Received 30 October 2022; revised 20 November 2022; accepted 4 December 2022. Date of publication 13 December 2022; date of current version 23 December 2022.

Digital Object Identifier 10.1109/OJCOMS.2022.3228813

A Low-Complexity and Adaptive Distributed Source Coding Design for Model Aggregation in Distributed Learning

NAIFU ZHANG AND MEIXIA TAO (Fellow, IEEE)

Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

CORRESPONDING AUTHOR: M. TAO (e-mail: mxtao@sjtu.edu.cn)

This work was supported in part by the National Key Research and Development Project of China under Grant 2019YFB1802702, and in part by the NSF of China under Grant 62125108. Part of this work is accepted by IEEE WCSP 2021 [1].

ABSTRACT A major bottleneck in distributed learning is the communication overhead of exchanging intermediate model update parameters between the worker nodes and the parameter server. Recently, it is found that local gradients among different worker nodes are correlated. Therefore, distributed source coding (DSC) can be applied to increase communication efficiency by exploiting such correlation. However, it is highly non-trivial to exploite the gradient correlations in distributed learning due to the unknown and time-varying gradient correlation. In this paper, we first propose a DSC framework, named successive Wyner-Ziv coding, for distributed learning based on quantization and Slepian-Wolf (SW) coding. We prove that the proposed framework can achieve the theoretically minimum communication cost from an information theory perspective. We also propose a low-complexity and adaptive DSC for distributed learning, including a gradient statistics estimator, rate controller, and a log-likelihood ratio (LLR) computer. The gradient statistics estimator estimates the gradient statistics online based only on the quantized gradients at previous iterations, hence it does not introduce extra communication cost. The computation complexity of the rate controller and the LLR computer is reduced to a linear growth in the number of worker nodes by introducing a semi-analytical Monte Carlo simulation. Finally, we design a DSC-based distributed learning process and find that the extra delay introduced by DSC does not scale with the number of worker nodes.

INDEX TERMS Distributed learning, model aggregation, distributed source coding, Slepian-Wolf coding.

I. INTRODUCTION

THE PROLIFERATION of mobile devices such as smartphones, tablets, and wearable devices has revolutionized people's daily lives. Due to the growing computation and sensing capabilities of these devices, a wealth of data has been generated each day, which can be used to train high-accurate machine learning models. It is becoming crucial to train big models in a distributed fashion in which large-scale datasets are distributed over multiple worker machines for parallel processing [2], [3]. Compared with traditional learning at a centralized data center, distributed learning offers several distinct advantages, such as preserving privacy, reducing network congestion, and leveraging distributed on-device computation.

Distributed learning generally requires the participating worker nodes to exchange intermediate model update parameters with the parameter server for global model aggregation repeatedly. With the fast-growing on-device computation capability, the communication overhead between the worker nodes and the parameter server has gradually become the performance bottleneck [2]. This is exacerbated in the cases of the federated learning paradigm [4], [5], and the cloud-edge AI systems [6]. In these computation paradigms, generally, the time for communication can be many orders of magnitude longer than the time for local computations [7]. It is thus essential to design communication-efficient distributed learning methods to reduce the communication cost during model aggregation. Gradient compression, such as quantization and sparsification, is an efficient approach to reduce the communication cost at each round of model update. Partially initiated by the 1-bit implementation of stochastic gradient descent (SGD) by Microsoft in [8], a large number of recent studies have revisited the idea of gradient quantization [9], [10], [11]. Other approaches for low-precision training focus on the sparsification of gradients, either by thresholding small entries or by random sampling [12], [13]. There also exist several approaches that combine quantization and sparsification to maximize performance gains, including quantized SGD (QSGD) [14] and TernGrad [15].

The above model compression schemes treat each model update from different worker nodes independently. In practice, the local updates are correlated since the model to be trained is the same for all worker nodes. Recently, it is found in [16], [17] that the model aggregation of distributed learning is inherently an indirect multi-terminal source coding problem, or the so-called CEO problem [18], where each worker node cannot observe directly the model update that is to be reconstructed at the parameter server, but is rather provided only with a noisy version. The work [19] also finds that the correlation of the local gradients produced by different worker nodes is strong using information-theoretic measures. Therefore, distributed source coding (DSC) can be applied for model aggregation to further increase the communication efficiency by exploiting such correlation.

Nevertheless, exploiting the gradient correlations in distributed learning is a non-trivial task. The most straightforward way is to apply the practical designs of DSC for the quadratic Gaussian CEO problem to distributed learning. The work [20] proposes an asymmetric DSC framework that essentially relies on quantization and Wyner-Ziv coding and can approach any point of the achievable rate region via source splitting. The practical DSC design [20] requires knowledge of the gradient statistics, which, however, are generally unknown and time-varying in the training process [21]. Even if the gradient statistics are known in the existing asymmetric Slepian-Wolf (SW) coding design [20], it still requires to redesign the encoders and decoder at each iteration due to the time-varying correlation. However, the complexity of computing the transmission rate at each iteration increases exponentially with the number of the worker nodes, and thus is unacceptable in a large-scale network.

Recently there have been some attempts to exploit the correlation of the local gradients [16], [19], [22]. The work [22] explores local memory similarity across worker nodes and designs a commutative compressor which works as follows: At each iteration, a leading worker node is selected, and all other worker nodes follow the leading worker's top-k index selection to sparsify their own local gradients. However, there is still redundant information that is not exploited between the sparsified local gradients. To this end, the work [19] exploits an autoencoder to capture the common information that exists in the local gradients. The autoencoder is trained using the local gradients collected from all the worker nodes at the initial iterations, which, however, introduces extra communication overhead since the local gradients for the training cannot be compressed. Moreover, the static encoders and decoders may fail to track the time-varying gradient correlation. In [16], the worker nodes are divided into two groups. The worker nodes in the first group transmit their local gradients without exploiting the correlation and the gradient statistics are estimated at the parameter server based on these local gradients. The worker nodes in the second group use nested scalar quantization and Slepian-Wolf (SW) coding to compress their local gradients. In this scheme, extra communication overhead is still needed since the local gradients of workers nodes in the first group cannot be compressed during correlation estimation. Besides, similar to asymmetric SW coding [20], the complexity of computing the transmission rate at each iteration increases exponentially with the number of worker nodes. Note that all these works [16], [19], [22] cannot theoretically guarantee that their proposed methods can fully exploit the correlation and achieve a theoretically minimum communication cost.

Motivated by the above issue, in this paper, we study the gradient compression scheme for distributed learning by fully exploiting the correlation of local gradients. The goal is to propose a DSC scheme to achieve the theoretically minimum communication cost, which is characterized by a sum-rate-distortion function. The low-complexity DSC scheme is required to track the time-varying correlation of local gradients and does not introduce extra communication cost. The main contributions of this paper are outlined below:

- *DSC design under known and static gradient statistics:* We propose a successive Wyner-Ziv coding framework for distributed learning based on quantization and SW coding. By applying ideal quantization and ideal SW coding, we first prove that the proposed framework can achieve the sum-rate-distortion function. To the best of our knowledge, this is the first work achieving the information-theoretic communication cost of model aggregation by exploiting the correlation. We then provide a multilevel syndrome-based SW coding implemented by low density parity check (LDPC) codes when gradient statistics are known. The proposed practical SW coding design is flexible and compatible with existing gradient quantization methods and can further reduce the communication overhead without loss of model accuracy.
- *Low-complexity and adaptive DSC for distributed learning:* We design three helper blocks, i.e., gradient statistics estimator, rate controller, and log-likelihood ratio (LLR) computer, to provide necessary information for the SW encoder and SW decoder at each iteration, so that practical SW coding can be applied to distributed learning when the gradient statistics are unknown and time-varying. The gradient statistics estimator estimates the gradient statistics online only based on the quantized gradients at previous iterations, hence it does not introduce extra communication cost. The rate

controller and LLR computer calculate the rate and LLR, respectively, of each bit-plane efficiently based on the estimated gradient statistics. The computation complexity is reduced to a linear growth in the number of worker nodes by introducing a semi-analytical Monte Carlo simulation. Finally, we design a DSC-based distributed learning process and find that the extra delay introduced by DSC does not scale with the number of worker nodes.

- *Experiment validation:* The performance of the three helper blocks and the communication cost of the proposed DSC are evaluated based on real-world datasets MNIST, CIFAR-10, and SVHN. The gradient statistics change slowly with iteration t and the proposed online gradient statistics estimation is effective. Compared with the traditional Monte Carlo simulation, the rate controller equipped with semi-analytical Monte Carlo simulation is more unbiased and achieves higher precision. It is also observed that the proposed DSC significantly reduces the communication cost without any loss of model accuracy in real-world datasets.

The rest of this paper is organized as follows. Section II provides the system model and problem formulation. Section III proposes DSC under known and static gradients statistics. Section IV introduces a low-complexity and adaptive DSC for distributed learning. Section V provides experiment results. Finally, we conclude the paper in Section VI.

Notations: An n-length sequence generated by random variable X is denoted by X^n, whose i-th element is denoted by $X[i]$, i.e., $X^n = (X[1], X[2], \dots, X[n])$.

II. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we first formulate the model aggregation in distributed learning as a quadratic Gaussian CEO problem under unbiased estimation constraint. Then, we review the minimum communication cost by exploiting the correlation between local gradients from an information theory perspective.

A. MODEL AGGREGATION IN DISTRIBUTED LEARNING

We consider a basic distributed learning framework, where a shared AI model (e.g., a classifier) is trained collaboratively across K worker nodes via the coordination of a parameter server.

Let $\mathcal{K} = \{1, \dots, K\}$ denote the set of worker nodes. Each worker node $k \in \mathcal{K}$ collects a fraction of training data, denoted as $\mathcal{S}_k$. It is assumed that the local datasets are independent and identical distributed (IID) for different $k's$. Let $\boldsymbol{w} \in \mathbb{R}^P$ denote the P-dimensional model parameter to be learned. The loss function measuring the model error is defined as

$$F(\boldsymbol{w}) = \sum_{k\in\mathcal{K}} \frac{|\mathcal{S}_k|}{|\mathcal{S}|} F_k(\boldsymbol{w}), \tag{1}$$

where $F_k(\boldsymbol{w}) = \frac{1}{|\mathcal{S}_k|}\sum_{i\in\mathcal{S}_k} f_i(\boldsymbol{w})$ is the loss function of worker node k quantifying the prediction error of the model $\boldsymbol{w}$ on the local dataset $\mathcal{S}_k$ collected at the k-th worker node, with $f_i(\boldsymbol{w})$ being the sample-wise loss function, and $\mathcal{S} = \bigcup_{k\in\mathcal{K}} \mathcal{S}_k$ is the union of all datasets. The objective of distributed learning is to minimize the loss function $F(\boldsymbol{w})$.

The minimization of $F(\boldsymbol{w})$ is typically carried out through the minibatched stochastic gradient descent (minibatched SGD) algorithm at each worker node. Specifically, each worker node k splits its local dataset $\mathcal{S}_k$ into mini-batches of size B, and draws one mini-batch $\mathcal{B}_k(t)$ randomly at each iteration t. It then calculates the local gradient as

$$\boldsymbol{g}_k(t) = \nabla \frac{1}{B} \sum_{i\in\mathcal{B}_k(t)} f_i(\boldsymbol{w}(t)). \tag{2}$$

At each iteration t, the parameter server is interested in the global gradient $\boldsymbol{g}(t)$, which is defined as $\boldsymbol{g}(t) \triangleq \nabla F(\boldsymbol{w}(t))$ The parameter server estimates the global gradient $\boldsymbol{g}(t)$ by aggregating $\hat{\boldsymbol{g}}(t) = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{g}_k(t)$, and then updates the model $\boldsymbol{w}(t)$ as

$$\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \eta(t)\hat{\boldsymbol{g}}(t), \tag{3}$$

with $\eta(t)$ being the learning rate at iteration t. This training process is repeated until the model converges. Since we focus on the design of communication-efficient model aggregation in the training process, the training iteration index t is omitted in the rest of this paper.

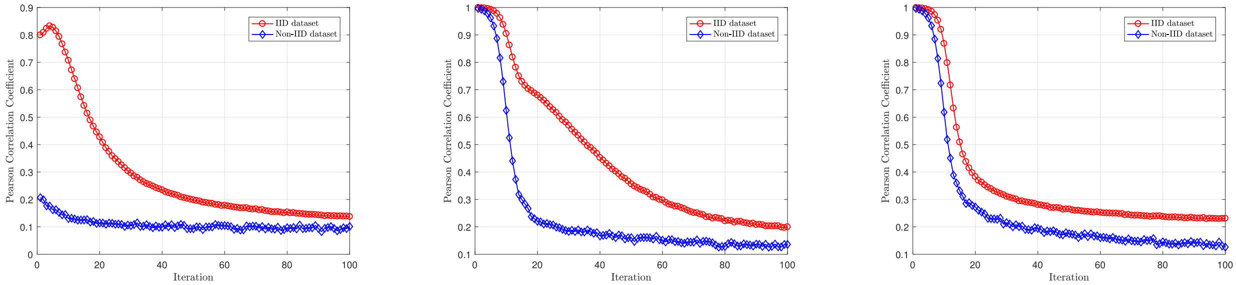
Model aggregation in distributed learning is essentially to estimate the global gradient $\boldsymbol{g}$ at the parameter server based on the local gradients $\boldsymbol{g}_k$ computed at worker nodes. As shown in [17], by minibatched SGD, this problem can be modeled as a Gaussian CEO problem in distributed source coding. Specifically, the P-dimensional global gradient $\boldsymbol{g}$ can be viewed as P realization of source X, which is a sequence of P IID real-valued Gaussian random variables of mean zero and variance σ_X^2.[1] Similarly, the P-dimensional local gradient $\boldsymbol{g}_k$ observed by each worker node k can be viewed as P realization of source Y_k, which is a noisy version of the global gradient X, i.e.,

$$Y_k[i] = X[i] + N_k[i], \forall i \in \{1, 2, \dots, P\}, \tag{4}$$

where the gradient noise N_k^P is a sequence of IID real-valued Gaussian random variables of mean zero and variance σ_N^2. The task of model aggregation is then to reconstruct X^P at the parameter server based on the noisy observations Y_k^P's at all the K worker nodes.[2] We call σ_X^2 the global gradient variance and σ_N^2 the gradient noise variance, respectively.

[1] As stated in [15], the gradient statistics in different layers can be different as gradients are back propagated. In experiment, we apply layer-wised DSC for the model aggregation, where the elements within one layer can be assumed IID. Nevertheless, for presentation convenience, we assume that the entire gradients are sequences of IID variables.

[2] Note that the assumption of gradient distribution follows principle of maximum entropy. The sufficiency of our proposed coding design will not be damaged even if these assumptions do not hold. Experimental validation of the Gaussian assumption of gradients can be found in [15], [17], [23].



(a) MLP model on MNIST.

(b) Resnet20 model on SVHN.

(c) Resnet56 model on CIFAR-10.

FIGURE 1. Experimental results of Pearson correlation coefficient of two local gradients over iterations for three datasets where the number of worker nodes is 10 and the local mini-batch size is 20.

The local gradient Y_k^P observed by each worker node k is separately encoded to $S_k = \phi_k(Y_k^P)$ and sent to the parameter server. The encoder function ϕ_k is defined by $\phi_k : \mathbb{R}^P \to \{1, 2, \ldots, 2^{PR_k}\}$, where R_k is the communication rate of worker node k. The parameter server observes S_k, for each $k = 1, 2, \ldots, K$, and outputs an unbiased estimation $\hat{X}^P$ of X^P by using decoder function ψ_K, i.e.,

$$\hat{X}^P = \psi_K(S_1, S_2, \ldots, S_K). \tag{5}$$

The estimation performance of $\hat{X}^P$ can be measured by MSE distortion given as

$$D\left(X^P, \hat{X}^P\right) = \frac{1}{P}\sum_{i=1}^{P} \mathbb{E}\left[\left(X[i] - \hat{X}[i]\right)^2\right]. \tag{6}$$

As shown in [24, Th. 6.3], the unbiased gradient estimator ensures the convergence of the model and the convergence rate of model training in machine learning depends on the mean square error (MSE) of the estimated gradient. Our aim is to design encoders $\{\phi_k\}_{k=1}^K$ and decoder ψ_K such that the total communication cost, i.e., $R_{sum}(D) = \sum_{k=1}^K R_k$, is minimized for a target gradient distortion $D(X^P, \hat{X}^P) \leq D$.

B. GRADIENT CORRELATION

We use Pearson correlation coefficients to measure the correlation between the local gradients. Let $\rho_{Y_k Y_j}$ denote Pearson correlation coefficients of two local gradients Y_k and Y_j, for $k \neq j$, which is given by

$$\rho_{Y_k Y_j} = \frac{\mathbb{E}\left[Y_k Y_j\right]}{\sqrt{\mathbb{E}\left[Y_k^2\right]\mathbb{E}\left[Y_j^2\right]}} \tag{7}$$

$$= \frac{\mathbb{E}\left[(X + N_k)\left(X + N_j\right)\right]}{\sqrt{\mathbb{E}\left[(X + N_k)^2\right]\mathbb{E}\left[\left(X + N_j\right)^2\right]}} \tag{8}$$

$$= \frac{\sigma_X^2}{\sigma_X^2 + \sigma_N^2}. \tag{9}$$

Note that the correlation between the local gradients fully depends on σ_X^2/σ_N^2. Specifically, the Pearson correlation coefficient approaches 0 when $\sigma_X^2/\sigma_N^2 \to 0$ and the Pearson correlation coefficient approaches 1 when $\sigma_X^2/\sigma_N^2 \to \infty$.

Fig. 1 illustrates the experimental results of Pearson correlation coefficient of two local gradients over iterations for three datasets, MNIST, SVHN, and CIFAR-10, where the number of worker nodes is 10 and the local mini-batch size is 20. These three models approach convergence at the end of the last iteration. Each value of Pearson correlation coefficient is obtained by averaging over 300 model trainings. Both IID and non-IID partitions are considered for the training dataset. For the former, we randomly partition the training samples into 100 equal shards, each of which is assigned to one particular worker node. While for the latter, we first sort the data by digit label, divide it into 200 equal shards, and randomly assign 2 shards to each worker node. The specific experiment setup can be found in Section V-A. It is observed that the correlation between the local gradients is large at the beginning of the model training and gradually decreases over iterations. Intuitively, in SGD-based learning, the correlation is strong at the beginning of the model training since the initial model is far away from the converge point. Then, the correlation decreases as the global model is converging and the local gradients are very small and uncorrelated when the model approaches the optimal solution. It is also observed that the correlation in IID partition are much larger than those in non-IID partition for all the three datasets. This indicates that the gradient distribution with non-IID dataset partition is more dispersive than that with IID dataset partition as expected.

C. REVIEW OF MINIMUM COMMUNICATION COSTS

To exploit the correlation between the local gradients, the work [17] obtains the sum-rate-distortion function for the model aggregation problem. This function characterizes the minimum total communication cost at given gradient estimation distortion D. The result is given as the following theorem.

Theorem 1 [17]: For every distortion D, the sum-rate-distortion function is

$$R_{sum}(D) = \frac{K}{2}\log\left(1 + \frac{\sigma_N^2}{KD - \sigma_N^2}\right) + \frac{1}{2}\log\left(1 + \frac{\sigma_X^2}{D}\right). \tag{10}$$

Remark 1: The sum-rate-distortion function is the sum of two nonnegative terms. The first term depends on the variance of gradient noise, σ_N^2, as well as the number of worker

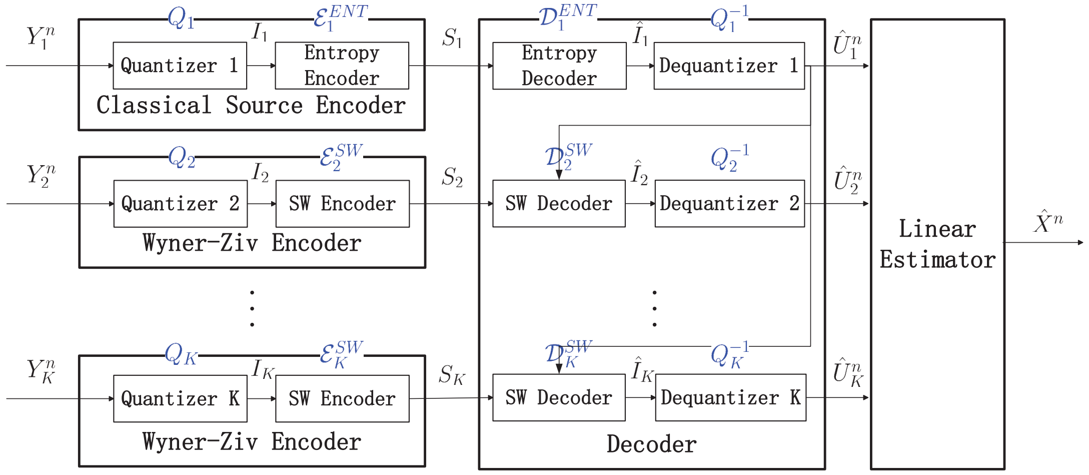


FIGURE 2. The proposed successive Wyner-Ziv coding framework for model aggregation in distributed learning.

nodes, K. It can be interpreted as the rate for quantizing K independent gradient noises. The second term depends on the variance of the global gradient, σ_X^2, which is the classical channel capacity for a Gaussian channel with a transmit power σ_X^2 and a noise power D. It thus can be interpreted as the rate for quantizing the global gradient. If each worker node transmits its own local gradient without distributed source coding, the sum-rate-distortion function reduces to $R_{in}(D) = \frac{K}{2}\log(1+\frac{\sigma_N^2}{KD-\sigma_N^2}) + \frac{K}{2}\log(1+\frac{\sigma_X^2}{KD})$. Note that the rate difference between $R_{in}(D)$ and $R_{sum}(D)$, i.e., $\frac{K}{2}\log(1+\frac{\sigma_X^2}{KD}) - \frac{1}{2}\log(1+\frac{\sigma_X^2}{D})$, represents the communication cost reduction by exploiting the correlation between worker nodes.

The work [17] only derives the minimum communication cost, but does not provide a practical coding scheme to achieve this bound. This paper aims to propose a practical coding design to achieve the sum-rate distortion function in Theorem 1.

III. DSC DESIGN UNDER KNOWN STATIC GRADIENT STATISTICS

In this section, we first propose a successive Wyner-Ziv coding framework and prove that the proposed framework can achieve the sum-rate-distortion function of the Gaussian CEO problem in (10). Then the practical SW coding in the framework is implemented by a multilevel syndrome-based method.

A. SUCCESSIVE WYNER-ZIV CODING FRAMEWORK

The successive Wyner-Ziv coding framework is depicted in Fig. 2, which consists of a pair of classical source encoder and decoder, $K-1$ pairs of Wyner-Ziv encoder and decoder, and a linear estimator. At each iteration, each P-dimensional gradient vector is divided into multiple n-length blocks and each block is separately transmitted by the proposed framework. More specifically, encoder 1 employs classical source coding to map Y_1^n to codeword U_1^n and output the corresponding index S_1. Encoder k, for $k = 2, \dots, K$, employs Wyner-Ziv encoding to map Y_k^n to codeword U_k^n and output the corresponding index S_k. The decoder first decodes codeword U_1^n from encoder 1, then successively decodes codeword U_k^n, for $k = 2, \dots, K$ from the encoder k with side information $U_1^n, \dots, U_{k-1}^n$. Finally, the decoder estimates $\hat{X}^n$ by employing a linear estimator of $U_1^n, \dots, U_K^n$.

Now, we introduce the proposed framework in detail. We first define each function in the framework, and then describe the encoding and decoding phases respectively. For each worker node k, the pair of quantizer and dequantizer is defined by:

$$Q_k : \mathcal{Y}_k^n \rightarrow \left\{1, 2, \dots, 2^{nR_k^Q}\right\}$$
$$Q_k^{-1} : \left\{1, 2, \dots, 2^{nR_k^Q}\right\} \rightarrow \mathcal{U}_k^n,$$

respectively, where R_k^Q is the quantization rate of the quantizer at worker node k, and $\mathcal{U}_k^n$ is an n-dimensional vector codebook of size $2^{nR_k^Q}$. At worker node 1, the pair of classical source encoder and decoder is defined by

$$\mathcal{E}_1^{ENT} : \left\{1, 2, \dots, 2^{nR_1^Q}\right\} \rightarrow \left\{1, 2, \dots, 2^{nR_1}\right\}$$
$$\mathcal{D}_1^{ENT} : \left\{1, 2, \dots, 2^{nR_1}\right\} \rightarrow \left\{1, 2, \dots, 2^{nR_1^Q}\right\},$$

respectively, where R_1 is the transmission rate of the worker node 1. For $k = 2, \dots, K$, the pair of SW encoder and decoder is defined by

$$\mathcal{E}_k^{SW} : \left\{1, 2, \dots, 2^{nR_k^Q}\right\} \rightarrow \left\{1, 2, \dots, 2^{nR_k}\right\}$$
$$\mathcal{D}_k^{SW} : \left\{1, 2, \dots, 2^{nR_k}\right\} \times \mathcal{U}_1^n \times \cdots \times \mathcal{U}_{k-1}^n \rightarrow \left\{1, 2, \dots, 2^{nR_k^Q}\right\}.$$

respectively, where R_k is the transmission rate of worker node k.

At the encoder side, worker node 1 first quantizes Y_1^n using codebook $\mathcal{U}_1^n$ by finding the vector codeword $U_1^n \in \mathcal{U}_1^n$ that is closest (e.g., in Euclidean distance) to Y_1^n, and outputs the quantization index I_1. Then the entropy encoder compresses I_1 as S_1, which is transmitted at rate R_1. For $k = 2, \dots, K$, worker node k first quantizes Y_k^n using codebook $\mathcal{U}_k^n$ and outputs the quantization index I_k. Then the SW encoder compresses I_k as S_k, which is transmitted at rate R_k.

At the decoder side, the parameter server first employs the entropy decoder and dequantizer 1 to reconstruct U_1^n as $\hat{U}_1^n$. Thus operations in the pair of classical source encoder and decoder for worker node 1 can be summarized as

$$\hat{U}_1^n = Q_1^{-1}\Big[\mathcal{D}^{ENT}\Big[\mathcal{E}^{ENT}\big[Q_1\big[Y_1^n\big]\big]\Big]\Big]. \tag{11}$$

The parameter server then employs the SW decoder to decode codeword U_k^n as $\hat{U}_k^n$ by using the previously decoded symbols $\hat{U}_1^n, \ldots, \hat{U}_{k-1}^n$ as side information. Thus, operations in the pair of Wyner-Ziv encoder and decoder for worker node $k \in \{2, \ldots, K\}$ can be summarized as

$$\hat{U}_k^n = Q_k^{-1}\Big[\mathcal{D}^{SW}\Big[\mathcal{E}^{SW}\big[Q_k\big[Y_k^n\big]\big], \hat{U}_1^n, \ldots, \hat{U}_{k-1}^n\Big]\Big]. \tag{12}$$

Finally, to reconstruct the sequence X^n, the parameter server employs a linear estimator, which implements the function $\psi : \mathcal{U}_1^n \times \cdots \times \mathcal{U}_K^n \to \mathcal{X}^n$ and is defined by

$$\hat{X}^n = \sum_{k=1}^{K} \alpha_k \hat{U}_k^n, \tag{13}$$

where α_k is the linear coefficient.

B. THEORETICAL ACHIEVABILITY

In this subsection, we prove that the achievable rate of the coding design can approach the sum-rate-distortion function in Theorem 1.

Given the gradient distortion D, we define the rate tuple $R_{\mathcal{K}}(D) \in \mathbb{R}_+^K$ by

$$\begin{aligned} R_k(D) = r_k &+ \frac{1}{2}\log\left(\frac{1}{\sigma_X^2} + \sum_{i=1}^{k} \frac{1-\exp(-2r_i)}{\sigma_N^2}\right) \\ &- \frac{1}{2}\log\left(\frac{1}{\sigma_X^2} + \sum_{i=1}^{k-1} \frac{1-\exp(-2r_i)}{\sigma_N^2}\right), k = 1, \ldots, K, \end{aligned} \tag{14}$$

where $(r_1, r_2, \ldots, r_K) \in \mathbb{R}_+^K$ satisfying

$$\sum_{k=1}^{K} \frac{1-\exp(-2r_k)}{\sigma_N^2} = \frac{1}{D}. \tag{15}$$

The following theorem states that our successive Wyner-Ziv coding framework can approach the rate tuple $R_{\mathcal{K}}(D)$ for any $(r_1, r_2, \ldots, r_K) \in \mathbb{R}_+^K$ satisfying (15).

Theorem 2: Let $(R_1^*, \ldots, R_K^*)$ be the rate tuple $R_{\mathcal{K}}(D^*)$. For any $\epsilon > 0$, there exists a block length of n, one classical source encoder and $K-1$ Wyner-Ziv encoders, which compress local gradients $Y_1^n, \ldots, Y_K^n$ at rates $(R_1, \ldots, R_K)$, respectively, and a decoder which reconstructs the global gradient X^n as $\hat{X}^n$, such that

$$\begin{cases} \mathbb{E}[\hat{X}^n | X^n = x^n] = x^n & (16) \\ \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \mathbb{E}[(X[i] - \hat{X}[i])^2] = D^* & (17) \\ R_k < R_k^* + \epsilon, k = 1, \ldots, K. & (18) \end{cases}$$

Proof: Let U_k be the auxiliary random variable defined as

$$U_k = Y_k + W_k, k = 1, \ldots, K, \tag{19}$$

where $W_k \sim \mathcal{N}(0, \sigma_{W_k}^2)$ is independent of $(X, Y_1, \ldots, Y_K)$ and is independent for different k. The random variable U_k can be interpreted as a quantized version of Y_k and the quantizer determines the variance of the quantization noise $\sigma_{W_k}^2$. The quantization noise W_k of an ideal quantization $Q_k(\cdot)$ is required to be Gaussian distributed and independent of the local gradient Y_k. We assume $\epsilon' < \epsilon$. To construct the random codebook for encoder k, for $k = 1, \ldots, K$, draw $2^{nI(U_k;Y_k)+n\epsilon'}$ n-length U_k vectors randomly according to the marginal of U_k, which is jointly typical with the observed vector Y_k^n (there will be at least one such codeword for large enough n with high probability). It can be proved by the generalized Markov lemma in [25, Lemma 5] that X^n, $\boldsymbol{Y}^n$ and $\boldsymbol{U}^n$ are jointly typical with a high probability because $U_k - Y_k - X - Y_j - U_j, k \neq j$ are Markov chains.

Since the encoder at worker node 1 employs classical source coding $\mathcal{E}_1^{ENT}$, the corresponding decoder can recover U_1^n accurately with a high probability if

$$R_1 > I(U_1; Y_1) + \epsilon', \tag{20}$$

where $I(U_1; Y_1) = R_1^*$ by letting $r_1 \triangleq I(U_1; Y_1|X)$. The encoder at worker node k, for $k = 2, \ldots, K$, employs SW coding $\mathcal{E}_k^{SW}$ and the ideal SW coding can be capable of compressing the quantized sources to their joint entropy. Specifically, the encoder partitions its $2^{nI(U_k;Y_k)+n\epsilon'}$ codewords into 2^{nR_k} bins. Then, the encoder picks a codeword U_k^n which is jointly typical with Y_k^n and sending the corresponding bin index. The corresponding decoder attempts to recover codeword U_k^n from the specified bin with side information $U_1^n, \ldots, U_{k-1}^n$. By the mutual packing lemma in [26, Lemma 12.2], no other set of codewords in the specified bin can be jointly typical with a high probability if

$$I(Y_k; U_k) - R_k < I(U_1, \ldots, U_{k-1}; U_k) - \epsilon', \tag{21}$$

and this condition can be rewritten as

$$\begin{aligned} R_k &> I(Y_k; U_k) - I(U_1, \ldots, U_{k-1}; U_k) + \epsilon' && (22) \\ &= h(U_k|U_1, \ldots, U_{k-1}) - h(U_k|Y_k) + \epsilon' && (23) \\ &= I(Y_k; U_k|U_1, \ldots, U_{k-1}) + \epsilon', && (24) \end{aligned}$$

where $h(\cdot)$ is the differential entropy function, and $I(Y_k; U_k|U_1, \ldots, U_{k-1}) = R_k^*$ by letting $r_k \triangleq I(U_k; Y_k|X)$. After recovering $\{U_k\}_{k=1}^K$, the parameter server reconstructs $\hat{X}^n$ using a linear estimator

$$\hat{X}^n = \sum_{k=1}^{K} \alpha_k U_k^n, \tag{25}$$

where $\alpha_k = \frac{(\sigma_N^2+\sigma_{W_k}^2)^{-1}}{\sum_{k=1}^{K}(\sigma_N^2+\sigma_{W_k}^2)^{-1}}$. The conditional expectation of $\hat{X}[i]$ is given by

$$\mathbb{E}\Big[\hat{X}[i] | X[i] = x[i]\Big] \tag{26}$$

$$= \mathbb{E}\left[\sum_{k=1}^{K} \alpha_k (x[i] + N_k[i] + W_k[i])\right] \tag{27}$$

$$= \sum_{k=1}^{K} \alpha_k x[i] \tag{28}$$

$$= x[i], \tag{29}$$

where the expectation is over gradient noise $N_k[i]$ and quantization noise $W_k[i]$ for $k = 1, 2, \ldots, K$. Hence, $\hat{X}^n$ is an unbiased estimator of X^n and its distortion is given by

$$\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\left[\left(X\left[i\right] - \hat{X}[i]\right)^2\right] = \left(\sum_{k=1}^{K}\frac{1}{\sigma_N^2 + \sigma_{W_k}^2}\right)^{-1}$$
$$= \left(\sum_{k=1}^{K}\frac{1-\exp(-2r_k)}{\sigma_N^2}\right)^{-1}$$
$$= D^*. \tag{30}$$

The proof of Theorem 2 is completed. ■

We set $r_1 = r_2 = \cdots = r_K = -\frac{1}{2}(1 - \frac{\sigma_N^2}{KD})$, where $(r_1, r_2, \ldots, r_K)$ satisfies (15). Based on Theorem 2, our successive Wyner-Ziv coding framework can approach the rate tuple $R_{\mathcal{K}}(D)$ and the corresponding sum rate is given by

$$\sum_{k\in\mathcal{K}} R_k(D)$$
$$= \sum_{k\in\mathcal{K}} r_k + \frac{1}{2}\log\left(\frac{1}{\sigma_X^2} + \frac{1}{D}\right) - \frac{1}{2}\log\left(\frac{1}{\sigma_X^2}\right) \tag{31}$$
$$= \frac{K}{2}\log\left(1 + \frac{\sigma_N^2}{KD - \sigma_N^2}\right) + \frac{1}{2}\log\left(1 + \frac{\sigma_X^2}{D}\right), \tag{32}$$

which is exactly the sum-rate-distortion function in (10).

Though the asymmetric Wyner-Ziv coding design in [20] also achieves the sum-rate-distortion function, its coding framework is more complex due to applying source splitting. Specifically, for a system with K worker nodes, our coding scheme only employs one entropy coding and $K-1$ Wyner-Ziv coding, while the asymmetric Wyner-Ziv coding design employs one entropy coding and $2(K-1)$ Wyner-Ziv coding thus requires more computation complexity.

C. PRACTICAL SW CODING DESIGN

The proposed successive Wyner-Ziv coding framework consists of two key components, quantization and SW coding. As for quantization, there has been extensive research on gradient quantization schemes [8], [9], [10], [11], including vector quantization [27], and they can be readily applied to the proposed framework. However for SW coding, there is no efficient coding scheme which uses random binning as in the achievability proof of Theorem 2. Hence, this subsection focuses on the practical SW coding design in the successive Wyner-Ziv coding framework.

In practice, error correction coding, such as LDPC code, provides a flexible solution to this problem. We will focus on syndrome-based SW coding [28], where the use of a linear parity-check channel code was suggested for partitioning all the binary source sequences into bins indexed by binary syndromes of a channel code. Although the syndrome-based SW coding can be directly applied to 1-bit quantization [8], in order to enable the SW coding scheme to be suitable for multi-bit quantization schemes, we provide a multilevel syndrome-based SW coding in the following.

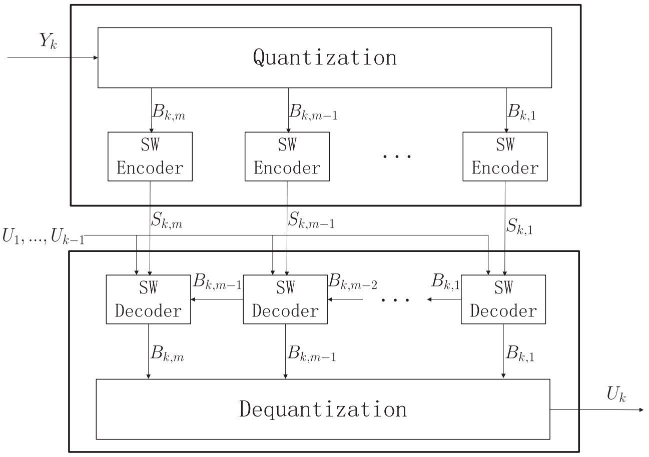


FIGURE 3. Block diagram of the multilevel syndrome-based SW coding scheme for each worker node $k \in \{2, \ldots, K\}$.

Fig. 3 shows the block diagram of the multilevel syndrome-based SW coding scheme for each worker node $k \in \{2, \ldots, K\}$. The local gradient Y_k is first quantized to a codeword U_k, which is then compressed using multilevel SW coding with $(U_1, \ldots, U_{k-1})$ as the decoder side information. Denote $J_k \in \{0, 1, \ldots, 2^{R_k} - 1\}$ as the index of U_k and write J_k as multiple bit-planes $B_{k,m}, B_{k,m-1}, \ldots, B_{k,1}$ in its binary representation, i.e., $J_k = \sum_{i=1}^{m} 2^{i-1}B_{k,i}$, where m is the number of the bit-planes, $B_{k,m}$ is the most significant bit and $B_{k,1}$ is the least significant bit. At first, $B_{k,1}$ is compressed using the first SW code at rate

$$R_{k,1} = H\big(B_{k,1}|U_1, \ldots, U_{k-1}\big), \tag{33}$$

then each $B_{k,i}$ for $i = 2, \ldots, m$ is compressed with the i-th SW code at rate

$$R_{k,i} = H\big(B_{k,i}|B_{k,1}, \ldots, B_{k,i-1}, U_1, \ldots, U_{k-1}\big), \tag{34}$$

in the ascending order of i, where $H(\cdot)$ is the entropy function. By the chain rule, we have

$$R_k = \sum_{i=1}^{m} R_{k,i} = H(J_k|U_1, \ldots, U_{k-1}) = H(U_k|U_1, \ldots, U_{k-1}). \tag{35}$$

By splitting U_k into multiple bit-planes, well-studied binary channel codes, such as LDPC, can be used to implement each SW coding of them. The idea is to split the space of input into bins, where elements with the same syndrome will be assigned to the same bin. Then we can consider each bin as a channel code and let $B_{k,i}$ pass through a hypothetical channel with the channel output $B_{k,1}, \ldots, B_{k,i-1}, U_1, \ldots, U_{k-1}$.

More specifically, consider the problem of the i-th SW coding at worker node k. Let $\boldsymbol{H}_{k,i}$ be the parity-check matrix

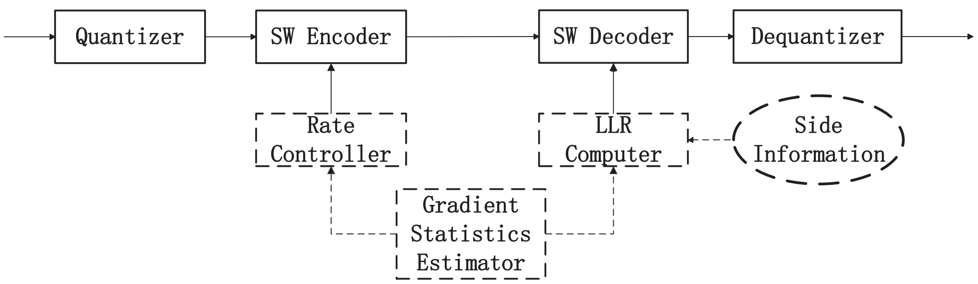


FIGURE 4. **Block diagram of the proposed low-complexity and adaptive DSC for distributed learning, where Wyner-Ziv coding block and helper block are in the solid and dash boxes, respectively.**

of a $(l_{k,i}, n)$ binary linear block code. The syndrome of the length-n binary sequence $\boldsymbol{b}_{k,i}$ is defined as $\boldsymbol{s}_{k,i} = \boldsymbol{H}_{k,i}\boldsymbol{b}_{k,i}$, which is a length-$l_{k,i}$ binary sequence.

- *Encoder:* The encoder simply computes and passes the syndrome of $\boldsymbol{b}_{k,i}$ to the decoder at rate $\frac{l_{k,i}}{n}$. By the SW theorem [29], we have
$$\frac{l_{k,i}}{n} \geq R_{k,i}. \tag{36}$$
- *Decoder:* Decoding $\boldsymbol{b}_{k,i}$ is similar to conventional channel decoding. However unlike conventional channel decoding, it will recover the source $\boldsymbol{B}_{k,i}$ as a code vector of the received syndrome (bin index) instead of a codeword. In practice, LDPC is efficient at decoding $\boldsymbol{b}_{k,i}$. Given the received syndrome $\boldsymbol{s}_{k,i}$ and the side information $\boldsymbol{b}_{k,1}, \ldots, \boldsymbol{b}_{k,i-1}, \boldsymbol{u}_1, \ldots, \boldsymbol{u}_{k-1}$, we can decode the source $\boldsymbol{b}_{k,i}$ using belief propagation.
- *Estimator:* The linear estimator jointly recovers $\hat{x}$ from all the received $\hat{u}_1, \ldots, \hat{u}_K$ at the parameter server. Given the quantization scheme, we can obtain the quantization noise variance $\sigma^2_{W_k}$ at encoder k. Hence, the linear estimator is proposed as
$$\hat{x} = \sum_{k=1}^{K} \frac{\left(\sigma_N^2 + \sigma_{W_k}^2\right)^{-1}}{\sum_{i=1}^{K}\left(\sigma_N^2 + \sigma_{W_i}^2\right)^{-1}} \hat{u}_k. \tag{37}$$
Then, the distortion of $\hat{x}$ is $(\sum_{k=1}^{K} \frac{1}{\sigma_N^2+\sigma_{W_k}^2})^{-1}$.

Remark 2: The proposed practical SW coding design is flexible and can be compatible with existing gradient quantization methods. Note that the proposed SW coding design cannot be directly applied to the case with non-IID dataset. This is because the gradients may not follow the Gaussian CEO distribution due to the non-IID gradient noise. In this case, we can apply existing methods, such as client clustering [30], [31], to mitigate the impact of non-IID datasets. Specifically, by client clustering, we can group the clients with the similar local training data into the same cluster, and then apply the proposed SW coding design within each cluster.

To design the LDPC code for the i-th bit-plane $B_{k,i}$, it is essential to estimate the desired transmission rate, i.e., $H(B_{k,i}|B_{k,1}, \ldots, B_{k,i-1}, U_1, \ldots, U_{k-1})$ and LLR of each bit-plane via gathering the statistics for the hypothetical channel with input $B_{k,i}$ and output $B_{k,1}, \ldots, B_{k,i-1}, U_1, \ldots, U_{k-1}$. In distributed learning, however, the desired transmission rate and the gradient statistics, are unknown and time-varying. Even if the gradient statistics are known, the complexity of computing the transmission rate at each iteration increases exponentially with the number of the worker nodes, which is unacceptable in a large-scale network. Therefore, the above practical SW coding scheme cannot be directly applied to distributed learning. In the next section, given the gradient statistics, we propose a low-complexity method, named semi-analytical Monte-Carlo, to calculate the statistics of the quantization variable. Based on the statistics of the quantization variable, the desired transmission rate and the LLR of each bit-plane can be easily calculated. Moreover, we also propose an online method to estimate the gradient statistics, i.e., the global gradient variance $\sigma_X^2(t)$ and the gradient noise variance $\sigma_N^2(t)$, at each iteration t.

IV. LOW-COMPLEXITY AND ADAPTIVE DSC FOR DISTRIBUTED LEARNING

This section proposes a low-complexity and adaptive DSC for distributed learning when the gradient statistics are unknown and time-varying. The block diagram is shown in Fig. 4, which consists of two blocks: Wyner-Ziv coding block and helper block. The Wyner-Ziv coding block consists of quantizer, SW encoder, SW decoder, and dequantizer, and it corresponds to the compression modules of the local gradient in the successive Wyner-Ziv framework. The helper block consists of the gradient statistics estimator, rate controller, and LLR computer, and it is located on the parameter server. The helper block provides necessary information for the SW encoder and SW decoder at each iteration, so that practical SW coding can be applied when the gradient statistics are unknown and time-varying. Specifically, the gradient statistics estimator estimates the gradient statistics online based on the quantized local gradients received in past iterations. The rate controller calculates the transmission rate of each worker node based on the estimated gradient statistics and informs all the worker nodes about the rate. The LLR computer calculates the LLR of each bit-plane based on the estimated gradient statistics and the side information (the quantized local gradients received at the current iteration) at the parameter server, and then informs the corresponding SW decoder.

In the following, we will introduce the implementation of these three helper blocks, respectively. Then, we design an efficient DSC-based distributed learning process. Finally, we analyze the complexity of the proposed DSC and the delay introduced by the extra communication and computation.

A. THE STATISTICS OF QUANTIZATION VARIABLE

In this subsection, we propose a low-complexity method to calculate the statistics of quantization variable given the gradient statistics $\sigma_X^2(t)$ and $\sigma_N^2(t)$. We compute the conditional probability $\Pr(B_{k,1} = b_{k,1}, \ldots, B_{k,i} = b_{k,i}|U_1 = u_1, \ldots, U_{k-1} = u_{k-1})$, which is an important component in calculating the transmission rate and LLR as will be shown in the next subsection. We define $Q_k(\cdot)$ as the quantization function at worker node k, and define $J(\cdot)$ as the binary representation function. Then we have

$$\begin{aligned}
&\Pr\Big(B_{k,1} = b_{k,1}, \ldots, B_{k,i} = b_{k,i}|U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\Big) && (38)\\
&= \frac{\Pr\Big(B_{k,1} = b_{k,1}, \ldots, B_{k,i} = b_{k,i}, U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\Big)}{\Pr(U_1 = u_1, \ldots, U_{k-1} = u_{k-1})} && (39)\\
&= \frac{\displaystyle\underset{\substack{J(Q_k(y_k))^{(i)}=b_{k,i},\ldots,b_{k,1}\\ Q_1(y_1)=u_1,\ldots,Q_{k-1}(y_{k-1})=u_{k-1}}}{\int\cdots\int} f(y_1, \ldots, y_k)dy_1, \ldots, dy_k}{\displaystyle\underset{Q_1(y_1)=u_1,\ldots,Q_{k-1}(y_{k-1})=u_{k-1}}{\int\cdots\int} f(y_1, \ldots, y_{k-1})dy_1, \ldots, dy_{k-1}}, && (40)
\end{aligned}$$

where $J(\cdot)^{(i)}$ represents the least significant i bits of the binary representation $J(\cdot)$ and the joint probability density function $f(y_1, \ldots, y_k)$ is given by

$$f(y_1, \ldots, y_k) = \int \frac{1}{\sigma_X\sqrt{2\pi}} e^{-\frac{x^2}{2\sigma_X^2}} \prod_{i=1}^{k} \frac{1}{\sigma_N\sqrt{2\pi}} e^{-\frac{(y_i-x)^2}{2\sigma_N^2}} dx. \quad (41)$$

1) TRADITIONAL MONTE CARLO METHOD

Note that it is very hard to calculate (40) analytically since it involves multi-dimensional integration. The work [32] uses Monte Carlo simulations to estimate (38) because they are more flexible and can be easily applied to different quantizers. Specifically, draw L independent samples from the distribution $X \sim N(0, \sigma_X^2)$ and $Y_k \sim N(X, \sigma_N^2)$ for each $k = 1, \ldots, K$, and let the l-th sample be denoted as x^l and y_k^l, respectively. Define $\mathbf{1}(\cdot)$ as an indicator function that equals one if its argument is true and zero otherwise. Based on these L samples, we can approximate the probability in (38) as

$$\begin{aligned}
&\Pr\Big(B_{k,1} = b_{k,1}, \ldots, B_{k,i} = b_{k,i}|U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\Big)\\
&\approx \frac{\sum_{l=1}^{L} \mathbf{1}\left(\substack{J(Q_k(y_k^l))^{(i)}=b_{k,i},\ldots b_{k,1}\\ Q_1(y_1^l)=u_1,\ldots,Q_{k-1}(y_{k-1}^l)=u_{k-1}}\right)}{\sum_{l=1}^{L} \mathbf{1}\big(Q_1(y_1^l) = u_1, \ldots, Q_{k-1}(y_{k-1}^l) = u_{k-1}\big)} \quad (42)
\end{aligned}$$

2) SEMI-ANALYTICAL MONTE CARLO METHOD

It is very expensive to calculate (42) accurately in the case of a large number of worker nodes because the number of samples required for the traditional Monte Carlo method increases exponentially with the number of worker nodes. Observing that the local gradients given the global gradient are independent over the worker nodes in the CEO problem, we propose a semi-analytical Monte Carlo simulation to estimate (38) alternatively. Specifically, we perform L times the following independent simulations. At the l-th simulation, we first draw a sample x^l from the distribution $X \sim N(0, \sigma_X^2)$. Given the global gradient x^l, the local gradients $\{U_k\}_{k=1}^K$ are independent of each other, then we have

$$\begin{aligned}
&\Pr\Big(U_i = u_i, U_j = u_j|X = x^l\Big)\\
&= \Pr\Big(U_i = u_i|X = x^l\Big)\Pr\Big(U_j = u_j|X = x^l\Big), \forall i \neq j \quad (43)
\end{aligned}$$

Then, by semi-analytical Monte Carlo simulation, we can approximate the conditional probability in (38) as

$$\begin{aligned}
&\Pr\Big(B_{k,1} = b_{k,1}, \ldots, B_{k,i} = b_{k,i}|U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\Big)\\
&\approx \frac{\frac{1}{L}\sum_{l=1}^{L} \Pr\Big(\substack{B_{k,1}=b_{k,1},\ldots,B_{k,i}=b_{k,i}\\ U_1=u_1,\ldots,U_{k-1}=u_{k-1}} |X = x^l\Big)}{\frac{1}{L}\sum_{l=1}^{L} \Pr\big(U_1 = u_1, \ldots, U_{k-1} = u_{k-1}|X = x^l\big)}\\
&= \frac{\sum_{l=1}^{L} \Pr\left(\substack{B_{k,1}=b_{k,1}\\ \cdots\\ B_{k,i}=b_{k,i}} |X = x^l\right) \prod_{i=1}^{k-1} \Pr\big(U_i = u_i|X = x^l\big)}{\sum_{l=1}^{L} \prod_{i=1}^{k-1} \Pr\big(U_i = u_i|X = x^l\big)}. \quad (44)
\end{aligned}$$

Note that the conditional probabilities on $X = x^l$ in (44) can be calculated analytically and independently. As will be shown in Section V-C, the required number of samples in this method does not scale with the number of worker nodes. Hence, the proposed semi-analytical simulation method is more computationally efficient than the traditional Monte Carlo simulation. Based on (38), we provide the rate controller and LLR computer for computing the transmission rate and the LLR, respectively.

B. RATE CONTROLLER AND LLR COMPUTER

The estimation of the transmission rate is critical. When the estimated rate is larger than the ground truth, the redundant information will be transmitted, otherwise, the decoding will fail. Armed with the semi-analytical Monte Carlo method, the rate controller estimates the transmission rate for the i-th bit-plane at worker node k as

$$\begin{aligned}
&H\big(B_{k,i}|B_{k,1}, \ldots, B_{k,i-1}, U_1, \ldots, U_{k-1}\big)\\
&= \sum_{\substack{b_{k,1}\\ \cdots\\ b_{k,i-1}}} \sum_{\substack{u_1\\ \cdots\\ u_{k-1}}} \Pr\begin{pmatrix} B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}\\ U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\end{pmatrix}\\
&\cdot \mathcal{H}\left(\Pr\left(B_{k,i} = 1|\begin{matrix} B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}\\ U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\end{matrix}\right)\right), \quad (45)
\end{aligned}$$

where

$$\mathcal{H}(p) = p\log_2\frac{1}{p} + (1-p)\log_2\frac{1}{1-p}. \quad (46)$$

Recall that the decoding of the i-th bit-plane of worker node k, i.e., $B_{k,i}$, can be viewed as channel decoding over the hypothetical channel with input $B_{k,i}$ and output $B_{k,1}, \ldots, B_{k,i-1}, U_1, \ldots, U_{k-1}$. The channel statistics, which is needed for decoding $B_{k,i}$, is entirely captured by the LLR $L_{ch}(b_{k,i})$. The LLR $L_{ch}(b_{k,i})$ can be obtained as shown at the top of the next page, where the conditional probability can also be calculated by semi-analytical Monte Carlo simulation (44).

C. GRADIENT STATISTICS ESTIMATION

In the above design of the rate controller and LLR computer, it is assumed that the parameter server has perfect knowledge on the gradient statistics $\sigma_X^2(t)$ and $\sigma_N^2(t)$. However, in distributed learning, the gradient statistics $\sigma_X^2(t)$ and $\sigma_N^2(t)$ are usually unknown and can even vary over time. In this subsection, we propose methods to estimate the global gradient variance $\sigma_X^2(t)$ and the gradient noise variance $\sigma_N^2(t)$ at each iteration t. We first consider an offline estimation where the perfect local gradients are available at the parameter server. Then we consider a more realistic online estimation where only the historical local gradients are available at the parameter server.

1) OFFLINE ESTIMATION WITH PERFECT LOCAL GRADIENTS

We first estimate $\sigma_N^2(t)$ and the sum of $\sigma_X^2(t)$ and $\sigma_N^2(t)$ based on the perfect local gradients, then we estimate $\sigma_X^2(t)$ by taking the difference between them.

Note that the local gradient is a noisy version of the global gradient, i.e., $Y_k(t) = X(t) + N_k(t)$, hence given the global gradient $X(t)$, the variance of $Y_k(t)$ is equal to the variance of $N_k(t)$. The local gradients $\{g_{k,p}(t)\}_{k\in\mathcal{K}}$ at dimension p from K worker nodes can be viewed as K samples of variable $Y_k(t)|X(t)$ with variance $\sigma_N^2(t)$. We can obtain the following P unbiased estimations of $\sigma_N^2(t)$ as

$$\hat{\sigma}_{N,p}^2(t) = \frac{1}{K-1}\sum_{k=1}^{K}\left(g_{k,p}(t) - \bar{g}_p(t)\right)^2, p = 1, 2, \ldots, P, \tag{48}$$

where $\bar{g}_p(t) = \frac{1}{K}\sum_{k=1}^{K} g_{k,p}(t)$ and P is the number of gradient dimension. Then we estimate $\sigma_N^2(t)$ by averaging these P estimations as

$$\hat{\sigma}_N^2(t) = \frac{1}{P}\sum_{p=1}^{P}\hat{\sigma}_{N,p}^2(t). \tag{49}$$

Note that the mean and the variance of the local gradient $Y_k(t)$ is given by

$$\mathbb{E}[Y_k(t)] = \mathbb{E}[X(t)] + \mathbb{E}[N_k(t)] = 0, \tag{50}$$

and

$$\begin{aligned}\mathrm{Var}(Y_k(t)) &= \mathbb{E}\Big[(Y_k(t) - \mathbb{E}[Y_k(t)])^2\Big] & (51)\\ &= \mathbb{E}\Big[(X(t) + N_k(t))^2\Big] & (52)\\ &= \mathbb{E}\Big[X^2(t)\Big] + \mathbb{E}\Big[N_k^2(t)\Big] & (53)\\ &= \sigma_X^2(t) + \sigma_N^2(t), & (54)\end{aligned}$$

respectively, where (53) results from the independence of $N_k(t)$ and $X(t)$. The local gradients $\{\boldsymbol{g}_k(t)\}_{k\in\mathcal{K}}$ from K worker nodes can be viewed as KP samples of variable $Y_k(t)$ with variance $\sigma_X^2(t) + \sigma_N^2(t)$. The unbiased estimation of $\sigma_X^2(t) + \sigma_N^2(t)$ is thus given by

$$\hat{\sigma}_{X+N}^2(t) = \frac{1}{KP}\sum_{k=1}^{K}\sum_{p=1}^{P} g_{k,p}^2(t). \tag{55}$$

Based on (49) and (55), $\sigma_X^2(t)$ can be estimated as

$$\hat{\sigma}_X^2(t) = \hat{\sigma}_{X+N}^2(t) - \hat{\sigma}_N^2(t). \tag{56}$$

2) ONLINE ESTIMATION WITH QUANTIZED LOCAL GRADIENTS

In practical distributed learning, the parameter server can only receive the quantized local gradients from worker nodes. In addition, the quantized local gradients of iteration t are not available at the parameter server at the beginning of the iteration t, and thus cannot be used for the estimation of $\sigma_X^2(t)$ and $\sigma_N^2(t)$. Hence, offline estimation with perfect local gradients is infeasible in practice. As an alternative, we propose an online estimation of $\sigma_X^2(t)$ and $\sigma_N^2(t)$ based on the quantized local gradients at the beginning of the iteration $t+1$. The main idea is to first estimate the sum of $\sigma_X^2(t)$ and $\sigma_N^2(t)$, then the optimal $\sigma_X^2(t)$ and $\sigma_N^2(t)$ are obtained by maximizing the likelihood of the received quantized local gradients.

Similar to the previous case, the parameter server can compute the estimation of $\sigma_X^2(t) + \sigma_N^2(t)$ with the help of worker nodes. Since the local gradient $\boldsymbol{g}_k(t)$ is available at worker node k and can be viewed as P samples of variable $Y_k(t)$ with variance $\sigma_X^2(t)+\sigma_N^2(t)$, worker node k can estimate $\sigma_X^2(t) + \sigma_N^2(t)$ by

$$\hat{\sigma}_{X+N,k}^2(t) = \frac{1}{P}\sum_{p=1}^{P} g_{k,p(t)}^2, k = 1, 2, \ldots, K. \tag{57}$$

$$\begin{aligned}&L_{ch}\big(b_{k,i}\big)\\ &= \log\frac{\Pr\big(B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}, U_1 = u_1, \ldots, U_{k-1} = u_{k-1}|B_{k,i} = 1\big)}{\Pr\big(B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}, U_1 = u_1, \ldots, U_{k-1} = u_{k-1}|B_{k,i} = 0\big)}\\ &= \log\frac{\Pr\big(B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}, U_1 = u_1, \ldots, U_{k-1} = u_{k-1}, B_{k,i} = 1\big)}{\Pr\big(B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}, U_1 = u_1, \ldots, U_{k-1} = u_{k-1}, B_{k,i} = 0\big)} - \log\frac{\Pr\big(B_{k,i} = 1\big)}{\Pr\big(B_{k,i} = 0\big)}\\ &= \log\frac{\Pr\big(B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}, B_{k,i} = 1|U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\big)}{\Pr\big(B_{k,1} = b_{k,1}, \ldots, B_{k,i-1} = b_{k,i-1}, B_{k,i} = 0|U_1 = u_1, \ldots, U_{k-1} = u_{k-1}\big)} - \log\frac{\Pr\big(B_{k,i} = 1\big)}{\Pr\big(B_{k,i} = 0\big)} \qquad (47)\end{aligned}$$

Then each worker node transmits this estimation to the parameter server along with the quantized local gradient. Note that the communication cost of transmitting this estimation is negligible compared to transmitting the high-dimensional local gradient. Based on the received K estimates, the parameter server can estimate $\sigma_X^2(t)+\sigma_N^2(t)$ by

$$\hat{\sigma}_{X+N}^2(t) = \frac{1}{K}\sum_{k=1}^{K}\hat{\sigma}_{X+N,k}^2(t). \tag{58}$$

Now we aim to compute the estimation of $\sigma_X^2(t)$ and $\sigma_N^2(t)$. At the beginning of the iteration $t+1$, the quantized local gradients $\{\boldsymbol{u}_k(t)\}_{k=1}^K$ and $\hat{\sigma}_{X+N}^2(t)$ are available at the parameter server. In the following, we apply maximum likelihood estimation for $\sigma_X^2(t)$ and $\sigma_N^2(t)$ based on the quantized local gradients $\{\boldsymbol{u}_k(t)\}_{k=1}^K$. The likelihood function is given by

$$L\Big(\sigma_X^2(t), \sigma_N^2(t) | \{\boldsymbol{u}_k(t)\}_{k=1}^K\Big) \tag{59}$$

$$= \Pr_{\sigma_X^2(t),\sigma_N^2(t)}\left(\boldsymbol{U}_1 = \boldsymbol{u}_1(t), \ldots, \boldsymbol{U}_K = \boldsymbol{u}_K(t)\right) \tag{60}$$

$$= \prod_{p=1}^{P}\Pr_{\sigma_X^2(t),\sigma_N^2(t)}\Big(U_1 = u_{1,p}(t), \ldots, U_K = u_{K,p}(t)\Big), \tag{61}$$

where the calculation of each probability is similar to (44) by applying semi-analytical Monte Carlo simulation. We estimate $\sigma_X^2(t)$ and $\sigma_N^2(t)$ by maximizing the likelihood function. By taking $\hat{\sigma}_{X+N}^2(t)$ as a constraint, this maximization problem is given by

$$\mathcal{P}_1 : \max_{\sigma_X^2(t),\sigma_N^2(t)>0} \quad L\Big(\sigma_X^2(t), \sigma_N^2(t) | \{\boldsymbol{u}_k(t)\}_{k=1}^K\Big) \tag{62a}$$

$$s.t. \quad \sigma_X^2(t) + \sigma_N^2(t) = \hat{\sigma}_{X+N}^2(t). \tag{62b}$$

Let $\tilde{\sigma}_X^2(t), \tilde{\sigma}_N^2(t)$ denote the optimal solution of problem $\mathcal{P}_1$. It is found empirically in [21] that the ratio of noise variance to gradient variance, i.e., $\frac{\sigma_N^2(t)}{\sigma_X^2(t)}$ changes slowly with iteration t. Moreover, it can be easily found that the gradient correlation only depends on the ratio of $\sigma_X^2(t)$ to $\sigma_N^2(t)$, and is not related to the absolute value of $\sigma_X^2(t)$ and $\sigma_N^2(t)$. Hence, the transmission rate and LLR calculations at iteration $t+1$ can be determined by the gradient statistics estimation $\tilde{\sigma}_X^2(t), \tilde{\sigma}_N^2(t)$.

D. DSC-BASED DISTRIBUTED LEARNING PROCESS

The proposed DSC-based distributed learning algorithm is presented in Algorithm 1. The timing diagram of the learning process is shown in Fig. 5, where 5 worker nodes are illustrated as an example. At the beginning of the iteration t, the parameter server broadcasts the global model $\boldsymbol{w}(t)$ to all the worker nodes (line 3). At the same time, the parameter server estimates the parameters $\sigma_X^2(t-1)$ and $\sigma_N^2(t-1)$ at the gradient statistics estimator (line 4) and calculates the transmission rate based on (45) at the rate controller (line 5). The parameter server informs all the worker nodes about the transmission rate (line 6). Then each worker node locally takes one step of minibatched SGD on the current model using its local dataset (line 8). Then, each worker node simply computes the scalar $\hat{\sigma}_{X+N,k}^2(t)$ based on the observed local gradient $\boldsymbol{g}_k(t)$ (line 9). Each worker node generates the parity-check matrix with the corresponding transmission rate (line 10). Each worker node also quantizes the observed local gradient $\boldsymbol{g}_k(t)$ as $Q(\boldsymbol{g}_k(t))$, converts it to binary representation $\boldsymbol{b}_{k,i}(t)$ (line 11) and computes the syndromes of $\boldsymbol{b}_{k,i}(t)$ for $i = 1, 2, \ldots, m$ (line 12). All the worker nodes send the syndromes and the scalars to the parameter server serially (line 13). Note that the transmission order in the successive Wyner-Ziv framework can be adjusted according to the order in which the local model training is completed. Upon a syndrome of a worker node arriving, the parameter server applies LDPC decoding to recover $\boldsymbol{b}_{k,i}(t)$ based on the received syndrome and LLR, then the LLR for the syndrome of the next bit-plane is calculated at the LLR computer (line 15). The LDPC decoding at the parameter server and the transmission of the next syndrome are executed in parallel, except for the decoding of the last syndrome. The linear estimator reconstructs $\hat{\boldsymbol{g}}(t)$ from all the quantized local gradients based on (37) (line 16). At the end of the iteration, the parameter server updates the global model (line 17). The process repeats itself until the model converges.

Algorithm 1 DSC-Based Distributed Learning

```
1: Initialize $\boldsymbol{w}(1)$, $\sigma_X^2(0)$ and $\sigma_N^2(0)$ at the parameter server;
2: for iteration $t = 1, ..., T$ do
3:     Broadcast global model $\boldsymbol{w}(t)$;
4:     Calculate $\tilde{\sigma}_X^2(t-1), \tilde{\sigma}_N^2(t-1)$ by solving $\mathcal{P}_1$;
5:     Calculate $R_{k,i}(t)$ based on (45);
6:     Inform each worker node $k$ of $\{R_{k,i}(t)\}_{i=1}^m$;
7:     for each worker node $k \in \mathcal{K}$ do
8:         Calculate $\boldsymbol{g}_k(t) = \nabla \frac{1}{B}\sum_{i\in\mathcal{B}_k(t)} f_i(\boldsymbol{w}(t))$;
9:         Calculate $\hat{\sigma}_{X+N,k}^2(t) = \frac{1}{P}\sum_{p=1}^{P} g_{k,p}^2(t)$;
10:        Generate $\boldsymbol{H}_{k,i}(t)$ satisfying $\frac{l_{k,i}}{n} \geq R_{k,i}(t)$;
11:        Perform quantization $\boldsymbol{b}_{k,m}(t), ..., \boldsymbol{b}_{k,1}(t) = J(Q(\boldsymbol{g}_k(t)))$;
12:        Generate syndrome $\boldsymbol{s}_{k,i}(t) = \boldsymbol{H}_{k,i}(t)\boldsymbol{b}_{k,i}(t)$;
13:        Send $\boldsymbol{s}_{k,1}(t), ..., \boldsymbol{s}_{k,m}(t)$ and $\hat{\sigma}_{X+N,k}^2(t)$ to parameter server;
14:     end for
15:     Recover $\hat{\boldsymbol{b}}_{k,i}(t)$ based on $L_{ch}(b_{k,i}(t))$ and $\boldsymbol{s}_{k,i}(t)$;
16:     $\hat{\boldsymbol{g}}(t) = \frac{1}{K}\sum_{k=1}^{K}\hat{\boldsymbol{u}}_k(t)$;
17:     $\boldsymbol{w}(t+1) = \boldsymbol{w}(t) - \eta(t)\hat{\boldsymbol{g}}(t)$;
18: end for
19: Return $\boldsymbol{w}(T+1)$;
```

E. EXTRA DELAY ANALYSIS

From the timing diagram, we can find that the time consumed by distributed learning mostly overlaps the time consumed by DSC, such that the total delay of DSC-based distributed learning is not the simple sum of them. In the following, we will show that the communication and computation of the helpers in DSC are sufficiently efficient such that the extra delay introduced by DSC is almost negligible.

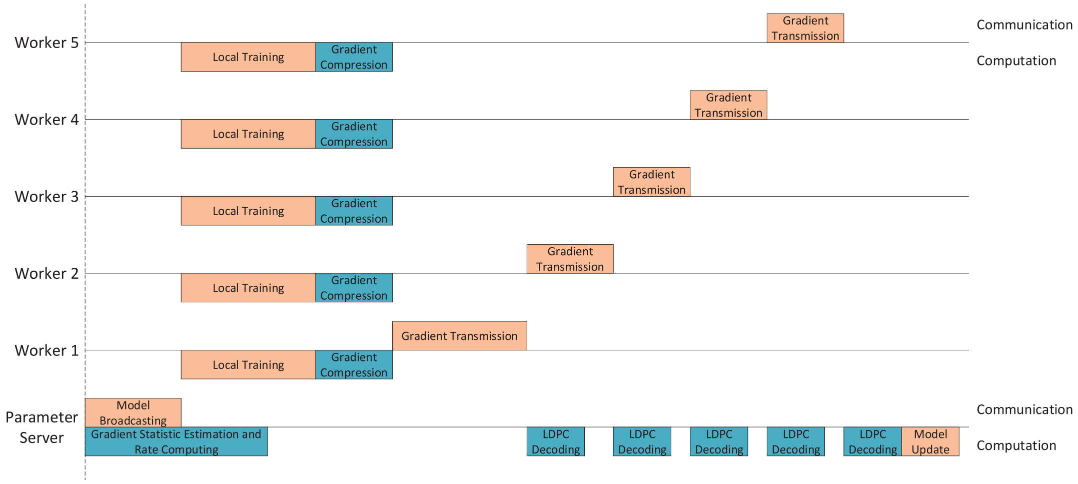


FIGURE 5. The timing diagram of DSC-based distributed learning process, where the graph depicts communication time above the horizontal axis and computation time below it, the time consumed by DSC is marked in blue, and the time consumed by distributed learning is marked in red.

1) COMMUNICATION

We reveal that the extra communication overhead introduced by DSC-based distributed learning is almost negligible. The generation of the parity-check matrix in each SW encoder requires the transmission rate computed by the parameter server. This requires an extra communication overhead from the parameter server to the distributed worker nodes. However, the transmission rate for each SW encoder only consists of m scalars, where m is the number of quantization bits, while the global gradient is a P-dimensional vector. In the considered experiment, m is not larger than 10 and P is in the order of 10^6. Hence, the extra communication overhead from the parameter server to the distributed worker nodes is negligible compared with that of transmitting the gradient vector.

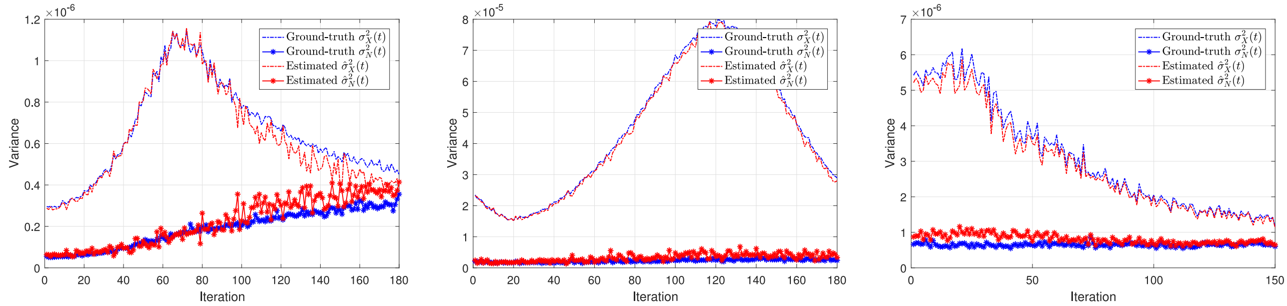
At iteration $t+1$, the transmission rate is redesigned based on the estimated gradient statistics, i.e., $\tilde{\sigma}_X^2(t), \tilde{\sigma}_N^2(t)$. To estimate $\tilde{\sigma}_X^2(t), \tilde{\sigma}_N^2(t)$, the gradient statistics estimator at the parameter server requires the quantized local gradients $\{\boldsymbol{u}_k(t)\}_{k=1}^K$ and scalars $\{\hat{\sigma}_{X+N,k}^2(t)\}_{k=1}^K$ in (57). The extra communication cost of transmitting these K scalars is negligible compared with that of transmitting the gradient vector.

In contrast, the work [19] exploits an autoencoder to capture the common information that exists in the local gradients. The autoencoder is trained using the perfect local gradients collected from all the worker nodes, which, however, introduces an extra communication overhead since the local gradients for the training cannot be compressed. The work [16] estimates the correlation based on the local gradients of a group of worker nodes at each iteration, and an extra communication overhead is still needed since the local gradients of workers nodes in this group cannot be compressed during correlation estimation.

2) COMPUTATION

We first reveal that the computational complexity of gradient statistics estimation and rate controller is only polynomial. Specifically, the complexity of gradient statistics estimation is $O(\frac{PLK}{\epsilon})$, where ϵ is the accuracy of problem $\mathcal{P}_1$ and the complexity of computing the likelihood function is $O(PLK)$. The complexity of computing the transmission rate is $O(2^{KM}LK^2M)$ since the number of bit-planes is $(K-1)M$, the number of conditions in (45) is 2^{KM} and the complexity of estimating each probability is $O(LK)$. In practice, we only estimate the transmission rate of the top K^{top} worker nodes, and the transmission rate of the rest worker nodes adopts the transmission rate of the K^{top}-th worker node. As shown in Fig. 9 in Section V-D, when a large number of worker nodes have transmitted the local gradients to the parameter server as side information, the transmission rate decreases slowly with the index of worker node. Note that the computation of gradient statistics estimator and rate controller and the communication of model broadcasting can perform in parallel. Hence, the gradient statistics estimator and rate controller do not bring extra delay in the case of limited communication bandwidth.

Now we demonstrate that LDPC decoding and computing LLR have polynomial complexity, and the throughput of practical LDPC decoders is high enough. The complexity of LDPC decoding is $O(MKPt^{BP}n)$, where t^{BP} is the maximum iteration times of BP algorithm and n is the block length. The complexity of computing LLR is $O(PLK^2M)$ since the number of bit-planes is $(K-1)M$, the number of LLR needs to be estimated in each bit-plane is P and the complexity of estimating each probability in (47), shown at the bottom of page 7, is $O(LK)$. In practice, the use of LDPC decoding can enjoy the benefits of its mature chip technology. Specifically, many advanced hardware architectures for LDPC decoders have been proposed in the literature [33], [34], which achieve high information throughput of 181Gbps and 588Gbps, respectively. Note that the LDPC decoding at the parameter server and the transmission of the next syndrome are executed in parallel. Hence, the extra delay can only be introduced by compressing the local gradients and decoding the last bit-plane of the last



(a) MLP model on MNIST. (b) Resnet20 model on SVHN. (c) Resnet56 model on CIFAR-10.

FIGURE 6. The ground truth and online estimation of $\sigma_X^2(t)$ and $\sigma_N^2(t)$ over iterations for three models, where the models are updated with ideal gradients without any transmission error.

local gradient in the case of limited communication bandwidth. Furthermore, these extra delay introduced by DSC is negligible in large-scale distributed learning systems since the extra delay introduced by DSC does not scale with the number of worker nodes and the communication delay in distributed learning increases linearly with the number of worker nodes.

V. EXPERIMENT RESULTS

In this section, we provide experiment results to evaluate the gradient statistics estimator and the rate controller equipped with semi-analytical Monte Carlo estimation. We also show the communication benefits for model aggregation in distributed learning resulting from the proposed DSC scheme.

A. EXPERIMENT SETUP

We conduct experiments in a simulated environment where the number of worker nodes involved in each training iteration is $K = 10$ if not specified otherwise. We evaluate the model training on the real-world datasets MNIST, SVHN, and CIFAR-10 datasets. The MNIST dataset consists of 10 categories ranging from digit 0 to 9 and a total of 70000 labeled data samples (60000 for training and 10000 for testing). The SVHN dataset includes 99289 labeled data samples (73257 for training and 26032 for testing). The CIFAR-10 dataset includes 60000 color images (50000 for training and 10000 for testing) of 10 different types of objects. In this paper, we consider IID data distribution, and we randomly partition the training samples into 100 equal shards, each of which is assigned to one particular worker node. We adopt multilayer perceptron (MLP) model on the MNIST dataset, Resnet20 model on the SVHN dataset and Resnet56 model on the CIFAR-10 dataset. The hyper-parameters are set as follows: momentum optimizer is 0.5, local batch size is 128 and learning rate $\eta = 0.01$.

B. GRADIENT STATISTICS AND ESTIMATION

We first study the gradient statistics and evaluate the gradient statistics estimator. Fig. 6 shows the ground truth and estimation of $\sigma_X^2(t)$ and $\sigma_N^2(t)$ over iterations for three models, where these three models approach convergence at the end of the last iteration. The ground truth of $\sigma_X^2(t)$ and $\sigma_N^2(t)$ are calculated offline based on the perfect local gradients, and the online estimation is based on the one-bit quantized value of local gradients. It is observed that the gap between the ground truth and the estimation of gradient statistics is very small, which indicates that the proposed method for estimating gradient statistics online is effective. Note that $\sigma_X^2(t)$ and $\sigma_N^2(t)$ changes slowly with iteration t, which is consistent with the result in [21]. It is also observed that the global gradient variance $\sigma_X^2(t)$ is large at the beginning of the model training and gradually decreases over iterations while the gradient noise variance $\sigma_N^2(t)$ remains approximately unchanged for all three models. Intuitively, in SGD-based learning, the initial model is far away from the converge point at the beginning of the model training, and the global gradient dominates the local gradients, hence $\sigma_X^2(t)$ is large at the beginning. When the model almost converges, the global gradient almost vanishes, hence $\sigma_X^2(t)$ gradually decreases to zero. On the other hand, the gradient noise variance $\sigma_N^2(t)$ remains approximately unchanged due to the randomness of local batches throughout the training process.

C. RATE CONTROLLER

We evaluate the performance of the proposed rate controller equipped with semi-analytical Monte Carlo simulation. For the one-bit quantization scheme, we estimate the rate of a single worker node by using traditional Monte Carlo simulation and semi-analytical Monte Carlo simulation, respectively, and each simulation is repeated 100 times, where the samples are drawn from the Gaussian CEO distribution with given gradient statistics.

Fig. 7 shows the mean of the estimated rate versus the number of samples L in each simulation, where $\sigma_X^2 = 10$, $\sigma_N^2 = 1$, $K = 10$, the number of simulations is 100. The ground truth is calculated by the semi-analytical Monte Carlo simulation with a large enough $L = 10^9$. It can be observed that the rates estimated by the semi-analytical Monte Carlo simulation are unbiased throughout the whole regime of L while there is a large bias in the rates estimated by traditional Monte Carlo simulation with low L. Specifically, the mean of estimated rates approaches zero for the traditional

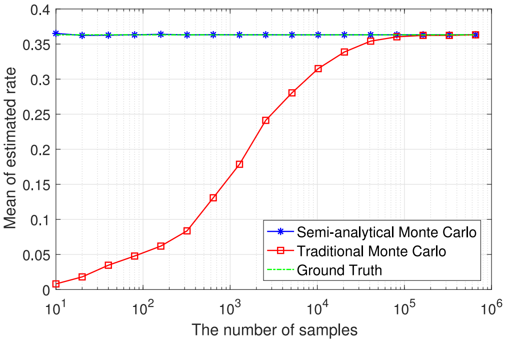


FIGURE 7. The mean of the estimated rate versus the number of samples L, where $\sigma_X^2 = 10$, $\sigma_N^2 = 1$, $K = 10$ and the number of simulations is 100.

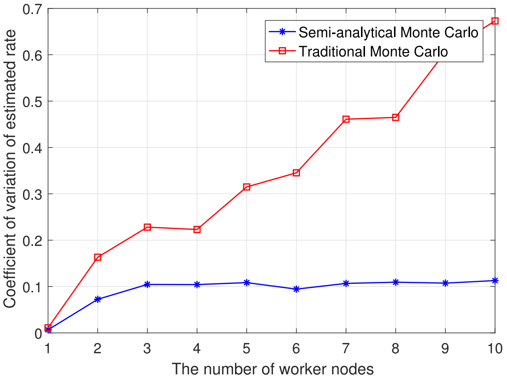


FIGURE 8. Coefficient of variation of the estimated rate versus the index of worker nodes k, where $\sigma_X^2 = 10$, $\sigma_N^2 = 1$, $L = 100$ and the number of simulations is 100.

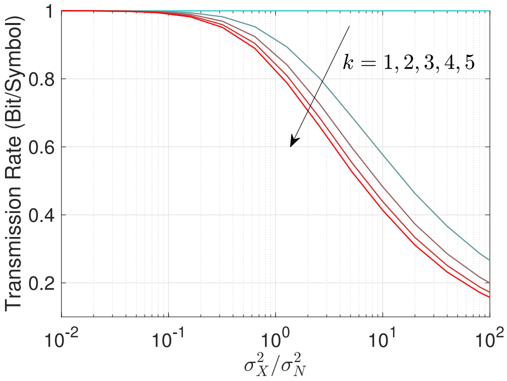


FIGURE 9. The rate of each worker nodes as a function of the ratio σ_X^2/σ_N^2 for one-bit quantization scheme, where the number of worker nodes is $K = 5$.

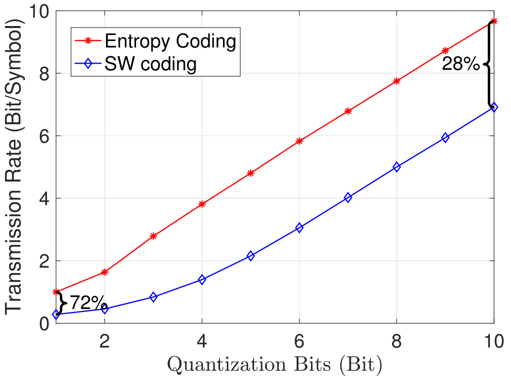


FIGURE 10. The rates of entropy coding and multilevel syndrome-based SW coding over the quantization bits for multiple-bit uniform quantization, where the ratio $\sigma_X^2/\sigma_N^2 = 100$.

monte carlo estimation when the number of samples is 10. In this case, the number of samples that meet any condition in (45) is not greater than 1 with large probability since the number of the conditions (i.e., $2^9 = 512$) is much larger than $L = 10$. This implies that all the estimated conditional probabilities in (45) are either 0 or 1, thus each conditional entropy in (45) estimated by traditional Monte Carlo simulation approaches zero. However, the ground-truth conditional entropy is greater than 0. This brings the gap of transmission rate between the ground truth and the traditional monte carlo estimation when L is small. In contrast, even if the number of samples L is small, the accurate conditional probability in (45) can be calculated analytically for each sample x^l by the proposed semi-analytical Monte Carlo simulation. Thus our scheme can estimate the transmission rate more accurately as shown in Fig. 7.

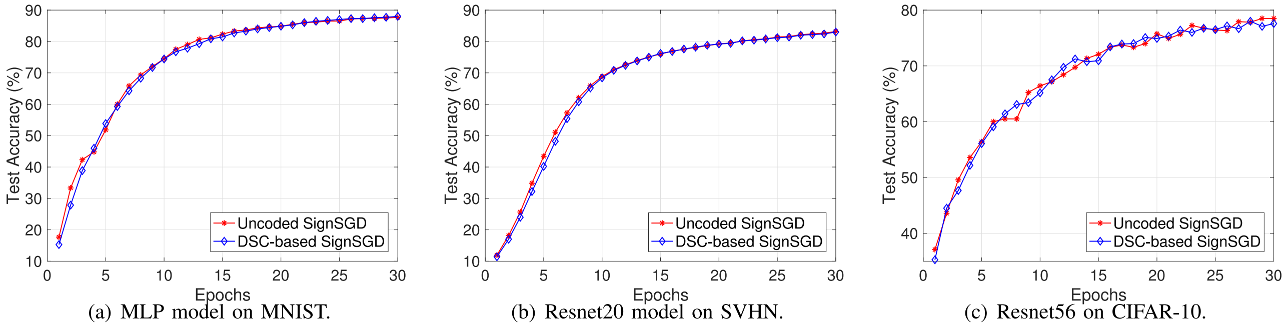
Fig. 8 shows the coefficient of variation (CV) of the estimated rate versus the index of worker node k, where $\sigma_X^2 = 10$, $\sigma_N^2 = 1$, $L = 100$ and the number of simulations is 100. It is observed that the CV of the rates estimated by the proposed scheme is stable throughout the whole regime of worker nodes k while the CV of the rates estimated by traditional Monte Carlo simulation increases over the index of worker nodes k. This is because the number of conditions in (45) grows exponentially over k, and the number of samples contained in each condition decreases dramatically with k, leading to reduced precision of the rate estimation for the traditional Monte Carlo simulation. In contrast, for each sample, the probability of each condition and its corresponding conditional probability, i.e., the first and second terms in (45), can be analytically calculated by the proposed method even if the index of worker node k is large, hence an increase in k does not reduce the precision of the rate estimation for proposed semi-analytical Monte Carlo method.

In summary, the proposed semi-analytical Monte Carlo simulation is unbiased and achieves higher precision than the traditional Monte Carlo simulation.

D. COMMUNICATION COST

We first demonstrate the communication cost with different gradient statistics and quantization bits by simulation and the rates results are calculated by semi-analytical Monte Carlo simulation with the number of samples $L = 10^6$.

Recall that the correlation between the local gradients depends on the ratio σ_X^2/σ_N^2. Fig. 9 shows the rate of each worker node as a function of the ratio σ_X^2/σ_N^2 for the one-bit quantization scheme, where the number of worker nodes is $K = 5$ and the number of samples is $L = 10^6$. It is observed that the communication cost of DSC decreases over the index of worker node due to the fact that the side information at

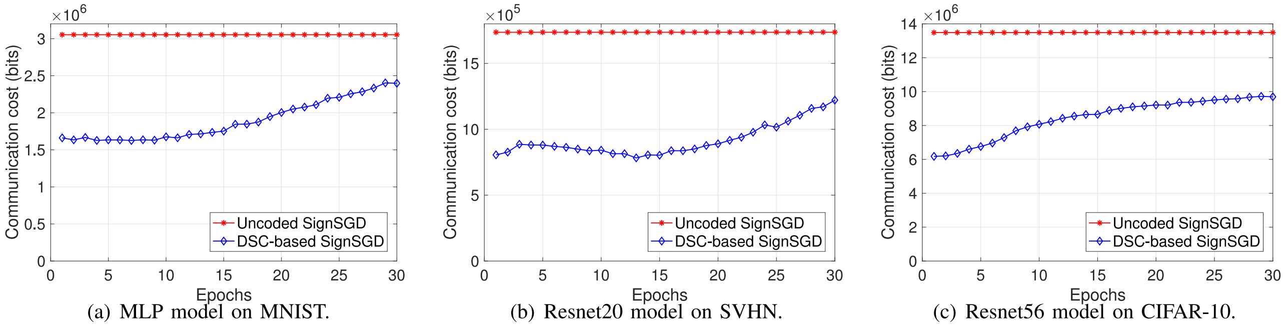


(a) MLP model on MNIST.

(b) Resnet20 model on SVHN.

(c) Resnet56 on CIFAR-10.

FIGURE 11. The test accuracy over epochs for uncoded SignSGD and DSC-based SignSGD on three models.



(a) MLP model on MNIST.

(b) Resnet20 model on SVHN.

(c) Resnet56 model on CIFAR-10.

FIGURE 12. The communication cost over epochs for uncoded SignSGD and DSC-based SignSGD on three models.

the parameter server increases. It is also observed that the rate for all the worker nodes decreases with the ratio σ_X^2/σ_N^2. This is because the correlation between the local gradients becomes stronger with the ratio σ_X^2/σ_N^2.

Fig. 10 shows the transmission rates of entropy coding and multilevel syndrome-based SW coding over the quantization bits for multiple-bit uniform quantization, where the ratio is given by $\sigma_X^2/\sigma_N^2 = 100$. The multilevel syndrome-based SW coding uses the quantized local gradient of one worker node as side information. It is observed that the transmission rates increase with the number of quantization bits for both coding schemes and the relative communication cost reduction (i.e., the rate reduction of SW coding normalized by the rates of entropy coding) decreases with the quantization bits. Specifically, the relative communication cost reduction is 72% when $m = 1$ and is 28% when $m = 10$. This is because the mutual information between the quantized gradient and side information is upper bounded by the mutual information between the perfect local gradients but the entropy of the quantized gradient increases without upper bound as the quantization bits increase. Therefore, using DSC for quantization schemes with low quantization levels can achieve a higher relative communication cost reduction.

The rates of each bit-plane in multilevel syndrome-based SW coding for multiple-bit uniform quantization are shown in Table 1, where $K = 3$, ratio $\sigma_X^2/\sigma_N^2 = 10$ and $m = 6$. It can be observed that the rates of most of the top bit-planes are almost equal to one, which indicates that the corresponding bit-planes can be transmitted without compression in practice. Therefore, the multilevel syndrome-based SW coding can only compress last few bit-planes almost without extra communication cost.

TABLE 1. The rates of each bit-plane in multilevel syndrome-based sw coding for multiple-bit uniform quantization, where K = 3, ratio σ_X^2/σ_N^2 = 10 and the quantization bits is 6.

Bit-plane	1	2	3	4	5	6
Worker 1	0.9999	0.9986	0.9975	0.9961	0.9907	0.8424
Worker 2	0.9859	0.9803	0.9740	0.9606	0.6395	0.0435
Worker 3	0.9831	0.9775	0.9710	0.9490	0.5043	0.0166

Finally, we evaluate the communication cost of the proposed adaptive SW coding scheme by experiments on models MLP, Resnet20 and Resnet56. Considering that most of the bit-planes of multi-bit quantization cannot benefit from DSC and one-bit quantization can achieve a higher relative communication cost reduction, we adopt signSGD quantization to show the communication efficiency of the proposed DSC. The practical syndrome-based SW encoder is based on irregular LDPC codes. In our experiments, the block length for LDPC code equals 2400, and the maximum number of iterations is set to 100 for LDPC decoding.

Fig. 11 and Fig. 12 compare the test accuracy and the communication cost, respectively, of the proposed DSC-based scheme over training epoches t. It is observed from Fig. 11 that the accuracy gap to the uncoded SignSGD scheme is very small due to the fact that the proposed SW coding design is lossless. Specifically, the decoding bit-error rate of the LDPC code is less than 1%. It is observed from Fig. 12 that the per-epoch communication cost of the SW coding scheme is significantly less than that of the uncoded

SignSGD. Moreover, the total communication cost to train the model is saved by the DSC-based SignSGD. Specifically, the DSC-based SignSGD saves 38%, 47% and 37% of total communication costs for MLP, Resnet20 and Resnet56 models, respectively. It is also observed that the communication cost reduction of the DSC-based scheme decreases when the model almost converges (i.e., epoch $>$ 20 in Fig. 12). Intuitively, when the correlation between the local gradient is stronger, the communication cost reduced by applying DSC is larger. At the beginning of the training, the correlation between the local gradients is strong considering that the global model is far away from the optimal solution. Then, the correlation decreases as the global model is converging and the local gradients are very small and uncorrelated when the model approaches the optimal solution. This observation is consistent with the results in Fig. 1.

VI. CONCLUSION

This paper proposed a distributed source coding framework for model aggregation in distributed learning. We proved that this coding framework approaches the minimum communication cost for distributed learning. We provided a multilevel syndrome-based SW coding scheme implemented by LDPC codes when gradient statistics are known, which can be applied to existing quantization schemes. We calculated the statistics of quantization variables given the gradient statistics by semi-analytical Monte Carlo simulation with low computation complexity. We also proposed an adaptive SW coding scheme that estimates the gradient statistics based on the observed quantized gradients at the parameter server and then dynamically adjusts the LDPC codes in each iteration based on the estimation results. The experiment results showed that the proposed coding scheme reduces the communication cost without any loss of the model accuracy.

REFERENCES

[1] N. Zhang and M. Tao, "An adaptive distributed source coding design for distributed learning," in *2021 13th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2021, pp. 1–5.

[2] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, Q. V. Le, M. Z. Mao, M. Ranzato, A. W. Senior, P. A. Tucker et al., "Large scale distributed deep networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1223–1231.

[3] A. Coates, B. Huval, T. Wang, D. Wu, B. Catanzaro, and N. Andrew, "Deep learning with cots hpc systems," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1337–1345.

[4] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial Intelligence and Statistics*, 2017, pp. 1273–1282.

[5] J. Konen, H. B. McMahan, F. X. Yu, P. Richtrik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," in *International Conference on Learning Representations*, 2018.

[6] 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," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[7] J. Huang, F. Qian, Y. Guo, Y. Zhou, Q. Xu, Z. M. Mao, S. Sen, and O. Spatscheck, "An in-depth study of lte: Effect of network protocol and application behavior on performance," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 363–374, 2013.

[8] 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. Interspeech*, 2014, pp. 1058–1062.

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

[10] S. Horvóth, C.-Y. Ho, L. Horvath, A. N. Sahu, M. Canini, and P. Richtarik, "Natural compression for distributed deep learning," in *Proceedings of Mathematical and Scientific Machine Learning*, 2022, pp. 129–141.

[11] S. Horvth, D. Kovalev, K. Mishchenko, P. Richtrik, and S. Stich, "Stochastic distributed learning with gradient quantization and double-variance reduction," *Optimization Methods and Software*, pp. 1–16, 2022.

[12] A. F. Aji and K. Heafield, "Sparse communication for distributed gradient descent," in *Proc. EMNLP*, 2017, pp. 440–445.

[13] Y. Lin, S. Han, H. Mao, Y. Wang, and B. Dally, "Deep gradient compression: Reducing the communication bandwidth for distributed training," in *International Conference on Learning Representations*, 2018.

[14] D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic, "QSGD: Communication-efficient SGD via randomized quantization and encoding," *Proc. Neural Inf. Process. Syst.*, vol. 3, pp. 1710–1721, 2018.

[15] 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. Neural Inf. Process. Syst.*, 2017, pp. 1508–1518.

[16] A. Abdi and F. Fekri, "Reducing communication overhead via CEO in distributed training," in *Proc. IEEE SPAWC Workshop*, 2019, pp. 1–5.

[17] N. Zhang, M. Tao, J. Wang, and F. Xu, "Fundamental limits of communication efficiency for model aggregation in distributed learning: A rate-distortion approach," *arXiv preprint arXiv:2206.13984*, 2022.

[18] T. Berger, Z. Zhang, and H. Viswanathan, "The CEO problem [multiterminal source coding]," *IEEE Trans. Inf. Theory*, vol. 42, no. 3, pp. 887–902, 1996.

[19] L. Abrahamyan, Y. Chen, G. Bekoulis, and N. Deligiannis, "Learned gradient compression for distributed deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–1, 2021.

[20] Y. Yang, V. Stankovic, Z. Xiong, and W. Zhao, "On multiterminal source code design," *IEEE Trans. Inf. Theory*, vol. 54, no. 5, pp. 2278–2302, 2008.

[21] N. Zhang and M. Tao, "Gradient statistics aware power control for over-the-air federated learning," *IEEE Trans. Wireless Commun.*, pp. 1–1, 2021.

[22] C.-Y. Chen, J. Ni, S. Lu, X. Cui, P.-Y. Chen, X. Sun, N. Wang, S. Venkataramani, V. V. Srinivasan, W. Zhang et al., "Scalecom: Scalable sparsified gradient compression for communication-efficient distributed training," *Advances in Neural Information Processing Systems*, vol. 33, pp. 13 551–13 563, 2020.

[23] L. Wang, W. Wu, J. Zhang, H. Liu, G. Bosilca, M. Herlihy, and R. Fonseca, "Superneurons: FFT-based gradient sparsification in the distributed training of deep neural networks," *arXiv preprint arXiv:1811.08596*, 2018.

[24] S. Bubeck et al., "Convex optimization: Algorithms and complexity," *Foundations and Trends® in Machine Learning*, vol. 8, no. 3-4, pp. 231–357, 2015.

[25] Y. Oohama, "The rate-distortion function for the quadratic gaussian CEO problem," *IEEE Trans. Inf. Theory*, vol. 44, no. 3, pp. 1057–1070, 1998.

[26] A. El Gamal and Y.-H. Kim, *Network information theory*.Cambridge university press, 2011.

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

[28] A. Wyner, "Recent results in the shannon theory," *IEEE Transactions on Information Theory*, vol. 20, no. 1, pp. 2–10, 1974.

[29] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inf. Theory*, vol. 19, no. 4, pp. 471–480, 1973.

[30] A. Ghosh, J. Chung, D. Yin, and K. Ramchandran, "An efficient framework for clustered federated learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 586–19 597, 2020.

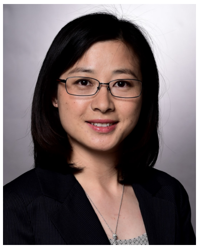
[31] K. Kopparapu and E. Lin, “Fedfmc: Sequential efficient federated learning on non-iid data,” *arXiv preprint arXiv:2006.10937*, 2020.

[32] Z. Liu, S. Cheng, A. Liveris, and Z. Xiong, “Slepian-wolf coded nested lattice quantization for wyner-ziv coding: High-rate performance analysis and code design,” *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4358–4379, 2006.

[33] R. Ghanaatian, A. Balatsoukas-Stimming, T. C. Mller, M. Meidlinger, G. Matz, A. Teman, and A. Burg, “A 588-gb/s ldpc decoder based on finite-alphabet message passing,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 26, no. 2, pp. 329–340, 2018.

[34] K. Cushon, P. Larsson-Edefors, and P. Andrekson, “A high-throughput low-power soft bit-flipping ldpc decoder in 28 nm fd-soi,” in *ESSCIRC 2018 - IEEE 44th European Solid State Circuits Conference (ESSCIRC)*, 2018, pp. 102–105.

NAIFU ZHANG received the B.Eng. and M.Eng. degrees in electrical engineering from Shanghai Jiao Tong University, Shanghai, China in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree with the Department of Electric Engineering. His research interests include coded caching, edge learning, and federated learning.



MEIXIA TAO (Fellow, IEEE) received the B.S. degree in electronic engineering from Fudan University, Shanghai, China, in 1999, and the Ph.D. degree in electrical and electronic engineering from the Hong Kong University of Science and Technology in 2003.

She is currently a Professor with the Department of Electronic Engineering, Shanghai Jiao Tong University, China. Her current research interests include wireless edge learning, coded caching, reconfigurable intelligence surfaces, and semantic communications.

Dr. Tao received the 2019 IEEE Marconi Prize Paper Award, the 2013 IEEE Heinrich Hertz Award for Best Communications Letters, the IEEE/CIC International Conference on Communications in China 2015 Best Paper Award, and the International Conference on Wireless Communications and Signal Processing 2012 Best Paper Award. She also receives the 2009 IEEE ComSoc Asia–Pacific Outstanding Young Researcher Award. She is an Associate Editor of the IEEE TRANSACTIONS ON INFORMATION THEORY and an Editor-at-Large of the IEEE Open Journal of the Communications Society. She served as a member of the Executive Editorial Committee of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2015 to 2019. She was also on the editorial board of several other journals as an Editor or a Guest Editor, including the IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS. She is currently serving as the TPC Co-Chair of IEEE ICC 2023.