

Online Proximal Learning over Jointly Sparse Multitask Networks With L∞,1 Regularization

Danqi Jin, Jie Chen, Cédric Richard, Jingdong Chen

▶ **To cite this version:**

Danqi Jin, Jie Chen, Cédric Richard, Jingdong Chen. Online Proximal Learning over Jointly Sparse Multitask Networks With L∞,1 Regularization. IEEE Transactions on Signal Processing, 2020, 68, pp.6319-6335. 10.1109/TSP.2020.3021247 . hal-03347269

HAL Id: hal-03347269
https://hal.science/hal-03347269

Submitted on 17 Sep 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Online Proximal Learning Over Jointly Sparse Multitask Networks With $\ell_{\infty,1}$ Regularization

Danqi Jin, *Student Member, IEEE*, Jie Chen, *Senior Member, IEEE*, Cédric Richard, *Senior Member, IEEE*, Jingdong Chen, *Senior Member, IEEE*

Abstract

Modeling relations between local optimum parameter vectors to estimate in multitask networks has attracted much attention over the last years. This work considers a distributed optimization problem with jointly sparse structure among nodes, that is, the local solutions have the same sparse support set. Several mixed norm have been proposed to address the jointly sparse structure in the literature. Among several candidates, the (reweighted) $\ell_{\infty,1}$-norm is element-wise separable, it is more convenient to evaluate their approximate proximal operators. Thus by introducing a (reweighted) $\ell_{\infty,1}$-norm penalty term at each node, and using a proximal gradient method to minimize the regularized cost, we devise a proximal multitask diffusion LMS algorithm which can promote joint-sparsity. Analyses are provided to characterize the algorithm behavior in the mean and mean-square sense. Simulation results are presented to show its effectiveness, as well as the accuracy of the theoretical findings.

Index Terms

Distributed optimization, diffusion strategy, joint sparsity, proximal algorithm, stochastic performance, (reweighted) $\ell_{\infty,1}$-norm.

I. Introduction

Because of their superior performance and wider stability range [1], diffusion strategies have been widely used in multi-agent networks to address estimation problems in a distributed and online manner. Several diffusion strategies have been introduced, and their performance analyzed in various situations, such as the diffusion LMS [2], RLS [3], and APA [4], as well as several of their variants [5], [6].

By referring to estimating an optimal parameter vector at a node as a task, and according to the relations between the optimal parameter vectors over the entire network, diffusion networks are further divided into single-task and multitask networks. In single-task networks, all nodes estimate the same parameter vector. Typical works related to single-task networks include [7]–[11]. With multitask networks, multiple but related parameter vectors are inferred simultaneously in a cooperative manner, so as to improve the estimation accuracy by exploiting the similarities between tasks. These similarities can be promoted with appropriate regularization terms. Squared ℓ_2-norm regularization is used in [12], and ℓ_1-norm regularization is considered in [13], [14]. For the latter, a subgradient and a proximal algorithm are introduced in [13] and [14], respectively. In [15] and [16], the authors derive solutions for other classes of multitask problems where the relations between the nodes are defined by common latent representations or local linear equality constraints, respectively. In [17], the authors solve a multitask problem by estimating the combination matrix. In [18], the authors address multitask problems over asynchronous networks and carry out a detailed theoretical analysis. In [19], the performance of multitask diffusion networks is analyzed for correlated noise and regressors. In [20], the authors propose a combination framework that aggregates several diffusion strategies. All the algorithms cited above are based on the diffusion LMS. In [21], the authors extend the diffusion APA to multitask framework in order to improve the robustness against correlated regressors. In [22], the authors improve the performance of the multitask diffusion APA via controlled inter-cluster cooperation. In [23], the authors propose a clustered multitask partial diffusion APA that transmits only a subset of the entries of the intermediate estimates, to provide a trade-off between the estimation performance and communication cost.

Multitask learning considerably enriches the possibilities of diffusion networks. Beyond the few examples listed above, there are also applications where the optimal parameter vectors have a jointly sparse structure, namely, local solutions have the same sparse support. Applications include, for instance, distributed spectrum sensing and channel identification in underwater and wireless communication networks with multiple sensors [24]. In spectrum sensing of sparse wide-band spectra in distributed wireless sensor networks, the parameter vectors at different nodes share the same intrinsic sparse structure. However, due to different channel fading effects, shadowing effects and transmission losses, the values at the nonzero entries of these parameter vectors are different [24]–[26]. For underwater and wireless communication networks, it has been shown that real-world underwater channels [27] and wireless communication channels [28] are inherently sparse with large delay spread. In addition,

The work of Jie Chen was supported in part by NSFC grant 61671382. The work of Jingdong Chen was supported in part by NSFC grant 61425005. The work of C. Richard work was supported through the UCA JEDI Investments in the Future project with the reference number ANR-15-IDEX-0001.
D. Jin, J. Chen and J. Chen are with Centre of Intelligent Acoustics and Immersive Communications at School of Marine Science and Technology, Northwestern Polytechnical University, Xi'an, China (danqijin@mail.nwpu.edu.cn, dr.jie.chen@ieee.org, jingdongchen@ieee.org). C. Richard is with Université Côte d'Azur, OCA, CNRS, France (cedric.richard@unice.fr).

the channel supports for neighboring antennas or nodes are approximately the same [28]. Indeed, times of arrival for closely spaced nodes and antennas are quite close, though the tap weights are actually different [24]. For more detailed information about this application, see Section VI-C. Taking jointly sparse structure into consideration can greatly improve the network performance.

Several works have been proposed to address problems with jointly sparse structure over diffusion networks. In [29], the authors consider the mixed $\ell_{2,0}$-norm. In [24], the authors devise an algorithm with regularizers such as $\ell_{2,0}$, $\ell_{2,1}$ and reweighted $\ell_{2,1}$ regularizers. They also conduct theoretical analyses of the algorithm behavior in the mean and mean-square sense. However, studies in [24], [29] are based on gradient-descent schemes, and use subgradient for non-differential regularization terms. For subgradient approaches, the subdifferential at a point may not be a singleton set, that is, it may be empty or consist of several elements. As a result, one may get stuck or have to choose one, respectively. Furthermore, even if the subdifferential is a singleton at each step, it might be highly discontinuous, so small deviations might lead to a singular behavior of the algorithm over iterations [30]. In this paper, we propose to use proximal operators to address the jointly sparse estimation problem and then avoid the weaknesses of subgradient-based approaches mentioned above. Proximal algorithms result in subproblems that often admit closed-form solutions or that can be solved efficiently with simple specialized methods [31]. In addition, they guarantee better convergence rates and stability than subgradient approaches [30], [32]. For single-task learning problems, proximal algorithms were first introduced in [33] to estimate sparse parameter vectors. They were used in [34] and [35] to optimize general stochastic costs with non-differential regularizers, and non-smooth regularizers, respectively. For multitask learning problems, related works only include [14]. In that paper, the authors consider the situation where the network is divided into several clusters. All nodes in a cluster are interested in estimating the same parameter vector, while nodes in adjacent clusters estimate parameter vectors that have a large number of similar entries. As a result the authors derive a closed-form proximal solution for an ℓ_1-norm regularizer that promotes similarities among clusters. A theoretical analysis of the steady-state behavior is provided. The algorithm in [14] is based on an extra exchange step of observations between neighboring nodes within a same cluster as well as a fusion step that averages local estimates. The proximal algorithm derived in the current work does not include these two steps.

In this paper, we introduce a proximal diffusion LMS strategy for multitask networks with jointly sparse structure. The main contributions of this work are summarized as follows:

- We derive approximate closed-form expressions for the $\ell_{\infty,1}$-norm and reweighted $\ell_{\infty,1}$-norm proximal operators.
- We derive a proximal multitask diffusion LMS algorithm to solve problems with jointly sparse structure.
- We conduct a theoretical analysis of the algorithm performance, including a condition for stability and a study of its transient behavior in the mean sense.

Notation. Normal font x, boldface small letters $\boldsymbol{x}$ and capital letters $\boldsymbol{X}$ denote scalars, column vectors and matrices, respectively. Symbol $[\,\cdot\,]_m$ denotes the m-th entry of its vector argument. The superscript $(\cdot)^\top$ denotes the transpose operator. The mathematical expectation is denoted by $\mathbb{E}\{\cdot\}$. The Gaussian distribution with mean μ and variance σ^2 is denoted by $\mathcal{N}(\mu,\sigma^2)$. Operator $|\cdot|$ takes the absolute value of its scalar or vector argument. Operator $\max\{\cdot,\cdot\}$ extracts the maximum value of its two arguments. Operator $\mathrm{diag}\{\cdot\}$ generates a diagonal matrix from its argument. Symbol $\preceq$ denotes a component-wise inequality. The symbol $\otimes$ denotes the Kronecker product. The set $\mathcal{N}_k$ denotes the neighbors of node k, including k itself, and $|\mathcal{N}_k|$ denotes its cardinality. The $\mathcal{N}_k^-$ denotes the neighbors of node k, excluding node k. Vector $\mathbb{1}_L$ is the all-one vector of dimension $L\times 1$.

II. Problem Formulation

Consider a connected network consisting of N nodes. Each node has access to streaming data $\{d_{k,n},\boldsymbol{x}_{k,n}\}$, where $\boldsymbol{x}_{k,n}$ is the $L\times 1$ real-valued regression vector at node k and time instant n, and $d_{k,n}$ denotes the observed real-valued signal. We assume that the data at each agent k and time instant n are related via the linear model:

$$d_{k,n}=\boldsymbol{x}_{k,n}^\top\boldsymbol{w}_k^\star+z_{k,n}, \tag{1}$$

where $\boldsymbol{w}_k^\star\in\mathbb{R}^{L\times 1}$ is the unknown system vector to estimate, and $z_{k,n}$ is a zero-mean additive noise. We assume that $z_{k,n}$ is independent of any other signal. Further, we assume that vectors $\{\boldsymbol{w}_k^\star\}_{k=1}^N$ are jointly sparse, namely, not only each $\boldsymbol{w}_k^\star$ is a sparse vector but, in addition, they all have the same support. The support of $\boldsymbol{w}_k^\star$ is defined as [36]:

$$\mathrm{supp}(\boldsymbol{w}_k^\star)\triangleq\{j:[\boldsymbol{w}_k^\star]_j\neq 0\}, \tag{2}$$

where $[\boldsymbol{w}_k^\star]_j$ is the j-th entry of $\boldsymbol{w}_k^\star$. By definition (2), jointly sparse structure means that:

$$\mathrm{supp}(\boldsymbol{w}_1^\star)=\cdots=\mathrm{supp}(\boldsymbol{w}_k^\star)=\cdots\mathrm{supp}(\boldsymbol{w}_N^\star). \tag{3}$$

Since it is not trivial to solve problems with jointly sparse structure directly, several mixed-norms have been introduced in the literature to alleviate this problem. They include the mixed $\ell_{2,1}$-norm [37], $\ell_{\infty,1}$-norm [38] and their reweighted versions [39]. We collect $\boldsymbol{w}_\ell^\star$ over the entire network into an $L\times N$ matrix, and we replace the k-th column $\boldsymbol{w}_k^\star$ by the optimization

variable $\boldsymbol{w}_k$. This leads us to consider:

$$\boldsymbol{W}_k^{\rm glob} \triangleq \big[\boldsymbol{w}_1^\star \quad \cdots \quad \boldsymbol{w}_{k-1}^\star \quad \boldsymbol{w}_k \quad \boldsymbol{w}_{k+1}^\star \quad \cdots \quad \boldsymbol{w}_N^\star\big]. \tag{4}$$

Since matrices defined in (4) differ from node to node, we use the subscript k in notation $\boldsymbol{W}_k^{\rm glob}$ to distinguish them. Evaluating the mixed $\ell_{p,1}$-norm of matrix $\boldsymbol{W}_k^{\rm glob}$ with $p = 2$ or ∞, so as to promote the jointly sparse structure, results in the following two steps:

Step 1: Evaluate the ℓ_p-norm of each row of $\boldsymbol{W}_k^{\rm glob}$, and stack the results into an intermediate vector of dimension $L \times 1$;

Step 2: Evaluate the ℓ_1-norm of the obtained intermediate vector to promote sparsity.

Though $\ell_{2,1}$-norm can be more efficient in some cases [40], we shall consider the $\ell_{\infty,1}$-norm and its reweighted form to promote the joint-sparsity. It is shown in [41] that the $\ell_{\infty,1}$ relaxation is exact in the case of normalized nonrepeating data. [38] introduced a recursive adaptive group lasso algorithm for real-time penalized least squares prediction by using the $\ell_{1,\infty}$ regularization (The anthors defined $\ell_{\infty,1}$-norm as $\ell_{1,\infty}$-norm in [38] for a vector with group sparsity structure). Each update minimizes a convex but nondifferentiable function optimization problem. The authors developed an online homotopy method to reduce the computational complexity. [41] proposed a collaborative convex framework for factoring a data matrix into a nonnegative matrices product. The authors used $\ell_{\infty,1}$ regularization to select the dictionary from the data and shown that this leads to an exact convex relaxation of ℓ_0 regularization in the case of distinct noise-free data. [42] proposed a class of group sparse RLS algorithms by using different penalty term to promote the group sparsity, where the $\ell_{\infty,1}$-norm is used as one of the penalty term and the subgradient of $\ell_{\infty,1}$-norm is evaluated. Both $\ell_{\infty,1}$-norm and its reweighted form are element-wise separable, which facilitates the derivation of the proximal operator in the current work. As we focus on distributed processing in the current work, only local information exchange is authorized. Thus, we restrict $\boldsymbol{W}_k^{\rm glob}$ to the local quantity:

$$\boldsymbol{W}_k \triangleq \big[\boldsymbol{w}_k,\ \boldsymbol{w}_\ell^\star \text{ with } \ell \in \mathcal{N}_k^-\big] \in \mathbb{R}^{L\times|\mathcal{N}_k|}. \tag{5}$$

We consider that the columns of $\boldsymbol{W}_k$ are sorted in increasing order according to the values of k and ℓ. In the sequel, we shall show how to evaluate the (reweighted) $\ell_{\infty,1}$-norm of $\boldsymbol{W}_k$.

III. Proximal Multitask Diffusion LMS

Before proceeding, we rewrite $\boldsymbol{W}_k$ in another way in order to facilitate the presentation:

$$\boldsymbol{W}_k = \big[\bar{\boldsymbol{w}}_{k,1}^\top \quad \cdots \quad \bar{\boldsymbol{w}}_{k,m}^\top \quad \cdots \quad \bar{\boldsymbol{w}}_{k,L}^\top\big]^\top, \tag{6}$$

where $\bar{\boldsymbol{w}}_{k,m}$ is the m-th row of matrix $\boldsymbol{W}_k$ and defined by

$$\bar{\boldsymbol{w}}_{k,m} \triangleq \big[[\boldsymbol{w}_k]_m,\ [\boldsymbol{w}_\ell^\star]_m \text{ with } \ell \in \mathcal{N}_k^-\big] \in \mathbb{R}^{1\times|\mathcal{N}_k|} \tag{7}$$

with $[\boldsymbol{w}_k]_m$ being the m-th entry of $\boldsymbol{w}_k$. Definition (7) means that $\bar{\boldsymbol{w}}_{k,m}$ is a *row* vector. Note that (5) and (6) are actually the same matrix, but represented in different ways. Also, $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty$ is defined as

$$\|\bar{\boldsymbol{w}}_{k,m}\|_\infty \triangleq \max\{|[\boldsymbol{w}_k]_m|, |[\boldsymbol{w}_\ell^\star]_m| \text{ with } \ell \in \mathcal{N}_k^-\} \tag{8}$$

which will be used in $g_1(\boldsymbol{w}_k)$ and $g_2(\boldsymbol{w}_k)$ of (11) and (12), respectively.

To determine the unknown vectors $\boldsymbol{w}_k^\star$ with jointly sparse structure, we consider the regularized cost at node k:

$$J_k(\boldsymbol{w}_k) = J_k'(\boldsymbol{w}_k) + \lambda_k g_i(\boldsymbol{w}_k), \tag{9}$$

where $J_k'(\boldsymbol{w}_k)$ is the mean-square error (MSE) defined as:

$$J_k'(\boldsymbol{w}_k) \triangleq \frac{1}{2}\,\mathbb{E}\left\{|d_{k,n} - \boldsymbol{x}_{k,n}^\top \boldsymbol{w}_k|^2\right\}. \tag{10}$$

The nonnegative parameter λ_k is used to control the regularization strength, $g_1(\boldsymbol{w}_k)$ and $g_2(\boldsymbol{w}_k)$ evaluate the $\ell_{\infty,1}$-norm and reweighted $\ell_{\infty,1}$-norm of $\boldsymbol{W}_k$, respectively, with:

$$g_1(\boldsymbol{w}_k) \triangleq \sum_{m=1}^{L} \|\bar{\boldsymbol{w}}_{k,m}\|_\infty, \tag{11}$$

$$g_2(\boldsymbol{w}_k) \triangleq \sum_{m=1}^{L} \log\left[1 + \frac{\|\bar{\boldsymbol{w}}_{k,m}\|_\infty}{\varepsilon}\right], \tag{12}$$

where $\varepsilon > 0$ is a parameter set by the user. Interpretations about the $\ell_{\infty,1}$-norm $g_1(\boldsymbol{w}_k)$ and reweighted $\ell_{\infty,1}$-norm $g_2(\boldsymbol{w}_k)$ are provided in Appendix A. At each node k, we then consider the convex optimization problem [43]:

$$\boldsymbol{w}_k^\dagger = \arg\min_{\boldsymbol{w}_k} J_k(\boldsymbol{w}_k). \tag{13}$$

Within the context of online learning, such optimization problem is usually solved via subgradient-based methods. In this paper, we propose to devise a proximal algorithm since these algorithms are usually more stable and have a better convergence rate than subgradient iterations [30], [32], [33]. Proximal gradient iteration consists of [31]:

$$\boldsymbol{w}_{k,n+1} = \mathrm{prox}_{\mu_k\lambda_k,g_i(\boldsymbol{w}_k)}\Big(\boldsymbol{w}_{k,n} - \mu_k\nabla J'_k(\boldsymbol{w}_{k,n})\Big), \tag{14}$$

where μ_k is a positive small step-size, and:

$$\mathrm{prox}_{\lambda,g_i(\boldsymbol{w}_k)}(\boldsymbol{v}) \triangleq \arg\min_{\boldsymbol{w}_k}\Big(g_i(\boldsymbol{w}_k) + \frac{1}{2\lambda}\|\boldsymbol{w}_k - \boldsymbol{v}\|_2^2\Big). \tag{15}$$

is the proximal operator. By introducing the intermediate quantity $\boldsymbol{\psi}_{k,n+1}$, we further decompose (14) into two steps:

$$\boldsymbol{\psi}_{k,n+1} = \boldsymbol{w}_{k,n} - \mu_k\nabla J'_k(\boldsymbol{w}_{k,n}), \tag{16}$$

$$\boldsymbol{w}_{k,n+1} = \mathrm{prox}_{\mu_k\lambda_k,g_i(\boldsymbol{w}_k)}\big(\boldsymbol{\psi}_{k,n+1}\big). \tag{17}$$

Equation (16) is the local update step, and (17) is the proximal step. Calculating the gradient of $J'_k(\boldsymbol{w}_k)$ at $\boldsymbol{w}_{k,n}$ and approximating the unknown statistical quantities with instantaneous quantities, (16) becomes:

$$\boldsymbol{\psi}_{k,n+1} = \boldsymbol{w}_{k,n} + \mu_k\boldsymbol{x}_{k,n}\big(d_{k,n} - \boldsymbol{x}_{k,n}^\top\boldsymbol{w}_{k,n}\big). \tag{18}$$

Combining (17) and (18) yields the proximal multitask diffusion LMS algorithm for networks with jointly sparse structure reported in **Algorithm** 1. This algorithm differs from the standard diffusion procedure devised in [8], [9] in the sense that it does not combine, in an explicit manner, the intermediate estimates $\boldsymbol{\psi}_{\ell,n+1}$ in the neighborhood of each node k. In our algorithm, the information on the support of the local estimates is shared by neighboring nodes via $g_i(\boldsymbol{w}_k)$.

Algorithm 1 Proximal multitask diffusion LMS

Initialize $\boldsymbol{w}_{k,0}$ for all $k = 1, 2, \cdots, N$, and repeat:

$$\begin{cases} \boldsymbol{\psi}_{k,n+1} = \boldsymbol{w}_{k,n} + \mu_k\boldsymbol{x}_{k,n}\big(d_{k,n} - \boldsymbol{x}_{k,n}^\top\boldsymbol{w}_{k,n}\big) \\ \boldsymbol{w}_{k,n+1} = \mathrm{prox}_{\mu_k\lambda_k,g_i(\boldsymbol{w}_k)}\big(\boldsymbol{\psi}_{k,n+1}\big) \end{cases} \tag{19}$$

IV. Proximal Operators Evaluation

To apply the algorithm, we need to derive closed-form expressions for the proximal operators. Equation (17) becomes:

$$\begin{aligned} \boldsymbol{w}_{k,n+1} &= \mathrm{prox}_{\mu_k\lambda_k,g_i(\boldsymbol{w}_k)}(\boldsymbol{\psi}_{k,n+1}) \\ &= \underset{\boldsymbol{w}_k}{\mathrm{argmin}}\,\big(g_i(\boldsymbol{w}_k) + \frac{1}{2\mu_k\lambda_k}\|\boldsymbol{w}_k - \boldsymbol{\psi}_{k,n+1}\|_2^2\big). \end{aligned} \tag{20}$$

As $g_i(\boldsymbol{w}_k)$ is separable over its all entries, the proximal operator can be evaluated in an element-wise manner as [31]:

$$\begin{aligned} &\big[\mathrm{prox}_{\mu_k\lambda_k,g_i(\boldsymbol{w}_k)}(\boldsymbol{\psi}_{k,n+1})\big]_m \\ &\qquad = \mathrm{prox}_{\mu_k\lambda_k,g_{i,m}([\boldsymbol{w}_k]_m)}([\boldsymbol{\psi}_{k,n+1}]_m) \end{aligned} \tag{21}$$

with:

$$g_{1,m}([\boldsymbol{w}_k]_m) \triangleq \|\bar{\boldsymbol{w}}_{k,m}\|_\infty \tag{22}$$

$$g_{2,m}([\boldsymbol{w}_k]_m) \triangleq \log\bigg[1 + \frac{\|\bar{\boldsymbol{w}}_{k,m}\|_\infty}{\varepsilon}\bigg], \tag{23}$$

where $\bar{\boldsymbol{w}}_{k,m}$ is the m-th row of matrix $\boldsymbol{W}_k$ in (5).

A. Approximate proximal operator of $\ell_{\infty,1}$-norm

Substituting $g_{1,m}$ into (21), we obtain:

$$\begin{aligned} [\boldsymbol{w}_{k,n+1}]_m = \underset{[\boldsymbol{w}_k]_m}{\mathrm{argmin}}\Big(&\max\{|[\boldsymbol{w}_k]_m|, |[\boldsymbol{w}_\ell^\star]_m| \text{ with } \ell \in \mathcal{N}_k^-\} \\ &+ \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2\Big). \end{aligned} \tag{24}$$

For the sake of simpler notations, we shall denote $[\boldsymbol{w}_{k,n+1}]_m$ by $\hat{w}$ as long as there is no ambiguity. We shall also denote the maximal value of $|[\boldsymbol{w}_\ell^\star]_m|$ for $\ell \in \mathcal{N}_k^-$ as $[\boldsymbol{w}_k^o]_m$. According to the relation between $|[\boldsymbol{w}_k]_m|$ and $[\boldsymbol{w}_k^o]_m$, we further split the problem into the following two cases:

- Case 1: $|[\boldsymbol{w}_k]_m| < [\boldsymbol{w}_k^o]_m$. In this case, (24) becomes:

$$\hat{w} = \underset{\substack{[\boldsymbol{w}_k]_m \\ |[\boldsymbol{w}_k]_m|<[\boldsymbol{w}_k^o]_m}}{\operatorname{argmin}} \; [\boldsymbol{w}_k^o]_m + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2. \tag{25}$$

Since the first term on the right-hand side (RHS) of (25) does not depend on $[\boldsymbol{w}_k]_m$, we conclude that:

$$\hat{w} = \begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m, & \text{if } |[\boldsymbol{\psi}_{k,n+1}]_m| < [\boldsymbol{w}_k^o]_m \\ [\boldsymbol{w}_k^o]_m, & \text{if } [\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{w}_k^o]_m \\ -[\boldsymbol{w}_k^o]_m, & \text{if } [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m. \end{cases} \tag{26}$$

- Case 2: $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m$. Equation (24) becomes:

$$\hat{w} = \underset{\substack{[\boldsymbol{w}_k]_m \\ |[\boldsymbol{w}_k]_m|\geq[\boldsymbol{w}_k^o]_m}}{\operatorname{argmin}} \; |[\boldsymbol{w}_k]_m| + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2. \tag{27}$$

We shall first discard the constraint $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m$ for reasons of simplicity, and denote by $\hat{w}^o$ the solution of the unconstrained problem. As the cost function in (27) is convex on $\mathbb{R}$, this constraint will be taken into account in the course of the calculation. Consider first:

$$\begin{aligned} \hat{w}^o &= \underset{[\boldsymbol{w}_k]_m}{\operatorname{argmin}} \, |[\boldsymbol{w}_k]_m| + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2 \\ &= \operatorname{prox}_{\mu_k\lambda_k, s([\boldsymbol{w}_k]_m)}\big([\boldsymbol{\psi}_{k,n+1}]_m\big) \end{aligned} \tag{28}$$

where function $s([\boldsymbol{w}_k]_m)$ is defined as:

$$s([\boldsymbol{w}_k]_m) \triangleq |[\boldsymbol{w}_k]_m|. \tag{29}$$

The optimality condition of (28) says that zero belongs to the subgradient set at the minimizer $\hat{w}^o$, that is, [31]

$$0 \in \partial|\hat{w}^o| + \frac{1}{\mu_k\lambda_k}\big(\hat{w}^o - [\boldsymbol{\psi}_{k,n+1}]_m\big) \tag{30}$$

This means that:

$$\big([\boldsymbol{\psi}_{k,n+1}]_m - \hat{w}^o\big) \in \mu_k\lambda_k \, \partial|\hat{w}^o| \tag{31}$$

with

$$\partial|\hat{w}^o| = \begin{cases} -1, & \text{if } \hat{w}^o < 0 \\ +1, & \text{if } \hat{w}^o > 0 \\ [-1,\, 1], & \text{if } \hat{w}^o = 0 \end{cases} \tag{32}$$

the subdifferential of the non-differentiable function $|\hat{w}^o|$ [31]. The third case in (32) means that, at $\hat{w}^o = 0$, $\partial|\hat{w}^o|$ can be any value within $[-1,\, 1]$.
Condition (31) leads to:

$$\hat{w}^o = \begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m + \mu_k\lambda_k, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m < -\mu_k\lambda_k \\ [\boldsymbol{\psi}_{k,n+1}]_m - \mu_k\lambda_k, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m > \mu_k\lambda_k \\ 0, \text{ otherwise.} \end{cases} \tag{33}$$

If $[\boldsymbol{w}_k^o]_m = 0$, problem (27) becomes unconstrained and we have:

$$\hat{w} = \hat{w}^o \tag{34}$$

Otherwise, since problem (27) is convex on $\mathbb{R}$, considering constraint $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m > 0$ with (33) yields:

$$\hat{w} = \tag{35}$$
$$\begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m + \mu_k\lambda_k, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m - \mu_k\lambda_k \\ -[\boldsymbol{w}_k^o]_m, \text{ if } -[\boldsymbol{w}_k^o]_m - \mu_k\lambda_k < [\boldsymbol{\psi}_{k,n+1}]_m < 0 \\ -[\boldsymbol{w}_k^o]_m \text{ or } [\boldsymbol{w}_k^o]_m, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m = 0 \\ [\boldsymbol{w}_k^o]_m, \text{ if } 0 < [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k \\ [\boldsymbol{\psi}_{k,n+1}]_m - \mu_k\lambda_k, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k \end{cases}$$

To evaluate the proximal operator (24), several issues have to be addressed.

1. One of the main issues is that we first need to know which of (26), (34) or (35) has to be applied as the proximal operator of (24). We shall now consider the following two cases: $[\boldsymbol{w}_k^o]_m = 0$ and $[\boldsymbol{w}_k^o]_m > 0$.

- Case A: $[\boldsymbol{w}_k^o]_m = 0$. Since condition $|[\boldsymbol{w}_k]_m| < [\boldsymbol{w}_k^o]_m$ of Case 1 cannot hold, we only consider Case 2. The proximal operator is given by $\hat{w}^o$ in (33). Interestingly, note that (33) is the proximal operator of the ℓ_1-norm regularizer for a scalar.
- Case B: $[\boldsymbol{w}_k^o]_m > 0$. Proximal operators (26) and (35) hold simultaneously. We shall choose the solution that minimizes the cost (24). As shown in Appendix B, we finally arrive at the following expression:

$$\hat{w} = \begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m + \mu_k\lambda_k, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m - \mu_k\lambda_k \\ -[\boldsymbol{w}_k^o]_m, \text{if } -[\boldsymbol{w}_k^o]_m - \mu_k\lambda_k < [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m \\ [\boldsymbol{\psi}_{k,n+1}]_m, \text{if } \big|[\boldsymbol{\psi}_{k,n+1}]_m\big| < [\boldsymbol{w}_k^o]_m \\ [\boldsymbol{w}_k^o]_m, \text{if } [\boldsymbol{w}_k^o]_m \leq [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k \\ [\boldsymbol{\psi}_{k,n+1}]_m - \mu_k\lambda_k, \text{ if } [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k \leq [\boldsymbol{\psi}_{k,n+1}]_m \end{cases} \tag{36}$$

2. Another issue is that $\hat{w}$ cannot be evaluated with (33) and (36) since $[\boldsymbol{w}_k^o]_m$ is unknown. To fix this problem, we follow a local one-step approximation strategy that has already proven its effectiveness in the literature [17], and which consists of using $\boldsymbol{\psi}_{\ell,n+1}$ as an approximation of $\boldsymbol{w}_\ell^\star$. One of the benefits of this approximation is that $\boldsymbol{\psi}_{\ell,n+1}$ can be transmitted by node ℓ to node k if the latter is in its neighborhood. An approximation of $[\boldsymbol{w}_k^o]_m$ is then given by $\max_{\ell\in\mathcal{N}_k^-}\{\big|[\boldsymbol{\psi}_{\ell,n+1}]_m\big|\}$, which allows node k to evaluate its proximal operator. This approximation will be taken into consideration in the theoretical analysis in Section V-A.

3. Before addressing the last issue, we need to point out the prominent role of Case A compared to Case B: unlike the proximal operator (36) in Case B, only the proximal operator (33) in Case A has the capability to drive $[\boldsymbol{w}_k]_m$ to zero and promote sparsity. Observe that condition $[\boldsymbol{w}_k^o]_m = 0$ has to be satisfied to trigger Case A, otherwise Case B is considered. Within the context of online learning with stochastic gradient descent algorithms, due to the existence of gradient noise, the estimates of $[\boldsymbol{w}_k^o]_m$ for zero-valued entries are actually nonzero-valued but vary around zero. Condition $[\boldsymbol{w}_k^o]_m = 0$ of Case A is thus seldom satisfied, and $[\boldsymbol{\psi}_{k,n+1}]_m$ is not driven to zero. To promote the sparsity of the estimates, since the true values of the non-zero entries are usually far away from zero, we introduce a small positive threshold value τ_1 instead of zero to make a distinguish between zero-valued and nonzero-valued entries. As a consequence, we arrive at conditions $[\boldsymbol{w}_k^o]_m \leq \tau_1$ to trigger Case A and $[\boldsymbol{w}_k^o]_m > \tau_1$ to select Case B.

We summarize our method in **Algorithm** 2.

Algorithm 2 Approximate proximal operator of $\ell_{\infty,1}$-norm

Initialization: Choose threshold value $\tau_1 > 0$.

Proximal operator: At each instant $n \geq 0$, for each node k, utilize $\boldsymbol{\psi}_{k,n+1}$ to evaluate $\boldsymbol{w}_{k,n+1}$ in an elementwise manner:

1) Calculate $[\boldsymbol{w}_k^o]_m$ as the maximal value of $\big|[\boldsymbol{\psi}_{\ell,n+1}]_m\big|$ for all $\ell \in \mathcal{N}_k^-$;
2) If $[\boldsymbol{w}_k^o]_m \leq \tau_1$, then calculate $[\boldsymbol{w}_{k,n+1}]_m$ as $\hat{w}^o$ via (33);
3) If $[\boldsymbol{w}_k^o]_m > \tau_1$, then calculate $[\boldsymbol{w}_{k,n+1}]_m$ as $\hat{w}$ via (36).

B. Approximate proximal operator of reweighted $\ell_{\infty,1}$-norm

Substituting the expression of $g_{2,m}$ into (21), we arrive at:

$$\begin{aligned}[\boldsymbol{w}_{k,n+1}]_m = \\ \underset{[\boldsymbol{w}_k]_m}{\operatorname{argmin}} \Bigg[& \log\left(1 + \frac{\max\big\{|[\boldsymbol{w}_k]_m|, |[\boldsymbol{w}_\ell^\star]_m| : \forall \ell \in \mathcal{N}_k^-\big\}}{\varepsilon}\right) \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2\Bigg]. \end{aligned} \tag{37}$$

Considering the same definition for $[\boldsymbol{w}_k^o]_m$ as before, and denoting $[\boldsymbol{w}_{k,n+1}]_m$ by $\hat{w}$ for the sake of conciseness, the problem can be split into two cases depending on the order relation between $[\boldsymbol{w}_k^o]_m$ and $|[\boldsymbol{w}_k]_m|$:

- Case 1: $|[\boldsymbol{w}_k]_m| < [\boldsymbol{w}_k^o]_m$. Equation (37) becomes:

$$\begin{aligned}\hat{w} = \underset{\substack{[\boldsymbol{w}_k]_m \\ |[\boldsymbol{w}_k]_m| < [\boldsymbol{w}_k^o]_m}}{\operatorname{argmin}} & \log\left(1 + \frac{[\boldsymbol{w}_k^o]_m}{\varepsilon}\right) \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2. \end{aligned} \tag{38}$$

This problem has a closed-form solution given by:

$$\hat{w} = \begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m, & \text{if } |[\boldsymbol{\psi}_{k,n+1}]_m| < [\boldsymbol{w}_k^o]_m \\ [\boldsymbol{w}_k^o]_m, & \text{if } [\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{w}_k^o]_m \\ -[\boldsymbol{w}_k^o]_m, & \text{if } [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m. \end{cases} \tag{39}$$

- Case 2: $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m$. In this case, (37) becomes:

$$\begin{aligned} \hat{w} = \underset{\substack{[\boldsymbol{w}_k]_m \\ |[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m}}{\operatorname{argmin}} & \log\Big(1 + \frac{|[\boldsymbol{w}_k]_m|}{\varepsilon}\Big) \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2. \end{aligned} \tag{40}$$

By first discarding the constraint $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m$ as we did it before, we obtain the following unconstrained optimization problem:

$$\begin{aligned} \hat{w}^o = \underset{[\boldsymbol{w}_k]_m}{\operatorname{argmin}} & \log\Big(1 + \frac{|[\boldsymbol{w}_k]_m|}{\varepsilon}\Big) \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2. \end{aligned} \tag{41}$$

Deriving the solution $\hat{w}^o$ directly from the optimality condition is a tough problem. We shall now alleviate this issue with a Majorization-Minimization (MM) approach [44], [45]. Conceptually, MM algorithms work by iteratively minimizing a simple surrogate function majorizing a given objective function. By introducing the auxiliary variable $[\boldsymbol{u}_k]_m$, problem (41) can be written as:

$$\begin{aligned} \hat{w}^o = \underset{\substack{[\boldsymbol{w}_k]_m, [\boldsymbol{u}_k]_m \\ |[\boldsymbol{w}_k]_m| \leq [\boldsymbol{u}_k]_m}}{\operatorname{argmin}} & \log\Big(1 + \frac{[\boldsymbol{u}_k]_m}{\varepsilon}\Big) \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2. \end{aligned} \tag{42}$$

Define:

$$f([\boldsymbol{u}_k]_m) \triangleq \log\Big(1 + \frac{[\boldsymbol{u}_k]_m}{\varepsilon}\Big). \tag{43}$$

It can be observed that function $f([\boldsymbol{u}_k]_m)$ is concave and below its tangent. So we have:

$$\begin{aligned} f([\boldsymbol{u}_k]_m) \leq & f([\boldsymbol{u}_k^{(t)}]_m) \\ & + f'([\boldsymbol{u}_k^{(t)}]_m) \cdot \big([\boldsymbol{u}_k]_m - [\boldsymbol{u}_k^{(t)}]_m\big) \end{aligned} \tag{44}$$

with $f'(\cdot)$ the first-order derivative of $f(\cdot)$, and where the superscript (t) denotes the iteration index. Using (44), we can construct a surrogate function for (42) as follows:

$$\begin{aligned} h([\boldsymbol{u}_k]_m, [\boldsymbol{w}_k]_m) \triangleq & \log\Big(1 + \frac{[\boldsymbol{u}_k^{(t)}]_m}{\varepsilon}\Big) + \frac{[\boldsymbol{u}_k]_m - [\boldsymbol{u}_k^{(t)}]_m}{\varepsilon + [\boldsymbol{u}_k^{(t)}]_m} \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2 \end{aligned} \tag{45}$$

Using the driving principle of MM algorithms, this yields the following optimization problem:

$$\begin{aligned} \big([\boldsymbol{w}_k^{(t+1)}]_m, [\boldsymbol{u}_k^{(t+1)}]_m\big) = & \underset{\substack{[\boldsymbol{w}_k]_m, [\boldsymbol{u}_k]_m \\ |[\boldsymbol{w}_k]_m| \leq [\boldsymbol{u}_k]_m}}{\operatorname{argmin}} \frac{[\boldsymbol{u}_k]_m}{\varepsilon + [\boldsymbol{u}_k^{(t)}]_m} \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2 \end{aligned} \tag{46}$$

which is equivalent to:

$$\begin{aligned} [\boldsymbol{w}_k^{(t+1)}]_m = \underset{[\boldsymbol{w}_k]_m}{\operatorname{argmin}} & \frac{|[\boldsymbol{w}_k]_m|}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|} \\ & + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2. \end{aligned} \tag{47}$$

Note that, in the construction of (46), we have dropped several constant terms unrelated to $[\boldsymbol{u}_k]_m$ and $[\boldsymbol{w}_k]_m$. Using the optimality condition of (47) at the minimizer $[\boldsymbol{w}_k^{(t+1)}]_m$, we have:

$$\big([\boldsymbol{\psi}_{k,n+1}]_m - [\boldsymbol{w}_k^{(t+1)}]_m\big) \in \frac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|} \cdot \partial \big|[\boldsymbol{w}_k^{(t+1)}]_m\big|. \tag{48}$$

Considering (32) with (48) yields:

$$[\boldsymbol{w}_k^{(t+1)}]_m = \tag{49}$$
$$\begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m + \dfrac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{if } [\boldsymbol{\psi}_{k,n+1}]_m < \dfrac{-\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|} \\ [\boldsymbol{\psi}_{k,n+1}]_m - \dfrac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{if } [\boldsymbol{\psi}_{k,n+1}]_m > \dfrac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|} \\ 0 \qquad \text{otherwise.} \end{cases}$$

Performing one iteration of (49) with $[\boldsymbol{w}_k^{(t)}]_m = [\boldsymbol{w}_{k,n}]_m$ at each instant n can be sufficient for approximating $\hat{w}^o$ with enough precision, as illustrated in the sequel. This value is then used to obtain an approximate solution $\hat{w}$ of problem (40) by taking the constraint $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m$ into account.
If $[\boldsymbol{w}_k^o]_m = 0$, problem (40) becomes unconstrained and we have:

$$\hat{w} = \hat{w}^o \tag{50}$$

where $\hat{w}^o$ is obtained by (49).
Otherwise, since problem (40) is convex on $\mathbb{R}$, considering constraint $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m > 0$ with (49) yields:

$$\hat{w} = \tag{51}$$
$$\begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m + \dfrac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{c}_k]_m \\ -[\boldsymbol{w}_k^o]_m, \text{ if } -[\boldsymbol{c}_k]_m < [\boldsymbol{\psi}_{k,n+1}]_m < 0 \\ -[\boldsymbol{w}_k^o]_m \text{ or } [\boldsymbol{w}_k^o]_m, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m = 0 \\ [\boldsymbol{w}_k^o]_m, \text{ if } 0 < [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{c}_k]_m \\ [\boldsymbol{\psi}_{k,n+1}]_m - \dfrac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{c}_k]_m \end{cases}$$

with $[\boldsymbol{c}_k]_m$ a constant defined as:

$$[\boldsymbol{c}_k]_m \triangleq [\boldsymbol{w}_k^o]_m + \frac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}. \tag{52}$$

As with the $\ell_{\infty,1}$-norm, we have to fix several issues.
1. We first need to check which of (39), (49) and (51) has to be applied to calculate the proximal operator. We consider the following two cases: $[\boldsymbol{w}_k^o]_m = 0$ and $[\boldsymbol{w}_k^o]_m > 0$.

- Case A: $[\boldsymbol{w}_k^o]_m = 0$. We focus on Case 2 since condition for Case 1 cannot hold. In Case 2, observe that any $[\boldsymbol{w}_k]_m$ calculated as $[\boldsymbol{w}_k^{(t+1)}]_m$ in (49) satisfies $|[\boldsymbol{w}_k]_m| \geq [\boldsymbol{w}_k^o]_m = 0$. Thus the proximal operator is given by (49).
- Case B: $[\boldsymbol{w}_k^o]_m > 0$. Case 1 and Case 2 can hold simultaneously. We must choose, among the resulting proximal operators, the one that minimizes cost (37). The derivation is provided in Appendix C. We obtain the following solution:

$$\hat{w} = \tag{53}$$
$$\begin{cases} [\boldsymbol{\psi}_{k,n+1}]_m + \dfrac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{c}_k]_m \\ -[\boldsymbol{w}_k^o]_m, \text{ if } -[\boldsymbol{c}_k]_m < [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m \\ [\boldsymbol{\psi}_{k,n+1}]_m, \text{ if } \big|[\boldsymbol{\psi}_{k,n+1}]_m\big| < [\boldsymbol{w}_k^o]_m \\ [\boldsymbol{w}_k^o]_m, \text{ if } [\boldsymbol{w}_k^o]_m \leq [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{c}_k]_m \\ [\boldsymbol{\psi}_{k,n+1}]_m - \dfrac{\mu_k \lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{c}_k]_m \end{cases}$$

2. Unlike the proximal operator (53) in Case B, only the proximal operator (49) in Case A has the capability to drive $[\boldsymbol{w}_k]_m$ to zero and promote sparsity. Thus, as in Section IV-A, we relax the condition $[\boldsymbol{w}_k^o]_m = 0$ by introducing a small

positive tolerance τ_2 to distinguish between zero and nonzero entries. This leads to the condition $[\boldsymbol{w}_k^o]_m \leq \tau_2$ for Case A, and $[\boldsymbol{w}_k^o]_m > \tau_2$ for Case B.

We summarize our method in **Algorithm** 3.

Algorithm 3 Approximate proximal operator of reweighted $\ell_{\infty,1}$-norm

Initialization: Choose threshold value $\tau_2 > 0$ and $\varepsilon > 0$.

Proximal operator: At each instant $n \geq 0$, for each node k, utilize $\boldsymbol{\psi}_{k,n+1}$ to evaluate $\boldsymbol{w}_{k,n+1}$ in an element-wise manner:

1) Calculate $[\boldsymbol{w}_k^o]_m$ as the maximal value of $\big|[\boldsymbol{\psi}_{\ell,n+1}]_m\big|$ for all $\ell \in \mathcal{N}_k^-$;
2) If $[\boldsymbol{w}_k^o]_m \leq \tau_2$, then calculate $[\boldsymbol{w}_{k,n+1}]_m$ as $[\boldsymbol{w}_k^{(t+1)}]_m$ via (49) with $[\boldsymbol{w}_k^{(t)}]_m = [\boldsymbol{w}_{k,n}]_m$;
3) If $[\boldsymbol{w}_k^o]_m > \tau_2$, then calculate $[\boldsymbol{w}_{k,n+1}]_m$ as $\hat{w}$ via (53).

V. Performance and Convergence Analyses

In this section, we analyse the performance and convergence property of **Algorithm** 2 and **Algorithm** 3 in an unified framework. Quantities specifically related to $\ell_{\infty,1}$-norm or reweighted $\ell_{\infty,1}$-norm are distinguished by the superscripts $^{(1)}$ and $^{(2)}$, respectively. Observe that the closed-forms expressions of both approximate proximal operators can be written compactly as:

$$\operatorname{prox}_{\mu_k\lambda_k, g_i(\boldsymbol{w}_k)}\big(\boldsymbol{\psi}_{k,n+1}\big) = \boldsymbol{\psi}_{k,n+1} - \boldsymbol{\gamma}_{k,n+1}^{(i)}, \tag{54}$$

with $\boldsymbol{\gamma}_{k,n+1}^{(i)}$ a $(L \times 1)$-dimensional vector.

For the $\ell_{\infty,1}$-norm, the m-th entry of $\boldsymbol{\gamma}_{k,n+1}^{(1)}$ is given by:

$$\big[\boldsymbol{\gamma}_{k,n+1}^{(1)}\big]_m = \begin{cases} -\mu_k\lambda_k, & \text{if } \big[\boldsymbol{\psi}_{k,n+1}\big]_m < -\mu_k\lambda_k \\ \big[\boldsymbol{\psi}_{k,n+1}\big]_m, & \text{if } \big|\big[\boldsymbol{\psi}_{k,n+1}\big]_m\big| \leq \mu_k\lambda_k \\ \mu_k\lambda_k, & \text{if } \big[\boldsymbol{\psi}_{k,n+1}\big]_m > \mu_k\lambda_k \end{cases} \tag{55}$$

if $[\boldsymbol{w}_k^o]_m \leq \tau_1$, and:

$$\big[\boldsymbol{\gamma}_{k,n+1}^{(1)}\big]_m = \tag{56}$$
$$\begin{cases} -\mu_k\lambda_k, \text{ if } \big[\boldsymbol{\psi}_{k,n+1}\big]_m \leq -[\boldsymbol{w}_k^o]_m - \mu_k\lambda_k \\ \big[\boldsymbol{\psi}_{k,n+1}\big]_m + [\boldsymbol{w}_k^o]_m, \text{ if } -[\boldsymbol{w}_k^o]_m - \mu_k\lambda_k < [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m \\ 0, \text{ if } \big|[\boldsymbol{\psi}_{k,n+1}]_m\big| < [\boldsymbol{w}_k^o]_m \\ \big[\boldsymbol{\psi}_{k,n+1}\big]_m - [\boldsymbol{w}_k^o]_m, \text{ if } [\boldsymbol{w}_k^o]_m \leq [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k \\ \mu_k\lambda_k, \text{ if } \big[\boldsymbol{\psi}_{k,n+1}\big]_m \geq [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k \end{cases}$$

if $[\boldsymbol{w}_k^o]_m > \tau_1$. For the reweighted $\ell_{\infty,1}$-norm, the m-th entry of $\boldsymbol{\gamma}_{k,n+1}^{(2)}$ is given by:

$$\big[\boldsymbol{\gamma}_{k,n+1}^{(2)}\big]_m = \begin{cases} \dfrac{-\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{if } \big[\boldsymbol{\psi}_{k,n+1}\big]_m < \dfrac{-\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|} \\ \big[\boldsymbol{\psi}_{k,n+1}\big]_m, \text{if } \big|\big[\boldsymbol{\psi}_{k,n+1}\big]_m\big| \leq \dfrac{\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|} \\ \dfrac{\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{if } \big[\boldsymbol{\psi}_{k,n+1}\big]_m > \dfrac{\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|} \end{cases} \tag{57}$$

if $[\boldsymbol{w}_k^o]_m \leq \tau_2$, and:

$$\big[\boldsymbol{\gamma}_{k,n+1}^{(2)}\big]_m = \tag{58}$$
$$\begin{cases} -\dfrac{\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{c}_k]_m \\ [\boldsymbol{\psi}_{k,n+1}]_m + [\boldsymbol{w}_k^o]_m, \text{ if } -[\boldsymbol{c}_k]_m < [\boldsymbol{\psi}_{k,n+1}]_m \leq -[\boldsymbol{w}_k^o]_m \\ 0, \text{ if } \big|[\boldsymbol{\psi}_{k,n+1}]_m\big| < [\boldsymbol{w}_k^o]_m \\ [\boldsymbol{\psi}_{k,n+1}]_m - [\boldsymbol{w}_k^o]_m, \text{ if } [\boldsymbol{w}_k^o]_m \leq [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{c}_k]_m \\ \dfrac{\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}, \text{ if } [\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{c}_k]_m \end{cases}$$

if $[\boldsymbol{w}_k^o]_m > \tau_2$. We observe in (55)–(56) that:

$$\big|\big[\boldsymbol{\gamma}_{k,n+1}^{(1)}\big]_m\big| \leq \mu_k\lambda_k \tag{59}$$

Thus, $\boldsymbol{\gamma}_{k,n+1}^{(1)}$ is absolutely bounded: $|\boldsymbol{\gamma}_{k,n+1}^{(1)}| \preceq \mu_k\lambda_k\mathbb{1}_L$.

From (57)–(58), we conclude that:

$$|\boldsymbol{\gamma}_{k,n+1}^{(2)}| \preceq b_k\mathbb{1}_L \tag{60}$$

where b_k is a constant defined as:

$$b_k \triangleq \frac{\mu_k\lambda_k}{\varepsilon}. \tag{61}$$

Substituting (54) into (19), we obtain:

$$\begin{cases} \boldsymbol{\psi}_{k,n+1} = \boldsymbol{w}_{k,n} + \mu_k\boldsymbol{x}_{k,n}\big(d_{k,n} - \boldsymbol{x}_{k,n}^{\top}\boldsymbol{w}_{k,n}\big) \\ \boldsymbol{w}_{k,n+1} = \boldsymbol{\psi}_{k,n+1} - \boldsymbol{\gamma}_{k,n+1}^{(i)}. \end{cases} \tag{62}$$

We shall now analyse the performance of proximal multitask diffusion LMS algorithm based on expression (62).

Define

$$\tilde{\boldsymbol{w}}_{k,n+1} \triangleq \boldsymbol{w}_{k,n+1} - \boldsymbol{w}_k^{\star}, \tag{63}$$

$$\tilde{\boldsymbol{\psi}}_{k,n+1} \triangleq \boldsymbol{\psi}_{k,n+1} - \boldsymbol{w}_k^{\star}. \tag{64}$$

By collecting $\boldsymbol{w}_k^{\star}$, $\boldsymbol{w}_{k,n+1}$, $\boldsymbol{\psi}_{k,n+1}$, $\tilde{\boldsymbol{w}}_{k,n+1}$, $\tilde{\boldsymbol{\psi}}_{k,n+1}$, $\boldsymbol{\gamma}_{k,n+1}^{(i)}$ over the entire network into block column vectors, we obtain quantities $\boldsymbol{w}^{\star}$, $\boldsymbol{w}_{n+1}$, $\boldsymbol{\psi}_{n+1}$, $\tilde{\boldsymbol{w}}_{n+1}$, $\tilde{\boldsymbol{\psi}}_{n+1}$, $\boldsymbol{\gamma}_{n+1}^{(i)}$, respectively. To facilitate the theoretical analysis, we introduce the following assumptions on the regression data and step-size. These assumptions are widely used in the analysis of adaptive filters [46]–[48] and diffusion networks [7], [9], [14].

A1 (*Independent Regressors*): The regression vector $\boldsymbol{x}_{k,n}$, generated from a zero-mean random process, is temporally stationary, white (over n) and spatially independent (over k) with covariance matrix $\boldsymbol{R}_{x,k} = \mathbb{E}\{\boldsymbol{x}_{k,n}\boldsymbol{x}_{k,n}^{\top}\} > 0$.

A2 (*Small step-sizes*): The step-sizes μ_k of the network are small enough, so that terms on the higher-order powers of the step-sizes can be ignored.

A. Mean behavior analysis

Subtracting $\boldsymbol{w}_k^{\star}$ from both sides of the first equation of (62), and using signal model (1) and block notations, we obtain:

$$\tilde{\boldsymbol{\psi}}_{n+1} = \boldsymbol{B}_n\tilde{\boldsymbol{w}}_n + \boldsymbol{U}\boldsymbol{h}_n, \tag{65}$$

where

$$\boldsymbol{B}_n \triangleq \boldsymbol{I} - \boldsymbol{U}\boldsymbol{M}_n \tag{66}$$

with

$$\boldsymbol{U} \triangleq \mathrm{diag}\{\mu_1, \mu_2, \cdots, \mu_N\} \otimes \boldsymbol{I}_L, \tag{67}$$

$$\boldsymbol{M}_n \triangleq \mathrm{diag}\big\{\boldsymbol{x}_{1,n}\boldsymbol{x}_{1,n}^{\top}, \boldsymbol{x}_{2,n}\boldsymbol{x}_{2,n}^{\top}, \cdots, \boldsymbol{x}_{N,n}\boldsymbol{x}_{N,n}^{\top}\big\}, \tag{68}$$

$$\boldsymbol{h}_n \triangleq \mathrm{col}\{\boldsymbol{x}_{1,n}z_{1,n}, \boldsymbol{x}_{2,n}z_{2,n}, \cdots, \boldsymbol{x}_{N,n}z_{N,n}\}. \tag{69}$$

Subtracting $\boldsymbol{w}_k^{\star}$ from both sides of the second equation of (62) and using block notations, we obtain:

$$\tilde{\boldsymbol{w}}_{n+1} = \tilde{\boldsymbol{\psi}}_{n+1} - \boldsymbol{\gamma}_{n+1}^{(i)}. \tag{70}$$

Combining (65) and (70) leads to:

$$\tilde{\boldsymbol{w}}_{n+1} = \boldsymbol{B}_n\tilde{\boldsymbol{w}}_n + \boldsymbol{U}\boldsymbol{h}_n - \boldsymbol{\gamma}_{n+1}^{(i)}. \tag{71}$$

Using **A1** and taking the expectation of (71), we obtain:

$$\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\} = \boldsymbol{B}\mathbb{E}\{\tilde{\boldsymbol{w}}_n\} - \mathbb{E}\{\boldsymbol{\gamma}_{n+1}^{(i)}\}, \tag{72}$$

where

$$\boldsymbol{B} \triangleq \mathbb{E}\{\boldsymbol{B}_n\} = \boldsymbol{I} - \boldsymbol{U}\boldsymbol{M}, \tag{73}$$

$$\big[\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1}^{(1)}\}\big]_m = \begin{cases} -\mu_k\lambda_k, & \text{if } \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m \le -\mathbb{E}\{[\boldsymbol{w}_k^o]_m\} - \mu_k\lambda_k \\ \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m + \mathbb{E}\{[\boldsymbol{w}_k^o]_m\}, & \text{if } -\mathbb{E}\{[\boldsymbol{w}_k^o]_m\} - \mu_k\lambda_k < \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m \le -\mathbb{E}\{[\boldsymbol{w}_k^o]_m\} \\ 0, & \text{if } \big|\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m\big| < \mathbb{E}\{[\boldsymbol{w}_k^o]_m\} \\ \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m - \mathbb{E}\{[\boldsymbol{w}_k^o]_m\}, & \text{if } \mathbb{E}\{[\boldsymbol{w}_k^o]_m\} \le \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m < \mathbb{E}\{[\boldsymbol{w}_k^o]_m\} + \mu_k\lambda_k \\ \mu_k\lambda_k, & \text{if } \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m \ge \mathbb{E}\{[\boldsymbol{w}_k^o]_m\} + \mu_k\lambda_k \end{cases} \tag{77}$$

with

$$\boldsymbol{M} \triangleq \mathbb{E}\{\boldsymbol{M}_n\} = \text{diag}\big\{\boldsymbol{R}_{x,1}, \boldsymbol{R}_{x,2}, \cdots, \boldsymbol{R}_{x,N}\big\}, \tag{74}$$

$$\mathbb{E}\{\boldsymbol{\gamma}_{n+1}^{(i)}\} = \text{col}\big\{\mathbb{E}\{\boldsymbol{\gamma}_{1,n+1}^{(i)}\}, \cdots, \mathbb{E}\{\boldsymbol{\gamma}_{N,n+1}^{(i)}\}\big\}. \tag{75}$$

To analyze iteration (72), we need to derive the explicit expression of $\mathbb{E}\{\boldsymbol{\gamma}_{n+1}^{(i)}\}$ firstly. Since it has different expressions for $\ell_{\infty,1}$-norm and reweighted $\ell_{\infty,1}$-norm, we shall now evaluate it separately.

1) $\ell_{\infty,1}$-norm: Taking the expectation of (55), (56), and approximating the inequality conditions by conditions on the expectation terms, the m-th entry of $\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1}^{(1)}\}$ is given by:

$$\big[\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1}^{(1)}\}\big]_m = \begin{cases} -\mu_k\lambda_k, \text{ if } \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m < -\mu_k\lambda_k \\ \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m, \text{ if } \big|\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m\big| \le \mu_k\lambda_k \\ \mu_k\lambda_k, \text{ if } \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m > \mu_k\lambda_k \end{cases} \tag{76}$$

if $\mathbb{E}\{[\boldsymbol{w}_k^o]_m\} \le \tau_1$, and it is defined by (77) if $\mathbb{E}\{[\boldsymbol{w}_k^o]_m\} > \tau_1$. To facilitate the derivation of the transient mean behavior in (76) and (77), we approximate the quantities appeared in "if" conditions by their corresponding expectations. It is difficult, if not impossible, to conduct theoretical analysis without this approximation. Note that this approximation is only used in the transient mean behavior analysis and have not been used in any other places. It is observed that the theoretical transient mean behavior obtained with this approximation match well with the results obtained via the Monte-Carlo simulation in Fig. 2 of Section VI-A. Vector $\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}$ is the k-th block vector of $\mathbb{E}\{\boldsymbol{\psi}_{n+1}\}$, which is evaluated as follows:

$$\mathbb{E}\{\boldsymbol{\psi}_{n+1}\} = \mathbb{E}\{\tilde{\boldsymbol{\psi}}_{n+1}\} + \boldsymbol{w}^\star, \tag{78}$$

with:

$$\mathbb{E}\{\tilde{\boldsymbol{\psi}}_{n+1}\} = \boldsymbol{B}\mathbb{E}\{\tilde{\boldsymbol{w}}_n\}. \tag{79}$$

On the other hand, $\mathbb{E}\{[\boldsymbol{w}_k^o]_m\}$ is approximated by the maximal value of $\big|\big[\mathbb{E}\{\boldsymbol{\psi}_{\ell,n+1}\}\big]_m\big|$ for all $\ell \in \mathcal{N}_k^-$.

2) Reweighted $\ell_{\infty,1}$-norm: As the distribution of $[\boldsymbol{w}_k^{(t)}]_m$ is unknown, we cannot evaluate exactly the expectations involving $[\boldsymbol{w}_k^{(t)}]_m$ in (57) and (58). We therefore consider the first-order Taylor series expansion as in [49]–[51]; the performance results obtained with the first-order Taylor series expansion match well with the Monte-Carlo results in Fig. 3 of Section VI-A. By expanding $f([\boldsymbol{w}_k^{(t)}]_m) \triangleq \mu_k\lambda_k/[\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|]$ around $\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}$ and taking expectation, we obtain the following approximation:

$$\mathbb{E}\bigg\{\frac{\mu_k\lambda_k}{\varepsilon + |[\boldsymbol{w}_k^{(t)}]_m|}\bigg\} \approx \frac{\mu_k\lambda_k}{\varepsilon + |\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}|}. \tag{80}$$

Taking the expectation of (57), (58) and using (80), as well as approximating the inequality conditions by conditions on the expectation terms, we obtain the explicit expression of $\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1}^{(2)}\}$, with its m-th entry given by (83) or (84) when $\mathbb{E}\{[\boldsymbol{w}_k^o]_m\} \le \tau_2$ and $\mathbb{E}\{[\boldsymbol{w}_k^o]_m\} > \tau_2$, respectively, with quantities:

$$\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\} = \big[\mathbb{E}\{\boldsymbol{w}_{k,n}\}\big]_m \tag{81}$$

$$\mathbb{E}\{[\boldsymbol{c}_k]_m\} = \mathbb{E}\{[\boldsymbol{w}_k^o]_m\} + \frac{\mu_k\lambda_k}{\varepsilon + \big|\big[\mathbb{E}\{\boldsymbol{w}_{k,n}\}\big]_m\big|}. \tag{82}$$

Note that several approximations, such as the first-order Taylor series expansion [49]–[51] and approximated the inequality conditions, are only used in the transient mean behavior of the reweighted $\ell_{\infty,1}$-norm, not in any other places. It is observed that the theoretical transient mean behavior obtained with these approximations match well with the results obtained via the Monte-Carlo simulation in Fig. 3 of Section VI-A.

From iteration (72), we obtain the following **Theorem 1** for the mean stability of proximal multitask diffusion LMS algorithm (19).

$$\big[\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1}^{(2)}\}\big]_m=\begin{cases}\dfrac{-\mu_k\lambda_k}{\varepsilon+\big|\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}\big|},\text{if }\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m<\dfrac{-\mu_k\lambda_k}{\varepsilon+\big|\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}\big|}\\ \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m,\text{if }\Big|\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m\Big|\le\dfrac{\mu_k\lambda_k}{\varepsilon+\big|\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}\big|}\\ \dfrac{\mu_k\lambda_k}{\varepsilon+\big|\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}\big|},\text{if }\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m>\dfrac{\mu_k\lambda_k}{\varepsilon+\big|\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}\big|}\end{cases}\tag{83}$$

$$\big[\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1}^{(2)}\}\big]_m=\begin{cases}-\dfrac{\mu_k\lambda_k}{\varepsilon+\big|\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}\big|},\text{ if }\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m\le-\mathbb{E}\{[\boldsymbol{c}_k]_m\}\\ \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m+\mathbb{E}\{[\boldsymbol{w}_k^o]_m\},\text{ if }-\mathbb{E}\{[\boldsymbol{c}_k]_m\}<\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m\le-\mathbb{E}\{[\boldsymbol{w}_k^o]_m\}\\ 0,\text{ if }\big|\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m\big|<\mathbb{E}\{[\boldsymbol{w}_k^o]_m\}\\ \big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m-\mathbb{E}\{[\boldsymbol{w}_k^o]_m\},\text{ if }\mathbb{E}\{[\boldsymbol{w}_k^o]_m\}\le\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m<\mathbb{E}\{[\boldsymbol{c}_k]_m\}\\ \dfrac{\mu_k\lambda_k}{\varepsilon+\big|\mathbb{E}\{[\boldsymbol{w}_k^{(t)}]_m\}\big|},\text{ if }\big[\mathbb{E}\{\boldsymbol{\psi}_{k,n+1}\}\big]_m\ge\mathbb{E}\{[\boldsymbol{c}_k]_m\}\end{cases}\tag{84}$$

Theorem 1. *(Mean stability)* Assume data model (1) and assumption **A1** hold. Then for any initial conditions, distributed networks endowed with the proximal multitask diffusion LMS algorithm (19) are stable in the mean, if step-sizes μ_k satisfy:

$$0<\mu_k<\frac{2}{\lambda_{\max}\{\boldsymbol{R}_{x,k}\}},\quad k=1,\cdots,N,\tag{85}$$

where $\lambda_{\max}\{\cdot\}$ denotes the maximal eigenvalue of its matrix argument. The block maximum norm of the bias can be upper bounded as:

$$\lim_{n\to\infty}\big\|\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\}\big\|_{b,\infty}\le\frac{\sqrt{L}\cdot\max_k\{\mu_k\lambda_k\}}{1-\|\boldsymbol{B}\|_{b,\infty}}\tag{86}$$

$$\lim_{n\to\infty}\big\|\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\}\big\|_{b,\infty}\le\frac{1}{\varepsilon}\frac{\sqrt{L}\cdot\max_k\{\mu_k\lambda_k\}}{1-\|\boldsymbol{B}\|_{b,\infty}}\tag{87}$$

for the $\ell_{\infty,1}$-norm and reweighted $\ell_{\infty,1}$-norm, respectively.

Proof: By iterating the RHS of (72) from time instant $n=0$, and proving the convergence of the obtained series, we arrive at condition (85) for step-size to ensure the mean stability of (19). For more details, see Appendix D. ■

Remark 1: Equation (85) provides an upper bound for step-size μ_k to ensure the mean stability of the distributed networks with the proximal multitask diffusion LMS algorithm (19). The upper bound is closely related to the second-order statistics $\boldsymbol{R}_{x,k}$ of the input signals. Equations (86) and (87) indicate that the proximal multitask diffusion LMS algorithm (19) is biased. The upper bound of the biases are proportional to the length of the system vector L, the step-size μ_k and the regularization parameter λ_k. The bias can be reduced by using a sufficiently small step-size μ_k or regularization parameter λ_k. Besides, for the reweighted $\ell_{\infty,1}$-norm, the bias is also inversely proportional to parameter ε. In addition, the upper bound of bias for the reweighted $\ell_{\infty,1}$-norm has an improvement with a factor $\frac{1}{\varepsilon}$ than that of the $\ell_{\infty,1}$-norm.

B. Mean-square behavior analysis

Under **A1** and using (71), then for any semi-positive definite matrix $\boldsymbol{\Sigma}$ of compatible dimension, the weighted mean-square behavior of $\tilde{\boldsymbol{w}}_{n+1}$ evaluates as:

$$\begin{aligned}\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\Sigma}}^2\}&=\mathbb{E}\{\|\tilde{\boldsymbol{w}}_n\|_{\boldsymbol{\Sigma}'}^2\}+\mathbb{E}\{\|\boldsymbol{U}\boldsymbol{h}_n\|_{\boldsymbol{\Sigma}}^2\}+\mathbb{E}\{\|\boldsymbol{\gamma}_{n+1}^{(i)}\|_{\boldsymbol{\Sigma}}^2\}\\&\quad-2\mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top\boldsymbol{B}_n^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\}-2\mathbb{E}\{\boldsymbol{h}_n^\top\boldsymbol{U}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\},\end{aligned}\tag{88}$$

where $\|\boldsymbol{x}\|_{\boldsymbol{\Sigma}}^2\triangleq\boldsymbol{x}^\top\boldsymbol{\Sigma}\boldsymbol{x}$, and

$$\boldsymbol{\Sigma}'\triangleq\mathbb{E}\{\boldsymbol{B}_n^\top\boldsymbol{\Sigma}\boldsymbol{B}_n\}.\tag{89}$$

Let $\boldsymbol{\sigma}\triangleq\text{vec}\{\boldsymbol{\Sigma}\}$ and $\boldsymbol{\sigma}'\triangleq\text{vec}\{\boldsymbol{\Sigma}'\}$, where $\text{vec}\{\cdot\}$ operator stacks the columns of its matrix argument on top of each other. Using the property of $\text{vec}\{\cdot\}$ operator, (89) becomes:

$$\boldsymbol{\sigma}'=\mathbb{E}\{\boldsymbol{B}_n^\top\otimes\boldsymbol{B}_n^\top\}\boldsymbol{\sigma}.\tag{90}$$

Under **A2** and ignoring terms on the second-order of the maximal step-size, we have the approximation for (90):

$$\boldsymbol{\sigma}' \approx \boldsymbol{F}\boldsymbol{\sigma} \tag{91}$$

with $\boldsymbol{F} \triangleq \boldsymbol{B}^\top \otimes \boldsymbol{B}^\top$. Define:

$$\boldsymbol{H} \triangleq \boldsymbol{U}\text{diag}\{\sigma_{z,1}^2\boldsymbol{R}_{x,1}, \cdots, \sigma_{z,N}^2\boldsymbol{R}_{x,N}\}\boldsymbol{U}^\top. \tag{92}$$

We then have:

$$\mathbb{E}\{\|\boldsymbol{U}\boldsymbol{h}_n\|_{\boldsymbol{\Sigma}}^2\} = [\text{vec}\{\boldsymbol{H}\}]^\top\boldsymbol{\sigma}. \tag{93}$$

To make the analysis tractable, we adopt approximation:

$$\mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top\boldsymbol{B}_n^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\} \approx \mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top\boldsymbol{B}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\}. \tag{94}$$

Since $|\boldsymbol{\gamma}_{k,n+1}^{(1)}| \preceq \mu_k\lambda_k\mathbb{1}_L$ and $|\boldsymbol{\gamma}_{k,n+1}^{(2)}| \preceq \frac{\mu_k\lambda_k}{\varepsilon}\mathbb{1}_L$, we conclude that $\boldsymbol{\gamma}_{n+1}^{(i)}$ is at most of the same order as the step-size. This implies that the last term on the RHS of (88) contains higher-order powers of the step-size, and can be ignored according to assumption **A2**. Finally, by using (91), (93) and (94), expression (88) becomes:

$$\begin{aligned}\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\sigma}}^2\} =& \mathbb{E}\{\|\tilde{\boldsymbol{w}}_n\|_{\boldsymbol{F}\boldsymbol{\sigma}}^2\} + [\text{vec}\{\boldsymbol{H}\}]^\top\boldsymbol{\sigma} + \mathbb{E}\{\|\boldsymbol{\gamma}_{n+1}^{(i)}\|_{\boldsymbol{\sigma}}^2\} \\ &- 2\mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top\boldsymbol{B}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\},\end{aligned} \tag{95}$$

where we use the notations $\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\Sigma}}^2\}$ and $\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\sigma}}^2\}$ interchangeably.

From iteration (95), we obtain the following **Theorem 2** to ensure the mean-square stability of the proximal multitask diffusion LMS algorithm (19).

Theorem 2. *(Mean-square stability)* Assume data model (1) and assumptions **A1**, **A2** hold. Further assume that approximation (91) is reasonable for sufficiently small step-sizes. Then for any initial conditions, distributed networks endowed with proximal multitask diffusion LMS algorithm (19) is stable in the mean-square sense, if the step-sizes μ_k are sufficiently small and satisfy (85).

Proof: By proving that the last two terms on the RHS of (95) is bounded, and iterating (95) from time instant $n = 0$, we obtain the condition for step-size to ensure the mean-square stability of (19). For more details, see Appendix E. ■

Remark 2: The weighted mean-square behavior of $\tilde{\boldsymbol{w}}_{n+1}$, that is $\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\Sigma}}^2\}$, can be decomposed as:

$$\begin{aligned}\mathbb{E}\big\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\Sigma}}^2\big\} &= \mathbb{E}\Big\{\big\|\boldsymbol{w}_{n+1} - \boldsymbol{w}^\star - \mathbb{E}\{\boldsymbol{w}_{n+1}\} + \mathbb{E}\{\boldsymbol{w}_{n+1}\}\big\|_{\boldsymbol{\Sigma}}^2\Big\} \\ &= \underbrace{\mathbb{E}\Big\{\big\|\tilde{\boldsymbol{w}}_{n+1} - \mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\}\big\|_{\boldsymbol{\Sigma}}^2\Big\}}_{\text{Variance term}} + \underbrace{\big\|\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\}\big\|_{\boldsymbol{\Sigma}}^2}_{\text{Bias term}}\end{aligned} \tag{96}$$

By substituting $\tilde{\boldsymbol{w}}_{n+1}$ of (71) and $\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\}$ of (72) into (96), we obtain an equivalent form of (95). Relation (96) is called the bias-variance decomposition. From (96), we observe that the stability of $\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\Sigma}}^2\}$ in **Theorem 2** ensures the stabilities of both the Variance term and the Bias term $\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\}$ of **Theorem 1**.

VI. Simulation Results

In this section, we present simulation results to validate the effectiveness of the algorithm. With the exception of the simulation results presented in Section VI-A, used to validate theoretical results in the mean behavior analysis and obtained by averaging over 500 independent Monte-Carlo runs, all other simulated curves were obtained by averaging over 100 independent Monte-Carlo runs.

A. Theoretical validation

We considered a connected network consisting of 16 nodes and 36 edges. The number of edges at each node was between 2 and 7, without taking into account the possible self-loops connecting each node to itself. Other characteristics of the network are listed in Table I. Each regressor $\boldsymbol{x}_{k,n}$ was generated from a zero-mean Gaussian distribution with covariance matrix $\boldsymbol{R}_{x,k} = \sigma_{x,k}^2\boldsymbol{I}_{30}$. Each additive noise $z_{k,n}$ was generated from a zero-mean Gaussian distribution with variance $\sigma_{z,k}^2$. Variances $\sigma_{x,k}^2$ and $\sigma_{z,k}^2$ at each node were generated randomly from a Gaussian distribution as shown in Fig. 1. Note that these variances settings are the same as in [12], [17].

To validate the theoretical results reported in the mean behavior analysis, we considered a stationary system identification scenario. The unknown system coefficients $\boldsymbol{w}_k^\star$ were generated such that the entire network has a jointly sparse structure with sparsity degree of $10/30$. Each nonzero element of $\boldsymbol{w}_k^\star$ was generated independently from a standard Gaussian distribution. The simulation results presented in this section were obtained by averaging over 500 independent runs.

TABLE I

CHARACTERISTICS OF THE NETWORK USED FOR MODEL VALIDATION. $\boldsymbol{L}$ IS THE LAPLACIAN MATRIX ASSOCIATED WITH THE GRAPH, $\lambda_2(\boldsymbol{L})$ IS THE ALGEBRAIC CONNECTIVITY [52] OF THE GRAPH, SIZE IS THE NUMBER OF NODES, DENSITY IS THE NUMBER OF NON-ZERO ENTRIES OF THE ADJACENCY MATRIX, AND DIAMETER IS THE MAXIMUM DISTANCE BETWEEN ANY TWO NODES [53].

Network	Size	Density	$\lambda_2(\boldsymbol{L})$	Diameter
Net	16	34%	0.9983	4

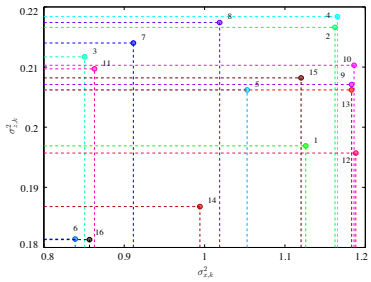


Fig. 1. Agent input and noise variances.

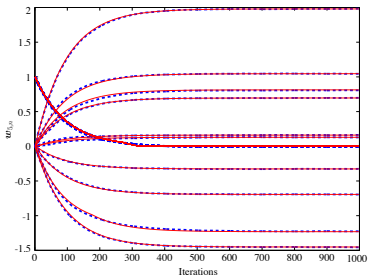


Fig. 2. Validation of the mean behavior model for $\ell_{\infty,1}$-regularization; Theory: red solid line; Monte-Carlo simulations: blue dashed line.

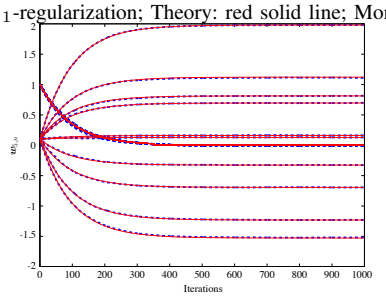


Fig. 3. Validation of the mean behavior model in the case of the reweighted $\ell_{\infty,1}$-regularization; Theory: red solid line; Monte-Carlo: blue dashed line.

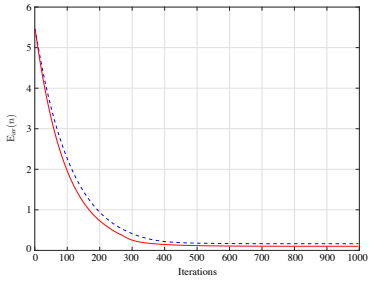


Fig. 4. The average error $\mathrm{E}_{\mathrm{av}}(n)$ of $\ell_{\infty,1}$-regularization; Theory: red solid line; Monte-Carlo simulations: blue dashed line.

The results are illustrated in Fig. 2 and Fig. 3 for the proximal multitask LMS with $\ell_{\infty,1}$-regularization and reweighted $\ell_{\infty,1}$-regularization, respectively. The good match between the theoretical results and the Monte-Carlo curves illustrates the accuracy of theoretical results in the mean behavior analysis.

We also considered the average error over the entire network defined by:

$$\mathrm{E}_{\mathrm{av}}(n) \triangleq \frac{1}{N}\sum_{k=1}^{N}\|\boldsymbol{w}_{k,n}-\boldsymbol{w}_k^{\star}\| \tag{97}$$

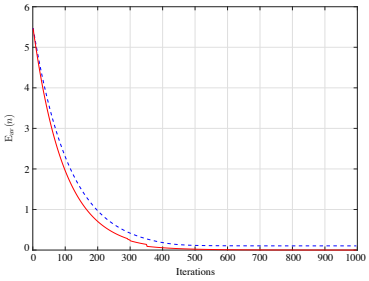


Fig. 5. The average error $\mathrm{E_{av}}(n)$ of the reweighted $\ell_{\infty,1}$-regularization; Theory: red solid line; Monte-Carlo simulations: blue dashed line.

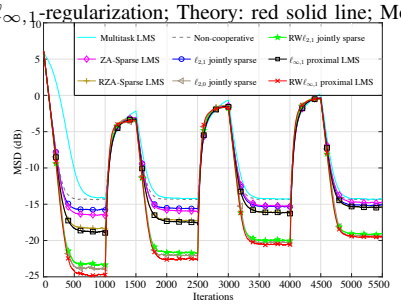


Fig. 6. Comparison of the proposed algorithms with several state-of-the-art algorithms for white inputs.

Though there is a small bias between theoretical and Monte-Carlo curves, the results reported in Fig. 4 and Fig. 5 confirm the accuracy of the theoretical models.

B. Numerical Simulations

1) Comparison with existing algorithms: We firstly considered a non-stationary jointly sparse system identification scenario with $\boldsymbol{w}_k^\star$ varying over time. Each nonzero entry of $\boldsymbol{w}_k^\star$ was generated independently from a standard Gaussian distribution. The evolution of $\boldsymbol{w}_k^\star$ was divided into four stationary stages and three transient stages. During stationary stages, sparse vectors $\boldsymbol{w}_k^\star$ were set to sparsity degree of $3/30$, $5/30$, $8/30$ and $10/30$, respectively. The transient stages were designed by using linear interpolation over 500 time instants. The regressors $\boldsymbol{x}_{k,n}$ were generated as those in Section VI-A for white inputs, while they were generated according to a zero-mean Gaussian distribution with covariance matrix $\boldsymbol{R}_{x,k} = \sigma_{x,k}^2 \boldsymbol{R}^\dagger$ for colored inputs, where $\boldsymbol{R}^\dagger$ is an 30×30 Hermite matrix with eigenvalue spread $\lambda_{\max}\{\boldsymbol{R}^\dagger\}/\lambda_{\min}\{\boldsymbol{R}^\dagger\} = 21$, and symbol $\lambda_{\min}\{\cdot\}$ denotes the minimal eigenvalue of its matrix argument. For comparison purpose, non-cooperative diffusion LMS algorithm, non-cooperative sparse diffusion LMS [6] with zero-attracting (ZA) regularizer and reweighted zero-attracting (RZA) regularizer, multitask diffusion LMS with adaptive combiner [17] and jointly sparse multitask diffusion LMS [24] with $\ell_{2,1}$-regularization, reweighted $\ell_{2,1}$-regularization ($\mathrm{RW}\ell_{2,1}$) and $\ell_{2,0}$-regularization were taken into consideration. We adopted a uniform step-size 0.01 for all algorithms. For these algorithms, we set their parameters so that they reach their best performance. To enable reproducible research, details about parameters used by these algorithms can be found in Table III of Appendix F.

The results are illustrated in Fig. 6 for white inputs. We observe that multitask LMS with adaptive combiner is the worst one among all competing algorithms, since it utilizes similarities between neighboring nodes to improve estimation accuracy. This does not necessarily exist in jointly sparse scenarios and may deteriorate the MSD performance. Since jointly sparse system can be regarded as a special case of general sparse systems, by using additional information about system sparsity, sparse diffusion LMS with ZA regularizer and RZA regularizer have better performance than the non-cooperative LMS. Similarly, all jointly sparse multitask algorithms considered in this comparative experiment perform better than the non-cooperative LMS. Observe that sparse diffusion LMS algorithms can perform slightly better than jointly sparse diffusion LMS algorithms as the parameter vectors to estimate become sparser, while they do not perform as well when these parameter vectors are less sparse. These findings illustrate the interest of exploiting joint sparsity as prior information. On the one hand, the proposed proximal multitask LMS with reweighted $\ell_{\infty,1}$-regularization performs better than all other algorithms as evidenced by its lowest steady-state MSD. On the other hand, the proposed proximal multitask LMS with $\ell_{\infty,1}$-regularization has similar performance to the sparse diffusion LMS with RZA regularizer when there are more zeros in the jointly sparse system to estimate, but the former performs better than the latter as the number of nonzero entries increases.

The results are illustrated in Fig. 7 for colored input signals. It can be observed that the convergence rate of all algorithms is slower than in the case of white inputs, and some algorithms have poorer performance than the non-cooperative algorithm when the parameter vectors to estimate are less sparse. Besides some conclusions that have been drawn for white inputs, we observe that the proximal multitask LMS algorithm with reweighted $\ell_{\infty,1}$-regularization still has the best performance.

Then we examined the performance of all algorithms as a function of the sparsity degree of the jointly sparse system to estimate. We considered the steady-state MSD as the measure of performance in this experiment. The results are illustrated in

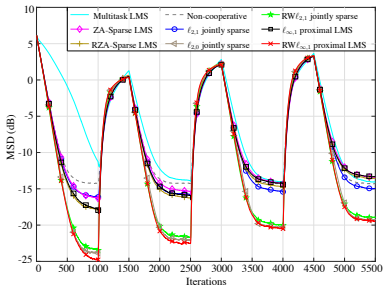


Fig. 7. Comparison of the proposed algorithms with several state-of-the-art algorithms for colored inputs.

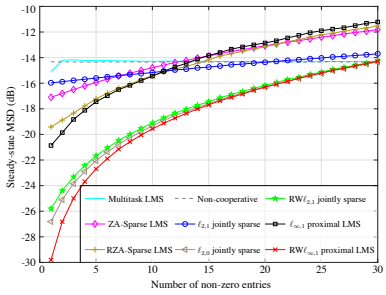


Fig. 8. Steady-state MSDs of all algorithms as a function of the sparsity degree.

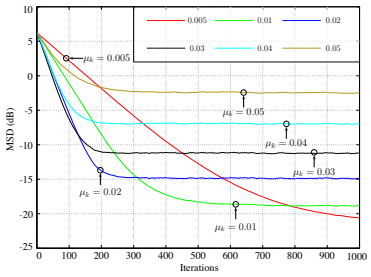


Fig. 9. Effect of the step-size μ_k on the MSD of the proximal LMS algorithm with $\ell_{\infty,1}$-regularization.

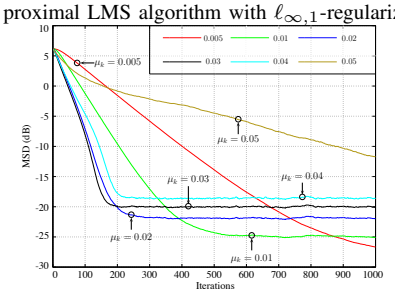


Fig. 10. Effect of the step-size μ_k on the MSD of the proximal LMS algorithm with reweighted $\ell_{\infty,1}$-regularization.

Fig. 8. Besides some conclusions that have been drawn before, we observe that some of the multitask algorithms with sparse and jointly sparse regularizers may have poorer performance than the non-cooperative LMS when the number of nonzero entries increases. Three algorithms, including the proximal multitask LMS with reweighted $\ell_{\infty,1}$-regularization, uniformly show better performance than all other algorithms for all sparsity degrees, and have similar performance to the non-cooperative LMS for totally non-sparse systems.

2) Effects of parameters setting: To examine the effects of parameters setting, including the step-size μ_k, the regularization parameter λ_k, the threshold values τ_1, τ_2 and the parameter ε, we considered a stationary system identification problem with sparsity degree of $3/30$. Each parameter was set to a same value for all nodes in the network. We examined the influence of one selected parameter at a time, setting all other parameters to fixed values, in order to facilitate comparison.

The effects of the step-size μ_k are illustrated in Fig. 9 and Fig. 10 for the proximal LMS with $\ell_{\infty,1}$-regularization and reweighted $\ell_{\infty,1}$-regularization, respectively. Observe that μ_k allows to control the trade-off between convergence rate and steady-state performance. A larger step-size results in a faster convergence rate at the cost of a larger steady-state MSD. A small step-size results in a more accurate estimation at the cost of a slower convergence speed.

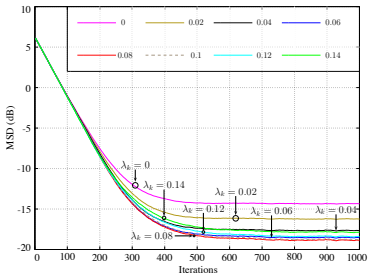


Fig. 11. Effect of the regularization parameter λ_k on the MSD of the proximal LMS algorithm with $\ell_{\infty,1}$-regularization.

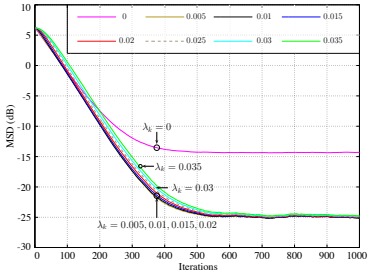


Fig. 12. Effect of the regularization parameter λ_k on the MSD of the proximal LMS algorithm with reweighted $\ell_{\infty,1}$-regularization.

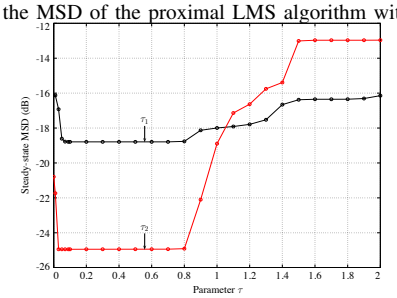


Fig. 13. Effect of parameters τ_1 and τ_2 on the steady-state MSD of the proximal LMS algorithm with $\ell_{\infty,1}$-regularization (black) and reweighted $\ell_{\infty,1}$-regularization (red), respectively.

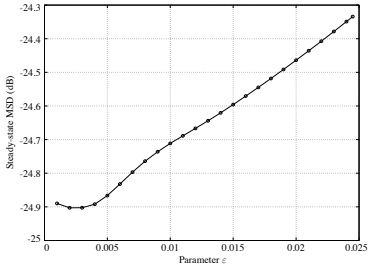


Fig. 14. Effect of parameter ε on the steady-state MSD of the proximal LMS algorithm with reweighted $\ell_{\infty,1}$-regularization.

The effects of the regularization parameter λ_k are illustrated in Fig. 11 and Fig. 12. We observe that increasing λ_k improves the convergence speed and the MSD at steady-state at first, and then degrades them. In this experiment, the critical values were 0.08 and 0.01 for the $\ell_{\infty,1}$ regularizer and the reweighted $\ell_{\infty,1}$ regularizer, respectively.

The effects of parameters τ_1 and τ_2 are shown in Fig. 13. Similar behaviors can be observed on the MSD at steady-state of both algorithms with respect to τ_1 and τ_2. The best values were obtained over interval $[0.05, 0.8]$ for both regularizers. First, these results show the need for introducing these two parameters. Second, they show that these two parameters can be appropriately selected over a large interval. Consider the $\ell_{\infty,1}$-norm regularizer. When $[\boldsymbol{w}_k^o]_m \leq \tau_1$, observe that the approximate proximal operator is given by (33) and corresponds to the ℓ_1-norm regularizer, which enjoys the following properties: On the one hand, it shrinks the estimates of zero-valued parameters to zero when $[\boldsymbol{w}_k^o]_m \leq \tau_1$; On the other hand, for small nonzero-valued entries, though introducing a bias when shrunk to zero, it lowers the estimates variance and results in a satisfying performance. Since jointly sparse systems are a special case of general sparse ones, for which the ℓ_1-norm regularizer is prescribed, it is expected that our algorithm works well for a large range of τ_1 values. The same reasoning applies to the reweighted $\ell_{\infty,1}$-norm regularizer and τ_2.

The effects of parameter ε in the reweighted $\ell_{\infty,1}$ regularizer are illustrated in Fig. 14. We set both parameters μ_k and λ_k to 0.01. We observe that the best value of ε resulting in the lowest MSD at steady-state was 0.002, but its effect on the

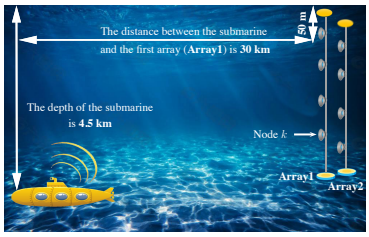


Fig. 15. Experimental setup for the practical application.

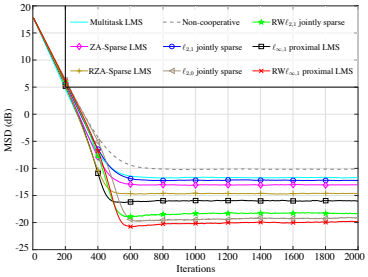


Fig. 16. Comparison of the proposed algorithms with several state-of-the-art algorithms for practical application.

steady-state MSD is weaker than the other parameters.

C. Practical application

We shall now validate the proposed algorithm in a practical application. This application follows the experimental setup described in [24]. In demodulation and decoding in underwater and wireless communication networks, a standalone low-cost sensor may not be able to decode or demodulate the source signal reliably due to the extra low signal-to-noise ratio (SNR) condition. It is better to adopt multiple sensors to collaboratively recover the information coming from a same source [54], [55] by combining and exchanging information within the sensor network. Practically, it is necessary to estimate the channel impulse responses between the source and each sensor before decoding. On the one hand, it has been shown that real-world underwater channels [27], and wireless communication channel [28], are sparse with large delay spread. On the other hand, the channel supports for neighboring antennas or nodes are approximately the same [28]. Indeed the times of arrival for closely spaced nodes and antennas are quite close, though the tap weights are different [24]. We shall now check that taking the jointly sparse property into consideration can improve the network performance.

Consider the problem of identifying underwater acoustic channels. We set the scenario presented in Fig. 15, where we have one source (the submarine) and two receiving sensor arrays. Both arrays are linear arrays and consist of 10 nodes with the same distance (3.75 m) between each of them. We assume that each node has the ability to communicate and process data. These two arrays are positioned in parallel with interspace 50 m, which gives us the distributed network with 20 nodes. Within each array, the nodes are connected one by one in a chain, and the corresponding pairs of nodes between two arrays are also connected. The underwater acoustic channels to estimate were generated via the BELLHOP model [56], which has been developed for predicting acoustic pressure fields in ocean environments. We used a white signal as input, and the additive noise at each node was white Gaussian. SNR conditions are listed in Table II.

TABLE II
SNR LEVEL IN DECIBEL (DB) FOR THE PRACTICAL APPLICATION. SINCE IT VARIES ACCORDING TO NODES, WE ENUMERATE THE MAXIMUM, MINIMUM AND MEAN VALUES.

SNR Level	Maximum	Minimum	Mean
SNR	7.89	6.19	7.07

We compared our algorithms with the algorithms considered in Section VI-B1. Parameters used by all them are listed in Table III. The results are depicted in Fig. 16. As can be observed, the proposed proximal multitask diffusion LMS algorithm with reweighted $\ell_{\infty,1}$ regularizer achieved the best performance in terms of the steady-state MSD.

VII. CONCLUSION

We considered the problem of estimating a set of parameter vectors in a distributed manner, where the local solutions have the same sparse support. We devised a proximal diffusion algorithm with (reweighted) $\ell_{\infty,1}$-norm regularization, with closed-form expressions for the regularizers. We conducted theoretical analyses of the algorithms behavior in the mean and mean-square sense. Simulation results illustrated the effectiveness of the proposed algorithms, as well as the accuracy of theoretical results. Integrating weighted network connection information to enhance this jointly-sparse estimation will be considered in future work.

APPENDIX A
INTERPRETATIONS ABOUT THE (REWEIGHTED) $\ell_{\infty,1}$-NORM

The $\ell_{\infty,1}$-norm of matrix $\boldsymbol{W}_k$ is defined as $g_1(\boldsymbol{w}_k)$ of (11). We focus on the interpretation of $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty$. From (8) we have that $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty = \max\{|[\boldsymbol{w}_k]_m|, |[\boldsymbol{w}_\ell^\star]_m| \text{ with } \ell \in \mathcal{N}_k^-\}$. Denote the maximal value of $|[\boldsymbol{w}_\ell^\star]_m|$ for $\ell \in \mathcal{N}_k^-$ as $[\boldsymbol{w}_k^o]_m$. Then $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty$ writes to $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty = \max\{|[\boldsymbol{w}_k]_m|, [\boldsymbol{w}_k^o]_m\}$. Now we consider the following three cases:

- Case a: The m-th row $\bar{\boldsymbol{w}}_{k,m}$ of matrix $\boldsymbol{W}_k$ corresponds to the zero-valued entries. In this case, $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty = \max\{|[\boldsymbol{w}_k]_m|, 0\} = \max\{|[\boldsymbol{w}_k]_m|\}$. Minimizing $g_1(\boldsymbol{w}_k)$ over $[\boldsymbol{w}_k]_m$ attracts the m-th entry $[\boldsymbol{w}_k]_m$ to zero.
- Case b: The m-th row $\bar{\boldsymbol{w}}_{k,m}$ corresponds to the non-zero valued entries, and $|[\boldsymbol{w}_k]_m| \leq [\boldsymbol{w}_k^o]_m$. In this case, $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty = \max\{|[\boldsymbol{w}_k]_m|, [\boldsymbol{w}_k^o]_m\} = [\boldsymbol{w}_k^o]_m$. Minimizing $g_1(\boldsymbol{w}_k)$ over $[\boldsymbol{w}_k]_m$ will not penalize $[\boldsymbol{w}_k]_m$.
- Case c: The m-th row $\bar{\boldsymbol{w}}_{k,m}$ corresponds to the non-zero valued entries, and $|[\boldsymbol{w}_k]_m| > [\boldsymbol{w}_k^o]_m$. In this case, $\|\bar{\boldsymbol{w}}_{k,m}\|_\infty = \max\{|[\boldsymbol{w}_k]_m|, [\boldsymbol{w}_k^o]_m\} = |[\boldsymbol{w}_k]_m|$. Minimizing $g_1(\boldsymbol{w}_k)$ over $[\boldsymbol{w}_k]_m$ will penalize $[\boldsymbol{w}_k]_m$ until $|[\boldsymbol{w}_k]_m| \leq [\boldsymbol{w}_k^o]_m$, which then becomes Case b.

Thus, the $\ell_{\infty,1}$-norm promotes the similarity between $|[\boldsymbol{w}_k]_m|$ and $[\boldsymbol{w}_k^o]_m$ for all m. Similar interpretations can be obtained for the reweighted $\ell_{\infty,1}$-norm of (12).

APPENDIX B
DERIVATION OF (36)

The initial problem (24) leads to (25) and (27) in Case 1 and Case 2, respectively. For ease of presentation, we define:

$$\bar{J}_1([\boldsymbol{w}_k]_m) \triangleq [\boldsymbol{w}_k^o]_m + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2 \tag{98}$$

$$\bar{J}_2([\boldsymbol{w}_k]_m) \triangleq |[\boldsymbol{w}_k]_m| + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2. \tag{99}$$

We select solution (26) or (35) depending on the value taken by costs (98) and (99). To save space, we partially present the derivation. The rest of the derivation can be obtained by following the same routine.

When $[\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k$, substituting $\hat{w} = [\boldsymbol{w}_k^o]_m$ of Case 1 into (98), we obtain:

$$\bar{J}_1([\boldsymbol{w}_k^o]_m) = [\boldsymbol{w}_k^o]_m + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k^o]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2. \tag{100}$$

Substituting $\hat{w} = [\boldsymbol{\psi}_{k,n+1}]_m - \mu_k\lambda_k$ of Case 2 into (99), we obtain:

$$\begin{aligned}\bar{J}_2(\hat{w}) &= |\hat{w}| + \frac{1}{2\mu_k\lambda_k}\big(\hat{w} - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2 \\ &\leq \bar{J}_2([\boldsymbol{w}_k^o]_m) = \bar{J}_1([\boldsymbol{w}_k^o]_m)\end{aligned} \tag{101}$$

since $\bar{J}_2(\hat{w})$ with $\hat{w} = [\boldsymbol{\psi}_{k,n+1}]_m - \mu_k\lambda_k$ is the minimal cost. Thus the proximal operator is given by:

$$\hat{w} = [\boldsymbol{\psi}_{k,n+1}]_m - \mu_k\lambda_k \tag{102}$$

for $[\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k$.

When $[\boldsymbol{w}_k^o]_m \leq [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{w}_k^o]_m + \mu_k\lambda_k$, since $\hat{w}$ of both Case 1 and Case 2 is given by:

$$\hat{w} = [\boldsymbol{w}_k^o]_m, \tag{103}$$

we arrive at proximal operator (103) directly.

When $0 < [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{w}_k^o]_m$, substituting $\hat{w} = [\boldsymbol{\psi}_{k,n+1}]_m$ of Case 1 into (98), we obtain:

$$\bar{J}_1\big([\boldsymbol{\psi}_{k,n+1}]_m\big) = [\boldsymbol{w}_k^o]_m. \tag{104}$$

Substituting $\hat{w} = [\boldsymbol{w}_k^o]_m$ of Case 2 into (99), we obtain:

$$\begin{aligned}\bar{J}_2([\boldsymbol{w}_k^o]_m) &= [\boldsymbol{w}_k^o]_m + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k^o]_m - [\boldsymbol{\psi}_{k,n+1}]_m\big)^2 \\ &\geq \bar{J}_1\big([\boldsymbol{\psi}_{k,n+1}]_m\big).\end{aligned} \tag{105}$$

Thus the proximal operator is given by:

$$\hat{w} = [\boldsymbol{\psi}_{k,n+1}]_m \tag{106}$$

for $0 < [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{w}_k^o]_m$.

By following the same routine, we obtain the proximal operator for $[\boldsymbol{\psi}_{k,n+1}]_m \leq 0$. Finally, by combining all these results, we arrive at the expression of proximal operator (36) when $[\boldsymbol{w}_k^o]_m > 0$.

APPENDIX C
DERIVATION OF (53)

Define:

$$\bar{J}_3([\boldsymbol{w}_k]_m) \triangleq \log\Big(1+\frac{[\boldsymbol{w}_k^o]_m}{\varepsilon}\Big) + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2 \tag{107}$$

$$\bar{J}_4([\boldsymbol{w}_k]_m) \triangleq \log\Big(1+\frac{|[\boldsymbol{w}_k]_m|}{\varepsilon}\Big) + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2. \tag{108}$$

The initial problem (37) leads to $\bar{J}_3([\boldsymbol{w}_k]_m)$ and $\bar{J}_4([\boldsymbol{w}_k]_m)$ in Case 1 and Case 2, respectively. We select solution (39), (49) or (51) by comparing costs (107) and (108) as that in Section IV-A. We partially present the derivation to save space.

When $0<[\boldsymbol{\psi}_{k,n+1}]_m<[\boldsymbol{w}_k^o]_m$, substituting $\hat{w}=[\boldsymbol{\psi}_{k,n+1}]_m$ of Case 1 into (107), we obtain:

$$\bar{J}_3\big([\boldsymbol{\psi}_{k,n+1}]_m\big) = \log\Big(1+\frac{[\boldsymbol{w}_k^o]_m}{\varepsilon}\Big). \tag{109}$$

Substituting $\hat{w}=[\boldsymbol{w}_k^o]_m$ of Case 2 into (108), we obtain:

$$\begin{aligned}\bar{J}_4([\boldsymbol{w}_k^o]_m) &= \log\Big(1+\frac{[\boldsymbol{w}_k^o]_m}{\varepsilon}\Big) + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k^o]_m-[\boldsymbol{\psi}_{k,n+1}]_m\big)^2 \\ &\geq \bar{J}_3\big([\boldsymbol{\psi}_{k,n+1}]_m\big).\end{aligned} \tag{110}$$

This means that the proximal operator is given by:

$$\hat{w} = [\boldsymbol{\psi}_{k,n+1}]_m \tag{111}$$

for $0<[\boldsymbol{\psi}_{k,n+1}]_m<[\boldsymbol{w}_k^o]_m$.

When $[\boldsymbol{w}_k^o]_m \leq [\boldsymbol{\psi}_{k,n+1}]_m < [\boldsymbol{c}_k]_m$, since $\hat{w}$ of both Case 1 and Case 2 is given by:

$$\hat{w} = [\boldsymbol{w}_k^o]_m, \tag{112}$$

we arrive at proximal operator (112) directly.

When $[\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{c}_k]_m$, substituting $\hat{w}=[\boldsymbol{w}_k^o]_m$ of Case 1 into (107), we obtain:

$$\begin{aligned}\bar{J}_3([\boldsymbol{w}_k^o]_m) &= \log\Big(1+\frac{[\boldsymbol{w}_k^o]_m}{\varepsilon}\Big) + \frac{1}{2\mu_k\lambda_k}\big([\boldsymbol{w}_k^o]_m - \big[\boldsymbol{\psi}_{k,n+1}\big]_m\big)^2 \\ &= \bar{J}_4([\boldsymbol{w}_k^o]_m).\end{aligned} \tag{113}$$

Further, since $\hat{w}=[\boldsymbol{\psi}_{k,n+1}]_m-\mu_k\lambda_k/(\varepsilon+|[\boldsymbol{w}_k^{(t)}]_m|)$ of Case 2 is an approximation of the minimizer $\hat{w}^o$ of (41), and according to the relation $\bar{J}_4(\hat{w}^o) \leq \bar{J}_4([\boldsymbol{w}_k^o]_m)$, we define the proximal operator as:

$$\hat{w} = [\boldsymbol{\psi}_{k,n+1}]_m - \frac{\mu_k\lambda_k}{\varepsilon+|[\boldsymbol{w}_k^{(t)}]_m|} \tag{114}$$

for $[\boldsymbol{\psi}_{k,n+1}]_m \geq [\boldsymbol{c}_k]_m$.

By following the same routine, we obtain the proximal operator for $[\boldsymbol{\psi}_{k,n+1}]_m \leq 0$. Finally, by combining all these results, we arrive at expression (53) when $[\boldsymbol{w}_k^o]_m > 0$.

APPENDIX D
PROOF OF **THEOREM 1**

Iterating (72) from $n=0$, we obtain:

$$\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\} = \boldsymbol{B}^{n+1}\mathbb{E}\{\tilde{\boldsymbol{w}}_0\} - \sum_{j=0}^{n}\boldsymbol{B}^j\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\} \tag{115}$$

where $\mathbb{E}\{\tilde{\boldsymbol{w}}_0\}$ is the initial condition. The convergence of (115) requires that both terms on the RHS to be convergent. For the first term, it requires that spectral radius $\rho(\boldsymbol{B}) < 1$ to ensure the convergence. For the second term, it is sufficient to prove that $\sum_{j=0}^{n}\big[\boldsymbol{B}^j\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\}\big]_m$ is convergent for $m = 1, \cdots, NL$. A series is absolutely convergent if each term is bounded by a term of an absolutely convergent series [6], [14]. Define $s_m \triangleq \big[\boldsymbol{B}^j\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\}\big]_m$. Since the block maximum norm of a block vector is larger than or equal to the largest absolute value of its entry, we have:

$$\begin{aligned}|s_m| &\le \big\|\boldsymbol{B}^j\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\}\big\|_{b,\infty}\\ &\le \big\|\boldsymbol{B}\big\|_{b,\infty}^{j}\cdot\big\|\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\}\big\|_{b,\infty}\\ &\le \big[\rho(\boldsymbol{B})\big]^j\cdot\gamma_{\max}^{(i)},\end{aligned} \tag{116}$$

where $\|\cdot\|_{b,\infty}$ is the block maximum norm [7]. The quantity $\big\|\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\}\big\|_{b,\infty}$ is finite for all j and n, and bounded by some constant $\gamma_{\max}^{(i)}$. Actually, from (76), (77), (83), (84) and following the routine for the boundness of $\boldsymbol{\gamma}_{k,n+1}^{(i)}$, we obtain:

$$\big|\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1-j}^{(1)}\}\big| \preceq \mu_k\lambda_k\mathbb{1}_L \tag{117}$$

$$\big|\mathbb{E}\{\boldsymbol{\gamma}_{k,n+1-j}^{(2)}\}\big| \preceq b_k\mathbb{1}_L \tag{118}$$

where the quantity b_k has been defined in (61).

Thus, we conclude that $\big\|\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\}\big\|_{b,\infty}$ is bounded, with

$$\big\|\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(1)}\}\big\|_{b,\infty} \le \gamma_{\max}^{(1)} \triangleq \sqrt{L}\cdot\max_k\big\{\mu_k\lambda_k\big\} \tag{119}$$

$$\big\|\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(2)}\}\big\|_{b,\infty} \le \gamma_{\max}^{(2)} \triangleq \frac{\sqrt{L}}{\varepsilon}\cdot\max_k\big\{\mu_k\lambda_k\big\}. \tag{120}$$

Consequently, convergence of (116) is ensured by condition $\rho(\boldsymbol{B}) < 1$. Step-sizes μ_k satisfying (85) ensure the mean stability of the network.

With step-sizes μ_k satisfying (85) to ensure $\rho(\boldsymbol{B}) < 1$, the block maximum norm of the bias can be bounded as $n \to \infty$:

$$\begin{aligned}\lim_{n\to\infty}\big\|\mathbb{E}\{\tilde{\boldsymbol{w}}_{n+1}\}\big\|_{b,\infty} &\le \lim_{n\to\infty}\sum_{j=0}^{n}\|\boldsymbol{B}^j\|_{b,\infty}\big\|\mathbb{E}\{\boldsymbol{\gamma}_{n+1-j}^{(i)}\}\big\|_{b,\infty}\\ &\le \frac{\gamma_{\max}^{(i)}}{1-\|\boldsymbol{B}\|_{b,\infty}}.\end{aligned} \tag{121}$$

Substituting (119) and (120) into (121), we arrive at (86) and (87), respectively.

APPENDIX E
PROOF OF **THEOREM 2**

Since $\boldsymbol{\Sigma}$ is a positive semi-definite matrix, and vector $\boldsymbol{\gamma}_{n+1}^{(i)}$ is uniformly bounded for all time instant n and $i = 1$ or 2, we have:

$$0 \le \mathbb{E}\{\|\boldsymbol{\gamma}_{n+1}^{(i)}\|_{\boldsymbol{\Sigma}}^2\} \le \kappa_1^{(i)} \tag{122}$$

for all n, where $\kappa_1^{(i)}$ is a positive constant depending on the jointly sparse regularizer. Since $\boldsymbol{\gamma}_{n+1}^{(i)}$ is uniformly bounded, vector $\boldsymbol{B}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}$ is also bounded for all n. Denote the bound of the largest component of $2\boldsymbol{B}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}$ in absolute-value sense as $\tau_{\max}^{(i)}$ for all n. We obtain:

$$\begin{aligned}\big|2\mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top\boldsymbol{B}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\}\big| &\le \tau_{\max}^{(i)}\sum_{k=1}^{N}\sum_{m=1}^{L}\big|\mathbb{E}\big\{\big[\tilde{\boldsymbol{w}}_{k,n}\big]_m\big\}\big|\\ &= \tau_{\max}^{(i)}\cdot\big\|\mathbb{E}\{\tilde{\boldsymbol{w}}_n\}\big\|_1.\end{aligned} \tag{123}$$

Given step-sizes μ_k satisfying condition (85) to ensure stability in the mean sense, we conclude that quantity $\big\|\mathbb{E}\{\tilde{\boldsymbol{w}}_n\}\big\|_1$ is upper bounded by some constant $\kappa_2^{(i)}$ for all n. Consequently, (123) becomes:

$$\big|2\mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top\boldsymbol{B}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\}\big| \le \tau_{\max}^{(i)}\cdot\kappa_2^{(i)}. \tag{124}$$

Define

$$\phi\left(\tilde{\boldsymbol{w}}_n, \boldsymbol{\gamma}_{n+1}^{(i)}\right) \triangleq \mathbb{E}\{\|\boldsymbol{\gamma}_{n+1}^{(i)}\|_{\boldsymbol{\sigma}}^2\} - 2\mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top\boldsymbol{B}^\top\boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\}. \tag{125}$$

Using (122)–(125), we have:

$$\begin{aligned}\left|\phi\left(\tilde{\boldsymbol{w}}_n,\boldsymbol{\gamma}_{n+1}^{(i)}\right)\right| &\leq \mathbb{E}\{\|\boldsymbol{\gamma}_{n+1}^{(i)}\|_{\boldsymbol{\sigma}}^2\}+\left|2\mathbb{E}\{\tilde{\boldsymbol{w}}_n^\top \boldsymbol{B}^\top \boldsymbol{\Sigma}\boldsymbol{\gamma}_{n+1}^{(i)}\}\right| \\ &\leq \kappa_1^{(i)}+\tau_{\max}^{(i)}\cdot\kappa_2^{(i)} \end{aligned} \tag{126}$$

for all n. Given a weighting matrix $\boldsymbol{\Sigma}$, the positive constant $\kappa_3^{(i)} \triangleq \kappa_1^{(i)}+\tau_{\max}^{(i)}\cdot\kappa_2^{(i)}$ can be written as a scaled multiple of $[\text{vec}\{\boldsymbol{H}\}]^\top\boldsymbol{\sigma}$ as:

$$\kappa_3^{(i)} = p\cdot[\text{vec}\{\boldsymbol{H}\}]^\top\boldsymbol{\sigma} \tag{127}$$

with $p\geq 0$ [6]. Using (95) and (125)–(127), we obtain an upper bound of $\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\sigma}}^2\}$ as:

$$\begin{aligned}\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\sigma}}^2\} &\leq \mathbb{E}\{\|\tilde{\boldsymbol{w}}_n\|_{\boldsymbol{F}\boldsymbol{\sigma}}^2\}+[\text{vec}\{\boldsymbol{H}\}]^\top\boldsymbol{\sigma}+\left|\phi\left(\tilde{\boldsymbol{w}}_n,\boldsymbol{\gamma}_{n+1}^{(i)}\right)\right| \\ &\leq \mathbb{E}\{\|\tilde{\boldsymbol{w}}_n\|_{\boldsymbol{F}\boldsymbol{\sigma}}^2\}+(1+p)\cdot[\text{vec}\{\boldsymbol{H}\}]^\top\boldsymbol{\sigma}. \end{aligned} \tag{128}$$

Iterating (128) from $n=0$, we obtain:

$$\begin{aligned}\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\sigma}}^2\} \leq\ & \mathbb{E}\{\|\tilde{\boldsymbol{w}}_0\|_{\boldsymbol{F}^{n+1}\boldsymbol{\sigma}}^2\} \\ & +(1+p)[\text{vec}\{\boldsymbol{H}\}]^\top\sum_{j=0}^{n}\boldsymbol{F}^j\boldsymbol{\sigma} \end{aligned} \tag{129}$$

where $\mathbb{E}\{\|\tilde{\boldsymbol{w}}_0\|^2\}$ is the initial condition. The stability of $\mathbb{E}\{\|\tilde{\boldsymbol{w}}_{n+1}\|_{\boldsymbol{\sigma}}^2\}$ requires the convergence of terms on the RHS of (129), which is ensured by condition $\rho(\boldsymbol{F})<1$. Since $\boldsymbol{F}=\boldsymbol{B}^\top\otimes\boldsymbol{B}^\top$, it is enough to select sufficiently small step-sizes μ_k satisfying (85) to ensure $\rho(\boldsymbol{F})<1$.

APPENDIX F
PARAMETERS USED BY ALGORITHMS IN SECTION VI-B1

Table III provides all parameters used by the algorithms in Section VI-B1. Notations used for these parameters are the same as those in the original references. Some pairs of columns, standing for different parameters, are merged into a single column for compactness. The corresponding symbols can be distinguished by the symbol "|". We used a uniform step-size 0.01 for all algorithms.

TABLE III
PARAMETERS USED BY ALL ALGORITHMS FOR SIMULATION.

Algorithms	λ_k | γ	ε	η	τ_1 | τ_2
proposed $\ell_{\infty,1}$ proximal LMS	0.08			0.1
proposed RW$\ell_{\infty,1}$ proximal LMS	0.01	0.01		| 0.05
ZA-Sparse LMS [6]	| 0.03			
RZA-Sparse LMS [6]	| 0.03	0.45		
$\ell_{2,0}$ jointly sparse [24]		8	0.1	
$\ell_{2,1}$ jointly sparse [24]			0.03	
RW$\ell_{2,1}$ jointly sparse [24]		100	1.5	

REFERENCES

[1] S.-Y. Tu and A. H. Sayed, "Diffusion strategies outperform consensus strategies for distributed estimation over adaptive netowrks," *IEEE Trans. Signal Process.*, vol. 60, no. 12, pp. 6217–6234, Dec. 2012.

[2] C. G. Lopes and A. H. Sayed, "Diffusion least-mean squares over adaptive networks: Formulation and performance analysis," *IEEE Trans. Signal Process.*, vol. 56, no. 7, pp. 3122–3136, Jul. 2008.

[3] F. Cattivelli, C. G. Lopes, and A. H. Sayed, "Diffusion recursive least-squares for distributed estimation over adaptive networks," *IEEE Trans. Signal Process.*, vol. 56, no. 5, pp. 1865–1877, May 2008.

[4] L. Li and J. Chambers, "Distributed adaptive estimation based on the APA algorithm over diffusion networks with changing topology," in *Proc. IEEE SSP*, 2009, pp. 757–760.

[5] Y. Liu, C. Li, and Z. Zhang, "Diffusion sparse least-mean squares over networks," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 4480–4485, Aug. 2012.

[6] P. Di Lorenzo and A. H. Sayed, "Sparse distributed learning based on diffusion adaptation," *IEEE Trans. Signal Process.*, vol. 61, no. 6, pp. 1419–1433, Mar. 2013.

[7] A. H. Sayed, "Diffusion adaptation over networks," in *Academic Press Libraray in Signal Processing*, R. Chellapa and S. Theodoridis, Eds., vol. 3, pp. 322–454. Elsevier, 2014.

[8] A. H. Sayed, "Adaptive networks," *Proc. of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[9] A. H. Sayed, *Adaptation, Learning, and Optimization over Networks*, vol. 7, Now Publishers Inc., Hanover, MA, USA, Jul. 2014.

[10] X. Mao, Y. Gu, and W. Yin, "Walk proximal gradient: An energy-efficient algorithm for consensus optimization," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 2048–2060, 2018.

[11] X. Mao, K. Yuan, Y. Hu, A. H. Sayed, and W. Yin, "Walkman: A communication-efficient random-walk algorithm for decentralized optimization," *IEEE Trans. Signal Process.*, vol. 68, pp. 2513 – 2528, 2020.

[12] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks," *IEEE Trans. Signal Process.*, vol. 62, no. 16, pp. 4129–4144, Aug. 2014.

[13] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion LMS with sparsity-based regularization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2015, pp. 3516–3520.

[14] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Trans. Signal Process.*, vol. 64, no. 23, pp. 6329–6344, Dec. 2016.

[15] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks with common latent representations," *IEEE J. Sel. Topics Signal Process.*, vol. 11, no. 3, pp. 563–579, 2017.

[16] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Diffusion LMS for multitask problems with local linear equality constraints," *IEEE Trans. Signal Process.*, vol. 65, no. 19, pp. 4979–4993, Oct. 2017.

[17] J. Chen, C. Richard, and A. H. Sayed, "Diffusion LMS over multitask networks," *IEEE Trans. Signal Process.*, vol. 63, no. 11, pp. 2733–2748, Jun. 2015.

[18] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Multitask diffusion adaptation over asynchronous networks," *IEEE Trans. Signal Process.*, vol. 64, no. 11, pp. 2835–2850, 2016.

[19] M. J. Piggott and V. Solo, "Stability of adaptive network algorithms in multitask environments," in *Proc. IEEE CDC*, 2017, pp. 1472–1477.

[20] D. Jin, J. Chen, C. Richard, J. Chen, and A. H. Sayed, "Affine combination of diffusion stretegies over networks," *IEEE Trans. Signal Process.*, vol. 68, pp. 2087–2104, Dec. 2020.

[21] V. C. Gogineni and M. Chakraborty, "Diffusion affine projection algorithm for multitask networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc.*, 2018, pp. 201–206.

[22] V. C. Gogineni and M. Chakraborty, "Improving the performance of multitask diffusion APA via controlled inter-cluster cooperation," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 67, no. 3, pp. 903–912, 2020.

[23] V. C. Gogineni and M. Chakraborty, "Partial diffusion affine projection algorithm over clustered multitask networks," in *Proc. IEEE ISCAS*, 2019, pp. 1–5.

[24] C. Li, S. Huang, Y. Liu, and Z. Zhang, "Distributed jointly sparse multitask learning over networks," *IEEE Trans. Cybern.*, vol. 48, no. 1, pp. 151–164, Jan. 2018.

[25] J. A. Bazerque and G. B. Giannakis, "Distributed spectrum sensing for cognitive radio networks by exploiting sparsity," *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1847–1862, 2010.

[26] M. F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R. G. Baraniuk, "Distributed compressed sensing of jointly sparse signals," in *Proc. Asilomar Conf. Signals Syst. Comput.*, Oct./Nov. 2005, pp. 1537–1541.

[27] M. Kocic, D. Brady, and M. Stojanovic, "Sparse equalization for realtime digital underwater acoustic communications," in *Proc. OCEANS*, San Diego, CA, USA, 1995, pp. 1417–1422.

[28] M. Masood, L. H. Afify, and T. Y. Al-Naffouri, "Efficient coordinated recovery of sparse channels in massive MIMO," *IEEE Trans. Signal Process.*, vol. 63, no. 1, pp. 104–118, 2015.

[29] Y. Gu and M. Wang, "Learning distributed jointly sparse systems by collaborative LMS," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2014, pp. 7228–7232.

[30] S. Banert and R. I. Bot, "A general double-proximal gradient algorithm for d.c. programming," *Math. Program.*, vol. 178, pp. 301–326, Nov. 2019.

[31] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[32] D. P. Bertsekas, "Incremental proximal methods for large scale convex optimization," *Math. Program.*, vol. 129, no. 2, pp. 163, Jun. 2011.

[33] W. M. Wee and I. Yamada, "A proximal splitting approach to regularized distributed adaptive estimation in diffusion networks," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, May 2013, pp. 5420–5424.

[34] S. Vlaski and A. H. Sayed, "Proximal diffusion for stochastic costs with non-differentiable regularizers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2015, pp. 3352–3356.

[35] S. Vlaski, L. Vandenberghe, and A. H. Sayed, "Diffusion stochastic optimization with non-smooth regularizers," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2016, pp. 4149–4153.

[36] J. Huang and T. Zhang, "The benefit of group sparsity," *The Annals of Statistics*, vol. 38, no. 4, pp. 1978–2004, Aug. 2010.

[37] L. Meier, S. Van De Geer, and P. Buhlmann, "The group lasso for logistic regression," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 1, pp. 53–71, 2008.

[38] Y. Chen and A. O. Hero, "Recursive $\ell_{1,\infty}$ group lasso," *IEEE Trans. Signal Process.*, vol. 60, no. 8, pp. 3978–3987, 2012.

[39] Y. Chen, Y. Gu, and A. O. Hero, "Regularized least-mean-square algorithms," *Arxiv preprint arXiv:1012.5066*, 2010.

[40] S. N. Negahban and M. J. Wainwright, "Simultaneous support recovery in high dimensions: Benefits and perils of block ℓ_1/ℓ_∞-regularization," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3841–3863, Jun. 2011.

[41] E. Esser, M. Moller, S. Osher, G. Sapiro, and J. Xin, "A convex model for nonnegative matrix factorization and dimensionality reduction on physical space," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3239–3252, 2012.

[42] E. M. Eksioglu, "Group sparse RLS algorithms," *International journal of adaptive control and signal processing*, vol. 28, no. 12, pp. 1398–1412, Dec. 2013.

[43] J. A. Tropp, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Process.*, vol. 86, no. 3, pp. 589 – 602, 2006.

[44] David R. Hunter, Kenneth Lange, Departments Of Biomathematics, and Human Genetics, "A tutorial on MM algorithms," *Amer. Statist*, pp. 30–37, 2004.

[45] E. J. Candès, M. B. Wakin, and S. P. Boyd, "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 877–905, 2008.

[46] A. H. Sayed, *Adaptive Filters*, John Wiley & Sons, Inc., 2008.

[47] D. Jin, J. Chen, C. Richard, and J. Chen, "Model-driven online parameter adjustment for zero-attracting LMS," *Signal Process.*, vol. 152, pp. 373 – 383, Nov. 2018.

[48] J. Chen, C. Richard, J. C. M. Bermudez, and P. Honeine, "Nonnegative least-mean-square algorithm," *IEEE Trans. Signal Process.*, vol. 59, no. 11, pp. 5225–5235, Nov. 2011.

[49] J. Arenas-Garcia, A. R. Figueiras-Vidal, and A. H. Sayed, "Mean-square performance of a convex combination of two adaptive filters," *IEEE Trans. Signal Process.*, vol. 54, no. 3, pp. 1078–1090, Mar. 2006.

[50] V. H. Nascimento, M. T. M. Silva, R. Candido, and J. Arenas-Garcia, "A transient analysis for the convex combination of adaptive filters," in *Proc. IEEE SSP*, Aug. 2009, pp. 53–56.

[51] M. T. M. Silva, V. H. Nascimento, and J. Arenas-Garcia, "A transient analysis for the convex combination of two adaptive filters with transfer of coefficients," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Mar. 2010, pp. 3842–3845.

[52] M. Fiedler, "Algebraic connectivity of graphs," *Czechoslovak Mathematical Journal*, vol. 23, no. 2, pp. 298–305, 1973.

[53] A. Simões and J. Xavier, "FADE: Fast and asymptotically efficient distributed estimator for dynamic networks," *IEEE Trans. Signal Process.*, vol. 67, no. 8, pp. 2080–2092, Apr. 2019.

[54] H. Zhu, G. B. Giannakis, and A. Cano, "Distributed in-network channel decoding," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3970–3983, 2009.

[55] H. Zhu, A. Cano, and G. B. Giannakis, "Distributed consensus-based demodulation: algorithms and error analysis," *IEEE Trans. Wireless Commun.*, vol. 9, no. 6, pp. 2044–2054, 2010.

[56] M. B. Porter, “The bellhop manual and user’s guide: Preliminary draft,” tech. rep., Heat, Light, and Sound Research, Inc., LaJolla, CA, USA, 2011 [Online] Available: http://oalib.hlsresearch.com/Rays/HLS-2010.pdf.

View publication stats