Communication-Efficient and Privacy-Aware Distributed Learning

Vinay Chakravarthi Gogineni, *Senior Member, IEEE*, Ashkan Moradi, *Member, IEEE*, Naveen K. D. Venkategowda, *Senior Member, IEEE*, Stefan Werner, *Fellow, IEEE*

***Abstract*—Communication efficiency and privacy are two key concerns in modern distributed computing systems. Towards this goal, this paper proposes partial sharing private distributed learning (PPDL) algorithms that offer communication efficiency while preserving privacy, thus making them suitable for applications with limited resources in adversarial environments. First, we propose a noise injection-based PPDL algorithm that achieves communication efficiency by sharing only a fraction of the information at each consensus iteration and provides privacy by perturbing the information exchanged among neighbors. To further increase privacy, local information is randomly decomposed into private and public substates before sharing with the neighbors. This results in a decomposition- and noise-injection-based PPDL strategy in which only a fraction of the perturbed public substate is shared during local collaborations, whereas the private substate is updated locally without being shared. To determine the impact of communication savings and privacy preservation on the performance of distributed learning algorithms, we analyze the mean and mean-square convergence of the proposed algorithms. Moreover, we investigate the privacy of agents by characterizing privacy as the mean squared error of the estimate of private information at the honest-but-curious adversary. The analytical results show a tradeoff between communication efficiency and privacy in proposed PPDL algorithms, while decomposition- and noise-injection-based PPDL improves privacy compared to noise-injection-based PPDL. Lastly, numerical simulations corroborate the analytical findings.**

***Index Terms*—Average consensus, communication efficiency, distributed learning, multiagent systems, privacy-preservation.**

I. Introduction

In distributed multiagent networks, local interactions among neighboring agents contribute to the overall network performance. For instance, in systems such as internet-of-things (IoT) and networked-control systems, agents engage in local learning and subsequent communication with neighbors to enhance collective learning performance and network robustness against dynamic changes [1]–[3]. The agents are often associated with limited computing and energy resources. Although local interactions improve learning performance, they consume a large amount of power and render the network vulnerable to adversaries [4]. In light of this, a distributed learning procedure that reduces the amount of communication overhead as much as possible without compromising the privacy of agents and overall learning accuracy is desirable.

In distributed information processing scenarios, such as consensus [5], [6], optimization [7], [8], filtering [9], and state estimation [10]–[12] privacy can be ensured through the implementation of cryptography-based mechanisms. Various cryptography-based security approaches, including partial homomorphic cryptography [13], [14] and Paillier encryption [15], were used to enforce privacy in average consensus and dynamic state estimation algorithms. In addition, homomorphic encryption has also been extended to federated learning to protect the private data of edge devices [16]. Although these cryptography-based approaches preserve the privacy of an individual agent against external adversaries, they are ineffective against privacy theft by honest-but-curious agents. Moreover, their utilization is limited in resource-constrained networks due to significant demands for communication and computation [14], [17], [18].

In contrast to cryptography-based security approaches, perturbation-based mechanisms are less complex and maintain the privacy of individual agents [19]–[21]. In this category, differential privacy techniques inject uncorrelated noise into the exchanged information to ensure data privacy [5]. Due to its simplicity, differential privacy has been extensively used in distributed learning frameworks, including federated learning [22]. However, differential privacy is a pessimistic approach and ignores any available side information about the adversary or learning tasks. The privacy-accuracy tradeoff can be improved by perturbing the shared information using correlated noise sequences with decaying variances [19]–[21]. In [23], a graph topology-aware noise injection-based distributed learning strategy is introduced. It effectively cancels out the added noise during local aggregation steps, thereby enhancing the performance and privacy of the distributed learning algorithm. Meanwhile, a decomposition-based privacy technique makes inference more challenging for adversaries by randomly decomposing private information into two substates, of which only one is shared among agents [24]–[27]. In [28], a privacy-aware collaborative estimation strategy is proposed in the context of networked vehicles, wherein the vehicles transform their sensitive information into an alternative domain prior to exchanging it with others. An approach for crafting transformation matrices is also proposed in [28];

A conference precursor was published in the proceedings of the 25th International Conference on Information Fusion (FUSION), 2022.

This work was partly supported by the Research Council of Norway. *Vinay Chakravarthi Gogineni and Ashkan Moradi contributed equally to this work.*

Vinay Chakravarthi Gogineni was with the Norwegian University of Science and Technology, Trondheim, Norway. Now, he is with the SDU Applied AI and Data Science, the Maersk Mc-Kinney Moller Institute, University of Southern Denmark, Denmark (e-mail: vigo@mmmi.sdu.dk). He was with the Norwegian University of Science and Technology, Norway.

Ashkan Moradi and Stefan Werner are with the Norwegian University of Science and Technology, Norway (e-mail: ashkan.moradi@ntnu.no, stefan.werner@ntnu.no). Stefan Werner is also with the Department of Information and Communications Engineering, Aalto University, 00076, Finland.

Naveen K. D. Venkategowda is with the Linköping University, Sweden (e-mail: naveen.venkategowda@liu.se).

however, the transformation is computationally intensive and thus unsuitable for resource-constrained scenarios. Despite providing privacy for individual agents, these approaches are ineffective from a communication perspective. Thus, designing a distributed mechanism that provides privacy while improving communication efficiency and accuracy remains challenging.

Several techniques have been suggested in the literature to reduce the frequency of information exchanges between agents, such as clustering [29], ordered transmission [30], agent selection [31], [32], and probabilistic communication [33]. Furthermore, the literature includes more sophisticated strategies for censoring redundant information exchanges in the estimation process [34]. Besides reducing communication frequency, a decrease in data payload also enhances communication efficiency, akin to the sign-bit transmission method in [35]. Censoring and quantization approach proposed in [36] enhances communication-efficiency by combining the above mentioned both features such as reduced transmission frequency and a reduction in data payload size. Moreover, a few other schemes perform dimensionality reduction [37] and 1-bit quantization [38] before sharing the information to further limit local interactions among agents. Federated learning framework also adopted these approaches to reduce the communication burden associated with edge devices [39], [40]. Even though these methods reduce communication costs, they add a substantial computational burden on agents. Furthermore, partial-sharing-based communication [41]–[43] reduces the consumption of resources by allowing agents to share only a fraction of information during each inter-node interaction. The ease of implementation has made partial-sharing concepts popular in distributed and federated learning [44]. Majority of the existing distributed learning approaches tend to address privacy and communication efficiency as separate entities. Therefore, this paper emphasizes a comprehensive distributed learning framework that concurrently mitigates communication overhead while enhancing agent privacy.

This paper proposes partial-sharing private distributed learning (PPDL) algorithms that achieve communication efficiency through partial sharing of information with neighbors and privacy by combining decomposition- and noise-injection-based average consensus techniques. We first develop a noise injection-based PPDL algorithm that enables agents to collaborate locally by sharing only a fraction of their perturbed information, thereby reducing resource consumption while maintaining privacy. We also propose a decomposition and noise injection-based PPDL algorithm in which agents decompose local information into public and private substates and participate in the learning algorithm by sharing only a fraction of their perturbed public substate. The private substate updates locally and will not be shared with neighbors. We conduct mean and mean-square convergence analyses of the proposed strategies to examine the impact of the partial-sharing-based communication along with the noise injection and decomposition methods on the performance.

We quantify the privacy of an agent in the presence of an honest-but-curious (HBC) adversary, which is a network agent that participates in the learning process but is curious about the private information of other agents. To this end, we define privacy metric as the mean squared error (MSE) of the estimate of private information at an HBC adversary. MSE is suited better to capture the information leakage and quantify the private information acquired by the adversary in distributed multiagent networks with limited and streaming data. Further, unlike differential privacy, the setup considered in this paper provides a specific attack model and information available to the adversary that can be exploited to obtain a better privacy-accuracy trade-off. In both noise-injection-based PPDL and decomposition- and noise-injection-based PPDL, the analysis shows a tradeoff between privacy and communication efficiency, where energy savings obtained by sharing a smaller portion of the information results in a lower privacy level. However, the decomposition and noise injection-based PPDL improve privacy significantly compared to the noise injection-based PPDL. Finally, numerical simulations are provided to validate the analytical results.

The remainder of this article is organized as follows. Section II presents the system model and provides background information. Section III proposes partial-sharing private distributed learning algorithms employing noise-injection- and decomposition-based techniques. Section IV presents the first and second-order convergence analysis of the proposed noise-injection-based PPDL and decomposition and noise injection-based PPDL, while Section V characterizes the agent privacy. Simulation results are demonstrated in Section VI, and Section VII concludes the article.

Mathematical notation: Scalars are denoted by lowercase letters, column vectors by bold lowercase, and matrices by bold uppercase. Superscripts $(\cdot)^{\mathrm{T}}$ and $(\cdot)^{-1}$ denote the transpose and inverse operators, respectively. The symbol $\mathbf{1}_K$ represents the $K \times 1$ column vector with all entries equal to one, and $\mathbf{I}_K$ is the $K \times K$ identity matrix. The right Kronecker product, right block Kronecker product and Hadamard product of two matrices are denoted by $\otimes$, $\otimes_b$ and $\odot$, respectively, while $\lambda_i(\mathbf{A})$ denotes the ith eigenvalue of matrix $\mathbf{A}$. Operators col$\{\cdot\}$ and blockdiag$\{\cdot\}$ denote column-wise stacking and block diagonalization operations, respectively. The trace operator is denoted as trace$(\cdot)$ and matrix diag$(\cdot)$ denotes a diagonal matrix whose diagonals are the elements of the argument vector.

II. Background and Problem Formulation

Consider a sensor network modeled as a connected graph $\mathcal{G} = \{\mathcal{N}, \mathcal{E}\}$, where the node set $\mathcal{N}$ represents the agents of the network and $\mathcal{E}$ is the set of edges that represent bidirectional communication links between the nodes, i.e., $(k, l) \in \mathcal{E}$ if nodes k and l are connected. Furthermore, the set $\mathcal{N}_k$ indicates the neighborhood of the node k, including itself, and the cardinality of the set $\mathcal{N}_k$ is denoted by $|\mathcal{N}_k|$, while $K = |\mathcal{N}|$ is the number of agents in the network.

At every time instant n, the node k has access to an input signal $\mathbf{x}_{k,n}$ and the desired signal $y_{k,n}$, whose relation is described as

$$y_{k,n} = \mathbf{x}_{k,n}^{\mathrm{T}} \mathbf{w}^{\star} + \nu_{k,n}, \tag{1}$$

where $\mathbf{w}^{\star} \in \mathbb{R}^L$ is the optimal parameter and needs to be estimated. The vector $\mathbf{x}_{k,n} \triangleq [x_{k,n}, x_{k,n-1}, \ldots, x_{k,n-L+1}]^{\mathrm{T}}$

is the input signal vector and the observation noise $\nu_{k,n}$ is a zero-mean Gaussian random sequence with variance $\sigma^2_{\nu,k}$. The estimate of $\mathbf{w}^\star$ at time instant n, i.e., $\mathbf{w}_n$, is chosen so that it minimizes

$$J_n \triangleq \frac{1}{K}\sum_{k\in\mathcal{N}} \mathrm{E}[e^2_{k,n}], \tag{2}$$

where $e_{k,n} \triangleq y_{k,n} - \hat{y}_{k,n}$ with $\hat{y}_{k,n}$ as the estimated filter output at node k. At every time instant n, $\mathbf{w}_n$ can be recursively updated in a steepest descent manner as

$$\mathbf{w}_{n+1} = \mathbf{w}_n - \frac{\mu}{2}\nabla J_n, \tag{3}$$

where ∇ denotes the gradient operator and μ is the positive real-valued gain. Using an instantaneous approximation of the gradient, the learning rule for $\mathbf{w}_n$ becomes

$$\mathbf{w}_{n+1} = \mathbf{w}_n + \mu\sum_{k\in\mathcal{N}} \mathbf{x}_{k,n}e_{k,n} = \frac{1}{K}\sum_{k\in\mathcal{N}} \boldsymbol{\psi}_{k,n+1}, \tag{4}$$

where $\boldsymbol{\psi}_{k,n+1}$ is the intermediate estimate of $\mathbf{w}^\star$ at node k and time instant n and is defined as

$$\boldsymbol{\psi}_{k,n+1} = \mathbf{w}_n + \eta\,\mathbf{x}_{k,n}\,e_{k,n}, \tag{5}$$

with $\eta = \mu K$ as the step size.

The average of the intermediate estimates $\boldsymbol{\psi}_{k,n+1}$ in (5) across the entire network can be evaluated in a distributed manner using an average consensus filter (ACF) [46]–[48]. In the ACF, agents update their states through an iterative process of sharing their information and utilizing the information of neighboring agents at each consensus iteration. Operation of the ACF at its mth iteration is stated as [47]

$$\mathbf{h}_{k,(m)} = \sum_{l\in\mathcal{N}_k} a_{lk}\,\mathbf{h}_{l,(m-1)}, \tag{6}$$

where $\mathbf{h}_{k,(m)}$ with $\mathbf{h}_{k,(0)} = \boldsymbol{\psi}_{k,n+1}$, is the state of the ACF at node k after m iterations. The combiner coefficients a_{lk} are non-negative and satisfy $\sum_{l\in\mathcal{N}_k} a_{lk} = 1$. If matrix $\mathbf{A}$ with $[\mathbf{A}]_{l,k} = a_{lk}$, is doubly stochastic and satisfies the conditions stated in [47], all agents reach consensus on the exact average, i.e.,

$$\lim_{m\to\infty} \mathbf{h}_{k,(m)} = \frac{1}{K}\sum_{l\in\mathcal{N}} \mathbf{h}_{l,(0)}. \tag{7}$$

To reach an average consensus, agents exchange local information $\boldsymbol{\psi}_{k,n+1}$ with their neighbors. Potential adversaries attempt to access the node-sensitive information by exploiting the shared information. Agents are therefore required to safeguard shared information when performing distributed learning in order to prevent node-sensitive data from being inferred by adversaries.

In particular, the input signal $\mathbf{x}_{k,n}$ is considered to be sensitive and must be protected since it contains private information about the agent k. We have seen that agents only exchange their intermediate estimates $\boldsymbol{\psi}_{k,n+1}$ in order to estimate the parameter $\mathbf{w}^\star$, which includes local information about the sensitive input signal $\mathbf{x}_{k,n}$. Consequently, since adversaries are able to infer $\mathbf{x}_{k,n}$ using $\boldsymbol{\psi}_{k,n+1}$, we consider the information shared, i.e., $\mathbf{h}_{k,(0)} = \boldsymbol{\psi}_{k,n+1}$, to be private and therefore needs to be protected. Moreover, since the distributed agents have limited battery and computational power, the inter-agent communication overhead must be minimized while maintaining the advantage of collaboration.

A. Noise Injection-based ACF

To protect the node-sensitive information from being inferred by adversaries, agents exchange perturbed versions of their private information [19]–[21]. Thus, the operation of noise injection-based privacy-preserving ACF at mth iteration is given by [19]:

$$\mathbf{h}_{k,(m)} = \sum_{l\in\mathcal{N}_k} a_{lk}\,\breve{\mathbf{h}}_{l,(m-1)}, \tag{8}$$

where $\breve{\mathbf{h}}_{l,(m-1)} = \mathbf{h}_{l,(m-1)} + \boldsymbol{\omega}_{l,(m-1)}$ is the perturbed local information from the lth agent and $\boldsymbol{\omega}_{l,(m-1)}$ is the perturbation noise at $(m-1)$th consensus iteration. The designed perturbation noise at each consensus iteration is given by

$$\boldsymbol{\omega}_{l,(m)} = \begin{cases} \boldsymbol{\nu}_{l,(0)}, & m = 0, \\ \phi^m\boldsymbol{\nu}_{l,(m)} - \phi^{m-1}\boldsymbol{\nu}_{l,(m-1)}, & \text{otherwise}, \end{cases} \tag{9}$$

where $\phi \in (0,1)$, same for all agents and $\boldsymbol{\nu}_{k,(m)} \in \mathbb{R}^L$ is a zero-mean Gaussian sequence with $\mathrm{E}[\boldsymbol{\nu}_{k,(m)}\boldsymbol{\nu}^{\mathrm{T}}_{k,(m)}] = \sigma^2_\nu\mathbf{I}_L$. If matrix $\mathbf{A}$ is a doubly stochastic matrix that satisfies the conditions stated in [47] and the perturbation noise follows (9), all agents reach a consensus on the average, i.e., $\bar{\mathbf{h}} = \frac{1}{K}\sum_{l\in\mathcal{N}}\mathbf{h}_{l,(0)}$, in the mean square sense [19].

B. Decomposition and Noise Injection based-ACF

Decomposition-based ACF takes a different approach from noise-injection-based ACF to preserve local information. In decomposition-based ACF, each agent k decomposes its local information $\mathbf{h}_{k,(0)}$ into public and private substates. The public substate is exchanged with neighbors while the private substate is updated internally and will not be observed by neighbors [24]. Although the private substate is invisible to neighbors, it contributes directly to the evolution of the public substate. To this end, agent k chooses the initial public and private substates $\boldsymbol{\alpha}_{k,(0)}$ and $\boldsymbol{\beta}_{k,(0)}$ randomly from the set of all real numbers such that

$$\boldsymbol{\alpha}_{k,(0)} + \boldsymbol{\beta}_{k,(0)} = 2\mathbf{h}_{k,(0)}. \tag{10}$$

To simplify the mathematical derivations, one can set $\boldsymbol{\alpha}_{k,(0)} = \theta\mathbf{h}_{k,(0)}$, where θ is randomly chosen from the uniform distribution $U(0,1)$. This simplification subsequently results $\boldsymbol{\beta}_{k,(0)} = (2-\theta)\mathbf{h}_{k,(0)}$. To further protect the node-sensitive information, at each consensus iteration m, agents share only a perturbed version of their public substate with their neighboring agents. Subsequently, at agent k, the decomposition and noise injection-based privacy-preserving ACF [27], at mth iteration is stated as

$$\begin{cases} \boldsymbol{\alpha}_{k,(m)} = \boldsymbol{\alpha}_{k,(m-1)} + \varepsilon\gamma_k\left(\boldsymbol{\beta}_{k,(m-1)} - \boldsymbol{\alpha}_{k,(m-1)}\right) \\ \qquad + \varepsilon\sum_{l\in\mathcal{N}_k}\gamma_{lk}\left(\breve{\boldsymbol{\alpha}}_{l,(m-1)} - \boldsymbol{\alpha}_{k,(m-1)}\right), \\ \boldsymbol{\beta}_{k,(m)} = \boldsymbol{\beta}_{k,(m-1)} + \varepsilon\gamma_k\left(\boldsymbol{\alpha}_{k,(m-1)} - \boldsymbol{\beta}_{k,(m-1)}\right), \end{cases} \tag{11}$$

where $\tilde{\alpha}_{l,(m-1)} = \alpha_{l,(m-1)} + \omega_{l,(m-1)}$ is the perturbed public substate from lth agent and the perturbation noise $\omega_{l,(m-1)}$ is the same as described in (9). The interaction weight between agents l and k is denoted by γ_{lk} that satisfies $\gamma_{lk} = \gamma_{kl}, \forall l,k$ and $\gamma_{l,k} = 0$ for $(l,k) \notin E$, while δ_k is the kth agent coupling weight that controls the level of contribution of each substate in the updating process. Moreover, we constrained γ_{lk} and all δ_k to reside in the interval $[\eta, 1)$ [24], where $\eta \in (0,1)$. The consensus parameter ε resides in the range $(0, 1/(\Delta + 1)]$, where $\Delta \triangleq \max_{k\in N} \sum_{l\in N_k} \gamma_{lk}$. It is important to note that both substates of the kth agent converge to the average consensus value for sufficiently large consensus iterations, i.e.,

$$\lim_{m\to\infty} \alpha_{k,(m)} = \lim_{m\to\infty} \beta_{k,(m)} = \frac{1}{|N|}\sum_{l\in N} \mathbf{h}_{l,(0)}. \tag{12}$$

III. Partial Sharing Private Distributed Learning Strategies

As can be seen from (6), collaboration between agents is vital for distributed learning. There is no exception to privacy-preserving distributed consensus techniques (9) and (11). Although collaboration among agents improves learning accuracy, it is resource-intensive. In multiagent networks, agents are usually limited in battery power and computational resources. Thus, it is essential to reduce inter-node communication while maintaining the benefits of inter-node cooperation. To this end, we aim to develop privacy-preserving distributed learning strategies that preserve privacy while also ensuring communication efficiency by using techniques to reduce local information exchange. Although several attempts have been made in the literature to improve the communication efficiency in distributed multiagent networks, none have been applied to the noise-injection- and decomposition-based privacy-preserving distributed consensus strategies. Therefore, by employing partial sharing-based communication [41], [42] among agents in private distributed consensus strategies, we achieve both privacy preservation and communication efficiency in a single framework.

A. Noise Injection-based PPDL

In the proposed noise injection-based PPDL, during each consensus iteration m, every agent only shares a fraction of the perturbed version of its private information with neighbors (i.e., L' entries of $\mathbf{h}_{k,(m)}$, with $L' \le L$) to reduce the inter-node communication overhead while maintaining privacy. At each agent k, instant n and consensus iteration m, the entry selection procedure is characterized by a diagonal selection matrix $\mathbf{S}_{k,n,(m)}$ that consists of L' numbers of ones and $L - L'$ numbers of zeros on its main diagonal. The position of ones in $\mathbf{S}_{k,n,(m)}$ indicates which entries of the perturbed private information are to be shared with neighbors. The selection of L' out of L entries can be made stochastically or sequentially as in [41], [42]. To keep the selection procedure simple, we adopt coordinated partial-sharing, a special case of sequential partial-sharing-based communication method [42]. In coordinated partial sharing-based communication, all agents are initialized with the same selection matrices i.e., $\mathbf{S}_{1,0,(0)} = \mathbf{S}_{2,0,(0)} \cdots \mathbf{S}_{K,0,(0)} = \mathbf{S}_{0,(0)}$. This implies every agent in the network shares the same portion of the perturbed private information with its neighbors. In the partial sharing-based communication, the selection matrix at the current consensus iteration $\mathbf{S}_{k,n,(m)}$ can be obtained by performing τ right circular shift operations on the main diagonal elements of the entry selection matrix used in the previous consensus iteration $\mathbf{S}_{k,n,(m-1)}$, i.e., $\mathrm{diag}\{\mathbf{S}_{k,n,(m)}\} = \mathrm{circularshift}(\mathrm{diag}\{\mathbf{S}_{k,n,(m-1)}\}, \tau)$, with $\mathbf{S}_{k,n,(0)} = \mathbf{S}_{k,n-1,(m)}$. Here, the integer τ indicates the number of right circular shifts, and $\mathrm{diag}\{\cdot\}$ operator returns a column vector that consists of the main diagonal elements of its argument matrix. Since every agent in the network uses the same selection matrix at each time instance n and consensus iteration m, we drop node index in $\mathbf{S}_{k,n,(m)}$ and continue with $\mathbf{S}_{n,(m)}$. In coordinated partial sharing-based communication, each entry of the perturbed private information will be shared L' times during L subsequent iterations. Thus, the frequency of each entry being shared is equal to $p_e = \frac{L'}{L}$.

Using the selection matrices, at agent k, noise injection-based privacy-preserving ACF is expressed alternatively as

$$\begin{aligned}\mathbf{h}_{k,(m)} = {} & a_{kk}\hat{\mathbf{h}}_{k,(m-1)} \\ & + \sum_{l\in N_k} a_{lk}\left(\mathbf{S}_{n,(m-1)}\hat{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)})\hat{\mathbf{h}}_{l,(m-1)}\right),\end{aligned} \tag{13}$$

where N_k indicates the neighborhood of node k excluding itself. Due to incomplete information exchange among agents, resulting from partial sharing, agents lack access to unshared portions of local information from neighbors. Allowing agents to use their local information instead effectively resolves this issue. At each agent k, we therefore substitute $(\mathbf{I} - \mathbf{S}_{n,(m-1)})\hat{\mathbf{h}}_{k,(m-1)}$ in the place of $(\mathbf{I} - \mathbf{S}_{n,(m-1)})\hat{\mathbf{h}}_{l,(m-1)}$ for $l \in N_k$ that results in

$$\begin{aligned}\mathbf{h}_{k,(m)} = {} & a_{kk}\hat{\mathbf{h}}_{k,(m-1)} \\ & + \sum_{l\in N_k} a_{lk}\left(\mathbf{S}_{n,(m-1)}\hat{\mathbf{h}}_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)})\hat{\mathbf{h}}_{k,(m-1)}\right)\end{aligned}$$

With further simplification and employing $a_{kk} + \sum_{l\in N_k} a_{lk} = 1$, we obtain

$$\begin{aligned}\mathbf{h}_{k,(m)} = {} & \hat{\mathbf{h}}_{k,(m-1)} \\ & + \sum_{l\in N_k} a_{lk}\mathbf{S}_{n,(m-1)}\left(\hat{\mathbf{h}}_{l,(m-1)} - \hat{\mathbf{h}}_{k,(m-1)}\right),\end{aligned} \tag{14}$$

The workflow of the noise injection-based PPDL is summarized in Algorithm 1.

B. Decomposition and Noise injection-based PPDL

In the proposed decomposition and noise injection-based PPDL, during each consensus iteration m, every agent only shares a portion of the perturbed version of its public substate with neighbors (i.e., L' entries of $\tilde{\alpha}_{k,(m)}$, with $L' \le L$) to reduce the inter-node communication overhead while preserving the privacy of its private substate. With the help of selection

Algorithm 1 Noise Injection-based PPDL

```
At each time index $n$ for nodes $k = 1, 2, \cdots, K$;
Initialize: $\mathbf{S}_{n,(0)}$ and ,
$\hat{y}_{k,n} = \mathbf{x}_{k,n}^{\mathrm{T}}\mathbf{w}_{k,n}$,
$e_{k,n} = y_{k,n} - \hat{y}_{k,n}$,
Local Update:
    $_{k,n+1} = \mathbf{w}_{k,n} + \mathbf{x}_{k,n} e_{k,n};$        (15)
ACF:
Set $\mathbf{h}_{k,(0)} = {}_{k,n+1}$,
For $m = 1$ to $M$
    Share $\mathbf{S}_{n,(m-1)}\hat{\mathbf{h}}_{k,(m-1)}$,
    Receive $\mathbf{S}_{n,(m-1)}\hat{\mathbf{h}}_{l,(m-1)} : \forall l \in \mathcal{N}_k$,
    Update $\mathbf{h}_{k,(m)}$ as in (14)
    $\mathrm{diag}\{\mathbf{S}_{n,(m)}\} = \mathrm{circularshift}(\mathrm{diag}\{\mathbf{S}_{n,(m-1)}\}, )$
Endfor
$\mathbf{w}_{k,n+1} = \mathbf{h}_{k,(M)}$.
```

matrices, at agent k, the update equations in (11) can be expressed alternatively as

$$\begin{cases} {}_{k,(m)} = {}_{k,(m-1)} + \varepsilon\, {}_k\left({}_{k,(m-1)} - {}_{k,(m-1)}\right) \\ \quad + \varepsilon \sum_{l\in\mathcal{N}_k} {}_{lk}\Big(\mathbf{S}_{n,(m-1)} e_{l,(m-1)} \\ \quad + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) e_{l,(m-1)} - {}_{k,(m-1)}\Big); \\ {}_{k,(m)} = {}_{k,(m-1)} + \varepsilon\, {}_k\left({}_{k,(m-1)} - {}_{k,(m-1)}\right). \end{cases} \tag{16}$$

Using the own public substate portion $(\mathbf{I} - \mathbf{S}_{n,(m-1)}) e_{k,(m-1)}$ in the place of unshared portion of neighbors public substate $(\mathbf{I} - \mathbf{S}_{n,(m-1)}) e_{l,(m-1)}$ for $l \in \mathcal{N}_k$, we finally have

$$\begin{cases} {}_{k,(m)} = {}_{k,(m-1)} + \varepsilon\, {}_k\left({}_{k,(m-1)} - {}_{k,(m-1)}\right) \\ \quad + \varepsilon \sum_{l\in\mathcal{N}_k} {}_{lk}\Big(\mathbf{S}_{n,(m-1)} e_{l,(m-1)} \\ \quad + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) e_{k,(m-1)} - {}_{k,(m-1)}\Big); \\ {}_{k,(m)} = {}_{k,(m-1)} + \varepsilon\, {}_k\left({}_{k,(m-1)} - {}_{k,(m-1)}\right). \end{cases} \tag{17}$$

The workflow of the proposed decomposition-based PPDL is summarized in Algorithm 2.

Communication Savings: During each consensus iteration, every agent in the network shares a subset of its local information with neighboring agents, i.e., L' number of elements. This approach enables the proposed algorithms to achieve communication savings, quantified by the percentage $\frac{(L-L')}{L} \times 100\%$. It is important to emphasize that the PPDL algorithms presented in this work offer communication savings while protecting local information without imposing additional computational burdens on the agents. The update process of selection matrices involves simple shifting operations, which do not require extensive computations like dimensionality reduction [37] and 1-bit quantization [38] methods.

Algorithm 2 Decomposition and Noise Injection-based PPDL

```
At each time index $n$ for nodes $k = 1, 2, \cdots, K$;
Initialize: $\mathbf{S}_{n,(0)}$ and ,
$\hat{y}_{k,n} = \mathbf{x}_{k,n}^{\mathrm{T}}\mathbf{w}_{k,n}$,
$e_{k,n} = y_{k,n} - \hat{y}_{k,n}$,
Local Update:
    $_{k,n+1} = \mathbf{w}_{k,n} + \mathbf{x}_{k,n} e_{k,n};$        (18)
ACF:
Set $\mathbf{h}_{k,(0)} = {}_{k,n+1}$, and partition it into two substates
Private substate: $_{k,(0)} = \mathbf{h}_{k,(0)}$,
Public substate: $_{k,(0)} = (2 - )\mathbf{h}_{k,(0)}$,
For $m = 1$ to $M$
    Share $\mathbf{S}_{n,(m-1)} e_{k,(m-1)}$;
    Receive $\mathbf{S}_{n,(m-1)} e_{l,(m-1)} : \forall l \in \mathcal{N}_k$
    $e'_{l,(m-1)} = \mathbf{S}_{n,(m-1)} e_{l,(m-1)} + (\mathbf{I} - \mathbf{S}_{n,(m-1)}) e_{k,(m-1)}$
        $_{k,(m)} = {}_{k,(m-1)} + \varepsilon\, {}_k({}_{k,(m-1)} - {}_{k,(m-1)})$
            $+ \varepsilon \sum_{l\in\mathcal{N}_k} {}_{lk}(e'_{l,(m-1)} - {}_{k,(m-1)});$        (19)
        $_{k,(m)} = {}_{k,(m-1)} + \varepsilon\, {}_k({}_{k,(m-1)} - {}_{k,(m-1)});$
    $\mathrm{diag}\{\mathbf{S}_{n,(m)}\} = \mathrm{circularshift}(\mathrm{diag}\{\mathbf{S}_{n,(m-1)}\}, )$,
Endfor
$\mathbf{w}_{k,n+1} = {}_{k,(M)}$.
```

IV. Learning Performance Analysis

Throughout this section, we examine the convergence behavior of the proposed PPDL strategies. In particular, we study the impact of partial-sharing-based communication on the convergence of privacy-preserving distributed learning.

To begin with, we first define certain network-level quantities at time instance n as follows: optimal parameter vector $\mathbf{w}^{\star}_{net} \triangleq \mathbf{1}_K \otimes \mathbf{w}^{\star}$, estimated parameter vector $\mathbf{w}_n \triangleq \mathrm{col}\{\mathbf{w}_{1,n}, \mathbf{w}_{2,n}, \ldots, \mathbf{w}_{K,n}\}$, at mth consensus iteration private information $\mathbf{h}_{(m)} \triangleq \mathrm{col}\{\mathbf{h}_{1,(m)}, \mathbf{h}_{2,(m)}, \ldots, \mathbf{h}_{K,(m)}\}$ with $\mathbf{h}_{(0)} \triangleq \mathrm{col}\{{}_{1,n}, {}_{2,n}, \ldots, {}_{K,n}\}$, private and public substates at mth consensus iteration ${}_{(m)} \triangleq \mathrm{col}\{{}_{1,(m)}, {}_{2,(m)}, \ldots, {}_{K,(m)}\}$ and ${}_{(m)} \triangleq \mathrm{col}\{{}_{1,(m)}, {}_{2,(m)}, \ldots, {}_{K,(m)}\}$, input data $\mathbf{X}_n \triangleq \mathrm{blockdiag}\{\mathbf{x}_{1,n}, \mathbf{x}_{2,n}, \ldots, \mathbf{x}_{K,n}\}$, and observation noise ${}_n \triangleq \mathrm{col}\{{}_{1,n}, {}_{2,n}, \ldots, {}_{K,n}\}$. Using the above definitions, the network-level data model and error vector can be expressed as

$$\begin{aligned} \mathbf{y}_n &\triangleq \mathrm{col}\{y_{1,n}, y_{2,n}, \ldots, y_{K,n}\} = \mathbf{X}_n^{\mathrm{T}}\mathbf{w}^{\star}_{net} + {}_n, \\ \mathbf{e}_n &\triangleq \mathrm{col}\left(e_{1,n}, e_{2,n}, \ldots, e_{K,n}\right) = \mathbf{y}_n - \mathbf{X}_n^{\mathrm{T}}\mathbf{w}_n. \end{aligned} \tag{20}$$

For establishing the convergence conditions and to obtain the closed-form expressions for network-level mean square error (MSE) of the proposed PPDL strategies, we make the following assumptions:

A1: For all $k \in \mathcal{N}$, the input signal vector $\mathbf{x}_{k,n}$ is drawn from a WSS multivariate random sequence with correlation matrix $\mathbf{R}_k \triangleq \mathrm{E}[\mathbf{x}_{k,n}\mathbf{x}_{k,n}^{\mathrm{T}}]$. Furthermore, the input signal vectors $\mathbf{x}_{k,n}$ and $\mathbf{x}_{l,m}$ are independent for all $k \neq l$ and $n \neq m$.

A2: The observation noise process ${}_{k,n}$ is assumed to be zero-mean i.i.d. and independent of any other quantity.

A3: The perturbation sequences ${}_{k,(m)}$ are assumed to be mutually independent for all agents $k \in \mathcal{N}$.

A4: For all $k \in \mathcal{N}$, the section matrix $\mathbf{S}_{n,(m)}$ is independent

of any other data. Additionally, the selection matrices $\mathbf{S}_{n;(m)}$ and $\mathbf{S}_{n^0;(m^0)}$ are independent for all $n \neq n^0$ and $m \neq m^0$.
A5: For sufficiently small learning rate , the terms involving higher-order powers of can be neglected.

Denoting the network-level weight error vector $\tilde{\mathbf{w}}_n = \mathbf{w}^?_{net} \;\; \mathbf{w}_n$, the network-level error can be alternatively expressed as $\mathbf{e}_n = \mathbf{X}_n^\mathsf{T}\tilde{\mathbf{w}}_n + \;\; {}_n$. Thus, the network-level mean square error (MSE) at time instance n: ${}_n = \frac{1}{K}\mathsf{E}[\mathbf{e}_n^\mathsf{T}\mathbf{e}_n] = \mathsf{E}[\tilde{\mathbf{w}}_n^\mathsf{T}\mathbf{X}_n\mathbf{X}_n^\mathsf{T}\tilde{\mathbf{w}}_n] + \mathsf{E}[\;{}_n^\mathsf{T}\;{}_n] = \mathsf{E}[k\tilde{\mathbf{w}}_nk^2_{\mathcal{R}}] + \mathrm{trace}(\mathbf{\Lambda}\;)$, where $\mathcal{R} = \mathsf{E}[\mathbf{X}_n\mathbf{X}_n^\mathsf{T}] = \mathrm{blockdiag}f\mathbf{R}_1;\mathbf{R}_2;:::;\mathbf{R}_Kg$ and $\mathbf{\Lambda} = \mathsf{E}[\;{}_n^\mathsf{T}\;{}_n] = \mathrm{diag}f\;{}^2_{;1};\;{}^2_{;2};:::;\;{}^2_{;K}g$. The term $\mathsf{E}[k\tilde{\mathbf{w}}_nk^2_{\mathcal{R}}]$ is often referred to as network-level excess MSE (EMSE).

A. Analysis of Noise Injection-based PPDL

Using the above stated network-level definitions, from (15), (14) and (20), the network-level recursion of noise injection-based PPDL is given by

$$\mathbf{w}_{n+1} = \mathcal{B}_n \;\; \mathbf{w}_n + \;\; \mathbf{X}_n \mathbf{e}_n \;\; + \;\; {}_n; \qquad (21)$$

with

$$\mathcal{B}_n = \mathop{\mathrm{Y}}_{i=0}^{m\;1} \mathcal{B}_{n;(i)}; \text{ and } {}_n = \mathop{\mathrm{X}}_{i=0}^{m\;1}\mathop{\mathrm{Y}}_{j=i}^{m\;1} \mathcal{B}_{n;(j)} \;\; !_{(i)}; \qquad (22)$$

where $\mathcal{B}_{n;(i)} = \mathbf{A} \;\; \mathbf{S}_{n;(i)} + \mathbf{I}_K \;\; (\mathbf{I}_L \;\; \mathbf{S}_{n;(i)})$ and $!_{(i)}\;,$ col$f!_{1;(i)};!_{2;(i)};:::;!_{K;(i)}g$. Here, the network-level perturbation noise vector is given by

$$!_{(m)} = \begin{cases} {}_{(0)}; & m = 0; \\ {}_{m}\;\;{}_{(m)} \;\;\; {}_{m\;\;1}\;\;{}_{(m\;\;1)}; & \text{otherwise}; \end{cases} \qquad (23)$$

with ${}_{(m)}\;,$ col$f\;{}_{1;(m)};\;{}_{2;(m)};:::;\;{}_{K;(m)}g$.

1) First-Order Convergence Analysis: Recalling that $\mathbf{w}^?_{net} = \mathcal{B}_n\mathbf{w}^?_{net}$ (since the row sum of $\mathcal{B}_{n;(i)}$ is unity and due to the structure of $\mathbf{w}^?_{net}$, $8i = 0;1;:::;m \;\; 1$; we will have $\mathcal{B}_{n;(i)}\mathbf{w}^?_{net} = \mathbf{w}^?_{net}$ and thus, $\mathrm{Q}_{i=0}^{m\;1}\mathcal{B}_{n;(i)}\mathbf{w}^?_{net} = \mathbf{w}^?_{net}$), form (21), the recursion for $\tilde{\mathbf{w}}_{n+1}$ can be stated as

$$\tilde{\mathbf{w}}_{n+1} = \mathcal{B}_n \;\; \mathbf{I}_{LK} \;\;\; \mathbf{X}_n\mathbf{X}_n^\mathsf{T} \;\; \tilde{\mathbf{w}}_n \;\;\; \mathcal{B}_n\mathbf{X}_n \;\; {}_n \;\;\; {}_n: \qquad (24)$$

Theorem 1: Let **A1**-**A4** hold true. Then, noise injection-based PPDL converges in the mean if and only if

$$0 < \;\; < \frac{2}{\max_{8p;k} f\;{}_p(\mathbf{R}_k)g} \qquad (25)$$

Proof: Taking the expectation $\mathsf{E}[\;]$ on both sides of (24) and using assumptions **A1**-**A4**, we have

$$\mathsf{E}[\tilde{\mathbf{w}}_{n+1}] = \mathsf{E}[\mathcal{B}_n] \;\; \mathbf{I}_{LK} \;\;\; \mathcal{R} \;\; \mathsf{E}[\tilde{\mathbf{w}}_n]: \qquad (26)$$

From (26), one can see that $\lim_{n\;!\;1}\mathsf{E}[\tilde{\mathbf{w}}_{net;n}]$ attains a finite value if and only if $k\mathsf{E}[\mathcal{B}_n] \;\mathbf{I}_{LK} \;\;\mathcal{R}\; k < 1$, where $k \;\; k$ is any matrix norm. In order to establish the convergence condition, we use the block maximum norm of the matrix, i.e., $k \;\; k_{b;\,1}$ [49]. From the properties of block maximum norm, one can write $k\mathsf{E}[\mathcal{B}_n] \;\mathbf{I}_{LK} \;\;\mathcal{R}\; k_{b;\,1} \;\;\; k\mathsf{E}[\mathcal{B}_n]k_{b;\,1}\, k\mathbf{I}_{LK} \;\;\mathcal{R}k_{b;\,1}$. Following the similar procedure as in [41], one can easily prove $\mathsf{E}[\mathbf{B}_{n;(i)}] = p_e(\mathbf{A} \;\;\mathbf{I}_L) + (1 \;\; p_e)\mathbf{I}_{LK}$ and its row sum equals to one. We then have

$$k\mathsf{E}[\mathcal{B}_n]k_{b;\,1} \;\;\; \mathop{\mathrm{Y}}_{i=0}^{m\;1} k\mathsf{E}[\mathbf{B}_{n;(i)}]k_{b;\,1} = 1: \qquad (27)$$

Thus, using [50, Lemma D. 5], it is seen that E $\tilde{\mathbf{w}}_n$ converges under $8p;k : j1 \;\;\; {}_p(\mathbf{R}_k)j < 1$. After some simplifications, we arrive at (25). ■

2) Second-Order Convergence Analysis: Let $\mathbf{\Sigma}$ be an arbitrary positive semi-definite matrix. Then from (24), the recursion for the weighted mean-square deviation (MSD), i.e., $\mathsf{E}[k\tilde{\mathbf{w}}_nk^2_{\mathbf{\Sigma}}] = \mathsf{E}[\tilde{\mathbf{w}}_n^\mathsf{T}\mathbf{\Sigma}\tilde{\mathbf{w}}_n]$, is stated as

$$\begin{aligned}\mathsf{E}[k\tilde{\mathbf{w}}_{n+1}k^2_{\mathbf{\Sigma}}] &= \mathsf{E}[k\tilde{\mathbf{w}}_nk^2_{\mathbf{\Sigma}^0_{\mathrm{NI}}}] + \;{}^2\mathsf{E}[\;{}_n^\mathsf{T}\mathbf{X}_n^\mathsf{T}\mathcal{B}_n^\mathsf{T}\mathbf{\Sigma}\mathcal{B}_n\mathbf{X}\;\;{}_n] \\ &\quad + \mathsf{E}[\;{}_n^\mathsf{T}\mathbf{\Sigma}\;\;{}_n]; \end{aligned} \qquad (28)$$

where the cross terms are vanished under the assumption **A2** and **A3**. The matrix $\mathbf{\Sigma}^0_{\mathrm{NI}}$ is given by

$$\mathbf{\Sigma}^0_{\mathrm{NI}} = \mathsf{E} \;\; \mathbf{I} \;\;\; \mathbf{X}_n\mathbf{X}_n^\mathsf{T} \;\; \mathcal{B}_n^\mathsf{T}\mathbf{\Sigma}\mathcal{B}_n \;\; \mathbf{I} \;\;\; \mathbf{X}_n\mathbf{X}_n^\mathsf{T} \;\; : \qquad (29)$$

Using the properties of block vectorization operator bvec$f\;g$ and block Kronecker product [51], one can relate $\;\;,\;$ bvec$f\mathbf{\Sigma}g$ and ${}^0_{\mathrm{NI}}\;,\;$ bvec$f\mathbf{\Sigma}^0_{\mathrm{NI}}g$ as

$$\begin{aligned} {}^0_{\mathrm{NI}} &= \mathrm{bvec} \;\; \mathsf{E}\;(\mathbf{I} \;\;\; \mathbf{X}_n\mathbf{X}_n^\mathsf{T})\mathcal{B}_n^\mathsf{T}\mathbf{\Sigma}\mathcal{B}_n(\mathbf{I} \;\;\; \mathbf{X}_n\mathbf{X}_n^\mathsf{T}) \\ &= \mathcal{F}_{\mathrm{NI}}^\mathsf{T} \;\; ; \end{aligned} \qquad (30)$$

with

$$\mathcal{F}_{\mathrm{NI}} = \mathcal{Q}_{\mathcal{B}} \;\;\; \mathcal{Q}_{\mathcal{B}}(\mathbf{I} \;\; {}_b\,\mathcal{R}) \;\;\; \mathcal{Q}_{\mathcal{B}}(\mathcal{R} \;\; {}_b\,\mathbf{I}); \qquad (31)$$

where

$$\mathcal{Q}_{\mathcal{B}} = \mathsf{E}[\mathcal{B}_n \;\; {}_b\,\mathcal{B}_n] = \mathop{\mathrm{Y}}_{i=0}^{m\;1}\mathsf{E}[\mathcal{B}_{n;(i)} \;\; {}_b\,\mathcal{B}_{n;(i)}]: \qquad (32)$$

The higher-order powers of are ignored in (31) under **A5**. We continue the analysis with this approximation. In the above $\mathsf{E}[\mathbf{B}_{n;(i)} \;\; \mathbf{B}_{n;(i)}] = p_e \;\; (\mathbf{A} \;\; \mathbf{A}) \;\; \mathbf{I}_{L^2} \;\; + (1 \;\; p_e)\mathbf{I}_{L^2K^2}$, and its row sum equals to one.

Next, the second term in the RHS of (28) can be evaluated as $\mathsf{E}[\;{}_n^\mathsf{T}\mathbf{X}_n^\mathsf{T}\mathcal{B}_n^\mathsf{T}\mathbf{\Sigma}\mathcal{B}_n\mathbf{X}_n\;{}_n] = \mathsf{E}[\mathrm{trace}(\mathcal{B}_n\mathbf{X}_n\;{}_n\;{}_n^\mathsf{T}\mathbf{X}_n^\mathsf{T}\mathcal{B}_n^\mathsf{T}\mathbf{\Sigma})] = \mathrm{trace}(\mathsf{E}[\mathcal{B}_n\mathbf{X}_n\mathsf{E}[\;{}_n\;{}_n^\mathsf{T}]\mathbf{X}_n^\mathsf{T}\mathcal{B}_n^\mathsf{T}]\mathbf{\Sigma}) = \mathrm{trace}(\mathsf{E}[\mathcal{B}_n\mathbf{\Phi}_n\mathcal{B}_n^\mathsf{T}]\mathbf{\Sigma})$, with $\mathbf{\Phi}_n = \mathbf{X}_n\mathbf{\Lambda}\;\mathbf{X}_n^\mathsf{T}$. From the properties of block vectorization, we finally have

$$\mathrm{trace}\;\;\mathsf{E}[\mathcal{B}_n\mathbf{\Phi}_n\mathcal{B}_n^\mathsf{T}]\mathbf{\Sigma} \;\; = \;\; {}^\mathsf{T}_{\mathrm{NI}} \;\; ; \qquad (33)$$

with

$${}_{\mathrm{NI}} = \mathrm{bvec}f\mathsf{E}[\mathcal{B}_n\mathbf{\Phi}_n\mathcal{B}_n^\mathsf{T}]g = \mathcal{Q}_{\mathcal{B}} \;\;\; ; \qquad (34)$$

where $\;\;,\;$ bvec$f\mathsf{E}[\mathbf{\Phi}_n]g =$ bvec$f\mathsf{E}[\mathbf{X}_n\mathbf{\Lambda}\;\mathbf{X}_n^\mathsf{T})]g$.

Finally, the last term on the RHS of (28) can be evaluated as

$$\begin{aligned} E[{}_n^{T}\Sigma\, {}_n] &= \sum_{i=0}^{m-1} E\left[\omega_{(i)}^{T}\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)^{T}\Sigma\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)\omega_{(i)}\right] \\ &= \sum_{i=0}^{m-1} E\left[\mathrm{trace}\left(\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)\omega_{(i)}\omega_{(i)}^{T}\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)^{T}\Sigma\right)\right] \\ &= \sum_{i=0}^{m-1}\mathrm{trace}\left(E\left[\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)\Lambda_{\omega,(i)}\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)^{T}\right]\Sigma\right), \end{aligned} \tag{35}$$

where $\Lambda_{\omega,(i)} \triangleq E[\omega_{(i)}\omega_{(i)}^{T}] = {}^{2}_{\omega,(i)}\mathbf{I}_{LK}$. From the properties of block vectorization one can simplify it further as

$$\begin{aligned} &\sum_{i=0}^{m-1}\mathrm{trace}\left(E\left[\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)\Lambda_{\omega,(i)}\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)^{T}\right]\Sigma\right) \\ &\qquad = \sum_{i=0}^{m-1}\varphi_i^{T}\,, \end{aligned} \tag{36}$$

with

$$\begin{aligned} \varphi_i &= \mathrm{bvec}\left(E\left[\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)\Lambda_{\omega,(i)}\left(\prod_{j=i}^{m-1}\mathcal{B}_{n,(j)}\right)^{T}\right]\right) \\ &= \prod_{j=i}^{m-1} E\left[\mathcal{B}_{n,(j)}\otimes_b\mathcal{B}_{n,(j)}\right]\,{}_{\omega,(i)}, \end{aligned} \tag{37}$$

where ${}_{\omega,(i)} \triangleq \mathrm{bvec}\{\Lambda_{\omega,(i)}\}$.

Integrating all of these results together, the recursion for the weighted MSD of noise injection-based PPDL can be stated as

$$\begin{aligned} E[\|\tilde{\mathbf{w}}_{n+1}\|^{2}_{\mathrm{bvec}^{-1}\{\}}] &= E[\|\tilde{\mathbf{w}}_{n}\|^{2}_{\mathrm{bvec}^{-1}\{\mathcal{F}_{\mathrm{NI}}^{T}\}}] + {}^{2}\,{}_{\mathrm{NI}}^{T} \\ &\quad + \sum_{i=0}^{m-1}\varphi_i^{T}\,, \end{aligned} \tag{38}$$

where $\mathrm{bvec}^{-1}\{\}$ is the reverse operation of block vectorization.

Theorem 2: Let **A1-A5** hold true and (38) represents weighted MSD dynamics of noise injection-based PPDL. Then, it exhibits stable MSD under

$$0 < \; < \frac{1}{\max_{\forall p,k}\{{}_p(\mathbf{R}_k)\}} \tag{39}$$

Proof: Iterating (38) downwards to $n = 0$, we obtain

$$\begin{aligned} E[\|\tilde{\mathbf{w}}_{n+1}\|^{2}_{\mathrm{bvec}^{-1}\{\}}] &= E[\|\tilde{\mathbf{w}}_{0}\|^{2}_{\mathrm{bvec}^{-1}\{(\mathcal{F}_{\mathrm{NI}}^{T})^{n+1}\}}] \\ &\quad + {}^{2}\,{}_{\mathrm{NI}}^{T}\left(\mathbf{I} + \sum_{j=1}(\mathcal{F}_{\mathrm{NI}}^{T})^{j}\right) \\ &\quad + \sum_{i=0}^{m-1}\varphi_i^{T}\left(\mathbf{I} + \sum_{j=1}(\mathcal{F}_{\mathrm{NI}}^{T})^{j}\right), \end{aligned} \tag{40}$$

where $\tilde{\mathbf{w}}_0 = \mathbf{w}^{\star}_{net} - \mathbf{w}_0$. For the convergence of $E[\|\tilde{\mathbf{w}}_n\|^{2}_{\Sigma}] = E[\|\tilde{\mathbf{w}}_n\|^{2}_{\mathrm{bvec}^{-1}\{\}}]$, $\|\mathcal{F}_{\mathrm{NI}}\|_{b,\infty} < 1$. From the properties of block maximum norm, we then can write

$$\begin{aligned} \|\mathcal{F}_{\mathrm{NI}}\|_{b,\infty} &\;\; \|\mathcal{Q}_{\mathcal{B}}\left(\mathbf{I} - \,(\mathbf{I}\otimes_b\mathcal{R}) - \,(\mathcal{R}\otimes_b\mathbf{I})\right)\|_{b,\infty} \\ &\;\; \|\mathcal{Q}_{\mathcal{B}}\|_{b,\infty}\|\mathbf{I} - \,(\mathbf{I}\otimes_b\mathcal{R}) - \,(\mathcal{R}\otimes_b\mathbf{I})\|_{b,\infty}, \end{aligned} \tag{41}$$

Using the fact that the row sum of $E[\mathbf{B}_{n,(i)}\otimes_b\mathbf{B}_{n,(i)}]$ is equal to one, we will have $\|\mathcal{Q}_{\mathcal{B}}\|_{b,\infty} = 1$. Therefore, the condition for the convergence of $E[\|\tilde{\mathbf{w}}_n\|^{2}_{\Sigma}]$ is $\|\mathbf{I} - \,(\mathbf{I}\otimes_b\mathcal{R}) - \,(\mathcal{R}\otimes_b\mathbf{I})\|_{b,\infty} < 1$, or, equivalently, $|1 - (\,{}_p(\mathcal{R}) + \,{}_q(\mathcal{R}))| < 1$, $p,q = 1,2,\,,LK$. Thus, a sufficient convergence condition is given by $0 < \; < \frac{1}{\max_{p=1,\,,LK}\,{}_p(\mathcal{R})}$, which proves (39). ■

3) Transient and Steady-State network-level EMSE: Using (38), $E[\|\tilde{\mathbf{w}}_{n+1}\|^{2}_{\mathrm{bvec}^{-1}\{\}}]$ and $E[\|\tilde{\mathbf{w}}_{n}\|^{2}_{\mathrm{bvec}^{-1}\{\}}]$ can be related as

$$\begin{aligned} &E[\|\tilde{\mathbf{w}}_{n+1}\|^{2}_{\mathrm{bvec}^{-1}\{\}}] \\ &\quad = E[\|\tilde{\mathbf{w}}_{n}\|^{2}_{\mathrm{bvec}^{-1}\{\}}] + {}^{2}\,{}_{\mathrm{NI}}^{T}(\mathcal{F}_{\mathrm{NI}}^{T})^{n} \\ &\quad + \sum_{i=0}^{m-1}\varphi_i^{T}(\mathcal{F}_{\mathrm{NI}}^{T})^{n} + E[\|\tilde{\mathbf{w}}_{e,0}\|^{2}_{\mathrm{bvec}^{-1}\{(\mathcal{F}_{\mathrm{NI}}^{T}-\mathbf{I})(\mathcal{F}_{\mathrm{NI}}^{T})^{n}\}}]. \end{aligned} \tag{42}$$

By choosing $= \mathrm{bvec}\{\mathcal{R}\}$, network-level EMSE of noise injection-based PPDL at time instance n can be obtained. Furthermore, letting $n \to \infty$ on both sides of (38), we have

$$\lim_{n\to\infty} E[\|\tilde{\mathbf{w}}_{n}\|^{2}_{\mathrm{bvec}^{-1}\{(\mathbf{I}-\mathcal{F}_{\mathrm{NI}}^{T})\}}] = {}^{2}\,{}_{\mathrm{NI}}^{T} + \sum_{i=0}^{m-1}\varphi_i^{T}. \tag{43}$$

The network-level steady-state EMSE of noise injection-based PPDL can be obtained by setting $= (\mathbf{I} - \mathcal{F}_{\mathrm{NI}}^{T})^{-1}\mathrm{bvec}\{\mathcal{R}\}$ in (43). As shown in (43), the steady-state performance of the noise injection-based PPDL depends on the number of consensus iterations and is degraded by the injected noise.

B. Analysis of Decomposition and Noise Injection-based PPDL

Using the above-stated network-level definitions and from (18)-(20), the network-level recursion of decomposition and noise injection-based PPDL is described as

$$\mathbf{w}_{n+1} = \mathcal{I}\;\mathcal{G}_n\Gamma\,\mathbf{w}_n + \;\mathbf{X}_n\mathbf{e}_n + \;{}_n\,, \tag{44}$$

with

$$\begin{aligned} \mathcal{I} &= \mathbf{I}_{LK}\quad \mathbf{0}_{LK}\,, \\ \mathcal{G}_n &= \prod_{i=0}^{m-1}\mathcal{G}_{n,(i)}, \\ {}_n &= \sum_{i=0}^{m-2}\prod_{j=i}^{m-2}\mathcal{G}_{n,(j+1)}\,\mathbf{C}_{n,(i)}\,\omega_{(i)} + \mathbf{C}_{n,(m-1)}\,\omega_{(m-1)}, \end{aligned} \tag{45}$$

where

$$\begin{aligned} \mathcal{G}_{n,(i)} &= \begin{matrix} \mathbf{I}_{LK} - \varepsilon\Theta - \varepsilon\mathbf{D}\,(\Delta\otimes\mathbf{I}_L)\,\mathcal{S}_{n,(i)} & \varepsilon\Theta \\ \varepsilon\Theta & \mathbf{I}_{LK} - \varepsilon\Theta \end{matrix}\,, \\ \mathbf{C}_{n,(i)} &= \begin{matrix} \varepsilon(\Delta\otimes\mathbf{I}_L)\mathcal{S}_{n,(i)} + \varepsilon\mathbf{D}(\mathbf{I}_{LK} - \mathcal{S}_{n,(i)}) \\ \mathbf{0} \end{matrix}\,, \\ \Gamma &= \begin{matrix} (2 - \;)\mathbf{I}_{LK} \\ \mathbf{I}_{LK} \end{matrix}\,. \end{aligned} \tag{46}$$

In the above $[\boldsymbol{\Delta}]_{l,k} = \delta_{lk}$, $\mathbf{D} = \mathrm{diag}\{\delta_1, \delta_2, \ldots, \delta_K\} \otimes \mathbf{I}_L$ with $\delta_l = \sum_{k\in\mathcal{N}_l} \delta_{lk}$, $\boldsymbol{\Theta} = \mathrm{diag}\{\theta_1, \theta_2, \ldots, \theta_K\} \otimes \mathbf{I}_L$ and $\mathcal{S}_{n,(i)} = \mathbf{I}_L \otimes \mathbf{S}_{n,(i)}$.

1) First-Order Convergence Analysis: Using the fact that $\mathbf{w}^{o}_{net} = \mathcal{I}\mathcal{G}_n\boldsymbol{\Gamma}\mathbf{w}^{o}_{net}$ (since the row sum of $\mathcal{G}_{n,(i)}$ is one for $i = 0, 1, \ldots, m-1$), then form (44), recursion for the weight error vector of the decomposition and noise injection-based PPDL can be expressed as

$$\tilde{\mathbf{w}}_{n+1} = \mathcal{I}\left(\mathcal{G}_n\boldsymbol{\Gamma}\left(\mathbf{I}_{LK} - \mu\mathbf{X}_n\mathbf{X}_n^{\mathsf{T}}\right)\tilde{\mathbf{w}}_n - \mu\mathcal{G}_n\boldsymbol{\Gamma}\mathbf{X}_n\boldsymbol{\eta}_n - \boldsymbol{\zeta}_n\right). \tag{47}$$

Theorem 3: Let the assumption **A1-A4** hold true. Then, the decomposition and noise injection-based PPDL converges in the mean if and only if

$$0 < \mu < \frac{2}{\max\limits_{\forall p,k}\{\lambda_p(\mathbf{R}_k)\}} \tag{48}$$

Proof: Taking expectation $\mathrm{E}[\cdot]$ on the both sides of (47) and using the assumptions **A1** − **A4**, we have

$$\mathrm{E}[\tilde{\mathbf{w}}_{n+1}] = \mathcal{I}\mathrm{E}[\mathcal{G}_n]\boldsymbol{\Gamma}\left(\mathbf{I}_{LK} - \mu\mathcal{R}\right)\mathrm{E}[\tilde{\mathbf{w}}_n]. \tag{49}$$

From (49), one can see that $\lim_{n\to\infty}\mathrm{E}[\tilde{\mathbf{w}}_n]$ attains a finite value if and only if $\|\mathcal{I}\mathrm{E}[\mathcal{G}_n]\boldsymbol{\Gamma}\left(\mathbf{I}_{LK} - \mu\mathcal{R}\right)\|_{b,\infty} < 1$. From the properties of the block maximum norm, one can write

$$\begin{aligned}\|\mathcal{I}\mathrm{E}[\mathcal{G}_n]\boldsymbol{\Gamma}&\left(\mathbf{I}_{LK} - \mu\mathcal{R}\right)\|_{b,\infty} \\ &\le \|\mathcal{I}\|_{b,\infty}\|\mathrm{E}[\mathcal{G}_n]\|_{b,\infty}\|\boldsymbol{\Gamma}\|_{b,\infty}\|\mathbf{I}_{LK} - \mu\mathcal{R}\|_{b,\infty}.\end{aligned} \tag{50}$$

From the definition of $\mathcal{I}$, $\boldsymbol{\Gamma}$, and $\|\mathrm{E}[\mathcal{G}_n]\|_{b,\infty}$, we have $\|\mathcal{I}\|_{b,\infty} = 1$, $\|\boldsymbol{\Gamma}\|_{b,\infty} = 1$, and $\|\mathrm{E}[\mathcal{G}_n]\|_{b,\infty} = \|\prod_{i=0}^{m-1}\mathrm{E}\left[\mathcal{G}_{n,(i)}\right]\|_{b,\infty} \le \prod_{i=0}^{m-1}\|\mathrm{E}[\mathcal{G}_{n,(i)}]\|_{b,\infty} = 1$ (since $\mathrm{E}[\mathcal{S}_{n,(i)}] = p_e\mathbf{I}_{LK}$, the row sum of $\mathrm{E}[\mathcal{G}_{n,(i)}]$ equals to one). Thus, the condition for the convergence of $\mathrm{E}\left[\tilde{\mathbf{w}}_n\right]$ is $\forall p,k: |1 - \mu\lambda_p(\mathbf{R}_k)| < 1$ and after some simplifications leads to (48). ■

2) Second-Order Convergence Analysis: Given an arbitrary positive semi-definite matrix $\boldsymbol{\Sigma}$, from (49), the recursion for the weighted MSD of decomposition and noise injection-based PPDL is

$$\begin{aligned}\mathrm{E}[\|\tilde{\mathbf{w}}_{n+1}\|^2_{\boldsymbol{\Sigma}}] = {} & \mathrm{E}[\|\tilde{\mathbf{w}}_n\|^2_{\boldsymbol{\Sigma}^0_{\mathrm{DNI}}}] \\ & + \mu^2\mathrm{E}[\boldsymbol{\eta}_n^{\mathsf{T}}\mathbf{X}_N^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}}\mathcal{G}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\mathcal{I}\mathcal{G}_n\boldsymbol{\Gamma}\mathbf{X}_n\boldsymbol{\eta}_n] \\ & + \mathrm{E}[\boldsymbol{\zeta}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\mathcal{I}\boldsymbol{\zeta}_n],\end{aligned} \tag{51}$$

where the cross terms turned zero under the assumption **A2** and **A3**. The matrix $\boldsymbol{\Sigma}^0_{\mathrm{DNI}}$ is given by

$$\boldsymbol{\Sigma}^0_{\mathrm{DNI}} = \mathrm{E}\left[\left(\mathbf{I} - \mu\mathbf{X}_n\mathbf{X}_n^{\mathsf{T}}\right)\boldsymbol{\Gamma}^{\mathsf{T}}\mathcal{G}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\mathcal{I}\mathcal{G}_n\boldsymbol{\Gamma}\left(\mathbf{I} - \mu\mathbf{X}_n\mathbf{X}_n^{\mathsf{T}}\right)\right]. \tag{52}$$

Then, $\boldsymbol{\sigma}$ and $\boldsymbol{\sigma}^0_{\mathrm{DNI}}$, $\mathrm{bvec}\{\boldsymbol{\Sigma}^0_{\mathrm{DNI}}\}$ can be related as

$$\begin{aligned}&\boldsymbol{\sigma}^0_{\mathrm{DNI}} \\ &= \mathrm{bvec}\left\{\mathrm{E}\left[\left(\mathbf{I} - \mu\mathbf{X}_n\mathbf{X}_n^{\mathsf{T}}\right)\boldsymbol{\Gamma}^{\mathsf{T}}\mathcal{G}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\mathcal{I}\mathcal{G}_n\boldsymbol{\Gamma}\left(\mathbf{I} - \mu\mathbf{X}_n\mathbf{X}_n^{\mathsf{T}}\right)\right]\right\} \\ &= \mathcal{F}^{\mathsf{T}}_{\mathrm{DNI}}\boldsymbol{\sigma},\end{aligned} \tag{53}$$

where

$$\mathcal{F}_{\mathrm{DNI}} = \mathcal{Q}_{\mathcal{G}} - \mu\mathcal{Q}_{\mathcal{G}}(\mathbf{I}\otimes_b\mathcal{R}) - \mu\mathcal{Q}_{\mathcal{G}}(\mathcal{R}\otimes_b\mathbf{I}), \tag{54}$$

with

$$\begin{aligned}\mathcal{Q}_{\mathcal{G}} &= (\boldsymbol{\Gamma}\otimes_b\boldsymbol{\Gamma})\mathrm{E}[\mathcal{G}_n\otimes_b\mathcal{G}_n](\mathcal{I}\otimes_b\mathcal{I}) \\ &= (\boldsymbol{\Gamma}\otimes_b\boldsymbol{\Gamma})\left(\prod_{i=0}^{m-1}\mathrm{E}[\mathcal{G}_{n,(i)}\otimes_b\mathcal{G}_{n,(i)}]\right)(\mathcal{I}\otimes_b\mathcal{I}).\end{aligned} \tag{55}$$

The higher-order powers of μ are ignored in (51) under **A5**. We continue the analysis with this approximation. Using the results $\mathrm{E}[\mathcal{S}_{n,(i)}] = p_e\mathbf{I}_{LK}$ and $\mathrm{E}[\mathcal{S}_{n,(i)}\otimes_b\mathcal{S}_{n,(i)}] = p_e\mathbf{I}_{L^2K^2}$, one can easily evaluate the term $\mathrm{E}[\mathcal{G}_{n,(i)}\otimes_b\mathcal{G}_{n,(i)}]$.

Next, the second term in the RHS of (51) can be evaluated as

$$\begin{aligned}\mathrm{E}[\boldsymbol{\eta}_n^{\mathsf{T}}\mathbf{X}_N^{\mathsf{T}}\boldsymbol{\Gamma}^{\mathsf{T}}\mathcal{G}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\mathcal{I}\mathcal{G}_n\boldsymbol{\Gamma}\mathbf{X}_n\boldsymbol{\eta}_n] & \\ = \mathrm{trace}\left(\mathrm{E}[\mathcal{I}\mathcal{G}_n\boldsymbol{\Gamma}\boldsymbol{\Phi}_n\boldsymbol{\Gamma}^{\mathsf{T}}\mathcal{G}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}]\boldsymbol{\Sigma}\right)& \\ = \boldsymbol{\gamma}^{\mathsf{T}}_{\mathrm{DNI}}\boldsymbol{\sigma},&\end{aligned} \tag{56}$$

where

$$\begin{aligned}\boldsymbol{\gamma}_{\mathrm{DNI}} &= \mathrm{bvec}\{\mathrm{E}[\mathcal{I}\mathcal{G}_n\boldsymbol{\Gamma}\boldsymbol{\Phi}_n\boldsymbol{\Gamma}^{\mathsf{T}}\mathcal{G}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}]\} \\ &= \mathcal{Q}_{\mathcal{G}}\boldsymbol{\phi}.\end{aligned} \tag{57}$$

The quantity $\boldsymbol{\phi}$ is the same as in Section IV-A2.

Finally, the last term on the RHS of (51) can be evaluated as

$$\begin{aligned}&\mathrm{E}[\boldsymbol{\zeta}_n^{\mathsf{T}}\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\mathcal{I}\boldsymbol{\zeta}_n] \\ &= \sum_{i=0}^{m-2}\mathrm{trace}\left(\mathcal{I}\,\mathrm{E}\left[\left(\prod_{j=i}^{m-2}\mathcal{G}_{n,(j+1)}\right)\mathbf{C}_{n,(i)}\boldsymbol{\Lambda}_{\omega,(i)}\mathbf{C}^{\mathsf{T}}_{n,(i)}\left(\prod_{j=i}^{m-2}\mathcal{G}_{n,(j+1)}\right)^{\mathsf{T}}\right]\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\right) \\ &\quad + \mathrm{trace}\left(\mathcal{I}\,\mathrm{E}\left[\mathbf{C}_{n,(m-1)}\boldsymbol{\Lambda}_{\omega,(m-1)}\mathbf{C}^{\mathsf{T}}_{n,(m-1)}\right]\mathcal{I}^{\mathsf{T}}\boldsymbol{\Sigma}\right) \\ &= \sum_{i=0}^{m-1}\boldsymbol{\vartheta}_i^{\mathsf{T}}\boldsymbol{\sigma},\end{aligned} \tag{58}$$

where $\boldsymbol{\Lambda}_{\omega,(i)}$ is the same given in the Section IV-A2, and

$$\begin{aligned}\boldsymbol{\vartheta}_i &= \mathrm{bvec}\left\{\mathcal{I}\,\mathrm{E}\left[\left(\prod_{j=i}^{m-2}\mathcal{G}_{n,(j+1)}\right)\mathbf{C}_{n,(i)}\boldsymbol{\Lambda}_{\omega,(i)}\mathbf{C}^{\mathsf{T}}_{n,(i)}\left(\prod_{j=i}^{m-2}\mathcal{G}_{n,(j+1)}\right)^{\mathsf{T}}\right]\mathcal{I}^{\mathsf{T}}\right\}, \\ &\quad + \mathrm{bvec}\left\{\mathcal{I}\mathrm{E}\left[\mathbf{C}_{n,(m-1)}\boldsymbol{\Lambda}_{\omega,(m-1)}\mathbf{C}^{\mathsf{T}}_{n,(m-1)}\right]\mathcal{I}^{\mathsf{T}}\right\} \\ &= (\mathcal{I}\otimes_b\mathcal{I})\left(\prod_{j=i+1}^{m-2}\mathrm{E}\left[\mathcal{G}_{n,(j+1)}\otimes_b\mathcal{G}_{n,(j+1)}\right]\right)\boldsymbol{\lambda}_{\omega,(i)} \\ &\qquad \mathrm{E}\left[\mathcal{G}_{n,(i)}\mathbf{C}_{n,(i)}\otimes_b\mathcal{G}_{n,(i)}\mathbf{C}_{n,(i)}\right] \\ &\quad + (\mathcal{I}\otimes_b\mathcal{I})\mathrm{E}\left[\mathbf{C}_{n,(m-1)}\otimes_b\mathbf{C}_{n,(m-1)}\right]\boldsymbol{\lambda}_{\omega,(m-1)},\end{aligned} \tag{59}$$

where $\boldsymbol{\lambda}_{\omega,(i)}$ is the same as defined in the Section IV-A2.

Putting all these results together, the recursion for the weighted MSD of decomposition and noise injection-based PPDL can be described as

$$\begin{aligned}\mathrm{E}[\|\tilde{\mathbf{w}}_{n+1}\|^2_{\mathrm{bvec}^{-1}\{\boldsymbol{\sigma}\}}] = \mathrm{E}[\|\tilde{\mathbf{w}}_n\|^2_{\mathrm{bvec}^{-1}\{\mathcal{F}^{\mathsf{T}}_{\mathrm{DNI}}\boldsymbol{\sigma}\}}] + \mu^2\boldsymbol{\gamma}^{\mathsf{T}}_{\mathrm{DNI}}\boldsymbol{\sigma}& \\ + \sum_{i=0}^{m-1}\boldsymbol{\vartheta}_i^{\mathsf{T}}\boldsymbol{\sigma}.&\end{aligned} \tag{60}$$

Theorem 4: Let **A1-A5** hold true and (60) describes weighted MSD dynamics of decomposition and noise injection-based PPDL. Then, it exhibits stable MSD under

$$0<\mu<\frac{1}{\max_{\forall p,k}\{\lambda_p(\mathbf{R}_k)\}} \tag{61}$$

Proof: Iterating (60), downwards to $n=0$, we obtain

$$\begin{aligned}\mathrm{E}[\|\tilde{\mathbf{w}}_{n+1}\|^2_{\mathrm{bvec}^{-1}\{\sigma\}}] &= \mathrm{E}[\|\tilde{\mathbf{w}}_0\|^2_{\mathrm{bvec}^{-1}\{(\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})^{n+1}\sigma\}}]\\ &\quad+\mu^2\gamma^{\mathrm{T}}_{\mathrm{DNI}}\Big(\mathbf{I}+\sum_{j=1}^{n}(\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})^j\Big)\sigma\\ &\quad+\sum_{i=0}^{m-1}\vartheta_i^{\mathrm{T}}\Big(\mathbf{I}+\sum_{j=1}^{n}(\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})^j\Big)\sigma.\end{aligned} \tag{62}$$

where $\tilde{\mathbf{w}}_0=\mathbf{w}^o_{net}-\mathbf{w}_0$. The convergence of $E[\|\tilde{\mathbf{w}}_n\|^2_{\mathrm{bvec}^{-1}\{\sigma\}}]$ requires $\|\mathcal{F}_{\mathrm{DNI}}\|_{b,\infty}<1$. From the properties of the block maximum norm, one can write

$$\|\mathcal{F}_{\mathrm{DNI}}\|_{b,\infty}\le\|\mathcal{Q}_{\mathcal{G}}\|_{b,\infty}\|\mathbf{I}-(\mathbf{I}\otimes_b\mu\mathcal{R})-(\mu\mathcal{R}\otimes_b\mathbf{I})\|_{b,\infty} \tag{63}$$

Using the fact that the row sum of $\mathcal{Q}_{\mathcal{G}}$ is one, the convergence of $\mathrm{E}[\|\tilde{\mathbf{w}}_n\|^2_{\Sigma}]$ requires $|1-\mu(\lambda_i(\mathcal{R})+\lambda_j(\mathcal{R}))|<1$, $i,j=1,2,\cdots,LK$, which in turn simplifies to (61). ■

3) Transient and Steady-State network-level EMSE: From (60), $\mathrm{E}[\|\tilde{\mathbf{w}}_{n+1}\|^2_{\mathrm{bvec}^{-1}\{\sigma\}}]$ and $\mathrm{E}[\|\tilde{\mathbf{w}}_n\|^2_{\mathrm{bvec}^{-1}\{\sigma\}}]$ can be related as

$$\begin{aligned}&\mathrm{E}[\|\tilde{\mathbf{w}}_{n+1}\|^2_{\mathrm{bvec}^{-1}\{\sigma\}}]\\ &=\mathrm{E}[\|\tilde{\mathbf{w}}_n\|^2_{\mathrm{bvec}^{-1}\{\sigma\}}]+\mu^2\gamma^{\mathrm{T}}_{\mathrm{DNI}}(\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})^n\sigma\\ &\quad+\sum_{i=0}^{m-1}\vartheta_i^{\mathrm{T}}(\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})^n\sigma+\mathrm{E}\Big[\|\tilde{\mathbf{w}}_{e,0}\|^2_{\mathrm{bvec}^{-1}\{(\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}}-\mathbf{I})(\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})^n\sigma\}}\Big].\end{aligned} \tag{64}$$

By selecting $\sigma=\mathrm{bvec}\{\mathcal{R}\}$, network-level EMSE of the decomposition and noise injection-based PPDL at time instance n, can be obtained. Letting $n\rightarrow\infty$ on both sides of (64), we obtain

$$\lim_{n\rightarrow\infty}\mathrm{E}[\|\tilde{\mathbf{w}}_n\|^2_{\mathrm{bvec}^{-1}\{(\mathbf{I}-\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})\sigma\}}]=\mu^2\gamma^{\mathrm{T}}_{\mathrm{DNI}}\sigma+\sum_{i=0}^{m-1}\vartheta_i^{\mathrm{T}}\sigma. \tag{65}$$

By substituting $\sigma=(\mathbf{I}-\mathcal{F}^{\mathrm{T}}_{\mathrm{DNI}})^{-1}\mathrm{bvec}\{\mathcal{R}\}$ in (65), the network-level steady-state EMSE of decomposition and noise-injection-based PPDL can be obtained. As shown in (65), the steady-state performance of the decomposition and noise injection-based PPDL depends on the number of consensus iterations and is degraded by the injected noise.

Remark 1: The convergence conditions of the noise-injection-based PPDL and combined decomposition and noise-injection-based PPDL strategies are those of the traditional distributed LMS (without communication-saving and data protection). Partial-sharing-based communication and noise perturbation do not alter the convergence of the proposed strategies.

Remark 2: When $L'=L$ and $\sigma^2=0$, the second term in (43) and (65) becomes zero. Thus, the steady-state MSD of the proposed strategies is the same as that of the traditional distributed LMS. On the other hand, when $L'<L$ and $\sigma^2>0$, the second term in (43) and (65) will be larger than zero. This means that the MSD of the proposed strategies at steady-state is slightly higher than that of the traditional distributed LMS. However, this slight degradation comes with enhanced privacy and communication efficiency, as detailed in the next section.

V. Privacy Analysis

This section examines the impact of communication savings on the privacy of agents. To this end, we analyze the privacy of agents in the presence of an HBC adversary and external eavesdropper. Before proceeding to the privacy analysis, we denote the privacy measure at agent l after m consensus iterations by $E_{l,(m)}$, which is defined as the mean squared estimation error at the adversary attempting to infer the private information $\mathbf{h}_{l,(0)}$, i.e.,

$$E_{l,(m)}\triangleq\mathrm{trace}\Big(\mathrm{E}\Big[(\hat{\mathbf{h}}_{l,(m)}-\mathbf{h}_{l,(0)})(\hat{\mathbf{h}}_{l,(m)}-\mathbf{h}_{l,(0)})^{\mathrm{T}}\Big]\Big), \tag{66}$$

where $\hat{\mathbf{h}}_{l,(m)}$ denotes the estimate of the private information at the adversary after m consensus iterations. The MSE is used here as a privacy metric to measure how accurately the adversary can estimate the private information of an agent. Unlike differential privacy scenarios, the attack model and information available to the adversary are specified, allowing the MSE metric to precisely quantify the information leakage. In the following, we quantify the privacy leakage of private information in both noise injection- and decomposition and noise injection-based PPDL algorithms.

A. Honest-but-Curious (HBC) Agent

The HBC agent is a network agent that has access to node-specific information and is curious about the private information of other agents. Without loss of generality, we consider agent k to be an HBC agent, attempting to estimate the private information of other agents.

1) Noise Injection-based PPDL: The HBC agent, i.e., agent k, attempts to estimate private information of other agents, i.e., $\mathbf{h}_{l,(0)}=\boldsymbol{\eta}_{l,n+1}$, for $l\in\mathcal{N}\backslash\{k\}$ using its own local information $\{\mathbf{h}_{k,(m)},\mathbf{S}_{n,(m)}\}$ and the shared information of its neighboring agents $\mathbf{S}_{n,(m)}\hat{\mathbf{h}}_{l,(m)}$, for $l\in\mathcal{N}_k$. The following theorem quantifies the privacy metric (66) for the noise injection-based PPDL algorithm.

Theorem 5: Let agent k be the HBC agent that has access to its own information and the exchanged information from its neighborhood at each consensus iteration m, i.e., $\{\mathbf{h}_{k,(m)},\mathbf{S}_{n,(m)},\mathbf{S}_{n,(m)}\hat{\mathbf{h}}_{l,(m)}\}$, for $l\in\mathcal{N}_k$. Then the privacy metric (66) at each agent l, after m consensus iterations is given by

$$E^{\mathrm{NI}}_{l,(m)}=\mathrm{trace}\Big((\mathbf{c}_l^{\mathrm{T}}\otimes\mathbf{I}_L)\mathbf{P}^{\mathrm{NI}}_{(m)}(\mathbf{c}_l\otimes\mathbf{I}_L)\Big), \tag{67}$$

where $\mathbf{c}_l$ is the $(K-1)$-dimensional canonical vector corresponding to agent l, which contains 1 as lth element and zeros elsewhere, and $\mathbf{P}^{\mathrm{NI}}_{(m)}$ is the associated error covariance to the maximum likelihood (ML) estimator at the HBC agent to estimate the private information $\mathbf{h}_{l,(0)}$, for $l\in\mathcal{N}\backslash\{k\}$.

Proof: The proof begins by stating the fact that the HBC agent is one of the network agents and has access to its own information and the exchanged information from its neighborhood at each consensus iteration m, i.e., $\{\mathbf{h}_{k;(m)}, \mathbf{S}_{n;(m)}, \mathbf{S}_{n;(m)}\hat{\mathbf{h}}_{l;(m)}\}$, for $l \in \mathcal{N}_k$. Then from (14), the network-level consensus operation in noise injection-based PPDL can be recursively computed as

$$\begin{aligned} \hat{\mathbf{h}}_{(1)} &= \mathcal{B}_{n;(0)}\mathbf{h}_{(0)} + \mathcal{B}_{n;(0)}\omega_{(0)} + \omega_{(1)}, \\ &\vdots \\ \hat{\mathbf{h}}_{(m)} &= \Big(\prod_{i=0}^{m} \mathcal{B}_{n;(i)}\Big)\mathbf{h}_{(0)} + \sum_{i=0}^{m}\Big(\prod_{j=i}^{m} \mathcal{B}_{n;(j)}\Big)\omega_{(i)}. \end{aligned} \tag{68}$$

Recall that the HBC agent already knows its own information and is only interested in estimating the private information of other agents. Thus, we reduce the dimension of the state parameters by removing the corresponding entries of the HBC agent from $\hat{\mathbf{h}}_{(m)}$, $\omega_{(m)}$, $\nu_{(m)}$, $\mathbf{h}_{(0)}$, and, $\mathcal{B}_{n;(m)}$, and continue the privacy analysis with their respective reduced quantities $\psi_{(m)}$, $\breve{\omega}_{(m)}$, $\breve{\nu}_{(m)}$, $\breve{\mathbf{h}}_{(0)}$, and, $\breve{\mathcal{B}}_{n;(m)}$. Subsequently, the reduced version of the network-level consensus operation in noise injection-based PPDL, as in (68), can be stated as

$$\psi_{(m)} = \Big(\prod_{i=0}^{m} \breve{\mathcal{B}}_{n;(i)}\Big)\breve{\mathbf{h}}_{(0)} + \sum_{i=0}^{m}\Big(\prod_{j=i}^{m} \breve{\mathcal{B}}_{n;(j)}\Big)\breve{\omega}_{(i)} \tag{69}$$

Without loss of generality, we consider agent K to be an HBC agent in the following. Thus, we consider $y_{(m)} = \mathbf{C}\psi_{(m)}$ as the observation vector at the HBC agent that comprises of the information of its neighboring agents. To captured the neighboring information at mth consensus iteration, we define $\mathbf{C} \triangleq \bar{\mathbf{C}}^{\mathrm{T}} \otimes \mathbf{I}_L$ with $\bar{\mathbf{C}} \in \mathbb{R}^{(K-1)\times|\mathcal{N}_K|}$. The lth column of $\bar{\mathbf{C}}$ represents the canonical vector corresponding to lth neighbor of agent K. Note that the canonical vector corresponding to agent l, denoted by $\mathbf{c}_l \in \mathbb{R}^{K-1}$, is a vector with 1 at lth element and zeros elsewhere. Following similar procedure as in [19] and after substituting (9) in (69), observation model at the HBC agent can be described as

$$\begin{aligned} y_{(m)} &= \mathbf{C}\psi_{(m)} \\ &= \mathbf{C}\Big(\prod_{i=0}^{m} \breve{\mathcal{B}}_{n;(i)}\Big)\breve{\mathbf{h}}_{(0)} + \epsilon^{m}\mathbf{C}\breve{\mathcal{B}}_{n;(m)}\breve{\nu}_{(m)} \\ &+ \mathbf{C}\sum_{i=0}^{m-1}\epsilon^{i}\Big(\prod_{j=i+1}^{m} \breve{\mathcal{B}}_{n;(j)}\Big)(\breve{\mathcal{B}}_{n;(i)} - \mathbf{I})\breve{\nu}_{(i)} \end{aligned} \tag{70}$$

After collecting all the observed information up to consensus iteration m, the overall observation vector can be stated as

$$\bar{y}_{(m)} = \bar{\mathcal{H}}_{(m)}\breve{\mathbf{h}}_{(0)} + \mathbf{F}_{(m)}\nu_{(m)}, \tag{71}$$

with

$$\begin{aligned} y_{(m)} &\triangleq \mathrm{col}\{y_{(0)}, y_{(1)}, \dots, y_{(m)}\}, \\ \bar{\mathcal{H}}_{(m)} &\triangleq \mathrm{col}\{\mathcal{H}_{(0)}, \mathcal{H}_{(1)}, \dots, \mathcal{H}_{(m)}\}, \\ \breve{\mathbf{h}}_{(0)} &\triangleq \mathrm{col}\{\mathbf{h}_{1;(0)}, \mathbf{h}_{2;(0)}, \dots, \mathbf{h}_{K-1;(m)}\}, \\ \nu_{(m)} &\triangleq \mathrm{col}\{\breve{\nu}_{(0)}, \cdots, \breve{\nu}_{(m)}\}, \end{aligned}$$

where $\mathcal{H}_{(m)} = \mathbf{C}\prod_{i=0}^{m}\breve{\mathcal{B}}_{n;(i)}$, and

$$\mathbf{F}_{(m)} = (\mathbf{I}_{m+1} \otimes \mathbf{C})\,\bar{\mathbf{F}}_{(m)},$$

with

$$\bar{\mathbf{F}}_{(m)} = \begin{bmatrix} \breve{\mathcal{B}}_{n;(0)} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{F}_{(1);(0)} & \epsilon\breve{\mathcal{B}}_{n;(1)} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{F}_{(2);(0)} & \epsilon\mathbf{F}_{(2);(1)} & \epsilon^{2}\breve{\mathcal{B}}_{n;(2)} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{F}_{(m);(0)} & \epsilon\mathbf{F}_{(m);(1)} & \epsilon^{2}\mathbf{F}_{(m);(2)} & \cdots & \epsilon^{m}\breve{\mathcal{B}}_{n;(m)} \end{bmatrix},$$

with $\mathbf{F}_{(m);(i)} = \prod_{t=i+1}^{m}\breve{\mathcal{B}}_{n;(t)}(\breve{\mathcal{B}}_{n;(i)} - \mathbf{I})$. Using the overall observation model in (71); the HBC agent can find a maximum likelihood (ML) estimate of private information vector $\breve{\mathbf{h}}_{(0)}$, with associated error covariance

$$\mathbf{P}^{\mathrm{NI}}_{(m)} = \Big(\bar{\mathcal{H}}^{\mathrm{T}}_{(m)}\big(\mathbf{F}_{(m)}\mathbf{\Lambda}_{\nu;(m)}\mathbf{F}^{\mathrm{T}}_{(m)}\big)^{-1}\bar{\mathcal{H}}_{(m)}\Big)^{-1}, \tag{72}$$

where $\mathbf{\Lambda}_{\nu;(m)} \triangleq \mathbb{E}\{\nu_{(m)}\nu^{\mathrm{T}}_{(m)}\} = \sigma^{2}\mathbf{I}$. As the HBC agent collects more information from neighbors, the mean squared error of the ML estimator decreases and the privacy metric (66) at each agent l can be obtained as

$$E^{\mathrm{NI}}_{l;(m)} = \mathrm{trace}\Big((\mathbf{c}^{\mathrm{T}}_{l} \otimes \mathbf{I}_L)\mathbf{P}^{\mathrm{NI}}_{(m)}(\mathbf{c}_l \otimes \mathbf{I}_L)\Big), \tag{73}$$

which completes the proof. ■

2) Decomposition and Noise Injection-based PPDL: Similar to the noise injection-based method, let agent k be an HBC agent, attempting to estimate the private information of other agents, i.e., $\mathbf{h}_{l;(0)} = \alpha_{l;n+1}$, for $l \in \mathcal{N}\setminus\{k\}$. The HBC agent has access to it own local information $\{\alpha_{k;(m)}, \beta_{k;(m)}, \mathbf{S}_{n;(m)}\}$ as well as the shared information by its neighbors $\mathbf{S}_{n;(m)}\mathbf{e}_{l;(m)}$ for $l \in \mathcal{N}_k$. We can observe that agent privacy depends on the availability of the interaction and coupling weights for all agents at the HBC adversary [27]. Thus, to investigate the privacy leakage in the worst-case scenario, we also assume that the HBC agent has also access to the coupling and interaction weight matrices $\mathbf{\Theta}$ and $\mathbf{\Delta}$.

Theorem 6: Let agent k be the HBC agent that has access not only to its own information and exchanged information in its neighborhood at every consensus iteration m, but also to the interaction and coupling weights of the entire network, i.e., $\{\alpha_{k;(m)}, \mathbf{S}_{n;(m)}, \mathbf{S}_{n;(m)}\mathbf{e}_{l;(m)}, \text{ for } l \in \mathcal{N}_k\} \cup \{\mathbf{\Theta}, \mathbf{\Delta}\}$, then, the privacy metric (66) at each agent l is obtained as

$$E^{\mathrm{DNI}}_{l;(m)} = \mathrm{trace}\Big((\mathbf{c}^{\mathrm{T}}_{l} \otimes \mathbf{I}_L)\mathbf{P}^{\mathrm{DNI}}_{(m)}(\mathbf{c}_l \otimes \mathbf{I}_L)\Big), \tag{74}$$

where $\mathbf{P}^{\mathrm{DNI}}_{(m)}$ is the associated error covariance to the ML estimator at the HBC agent to estimate the private information $\mathbf{h}_{l;(0)}$, for $l \in \mathcal{N}\setminus\{k\}$.

Proof: The proof begins by stating that the available information set at the HBC agent can be represented as $\{\alpha_{k;(m)}, \mathbf{S}_{n;(m)}, \mathbf{S}_{n;(m)}\mathbf{e}_{l;(m)}, \text{ for } l \in \mathcal{N}_k\} \cup \{\mathbf{\Theta}, \mathbf{\Delta}\}$. On the other hand, using (19), the network-level consensus operation in decomposition and noise injection-based PPDL can be recursively computed as

$$
\begin{aligned}
{}_{(m)} = &\prod_{i=0}^{m} \mathcal{G}_{n;(i)} \; {}_{(0)} + \mathbf{C}_{n;(m)}\, \omega_{(m)} \\
&+ \sum_{i=0}^{m-1} \prod_{j=i}^{m-1} \mathcal{G}_{n;(j+1)} \; \mathbf{C}_{n;(i)}\, \omega_{(i)} \qquad (75)
\end{aligned}
$$

where ${}_{(m)} \; , \; [\,{}^{\mathrm{T}}_{(m)}; \; {}^{\mathrm{T}}_{(m)}]^{\mathrm{T}}$.

Since the HBC agent already knows its own information and is only interested in estimating the private information of other agents, we reduce the dimension of the state parameters by removing the corresponding entries of the HBC agent from quantities of $\mathcal{G}_{n;(m)}$, $\mathbf{C}_{n;(m)}$, ${}_{(m)}$, ${}_{(m)}$, and ${}_{(m)}$ and use $\check{\mathcal{G}}_{n;(m)}$and $\check{\mathbf{C}}_{n;(m)}$, $\check{}_{(m)}$, $\check{}_{(m)}$, and $\check{}_{(m)}$, respectively. Following the same procedure as in the noise-injection-based PPDL method, the reduced version of the network-level consensus operation in the decomposition and noise injection-based PPDL, as in (75), can be stated as

$$
\begin{aligned}
\check{}_{(m)} = &\prod_{i=0}^{m} \check{\mathcal{G}}_{n;(i)} \; \check{}_{(0)} + \check{\mathbf{C}}_{n;(m)}\, \check{\omega}_{(m)} \\
&+ \sum_{i=0}^{m-1} \prod_{j=i}^{m-1} \check{\mathcal{G}}_{n;(j+1)} \; \check{\mathbf{C}}_{n;(i)}\, \check{\omega}_{(i)} \qquad (76)
\end{aligned}
$$

where $\check{}_{(m)} \; , \; [\,\check{}^{\mathrm{T}}_{(m)}; \; \check{}^{\mathrm{T}}_{(m)}]^{\mathrm{T}}$. Without loss of generality, we consider the case where agent K is an HBC agent. At the HBC agent, let ${}^{0}_{(m)} = \mathbf{C}^{0}\check{}_{(m)}$ be the observation vector that comprises of the information captured at mth consensus iteration. Thus, the matrix $\mathbf{C}^{0}$ is defined as $\mathbf{C}^{0} \; , \; [\mathbf{C}; \mathbf{0}]$ and contains two blocks: the first block captures the public substates of the neighbors, and the second block shows that it does not have access to the private substates. Then, following the similar procedure as stated in the noise injection-based PPDL method and after substituting (9) in (76), the observation model at the HBC agent can be described as

$$
\begin{aligned}
{}^{0}_{(m)} &= \mathbf{C}^{0}\check{}_{(m)} \\
&= \mathbf{C}^{0} \prod_{i=0}^{m} \check{\mathcal{G}}_{n;(i)} \; \check{}_{(0)} + {}^{m}\mathbf{C}^{0}\check{\mathbf{C}}_{n;(m)}\, \check{}_{(m)} \qquad (77) \\
&+ \mathbf{C}^{0} \sum_{i=0}^{m-1} {}^{i} \prod_{j=i+2}^{m} \check{\mathcal{G}}_{n;(j)} \; (\check{\mathcal{G}}_{n;(i+1)}\check{\mathbf{C}}_{n;(i)} - \check{\mathbf{C}}_{n;(i+1)})\, \check{}_{(i)}
\end{aligned}
$$

Subsequently, by collecting all the observed information up to consensus iteration m, the overall observation vector can be stated as

$$
{}^{0}_{(m)} = \bar{\mathcal{H}}'_{(m)}\check{}_{(0)} + \mathbf{F}^{0}_{(m)} \; {}_{(m)}; \qquad (78)
$$

with

$$
\begin{aligned}
{}^{0}_{(m)} \; &, \; \mathrm{col}\{\,{}^{0}_{(0)}; \; {}^{0}_{(1)}; \ldots; \; {}^{0}_{(m)}\} \\
\bar{\mathcal{H}}'_{(m)} \; &, \; \mathrm{col}\{\mathcal{H}'_{(0)}; \mathcal{H}'_{(2)}; \ldots; \mathcal{H}'_{(m)}\}:
\end{aligned}
$$

where $\mathcal{H}'_{(m)} = \mathbf{C}^{0} \prod_{i=0}^{m} \check{\mathcal{G}}_{n;(i)}$, and

$$
\mathbf{F}^{0}_{(m)} = (\mathbf{I}_{m+1} \quad \mathbf{C}^{0})\, \mathbf{F}^{0}_{(m)};
$$

with

$$
\mathbf{F}^{0}_{(m)} = \begin{bmatrix}
\check{\mathbf{C}}_{n;(0)} & \mathbf{0} & \mathbf{0} & & \mathbf{0} \\
\mathbf{F}^{0}_{(1);(0)} & \check{\mathbf{C}}_{n;(1)} & \mathbf{0} & & \mathbf{0} \\
\mathbf{F}^{0}_{(2);(0)} & \mathbf{F}^{0}_{(2);(1)} & {}^{2}\check{\mathbf{C}}_{n;(2)} & & \mathbf{0} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\mathbf{F}^{0}_{(m);(0)} & \mathbf{F}^{0}_{(m);(1)} & {}^{2}\mathbf{F}^{0}_{(m);(2)} & & {}^{m}\check{\mathbf{C}}_{n;(m)}
\end{bmatrix};
$$

where

$$
\mathbf{F}^{0}_{(m);(i)} = \prod_{j=i+2}^{m} \check{\mathcal{G}}_{n;(j)}(\check{\mathcal{G}}_{n;(i+1)}\check{\mathbf{C}}_{n;(i)} - \check{\mathbf{C}}_{n;(i+1)}):
$$

Using the overall observation model in (78); the HBC agent can find an ML estimate of the private information of agents $\check{\mathbf{h}}_{(0)} = \frac{1}{2}\bar{\mathbf{I}}\check{}_{(0)}$ with $\bar{\mathbf{I}} = [\mathbf{I}; \mathbf{I}]$ and $\check{}_{(0)} = [\,\check{}^{\mathrm{T}}_{(0)}; \; \check{}^{\mathrm{T}}_{(0)}]^{\mathrm{T}}$. Thus, the associated error covariance to estimate private information of agents $\check{\mathbf{h}}_{(0)}$ can be expressed as

$$
\mathbf{P}^{\mathrm{DNI}}_{(m)} = \frac{1}{4}\bar{\mathbf{I}} \left(\bar{\mathcal{H}}'^{\mathrm{T}}_{(m)} \left(\mathbf{F}^{0}_{(m)}\boldsymbol{\Lambda}_{\;;(m)}\mathbf{F}^{0\,\mathrm{T}}_{(m)} \right)^{-1} \bar{\mathcal{H}}'_{(m)} \right)^{-1} \bar{\mathbf{I}}^{\mathrm{T}}: \qquad (79)
$$

As the HBC agent collects more information from neighbors, the mean squared error of the ML estimator decreases and the privacy metric (66) at each agent l can be obtained as

$$
E^{\mathrm{DNI}}_{l;(m)} = \mathrm{trace}\left((\mathbf{c}_l^{\mathrm{T}} \quad \mathbf{I}_L)\mathbf{P}^{\mathrm{DNI}}_{(m)}(\mathbf{c}_l \quad \mathbf{I}_L) \right); \qquad (80)
$$

which completes the proof. ■

B. External Eavesdropper

The external eavesdropper is an adversary outside the network that knows the network topology and can access the information exchanged between agents. Therefore, it generally produces a more accurate estimate of the private information [24]. In the proposed communication-efficient and privacy-aware distributed learning strategies, however, the selection matrix and circular shift variable are invisible to external eavesdroppers since they are initialized during network establishment and are never shared during collaboration. Further, due to partial information sharing, an external eavesdropper can only access a portion of entries in the perturbed private information during each consensus m, while being unable to determine their position and the size of the private information. Thus, the proposed communication-efficient and privacy-preserving PPDL strategies are resilient against external eavesdroppers with no information leakage.

However, in the worst-case scenario, when the initial selection matrices, the right-circular shift parameter, and the size of the private information vector are accessible to the adversary, the external eavesdropper can construct an observer and estimate the private information. Besides the information mentioned above, i.e., $I_{\mathrm{EE}} = \{\mathbf{S}_{n;(m)}\tilde{\mathbf{h}}_{l;(m)}$ for $l \in N_g \cup \{\mathbf{S}_{n;(0)}; \; ; L\}$, the eavesdropper must also be able to access the consensus weights of the noise injection PPDL and the coupling and interaction weights of the decomposition and noise injection PPDL in order to complete the construction of the local observer and estimate the private information.

Remark 3: Assuming that an external eavesdropper has knowledge of the selection matrices, the circular shift parameter, the vector size of private information, consensus weights of the noise injection PPDL, and the coupling and interaction weights of the decomposition and noise injection PPDL is an unrealistic scenario in practice. Furthermore, conducting a comprehensive analysis of the eavesdropper's ability to accurately estimate private information through the construction of a local observer is beyond the scope of this work, but it will be addressed in future research.

VI. Numerical Simulations

To demonstrate the effectiveness of noise-injection- as well as decomposition- and noise-injection-based PPDL strategies, we conducted a series of simulations in the context of system identification. For this, we considered a random network of $K = 50$ agents, with topology shown in Fig. 1, with the goal of estimating an unknown system of length $L = 32$.

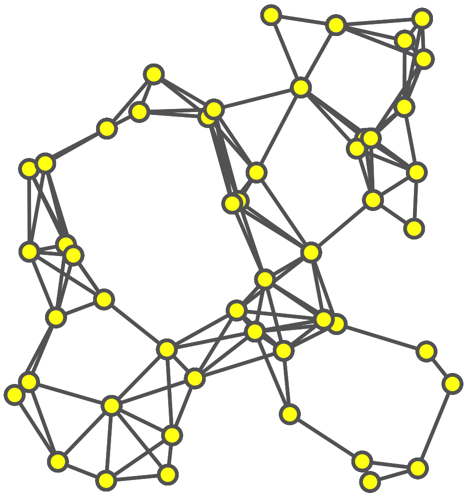
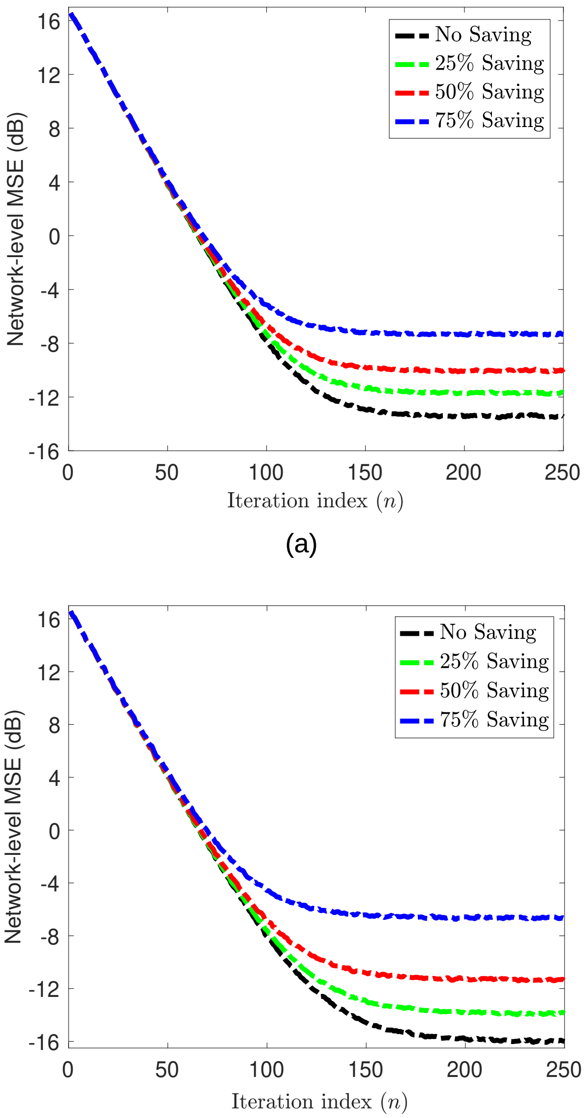


Fig. 1. Network topology.

The input signal $x_{k;n}$ and observation noise sequence $_{k;n}$, were drawn from zero-mean Gaussian distribution with variance $^{2}_{x;k} = 1$ and $^{2}_{;k}$ 2 $U(0{:}008; 0{:}03)$, respectively. Metropolis rule [47] was used to obtain non-negative coefficients a_{lk} in the average consensus operations of the noise injection-based PPDL. The interaction weights of the decomposition-based method were set as $\mathbf{\Delta} = 0{:}8\mathbf{E}$ where $\mathbf{E}$ denotes the adjacency matrix of the network. The elements of the coupling weight $_{k}$ were chosen independently from a distribution $U(;1)$ where $ = 0{:}8$ and we set $ = 0{:}9$. Average consensus operations for the proposed PPDL algorithms are iterated $M = 40$ times and the perturbation noise sequence at each agent follows (9). The proposed communication-efficient and privacy-preserving distributed learning strategies were simulated under coordinated partial-sharing scheme for different values of L^{0} (say $0{:}75L$, $0{:}5L$, $0{:}25L$, implying 25%, 50% and 75% communication saving). The network-level MSE, which is given by $\frac{1}{K}\mathrm{E}[\mathbf{e}_n^{\mathsf{T}}\mathbf{e}_n]$), was considered as the performance metric. The simulation results were obtained by averaging over 200 independent experiments.



(a)

(b)

Fig. 2. Learning curves of the proposed communication-efficient and privacy-preserving distributed learning strategies: (a). Noise-injection-based PPDL. (b). Decomposition- and noise-injection-based PPDL.

First, the proposed strategies were simulated for perturbation noise variance $^{2} = 5$ and the corresponding learning curves (i.e., network-level MSE in dB vs iteration index n) are shown in Fig. 2. From Fig. 2, we see that the proposed distributed learning strategies simultaneously achieve communication efficiency and preserve privacy at the cost of a slight degradation in performance. These strategies exhibit a tradeoff between communication-saving and estimation performance, i.e., as the communication-saving increases the estimation performance deteriorates. Even with 50% communication saving, the proposed strategies are able to achieve comparable performance with that of full communication case. It is also evident from Fig. 2 that the decomposition and noise injection-based PPDL exhibits better estimation performance than the noise injection-based PPDL because it injects perturbation noise only into a fraction of the public state, thus minimizing

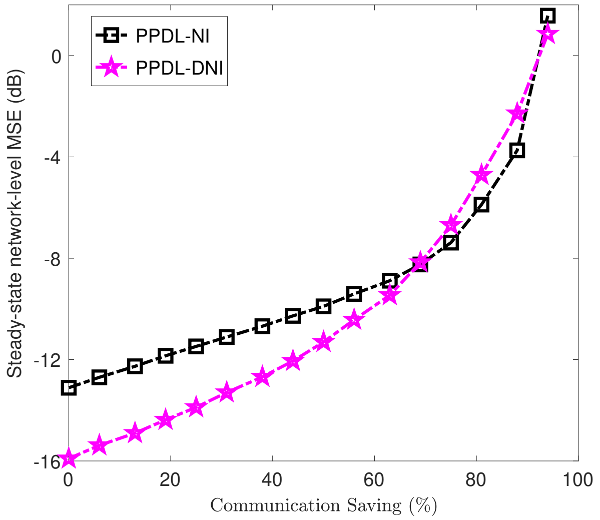
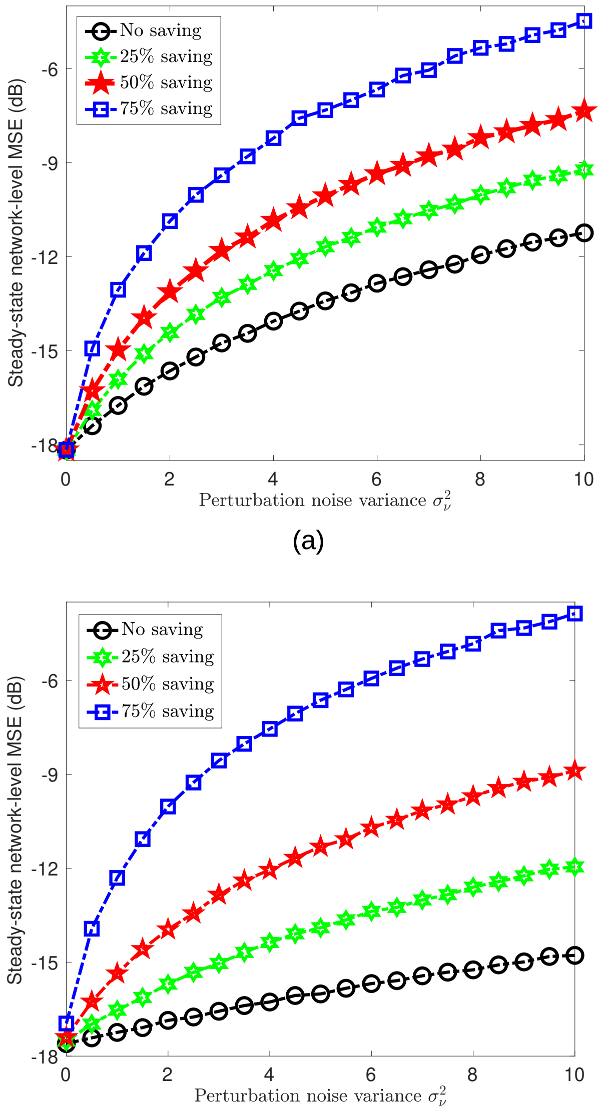


Fig. 3. Communication saving vs. steady-state network-level MSE for ^{2} = 5.



(a)

(b)

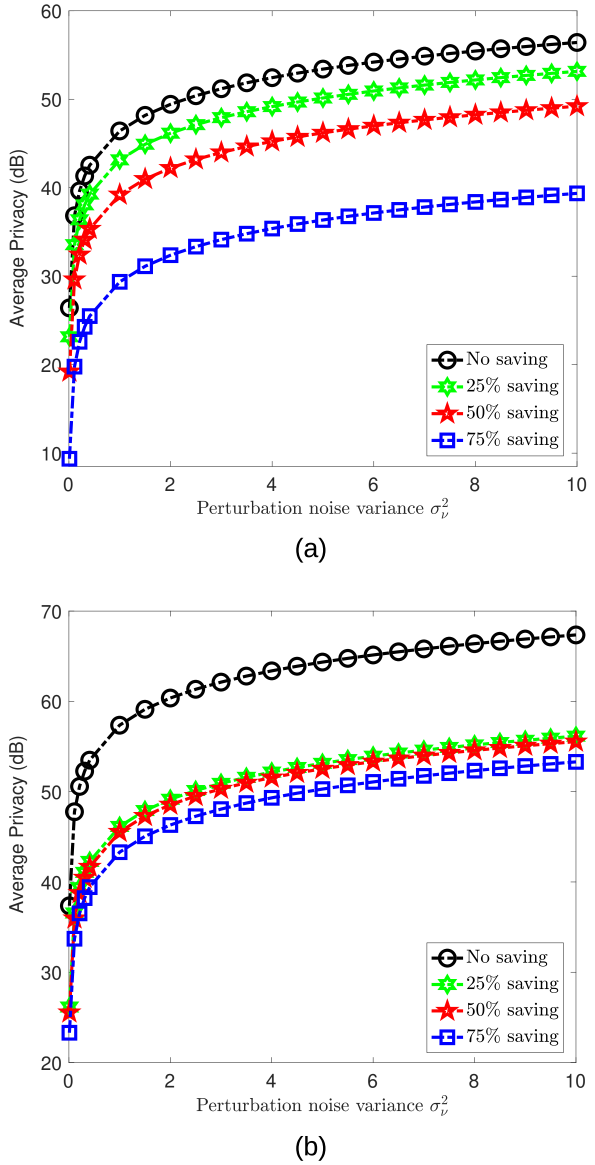
Fig. 4. Steady-state network-level MSE vs. perturbation noise variance: (a). Noise-injection-based PPDL. (b). Decomposition- and noise-injection-based PPDL.

the overall contamination of the private information.

To delve deeper into this, Fig. 3 illustrates the percentage of communication savings versus the steady-state mean square error (MSE) in decibels (dB) for the proposed strategies. Based on the findings depicted in Fig. 3, it is evident that the decomposition and noise injection-based PPDL approach outperforms the noise injection-based PPDL approach in terms of estimation performance up to a communication saving of 65%. However, beyond this point, both algorithms demonstrate similar performance.

Second, we examine the robustness of the proposed communication-efficient and privacy-preserving distributed learning strategies against the perturbation noise variance ^{2}, by varying ^{2} from 0 to 10. The steady-state network-level MSE of the proposed strategies against ^{2} is displayed in Fig. 4. From Fig. 4, we see that the performance of the proposed strategies deteriorates slightly as the variance of the perturbation noise increases. However, Fig. 4 also shows that the performance degradation is limited with the combined decomposition- and noise-injection-based PPDL when compared to the pure noise-injection-based PPDL. This behavior is due to the noise-free private substate. Noise injected with a higher variance degrades the learning performance regardless of the level of communication efficiency. However, the performance degradation is more pronounced when the communication savings are high. In other words, both proposed PPDL strategies become more sensitive to the perturbation noise variance when a smaller fraction of the information is shared at each time instant.

Third, we investigate the impact of communication savings on agent privacy in different distributed learning scenarios. In order to investigate the privacy performance of different algorithms, we use the average privacy of network agents, defined as E , $\frac{1}{K}$ P $^{K}_{l=1}$ ^{1} $E_{l;(m)}$ with $E_{l;(m)}$ in (66), as a privacy measure. The average privacy E of the proposed communication-efficient and privacy-preserving distributed learning strategies is demonstrated in Fig. 5 by varying the perturbation noise variance ^{2} from 0 to 10. From Fig. 5, we see that increasing the variance of the perturbation noise increases the average privacy regardless of the level of the communication savings in both noise injection-based PPDL and decomposition and noise injection-based PPDL. Fig. 5 also shows a tradeoff between communication savings and privacy in both noise-injection-based PPDL, and decomposition and noise injection-based PPDL, where sharing a smaller fraction of the information at each time instant results in a lower level of average privacy in the network. Even when a larger fraction of the information is shared at each iteration, the SNR at the HBC does not increase due to higher cumulative noise present in the elements of private information. This leads to higher estimation error at the HBC when it attempts to estimate the private information of a given agent, which ensures a higher level of privacy. From Fig. 5 (b), it is also evident that the decomposition and



(a)

(b)

Fig. 5. Average privacy versus perturbation noise variance: (a). Noise-injection-based PPDL. (b). Decomposition- and noise-injection-based PPDL.

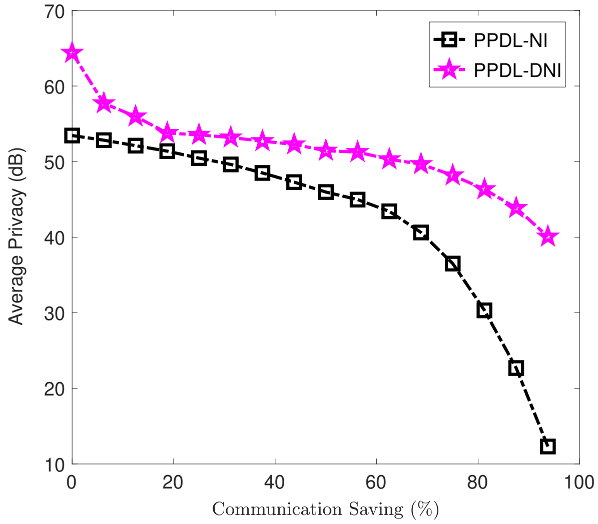
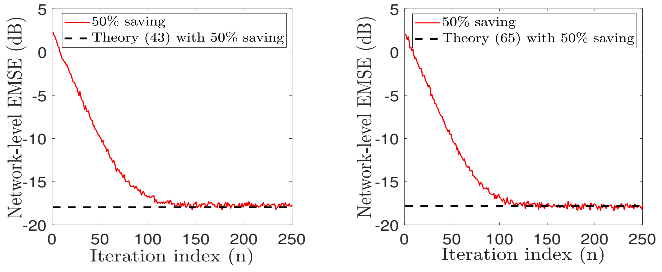


Fig. 6. Communication saving vs. average privacy for $^2 = 4$.



(a)

(b)

Fig. 7. Learning curves of the proposed PPDL strategies: (a). Noise-injection-based PPDL. (b). Decomposition- and noise-injection-based PPDL.

noise injection-based PPDL offers better privacy compared to the noise injection-based PPDL for a given value of ^{2}. Although privacy decreases with increased communication efficiency, the privacy achieved by decomposition and noise injection-based PPDL with 75% communication savings is better than the privacy achieved by noise injection-based PPDL without communication savings, which is again due to the noise-free private substate. The privacy-performance trade-off can be examined by analyzing plots in Fig. 4 and Fig. 5 to determine the perturbation level. Thus, the amount of the injected noise can be established based on the sought privacy level, Fig. 5, and tolerable performance degradation, Fig. 4, for the application at hand.

To further investigate the impact of communication savings on privacy, Fig. 6 illustrates the percentage of communication savings versus the average privacy of agents in dB for proposed PPDL strategies using the perturbation noise variance $^2 = 4$. It can be seen that the decomposition and noise injection-based PPDL approach outperforms the noise injection-based PPDL approach in terms of the average privacy provided. Furthermore, Fig. 6 illustrates that in both strategies, increasing the communication-saving percentage decreases the obtained average privacy, indicating a tradeoff between communication savings and privacy that must be taken into account based on the particular application.

Finally, in order to determine the accuracy of the analytical MSE expressions in (43) and (65), we plotted these equations alongside their numerical equivalents. Due to limited resources (i.e., memory and hardware limitations), the comparison results are shown for $K = 20$ and $L = 4$ while the remaining parameters are the same as the first experiment. Figs. 7 illustrate the learning curves of the proposed strategies when $L^{0} = 2$ (i.e., 50% saving). We see that the theoretical results match the simulation results. Due to the small parameter vector length L, both methods perform similarly in this experiment.

VII. Conclusions

This paper proposed partial-sharing private distributed learning (PPDL) algorithms that offer communication efficiency while preserving privacy. The proposed noise-injection-based PPDL achieved communication efficiency and privacy by allowing each agent to share only a fraction of perturbed private information among neighbors. On the other hand, the decomposition- and noise-injection-based PPDL randomly decomposed the local information into public and private substates and partially shared the perturbed version of the public information to achieve both communication efficiency and privacy. Mean and mean-square convergence analyses were conducted to determine the impact of communication savings and privacy preservation on the performance of the proposed PPDL algorithms. The agent privacy was characterized in the presence of an HBC adversary, and the impact of the communication saving on privacy was also analyzed. Analytical results showed that agent privacy is improved under the decomposition and noise injection PPDL, and communication efficiency in the proposed PPDL algorithms is achieved at the price of learning performance and privacy. Numerical simulations validated the analytical findings and showed that the decomposition and noise injection PPDL with 50% communication savings achieved nearly the same learning performance and improved privacy compared to the noise injection PPDL without communication savings.

References

[1] A. H. Sayed, "Adaptive Networks," *Proc. IEEE,* vol. 102, no. 4, pp. 460-497, Apr. 2014.

[2] M. Chen, D. Gündüz, K. Huang, W. Saad, M. Bennis, A. V. Feljan and H. V. Poor, "Distributed learning in wireless networks: Recent progress and future challenges," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 12, pp. 3579-3605, Dec. 2021.

[3] T.H. Chang, M. Hong, H.T. Wai, X. Zhang and S. Lu, "Distributed learning in the nonconvex world: From batch data to streaming and beyond," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 26–38, May 2020.

[4] Q. Li, J. S. Gundersen, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed processing: Metrics, bounds and algorithms," *IEEE Trans. Inf. Forensics Security.*, vol. 16, no. 3, pp. 2090–2103, Jan. 2021.

[5] J. He, L. Cai and X. Guan, "Differential private noise adding mechanism and its application on consensus algorithm," *IEEE Trans. Signal Process.*, vol. 68, pp. 4069-4082, Jul. 2020.

[6] N. K. D. Venkategowda and S. Werner, "Privacy-preserving distributed maximum consensus," *IEEE Signal Process. Lett.*, vol. 27, no. 10, pp. 1839-1843, Oct. 2020.

[7] Q. Li, R. Heusdens and M. G. Christensen, "Privacy-preserving distributed optimization via subspace perturbation: A general framework," *IEEE Trans. Signal Process.*, vol. 68, no. 10, pp. 5983-5996, Oct. 2020.

[8] Z. Huang, S. Mitra and N. Vaidya, "Differentially private distributed optimization," in *Proc. Int. Conf. Distributed Comput. and Netw.*, 2015, pp. 1–10.

[9] Q. Li, M. Coutino, G. Leus and M. G. Christensen, "Privacy-preserving distributed graph filtering," in *Proc. European. Signal Process. Conf.*, 2021, pp. 2155-2159.

[10] Y. Mo and B. Sinopoli, "Secure estimation in the presence of integrity attacks," *IEEE Trans. Autom. Control*, vol. 60, no. 4, pp. 1145-1151, Apr. 2015.

[11] Z. Guo, D. Shi, D. E. Quevedo and L. Shi, "Secure state estimation against integrity attacks: A Gaussian mixture model approach," *IEEE Trans. Signal Process.*, vol. 67, no. 1, pp. 194–207, Jan. 2019.

[12] L. Su and S. Shahrampour, "Finite-time guarantees for Byzantine-resilient distributed state estimation with noisy measurements," *IEEE Trans. Autom. Control*, vol. 65, no. 9, pp. 3758–3771, Sep. 2020.

[13] M. Ruan, H. Gao and Y. Wang, "Secure and privacy-preserving consensus," *IEEE Trans. Autom. Control*, vol. 64, no. 10, pp. 4035-4049, Oct. 2019.

[14] C. Zhang, M. Ahmad and Y. Wang, "ADMM based privacy-preserving decentralized optimization," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 565-580, Mar. 2019.

[15] Y. Ni, J. Wu, L. Li and L. Shi, "Multi-party dynamic state estimation that preserves data and model privacy," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 2288-2299, Jan. 2021.

[16] X. Yin, Y. Zhu and J. Hu,"A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions," *ACM Comp. Surveys.*, vol. 54, no. 6, pp. 1-36, Jul. 2021.

[17] R. L. Lagendijk, Z. Erkin and M. Barni, "Encrypted signal processing for privacy protection: Conveying the utility of homomorphic encryption and multiparty computation," *IEEE Signal Process. Mag.*, vol. 30, no. 1, pp. 82-105, Jan. 2013.

[18] I. Damgård, V. Pastro, N. Smart and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Proc. Annu. Cryptology Conf.*, 2012, pp. 643-662.

[19] Y. Mo and R. M. Murray, "Privacy preserving average consensus," *IEEE Trans. Autom. Control*, vol. 62, no. 2, pp. 753–765, Feb. 2017.

[20] J. He, L. Cai and X. Guan, "Preserving data-privacy with added noises: Optimal estimation and privacy analysis," *IEEE Trans. Inf. Theory*, vol. 64, no. 8, pp. 5677-5690, Aug. 2018.

[21] J. He, L. Cai, C. Zhao, P. Cheng and X. Guan, "Privacy-preserving average consensus: Privacy analysis and algorithm design," *IEEE Trans. Signal Inf. Process. Netw.*, vol. 5, no. 1, pp. 127-138, Mar. 2019.

[22] T. Liu, B. Di, B. Wang and L. Song, "Loss-Privacy tradeoff in federated edge learning," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 3, pp. 546-558, Apr. 2022.

[23] E. Rizk, S. Vlaski and A. H. Sayed, "Enforcing privacy in distributed learning with performance guarantees," *arXiv preprint*, arXiv:2301.06412, Jan. 2023.

[24] Y. Wang, "Privacy-preserving average consensus via state decomposition," *IEEE Trans. Autom. Control*, vol. 64, no. 11, pp. 4711-4716, Nov. 2019.

[25] K. Zhang, Z. Li, Y. Wang, A. Louati and J. Chen, "Privacy-preserving dynamic average consensus via state decomposition: Case study on multi-robot formation control," *Automatica*, vol. 139, no. 1, pp. 110182, May 2022.

[26] W. Wang, D. Li, X. Wu and S. Xue, "Average consensus for switching topology networks with privacy protection," in *Proc. Chinese Automat. Congr.*, 2019, pp. 1098-1102.

[27] A. Moradi and N. K. D. Venkategowda and S. P. Talebi and S. Werner, "Privacy-Preserving Distributed Kalman Filtering," *IEEE Trans. Signal Process.*, vol. 70, pp. 3074-3089, Jun. 2022.

[28] H. Gao, Z. Li and Y. Wang, "Privacy-preserving collaborative estimation for networked vehicles with application to collaborative road profile estimation," *IEEE Trans. Intell. Transp.*, vol. 23, no. 10, pp. 17301-17311, Oct. 2022.

[29] S. Bandyopadhyay and E. J. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *Proc. Conf. Comput. Commun.*, 2003, pp. 1713-1723.

[30] R. S. Blum and B. M. Sadler, "Energy efficient signal detection in sensor networks using ordered transmissions," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3229-3235, Jul. 2008.

[31] R. Arablouei, S. Werner, K. Doğançay and Y-F. Huang, "Analysis of a reduced-communication diffusion LMS algorithm," *Signal Process.*, vol. 117, pp. 355-361, Dec. 2015.

[32] D. Bajovic, B. Sinopoli and J. Xavier, "Sensor selection for event detection in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 10, pp. 4938-4953, Oct. 2011.

[33] C. G. Lopes and A. H. Sayed, "Diffusion adaptive networks with changing topologies," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2008, pp. 3285-3288.

[34] D. Berberidis, V. Kekatos and G. B. Giannakis, "Online censoring for large-scale regressions with application to streaming big data," *IEEE Trans. Signal Process.,* vol. 64, no. 15, pp. 3854-3867, Aug. 2016.

[35] A. Ribeiro, G. B. Giannakis and S. I. Roumeliotis, "SOI-KF: Distributed Kalman filtering with low-cost communications using the sign of innovations," *IEEE Trans. Signal Process.*, vol. 54, no. 12, pp. 4782-4795, Dec. 2006.

[36] E. J. Msechu and G. B. Giannakis, "Sensor-centric data reduction for estimation with WSNs via censoring and quantization," *IEEE Trans. Signal Process.,* vol. 60, no. 1, pp. 400-414, Jan. 2012.

[37] S. Chouvardas, K. Slavakis and S. Theodoridis, "Trading off complexity with communication costs in distributed adaptive learning via Krylov subspaces for dimensionality reduction," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 2, pp. 257-273, Apr. 2013.

[38] S. Xie and H. Li, "Distributed LMS estimation over networks with quantized communications," *Int. J. Control*, vol. 86, no. 3, pp. 478-492, Apr. 2013.

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

[40] X. Wu, X. Yao and C-L. Wang, "FedSCR: Structure-based communication reduction for federated learning," *IEEE Trans. Parallel Distrib. Syst.*, vol. 32, no. 7, pp. 1565-1577, Jul. 2021.

[41] R. Arablouei, S. Werner, Y. F. Huang and K. Doğançay, "Distributed least mean-square estimation with partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 2, pp. 472-484, Jan. 2013.

[42] R. Arablouei, K. Doğançay, S. Werner and Y. F. Huang, "Adaptive distributed estimation based on recursive least-squares and partial diffusion," *IEEE Trans. Signal Process.*, vol. 62, no. 14, pp. 3510-3522, Jul. 2014.

[43] V. C. Gogineni and M. Chakraborty, "Partial diffusion affine projection algorithm over clustered multitask networks," in *Proc. IEEE Int. Symp. Circuits and Syst.*, 2019, pp. 1-5.

[44] V. C. Gogineni, S. Werner, Y-F. Huang and A. Kuh, "Communication-efficient online federated learning strategies for kernel regression," *IEEE Internet of Things J.*, vol. 10, no. 5, pp. 4531-4544, Mar. 2023.

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

[46] I. D. Schizas, G. Mateos and G. B. Giannakis, "Distributed LMS for consensus-based in-network adaptive processing," *IEEE Trans. Signal Process.*, vol. 57, no. 6, pp. 2365-2382, Jun. 2009.

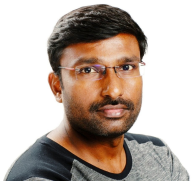
[47] L. Xiao, S. Boyd and S. Lall, "A scheme for robust distributed sensor fusion based on average consensus," in *Proc. Int. Conf. Info. Process. in Sensor Networks*, 2005, pp. 63–70.

[48] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. & Control Lett.*, vol. 53, no. 1, pp. 65-78, 2004.

[49] A. H. Sayed, *Adaptation, Learning, and Optimization over Networks,* Boston-Delft: NOW, 2014.

[50] A. H . Sayed, "Diffusion adaptation over networks," in *Academic Press Library in Signal Processing,* R. Chellappa and S. Theodoridis, Eds., pp. 322-454, Elsevier, 2013. Also available as arXiv:1205.4220 [cs.MA], May 2012.

[51] R. H. Koning, H. Neudecker and T. Wansbeek, "Block Kronecker products and the vecb operator," *Linear Algebra and its Applications*, vol. 149, pp. 165-184, 1991.



Vinay Chakravarthi Gogineni (Senior Member, IEEE) received the Bachelor's degree in electronics and communication engineering from Jawaharlal Nehru Technological University, Andhra Pradesh, India, in 2005, the Master's degree in communication engineering from VIT University, India, in 2008, and the Ph.D. degree in electronics and electrical communication engineering from Indian Institute of Technology Kharagpur, India in 2019. Currently, he is an Assistant Professor at SDU Applied AI and Data Science, The Maersk Mc-Kinney Moller Institute, University of Southern Denmark. Prior to this, he worked as a postdoctoral research fellow at NTNU and Simula, Norway. From 2008 to 2011, he was with a couple of MNCs in India. His research interests include statistical signal processing, distributed machine learning, geometric deep learning, and their application in healthcare. He was a recipient of the ERCIM Alain Bensoussan Fellowship in 2019 and the Best Paper Award at APSIPA ASC-2021, Tokyo, Japan. He is a member of the editorial board for the IEEE Sensors Journal.

Ashkan Moradi (Member, IEEE) holds an M.Sc. degree in Telecommunication Networks from the University of Tehran (2016) and a Ph.D. degree in "Distributed Learning and Estimation with Enhanced Privacy and Security" from the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU, 2023). Presently, he holds a position as a Postdoctoral Researcher at the Department of Circulation and Medical Imaging at NTNU. His research proficiency spans across areas such as Federated Learning, with a specific emphasis on cancer detection and organ segmentation, as well as distributed learning and estimation, with a primary focus on information privacy.

Naveen K. D. Venkategowda (S'12–M'17) received the B.E. degree in electronics and communication engineering from Bangalore University, Bengaluru, India, in 2008, and the Ph.D. degree in electrical engineering from Indian Institute of Technology, Kanpur, India, in 2016. He is currently an Universitetslektor at the Department of Science and Technology, Linköping University, Sweden. From Oct. 2017 to Feb. 2021, he was postdoctoral researcher at the Department of Electronic Systems, Norwegian University of Science and Technology, Trondheim, Norway. He was a Research Professor at the School of Electrical Engineering, Korea University, South Korea from Aug. 2016 to Sep. 2017. He was a recipient of the TCS Research Fellowship (2011-15) from TCS for graduate studies in computing sciences and the ERCIM Alain Bensoussan Fellowship in 2017.

Stefan Werner (Fellow, IEEE) received the M.Sc. Degree in electrical engineering from the Royal Institute of Technology, Stockholm, Sweden, in 1998, and a D.Sc. degree (Hons.) in electrical engineering from the Signal Processing Laboratory, Helsinki University of Technology, Espoo, Finland, in 2002. He is a Professor at the Department of Electronic Systems, Norwegian University of Science and Technology (NTNU), Director of IoT@NTNU, and Adjunct Professor at Aalto University in Finland. He was a visiting Melchor Professor with the University of Notre Dame during the summer of 2019 and an Adjunct Senior Research Fellow with the Institute for Telecommunications Research, University of South Australia, from 2014 to 2020. He held an Academy Research Fellowship, funded by the Academy of Finland, from 2009 to 2014. His research interests include adaptive and statistical signal processing, wireless communications, and security and privacy in cyber-physical systems. He is a member of the editorial boards for the EURASIP Journal of Signal Processing and the IEEE Transactions on Signal and Information Processing over Networks.