# Communication-Efficient Distributed Learning: An Overview

Xuanyu Cao, *Senior Member, IEEE*, Tamer Başar, *Life Fellow, IEEE*, Suhas Diggavi, *Fellow, IEEE*, Yonina C. Eldar, *Fellow, IEEE*, Khaled B. Letaief, *Fellow, IEEE*, H. Vincent Poor, *Life Fellow, IEEE*, and Junshan Zhang, *Fellow, IEEE*

***Abstract*— Distributed learning is envisioned as the bedrock of next-generation intelligent networks, where intelligent agents, such as mobile devices, robots, and sensors, exchange information with each other or a parameter server to train machine learning models collaboratively without uploading raw data to a central entity for centralized processing. By utilizing the computation/communication capability of individual agents, the distributed learning paradigm can mitigate the burden at central processors and help preserve data privacy of users. Despite its promising applications, a downside of distributed learning is its need for iterative information exchange over wireless channels, which may lead to high communication overhead unaffordable in many practical systems with limited radio resources such as energy and bandwidth. To overcome this communication bottleneck, there is an urgent need for the development of communication-efficient distributed learning algorithms capable of reducing the communication cost and achieving satisfactory learning/optimization performance simultaneously. In this paper, we present a comprehensive survey of prevailing methodologies for communication-efficient distributed learning, including reduction of the number of communications, compression and quantization of the exchanged information, radio resource management for efficient learning, and game-theoretic mechanisms incentivizing user participation. We also point out potential directions for future research to further enhance the communication efficiency of distributed learning in various scenarios.**

***Index Terms*— Distributed learning, communication efficiency, event-triggering, quantization, compression, sparsification, resource allocation, incentive mechanisms, single-task learning, multitask learning, meta-learning, online learning.**

Manuscript received 13 April 2022; revised 10 August 2022; accepted 12 August 2022. Date of publication 6 February 2023; date of current version 17 March 2023. This work was supported in part by the U.S. National Science Foundation under Grant CNS-2128448, Grant 2007714, Grant 2139304, and Grant 2146838, in part by the U.S. Army Research Office under MURI Grant AG285, in part by the U.S. Army Research Laboratory Cooperative under Agreement W911NF-17-2-0196, and in part by the National Natural Science Foundation of China under Grant 62203373. *(Corresponding author: Xuanyu Cao.)*

Xuanyu Cao and Khaled B. Letaief are with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong (e-mail: eexcao@ust.hk; eekhaled@ust.hk).

Tamer Başar is with the Department of Electrical and Computer Engineering, University of Illinois Urbana–Champaign, Urbana–Champaign, IL 61801 USA (e-mail: basar1@illinois.edu).

Suhas Diggavi is with the Department of Electrical and Computer Engineering, The University of California, Los Angeles, CA 90095 USA (e-mail: suhas@ee.ucla.edu).

Yonina C. Eldar is with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 7610001, Israel (e-mail: yonina.eldar@weizmann.ac.il).

H. Vincent Poor is with the Department of Electrical and Computer Engineering, Princeton University, Princeton, NJ 08544 USA (e-mail: poor@princeton.edu).

Junshan Zhang is with the Department of Electrical and Computer Engineering, The University of California, Davis, CA 95616 USA (e-mail: jazh@ucdavis.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/JSAC.2023.3242710.

Digital Object Identifier 10.1109/JSAC.2023.3242710

## I. Introduction

MACHINE learning is one of the most important technologies to enable ubiquitous artificial intelligence (AI). In conventional centralized machine learning, all data is delivered from data owners to a central entity, which conducts centralized training and then sends the trained model to users of AI services. Such a centralized learning paradigm has several disadvantages. First, transmitting huge amount of raw data to a central processor can lead to significant traffic congestion and large communication delay. This renders centralized learning inappropriate for time-sensitive applications such as autonomous driving. Second, conducting the entire training procedure in a centralized manner causes substantial, if not prohibitive, computation burden for the central processor, and may lead to large computation latency. Third, the training data of individual users may contain private sensitive information (e.g., health data and financial data) and users with privacy concerns may not be willing to share their raw data with others.

To resolve the aforementioned issues, distributed learning has emerged as an alternative paradigm, where data owners train machine learning models collaboratively and distributively without uploading raw data to a central entity for centralized processing. In distributed learning, by utilizing their communication and computation resources, intelligent devices (e.g., smartphones) conduct local training steps by using local datasets and exchange information with other devices or a parameter server. Such a framework alleviates the computation and communication burden of centralized learning, and helps preserve data privacy of users. Due to its great potential, distributed learning has been extensively studied in the past decades and many distributed learning/optimization algorithms have been proposed for a variety of distributed learning settings (e.g., single-task learning, personalized learning, online learning, fully decentralized learning over networks, and more). Examples include distributed (sub)gradient descent

[1], distributed primal-dual method [2], alternating direction method of multipliers [3], distributed Newton's method [4], etc. The convergence performance of these distributed learning algorithms has been comprehensively analyzed for learning problems under various conditions (convexity, nonconvexity, strong convexity, smoothness, etc.).

Distributed learning algorithms require agents to exchange information with each other or a parameter server. The information often needs to be transmitted over wireless channels and may consume substantial amount of radio resources (e.g., energy and bandwidth), which are scarce in practice. For instance, mobile devices may have scarce energy due to their limited battery capacity, and can use very narrow bandwidth in communication systems located in densely populated urban regions. Sensors deployed in the wild may have little energy supply and are difficult to recharge when they run out of energy.

In conventional distributed learning algorithms, agents have to send high-dimensional *dense real-valued* vectors to other agents or the parameter server in *every* time slot, leading to high radio resource consumption. To cope with communication factors such as channel fading and noise, agents need to make the most of their limited radio resources to align well with the nature of distributed learning algorithms. Moreover, due to the substantial consumption of radio resources, agents are not well motivated to participate in distributed learning algorithms when they are deficient in resources. If not addressed adequately, the scarcity of wireless resources may greatly restrict the application of distributed learning in many practical scenarios.

To reduce the communication overhead of distributed learning algorithms, a variety of methods have been proposed in the literature. Methodologies for communication-efficient distributed learning can be divided into four categories. The first type of methods aim to reduce the number of communication rounds of distributed learning algorithms and require information exchange only when necessary [5]. The conditions for communications to occur are devised to balance the tradeoff between learning performance and communication overhead. Alternatively, the second type of methods seek to compress the information to be sent into finite number of bits or sparse vectors through data compression techniques such as quantization [6] and sparsification [7], [8]. The compression methods and learning algorithms are designed jointly to mitigate the negative impact of compressed communications on the learning performance. The third type of methods take practical wireless communication factors (noise, fading, interference, etc.) into consideration and aim to manage radio resources optimally for learning purposes (e.g., [9]). The goal is to achieve the best learning performance under radio resource budget constraints. The fourth type of works investigate the strategic behavior of agents in distributed learning [10]. Game-theoretic mechanisms are designed to incentivize agent participation in distributed learning algorithms, which consume agents' precious communication resources. There are works combining the aforementioned four types of techniques to further mitigate the communication overhead.

In this paper, we present a holistic overview of existing works on communication-efficient distributed learning. The organization of the paper is depicted in Fig. 1 and elucidated as follows.

- In Section II, we provide a brief overview of the basic problem formulations, algorithms, and convergence results of distributed learning, which is categorized into two scenarios, namely, distributed learning in the presence of a central parameter server and fully decentralized learning over networks without parameter servers. For both scenarios, we first consider single-task learning, where all agents seek to learn a common model. Then, we consider personalized learning (including multitask learning and meta-learning), where different agents aim to learn different (but related) models.
- In Section III, we survey communication-efficient distributed learning algorithms that reduce the number of communication rounds. Such algorithms may conduct multiple local update steps between consecutive communication rounds according to some pre-defined rules, or trigger communications only when certain conditions are met as the algorithms progress. We also provide an overview of results on characterizing the fundamental lower bounds for the number of communications needed to achieve certain learning performance guarantees. We then introduce several possible future research directions for reducing the number of communications in various distributed learning settings.
- In Section IV, we consider communication-efficient distributed learning algorithms using compressed communications to reduce redundant information transmission. These compression techniques include quantization, sparsification, error-compensated compression, as well as other methods exploiting special structures (e.g., low rank) of the exchanged information. Potential directions for future works on distributed learning with compressed communications are also mentioned.
- In Section V, we survey resource management techniques for distributed learning, which seek to achieve the best learning performance under radio resource budget constraints. We review results on both power allocation and bandwidth allocation, including their integration with other communication-efficient techniques such as user selection. We further point out some future research directions on this topic.
- In Section VI, we review several recent works on game-theoretic incentive mechanism design for encouraging user participation in distributed learning algorithms, which consume substantial amount of radio resources of users. Some potential future directions are also discussed.
- In Section VII, we conclude the paper.

## II. Preliminaries of Distributed Learning

In this section, we provide a brief overview of distributed learning, a research topic extensively studied over multiple decades. We categorize distributed learning settings based on the presence or absence of a central entity coordinating the learning processes. For both scenarios, we present the basic problem formulations, prevailing algorithms, and convergence results.

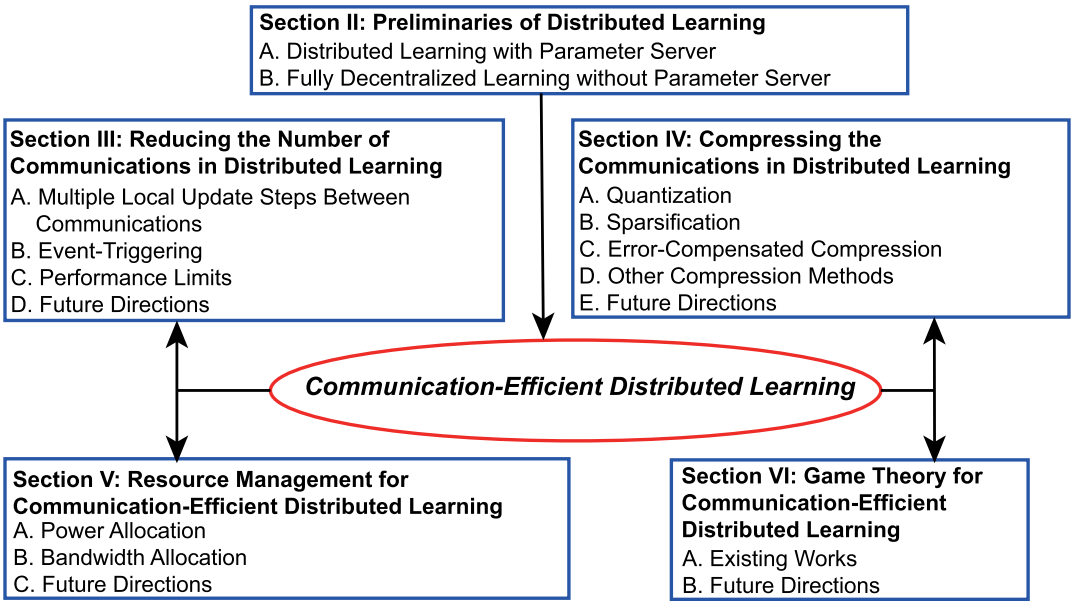
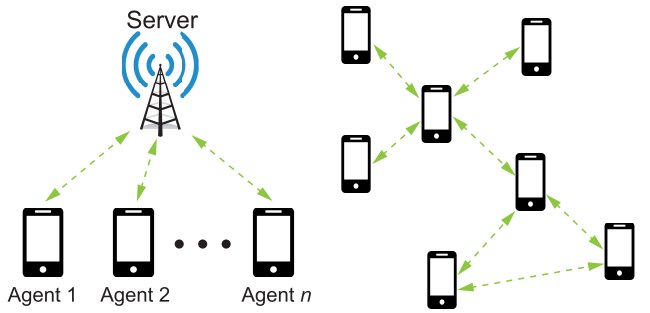



Fig. 1. Organization of the paper.




(a) Distributed learning in a server-agent system (b) Fully decentralized learning over a network

Fig. 2. Two multi-agent systems for distributed learning.

### A. Distributed Learning With Parameter Server

We first consider distributed learning over a system consisting of multiple agents and a central parameter server (abbreviated as server henceforth), as illustrated in Fig. 2-(a), where the server is able to exchange information with all agents. Such multi-agent systems are ubiquitous. For instance, in federated learning (FL) over cellular networks, the base station (server) can communicate with the mobile devices (agents) [11], [12], [13]. In sensor networks, the fusion center (server) can exchange information with the sensors (agents). In the following, we categorize distributed learning problems into two classes depending on whether the model parameters of the agents are the same or not.

*1) Single-Task Learning:* Let $L(\boldsymbol{x};\boldsymbol{u},d)$ be the loss function of the learning problem, where $\boldsymbol{x},\boldsymbol{u},d$ are the model parameter, input feature, and output value or label, respectively. For example, we have $L(\boldsymbol{x};\boldsymbol{u},d)=(\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x}-d)^2$ for linear regression, and $L(\boldsymbol{x};\boldsymbol{u},d)=\log(1+\exp(-d\cdot\boldsymbol{u}^{\mathsf{T}}\boldsymbol{x}))$ for logistic regression ($d=\pm 1$). The most standard and commonly used distributed learning setting is the single-task learning problem below

$$\min_{\boldsymbol{x}}\ f(\boldsymbol{x}):=\sum_{i=1}^{n} f_i(\boldsymbol{x}), \tag{1}$$

where $f_i(\boldsymbol{x})=\sum_{k\in\mathcal{S}_i} L(\boldsymbol{x};\boldsymbol{u}_{ik},d_{ik})$ is the local loss function of agent $i$, and $\{\boldsymbol{u}_{ik},d_{ik}\}_{k\in\mathcal{S}_i}$ is the training set of agent $i$. Problem (1) is referred to as empirical risk minimization or consensus optimization in the literature of distributed optimization. In such a single-task learning problem, agents aim to learn a common model $\boldsymbol{x}$ collaboratively based on all agents' training data. For instance, in sensor networks, sensors may seek to estimate the location of an object jointly by using every sensor's local measurements. In deep learning, to alleviate the computational burden of training, data may be distributed among multiple computers, which collaborate to train a common neural network in parallel.

Problem (1) has been studied for decades [14], and a variety of algorithms have been proposed. One of the most standard algorithms is gradient descent (GD). At each time $t$, the server broadcasts the current model $\boldsymbol{x}(t)$ to all agents. Each agent $i$ computes the local gradient $\nabla f_i(\boldsymbol{x}(t))$ by using local training data, and sends it to the server. The server then aggregates all the local gradients, and updates the model according to

$$\boldsymbol{x}(t+1)=\boldsymbol{x}(t)-\eta_t\sum_{i=1}^{n}\nabla f_i(\boldsymbol{x}(t)),$$

where $\eta_t>0$ is the stepsize. If each $f_i$ is convex and has Lipschitz continuous gradient with constant $L_i$, then a fixed stepsize $\eta_t=\eta\le\frac{1}{\sum_{i=1}^n L_i}$ will guarantee that the GD algorithm converges at rate $\mathcal{O}(1/t)$.

Another popular algorithm for solving problem (1) is the distributed alternating direction method of multipliers (ADMM). At each time $t$, each agent $i$ sends its current local model $\boldsymbol{x}_i(t)$ and local multiplier $\boldsymbol{\lambda}_i(t)$ to the server. The server broadcasts $\boldsymbol{z}(t+1)=\bar{\boldsymbol{x}}(t)+\frac{1}{\rho}\bar{\boldsymbol{\lambda}}(t)$ to all agents, where $\bar{\boldsymbol{x}}(t)=\frac{1}{n}\sum_{i=1}^n\boldsymbol{x}_i(t)$, $\bar{\boldsymbol{\lambda}}(t)=\frac{1}{n}\sum_{i=1}^n\boldsymbol{\lambda}_i(t)$, and $\rho>0$ is an algorithm parameter. Then, each agent $i$ updates its local model and multiplier in parallel as follows:

$$\boldsymbol{x}_i(t+1)=\arg\min_{\boldsymbol{x}_i}\left\{f_i(\boldsymbol{x}_i)+\frac{\rho}{2}\left\|\boldsymbol{x}_i+\frac{1}{\rho}\boldsymbol{\lambda}_i(t)\right.\right.$$

$$-\boldsymbol{z}(t+1)\|^2\Big\},$$
$$\boldsymbol{\lambda}_i(t+1)=\boldsymbol{\lambda}_i(t)+\rho(\boldsymbol{x}_i(t+1)-\boldsymbol{z}(t+1)).$$

When the loss functions $f_i$'s are strongly convex and have Lipschitz continuous gradients, distributed ADMM converges to the global optimal solution at a linear rate [15]. Many other optimization algorithms can also be used to solve single-task distributed learning problem (1), such as momentum acceleration methods (e.g., heavy ball and Nesterov's algorithms), and (quasi-)Newton's methods.

In some applications, the training data changes with time. Agents may collect new data in real time and discard outdated data. Correspondingly, the loss functions also vary across time. Such a scenario is referred to as online learning and has been investigated extensively [16], [17]. Let us denote the local loss function of agent $i$ at time $t$ by $f_{i,t}$ and let $f_t(\boldsymbol{x})=\sum_{i=1}^n f_{i,t}(\boldsymbol{x})$ be the global loss function at time $t$. Let $\{\boldsymbol{x}^*(t)\}$ be some performance benchmark, e.g., the dynamic optimal model $\boldsymbol{x}^*(t)=\arg\min_{\boldsymbol{x}\in\mathcal{X}} f_t(\boldsymbol{x})$ or the best fixed model $\boldsymbol{x}^*(t)=\boldsymbol{x}^*=\arg\min_{\boldsymbol{x}\in\mathcal{X}}\sum_{t=1}^T f_t(\boldsymbol{x})$, where $\mathcal{X}$ is the set of admissible model parameters and $T$ is the time horizon. Our goal is to determine a series of model parameters $\boldsymbol{x}(t)$ sequentially such that the *regret*, i.e., $\sum_{t=1}^T f_t(\boldsymbol{x}(t))-\sum_{t=1}^T f_t(\boldsymbol{x}^*(t))$, is minimized. In particular, if the regret is sublinear with respect to $T$, then the time-average loss incurred by the selected models $\boldsymbol{x}(t)$ is no greater than that of the benchmark $\boldsymbol{x}^*(t)$ asymptotically, as $T$ goes to infinity. One of the most standard online optimization algorithms is online gradient descent (OGD) [18], i.e.,

$$\boldsymbol{x}(t+1)=\mathcal{P}_{\mathcal{X}}\left(\boldsymbol{x}(t)-\eta_t\sum_{i=1}^n\nabla f_{i,t}(\boldsymbol{x}(t))\right), \tag{2}$$

where $\mathcal{P}_{\mathcal{X}}$ stands for projection onto $\mathcal{X}$. In the algorithm, the server broadcasts the current model $\boldsymbol{x}(t)$ to the agents and each agent $i$ sends the local gradient $\nabla f_{i,t}(\boldsymbol{x}(t))$ to the server. With the stepsize being $\eta_t=\frac{1}{\sqrt{t}}$, under certain technical assumptions, it has been shown that the regret of OGD is upper bounded by $\mathcal{O}(\sqrt{T})$ and is thus sublinear [18].

*2) Personalized Learning:* In practice, different agents may have different model parameters to learn, in which case problem (1) is not a suitable formulation. Such a scenario is referred to as personalized learning, where each agent has its own personal model to infer. One viable formulation for personalized learning is *multitask learning*, where each agent $i$ seeks to learn its own model $\boldsymbol{x}_i$. Even though the models of different agents are distinct, they are still related and we should take their relationship into account when formulating the learning problem. This leads to the following standard formulation for multitask learning [19]:

$$\min_{\boldsymbol{X},\boldsymbol{\Omega}}\ \sum_{i=1}^n f_i(\boldsymbol{x}_i)+\gamma\cdot\mathrm{tr}\left(\boldsymbol{X}\boldsymbol{\Omega}^{-1}\boldsymbol{X}^{\mathsf{T}}\right), \tag{3a}$$
$$\text{s.t. } \boldsymbol{\Omega}\succeq\boldsymbol{0},\ \mathrm{tr}(\boldsymbol{\Omega})=1, \tag{3b}$$

where $\boldsymbol{X}=[\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n]$, $\gamma>0$ is a regularization parameter, and $\mathrm{tr}(\cdot)$ stands for the trace of a matrix. The matrix $\boldsymbol{\Omega}$ characterizes the relationship between the models of different agents, and the regularization term $\mathrm{tr}\left(\boldsymbol{X}\boldsymbol{\Omega}^{-1}\boldsymbol{X}^{\mathsf{T}}\right)$ is used to promote such relationship in the learning outcome. In problem (3), we aim to learn both the models of all agents and the relationship between these models jointly. To this end, we can use alternating optimization methods [19], [20]. In other words, the agents first optimize over $\boldsymbol{X}$ with fixed $\boldsymbol{\Omega}$ in a parallel manner and send their local models to the server. Then, the server optimizes over $\boldsymbol{\Omega}$ with fixed $\boldsymbol{X}$, and broadcasts the new relationship $\boldsymbol{\Omega}$ to all agents. Under certain technical conditions, convergence of such alternating optimization methods to the globally optimal solution can be guaranteed [19].

In addition to multitask learning, another recently popular framework for personalized learning is *meta-learning* initiated in [21] and [22]. In meta-learning, agents collaborate to learn a common *meta-model*. Starting from the meta-model, an agent can adapt to new tasks readily by using very limited local data and simple training iterations, e.g., a few gradient descent steps. The most standard form of meta-learning can be cast as the following optimization problem:

$$\min_{\boldsymbol{x}}\ \sum_{i=1}^n f_i(\boldsymbol{x}-\alpha\nabla f_i(\boldsymbol{x})), \tag{4}$$

where $\alpha>0$ is the stepsize for local adaptation, i.e., one-step gradient descent. Let $F_i(\boldsymbol{x}):=f_i(\boldsymbol{x}-\alpha\nabla f_i(\boldsymbol{x}))$ be the meta-function of agent $i$. To solve problem (4), we can still use GD algorithm, i.e., $\boldsymbol{x}(t+1)=\boldsymbol{x}(t)-\eta_t\sum_{i=1}^n\nabla F_i(\boldsymbol{x}(t))$, where each agent $i$ sends $\nabla F_i(\boldsymbol{x}(t))=(\boldsymbol{I}-\alpha\nabla^2 f_i(\boldsymbol{x}(t)))\nabla f_i(\boldsymbol{x}(t)-\alpha\nabla f_i(\boldsymbol{x}(t)))$ to the server at each time $t$. Various distributed personalized learning algorithms have been studied in [23], [24], [25], and [26] from the viewpoint of distributed optimization.

### B. Fully Decentralized Learning Without Parameter Server

Many multi-agent systems do not have any central entity capable of communicating with all agents. Instead, the agents form a network, where two agents linked by an edge are able to exchange information with each other, as illustrated in Fig. 2-(b). For instance, large-scale sensor networks may not have fusion centers, and sensors can only communicate with other nearby sensors. In ad hoc networks without base stations (e.g., battlefield networks without communication infrastructure), mobile devices can only communicate with other nearby devices. In the absence of central servers, the learning algorithms have to be fully decentralized and only communications between one-hop neighbors are allowed. In the following, we discuss fully decentralized single-task learning and multitask learning over multi-agent networks without central entities.

*1) Single-Task Learning:* One of the most prevailing fully decentralized optimization algorithms for solving the single-task learning problem (1) is the decentralized gradient descent (DGD) method proposed by [1]. Let $\mathcal{N}_i$ be the set of neighbors of agent $i$. In DGD, each agent $i$ updates its local model $\boldsymbol{x}_i(t)$ by using a convex combination of its neighbors' local models, followed by a local gradient descent step, i.e.,

$$\boldsymbol{x}_i(t+1)=\sum_{j\in N_i\cup\{i\}} a_{ij}\boldsymbol{x}_j(t)-\eta_t\nabla f_i(\boldsymbol{x}_i(t)),$$

where $a_{ij}$ is the $(i,j)$-th entry of a doubly stochastic weight matrix $\boldsymbol{A}$. We have $a_{ij}=0$ for $j\notin\mathcal{N}_i\cup\{i\}$, so that each agent only communicates with its neighbors. It has been shown in [1] that, if a constant stepsize $\eta_t$ is used, all local models converge to a neighborhood of the optimal solution to (1) with rate $\mathcal{O}(1/t)$. If diminishing stepsizes are used, the DGD algorithm can converge to the exact optimal solution with rate $\mathcal{O}(1/\sqrt{t})$. Since the seminal work [1], a variety of first-order decentralized optimization algorithms have been developed to solve the consensus optimization problem (1) in various settings, including constrained decentralized optimization in [27], decentralized optimization over time-varying networks in [28], decentralized optimization over directed networks (e.g., the push-pull algorithm, in [29]), and decentralized optimization over time-varying directed networks (e.g, the push-subgradient algorithm, in [30]). Additionally, by using gradient information of the last two steps, the EXTRA algorithm proposed in [31] can converge to the exact optimal solution with a constant stepsize. To accelerate the convergence rate, distributed Nesterov gradient descent algorithm was developed in [32]. Distributed zero-order algorithms with gradient tracking were studied in [33], where one could only evaluate the objective functions at finitely many points. Further, second-order decentralized optimization algorithms have also been studied, such as decentralized Newton's method with truncated approximation of inverse Hessian matrices in [4], and decentralized BFGS algorithm (a quasi-Newton method using gradient information to approximate Newton steps) [34].

In addition to the aforementioned primal-domain methods, primal-dual algorithms have also been developed for decentralized optimization problems. One of the most widely used primal-dual methods for solving problem (1) is the decentralized ADMM, in which each agent $i$ updates its local model $\boldsymbol{x}_i(t)$ (i.e., primal variable) and multiplier $\boldsymbol{\phi}_i(t)$ (i.e., dual variable) as follows:

$$\boldsymbol{x}_i(t+1)=\arg\min_{\boldsymbol{x}_i}\Bigg\{f_i(\boldsymbol{x}_i)+\boldsymbol{\phi}_i(t)^{\mathsf{T}}\boldsymbol{x}_i+\rho|\mathcal{N}_i|\|\boldsymbol{x}_i\|^2-\rho\left(|\mathcal{N}_i|\boldsymbol{x}_i(t)+\sum_{j\in\mathcal{N}_i}\boldsymbol{x}_j(t)\right)^{\mathsf{T}}\boldsymbol{x}_i\Bigg\}, \tag{5a}$$

$$\boldsymbol{\phi}_i(t+1)=\boldsymbol{\phi}_i(t)+\rho\left(|\mathcal{N}_i|\boldsymbol{x}_i(t+1)-\sum_{j\in\mathcal{N}_i}\boldsymbol{x}_j(t+1)\right), \tag{5b}$$

where $|\cdot|$ stands for the cardinality of a set. At each time $t$, each agent $i$ needs to broadcast its current local model $\boldsymbol{x}_i(t)$ to all the neighbors in $\mathcal{N}_i$. When the loss functions are strongly convex and have Lipschitz continuous gradients, it has been shown in [3] that decentralized ADMM has linear convergence rate. Following [3], a series of variants of decentralized ADMM have been developed. To reduce the computational burden and avoid solving optimization subproblems in each iteration, linearized ADMM and quadratically approximated ADMM have been proposed in [35] and [36], which use linear and quadratic approximations for $f_i(\boldsymbol{x}_i)$ in step (5a) to obtain closed-form update equations.

When the training data is collected in real-time and the loss functions are time-varying, decentralized online optimization problems have been studied. A decentralized online gradient descent algorithm was developed in [37], where the regret of every agent was upper bounded by $\mathcal{O}(\sqrt{T})$. A decentralized online saddle-point algorithm was proposed in [38], and a decentralized online push-sum algorithm was developed for directed graphs in [39]. Moreover, dynamic decentralized ADMM was studied in [40] and was shown to converge to a neighborhood of the dynamic optimal solution, where the size of the neighborhood depended on the variation speed of the loss functions.

*2) Multitask Learning:* In addition to the single-task problem (1), decentralized multitask learning problems over multi-agent networks without any central entity have also been studied in the literature. In such a case, each agent $i$ has an individual model $\boldsymbol{x}_i$ to learn, and the local models of different agents are related. One of the most common methods of characterizing this relationship is to introduce a link cost $g_{ij}(\boldsymbol{x}_i,\boldsymbol{x}_j)$ for each pair of neighboring agents $i$ and $j$. For instance, $g_{ij}(\boldsymbol{x}_i,\boldsymbol{x}_j)=\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2-b_{ij}^2$ can be used to make neighbors' models close to each other, where $b_{ij}$ is some constant. The link costs are either added to the objective function of the learning problem, i.e.,

$$\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n}\ \sum_{i=1}^n f_i(\boldsymbol{x}_i)+\sum_{i=1}^n\sum_{j\in\mathcal{N}_i}g_{ij}(\boldsymbol{x}_i,\boldsymbol{x}_j), \tag{6}$$

or used as constraints of the learning problem, i.e.,

$$\min_{\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n}\ \sum_{i=1}^n f_i(\boldsymbol{x}_i) \tag{7a}$$

$$\text{s.t. } g_{ij}(\boldsymbol{x}_i,\boldsymbol{x}_j)\le 0,\quad \forall i,j\in\mathcal{N}_i. \tag{7b}$$

For problem (6), decentralized linearized ADMM and decentralized Newton's method were proposed in [41] and [42], respectively, both of which could achieve linear convergence rate. For problem (7), a primal-dual optimization method was developed in [43] to handle the constraints, and convergence rate for the objective and constraint functions were shown to be $\mathcal{O}(t^{-1/2})$ and $\mathcal{O}(t^{-1/4})$, respectively.

Furthermore, when the training data is collected sequentially and the loss functions vary across time, decentralized multitask adaptive learning algorithms have been proposed in [44] and [45], where agents are clustered and neighboring clusters have similar models. When the model parameters are sparse, ADMM-based and subgradient-based decentralized multitask adaptive learning algorithms have been developed in [46].

## III. Reducing the Number of Communications in Distributed Learning

Conventional distributed learning algorithms require agents to exchange information with the server or neighboring agents in every time instant, which can lead to a high communication overhead. In this section, we provide an overview of communication-efficient distributed learning algorithms that

reduce the number of communications, and point out several potential directions for future work.

### A. Multiple Local Update Steps Between Communications

One of the most commonly used approaches for improving the communication efficiency of distributed learning is to exchange information periodically instead of at every time instant. Between consecutive communications, agents conduct multiple steps of local model updates based on local data. In [5], such a method for distributed learning in a server-agent system has been investigated. Let $\tau \in \{1, 2, \ldots\}$ be the number of local model update steps between two consecutive global aggregations, i.e., communications between the server and the agents. When the time index $t$ is not an integer multiple of $\tau$, each agent $i$ conducts a local gradient descent step to update the local model $\boldsymbol{x}_i(t)$, i.e.,

$$\boldsymbol{x}_i(t) = \boldsymbol{x}_i(t-1) - \eta \nabla f_i(\boldsymbol{x}_i(t-1)).$$

Otherwise, when $t$ is an integer multiple of $\tau$, each agent $i$ sends $\boldsymbol{x}_i(t-1) - \eta \nabla f_i(\boldsymbol{x}_i(t-1))$ to the server. The server aggregates the information from all agents to obtain

$$\widetilde{\boldsymbol{x}}(t) = \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i(t-1) - \eta \nabla f_i(\boldsymbol{x}_i(t-1))).$$

Then, the server broadcasts $\widetilde{\boldsymbol{x}}(t)$ to all agents and each agent $i$ updates its new local model to be $\boldsymbol{x}_i(t) = \widetilde{\boldsymbol{x}}(t)$. In such an algorithm, global aggregations occur once every $\tau$ time instants.

Suppose we are concerned with $M$ types of radio resources, e.g., energy and bandwidth, and the budget for type-$m$ resource is $R_m$, $m = 1, \ldots, M$. When models are updated locally without information exchange (i.e., $t$ is not an integer multiple of $\tau$), the multi-agent system consumes $c_m$ amount of type-$m$ resource. If, in addition to local model update steps, global aggregation happens and information exchange between the server and the agents is needed, the system consumes $b_m$ amount of type-$m$ resource. We usually have $b_m > c_m$ since global aggregation consumes additional resources. Let $T$ be the number of time instants of the algorithm and $K = T/\tau$ be the number of global aggregations. When global aggregation occurs, the server sets $\widetilde{f} \leftarrow \min\{f(\widetilde{\boldsymbol{x}}(t)), \widetilde{f}\}$ so that $\widetilde{f}$ records the best loss function values at time $t = 0, \tau, 2\tau, \ldots$. Our goal is to achieve the best loss function values subject to the resource constraints, i.e.,

$$\min_{\tau, K \in \{1,2,\ldots\}} \min_{k=0,\ldots,K} f(\widetilde{\boldsymbol{x}}(k\tau)) \tag{8a}$$

$$\text{s.t. } (T+1)c_m + (K+1)b_m \leq R_m, \quad \forall m = 1, \ldots, M, \tag{8b}$$

$$T = \tau K, \tag{8c}$$

where the additional "+1" in (8b) accounts for the last global aggregation. In [5], an algorithm for solving problem (8) approximately has been proposed when the resource consumptions $\{c_m, b_m\}$ are known in advance. When $\{c_m, b_m\}$ are unknown and can vary with time, a control algorithm estimating the parameters and adjusting the values of $\tau$ on-the-fly has been developed.

Similarly, the federated averaging (FedAvg) algorithm in [47] lets each agent conduct multiple steps of local model updates between two global aggregations in distributed learning of deep networks. In addition, FedAvg selects a dynamic subset of agents, instead of all agents, to participate in model updating and global aggregation, which further improves the communication efficiency. It was shown in [47] through extensive numerical experiments that FedAvg could reduce the communication overhead by one to two orders of magnitudes. Further, in [48], the authors studied rigorously the reason why periodic model averaging (i.e., global aggregation) could work as well as parallel mini-batch SGD (with global aggregation in every time instant) and achieve linear speedup with respect to the number of agents. In particular, it was shown that the dominant term in the convergence bound for distributed learning with periodic model averaging was $\mathcal{O}(1/\sqrt{nt})$, which was not affected by the model-averaging period. Further, in [49], the convergence rate of local stochastic gradient descent (SGD) was analyzed, where global aggregation occurred only at certain time instants. For smooth strongly convex learning problems, it was shown that local SGD converged at the same rate as standard mini-batch SGD did. By using local SGD, the number of communication rounds could be reduced by a factor of $\mathcal{O}(\sqrt{T})$, where $T$ is the total number of update steps. Additionally, post-local SGD, i.e., a mixture of mini-batch SGD and local SGD, was proposed in [50], and was shown to achieve better tradeoff between communication efficiency and generalized performance for deep learning. The convergence rate of local SGD with periodic averaging was further analyzed in [51] for nonconvex loss functions satisfying the Polyak-Łojasiewicz condition. It was shown that $\mathcal{O}((nT)^{\frac{1}{3}})$ rounds of communications suffice to achieve a convergence rate of $\mathcal{O}(1/nT)$, which maintained linear speedup with respect to the number of agents. Further, the number of local model updates per round of global aggregation was adjusted in an adaptive manner in [52], so that the runtime of the distributed learning algorithm was minimized when communication delay existed. FL with heterogeneous number of local updates among agents was studied in [53]. The authors developed a novel FL algorithm to compensate for the heterogeneity caused by agents' different computation speeds and dataset sizes. Additionally, to improve the convergence rate of local SGD, a slow momentum algorithm was proposed in [54], where agents performed local momentum model update and synchronized periodically through global aggregations. A comprehensive comparison between local SGD and mini-batch SGD was presented in [55], and it was shown that the two algorithms could outperform each other in certain regimes.

In addition to local SGD, SGD with elastic averaging was proposed in [56], where proximal terms were included in the loss functions to allow some slacks between the local models at the agents and the global model at the server. The approach was shown to have better learning performance in the deep learning setting where many local minima existed. Momentum versions of elastic averaging SGD were also developed in [56]. Further, cooperative SGD, a unified framework for a variety of local SGD algorithms (e.g., local SGD with averaging, elastic averaging SGD, and decentralized local

SGD over networks without central server), was proposed and analyzed in [57], which improved upon prior results on local SGD in terms of convergence bounds and applicability. A new decentralized primal-dual algorithm named decentralized communication sliding method was developed in [58] for networked multi-agent networks without central entities, where inter-agent communications were skipped while individual agents solved local optimization subproblems iteratively. In [59], the authors investigated semi-decentralized FL over a clustered network, which consisted of a server and multiple clusters of agents. Each cluster was comprised of a cluster head and multiple normal agents. Within each cluster, agents performed multiple SGD iterations based on local datasets and aperiodically engaged in consensus procedures within the cluster by using fully decentralized device-to-device (D2D) communications. Meanwhile, the cluster heads conducted inter-cluster model aggregation through the help of the central server. Within such a framework, an adaptive control algorithm was developed in [59] to tune the stepsize, D2D communication rounds, and global aggregation periods, with the goal of minimizing the overall system loss due to energy consumption, delay, and FL performance. Moreover, in [60], a hierarchical FL framework was presented, where clients, edge servers and cloud server exchanged information with each other to learn collaboratively.

Different from GD, a communication-efficient dual coordinate ascent algorithm was put forth in [61], where local computation was used in a primal-dual method to reduce the communication overhead dramatically. A communication-efficient federated deep learning method was proposed in [62], where parameters of the deep layers were updated less frequently than those of the shallow layers to reduce the communication overhead. A temporally weighted aggregation strategy was introduced at the server to make use of the previously trained local models of the agents. Besides single-task learning problem (1), meta-learning (c.f. problem (4)) algorithms with reduced number of communications have also been studied to facilitate communication-efficient personalized learning. In [23], a personalized FedAvg algorithm was proposed for distributed meta-learning problems, where a subset of agents conducted multiple local gradient descent steps with respect to their local meta-functions and global aggregation was performed periodically. For nonconvex loss functions, the convergence rate (to a first-order stationary point) of the algorithm was analyzed, and the impact of the closeness of the underlying distributions of agents' data (measured in terms of total variation and Wasserstein distance) on the learning performance was characterized.

### B. Event-Triggering

The communication patterns of distributed learning algorithms in the aforementioned works in the previous subsection follow some predefined rules independent from the algorithm iterates, e.g., periodic global aggregation with a predefined period. Another generic approach to reducing the number of communications is to exchange information only when certain conditions related to the algorithm iterates are met during algorithm execution. Such an approach is named *event-triggering*, where communications occur only when a certain event is triggered. The triggering event can be devised so that the information is exchanged only when necessary. This can potentially reduce the communication cost without degrading the learning performance much.

We use the event-triggered projected DGD algorithm for problem (1) in [63] as a concrete example to illustrate the event-triggering approach. Consider a fully decentralized network without central entities. Each agent $i$ sends its local model $\boldsymbol{x}_i(t)$ to its neighbors only when certain conditions are met. In addition to $\boldsymbol{x}_i(t)$, each agent $i$ maintains another variable $\widetilde{\boldsymbol{x}}_i(t)$, which stands for the latest sent local model up to time $t$. Thus, at time $t$, agent $i$ has access to $\boldsymbol{x}_i(t), \widetilde{\boldsymbol{x}}_i(t), \{\widetilde{\boldsymbol{x}}_j(t)\}_{j\in\mathcal{N}_i}$. Then, agent $i$ updates its local model as follows:

$$\boldsymbol{x}_i(t+1) = \mathcal{P}_{\mathcal{X}}\left(\boldsymbol{x}_i(t) + \sum_{j\in\mathcal{N}_i} a_{ij}\left(\widetilde{\boldsymbol{x}}_j(t) - \widetilde{\boldsymbol{x}}_i(t)\right) - \eta\nabla f_i(\boldsymbol{x}_i(t))\right),$$

where $\mathcal{X}$ is a common constraint set for the local models, and $\boldsymbol{A} = [a_{ij}]$ is a symmetric doubly stochastic weight matrix. The triggering event for communications depends on the gap between the new local model $\boldsymbol{x}_i(t+1)$ and the latest sent local model $\widetilde{\boldsymbol{x}}_i(t)$. Let $C_i(t)$ be the triggering threshold of agent $i$ at time $t$. If $\|\boldsymbol{x}_i(t+1)-\widetilde{\boldsymbol{x}}_i(t)\| \geq C_i(t)$, agent $i$ sends $\boldsymbol{x}_i(t+1)$ to all neighbors and sets $\widetilde{\boldsymbol{x}}_i(t+1) = \boldsymbol{x}_i(t+1)$. Otherwise, agent $i$ does not send anything and sets $\widetilde{\boldsymbol{x}}_i(t+1) = \widetilde{\boldsymbol{x}}_i(t)$. In other words, agents communicate with neighbors only when the differences between the latest sent models and the current true models are large enough. The impact of the event-triggering thresholds $\{C_i(t)\}$ on the performance of the decentralized learning algorithm was analyzed in [63]. Convergence could be guaranteed as long as the event-triggering thresholds are square-summable. If the loss functions are strongly convex and the event-triggering thresholds are geometrically decaying, the local models converge to some neighborhood of the optimal solution with linear convergence rate, where the size of the neighborhood is proportional to the constant stepsize $\eta$.

In [64], the authors proposed an event-triggered multi-agent optimization algorithm over a complete network, where each agent was able to communicate with all other agents. Each agent sent its current local model to others when it detected that other agents' estimates of its local model were sufficiently different from the true local model. Later, an edge-based event-triggered projected DGD algorithm over fully decentralized networks was developed in [65], where an agent sent its current local model to one of its neighbors only when the difference between the current model and the latest sent one was larger than an edge-specific threshold. With diminishing stepsizes and event-triggering thresholds, the convergence of the algorithm was analyzed for convex loss functions and the impact of the triggering thresholds on the convergence rate was investigated. The convergence rate of event-triggered decentralized SGD was further analyzed in [66] for nonconvex loss functions, in the presence of diminishing stepsizes and triggering thresholds. Moreover, a continuous-time decentralized

event-triggered DGD algorithm was proposed in [67], which was independent of the parameters of the loss functions and free of Zeno behavior (i.e., not requiring infinite number of communications within a finite period of time).

A decentralized event-triggered continuous-time zero-gradient-sum algorithm was proposed in [68], where the triggering condition depended on the distance between the latest sent local model and the current local model as well as the consensus gap between the neighboring agents' models. The algorithm was shown to be free of Zeno behavior. In particular, the inter-communication time was lower bounded by some positive constant. For strongly convex loss functions, exponential convergence rate of the algorithm to the optimal solution was established. Moreover, event-triggered decentralized zero-gradient-sum algorithms over directed networks were proposed in [69], where both continuous-time and discrete-time algorithms were considered. Further, in [70], the event-triggering approach was applied to a more general distributed optimization problem with affine constraints, which encompassed the distributed learning problem (1) and the network utility maximization problem as special cases. An event-triggered augmented Lagrangian method was put forth, where the triggering condition was related to the primal gradient of the augmented Lagrangian. In addition, a decentralized event-triggered gradient tracking algorithm was proposed in [71], where linear convergence to the optimal solution was established by using sporadic communications. A decentralized event-triggered gradient-push algorithm over directed networks was developed in [72], and the convergence of the algorithm was established under summable stepsizes and triggering thresholds. Moreover, a decentralized event-triggered coordinate descent algorithm was studied in [73]. An event-triggered (a.k.a. communication-censored) decentralized ADMM algorithm was developed in [74]. It was shown that the censored ADMM converged to the optimal solution if the loss functions were convex and the event triggering thresholds were summable. If the loss functions were strongly convex and the triggering thresholds were decaying geometrically, then the censored ADMM exhibited linear convergence rate. Further, when the loss functions were time-varying, an event-triggered decentralized online subgradient method was developed in [75], where the impact of the triggering thresholds on the regret of each agent was characterized explicitly.

The integration of event-triggering and quantization was considered in [76], which proposed a continuous-time event-triggered DGD algorithm with dynamic quantization. The dynamic quantization scheme consisted of a dynamic encoder for the transmitting agent and a dynamic decoder for the receiving agent. The scheme quantized the difference of the latest sent local model and current local model with increasing accuracy, which made use of the convergence effect of the algorithm. It was shown in [76] that the algorithm could converge to the optimal solution without encountering Zeno behavior. Analogously, a discrete-time event-triggered quantized DGD algorithm was developed in [77] for time-varying directed graphs, where the dynamic quantization scheme still included dynamic encoding and decoding methods with finite number of quantization levels. It was shown that the algorithm could converge to the optimal solution even with one-bit information exchange in each time instant with triggered event, and the convergence rate was $\mathcal{O}(\log t/\sqrt{t})$ for convex loss functions.

An event-triggered distributed learning algorithm termed lazily aggregated gradient (LAG) for server-agent systems was developed in [78]. In LAG, each agent sent the difference of the current local gradient and the last sent local gradient to the server when this difference was larger than some threshold related to the weighted temporal variation of the global model. Meanwhile, the server sent the current model to an agent only when the difference between the local model of the agent and the global model of the server was larger than some threshold pertaining to the temporal variation of the global model. In other words, LAG conducted event-triggering for both the downlink and uplink communications between the server and the agents. It was shown in [78] that LAG exhibited linear convergence rate and $\mathcal{O}(1/t)$ convergence rate for the scenarios of strongly convex loss functions and convex loss functions, respectively. When the loss functions were nonconvex, LAG converged to a first-order stationary point with rate $\mathcal{O}(1/\sqrt{t})$. In addition, LAG with quantized gradients was put forth in [79], which saved both the number of communication rounds and the number of bits per communication round. For strongly convex loss functions, such an algorithm was shown to have the same linear convergence rate as standard GD did. The LAG algorithm was further extended to policy gradient (PG) method for reinforcement learning (RL) in [80]. It was shown that the LAG approach could achieve the same convergence rate as vanilla PG method did, and the number of communications could be significantly reduced, especially when the reward functions of the agents were sufficiently heterogeneous. Other approaches to reducing the number of communications include dynamically increasing batch sizes in parallel SGD to achieve the best tradeoff between communication and computation (measured by the number of stochastic gradients called) [81], and properly infusing redundancy to the training data for distributed SGD [82].

### C. Performance Limits

Several papers have investigated the fundamental lower bounds on the number of communications to achieve certain learning performance guarantees.

The communication complexity of distributed convex optimization was investigated in [83]. The paper considered a simple setting where each of two processors had access to a different convex function $f_i$, $i = 1, 2$. The two processors exchanged binary information with each other until they found a point minimizing $f_1(\boldsymbol{x})+f_2(\boldsymbol{x})$ (corresponding to single-task learning problem (1) with two agents) within some error $\epsilon$. It was shown in [83] that the minimal number of communication rounds to achieve this goal was $\Omega(d\log(1/\epsilon))$, where $d$ was the dimension of the decision variable (i.e., the model parameter in the context of learning). In [84], the authors

studied lower bounds for the number of communication rounds needed to solve distributed learning problems over complete networks, where each agent was capable of broadcasting to everyone. They identified cases where existing distributed learning algorithms were worst-case optimal, as well as scenarios where improvements were possible. They showed that, if the loss functions of different agents were not similar, a large number of communications was necessary even when agents had infinite computation power. Lower bounds for the communication complexity of solving distributed linear systems and linear programming were studied in [85]. Further, the minimax communication complexity of distributed convex stochastic optimization problems was examined in [86], where every agent had access to the stochastic gradients of a common objective function. Lower bounds on the number of communications and the corresponding optimal algorithm with matching upper bounds (up to logarithmic factors) were presented. In addition, information-theoretic lower bounds on the query complexity of stochastic convex optimization were investigated in [87] and [88].

### D. Future Directions

Several potential directions for future work in this domain are listed below.

*1) Reducing the Number of Communications for Distributed Online Learning:* There are relatively few works on reducing the number of communications for distributed online learning. In [75], an event-triggered distributed online subgradient method was developed to reduce the number of communications for distributed online learning. Nevertheless, reference [75] did not quantify the communication overhead of the algorithm explicitly and did not study the optimal tradeoff between learning performance and communication cost. Moreover, [75] was focused on online single-task learning and did not take other forms of learning problems into consideration. Many aspects of distributed online learning with reduced number of communications are yet to be explored.

For distributed online learning in server-agent systems, the conventional approach is to use the OGD algorithm (2), where global aggregation occurs at every time instant. Alternatively, we can let each agent conduct local GD update steps for $\tau$ time instants based on the local training data collected during this time period (i.e., the local loss functions during this period). Every $\tau$ time instants, each agent sends the difference of the local model, i.e., $\boldsymbol{x}_i(t+\tau) - \boldsymbol{x}_i(t)$, to the server. The server aggregates all the local models and computes the new global model, which is broadcast to all agents. Suppose $K$ rounds of global aggregation occur during the execution of the algorithm, i.e., $T = K\tau$ time instants in total. We can then characterize the relation between the parameters $\tau, K$ and the regret of the online learning algorithm through regret analysis. Under given communication resource budgets, one can seek to obtain the optimal $\tau, K$ yielding the minimal time-average regret. It is also possible to extend this framework to other distributed online learning problems, such as those with constraints not amenable to computationally efficient projection operators. These problems can be handled through primal-dual methods using Lagrangian (c.f. [2]), and we can study the optimal tradeoff between the communication overhead and learning performance as measured by regret and constraint violations.

In addition, it is possible to revisit the event-triggered decentralized online optimization problem over fully decentralized networks without any central server. In [75], the relation between event-triggering thresholds and the regret of each agent has been characterized. One can further study the relation between event-triggering thresholds and the communication overhead, based on which the optimal triggering thresholds can be designed to achieve the best regret under given communication budget.

*2) Reducing the Number of Communications for Distributed Personalized Learning:* Most of prior works on distributed learning algorithms with reduced number of communications are focused on single-task learning problems. Two exceptions are [23] and [25], where infrequent communications are conducted to alleviate the communication overhead for solving personalized learning problems. One can further develop an event-triggered approach to distributed meta-learning so that the number of local updates per communication round is not a fixed number and can vary according to the needs of the algorithm based on triggering rules. It would also be possible to consider distributed online meta-learning algorithms with reduced number of communications, when the training data is collected in real time.

Additionally, one can study distributed multitask learning algorithms for solving problems (3) (for server-agent systems), and (6), (7) (for fully decentralized networks without central servers) by using reduced number of communications. It would be possible to examine the relations between the communication patterns (e.g., number of local updates per communication round or event-triggering rules) and the learning performance, and design the best algorithms to achieve the best tradeoff between communication cost and learning performance.

*3) Performance Limits for Generic Distributed Learning Problems:* Even though some prior works have studied the fundamental performance tradeoff between the number of communications and the learning performance, they are only concerned with specific scenarios of distributed learning, e.g., linear programming, and learning over complete graphs. We still lack a clear understanding of the fundamental tradeoff between learning performance and number of communications in the general distributed learning setting. One can start from the most basic setting, namely the static single-task distributed learning problem (1), and determine lower bounds on the number of communications needed to arrive at an $\epsilon$-suboptimal model. It would be interesting to see if standard algorithms (e.g., GD and ADMM) or their variants can achieve the best communication complexity. If not, one can seek to design such optimal (in order sense) algorithms achieving the best learning performance with limited communication budget. Afterwards, it would be possible to extend the framework to more complicated scenarios, such as distributed online learning and distributed personalized learning.

## IV. Compressing the Communications in Distributed Learning

In addition to reducing the number of communication rounds, another general approach to improving the communication efficiency of distributed learning algorithms is to compress the information exchanged in each communication round. In this section, an overview of distributed learning algorithms using compressed communications is presented.

### A. Quantization

One of the most widely used compression methods for distributed learning is quantization, where the information to be exchanged is transformed into discrete values that can be encoded into a finite number of bits. Quantization techniques can reduce the number of communicated bits, and thus enable distributed learning in systems with scarce communication bandwidth, such as crowded Metropolitan areas with scarce spectrum resources. The study of quantized incremental distributed learning algorithms was pioneered in [6], which aimed at solving the single-task learning problem (1) by using a finite number of bits per communication. In the incremental algorithm, all agents were numbered (labeled) in advance and took turns to update the model parameters according to the order prescribed by the labeling. The model updates were cycled through the network. Let $\Lambda \subset \mathbb{R}^d$ be a $d$-dimensional lattice, where each entry of $\boldsymbol{x} \in \Lambda$ is an integer multiple of some given $\delta > 0$ (the width of the lattice). Denote the set of possible model parameters by $\mathcal{X}$. At each cycle $k$, an agent $i$ receives the model $\boldsymbol{x}_{i-1,k}$ from its predecessor, agent $i-1$, and computes the new model by a quantized gradient descent step as follows:

$$\boldsymbol{x}_{i,k} = Q(\boldsymbol{x}_{i-1,k} - \eta \nabla f_i(\boldsymbol{x}_{i-1,k})),$$

where the quantizer $Q$ is the projection operator associated with the set $\mathcal{X} \cap \Lambda$. Then, agent $i$ sends the new model $\boldsymbol{x}_{i,k}$ to its successor, agent $i+1$. After one completed cycle of model updates, we set $\boldsymbol{x}_{0,k+1} = \boldsymbol{x}_k = \boldsymbol{x}_{n,k}$ and begin the next cycle. It has been shown in [6] that, as $k$ goes to infinity, the gap between $f(\boldsymbol{x}_k)$ and the optimal value is upper bounded by some number pertaining to the quantization resolution $\delta$. The smaller $\delta$ is, the better the learning performance becomes (i.e., the smaller $f(\boldsymbol{x}_k)$ becomes).

The approach in [6] required the network to maintain an ordering of the agents and was not fully decentralized. Alternatively, a fully decentralized quantized DGD algorithm was proposed in [89], where each agent $i$ updated its local model in each time slot $t$ as follows:

$$\boldsymbol{x}_i(t+1) = Q\left(\sum_{j \in N_i \cup \{i\}} a_{ij} \boldsymbol{x}_j(t) - \eta_t \nabla f_i(\boldsymbol{x}_i(t))\right),$$

so that each agent only needed to transmit a quantized local model to its neighbors. The impact of the number of quantization levels on the convergence rate was investigated. To mitigate the negative effect of quantization on the learning performance, a universal vector quantization scheme was put forth in [90] for FL over rate-constrained wireless channels in a server-agent system. It was shown that the distortion due to quantization vanishes as the number of agents increases. Moreover, a distributed dual-averaging method using quantized communications was developed in [91]. When deterministic quantizers were used, the algorithm converged to a suboptimal point, where the suboptimality depended on the quantization resolution. When probabilistic quantizers were used, the algorithm converged to the optimal solution in expectation, and the impact of quantization resolution on the convergence rate was investigated. Analogously, quantized ADMM algorithm was studied in [92] by using deterministic and probabilistic quantizers, and the effect of quantization accuracy on the learning performance was characterized. In addition, a variant of DGD using multiple quantized consensus communication steps per local gradient descent was proposed in [93] to allow more flexible tradeoff between communication and computational costs.

A compression scheme named quantized SGD (QSGD) was proposed in [94] to allow for a smooth tradeoff between communication bandwidth and convergence time of the learning algorithms. QSGD enjoyed guaranteed convergence for both convex and nonconvex loss functions, and could be equipped with stochastic variance-reduction techniques to further accelerate convergence. Another method of achieving convergence to the exact optimal solution by exchanging only quantized values was proposed in [95], which studied fully decentralized quantized optimization problems over networks. At each time $t$, each agent $i$ updated its local model as follows:

$$\boldsymbol{x}_i(t+1) = (1 - \epsilon + \epsilon a_{ii})\boldsymbol{x}_i(t) + \epsilon \sum_{j \in \mathcal{N}_i} a_{ij} Q(\boldsymbol{x}_j(t)) - \eta \epsilon \nabla f_i(\boldsymbol{x}_i(t)),$$

where $\epsilon$ is some positive parameter to be chosen and $\eta$ is the stepsize. The stochastic quantizer $Q$ was assumed to be unbiased and have bounded variance. By setting $\epsilon = \mathcal{O}(1/T^{\frac{3\gamma}{2}})$ and $\eta = \mathcal{O}(1/T^{\frac{\gamma}{2}})$, it was shown for strongly convex loss functions that $\mathbb{E}[\|\boldsymbol{x}_i(T) - \boldsymbol{x}^*\|^2] \leq \mathcal{O}(1/T^{\gamma})$, where $\gamma$ is an arbitrary number in $(0, 1/2)$ and $T$ is the number of time slots. The algorithm achieved exact convergence to the optimal solution $\boldsymbol{x}^*$ by allocating diminishing weights to the quantized information received from neighbors. A similar approach was adopted in [96], where the weights for neighboring agents' quantized models converged to zero as the quantized DGD algorithm progressed. It was shown that, with random quantization schemes, the convergence rates for convex loss functions and strongly convex loss functions were $\mathcal{O}\left(\frac{\delta^2}{(1-\sigma)^2}\frac{\log t}{t^{\frac{1}{4}}}\right)$ and $\mathcal{O}\left(\frac{\delta^2}{(1-\sigma)^3}\frac{\log t}{t^{\frac{1}{3}}}\right)$, respectively, where $\delta$ is the length of the quantization interval and $1-\sigma$ is the spectral gap of the underlying communication graph.

A decentralized lazy mirror descent method with differential exchanges was developed in [97] for fully decentralized learning problems over rate-constrained noisy wireless channels. To combat the channel noise and rate constraints, the algorithm used quantization and power control techniques jointly. Besides local models, agents also maintained the disagreements in their estimates of neighbors' local models due to noise and rate constraints, and exchanged the quantized

differences with neighbors. To guarantee convergence to the optimal solution, the algorithm designed two sequences. One sequence controlled the consensus rate (i.e., the weights of neighbors' noisy quantized information), and the other one controlled the transmission power when sending the differential signals. The impact of transmission power and quantization resolution on the convergence rate was characterized. A quantized FL algorithm was devised in [98], where transmission power and quantization bits were jointly allocated across the agents to minimize the communication errors.

An iteratively refined quantization scheme was proposed for inexact (accelerated) proximal gradient methods in [99]. During the progression of the algorithm, the center of the quantization range changed as the estimates of the optimal point varied, and the quantization range shrank as the estimates became more and more accurate. If the loss functions were strongly convex, with appropriately designed dynamic quantization scheme (appropriate shrinkage rate of the quantization range), the algorithm converged to the optimal solution at linear rate. A similar approach based on DGD was adopted in [100], where an adaptive quantization scheme was used. As the algorithm progresses, one becomes more and more confident on the location of the optimal solution and adjusts the quantization codebook accordingly to make the quantized values more accurate. For convex or strongly convex loss functions, it was shown in [100] that such an adaptive quantization approach would not degrade the convergence rate compared to vanilla DGD with perfect communications, except for constant factors depending on the quantization resolution. Following this line of research, in [101], the authors designed dynamic quantization methods compressing the exchanged information into a few bits while still maintaining the linear convergence rate of the distributed learning algorithms. The convergence time of the algorithm was characterized as a function of the information transmission rate. Similar dynamic quantizers were applied to distributed gradient tracking algorithms to achieve linear convergence rates by using finite-bit communications in [102] and [103]. By using an analogous dynamic quantization scheme, [104] sought to minimize the number of quantization levels for achieving exact convergence. Exploiting dynamic quantizers, [105] explored the minimal number of quantization levels to ensure convergence of DGD over time-varying directed graphs. It was shown that one-bit communications sufficed when certain system parameters were chosen properly.

A hierarchical gradient quantization method for distributed learning was proposed in [106]. The stochastic gradient was decomposed into its norm and normalized gradient blocks, which were quantized using a uniform quantizer and a low-dimensional Grassmannian codebook, respectively. A bit-allocation scheme was used to determine the resolution of the low-dimensional quantizers for the gradient blocks. The convergence rate of this algorithm was analyzed in terms of the quantization bits. A double quantization method for distributed learning was developed in [107], where both the gradients (uplink transmission) and the models (downlink transmission) were quantized. The method was amenable to asynchronous implementation, and could be combined with gradient sparsification and momentum techniques to further improve the communication efficiency and convergence rate. Moreover, a quantized Frank-Wolfe algorithm was put forth in [108] to obtain a communication-efficient projection-free (thus alleviating the computational burden) approach. The convergence of the algorithm was analyzed for both convex and nonconvex problems. Quantized saddle-point algorithms were developed in [109] for decentralized stochastic optimization with pairwise constraints between neighbors, which could be used for multitask learning. The impact of quantization resolution on the convergence rate of the algorithms was examined for both the sample feedback and the bandit feedback (where only the values of the loss functions at two random points were revealed at each time) settings. Quantization of data instead of gradients was proposed in [110], which outperforms gradient compression significantly when model dimension is large.

A more aggressive quantization approach is to compress the exchanged information to two possible values, i.e., one bit, or three possible values. A ternary gradient approach was proposed for distributed learning in [111], where only three possible values were transmitted. The convergence of the algorithm was established theoretically. It was shown via numerical experiments that the algorithm could reduce the bandwidth requirement significantly without affecting the learning performance much. In addition, a signSGD algorithm was studied in [112], where each agent sent only the signs of the local gradients and the server used a majority vote to aggregate the signs. An FL algorithm using one-bit gradient quantization and over-the-air majority rule aggregation was proposed in [113] for distributed learning over noisy fading wireless channels. The effects of wireless communication factors, e.g., channel fading, noise, channel estimation errors, were investigated comprehensively. It was shown that the negative effects of these factors vanished as the number of agents grew. Another one-bit quantization approach proposed in [114] used only the signs of the relative models of neighbors, i.e., the signs of the differences between agents' models and neighbors' models. In the model adopted, at each time $t$, each agent $i$ updates its local model according to

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \gamma\eta_t \sum_{j\in\mathcal{N}_i} a_{ij}\,\mathrm{sgn}(\boldsymbol{x}_j(t)-\boldsymbol{x}_i(t)) - \eta_t\nabla f_i(\boldsymbol{x}_i(t)),$$

where $\gamma > 0$ is some algorithm parameter to be chosen. It was shown in [114] that the convergence of the algorithm could be guaranteed if $\gamma$ is sufficiently large, and the convergence rate was the same as that of the vanilla DGD using the exact models of neighbors. The DGD algorithm based on signs of relative models was extended to the online scenario in [115], where the training data was collected sequentially and the loss functions varied across time. It was proved that the method could achieve the same regret (in order sense) as standard OGD did. Additionally, an FL framework of training binary neural networks (BNNs) with binary model parameters was proposed in [116], where agents only needed to upload binary parameters to the server. Conditions ensuring

the convergence of the proposed BNN training algorithm were derived theoretically.

To further reduce the communication overhead, quantization techniques can be used in conjunction with other methods. An FL algorithm using quantization, probabilistic device selection, and resource allocation jointly was proposed in [117]. The method could improve the learning performance and reduce the training time significantly. Quantization techniques were integrated with variance reduction to further accelerate the convergence in [118]. Moreover, in [119], an FL algorithm combining periodic averaging, partial agent participation, and quantization was developed. The impact of these communication-efficient techniques on the convergence rate was investigated for strongly convex as well as nonconvex problems. Convergence analysis of the FedAvg algorithm with non-i.i.d. dataset distributions, partial agent participation, and finite-precision quantization was presented in [120]. It was shown that, to achieve $\mathcal{O}(1/t)$ convergence rate, transmitting the models required a logarithmic number of quantization levels, while transmitting the model differentials required only a constant number of quantization levels. A joint quantization and noise insertion approach for distributed learning was put forth in [121], which was able to achieve differential privacy and communication efficiency simultaneously.

### B. Sparsification

In addition to quantization, another popular approach to compressing the communications in distributed learning algorithms is sparsification, where only a small subset of entries of the raw information vectors are transmitted.

It was observed in [122] and [7] that most entries of the gradients used in DGD algorithm are very close to zero. Motivated by this observation, in [122], the authors proposed to map 99% of the gradient entries to zero and only transmit the rest. Empirical experiments indicated that this could reduce the communication cost significantly without degrading the learning performance much. The author of [7] also reduced the amount of communications by three orders of magnitude for training deep neural networks. The authors in [123] proposed to sparsify gradients used in SGD based on their magnitudes. Combining sparsified gradients and local error correction, the algorithm could provide convergence guarantees for both convex and nonconvex loss functions. A variant of parallel block coordinate descent algorithm based on independent sparsification of local gradients was proposed in [124]. Moreover, [125] proposed a sparsification scheme that minimized the total error incurred by sparsification throughout the learning processes under total communication budget constraint. It was found that the hard-threshold sparsifier, a variant of the Top-$k$ sparsifier (sending the $k$ entries with largest magnitudes and discarding the rest) with $k$ determined by a constant threshold, was the optimal sparsifier under such a criterion. For convex as well as nonconvex loss functions, the convergence of distributed learning algorithms using such a hard-threshold sparsifier in conjunction with error feedback was analyzed. It was proved in [125] that the algorithm had the same asymptotic convergence and linear speedup properties as SGD, and unlike conventional Top-$k$ sparsifier, had no performance loss due to data heterogeneity. To further reduce the communication overhead of distributed learning, a global Top-$k$ sparsifier was proposed in [126], where the $k$ gradient entries with globally largest absolute values from all agents were transmitted. It was shown that such a sparsifier incurred much less communication cost compared to conventional local Top-$k$ sparsifier. Additionally, a modified sparsified SGD algorithm, namely the global renovating SGD, was proposed in [127], where previous-round global gradients were utilized to estimate the current global gradient and renovate the current zero-sparsified gradients. While mitigating the communication overhead, the algorithm made the convergence direction closer to the centralized optimization, thus accelerating the distributed learning. Convergence guarantees of rate $\mathcal{O}(1/\sqrt{t})$ were provided for nonconvex learning problems.

The impact of wireless communication factors (e.g., channel fading, noise, power control) on sparsified distributed learning algorithms has also been investigated in the literature. FL over bandwidth-limited fading multiple access channels was studied in [128]. The authors proposed a compressed analog distributed SGD algorithm, where agents first sparsified their local gradients and then projected the resultant sparse vector into a low-dimensional vector for bandwidth reduction. Through bandwidth-limited wireless channels, these low-dimensional vectors from the agents were sent to the server, where the aggregation was conducted by over-the-air computations. A power allocation scheme was devised to align the received gradients at the server. A convergence analysis for this approach was presented in [129]. It was shown that the probability of reaching a small neighborhood of the optimal solution converged to one as time went to infinity. In [130], an online learning approach was developed to minimize the overall training time of FL algorithms and achieve the near-optimal communication-computation tradeoff by controlling the sparsity of the gradients. A compressive sensing (CS) approach was proposed in [131] for FL over massive MIMO systems, where sparse signals constructed from local gradients were transmitted by devices and a CS algorithm was developed to reconstruct local gradients at the central server.

In [132], the authors integrated sparsification with atomic decomposition (e.g., singular value decomposition, Fourier transform), where the atoms of the atomic decomposition of the gradients were sparsified. Notable methods such as QSGD in [94] and TernGrad in [111] could be regarded as special cases of sparsified atomic decomposition algorithm. It was shown in [132] that sparsifiying the singular values of neural network gradients, rather than their entries, led to significantly faster distributed training. A convex optimization formulation for minimizing the coding length of the stochastic gradients in distributed learning was proposed in [133], where entries of the gradients were randomly dropped out and the remaining entries were amplified to keep the sparsified gradients unbiased. A simple and fast algorithm for solving this optimization problem was developed with guaranteed sparsity. The convergence rates of distributed learning algorithms with sparse model averaging and gradient quantization were investigated for both convex and nonconvex

problems in [134]. Besides first-order algorithms, second-order distributed learning algorithms with sparsification were also studied. In [135], a distributed approximated Newton's method was proposed based on $\delta$-approximate compressors, which included Top-$k$ sparsifier as a special case. It was shown that the algorithm was able to achieve the same rate of convergence as state-of-the-art second-order distributed learning algorithms by incurring much less communication overhead. Sparsification was also applied to deep learning in [136], where only the important entries of the gradients were sent. Momentum residual accumulation was designed for tracking outdated residual gradient coordinates to avoid low convergence rate caused by sparse updates. Sparsified gradient descent algorithm was implemented as a library in [137].

### C. Error-Compensated Compression

Compressing the exchanged information usually leads to errors in distributed learning. As the learning algorithms progress, the errors caused by compression in each time slot accumulate and may degrade the learning performance severely. A remedy to this issue is to provide error feedback to the agents, who compensate for the errors dynamically to avoid error accumulation. Recently, following this general approach, a series of distributed learning algorithms with error-compensated compression have been developed, which can reduce the communication overhead significantly without compromising the learning performance much.

We use here the communication-compressed decentralized SGD algorithm proposed in [138] as an illustrative example for the error-compensated compression approach. The problem considered in [138] is the single-task decentralized learning problem (1) over a connected undirected network. The expected local loss function of each agent $i$ is given by $f_i(\boldsymbol{x}) = \mathbb{E}_{\xi_i \sim \mathcal{D}_i}[F_i(\boldsymbol{x}, \xi_i)]$, where $\xi_i$ is the local data, $\mathcal{D}_i$ is the data distribution, and $F_i$ is the loss function. Let $Q : \mathbb{R}^d \mapsto \mathbb{R}^d$ be a (possibly probabilistic) compression operator satisfying the following property:

$$\mathbb{E}_Q[\|Q(\boldsymbol{x}) - \boldsymbol{x}\|^2] \leq (1-\delta)\|\boldsymbol{x}\|^2, \quad \forall \boldsymbol{x} \in \mathbb{R}^d, \tag{9}$$

where the expectation is taken with respect to the internal randomness of the compressor $Q$, and $\delta \in (0, 1)$ is a constant. Many popular compressors satisfy property (9), including sparsifiers (e.g., Top-$k$ and Rand-$k$ (randomly picking $k$ out of $d$ entries to transmit)), random gossiping (transmitting with certain probability), and other random quantizers. Each agent $i$ maintains $(|\mathcal{N}_i|+2)$ variables, namely $\boldsymbol{x}_i(t), \{\widehat{\boldsymbol{x}}_j(t)_{j\in\mathcal{N}_i\cup\{i\}}\}$, where $\widehat{\boldsymbol{x}}_j(t)$ is an approximate local model of agent $j$. In each time $t$, agent $i$ first samples $\xi_i(t) \sim \mathcal{D}_i$. Then it updates its local model according to

$$\boldsymbol{x}_i(t+1) = \boldsymbol{x}_i(t) + \gamma \sum_{j\in\mathcal{N}_i} a_{ij}(\widehat{\boldsymbol{x}}_j(t) - \widehat{\boldsymbol{x}}_i(t)) - \eta_t \nabla F_i(\boldsymbol{x}_i(t), \xi_i(t)), \tag{10}$$

where $\gamma > 0$ is an algorithm parameter to be selected. Note that in (10), agent $i$ uses the approximate model $\widehat{\boldsymbol{x}}_j(t)$ instead of the exact model $\boldsymbol{x}_j(t)$ ($j \in \mathcal{N}_i$), which is not accessible to agent $i$. Afterwards, agent $i$ computes

$$\boldsymbol{q}_i(t) = Q(\boldsymbol{x}_i(t+1) - \widehat{\boldsymbol{x}}_i(t)), \tag{11}$$

and sends $\boldsymbol{q}_i(t)$ to all neighbors in $\mathcal{N}_i$. Symmetrically, it receives $\boldsymbol{q}_j(t)$ from all neighbors $j \in \mathcal{N}_i$, and updates the approximate local model by

$$\widehat{\boldsymbol{x}}_j(t+1) = \boldsymbol{q}_j(t) + \widehat{\boldsymbol{x}}_j(t), \quad \forall j \in \mathcal{N}_i \cup \{i\}. \tag{12}$$

This algorithm conducts error-compensation in steps (11) and (12). Specifically, step (11) compresses the difference between the new local model $\boldsymbol{x}_i(t+1)$ and the previous *approximate* local model $\widehat{\boldsymbol{x}}_i(t)$, which contains errors caused by compressed communications so far. Thus, in (12), $\boldsymbol{q}_j(t)$ is able to partially offset the compression errors in the previous approximate model $\widehat{\boldsymbol{x}}_j(t)$. In particular, if $Q$ is replaced by an identity mapping at time $t$, then combining (11) and (12) yields $\widehat{\boldsymbol{x}}_i(t+1) = \boldsymbol{x}_i(t+1)$ readily (i.e., zero error), no matter how large the gap $\widehat{\boldsymbol{x}}_i(t) - \boldsymbol{x}_i(t)$ was previously. It has been shown in [138] that, if the loss functions $\{f_i\}$ are $\mu$-strongly convex, then the algorithm converges at rate $\mathcal{O}(\frac{\sigma^2}{\mu n t})$, where $\sigma^2$ is the variance of the stochastic gradients $\nabla F_i(\boldsymbol{x}, \xi_i)$. This recovers the convergence rate of mini-batch SGD with perfect communications. In the convergence bound, communication compression (e.g., the compression accuracy factor $\delta$) only affects higher order terms that are negligible as time $t$ goes to infinity. This suggests that the error-compensated decentralized learning algorithm in [138] is able to reduce communication overhead significantly (by sending information compressed by $Q$) without degrading the learning performance much. Numerical experiments show that, to achieve the same learning performance, the number of bits communicated by the error-compensated algorithm is smaller than that of vanilla SGD by orders of magnitude. Decentralized learning algorithms with error-compensated compressed communications were also studied in [139], where two different compression strategies, namely extrapolation compression and difference compression, were used. When the compressors were unbiased and had bounded variances, it was shown for nonconvex learning problems that the algorithm converged at rate $\mathcal{O}(1/\sqrt{nt})$, matching the convergence rate of centralized learning with perfect communications. Error-compensated compression and event-triggered communications were combined to further improve the communication efficiency of decentralized optimization algorithms in [140]. Further, momentum SGD with error-compensated compressed communications was studied in [141], which imposed weaker assumptions on the variance and dissimilarity of the gradients. Decentralized optimization with sparsification and error-compensated compression was investigated in [142].

Distributed learning algorithms with error-compensated communication compression have also been studied for server-agent systems. A sparsified SGD algorithm with error-compensation was developed in [143], and was shown to converge at the same rate as vanilla SGD. Distributed SGD with error-compensated stochastic quantization was proposed in [144], and its convergence was analyzed for the case of

quadratic optimization, though its convergence rate was not shown to be the same as vanilla SGD. Error-compensated signSGD was developed in [145], and the algorithm was shown to achieve the same convergence rate as vanilla SGD. An asynchronous error-compensated distributed SGD algorithm composing quantization and sparsification was proposed in [146], where each agent communicated with the server infrequently at different time instants. It was shown in [146] that despite this aggressive compression, the algorithm could achieve the same convergence rate as vanilla SGD for both convex and nonconvex problems. A general framework for devising and analyzing error-compensated quantized distributed learning algorithms was presented in [147], where linear convergence rates could be guaranteed. Linearly converging error-compensated distributed SGD with improved convergence rate was developed in [148] based on loopless Katyusha method. Error-compensated communication compression was further extended to distributed learning algorithms with variance reduction techniques in [149], where the variance of stochastic gradient was reduced by taking a moving average over all historical gradients. In such a case, only using the compression error in the previous time instant was not enough for fully compensating for the compression errors. An error-compensation algorithm using the compression errors from the previous two time instants was proposed and was shown to achieve the same convergence rate as the case without compression. A distributed SGD with double-pass error-compensated compression was proposed in [150], where the compression was conducted at both the server and the agents. Hessian-based error-compensated compression was developed in [151], which was especially suitable for ill-conditioned problems. A saddle-point algorithm with error-compensated compression was studied in [152] to solve decentralized multitask learning problems.

### D. Other Compression Methods

In addition to quantization, sparsification and error-compensated compression techniques, researchers have devised other communication compression methods for distributed learning algorithms. In [153], the models sent by the agents to the server were restricted to have certain structures such as low-rank in order to reduce the communication overhead. In [8], a variety of techniques were employed to reduce the communication bandwidth of distributed learning algorithms comprehensively, including momentum correction, local gradient clipping, momentum factor masking, and warm-up training. In [154], a low-rank gradient compressor based on power iterations was proposed for distributed learning that could achieve test performance on par with SGD. Communication-efficient FL algorithms based on sketching were devised in [155]. Additionally, communication-efficient multi-agent actor-critic algorithm for multi-agent RL over directed graphs was examined in [156], where each agent only sent two scalars at each time.

FL based on over-the-air computation was proposed in [157] to reduce the bandwidth requirement by exploiting the superposition property of wireless multiple-access channels. The algorithm used joint device selection and beamforming design, which were modeled as a sparse low-rank optimization problem. To solve this nonconvex problem, a difference-of-convex (DC) algorithm with global convergence guarantee was developed. The effects of over-the-air analog aggregation (e.g., waveform superposition and communication latency reduction) on the performance of FL algorithms were further investigated in [158] and [159]. Moreover, [160] developed a band-limited coordinate descent approach by $k$-sparsifying the gradients and transmitting the gradient entries over $k$ subcarriers through wireless channels. Learning-driven communication error minimization was studied by jointly optimizing the power allocation and learning rates. In [161], the learning rate of the FL algorithm was optimized dynamically and beamforming subject to power constraints was also designed.

### E. Future Directions

We provide two potential directions for future work on distributed learning with compressed communications.

*1) Communication Compression for Distributed Online Learning:* Prior work on distributed online learning with compressed communications is rather limited. A decentralized online learning algorithm using the signs of the relative local models of neighboring agents has been proposed in [115]. The approach required each agent to be able to observe the signs of the models of neighbors relative to its own model, which might not be the case in practice. One future direction of research would be to devise distributed online learning algorithms that quantize/compress the local models directly (instead of the *relative* local models, i.e., the difference between neighbors' models). The quantization/compression schemes will have to be designed such the degradation of online learning performance (e.g., regret and constraint violations) is minimal.

One possible approach is to design a dynamic quantizer, which adjusts the length and the center of the quantization interval on-the-fly. Specifically, as the algorithm progresses and becomes more confident about the location of the dynamic optimal solution, the length of the quantization interval could be shrunk, leading to higher quantization resolution. This can potentially reduce the communication overhead of distributed online algorithms without hurting the regret and constraint violations in order sense. A challenge to this approach is that, unlike static learning problems, the optimal solutions of online learning problems change with time and it is more difficult to locate them with high confidence. Technical assumptions limiting the temporal variation speed of the online problems may be needed to resolve this issue. Another approach would be to adjust the algorithm parameters, e.g., the combination weights of decentralized OGD algorithm, so that the impact of the quantization errors vanishes gradually as the algorithm progresses.

It is also possible to devise error-compensated compression schemes for distributed online learning. When the training data is collected sequentially and loss functions vary with time, it would be interesting to see if the compression errors can still be compensated for dynamically so that the regret is not affected by communication compression in order sense.

*2) Performance Limits Under Communication Rate Constraints:* Most existing works on distributed learning with compressed communications are focused on algorithm design. Yet little is known about the fundamental performance limits of distributed learning when communications are compressed. With limited communication bandwidth, the data rate of information exchange in distributed learning algorithms is constrained. Under such communication rate constraints, one would seek to establish lower bounds for the training loss (e.g., the gap between the loss functions of the trained model and the optimal model) or testing performance (e.g., generalization error), and ascertain the impact of communication rate on these lower bounds. It would be interesting to see if existing learning algorithms with compressed communications can achieve such lower bounds in order sense. If not, one could look into designing novel compression methods to match the derived performance lower bounds.

## V. Resource Management for Communication-Efficient Distributed Learning

The information exchange required by distributed learning algorithms consumes substantial amount of radio resources, such as energy and bandwidth, which are scarce in many practical circumstances. In this section, we provide an overview of resource management techniques for communication-efficient distributed learning algorithms, which seek to achieve the best learning performance under resource budget constraints.

### A. Power Allocation

A variety of power allocation schemes have been proposed to obtain satisfactory performance for FL under energy constraints. For FL over wireless networks, in [162], the authors took transmission energy (arising from sending local models to the server) and computation energy (arising from the local training steps) into consideration, and minimized the total energy consumption subject to constraints on computation and communication latencies. An iterative algorithm was developed to solve this optimization problem, where closed-form solutions for time/power/bandwidth allocation were derived. In [9], a joint transmit power allocation and device selection problem was studied to achieve the best FL performance over wireless networks. A closed-form expression for the convergence rate of the FL algorithm was first derived to quantify the impact of wireless factors on the training loss. Then, based on this convergence rate, the optimal scheme for transmit power allocation, user selection, and uplink resource block allocation was developed. Additionally, a resource allocation problem was formulated in [163] and [164] to achieve the optimal tradeoff between FL convergence and energy consumption. Such a resource allocation problem was nonconvex, and was decomposed into three convex subproblems. The globally optimal resource allocation scheme was obtained by characterizing the solution structures of the subproblems.

Convergence analysis of FL over noisy fading wireless channels was studied in [165] recently. Power allocation problems were formulated to minimize the convergence bound subject to a set of average and maximum power constraints at individual edge devices. The problems were transformed into convex forms, and their structured optimal solutions, appearing in the form of regularized channel inversion, were obtained by using the Lagrangian duality method. Moreover, FL system with over-the-air analog gradient aggregation was examined in [166]. Dynamic agent participation scheduling and power allocation schemes were proposed to optimize the training performance under energy constraints of the agents, where both the communication and computation energy consumptions were taken into account. The energy consumption of FL algorithms has been studied by other approaches as well, beyond power control. In [167], the total cost of FL, arising from the training time and energy consumption, was minimized by choosing agent participation and the number of local iterations. Solution properties of the formulated problem were derived to identify the design principles of FL algorithms. Further, a semi-asynchronous federated learning algorithm was developed in [168], where the server aggregated a certain number of local models based on their arrival orders in each time. A convergence bound for the algorithm was established, and the training time was minimized under communication cost constraints and FL accuracy constraints by choosing an appropriate number of participating agents.

### B. Bandwidth Allocation

Bandwidth allocation has also been investigated extensively and is often utilized in conjunction with other techniques (such as agent selection and power control) to improve the communication efficiency of FL. In [169], for FL over wireless networks, a stochastic optimization problem minimizing the long-term learning loss under long-term energy constraints was studied by selecting agent participation and allocating bandwidth. An algorithm utilizing only the currently available wireless channel information was devised to solve this stochastic optimization problem. A joint probabilistic user selection and resource block (spectrum bands) allocation scheme was developed in [170] to minimize the training loss and convergence time of the FL algorithm, where only those users with significant impact on the global model were selected to upload their local models. A joint bandwidth allocation and device selection scheme was proposed in [171] to maximize the training accuracy subject to total training time constraints for latency-constrained FL. Moreover, joint power and bandwidth allocation was investigated in [172] to minimize the energy consumption, computation cost and time cost of the FL algorithm. In [173], a channel allocation problem was investigated to minimize the training delays subject to differential privacy and training performance constraints. A joint bandwidth allocation and user selection problem was examined in [174] in the scenario of visible light communication.

Asynchronous FL with limited wireless resources was studied in [175]. A metric named effectivity score was proposed to represent the amount of learning. An asynchronous learning-aware transmission scheduling (ALS) problem to maximize the effectivity score subject to resource constraints (e.g., spectrum constraints) was formulated. When the statistical information of the system uncertainties (e.g., channel conditions, data

arrivals, and radio resource availability) was unknown, the scheduling problem could be solved through a Bayesian learning approach. Hierarchical FL was introduced in [176], where small-cell base stations coordinated the mobile users within their cells and periodically exchanged model updates with the main base station, i.e., the server. A method was proposed to optimize the allocation of subcarriers so as to reduce the communication latency of the FL algorithm. In addition, a collaborative FL architecture supporting deep neural network (DNN) training was considered in [177], which sought to optimally select participating devices and allocate computing and spectrum resources. A stochastic optimization problem with the objective of minimizing learning loss while satisfying delay and long-term energy consumption requirements was formulated. A deep multi-agent reinforcement learning approach was developed to solve the problem. Moreover, FL with the assistance of intelligent reflection surface was proposed in [178], and the delay of FL was analyzed in [179].

### C. Future Directions

Several promising directions for future research on resource management in distributed learning are discussed below.

*1) Improving Communication Efficiency and Data Privacy Simultaneously:* In distributed learning, the loss functions of the agents depend on their local private data, which often contain sensitive information, e.g., health information and financial information. Even though the agents do not need to share their raw data with others in a distributed learning setting, the exchanged information between agents and the server may still be overheard and utilized by malicious adversaries to (partially) infer the private data of the agents. The noise in wireless channels, a nuisance from the perspective of communication, can help preserve the data privacy by preventing adversaries from inferring the private data of agents accurately based on the overheard noisy information. Transmission power of agents also influence the data privacy of agents. Large transmission power enhances the signal-to-noise ratio at adversaries and makes it easier to infer the private data of agents. On the other hand, low transmission power hinders accurate information exchange in distributed learning algorithms, and thus degrades the learning performance. It is therefore imperative to devise power allocation schemes (across time and agents) to balance the learning performance and data privacy under energy budget constraints of agents. The goal is to achieve an optimal tradeoff between learning performance, data privacy, and communication efficiency for distributed learning over wireless networks.

*2) Impact of Wireless Interference in Decentralized Networks:* Most of the existing works on resource management for communication-efficient distributed learning are focused on the server-agent setting, where all agents communicate with a server through a multiple-access channel. When the multi-agent system is a fully decentralized network without a central server, each agent exchanges information with its neighbors and the concurrent information transmissions of different agents can cause mutual interference, which affects the communication accuracy and the learning performance. It is therefore important to design novel transmission scheduling and power allocation mechanisms to mitigate the negative effects of wireless interference on decentralized learning. For instance, the information transmission should be scheduled so that agents located close to each other do not transmit simultaneously to avoid strong interference. In addition, agents should avoid using high transmission power to compensate for poor channel conditions as this would lead to strong interference to nearby agents.

*3) Resource Allocation for Communication-Efficient Distributed Online Learning:* Most existing resource allocation schemes are designed for communication-efficient distributed learning in static settings, where the loss functions are fixed. When the training data is collected sequentially and learning is conducted in real time, it is still unclear how to allocate radio resources (e.g., power and bandwidth) to achieve the best online learning performance under resource budget constraints. One can first study the impact of limited radio resources on the regret and constraint violations of various distributed online learning algorithms (e.g., distributed OGD, online saddle-point algorithm). Then, one can minimize the performance bounds on the regret and constraint violations by allotting the radio resources in an optimal manner.

## VI. Game Theory for Communication-Efficient Distributed Learning

The information exchange required by distributed learning algorithms consumes substantial amount of communication resources, which are often scarce for users. For instance, mobile devices may have limited amount of energy and communication bandwidth. Therefore, users may not be willing to participate in the distributed learning algorithms or may not devote sufficient radio resources to the learning algorithms, which would then lead to deterioration of the learning performance. To cope with this challenge, several recent works have devised game-theoretic mechanisms to compensate for the resource consumption of users and incentivize their participation in distributed learning. In this section, we present an overview of this line of research and point out several potential directions for future research.

### A. Existing Works

In [10], the authors studied FL in a server-agent system, and sought to design an optimal incentive mechanism from the server's perspective in the presence of users' multi-dimensional private information, including training cost (such as communication and computation energy cost) and communication delays. The authors proposed a multi-dimensional contract-theoretic mechanism, which summarized users' multi-dimensional private information into a one-dimensional criterion that entails a complete ordering of users. Analysis in various information scenarios was conducted to reveal the impact of information asymmetry levels on the server's optimal strategy. Reputation was introduced as a metric to measure the reliability and trustworthiness of the mobile users in [180]. A reputation-based user selection scheme was developed for reliable FL by using a multiweight subjective logic model.

An incentive mechanism combining reputation and contract theory was devised to motivate high-reputation users with high-quality data to participate in the FL algorithm. Moreover, in [181], an incentive mechanism based on deep reinforcement learning was devised for FL to determine the optimal pricing strategy for the server and the optimal training strategies for the users, where the utility functions of users took their communication and computation costs into account. Auction mechanisms were proposed in [182] to incentivize users to contribute communication/computation resources and private data to FL algorithms. An approximate strategy-proof mechanism with guaranteed truthfulness, individual rationality and computational efficiency was designed. To further improve the social welfare, an automated strategy-proof mechanism based on deep reinforcement learning was also devised. Additionally, a hierarchical FL framework was studied in [183], where users first transmitted local models to edge servers for intermediate aggregation, and then edge servers communicated with the model owner for global aggregation. Such an approach could reduce the number of global communications and mitigate the straggler effect of users. A hierarchical game was proposed for the edge association and resource allocation problem, where users' strategies were their edge associations and the edge servers' strategies were their bandwidth allocation schemes. The lower-level interaction between the users were modeled as an evolutionary game. The upper-level interaction between the edge servers and the model owner was modeled as a Stackelberg differential game, where the model owner decided an optimal reward scheme given the expected bandwidth allocation strategies of the edge servers.

### B. Future Directions

Two possible future directions on the use of game theory for communication-efficient distributed learning are presented below.

*1) Game Theory for Communication-Efficient Fully Decentralized Learning over Networks:* Most existing works on the use of game theory for distributed learning are focused on server-agent systems, in which a central server interacts with a set of strategic agents. Yet little is known about the strategic behavior of agents in a fully decentralized learning setting over a network without central entity. Decentralized learning algorithms require agents to exchange information with neighbors to facilitate collaborative learning. When determining the amount of radio resources (e.g., energy and bandwidth) devoted to information transmission, agents need to take into consideration both their local resource budget constraints and the interference to other nearby agents sending/receiving information concurrently. For example, if an agent sends information with very large transmission power, other nearby concurrent transmissions cannot be received accurately due to the strong interference, which may degrade the collaborative learning performance. Using a non-cooperative game framework, one can study the strategic behavior (e.g., power control and spectrum usage) of agents in such a decentralized learning setting. It would be interesting to examine the price-of-anarchy by comparing the learning performance of a non-cooperative game and that of a fully cooperative scenario with globally optimal resource allocation scheme. Further, one may devise game-theoretic incentive mechanisms (e.g., auction and bargaining) to guide agents' behavior and ameliorate the performance of decentralized learning.

*2) Incentive Mechanism Design for Communication-Efficient Personalized Learning:* Existing incentive mechanisms are mostly designed for single-task distributed learning (problem (1)), where all agents collaborate to learn a common model. It would be interesting to study the strategic behavior of agents in distributed personalized learning, where each agent has its own model to train and different agents' models are distinct (but related). In personalized learning, e.g., multitask learning (problems (3), (6), (7)) and meta-learning (problem (4)), each agent aims to obtain the best personal model by using its scarce radio resources, and is indifferent about the learning accuracy of other agents' models. The decision making processes are coupled across agents since the local models of individual agents are related. Through a non-cooperative game framework, one can investigate the equilibrium resource allocation strategies of agents and the performance of personalized learning algorithms at the equilibrium. To improve the learning performance, one may further devise game-theoretic mechanisms incentivizing agents to contribute sufficient radio resources to personalized learning algorithms.

## VII. Conclusion

In this paper, we have presented a holistic overview of communication-efficient distributed learning. First, we have surveyed methods reducing the number of communication rounds for distributed learning, including multiple local training steps between consecutive communications and event-triggered communications. Second, we have reviewed various communication compression schemes for distributed learning, such as quantization, sparsification, and error-compensated compression. Third, resource management techniques, e.g., power control and bandwidth allocation, have been presented to make the most of the limited radio resources to achieve the best learning performance. Finally, several recent studies on the game-theoretic mechanism design for incentivizing user participation in distributed learning have been discussed. In addition to reviewing existing works, for each of these communication-efficient distributed learning methods, we have also pointed out potential directions for future research.

## References

[1] A. Nedic and A. Ozdaglar, "Distributed subgradient methods for multi-agent optimization," *IEEE Trans. Autom. Control*, vol. 54, no. 1, pp. 48–61, Jan. 2009.

[2] M. Mahdavi, R. Jin, and T. Yang, "Trading regret for efficiency: Online convex optimization with long term constraints," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 2503–2528, 2012.

[3] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 62, no. 7, pp. 1750–1761, Apr. 2014.

[4] A. Mokhtari, Q. Ling, and A. Ribeiro, "Network Newton distributed optimization methods," *IEEE Trans. Signal Process.*, vol. 65, no. 1, pp. 146–161, Jan. 2016.

[5] S. Wang et al., "Adaptive federated learning in resource constrained edge computing systems," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1205–1221, 2019.

[6] M. G. Rabbat and R. D. Nowak, "Quantized incremental algorithms for distributed optimization," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 4, pp. 798–808, Apr. 2005.

[7] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *Proc. 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015.

[8] Y. Lin, S. Han, H. Mao, Y. Wang, and W. J. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *Proc. Int. Conf. Learn. Represent.*, 2018.

[9] M. Chen, Z. Yang, W. Saad, C. Yin, H. V. Poor, and S. Cui, "A joint learning and communications framework for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 269–283, Oct. 2020.

[10] N. Ding, Z. Fang, and J. Huang, "Optimal contract design for efficient federated learning with multi-dimensional private information," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 1, pp. 186–200, Jan. 2020.

[11] P. Kairouz et al., "Advances and open problems in federated learning," *Found. Trends Mach. Learn.*, 2021.

[12] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, "Edge artificial intelligence for 6G: Vision, enabling technologies, and applications," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 5–36, Jan. 2022.

[13] T. Gafni, N. Shlezinger, K. Cohen, Y. C. Eldar, and H. V. Poor, "Federated learning: A signal processing perspective," *IEEE Signal Process. Mag.*, vol. 39, no. 3, pp. 14–41, May 2022.

[14] D. Bertsekas and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods*. Upper Saddle River, NJ, USA: Prentice-Hall, 1989.

[15] S. Boyd, N. Parikh, and E. Chu, *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Delft, The Netherlands: Zuid-Holland, 2011.

[16] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, no. 2, pp. 107–194, 2011.

[17] E. Hazan, "Introduction to online convex optimization," *Found. Trends Optim.*, vol. 2, nos. 3–4, pp. 157–325, 2016.

[18] M. Zinkevich, "Online convex programming and generalized infinitesimal gradient ascent," in *Proc. 20th Int. Conf. Mach. Learn.*, 2003, pp. 928–936.

[19] S. Liu, S. J. Pan, and Q. Ho, "Distributed multi-task relationship learning," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2017, pp. 937–946.

[20] Y. Zhang and D.-Y. Yeung, "A regularization approach to learning task relationships in multitask learning," *ACM Trans. Knowl. Discovery Data*, vol. 8, no. 3, pp. 1–31, Jun. 2014.

[21] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[22] C. Finn, K. Xu, and S. Levine, "Probabilistic model-agnostic meta-learning," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 9537–9548.

[23] A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized federated learning with theoretical guarantees: A model-agnostic meta-learning approach," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3557–3568.

[24] C. T. Dinh, N. Tran, and J. Nguyen, "Personalized federated learning with Moreau envelopes," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 21394–21405.

[25] K. Ozkara, N. Singh, D. Data, and S. Diggavi, "QuPeD: Quantized personalization via distillation with applications to federated learning," in *Adv. Neural Inf. Process. Syst.*, vol. 34, 2021.

[26] F. Hanzely, S. Hanzely, S. Horváth, and P. Richtárik, "Lower bounds and optimal algorithms for personalized federated learning," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 2304–2315.

[27] A. Nedic, A. Ozdaglar, and P. A. Parrilo, "Constrained consensus and optimization in multi-agent networks," *IEEE Trans. Autom. Control*, vol. 55, no. 4, pp. 922–938, Apr. 2010.

[28] A. Nedić, A. Olshevsky, and W. Shi, "Achieving geometric convergence for distributed optimization over time-varying graphs," *SIAM J. Optim.*, vol. 27, no. 4, pp. 2597–2633, 2017.

[29] S. Pu, W. Shi, J. Xu, and A. Nedić, "Push–pull gradient methods for distributed optimization in networks," *IEEE Trans. Autom. Control*, vol. 66, no. 1, pp. 1–16, Feb. 2020.

[30] A. Nedić and A. Olshevsky, "Distributed optimization over time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 60, no. 3, pp. 601–615, Mar. 2015.

[31] W. Shi, Q. Ling, G. Wu, and W. Yin, "Extra: An exact first-order algorithm for decentralized consensus optimization," *J. Optim.*, vol. 25, no. 2, pp. 944–966, May 2015.

[32] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2566–2581, Jun. 2019.

[33] Y. Tang, J. Zhang, and N. Li, "Distributed zero-order algorithms for nonconvex multiagent optimization," *IEEE Trans. Control Netw. Syst.*, vol. 8, no. 1, pp. 269–281, Mar. 2021.

[34] M. Eisen, A. Mokhtari, and A. Ribeiro, "Decentralized quasi-Newton methods," *IEEE Trans. Signal Process.*, vol. 65, no. 10, pp. 2613–2628, May 2017.

[35] Q. Ling, W. Shi, G. Wu, and A. Ribeiro, "DLM: Decentralized linearized alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 63, no. 15, pp. 4051–4064, Aug. 2015.

[36] A. Mokhtari, W. Shi, Q. Ling, and A. Ribeiro, "DQM: Decentralized quadratically approximated alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 64, no. 19, pp. 5158–5173, Mar. 2016.

[37] F. Yan, S. Sundaram, S. V. N. Vishwanathan, and Y. 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.

[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] M. Akbari, B. Gharesifard, and T. Linder, "Distributed online convex optimization on time-varying directed graphs," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 3, pp. 417–428, Sep. 2015.

[40] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1185–1197, Dec. 2013.

[41] X. Cao and K. J. R. Liu, "Distributed linearized ADMM for network cost minimization," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 4, no. 3, pp. 626–638, Sep. 2018.

[42] X. Cao and K. J. R. Liu, "Distributed Newton's method for network cost minimization," *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 1278–1285, Mar. 2021.

[43] A. Koppel, B. M. Sadler, and A. Ribeiro, "Proximity without consensus in online multiagent optimization," *IEEE Trans. Signal Process.*, vol. 65, no. 12, pp. 3062–3077, Mar. 2017.

[44] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.

[45] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.

[46] X. Cao and K. J. R. Liu, "Decentralized sparse multitask RLS over networks," *IEEE Trans. Signal Process.*, vol. 65, no. 23, pp. 6217–6232, Dec. 2017.

[47] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2017, pp. 1273–1282.

[48] H. Yu, S. Yang, and S. Zhu, "Parallel restarted SGD with faster convergence and less communication: Demystifying why model averaging works for deep learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 5693–5700.

[49] S. U. Stich, "Local SGD converges fast and communicates little," in *Proc. Int. Conf. Learn. Represent.*, 2019.

[50] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, "Don't use large mini-batches, use local SGD," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[51] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. R. Cadambe, "Local SGD with periodic averaging: Tighter analysis and adaptive synchronization," in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 11082–11094.

[52] J. Wang and G. Joshi, "Adaptive communication strategies to achieve the best error-runtime trade-off in local-update SGD," in *Proc. Conf. Mach. Learn. Syst.*, 2019, pp. 212–229.

[53] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 7611–7623.

[54] J. Wang, V. Tantia, N. Ballas, and M. Rabbat, "SlowMo: Improving communication-efficient distributed SGD with slow momentum," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[55] B. Woodworth et al., "Is local SGD better than minibatch SGD?" in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 10334–10343.

[56] S. Zhang, A. Choromanska, and Y. LeCun, "Deep learning with elastic averaging SGD," in *Adv. Neural Inf. Process. Syst.*, 2015.

[57] J. Wang and G. Joshi, "Cooperative SGD: A unified framework for the design and analysis of local-update SGD algorithms," *J. Mach. Learn. Res.*, vol. 22, no. 213, pp. 1–50, 2021.

[58] G. Lan, S. Lee, and Y. Zhou, "Communication-efficient algorithms for decentralized and stochastic optimization," *Math. Program.*, vol. 180, nos. 1–2, pp. 237–284, Mar. 2020.

[59] F. P.-C. Lin, S. Hosseinalipour, S. S. Azam, C. G. Brinton, and N. Michelusi, "Semi-decentralized federated learning with cooperative D2D local model aggregations," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3851–3869, Dec. 2021.

[60] L. Liu, J. Zhang, S. H. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.

[61] M. Jaggi et al., "Communication-efficient distributed dual coordinate ascent," in *Adv. Neural Inf. Process. Syst.*, 2014, pp. 3068–3076.

[62] Y. Chen, X. Sun, and Y. Jin, "Communication-efficient federated deep learning with layerwise asynchronous model update and temporally weighted aggregation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4229–4238, Oct. 2019.

[63] C. Liu, H. Li, Y. Shi, and D. Xu, "Distributed event-triggered gradient method for constrained convex minimization," *IEEE Trans. Autom. Control*, vol. 65, no. 2, pp. 778–785, Feb. 2020.

[64] M. Zhong and C. G. Cassandras, "Asynchronous distributed optimization with event-driven communication," *IEEE Trans. Autom. Control*, vol. 55, no. 12, pp. 2735–2750, Dec. 2010.

[65] Y. Kajiyama, N. Hayashi, and S. Takai, "Distributed subgradient method with edge-based event-triggered communication," *IEEE Trans. Autom. Control*, vol. 63, no. 7, pp. 2248–2255, Jul. 2018.

[66] J. George and P. Gurram, "Distributed stochastic gradient descent with event-triggered communication," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 7169–7178.

[67] Z. Wu, Z. Li, Z. Ding, and Z. Li, "Distributed continuous-time optimization with scalable adaptive event-based mechanisms," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 50, no. 9, pp. 3252–3257, Sep. 2018.

[68] W. Du, X. Yi, J. George, K. H. Johansson, and T. Yang, "Distributed optimization with dynamic event-triggered mechanisms," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 969–974.

[69] W. Chen and W. Ren, "Event-triggered zero-gradient-sum distributed consensus optimization over directed networks," *Automatica*, vol. 65, pp. 90–97, Mar. 2016.

[70] P. Wan and M. D. Lemmon, "Event-triggered distributed optimization in sensor networks," in *Proc. Int. Conf. Inf. Process. Sensor Netw.*, 2009, pp. 49–60.

[71] L. Gao, S. Deng, H. Li, and C. Li, "An event-triggered approach for gradient tracking in consensus-based distributed optimization," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 2, pp. 510–523, Mar. 2022.

[72] J. Kim and W. Choi, "Gradient-push algorithm for distributed optimization with event-triggered communications," 2021, *arXiv:2111.06315*.

[73] B. Hu, Z.-H. Guan, G. Chen, and X. Shen, "A distributed hybrid event-time-driven scheme for optimization over sensor networks," *IEEE Trans. Ind. Electron.*, vol. 66, no. 9, pp. 7199–7208, Sep. 2018.

[74] Y. Liu, W. Xu, G. Wu, Z. Tian, and Q. Ling, "Communication-censored ADMM for decentralized consensus optimization," *IEEE Trans. Signal Process.*, vol. 67, no. 10, pp. 2565–2579, Mar. 2019.

[75] X. Cao and T. Basar, "Decentralized online convex optimization with event-triggered communications," *IEEE Trans. Signal Process.*, vol. 69, pp. 284–299, 2021.

[76] S. Liu, L. Xie, and D. E. Quevedo, "Event-triggered quantized communication-based distributed convex optimization," *IEEE Trans. Control Netw. Syst.*, vol. 5, no. 1, pp. 167–178, Mar. 2016.

[77] H. Li, S. Liu, Y. C. Soh, and L. Xie, "Event-triggered communication and data rate constraint for distributed optimization of multiagent systems," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 11, pp. 1908–1919, Nov. 2017.

[78] T. Chen, G. B. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Adv. Neural Inf. Process. Syst.*, 2018.

[79] J. Sun, T. Chen, G. B. Giannakis, and Z. Yang, "Communication-efficient distributed learning via lazily aggregated quantized gradients," in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 3370–3380.

[80] T. Chen, K. Zhang, G. B. Giannakis, and T. Basar, "Communication-efficient policy gradient methods for distributed reinforcement learning," *IEEE Trans. Control Netw. Syst.*, vol. 9, no. 2, pp. 917–929, Jun. 2022.

[81] H. Yu and R. Jin, "On the computation and communication complexity of parallel SGD with dynamic batch sizes for stochastic non-convex optimization," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7174–7183.

[82] F. Haddadpour, M. M. Kamani, M. Mahdavi, and V. Cadambe, "Trading redundancy for communication: Speeding up distributed SGD for non-convex optimization," in *Proc. Int. Conf. Mach. Learn.*, pp. 2545–2554, 2019.

[83] J. N. Tsitsiklis and Z.-Q. Luo, "Communication complexity of convex optimization," *J. Complex.*, vol. 3, no. 3, pp. 231–243, 1987.

[84] Y. Arjevani and O. Shamir, "Communication complexity of distributed convex learning and optimization," in *Adv. Neural Inf. Process. Syst.*, 2015, pp. 1756–1764.

[85] S. S. Vempala, R. Wang, and D. P. Woodruff, "The communication complexity of optimization," in *Proc. 14th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2020, pp. 1733–1752.

[86] B. Woodworth, B. Bullins, O. Shamir, and N. Srebro, "The min-max complexity of distributed stochastic convex optimization with intermittent communication," in *Proc. 34th Annu. Conf. Learn. Theory*, 2021.

[87] A. Agarwal, M. J. Wainwright, P. Bartlett, and P. Ravikumar, "Information-theoretic lower bounds on the Oracle complexity of convex optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 22, 2009.

[88] P. Mayekar and H. Tyagi, "RATQ: A universal fixed-length quantizer for stochastic optimization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 1399–1409.

[89] A. Nedic, A. Olshevsky, A. Ozdaglar, and J. N. Tsitsiklis, "Distributed subgradient methods and quantization effects," in *Proc. 47th IEEE Conf. Decis. Control*, Dec. 2008, pp. 4177–4184.

[90] N. Shlezinger, M. Chen, Y. C. Eldar, H. V. Poor, and S. Cui, "UVeQFed: Universal vector quantization for federated learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 500–514, 2020.

[91] D. Yuan, S. Xu, H. Zhao, and L. Rong, "Distributed dual averaging method for multi-agent optimization with quantized communication," *Syst. Control Lett.*, vol. 61, no. 11, pp. 1053–1061, Nov. 2012.

[92] S. Zhu and B. Chen, "Quantized consensus by the ADMM: Probabilistic versus deterministic quantizers," *IEEE Trans. Signal Process.*, vol. 64, no. 7, pp. 1700–1713, Apr. 2015.

[93] A. S. Berahas, C. Iakovidou, and E. Wei, "Nested distributed gradient methods with adaptive quantized communication," in *Proc. IEEE 58th Conf. Decis. Control*, Dec. 2019, pp. 1519–1525.

[94] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1709–1720.

[95] A. Reisizadeh, A. Mokhtari, H. Hassani, and R. Pedarsani, "An exact quantized decentralized gradient descent algorithm," *IEEE Trans. Signal Process.*, vol. 67, no. 19, pp. 4934–4947, Oct. 2019.

[96] T. T. Doan, S. T. Maguluri, and J. Romberg, "Convergence rates of distributed gradient methods under random quantization: A stochastic approximation approach," *IEEE Trans. Autom. Control*, vol. 66, no. 10, pp. 4469–4484, Oct. 2021.

[97] R. Saha, S. Rini, M. Rao, and A. Goldsmith, "Decentralized optimization over noisy, rate-constrained networks: Achieving consensus by communicating differences," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 2, pp. 449–467, Oct. 2022.

[98] Y. Wang, Y. Xu, Q. Shi, and T.-H. Chang, "Quantized federated learning under transmission delay and outage constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 323–341, Jan. 2022.

[99] Y. Pu, M. N. Zeilinger, and C. N. Jones, "Quantization design for distributed optimization," *IEEE Trans. Autom. Control*, vol. 62, no. 5, pp. 2107–2120, May 2017.

[100] T. T. Doan, S. T. Maguluri, and J. Romberg, "Fast convergence rates of distributed subgradient methods with adaptive quantization," *IEEE Trans. Autom. Control*, vol. 66, no. 5, pp. 2191–2205, May 2021.

[101] S. Magnússon, H. Shokri-Ghadikolaei, and N. Li, "On maintaining linear convergence of distributed learning and optimization under limited communication," *IEEE Trans. Signal Process.*, vol. 68, pp. 6101–6116, 2020.

[102] Y. Kajiyama, N. Hayashi, and S. Takai, "Linear convergence of consensus-based quantized optimization for smooth and strongly convex cost functions," *IEEE Trans. Autom. Control*, vol. 66, no. 3, pp. 1254–1261, Mar. 2021.

[103] C.-S. Lee, N. Michelusi, and G. Scutari, "Finite rate quantized distributed optimization with geometric convergence," in *Proc. 52nd Asilomar Conf. Signals, Syst., Comput.*, Oct. 2018, pp. 1876–1880.

[104] P. Yi and Y. Hong, "Quantized subgradient algorithm and data-rate analysis for distributed optimization," *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 4, pp. 380–392, Dec. 2014.

[105] H. Li, C. Huang, G. Chen, X. Liao, and T. Huang, "Distributed consensus optimization in multiagent networks with time-varying directed topologies and quantized communication," *IEEE Trans. Cybern.*, vol. 47, no. 8, pp. 2044–2057, Aug. 2017.

[106] Y. Du, S. Yang, and K. Huang, "High-dimensional stochastic gradient quantization for communication-efficient edge learning," *IEEE Trans. Signal Process.*, vol. 68, pp. 2128–2142, 2020.

[107] Y. Yu, J. Wu, and L. Huang, "Double quantization for communication-efficient distributed optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 4438–4449.

[108] M. Zhang, L. Chen, A. Mokhtari, H. Hassani, and A. Karbasi, "Quantized Frank-Wolfe: Communication-efficient distributed optimization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 3696–3706.

[109] X. Cao and T. Basar, "Decentralized multi-agent stochastic optimization with pairwise constraints and quantized communications," *IEEE Trans. Signal Process.*, vol. 68, pp. 3296–3311, 2020.

[110] O. A. Hanna, Y. H. Ezzeldin, C. Fragouli, and S. Diggavi, "Quantization of distributed data for learning," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 3, pp. 987–1001, Sep. 2021.

[111] W. Wen et al., "TernGrad: Ternary gradients to reduce communication in distributed deep learning," in *Adv. Neural Inf. Process. Syst.*, 2017, pp. 1508–1518.

[112] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar, "SignSGD: Compressed optimisation for non-convex problems," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 560–569.

[113] G. Zhu, Y. Du, D. Gunduz, and K. Huang, "One-bit over-the-air aggregation for communication-efficient federated edge learning: Design and convergence analysis," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 2120–2135, Mar. 2021.

[114] J. Zhang, K. You, and T. Başar, "Distributed discrete-time optimization in multiagent networks using only sign of relative state," *IEEE Trans. Autom. Control*, vol. 64, no. 6, pp. 2352–2367, Jun. 2019.

[115] X. Cao and T. Başar, "Decentralized online convex optimization based on signs of relative states," *Automatica*, vol. 129, Jul. 2021, Art. no. 109676.

[116] Y. Yang, Z. Zhang, and Q. Yang, "Communication-efficient federated learning with binary neural networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3836–3850, Dec. 2021.

[117] M. Chen, N. Shlezinger, H. V. Poor, Y. C. Eldar, and S. Cui, "Communication-efficient federated learning," *Proc. Nat. Acad. Sci. USA*, vol. 118, no. 17, pp. 1–8, 2021.

[118] S. Horváth, D. Kovalev, K. Mishchenko, S. Stich, and P. Richtárik, "Stochastic distributed learning with gradient quantization and variance reduction," 2019, *arXiv:1904.05115*.

[119] A. Reisizadeh, A. Mokhtari, H. Hassani, A. Jadbabaie, and R. Pedarsani, "FedPAQ: A communication-efficient federated learning method with periodic averaging and quantization," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2020, pp. 2021–2031.

[120] S. Zheng, C. Shen, and X. Chen, "Design and analysis of uplink and downlink communications for federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2150–2167, Jul. 2021.

[121] N. Agarwal, A. T. Suresh, F. Yu, S. Kumar, and H. B. Mcmahan, "CpSGD: Communication-efficient and differentially-private distributed SGD," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 7575–7586.

[122] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017.

[123] D. Alistarh, T. Hoefler, M. Johansson, N. Konstantinov, S. Khirirat, and C. Renggli, "The convergence of sparsified gradient methods," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 5973–5983.

[124] K. Mishchenko, F. Hanzely, and P. Richtárik, "99% of worker-master communication in distributed optimization is not needed," in *Proc. Conf. Uncertainty Artif. Intell.*, 2020, pp. 979–988.

[125] A. Sahu, A. Dutta, A. M. Abdelmoniem, T. Banerjee, M. Canini, and P. Kalnis, "Rethinking gradient sparsification as total error minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1–14.

[126] S. Shi et al., "A distributed synchronous SGD algorithm with global top-k sparsification for low bandwidth networks," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst.*, Jul. 2019, pp. 2238–2247.

[127] W. Ning et al., "Following the correct direction: Renovating sparsified SGD towards global optimization in distributed edge learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 2, pp. 499–514, Oct. 2022.

[128] M. M. Amiri and D. Gündüz, "Federated learning over wireless fading channels," *IEEE Trans. Wireless Commun.*, vol. 19, no. 5, pp. 3546–3557, May 2020.

[129] M. M. Amiri and D. Gündüz, "Machine learning at the wireless edge: Distributed stochastic gradient descent over-the-air," *IEEE Trans. Signal Process.*, vol. 68, pp. 2155–2169, 2020.

[130] P. Han, S. Wang, and K. K. Leung, "Adaptive gradient sparsification for efficient federated learning: An online learning approach," in *Proc. IEEE 40th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Nov. 2020, pp. 300–310.

[131] Y.-S. Jeon, M. M. Amiri, J. Li, and H. V. Poor, "A compressive sensing approach for federated learning over massive MIMO communication systems," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1990–2004, Mar. 2021.

[132] H. Wang, S. Sievert, Z. Charles, S. Liu, S. Wright, and D. Papailiopoulos, "ATOMO: Communication-efficient learning via atomic sparsification," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 9872–9883.

[133] J. Wangni, J. Wang, J. Liu, and T. Zhang, "Gradient sparsification for communication-efficient distributed optimization," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 1306–1316.

[134] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 2530–2541.

[135] A. Ghosh, R. K. Maity, A. Mazumdar, and K. Ramchandran, "Communication efficient distributed approximate Newton method," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2020, pp. 2539–2544.

[136] Z. Tao and Q. Li, "eSGD: Communication efficient distributed deep learning on the edge," in *Proc. USENIX Workshop Hot Topics Edge Comput.*, 2018.

[137] C. Renggli, S. Ashkboos, M. Aghagolzadeh, D. Alistarh, and T. Hoefler, "SparCML: High-performance sparse communication for machine learning," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, Nov. 2019, pp. 1–15.

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

[139] H. Tang, S. Gan, C. Zhang, T. Zhang, and J. Liu, "Communication compression for decentralized training," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 7663–7673.

[140] N. Singh, D. Data, J. George, and S. Diggavi, "SPARQ-SGD: Event-triggered and compressed communication in decentralized optimization," *IEEE Trans. Autom. Control*, vol. 68, no. 2, pp. 721–736, Feb. 2022.

[141] N. Singh, D. Data, J. George, and S. Diggavi, "SQuARM-SGD: Communication-efficient momentum SGD for decentralized optimization," *IEEE J. Sel. Areas Inf. Theory*, vol. 2, no. 3, pp. 954–969, Sep. 2021.

[142] A. Koloskova, T. Lin, S. U. Stich, and M. Jaggi, "Decentralized deep learning with arbitrary communication compression," in *Proc. Int. Conf. Learn. Represent.*, 2020.

[143] S. U. Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Adv. Neural Inf. Process. Syst.*, 2018, pp. 4452–4463.

[144] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized SGD and its applications to large-scale distributed optimization," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5325–5333.

[145] S. P. Karimireddy, Q. Rebjock, S. Stich, and M. Jaggi, "Error feedback fixes signSGD and other gradient compression schemes," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3252–3261.

[146] D. Basu, D. Data, C. Karakus, and S. Diggavi, "Qsparse-local-SGD: Distributed SGD with quantization, sparsification, and local computations," in *Adv. Neural Inf. Process. Syst.*, 2019, pp. 1–12.

[147] E. Gorbunov, D. Kovalev, D. Makarenko, and P. Richtárik, "Linearly converging error compensated SGD," in *Adv. Neural Inf. Process. Syst.*, 2020, pp. 1–12.

[148] X. Qian, P. Richtárik, and T. Zhang, "Error compensated distributed SGD can be accelerated," in *Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–13.

[149] H. Tang, Y. Li, J. Liu, and M. Yan, "ErrorCompensatedX: Error compensation for variance reduced algorithms," in *Adv. Neural Inf. Process. Syst.*, 2021, pp. 1–12.

[150] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu, "DoubleSqueeze: Parallel stochastic gradient descent with double-pass error-compensated compression," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6155–6165.

[151] S. Khirirat, S. Magnusson, and M. Johansson, "Compressed gradient methods with hessian-aided error compensation," *IEEE Trans. Signal Process.*, vol. 69, pp. 998–1011, 2021.

[152] N. Singh, X. Cao, S. Diggavi, and T. Basar, "Decentralized multi-task stochastic optimization with compressed communications," 2021, *arXiv:2112.12373*.

[153] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. and Bacon, "Federated learning: Strategies for improving communication efficiency," in *Proc. NIPS Workshop Private Multi-Party Mach. Learn.*, 2016.

[154] T. Vogels, S. P. Karinireddy, and M. Jaggi, "PowerSGD: Practical low-rank gradient compression for distributed optimization," in *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–10.

[155] D. Rothchild et al., "FetchSGD: Communication-efficient federated learning with sketching," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8253–8265.

[156] Y. Lin et al., "A communication-efficient multi-agent actor-critic algorithm for distributed reinforcement learning," in *Proc. IEEE 58th Conf. Decis. Control*, Dec. 2019, pp. 5562–5567.

[157] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Trans. Wireless Commun.*, vol. 19, no. 3, pp. 2022–2035, Mar. 2020.

[158] G. Zhu, Y. Wang, and K. Huang, "Broadband analog aggregation for low-latency federated edge learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 491–506, Jan. 2019.

[159] T. Sery, N. Shlezinger, K. Cohen, and Y. Eldar, "Over-the-air federated learning from heterogeneous data," *IEEE Trans. Signal Process.*, vol. 69, pp. 3796–3811, 2021.

[160] J. Zhang, N. Li, and M. Dedeoglu, "Federated learning over wireless networks: A band-limited coordinated descent approach," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, May 2021, pp. 1–10.

[161] C. Xu, S. Liu, Z. Yang, Y. Huang, and K.-K. Wong, "Learning rate optimization for federated learning exploiting over-the-air computation," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3742–3756, Dec. 2021.

[162] Z. Yang, M. Chen, W. Saad, C. S. Hong, and M. Shikh-Bahaei, "Energy efficient federated learning over wireless communication networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 3, pp. 1935–1949, Mar. 2020.

[163] N. H. Tran, W. Bao, A. Zomaya, M. N. H. Nguyen, and C. S. Hong, "Federated learning over wireless networks: Optimization model design and analysis," in *Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM)*, Apr. 2019, pp. 1387–1395.

[164] C. T. Dinh et al., "Federated learning over wireless networks: Convergence analysis and resource allocation," *IEEE/ACM Trans. Netw.*, vol. 29, no. 1, pp. 398–409, Nov. 2021.

[165] X. Cao, G. Zhu, J. Xu, Z. Wang, and S. Cui, "Optimized power control design for over-the-air federated edge learning," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 342–358, Jan. 2022.

[166] Y. Sun, S. Zhou, Z. Niu, and D. Gunduz, "Dynamic scheduling for over-the-air federated edge learning with energy constraints," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 227–242, Jan. 2022.

[167] B. Luo, X. Li, S. Wang, J. Huang, and L. Tassiulas, "Cost-effective federated learning in mobile edge networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3606–3621, Dec. 2021.

[168] Q. Ma, Y. Xu, H. Xu, Z. Jiang, L. Huang, and H. Huang, "FedSA: A semi-asynchronous federated learning mechanism in heterogeneous edge computing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3654–3672, Dec. 2021.

[169] J. Xu and H. Wang, "Client selection and bandwidth allocation in wireless federated learning networks: A long-term perspective," *IEEE Trans. Wireless Commun.*, vol. 20, no. 2, pp. 1188–1200, Feb. 2020.

[170] M. Chen, H. V. Poor, W. Saad, and S. Cui, "Convergence time optimization for federated learning over wireless networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 4, pp. 2457–2471, Apr. 2020.

[171] W. Shi, S. Zhou, Z. Niu, M. Jiang, and L. Geng, "Joint device scheduling and resource allocation for latency constrained wireless federated learning," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 453–467, Jan. 2020.

[172] S. Wan, J. Lu, P. Fan, Y. Shao, C. Peng, and K. B. Letaief, "Convergence analysis and system design for federated learning over wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3622–3639, Dec. 2021.

[173] K. Wei et al., "Low-latency federated learning over wireless channels with differential privacy," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 1, pp. 290–307, Nov. 2022.

[174] W. Huang, Y. Yang, M. Chen, C. Liu, C. Feng, and H. V. Poor, "Wireless network optimization for federated learning with model compression in hybrid VLC/RF systems," *Entropy*, vol. 23, no. 11, p. 1413, Oct. 2021.

[175] H.-S. Lee and J.-W. Lee, "Adaptive transmission scheduling in wireless networks for asynchronous federated learning," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3673–3687, Dec. 2021.

[176] M. S. H. Abad, E. Ozfatura, D. Gunduz, and O. Ercetin, "Hierarchical federated learning ACROSS heterogeneous cellular networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 8866–8870.

[177] W. Zhang et al., "Optimizing federated learning in distributed industrial IoT: A multi-agent approach," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3688–3703, Dec. 2021.

[178] Z. Wang et al., "Federated learning via intelligent reflecting surface," *IEEE Trans. Wireless Commun.*, vol. 21, no. 2, pp. 808–822, Feb. 2022.

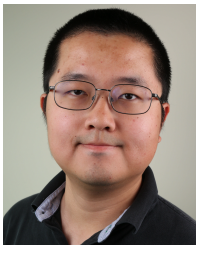
[179] L. Li et al., "Delay analysis of wireless federated learning based on saddle point approximation and large deviation theory," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3772–3789, Dec. 2021.

[180] J. Kang, Z. Xiong, D. Niyato, S. Xie, and J. Zhang, "Incentive mechanism for reliable federated learning: A joint optimization approach to combining reputation and contract theory," *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10700–10714, Dec. 2019.

[181] Y. Zhan, P. Li, Z. Qu, D. Zeng, and S. Guo, "A learning-based incentive mechanism for federated learning," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6360–6368, Jul. 2020.

[182] Y. Jiao et al., "Toward an automated auction framework for wireless federated learning services market," *IEEE Trans. Mobile Comput.*, vol. 20, no. 10, pp. 3034–3048, Oct. 2021.

[183] W. Y. B. Lim, J. S. Ng, Z. Xiong, D. Niyato, C. Miao, and D. I. Kim, "Dynamic edge association and resource allocation in self-organizing hierarchical federated learning networks," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3640–3653, Dec. 2021.



**Xuanyu Cao** (Senior Member, IEEE) received the B.S. degree in electrical engineering from Shanghai Jiao Tong University, in 2013, and the M.S. and Ph.D. degrees in electrical engineering from the University of Maryland, College Park, in 2016 and 2017, respectively. From August 2017 to October 2021, he was successively a Postdoctoral Research Associate with the Department of Electrical Engineering, Princeton University, and the Coordinated Science Laboratory, University of Illinois, Urbana–Champaign. Since October 2021, he has been an Assistant Professor with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong. His research interests encompass distributed/online optimization, communication-efficient distributed learning, federated learning over wireless networks, and game theory and network economics. He is a TPC member of ACM MobiHoc 2022 and 2023. He is an Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY and the Lead Guest Editor of the Special Issue on Communication-Efficient Distributed Learning over Networks in IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS.

**Tamer Başar** (Life Fellow, IEEE) received the B.S.E.E. degree from the Robert College, Istanbul, and the M.S., M.Phil., and Ph.D. degrees from Yale University. He has been with the University of Illinois Urbana–Champaign, since 1981, where he is currently a Swanlund Endowed Chair Emeritus and the Center for Advanced Study (CAS) Professor Emeritus of electrical and computer engineering, with also affiliations with the Coordinated Science Laboratory, Information Trust Institute, and Mechanical Science and Engineering. At Illinois, he has also served as the Director of CAS (2014–2020), an Interim Dean of engineering (2018), and an Interim Director of the Beckman Institute (2008–2010). He has over 1000 publications in systems, control, communications, optimization, networks, and dynamic games, including books on non-cooperative dynamic game theory, robust control, network security, wireless and communication networks, and stochastic networked control. His current research interests include stochastic teams, games, networks, risk-sensitive estimation and control, mean-field game theory, multi-agent systems and learning, data-driven distributed optimization, epidemics modeling and control over networks, strategic information transmission, spread of disinformation, deception, security and trust, energy systems, and cyber-physical systems. He is a member of the U.S. National Academy of Engineering and a fellow of IFAC and SIAM. He received the Wilbur Cross Medal in 2021. He has received several awards and recognitions over the years, including the highest awards of IEEE CSS, IFAC, AACC, and ISDG, the IEEE Control Systems Award, and a number of international honorary doctorates and professorships. He has served as the President of IEEE Control Systems Society (CSS), International Society of Dynamic Games (ISDG), and American Automatic Control Council (AACC). He was the Editor-in-Chief of *Automatica*, from 2004 to 2014. He is currently an editor of several book series.

**Suhas Diggavi** (Fellow, IEEE) received the bachelor's degree from IIT, Delhi, and the Ph.D. degree from Stanford University.

He is currently a Professor of electrical and computer engineering at UCLA. He has worked as a Principal Member Research Staff at AT&T Shannon Laboratories and directed the Laboratory for Information and Communication Systems (LICOS), EPFL. At UCLA, he directs the Information Theory and Systems Laboratory. He has eight issued patents. His research interests include information theory and its applications to several areas, including machine learning, security and privacy, wireless networks, data compression, cyber-physical systems, bioinformatics, and neuroscience.

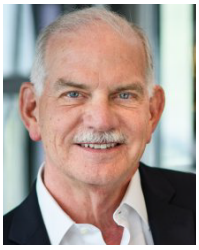
Dr. Diggavi was selected as a Guggenheim fellow in 2021. He has received several recognitions for his research from IEEE and ACM, including the 2013 IEEE Information Theory Society and Communications Society Joint Paper Award, the 2021 ACM Conference on Computer and Communications Security (CCS) Best Paper Award, the 2013 ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc) Best Paper Award, and the 2006 IEEE Donald Fink Prize Paper Award. He also received the 2019 Google Faculty Research Award, the 2020 Amazon Faculty Research Award, and the 2021 Facebook/Meta Faculty Research Award. He has helped to organize IEEE and ACM conferences, including serving as the Technical Program Co-Chair for 2012 IEEE Information Theory Workshop (ITW), the Technical Program Co-Chair for the 2015 IEEE International Symposium on Information Theory (ISIT), and General Co-Chair for ACM Mobihoc 2018. He was an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY, *ACM/IEEE Transactions on Networking*, other journals and special issues, and the program committees of several conferences. He served as an IEEE Distinguished Lecturer and also served on Board of Governors for the IEEE Information Theory Society (2016–2021). More information can be found at http://licos.ee.ucla.edu.

**Yonina C. Eldar** (Fellow, IEEE) received the B.Sc. degree in physics and electrical engineering from Tel Aviv University (TAU), Tel Aviv, Israel, in 1995 and 1996, respectively, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 2002. She is currently a Professor with the Department of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel, where she heads the Center for Biomedical Engineering. She is also a Visiting Professor at MIT, a Visiting Scientist at the Broad Institute, an Adjunct Professor at Duke University, and a Visiting Professor at Stanford. She is a member of the Israel Academy of Sciences and Humanities and a EURASIP fellow. She was a member of the Young Israel Academy of Science and Humanities and the Israel Committee for Higher Education. She was a Horev Fellow of the Leaders in Science and Technology Program at the Technion and was selected as one of the 50 most influential women in Israel and Asia. She has received many awards for excellence in research and teaching, including the IEEE Signal Processing Society Technical Achievement Award, in 2013, the IEEE/AESS Fred Nathanson Memorial Radar Award, in 2014, and the IEEE Kiyo Tomiyasu Award, in 2016. She is the Editor-in-Chief of *Foundations and Trends in Signal Processing* and serves on many IEEE committees.

**Khaled B. Letaief** (Fellow, IEEE) received the B.S. (Hons.), M.S., and Ph.D. degrees in electrical engineering from Purdue University, West Lafayette, IN, USA, in December 1984, August 1986, and May 1990, respectively, the Ph.D. degree (honoris causa) from the University of Johannesburg, South Africa, in 2022.

He is an internationally recognized Leader in wireless communications and networks. From 1990 to 1993, he was a Faculty Member at the University of Melbourne, Australia. Since 1993, he has been with The Hong Kong University of Science and Technology (HKUST), where he has held many administrative positions, including an Acting Provost, the Dean of Engineering, the Head of the Electronic and Computer Engineering Department, the Director of the Wireless IC Design Center, the founding Director of the Huawei Innovation Laboratory, and the Director of the Hong Kong Telecom Institute of Information Technology. From September 2015 to March 2018, he joined HBKU, as a Provost as help to establish a research-intensive university in Qatar, in partnership with strategic partners, that include Northwestern University, Carnegie Mellon University, Cornell, and Texas A&M.

Dr. Letaief is a member of the U.S. National Academy of Engineering, a fellow of the Hong Kong Institution of Engineers, and a member of the India National Academy of Sciences and the Hong Kong Academy of Engineering Sciences. He is currently serving as a member for the IEEE Board of Directors. He was a recipient of many distinguished awards and honors, including the 2022 IEEE Communications Society Edwin Howard Armstrong Achievement Award, the 2021 IEEE Communications Society Best Survey Paper Award, the 2019 IEEE Communications Society and Information Theory Society Joint Paper Award, the 2016 IEEE Marconi Prize Paper Award in Wireless Communications, the 2011 IEEE Communications Society Harold Sobol Award, the 2010 Purdue University Outstanding Electrical and Computer Engineer Award, the 2007 IEEE Communications Society Joseph LoCicero Publications Exemplary Award, and over 19 IEEE Best Paper Awards. He is also recognized by Thomson Reuters as an ISI Highly Cited Researcher and was listed among the 2020 top 30 of AI 2000 Internet of Things Most Influential Scholars. He also served as the President for the IEEE Communications Society (2018–2019), the world's leading organization for communications professionals with headquarter in New York City and members in 162 countries. He is well recognized for his dedicated service to professional societies and IEEE, where he has served in many leadership positions. These include the founding Editor-in-Chief of the prestigious IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS.



**H. Vincent Poor** (Life Fellow, IEEE) received the Ph.D. degree in EECS from Princeton University in 1977. From 1977 to 1990, he was on the faculty of the University of Illinois at Urbana–Champaign. Since 1990, he has been on the faculty at Princeton, where he is currently the Michael Henry Strater University Professor. From 2006 to 2016, he served as the Dean of Princeton's School of Engineering and Applied Science. He has also held visiting appointments at several other universities, including most recently at Berkeley and Cambridge. His research interests are in the areas of information theory, machine learning and network science, and their applications in wireless networks, energy systems, and related fields. Among his publications in these areas is the recent book *Machine Learning and Wireless Communications* (Cambridge University Press in 2022). He is a member of the National Academy of Engineering and the National Academy of Sciences; and a foreign member of the Chinese Academy of Sciences, the Royal Society, and other national and international academies. He received the IEEE Alexander Graham Bell Medal in 2017.

**Junshan Zhang** (Fellow, IEEE) received the Ph.D. degree from the School of ECE, Purdue University, in August 2000. He was on the Faculty of the School of ECEE, Arizona State University, from 2000 to 2021. He is a Professor at the ECE Department, University of California Davis. His research interests include information networks and data science, including edge intelligence, reinforcement learning, continual learning, network optimization and control, game theory, with applications in connected and automated vehicles, 5G and beyond, wireless networks, the IoT data privacy/security, and smart grid.

He was a recipient of the ONR Young Investigator Award in 2005 and the NSF CAREER award in 2003. He also received the IEEE Wireless Communication Technical Committee Recognition Award in 2016. His papers have won a few awards, including the Best Student Paper award at WiOPT 2018, the Kenneth C. Sevcik Outstanding Student Paper Award of ACM SIGMETRICS/IFIP Performance 2016, the Best Paper Runner-Up Award of IEEE INFOCOM 2009 and IEEE INFOCOM 2014, and the Best Paper Award at IEEE ICC 2008 and ICC 2017. He was a TPC Co-Chair of a number of major conferences in communication networks, including IEEE INFOCOM 2012 and ACM MOBIHOC 2015. He was the General Chair of ACM/IEEE SEC 2017 and WiOPT 2016. Building on his research findings, he has Co-Founded Smartiply Inc., a Fog Computing startup company, delivering boosted network connectivity and embedded artificial intelligence. He is currently serving as the Editor-in-chief for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and a Senior Editor for IEEE/ACM TRANSACTIONS ON NETWORKING. He was a Distinguished Lecturer of the IEEE Communications Society.