QC-ODKLA: Quantized and Communication-Censored Online Decentralized Kernel Learning via Linearized ADMM

Ping Xu, *Member, IEEE*, Yue Wang, *Senior Member, IEEE*, Xiang Chen, *Member, IEEE*, and Zhi Tian, *Fellow, IEEE*

***Abstract*— This article focuses on online kernel learning over a decentralized network. Each agent in the network receives online streaming data and collaboratively learns a globally optimal nonlinear prediction function in the reproducing kernel Hilbert space (RKHS). To overcome the curse of dimensionality issue in traditional online kernel learning, we utilize random feature (RF) mapping to convert the nonparametric kernel learning problem into a fixed-length parametric one in the RF space. We then propose a novel learning framework, named online decentralized kernel learning via linearized ADMM (ODKLA), to efficiently solve the online decentralized kernel learning problem. To enhance communication efficiency, we introduce quantization and censoring strategies in the communication stage, resulting in the quantized and communication-censored ODKLA (QC-ODKLA) algorithm. We theoretically prove that both ODKLA and QC-ODKLA can achieve the optimal sublinear regret $\mathcal{O}(\sqrt{T})$ over T time slots. Through numerical experiments, we evaluate the learning effectiveness, communication efficiency, and computation efficiency of the proposed methods.**

***Index Terms*— Communication censoring, decentralized online kernel learning, linearized alternating direction method of multiplier (ADMM), quantization, random feature (RF) mapping.**

I. Introduction

DECENTRALIZED online learning has been widely studied in the last decades, mostly motivated by its broad applications in networked multi-agent systems, such as wireless sensor networks, robotics, and the internet of things, etc., [1], [2]. In these systems, a number of agents collect their own online streaming data and aim to learn a common functional model through local information exchange. This objective is usually achieved by decentralized online convex optimization [3], [4], [5], [6]. With an online gradient descent–based algorithm [7], or through online alternating direction method of multipliers (ADMMs) [4], a static regret $\mathcal{O}(\sqrt{T})$ can be achieved over a time horizon T. Further, if the cost functions are strictly convex, an efficient algorithm based on the Newton method achieves a regret bound of $\mathcal{O}(\log T)$ [8]. However, all these works assume that the functional model to be learned by agents is linear, which may not be always true in practical applications.

Motivated by the universality of kernel methods in approximating nonlinear functions, this article aims to solve the decentralized online kernel learning problem, where the common function to be learned by agents is assumed to be nonlinear and belongs to the reproducing kernel Hilbert space (RKHS). However, directly applying kernel methods for decentralized online learning is a formidably challenging task because they adopt nonparametric models, where the number of model variables grows proportionally to the data size. This incurs the curse of dimensionality issue when data size becomes large over time. Additionally, the data-dependent decision variables hinder consensus optimization, especially when the data sizes vary among different agents and across time, as well as under certain circumstances where raw data exchange is prohibited [9]. Another key issue in decentralized (kernel) learning is its reliance on iterative local communications for computational feasibility and efficiency. This incurs frequent communications among agents to exchange their locally computed updates of the shared learning model, which can cause tremendous communication overhead in terms of both link bandwidth and transmission power. Therefore, it is crucial to design both communication- and computation-efficient distributed online kernel learning algorithms with data privacy protection.

To alleviate the computational complexity of kernel methods, [10], [11] propose to restrict the number of parameters to be estimated. Random feature (RF) based methods approximate the kernel function using a small number of features that are randomly sampled from a distribution independent of the training data [12], [13], [14]. In contrast, Nyström methods approximate the kernel matrix by randomly selecting a subset of training data to form its basis functions [15], [16]. Compared to Nyström methods, the data-independent RF-based methods not only circumvent the curse of dimensionality problem but also enable consensus optimization without any raw data exchange among agents, making them popular in many decentralized kernel learning works, including

Manuscript received 14 July 2022; revised 13 June 2023; accepted 18 August 2023. This work was supported in part by the National Science Foundation of the U.S. under Grant 1741338, Grant 1939553, Grant 2003211, Grant 2128596, and Grant 2136202; and in part by the Virginia Research Investment Fund through Commonwealth Cyber Initiative under Grant 223996. *(Corresponding author: Ping Xu.)*

Ping Xu is with the Department of Electrical and Computer Engineering, The University of Texas Rio Grande Valley, Edinburg, TX 78539 USA (e-mail: ping.t.xu@utrgv.edu).

Yue Wang is with the Department of Computer Science, Georgia State University, Atlanta, GA 30302 USA.

Xiang Chen and Zhi Tian are with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030 USA.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2023.3310499.

Digital Object Identifier 10.1109/TNNLS.2023.3310499

batch-form learning [9], [17] and online streaming learning [18], [19], [20].

To improve communication efficiency in decentralized learning, Nesterov's gradient is harnessed for achieving fast convergence [21]. Quantization [19], [22], [23] and sparsification [24], [25] methods are employed to compress transmitted information, while random node selection and asynchronous updating are utilized to reduce the number of transmissions per iteration [26], [27]. In contrast to random node selection, a more intuitive approach involves evaluating the importance of a message to avoid unnecessary transmissions. This is often achieved through the adoption of a communication-censoring or event-triggering scheme, which adaptively determines whether a message is informative enough to be transmitted during the iterative optimization process [9], [28], [29], [30].

In this article, we thus focus on the decentralized online kernel learning problem in networked multi-agent systems and aim to develop algorithms that are both communication- and computation-efficient. Relative to prior art, our contributions are summarized as follows.

1) We first utilize RF mapping to transform the original nonparametric data-dependent learning problem into a parametric fixed-size data-independent learning problem to circumvent the curse of dimensionality issue in traditional kernel methods and enable consensus optimization in a decentralized setting in the RF space. Different from existing gradient descent–based method [18], [19] or the standard ADMM algorithm [20], we propose to solve the decentralized kernel learning problem through linearized ADMM. This leads to the development of the **O**nline **d**ecentralized **k**ernel learning via **l**inearized ADMM (ODKLA) algorithm. In ODKLA, the local cost function of each agent is replaced by its first-order approximation centered at the current iterate and results in a closed-form primal update when the local cost function is convex. Compared with the standard ADMM [19], [20] that solve sub-optimization problems at each iteration to get the updates of the primal variables, ODKLA is more computationally efficient. Furthermore, ODKLA is essentially a variant of the higher-order ADMM and thus achieves faster convergence compared with the diffusion-based first-order gradient descent methods [18]. Additionally, since no raw data is exchanged among agents and the mapping from the original data space to the RF space is not one-to-one mapping, data privacy is protected to a certain level.

2) To reduce the communication cost, we develop the quantized and communication-censored online decentralized kernel learning via linearized ADMM (QC-ODKLA) algorithm by introducing a communication-censoring strategy and a quantization strategy. The communication-censoring strategy allows each agent to autonomously skip unnecessary communications when its local update is not informative enough for transmission, while the quantization strategy restricts the total number of transmitted bits throughout the learning process. In this way, the communication efficiency can be boosted at almost no sacrifice to the learning performance. In the absence of both strategies, QC-ODKLA degenerates to ODKLA. Compared to works such as [18] and [20] that do not have any communication-saving strategies, and [19] that solely utilizes the quantization strategy, our approach can further save communication resources.

3) In addition, we analyze the regret bound of QC-ODKLA. We show that when all techniques are adopted (linearized ADMM, quantization, and communication censoring), QC-ODKLA is still able to achieve the optimal sublinear regret of $\mathcal{O}(\sqrt{T})$ over T time slots under mild conditions, that is, the communication-censoring thresholds should be decaying. This indicates that the proposed QC-ODKLA algorithm enables every agent in the network to learn a common function that has a diminishing gap from the hindsight best function under mild conditions. The analysis also provides guidelines for tuning the parameters of QC-ODKLA, including the step size, the censoring function, and the quantization level. It also characterizes how the regret bound is affected by the properties of the cost functions and the communication graph.

4) Finally, we test the performance of our proposed ODKLA and QC-ODKLA algorithms on extensive real datasets. The results corroborate that both ODKLA and QC-ODKLA exhibit attractive learning performance and computation efficiency, while QC-ODKLA is highly communication efficient. Such salient features make it an attractive solution for broad applications where decentralized learning from streaming data is at its core.

The remaining of this article is organized as follows. Section II provides some preliminaries for decentralized kernel learning. Section III formulates the online decentralized kernel learning problem. Section IV develops the online decentralized kernel learning algorithms, including both ODKLA and QC-ODKLA. Section V presents the theoretical results. Section VI tests the proposed methods by real datasets. Concluding remarks are summarized in Section VII.

Notation: $\mathbb{R}$ denotes the set of real numbers. $\|\cdot\|_2$ denotes the Euclidean norm of vectors and $\|\cdot\|_F$ denotes the Frobenius norm of matrices. $|\cdot|$ denotes the cardinality of a set. $\mathbf{A}$ denotes a matrix, $\mathbf{a}$ denotes a vector, and a denotes a scalar.

II. Preliminaries

A. Network and Communication Models

1) Network Model: Consider a bidirectionally connected network of N agents and r arcs, whose underlying undirected communication graph is denoted as $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of agents with cardinality $|\mathcal{N}| = N$ and $\mathcal{A}$ is the set of undirected arcs with cardinality $|\mathcal{A}| = r$. Two agents i and j are called as neighbors when $(i, j) \in \mathcal{A}$ and, by the symmetry of the network, $(j, i) \in \mathcal{A}$. For agent i, its one-hop neighbors are in the set $\mathcal{N}_i = \{j|(j, i) \in \mathcal{A}\}$ with cardinality $|\mathcal{N}_i|$, which is also known as the degree d_i of agent i. The degree matrix of the communication graph is $\boldsymbol{D} \in \mathbb{R}^{N\times N}$, which is diagonal with the ith diagonal element being $d_i, \forall i$.

Define the symmetric adjacency matrix associated with the communication graph as $\boldsymbol{W} \in \mathbb{R}^{N\times N}$, whose (i, j)th entry is 1 if agent i and j are neighbors or 0 otherwise. Define the unsigned incidence matrix and the signed incidence matrix of the communication graph as $\mathbf{S}_+ \in \mathbb{R}^{N\times 2r}$ and $\mathbf{S}_- \in \mathbb{R}^{N\times 2r}$, respectively. According to [31], we have

$$\boldsymbol{D} + \boldsymbol{W} = \frac{1}{2}\mathbf{S}_+\mathbf{S}_+^\top$$
$$\boldsymbol{D} - \boldsymbol{W} = \frac{1}{2}\mathbf{S}_-\mathbf{S}_-^\top.$$

2) Communication Model: In this article, we consider synchronous communications. That is, the iterative process of algorithm implementation consists of three stages: communication, observation, and computation. In the communication stage, each agent broadcasts its state variable to its neighbors and receives state variables from its neighbors according to the communication-censoring rule, which shall be introduced later. After communicating with its neighbors, each agent collects its streaming data and formulates its own local objective function in the observation stage. In the computation stage, each agent carries out local updates based on the observed data, local objective function, and state variables.

B. RF Mapping

RF mapping is proposed to make kernel methods scalable for large datasets [12]. For a shift-invariant kernel that satisfies $\kappa(\mathbf{x}_t, \mathbf{x}_\tau) = \kappa(\mathbf{x}_t - \mathbf{x}_\tau)$, $\forall t$, $\forall \tau$, if $\kappa(\mathbf{x}_t - \mathbf{x}_\tau)$ is absolutely integrable, then its Fourier transform $p_\kappa(\boldsymbol{\omega})$ is guaranteed to be nonnegative ($p_\kappa(\boldsymbol{\omega}) \geq 0$), and hence can be viewed as its probability density function (pdf) when κ is scaled to satisfy $\kappa(0) = 1$ [32]. Therefore, we have

$$\begin{aligned}\kappa(\mathbf{x}_t, \mathbf{x}_\tau) &= \int p_\kappa(\boldsymbol{\omega}) e^{j\boldsymbol{\omega}^\top(\mathbf{x}_t - \mathbf{x}_\tau)} d\boldsymbol{\omega} \\ &= \mathbb{E}_{\boldsymbol{\omega}}[\phi(\mathbf{x}_t, \boldsymbol{\omega})\phi^*(\mathbf{x}_\tau, \boldsymbol{\omega})] \qquad (1)\end{aligned}$$

where $\mathbb{E}$ denotes the expectation operator, $\phi(\mathbf{x}, \boldsymbol{\omega}) := e^{j\boldsymbol{\omega}^\top \mathbf{x}}$ with $\boldsymbol{\omega} \in \mathbb{R}^d, \mathbf{x} \in \mathbb{R}^d$, and $*$ being the complex conjugate operator. In (1), the first equality is the result of the Fourier inversion theorem, and the second equality arises by viewing $p_\kappa(\boldsymbol{\omega})$ as the pdf of $\boldsymbol{\omega}$.

The main idea of the RF mapping method is to approximate the kernel function $\kappa(\mathbf{x}_t, \mathbf{x}_\tau)$ by the sample average

$$\hat{\kappa}_L(\mathbf{x}_t, \mathbf{x}_\tau) := \frac{1}{L}\sum_{l=1}^{L} \phi(\mathbf{x}_t, \boldsymbol{\omega}_l)\phi^*(\mathbf{x}_\tau, \boldsymbol{\omega}_l) \qquad (2)$$

where $\{\boldsymbol{\omega}_l\}_{l=1}^L$ are randomly drawn from the distribution $p_\kappa(\boldsymbol{\omega})$. For implementation, the following real-valued mapping is usually adopted:

$$\phi(\mathbf{x}, \boldsymbol{\omega}) = [\cos(\boldsymbol{\omega}^\top \mathbf{x}), \sin(\boldsymbol{\omega}^\top \mathbf{x})]^\top. \qquad (3)$$

III. Problem Statement

Consider the network model described in Section II-A, each agent in the network only has access to its locally observed data composed of independently and identically distributed (i.i.d) input-label pairs $\{\mathbf{x}_{i,t}, y_{i,t}\}_{t=1}^T$ obeying an unknown probability distribution p on $\mathcal{X} \times \mathcal{Y}$, with $\mathbf{x}_{i,t} \in \mathbb{R}^d$ and $y_{i,t} \in \mathbb{R}$. The decentralized learning task is to find a nonlinear prediction function f such that $y_{i,t} = f(\mathbf{x}_{i,t}) + e_{i,t}$ for $\{\{\mathbf{x}_{i,t}, y_{i,t}\}_{t=1}^T\}_{i=1}^N$, where the error term $e_{i,t}$ is minimized accordingly to certain optimality metric. This is usually achieved by minimizing the empirical risk

$$f^\star = \arg\min_{f\in\Omega} \sum_{i=1}^{N}\sum_{t=1}^{T} \ell(f(\mathbf{x}_{i,t}), y_{i,t}) + \lambda\|f\|_\Omega^2 \qquad (4)$$

where $\ell(\cdot, \cdot)$ is a nonnegative loss function, Ω is the function space f belongs to, and $\lambda > 0$ is a regularization parameter that controls overfitting. For regression problems, a common loss function is the quadratic loss. For binary classifications, the common loss functions are the hinge loss $\ell(y, \hat{y}) = \max(0, 1 - y\hat{y})$ and the logistic loss $\ell(y, \hat{y}) = \log(1 + e^{-y\hat{y}})$.

Assume f belongs to the RKHS $\mathcal{H} := \{f|f(\mathbf{x}) = \sum_{t=1}^\infty \alpha_t \kappa(\mathbf{x}, \mathbf{x}_t)\}$ induced by a shift-invariant positive semidefinite kernel $\kappa(\mathbf{x}, \mathbf{x}_t) : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$. Thus, the optimal solution of (4) admits

$$\hat{f}_\kappa^\star(\mathbf{x}) = \sum_{i=1}^{N}\sum_{t=1}^{T} \alpha_{i,t}\kappa(\mathbf{x}, \mathbf{x}_{i,t}) := \boldsymbol{\alpha}^\top \boldsymbol{\kappa}(\mathbf{x}) \qquad (5)$$

where $\boldsymbol{\alpha} = [\alpha_{1,1}, \ldots, \alpha_{N,T}]^\top \in \mathbb{R}^{NT}$ is the coefficient vector to be learned and $\boldsymbol{\kappa}(\mathbf{x}) = [\kappa(\mathbf{x}, \mathbf{x}_{1,1}), \ldots, \kappa(\mathbf{x}, \mathbf{x}_{N,T})]^\top$. The kernel function can be linear, Gaussian, or Laplacian. In this article, we adopt a Gaussian kernel $\kappa(\mathbf{x}, \mathbf{x}_t) = \exp(-\|\mathbf{x} - \mathbf{x}_t\|_2^2/(2\sigma^2))$ with a pre-defined bandwidth σ.

Notice that the parameter $\boldsymbol{\alpha}$ to be learned is data dependent and its size grows linear with the number of data points. This incurs two problems. First, since the size of $\boldsymbol{\alpha}$ grows linearly with T, the computational complexity of estimating $\boldsymbol{\alpha}$ becomes an issue when T grows large. Second, since $\boldsymbol{\alpha}$ is data dependent, to learn a common functional model represented by (5) in the decentralized network means that raw data $\{\mathbf{x}_{i,t}, y_{i,t}\}, \forall i, \forall t$ are also required to be communicated, which raises the privacy concerns, especially when the raw data contains sensitive information [9].

To circumvent the curse of dimensionality issue and prevent raw data exchange, we adopt the RF mapping method described in Section II-B. For the adopted Gaussian kernel, its pdf is a normal distribution with $p_\kappa(\boldsymbol{\omega}) \sim \mathbf{N}(\mathbf{0}, \sigma^{-2}\mathbf{I})$. Then, the function $f^\star$ to be learned in (4) can be approximated by the following representation:

$$\hat{f}^\star(\mathbf{x}) = \boldsymbol{\theta}^\top \boldsymbol{\phi}_L(\mathbf{x}) \qquad (6)$$

where $\boldsymbol{\theta} \in \mathbb{R}^{2L}$ is the decision vector to be learned in the RF space, and $\boldsymbol{\phi}_L(\mathbf{x})$ is the mapped data in the RF space using (3)

$$\boldsymbol{\phi}_L(\mathbf{x}) := \sqrt{\frac{1}{L}}[\phi(\mathbf{x}, \boldsymbol{\omega}_1), \ldots, \phi(\mathbf{x}, \boldsymbol{\omega}_L)]^\top. \qquad (7)$$

Here, the new decision variable $\boldsymbol{\theta}$ is data independent. Therefore, when nodes in the decentralized network communicate, they only need to communicate the new parameter $\boldsymbol{\theta}$, as shown by Algorithms 1 and 2. Moreover, the size of $\boldsymbol{\theta}$ is fixed and determined by the number of RFs chosen, which is usually much smaller than the number of data points ($L \ll T$). In addition, the mapping from $\mathbf{x}$ to $\boldsymbol{\phi}_L(\mathbf{x})$ is not one to one, which further relieves the sensitive raw data leakage problem.

With the approximation (6), the decentralized kernel learning problem is formulated as follows:

$$\begin{aligned}&\min_{\{\boldsymbol{\theta}_i, z_{ij}\}} \sum_{i=1}^{N}\left[\sum_{t=1}^{T} \ell\left(\boldsymbol{\theta}_i^\top \boldsymbol{\phi}_L(\mathbf{x}_{i,t}), y_{i,t}\right) + \frac{\lambda}{N}\|\boldsymbol{\theta}_i\|^2\right]\\ &\text{s.t. } \boldsymbol{\theta}_i = z_{ij}, \quad \boldsymbol{\theta}_j = z_{ij} \quad \forall (i,j) \in \mathcal{A}\end{aligned} \tag{8}$$

where $\boldsymbol{\theta}_i$ and $\boldsymbol{\theta}_j$ are the local copies of the global parameter $\boldsymbol{\theta}$ associated with agents $i, j \in \mathcal{N}$, respectively. The constraint in (8) enforces the consensus constraint on neighboring agents i and j using an auxiliary variable z_{ij}. The optimization problem can then be solved using DKLA proposed in [9]. A communication-censored algorithm (COKE) is also proposed in [9] to improve the communication efficiency of DKLA.

However, both DKLA and COKE operate in batch form when all data are available. Whereas in many real-life applications, function learning tasks are expected to perform in an online fashion with sequentially arriving data. In this article, we consider the case that each agent collects the data points $\{\mathbf{x}_{i,t}, y_{i,t}\}_{t=1}^{T}, \forall i$ in an online fashion, and the parameter is estimated based on instantaneous data samples. To achieve an optimal sublinear regrets from the optimal performance of (8), we customize the general online decentralized ADMMs algorithm proposed in [4] to decentralized online kernel learning to efficiently solve the online kernel learning problem over a decentralized network. At every time t, decentralized online kernel learning (approximately) solves an optimization problem to obtain the update $\boldsymbol{\theta}_{i,t+1}$ from the current decision $\boldsymbol{\theta}_{i,t}$ and the newly arrived data

$$\begin{aligned}&\min_{\{\boldsymbol{\theta}_i, z_{ij}\}} \sum_{i=1}^{N} \mathcal{L}_{i,t}(\boldsymbol{\theta}_i) + \frac{\eta_t}{2}\sum_{i=1}^{N}\|\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i,t}\|^2\\ &\text{s.t. } \boldsymbol{\theta}_i = z_{ij}, \quad \boldsymbol{\theta}_j = z_{ij} \quad \forall (i,j) \in \mathcal{A}\end{aligned} \tag{9}$$

where $\mathcal{L}_{i,t}(\boldsymbol{\theta}_i) := \ell(\boldsymbol{\theta}_i^\top \boldsymbol{\phi}_L(\mathbf{x}_{i,t}), y_{i,t}) + (\lambda/N)\|\boldsymbol{\theta}_i\|^2$ is the local instantaneous cost function dependent of the new data only, whereas $\boldsymbol{\theta}_{i,t}$ captures the influence of all the past data.

In Section IV, we first propose a computation-efficient algorithm to solve (9). We then utilize communication-censoring and quantization strategies to improve the communication efficiency of the proposed algorithm.

IV. Algorithm Development

In this section, we first utilize linearized ADMM to efficiently solve (9) and then add the censoring and quantization techniques to develop a communication-efficient decentralized online kernel learning algorithm.

For notational clarity, we define $\boldsymbol{\Theta} = [\boldsymbol{\theta}_1^\top; \boldsymbol{\theta}_2^\top; \ldots; \boldsymbol{\theta}_N^\top] \in \mathbb{R}^{N\times 2L}$ that contains all the local copies $\boldsymbol{\theta}_i$ and $\boldsymbol{Z} = [\cdots; z_{ij}^\top; \cdots] \in \mathbb{R}^{2r\times 2L}$. We further define the aggregated function as $\mathcal{L}_t(\boldsymbol{\Theta}) := \sum_{i=1}^{N}\mathcal{L}_{i,t}(\boldsymbol{\theta}_i)$. With these definitions, we rewrite (9) in a matrix form for the $\boldsymbol{\Theta}_{t+1}$ update

$$\begin{aligned}&\min_{\{\boldsymbol{\Theta}, \boldsymbol{Z}\}} \mathcal{L}_t(\boldsymbol{\Theta}) + \frac{\eta_t}{2}\|\boldsymbol{\Theta} - \boldsymbol{\Theta}_t\|^2\\ &\text{s.t. } \boldsymbol{A}\boldsymbol{\Theta} + \boldsymbol{B}\boldsymbol{Z} = \mathbf{0}_{4r\times 2L}\end{aligned} \tag{10}$$

where $\boldsymbol{A} = (1/2)[\mathbf{S}_+^\top + \mathbf{S}_-^\top; \mathbf{S}_+^\top - \mathbf{S}_-^\top] \in \mathbb{R}^{4r\times N}$ and $\boldsymbol{B} = [-\boldsymbol{I}_{2r}; -\boldsymbol{I}_{2r}]$.

A. ODKLA: Online Decentralized Kernel Learning via Linearized ADMM

Standard ADMM to solve (10) starts from formulating the augmented Lagrangian of (10) as follows:

$$\begin{aligned}\mathbb{L}_t(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Lambda}) &= \mathcal{L}_t(\boldsymbol{\Theta}) + \frac{\eta_t}{2}\|\boldsymbol{\Theta} - \boldsymbol{\Theta}_t\|_F^2\\ &\quad + \langle \boldsymbol{\Lambda}, \boldsymbol{A}\boldsymbol{\Theta} + \boldsymbol{B}\boldsymbol{Z}\rangle + \frac{\rho}{2}\|\boldsymbol{A}\boldsymbol{\Theta} + \boldsymbol{B}\boldsymbol{Z}\|_F^2\end{aligned} \tag{11}$$

where ρ is the penalty parameter, $\boldsymbol{\Lambda} = [\boldsymbol{\beta}; \boldsymbol{\lambda}] \in \mathbb{R}^{4r\times 2L}$ is the Lagrange multiplier associated with the constraint $\boldsymbol{A}\boldsymbol{\Theta} + \boldsymbol{B}\boldsymbol{Z} = \mathbf{0}$. At time t, the updates of the primal variables $\boldsymbol{\Theta}_{t+1}$, $\boldsymbol{Z}_{t+1}$ and the dual variable $\boldsymbol{\Lambda}_{t+1}$ are sequentially given by

$$\boldsymbol{\Theta}_{t+1} := \arg\min_{\boldsymbol{\Theta}} \mathbb{L}_t(\boldsymbol{\Theta}, \boldsymbol{Z}_t, \boldsymbol{\Lambda}_t) \tag{12}$$

$$\boldsymbol{Z}_{t+1} := \arg\min_{\boldsymbol{Z}} \mathbb{L}_t(\boldsymbol{\Theta}_{t+1}, \boldsymbol{Z}, \boldsymbol{\Lambda}_t) \tag{13}$$

$$\boldsymbol{\Lambda}_{t+1} = \boldsymbol{\Lambda}_t + \rho(\boldsymbol{A}\boldsymbol{\Theta}_{t+1} + \boldsymbol{B}\boldsymbol{Z}_{t+1}). \tag{14}$$

Note that given the instantaneous loss $\mathcal{L}_t$, iterates (12)–(14) only run once, and thus the optimization problem in (10) is only approximately solved.

It has been proven in [4] that with initializations $\boldsymbol{\beta}_1 = -\boldsymbol{\lambda}_1$, and $\boldsymbol{Z}_1 = (1/2)\mathbf{S}_+^\top\boldsymbol{\Theta}_1$, the update of the auxiliary variable $\boldsymbol{Z}_t$ is not necessary and the Lagrange multiplier $\boldsymbol{\Lambda}$ can be replaced by a lower dimensional variable $\boldsymbol{\Gamma} := [\gamma_1^\top; \ldots; \gamma_N^\top] \in \mathbb{R}^{N\times 2L}$. The simplified updates of ADMM for general online decentralized optimization refer to [4]. Though simplified, the general decentralized ADMM still involves solving local optimization problems for the primal variables update, thus is computational intensive.

To reduce the computation complexity of ADMM, we replace $\mathcal{L}_t(\boldsymbol{\Theta})$ in (12) by its linear approximation $\mathcal{L}_t(\boldsymbol{\Theta}_t) + \langle \partial\mathcal{L}_t(\boldsymbol{\Theta}_t), \boldsymbol{\Theta} - \boldsymbol{\Theta}_t\rangle$ at $\boldsymbol{\Theta} = \boldsymbol{\Theta}_t$, and develop the ODKLA algorithm where the iterates of $\boldsymbol{\Theta}_{t+1}$ and $\boldsymbol{\Gamma}_{t+1}$ are generated by the simplified recursions

$$\begin{aligned}\boldsymbol{\Theta}_{t+1} = (\eta_t\mathbf{I} + 2\rho\boldsymbol{D})^{-1}\Big[&(\rho(\boldsymbol{D} + \boldsymbol{W}) + \eta_t\mathbf{I})\boldsymbol{\Theta}_t\\ &- \boldsymbol{\Gamma}_t - \partial\mathcal{L}_t(\boldsymbol{\Theta}_t)\Big]\end{aligned} \tag{15}$$

$$\boldsymbol{\Gamma}_{t+1} = \boldsymbol{\Gamma}_t + \rho(\boldsymbol{D} - \boldsymbol{W})\boldsymbol{\Theta}_{t+1}. \tag{16}$$

The ODKLA algorithm can be implemented distributedly. Specifically, each agent i only needs to update a primal

Algorithm 1 ODKLA (Run at Agent i)

```
Require: Kernel $\kappa$, hyperparameters $(L, \eta_t)$, initialize local
    variables to $\boldsymbol{\theta}_{i,1} = \mathbf{0}$, and $\gamma_{i,1} = 0$.
Draw $L$ i.i.d. samples $\{\boldsymbol{\omega}_l\}_{l=1}^{L}$ from $p_\kappa(\boldsymbol{\omega})$ according to a
    common random seed.
for iterations $t = 1, 2, \ldots, T$ do
    Receive a streaming data $(\mathbf{x}_{i,t}, y_{i,t})$
    Construct $\boldsymbol{\phi}(\mathbf{x}_{i,t})$ via (7).
    Update local primal variable $\boldsymbol{\theta}_{i,t+1}$ via (17).
    Transmit $\boldsymbol{\theta}_{i,t+1}$ to neighbors and receive $\boldsymbol{\theta}_{j,t+1}$ from
        neighbors $j \in \mathcal{N}_i$.
    Update local dual variable $\gamma_{i,t+1}$ via (18).
end for
```

variable $\boldsymbol{\theta}_i$ and a dual variable γ_i with the following iterations:

$$\boldsymbol{\theta}_{i,t+1} = \boldsymbol{\theta}_{i,t} - \frac{1}{\eta_t + 2\rho d_i}\Bigg[\partial\mathcal{L}_{i,t}(\boldsymbol{\theta}_{i,t}) + \rho\sum_{j\in\mathcal{N}_i}(\boldsymbol{\theta}_{i,t} - \boldsymbol{\theta}_{j,t}) + \gamma_{i,t}\Bigg] \tag{17}$$

$$\gamma_{i,t+1} = \gamma_{i,t} + \rho\sum_{j\in\mathcal{N}_i}(\boldsymbol{\theta}_{i,t+1} - \boldsymbol{\theta}_{j,t+1}). \tag{18}$$

Note that with linearized ADMM, at each time t, ODKLA has closed-form solutions for all agents to update their primal variables, instead of solving optimization problems as in (12). Thus, the computational efficiency is improved. The ODKLA algorithm is outlined in Algorithm 1.

Remark 1: Our paper shares similar problem formulation (9) as [20] since our algorithms are both developed from the general decentralized online ADMM framework proposed by [4]. However, our methods differ from [20] in two ways. First, we utilize linearized ADMM to solve the decentralized kernel learning problem while [20] adopts the standard ADMM method. Compared with [20], our algorithms enjoy light computation. Second, we also develop the communication-efficient algorithm in the next section using quantization and communication-censoring strategies while the communication efficiency is not discussed in [20].

B. QC-ODKLA: Quantized and Communication-Censored ODKLA

ODKLA resolves the challenges caused by streaming data in decentralized network setting in a computationally efficient manner. However, as seen in (17) and (18), agents communicate all the time which causes low communication efficiency. Thus, we introduce communication-censoring and quantization strategies to deal with the limited communication resource situation and develop the QC-ODKLA algorithm.

To start, we introduce a new state variable $\hat{\boldsymbol{\theta}}_{i,t}$ for each agent i to record its latest broadcast primal variable up to time t. Then, the difference between agent i's updated state $\boldsymbol{\theta}_{i,t+1}$ and its previously transmitted state $\hat{\boldsymbol{\theta}}_{i,t}$ at time t is defined as follows:

$$\boldsymbol{h}_{i,t} = \boldsymbol{\theta}_{i,t+1} - \hat{\boldsymbol{\theta}}_{i,t}. \tag{19}$$

We then introduce an evaluation function

$$H_{i,t} = \|\boldsymbol{h}_{i,t}\|_2 - \alpha\beta^t \tag{20}$$

to evaluate if the local updates $\boldsymbol{\theta}_{i,t+1}$ are informative enough to be transmitted, with predefined positive constants $\alpha > 0$ and $\beta < 1$. If $H_{i,t} \geq 0$, then $\boldsymbol{\theta}_{i,t+1}$ is deemed informative, and agent i is allowed to transmit a quantized update $Q(\boldsymbol{\theta}_{i,t+1})$ to its neighbors. Here, the quantization is introduced to reduce the communication cost from the perspective of bit numbers per transmission. To facilitate the measurement and analysis of the impact of quantization, we adopt the difference-based quantization scheme proposed in [33]. That is, at time t, instead of quantizing $\boldsymbol{\theta}_{i,t+1}$, we quantize the difference $\boldsymbol{h}_{i,t}$. Specifically, for each element $h_{i,t}^l, l = 1, \ldots, 2L$ within the range of $[u, v)$, if we restrict the number of transmission bits to be b, then we can evenly divide the range $[u, v)$ to be $q = 2^b$ intervals of equal length $\Delta = (v-u)/q$. Then the rounding quantizer $Q(\cdot)$ applied to $h_{i,t}^l$ outputs

$$Q(h_{i,t}^l) = u + \left(\left\lfloor\frac{h_{i,t}^l - u}{\Delta}\right\rfloor + \frac{1}{2}\right)\Delta \tag{21}$$

where $\lfloor\cdot\rfloor$ is the floor operation. In practice, it is not necessary to transmit $Q(h_{i,t}^l)$, instead, we can simply transmit the integer $k := \lfloor(h_{i,t}^l - u/\Delta)\rfloor$ using the b bits. Thus, the total number of bits for agent i to transmit the quantized difference $Q(\boldsymbol{h}_{i,t})$ to its neighbors is only $2Lb$ bits.

The whole communication process thus involves three parts: evaluation, quantization, and states update. If $H_{i,t} \geq 0$, then $\boldsymbol{\theta}_{i,t+1}$ is deemed informative, and agent i is allowed to transmit a quantized difference $Q(\boldsymbol{h}_{i,t})$ to its neighbors and updates its local state as $\hat{\boldsymbol{\theta}}_{i,t+1} = \hat{\boldsymbol{\theta}}_{i,t} + Q(\boldsymbol{h}_{i,t})$. Otherwise, $\boldsymbol{\theta}_{i,t+1}$ is censored, agent i sets $\hat{\boldsymbol{\theta}}_{i,t+1} = \hat{\boldsymbol{\theta}}_{i,t}$, and no information is transmitted. Similarly, upon receiving $Q(\boldsymbol{h}_{j,t})$ from its neighbor j, agent i updates the state variables of its neighbor's as $\hat{\boldsymbol{\theta}}_{j,t+1} = \hat{\boldsymbol{\theta}}_{j,t} + Q(\boldsymbol{h}_{j,t})$, otherwise, $\hat{\boldsymbol{\theta}}_{j,t+1} = \hat{\boldsymbol{\theta}}_{j,t}$.

With the communication-censoring rule and quantization scheme, the primal and dual updates in (17) and (18) become

$$\boldsymbol{\theta}_{i,t+1} = \boldsymbol{\theta}_{i,t} - \frac{1}{\eta_t + 2\rho d_i}\Bigg[\partial\mathcal{L}_{i,t}(\boldsymbol{\theta}_{i,t}) + \rho\sum_{j\in\mathcal{N}_i}(\hat{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{j,t}) + \gamma_{i,t}\Bigg] \tag{22}$$

$$\gamma_{i,t+1} = \gamma_{i,t} + \rho\sum_{j\in\mathcal{N}_i}\left(\hat{\boldsymbol{\theta}}_{i,t+1} - \hat{\boldsymbol{\theta}}_{j,t+1}\right) \tag{23}$$

and the total numbers of transmissions and bits are both reduced in the optimization and learning process. We summarize the QC-ODKLA algorithm in Algorithm 2.

Algorithm 2 QC-ODKLA (Run at Agent i)

```
Require: Kernel $\kappa$, hyperparameters $(L, \rho, \alpha, \beta)$, initialize local variables to $\boldsymbol{\theta}_{i,1} = \mathbf{0}$, and $\gamma_{i,1} = \mathbf{0}$, $\hat{\boldsymbol{\theta}}_{i,1} = Q(\boldsymbol{\theta}_{i,1})$ and $\hat{\boldsymbol{\theta}}_{j,1} = Q(\boldsymbol{\theta}_{j,1})$ for all $j \in \mathcal{N}_i$.
Draw $L$ i.i.d. samples $\{\boldsymbol{\omega}_l\}_{l=1}^{L}$ from $p_\kappa(\boldsymbol{\omega})$ according to a common random seed.
for iterations $t = 1, 2, \ldots, T$ do
    Receive a streaming data $(\mathbf{x}_{i,t}, y_{i,t})$
    Construct $\boldsymbol{\phi}(\mathbf{x}_{i,t})$ via (7).
    Update local primal variable $\boldsymbol{\theta}_{i,t+1}$ by solving (22).
    Calculate the difference $\boldsymbol{h}_{i,t}$ via (19) and quantize it as $Q(\boldsymbol{h}_{i,t})$ via (21).
    If (20) is nonnegative, transmit $Q(\boldsymbol{h}_{i,t})$ to neighbors and set $\hat{\boldsymbol{\theta}}_{i,t+1} = \hat{\boldsymbol{\theta}}_{i,t} + Q(\boldsymbol{h}_{i,t})$. Else, set $\hat{\boldsymbol{\theta}}_{i,t+1} = \hat{\boldsymbol{\theta}}_{i,t}$ and do not transmit.
    If receiving $Q(\boldsymbol{h}_{j,t})$ from neighbors $j$, update $\hat{\boldsymbol{\theta}}_{j,t+1} = \hat{\boldsymbol{\theta}}_{j,t} + Q(\boldsymbol{h}_{j,t})$. Else, set $\hat{\boldsymbol{\theta}}_{j,t+1} = \hat{\boldsymbol{\theta}}_{j,t}$.
    Update local dual variable $\gamma_{i,t+1}$ via (23).
end for
```

V. Regret Analysis

In this section, we analyze the regret bound of QC-ODKLA. As in [18], we define the cumulative network regret of online decentralized learning as follows:

$$\mathcal{R}(T) = \sum_{t=1}^{T}\sum_{i=1}^{N}\big(\mathcal{L}_{i,t}(\boldsymbol{\theta}_{i,t}) - \mathcal{L}_{i,t}(\boldsymbol{\theta}^\star)\big) \tag{24}$$

where $\boldsymbol{\theta}^\star$ is the optimal solution of (8) that assumes all data are available. We prove that QC-ODKLA achieves the optimal sublinear regret $\mathcal{O}(\sqrt{T})$ for convex local cost functions $\mathcal{L}_{i,t}$. Since ODKLA is a special case of QC-ODKLA where both the quantization and communication-censoring strategies are absent, the regret analysis of QC-ODKLA extends to ODKLA straightforwardly. The following commonly used assumptions are adopted.

Assumption 1: The local cost functions $\mathcal{L}_{i,t}(\boldsymbol{\theta})$ are convex and differentiable with respect to $\boldsymbol{\theta}$. Also, assume the gradients of the local cost functions are Lipschitz continuous with constants $C_{\mathcal{L}_i} > 0, \forall\, i$. That is, $\|\partial\mathcal{L}_{i,t}(\boldsymbol{\theta})\|_2 \le C_{\mathcal{L}_i}, \forall\, i$. The maximum Lipschitz constant is $C_{\mathcal{L}} := \max_i C_{\mathcal{L}_i}$.

Assumption 2: The estimates $\boldsymbol{\theta}_{i,t}$ and the optimal solution $\boldsymbol{\theta}^\star$ of (8) are bounded. That is, $\|\boldsymbol{\theta}_{i,t}\|_2 \le C_\theta$, and $\|\boldsymbol{\theta}^\star\|_2 \le C_\theta$.

Note that all assumptions are standard in online decentralized kernel learning [18], [19], [20]. The convexity of local cost functions are easily satisfied in learning problems if the local cost functions are square loss or the hinge loss.

To study the regret bound for QC-ODKLA, we notice that the difference of QC-ODKLA and ODKLA is the communication-censoring step and quantization step in the communication stage, which introduces an error if an update is censored and/or quantized in a transmission. Define the introduced error for agent i at time t as follows:

$$\boldsymbol{e}_{i,t} := \boldsymbol{\theta}_{i,t} - \hat{\boldsymbol{\theta}}_{i,t}. \tag{25}$$

Then, the overall introduced error at time t can be concatenated as $\boldsymbol{E}_t := [\boldsymbol{e}_{1,t}^\top; \boldsymbol{e}_{2,t}^\top; \ldots; \boldsymbol{e}_{N,t}^\top]$. We first show that the overall introduced error in QC-ODKLA is upper bounded by the quantization error and the pre-defined threshold parameters.

Lemma 1: For the updates (22) and (23), under Assumptions 1 and 2, if the quantized difference $Q(\boldsymbol{h}_{i,t})$ is only allowed to transmit when $H_{i,t} \ge 0$ for the pre-defined threshold parameters α and β, then, for any time $t > 0$, the overall error introduced in the QC-ODKLA is upper bounded by

$$\|\boldsymbol{E}_t\|_F^2 \le \zeta := \max\{\sqrt{N}\alpha\beta, \sqrt{2NL}\Delta/2\} \tag{26}$$

where Δ is the length of the quantization interval.

Proof: Define $\delta\hat{\boldsymbol{\theta}}_{i,t} = \hat{\boldsymbol{\theta}}_{i,t} - \hat{\boldsymbol{\theta}}_{i,t-1}$, the introduced error for each agent i can be represented as follows:

$$\begin{aligned}\boldsymbol{e}_{i,t} &= \boldsymbol{\theta}_{i,t} - \hat{\boldsymbol{\theta}}_{i,t}\\ &= \boldsymbol{\theta}_{i,t} - \hat{\boldsymbol{\theta}}_{i,t-1} - \delta\hat{\boldsymbol{\theta}}_{i,t}\\ &= \boldsymbol{h}_{i,t-1} - \delta\hat{\boldsymbol{\theta}}_{i,t}.\end{aligned} \tag{27}$$

According to the censoring rule, if $\|\boldsymbol{h}_{i,t-1}\|_2 \ge \alpha\beta^{t-1}$ for $t \ge 1$, we have $\delta\hat{\boldsymbol{\theta}}_{i,t} = Q(\boldsymbol{h}_{i,t-1})$, which implies $\|\boldsymbol{e}_{i,t}\|_2 = \|\boldsymbol{h}_{i,t-1} - Q(\boldsymbol{h}_{i,t-1})\|_2 \le \sqrt{2L}\Delta/2$. Otherwise, if $\|\boldsymbol{h}_{i,t-1}\|_2 < \alpha\beta^{t-1}$ for $t \ge 1$, we have $\delta\hat{\boldsymbol{\theta}}_{i,t} = \mathbf{0}$, which implies $\|\boldsymbol{e}_{i,t}\|_2 = \|\boldsymbol{h}_{i,t-1}\|_2 \le \alpha\beta^{t-1} \le \alpha\beta$ since $\beta < 1$. Therefore, the overall introduced error $\|\boldsymbol{E}_t\|_F^2 \le \max\{\sqrt{N}\alpha\beta, \sqrt{2NL}\Delta/2\}$. ■

With Lemma 1, we are ready to establish the network regret bound of QC-ODKLA.

Theorem 1: Under Assumptions 1 and 2, if the quantized difference $Q(\boldsymbol{h}_{i,t})$ is only allowed to transfer when $H_{i,t} \ge 0$ for the pre-defined threshold parameters $\alpha > 0$ and $\beta < 1$, then, for any time $t > 0$, the cumulative network regret (24) generated by the updates (22) and (23) satisfies

$$\mathcal{R}(T) \le \left(\sqrt{N}C_\theta + \frac{1}{\sigma_{\max}^2(\mathbf{S}_-)}C_{\mathcal{L}} + \sigma_{\max}^2(\mathbf{S}_-)\zeta\right)\mathcal{O}\left(\sqrt{T}\right) \tag{28}$$

if $\eta_t = \rho = 1/\mathcal{O}(\sqrt{T})$.

Proof: See Appendix. ■

Remark 2: Note that in addition to the network size (N) and topology ($\mathbf{S}_-$), the communication-censoring and quantization strategies (incorporated in ζ) also affect the cumulative network regret, which creates a trade-off between the communication efficiency and the online learning performance.

VI. Experiments

This section evaluates the effectiveness of our proposed QC-ODKLA algorithm in saving communication and computation resources on various online regression tasks with real-world datasets. The loss function is assumed to be a squared loss, resulting in the following instantaneous cost function for agent $i, \forall i \in \mathcal{N}$:

$$\mathcal{L}_{i,t}(\boldsymbol{\theta}_i) := (\boldsymbol{\theta}_i^\top\boldsymbol{\phi}_L(\mathbf{x}_{i,t}) - y_{i,t})^2 + \frac{\lambda}{N}\|\boldsymbol{\theta}_i\|^2. \tag{29}$$

For the sake of comparison, the following decentralized online learning algorithms will be used for our experiments.

1) *RFF-DOKL:* The decentralized online kernel learning algorithm that is developed based on online gradient descent and a diffusion strategy [18].

2) *DOKL:* The decentralized online kernel learning algorithm that is developed based on online ADMM [20].
3) *ODKLA:* The proposed decentralized online kernel learning that is developed based on linearized ADMM.
4) *QC-ODKLA:* The proposed communication-efficient decentralized online kernel learning that is developed based on linearized ADMM. The communication efficiency is achieved by communication censoring and quantization.

There are other efficient kernel-based learning algorithms, for example, the parsimonious kernel learning method [11], the Nyström method [15], the multi-kernel learning algorithms [19], [20], etc. However, since we consider the case that data are only locally available and cannot be shared among agents to ensure raw data privacy, the parsimonious kernel learning method and the Nyström method are excluded for comparison. Although learning with multi-kernel has more flexibility and may achieve better learning performance than utilizing a single kernel, it is not our objective to compare the learning performance of single and multiple kernels in this article. We remark that the multi-kernel learning scheme can be easily incorporated into our work though, as proposed by [20], to achieve both high learning performance and communication-computation efficiency.

A. Description of Online Datasets and Experimental Settings

The regression tasks are carried out on six datasets available at the UCI machine learning repository [34]. The detailed descriptions of the six datasets are listed below.

1) *Tom's Hardware:* This dataset contains $T_{\text{total}} = 11\,000$ samples with $\mathbf{x}_t \in \mathbb{R}^{96}$ including the number of created discussions and authors interacting of a topic and $y_t \in \mathbb{R}$ representing the average number of display to a visitor about that topic [35].
2) *Twitter:* This dataset consists of $T_{\text{total}} = 98\,700$ samples with $\mathbf{x}_t \in \mathbb{R}^{77}$ being a feature vector reflecting the number of new interactive authors and the lengths of discussions on a given topic, etc., and $y_t \in \mathbb{R}$ representing the average number of active discussions on a certain topic. The learning task is to predict the popularity of these topics [35].
3) *Energy:* This dataset contains $T_{\text{total}} = 18\,600$ samples with $\mathbf{x}_t \in \mathbb{R}^{28}$ describing the humidity and temperature in different areas of the house, pressure, wind speed, and viability outside, while y_t denotes the total energy consumption in the house [36].
4) *Air Quality:* This dataset contains $T_{\text{total}} = 7320$ samples measured by a gas multi-sensor device in an Italian city, where $\mathbf{x}_t \in \mathbb{R}^{13}$ represents the hourly concentration of CO, NOx, NO2, etc, while y_t denotes the concentration of polluting chemicals in the air [37].
5) *Conductivity:* This dataset contains $T_{\text{total}} = 21\,260$ samples extracted from superconductors, where $\mathbf{x}_t \in \mathbb{R}^{81}$ represents critical information to construct superconductor such as density and mass of atoms. The task is to predict the critical temperature, which creates superconductor [38].
6) *Blood Data:* This dataset contains $T_{\text{total}} = 61\,000$ samples recorded by patient monitors at different hospitals, where $\mathbf{x}_t \in \mathbb{R}^2$ and the goal is to predict the blood pressure based on several physiological parameters from photoplethysmography and electrocardiogram signals [39].

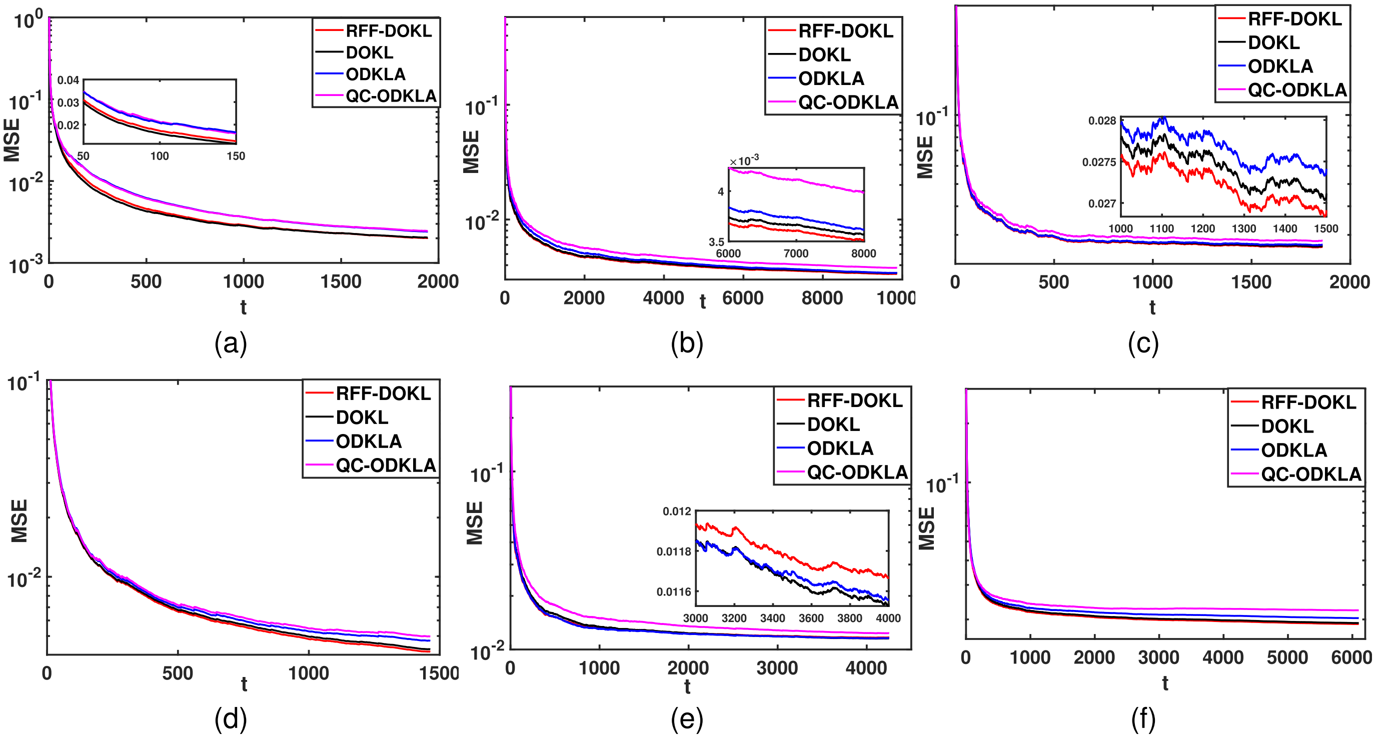
All experiments are conducted using MATLAB 2021 on an Intel CPU @ 3.6 GHz (32 GB RAM) desktop. For each dataset, the T_{total} data samples are randomly shuffled and then partitioned among N nodes so that each node has $T = T_{\text{total}}/N$ samples. The features in data are normalized so that all values are between 0 and 1. Throughout the simulation, we adopt the Gaussian kernel for our learning tasks, whose bandwidth is fine tuned through grid search for each task. The Gaussian kernel bandwidth is fined tuned to be $\sigma = 0.5$ for Tom's hardware, Twitter, air quality, and blood datasets. For conductivity and energy datasets, $\sigma = 1$ and $\sigma = 0.1$, respectively. The regularization parameter $\lambda = 10^{-4}$. The number of RFs adopted for RF approximation is $L = 50$ throughout the simulations, which is the same as [20]. The stepsize ρ and η_t are fine tuned via grid search for each method and each dataset individually. The censoring threshold parameters are $\alpha = 2, \beta = 0.9$ for energy data, and $\alpha = 4, \beta = 0.99$ for all the other datasets. The quantization level for QC-ODKLA algorithm is set to be $q = 8$ to achieve a balance in saving communication and good learning performance.

The connected graphs are randomly generated with $N = 5$ or $N = 10$ nodes, each with a moderate connection. The probability of attachment per node equals to 0.5, that is, any pair of two nodes are connected with a probability of 0.5. For Twitter, conductivity, and blood datasets, whose datasize are large, we use a 10-node network. The remaining datasets use a 5-node network.

B. Performance Evaluations

We demonstrate the effectiveness of our proposed QC-ODKLA algorithm from three aspects, that is: 1) the learning performance in terms of mean-squared-error (MSE); 2) the communication efficiency in terms of the total number of communications triggered and total number of bits transferred; and 3) the computation efficiency in terms of the running time to conduct each learning task. For fair comparison, all hyperparameters are tuned to be the best for each algorithm, as specified in Section VI-A.

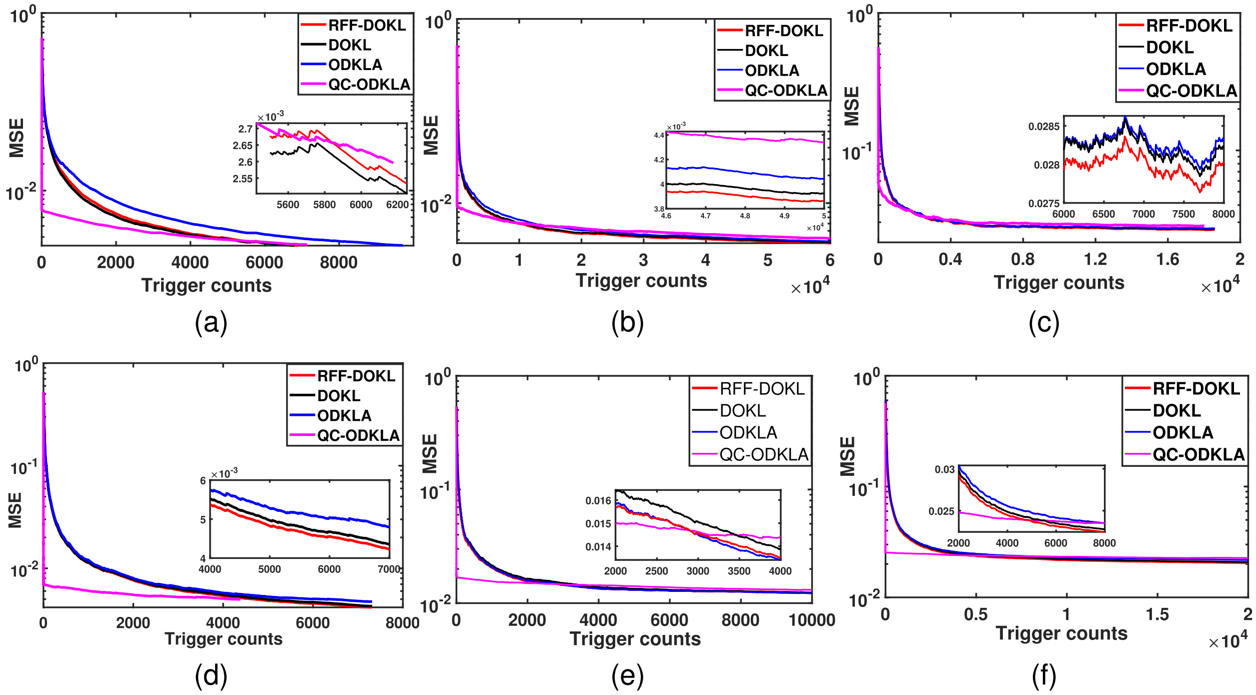
1) MSE Performance: We first evaluate the learning performance of all algorithms by their MSE, which is commonly adopted in online learning problems [18], [20]. From Fig. 1, we can see that the learning performance of ODKLA, RFF-DOKL, and DOKL are very close while the trivial difference comes from the distinction of specific datasets. Notice that in ODKLA, we utilize the linearized ADMM instead of the standard ADMM, and the negligible gap indicates that the learning performance scarification can be ignored. For the proposed QC-ODKLA algorithm that utilizes the quantization and



(a) (b) (c)

(d) (e) (f)

Fig. 1. Comparisons of MSE performance of various methods in online regression tasks. (a) Tom's hardware. (b) Twitter. (c) Energy. (d) Air quality. (e) Conductivity. (f) Blood.



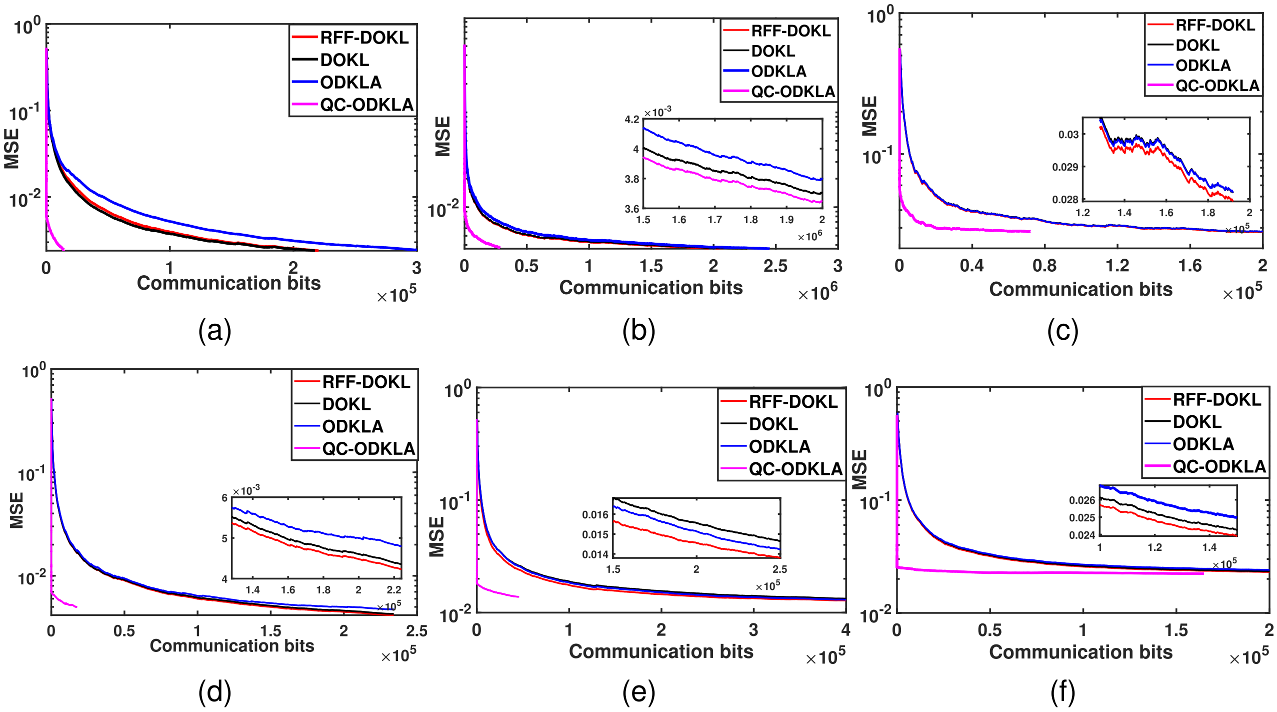
(a) (b) (c)

(d) (e) (f)

Fig. 2. Comparisons of MSE versus triggering numbers of various methods in online regression tasks. (a) Tom's hardware. (b) Twitter. (c) Energy. (d) Air quality. (e) Conductivity. (f) Blood.

communication-censoring strategies to save communications, there are negligible performance degradation on half of the simulated datasets (Twitter, energy, conductivity), while for the remaining datasets, the performance degradation can also be traded with the communication efficiency and computation efficiency of it, which will be shown below.

2) Communication Efficiency: We then evaluate the communication efficiency among different algorithms. We present the MSE performance versus trigger counts in Fig. 2 and MSE performance versus communication bits in Fig. 3. Fig. 2 shows that QC-ODKLA triggers a few transmissions in the early learning stage, which greatly improves the communication



(a) (b) (c)

(d) (e) (f)

Fig. 3. Comparisons of MSE versus communication bits of various methods in online regression tasks. (a) Tom's hardware. (b) Twitter. (c) Energy. (d) Air quality. (e) Conductivity. (f) Blood.

TABLE I

Running Time of Four Algorithms on Six Datasets

Data set	RFF-DOKL	DOKL	ODKLA	QC-ODKLA
Tom's	0.3541s	2.0613s	0.4455s	0.7247s
Twitter	4.6148s	21.8808s	6.2351s	9.4218s
Energy	0.8584s	4.2681s	1.1600s	1.7928s
Air quality	0.2760s	1.6808s	0.4717s	0.5510s
Conductivity	0.7952s	4.4285s	0.9784s	1.6201s
Blood	2.6194s	13.6249s	3.6104s	5.4290s

efficiency. Further, thanks to the quantization, QC-ODKLA only needs 3 bits to transmit an element, the total number of communication bits is also greatly reduced accordingly. For other methods to transmit each element of updates, suppose the agent uses a 32-bit CPU operating mode, then the communication cost is 32 bits per iteration per agent per element. Therefore, QC-ODKLA is corroborated to greatly reduce the communication cost.

3) Computation Efficiency: Finally, we evaluate the computation efficiency of all algorithms by their running time on six datasets, which is recorded in Table I. RFF-DOKL is a gradient descent–based first-order algorithm, which achieves the highest computation efficiency. Comparing ODKLA with the ADMM-based DOKL method, we see that the linearization step reduces a large amount of computation of the standard ADMM. Under the circumstance that online streaming data vary fast, a computation-efficient algorithm is preferred, reflecting the advantages of the proposed ODKLA and QC-ODKLA algorithms. Also, note that QC-ODKLA is computationally slower than ODKLA since the communication-censoring and quantization steps consume computation resources.

In summary, we show that the proposed QC-ODKLA algorithm achieves a good balance in learning performance, communication efficiency, and computation efficiency compared with the state of the art online decentralized kernel-based algorithms.

VII. Conclusion

This article studies the online decentralized kernel learning problem under communication constraints for multi-agent systems. We utilize RF mapping to circumvent the curse of dimensionality issue caused by the increasing size of sequentially arriving data. To efficiently solve such a challenging problem, we then develop a novel online decentralized kernel learning algorithm via linearized ADMM. We integrate the communication-censoring and quantization strategies into the proposed ODKLA algorithm (QC-ODKLA) to further save communication overheads. We derive the sublinear regret bound for QC-ODKLA theoretically, and verify their effectiveness in learning performance, communication, and computation efficiencies via simulations on various real datasets. Future work will be devoted to multi-kernel learning and dynamic kernel learning.

Appendix

Proof: Define $\mathbf{\Theta}^\star = [\boldsymbol{\theta}^{\star\top}; \ldots; \boldsymbol{\theta}^{\star\top}] \in \mathbb{R}^{N\times 2L}$, which is the stack of N copies of $\boldsymbol{\theta}^\star$, and $\mathcal{L}_t(\mathbf{\Theta}^\star) := \sum_{i=1}^{N} \mathcal{L}_{i,t}(\boldsymbol{\theta}^\star)$, we rewrite (24) as follows:

$$\begin{aligned}\mathbf{Reg}_T^S &= \sum_{t=1}^{T}\left(\sum_{i=1}^{N}\mathcal{L}_{i,t}(\boldsymbol{\theta}_{i,t}) - \sum_{i=1}^{N}\mathcal{L}_{i,t}(\boldsymbol{\theta}^\star)\right)\\ &= \sum_{t=1}^{T}\big(\mathcal{L}_t(\boldsymbol{\Theta}_t) - \mathcal{L}_t(\boldsymbol{\Theta}^\star)\big). \qquad (30)\end{aligned}$$

To analyze the regret bound of QC-ODKLA, we first represent the matrix form update of QC-ODKLA updates (22)–(23) as follows:

$$\begin{aligned}\boldsymbol{\Theta}_{t+1} &= \boldsymbol{\Theta}_t - (\eta_t\mathbf{I} + 2\rho\boldsymbol{D})^{-1}\Big[\partial\mathcal{L}_t(\boldsymbol{\Theta}_t)\\ &\qquad + \rho(\boldsymbol{D} - \boldsymbol{W})\hat{\boldsymbol{\Theta}}_t + \boldsymbol{\Gamma}_t\Big] \qquad (31)\\ \boldsymbol{\Gamma}_{t+1} &= \boldsymbol{\Gamma}_t + \rho(\boldsymbol{D} - \boldsymbol{W})\hat{\boldsymbol{\Theta}}_{t+1} \qquad (32)\end{aligned}$$

where $\hat{\boldsymbol{\Theta}}_t = [\hat{\boldsymbol{\theta}}_{1,t}^\top; \ldots; \hat{\boldsymbol{\theta}}_{N,t}^\top] \in \mathbb{R}^{N\times 2L}$. Note that the censoring and quantization are implemented after step (31) and before step (32).

The definitions of the introduced error in (25) and the overall introduced error $\boldsymbol{E}_t$ is equivalent to $\boldsymbol{E}_t := \boldsymbol{\Theta}_t - \hat{\boldsymbol{\Theta}}_t$. With the equality $\mathbf{D} - \mathbf{W} = (1/2)\mathbf{S}_-\mathbf{S}_-^\top$, we can obtain the equivalent form of (31) and (32), respectively, as follows:

$$\begin{aligned}\boldsymbol{\Theta}_{t+1} &= \boldsymbol{\Theta}_t - (\eta_t\mathbf{I} + 2\rho\boldsymbol{D})^{-1}\\ &\quad \times\Big[\partial\mathcal{L}_t(\boldsymbol{\Theta}_t) + \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top\boldsymbol{\Theta}_t - \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top\boldsymbol{E}_t + \boldsymbol{\Gamma}_t\Big] \qquad (33)\\ \boldsymbol{\Gamma}_{t+1} &= \boldsymbol{\Gamma}_t + \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top\boldsymbol{\Theta}_{t+1} - \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top\boldsymbol{E}_{t+1}. \qquad (34)\end{aligned}$$

Observe from (34) that $\boldsymbol{\Gamma}_{t+1}$ stays in the column space of $\mathbf{S}_-\mathbf{S}_-^\top$ if $\boldsymbol{\Gamma}_1$ is also initialized therein. Therefore, we introduce variables $\boldsymbol{\beta}_t \in \mathbb{R}^{2r\times 2L}$, which stay in the column space of $\mathbf{S}_-^\top$, and let $\boldsymbol{\Gamma}_t = \mathbf{S}_-\boldsymbol{\beta}_t$ for any $t \geq 1$. Then, (34) is equivalent to

$$\boldsymbol{\beta}_{t+1} = \boldsymbol{\beta}_t + \frac{\rho}{2}\mathbf{S}_-^\top\boldsymbol{\Theta}_{t+1} - \frac{\rho}{2}\mathbf{S}_-^\top\boldsymbol{E}_{t+1}. \qquad (35)$$

Using (34) and $\boldsymbol{\Gamma}_t = \mathbf{S}_-\boldsymbol{\beta}_t$ to eliminate $\boldsymbol{\Gamma}_t$, we rewrite (33) as follows:

$$\begin{aligned}\boldsymbol{\Theta}_{t+1} &= \boldsymbol{\Theta}_t - (\eta_t\mathbf{I} + 2\rho\boldsymbol{D})^{-1}\\ &\quad \times\Big[\partial\mathcal{L}_t(\boldsymbol{\Theta}_t) + \mathbf{S}_-\boldsymbol{\beta}_{t+1} + \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top(\boldsymbol{E}_{t+1} - \boldsymbol{E}_t)\\ &\qquad + \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\Big]. \qquad (36)\end{aligned}$$

The following analysis is based on the equivalent form of the QC-ODKLA algorithm given by (36) and (35). The Karush–Kuhn–Tucker (KKT) conditions of (10) are

$$\begin{aligned}\partial\mathcal{L}_t(\boldsymbol{\Theta}^\star) + \eta_t(\boldsymbol{\Theta}^\star - \boldsymbol{\Theta}_t) + \mathbf{S}_-\boldsymbol{\beta}^\star = \mathbf{0} \qquad (37a)\\ \mathbf{S}_-^\top\boldsymbol{\Theta}^\star = \mathbf{0} \qquad (37b)\\ \frac{1}{2}\mathbf{S}_+^\top\boldsymbol{\Theta}^\star = \boldsymbol{Z}^\star \qquad (37c)\end{aligned}$$

where $(\boldsymbol{\Theta}^\star, \boldsymbol{Z}^\star, \boldsymbol{\beta}^\star)$ is the optimal primal-dual triplet.

Rearrange terms in (36) to place $\partial\mathcal{L}_t(\boldsymbol{\Theta}_t)$ at the left side, we have

$$\begin{aligned}\partial\mathcal{L}_t(\boldsymbol{\Theta}_t) &= \Big(\eta_t\mathbf{I} + 2\rho\boldsymbol{D} - \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top\Big)(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\\ &\quad + \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \mathbf{S}_-\boldsymbol{\beta}_{t+1}\\ &= \Big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\Big)(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\\ &\quad + \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \mathbf{S}_-\boldsymbol{\beta}_{t+1} \qquad (38)\end{aligned}$$

where the second equality utilizes $\mathbf{D} - \mathbf{W} = (1/2)\mathbf{S}_-\mathbf{S}_-^\top$ and $\mathbf{D} + \mathbf{W} = (1/2)\mathbf{S}_+\mathbf{S}_+^\top$ such that $2\mathbf{D} = (1/2)\mathbf{S}_-\mathbf{S}_-^\top + (1/2)\mathbf{S}_+\mathbf{S}_+^\top$. We consider to bound the instantaneous regret $\mathcal{L}_t(\boldsymbol{\Theta}_t) - \mathcal{L}_t(\boldsymbol{\Theta}^\star)$ at time t first. With Assumption 1, it holds

$$\mathcal{L}_t(\boldsymbol{\Theta}_t) - \mathcal{L}_t(\boldsymbol{\Theta}^\star) \leq \langle\partial\mathcal{L}_t(\boldsymbol{\Theta}_t), \boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\rangle. \qquad (39)$$

Substitute the expression of $\partial\mathcal{L}_t(\boldsymbol{\Theta}_t)$ in (38) into (39) yields

$$\begin{aligned}&\mathcal{L}_t(\boldsymbol{\Theta}_t) - \mathcal{L}_t(\boldsymbol{\Theta}^\star)\\ &\quad\leq \Big\langle\Big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\Big)(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\Big\rangle\\ &\qquad + \Big\langle\frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \mathbf{S}_-\boldsymbol{\beta}_{t+1}, \boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\Big\rangle. \qquad (40)\end{aligned}$$

Now we reorganize the two terms on the right-hand side of (40). For the first term, we have

$$\begin{aligned}&\Big\langle\Big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\Big)(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\Big\rangle\\ &\quad\leq \sigma_{\max}\Big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\Big)\langle\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}, \boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\rangle\\ &\quad= \frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\Big(\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\|_F^2 - \|\boldsymbol{\Theta}_{t+1} - \boldsymbol{\Theta}^\star\|_F^2\\ &\qquad + \|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}\|_F^2\Big) \qquad (41)\end{aligned}$$

where $\sigma_{\max}(\eta_t\mathbf{I} + (\rho/2)\mathbf{S}_+\mathbf{S}_+^\top)$ denotes the maximum singular value of $\eta_t\mathbf{I} + (\rho/2)\mathbf{S}_+\mathbf{S}_+^\top$.

For the second term, we have

$$\begin{aligned}&\Big\langle\frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \mathbf{S}_-\boldsymbol{\beta}_{t+1}, \boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\Big\rangle\\ &= \Big\langle\frac{\rho}{2}\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \boldsymbol{\beta}_{t+1}, \mathbf{S}_-^\top\big(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\big)\Big\rangle\\ &\overset{(a)}{=} \Big\langle\frac{\rho}{2}\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \boldsymbol{\beta}_{t+1}, \mathbf{S}_-^\top\boldsymbol{\Theta}_t\Big\rangle\\ &\overset{(b)}{=} \Big\langle\frac{\rho}{2}\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \boldsymbol{\beta}_{t+1}, \frac{2}{\rho}(\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}) + \mathbf{S}_-^\top\boldsymbol{E}_t\Big\rangle\\ &= \Big\langle\boldsymbol{\beta}_{t-1} - 2\boldsymbol{\beta}_t + \frac{\rho}{2}\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \frac{2}{\rho}(\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}) + \mathbf{S}_-^\top\boldsymbol{E}_t\Big\rangle\\ &\overset{(c)}{=} -\frac{2}{\rho}\langle\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}, \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\rangle - \frac{2}{\rho}\langle\boldsymbol{\beta}_t, \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\rangle\\ &\quad + \langle\boldsymbol{\beta}_{t-1} - \boldsymbol{\beta}_t, \mathbf{S}_-^\top\boldsymbol{E}_t\rangle + \langle\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\rangle\\ &\quad + \frac{\rho}{2}\langle\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \mathbf{S}_-^\top\boldsymbol{E}_t\rangle - \langle\boldsymbol{\beta}_t, \mathbf{S}_-^\top\boldsymbol{E}_t\rangle\\ &= -\frac{2}{\rho}\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\|_F^2 - \frac{2}{\rho}\|\boldsymbol{\beta}_t\|_F^2 + \frac{2}{\rho}\langle\boldsymbol{\beta}_t, \boldsymbol{\beta}_{t-1}\rangle - \langle\boldsymbol{\beta}_t, \mathbf{S}_-^\top\boldsymbol{E}_t\rangle\\ &\quad + \langle\boldsymbol{\beta}_{t-1} - \boldsymbol{\beta}_t, \mathbf{S}_-^\top\boldsymbol{E}_t\rangle + \langle\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\rangle\\ &\quad + \frac{\rho}{2}\langle\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \mathbf{S}_-^\top\boldsymbol{E}_t\rangle \qquad (42)\end{aligned}$$

where (a) comes from the KKT condition (37b), (b) and (c) are obtained by utilizing (35).

Next, we will utilize Young's inequality to bound the inner product terms in (42), which are

$$\begin{aligned}
\frac{2}{\rho}\langle \boldsymbol{\beta}_t, \boldsymbol{\beta}_{t-1}\rangle &\le \frac{2}{\rho}\left(\frac{1}{2\eta_1}\|\boldsymbol{\beta}_t\|_F^2 + \frac{\eta_1}{2}\|\boldsymbol{\beta}_{t-1}\|_F^2\right)\\
&= \frac{1}{\rho\eta_1}\|\boldsymbol{\beta}_t\|_F^2 + \frac{\eta_1}{\rho}\|\boldsymbol{\beta}_{t-1}\|_F^2,\\
\langle \boldsymbol{\beta}_{t-1} - \boldsymbol{\beta}_t, \mathbf{S}_-^\top \boldsymbol{E}_t\rangle &\le \frac{1}{2\eta_2}\|\boldsymbol{\beta}_{t-1} - \boldsymbol{\beta}_t\|_F^2\\
&\quad + \frac{\eta_2}{2}\|\mathbf{S}_-^\top \boldsymbol{E}_t\|_F^2\\
-\langle \boldsymbol{\beta}_t, \mathbf{S}_-^\top \boldsymbol{E}_t\rangle &\le \frac{1}{2\eta_3}\|\boldsymbol{\beta}_t\|_F^2 + \frac{\eta_3}{2}\|\mathbf{S}_-^\top \boldsymbol{E}_t\|_F^2\\
\langle \mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\rangle &\le \frac{1}{2\eta_4}\|\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\|_F^2\\
&\quad + \frac{\eta_4}{2}\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\|_F^2\\
\frac{\rho}{2}\langle \mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}), \mathbf{S}_-^\top \boldsymbol{E}_t\rangle &\le \frac{\rho}{4\eta_5}\|\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\|_F^2\\
&\quad + \frac{\rho\eta_5}{4}\|\mathbf{S}_-^\top \boldsymbol{E}_t\|_F^2 \qquad (43)
\end{aligned}$$

where $\eta_1, \eta_2, \eta_3, \eta_4, \eta_5$ are any positive constants.

Substitute (43) into (42) gives

$$\begin{aligned}
&\left\langle \frac{\rho}{2}\mathbf{S}_-\mathbf{S}_-^\top(\boldsymbol{E}_t - \boldsymbol{E}_{t+1}) - \mathbf{S}_-\boldsymbol{\beta}_{t+1}, \boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\right\rangle\\
&\le \left(\frac{1}{2\eta_2} + \frac{\eta_4}{2} - \frac{2}{\rho}\right)\|\boldsymbol{\beta}_t - \boldsymbol{\beta}_{t-1}\|_F^2 + \frac{\eta_1}{\rho}\|\boldsymbol{\beta}_{t-1}\|_F^2\\
&\quad + \left(\frac{1}{\rho\eta_1} + \frac{1}{2\eta_3} - \frac{2}{\rho}\right)\|\boldsymbol{\beta}_t\|_F^2 + +\Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\\
&\quad \times \|\mathbf{S}_-^\top \boldsymbol{E}_t\|_F^2 + \left(\frac{1}{2\eta_4} + \frac{\rho}{4\eta_5}\right)\|\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\|_F^2\\
&= \left(\frac{1}{2\eta_2} + \frac{\eta_4}{2} + \frac{1}{\rho\eta_1} + \frac{1}{2\eta_3} - \frac{4}{\rho}\right)\|\boldsymbol{\beta}_t\|_F^2\\
&\quad + \left(\frac{1}{2\eta_2} + \frac{\eta_4}{2} + \frac{\eta_1}{\rho} - \frac{2}{\rho}\right)\|\boldsymbol{\beta}_{t-1}\|_F^2\\
&\quad + \left(\frac{1}{2\eta_4} + \frac{\rho}{4\eta_5}\right)\|\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\|_F^2\\
&\quad + \Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\|\mathbf{S}_-^\top \boldsymbol{E}_t\|_F^2 + \left(\frac{2}{\rho} - \frac{1}{2\eta_2} - \frac{\eta_4}{2}\right)\\
&\quad \times \big\langle \boldsymbol{\beta}_t, \boldsymbol{\beta}_{t-1}\big\rangle\\
&\le c_1\|\boldsymbol{\beta}_t\|_F^2 + c_2\|\boldsymbol{\beta}_{t-1}\|_F^2 + \left(\frac{1}{2\eta_4} + \frac{\rho}{4\eta_5}\right)\|\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\|_F^2\\
&\quad + \Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\|\mathbf{S}_-^\top \boldsymbol{E}_t\|_F^2 \qquad (44)
\end{aligned}$$

where c_1 and c_2 are defined as follows:

$$\begin{aligned}
c_1 &:= \frac{1}{2\eta_2} - \frac{4}{\rho} + \frac{\eta_4}{2} + \frac{1}{\rho\eta_1} + \frac{1}{2\eta_3} + \frac{2}{\rho\eta_6} - \frac{1}{2\eta_2\eta_6} - \frac{\eta_4}{2\eta_6}\\
c_2 &:= \frac{\eta_1}{\rho} - \frac{2}{\rho} + \frac{1}{2\eta_2} + \frac{\eta_4}{2} + \frac{2\eta_6}{\rho} - \frac{\eta_6}{2\eta_2} - \frac{\eta_4\eta_6}{2}.
\end{aligned}$$

With (44) and (41), we obtain an upper bound for (40), which is

$$\begin{aligned}
&\mathcal{L}_t(\boldsymbol{\Theta}_t) - \mathcal{L}_t\big(\boldsymbol{\Theta}^\star\big)\\
&\le \frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\big(\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\|_F^2 - \|\boldsymbol{\Theta}_{t+1} - \boldsymbol{\Theta}^\star\|_F^2\big)\\
&\quad + \frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}\|_F^2 + c_1\|\boldsymbol{\beta}_t\|_F^2\\
&\quad + c_2\|\boldsymbol{\beta}_{t-1}\|_F^2 + \left(\frac{1}{2\eta_4} + \frac{\rho}{4\eta_5}\right)\|\mathbf{S}_-^\top(\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1})\|_F^2\\
&\quad + \Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\|\mathbf{S}_-^\top \boldsymbol{E}_t\|_F^2\\
&\le \frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\big(\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\|_F^2 - \|\boldsymbol{\Theta}_{t+1} - \boldsymbol{\Theta}^\star\|_F^2\big)\\
&\quad + c_1\|\boldsymbol{\beta}_t\|_F^2 + c_2\|\boldsymbol{\beta}_{t-1}\|_F^2 + \Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\\
&\quad \times \sigma_{\max}^2(\mathbf{S}_-)\|\boldsymbol{E}_t\|_F^2 + \left(\frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\right.\\
&\qquad\qquad\qquad \left. + \frac{\sigma_{\max}^2(\mathbf{S}_-)}{2\eta_4} + \frac{\rho\sigma_{\max}^2(\mathbf{S}_-)}{4\eta_5}\right)\\
&\quad \times \|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}\|_F^2. \qquad (45)
\end{aligned}$$

We then utilize (36) to rewrite $\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}$ as follows:

$$\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1} = (\eta_t\mathbf{I} + 2\rho\boldsymbol{D})^{-1}\Big(\partial\mathcal{L}_t(\boldsymbol{\Theta}_t) + 2\mathbf{S}_-\boldsymbol{\beta}_t - \mathbf{S}_-\boldsymbol{\beta}_{t-1}\Big) \qquad (46)$$

and bound $\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}\|_F^2$ as follows:

$$\begin{aligned}
&\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}_{t+1}\|_F^2\\
&= \Big\|(\eta_t\mathbf{I} + 2\rho\boldsymbol{D})^{-1}\Big(\partial\mathcal{L}_t(\boldsymbol{\Theta}_t) + 2\mathbf{S}_-\boldsymbol{\beta}_t - \mathbf{S}_-\boldsymbol{\beta}_{t-1}\Big)\Big\|_F^2\\
&\le \frac{1}{\sigma_{\min}^2(\eta_t\mathbf{I} + 2\rho\boldsymbol{D})}\|\partial\mathcal{L}_t(\boldsymbol{\Theta}_t)\|_F^2 + \frac{4\sigma_{\max}^2(\mathbf{S}_-)}{\sigma_{\min}^2(\eta_t\mathbf{I} + 2\rho\boldsymbol{D})}\|\boldsymbol{\beta}_t\|_F^2\\
&\quad + \frac{\sigma_{\max}^2(\mathbf{S}_-)}{\sigma_{\min}^2(\eta_t\mathbf{I} + 2\rho\boldsymbol{D})}\|\boldsymbol{\beta}_{t-1}\|_F^2 \qquad (47)
\end{aligned}$$

where $\sigma_{\min}(\eta_t\mathbf{I} + 2\rho\boldsymbol{D})$ is the lower bound of the nonzero singular values of $\eta_t\mathbf{I} + 2\rho\boldsymbol{D}$.

Substitute (47) into (45) we obtain

$$\begin{aligned}
&\mathcal{L}_t(\boldsymbol{\Theta}_t) - \mathcal{L}_t(\boldsymbol{\Theta}^\star)\\
&\le \frac{\sigma_{\max}(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top)}{2}\Big(\|\boldsymbol{\Theta}_t - \boldsymbol{\Theta}^\star\|_F^2 - \|\boldsymbol{\Theta}_{t+1} - \boldsymbol{\Theta}^\star\|_F^2\Big)\\
&\quad + (c_1 + 4\,c_{\mathcal{N}})\|\boldsymbol{\beta}_t\|_F^2 + (c_2 + c_{\mathcal{N}})\|\boldsymbol{\beta}_{t-1}\|_F^2 + \frac{c_{\mathcal{N}}}{\sigma_{\max}^2(\mathbf{S}_-)}\\
&\quad \times \|\partial\mathcal{L}_t(\boldsymbol{\Theta}_t)\|_F^2 + \Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\sigma_{\max}^2(\mathbf{S}_-)\|\boldsymbol{E}_t\|_F^2 \qquad (48)
\end{aligned}$$

where $c_{\mathcal{N}}$ is defined as follows:

$$\begin{aligned}
c_{\mathcal{N}} := &\left(\frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2} + \frac{\sigma_{\max}^2(\mathbf{S}_-)}{2\eta_4} + \frac{\rho\sigma_{\max}^2(\mathbf{S}_-)}{4\eta_5}\right)\\
&\qquad \frac{\sigma_{\max}^2(\mathbf{S}_-)}{\sigma_{\min}^2(\eta_t\mathbf{I} + 2\rho\boldsymbol{D})}.
\end{aligned}$$

Carefully choose $\eta_1, \eta_2, \eta_3, \eta_4, \eta_5$, and η_6, we can make $c_1 + 4c_{\mathcal{N}} = -(c_2 + c_{\mathcal{N}}) = -c$, where $c > 0$. One example is to set

$\eta_4 = (2\eta_6/((\eta_6-1)^2))((1/\rho)(\eta_1 + (1/\eta_1) + (2/\eta_6) + 2\eta_6) + (1/2\eta_2)(2-\eta_6-(1/\eta_6)) + (1/2\eta_3) + 5c_{\mathcal{N}})$. Then (48) can be further simplified as follows:

$$\begin{aligned}
&\mathcal{L}_t(\mathbf{\Theta}_t) - \mathcal{L}_t(\mathbf{\Theta}^\star)\\
&\le \frac{\sigma_{\max}(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top)}{2}\Big(\|\mathbf{\Theta}_t - \mathbf{\Theta}^\star\|_F^2 - \|\mathbf{\Theta}_{t+1} - \mathbf{\Theta}^\star\|_F^2\Big)\\
&\quad + c(\|\boldsymbol{\beta}_{t-1}\|_F^2 - \|\boldsymbol{\beta}_t\|_F^2) + \frac{c_{\mathcal{N}}}{\sigma_{\max}^2(\mathbf{S}_-)}\|\partial\mathcal{L}_t(\mathbf{\Theta}_t)\|_F^2\\
&\quad + \Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\sigma_{\max}^2(\mathbf{S}_-)\|\boldsymbol{E}_t\|_F^2. \qquad (49)
\end{aligned}$$

Summarizing both sides of (49) from $t = 1$ to $t = T$ leads to the accumulated network regret $\mathcal{R}(T)$

$$\begin{aligned}
&\mathcal{R}(T)\\
&\le \frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\big(\|\mathbf{\Theta}_1 - \mathbf{\Theta}^\star\|_F^2 - \|\mathbf{\Theta}_{T+1} - \mathbf{\Theta}^\star\|_F^2\big)\\
&\quad + c\big(\|\boldsymbol{\beta}_0\|_F^2 - \|\boldsymbol{\beta}_T\|_F^2\big) + \sum_{t=1}^{T}\frac{c_{\mathcal{N}}}{\sigma_{\max}^2(\mathbf{S}_-)}\|\partial\mathcal{L}_t(\mathbf{\Theta}_t)\|_F^2\\
&\quad + \sum_{t=1}^{T}\Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\sigma_{\max}^2(\mathbf{S}_-)\|\boldsymbol{E}_t\|_F^2\\
&\le \frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\|\mathbf{\Theta}_1 - \mathbf{\Theta}^\star\|_F^2 + \sum_{t=1}^{T}\frac{c_{\mathcal{N}}}{\sigma_{\max}^2(\mathbf{S}_-)}\\
&\quad \times \|\partial\mathcal{L}_t(\mathbf{\Theta}_t)\|_F^2 + c\|\boldsymbol{\beta}_0\|_F^2 + \sum_{t=1}^{T}\Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\\
&\quad \times \sigma_{\max}^2(\mathbf{S}_-)\|\boldsymbol{E}_t\|_F^2\\
&= \frac{\sigma_{\max}\big(\eta_t\mathbf{I} + \frac{\rho}{2}\mathbf{S}_+\mathbf{S}_+^\top\big)}{2}\|\mathbf{\Theta}^\star\|_F^2 + \sum_{t=1}^{T}\frac{c_{\mathcal{N}}}{\sigma_{\max}^2(\mathbf{S}_-)}\|\partial\mathcal{L}_t(\mathbf{\Theta}_t)\|_F^2\\
&\quad + \sum_{t=1}^{T}\Big(\frac{\eta_2}{2} + \frac{\eta_3}{2} + \frac{\rho\eta_5}{4}\Big)\sigma_{\max}^2(\mathbf{S}_-)\|\boldsymbol{E}_t\|_F^2. \qquad (50)
\end{aligned}$$

The last equality comes from the initialization that $\mathbf{\Theta}_1 = \mathbf{0}$ and $\boldsymbol{\beta}_1 = \mathbf{0}$ and thus $\boldsymbol{\beta}_0 = \mathbf{0}$. Assumption 2 assumes $\|\boldsymbol{\theta}^\star\|_F \le C_\theta$, which implies $\|\mathbf{\Theta}^\star\|_F \le \sqrt{N}C_\theta$. Assumption 1 assumes $\|\partial\mathcal{L}_t(\mathbf{\Theta}_t)\|_F \le C_{\mathcal{L}}$. Then, setting $\rho = \eta_t = \eta_2 = \eta_3 = 1/\mathcal{O}(\sqrt{T})$, the sublinear regret is achieved

$$\mathcal{R}(T) \le (\sqrt{N}C_\theta + \frac{1}{\sigma_{\max}^2(\mathbf{S}_-)}C_{\mathcal{L}} + \sigma_{\max}^2(\mathbf{S}_-)\zeta)\mathcal{O}(\sqrt{T}) \qquad (51)$$

where $\zeta := \max\{\sqrt{N}\alpha\beta, \sqrt{2NL}\Delta/2\}$ with α and β being the predefined censoring threshold parameters and Δ being the length of the quantization interval.

References

[1] J. Liang, Z. Wang, B. Shen, and X. Liu, "Distributed state estimation in sensor networks with randomly occurring nonlinearities subject to time delays," *ACM Trans. Sensor Netw.*, vol. 9, no. 1, pp. 1–18, Nov. 2012.

[2] W. Ren, R. W. Beard, and E. M. Atkins, "Information consensus in multivehicle cooperative control," *IEEE Control Syst. Mag.*, vol. 27, no. 2, pp. 71–82, Apr. 2007.

[3] D. Mateos-Núñez and J. Cortés, "Distributed online convex optimization over jointly connected digraphs," *IEEE Trans. Netw. Sci. Eng.*, vol. 1, no. 1, pp. 23–37, Jan. 2014.

[4] H.-F. Xu, Q. Ling, and A. Ribeiro, "Online learning over a decentralized network through ADMM," *J. Oper. Res. Soc. China*, vol. 3, no. 4, pp. 537–562, Dec. 2015.

[5] Y. Zhao, C. Yu, P. Zhao, H. Tang, S. Qiu, and J. Liu, "Decentralized online learning: Take benefits from others' data without sharing your own to track global trend," 2019, *arXiv:1901.10593*.

[6] P. Sharma, P. Khanduri, L. Shen, D. J. Bucci Jr., and P. K. Varshney, "On distributed online convex optimization with sublinear dynamic regret and fit," 2020, *arXiv:2001.03166*.

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

[8] E. Hazan, A. Agarwal, and S. Kale, "Logarithmic regret algorithms for online convex optimization," *Mach. Learn.*, vol. 69, nos. 2–3, pp. 169–192, Oct. 2007.

[9] P. Xu, Y. Wang, X. Chen, and Z. Tian, "COKE: Communication-censored decentralized kernel learning," *J. Mach. Learn. Res.*, vol. 22, no. 196, pp. 1–35, 2021.

[10] T. Le, V. Nguyen, T. D. Nguyen, and D. Phung, "Nonparametric budgeted stochastic gradient descent," in *Artificial Intelligence and Statistics*. Cadiz, Spain: PMLR, 2016, pp. 572–654.

[11] A. Koppel, G. Warnell, E. Stump, and A. Ribeiro, "Parsimonious online learning with kernels via sparse projections in function space," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2017, pp. 4671–4675.

[12] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1177–1184.

[13] B. Dai et al., "Scalable kernel methods via doubly stochastic gradients," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3041–3049.

[14] T. D. Nguyen, T. Le, H. Bui, and D. Phung, "Large-scale online kernel learning with random feature reparameterization," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 2543–2549.

[15] C. Williams and M. Seeger, "Using the Nyström method to speed up kernel machines," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 13, 2000, pp. 1–13.

[16] A. Gittens and M. W. Mahoney, "Revisiting the Nyström method for improved large-scale machine learning," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 3977–4041, 2016.

[17] D. Richards, P. Rebeschini, and L. Rosasco, "Decentralised learning with random features and distributed gradient descent," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 8105–8115.

[18] P. Bouboulis, S. Chouvardas, and S. Theodoridis, "Online distributed learning over networks in RKH spaces using random Fourier features," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1920–1932, Apr. 2018.

[19] Y. Shen, S. Karimi-Bidhendi, and H. Jafarkhani, "Distributed and quantized online multi-kernel learning," *IEEE Trans. Signal Process.*, vol. 69, pp. 5496–5511, 2021.

[20] S. Hong and J. Chae, "Distributed online learning with multiple kernels," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 3, pp. 1263–1277, Mar. 2023.

[21] G. Qu and N. Li, "Accelerated distributed Nesterov gradient descent," *IEEE Trans. Autom. Control*, vol. 65, no. 6, pp. 2566–2581, Jun. 2020.

[22] S. Zhu, M. Hong, and B. Chen, "Quantized consensus ADMM for multi-agent distributed optimization," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 4134–4138.

[23] M. Zhang, L. Chen, A. Mokhtari, H. Hassani, and A. Karbasi, "Quantized Frank–Wolfe: Faster optimization, lower communication, and projection free," 2019, *arXiv:1902.06332*.

[24] S. U Stich, J.-B. Cordonnier, and M. Jaggi, "Sparsified SGD with memory," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 4447–4458.

[25] I. E. K. Harrane, R. Flamary, and C. Richard, "On reducing the communication cost of the diffusion LMS algorithm," *IEEE Trans. Signal Inf. Process. over Netw.*, vol. 5, no. 1, pp. 100–112, Mar. 2019.

[26] M. Li, D. G. Andersen, A. J. Smola, and K. Yu, "Communication efficient distributed machine learning with the parameter server," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 19–27.

[27] Y. Yu, J. Wu, and L. Huang, "Double quantization for communication-efficient distributed optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4440–4451.

[28] T. Chen, G. Giannakis, T. Sun, and W. Yin, "LAG: Lazily aggregated gradient for communication-efficient distributed learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 5050–5060.

[29] 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, May 2019.

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

[31] F. R. K. Chung and F. C. Graham, *Spectral Graph Theory*. Providence, RI, USA: Amer. Math. Soc., 1997.

[32] S. Bochner, *Harmonic Analysis and the Theory of Probability*. North Chelmsford, MA, USA: Courier Corp., 2005.

[33] Y. Liu, G. Wu, Z. Tian, and Q. Ling, “DQC-ADMM: Decentralized dynamic ADMM with quantized and censored communications,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3290–3304, Aug. 2022.

[34] M. Kelly, R. Longjohn, and K. Nottingham. *The UCI Machine Learning Repository*. [Online]. Available: https://archive.ics.uci.edu

[35] F. Kawala, A. Douzal, E. Gaussier, and E. Diemert, “Buzz in social media,” *UCI Mach. Learn. Repository*, 2013. [Online]. Available: https://doi.org/10.24432/C56G6V

[36] L. M. Candanedo, V. Feldheim, and D. Deramaix, “Data driven prediction models of energy use of appliances in a low-energy house,” *Energy Buildings*, vol. 140, pp. 81–97, Apr. 2017.

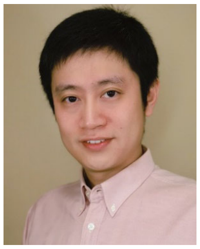
[37] S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia, “On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario,” *Sens. Actuators B, Chem.*, vol. 129, no. 2, pp. 750–757, Feb. 2008.

[38] K. Hamidieh, “A data-driven statistical model for predicting the critical temperature of a superconductor,” *Comput. Mater. Sci.*, vol. 154, pp. 346–354, Nov. 2018.

[39] M. Kachuee, M. M. Kiani, H. Mohammadzade, and M. Shabany, “Cuffless high-accuracy calibration-free blood pressure estimation using pulse transit time,” in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 1006–1009.

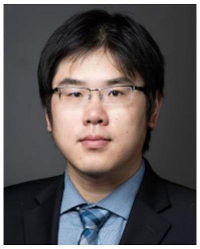
Ping Xu (Member, IEEE) received the B.E. degree in electronics and information engineering from Northwestern Polytechnical University, Xi’an, China, in 2015, and the master’s and Ph.D. degrees in electrical engineering from George Mason University, Fairfax, VA, USA, in 2018 and 2022, respectively, under the guidance of Dr. Z. Tian.

Since August 2023, she has been with the Department of Electrical and Computer Engineering, University of Texas Rio Grande Valley, Edinburg, TX, USA. Her research interests include decentralized learning and optimization, signal processing, dynamical systems, and cooperative control.



Yue Wang (Senior Member, IEEE) received the Ph.D. degree in communication and information system from the School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China, in 2011.

He was a Research Assistant Professor with the Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA, USA. He is currently an Assistant Professor with the Department of Computer Science, Georgia State University, Atlanta, GA, USA. His general interests include machine learning, signal processing, wireless communications, and their applications in cyber-physical systems. His specific research focuses on compressive sensing, massive multiple input multiple output (MIMO), millimeter-wave communications, wideband spectrum sensing, cognitive radios, direction of arrival (DoA) estimation, high-dimensional data analysis, and distributed optimization and learning.



Xiang Chen (Member, IEEE) received the Ph.D. degree in computer engineering from the University of Pittsburgh, Pittsburgh, PA, USA, in 2016, under the guidance of Dr. Y. Chen.

After that, he joined George Mason University, Fairfax, VA, USA, and founded the Intelligence Fusion Laboratory. His research works focus on high-performance computing, artificial intelligence, large-scale systems, and various mobile and edge applications.

Dr. Chen also received the NSF CAREER Award, the Best Paper Award in DATE, and the several other competition awards. With close collaboration with industrial labs and vast universities, he is currently leading multiple research projects funded by NSF and Air Force Research Lab (AFRL).

Zhi Tian (Fellow, IEEE) was on the Faculty of Michigan Technological University, Houghton, MI, USA, from 2000 to 2014. She has been a Professor with the Electrical and Computer Engineering Department of George Mason University, Fairfax, VA, USA, since 2015. She served as the Program Director with the U.S. National Science Foundation from 2012 to 2014. Her research interests include the areas of statistical signal processing, wireless communications, machine learning, and estimation and detection theory. Her research focuses on compressed sensing for random processes, statistical inference of network data, distributed network optimization and learning, and millimeter-wave communications.

Dr. Tian was a Member-at-Large of the Board of Governors of the IEEE Signal Processing Society for the term of 2019–2021. She was an IEEE Distinguished Lecturer for both the IEEE Communications Society and the IEEE Vehicular Technology Society. She received the IEEE Communications Society TCCN Publication Award in 2018. She served as an Associate Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS and IEEE TRANSACTIONS ON SIGNAL PROCESSING.