<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper published in *IEEE Transactions on Signal Processing*. This paper has been peer-reviewed but does not include the final publisher proof-corrections or journal pagination.

Citation for the original published paper (version of record):

Ye, Y., Xiao, M., Skoglund, M. (2021)
Randomized Neural Networks Based Decentralized Multi-Task Learning via Hybrid
Multi-Block ADMM
IEEE Transactions on Signal Processing, 69: 2844-2857
<https://doi.org/10.1109/TSP.2021.3078625>

Access to the published version may require subscription.

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-298637>

Randomized Neural Networks based Decentralized Multi-Task Learning via Hybrid Multi-Block ADMM

Yu Ye, *Student Member, IEEE*, Ming Xiao, *Senior Member, IEEE*, and Mikael Skoglund, *Fellow, IEEE*

Abstract—In multi-task learning (MTL), related tasks learn jointly to improve generalization performance. To exploit the high learning speed of feed-forward neural networks (FNN), we apply the randomized single-hidden layer FNN (RSF) to the MTL problem, where the output weights of RSFs for all the tasks are learned collaboratively. We first present the RSF based MTL problem in the centralized setting, which is solved by the proposed MTL-RSF algorithm. Due to the fact that many data sets of different tasks are geo-distributed, decentralized machine learning is studied. We formulate the decentralized MTL problem based on RSF as majorized multi-block optimization with coupled bi-convex objective functions. To solve the problem, we propose the DMTL-RSF algorithm, which is a hybrid Jacobian and Gauss-Seidel Proximal multi-block alternating direction method of multipliers (ADMM). Further, to reduce the computation load of DMTL-RSF, DMTL-RSF with first-order approximation (FO-DMTL-RSF) is presented. Theoretical analysis shows that the convergence to the stationary point of proposed decentralized algorithms can be guaranteed conditionally. Through simulations, we demonstrate the convergence of presented algorithms, and also show that they can outperform existing MTL methods. Moreover, by adjusting the dimension of hidden feature space, there exists a trade-off between communication load and learning accuracy for DMTL-RSF.

Index Terms—Multi-task learning; randomized feed-forward neural networks; decentralized optimization

I. INTRODUCTION

Machine learning usually requires a large amount of training samples to obtain an accurate learner, e.g., the deep neural network with a tremendous number of model parameters. However, in many real-world applications, it is hard to collect enough samples for training. One efficient solution when in shortage of data is Multi-Task Learning (MTL) [1], of which the main goal is to improve generalization performance by leveraging the domain-specific information contained in the training samples of tasks. In MTL, the learning tasks are assumed to be related to each other, and it is found that learning them jointly can achieve better generalization performance rather than learning individually.

The MTL approach seeks to learn the relationship of model parameters among tasks, and it has been extensively investigated [2]–[13]. In [2], it is assumed that the model parameters of task t can be represented as $\mathbf{w}_t = \mathbf{w}_0 + \mathbf{v}_t$, where $\mathbf{w}_0$ is common across tasks and $\mathbf{v}_t$ is task specific deviation. The clustered multi-task learning (CMTL) is studied in [3],

which learned the clustered structure of tasks by identifying representative ones. While in [4]–[9], the multi-task feature learning (MTFL) is considered, based on the assumption that the model parameters of task t can be given by $\mathbf{w}_t = \mathbf{U}\mathbf{A}_t$. The feature space $\mathbf{U}$ is common for all the tasks while $\mathbf{A}_t$ is task specified. In [4], [5] and [6], ℓ_1 -norm penalty and ℓ_2 -norm penalty are considered in the objective functions, respectively. In [7], the tasks are assumed to be in disjoint groups. While in [8], the tasks in different groups are allowed to overlap with each other in bases. In [12], the model parameters for task t is assumed to be $\mathbf{w}_t = \mathbf{U}_t + \Theta\mathbf{v}_t$, where Θ captures the correlation among tasks. Combining the assumptions in [2] and [4], the shared features and shared parameters are considered simultaneously through assuming $\mathbf{w}_t = \mathbf{U}(\mathbf{A}_0 + \mathbf{A}_t)$. In this way both feature relatedness and model relatedness can be modeled.

In [2]–[12], the data from all the tasks are in a central location. However in many real-world applications, the data of different tasks may be separately located in different machines. Thus the centralized methods may be inefficient. MTL on distributed networks is an active research area and has attracted lots of research interests. Recently, various distributed MTL problems have been studied [14]–[18]. Utilizing the equivalent convex optimization formulation in [5], which characterizes the correlation between model parameters $\mathbf{w}_t$ by a matrix $\mathbf{\Omega}$, the distributed multi-task relationship learning is studied in [19]–[21]. In [22], a communication-efficient estimator based on the debiased lasso is presented. Reference [23] learned a shared predictive structure for tasks by extending [12] to a distributed setting. Following the assumptions of MTFL in [4], two communication-efficient subspace pursuit algorithms are provided in [24]. However, the strategies in [24] can only be implemented in master-slave network structures.

For the centralized MTL problems studied in [4], [5], [7], [8], [10], *alternating optimization* (AO) method is utilized to obtain the solutions. In each iteration of the AO method, one variable is optimized while another is fixed [25]. Yet in distributed MTL (DMTL), objectives and variables may be separated and the AO method is invalid. Hence one of the challenges for DMTL is to design distributed optimization algorithms. Confronted with this problem, the alternating direction method of multipliers (ADMM) [26]–[30] has been shown as an efficient solution, which was used to solve a global consensus problem with regularization in [31]. Extending the *global consensus problem* [26] to multi-block convex optimization with separable objectives and linear constraints,

Yu Ye, Ming Xiao and Mikael Skoglund are with the School of Electrical Engineering and Computer Science, Royal Institute of Technology (KTH), Stockholm, Sweden (email: yu9@kth.se, mingx@kth.se, skoglund@kth.se).

a Jacobian Proximal ADMM is proposed in [32], where the variables are updated in a parallel manner. The ADMM based algorithms for multi-block convex optimizations with coupled objectives are provided in [33]–[38]. A Gauss-Seidel type ADMM with first-order approximation is provided to solve the general multi-block optimization with coupled objective function in [34]. Different from the Jacobian type method, the variables are sequentially optimized for Gauss-Seidel method. In [23], the optimization method that integrates block coordinate descent method (BCD) with the inexact ADMM is utilized for distributed learning. While a BCD for regularized multi-convex optimization is considered in [39]. Furthermore, the ADMM in nonconvex and nonsmooth optimization is studied in [40].

Since the tasks are trained together both for centralized and distributed MTL, the model will become complicated and hence significantly reduce the training speed, especially for neural networks (NN) with multiple layers. To balance the training speed and the generalization performance of MTL and DMTL, the randomized single-hidden layer feed-forward neural networks (RSF)¹ can be utilized for the basic tasks. Because there is only one hidden layer in RSF, and the hidden nodes can be chosen randomly, the output weights of RSF can be analytically determined. According to the analysis and experiments in [41], the RSF can provide good generalization performance in most learning cases and learn thousands of times faster than conventional popular learning algorithms for feed-forward neural networks. The distributed RSF is studied in [31], [42], where one single task is distributed to several workers in parallel.

Based on above observation, we will study the MTL approach with RSF implementation by considering both centralized and decentralized scenarios. Different from [2]–[12], [14]–[17], [19]–[24], the input data for tasks is first randomly mapped to a hidden feature space. In our MTL method [1], the weights of RSF for related tasks are correlated. Hence instead of learning the weights of RSF individually, the tasks can leverage training data sets from each other to obtain model parameters with better generalization performance. This can be achieved for the centralized case since all the data sets are available in a single node. However, when tasks are located separately, transmitting training datasets across tasks is costly, as well as may cause the privacy leakage problem. Thus we utilize the decentralized optimization method to update each local weights through exchanging intermediate information. To the best of our knowledge, DMTL with RSF implementation has not been studied before. The main contributions of this paper are listed as follows.

- We study the centralized multi-task learning machine based on RSF (MTL-RSF). Based on *Alternating Optimization* (AO) method, the MTL-RSF algorithm is proposed to obtain shared subspace across tasks and the task-specified weights. We show that the proposed MTL-RSF algorithm can be guaranteed to converge to a stationary point of the problem. Then we extend the centralized learning problem to a decentralized setting by introducing

sharing variables for localized task objectives as well as a consensus constraint. The algorithm DMTL-RSF of hybrid Jacobian and Gauss-Seidel Proximal multi-block alternating direction method of multipliers (ADMM) is proposed to solve the decentralized learning problem;

- We generalize the decentralized learning problem as majorized multi-block optimization problem with coupled bi-convex objective functions, which was not been studied before, to our best knowledge. Through theoretical analysis, we prove that the DMTL-RSF converges to a stationary point when algorithm parameters meet specific conditions.
- To reduce the computation load of DMTL-RSF, we propose the FO-DMTL-RSF algorithm, which utilizes the first-order approximation in the update process. The convergence of FO-DMTL-RSF to stationary points are proved.
- By simulations, we show the convergence of algorithms MTL-RSF, DMTL-RSF and FO-DMTL-RSF. By experiments on real-world datasets, we also show that MTL-RSF, DMTL-RSF and FO-DMTL-RSF can outperform state-of-the-art MTL approaches in terms of the generalization performance. Moreover, the trade-off between communication load and learning accuracy for DMTL-RSF and FO-DMTL-RSF are shown through experiments.

The rest of this paper is organized as follows. In Section I, we formulate the RSF based MTL problem in a centralized setting, which is solved by the proposed MTL-RSF algorithm. Then the DMTL problem with RSF implementation is presented in Section III. Two fully decentralized ADMM approach DMTL-RSF and FO-DTML-RSF are provided, the convergence analysis of which is presented in Section IV. Numerical results are given in Section V to show the generalization performance of the proposed approaches. Finally, we draw conclusions in Section VI.

Notation: $\text{tr}(\mathbf{A})$, $\|\mathbf{A}\|$ and $\mathbf{A}^\top$ denote trace, Frobenius norm and transpose of matrix $\mathbf{A}$, respectively. $\otimes$ presents Kronecker product. $\langle \mathbf{A}, \mathbf{B} \rangle$ denotes Frobenius inner product of matrices $\mathbf{A}$ and $\mathbf{B}$. $\text{vec}(\mathbf{A})$ presents the vectorization of matrix $\mathbf{A}$.

II. MULTI-TASK LEARNING BASED ON RSF

In this section, we will first present the principles of RSF. Then we shall integrate the RSF with centralized MTL by modifying the models of RSF. Though the stochastic configuration networks (SCN) in [43] also employs randomized learning technique and it outperforms RSF, we use RSF to develop our models due to its straightforward implementation.

A. Principles of RSF

RSF refers to a class of single-hidden-layer feed-forward neural networks (FNNs) shown in Fig. 1 (a), where the hidden layer needs not be tuned. The output of an RSF network with L hidden nodes is formulated as

$$\mathbf{Y}_L(\mathbf{X})^\top = \sum_{l=1}^L \beta_l^\top h_l(\mathbf{X}) = \mathbf{h}(\mathbf{X})\boldsymbol{\beta}, \quad (1)$$

¹It is also referred as extreme learning machine (ELM).

Fig. 1. Two different model structures for task t : (a) typical RSF; (b) multi-task learning based on RSF with a common shared subspace U and task specified weight A_t .

where $\beta = [\beta_1, \dots, \beta_L]^\top \in \mathbb{R}^{L \times d}$ is the output weights. $\mathbf{h}(X) = [h_1(X), \dots, h_L(X)]$ is the feature mapping $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^{1 \times L}$, which maps input variable $X \in \mathbb{R}^n$ to L -dimension hidden-layer feature space. Note that $\mathbf{h}(X)$ is a row vector. The component $h_l(X) = G(\mathbf{w}_l, b_l, X) : \mathbb{R}^n \rightarrow \mathbb{R}$ denotes the impulse function of the l -th hidden node given by activation function $G(\mathbf{w}_l, b_l, X) = g(\mathbf{w}_l^\top X + b_l)$. The similar idea of randomization over hidden neurons is also analyzed in the random vector version of the functional-link (RVFL) net [44]. The output weights β of RSF need to be learned from a training data set $\mathcal{D} = \{(X_i, T_i) | X \in \mathbb{R}^n, T_i \in \mathbb{R}^d, i = 1, \dots, N\}$ by solving

$$\beta^* := \arg \min_{\beta} \frac{1}{2} \|H\beta - T\|^2 + \frac{\mu}{2} \|\beta\|^2, \quad (2)$$

where $T = [T_1, \dots, T_N]^\top \in \mathbb{R}^{N \times d}$, $\frac{\mu}{2} \|\cdot\|^2$ is the regularization term [41], and $H = [\mathbf{h}(X_1)^\top, \dots, \mathbf{h}(X_N)^\top]^\top \in \mathbb{R}^{N \times L}$ represents the hidden layer output matrix. RSF can be implemented on a single machine since the parameters can be obtained explicitly.

B. Centralized MTL with RSF

In what follows, we apply the RSF method to MTL, where each task implements its own RSF scheme. Through learning a shared feature space of the hidden layer among tasks to transfer the knowledge of training data, we present how each task can improve the generalization performance. Suppose that there are m tasks with each data set as $\mathcal{D}_t = \{(X_{t,i}, T_{t,i}) | i = 1, \dots, N_t, t \in \mathcal{V}\}$, where N_t is the number of training samples and $\mathcal{V} = \{1, \dots, m\}$ is the set of tasks. Therefore the corresponding hidden layer feature space for task t is expressed as $H_t = [\mathbf{h}(X_{t,1})^\top, \dots, \mathbf{h}(X_{t,N_t})^\top]^\top \in \mathbb{R}^{N_t \times L}$. The output of RSF for task t is $f_t = H_t \beta_t$, where $\beta_t = [\beta_{t,1}, \dots, \beta_{t,L}] \in \mathbb{R}^{L \times d}$. Following [8], we assume that there are r latent basis tasks in MTL and each observed task can be represented as a subset of these basis tasks. Thus there exists a subspace of the hidden layer feature space, with which all the tasks can share related representation. Hence from [8], the output weights can be evaluated by $\beta_t = U A_t$, where $U \in \mathbb{R}^{L \times r}$ presents the predictive structure of the tasks while $A_t \in \mathbb{R}^{r \times d}$ determines the grouping structure. But different from [8], in the multi-task feature learning based on RSF (MTL-RSF), the input data X is

Algorithm 1: MTL-RSF

```

1: Input: Hidden layer feature space  $\{H_t | t \in \mathcal{V}\}$ 
2: Output:  $U$  and  $\{A_t | t \in \mathcal{V}\}$ 
3: initialization:  $\{A_t^0 = \mathbf{1} | t \in \mathcal{V}\}$ 
4: for  $k = 0, 1, \dots$  do
5:   update  $U^{k+1}$  according to (4);
6:   for  $t = 1$  to  $m$  do
7:     update  $A_t^{k+1}$  according to (6);
8:   end for
9: end for

```

---

first randomly mapped into a hidden feature space¹. Then we can conclude the structure of MTL-RSF as Fig. 1 (b), where a common *Linear Transform* (LT) layer is stacked between the hidden layer and output layer for each task. U is the weight for LT layer and shared among tasks. A_t is the output weight and is task specific. To obtain the optimal feature space U and $\{A_t\}$, we will solve the following problem

$$(P-1) : \min_{U, A} \sum_{t \in \mathcal{V}} \frac{1}{2} \|H_t U A_t - T_t\|^2 + \frac{\mu_1}{2} \|U\|^2 + \frac{\mu_2}{2} \|A\|^2,$$

where $A = [A_1^\top, \dots, A_m^\top]^\top$ and $T_t = [T_{t,1}, \dots, T_{t,N_t}]^\top \in \mathbb{R}^{N_t \times d}$. μ_1 and μ_2 are the trade-off parameters between the training error and regularization. It is assumed that each RSF utilizes the same number of hidden nodes L . The random weights and biases of hidden neurons are the same as $\{\mathbf{w}_l, b_l | l = 1, \dots, L\}$.

The cost function in (P-1) is bi-convex in U and A . Hence we adopt *alternating optimization* (AO) method to achieve a local minimum. When A^k is fixed, for the iteration $k+1$, the problem reduces to

$$U^{k+1} := \arg \min_U \sum_{t \in \mathcal{V}} \frac{1}{2} \|H_t U A_t^k - T_t\|^2 + \frac{\mu_1}{2} \|U\|^2. \quad (3)$$

(3) is convex in U and a closed-form solution can be obtained. Setting the derivative to zero, and according to the property of Kronecker product $\text{vec}(AXB) = (B^\top \otimes A) \text{vec}(X)$, it has

$$\begin{aligned} \text{vec}(U^{k+1}) = & \left(\sum_{t \in \mathcal{V}} (A_t^k (A_t^k)^\top) \otimes (H_t^\top H_t) + \mu_1 I \right)^{-1} \\ & \times \sum_{t \in \mathcal{V}} \text{vec}(H_t^\top T_t (A_t^k)^\top). \end{aligned} \quad (4)$$

When U^{k+1} is fixed, the problem reduces to solving (5) for each task separately,

$$A_t^{k+1} := \arg \min_{A_t} \frac{1}{2} \|H_t U^{k+1} A_t - T_t\|^2 + \frac{\mu_2}{2} \|A_t\|^2. \quad (5)$$

(5) is convex over A_t and the solution can be easily obtained by

$$A_t^{k+1} = \left((U^{k+1})^\top H_t^\top H_t U^{k+1} + \mu_2 I \right)^{-1} (U^{k+1})^\top H_t^\top T_t. \quad (6)$$

¹The proposed algorithms and theory regarding convergence can be applied to the general multi-task learning problem, which seeks to learn T regression functions f_t of the form $f_t(x) = \langle A_t, U^\top \phi(x) \rangle$ with any feature mapping $\phi(x) : \mathbb{R}^n \rightarrow \mathbb{R}^L$.

Then we propose the multi-task learning based on RSF algorithm (MTL-RSF) to solve (P-1). In MTL-RSF, each task initializes $A_t^0 = \mathbf{1}$. Then the iterations are executed, and the shared representation U and $\{A_t\}$ are successively optimized in each iteration. Based on the analysis in [25], we provide the convergence of MTL-RSF as follow.

Lemma 1. *With $k \rightarrow \infty$, the sequence $\{U^k, A^k\}$ generated by MTL-RSF converges to a stationary point (U^*, A^*) of problem (P-1).*

Proof: See the Appendix A. ■

Furthermore, from [25], it can be shown that the objective of (P-1) is non-increasing with iterations in MTL-RSF. Since the objective of (P-1) is lower bounded, we conclude that the stationary point (U^*, A^*) can be a local minimum.

III. DECENTRALIZED MULTI-TASK LEARNING WITH RSF

A. Motivation and basics

In many real-world applications, the data of different tasks may be geo-distributed over different machines. However, the MTL-RSF algorithm proposed in different locations can only work on a single machine, which has the whole data sets of all tasks. Due to the heavy communication load caused by transmitting the data or the constraint of data privacy and security, it may be impossible to send data of different tasks to a master machine to perform the MTL-RSF algorithm. Therefore we consider the decentralized multi-task feature learning based on RSF (DMTL-RSF), where the shared representation U for hidden layer feature is obtained through information exchange between tasks. Meanwhile, different from the distributed RSF (DRSF) in [31] where one identical output is desired, we learn the distinctive predictive model for each task based on its local data distribution.

In DMTL-RSF, we consider a setup with m tasks distributed over m agents, i.e., one agent for one task. We denote the multi-agent network as an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the agents and $\mathcal{E}$ includes the links.

Assumption 1. *The undirected graph $\mathcal{G}$ is connected.*

Assumption 1 implies that any two agents in the network can always influence each other in the long run. Each agent implements its own RSF scheme locally with the same number of hidden neurons as L and the identical random weights and biases as $\{w_l, b_l | l = 1, \dots, L\}$. The goal of DMTL-RSF is to find the sharing representation U cooperatively across agents, as well as the predictive model $\{A_t | t \in \mathcal{V}\}$ for all the agents by minimizing the overall expected loss.

Remark 1. *One naive approach to solving (P-1) in a decentralized way is to exchange the information $\{(A_t^k (A_t^k)^T) \otimes (H_t^T H_t), H_t^T T_t (A_t^k)^T | t \in \mathcal{V}\}$ among all the agents. However, when dimensions L and r are large, the amount of exchanged information becomes huge. This significantly increases communication load. Another limitation is that this strategy can only work efficiently in the master-slave structure.*

Hence we will solve problem (P-1) through adopting decentralized optimization method instead. In the following, we

will consider the decentralized network structure and propose the decentralized algorithm.

B. Problem Formulation

In order to find optimal U , information on local weights needs to be exchanged among agents. Thus, we exploit the alternating direction method of multipliers (ADMM) to ensure all machines agree with the same subspace by solving the following problem,

$$(P-2) : \min_{U, A} \sum_{t \in \mathcal{V}} \left(\frac{1}{2} \|H_t U_t A_t - T_t\|^2 + \frac{\mu_1}{2m} \|U_t\|^2 + \frac{\mu_2}{2} \|A_t\|^2 \right),$$

$$s.t. \sum_{t \in \mathcal{V}} C_t U_t = \mathbf{0}.$$

Compared with the centralized scenario, agent t holds a local subspace variable U_t . To ensure a unique subspace, the constraint of $\{U_t\}$ is introduced. For any $(i, j) \in \mathcal{E} (i, j \in \mathcal{V})$, the subspaces of $U_i = U_j$ is required. Let $i \in \{1, \dots, |\mathcal{E}|\}$ and $C = [C_1, \dots, C_m]$ be composed of $|\mathcal{E}| \times m$ blocks of $L \times L$ matrices, where $C_t \in \mathbb{R}^{|\mathcal{E}| \times L \times L}$ contains $|\mathcal{E}|$ blocks. If $(t_1, t_2) = \mathcal{E}_i, \forall t_1, t_2 \in \mathcal{V}$, the (i, t_1) -th block and the (i, t_2) -th block of C are respectively $I^{L \times L}$ and $-I^{L \times L}$. Other blocks in C are zero matrices $\mathbf{0}^{L \times L}$.

According to ADMM [21], the augmented Lagrangian function for problem (P-2) is given by

$$\mathcal{L}(U, A, \lambda) = \sum_{t \in \mathcal{V}} \left(\frac{1}{2} \|H_t U_t A_t - T_t\|^2 + \frac{\mu_1}{2m} \|U_t\|^2 + \frac{\mu_2}{2} \|A_t\|^2 \right) + \langle \lambda, \sum_{t \in \mathcal{V}} C_t U_t \rangle + \frac{\rho}{2} \left\| \sum_{t \in \mathcal{V}} C_t U_t \right\|^2, \quad (7)$$

where $\lambda = [\lambda_1^T, \dots, \lambda_{|\mathcal{E}|}^T]^T \in \mathbb{R}^{|\mathcal{E}| \times L \times r}$ is a Lagrange multiplier and $U = [U_1^T, \dots, U_m^T]^T, \rho > 0$. Since Lagrangian $\mathcal{L}$ is separable in U_t and A_t , we can optimize them in parallel across the agents. However, since U_t and A_t are coupled variables, we can follow AO to update them sequentially. Hence from the iteration of proximal Jacobi ADMM [32], we update the variables by

$$U_t^{k+1} := \arg \min_{U_t} \mathcal{L}(U_t, U_{-t}^k, A^k, \lambda^k) + \frac{1}{2} \|U_t - U_t^k\|_{P_t}^2; \quad (8a)$$

$$\lambda^{k+1} := \lambda^k - \rho \gamma^{k+1} \sum_{t \in \mathcal{V}} C_t U_t^{k+1}; \quad (8b)$$

$$A_t^{k+1} := \arg \min_{A_t} \mathcal{L}(U^{k+1}, A_t, A_{-t}^k, \lambda^k) + \frac{1}{2} \|A_t - A_t^k\|_{Q_t}^2, \quad (8c)$$

where $U_{-t} \triangleq \{U_1, \dots, U_{t-1}, U_{t+1}, \dots, U_m\}$ and $A_{-t} \triangleq \{A_1, \dots, A_{t-1}, A_{t+1}, \dots, A_m\}$. Since $\{U_t\}$ and $\{A_t\}$ are matrices, for $z \in \mathbb{R}^{n \times m}$, we define the G -norm as $\|z\|_G^2 = \text{tr}(z^T G z)$. Without loss of generality, we assume $\{P_t\}$ and $\{Q_t\}$ are

Algorithm 2: DMTL-RSF

1: **Input:** $\{H_t, T_t, C_t, P_t, Q_t | t \in \mathcal{V}\}, \delta$
2: **Output:** $\{U_t, A_t | t \in \mathcal{V}\}$
3: initialization: $\{U_t^0 = \mathbf{1}, A_t^0 = \mathbf{1} | t \in \mathcal{V}\}, \lambda^0 = \mathbf{0}$;
4: **for** $k = 0, 1, \dots$ **do**
5:   agents $t = 1$ to m **do**:
6:   update U_t^{k+1} in parallel by (10), and share with the
   neighboring agents;
7:   pick $\gamma_i^{k+1} \in (0, \frac{\delta \|\hat{C}_i(U_t^k - U_t^{k+1})\|^2}{\|\hat{C}_i U_t^{k+1}\|^2}]$, $i \in \{1, \dots, |\mathcal{E}|\}$;
8:   update λ^{k+1} according to (8b);
9:   update A_t^{k+1} in parallel by (12);
10: **end for**

Fig. 2. The updates in the k -th iteration of DMTL-RSF: an example.

diagonal matrices of which the diagonals are positive. In the updated process of U_t^{k+1} , it is equivalent to solve

$$U_t^{k+1} := \arg \min_{U_t} \frac{1}{2} \|H_t U_t A_t^k - T_t\|^2 + \frac{\mu_1}{2m} \|U_t\|^2 + \langle \lambda^k, C_t U_t \rangle + \frac{\rho}{2} \left\| C_t U_t + \sum_{i \in \mathcal{V}, i \neq t} C_i U_i^k \right\|^2 + \frac{1}{2} \|U_t - U_t^k\|_{P_t}^2. \quad (9)$$

Setting the derivative of (9) to 0 and applying vectorization, we obtain

$$\begin{aligned} \text{vec}(U_t^{k+1}) &= \left((A_t^k (A_t^k)^\top) \otimes (H_t^\top H_t) + I \otimes \left(\frac{\mu_1}{m} I + \rho C_t^\top C_t + P_t \right) \right)^{-1} \\ &\quad \times \text{vec} \left(H_t^\top T_t (A_t^k)^\top - \rho C_t^\top \sum_{i \in \mathcal{V}, i \neq t} C_i U_i^k - C_t^\top \lambda^k + P_t U_t^k \right). \end{aligned} \quad (10)$$

While updating A_t^{k+1} as (10), it is equivalent to solve

$$A_t^{k+1} := \arg \min_{A_t} \frac{1}{2} \|H_t U_t^{k+1} A_t - T_t\|^2 + \frac{\mu_2}{2} \|A_t\|^2 + \frac{1}{2} \|A_t - A_t^k\|_{Q_t}^2. \quad (11)$$

Setting the derivative to 0, we get the optimal solution of (20) as

$$A_t^{k+1} = \left((U_t^{k+1})^\top H_t^\top H_t U_t^{k+1} + Q_t + \mu_2 I \right)^{-1} \times \left((U_t^{k+1})^\top H_t^\top T_t + Q_t A_t^k \right). \quad (12)$$

With above analysis, we summarize the DMTL-RSF algorithm in Algorithm 2, where $\gamma = \text{blkdiag}(\gamma_1 I, \dots, \gamma_{|\mathcal{E}|} I) \in \mathbb{N}^{|\mathcal{E}|L \times |\mathcal{E}|L}$, $\mathbf{C} = [C_1, \dots, C_m]$ and $\mathbf{C} = [\hat{C}_1^\top, \dots, \hat{C}_{|\mathcal{E}|}^\top]^\top$.

Algorithm 3: First-order DMTL-RSF (FO-DMTL-RSF)

1: Follow steps 1-10 of Algorithm 2, but update U_t^{k+1} according to (14) in step 6.

$\hat{C}_i \in \mathbb{R}^{L \times mL}$ ($i \in \{1, \dots, |\mathcal{E}|\}$) is determined by the i -th link in graph $\mathcal{G}$. $\delta (> 0)$ is predetermined parameter and γ can balance the primal and dual residuals, which will be given by following analysis. In the k -th iteration of DMTL-RSF, U_t and A_t are updated successively at each agent. Moreover, the update process of $\{U_t\}$ and $\{A_t\}$ is carried out in parallel across agents. Hence the DMTL-RSF approach is a hybrid Jacobian and Gauss-Seidel Proximal multi-block ADMM algorithm. Since a unique subspace is expected for all agents, the update of U_t^{k+1} requires the information of U_t^k and $\{U_i^k | i \in \mathcal{V}_t\}$, where $\mathcal{V}_t (\subset \mathcal{V})$ includes the agents that connect with agent t . $\{A_t\}$, as the local predictive model, are specific among agents, and are updated privately without any sharing. One example for the updates of DMTL-RSF in the k -th iteration is shown as Fig. 2.

C. Computation efficient DMTL-RSF

Regarding the proposed algorithm DMTL-RSF, the update of U_t and A_t will consume substantial computing resource since the matrix inverse needs to be calculated in each iteration. This becomes especially severe when the dimension of hidden feature L and the number of basic tasks r are large. To reduce the burden on the computation of DMTL-RSF, we consider applying a first-order approximation for the updating process of U_t . Denoting $F_t(U_t, A_t) = \frac{1}{2} \|H_t U_t A_t - T_t\|^2 + \frac{\mu_1}{2m} \|U_t\|^2 + \frac{\mu_2}{2} \|A_t\|^2$, the function is approximated by $F_t(U_t, A_t^k) \approx F_t(U_t^k, A_t^k) + \langle \nabla_{U_t} F_t(U_t^k, A_t^k), U_t - U_t^k \rangle$ with fixed A_t^k . Then the update of U_t for agent t can be evaluated as

$$U_t^{k+1} := \arg \min_{U_t} \langle \nabla_{U_t} F_t(U_t^k, A_t^k), U_t - U_t^k \rangle + \langle \lambda^k, C_t U_t \rangle + \frac{\rho}{2} \left\| C_t U_t + \sum_{i \neq t} C_i U_i^k \right\|^2 + \frac{1}{2} \|U_t - U_t^k\|_{P_t}^2. \quad (13)$$

Hence the closed form representation of U_t^{k+1} is given by

$$U_t^{k+1} = (\rho C_t^\top C_t + P_t)^{-1} \left(-H_t^\top H_t U_t^k A_t^k (A_t^k)^\top + H_t^\top T_t (A_t^k)^\top - \frac{\mu_1}{m} U_t^k - \rho C_t^\top \sum_{i \in \mathcal{V}, i \neq t} C_i U_i^k - C_t^\top \lambda^k + P_t U_t^k \right). \quad (14)$$

Comparing (14) with (10), since the calculation of inversion becomes fixed, the complexity of computation is significantly reduced. Moreover, if we further introduce the approximation for the update of A_t , (7) can be reduced to the gradient descent method. By substituting (10) with (14) in algorithm 2, we can obtain the Algorithm 3, namely first-order DMTL-RSF (FO-DMTL-RSF).

IV. CONVERGENCE ANALYSIS

In what follows, we first discuss the difference between proposed decentralized algorithms and state-of-the-art methods. Then detailed convergence analysis will be provided.

A. Discussions

Denoting $f_t(U_t, A_t) = \frac{1}{2}\|H_t U_t A_t - T_t\|^2$, $g_1(\cdot) = \frac{\mu_1}{2m}\|\cdot\|^2$ and $g_2(\cdot) = \frac{\mu_2}{2}\|\cdot\|^2$, problem (P-2) can be generalized as

$$\min_{U, A} \sum_{t \in \mathcal{V}} (f_t(U_t, A_t) + g_1(U_t) + g_2(A_t)), \text{ s.t. } \mathbf{C}\mathbf{U} + \mathbf{D}\mathbf{A} = \mathbf{b}, \quad (15)$$

where $\mathbf{D} = \mathbf{0}$ and $\mathbf{b} = \mathbf{0}$. It is worth noting that $f_t(U_t, A_t)$ can only be convex when either U_t or A_t is fixed. Hence (15) is a majorized multi-block problem with coupled bi-convex objective functions. A more generalized presentation of (15) is the majorized model, which tries to solve

$$\min_{U, A} \phi(\mathbf{U}, \mathbf{A}) + \sum_{t \in \mathcal{V}} \varphi_t(U_t) + \sum_{t \in \mathcal{V}} \psi_t(A_t), \text{ s.t. } \mathbf{C}\mathbf{U} + \mathbf{D}\mathbf{A} = \mathbf{b}. \quad (16)$$

From the Multi-block ADMM algorithm in [34], problem (16) can be solved by the BCD method with Gauss-Seidel type, which optimizes the variable sequentially while fixing the remaining blocks at their last updated values. However, the sequential update behavior is non-efficient for solving the decentralized optimization problem. In [33], a hybrid Jacobian and Gauss-Seidel proximal block coordinate update (BCU) method is presented to solve a linearly constrained multi-block structured problem with a quadratic term in the objective. In [38] it shows that problem (16) can also be solved by a regularized version of the Jacobi algorithm. However, all the references mentioned above assume that the objective function is convex, which hence cannot be applied to solving (15). Though the methods for multi-convex and nonconvex optimization are provided in [39] and [40], they only focus on studying the algorithms with Gauss-Seidel type.

Since the fully Gauss-Seidel update usually performs better than the fully Jacobian update empirically [33], we integrate Jacobi-Proximal ADMM [32] with Gauss-Seidel update in Algorithm 2 to solve problem (15) in a hybrid way.

B. Convergence of (FO-)DMTL-RSF

Because $f_t(\cdot, \cdot)$ is bi-convex and $g_1(\cdot)$ and $g_2(\cdot)$ are strongly convex functions, following inequalities are useful for proving the convergence of DMTL-RSF and FO-DMTL-RSF, where $\langle \mathbf{A}, \mathbf{B} \rangle = \text{tr}(\mathbf{A}^\top \mathbf{B})$ ($\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$).

Proposition 1. For $\forall t \in \mathcal{V}$ and $U_t^1, U_t^2 \in \mathcal{U}$, $A_t^1, A_t^2 \in \mathcal{A}$, we have

$$\begin{aligned} f_t(U_t^1, A_t^1) - f_t(U_t^2, A_t^1) &\geq \langle \nabla_{U_t} f_t(U_t^2, A_t^1), U_t^1 - U_t^2 \rangle, \\ f_t(U_t^1, A_t^1) - f_t(U_t^1, A_t^2) &\geq \langle \nabla_{A_t} f_t(U_t^1, A_t^2), A_t^1 - A_t^2 \rangle. \end{aligned} \quad (17)$$

With the strong convexity of $g_1(\cdot)$ and $g_2(\cdot)$, we have

$$\begin{aligned} g_1(U_t^1) - g_1(U_t^2) &\geq \langle \nabla g_1(U_t^2), U_t^1 - U_t^2 \rangle + \frac{\sigma}{2} \|U_t^1 - U_t^2\|^2, \\ g_2(A_t^1) - g_2(A_t^2) &\geq \langle \nabla g_2(A_t^2), A_t^1 - A_t^2 \rangle + \frac{\sigma}{2} \|A_t^1 - A_t^2\|^2. \end{aligned} \quad (18)$$

Proof: From the bi-convexity of function $f_t(\cdot, \cdot)$, (17) can be shown directly. With $\sigma \in (0, \min\{\frac{\mu_1}{m}, \mu_2\})$, both $g_1(x) - \frac{\sigma}{2}\|x\|^2$ and $g_2(x) - \frac{\sigma}{2}\|x\|^2$ are convex. Hence $g_1(x)$ and $g_2(x)$ are strongly convex. ■

Since $\rho \mathbf{C}_t^\top \mathbf{C}_t$ is a diagonal matrix, we follow the *prox-linear case* to set diagonal matrices $\mathbf{P}_t = \tau_t \mathbf{I} - \rho \mathbf{C}_t^\top \mathbf{C}_t$ and $\mathbf{Q}_t = \zeta_t \mathbf{I}$. To prove the convergence of (FO-)DMTL-RSF, Our analysis mainly focuses on showing that the augmented Lagrangian $\mathcal{L}$ defined in (7) is lower bounded and monotonically non-increasing with iterations. Defining $\sigma_{t, \max}$ as the largest eigenvalue of $\mathbf{C}_t^\top \mathbf{C}_t$, then we have the following result.

Lemma 2. (Sufficient descent of $\mathcal{L}$ during $\mathbf{U}$ and λ update) For a given $\delta > 0$, choosing $\gamma_i^{k+1} \in (0, \frac{\delta \|\widehat{\mathbf{C}}_i(\mathbf{U}^k - \mathbf{U}^{k+1})\|^2}{\|\widehat{\mathbf{C}}_i \mathbf{U}^{k+1}\|^2}]$ ($i \in \mathcal{E}$) and $\tau_i \geq \rho m(\delta + \frac{1}{2})\sigma_{t, \max} - \frac{\sigma}{2}$, the iterations in DMTL-RSF satisfy

$$\mathcal{L}(\mathbf{U}^k, \mathbf{A}^k, \lambda^k) - \mathcal{L}(\mathbf{U}^{k+1}, \mathbf{A}^k, \lambda^{k+1}) \geq 0. \quad (19)$$

Proof: See the Appendix B. ■

Since agents are fully decentralized, a unique γ^{k+1} can not be guaranteed. Supposing $\mathbf{C}_t^\top \mathbf{C}_t = |d_t| \mathbf{I}$ and thus $\|\widehat{\mathbf{C}}_i \mathbf{U}^{k+1}\|^2 = \|U_t^{k+1} - U_j^{k+1}\|^2$ where $(t, j) \in \mathcal{E}$, it demonstrates that γ_i^{k+1} relates to the *primal* residual among agents t and j . The main idea is to bound the *primal* residue by the *dual* residual. This is because $\|U_t^{k+1} - U_j^{k+1}\|^2 \leq \frac{\delta}{\gamma_i^{k+1}} \|(U_t^k - U_j^k) - (U_t^{k+1} - U_j^{k+1})\|^2 \leq \frac{2\delta}{\gamma_i^{k+1}} (\|U_t^k - U_t^{k+1}\|^2 + \|U_j^k - U_j^{k+1}\|^2)$. Hence the δ and γ_i^{k+1} can balance the *primal* and *dual* residuals. Further, δ can also be adjusted in each iteration.

Lemma 3. (Sufficient descent of $\mathcal{L}$ during $\mathbf{A}$ update) Letting $\zeta_t \geq 0$, then the iterations in DMTL-RSF satisfy

$$\mathcal{L}(\mathbf{U}^{k+1}, \mathbf{A}^k, \lambda^{k+1}) - \mathcal{L}(\mathbf{U}^{k+1}, \mathbf{A}^{k+1}, \lambda^{k+1}) \geq 0. \quad (20)$$

Proof: See the Appendix C. ■

To combine the Lemmas 2 and 3, we can conclude that

$$\begin{aligned} \mathcal{L}(\mathbf{U}^k, \mathbf{A}^k, \lambda^k) - \mathcal{L}(\mathbf{U}^{k+1}, \mathbf{A}^{k+1}, \lambda^{k+1}) \\ \geq \sum_{t \in \mathcal{V}} \left(c_{t,1} \|U_t^k - U_t^{k+1}\|^2 + c_{t,2} \|A_t^k - A_t^{k+1}\|^2 \right), \end{aligned} \quad (21)$$

where $c_{t,1} = \tau_t + \frac{\sigma}{2} - \rho m(\delta + \frac{1}{2})\sigma_{t, \max}$ and $c_{t,2} = \zeta_t + \frac{\sigma}{2}$. Moreover, when $c_{t,1}, c_{t,2} \geq 0$, the decrease of $\mathcal{L}$ with iterations in DMTL-RSF can be guaranteed.

Lemma 4. (Boundedness) The sequence $\{\mathbf{U}^k, \mathbf{A}^k, \lambda^k\}$ generated by DMTL-RSF is bounded. $\mathcal{L}(\mathbf{U}^k, \mathbf{A}^k, \lambda^k)$ is lower bounded for all $k \in \mathbb{N}$ and converges as $k \rightarrow \infty$.

Proof: See the Appendix D. ■

Lemma 5. (Partial gradient bound) For any partial gradient $\partial \mathcal{L}(\mathbf{U}^{k+1}, \mathbf{A}^{k+1}, \lambda^{k+1})$, there exists $c > 0$ such that

$$\partial \mathcal{L}(\mathbf{U}^{k+1}, \mathbf{A}^{k+1}, \lambda^{k+1}) \leq c \sum_{t \in \mathcal{V}} \left(\|U_t^k - U_t^{k+1}\|^2 + \|A_t^k - A_t^{k+1}\|^2 \right). \quad (22)$$

Proof: See the Appendix E. ■

Based on Lemmas 2,3 and 4, we now establish the convergence of DMTL-RSF.

Theorem 1. Letting $\tau_t \geq \rho m(\delta + \frac{1}{2})\sigma_{t,\max} - \frac{\sigma}{2}$ and $\zeta_t \geq 0$, the sequence $\{U^k, A^k, \lambda^k\}$ generated by DMTL-RSF converges to a stationary point (U^*, A^*, λ^*) of $\mathcal{L}$ as $k \rightarrow \infty$.

Proof: By Lemmas 2 and 3, $\mathcal{L}(U^k, A^k, \lambda^k)$ is monotonically non-increasing and lower bounded, and therefore from (22), $\|U_t^k - U_t^{k+1}\|^2 \rightarrow 0$ and $\|A_t^k - A_t^{k+1}\|^2 \rightarrow 0$ as $k \rightarrow \infty$. Moreover from the proof of Lemma 2, $\|\lambda^k - \lambda^{k+1}\|^2 \rightarrow 0$ also holds when $k \rightarrow \infty$. Supposing that the sequence (U^k, A^k, λ^k) converges to a limit point (U^*, A^*, λ^*) , then from Lemma 4, there exists a convergent subsequence $(U^{k_s}, A^{k_s}, \lambda^{k_s})_{s \in \mathbb{N}}$ such that $(U^{k_s}, A^{k_s}, \lambda^{k_s}) \rightarrow (U^*, A^*, \lambda^*)$ as $s \rightarrow \infty$. Based on Lemma 5, $\|\partial \mathcal{L}(U^k, A^k, \lambda^k)\|^2 \rightarrow 0$ when $k \rightarrow \infty$, and in particular $\|\partial \mathcal{L}(U^{k_s}, A^{k_s}, \lambda^{k_s})\|^2 \rightarrow 0$ as $s \rightarrow \infty$. Since $f_t(\cdot, \cdot)$, $g_1(\cdot)$ and $g_2(\cdot)$ are continuous, we have $\lim_{s \rightarrow \infty} \mathcal{L}(U^{k_s}, A^{k_s}, \lambda^{k_s}) \rightarrow \mathcal{L}(U^*, A^*, \lambda^*)$. By Proposition 2 in [40], we have $\partial \mathcal{L}(U^*, A^*, \lambda^*) = 0$, which demonstrates that the limit point (U^*, A^*, λ^*) is a stationary point. That concludes the proof. ■

Following the analysis for DMTL-RSF, we next provide the convergence analysis of FO-DMTL-RSF.

Proposition 2. (Block-coordinate Lipschitz continuous) For $U_t^1, U_t^2 \in \mathcal{U}$ and $A_t \in \mathcal{A}$, there exists $L_t > 0^2$ such that

$$\|\nabla_{U_t} F_t(U_t^1, A_t) - \nabla_{U_t} F_t(U_t^2, A_t)\| \leq L_t \|U_t^1 - U_t^2\|. \quad (23)$$

Proof: Refer to the proof of Lemma 1, but with a different L_t rather than L_U . ■

Lemma 6. (Sufficient descent of $\mathcal{L}$ during U and λ update) Following the same update strategy for γ_i^{k+1} in Lemma 2, by guaranteeing $\tau_t \geq L_t + \rho m(\delta + \frac{1}{2})\sigma_{t,\max} - \frac{\sigma}{2}$, then the iterations in FO-DMTL-RSF satisfy

$$\mathcal{L}(U^k, A^k, \lambda^k) - \mathcal{L}(U^{k+1}, A^k, \lambda^{k+1}) \geq 0. \quad (24)$$

Proof: See the Appendix F. ■

Since the update of A_t is less complex compared with that of U_t , we do not introduce the first-order approximation. Hence with Lemma 3, the decreasing of $\mathcal{L}$ during A updating also holds with setting $\zeta_t \geq 0$.

Lemma 7. (Partial gradient bound) For the sequence $\{U^k, A^k, \lambda^k\}$ generated in FO-DMTL-RSF, and any partial gradient $\partial \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})$, there exists $c > 0$ such that

$$\partial \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1}) \leq c \sum_{t \in \mathcal{V}} \left(\|U_t^k - U_t^{k+1}\|^2 + \|A_t^k - A_t^{k+1}\|^2 \right). \quad (25)$$

Proof: See the Appendix G. ■

Then based on Lemmas 6 and 7, we can conclude the convergence of FO-DMTL-RSF as follows.

Theorem 2. Letting $\tau_t \geq L_t + \rho m(\delta + \frac{1}{2})\sigma_{t,\max} - \frac{\sigma}{2}$ and $\zeta_t \geq 0$, the sequence $\{U^k, A^k, \lambda^k\}$ generated by FO-DMTL-RSF converges to a stationary point (U^*, A^*, λ^*) of $\mathcal{L}$ as $k \rightarrow \infty$.

Proof: The proof is similar as that of Theorem 1. ■

²Since the boundedness of U and A is proved in Lemma 3, from the proof of Lemma 5 we suppose that the Lipschitz constant for gradient $\nabla F(U, A)$ in bounded set is consistent with the block-coordinate Lipschitz continuous constant L_t .

As presented in Theorems 1 and 2, we need to choose larger proximal penalty τ_t for FO-DMTL-RSF than that of DMTL-RSF, while the conditions for other parameters are the same. Note that the Theorem 1 and Theorem 2 present the sufficient conditions for the convergence guarantee. Furthermore, convergence analysis for both DMTL-RSF and FO-DMTL-RSF can be applied to the general problem (16) with multi-convex objective $\phi(U, A)$. Since we do not specify the mapping $G(w, b, X)$, the results in Theorems 1 and 2 hold for different choice of activation functions but with appropriate parameters correspondingly.

V. NUMERICAL EXPERIMENTS

In this section, we will first provide detailed numerical results on the convergence behavior of algorithms MTL-RSF and DMTL-RSF. Then the generalization performance of proposed methods are compared with state-of-the-art approaches. Finally we will discuss the communication requirement of the decentralized method DMTL-RSF over learning accuracy. Simulations are carried out in MATLAB R2019a on a 2.60 GHz computer with 8 GB memory.

A. Convergence experiments

We first evaluate the convergence behavior of proposed algorithms for solving problems (P-1) and (P-2). For simplicity, we set $m = \{5, 10\}$, $L = \{5, 10, 50\}$, $N_t = 100$, $r = 2$ and $d = 1$. The regularization parameters are chosen as $\mu = \nu = 2$. The hidden layer feature matrix H_t and the training labels T_t are generated randomly according to the uniform distribution $U(0, 1)$. Denoting $H = [H_1^T, \dots, H_m^T]^T$, the columns of H are normalized. For the decentralized scenario, we consider the network structure illustrated in Fig. 2 (a). We set the parameter $\rho = 1$. With Prox-linear Proximal we denote $P_t = \tau_t I - \rho C_t^T C_t$ and $Q_t = \zeta_t I$. Without loss of generality, we set $C_t^T C_t = d_t I$ where d_t is the degree of agent t in graph $\mathcal{G}$. We generate the connected network $\mathcal{G}$ randomly with m agents and $|\mathcal{E}| = \frac{m(m-1)}{2}\eta$ links. The connection density is set as $\eta = 0.4$.

The convergence behavior of MTL-RSF, DMTL-RSF and FO-DMTL-RSF are presented in Fig. 3. Besides, to show the advantage of the proposed decentralized algorithms, we compare the convergence performance with decentralized gradient descent (DGD) method [45]. In each iteration of DGD, the update of $\{U_t\}$ is first implemented in parallel as (4) in [45]. After sharing $\{U_t\}$ across agents, $\{A_t\}$ is updated through gradient descent approach. We set the step sizes for updating $\{U_t\}$ and $\{A_t\}$ as 0.1 in DGD.

Since MTL-RSF is a centralized method, it has the fastest convergence speed in Fig. 3. While DGD has the slowest convergence speed among the methods. We denote τ_t, ζ_t the parameters of DMTL-RSF while $\bar{\tau}_t, \bar{\zeta}_t$ as those of FO-DMTL-RSF. For Fig. 3 (a) and (b), we set $L = 10$, $m = 5$, $\delta = 10$ and $\gamma_i^{k+1} = \min\{1, \frac{\delta \|\bar{C}_i U^k - \bar{C}_i U^{k+1}\|^2}{\|\bar{C}_i U^{k+1}\|^2}\}$. As shown in Fig. 3 (a), both MTL-RSF and DMTL-RSF can converge except FO-DMTL-RSF, where $\tau_t = \bar{\tau}_t = 1 + d_t$ and $\zeta_t = \bar{\zeta}_t = 1$. With $\tau_t = \bar{\tau}_t = 2 + d_t$ and $\zeta_t = \bar{\zeta}_t = 2$, the convergence happens in Fig. 3 (b), where conditions in Theorems 1 and 2

Fig. 3. (a) Objective value with $\tau_t = \bar{\tau}_t = 1 + d_t$, $\zeta_t = \bar{\zeta}_t = 1$; (b) Objective value with $\tau_t = \bar{\tau}_t = 2 + d_t$, $\zeta_t = \bar{\zeta}_t = 2$; (c), (d) evolution of $U_t^k(\bar{U}_t^k)$ and $A_t^k(\bar{A}_t^k)$ with $\tau_t = 1 + d_t$, $\zeta_t = 1$ and $\bar{\tau}_t = 2 + d_t$, $\bar{\zeta}_t = 2$; (e), (f) accuracy for $U_t^k(\bar{U}_t^k)$ and $A_t^k(\bar{A}_t^k)$ with $\tau_t = 1 + d_t$, $\zeta_t = 1$ and $\bar{\tau}_t = 2 + d_t$, $\bar{\zeta}_t = 2$.

are satisfied. However, comparing (a) with (b), the convergence speed for these two cases is different for both DMTL-RSF and FO-DMTL-RSF. This is because larger parameters τ_t ($\bar{\tau}_t$) and ζ_t ($\bar{\zeta}_t$) reduce the update step of U_t and A_t . Thus, we can trade off the convergence speed for weaker convergence requirement by adjusting the parameters of algorithms DMTL-RSF and FO-DMTL-RSF. In Fig. 3, though the FO-DMTL-RSF decreases faster than DMTL-RSF at the beginning, there exists jitter in the objective value for FO-DMTL-RSF method, which is due to the first-order approximation. This also explains the slow convergence for DGD method.

We also reveal the element evolution of U_t and A_t for DMTL-RSF and FO-DMTL-RSF in Fig. 3 (c) and (d), respectively, where $\tau_t = 1 + d_t$, $\bar{\tau}_t = 2 + d_t$, $\zeta_t = 1$, $\bar{\zeta}_t = 2$ and $L = 10$, $m = 10$. Since $U_t \in \mathbb{R}^{10 \times 2}$, we only compare the element $U_t^k(1, 1)$ and $\bar{U}_t^k(1, 1)$ with $\bar{U}^{k=1000}(1, 1)$ obtained by MTL-RSF algorithm, which is presented by the dashed line in Fig. 3 (c). From this figure, we can conclude that algorithm DMTL-RSF can ensure all the agents a unique subspace U^* . Moreover in Fig. 3 (d), the local predictor $A_t^k(1, 1)$ and $\bar{A}_t^k(1, 1)$ of tasks updated in DMTL-RSF and FO-DMTL-RSF algorithms converges to the corresponding $\bar{A}_t^{k=1000}(1, 1)$ from MTL-RSF with carrying out more iterations. Hence the optimal A_t^* can also be guaranteed for each task. The evolution of $\bar{U}_t^k$ and $\bar{A}_t^k$ jitters with iterations. This is consistent with Fig. 3 (b).

The accuracy of $U_t^k(\bar{U}_t^k)$ and $A_t^k(\bar{A}_t^k)$, which are defined as $(\frac{1}{mLr} \sum_{t=1}^m \|U_t^k(\bar{U}_t^k) - \bar{U}^{k=1000}\|^2)^{1/2}$ and $(\frac{1}{mrd} \sum_{t=1}^m \|A_t^k(\bar{A}_t^k) - \bar{A}_t^{k=1000}\|^2)^{1/2}$, is presented in Fig. 3 (e) and (f). Without specifying, we set $m = 5$, $L = 10$, $\tau_t = 1 + d_t$, $\bar{\tau}_t = 2 + d_t$ and $\zeta_t = 1$, $\bar{\zeta}_t = 2$. With a fixed agent number and input dimension, the accuracy for FO-DMTL-RSF has a faster decrease speed at the beginning of iterations than DMTL-RSF. However, when k grows, the DMTL-RSF can guarantee more accurate outputs.

When we scale up the problem size, i.e., increasing the number of agents m or the hidden feature space dimension L , the convergence speed of DMTL-ELM reduces with fixed step size τ_t and ζ_t . Besides, from Fig. 3 (e) and (f), the proposed methods, i.e., DMTL-RSF and FO-DMTL-RSF, have a faster convergence speed, and can guarantee much higher accuracy than DGD.

B. Generalization Performance

We perform extensive studies to evaluate the proposed approaches by empirical comparison with the following baselines for centralized and distributed MTL, respectively.

- *Local RSF*: a baseline single-task learning method utilizing RSF, in which the output weight of each task is learned separately by tasks;
- *Multi-Task Feature Learning (MTFL)* [5]: an AO based algorithm solving the equivalent convex problem with respect to the weight w_t of tasks and the correlation matrix Ω ;
- *Grouping and Overlap for Multi-Task learning (GO-MTL)* [8]: a framework for multi-task learning assuming that each task parameter vector is a linear combination of a finite number of underlying basis tasks;
- *Distributed Gradient Subspace Pursuit (DGSP)* [24]: a distributed approach aiming to learn an unknown shared low-dimensional subspace for related tasks. The subspace gets supplemented at all tasks in each iteration;
- *Distributed Newton Subspace Pursuit (DNSP)* [24]: the same as approach DGSP except the way to pursue the shared subspace, which substitutes gradient direction with Newton direction in iterations.

The Local RSF is a separate approach. MTFL and GO-MTL are centralized methods, while DGSP and DNSP are

Type	Local method				Centralized methods							
Data	Local RSF				MTFL		GO-MTL		MTL-RSF			
	ReLU		Sigmoid		Testing error	Training time	Testing error	Training time	ReLU		Sigmoid	
	Testing error	Training time	Testing error	Training time					Testing error	Training time	Testing error	Training time
USPS	5.89	0.007	4.26	0.009	4.67	0.10	6.30	7.52	3.89	198.4	3.49	226.1
MNIST	8.60	0.006	6.58	0.004	6.84	0.20	9.76	8.10	5.82	228.7	5.90	244.2

Type	Distributed methods											
Data	DGSP		DNSP		DMTL-RSF				FO-DMTL-RSF			
	Testing error	Training time	Testing error	Training time	ReLU		Sigmoid		ReLU		Sigmoid	
					Testing error	Training time	Testing error	Training time	Testing error	Training time	Testing error	Training time
USPS	5.05	0.03	4.47	0.04	4.35	163.7	3.54	184.2	4.78	20.1	3.89	22.5
MNIST	7.90	0.04	7.35	0.07	6.72	171.5	5.96	192.6	7.20	18.0	6.20	19.7

TABLE I
COMPARISON OF TESTING ERROR (%) AND TRAINING TIME (S) FOR DIFFERENT LEARNING APPROACHES AND TRAINING DATA SETS.

Fig. 4. Testing error comparison for Local RSF, MTL-RSF, DMTL-RSF and FO-DMTL-RSF with state-of-the-art methods over USPS dataset.

distributed approaches. Since the methods DGSP and DNSP can only work in the master-slave structure, we consider the decentralized setting for DMTL-RSF and FO-DMTL-RSF. Without loss of generality, we set $\mathcal{E} = \{(1, i) | \forall i \in \mathcal{V}, i \neq 1\}$.

Our goal is to compare the proposed centralized and decentralized multi-task learning methods with the corresponding baseline approaches. For completeness, we examine the proposed MTL-RSF, DMTL-RSF and FO-DMTL-RSF algorithms as well as state-of-the-art methods over the following two real-world datasets with classification tasks. According to experiments in [41], we adopt Sigmoid additive hidden node of which the activation function is

$$G(\mathbf{w}_l, b_l, \mathbf{X}) = \frac{1}{1 + \exp(-(\mathbf{w}_l^\top \mathbf{X} + b))}. \quad (26)$$

For comparison, we also evaluate the generalization performance of RSF based methods by adopting Rectified Linear Units (ReLU) as activation function, specified by

$$G(\mathbf{w}_l, \mathbf{X}) = \max\{0, \mathbf{w}_l^\top \mathbf{X}\}. \quad (27)$$

For algorithms MTL-RSF, DMTL-RSF and FO-DMTL-RSF, we run the iterations for $k = 100$ times, respectively.

Meanwhile in DMTL-RSF and FO-DMTL-RSF, we use *Standard Proximal* P_t and Q_t , and set parameters $\rho = 1$, $\delta = 100$, $\gamma_i^{k+1} = \min\{1, \frac{\delta \|\hat{C}_i(\mathbf{U}^k - \mathbf{U}^{k+1})\|^2}{\|\hat{C}_i \mathbf{U}^{k+1}\|^2}\}$. We run simulations over the following data sets for 100 times and average the testing error and training time.

*USPS digits dataset*²: This is a handwritten digits dataset [7] with 10 classes and 256 input dimension. The images are preprocessed using principle component analysis (PCA) and dimensionality is reduced to 64 with retaining almost 95% of the variance. We extract 1350 samples, 900 samples out of which are used for training while the other 450 samples are used for testing. We set the task number as $m = 10$, where each task conducts classification over 3 random classes. Meanwhile the training and testing samples for each task are randomly and equivalently allocated. The regularization parameters of MTL-RSF, DMTL-RSF and FO-DMTL-RSF are chosen as $\mu_1 = \mu_2 = 10$ with ReLU and $\mu_1 = \mu_2 = \sqrt{10}$ with Sigmoid, respectively. We set $\tau_t = 10 + d_t$ and $\zeta_t = 30$ for DMTL-RSF while $\bar{\tau}_t = 20 + d_t$, $\bar{\zeta}_t = 30$ for FO-DMTL-RSF. The parameters for MTFL are set as $\gamma = 10$ and $\epsilon = 20$, while the regularization parameter for other methods is set as $\lambda = 10$. The hyperparameters for different methods are optimized through exhaustive experiments.

The generalization performance for the proposed methods is presented in Fig. 4. By sharing information across tasks to learn output weights, the MTL-RSF, DMTL-RSF and FO-DMTL-RSF schemes can achieve smaller testing error than the separate learning method Local RSF. Since DMTL-RSF can converge to the outputs of MTL-RSF with $k = 100$, the DMTL-RSF has ignorable performance loss over testing error compared with MTL-RSF. Due to the first-order approximation, the outputs of FO-DMTL-RSF does not approach the optimum with $k = 100$ iterations, the testing error performance decays from DMTL-RSF but still outperform Local-RSF method. From Fig. 4, the testing error of MTL-RSF, DMTL-RSF with $L \geq 150$ and FO-DMTL-RSF with $L \geq 200$ are smaller than that of MTFL and DNSP methods. Since the dimension of A_t is irrelevant to L , the dimension of shared

²Thanks [7] for providing us the data, which is available at <https://zhuoliang.me/research.html>.

Fig. 5. Testing error of DMTL-RSF with the communication ratio of DMTL-RSF to DNSP over the USPS dataset.

subspace U will be expanded with L , which separates the latent tasks more easily. Hence the generalization performance for all the methods based on RSF can be improved. The testing errors of Local RSF, MTL-RSF and DMTL-RSF and FO-DMTL-RSF in Table 1 are obtained with $L = 300$, where $\tau_t = 20 + d_t$, $\zeta_t = 40$ for DMTL-RSF and $\bar{\tau}_t = 30 + d_t$, $\bar{\zeta}_t = 40$ for FO-DMTL-RSF.

MNIST digits dataset: This is also a digit dataset [7] with 10 classes and 784 input dimensions. We reduce the dimensions of images to 87 by using PCA. We extract 1350 samples, 900 samples out of which are used for training while the other 450 samples are used for testing. We set the task number as $m = 10$, where each task conducts classification over 3 random classes. The training and testing samples for each task are randomly and equivalently allocated. The regularization parameters of MTL-RSF, DMTL-RSF and FO-DMTL-RSF are chosen as $\mu_1 = \mu_2 = 20$ with ReLU and $\mu_1 = \mu_2 = \sqrt{20}$ with Sigmoid, respectively. We set $\lambda = 100$ for GO-MTL. We set the parameters for MTL as $\gamma = 10$ and $\epsilon = 20$, and the regularization parameter for other methods is set as $\lambda = 20$. The hyperparameters for different methods are optimized through exhaustive experiments. To obtain the testing errors given in Table 1, we set the number of hidden neurons as $L = 300$ for Local RSF, MTL-RSF and DMTL-RSF methods, and $\tau_t = 20 + d_t$, $\zeta_t = 40$ for DMTL-RSF and $\bar{\tau}_t = 30 + d_t$, $\bar{\zeta}_t = 40$ for FO-DMTL-RSF.

Table 1 presents that the testing error achieved by MTL-RSF, DMTL-RSF and FO-DMTL-RSF can outperform other MTL methods over the tested data sets. We can also see that, with USPS dataset, the testing errors achieved by MTL-RSF and DMTL-RSF outperform local RSF by 34.0% and 26.2% with ReLU, and 18.1% and 16.9% with Sigmoid. While with MNIST dataset, the testing errors achieved by MTL-RSF and DMTL-RSF outperform local RSF by 32.3% and 21.9% with ReLU, and 10.3% and 9.4% with Sigmoid, respectively. Numerical results demonstrate that in the shortage of training data, the proposed multi-task learning algorithms, which train the models of tasks together, can effectively improve the generation performance for each task from separate learning.

Meanwhile, the proposed DMTL-RSF algorithm can achieve almost the same testing error compared with centralized MTL-RSF. Though with decayed generalization performance with same iteration k , the training time of FO-DMTL-RSF is much faster than MTL-RSF and DMTL-RSF. Among all the tested methods, the Local RSF has the shortest training time.

C. Communication load of DMTL

The communication load of distributed learning approaches is important in real-world applications, since excessive information exchanged can cause system overload. Since the communication load of DMTL-RSF and FO-DMTL-RSF are the same when the iterations k is identical. Then we only evaluate the communication loads and testing error of the proposed DMTL-RSF algorithm compared with the DMTL method DNSP, which can achieve better generalization performance than DGSP. In each iteration of DMTL-RSF, the information that agent t broadcasts to its neighbors is U_t . Hence while the decentralized network structure is fixed, the communication load of DMTL-RSF is determined by the number of iterations k and dimension of U_t , which is the number of hidden neurons L . In each iteration of the DNSP method, which works in the master-slave network structure, the updated U_t in slaves are first sent to the master. Then a new column u of the desired subspace U is calculated and broadcast to slaves. But the difference between DNSP and DMTL-RSF method is that the iterations of the former is only r , which is the number of latent tasks, instead of the k in the latter one. Thus for the master-slave structure, the communication load of DMTL-RSF is $\frac{2kL}{(r+1)n}$ times that of DNSP.

In Fig. 5, we present the testing error of DMTL-RSF algorithm with respect to the communication load ratio of DMTL-RSF to DNSP method when using USPS dataset and Sigmoid activation function, where $L \in \{100, 150, 200, 250, 300\}$ and $k \in \{25, 50, 100\}$. Since the number of latent tasks is identical, increasing k and L will enlarge the communication load of DMTL-RSF, as well as guarantee a smaller testing error. Thus we can conclude that there exists a trade-off between the generalization performance of DMTL-RSF and the communication load. For the considered regions of k and L in Fig. 5, the communication load of DNSP is smaller than that of DMTL-RSF since the ratio is larger than 1. When we fix the iteration rounds as $k = 25$ of DMTL-RSF, the testing error of DNSP is always smaller than that of DMTL-RSF even rough increasing the hidden neurons L in considered region. This is because the outputs $\{U_t\}$ and $\{A_t\}$ of DMTL-RSF with $k = 25$ are far away from the optimal value. However, when k increase to 50 and 100, the DMTL-RSF can achieve better generalization performance than DNSP by enlarging L , with larger communication load than DNSP. Moreover from Fig. 5, when we set $k = 50$, $L = 300$, the generalization performance is better than that of $k = 50$, $L = 200$, but with a smaller communication cost. Thus, in real applications, we can choose appropriate values of k and L to balance the communication load and generalization performance of DMTL-RSF.

VI. CONCLUSIONS

We study the multi-task learning problem based on an RSF implementation. We first introduce the centralized MTL problem based on RSF, and present the AO-based algorithm MTL-RSF to solve the problem. Then we extend to the decentralized scenario by formulating the DMTL problem as a majorized multi-block optimization with coupled bi-convex objective functions. It is solved by our proposed algorithm DMTL-RSF, which is a hybrid Jacobian and Gauss-Seidel ADMM. The computation load of DMTL-RSF is further reduced by introducing first-order approximation in the proposed FO-DMTL-RSF algorithm. Through analysis we prove the convergence of MTL-RSF, DMTL-RSF and FO-DMTL-RSF, as well as presenting the required conditions for parameters. Simulations verified the convergence of all the approaches, and show that the generalization performance of MTL-RSF, DMTL-RSF and FO-DMTL-RSF can outperform state-of-the-art MTL methods by randomly mapping the input of each task to a hidden feature space and learning the output weights together across tasks. Moreover through adjusting the number of hidden neurons of RSF and iteration rounds, we present the trade-off between communication load and the generalization performance of DMTL-RSF algorithm.

VII. APPENDICES

A. Proof of Lemma 1

We simplify (P-1) as $F(U, A) = f(U, A) + g_1(U) + g_2(A)$, where $g_1(\cdot)$ and $g_2(\cdot)$ are the regularization terms. $g_1(\cdot)$ and $g_2(\cdot)$ are differentiable convex functions and $f(\cdot, \cdot)$ is a continuously differentiable function over U and A . We first prove that the gradient of $f(\cdot, \cdot)$ is *block-coordinate Lipschitz continuous*. Since when $\|U\| \rightarrow \infty$ or $\|A\| \rightarrow \infty$, it makes $F(U, A) \rightarrow \infty$. Hence we have $U < \infty$ and $A < \infty$. Then for a fixed $A \in \mathcal{A}$, there exists $L_U \in (0, \infty)$ such that

$$\begin{aligned} & \|\nabla_U f(U + d_U, A) - \nabla_U f(U, A)\| \\ &= \left\| \sum_{t \in \mathcal{V}} H_t^\top H_t d_U A_t A_t^\top + \mu_1 d_U \right\| \\ &\stackrel{(a)}{=} \left\| \left(\sum_{t \in \mathcal{V}} (A_t A_t^\top) \otimes (H_t^\top H_t) + \mu_1 I \right) d_U \right\| \\ &\stackrel{(b)}{\leq} \left\| \sum_{t \in \mathcal{V}} (A_t A_t^\top) \otimes (H_t^\top H_t) + \mu_1 I \right\| \|d_U\| \leq L_U \|d_U\|, \quad (28) \end{aligned}$$

where (a) is since $\|\cdot\| = \|\text{vec}(\cdot)\|$, while (b) is from the *sub-multiplicativity* of frobenius norm. The L_U is Lipschitz constant for block U . Similarly, with fixed $U \in \mathcal{U}$, we have

$$\begin{aligned} & \|\nabla_A f(U, A + d_A) - \nabla_A f(U, A)\| = \\ & \left\| (\text{blkdiag}(U^\top H_1^\top H_1 U, \dots, U^\top H_m^\top H_m U) + \mu_2 I) d_A \right\| \leq L_A \|d_A\|. \quad (29) \end{aligned}$$

Correspondingly L_A is Lipschitz constant for block A . Hence according to Lemma 3.2 in [25], with $k \rightarrow \infty$, the sequence $\{U^k, A^k\}$ generated by DMTL-RSF will converge to a stationary point (U^*, A^*) .

B. Proof of Lemma 2

Recall the update of U_t^{k+1} in (10), we obtain the optimality condition as

$$\begin{aligned} & \nabla_{U_t} f_t(U_t^{k+1}, A_t^k) + \nabla_{g_1}(U_t^{k+1}) \\ &= -C_t^\top \lambda^k - \rho C_t^\top (C_t U_t^{k+1} + \sum_{i \neq t} C_i U_i^k) + P_t (U_t^k - U_t^{k+1}) \\ &= -C_t^\top \lambda^k + \rho C_t^\top C_t (U_t^k - U_t^{k+1}) - \rho C_t^\top C U^k + P_t (U_t^k - U_t^{k+1}) \\ &= -C_t^\top \lambda^k + \tau_t (U_t^k - U_t^{k+1}) - \rho C_t^\top C U^k. \quad (30) \end{aligned}$$

Using the inequalities in Proposition 2, we have

$$\begin{aligned} & f_t(U_t^k, A_t^k) + g_1(U_t^k) - f_t(U_t^{k+1}, A_t^k) - g_1(U_t^{k+1}) \\ &\geq \langle \nabla_{U_t} f_t(U_t^{k+1}, A_t^k) + \nabla_{g_1}(U_t^{k+1}), U_t^k - U_t^{k+1} \rangle \\ &\quad + \frac{\sigma}{2} \|U_t^k - U_t^{k+1}\|^2 \\ &= -\langle \lambda^k, C_t (U_t^k - U_t^{k+1}) \rangle - \rho \langle C U^k, C_t (U_t^k - U_t^{k+1}) \rangle \\ &\quad + \left(\tau_t + \frac{\sigma}{2} \right) \|U_t^k - U_t^{k+1}\|^2. \quad (31) \end{aligned}$$

Sum the above inequality over all t to get

$$\begin{aligned} & \sum_{t \in \mathcal{V}} [f_t(U_t^k, A_t^k) + g_1(U_t^k) - f_t(U_t^{k+1}, A_t^k) - g_1(U_t^{k+1})] \\ &\geq -\langle \lambda^k, C(U^k - U^{k+1}) \rangle - \rho \langle C U^k, C(U^k - U^{k+1}) \rangle \\ &\quad + \sum_{t \in \mathcal{V}} \left(\tau_t + \frac{\sigma}{2} \right) \|U_t^k - U_t^{k+1}\|^2. \quad (32) \end{aligned}$$

To track the difference of $\mathcal{L}$ caused by updating U and λ , we should also analyze the following term.

$$\begin{aligned} & \langle \lambda^k, C U^k \rangle - \langle \lambda^{k+1}, C U^{k+1} \rangle + \frac{\rho}{2} \|C U^k\|^2 - \frac{\rho}{2} \|C U^{k+1}\|^2 \\ &= \langle \lambda^k, C(U^k - U^{k+1}) \rangle + \langle \lambda^k - \lambda^{k+1}, C U^{k+1} \rangle \\ &\quad + \frac{\rho}{2} \|C U^k\|^2 - \frac{\rho}{2} \|C U^{k+1}\|^2 \\ &\stackrel{(a)}{=} \langle \lambda^k, C(U^k - U^{k+1}) \rangle + \langle \lambda^k - \lambda^{k+1}, C U^{k+1} \rangle \\ &\quad + \frac{\rho}{2} \|C(U^k - U^{k+1})\|^2 + \rho \langle C U^{k+1}, C(U^k - U^{k+1}) \rangle, \quad (33) \end{aligned}$$

where (a) is from truth that $\|a + c\|^2 - \|b + c\|^2 = \|a - b\|^2 + 2\langle a - b, b + c \rangle$. Then combining (32) and (33) together with $\lambda^{k+1} = \lambda^k + \rho \gamma^{k+1} C U^{k+1}$, the target becomes

$$\begin{aligned} & \mathcal{L}(U^k, A^k, \lambda^k) - \mathcal{L}(U^{k+1}, A^k, \lambda^{k+1}) \\ &= \sum_{t \in \mathcal{V}} [f_t(U_t^k, A_t^k) + g_1(U_t^k) - f_t(U_t^{k+1}, A_t^k) - g_1(U_t^{k+1})] \\ &\quad + \langle \lambda^k, C U^k \rangle - \langle \lambda^{k+1}, C U^{k+1} \rangle + \frac{\rho}{2} \|C U^k\|^2 - \frac{\rho}{2} \|C U^{k+1}\|^2 \\ &\geq \langle \lambda^k - \lambda^{k+1}, C U^{k+1} \rangle - \frac{\rho}{2} \|C(U^k - U^{k+1})\|^2 \\ &\quad + \sum_{t \in \mathcal{V}} \left(\tau_t + \frac{\sigma}{2} \right) \|U_t^k - U_t^{k+1}\|^2 \\ &= -\frac{1}{\rho} \langle \lambda^k - \lambda^{k+1}, (\gamma^{k+1})^{-1} (\lambda^k - \lambda^{k+1}) \rangle - \frac{\rho}{2} \|C(U^k - U^{k+1})\|^2 \\ &\quad + \sum_{t \in \mathcal{V}} \left(\tau_t + \frac{\sigma}{2} \right) \|U_t^k - U_t^{k+1}\|^2. \quad (34) \end{aligned}$$

Since our goal is to guarantee the decrease of $\mathcal{L}$ with iterations, we then turn to bound the terms $\langle \lambda^k - \lambda^{k+1}, (\gamma^{k+1})^{-1} (\lambda^k - \lambda^{k+1}) \rangle$ and $\|C(U^k - U^{k+1})\|^2$ by

$\sum_{t \in \mathcal{V}} \|U_t^k - U_t^{k+1}\|^2$. Since agents are decentralized, the parameter $\gamma_i (i = 1, \dots, |\mathcal{E}|)$ is chosen by connected agents separately. Denoting $\lambda = [\lambda_1^\top, \dots, \lambda_{|\mathcal{E}|}^\top]^\top$ and ensuring $\gamma_i^{k+1} \leq \frac{\delta \|\widehat{C}_i(U^k - U^{k+1})\|^2}{\|\widehat{C}_i U^{k+1}\|^2}$, then we can derive

$$\begin{aligned} & \frac{1}{\rho} \langle \lambda^k - \lambda^{k+1}, (\gamma^{k+1})^{-1} (\lambda^k - \lambda^{k+1}) \rangle = \sum_{i \in \mathcal{E}} \frac{1}{\rho \gamma_i^{k+1}} \|\lambda_i^k - \lambda_i^{k+1}\|^2 \\ & = \sum_{i \in \mathcal{E}} \rho \gamma_i^{k+1} \|\widehat{C}_i U^{k+1}\|^2 \leq \sum_{i \in \mathcal{E}} \rho \delta \|\widehat{C}_i (U^k - U^{k+1})\|^2 \\ & = \rho \delta \|C(U^k - U^{k+1})\|^2. \end{aligned} \quad (35)$$

Further, we have

$$\begin{aligned} \|C(U^k - U^{k+1})\|^2 &= \left\| \sum_{t \in \mathcal{V}} C_t (U_t^k - U_t^{k+1}) \right\|^2 \\ &\stackrel{(a)}{\leq} m \sum_{t \in \mathcal{V}} \|C_t (U_t^k - U_t^{k+1})\|^2 \stackrel{(b)}{<} m \sum_{t \in \mathcal{V}} \sigma_{t, \max} \|U_t^k - U_t^{k+1}\|^2, \end{aligned} \quad (36)$$

where (a) is from the AM-GM inequality. Since $C_t^\top C_t$ is Hermitian matrix and also positive definite, (b) can be satisfied. Then we can guarantee the decrease of $\mathcal{L}$ by updating U and λ in DMTL-RSF, if we ensure $\tau_t \geq \rho m (\delta + \frac{1}{2}) \sigma_{t, \max} - \frac{\sigma}{2}$,

$$\begin{aligned} & \mathcal{L}(U^k, A^k, \lambda^k) - \mathcal{L}(U^{k+1}, A^k, \lambda^{k+1}) \\ & > \sum_{t \in \mathcal{V}} \left(\tau_t + \frac{\sigma}{2} - \rho m \left(\delta + \frac{1}{2} \right) \sigma_{t, \max} \right) \|U_t^k - U_t^{k+1}\|^2 \geq 0. \end{aligned} \quad (37)$$

That completes the proof.

C. Proof of Lemma 3

When updating A_t at agents in parallel, U_t^{k+1} and λ^{k+1} are fixed. From the optimality condition of (21) we have

$$\nabla_{A_t} f_t(U_t^{k+1}, A_t^{k+1}) + \nabla g_2(A_t^{k+1}) = \zeta_t (A_t^k - A_t^{k+1}). \quad (38)$$

Then with Proposition 1 and letting $\zeta_t \geq 0$, the evolution of $\mathcal{L}$ by updating A follows

$$\begin{aligned} & \mathcal{L}(U^{k+1}, A^k, \lambda^{k+1}) - \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1}) \\ &= \sum_{t \in \mathcal{V}} (f_t(U_t^{k+1}, A_t^k) - f_t(U_t^{k+1}, A_t^{k+1}) + g_2(A_t^k) - g_2(A_t^{k+1})) \\ &\geq \sum_{t \in \mathcal{V}} (\langle \nabla_{A_t} f_t(U_t^{k+1}, A_t^{k+1}) + \nabla g_2(A_t^{k+1}), A_t^k - A_t^{k+1} \rangle \\ &\quad + \frac{\sigma}{2} \|A_t^k - A_t^{k+1}\|^2) \\ &= \sum_{t \in \mathcal{V}} \left(\zeta_t + \frac{\sigma}{2} \right) \|A_t^k - A_t^{k+1}\|^2 \geq 0. \end{aligned} \quad (39)$$

That completes the proof.

D. Proof of Lemma 4

To ensure the boundedness of the sequence (U^k, A^k, λ^k) , we only need the coercivity of the objective function within the feasible set. Define the feasible set $\mathcal{F} := \{(U, A) : CU = \mathbf{0}\}$. Since $F(U, A) \rightarrow \infty$ while $(U, A) \in \mathcal{F}$ and $\|(U, A)\| \rightarrow \infty$,

the objective function $F(U, A)$ is coercive over $\mathcal{F}$. Hence we can deduce that $\{U^k, A^k\}$ are bounded. Furthermore (35) implies the boundedness of λ^k . Since $F(U^k, A^k) \geq \min_{U, A} \{F(U, A) : CU = \mathbf{0}\} \geq 0 > -\infty$, then with boundedness of $\{U^k, A^k, \lambda^k\}$ we have

$$\mathcal{L}(U^k, A^k, \lambda^k) = F(U^k, A^k) + \langle \lambda^k, CU^k \rangle + \frac{\rho}{2} \|CU^k\|^2 > -\infty. \quad (40)$$

From Lemmas 2 and 3, $\mathcal{L}(U^k, A^k, \lambda^k)$ is monotonically non-increasing, therefore it will converges with $k \rightarrow \infty$.

E. Proof of Lemma 5

We first prove that the derivative $\nabla_{\lambda} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})$ is upper bounded. Since $\gamma_i^{k+1} \|\widehat{C}_i U^{k+1}\|^2 \leq \delta \|\widehat{C}_i (U^k - U^{k+1})\|^2$, there exists $c_0 \geq \max\{\frac{\delta}{\gamma_i^{k+1}}\}$ such that

$$\begin{aligned} \|\nabla_{\lambda} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})\|^2 &= \|CU^{k+1}\|^2 = \sum_{i \in \mathcal{E}} \|\widehat{C}_i U^{k+1}\|^2 \\ &\leq \sum_{i \in \mathcal{E}} \frac{\delta}{\gamma_i^{k+1}} \|\widehat{C}_i (U^k - U^{k+1})\|^2 \leq c_0 \|C(U^k - U^{k+1})\|^2 \\ &\leq c_0 m \sum_{t \in \mathcal{V}} \sigma_{t, \max} \|U_t^k - U_t^{k+1}\|^2. \end{aligned} \quad (41)$$

With the equality (38), we can derive

$$\|\nabla_{A_t} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})\|^2 = \zeta_t^2 \|A_t^k - A_t^{k+1}\|^2. \quad (42)$$

Next, we show that $\nabla_{U_t} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})$ can also be upper bounded as

$$\begin{aligned} & \|\nabla_{U_t} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})\|^2 \\ &= \|\nabla_{U_t} F_t(U_t^{k+1}, A_t^{k+1}) + C_t^\top (\lambda^{k+1} + CU^{k+1})\|^2 \\ &\leq 2 \|\nabla_{U_t} F_t(U_t^{k+1}, A_t^{k+1}) - \nabla_{U_t} F_t(U_t^{k+1}, A_t^k)\|^2 \\ &\quad + 2 \|\nabla_{U_t} F_t(U_t^{k+1}, A_t^k) + C_t^\top (\lambda^{k+1} + CU^{k+1})\|^2 \\ &\stackrel{(a)}{\leq} 2 \|\nabla_{U_t} F_t(U_t^{k+1}, A_t^k) + C_t^\top (\lambda^{k+1} + \rho CU^{k+1})\|^2 \\ &\quad + 2L_t^2 \|A_t^k - A_t^{k+1}\|^2 \\ &= 2 \|C_t^\top (\lambda^{k+1} - \lambda^k) + \tau_t (U_t^k - U_t^{k+1}) - \rho C_t^\top C(U^k - U^{k+1})\|^2 \\ &\quad + 2L_t^2 \|A_t^k - A_t^{k+1}\|^2 \\ &\leq 6\sigma_{t, \max} \|\lambda^k - \lambda^{k+1}\|^2 + 6\rho^2 \sigma_{t, \max} \|C(U^k - U^{k+1})\|^2 \\ &\quad + 6\tau_t^2 \|U_t^k - U_t^{k+1}\|^2 + 2L_t \|A_t^k - A_t^{k+1}\|^2 \\ &\leq 6\rho^2 (\delta + 1) \sigma_{t, \max} \|C(U^k - U^{k+1})\|^2 + 6\tau_t^2 \|U_t^k - U_t^{k+1}\|^2 \\ &\quad + 2L_t^2 \|A_t^k - A_t^{k+1}\|^2 \\ &\leq 6\rho^2 m (\delta + 1) \sigma_{t, \max} \sum_{j \in \mathcal{V}} \sigma_{j, \max} \|U_j^k - U_j^{k+1}\|^2 \\ &\quad + 6\tau_t^2 \|U_t^k - U_t^{k+1}\|^2 + 2L_t^2 \|A_t^k - A_t^{k+1}\|^2, \end{aligned} \quad (43)$$

where (a) is since the boundedness of $\{U^k, A^k\}$ makes the gradient of $F_t(\cdot, \cdot)$ Lipschitz continuous hold in any bounded set. Then with (41)-(43) we conclude that there exists $c \geq 0$ such that

$$\begin{aligned} & \partial \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1}) \\ & \leq c \sum_{t \in \mathcal{V}} \left(\|U_t^k - U_t^{k+1}\|^2 + \|A_t^k - A_t^{k+1}\|^2 \right), \end{aligned} \quad (44)$$

which completes the proof.

F. Proof of Lemma 6

Recalling the update of U_t^{k+1} in FO-DMTL-RSF, the optimality condition deduces

$$\nabla_{U_t} f_t(U_t^k, A_t^k) + \nabla_{g_1}(U_t^k) = \rho C_t^\top C_t(U_t^k - U_t^{k+1}) - C_t^\top \lambda^k - \rho C_t^\top C U^k + P_t(U_t^k - U_t^{k+1}). \quad (45)$$

Thus with Proposition 2, we have

$$\begin{aligned} & f_t(U_t^k, A_t^k) + g_1(U_t^k) - f_t(U_t^{k+1}, A_t^k) - g_1(U_t^{k+1}) \\ & \geq \langle \nabla_{U_t} f_t(U_t^{k+1}, A_t^k) + \nabla_{g_1}(U_t^{k+1}), U_t^k - U_t^{k+1} \rangle \\ & \quad + \frac{\sigma}{2} \|U_t^k - U_t^{k+1}\|^2 \\ & = \langle \nabla_{U_t} f_t(U_t^k, A_t^k) + \nabla_{g_1}(U_t^k), U_t^k - U_t^{k+1} \rangle + \frac{\sigma}{2} \|U_t^k - U_t^{k+1}\|^2 \\ & \quad + \langle \nabla_{U_t} F_t(U_t^{k+1}, A_t^k) - \nabla_{U_t} F_t(U_t^k, A_t^k), U_t^k - U_t^{k+1} \rangle \\ & \geq \langle \nabla_{U_t} f_t(U_t^k, A_t^k) + \nabla_{g_1}(U_t^k), U_t^k - U_t^{k+1} \rangle + \frac{\sigma}{2} \|U_t^k - U_t^{k+1}\|^2 \\ & \quad - \|\nabla_{U_t} F_t(U_t^{k+1}, A_t^k) - \nabla_{U_t} F_t(U_t^k, A_t^k)\| \|U_t^k - U_t^{k+1}\| \\ & \geq \langle \nabla_{U_t} f_t(U_t^k, A_t^k) + \nabla_{g_1}(U_t^k), U_t^k - U_t^{k+1} \rangle \\ & \quad + \left(\frac{\sigma}{2} - L_t\right) \|U_t^k - U_t^{k+1}\|^2 \\ & = -\langle \lambda^k, C_t(U_t^k - U_t^{k+1}) \rangle - \rho \langle C U^k, C_t(U_t^k - U_t^{k+1}) \rangle \\ & \quad + \left(\tau_t + \frac{\sigma}{2} - L_t\right) \|U_t^k - U_t^{k+1}\|^2. \end{aligned} \quad (46)$$

Following the analysis presented in the Proof of Lemma 2, we can also quantify the change of $\mathcal{L}$ as

$$\begin{aligned} & \mathcal{L}(U^k, A^k, \lambda^k) - \mathcal{L}(U^{k+1}, A^k, \lambda^{k+1}) > \\ & \sum_{t \in \mathcal{V}} \left(\tau_t + \frac{\sigma}{2} - L_t - \rho m \left(\delta + \frac{1}{2} \right) \sigma_{t, \max} \right) \|U_t^k - U_t^{k+1}\|^2. \end{aligned} \quad (47)$$

So to make $\mathcal{L}$ decrease with updating U and λ in FO-DMTL-RSF, we only need to guarantee $\tau_t > L_t + \rho m(\delta + \frac{1}{2})\sigma_{t, \max} - \frac{\sigma}{2}$.

G. Proof of Lemma 7

Comparing FO-DMTL-RSF and DMTL-RSF, the update of λ and A are the same. Hence similar as the proof of Lemma 5, we can show that $\|\nabla_{\lambda} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})\|^2$ and $\|\nabla_{A_t} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})\|^2$ are also bounded by $c \sum_{t \in \mathcal{V}} (\|U_t^k - U_t^{k+1}\|^2 + \|A_t^k - A_t^{k+1}\|^2)$ with $c \geq 0$, where

$$\begin{aligned} & \|\nabla_{U_t} \mathcal{L}(U^{k+1}, A^{k+1}, \lambda^{k+1})\|^2 \\ & = \|\nabla_{U_t} F_t(U_t^{k+1}, A_t^{k+1}) + C_t^\top (\lambda^{k+1} + C U^{k+1})\|^2 \\ & \leq 2 \|\nabla_{U_t} F_t(U_t^{k+1}, A_t^{k+1}) - \nabla_{U_t} F_t(U_t^k, A_t^k)\|^2 \\ & \quad + 2 \|\nabla_{U_t} F_t(U_t^k, A_t^k) + C_t^\top (\lambda^{k+1} + C U^{k+1})\|^2 \\ & \leq 6\rho^2 m(\delta + 1) \sigma_{t, \max} \sum_{j \in \mathcal{V}} \sigma_{j, \max} \|U_j^k - U_j^{k+1}\|^2 \\ & \quad + (6\tau_t^2 + 2L_t^2) \|U_t^k - U_t^{k+1}\|^2 + 2L_t^2 \|A_t^k - A_t^{k+1}\|^2. \end{aligned} \quad (48)$$

This completes the proof.

REFERENCES

- [1] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.
- [2] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2004, pp. 109–117.
- [3] Q. Zhou and Q. Zhao, "Flexible clustered multi-task learning by learning representative tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 266–278, 2016.
- [4] A. Argyriou, T. Evgeniou, and M. Pontil, "Multi-task feature learning," in *Advances in neural information processing systems*, 2007, pp. 41–48.
- [5] —, "Convex multi-task feature learning," *Machine Learning*, vol. 73, no. 3, pp. 243–272, 2008.
- [6] Y. Amit, M. Fink, N. Srebro, and S. Ullman, "Uncovering shared structures in multiclass classification," in *Proceedings of the 24th international conference on Machine learning*. ACM, 2007, pp. 17–24.
- [7] Z. Kang, K. Grauman, and F. Sha, "Learning with whom to share in multi-task feature learning," in *ICML*, 2011, pp. 521–528.
- [8] A. Kumar and H. Daume III, "Learning task grouping and overlap in multi-task learning," *arXiv preprint arXiv:1206.6417*, 2012.
- [9] B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil, "Exploiting unrelated tasks in multi-task learning," in *International Conference on Artificial Intelligence and Statistics*, 2012, pp. 951–959.
- [10] Y. Zhang and D.-Y. Yeung, "A convex formulation for learning task relationships in multi-task learning," *arXiv preprint arXiv:1203.3536*, 2012.
- [11] Y. Li, X. Tian, T. Liu, and D. Tao, "On better exploring and exploiting task relationships in multitask learning: Joint model and feature learning," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 5, pp. 1975–1985, 2018.
- [12] J. Chen, L. Tang, J. Liu, and J. Ye, "A convex formulation for learning a shared predictive structure from multiple tasks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 5, pp. 1025–1038, 2013.
- [13] J. Chen, C. Richard, and A. H. Sayed, "Multitask diffusion adaptation over networks with common latent representations," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 3, pp. 563–579, 2017.
- [14] R. Nassif, C. Richard, A. Ferrari, and A. H. Sayed, "Proximal multitask learning over networks with sparsity-inducing coregularization," *IEEE Transactions on Signal Processing*, vol. 64, no. 23, pp. 6329–6344, 2016.
- [15] J. Li, T. Arai, Y. Baba, H. Kashima, and S. Miwa, "Distributed multi-task learning for sensor network," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2017, pp. 657–672.
- [16] C. Li, S. Huang, Y. Liu, and Z. Zhang, "Distributed jointly sparse multitask learning over networks," *IEEE transactions on cybernetics*, vol. 48, no. 1, pp. 151–164, 2018.
- [17] C. Zhang, P. Zhao, S. Hao, Y. C. Soh, B. S. Lee, C. Miao, and S. C. Hoi, "Distributed multi-task classification: a decentralized online learning approach," *Machine Learning*, vol. 107, no. 4, pp. 727–747, 2018.
- [18] T. Zhang, G. Su, C. Qing, X. Xu, B. Cai, and X. Xing, "Hierarchical lifelong learning by sharing representations and integrating hypothesis," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2019.
- [19] S. Liu, S. J. Pan, and Q. Ho, "Distributed multi-task relationship learning," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2017, pp. 937–946.
- [20] V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated multi-task learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4424–4434.
- [21] W. Wang, J. Wang, M. Kolar, and N. Srebro, "Distributed stochastic multi-task learning with graph regularization," *arXiv preprint arXiv:1802.03830*, 2018.
- [22] J. Wang, M. Kolar, and N. Srebro, "Distributed multi-task learning," in *Artificial Intelligence and Statistics*, 2016, pp. 751–760.
- [23] J. Hua, C. Li, and H.-L. Shen, "Distributed learning of predictive structures from multiple tasks over networks," *IEEE Transactions on Industrial Electronics*, vol. 64, no. 5, pp. 4246–4256, 2017.
- [24] J. Wang, M. Kolar, and N. Srebro, "Distributed multi-task learning with shared representation," *arXiv preprint arXiv:1603.02185*, 2016.
- [25] A. Beck, "On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes," *SIAM Journal on Optimization*, vol. 25, no. 1, pp. 185–209, 2015.

- [26] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [27] J. F. Mota, J. M. Xavier, P. M. Aguiar, and M. Püschel, “D-admm: A communication-efficient distributed algorithm for separable optimization,” *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2718–2723, 2013.
- [28] T.-H. Chang, M. Hong, and X. Wang, “Multi-agent distributed optimization via inexact consensus admm,” *IEEE Transactions on Signal Processing*, vol. 63, no. 2, pp. 482–497, 2014.
- [29] Q. Ling, Y. Liu, W. Shi, and Z. Tian, “Weighted admm for fast decentralized network optimization,” *IEEE Transactions on Signal Processing*, vol. 64, no. 22, pp. 5930–5942, 2016.
- [30] L. Majzoobi, F. Lahouti, and V. Shah-Mansouri, “Analysis of distributed admm algorithm for consensus optimization in presence of node error,” *IEEE Transactions on Signal Processing*, vol. 67, no. 7, pp. 1774–1784, 2019.
- [31] M. Luo, L. Zhang, J. Liu, J. Guo, and Q. Zheng, “Distributed extreme learning machine with alternating direction method of multiplier,” *Neurocomputing*, vol. 261, pp. 164–170, 2017.
- [32] W. Deng, M.-J. Lai, Z. Peng, and W. Yin, “Parallel multi-block admm with $\mathcal{O}(1/k)$ convergence,” *Journal of Scientific Computing*, vol. 71, no. 2, pp. 712–736, May 2017.
- [33] Y. Xu, “Hybrid jacobian and gauss–seidel proximal block coordinate update methods for linearly constrained convex programming,” *SIAM Journal on Optimization*, vol. 28, no. 1, pp. 646–670, 2018.
- [34] X. Gao and S.-Z. Zhang, “First-order algorithms for convex optimization with nonseparable objective and coupled constraints,” *Journal of the Operations Research Society of China*, vol. 5, no. 2, pp. 131–159, Jun 2017.
- [35] Y. Cui, X. Li, D. Sun, and K.-C. Toh, “On the convergence properties of a majorized alternating direction method of multipliers for linearly constrained convex optimization problems with coupled objective functions,” *Journal of Optimization Theory and Applications*, vol. 169, no. 3, pp. 1013–1041, 2016.
- [36] A. Beck and L. Tetruashvili, “On the convergence of block coordinate descent type methods,” *SIAM journal on Optimization*, vol. 23, no. 4, pp. 2037–2060, 2013.
- [37] F. Liu, L. Xu, Y. Sun, and D. Han, “A proximal alternating direction method for multi-block coupled convex optimization,” *Journal of Industrial & Management Optimization*, pp. 1171–1197, 2018.
- [38] G. Banjac, K. Margellos, and P. J. Goulart, “On the convergence of a regularized jacobi algorithm for convex optimization,” *IEEE Transactions on Automatic Control*, vol. 63, no. 4, pp. 1113–1119, April 2018.
- [39] Y. Xu and W. Yin, “A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion,” *SIAM Journal on Imaging Sciences*, vol. 6, no. 3, pp. 1758–1789, 2013.
- [40] Y. Wang, W. Yin, and J. Zeng, “Global convergence of admm in nonconvex nonsmooth optimization,” *Journal of Scientific Computing*, vol. 78, no. 1, pp. 29–63, Jan 2019.
- [41] G.-B. Huang, H. Zhou, X. Ding, and R. Zhang, “Extreme learning machine for regression and multiclass classification,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.
- [42] X. Bi, X. Zhao, G. Wang, P. Zhang, and C. Wang, “Distributed extreme learning machine with kernels based on mapreduce,” *Neurocomputing*, vol. 149, pp. 456–463, 2015.
- [43] D. Wang and M. Li, “Stochastic configuration networks: Fundamentals and algorithms,” *IEEE Transactions on Cybernetics*, vol. 47, no. 10, pp. 3466–3479, Oct 2017.
- [44] B. Igel'nik and Yoh-Han Pao, “Stochastic choice of basis functions in adaptive function approximation and the functional-link net,” *IEEE Transactions on Neural Networks*, vol. 6, no. 6, pp. 1320–1329, Nov 1995.
- [45] K. Yuan, Q. Ling, and W. Yin, “On the convergence of decentralized gradient descent,” *SIAM J. Optim.*, vol. 26, no. 3, pp. 1835–1854, 2016.