Online Distributed Optimization with Efficient Communication via Temporal Similarity

Juncheng Wang*, Ben Liang*, Min Dong†, Gary Boudreau‡, and Ali Afana‡
*Department of Electrical and Computer Engineering, University of Toronto, Canada,
†Department of Electrical, Computer and Software Engineering, Ontario Tech University, Canada, ‡Ericsson Canada, Canada

***Abstract*—We consider online distributed optimization in a networked system, where multiple devices assisted by a server collaboratively minimize the accumulation of a sequence of global loss functions that can vary over time. To reduce the amount of communication, the devices send quantized and compressed local decisions to the server, resulting in noisy global decisions. Therefore, there exists a tradeoff between the optimization performance and the communication overhead. Existing works separately optimize computation and communication. In contrast, we jointly consider computation and communication over time, by encouraging temporal similarity in the decision sequence to control the communication overhead. We propose an efficient algorithm, termed Online Distributed Optimization with Temporal Similarity (ODOTS), where the local decisions are both computation- and communication-aware. Furthermore, ODOTS uses a novel tunable virtual queue, which completely removes the commonly assumed Slater's condition through a modified Lyapunov drift analysis. ODOTS delivers provable performance bounds on both the optimization objective and constraint violation. As an example application, we apply ODOTS to enable communication-efficient federated learning. Our experimental results based on real-world image classification demonstrate that ODOTS obtains higher classification accuracy and lower communication overhead compared with the current best alternatives for both convex and non-convex loss functions.**

I. Introduction

Distributed optimization has become an essential tool for modern machine learning applications, which require ample storage, computation, and data. It avoids overburdening any single server and is robust to failures by coordinating multiple local devices to process the machine learning tasks. It can also alleviate privacy concerns by keeping the data local. However, the migration of optimization from the central server to local devices can incur a surge of communication overhead between them [1], [2]. This calls for *communication-efficient* distributed optimization [3]. Most existing works on communication-efficient distributed learning consider computation and communication *separately* [4]-[17], *i.e.,* communication designs such as quantization and compression come *after* the machine-learning model parameters are already determined, for example, by standard gradient descent. However, since communication efficiency is strongly dependent on the information being transmitted [18], one can further improve the learning performance by proactively designing the model parameters for both learning accuracy and communication efficiency. In other words, *joint* consideration of computation and communication would take into fuller account the mutual impact between them.

Furthermore, most existing works focus on *offline* optimization, which does not allow time-varying loss functions or account for any long-term constraints. However, in many practical machine learning applications, *e.g.,* network traffic classification [19], dynamic user profiling [20], and real-time video analysis [21], random data samples arrive in a streaming fashion, and consequently the loss functions vary over time. These applications require *online optimization*, where we compute a sequence of optimization decisions that are adaptive to the unpredictable system dynamics over time [22], [23].

This motivates us to pose the following key question: *How to design an online distributed optimization algorithm that jointly considers computation and communication over time?* In particular, we are interested in a design that takes into account the interdependence of the optimization decisions over time to reduce the communication overhead, while providing performance guarantees on both the optimization and communication performance metrics.

To answer the above key question, we must address several challenges: 1) Since the communication overhead depends on the local decisions transmitted from the devices to the server, when updating the local decisions, we must consider both their optimization performance and communication cost. 2) Lossy quantization substantially reduces the communication overhead but at the same time generates errors in the optimization decisions, and these errors propagate in the iterative computation process over time. 3) Due to the tight coupling between computation and communication, we must properly balance their joint impact on both the optimization performance and the convergence speed. 4) Both computation and communication needs to be properly formulated and designed to account for the unpredictable fluctuations in the environment over time.

In this context, the contributions of this paper are as follows:

- We formulate an online distributed optimization problem where the server computes a sequence of global optimization decisions to minimize the accumulated global loss, by aggregating the quantized and compressed local decisions communicated from the devices. To reduce the communication overhead, we encourage temporal similarity in the computed sequence of local decisions at the devices by enforcing an average long-term decision dis-similarity constraint. Thus, we consider both the

This work has been funded in part by Ericsson Canada and by the Natural Sciences and Engineering Research Council (NSERC) of Canada. The authors have provided public access to their code or data at https://github.com/juncheng-wang/INFOCOM2023-ODOTS.

optimization and communication performance metrics. To the best of our knowledge, this form of online distributed optimization with joint computation and communication consideration has not been studied in the literature.

- We propose an efficient algorithm to solve this problem, termed Online Distributed Optimization with Temporal Similarity (ODOTS). The local decisions yielded by ODOTS are adaptive to the unpredictable fluctuations of the loss functions while accounting for the decision dissimilarity constraint violation to limit the communication overhead. ODOTS achieves this via a novel tunable virtual queue that requires a modified Lyapunov drift analysis technique. Notably, this removes the requirement for Slater's condition, which is commonly assumed in existing virtual-queue-based online optimization algorithms.
- We analyze the tight coupling between computation and communication, and their joint impact on the optimization performance and convergence speed of ODOTS. Our analysis shows that for all sequences of time-varying weights on the devices, ODOTS achieves $\mathcal{O}(\max\{T^{\frac{1+\mu}{2}}, T^{\frac{3+\nu}{4}}\})$ performance gap to the centralized per-slot optimal decision sequence and $\mathcal{O}(\max\{T^{\frac{3+\mu}{4}}, T^{\frac{7+\nu}{8}}\})$ violation of the long-term decision dis-similarity constraint over T time slots, where μ represents the growth rate of the centralized per-slot optimizer and the quantization error, and ν measures the accumulated variation of the time-varying weights.
- As an example application, we apply ODOTS to enable communication-efficient federated learning. We study the impact of system parameters on the performance of ODOTS, by experimenting with real-world image classification datasets. Our experimental results demonstrate that for both convex and non-convex loss functions, ODOTS obtains higher test accuracy with lower communication overhead, compared with the current best alternatives under different scenarios.

II. Related Work

A. Communication Efficiency in Distributed Optimization

Distributed optimization has been widely studied (see [24] and references therein). For example, offline distributed dual averaging and mirror descent algorithms were proposed in [25] and [26]. These two algorithms were respectively extended in [27] and [28] to the online setting. However, these works do not explicitly consider the communication efficiency.

Distributed approximate Newton-typed algorithm and alternating direction method of multipliers algorithm were proposed in [29] and [30] to reduce the number of iterations for efficient communication. Distributed gradient descent with event-trigger communication was considered in [31]. A general communication-efficient distributed dual coordinate ascent framework was proposed in [32], which used local computation in a primal-dual setting for reduced communication. However, the above works all assume error-free communication, and they ignore the opportunity to reduce the communication overhead via information similarity.

B. Communication-Efficient Distributed Learning

The original federated averaging algorithm increases the number of local updates to reduce the communication overhead [4]. An adaptive model aggregation approach was proposed in [5] under communication resource constraints. Quantization schemes have been adopted in distributed learning to reduce the number of transmitted bits by mapping the model parameters to a small set of discrete values. For example, 1-bit and multi-bit quantization methods were developed in [6] and [7]. Some other variations include error compensation [8], variance reduction [9], and ternary quantization [10]. Sparsification schemes select a portion of the model parameters for communication. For example, threshold-based and top-k selection schemes were proposed in [11] and [12]. Quantization and sparsification have also been applied simultaneously in [13]. However, the above works do not utilize the model similarity for more efficient communication.

Model similarity was utilized in [14] to further reduce the number of transmitted bits via conditional entropy coding. By using the autoencoder technique originally proposed for image compression, model compression was trained in [15]. Scalable sparsified model compression in combination with error-correction techniques was proposed in [16]. An innovation-based quantization scheme was proposed in [17]. However, the above works have the following fundamental limitations: 1) Their separate consideration of model training and compression overlooks the opportunity to select model parameters that can improve the communication efficiency; 2) Their offline optimization does not fully account for the unpredictable system variations during the learning process.

There is a recent branch of federated learning that utilizes analog communication, where model aggregation can be conducted over the air to reduce latency and communication overhead. For example, the aggregation error caused by noisy channel and model quantization was minimized through power allocation at each iteration in [33]. Online model updating under long-term power constraints was considered in [34]. However, over-the-air model aggregation requires strict symbol-level synchronization among the devices and a large number of subchannels to separately communicate each of the model parameters. It is outside the scope of this work, which is designed for the common digital communication system.

C. Online Convex Optimization and Lyapunov Optimization

Due to the dynamic nature of the iterative computation and communication over time, a part of our solution resembles online convex optimization (OCO) [23], especially distributed constrained OCO with consensus [35]-[39]. However, the OCO framework mainly concerns delayed information feedback with error-free communication, which is inherently different from the joint online computation and communication framework in this work.

Since our work considers online optimization with a long-term constraint, it is also related to Lyapunov optimization [40], which minimizes a weighted sum of the loss and constraint functions at each time. However, directly minimizing

the loss function can be difficult; *e.g.,* in distributed learning, it means directly solving for the optimal global model. Furthermore, ODOTS is a gradient-descent-typed algorithm, which substantially differs from Lyapunov optimization.

III. System Model and Problem Formulation

A. Online Distributed Optimization Objective

Consider a networked system consists of N local devices and a server. The system operates in a time-slotted fashion with time indexed by t. Let $f_t^n(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ be the *local* loss function of device n at time t, which may change over time. We are interested in an *online distributed* optimization problem with a *global* loss function $f_t(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}$ at each time t. It is defined as the weighted average of the local loss functions $\{f_t^n(\mathbf{x})\}$, given by

$$f_t(\mathbf{x}) \triangleq \sum_{n=1}^{N} w_t^n f_t^n(\mathbf{x}) \tag{1}$$

where $w_t^n \geq 0$ is the weight of device n, and satisfies $\sum_{n=1}^N w_t^n = 1$. Note that we also allow w_t^n to vary over time. The goal of online distributed optimization is to compute at the server a sequence of global decisions $\{\mathbf{x}_t\}$ that minimizes the accumulated global loss over a finite time horizon T, *i.e.,*

$$\min_{\{\mathbf{x}_t\}} \quad \sum_{t=1}^{T} f_t(\mathbf{x}_t). \tag{2}$$

As an example, in distributed learning, random training data may arrive at the devices over time as a continuous stream. At each time t, each device n collects its *local* dataset denoted by $\mathcal{D}_t^n$. The i-th data sample in $\mathcal{D}_t^n$ is represented by $(\mathbf{u}_t^{n,i}, v_t^{n,i})$, where $\mathbf{u}_t^{n,i}$ is a data feature vector and $v_t^{n,i}$ is its true label. Let $l(\mathbf{x}; \mathbf{u}_t^{n,i}, v_t^{n,i}) : \mathbb{R}^d \to \mathbb{R}$ be a training loss function to indicate how the learning model $\mathbf{x} \in \mathbb{R}^d$ performs on each data sample $(\mathbf{u}_t^{n,i}, v_t^{n,i})$, *e.g.,* it can be defined as the cross-entropy loss for logistic regression (see Section VI-B). In this case, the local loss function $f_t^n(\mathbf{x})$ is the averaged losses of the data samples in $\mathcal{D}_t^n$, given by

$$f_t^n(\mathbf{x}) = \frac{1}{|\mathcal{D}_t^n|} \sum_{i=1}^{|\mathcal{D}_t^n|} l(\mathbf{x}; \mathbf{u}_t^{n,i}, v_t^{n,i}) \tag{3}$$

where $|\mathcal{D}_t^n|$ is the cardinality of $\mathcal{D}_t^n$. When we set the local weight as $w_t^n = \frac{|\mathcal{D}_t^n|}{\sum_{m=1}^N |\mathcal{D}_t^m|}$ for each device n, the global loss $f_t(\mathbf{x})$ in (1) is equivalent to the averaged losses incurred by the global dataset $\bigcup_{n=1}^N \{\mathcal{D}_t^n\}$. Note that due to the fluctuations of the available computation resources, each device n may process different amounts of data samples over time, leading to a sequence of time-varying weights $\{w_t^n\}$.

B. Local Decision Quantization and Compression

For distributed minimization of the accumulated global loss, each device n generates a sequence of its local decisions $\{\mathbf{x}_t^n\}$. The server aggregates the local decisions into a sequence of global decisions. Transmitting the local decisions $\{\mathbf{x}_t^n\}$ from the N devices to the server can cause a large amount of communication overhead. This can be challenging and time-consuming, *e.g.,* for neural network training in the wireless environment, which can include millions of model parameters in each $\mathbf{x}_t^n$. In practical systems, communicating the local decisions from the devices to the server has been observed to be a significant performance bottleneck [1]-[3].

For efficient communication, the local decisions are usually quantized before transmission to the server. At each time t, after obtaining the local decision $\mathbf{x}_t^n$, each device n generates a quantized local decision $\hat{\mathbf{x}}_t^n$, by projecting each element of $\mathbf{x}_t^n$ to its closest point in a uniformly distributed grid with $s = 2^b$ quantization levels, where b is the quantization bit length.[1] In particular, the i-th element $x_t^{n,i}$ of $\mathbf{x}_t^n$ is quantized as $\hat{x}_t^{n,i}$, given by

$$\hat{x}_t^{n,i} = x_{\max} \cdot \text{sign}(x_t^{n,i}) \cdot \text{map}(x_t^{n,i}; x_{\max}, s) \tag{4}$$

where $x_{\max}$ is the maximum decision value, $\text{sign}(x) \in \{-1, 1\}$ returns the sign of x with $\text{sign}(0) = 1$, and

$$\text{map}(x; x_{\max}, s) = \left\lfloor \frac{|x|}{x_{\max}} \cdot (s-1) + \frac{1}{2} \right\rfloor \tag{5}$$

with $\lfloor a \rfloor$ being the floor function. Note that $x_{\max}$ can be easily enforced to the decision parameters by setting a set of short-term constraints on $\mathbf{x}_t^n$, given by

$$\mathcal{X} \triangleq \{\mathbf{x} : -x_{\max}\mathbf{1} \preceq \mathbf{x} \preceq x_{\max}\mathbf{1}\} \tag{6}$$

with $\mathbf{1}$ being a vector of all 1's.

Communicating the quantized local decisions requires efficient encoding to convert $\hat{\mathbf{x}}_t^n$ into bit streams. There are two common encoding approaches to compress $\hat{\mathbf{x}}_t^n$: 1) simple encoding that does not utilize any correlation in the sequence of decisions, such as Elias coding [41] and entropy coding [42]; and 2) more complicated encoding approach that utilizes the decision similarity, such as Wyner-Ziv coding [43] and conditional entropy coding [44], [45]. For example, consider the ideal conditional entropy coding. Let $H(\hat{\mathbf{x}}_t^n)$ be the marginal entropy of $\hat{\mathbf{x}}_t^n$. It measures the number of bits to communicate $\hat{\mathbf{x}}_t^n$ using entropy coding. Let $H(\hat{\mathbf{x}}_t^n | \hat{\mathbf{x}}_{t-1}^n)$ be the conditional entropy of $\hat{\mathbf{x}}_t^n$ given $\hat{\mathbf{x}}_{t-1}^n$, which represents the number of bits to communicate $\hat{\mathbf{x}}_t^n$ using conditional entropy coding, when $\hat{\mathbf{x}}_{t-1}^n$ is known at the destination. Due to the correlation between $\hat{\mathbf{x}}_{t-1}^n$ and $\hat{\mathbf{x}}_t^n$, their mutual information $H(\hat{\mathbf{x}}_t^n) - H(\hat{\mathbf{x}}_t^n | \hat{\mathbf{x}}_{t-1}^n)$ can be high. Therefore, conditional entropy coding can substantially reduce the communication overhead compared with independent entropy coding [44], [45].

The quantized and compressed local decisions are losslessly conveyed to the server through standard channel coding techniques. However, due to lossy quantization, the server can only compute a *noisy* global decision $\hat{\mathbf{x}}_{t+1}$, given by

$$\hat{\mathbf{x}}_{t+1} = \sum_{n=1}^{N} w_t^n \hat{\mathbf{x}}_t^n = \mathbf{x}_{t+1} + \mathbf{n}_{t+1} \tag{7}$$

[1]Other techniques may be combined to further reduce the communication overhead. For example, each device n can first perform *sparsification* and then quantization to generate $\hat{\mathbf{x}}_t^n$. It will cause additional errors to the global decision $\hat{\mathbf{x}}_{t+1}$ (7). However, these errors can be included in $\mathbf{n}_{t+1}$ and do not impact our performance analysis later.

where $\mathbf{x}_{t+1} = \sum_{n=1}^{N} w_t^n \mathbf{x}_t^n$ is the *noiseless* global decision and $\mathbf{n}_{t+1} = \hat{\mathbf{x}}_{t+1} - \mathbf{x}_{t+1}$ is the global quantization error. The server then broadcasts $\hat{\mathbf{x}}_{t+1}$ to all N devices, and each device uses $\hat{\mathbf{x}}_{t+1}$ and its local loss function at time $t+1$ to compute the next local decision $\mathbf{x}_{t+1}^n$.

For ease of exposition, we assume the server uses standard channel coding techniques, such that $\hat{\mathbf{x}}_{t+1}$ can be received by all devices in an error-free fashion. However, lossy transmission of $\hat{\mathbf{x}}_{t+1}^n$ can be easily combined with our proposed algorithm and its performance analysis.

C. ODOTS Problem Formulation

Our goal is to jointly consider the global loss minimization and the local decision communication overhead over time. However, it is challenging to directly model a temporal-similarity encoding scheme during decision updating, since it depends on the joint probability density of $\hat{\mathbf{x}}_t^n$ and $\hat{\mathbf{x}}_{t-1}^n$. We observe that for different encoding schemes, an importance measure of the coding length is the difference between the information sources, *e.g.,* $\hat{\mathbf{x}}_t^n - \hat{\mathbf{x}}_{t-1}^n$, as it approximates the amount of new information to be encoded. Further note that the quantized local decision $\hat{\mathbf{x}}_t^n$ is generated only *after* computing the local decision $\mathbf{x}_t^n$. That is to say we can only optimize $\mathbf{x}_t^n$ instead of $\hat{\mathbf{x}}_t^n$ during the decision updating process. Therefore, we resort to limiting the amount of decision dis-similarity $\|\mathbf{x}_t^n - \hat{\mathbf{x}}_{t-1}^n\|^2$ to control the communication overhead, where $\|\cdot\|$ represents the Euclidean norm.

We aim at computing a sequence of local decisions $\{\mathbf{x}_t^n \in \mathcal{X}\}$ to minimize the accumulated loss yielded by the noisy global decision sequence $\{\hat{\mathbf{x}}_t\}$, while ensuring that the average long-term decision dis-similarity constraint is satisfied. This leads to the following online distributed optimization problem:

$$\mathbf{P1}: \quad \min_{\{\mathbf{x}_t^n \in \mathcal{X}\}} \quad \sum_{t=1}^{T} f_t(\hat{\mathbf{x}}_t)$$
$$\text{s.t.} \quad \frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N} g_t^n(\mathbf{x}_t^n) \leq 0 \tag{8}$$

where $\hat{\mathbf{x}}_t$ is the noisy global decision in (7) and $g_t^n(\mathbf{x})$ is the constraint function defined as

$$g_t^n(\mathbf{x}) \triangleq \|\mathbf{x} - \hat{\mathbf{x}}_{t-1}^n\|^2 - \epsilon \tag{9}$$

with ϵ being the allowed average decision dis-similarity.

Note that **P1** is an online optimization problem due to the time-varying loss and constraint functions. In **P1**, the global loss $f_t(\hat{\mathbf{x}}_t)$ is determined by the quantized local decisions $\{\hat{\mathbf{x}}_t^n\}$. The decision dis-similarity constraint $g_t^n(\mathbf{x}_t^n)$ also depends on the quantized local decision $\hat{\mathbf{x}}_{t-1}^n$. Solving **P1** requires simultaneous consideration of computation and communication over time.

Furthermore, compared with the standard error-free optimization problem (2), the additional long-term constraint in (8) of **P1** requires a more complicated *constrained* online distributed optimization algorithm, especially since the local loss functions $\{f_t^n(\mathbf{x})\}$, weights $\{w_t^n\}$, and quantized decisions $\{\hat{\mathbf{x}}_t^n\}$ all can vary over time. It is therefore difficult to obtain the globally optimal solution to **P1**, which would require *centralized* computation with *a priori* information of $\{f_t^n(\mathbf{x})\}$, $\{w_t^n\}$, and $\{\hat{\mathbf{x}}_t^n\}$ over T time slots.

A commonly used *centralized per-slot optimal* solution benchmark $\{\mathbf{x}_t^{\text{ctr}}\}$ for **P1** is given by [39], [46]-[49][2]

$$\mathbf{x}_t^{\text{ctr}} \in \arg\min\{f_t(\mathbf{x}) | g_t^n(\mathbf{x}) \leq 0, \forall n\}. \tag{10}$$

Note that $\mathbf{x}_t^{\text{ctr}}$ is computed without considering any errors, and it requires global information. Furthermore, as explained in Section II-C, directly minimizing $f_t(\mathbf{x})$ as in (10) can be difficult, especially for machine learning tasks. In this work, we aim to develop a constrained online distributed optimization algorithm to compute an online distributed solution sequence $\{\mathbf{x}_t^n\}$ to **P1** with sublinear performance gap to $\{\mathbf{x}_t^{\text{ctr}}\}$, *i.e.,* $\sum_{t=1}^{T}(f_t(\hat{\mathbf{x}}_t) - f_t(\mathbf{x}_t^{\text{ctr}})) = o(T)$, and sublinear constraint violation, *i.e.,* $\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N} g_t^n(\mathbf{x}_t^n) = o(T)$. Sublinearity in performance gap and constraint violation is important; it implies that the online distributed solution approaches to $\{\mathbf{x}_t^{\text{ctr}}\}$ in terms of its time-averaged performance and the long-term constraint is asymptotically satisfied.

IV. Online Distributed Optimization with Temporal Similarity

In this section, we present details of the ODOTS algorithm at the devices and the server. The local decisions yielded by ODOTS are both computation- and communication-aware, and are in closed forms that can be computed efficiently.

A. Tunable Virtual Queue

We first introduce a novel *tunable* virtual queue Q_t^n at each device n to account for the long-term constraint (8) in **P1**, with the following updating rule:

$$Q_{t+1}^n = \left[(1-\gamma^2)Q_t^n + \gamma\eta g_t^n(\mathbf{x}_t^n)\right]_+ \tag{11}$$

where $\gamma \in (0,1)$ is a tuning factor on the virtual queue, $\eta > 0$ is a weighting factor on the constraint function, and $[a]_+ = \max\{a, 0\}$ is a projection operator.[3] The role of Q_t^n is similar to a Lagrangian multiplier for **P1** or a backlog queue for the constraint violation. The concept of virtual queue was also used in [40] and [46]-[49] for Lyapunov optimization and *centralized* constrained OCO. However, unique to our virtual queue updating rule (11), there is an additional $-\gamma^2 Q_t^n$ term to prevent Q_{t+1}^n from becoming too large, and the constraint violation $g_t^n(\mathbf{x}_t^n)$ is scaled by $\gamma\eta$ to control how fast the virtual queue varies over time.

This new tunable virtual queue updating rule (11) will be shown later in Section V-B to provide a simple upper bound on Q_t^n, which does not require the Slater's condition that is commonly assumed for the virtual-queue-based online

[2]The solution benchmark used in [35]-[38] is *fixed* over time.

[3]As will be shown later in Section V-E, η as a constant does not change the growth rate of the performance gap or the constraint violation. However, η can be useful in some numerical experiments as a hyper parameter, especially when the values of the loss and constraint functions differ too much.

Algorithm 1 ODOTS: Device n's algorithm

1: Initialize $\hat{\mathbf{x}}_1 = \mathbf{0}$ and $Q_1^n = 0$. For each t, do:
2: Update local decision $\mathbf{x}_t^n$ by solving $\mathbf{P2}^n$ via (12).
3: Update local virtual queue Q_{t+1}^n via (11).
4: Update quantized local decision $\hat{\mathbf{x}}_t^n$ via (4).
5: Transmit $\hat{\mathbf{x}}_t^n$ via conditional entropy coding.

optimization algorithms [40], [46]-[49].[4] However, without the Slater's condition, we can no longer directly transfer the virtual queue upper bound to the constraint violation bound. To overcome this technical difficulty, as shown later in Section V-B, we will bound the constraint violation using a new modified Lyapunov drift analysis technique.

*B. Decomposition of **P1***

We convert **P1** into a set of local optimization problems, one for each device n at each time t, given by

$$\mathbf{P2}^n: \min_{\mathbf{x}\in\mathcal{X}} \langle \nabla f_t^n(\hat{\mathbf{x}}_t), \mathbf{x}-\hat{\mathbf{x}}_t\rangle + \alpha\|\mathbf{x}-\hat{\mathbf{x}}_t\|^2 + \eta Q_t^n g_t^n(\mathbf{x})$$

where $\alpha > 0$ is a step-size parameter that controls the gradient descent step and $\langle \mathbf{a}, \mathbf{b}\rangle$ represents the inner product of vectors $\mathbf{a}$ and $\mathbf{b}$. Note that $\mathbf{P2}^n$ is a per-device per-slot optimization problem using the current local loss function $f_t^n(\mathbf{x})$, tunable virtual queue length Q_t^n, and the previous quantized local decision $\hat{\mathbf{x}}_{t-1}^n$. It is under short-term constraints only. Furthermore, the local gradient $\nabla f_t^n(\hat{\mathbf{x}}_t)$ is evaluated using the noisy global decision $\hat{\mathbf{x}}_t$ and the regularization $\|\mathbf{x}-\hat{\mathbf{x}}_t\|^2$ is also on $\hat{\mathbf{x}}_t$ to enable local gradient descent based on $\hat{\mathbf{x}}_t$. Compared with the original **P1**, the long-term decision dis-similarity constraint has been converted into controlling $g_t^n(\mathbf{x}_t^n)$ to maintain the queue stability as shown in the third term of the objective in $\mathbf{P2}^n$. The intuition of solving $\mathbf{P2}^n$ is to minimize an upper bound on a modified drift plus penalty plus violation term (see (22) in Section V-B) to trade off loss minimization and constraint violation over time.

Note that the constraint function $g_t^n(\mathbf{x})$ is convex and the feasible set $\mathcal{X}$ is affine with respect to (w.r.t.) $\mathbf{x}$. Furthermore, the first two terms in the objective of $\mathbf{P2}^n$ are affine and convex w.r.t. $\mathbf{x}$, respectively. Therefore, $\mathbf{P2}^n$ is a convex optimization problem and therefore can be solved efficiently.

C. ODOTS Algorithm

In the following, we provide a closed-form solution to $\mathbf{P2}^n$. It is easy to see that the gradient of the objective function of $\mathbf{P2}^n$ is

$$\nabla f_t^n(\hat{\mathbf{x}}_t) + 2\alpha(\mathbf{x}-\hat{\mathbf{x}}_t) + 2\eta Q_t^n(\mathbf{x}-\hat{\mathbf{x}}_{t-1}^n).$$

Then, the optimal solution to $\mathbf{P2}^n$ can be obtained by setting this gradient to zero and then projecting it onto $\mathcal{X}$. The resulting local decision update is in a closed form, given by

$$\mathbf{x}_t^n = \Big[\frac{\alpha}{\alpha+\eta Q_t^n}\Big(\hat{\mathbf{x}}_t + \frac{\eta Q_t^n}{\alpha}\hat{\mathbf{x}}_{t-1}^n - \frac{1}{2\alpha}\nabla f_t^n(\hat{\mathbf{x}}_t)\Big)\Big]_{-x_{\max}\mathbf{1}}^{x_{\max}\mathbf{1}} \quad (12)$$

[4]The Slater's condition precludes dealing with equality constraints and can be restrictive to many practical applications. For example, it does not hold if we set $\epsilon = 0$ in the constraint function (9).

Algorithm 2 ODOTS: Server's algorithm

1: Initialize and broadcast α, γ, and η. For each t, do:
2: Receive quantized local decisions $\{\hat{\mathbf{x}}_t^n\}$.
3: Update noisy global decision $\hat{\mathbf{x}}_{t+1}$ via (7).
4: Broadcast $\hat{\mathbf{x}}_{t+1}$ to all devices.

where $[\mathbf{a}]_{\mathbf{b}}^{\mathbf{c}} = \min\{\mathbf{c}, \max\{\mathbf{a}, \mathbf{b}\}\}$ is an entry-wise projection operator.

Note that the local decision update (12) is scaled by a factor $\frac{\alpha}{\alpha+\eta Q_t^n}$ that depends on the ratio of the tunable virtual queue length Q_t^n and the gradient descent step size α. The values of Q_t^n and α tune the relative weights on the global decision $\hat{\mathbf{x}}_t$ and the previous quantized local decision $\hat{\mathbf{x}}_{t-1}^n$ on the new local decision update. When Q_t^n is small, *i.e.,* the scale on the decision update $\frac{\alpha}{\alpha+\eta Q_t^n}$ is close to 1 and the weight $\frac{\eta Q_t^n}{\alpha}$ on $\hat{\mathbf{x}}_{t-1}$ is close to 0, (12) becomes the standard projected local gradient descent based on the noisy global decision $\mathbf{x}_t^n = [(\hat{\mathbf{x}}_t - \frac{1}{2\alpha}\nabla f_t^n(\hat{\mathbf{x}}_t))]_{-x_{\max}\mathbf{1}}^{x_{\max}\mathbf{1}}$ to minimize the loss. Otherwise, when Q_t^n is relatively large compared with α, *i.e.,* $\frac{\alpha}{\alpha+\eta Q_t^n}$ is close to 0 and $\frac{\eta Q_t^n}{\alpha}$ is large, the gradient descent is *stalled* and (12) is close to $\hat{\mathbf{x}}_{t-1}^n$, which reduces the communication overhead due to the resulting high interdependence between $\hat{\mathbf{x}}_t^n$ and $\hat{\mathbf{x}}_{t-1}^n$. Therefore, the local decision update by ODOTS is both computation- and communication aware, *i.e.,* automatically balancing the improvement in optimization and the cost in communication over time.

We summarize the devices' algorithm and the server's algorithm in Algorithms 1 and 2. The choices of algorithm parameters α, γ, and η will be discussed in Section V-E, after we derive the bounds on the performance gap and constraint violation for ODOTS.

V. Performance Bounds of ODOTS

In this section, we further show that ODOTS provides strong performance guarantees in both the optimization objective and the temporal decision dis-similarity constraint. In particular, the unique design of ODOTS requires new analysis techniques to account for the impact of the noisy decision update and the tunable virtual queue.

A. Preliminaries

We make the following standard assumptions in the performance analysis of ODOTS.

Assumption 1. The local loss function $f_t^n(\mathbf{x})$ is convex, *i.e.,*

$$f_t^n(\mathbf{y}) \geq f_t^n(\mathbf{x}) + \langle \nabla f_t^n(\mathbf{x}), \mathbf{y}-\mathbf{x}\rangle,\ \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \forall n, \forall t. \quad (13)$$

Assumption 2. The local loss function $f_t^n(\mathbf{x})$ has bounded gradient $\nabla f_t^n(\mathbf{x})$: $\exists D > 0$, *s.t.,*

$$\|f_t^n(\mathbf{x})\| \leq D, \quad \forall \mathbf{x} \in \mathbb{R}^d, \forall n, \forall t. \quad (14)$$

Assumptions 1 and 2 are common in existing studies on online distributed optimization. Nevertheless, later in Section VI-C, we empirically show that ODOTS also works well for general non-convex loss functions.

The following lemma shows that **P1** satisfies the following properties: 1) The feasible set $\mathcal{X}$ is bounded; 2) The quantization error $\mathbf{n}_t$ is bounded; 3) The constraint function $g_t^n(\mathbf{x})$ is bounded. The proof is omitted due to the page limit.

Lemma 1. Our formulated **P1** satisfies the following:

$$\|\mathbf{x}-\mathbf{y}\| \leq R, \quad \forall \mathbf{x},\mathbf{y} \in \mathcal{X}, \tag{15}$$

$$\|\mathbf{n}_t\| \leq \delta, \quad \forall t, \tag{16}$$

$$|g_t^n(\mathbf{x})| \leq G, \quad \forall \mathbf{x} \in \mathcal{X}, \forall n, \forall t. \tag{17}$$

where $R=2\sqrt{d}x_{\max}$, $\delta=\frac{R}{4(s-1)}$, and $G=\max\{\epsilon, R^2+\delta^2-\epsilon\}$.

B. Bounds on the Tunable Virtual Queue and Modified Lyapunov Drift

We first provide an upper bound on the tunable virtual queue.

Lemma 2. The tunable virtual queue generated by ODOTS is upper bounded as follows:

$$Q_t^n \leq \frac{\eta G}{\gamma}, \quad \forall n, \forall t. \tag{18}$$

Proof: We prove by induction. We have $Q_1^n = 0 \leq \frac{\eta G}{\gamma}$ by initialization. Suppose $Q_\tau^n \leq \frac{\eta G}{\gamma}$ for some $\tau \geq 1$. We have

$$\begin{aligned} Q_{\tau+1}^n &\overset{(a)}{\leq} |(1-\gamma^2)Q_\tau^n + \gamma\eta g_\tau^n(\mathbf{x}_\tau^n)| \\ &\overset{(b)}{\leq} (1-\gamma^2)Q_\tau^n + \gamma\eta|g_\tau^n(\mathbf{x}_\tau^n)| \overset{(c)}{\leq} (1-\gamma^2)\frac{\eta G}{\gamma} + \gamma\eta G = \frac{\eta G}{\gamma} \end{aligned}$$

where (a) follows directly from (11); (b) is because of $Q_t^n \geq 0, \forall t$, $\gamma \in (0,1)$, and the triangle inequality; and (c) follows from induction and the bound on $g_t^n(\mathbf{x})$ in (17). ■

Although our tunable virtual queue updating rule (11) yields a simple upper bound on Q_t^n in (18), unfortunately it also breaks the key connection between the virtual queue bound and the constraint violation bound used by [40], [46]-[49] in their performance analysis. To proceed with our analysis, we define a *modified* Lyapunov drift for each device n as

$$\Theta_t^n = \frac{1}{2\gamma}(Q_{t+1}^n - U)^2 - \frac{1}{2\gamma}(Q_t^n - U)^2. \tag{19}$$

where $U \geq 0$ is a *virtual* regularization factor on the quadratic Lyapunov function. Note that U is introduced only to enable our performance bound analysis, and ODOTS does not require the value of U to run. Using the result in Lemma 2, we provide an upper bound on Θ_t^n, which regains the connection between the tunable virtual queue and the constraint violation.

Lemma 3. The modified Lyapunov drift is upper bounded by

$$\Theta_t^n \leq \eta Q_t^n g_t^n(\mathbf{x}_t^n) - U\eta g_t^n(\mathbf{x}_t^n) + 2\gamma\eta^2 G^2 + \frac{\gamma}{2}U^2, \ \forall n, \forall t. \tag{20}$$

Proof: From the tunable virtual queue updating rule (11) and the fact that $|[a]_+ - [b]_+| \leq |a-b|$, we have

$$\begin{aligned} (Q_{t+1}^n - U)^2 &\leq \big((1-\gamma^2)Q_t^n + \gamma\eta g_t^n(\mathbf{x}_t^n) - U\big)^2 \\ &= (Q_t^n - U)^2 + \gamma^2\big(\eta g_t^n(\mathbf{x}_t^n) - \gamma Q_t^n\big)^2 + 2\gamma\eta Q_t^n g_t^n(\mathbf{x}_t^n) \\ &\quad - 2\gamma\eta U g_t^n(\mathbf{x}_t^n) - 2\gamma^2(Q_t^n - U)Q_t^n. \end{aligned} \tag{21}$$

We now bound the terms on the right-hand side (RHS) of (21). From the bound on $g_t^n(\mathbf{x})$ in (17) and the bound on Q_t^n in (18), we have $|\eta g_t^n(\mathbf{x}_t^n) - \gamma Q_t^n| \leq \eta|g_t^n(\mathbf{x}_t^n)| + \gamma Q_t^n \leq 2\eta G$. For the last term on the RHS of (21), we have $-2(Q_t^n - U)Q_t^n = U^2 - (Q_t^n)^2 - (Q_t^n - U)^2 \leq U^2$. Substituting the above two inequalities into (21) and dividing both sides of the resulting inequality by 2γ, we prove (20). ■

From the upper bound on Θ_t^n in (20) and noting that $2\gamma\eta^2G^2 + \frac{\gamma}{2}U^2$ in (20) is a constant, we can see that solving $\mathbf{P2}^n$ for each device n is equivalent to minimizing an upper bound on the following modified *drift plus penalty plus violation* term at each time t:

$$\Theta_t^n + \langle \nabla f_t^n(\hat{\mathbf{x}}_t), \mathbf{x} - \hat{\mathbf{x}}_t\rangle + \alpha\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2 + U\eta g_t^n(\mathbf{x}). \tag{22}$$

This is similar to the Lyapunov optimization approach [40] that minimizes a drift plus penalty term at each time. However, the penalty term in standard Lyapunov optimization is the loss function itself. As explained in Section II-C, for machine learning tasks in distributed learning, this means finding the optimal model within a single time slot and is impossible in general. Instead, we use the penalty term $\langle \nabla f_t^n(\hat{\mathbf{x}}_t), \mathbf{x} - \hat{\mathbf{x}}_t\rangle + \alpha\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2$ to enable local gradient descent for the global loss minimization. Note that when the virtual penalty factor U on the quadratic Lyapunov function is nonzero, (22) also includes a violation term $U\eta g_t^n(\mathbf{x})$. This is introduced to help bound the constraint violation, since the upper bound (18) on our tunable virtual queue is not directly transferable to the constraint violation bound anymore.

C. Bound on the Performance Gap

Using the results in Lemmas 1-3, the following lemma provides an upper bound on the weighted sum of the per-slot local loss and constraint violation $f_t^n(\hat{\mathbf{x}}_t) + U\eta g_t^n(\mathbf{x}_t^n)$ by ODOTS.

Lemma 4. The weighted sum of the per-slot local loss and constraint violation yielded by ODOTS is upper bounded by

$$\begin{aligned} f_t^n(\hat{\mathbf{x}}_t) + U\eta g_t^n(\mathbf{x}_t^n) &\leq f_t^n(\mathbf{x}_t^{\text{ctr}}) + \frac{D^2}{4\alpha} + 2\gamma\eta^2G^2 + \frac{\gamma}{2}U^2 \\ &\quad - \Theta_t^n + \alpha\big(\phi_t + \psi_t^n + \|\mathbf{n}_t\|^2 + 2R(\|\mathbf{n}_t\| + \pi_t)\big), \ \forall n, \forall t \end{aligned} \tag{23}$$

where $\phi_t \triangleq \|\mathbf{x}_t^{\text{ctr}} - \mathbf{x}_t\|^2 - \|\mathbf{x}_{t+1}^{\text{ctr}} - \mathbf{x}_{t+1}\|^2$, $\psi_t^n \triangleq \|\mathbf{x}_t^{\text{ctr}} - \mathbf{x}_{t+1}\|^2 - \|\mathbf{x}_t^{\text{ctr}} - \mathbf{x}_t^n\|^2$, and $\pi_t \triangleq \|\mathbf{x}_t^{\text{ctr}} - \mathbf{x}_{t+1}^{\text{ctr}}\|$.

Proof: We require the following lemma, which is copied from Lemma 2.8 in [23].

Lemma 5. ([23, Lemma 2.8]) Let $\mathcal{Z} \in \mathbb{R}^z$ be a nonempty convex set. Let $h(\mathbf{z}) : \mathbb{R}^z \to \mathbb{R}$ be a 2ϱ-strongly convex function over $\mathcal{Z}$ w.r.t. any norm $\|\cdot\|'$. Let $\mathbf{w} = \arg\min_{\mathbf{z}\in\mathcal{Z}} h(\mathbf{z})$. Then, for any $\mathbf{u} \in \mathcal{Z}$, we have $h(\mathbf{w}) \leq h(\mathbf{u}) - \varrho\|\mathbf{u} - \mathbf{w}\|'^2$.

The objective function of $\mathbf{P2}^n$ is 2α-strongly convex over $\mathcal{X}$ w.r.t. $\|\cdot\|$ due to the regularization term $\alpha\|\mathbf{x} - \hat{\mathbf{x}}_t\|^2$. Since $\mathbf{x}_t^n$ is the optimal solution to $\mathbf{P2}^n$, from Lemma 5, we have

$$\begin{aligned} &\langle \nabla f_t^n(\hat{\mathbf{x}}_t), \mathbf{x}_t^n - \hat{\mathbf{x}}_t\rangle + \alpha\|\mathbf{x}_t^n - \hat{\mathbf{x}}_t\|^2 + \eta Q_t^n g_t^n(\mathbf{x}_t^n) \\ &\quad \leq \langle \nabla f_t^n(\hat{\mathbf{x}}_t), \mathbf{x}_t^{\text{ctr}} - \hat{\mathbf{x}}_t\rangle + \eta Q_t^n g_t^n(\mathbf{x}_t^{\text{ctr}}) \\ &\qquad + \alpha(\|\mathbf{x}_t^{\text{ctr}} - \hat{\mathbf{x}}_t\|^2 - \|\mathbf{x}_t^{\text{ctr}} - \mathbf{x}_t^n\|^2). \end{aligned} \tag{24}$$

We now bound the last term on the RHS of (24). We have

$$\begin{aligned}
&\|\mathbf{x}_t^{\text{ctr}}-\hat{\mathbf{x}}_t\|^2-\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t^n\|^2\\
&=\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t+\mathbf{x}_t-\hat{\mathbf{x}}_t\|^2-\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_{t+1}^{\text{ctr}}+\mathbf{x}_{t+1}^{\text{ctr}}-\mathbf{x}_{t+1}\|^2\\
&\quad+(\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_{t+1}\|^2-\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t^n\|^2)\\
&\overset{(a)}{\leq}\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t\|^2+\|\mathbf{x}_t-\hat{\mathbf{x}}_t\|^2+2\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t\|\|\mathbf{x}_t-\hat{\mathbf{x}}_t\|\\
&\quad-\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_{t+1}^{\text{ctr}}\|^2-\|\mathbf{x}_{t+1}^{\text{ctr}}-\mathbf{x}_{t+1}\|^2\\
&\quad+2\|\mathbf{x}_{t+1}^{\text{ctr}}-\mathbf{x}_{t+1}\|\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_{t+1}^{\text{ctr}}\|+\psi_t^n\\
&\overset{(b)}{\leq}(\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t\|^2-\|\mathbf{x}_{t+1}^{\text{ctr}}-\mathbf{x}_{t+1}\|^2)+\|\mathbf{n}_t\|^2\\
&\quad+2\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t\|\|\mathbf{n}_t\|+2\|\mathbf{x}_{t+1}^{\text{ctr}}-\mathbf{x}_{t+1}\|\pi_t+\psi_t^n\\
&\overset{(c)}{\leq}\phi_t+\|\mathbf{n}_t\|^2+2R\|\mathbf{n}_t\|+2R\pi_t+\psi_t^n. \qquad (25)
\end{aligned}$$

where (a) follows from $\|\mathbf{a}+\mathbf{b}\|^2\leq\|\mathbf{a}\|^2+\|\mathbf{b}\|^2+2\|\mathbf{a}\|\|\mathbf{b}\|$, $-\|\mathbf{a}+\mathbf{b}\|^2\leq-\|\mathbf{a}\|^2-\|\mathbf{b}\|^2+2\|\mathbf{a}\|\|\mathbf{b}\|$, and the definition of ψ_t^n; (b) is because of the definitions of $\mathbf{n}_t$ and π_t; and (c) follows from the bound of $\mathcal{X}$ in (15) and the definition of ϕ_t.

Substituting (25) into (24), and noting that $Q_t^n g_t^n(\mathbf{x}_t^{\text{ctr}})\leq 0$ and $\langle\nabla f_t^n(\hat{\mathbf{x}}_t),\mathbf{x}_t^{\text{ctr}}-\hat{\mathbf{x}}_t\rangle\leq f_t^n(\mathbf{x}_t^{\text{ctr}})-f_t^n(\hat{\mathbf{x}}_t)$, we have

$$\begin{aligned}
f_t^n(\hat{\mathbf{x}}_t)\leq f_t^n(\mathbf{x}_t^{\text{ctr}})-\langle\nabla f_t^n(\hat{\mathbf{x}}_t),\mathbf{x}_t^n-\hat{\mathbf{x}}_t)\rangle-\alpha\|\mathbf{x}_t^n-\hat{\mathbf{x}}_t\|^2\\
-\eta Q_t^n g_t^n(\mathbf{x}_t^n)+\alpha\big(\phi_t+\psi_t^n+\|\mathbf{n}_t\|^2+2R(\|\mathbf{n}_t\|+\pi_t)\big). \quad (26)
\end{aligned}$$

Completing the square and noting that $\nabla f_t^n(\mathbf{x})$ is bounded in (14), we can show that $-\langle\nabla f_t^n(\hat{\mathbf{x}}_t),\mathbf{x}_t^n-\hat{\mathbf{x}}_t)\rangle-\alpha\|\mathbf{x}_t^n-\hat{\mathbf{x}}_t\|^2\leq\frac{D^2}{4\alpha}$. Substituting (20) and the above inequality into (26), we have (23). ∎

Based on the result in Lemma 4, we provide an upper bound on the performance gap to the centralized per-slot optimal solution sequence $\{\mathbf{x}_t^{\text{ctr}}\}$ for ODOTS in the following theorem.

Theorem 6. The performance gap to $\{\mathbf{x}_t^{\text{ctr}}\}$ by DOTS is upper bounded by

$$\begin{aligned}
\sum_{t=1}^{T}(f_t(\hat{\mathbf{x}}_t)-f_t(\mathbf{x}_t^{\text{ctr}}))\leq\frac{D^2T}{4\alpha}+2\gamma\eta^2G^2T+\frac{\eta^2G^2\Omega_T}{2\gamma^3}\\
+\alpha\big(R^2+\Lambda_{2,T}+2R(\Lambda_T+\Pi_T)\big) \quad (27)
\end{aligned}$$

where $\Pi_T\triangleq\sum_{t=1}^{T}\pi_t$, $\Omega_T\triangleq\sum_{t=1}^{T}\sum_{n=1}^{N}(w_{t+1}^n-w_t^n)$, $\Lambda_T\triangleq\sum_{t=1}^{T}\|\mathbf{n}_t\|$, and $\Lambda_{2,T}\triangleq\sum_{t=1}^{T}\|\mathbf{n}_t\|^2$.

Proof: Multiplying both sides of (23) by w_t^n, setting $U=0$, and summing the resulting inequality over n and t, we have

$$\begin{aligned}
\sum_{t=1}^{T}\big(f_t(\hat{\mathbf{x}}_t)-f_t(\mathbf{x}_t^{\text{ctr}})\big)\overset{(a)}{\leq}\frac{D^2T}{4\alpha}+2\gamma\eta^2G^2T-\sum_{t=1}^{T}\sum_{n=1}^{N}w_t^n\Theta_t^n\\
+\alpha\sum_{t=1}^{T}\Big(\phi_t+\sum_{n=1}^{N}w_t^n\psi_t^n\Big)+\alpha\big(\Lambda_{2,T}+2R(\Lambda_T+\Pi_T)\big) \quad (28)
\end{aligned}$$

where (a) follows from the definitions of $\Lambda_{2,T}$, Λ_T, and Π_T.

We now bound the terms on the RHS of (28). From $Q_1^n=0$, the bound on Q_t^n in (18), and the definition of Ω_T, we have

$$\begin{aligned}
-\sum_{t=1}^{T}\sum_{n=1}^{N}w_t^n\Theta_t^n=\frac{1}{2\gamma}\sum_{t=1}^{T}\sum_{n=1}^{N}\big(w_t^n(Q_t^n)^2-w_{t+1}^n(Q_{t+1}^n)^2\big)\\
+\frac{1}{2\gamma}\sum_{t=1}^{T}\sum_{n=1}^{N}(w_{t+1}^n-w_t^n)(Q_{t+1}^n)^2\leq\frac{\eta^2G^2\Omega_T}{2\gamma^3}. \quad (29)
\end{aligned}$$

From the separate convexity of the Euclidean norm, we have

$$\begin{aligned}
\sum_{n=1}^{N}w_t^n\psi_t^n=\sum_{n=1}^{N}w_t^n\Big(\Big\|\sum_{m=1}^{N}w_t^m(\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t^m)\Big\|^2-\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t^n\|^2\Big)\\
\leq\sum_{n=1}^{N}w_t^n\Big(\sum_{m=1}^{N}\big(w_t^m\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t^m\|^2\big)-\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t^n\|^2\Big)=0. \quad (30)
\end{aligned}$$

Substituting (29) and (30) into (28), and noting that $\sum_{t=1}^{T}\phi_t\leq\|\mathbf{x}_1^{\text{ctr}}-\mathbf{x}_1\|^2\leq R^2$, we prove (27). ∎

D. Bound on the Constraint Violation

We now proceed to provide an upper bound on the constraint violation for ODOTS. The virtual-queue-based online optimization algorithms [40], [46]-[49] bound the constraint violation via the virtual queue bound, which requires Slater's condition (or its relaxed version in [48]). Instead, we resort to bound the constraint violation by properly setting the virtual penalty factor U in the modified Lyapunov drift Θ_t^n (19).

Theorem 7. The constraint violation yielded by ODOTS is upper bounded by

$$\begin{aligned}
\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}g_t^n(\mathbf{x}_t^n)\leq\Big(\frac{2\gamma^2T+2}{\gamma\eta^2}\Big)^{\frac{1}{2}}\Big(\frac{D^2T}{4\alpha}+2\gamma\eta^2G^2T\\
+D(R+\delta)T+\alpha\big(R^2(1+\Xi_T)+\Lambda_{2,T}+2R(\Lambda_T+\Pi_T)\big)\Big)^{\frac{1}{2}} \quad (31)
\end{aligned}$$

where $\Xi_T\triangleq\sum_{t=1}^{T}\sum_{n=1}^{N}\big(w_t^n-\frac{1}{N}\big)$.

Proof: Summing (23) over n and t, and dividing both sides of the resulting inequality by N, we have

$$\begin{aligned}
\frac{U\eta}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}g_t^n(\mathbf{x}_t^n)\leq\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}\big(f_t^n(\mathbf{x}_t^{\text{ctr}})-f_t^n(\hat{\mathbf{x}}_t)\big)\\
+\frac{D^2T}{4\alpha}+2\gamma\eta^2G^2T+\frac{\gamma T}{2}U^2-\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}\Theta_t^n\\
+\alpha\sum_{t=1}^{T}\Big(\phi_t+\sum_{n=1}^{N}\frac{\psi_t^n}{N}\Big)+\alpha\big(\Lambda_{2,T}+2R(\Lambda_T+\Pi_T)\big). \quad (32)
\end{aligned}$$

We now bound the terms on the RHS of (32). From the convexity of $f_t^n(\mathbf{x})$ in (13), and the bounds on $\nabla f_t^n(\mathbf{x})$, $\mathcal{X}$, $\|\mathbf{n}_t\|$ in (14), (15), (16), we have

$$\begin{aligned}
f_t^n(\mathbf{x}_t^{\text{ctr}})-f_t^n(\hat{\mathbf{x}}_t)\leq\langle\nabla f_t^n(\mathbf{x}_t^{\text{ctr}}),\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t-\mathbf{n}_t\rangle\\
\leq\|\nabla f_t^n(\mathbf{x}_t^{\text{ctr}})\|(\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t\|+\|\mathbf{n}_t\|)\leq D(R+\delta). \quad (33)
\end{aligned}$$

Similar to the proof of (30), we can show that

$$\sum_{t=1}^{T}\sum_{n=1}^{N}\frac{\psi_t^n}{N}\leq\sum_{t=1}^{T}\sum_{n=1}^{N}\big(w_t^n-\frac{1}{N}\big)\|\mathbf{x}_t^{\text{ctr}}-\mathbf{x}_t^n\|^2\leq R^2\Xi_T. \quad (34)$$

Also, noting that $Q_1^n=0$ by initialization, we have

$$-\sum_{t=1}^{T}\Theta_t^n=\frac{1}{2\gamma}\sum_{t=1}^{T}\big((Q_t^n-U)^2-(Q_{t+1}^n-U)^2\big)\leq\frac{U^2}{2\gamma}. \quad (35)$$

Substituting $\sum_{t=1}^{T}\phi_t \leq R^2$ and (33)-(35) into (32) with $U=\frac{\gamma\eta}{\gamma^2T+1}[\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}g_t^n(\mathbf{x}_t^n)]_+$, and noting that $a\leq[a]_+$, we have

$$\frac{\gamma\eta^2}{2\gamma^2T+2}\Big[\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}g_t^n(\mathbf{x}_t^n)\Big]_+^2\leq\frac{D^2T}{4\alpha}+2\gamma\eta^2G^2T$$
$$+D(R+\delta)T+\alpha\big(R^2(1+\Xi_T)+\Lambda_{2,T}+2R(\Lambda_T+\Pi_T)\big).$$

Taking the square root on both side of the above inequality, we prove (31). ∎

E. Discussion on the Performance Bounds

We now discuss the sufficient conditions for ODOTS to yield sublinear performance gap and constraint violation. We define parameters $\mu\in[0,1]$ and $\nu\in[0,1]$ to represent the time variability of the underlying system, such that $\max\{\Pi_T,\Xi_T,\Lambda_{2,T},\Lambda_T\}=O(T^\mu)$ and $\Omega_T=\mathcal{O}(T^\nu)$. Note that Ξ_T and Ω_T are the accumulated variation measures of the time-varying weights $\{w_t^n\}$ on the devices (see Theorems 6 and 7 for definition). An important special case is $w_t^n=\frac{1}{N},\forall n,\forall t$, *i.e.,* the devices have time-invariant equal weights. From Theorems 6 and 7, we can derive the following corollary regarding the performance gap and constraint violation bounds, depending on whether w_t^n is time-varying. The proof is omitted for brevity.

Corollary 8. *Time-varying weight:* Let $\alpha=T^{\frac{1-\mu}{2}}$, $\gamma=T^{\frac{\nu-1}{4}}$, and $\eta=\mathcal{O}(1)$ in ODOTS. We have $\sum_{t=1}^{T}(f_t(\hat{\mathbf{x}}_t)-f_t(\mathbf{x}_t^{\text{ctr}}))=\mathcal{O}(\max\{T^{\frac{1+\mu}{2}},T^{\frac{3+\nu}{4}}\})$ and $\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}g_t^n(\mathbf{x}_t^n)=\mathcal{O}(\max\{T^{\frac{3+\mu}{4}},T^{\frac{7+\nu}{8}}\})$.

Time-invariant equal weight: Suppose $w_t^n=\frac{1}{N},\forall n,\forall t$ such that $\Xi_T=0$ and $\Omega_T=0$. Let $\alpha=T^{\frac{1-\mu}{2}}$, $\gamma=T^{-\frac{1}{2}}$, and $\eta=\mathcal{O}(1)$ in ODOTS. We have $\sum_{t=1}^{T}(f_t(\hat{\mathbf{x}}_t)-f_t(\mathbf{x}_t^{\text{ctr}}))=\mathcal{O}(T^{\frac{1+\mu}{2}})$ and $\frac{1}{N}\sum_{t=1}^{T}\sum_{n=1}^{N}g_t^n(\mathbf{x}_t^n)=\mathcal{O}(T^{\frac{3}{4}})$.

In particular, if $\mu<1$ and $\nu<1$, *i.e.,* the system variations are sublinear in T, both the performance gap and constraint violation are sublinear in T. We remark here that sublinear system variations is a standard necessary (but generally insufficient) condition for sublinear performance bounds in online optimization with unpredictable dynamics [39], [46]-[49].

VI. Application to Federated Learning

As an example to study the performance of ODOTS in practical systems, we apply it to federated learning (FL) [4], where multiple local devices cooperate to train a machine-learning model with the assistance of a server. We present numerical results to demonstrate the performance advantage of ODOTS over state-of-the-art alternatives, based on real-world image classification datasets for both convex and non-convex loss functions.

A. Simulation Setup

We consider a FL system with $N=10$ devices and a server. We evaluate our results on the popular MNIST dataset [50]. Its training dataset $\mathcal{D}$ consists of 6×10^4 data samples and its test dataset $\mathcal{E}$ has 1×10^4 data samples. Each data sample $(\mathbf{u},v)$ represents an image with 28×28 pixels and $V=10$ possible labels, *i.e.,* $\mathbf{u}\in\mathbb{R}^{784}$ and $v\in\{1,\ldots,V\}$. We study the scenario where each local dataset $\mathcal{D}_t^n$ at device n only contains data samples of label n, such that the data is non-i.i.d. We assume device n randomly selects $|\mathcal{D}_t^n|=20$ data samples at each time t, such that the devices share the same weight $w_t^n=\frac{1}{N}$. We have also conducted experiments on time-varying weights and different datasets, which show a similar trend as the simulation results in this paper. Due to the page limit, we do not include them. This is to emulate the online FL scenario where data samples arrive at the devices over time.

We compare ODOTS with the following schemes.

- *Error-free FL:* We alternates local model update $\mathbf{x}_t^n=\mathbf{x}_t-\frac{1}{2\alpha}\nabla f_t^n(\mathbf{x}_t)$ and global model update $\mathbf{x}_{t+1}=\sum_{n=1}^{N}w_t^n\mathbf{x}_t^n$ at each time t. It represents the idealized standard FL algorithm where the communication is error free [4].
- *Primal-dual GD:* The primal-dual gradient descent (GD) algorithm in [39] is the current best solution for distributed constrained online convex optimization with consensus. We implement it to solve **P1**, except using the same current information on the loss and constraint functions as ODOTS.
- *QFL-CE:* We adopt the quantized federated learning (QFL) scheme in [7] by perform local model update (*i.e.,* (12) with $Q_t^n=0$) and quantization (*i.e.,* (4)) at each time t. We implement the same conditional entropy (CE) coding as ODOTS for QFL.[5] The server then updates its noisy global model (*i.e.,* (7)). This is a state-of-the-art approach where model training and compression are separately designed.

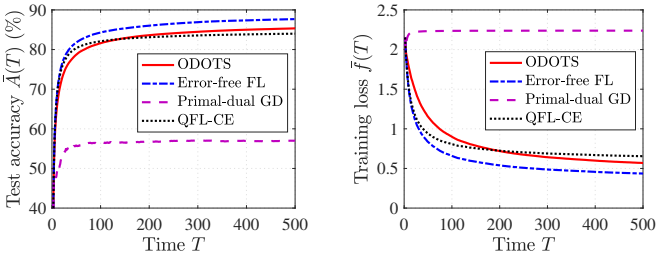
B. Convex Loss: Logistic Regression

We consider the cross-entropy loss for multinomial logistic regression, given by $l(\mathbf{x};\mathbf{u},v)=-\sum_{j=1}^{V}1\{v=j\}\log\frac{\exp(\langle\mathbf{x}[j],\mathbf{u}\rangle)}{\sum_{k=1}^{V}\exp(\langle\mathbf{x}[k],\mathbf{u}\rangle)}$, where $\mathbf{x}=[\mathbf{x}[1]^T,\ldots,\mathbf{x}[V]^T]^T$ with $\mathbf{x}[j]\in\mathbb{R}^{784}$ being the model for label j. The entire model $\mathbf{x}$ is thus of dimension $d=7840$. Our computation performance metrics are the time-averaged test accuracy $\bar{A}(T)=\frac{1}{|\mathcal{E}|T}\sum_{t=1}^{T}\sum_{i=1}^{|\mathcal{E}|}1\Big\{\text{argmax}_j\Big\{\frac{\exp(\langle\hat{\mathbf{x}}_t[j],\mathbf{u}^i\rangle)}{\sum_{k=1}^{V}\exp(\langle\hat{\mathbf{x}}_t[k],\mathbf{u}^i\rangle)}\Big\}=v^i\Big\}$, and the time-averaged training loss $\bar{f}(T)=\frac{1}{T}\sum_{t=1}^{T}\sum_{n=1}^{N}\frac{w_t^n}{|\mathcal{D}_t^n|}\sum_{i=1}^{|\mathcal{D}_t^n|}l(\hat{\mathbf{x}}_t;\mathbf{u}_t^{n,i},v_t^{n,i})$. Our communication performance metrics are the total number of transmitted bits using the conditional entropy coding $B(T)=\sum_{t=1}^{T}\sum_{n=1}^{N}H(\hat{\mathbf{x}}_t^n|\hat{\mathbf{x}}_{t-1}^n)$ and the time-averaged decision dis-similarity $\bar{g}(T)=\frac{1}{TN}\sum_{t=1}^{T}\sum_{n=1}^{N}\|\mathbf{x}_t^n-\hat{\mathbf{x}}_{t-1}^n\|^2$.[6]

Fig. 1 shows $\bar{A}(T)$, $\bar{f}(T)$, $B(T)$, and $\bar{g}(T)$ versus T. We set the decision dis-similarity limit $\epsilon=1e^{-6}$. We set the quantization bit length $b=5$ for ODOTS and $b=4$ for

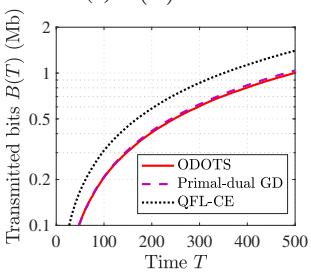
[5]The Elias coding used in [7] does not use any model similarity, and thus incurs more communication overhead compared with the CE coding.

[6]We use the histogram method to estimate the joint probability distribution of $\hat{\mathbf{x}}_t^n$ and $\hat{\mathbf{x}}_{t-1}^n$ and then compute the conditional entropy $H(\hat{\mathbf{x}}_t^n|\hat{\mathbf{x}}_{t-1}^n)$.

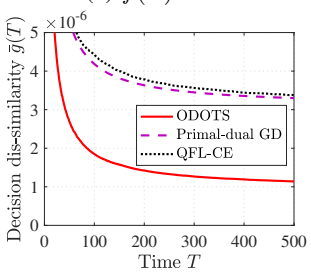


(a) $\bar{A}(T)$ vs. T.

(b) $\bar{f}(T)$ vs. T.



(c) $B(T)$ vs. T.



(d) $\bar{g}(T)$ vs. T.

Fig. 1: Test accuracy $\bar{A}(T)$, training loss $\bar{f}(T)$, transmitted bits $B(T)$, and decision dis-similarity $\bar{g}(T)$ vs. time T.

Primal-dual GD and QFL-CE. We set the maximum decision limit $x_{\max} = 1 \times 10^{-3}$, step-size $\alpha = 1 \times 10^{5}$, tuning factor $\gamma = 0.5$, and weighting factor $\eta = 5 \times 10^{5}$ in ODOTS. We use the same parameter values for the other schemes if any is used. We note that despite the higher quantization bit length b in ODOTS, due to its inherent communication efficiency, its total number of transmitted bits remains lower than both Primal-dual GD and QFL-CE. We observe that the test accuracy yielded by ODOTS is over 25% higher than Primal-dual GD. This is because Primal-dual GD performs dual gradient descent to control the constraint violation, which can deteriorate its performance when the gradient directions of loss and constraint functions deviate much from each other. Compared with QFL-CE, ODOTS achieves higher test accuracy and incurs abound 30% less communication overhead, thanks to its joint consideration of computation and communication over time. Also, we observe that ODOTS converges slightly slower than QFL-CE at the early training stage, this is because the value of the tunable virtual queue Q_t^n in (11) is relatively large at the beginning to reduce the transmitted bits.

In Fig. 2, we compare the final test accuracy $A(T)$ between ODOTS and QFL-CE under different total transmitted bits $B(T)$. We vary the quantization bit length b in QFL-CE to trade off its computation and communication performance. For ODOTS, we also vary ϵ for any given b value. The final test accuracies yielded by QFL-CE and ODOTS both decrease as b decreases due to the increased quantization errors. However, for any operating point on the QFL-CE curve, we can always find a combination of b and ϵ for ODOTS that achieves higher test accuracy while incurring less communication overhead. Furthermore, their difference in test accuracy grows dramatically as the number of transmitted bits decreases. This suggests that ODOTS is particularly advantageous in systems with a tight communication budget.

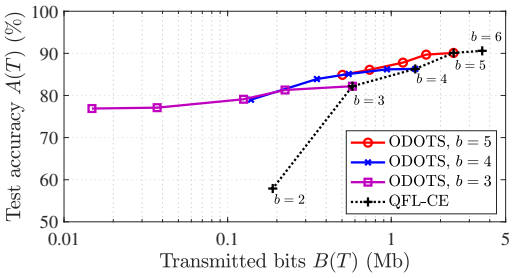


Fig. 2: Final test accuracy $A(T)$ vs. transmitted bits $B(T)$ for convex logistic regression.

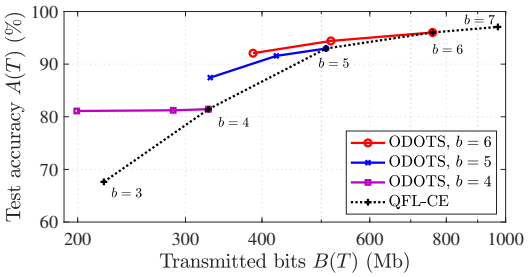


Fig. 3: Final test accuracy $A(T)$ vs. transmitted bits $B(T)$ for non-convex convolutional neural network training.

C. Non-Convex Loss: Convolutional Neural Network Training

The performance analysis of ODOTS in Section V requires convex loss functions. To further evaluate the performance of ODOTS for non-convex loss functions, we consider training a convolutional neural network for MNIST classification, with 784 pixels as input, a convolutional layer with 10 filters each of size 9×9, a ReLU hidden layer with 100 neurons, and a softmax output layer with 10 neurons. The total number of model parameters is $d = 101,810$. We set $x_{\max} = 1$, $\alpha = 2$, $\gamma = 0.5$, and $\eta = 0.01$ in ODOTS. Similar to Fig. 2, Fig. 3 compares the performance of ODOTS and QFL-CE in this scenario. Note that the number of transmitted bits is substantially higher due to the larger number of model parameters, compared with the convex logistic regression scenario. We again observe similar trends as in Fig. 2, with ODOTS substantially outperforming QFL-CE especially when the number of transmitted bits is moderate to low.

VII. Conclusions

We consider online distributed optimization in networked systems, under a long-term decision dis-similarity constraint to control the communication overhead. We propose an efficient ODOTS algorithm to balance the improvement in optimization and the cost of communication over time via a novel tunable virtual queue. Through a modified Lyapunov drift analysis, we show that ODOTS achieves sublinear performance gap from the centralized per-slot optimizer and sublinear constraint violation simultaneously. When applying ODOTS to federated learning, our experimental results demonstrate that ODOTS can have substantial performance advantage over state-of-the-art approaches, in terms of both improved test accuracy and reduced communication overhead. ODOTS is advantageous especially in systems with a tight communication budget.

REFERENCES

[1] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Commun. Surveys Tuts.*, vol. 19, pp. 2322–2358, 2017.

[2] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proc. IEEE*, vol. 107, pp. 1738–1762, Aug. 2019.

[3] M. I. Jordan, J. D. Lee, and Y. Yang, "Communication-efficient distributed statistical inference," *J. Amer. Stat. Assoc.*, vol. 19, pp. 2322–2358, Feb. 2018.

[4] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. Y. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. Intel. Conf. Artif. Intell. Statist. (AISTATS)*, 2017.

[5] S. Wang *et al.*, "When edge meets learning: Adaptive control for resource-constrained distributed machine learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2018.

[6] F. Seide, H. Fu, J. Droppo, G. Li, and D. Yu, "1-bit stochastic gradient descent and its application to data-parallel distributed training of speech DNNs," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2014.

[7] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via gradient quantization and encoding," in *Proc. Adv. Neural Info. Proc. Sys. (NIPS)*, 2017.

[8] J. Wu, W. Huang, J. Huang, and T. Zhang, "Error compensated quantized SGD and its applications to large-scale distributed optimization," in *Proc. Intel. Conf. Mach. Learn. (ICML)*, 2018.

[9] H. Zhang, J. Li, K. Kara, D. Alistarh, J. Liu, and C. Zhang, "ZipML: Training linear models with end-to-end low precision, and a little bit of deep learning," in *Proc. Intel. Conf. Mach. Learn. (ICML)*, 2017.

[10] W. Wen, C. Xu, F. Yan, C. Wu, Y. Wang, Y. Chen, and H. Li, "Terngrad: Ternary gradients to reduce communication in distributed deep learning," in *Proc. Adv. Neural Info. Proc. Sys. (NIPS)*, 2017.

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

[12] N. Strom, "Scalable distributed DNN training using commodity GPU cloud computing," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (INTERSPEECH)*, 2015.

[13] P. Jiang and G. Agrawal, "A linear speedup analysis of distributed deep learning with sparse and quantized communication," in *Proc. Adv. Neural Info. Proc. Sys. (NIPS)*, 2018.

[14] A. Abdi and F. Fekri, "Reducing communication overhead via CEO in distributed training," in *Proc. IEEE Intel. Workshop on Signal Process. Advances in Wireless Commun. (SPAWC)*, 2019, pp. 1–5.

[15] L. Abrahamyan, Y. Chen, G. Bekoulis, and N. Deligiannis, "Learned gradient compression for distributed deep learning," *IEEE Trans. Neural Netw. Learn. Syst.*, Jun. 2021.

[16] C.-Y. Chen *et al.*, "Scalecom: Scalable sparsified gradient compression for communication-efficient distributed training," in *Proc. Adv. Neural Info. Proc. Sys. (NIPS)*, 2020.

[17] J. Sun, T. Chen, G. Giannakis, and Z. Yang, "Communication-efficient distributed learning via lazily aggregated quantized gradients," in *Proc. Adv. Neural Info. Proc. Sys. (NIPS)*, 2019.

[18] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York: Wiley, 1991.

[19] M. Soysal and E. G. Schmidt, "Machine learning algorithms for accurate flow-based network traffic classification: Evaluation and comparison," *Perform. Eval.*, vol. 67, pp. 451–467, 2010.

[20] S. Liang, X. Zhang, Z. Ren, and E. Kanoulas, "Dynamic embeddings for user profiling in twitter," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2018.

[21] C. Gutterman *et al.*, "Requet: Real-time QoE metric detection for encrypted YouTube traffic," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 16, pp. 1–28, 2020.

[22] N. Cesa-Bianchi and G. Lugosi, *Prediction, Learning, and Games*. Cambridge University Press, 2006.

[23] S. Shalev-Shwartz, "Online learning and online convex optimization," *Found. Trends Mach. Learn.*, vol. 4, pp. 107–194, Feb. 2012.

[24] T. Yang *et al.*, "A survey of distributed optimization," *Annu. Rev. Control*, vol. 46, pp. 278–305, 2019.

[25] J. C. Duchi, A. Agarwal, and M. J. Wainwright, "Dual averaging for distributed optimization: Convergence analysis and network scaling," *IEEE Trans. Automat. Contr.*, vol. 57, no. 3, pp. 592–606, 2012.

[26] M. Raginsky and J. Bouvrie, "Continuous-time stochastic mirror descent on a network: Variance reduction, consensus, convergence," in *IEEE Conf. Decision Control (CDC)*, 2012.

[27] S. Hosseini, A. Chapman, and M. Mesbahi, "Online distributed optimization via dual averaging," in *IEEE Conf. Decision Control (CDC)*, 2013.

[28] S. Shahrampour and A. Jadbabaie, "Distributed online optimization in dynamic environments using mirror descent," *IEEE Trans. Automat. Control*, vol. 63, pp. 714–725, Mar. 2018.

[29] O. Shamir, N. Srebro, and T. Zhang, "Communication-efficient distributed optimization using an approximate Newton-type method," in *Proc. Intel. Conf. Mach. Learn. (ICML)*, 2014.

[30] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Pschel, "D-ADMM: A communication-efficient distributed algorithm for separable optimization," *IEEE Trans. Signal Process.*, vol. 61, pp. 2718–2723, 2013.

[31] C. Liu, H. Li, Y. Shi, and D. Xu, "Distributed event-triggered gradient method for constrained convex minimization," *IEEE Trans. Automat. Contr.*, vol. 65, pp. 778–785, Feb. 2020.

[32] M. Jaggi, V. Smith, M. Takac, J. Terhorst, S. Krishnan, T. Hofmann, and M. I. Jordan, "Communication-efficient distributed dual coordinate ascent," in *Proc. Adv. Neural Info. Proc. Sys. (NIPS)*, 2014.

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

[34] J. Wang, M. Dong, B. Liang, G. Boudreau, and H. Abou-Zeid, "Online model updating with analog aggregation in wireless edge learning," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2022.

[35] S. Lee and M. M. Zavlanos, "Distributed primal-dual methods for online constrained optimization," in *Proc. Amer. Control Conf. (ACC)*, 2016.

[36] D. Yuan, D. W. C. Ho, and G.-P. Jiang, "An adaptive primal-dual subgradient algorithm for online distributed constrained optimization," *IEEE Trans. Cybern.*, vol. 48, pp. 3045–3055, Nov. 2018.

[37] D. Yuan, A. Proutiere, and G. Shi, "Distributed online optimization with long-term constraints," *IEEE Trans. Automat. Contr.*, vol. 67, pp. 1089–1104, Mar. 2022.

[38] S. Paternain, S. Lee, M. M. Zavlanos, and A. Ribeiro, "Distributed constrained online learning," *IEEE Trans. Signal Process.*, vol. 68, pp. 3486–3499, Jun. 2020.

[39] P. Sharma, P. Khanduri, L. Shen, D. J. Bucci, and P. K. Varshney, "On distributed online convex optimization with sublinear dynamic regret and fit," in *Proc. Asilomar Conf. Signal Sys. Comput. (ASILOMARSSC)*, 2021.

[40] M. J. Neely, *Stochastic Network Optimization with Application on Communication and Queueing Systems*. Morgan & Claypool, 2010.

[41] P. Elias, "Predictive coding–I," *IRE Trans. Inf. Theory*, vol. 1, pp. 16–24, 1955.

[42] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proc. IRE*, vol. 40, pp. 1098–1101, 1952.

[43] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. Inf. Theory*, vol. 22, pp. 1–10, 1976.

[44] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, pp. 471–480, Jul. 1973.

[45] S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (DISCUS): design and construction," *IEEE Trans. Inf. Theory*, vol. 49, no. 3, pp. 626–643, 2003.

[46] H. Yu, M. J. Neely, and X. Wei, "Online convex optimization with stochastic constraints," in *Proc. Adv. Neural Info. Proc. Sys. (NIPS)*, 2017.

[47] X. Cao, J. Zhang, and H. V. Poor, "A virtual-queue-based algorithm for constrained online convex optimization with applications to data center resource allocation," *IEEE J. Sel. Topics Signal Process.*, vol. 12, pp. 703–716, Aug. 2018.

[48] X. Wei, H. Yu, and M. J. Neely, "Online primal-dual mirror descent under stochastic constraints," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 4, Jun. 2020.

[49] J. Wang, M. Dong, B. Liang, G. Boudreau, and H. Abou-Zeid, "Delay-tolerant OCO with long-term constraints: Algorithm and its application to network resource allocation," *IEEE/ACM Trans. Netw.*, Jul. 2022.

[50] Y. LeCun, C. Cortes, and C. Burges, "The MNIST database," 1998. [Online]. Available: http://yann.lecun.com/exdb/mnist/