Peer-to-Peer Variational Federated Learning Over Arbitrary Graphs

Xinghan Wang, Anusha Lalitha, Tara Javidi, *Fellow, IEEE*, and Farinaz Koushanfar, *Fellow, IEEE*

***Abstract*—This paper proposes a federated supervised learning framework over a general peer-to-peer network with agents that act in a variational Bayesian fashion. The proposed framework consists of local agents where each of which keeps a local *"posterior probability distribution"* over the parameters of a global model; the updating of the posterior over time happens in a local fashion according to two subroutines of: 1) variational model training given (a batch of) local labeled data, and 2) asynchronous communication and model aggregation with the 1-hop neighbors. Inspired by the popular federated learning (model averaging), the framework allows the training data to remain distributed on mobile devices while utilizing a peer-to-peer model aggregation in a social network. The proposed framework is shown to allow for a systematic treatment of model aggregation over any arbitrary connected graph with consistent (in general, non-iid) local labeled data. Specifically, under mild technical conditions, the proposed algorithm allows agents with local data to learn a shared model explaining the global training data in a decentralized fashion over an arbitrary peering/connectivity graph. Furthermore, the rate of convergence is characterized and shown to be a function of each individual agent's data quality weighted by its eigenvector centrality. Empirically, the proposed methodology is shown to work well with efficient variation Bayesian inference techniques to train Bayesian neural networks in a decentralized manner even when the local data batches are not identically distributed.**

***Index Terms*—Federated learning, variational Bayes, peer-to-peer network, decentralized learning.**

I. Introduction

PERSONAL edge devices can often use their locally observed data to learn machine learning models that improve the user experience. However, the use of local data for learning globally rich machine learning models has to address two important challenges. First of all, this type of localized data, in isolation from the data collected by other devices, might be statistically insufficient to learn a global model. Secondly, there might be severe restrictions on sharing raw forms of personal/local data due to privacy and communication cost concerns. In light of these challenges and restrictions, an alternative approach has emerged which leaves the training data distributed on the edge devices while enabling the decentralized learning of a shared model. This alternative, known as *Federated Learning*, is based on edge devices' periodic communication with a central (cloud-based) server responsible for iterative model aggregation. While addressing the privacy constraints on raw data sharing, and significantly reducing the communication overload as compared to synchronized stochastic gradient descent (SGD), this approach falls short in fully decentralizing the training procedure. Many practical peer-to-peer networks are dynamic and a regular access to a fixed central server, which coordinates the learning across devices, is not always possible. Existing methods based on federated learning cannot handle such general networks where central server is absent and/or when the data has severe heterogeneity across the network.

To summarize, some of the major challenges encountered in a fully decentralized learning paradigm are: (i) *Statistical Insufficiency and non-IID Data Distributions:* The local and individually observed data distributions are likely to be less rich than the global training set. For example, a subset of features associated with the global model may be missing locally. (ii) *Restriction on Data Exchange:* Due to privacy concerns, agents do not share their raw training data with the neighbors. Furthermore, model parameter sharing has been shown to reduce the communication requirements significantly. (iii) *Lack of Synchronization:* There may not be a single agent with whom every agent communicates, which can synchronize the learning periodically. (iv) *Localized Information Exchange:* Agents are likely to limit their interactions and information exchange to a small group of their peers which can be viewed as the 1-hop neighbors on the social network graph. Furthermore, information obtained from different peers might be viewed differently, requiring a heterogeneous model aggregation strategy.

Contributions: We consider a fully decentralized learning paradigm where agents iteratively update their models using local data and aggregate information from their neighbors to their local models. In particular, we consider a learning rule where agents take a variational (Bayes) learning approach via the introduction of a posterior distribution over a parameter space characterizing the unknown global model.

Our contributions are as follows:

1) On the algorithmic side, our decentralized learning rule is inspired by works on social learning, distributed

Manuscript received 2 January 2022; revised 27 April 2022; accepted 1 July 2022. Date of publication 11 July 2022; date of current version 14 November 2022. This work was supported by NSF-CNS under Award 2016737. *(Corresponding author: Xinghan Wang.)*

Xinghan Wang is with the Computer Science and Engineering Department, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: x2wang@eng.ucsd.edu).

Anusha Lalitha was with the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92093 USA. She is now with the AI Vertical Services, AWS AI Labs, Cupertino, CA 95014 USA (e-mail: anlalith@amazon.com).

Tara Javidi and Farinaz Koushanfar are with the Electrical and Computer Engineering Department, University of California at San Diego, La Jolla, CA 92093 USA (e-mail: tjavidi@eng.ucsd.edu; fkoushanfar@eng.ucsd.edu).

Digital Object Identifier 10.1109/JSAIT.2022.3189051

hypothesis testing literature [1]–[4]. Our social learning rule builds on our recent work on federated log posterior averaging [5], [6] and generalizes posterior averaging propose in [7].

2) We provide the first theoretical guarantees for the realizable case where the hypothesis (parametric model) class contains the true labeling function.
   - We prove that, under mild technical assumptions, each agent will eventually learn the true parameters associated with global model, with potentially non-IID local data distributions.
   - We provide analytical characterization of the rate of convergence of the posterior probability at each agent in the network as a function of network structure and local learning capacity as measured by the relative entropy.
3) Unlike prior work, we allow a general network structure as well as non-IID data distributions. As a consequence, our work provides first known theoretical guarantees on convergence for a variational federated learning on an arbitrary connected graph.
4) In addition to our theoretical results, we investigate the performance of our proposed variational learning empirically.
   - We show that the practical advantage of our approach for non-IID data over the classical federated averaging especially as the complexity of the task increases.
   - We illustrate the impact of social network structure on the model convergence.
   - We also demonstrate the ability of our approach to scale up in a time-varying asynchronous network.
   - We demonstrate the ability of our approach to deal with all 4 cases of non-IID local data.

In this regard, our work bridges the theoretical gap between decentralized training methodologies, Bayesian statistical learning, distributed hypothesis testing, and the computational advantages of variational Bayes' methods.

Related Work: Our fully decentralized training methodology extends federated learning [8]–[10] to general graphs in a Bayesian setting and does away with the need of having a centralized controller or IID data distribution across the network. Our learning rule generalizes various Bayesian inference techniques such as [11]–[14] and variational continual learning techniques such as [11], [13] to the decentralized learning/training case.

Our work can be viewed as a Bayesian variant of communication-efficient methods based on Stochastic Gradient Descent (SGD) [15]–[17] that allow the agents to make several local computations and then periodically average the local models, with the consensus step inspired by previous works on social learning and distributed hypothesis testing [1]–[4]. We contrast our work with the vast literature on decentralized optimization based on SGD [18]–[28], where local (stochastic) gradients are computed for each instance of data and communication happens at a rate comparable to number of local updates. The communication rules of decentralized SGD methods originate from prior works on distributed averaging [29], [30], and are either *gossip-based* for doubly stochastic communication matrix [22], [23], [26], or *PushSum-based* [31] for column stochastic communication matrix [27], [28]. In these works typically strong convexity of local objective functions is assumed, with nodes having IID data, and the algorithms and theoretical results are presented in the *realizable* setting, where there exist a parameterization of the model that agrees with the true underlying data generating model. For an overview on the decentralized optimization methods refer to the survey [25].

We also note the relation and difference between our work and decentralized variational Bayesian inference [13], [32]. While these works and ours utilize Bayesian learning agents, [32] merges the local posteriors in a *one-shot* manner after observing all data, and [13] works in a *streaming* setting where data is distributed from a central server and the goal is infer the posterior relevant to the most recent data. Our work combines the advantage of allowing uncertainty from using a Bayesian-like posterior, and the periodic averaging aspect from FedAvg [10]. A recent work [7] fits Gaussian posteriors to local datasets in a federated way, however it is unable to provide theoretical guarantees on convergence rate and can be viewed as a special case to our work when the problem is *realizable* and posterior distribution can shown to remain in the Gaussian family.

Notation: We use boldface for vectors $\mathbf{v}$ and denote its i-th element by v_i. Let $[n] = \{1, 2, \ldots, n\}$. Let $\mathcal{P}(A)$ and $|A|$ denote the set of all probability distributions and the number of elements respectively on a set A. Let $G(\theta, \sigma^2)$ denote the pdf of a Gaussian random variable with mean θ and variance σ^2. Let $D_{\mathrm{KL}}(P_Z||P'_Z)$ be the Kullback–Leibler (KL) divergence between two probability distributions $P_Z, P'_Z \in \mathcal{P}(\mathcal{Z})$.

II. The Model: Decentralized Learning

In this section, we formally describe the label generation model at each node, the communication graph, and a criterion for successful learning over the network.

Consider a group of N individual nodes. Each node $i \in [N]$ has access to a dataset $\mathcal{D}^{(i)}$ consisting of instance-label pairs, $(X_t^{(i)}, Y_t^{(i)})$ where $i \in [N]$ and $t \geq 1$. Each instance $X_t^{(i)} \in \mathcal{X}_i \subseteq \mathcal{X}$, where $\mathcal{X}_i$ denotes the local instance space of node i and $\mathcal{X}$ denotes a global instance space $\mathcal{X} \subseteq \cup_{i=1}^N \mathcal{X}_i$. Similarly, let $\mathcal{Y}$ denote the set of all possible labels over all the nodes.[1] The samples $\{X_1^{(i)}, X_2^{(i)}, \ldots, X_t^{(i)}\}$ are independent and identically distributed (IID) over time, and are generated according to a distribution $\mathsf{P}_X^{(i)} \in \mathcal{P}(\mathcal{X}_i)$. We view the model generating the labels for each node i as a probabilistic model with a distribution $\mathsf{P}_{Y|X}(y|x)$, $\forall\, y \in \mathcal{Y}, \forall x \in \mathcal{X}$.

The learners' objective is to (collaboratively) approximate the probabilistic labeling function $\mathsf{P}_{Y|X}(y|x)$ with a parametric probabilistic model $f(\cdot|X, \theta)$ with $\theta \in \Theta \subseteq \mathbb{R}^d$ representing the model parameter(s). The next two examples highlight the two complementary and necessary components of learning here: 1) model training utilizing local data, and 2) communication and model aggregation across many agents.

[1]Some examples include, $\mathcal{Y} = \mathbb{R}$ for regression and $\mathcal{Y} = \{0, 1\}$ for binary classification.

Example 1 (Decentralized Linear Regression): Consider a linear regression problem [33] with P_X denoting the input distribution over $\mathbb{R}^d$ and latent variable $\theta^* \in \mathbb{R}^d$, which for each input $x \in \mathbb{R}^d$ generates a label $y \in \mathbb{R}$ as $y = \langle \theta^*, x \rangle + \eta$, where $\eta \sim \mathcal{N}(0, \alpha^2)$. Now consider the problem of two agents learning labeling function $\mathsf{P}_{Y|X}(\cdot|x) \sim \mathcal{N}(\langle \theta^*, x \rangle, \alpha^2)$ where the agents' local input distributions is restricted to the marginal distributions over sets $\mathcal{X}_1 := \{x \in \mathbb{R}^d : [x_1, \ldots, x_m, 0, \ldots, 0]\}$ and $\mathcal{X}_2 := \{x \in \mathbb{R}^d : [0, \ldots, 0, x_{m+1}, \ldots, x_d]\}$ for some integer m. Let $\mathsf{P}_X^{(1)}$ be the input distribution with support over $\mathcal{X}_1$ and $\mathsf{P}_X^{(2)}$ be the input distribution with support over $\mathcal{X}_2$ and define $\mathsf{P}_X(x) := \mathsf{P}_X^{(1)}(x_{1:m}) \times \mathsf{P}_X^{(2)}(x_{m+1:d})$ for any $x \in \mathbb{R}^d$. It is clear that the observed input distribution for learner 1 lies in $\mathcal{X}_1$, hence learner 1, in isolation, can only learn the first m coordinates of the inputs; similarly, learner 2 can only access the remaining $d - m$ coordinates locally.

Example 2 (Decentralized Classification With Bayesian Neural Networks): Consider the problem of training a neural network (NN) with weights $\theta \in \mathbb{R}^d$ and output layer $f(\cdot|x, \theta)$ to approximate the true probabilistic labeling function $\mathsf{P}_{Y|X}(\cdot|x)$[2] generating labels in $\mathcal{Y}$. Two variational (Bayesian) learners rely on labeled training data $(X_t^{(i)}, Y_t^{(i)})$ which are generated according to a distribution $\mathsf{P}_X^{(i)} \in \mathcal{P}(\mathcal{X}_i)$ to calculate the posterior distribution of the weights $\mathsf{P}(\theta|\mathcal{D})$ given the training data.[3] Two agents are tasked to arrive at a classifier even when their observations are restricted to non-overlapping partitions of the label space, $\mathcal{Y}_1$ and $\mathcal{Y}_2$ and the input distributions $\mathsf{P}_X^{(1)}$, and $\mathsf{P}_X^{(2)}$.

In both examples above, neither learners can learn the global modeling in isolation unless there is a method for communication and model aggregation.

Centralized Variational Learning: We first consider the centralized setting as a benchmark in which a (super-)learner is assumed to have access to all agents' training data samples $\mathcal{D} = \{(X_t^{(i)}, Y_t^{(i)}) \in (\mathcal{X}, \mathcal{Y})\}_{i \in [N], t \geq 1}$, where samples are independent and identically distributed (IID) across time with joint distribution $\mathsf{P}_{XY} = \mathsf{P}_X \mathsf{P}_{Y|X}$ and P_X is a measure on $\mathcal{X}$. This centralized learner approximates the true probabilistic labeling function with a parametric probabilistic model with $\theta \in \Theta$.

Towards this objective, a variational (Bayesian) learner places a prior $q^{(0)} = p_\theta \in \mathcal{P}(\Theta)$ on the latent parameter/variable θ and computes the posterior distribution $q^{(t)}(\theta|\mathcal{D}) \in \mathcal{P}(\Theta)$ after observing $\mathcal{D}_{1:t} = \{(\mathbf{X}_s, \mathbf{Y}_s)\}_{1 \leq s \leq t} = \{(X_s^{(i)}, Y_s^{(i)}) \in (\mathcal{X}, \mathcal{Y})\}_{i \in [N], 1 \leq s \leq t}$. These updates follow the variational learner's preference and iterative fashion:

$$q^{(t)} = \underset{\pi \in \mathcal{Q}}{\operatorname{argmin}} \left\{ D_{KL}\left(\pi \middle\| q^{(t-1)}\right) + \mathbb{E}_\pi\left[-\log f(\mathbf{Y}_t|\mathbf{X}_t, \cdot)\right] \right\}. \quad (1)$$

Bernstein-von Mises theorem under model misspecification [35] (See Theorem 2 in the Appendix) asserts that the posterior converges to a point mass at θ^*, explaining the observed labeling distribution:

$$\theta^* \in \underset{\theta \in \Theta}{\operatorname{argmin}} \mathbb{E}_{\mathsf{P}_X}\left[D_{KL}\left(\mathsf{P}_{Y|X}(\cdot|X) || f(\cdot|X, \theta)\right)\right] \quad (2)$$

Note that given posterior $q^{(t)}(\theta|\mathcal{D}_{1:t})$, the predictive distribution of the label of a new test input $\hat{X}$ sampled from P_X is given by

$$\mathsf{P}_t(\cdot|\hat{X}) := \int f\left(\cdot|\hat{X}, \theta\right) dq^{(t)}(\theta|\mathcal{D}_{1:t})$$

This means that as the posterior $q^{(t)}(\theta|\mathcal{D}_{1:t})$ concentrates on θ^*, the predictive label distribution converges to the best explanation for the observed data.

Definition 1: A learning problem is said to be realizable if there exists a $\theta^* \in \Theta$ such that $f(\cdot|X, \theta^*) = \mathsf{P}_{Y|X}(\cdot|X)$ almost surely under P_X. In this case, as shown in Equation (2), it is possible to derive the expected loss to zero.

Decentralized Variational Learning: In this setting, while that the data samples across all learners are labeled by the same unknown probabilistic labeling function $\mathsf{P}_{Y|X}(y|x)$ for all $y \in \mathcal{Y}$ given any $x \in \mathcal{X}$, each agent only has access to local data. In other words, each agent $i \in [N]$ approximates this labeling function by $f^{(i)}(\cdot|X, \theta)$ for any $X \in \mathcal{X}$. The goal is to agree on a global labeling function f parameterized by $\theta \in \Theta$ which approximates $\mathsf{P}_{Y|X}$ optimally with respect to the global distribution P_X. Formally, the agents are required to collectively optimize the following objective function:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \sum_{i=1}^{N} \mathbb{E}_{\mathsf{P}_X}\left[D_{KL}\left(\mathsf{P}_{Y|X}(\cdot|X) || f^{(i)}(\cdot|X, \theta)\right)\right].$$

In this work, we assume variational (Bayesian) agents: agent i places a prior $q_i^{(0)} = p_\theta^{(i)} \in \mathcal{P}(\Theta)$ on the latent variable θ and infers the posterior distribution $q_i^{(t)}(\theta|\mathcal{D}_{1:t}^{(i)}) \in \mathcal{P}(\Theta)$ after observing t batches of local data, $\mathcal{D}_{1:t}^{(i)} = (X_s^{(i)}, Y_s^{(i)})_{1 \leq s \leq t}$, drawn IID from the distribution $\mathsf{P}_{XY}^{(i)}$.

Let us first consider the case of an isolated variational learner whose posterior updates are limited to $\mathcal{D}^{(i)}$. In this setting, agent i's observation is shaped by the local data distribution and might be informationally deficient with respect to the global model. When $\mathsf{P}_{XY}^{(i)} = \mathsf{P}_{XY}$ for all i, then Bernstein-Von Mises Theorem states that the posterior distribution $\mathsf{P}(\theta|\mathcal{D}_{1:t}^{(i)})$, where $\mathcal{D}_{1:t}^{(i)} := \{(X_\tau^{(i)}, Y_\tau^{(i)}) : \tau \in [t]\}$, is guaranteed to converge to a globally consistent parameter. In this case, statistically the agents can all learn the global model from local data and there is no need for communication except for boosting the convergence rate.

Definition 2: We call the local datasets to have *non-IID data distributions* when there exists $i \in [N]$ for which $\mathsf{P}_{XY}^{(i)} \neq \mathsf{P}_{XY}$. Prior work in federated learning [36]–[38] summarizes the following cases of *non-IID data distribution*:

1) *label distribution skew:* $\mathsf{P}_Y \neq \mathsf{P}_Y^{(i)}$.
2) *feature distribution skew:* $\mathsf{P}_X \neq \mathsf{P}_X^{(i)}$.
3) *same label different features:* $\mathsf{P}_{X|Y} \neq \mathsf{P}_{X|Y}^{(i)}$.
4) *quantity skew:* $\mathsf{P}_{XY} = \mathsf{P}_{XY}^{(i)}$, but the number of data samples on each learner is different.

When the data available to agents locally have non-IID distributions, i.e., $\mathsf{P}_{XY}^{(i)} \neq \mathsf{P}_{XY}$, there is no guarantee that

[2]For the case of classification, $f(\cdot|x, \theta)$ denotes the final softmax layer, and for the case of regression, $f(\cdot|x, \theta)$ is a Gaussian distribution with mean as NN output $f(\cdot|x, \theta)$

[3]While exact Bayesian inference on the weights of a neural network is intractable due to large parameter space, in the recent years variational methods to approximate exact Bayesian updates enables Bayesian NN to be learned in a computationally efficient manner [12], [14], [34].

the local models will converge in a globally consistent manner. Therefore, each learner i must aim not only to arrive at a good approximation to its local observations, $\mathcal{D}^{(i)} = \{(X_t^{(i)}, Y_t^{(i)})\}_{t\geq 1}$, but also collaborate with other agents to ensure consenting on a consistent model across. This all suggests that minimizing (2) requires communications across learners, which gives rise to our communication model and necessitates our decentralized learning rule.

Before we provide the communications/social network graph model, we note that if $\mathsf{P}_{XY}^{(i)} = \mathsf{P}_{XY}$ for all learners $i \in [N]$, then the distributions of local dataset of all learners are identical with the global distribution. This case, which is known as federated learning with *IID data distribution*, is not the focus of this work. It is simple to note that if $\mathcal{D}^{(i)} = \{(X_t^{(i)}, Y_t^{(i)})\}_{t\geq 1}$ is identically distributed across nodes according to the joint distribution $\mathsf{P}_{XY}(x, y) = \mathsf{P}_X(x)\mathsf{P}_{Y|X}(y|x) = \mathsf{P}_Y(y)\mathsf{P}_{X|Y}(x|y)$, then each agent is informationally equivalent to the centralized learner in that the agent is only required to wait longer to essentially receive n times more training data. In other words, in our analysis, we do not consider the problem of data *quantity skew* across the agents [38] since when $\mathsf{P}_{XY} = \mathsf{P}_{XY}^{(i)}$, the difference in the number of data samples on each learner only impacts the rate of learning and not the convergence to θ^* itself which is the focus of this work. We do, on the other hand, consider this phenomenon in our experiments.

Arbitrary Graph for Communication and Model Aggregation: We model the communication between learners via a directed graph with vertex set $[N]$. We define the neighborhood of learner i, denoted by $\mathcal{N}(i)$, as the set of all learners j who have an edge going from j to i. We assume $i \in \mathcal{N}(i)$. Furthermore, if learner $j \in \mathcal{N}(i)$, learner i receives information from learner j. The social interaction of the learners is characterized by a stochastic matrix W, $\sum_{j=1}^N W_{ij} = 1$ and $W_{ii} = 1 - \sum_{j=1, j\neq i}^N W_{ij}$. The weight $W_{ij} \in [0, 1]$ is strictly positive if and only if $j \in \mathcal{N}(i)$. The weight W_{ij} denotes the confidence learner i has on the information it receives from learner j.

III. Peer-to-Peer Variational Learning Over an Arbitrary Graph

We now present our peer-to-peer variational learning rule. We assume that at each time instant $t \geq 0$, every learner $i \in [N]$ gets a batch of B observations $(\mathbf{X}_t^{(i)}, \mathbf{Y}_t^{(i)}) = (X_{Bt+1:B(t+1)}^{(i)}, Y_{Bt+1:B(t+1)}^{(i)})$. Let Θ denote the latent variable space and typically we choose $\Theta = \mathbb{R}^d$ for some $d > 1$. Each learner takes a Bayesian-like approach and places a prior distribution $p_\theta^{(i)} \in \mathcal{P}(\Theta)$ over the latent variable. Let $q_i^{(t)}$ denote the density of the posterior maintained by each learner i at time t. We introduce a decentralized learning rule which generalizes a learning rule considered in the social learning literature [2]–[4], to obtain the posterior $q_i^{(t)}$ at each learner i at every time instant t. We restrict posterior distributions to a predetermined family of parametric distributions. This allows us to implement the decentralized algorithm in a computationally tractable manner. Let $\mathcal{Q} \subseteq \mathcal{P}(\Theta)$ be a family of posterior parametric distributions.

Each learner i starts with $q_i^{(0)} = p_\theta^{(i)}$ and at every time step $t \geq 0$ the following events happen at every learner $i \in [N]$:

1) Draw a batch of B i.i.d samples $(\mathbf{X}_t^{(i)}, \mathbf{Y}_t^{(i)})$ from distribution $\mathsf{P}_{XY}^{(i)}$.
2) *Approximate Bayesian Update Step:* Approximate the local Bayesian update on $q_i^{(t-1)}$ to form a public posterior $b_i^{(t)}$ using the following rule:

$$b_i^{(t)} = \underset{\pi\in\mathcal{Q}}{\operatorname{argmin}} \Big\{ D_{KL}\Big(\pi \,\Big\|\, q_i^{(t-1)}\Big) + \mathbb{E}_\pi\Big[-\log f\Big(\mathbf{Y}_t^{(i)}|\mathbf{X}_t^{(i)}, \cdot\Big)\Big]\Big\}. \quad (3)$$

3) *Communication Step:* Learner i sends $b_i^{(t)}$ to learner j if $i \in \mathcal{N}(j)$ and receives $b_j^{(t)}$ from neighbors $j \in \mathcal{N}(i)$.
4) *Consensus Step:* Update posterior distribution by averaging the log posterior distributions received from neighbors, i.e., for each $\theta \in \Theta$,

$$q_i^{(t)}(\theta) = \frac{\exp\Big(\sum_{j=1}^N W_{ij} \log b_j^{(t)}(\theta)\Big)}{\int_\Theta \exp\Big(\sum_{j=1}^N W_{ij} \log b_j^{(t)}(\phi)\Big) d\phi}. \quad (4)$$

Remark 1 (Approximate Bayesian Update): Minimization performed in Equation (3) is referred to as Variational Inference (VI) and the minimand is referred to as the variational free energy or evidence lower bound (ELBO) [12], [14], [34], [39].

Remark 2 (Gaussian Case): The variational computation as well as the normalization involved in consensus step (Equation (4)) require multi-dimensional integration. However, in most practical application Gaussian posterior distributions are used to approximate the true posterior and the consensus step reduces to updating the mean and covariance matrices. We will show this in Lemma 1 in Section IV. This also shows that our algoirthm is closely related to the algorithm proposed in [7] where instead of following a consensus step on log posteriors, the authors propose a direct averaging of posteriors.

IV. Main Results

In this section, we first provide our analytical guarantee for the convergence of our proposed algorithm. Additionally, we provide a special case of our algorithm where the variational posterior class $\mathcal{Q}$ is chosen to be Gaussian. This special case, which later is used in Section V is shown to reduce the complexity of the algorithm substantially and allow for a closed form characterization of our consensus step.

We now make the following assumptions for the main theorem.

Assumption 1: The network is a connected aperiodic graph. Specifically, W is an aperiodic and irreducible stochastic matrix.

Assumption 2: For all agents $i \in [N]$, assume: (i) The prior $b_i^{(0)}(\theta) > 0$ for all $\theta \in \Theta$. (ii) There exists an $\alpha > 0$, $L > 0$ such that $\alpha < f^{(i)}(y|x, \theta) < L$, for all $y \in \mathcal{Y}$, $\theta \in \Theta$ and $x \in \mathcal{X}$. This guarantees the log-likelihood ratio $|\log \frac{f^{(i)}(y|x,\theta)}{f^{(i)}(y|x,\theta')}|$ is bounded for all $i \in [N]$, $y \in \mathcal{Y}$, $\theta \in \Theta$ and $x \in \mathcal{X}$.

Assumption 3: Let parameter set Φ be a compact subset of $\mathbb{R}^d$, and assume there exists a quantization of Φ with quantization points in Θ such that Θ is an r-covering of Φ. Specifically, we assume there exists a set $\Theta \subset \Phi$ of finite cardinality M that is an r-covering of Φ, i.e., $\Phi \subset \cup_{\theta\in\Theta}\mathcal{B}_r(\theta)$, where $\mathcal{B}_r(\theta) :=$

$$\left\{\psi \in \Phi : \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{\mathsf{P}_X}\left[D_{KL}\left(f^{(i)}(\cdot|X,\theta)||f^{(i)}(\cdot|X,\psi)\right)\right] \le r\right\}$$

Assumption 4: For all agents $i \in [N]$, there exists an optimal $\theta^* \in \Phi$, such that $f^{(i)}(\cdot|X,\theta^*) = \mathsf{P}_{Y|X}(\cdot|X)$ almost surely under P_X.

These assumptions are natural. Assumption 1 states that one can always restrict attention to the connected components of the social network where the information gathered locally by the agents can be disseminated within the component. Assumption 2 prevents the degenerate case where a zero Bayesian prior prohibits learning. Assumption 3 assumes when θ is continuous it belongs to a compact set $\Phi \subset \mathbb{R}^d$. Assumption 4 assume the learning problem is realizable for all agents.

Theorem 1: Let $\mathcal{Q} = \mathcal{P}(\Theta)$ and $\epsilon > 0$. Under assumptions 1, 2, 3 and 4, with potentially non-IID local dataset as defined in Definition 2, using the decentralized learning algorithm in Section III, with probability at least $1-\delta$ we have

$$\max_{i\in[N]}\max_{\theta\in\Theta\setminus\mathcal{B}_r(\theta^*)} b_i^{(T)}(\theta) < e^{-T(K(\Theta)-\epsilon)}$$

when the number of communication rounds satisfies $T \ge \frac{8C\log\frac{N|\Theta|}{\delta}}{\epsilon^2(1-\lambda_{\max}(W))}$, where we define the rate of convergence of the posterior distribution as follows

$$K(\Theta) := \min_{\theta\in\Theta\setminus\mathcal{B}_r(\theta^*)}\sum_{j=1}^{N} v_j I_j\left(\theta^*,\theta\right), \tag{5}$$

and $I_j(\theta^*,\theta) := \mathbb{E}_{\mathsf{P}_X^{(j)}}[D_{\mathrm{KL}}(\mathsf{P}_{Y|X}(\cdot|\mathbf{X})||f^{(j)}(\cdot|\theta,\mathbf{X})) - D_{\mathrm{KL}}(\mathsf{P}_{Y|X}(\cdot|\mathbf{X})||f^{(j)}(\cdot|\theta^*,\mathbf{X}))]$, where eigenvector centrality $\mathbf{v} = [v_1, v_2, \ldots, v_N]$ is the unique stationary distribution of W with strictly positive components, furthermore define $\lambda_{\max}(W) := \max_{1\le i\le N-1}\lambda_i(W)$, where $\lambda_i(W)$ denotes i-th eigenvalue of W counted with algebraic multiplicity and $\lambda_0(W) = 1$, and $C := |\log\frac{L}{\alpha}|$.

Proof of the Theorem 1 is provided in the Appendix and relies on the following remark.

Remark 3: If $\mathcal{Q} = \mathcal{P}(\Theta)$, then update performed in equation (3) reduces to a Bayesian update with prior $q_i^{(t-1)}$ and likelihood function $f^{(i)}(\mathbf{Y}_t^{(i)}|\mathbf{X}_t^{(i)},\cdot)$, i.e., for $\theta\in\Theta$,

$$b_i^{(t)}(\theta) = \frac{f^{(i)}\left(\mathbf{Y}_t^{(i)}|\mathbf{X}_t^{(i)},\theta\right)q_i^{(t-1)}(\theta)}{\int_\Theta f^{(i)}\left(\mathbf{Y}_t^{(i)}|\mathbf{X}_t^{(i)},\phi\right)q_i^{(t-1)}(\phi)d\phi}. \tag{6}$$

Remark 4: Proof of Theorem 1 relies on Assumption 4, i.e., the realizable learning problems. We conjecture that this assumption can be relaxed to arrive at a decentralized variant of Bernstein-Von Mises Theorem. While, this extension remains an interesting future research direction, we investigate relaxing the assumption empirically in Section V where we consider the non-realizable case of decentralized training of a neural networks over an arbitrary network with non-IID data.

Remark 5: Theorem 1 indicates the posterior belief over the parameters θ outside of $\mathcal{B}_r(\theta^*)$ shrinks to zero exponentially fast, with exponent characterized by $K(\Theta)$. The rate of convergence characterized by (5) is a function of the agent's ability to distinguish between the parameters given by the KL-divergences and structure of the weighted network which is captured by the eigenvector centrality $\mathbf{v}$ of the agents. Hence, every agent influences the rate in two ways. Firstly, if the agent has higher eigenvector centrality (i.e., the agent is centrality located), it has larger influence over the posterior distributions of other agents as a result has a greater influence over the rate of exponential decay as well. Secondly, if the agent has high KL-divergence (i.e., highly informative local observations that can distinguish between parameters), then again it increases the rate. If an influential agent has highly informative observations then it boosts the rate of convergence. We will illustrate this through extensive simulations in Section V.

Corollary 1 (Average Expected Loss): Define $\hat{\theta}_i^{(t)} := \operatorname{argmax}_{\theta\in\Theta} b_i^{(t)}(\theta)$. Let $l(\hat{y},y)$ denote the loss function of predicting $\hat{y}$ given true label y. The expected loss of agent i under θ is given by $L_i(\theta) = \mathbb{E}_{\mathsf{P}_X}[\int_{\mathcal{Y}} l(\hat{y},y)f^{(i)}(\hat{y}|x,\theta)dy]$. Assume $|l(\hat{y},y)| \le B$ for any $\hat{y},y\in\mathcal{Y}$. Under assumptions 1, 2, 3 and 4, using the decentralized learning algorithm in Section III, with probability at least $1-\delta$ we have

$$\frac{1}{N}\sum_{i=1}^{N}|L_i(\theta^*) - L_i\left(\hat{\theta}_i^{(T)}\right)| \le \frac{B\sqrt{r}}{2}$$

When the number of communication rounds satisfies $T \ge \frac{8C\log\frac{N|\Theta|}{\delta}}{\epsilon^2(1-\lambda_{\max}(W))}$.

Proof of Corollary 1 is provided in the Appendix.

Corollary 2: Let $\Theta \subset \mathbb{R}^d$ be a finite set, i.e., $\Theta = \{\theta_1,\ldots,\theta_M\}$. Let $\mathcal{Q} = \mathcal{P}(\Theta)$ and $\epsilon > 0$. Under assumptions 1, 2, and assume for all agents $i\in[N]$, there exists an optimal $\theta^*\in\Theta$, such that $f^{(i)}(\cdot|X,\theta^*) = \mathsf{P}_{Y|X}(\cdot|X)$ almost surely under P_X. Then using the decentralized learning algorithm in Section III, with probability at least $1-\delta$ we have

$$\max_{i\in[N]}\max_{\theta\neq\theta^*} b_i^{(T)}(\theta) < e^{-T(K(\Theta)-\epsilon)}$$

when $T \ge \frac{8C\log\frac{NM}{\delta}}{\epsilon^2(1-\lambda_{\max}(W))}$, and the rate of convergence is: $K(\Theta) = \min_{\theta\neq\theta^*}\sum_{j=1}^{N} v_j I_j(\theta^*,\theta)$.

At the end of this section, we discuss a special case of our algorithm where in both variational training and consensus step, we restrict our attention to the family of Gaussian distributions. Let $\mathcal{Q}$ denote the family of Gaussian posterior distributions with pdf given by $G(\mu,\Sigma)$.

Here we first note that If $\mathcal{Q}\subsetneq\mathcal{P}(\Theta)$, equation (3), finds a distribution $\pi\in\mathcal{Q}$ which satisfies the following

$$\operatorname*{argmin}_{\pi\in\mathcal{Q}} D_{KL}\left(\pi\left|\left|\frac{f^{(i)}\left(\mathbf{Y}_t^{(i)}|\mathbf{X}_t^{(i)},\theta\right)q_i^{(t-1)}(\theta)}{\int_\Theta f^{(i)}\left(\mathbf{Y}_t^{(i)}|\mathbf{X}_t^{(i)},\phi\right)q_i^{(t-1)}(\phi)d\phi}\right.\right). \tag{7}$$

In other words, equation (7) projects the distribution obtained via equation (6) onto the allowed family of posterior distributions $\mathcal{Q}$ by employing KL-divergence minimization.

Lemma 1 (Posterior Merging With Gaussians): Let $(\mu_i^{(t)}, \Sigma_i^{(t)})$ denote the mean and the covariance matrix of $b_i^{(t)}$ at learner i. Then the posterior distribution $q_i^{(t)}$ obtained after the consensus step also belongs to $\mathcal{Q}$. Furthermore, the closed-form update of the mean and covariance matrix $(\widetilde{\mu}_i^{(t)}, \widetilde{\Sigma}_i^{(t)})$ of $q_i^{(t)}$ is given as follows

$$\left(\widetilde{\Sigma}_i^{(t)}\right)^{-1} = \sum_{j=1}^{N} W_{ij} \Sigma_j^{(t)^{-1}}, \quad \widetilde{\mu}_i^{(t)} = \widetilde{\Sigma}_i^{(t)} \sum_{j=1}^{N} W_{ij} \left(\Sigma_j^{(t)}\right)^{-1} \mu_j^{(t)}. \tag{8}$$

Derivation of equation (8) is provided in the Appendix.

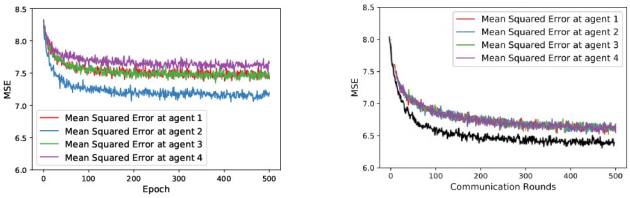
V. Experiments

We construct experiments in the decentralized Bayesian linear regression setup (Example 1) and decentralized classification setup (Example 2) over synthetic and real-world benchmark datasets. We demonstrate the performance of the proposed decentralized learning rule in all cases of non-IID data distributions mentioned in Definition 2. Furthermore, we discuss the effect of eigenvector centrality on the rate of convergence, compare our learning rule with FedAvg [10] over a federated dataset, and show how our method scales with larger number of agents in a time-varying network.

A. Decentralized Bayesian Linear Regression

To illustrate our approach, we construct an example of Bayesian linear regression (Example 1) in the realizable setting over the network with 4 agents. Let $\boldsymbol{\theta}^* = [-0.3, 0.5, 0.5, 0.1]^T$ and let noise be distributed as $\eta \sim \mathcal{N}(0, \alpha^2)$ where $\alpha = 0.8$. Agent i makes observations $(\mathbf{x}, y)$, where $\mathbf{x} = [0 \ldots, 0, x_i, 0, \ldots, 0]^T$ and x_i is sampled from Unif$[-1, 1]$ for $i = 1$, Unif$[-1.5, 1.5]$ for $i = 2$, Unif$[-1.25, 1.25]$ for $i = 3$, and Unif$[-0.75, 0.75]$ for $i = 4$. We assume each agent starts with a Gaussian prior over $\boldsymbol{\theta}$ with zero mean vector and covariance matrix given by diag$[0.5, 0.5, 0.5, 0.5]$, where diag$(\mathbf{x})$ denotes a diagonal matrix with diagonal elements given by vector $\mathbf{x}$. The social interaction weights are given as $\mathbf{W}_1 = [0.5, 0.5, 0, 0]$, $\mathbf{W}_2 = [0.3, 0.1, 0.3, 0.3]$, $\mathbf{W}_3 = [0, 0.5, 0.5, 0]$ and $\mathbf{W}_4 = [0, 0.5, 0, 0.5]$. Since each agent starts with a Gaussian prior over Θ, the posterior distribution after a Bayesian update remains Gaussian, which implies $\mathcal{Q}$ remains fixed as the family of Gaussian distributions and the consensus step reduces to equation (8).

We show that our proposed social learning framework enables a fully decentralized and fast learning of a global model even when the local data is severely deficient. More specifically, we assume that each agent makes observations along only one coordinate of $\mathbf{x}$ even though the global test set consists of observations belonging to any $\mathbf{x}$. Note that this is a case of extreme non-IID data partition across the agents and corresponds to case (2) of *non-IID* in Definition 2. Fig. 1(b) shows that the MSE of all agents, when trained using the decentralized learning rule, is much lower than training separately with no communication, and matches that of a central agent, implying that the agents converge to the true $\boldsymbol{\theta}^*$ as our theory predicts.



(a) Learning without cooperation (b) Learning with cooperation

Fig. 1. Mean Squared Error (MSE) of the predictions over a test dataset under two cases: (i) all agents, despite the severe deficiency of their observations, learn without cooperation using local training data only, and (ii) agents learn using the proposed decentralized learning rule. The black curve represents a benchmark scenario a central agent learns the model with access to all coordinates of the training data.

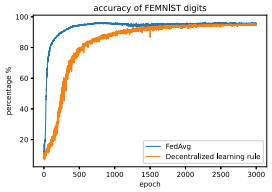
B. Decentralized Image Classification

To illustrate the performance of our learning rule on real datasets we consider the problem of decentralized training of Bayesian neural networks for an image classification task on the MNIST digits dataset [40], the Fashion-MNIST (FMNIST) dataset [41] and the Federated Extended-MNIST dataset from LEAF [42]. For all our experiments we consider multilayered Bayesian NN and employ Monte Carlo to obtain the predictions.

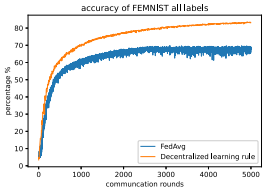
We divide the training dataset into non-overlapping subsets. Hence, agents must learn $\mathbf{b}_i^{(n)}$ such that the resulting predictive distribution can perform well over the global dataset without sharing the local data and hence not having seen input example associated with the labels that are missing locally.

1) Comparison to FedAvg on Federated Dataset: In this section, we compare the proposed decentralized update rule with Federated Averaging (FedAvg) [10] on a public federated dataset called Federated Extended-MNIST (FEMNIST) [42]. The Federated Extended-MNIST (FEMNIST) dataset is part of LEAF, a benchmark dataset for federated learning [42]. The dataset has handwritten digits and lower/upper case alphabets by different writers, therefore there is a total of 62 labels, namely $[0, \ldots, 9]$ and $[a, \ldots, z, A, \ldots, Z]$. The dataset has a total of 3550 writers with an average of 226 images per writer, where the images are of size 28 by 28. In the experiment setup, each learner only sees digits/alphabets from a single writer, therefore there is significant *feature skew* (case (2) of *non-IID* in Definition 2).

For this experiment we use a multi-layer Convolutional Neural Network. The input image is first passed through a convolution layer with 6 channels, size 5 by 5, followed by a 2DMaxpool of size 2 by 2 and a ReLU. Then the output goes through another convolution layer with 16 channels, size 5 by 5, followed by a 2DMaxpool of size 2 by 2 and a ReLU. Then the output is forwarded to 2 linear layers with 784 and 120 hidden units each. We choose $\mathcal{Q}$ to be the family of Gaussian mean-field approximate posterior distributions with pdf given

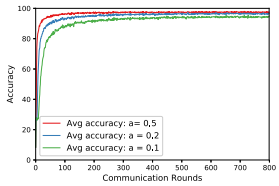


(a) On digits only

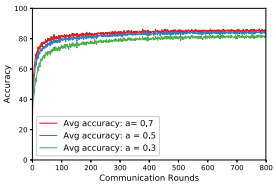


(b) On digits and upper/lower case alphabets

Fig. 2. Figure shows average accuracies over all nodes of proposed decentralized learning algorithm and FedAvg on FEMNIST dataset.



(a) On MNIST dataset



(b) On FMNIST dataset

Fig. 3. Figure shows the variation in the average accuracy over a star network topology as the eigenvector centrality of the central agent is varied.

by $G(\boldsymbol{\theta}, \boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a strictly diagonal matrix [11], [12]. As discussed in Remark. 3 this corresponds to performing variational inference to obtain a Gaussian approximation of the local posterior distribution, i.e., minimizing the variational free energy given in equation (3) over $\mathcal{Q}$. While we compute the KL divergence in (3) in a closed form, we employ simple Monte Carlo to compute the gradients using Bayes by Backprop [12], [14].

To emulate the federated learning setup, where the beliefs of all learners are merged with equal weight to form a public belief, we use a densely connected network with $W_{ij} = 1/N$ for all $i, j \in [N]$. We note that FedAvg can also be viewed as a decentralized SGD method over a fully connected network. We use a total of 50 learners, chosen randomly from the 3550 available writers. We use local training batch size $B = 10$, local epochs $E = 1$ and a total of 3000 communication rounds for digits only and 5000 communication rounds for digits and alphabets. For all agents, we use Adam optimizer [43] with initial learning rate of 0.001 and learning rate decay of 0.99 per communication round.

On only digits, our method is on par with FedAvg, with accuracy $\sim 96\%$. On both digits and alphabets, our method significantly outperforms FedAvg with accuracy over 83% after 5000 communication rounds, comparing to $\sim 69\%$ from FedAvg. We note that classifying both digits/alphabets is significantly harder than just digits, and by putting a prior on the latent parameters our method encodes uncertainty in the local estimates of the true parameter and thus prevents premature convergence to local data, as is suffered by FedAvg.

2) Effect of Eigenvector Centrality: In this section, we investigate how the eigenvector centrality of an agent affect the rate of convergence to the true parameter. We examine this on a network with a star topology, where a central agent is connected to 8 other edge agents. Let the social interaction weights for the central agent be $\mathbf{W}_1 = [1/9, \ldots, 1/9]$. For $a \in (0, 1)$, we assume that an edge agent i puts a confidence $\mathbf{W}_{i1} = a$ on the central agent, $\mathbf{W}_{ii} = 1 - a$ on itself and zero on others. Note that as the confidence a which the edge agents put on the central agent increases, the eigenvector centrality of the central agent v_1 increases, i.e., central agent becomes more influential over the network.

We consider two datasets: (i) the MNIST digits dataset [40] where each image is assigned a label in $\mathcal{Y} = [0, \ldots, 9]$ and (ii) the Fashion-MNIST (FMNIST) dataset [41] where each image is assigned a label in $\mathcal{Y} =$ [`t-shirt`, `trouser`, `pullover`, `dress`, `coat`, `sandal`, `shirt`, `sneaker`, `bag`, `ankle-boot`]. Both datasets consist of 60,000 training images and 10,000 testing images of size 28 by 28. For all our experiments we consider a fully connected NN with 2-hidden layers with 200 units each using ReLU activations which is same as the architecture considered in the context of federated learning in [10]. Again as in Section V-B1, we choose $\mathcal{Q}$ to be the family of Gaussian mean-field approximate posterior distributions.

We partition the dataset such that the central agent has more informative local observations. Specifically, the central agent sees more labels and also has the largest number of samples, which corresponds to *label skew* (case (1)) and *quantity skew* (case (4)) of *non-IID* data distribution in Definition 2.

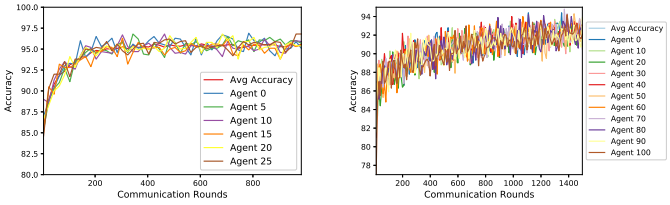
We vary confidence a which the edge agents put on the central agent over $[0.1, 0.2, 0.3, 0.5, 0.7]$, and therefore the eigenvector centrality of the central agent v_1 increases as $[0.1, 0.18, 0.25, 0.36, 0.44]$. We partition the MNIST dataset into two subsets so that the central agent dataset has all images of labels $[2, \ldots, 9]$ and edge agents has all images of labels $[0, 1]$. To ensure all the edge agents has equal number of images, we shuffle the images with labels $[0, 1]$ and partition them into 8 non-overlapping subsets. Similarly, for Fashion-MNIST (FMNIST) dataset, we first partition into two subsets so that central agent has access to labels [`t-shirt`, `pullover`, `dress`, `coat`, `shirt`, `bag`] and edge agents have access to labels [`trouser`, `sandal`, `sneaker`, `ankle-boot`]. We shuffle the images with labels [`trouser`, `sandal`, `sneaker`, `ankle-boot`] and partition them into 8 non-overlapping subsets.

We ensure that all agents has same number of local updates u per communication round, which is equal to $(\lfloor * \rfloor n_{edge}/B)E$, here n_{edge} denotes the number of training samples for each edge node. For the central agent, this means that for each local epoch, the central agent is trained on a random subset of its local dataset, whereas the edge agents use all the local dataset. For all agents, we use Adam optimizer [43] with initial learning rate of 0.001 and learning rate decay of 0.99 per communication round.

From equation (5) we know that placing more confidence a on the central agent increases the rate of convergence to the true parameter and increases rate of convergence of the test dataset accuracy. This is demonstrated in Fig. 3(a) and Fig. 3(b) where both accuracy and the rate of convergence improve as a increases. In other words, rate of convergence and the average accuracy is the highest when the agent with

TABLE I
SETTINGS FOR STAR TOPOLOGY NETWORK EXPERIMENT: E IS NUMBER OF LOCAL EPOCHS, B IS THE LOCAL MINIBATCH SIZE, u IS THE NUMBER OF LOCAL UPDATES PER COMMUNICATION ROUND, η IS THE INITIAL LEARNING RATE FOR ALL AGENTS, ϵ IS THE LEARNING RATE DECAY RATE, n_{center} IS THE DATASET SIZE OF THE CENTRAL AGENT, n_{edge} IS THE DATASET SIZE OF EACH OF THE EDGE AGENT

Experiment	E	B	u	η	ϵ	n_{center}	n_{edge}	comm rounds
MNIST	5	50	155	0.001	0.99	47335	1583	800
FMNIST	5	100	150	0.001	0.99	36000	3000	800



(a) Average accuracy over all 26 nodes. (b) Accuracy over all 100 nodes.

Fig. 4. Figure shows the accuracies of selected agents in a time-varying network.

most informative local observations has most influence on the network. Note that assigning too much confidence on the most informative agents can potentially hinder other nodes from learning from their local dataset.

3) Asynchronous Decentralized Learning on Time-Varying Networks: Now we implement our learning rule on time-varying networks which model practical peer-to-peer networks where synchronous updates are not easy or very costly to implement. We consider a time-varying network of $N+1$ agents numbered as $\{0, 1, \ldots, N\}$. At any give time, only N_0 agents are connected to agent 0 in a star topology. For $k \in [N/N_0]$, let $\mathcal{G}_k$ denote a graph with a star topology where the central agent 0 is connected to edge agents whose indices belong to $\{N_0(k-1)+1, \ldots, N_0 k\}$. This implies at any given time only a small fraction of agents N_0/N are training over their local data. Note that $\cup_{k=1}^{N/N_0} \mathcal{G}_k$ is strongly connected network over all $N+1$ agents. The social interaction weights for the central agent are $\mathbf{W}_0 = [1/N_0+1, \ldots, 1/N_0+1]$. Let $a = 0.5$. An edge agent $i \in \mathcal{G}_k$ puts a confidence $\mathbf{W}_{i0} = a$ on the central agent 0, $\mathbf{W}_{ii} = 1-a$ on itself and zero on others. The MNIST dataset is divided in an i.i.d manner, i.e., data is shuffled and each agent is randomly assigned approximately $(60{,}000/N+1)$ samples. As demonstrated in Fig. 4, for $N = 25, N_0 = 5$, we obtain an average accuracy of 96.5% over all agents and 95.1% accuracy at the central agent and for $N = 100, N_0 = 10$, we obtain an average accuracy of 92.3% over all agents and 93.1% accuracy at the central agent. This also demonstrates that decentralized learning can be achieved with as few as 600 samples locally.

VI. CONCLUSION AND FUTURE WORK

This paper considers the problem of peer-to-peer decentralized variational learning over an arbitrary social network. Building on prior work on distributed hypothesis testing [1] and variational Bayesian inference [12]–[14], we propose a fully decentralized variational learning algorithm consisting of two main components of variational updating based on local data and log-posterior averaging. In the realizable setting, we show this method to converge to the correct global labeling function and provide an analytical and closed-form characterization of the rate of convergence. Empirically, we validate our theoretical finding, illustrate the advantages over existing methods such as federated averaging when the data is non-IID across the agents. We also illustrate how the choice of the underlying social network impacts the rate of convergence.

TABLE II
SETTINGS FOR TIME-VARYING NETWORK EXPERIMENT: E IS NUMBER OF LOCAL EPOCHS, B IS THE LOCAL MINIBATCH SIZE, u IS THE NUMBER OF LOCAL UPDATES PER COMMUNICATION ROUND, η IS THE INITIAL LEARNING RATE FOR ALL AGENTS, ϵ IS THE LEARNING RATE DECAY RATE, n IS THE DATASET SIZE OF ANY AGENT. SINCE ALL AGENTS HAVE SAME NUMBER OF SAMPLES, THEY AUTOMATICALLY HAVE EQUAL NUMBER OF LOCAL UPDATES PER COMMUNICATION ROUND. ADAM OPTIMIZER IS USED FOR ALL AGENTS

Experiment	E	B	u	η	ϵ	n	comm rounds
N = 25	1	50	47	0.001	0.99	2307	1000
N = 100	2	10	120	0.001	0.998	594	1000

APPENDIX

A. Bernstein-Von Mises Theorem Under Model Misspecification

Theorem 2: Let $q^{(t)}(\theta|\mathcal{D}) \in \mathcal{P}(\Theta)$ be the posterior distribution after observing $\mathcal{D}_{1:t} = \{(\mathbf{X}_s, \mathbf{Y}_s)\}_{1\le s\le t} = \{(X_s^{(i)}, Y_s^{(i)}) \in (\mathcal{X}, \mathcal{Y})\}_{i\in[n], 1\le s\le t}$, where $\mathcal{D}_{1:t}$ is drawn i.i.d. from $\mathsf{P}_{XY} = \mathsf{P}_X \mathsf{P}_{Y|X}$. Let $p_\theta \in \mathcal{P}(\Theta)$ be the prior distribution over Θ. The Bernstein-Von Mises Theorem asserts that, for any measurable set $A \subset \Theta$, we have,

$$\sup_A |q^{(t)}(A) - N_{\hat{\theta}_t, (tV_{\theta^*})^{-1}}(A)| \xrightarrow{\mathsf{P}_{XY}} 0$$

where

$$\theta^* \in \operatorname*{argmin}_{\theta\in\Theta} \mathbb{E}_{\mathsf{P}_X}\left[D_{KL}\left(\mathsf{P}_{Y|X}(\cdot|X)||f(\cdot|X,\theta)\right)\right]$$

and V_{θ^*} is the negative Hessian matrix of the above expected KL divergence, and $\hat{\theta}_t$ is some suitable estimators, typically taken as the maximum likelihood estimator that satisfies the sequence $\sqrt{t}(\hat{\theta}_t - \theta^*)$ is asymptotically normal with zero mean. Here, it is not required that there exists a $\theta^* \in \Theta$ such that $f(\cdot|X,\theta^*) = \mathsf{P}_{Y|X}(\cdot|X)$ almost surely under P_X. In other words, Bernstein-Von Mises Theorem under model misspecification guarantees convergence of the posterior regardless of realizability.

B. Consensus Step on Gaussian Distributions

Let $(\boldsymbol{\mu}_i^{(t)}, \boldsymbol{\Sigma}_i^{(t)})$ denote the mean and the covariance matrix of $\mathbf{b}_i^{(t)}$ at agent i at time t. Using equation (4), we have

$$\sum_{j=1}^{N} W_{ij} \ln G\left(\boldsymbol{\theta}, \boldsymbol{\mu}_j^{(t)}, \boldsymbol{\Sigma}_j^{(t)}\right)$$

$$= -\frac{1}{2}\sum_{j=1}^{N} W_{ij}\left((\boldsymbol{\theta} - \boldsymbol{\mu}_j^{(t)})^{\mathrm{T}} {\boldsymbol{\Sigma}_j^{(t)}}^{-1} (\boldsymbol{\theta} - \boldsymbol{\mu}_j^{(t)})\right)$$

$$
\begin{aligned}
&- \frac{1}{2}\sum_{i=1}^{N} W_{ij}\ln(2\pi)^k\left|\boldsymbol{\Sigma}_j^{(t)}\right| \\
&= -\frac{1}{2}\left(\boldsymbol{\theta}^T\sum_{j=1}^{N} W_{ij}\boldsymbol{\Sigma}_j^{(t)^{-1}}\boldsymbol{\theta} + \sum_{j=1}^{N}\boldsymbol{\mu}_j^{(t)^T} W_{ij}\boldsymbol{\Sigma}_j^{(t)^{-1}}\boldsymbol{\mu}_j^{(t)}\right) \\
&+ \frac{1}{2}\left(\sum_{j=1}^{N}\boldsymbol{\mu}_j^{(t)^T} W_{ij}\boldsymbol{\Sigma}_j^{(t)^{-1}}\boldsymbol{\theta} + \boldsymbol{\theta}^T\sum_{j=1}^{N} W_{ij}\boldsymbol{\Sigma}_j^{(t)^{-1}}\boldsymbol{\mu}_i^{(t)}\right) \\
&-\frac{1}{2}\sum_{j=1}^{N} W_{ij}\ln(2\pi)^k\left|\boldsymbol{\Sigma}_j^{(t)}\right|.
\end{aligned}
$$

Noting that $(\boldsymbol{\theta}-\mathbf{A})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{\theta}-\mathbf{A}) = \boldsymbol{\theta}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta} - \mathbf{A}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\theta} - \boldsymbol{\theta}^T\boldsymbol{\Sigma}^{-1}\mathbf{A}+\mathbf{A}^T\boldsymbol{\Sigma}^{-1}\mathbf{A}$. By completing the squares we obtain $\mathbf{q}_i^{(t)}$ is Gaussian distribution with the last term becoming part of the normalization constant after exponentiating. and we have

$$
\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{i}}^{(t)^{-1}} = \sum_{j=1}^{N} W_{ij}\boldsymbol{\Sigma}_j^{(t)^{-1}},
$$

and

$$
\begin{aligned}
\widetilde{\boldsymbol{\Sigma}}_{\boldsymbol{i}}^{(t)^{-1}}\widetilde{\boldsymbol{\mu}}_i^{(t)} &= \sum_{j=1}^{N} W_{ij}\boldsymbol{\Sigma}^{(t)}{}_j^{-1}\boldsymbol{\mu}_j^{(t)} \\
\implies \widetilde{\boldsymbol{\mu}}_i^{(t)} &= \widetilde{\boldsymbol{\Sigma}}_i^{(t)}\sum_{j=1}^{N} W_{ij}\boldsymbol{\Sigma}_j^{(t)^{-1}}\boldsymbol{\mu}_j^{(t)}.
\end{aligned}
$$

C. Proof of Theorem 1

Before proving the theorem we first present a lemma on the stochastic matrix W.

Lemma 2 [2]: For an irreducible and aperiodic stochastic matrix W, the stationary distribution $\mathbf{v} = [v_1, v_2, \ldots, v_N]$ is unique and has strictly positive components and satisfies $v_i = \sum_{j=1}^{n} v_j W_{ji}$. Furthermore, for any $i \in [N]$ the weight matrix satisfies

$$
\sum_{k=1}^{n}\sum_{j=1}^{N}\left|[W^k]_{ij} - v_j\right| \le \frac{4\log N}{1-\lambda_{\max}(W)},
$$

where $\lambda_{\max}(W) = \max_{i\in[N-1]}\lambda_i(W)$, and $\lambda_i(W)$ denotes eigenvalue of W counted with algebraic multiplicity and $\lambda_0(W) = 1$.

The proof of Theorem 1 is based the proof provided in [2]–[4]. For the ease of exposition, let $b_i^{(0)}(\theta) = \frac{1}{|\Theta|}$ for all $\theta \in \Theta$. We begin with the following recursion for each node $i \in [N]$ and for any $\theta \in \Theta \setminus \mathcal{B}_r(\theta^*)$,

$$
\frac{1}{T}\log\frac{b_i^{(T)}(\theta^*)}{b_i^{(T)}(\theta)} = \frac{1}{T}\sum_{j=1}^{N}\sum_{t=0}^{T}\left[W^t\right]_{ij} z_j^{(T-t)}\left(\theta^*,\theta\right),
$$

where

$$
z_j^{(t)}\left(\theta^*,\theta\right) = \log\frac{f^{(j)}\left(\mathbf{Y}_t^{(j)}|\mathbf{X}_t^{(j)},\theta^*\right)}{f^{(j)}\left(\mathbf{Y}_t^{(j)}|\mathbf{X}_t^{(j)},\theta\right)}.
$$

From the above recursion we have

$$
\begin{aligned}
\frac{1}{T}\log\frac{b_i^{(T)}(\theta^*)}{b_i^{(T)}(\theta)} &= \frac{1}{T}\sum_{j=1}^{N} v_j\left(\sum_{t=0}^{T} z_j^{(T-t)}\left(\theta^*,\theta\right)\right) \\
&\quad + \frac{1}{T}\sum_{j=1}^{N}\left(\sum_{t=0}^{T}\left([W^t]_{ij} - v_j\right) z_j^{(T-t)}(\theta^*,\theta)\right) \\
&\ge \frac{1}{T}\sum_{j=1}^{N} v_j\left(\sum_{t=0}^{T} z_j^{(t)}(\theta^*,\theta)\right) \\
&\quad - \frac{1}{T}\sum_{j=1}^{N}\sum_{t=0}^{T}\left|[W^t]_{ij} - v_j\right|\left|z_j^{(t)}(\theta^*,\theta)\right| \\
&\overset{(a)}{\ge} \frac{1}{T}\sum_{j=1}^{N} v_j\left(\sum_{t=0}^{T} z_j^{(t)}\left(\theta^*,\theta\right)\right) - \frac{4C\log N}{T(1-\lambda_{\max}(W))}
\end{aligned}
$$

where (a) follows from Lemma 2 and the boundedness assumption of log-likelihood ratios. Now fix $T \ge \frac{8C\log N}{\epsilon(1-\lambda_{\max}(W))}$, since $b_i^{(T)}(\theta^*) \le 1$ (since $b_i^{(T)}(\cdot)$ is a pmf over Θ) we have

$$
-\frac{1}{T}\log b_i^{(T)}(\theta) \ge -\frac{\epsilon}{2} + \frac{1}{T}\sum_{j=1}^{N} v_j\left(\sum_{t=0}^{T} z_j^{(t)}\left(\theta^*,\theta\right)\right).
$$

Furthermore, we have

$$
\begin{aligned}
&\mathsf{P}\left(-\frac{1}{T}\log b_i^{(T)}(\theta) \le \sum_{j=1}^{N} v_j I_j(\theta^*,\theta) - \epsilon\right) \\
&\le \mathsf{P}\left(\frac{1}{T}\sum_{j=1}^{N} v_j\sum_{t=0}^{T} z_j^{(t)}\left(\theta^*,\theta\right) \le \sum_{j=1}^{N} v_j I_j\left(\theta^*,\theta\right) - \frac{\epsilon}{2}\right)
\end{aligned}
$$

where recall that

$$
\begin{aligned}
I_j\left(\theta^*,\theta\right) &= \mathbb{E}\left[z_j\left(\theta^*,\theta\right)\right] \\
&= \mathbb{E}_{\mathsf{P}_X^{(j)}}\Big[D_{\mathrm{KL}}\Big(\mathsf{P}_{Y|X}(\cdot|\mathbf{X})||f^{(j)}(\cdot|\theta,\mathbf{X})\Big) \\
&\qquad - D_{\mathrm{KL}}\Big(\mathsf{P}_{Y|X}(\cdot|\mathbf{X})||f^{(j)}\left(\cdot|\theta^*,\mathbf{X}\right)\Big)\Big].
\end{aligned}
$$

Now for any $j \in [N]$ note that

$$
\begin{aligned}
&\sum_{j=1}^{N} v_j\sum_{t=0}^{T} z_j^{(t)}\left(\theta^*,\theta\right) - T\sum_{j=1}^{N} v_j I_j\left(\theta^*,\theta\right) \\
&= \sum_{t=0}^{T}\left(\sum_{j=1}^{N} v_j z_j^{(t)}\left(\theta^*,\theta\right) - \sum_{j=1}^{N} v_j\mathbb{E}\left[z_j^{(t)}\left(\theta^*,\theta\right)\right]\right).
\end{aligned}
$$

For any $\theta \notin \mathcal{B}_r(\theta^*)$, applying McDiarmid's inequality for all $\epsilon > 0$ and for all $T \ge 1$ we have

$$
\begin{aligned}
&\mathsf{P}\left(\sum_{t=0}^{T}\left(\sum_{j=1}^{N} v_j z_j^{(t)}\left(\theta^*,\theta\right) - \sum_{j=1}^{N} v_j\mathbb{E}\left[z_j^{(t)}\left(\theta^*,\theta\right)\right]\right) \le -\frac{\epsilon T}{2}\right) \\
&\le e^{-\frac{\epsilon^2 T}{2C}}.
\end{aligned}
$$

Hence, for all $\theta \notin \mathcal{B}_r(\theta^*)$, for $T \ge \frac{8C\log N}{\epsilon(1-\lambda_{\max}(W))}$ we have

$$
\mathsf{P}\left(\frac{-1}{T}\log b_i^{(T)}(\theta) \le \sum_{j=1}^{N} v_j I_j\left(\theta^*,\theta\right) - \epsilon\right) \le e^{-\frac{\epsilon^2 T}{4C}},
$$

which implies

$$\mathsf{P}\left(b_i^{(T)}(\theta) \geq e^{-T\left(\sum_{j=1}^N v_j I_j(\theta^*,\theta)-\epsilon\right)}\right) \leq e^{-\frac{\epsilon^2 T}{4C}}.$$

Using this we obtain a bound on the worst case error over all θ and across the entire network as follows

$$\mathsf{P}\left(\max_{i\in[N]}\max_{\theta\in\Theta\setminus\mathcal{B}_r(\theta^*)} b_i^{(T)}(\theta) \geq e^{-T(K(\Theta)-\epsilon)}\right) \leq N|\Theta|e^{-\frac{\epsilon^2 T}{4C}},$$

where $K(\Theta) := \min_{\theta\in\Theta\setminus\mathcal{B}_r(\theta^*)} \sum_{j=1}^N v_j I_j(\theta^*,\theta)$. From Lemma 2 we have that $K(\Theta) > 0$. Then, with probability at least $1-\delta$ we have

$$\max_{i\in[N]}\max_{\theta\in\Theta\setminus\mathcal{B}_r(\theta^*)} b_i^{(T)}(\theta) < e^{-T(K(\Theta)-\epsilon)},$$

when the number of samples satisfies

$$T \geq \frac{4C\log\frac{N|\Theta|}{\delta}}{\epsilon^2}.$$

However, recall we require $T \geq \frac{8C\log N}{\epsilon(1-\lambda_{\max}(W))}$, a reasonable lower bound is therefore

$$T \geq \frac{8C\log\frac{N|\Theta|}{\delta}}{\epsilon^2(1-\lambda_{\max}(W))}.$$

D. Proof of Corollary 1

$$\begin{aligned}
&\frac{1}{N}\sum_{i=1}^N \left|L_i(\theta^*) - L_i\left(\hat{\theta}_i^{(T)}\right)\right| \\
&\leq \frac{B}{N}\sum_{i=1}^N \mathbb{E}_{\mathsf{P}_X}\left[\int_{\mathcal{Y}}\left|f^{(i)}(\hat{y}|x,\theta^*) - f^{(i)}(\hat{y}|x,\hat{\theta}_i^{(T)})\right|dy\right] \\
&\leq \frac{B}{2N}\sum_{i=1}^N \mathbb{E}_{\mathsf{P}_X}\left[\sqrt{D_{KL}\left(f^{(i)}(\hat{y}|x,\theta^*)||f^{(i)}\left(\hat{y}|x,\hat{\theta}_i^{(T)}\right)\right)}\right] \\
&\leq \frac{B}{2}\sqrt{\frac{1}{N}\sum_{i=1}^N \mathbb{E}_{\mathsf{P}_X}\left[D_{KL}\left(f^{(i)}(\hat{y}|x,\theta^*)||f^{(i)}\left(\hat{y}|x,\hat{\theta}_i^{(T)}\right)\right)\right]} \\
&\leq \frac{B\sqrt{r}}{2}
\end{aligned}$$

where the third to last inequality follows from Pinsker's Inequality, the second to last inequality follows from Jensen's Inequality, and the last line follows from Theorem 1 and Assumption 3.

REFERENCES

[1] A. Lalitha, A. D. Sarwate, and T. Javidi, "Social learning and distributed hypothesis testing," in *Proc. IEEE Int. Symp. Inf. Theory*, 2014, pp. 551–555.

[2] S. Shahrampour, A. Rakhlin, and A. Jadbabaie, "Distributed detection: Finite-time analysis and impact of network topology," *IEEE Trans. Autom. Control*, vol. 61, no. 11, pp. 3256–3268, Nov. 2016.

[3] A. Nedić, A. Olshevsky, and C. A. Uribe, "Fast convergence rates for distributed non-Bayesian learning," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5538–5553, Nov. 2017.

[4] A. Lalitha, T. Javidi, and A. D. Sarwate, "Social learning and distributed hypothesis testing," *IEEE Trans. Inf. Theory*, vol. 64, no. 9, pp. 6161–6179, Sep. 2018.

[5] A. Lalitha, S. Shekhar, T. Javidi, and F. Koushanfar, "Fully decentralized federated learning," in *Proc. NeurIPS Workshop Bayesian Deep Learn.*, 2018, pp. 1–9.

[6] A. Lalitha, O. C. Kilinc, T. Javidi, and F. Koushanfar, "Peer-to-peer federated learning on graphs," 2019, *arXiv:1901.11173*.

[7] M. Al-Shedivat, J. Gillenwater, E. Xing, and A. Rostamizadeh, "Federated learning via posterior averaging: A new perspective and practical algorithms," 2021, *arXiv:2010.05273*.

[8] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016, *arXiv:1610.02527*.

[9] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon, "Federated learning: Strategies for improving communication efficiency," 2016, *arXiv:1610.05492*.

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

[11] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner, "Variational continual learning," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–27.

[12] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight uncertainty in neural networks," in *Proc. 32nd Int. Conf. Mach. Learn. Vol. 37*, 2015, pp. 1613–1622. [Online]. Available: JMLR.org

[13] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan, "Streaming variational bayes," in *Advances in Neural Information Processing Systems*, vol. 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2013, pp. 1727–1735.

[14] D. P. Kingma, T. Salimans, and M. Welling, "Variational dropout and the local reparameterization trick," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2015, pp. 2575–2583.

[15] V. Smith, S. Forte, C. Ma, M. Takáč, M. I. Jordan, and M. Jaggi, "CoCoA: A general framework for communication-efficient distributed optimization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 8590–8638, 2018.

[16] P. Chaudhari, C. Baldassi, R. Zecchina, S. Soatto, A. Talwalkar, and A. Oberman, "Parle: Parallelizing stochastic gradient descent," 2017, *arXiv:1707.00424*.

[17] T. Lin, S. U. Stich, K. K. Patel, and M. Jaggi, "Don't use large mini-batches, use local SGD," 2018, *arXiv:1808.07217*.

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

[19] E. Wei and A. Ozdaglar, "Distributed alternating direction method of multipliers," in *Proc. IEEE 51st IEEE Conf. Decis. Control (CDC)*, Dec. 2012, pp. 5445–5450.

[20] H. Tang, X. Lian, M. Yan, C. Zhang, and J. Liu, "d^2: Decentralized training over decentralized data," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80. Stockholm, Sweden, Jul. 2018, pp. 4848–4856.

[21] L. Bottou, F. Curtis, and J. Nocedal, "Optimization methods for large-scale machine learning," *SIAM Rev.*, vol. 60, no. 2, pp. 223–311, 2018.

[22] X. Lian, C. Zhang, H. Zhang, C.-J. Hsieh, W. Zhang, and J. Liu, "Can decentralized algorithms outperform centralized algorithms? a case study for decentralized parallel stochastic gradient descent," in *Advances in Neural Information Processing Systems 30*, I. Guyon *et al.*, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2017, pp. 5330–5340.

[23] Z. Jiang, A. Balu, C. Hegde, and S. Sarkar, "Collaborative deep learning in fixed topology networks," in *Advances in Neural Information Processing Systems*, vol. 30, I. Guyon *et al.*, Eds. Red Hook, NY, USA: Curran Assoc., Inc., 2017, pp. 5904–5914.

[24] P. H. Jin, Q. Yuan, F. N. Iandola, and K. Keutzer, "How to scale distributed deep learning?" 2016, *arXiv:1611.04581*.

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

[26] X. Lian, W. Zhang, C. Zhang, and J. Liu, "Asynchronous decentralized parallel stochastic gradient descent," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 3043–3052.

[27] A. Nedić and A. Olshevsky, "Stochastic gradient-push for strongly convex functions on time-varying directed graphs," *IEEE Trans. Autom. Control*, vol. 61, no. 12, pp. 3936–3947, Dec. 2016.

[28] M. Assran, N. Loizou, N. Ballas, and M. Rabbat, "Stochastic gradient push for distributed deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 344–353.

[29] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2508–2530, Jun. 2006.
[30] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control Lett.*, vol. 53, no. 1, pp. 65–78, 2004.
[31] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," in *Proc. 44th Annu. IEEE Symp. Found. Comput. Sci.*, 2003, pp. 482–491.
[32] T. Campbell and J. P. How. "Decentralized variational Bayesian inference," 2014, *arXiv:1403.7471v1*.
[33] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics). Berlin, Germany: Springer-Verlag, 2006.
[34] Y. Gal, *Uncertainty in Deep Learning*, Univ. Cambridge, Cambridge, U.K., 2016.
[35] B. J. K. Kleijn and A. W. van der Vaart, "The Bernstein-Von-Mises theorem under misspecification," *Electron. J. Stat.*, vol. 6, pp. 354–381, Mar. 2012.
[36] P. Kairouz *et al.*, "Advances and open problems in federated learning," 2021, *arXiv:1912.04977*.
[37] Q. Li, Y. Diao, Q. Chen, and B. He, "Federated learning on non-IID data silos: An experimental study," 2021, *arXiv:2102.02079*.
[38] H. Zhu, J. Xu, S. Liu, and Y. Jin, "Federated learning on non-IID data: A survey," *Neurocomputing*, vol. 465, pp. 371–390, Nov. 2021.
[39] K. P. Murphy, *Machine Learning: A Probabilistic Perspective*. Cambridge, MA, USA: MIT Press, 2012.
[40] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, Nov. 1998.
[41] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.
[42] S. Caldas *et al.*, "LEAF: A benchmark for federated settings," 2019, *arXiv:1812.01097*.
[43] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2015, *arXiv:1412.6980*.

Xinghan Wang received the B.S. and M.S. degrees in electrical and systems engineering from Washington University in St. Louis in 2018. He is currently pursuing the Ph.D. degree with the University of California at San Diego, where he works on federated learning, multiarmed bandits, and reinforcement learning.

Anusha Lalitha received the B.Tech. degree in electrical engineering from IIT Gandhinagar, India, in 2012, and the M.S. and Ph.D. degrees in electrical engineering from the University of California at San Diego in 2015 and 2019, respectively, where she broadly worked in information theory and statistics. She is an Applied Scientist II for AI Vertical Services with AWS AI Labs, where she works on explainable personalization and recommendations. Before that, she worked with Amazon Alexa on automatic speech recognition. Before joining Amazon, she was a Postdoctoral Researcher with the Department of Electrical Engineering, Stanford University, where she worked on decentralized learning.

Tara Javidi (Fellow, IEEE) received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, the M.S. degree in electrical engineering (systems) and in applied mathematics (stochastic analysis) from the University of Michigan at Ann Arbor, Ann Arbor, and the Ph.D. degree in electrical engineering and computer science from the University of Michigan at Ann Arbor in 2002. She is a Jacobs Family Scholar, a Halicioglu Data Science Fellow, and a Professor of Electrical and Computer Engineering with the University of California at San Diego, San Diego. She is the Editor-in-Chief of IEEE JOURNAL ON SELECTED AREAS IN INFORMATION THEORY in 2022–2024.

Farinaz Koushanfar (Fellow, IEEE) received the B.S. degree in electrical engineering from the Sharif University of Technology, Tehran, Iran, and the M.A. degree in statistics and machine learning and the Ph.D. degree in electrical engineering and computer science from the University of California at Berkeley in 2005. She is a Henry Booker Faculty Scholar Professor of Electrical and Computer Engineering with the University of California at San Diego.