Broad Multitask Learning System With Group Sparse Regularization

Jintao Huang, Chuangquan Chen, *Member, IEEE*, Chi-Man Vong, *Senior Member, IEEE*, and Yiu-Ming Cheung, *Fellow, IEEE*

***Abstract*— The broad learning system (BLS) featuring lightweight, incremental extension, and strong generalization capabilities has been successful in its applications. Despite these advantages, BLS struggles in multitask learning (MTL) scenarios with its limited ability to simultaneously unravel multiple complex tasks where existing BLS models cannot adequately capture and leverage essential information across tasks, decreasing their effectiveness and efficacy in MTL scenarios. To address these limitations, we proposed an innovative MTL framework explicitly designed for BLS, named group sparse regularization for broad multitask learning system using related task-wise (BMtLS-RG). This framework combines a task-related BLS learning mechanism with a group sparse optimization strategy, significantly boosting BLS's ability to generalize in MTL environments. The task-related learning component harnesses task correlations to enable shared learning and optimize parameters efficiently. Meanwhile, the group sparse optimization approach helps minimize the effects of irrelevant or noisy data, thus enhancing the robustness and stability of BLS in navigating complex learning scenarios. To address the varied requirements of MTL challenges, we presented two additional variants of BMtLS-RG: BMtLS-RG with sharing parameters of feature mapped nodes (BMtLS-RGf), which integrates a shared feature mapping layer, and BMtLS-RGf and enhanced nodes (BMtLS-RGfe), which further includes an enhanced node layer atop the shared feature mapping structure. These adaptations provide customized solutions tailored to the diverse landscape of MTL problems. We compared BMtLS-RG with state-of-the-art (SOTA) MTL and BLS algorithms through comprehensive experimental evaluation across multiple practical MTL and UCI datasets. BMtLS-RG outperformed SOTA methods in 97.81% of classification tasks and achieved optimal performance in 96.00% of regression tasks, demonstrating its superior accuracy and robustness. Furthermore, BMtLS-RG exhibited satisfactory training efficiency, outperforming existing MTL algorithms by 8.04–42.85 times.**

***Index Terms*— Broad learning system (BLS), group sparse regularization, multitask learning (MTL), task relation.**

Manuscript received 7 August 2023; revised 5 April 2024; accepted 11 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62201402; in part by Shenzhen Science and Technology Innovation Committee under Grant SGDX20220530111001006; in part by NSFC/Research Grants Council (RGC) Joint Research Scheme under Grant N_HKBU214/21; in part by the General Research Fund of RGC under Grant 12201321, Grant 12202622, and Grant 1220323; in part by RGC Senior Research Fellow Scheme under Grant SRFS2324-2S02; in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2023A1515011978; and in part by Hong Kong and Macau Joint Research and Development Fund of Wuyi University under Grant 2021WGALH19. *(Jintao Huang and Chuangquan Chen contributed equally to this work.) (Corresponding authors: Chi-Man Vong; Yiu-Ming Cheung.)*

Jintao Huang and Yiu-Ming Cheung are with the Department of Computer Science, Hong Kong Baptist University, Hong Kong, SAR, China (e-mail: csjthuang@comp.hkbu.edu.hk; ymc@comp.hkbu.edu.hk).

Chuangquan Chen is with the School of Electronics and Information Engineering, Wuyi University, Jiangmen 529020, China (e-mail: chenchuangquan87@163.com).

Chi-Man Vong is with the Department of Computer and Information Science, University of Macau, Macau, SAR, China (e-mail: cmvong@um.edu.mo).

This article has supplementary downloadable material available at https://doi.org/10.1109/TNNLS.2024.3416191, provided by the authors.

Digital Object Identifier 10.1109/TNNLS.2024.3416191

I. Introduction

DEEP neural networks (DNNs) possess remarkable feature extraction and nonlinear approximation capabilities [1]. Nevertheless, they still encounter challenges in practical applications, like vanishing or exploding gradients, convergence to local optima, and slow training efficiency, just to name a few. In particular, the weight updates in DNNs through layer-by-layer gradients during backpropagation are highly time-consuming [2], [3]. To overcome these limitations, the broad learning system (BLS) emerges as an efficient and effective lightweight machine learning approach [4]. It has the potential to revolutionize traditional artificial intelligence methods and finds applications across various research fields in the era of big data [5]. BLS offers notable advantages regarding high accuracy and efficiency when handling ultrahigh-dimensional massive datasets [6].

BLS has increasingly been applicable for practical tasks with its superior capabilities over DNNs, owing to its efficiency and adaptability in handling complex data structures, which is substantiated through various researches with three advantages inherent to the BLS framework: 1) enhanced feature extraction through sparse coding: BLS incorporates sparse coding techniques, facilitating the rapid and precise identification and extraction of critical features from extensive datasets, which underscore the system's proficiency in feature extraction processes [5], [6]; 2) dynamic and incremental updating mechanism: the architecture of BLS is designed to allow the dynamic addition of nodes, thereby enabling efficient model updating and reconstruction without necessitating complete retraining, presenting a flexible adaptation approach for the BLS [7], [8]; and 3) efficient output weight calculation via pseudo-inverse method: by employing the pseudo-inverse method for calculating output weights, BLS offers a high-efficiency solution that addresses and mitigates common issues encountered in neural networks, such as gradient vanishing or explosion [9].

These advantages have cultivated the development of various BLS-based structures and algorithms, finding applications across various domains. Notably, in computer vision, the

introduction of one-class BLSs (OCBLSs) and their stacked variants (STs-OCBLSs) by Yang et al. [10] have achieved significant advancements in industrial intrusion detection scenarios. Additionally, the temporally BLS (TBLS) developed by Sheng et al. [11] enforces temporal consistency in video frame analysis İn the sphere of EEG signal processing for brain–computer interface (BCI) research, the integration of BLS into complex networks is exemplified by Gao et al. 's [12] work on visual evoked potential (VEP)-based BCIs. Similarly, Issa et al.'s [13] application of BLS for user-independent emotion classification through EEG signals demonstrates the system's versatility in bio-signal processing. Furthermore, BLS's application extends to time-series prediction, where Han et al. [14] introduced the maximum information exploitation BLS (MIE-BLS) for chaotic time-series modeling. Yi et al. [15] developed an intergroup cascade BLS for optimized chaotic Time-series prediction. In automatic control, the potential of BLS in predicting the static voltage stability index online has been explored by Yang et al. [16], highlighting its applicability in electrical engineering. Additionally, Yuan et al.'s [17] integration of BLS with online adaptive dynamic programming (ADP) controllers for optimal control of nonlinear systems showcases the system's adaptability and effectiveness in control theory. The flexibility and broad applicability of BLS have been further evidenced through its integration into various specialized learning frameworks, including fuzzy-based BLS [18], [19], class imbalance-based BLS [20], weakly supervised learning [21], [22], [23], semi-supervised learning [24], and online learning [25].

Existing research on BLS has made significant advancements in improving both accuracy and efficiency [26], [27], [28]. Nonetheless, a notable limitation of current BLS-related methods is their reliance on single-task learning (STL) models [29], [30]. In other words, these state-of-the-art (SOTA) BLS-based methods are designed to address individual learning tasks effectively [29], [30]. When faced with complicated learning tasks, BLS can only decompose them into multiple independent single tasks for learning and then combines the results of each task to obtain the outcome. To achieve stable learning results in such scenarios, BLS often requires a substantial amount of labeled data for thorough training. Without sufficient training data, the risk of underfitting arises, leading to diminished learning performance [31]. However, as task complexity and data scale increase, higher demands are placed on the quantity and quality of manually annotated data. Moreover, in practical applications, valuable knowledge and information from other related tasks are often disregarded when each task is learned independently. This results in redundant training efforts and inefficient use of learning resources, ultimately limiting the performance improvement of BLS.

Furthermore, real-life situations often involve multiple-related learning tasks [32]. For example, for a primary face-learning task, followed by four related learning tasks: expression recognition, face recognition, age estimation, and gender classification. Traditional STL methods, including existing BLS, would require separate training for each task using individual datasets, leading to distinct machine-learning models. However, these four learning tasks share common characteristics. For instance, they all pertain to face-related learning tasks, and their datasets contain images of faces. This scenario is prevalent in practical applications, where multiple learning tasks exhibit interconnections and overlap. In summary, the existing BLS methods face challenges such as the requirement for abundant annotated data, the neglect of valuable knowledge from related tasks, and the prevalence of multiple interconnected learning tasks in real-life applications.

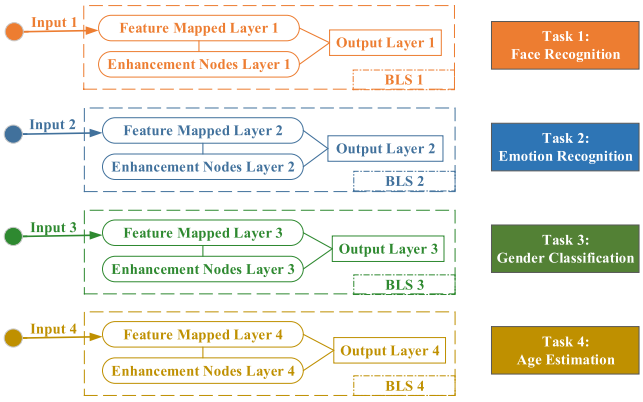


Fig. 1. Example of existing BLS tackling MTL problem.

To address this phenomenon effectively, the research field of multitask learning (MTL) has emerged, garnering extensive attention and exploration [33]. MTL enables the simultaneous processing of multiple-related tasks and leverages the inherent relationships between tasks to enhance the generalization performance of STL. By incorporating inductive bias, MTL improves the model's generalization ability by jointly learning shared information across multiple tasks. This approach has found successful applications in various domains, such as computer vision [34], [35], self-driving [36], [37], speech recognition [38], [39], and medical detection [40], [41]. Given the increasing complexity and scale of scenarios, BLS needs to handle, addressing MTL within the BLS framework becomes crucial and noteworthy. Nevertheless, there has been no previous research on MTL methods designed explicitly for BLS. Therefore, our objective is to pioneer the extension of BLS into MTL. However, applying BLS directly to MTL poses significant challenges and obstacles.

A. Task Overlapping

Multiple tasks may share common features, but variations and differences can also exist among them. Directly applying BLS to MTL may fail to capture the unique characteristics of individual tasks, potentially leading to suboptimal performance. As shown in Fig. 1, the features are extracted through a separate feature mapping function for each independent task, then separately processed through a nonlinear transformation in the enhancement nodes layer.

B. Model Complexity

When dealing with multiple tasks, the complexity of the model increases significantly. With its lightweight and efficient nature, BLS may struggle to accommodate the added complexity, potentially resulting in compromised performance and scalability.

C. Optimization Difficulty

Simultaneously optimizing multiple tasks in the BLS framework poses a non-trivial optimization problem. Jointly optimizing diverse tasks with different objectives, constraints, and data distributions requires innovative strategies for effective and efficient learning.

D. Data Heterogeneity

The datasets associated with different tasks may exhibit heterogeneity in data distribution, feature representation, and label characteristics. Devising techniques that can handle such data heterogeneity effectively within the BLS-based MTL framework is essential. Addressing these challenges is crucial for successfully integrating MTL into the BLS framework and unlocking its potential in tackling complex and diverse real-world scenarios.

To address these challenges and enable the successful integration of MTL into the BLS framework, we propose a new multitask-based BLS called group sparse regularization for broad multitask learning system using related task-wise (BMtLS-RG), which utilizes *task correlation* and *group sparsity techniques* [42]. The distinctive **technical innovations** of BMtLS-RG are as follows.

1) Task Correlation Technique: BMtLS-RG leverages task correlation to effectively facilitate joint learning of multiple tasks within the BLS framework. We introduce a novel weight fusion criterion considering the correlation between different tasks. By considering the interdependencies and relationships between tasks, BMtLS-RG enables sharing parameters and learned information across tasks, improving accuracy and performance in MTL scenarios.

2) Group Sparsity Optimization: BMtLS-RG integrates a group sparsity technology widely used in statistics, signal processing, and machine learning [43], [44], aiming to mitigate the influence of noise data or negative information that may arise when dealing with MTL. We redesign the objective optimization function, incorporating Lasso regularization for combined input nodes (i.e., L_1-norm regularization), task-wise sparse regularization, and group sparse regularization. By effectively suppressing the impact of irrelevant or noisy data, BMtLS-RG enhances the robustness and stability of the system during MTL tasks.

3) Efficiency-Enhancing Variant Structures: To further improve learning efficiency, we proposed two additional variants of BMtLS-RG based on the degree of information sharing in tasks.

1) *BMtLS-RGf (sharing feature mapping layer)*: This variant introduces a sharing feature mapping layer, which allows for the efficient sharing of feature representations across different tasks. By exploiting shared feature mappings, BMtLS-RG with sharing parameters of feature mapped nodes (BMtLS-RGf) enhances the system's adaptability in diverse MTL scenarios.
2) *BMtLS-RGfe (sharing feature mapping layer and enhanced node layer)*: Building upon BMtLS-RGf, this variant incorporates an enhanced node layer that further improves the system's performance. The enhanced node layer leverages optimized parameters to adaptively adjust the learning process and make BMtLS-RGf and enhanced nodes (BMtLS-RGfe) even more effective in addressing different MTL scenarios.

4) Contributions: Our research introduces a novel framework, BMtLS-RG, joining MTL with the BLS to tackle challenges applying BLS to MTL scenarios, like overlapping tasks and complex optimization. The proposed method uses task correlation techniques and group sparsity optimization to advance handling practical problems across various domains significantly. The task correlation enables joint learning by leveraging task interdependencies, which enhances parameter sharing and information across tasks, thereby improving accuracy and performance. Group sparsity optimization, incorporating Lasso regularization, targets irrelevant or noisy data, thereby boosting system robustness and stability in MTL settings. We also present two efficiency-boosting variants, BMtLS-RGf and BMtLS-RGfe, which enhance adaptability and performance through shared feature mapping layers and an enhanced node layer, further contributing to BMtLS-RG's accuracy, stability, and efficiency.

The rest of our article is organized as follows. Section II makes an overview of MTL and BLS. Section III gives the details of the basic proposed method. Section VI gives the different variants of the proposed BMtLS-RG. Experimental results and analysis are given in Section V. Finally, we draw conclusion in Section VI.

II. Preliminaries

A. Multitask Learning

In MTL, we consider a set of T learning tasks represented by the dataset vectors $\{\mathbf{T}_t\}_{t=1}^{T}$, where all tasks or a subset of tasks are related [31], [45]. For the tth task$\mathbf{T}_t$, we have a dataset $\mathbf{D}_t$ consisting of N_t sample-label pairs where N_t is the instance number of tth task, i.e., $\mathbf{D}_t = \{\mathbf{X}^t, \mathbf{Y}^t\}_{t=1}^{T}$, where$\mathbf{X}^t$ denotes the instance data matrix, and $\mathbf{Y}^t$ is the label matrix for each $\boldsymbol{T}_t$. Here, $\boldsymbol{x}_i^t \in \mathbf{X}^t = [\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, \ldots, \boldsymbol{x}_{N_t}^t]$ represents the ith instance of the tth task. In the case where different tasks share the same feature space, meaning that d_i equals d_j for any $i \neq j$ where d_i means the feature space of ith instance, we refer to it as homogeneous-feature MTL. On the other hand, if tasks have different feature spaces, we refer to it as heterogeneous feature MTL. By default, unless otherwise specified, the MTL setting assumes homogeneous-feature MTL.

B. Broad Learning System

The BLS is an improved flat network based on the single-hidden layer feedforward network (SLFN), known for its high efficiency and strong generalization capabilities [8], [42]. It consists of four main layers: the input layer, feature-mapping layer, enhanced nodes layer, and output layer. Notably, the BLS framework allows for obtaining the global optimal solution without the need for dynamic adjustment of the input weights and biases in the network.

Formally, consider a dataset represented as $\mathbf{M} = \{\mathbf{X}, \mathbf{Y}\}$, where $\mathbf{X}$ denotes the input features and $\mathbf{Y}$ represents the corresponding output labels. The dataset comprises n groups

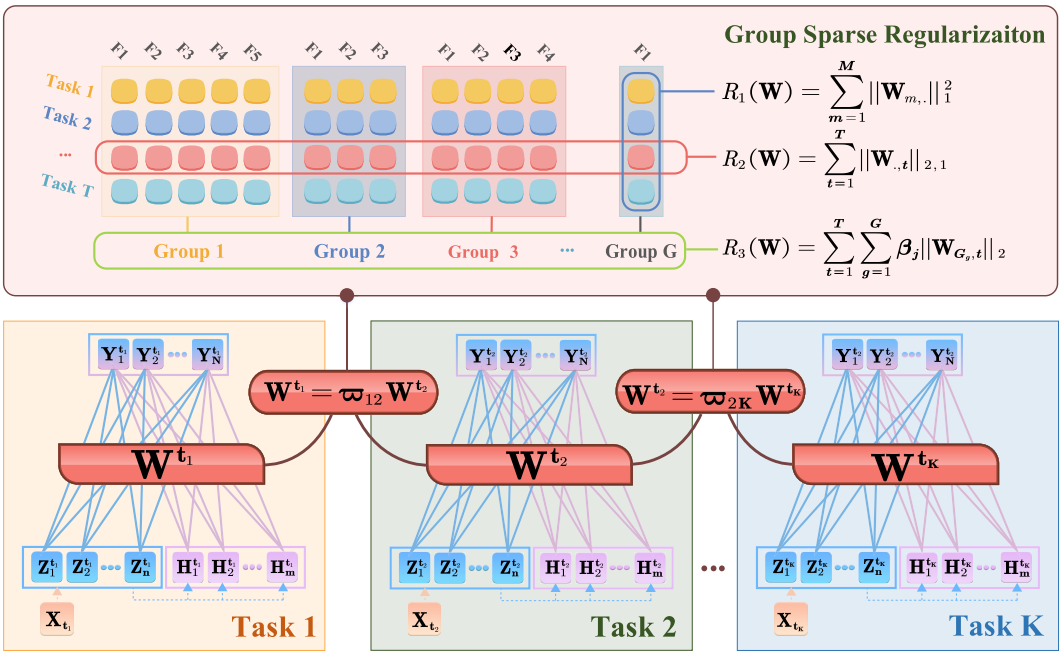


Fig. 2. Our proposed BMtLS-RG framework. There are two key components: 1) task-relation mechanism by leveraging the inter-task correlations in MTL and 2) group sparse regularization, incorporating three group sparse regularization terms. $R_1(\mathrm{W})$ is the input node regularized term, $R_2(\mathrm{W})$ is the task-wise sparse regularized term, and $R_3(\mathrm{W})$ is the group sparse regularized term. F_1–F_5 are the representative feature spaces.

of mapped features. In this context, the ith feature mapping nodes can be mathematically defined as follows:

$$\boldsymbol{Z}_i = \varphi_i\left(\mathbf{X}W_{fi} + \beta_{fi}\right), \ i = 1, 2, \ldots, n \tag{1}$$

where the weights W_{fi} and biases term β_{fi} are randomly generated with the proper dimensions. $\varphi_i(\cdot)$ is usually a linear transformation. Additionally, to obtain better mapped features, W_{fi} is fine-tuned by a sparse autoencoder. The outputs of all groups of mapped features are denoted as

$$\boldsymbol{Z}^n \triangleq (\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n). \tag{2}$$

The output matrix of the jth group of enhancement nodes connected by $\boldsymbol{Z}^n$ is defined as

$$\boldsymbol{H}_j \triangleq \xi_j\left(\boldsymbol{Z}^n W_{ej} + \beta_{ej}\right), \ j = 1, 2, \ldots, m \tag{3}$$

where W_{ej} and β_{ej} are same to (1) which are randomly generated from [0, 1] to connect outputs of mapped features $\boldsymbol{Z}^n$. ξ_j is a nonlinear activation function. Concatenating m enhancement nodes as the output matrix of the enhancement layer: $\boldsymbol{H}^m \triangleq (\boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m)$. Therefore, the output of the BLS $\widehat{\mathbf{Y}}$ is denoted as

$$\begin{aligned}\widehat{\mathbf{Y}} &= [\boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_n | \boldsymbol{H}_1, \boldsymbol{H}_2, \ldots, \boldsymbol{H}_m]\mathbf{W} \\ &= \left[\boldsymbol{Z}^n | \boldsymbol{H}^m\right]\mathbf{W}\end{aligned} \tag{4}$$

where $\mathbf{W}$ is denoted as the final output weight of the BLS.

III. Proposed Method

The BMtLS-RG framework is an innovative approach to integrating MTL with the BLS, addressing the specific challenges of task overlap, model complexity, optimization difficulty, and data diversity inherent in applying BLS to MTL scenarios. BMtLS-RG consists of two key innovations: task correlation techniques and group sparsity optimization. Task correlation harnesses the interconnectedness of multiple tasks to facilitate shared learning and parameter optimization, significantly boosting accuracy and performance across tasks. Meanwhile, group sparsity optimization, incorporating Lasso regularization, targets the reduction of noise and irrelevant data, thereby enhancing the robustness and stability of the learning model. As illustrated in Fig. 2, this framework aims to overcome the challenges mentioned above and enhance the performance of BLS in MTL settings.

A. Broad Multitask Learning System Using Related Task-Wise

Given a multitask dataset $\{\mathbf{T}_t\}_{t=1}^T$ with T tasks, for each task $\boldsymbol{T}_t$, let $\mathbf{X} \in \boldsymbol{R}^{N_t \times d}$ be the input data with d-dimension feature space, and $\mathbf{Y} \in \boldsymbol{R}^{N_t \times 1}$ is the output. As shown in Fig. 3, for each task T_t, the objective function of basic BLS with multiple tasks, namely BMtLS, could be defined as

$$\arg\min_{\mathbf{W}_t}\left(\frac{1}{2}\|\mathbf{Y} - \mathbf{A}_t\mathbf{W}_t\|^2 + \frac{\lambda}{2}\|\mathbf{W}_t\|^2\right) \tag{5}$$

where $\mathbf{W}_t$ is the output weight of m-task, $\mathbf{A} = [\boldsymbol{Z}^n | \boldsymbol{H}^m]$ is the concatenated final input matrix where the dimension is $M = n + m$, and λ is the trade-off parameter. For all the task, we can obtain the model by combined with each objective function through (5) as

$$O_{\mathrm{BMtLS}}(\mathbf{W}) = \arg\min_{\mathbf{W}_t}\left(\sum_{t=1}^{T}\frac{1}{2}\|\mathbf{Y} - \mathbf{A}_t\mathbf{W}_t\|^2 + \frac{\lambda}{2}\|\mathbf{W}_t\|^2\right). \tag{6}$$

As demonstrated in (6), BMtLS is capable of effectively utilizing the shared information from the feature mapping layer and the enhancement node layer across different tasks, thereby enhancing the predictive performance of the model. Nevertheless, the existing BMtLS approach only captures the shared information among all tasks without considering the

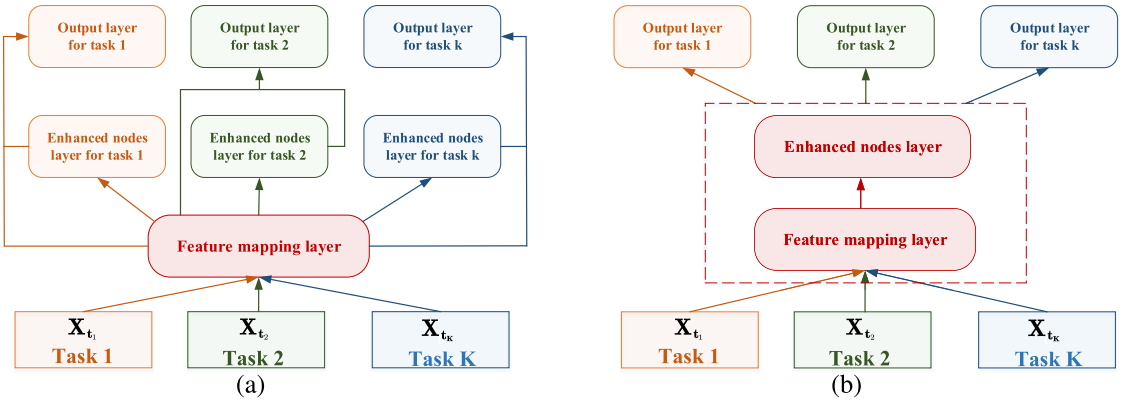


Fig. 3. Two different variants of BMtLS-RG. (a) BMtLS-RGf. (b) BMtLS-RGfe.

varying degrees of association between tasks. Furthermore, if BMtLS employs BLS for MTL, it may encounter issues of overfitting. To address these limitations and uncover the common information within the shared knowledge structure among tasks, we introduce additional constraints to the output weight parameters, representing the relationship between two tasks: $\mathbf{W}_t = \mathbf{W}_0 \varpi_t$, where $\mathbf{W}_0$ is a weight parameter shared by all tasks, and ϖ_t represents task-specific weights for each individual task. Consequently, the refined BMtLS formulation incorporated task relations is defined as

$$O_{\text{BMtLS-R}}(\mathbf{W}) = \arg\min_{\mathbf{W}_t} \left(\sum_{t=1}^{T} \frac{1}{2} \|\mathbf{Y}_t - \mathbf{A}_t \mathbf{W}_t\|^2 + \frac{\lambda}{2} \|\mathbf{W}_t\|^2 \right) \quad \text{s.t. } \mathbf{W}_t = \varpi_{tk} \mathbf{W}_k, k \neq t. \tag{7}$$

B. Group Sparse Regularization for BMtLS-R

To mitigate the issue of overfitting that may arise during MTL in the BLS architecture, which employs Lasso regularization (i.e., L_1-norm regularization), we propose an enhancement to the model by incorporating group sparsity regularization. This regularization approach jointly applies the L_1 and L_2 norms to regularize the model. Group sparsity regularization ensures that the L_2-norm imposes minimal constraints on similarity while simultaneously maintaining the sparse characteristic of the L_1 norm for each task. Consequently, with the integration of group sparsity regularization, the enhanced framework is denoted as BMtLS-RG

$$\begin{aligned} &O_{\text{BMtLS-RG}}(\mathbf{W}) \\ &= \arg\min_{\mathbf{W}_t} \left[\sum_{t=1}^{T} \frac{1}{2} \|\mathbf{Y}_t - \mathbf{A}_t \mathbf{W}_t\|^2 \right. \\ &\qquad \left. + \frac{\lambda_1}{2} \|\mathbf{W}\|_1^2 + \lambda_2 R_2(\mathbf{W}) + \lambda_3 R_3(\mathbf{W}) \right] \\ &\text{s.t.} \mathbf{W}_t = \varpi_{tk} \mathbf{W}_k, k \neq t \end{aligned} \tag{8}$$

where $R_2(\mathbf{W})$ is the task-wise sparse regularized term, and $R_3(\mathbf{W})$ is the group sparse regularized term, which are obtained by

$$R_2(\mathbf{W}) = \|\mathbf{W}\|_{2,1} = \sum_{t=1}^{T} \|\mathbf{W}_{.,t}\|_{2,1} \tag{9}$$

$$R_3(\mathbf{W}) = \|\mathbf{W}_{\mathbf{G}}\|_{2,1} = \sum_{t=1}^{T} \sum_{g=1}^{G} \beta_g \|\mathbf{W}_{G_g,t}\|_{2,1}. \tag{10}$$

Additionally, suppose $R_1(\mathbf{W}) = \sum_{d=1}^{D} \|\mathbf{W}_{d,.}\|_1^2$ is the input node regularized term, where D is the dimension of $\mathbf{A}$. G in (10) denotes the number of feature groups. Accordingly, (8) can be rewritten as

$$O_{\text{BMtLS-RG}}(\mathbf{W}) = \arg\min_{\Theta} \ [\Theta(\mathbf{W}) + \lambda R(\mathbf{W})] \quad \text{s.t. } \mathbf{W}_t = \varpi_{tk} \mathbf{W}_k, k \neq t \tag{11}$$

where

$$\Theta(\mathbf{W}) = \sum_{t=1}^{T} \frac{1}{2} \|\mathbf{Y}_t - \mathbf{A}_t \mathbf{W}_t\|^2 \tag{12}$$

$$\lambda R(\mathbf{W}) = \lambda_1 R_1(\mathbf{W}) + \lambda_2 R_2(\mathbf{W}) + \lambda_3 R_3(\mathbf{W}). \tag{13}$$

Owing to the interdependence introduced by (11), the direct utilization of the Lagrangian function to convert the formulation into an unconstrained optimization problem is not feasible. In line with the principles of alternating direction method of multipliers (ADMMs), we introduce new variables to facilitate the transformation of the aforementioned constraints

$$O_{\text{BMtLS-RG}}(\mathbf{W}) = \arg\min_{\Theta} \ [\Theta(\mathbf{W}) + \lambda R(\mathbf{W})] \quad \text{s.t. } \mathbf{W}_t = \mathbf{Q}_{tk}, \varpi_{tk} \mathbf{W}_k = \mathbf{Q}_{tk}. \tag{14}$$

To address the optimization problem stated in (14), we employ the ADMM in combination with the Lagrangian function. This involves the following steps:

$$\begin{aligned} L(\mathbf{W}_t, \mathbf{S}, \mathbf{P}, \mathbf{Q}_{tk}, \varpi_{tk}) &= \Theta(\mathbf{W}_t) + \lambda R(\mathbf{S}) \\ &+ \text{Tr}\left[\mathbf{P}^\top (\mathbf{W}_t - \mathbf{S})\right] + \frac{\alpha}{2} \|\mathbf{W}_t - \mathbf{S}\|^2 \\ &+ \sum_{t=1}^{T} \sum_{k=1, k\neq t}^{T} [\beta_1 (\mathbf{W}_t - \mathbf{Q}_{tk}) \\ &+ \frac{\alpha}{T} \|\mathbf{W}_t - \mathbf{Q}_{tk}\|^2] \\ &+ \sum_{t=1}^{T} \sum_{k=1, k\neq t}^{T} [\beta_2 (\varpi_{tk} \mathbf{W}_k - \mathbf{Q}_{tk}) \\ &+ \frac{\alpha}{T} \|\mathbf{Q}_{tk} - \varpi_{tk} \mathbf{W}_k\|^2] \end{aligned} \tag{15}$$

where λ, α are the trade-off parameters, and β_1, β_2 are the non-negative Lagrange multipliers. $\mathbf{P}$ is the augmented Lagrangian multiplier, and $\mathbf{S}$ is the slack variables. To obtain the solution of each parameters in (14) of tth task for $(i+1)$th iteration, minimize the Lagrangian loss function in an iterative manner.

C. Optimization of BMtLS-RG

1) Updating $\mathbf{W}_t^{i+1}$:

$$\begin{aligned}&\frac{\partial L(\mathbf{W}_t,\mathbf{S},\mathbf{P},\mathbf{Q}_{tk},\varpi_t)}{\partial \mathbf{W}_t}\\&=\left[\mathbf{A}^\top\mathbf{A}+\alpha I+2\alpha\frac{(T-1)}{T}\right)I\Big]\mathbf{W}_t\\&\quad-\mathbf{A}^\top\mathbf{Y}+P-\alpha\mathbf{S}+\beta_1(T-1)\\&\quad-\frac{2\alpha}{T}\sum_{k=1,k\neq t}^{T}\mathbf{Q}_{tk}\end{aligned}\tag{16}$$

$$\begin{aligned}\mathbf{W}_t^{i+1}&=\left[\mathbf{A}^\top\mathbf{A}+\alpha\left(1+2\frac{(T-1)}{T}\right)I\right]^{-1}\\&\quad\times\Bigg[\mathbf{A}^\top\mathbf{Y}-\mathbf{P}^i+\alpha\mathbf{S}^i\\&\qquad\times-\beta_1^i(T-1)+\frac{2\alpha}{T}\sum_{k=1,k\neq t}^{T}\mathbf{Q}_{tk}^i\Bigg].\end{aligned}\tag{17}$$

2) Updating $\mathbf{S}^{i+1}$:

$$\frac{\partial L(\mathbf{W}_t,\mathbf{S},\mathbf{P},\mathbf{Q}_{tk},\varpi_t)}{\partial \mathbf{S}}=\frac{\partial \lambda R(\mathbf{S})}{\partial \mathbf{S}}-\mathbf{P}+\alpha(\mathbf{W}_t-\mathbf{S})\tag{18}$$

$$\arg\min_{\mathbf{K}}\left[\lambda R(\mathbf{S})+\frac{\alpha}{2}\|\mathbf{S}-\mathbf{K}\|^2\right]\tag{19}$$

where $\mathbf{K}=\mathbf{W}_t+\frac{1}{\alpha}\mathbf{P}$, the goal is to be able to compute K efficiently. It can be shown that the proximal operator for the composite regularizer can be computed efficiently in three steps, and all of these steps can be executed efficiently using suitable extensions of soft-thresholding

$$\begin{cases}\boldsymbol{\Phi}^{i+1}=\arg\min_{\boldsymbol{\Phi}}\left[\frac{\lambda_1}{2}\|\boldsymbol{\Phi}\|_1^2+\frac{\alpha}{2}\|\boldsymbol{\Phi}-\mathbf{K}\|\right]\\\boldsymbol{\Psi}^{i+1}=\arg\min_{\boldsymbol{\Psi}}\left[\frac{\lambda_1}{2}\|\boldsymbol{\Psi}\|_{2,1}^2+\frac{\alpha}{2}\|\boldsymbol{\Psi}-\boldsymbol{\Phi}\|\right]\\\mathbf{S}^{i+1}=\arg\min_{\mathbf{S}}\left[\frac{\lambda_1}{2}\|\mathbf{S}_G\|_{2,1}^2+\frac{\alpha}{2}\|\mathbf{S}-\boldsymbol{\Psi}\|\right]\end{cases}\tag{20}$$

$$\begin{cases}\phi_{i,t}=\frac{\max\{\|k_{i,t}\|_1-\frac{\lambda_1}{\alpha},0\}}{\|k_{i,t}\|_1}k_{i,t}\\\psi_{i.}=\frac{\max\{\|\phi_{i,}\|_2-\frac{\lambda_2}{\alpha},0\}}{\|\phi_{i,}\|_2}\phi_{i,}\\s_{G_g,t}=\frac{\max\{\|\psi_{G_g,t}\|_2-\frac{\lambda_3}{\alpha}\beta_g,0\}}{\|\psi_{G_g,t}\|_2}\psi_{G_g,t}\end{cases}\tag{21}$$

where $\phi_{i.}$, $k_{i.}$, and $\psi_{i.}$ are the ith row of $\boldsymbol{\Phi}^{i+1}$, $\mathbf{K}^{i+1}$, and $\boldsymbol{\Psi}^{i+1}$, $s_{G_g,t}$, $\psi_{G_g,t}$ are rows in group G_g for tth task of $\mathbf{S}^{i+1}$ and $\boldsymbol{\Psi}^{i+1}$, respectively.

3) Updating $\mathbf{Q}_{tk}^{i+1}$:

$$\begin{aligned}&\frac{\partial L(\mathbf{W}_t,\mathbf{S},\mathbf{P},\mathbf{Q}_{tk},\varpi_t)}{\partial \mathbf{Q}_{tk}}\\&=\beta_1-\beta_2-+\frac{2\alpha}{T}\varpi_{tk}\mathbf{W}_k-\frac{4\alpha}{T}\mathbf{Q}_{tk}\\&\Rightarrow\mathbf{Q}_{tk}^{i+1}=\frac{T\left(\beta_2^i-\beta_1^i-\frac{2\alpha}{T}\varpi_{tk}^{i+1}\mathbf{W}_k^{i+1}\right)}{4\alpha}.\end{aligned}\tag{22}$$

4) Updating ϖ_t^{i+1}:

$$\varpi_{tk}^{i+1}=\frac{T\beta_1^i\mathbf{W}_k^{i+1}+2\alpha\left(\mathbf{W}_t^{i+1}\right)^\top\mathbf{Q}_{tk}^{i+1}}{2\alpha\left(\mathbf{W}_t^{i+1}\right)^\top\mathbf{W}_t^{i+1}}.\tag{23}$$

5) Updating $\mathbf{P}^{i+1}$, β_1^{i+1}, β_2^{i+1}:

$$\begin{cases}\mathbf{P}^{i+1}=\mathbf{P}^i+\alpha\left(\mathbf{W}_t^{i+1}-\mathbf{S}^{i+1}\right)\\\beta_1^{i+1}=\beta_1^i+\alpha\left(\mathbf{W}_t^{i+1}-\mathbf{Q}_{tk}^{i+1}\right)\\\beta_2^{i+1}=\beta_2^i+\alpha\left(\mathbf{W}_t^{i+1}-\mathbf{Q}_{tk}^{i+1}\right).\end{cases}\tag{24}$$

Algorithm 1 Broad Multitask Learning System via Related Task and Group Sparse (BMtLS-RG)

Input: A multi-task dataset $\{\mathbf{T}_t\}_{t=1}^T$ with T tasks;
number of feature mapped nodes n;
number of enhancement nodes m;
user-specified coefficients λ, α;
number of iterations $iter$.
Output: *the output weighted matrix* $\mathbf{W}$.

```
Step 1: Initialized procedure:
    Initializing S, P, Q_tk = [0 ⋯ 0; ⋮ ⋱ ⋮; 0 ⋯ 0];
    Initializing ϖ_t, β_1, β_2 = 0, i = 1;
    For t = 1 to T do:
        Step 2: Feature processing procedure:
            Obtaining the Z^n for each task t;
            Obtaining the H^m for each task t;
        Step 3: Updating procedure:
            Repeat:
                Updating the W_t^{i+1} by (18);
                Updating the S^{i+1} by (19);
                Updating the Q_tk^{i+1} by (24);
                Updating the ϖ_t^{i+1} by (26);
                Updating the P^{i+1}, β_1^{i+1}, β_2^{i+1} by (27);
                i = i + 1;
            Until i = iter.
    end For
```

IV. Different Variants of BMtLS-RG

Section III introduces BMtLS-RG by incorporating task pairing association and group sparse mechanism. With BMtLS-RG, the parameters and relevant information of the feature mapping and enhanced node layers are decoupled from specific tasks, which can improve model performance and training efficiency in MTL scenarios. However, sharing information between nodes can further enhance results in certain situations. Thus, in this section, we explore variant models of BMtLS-RG that involve sharing information from the feature mapping layer or both the feature mapping layer and the enhanced node layer. Specifically, we delve into BMtLS-RGf [shown in Fig. 3(a)], which includes a shared feature mapping layer, and BMtLS-RGfe [shown in Fig. 3(b)], which shares both the feature mapping layer and enhanced node layer. It is important to note that the enhanced node layer cannot be shared independently in practical situations due to the need for transformation after the feature mapping nodes.

A. BMtLS-RG With Sharing Parameters of Feature Mapped Nodes

In BMtLS-RGf, the feature mapping node layer is shared among tasks, meaning that the feature mapping layer of all tasks utilizes a unified parameter and structure. This implies that, in BMtLS-RGf, a consistent transformation of the feature mapping layer is applied to all tasks.

Referring to (8), where $\mathbf{A}_t = [\boldsymbol{Z}_t^n | \boldsymbol{H}_t^m]$ represents the concatenated input layer consisting of n feature mapping nodes and m enhanced nodes for each task. In BMtLS-RGf, $\mathbf{A}_t$ can be modified as follows:

$$\forall t \in [1, T], \ \mathbf{A}_t = \left[\boldsymbol{Z}_T^n | \boldsymbol{H}_t^m\right] = \left[\widehat{\mathbf{Z}} | \mathbf{H}_t\right] \tag{25}$$

where $\widehat{\mathbf{Z}}$ represents the shared feature mapping layer for all tasks, while $\mathbf{H}_t$ represents the enhancement nodes layer specific to the tth task. Accordingly, the optimization function of BMtLS-LGf in (11) can be refined as

$$\begin{aligned} O_{\text{BMtLS-RGf}}(\mathbf{W}) = \arg\min_{\Theta} & \left[\sum_{t=1}^{T} \frac{1}{2} \|\mathbf{Y}_t - \left[\widehat{\mathbf{Z}} | \mathbf{H}_t\right] \mathbf{W}_t\|^2 \right. \\ & \left. + \lambda R(\mathbf{W})\right] \\ \text{s.t. } & \mathbf{W}_t = \mathbf{Q}_{tk}, \varpi_{tk} \mathbf{W}_k = \mathbf{Q}_{tk}. \end{aligned} \tag{26}$$

Therefore, the detailed algorithm of BMtLS-LGf is shown in Algorithm 2.

Algorithm 2 BMtLS-RG With Shared Feature Mapped Nodes (BMtLS-RGf)

Input: The same to Algorithm 1
Output: *the output weighted matrix* $\mathbf{W}$.
Step 1: The same to Algorithm 1.
Step 2: Feature processing procedure:
01. Obtaining the $\widehat{\mathbf{Z}}$ for all task $t \in T$;
02. **For** $t = 1$ **to** T **do:**
03. Obtaining the $\boldsymbol{H}^m$ for each task t;
Step 3: The same to Algorithm 1.
04. **end For**

B. BMtLS-RG With Sharing Parameters of Feature Mapped Nodes and Enhanced Nodes

In BMtLS-RGfe, the feature mapping node layer and the enhanced node layer are shared among all tasks, meaning they utilize the same parameters and structures. Consequently, in BMtLS-RGfe, a unified nonlinear transformation is applied to the shared feature mapping layer and the enhanced node layer for all tasks. This unified transformation ensures consistent feature mapping and enhancement processing across tasks. Accordingly, in BMtLS-RGfe, $\mathbf{A}_t$ in (28) can be refined as

$$\forall t \in [1, T], \ \mathbf{A}_t = \left[\boldsymbol{Z}_T^n | \boldsymbol{H}_T^m\right] = \left[\widehat{\mathbf{Z}} | \widehat{\mathbf{H}}\right] \tag{27}$$

where $\widehat{\mathbf{H}}$ is the enhanced node layer for all the tasks.

Therefore, the optimization function of BMtLS-LGfe in (11) can be refined as

$$\begin{aligned} O_{\text{BMtLS-RGfe}}(\mathbf{W}) = \arg\min_{\Theta} & \left[\sum_{t=1}^{T} \frac{1}{2} \|\mathbf{Y}_t - \left[\widehat{\mathbf{Z}} | \widehat{\mathbf{H}}\right] \mathbf{W}_t\|^2 \right. \\ & \left. + \lambda R(\mathbf{W})\right] \\ \text{s.t. } & \mathbf{W}_t = \mathbf{Q}_{tk}, \varpi_{tk} \mathbf{W}_k = \mathbf{Q}_{tk}. \end{aligned} \tag{28}$$

The algorithm of BMtLS-LGfe is displayed in Algorithm 3.

Algorithm 3 BMtLS-RG With Shared Feature Mapped Nodes and Enhanced Nodes (BMtLS-RGfe)

Input: The same to Algorithm 1
Output: *the output weighted matrix* $\mathbf{W}$.
Step 1: The same to Algorithm 1.
Step 2: Feature processing procedure:
01. Obtaining the $\widehat{\mathbf{Z}}$ for all task $t \in T$;
02. Obtaining the $\widehat{\mathbf{H}}$ for each task $t \in T$;
03. **For** $t = 1$ **to** T **do:**
Step 3: The same to Algorithm 1.
04. **end For**

V. Experiments

In this section, we conducted a comprehensive set of experiments on various practical multitask datasets to evaluate the effectiveness of our proposed methods, namely BMtLS-RG, BMtLS-RGf, and BMtLS-RGfe. The experiments were designed to address the following objectives.

A. Comparison With SOTA STL Methods

We compared the performance of BMtLS-RG against other widely used STL methods, including BLS [4], FBLS [18], CFBLS [5], REMBLS [7], and MSBLS [46], using different variants of the BLS.

B. Comparison With SOTA MTL Methods

We compared the performance of BMtLS-RG against other SOTA MTL methods, such as VSTG [47], MTPL [32], ATMTL [48], MWAN [49] and MRN [50].

C. Comparison Among Proposed Methods

We conducted comparisons among our proposed methods, BMtLS-RG, BMtLS-RGf, and BMtLS-RGfe, to evaluate their relative performance.

D. Ablation Experiments

We performed ablation experiments on BMtLS-RG to investigate the impact of different components and settings of the proposed method.

All the experiments were conducted on a computer system equipped with an Intel[1] Core[2] i7-9700K CPU running at

[1]Registered trademark.
[2]Trademarked.

TABLE I
13 BENCHMARK DATASETS FOR MTL

No	Dataset	Instances	Features	Tasks	Type
1	Yeast(Yea.)	1484	8	8	Classification
2	Abalone(Aba.)	4177	8	28	Classification
3	Isolet(Iso.)	7794	617	26	Classification
4	Audit(Aud.)	777	18	2	Classification
5	Avila(Avi.)	20867	10	12	Classification
6	Car	1728	6	4	Classification
7	Landmine(Land.)	20010	9	29	Classification
8	MNIST_USPS(MNI.)	1440	256	10	Classification
9	Machine	209	8	4	Regression
10	CCPP	9568	4	10	Regression
11	UJIndoorLoc	11937	520	15	Regression
12	QSAR-BCF	779	10	5	Regression
13	Sarcos	44484	21	7	Regression

3.60 GHz and 32.00 GB of memory, using the Windows 10 operating system. To evaluate the effectiveness of our proposed BMtLS-RG method, we conducted experiments on a set of ten UCI datasets and three practical multitask datasets. The details of these datasets are provided in Table I. For the six UCI classification datasets, namely Yeast, Abalone, Isolet, Audit, Avila, and Car, we converted them into multitask datasets based on the number of categories present in each dataset. This conversion allows us to explore the performance of BMtLS-RG in an MTL setting for classification tasks. Additionally, we selected four UCI regression datasets, namely Machine, CCPP, UJIndoorLoc, and QSAR-BCF. These datasets provide regression problems, and we included them in our experiments to assess the performance of BMtLS-RG in multitask regression scenarios. Please refer to Table I for more detailed information about these datasets. Due to limitations, only some of the experimental data is displayed. For more experimental data, please refer to the supplementary materials.

E. Comparisons Between STL and MTL Methods

To comprehensively evaluate the effectiveness of our proposed algorithm, we conducted experiments comparing it with four STL algorithms: BLS, FBLS, CFBLS, and MSBLS. Additionally, we selected three SOTA MTL algorithms: MTPL, VSTG, and ATMTL, as representative benchmarks for comparison. In our experiments, we set the number of feature mapping nodes to 200 and the number of enhanced nodes to 500 for BLS, CFBLS, MSBLS, and BMtLS-RG. For FBLS, we used 200 fuzzy subsystems and 500 enhanced nodes. These settings ensure consistency across the compared algorithms. We conducted the experiments on eight classification datasets and five regression datasets. The datasets were split into 80% training and 20% test data. The results of the experiments are presented in Table II for classification tasks and Table III for regression tasks, respectively. These tables comprehensively compare the performance of our proposed algorithm (BMtLS-RG) with the selected STL and MTL algorithms across various datasets.

Table II presents the findings from evaluating four widely recognized metrics for assessing classification algorithms: accuracy, precision, recall, and F1-score. Our introduced algorithm, BMtLS-RG, demonstrates superior performance in 97.81% of instances evaluated. Specifically, within the Landmine and USPS-MNIST datasets, BMtLS-RG surpasses nine competing algorithms across all four metrics. The observed enhancements span from 1.24% to 28.14% for accuracy, 2.99% to 25.96% for precision, 1.46% to 27.17% for recall, and 1.89% to 26.40% for F1-score, underscoring the algorithm's exceptional performance on these datasets. The analysis reveals that MTL approaches significantly outperform STL methods in classification outcomes. Across eight datasets, these findings suggest that singular learning architectures are insufficient for attaining optimal classification accuracy within MTL contexts. Compared to single-task models like BLS, the enhanced BMtLS-RG, as proposed in this study, shows marked improvements in all evaluated scenarios, affirming the efficacy of the proposed multitask model, which leverages group sparse BLS. Moreover, BMtLS-RG outstrips four other MTL methodologies by margins ranging from 2.51% to 10.12%, evidencing its superiority across diversified datasets.

Table III shows the performance of ten comparative methods on five multitask regression datasets, evaluated through metrics such as root-mean-square error (RMSE) and mean absolute percentage error (MAPE). The proposed BMtLS-RG algorithm achieves the optimal results across all five regression datasets. Relative to nine other algorithms, BMtLS-RG exhibits significant improvements in RMSE and MAPE across these datasets. On average, BMtLS-RG's RMSE metrics are enhanced by factors ranging from 4.99 to 26.87 times, and its MAPE metrics by factors ranging from 2.57 to 8.65 times. It is particularly noteworthy that BMtLS-RG outclasses existing multitask learning algorithms on the real multitask regression dataset, Sarcos. However, due to extended training durations within the current experimental setup, the MTPL and MWAN algorithms failed to yield results within a 5-h window, necessitating the exclusion of their RMSE and MAPE metrics. The analysis also provides the average training durations for eight algorithms across the five datasets. It is observed that the five STL models represented by BLS require significantly less training time than their MTL counterparts. The BMtLS-RG model benefits from

TABLE II
CLASSIFICATION PERFORMANCE OF TEN COMPARING ALGORITHMS ON EIGHT DATASETS

Comparing Algorithms	Accuracy(%)								Precision(%)							
	Yea.	Aba.	Iso.	Aud.	Avi.	Car	Land.	MNI.	Yea.	Aba.	Iso.	Aud.	Avi.	Car	Land.	MNI.
BLS	54.27	49.27	49.83	40.37	63.67	62.76	74.22	64.47	42.63	47.33	50.89	49.87	52.28	59.80	59.73	51.45
FBLS	56.95	47.06	52.03	49.44	68.72	64.19	77.11	68.61	45.05	41.12	54.28	51.87	62.75	65.64	64.05	58.28
CFBLS	55.24	50.78	58.47	43.15	66.51	68.40	68.00	71.67	44.53	45.29	55.56	55.87	60.99	61.38	59.72	55.48
MSBLS	53.29	58.98	61.36	52.53	67.33	68.34	72.00	72.39	44.32	47.76	65.30	59.87	63.05	69.21	58.26	61.82
REMBLS	55.37	57.22	58.38	55.46	67.21	69.51	76.10	72.09	45.12	45.66	62.40	61.77	62.15	67.12	59.63	62.11
MTPL	62.93	63.37	73.75	69.55	79.44	84.74	93.05	83.80	58.28	65.42	70.49	74.15	79.79	81.94	73.27	72.81
VSTG	63.24	59.54	77.84	74.25	**84.71**	81.85	94.29	87.92	60.69	63.22	**75.85**	72.55	77.88	80.87	79.65	77.41
ATMTL	69.95	70.82	74.55	77.44	81.32	90.93	95.17	85.89	61.12	67.34	73.46	76.39	80.10	82.38	81.23	74.74
MWAN	64.32	66.47	**78.19**	76.52	82.13	89.88	93.11	88.12	63.77	66.12	73.11	74.39	78.19	81.77	80.55	80.41
BMtLS-RG	**70.08**	**75.17**	76.37	**77.82**	84.54	**91.74**	**96.41**	**92.67**	**65.65**	**70.15**	75.72	**78.91**	**82.29**	**83.67**	**84.22**	**81.94**
Comparing Algorithms	Recall(%)								F1-Score(%)							
	Yea.	Aba.	Iso.	Aud.	Avi.	Car	Land.	MNI.	Yea.	Aba.	Iso.	Aud.	Avi.	Car	Land.	MNI.
BLS	58.43	41.73	46.92	50.00	40.63	63.65	59.76	53.54	44.36	41.80	48.80	48.44	45.72	61.65	59.73	54.48
FBLS	58.03	41.15	50.44	50.00	48.41	51.87	68.39	61.40	49.41	42.13	50.20	49.40	56.45	48.52	66.13	61.79
CFBLS	52.14	43.94	54.38	50.00	44.04	56.09	58.87	64.53	45.24	44.59	54.90	51.14	51.09	53.13	59.04	62.98
MSBLS	57.23	47.73	63.66	50.00	46.47	62.49	57.02	67.05	49.08	47.75	64.45	58.44	55.54	60.79	57.60	57.37
REMBLS	58.22	49.88	62.14	50.00	47.12	69.51	66.30	65.19	51.67	47.77	62.27	55.89	56.64	68.31	62.97	63.65
MTPL	71.94	61.15	70.51	70.11	69.60	78.95	81.48	70.24	58.42	61.22	70.55	72.15	**76.37**	82.82	81.35	71.24
VSTG	73.26	59.35	69.71	72.12	71.46	80.91	80.45	73.15	58.91	60.75	**72.61**	73.55	74.60	82.77	80.53	75.19
ATMTL	74.98	65.25	70.42	**75.62**	70.21	**84.15**	82.73	75.04	60.41	64.75	70.14	74.11	75.12	84.57	81.55	76.12
MWAN	70.32	67.44	71.91	72.25	70.92	80.89	83.11	78.12	67.05	66.78	70.51	73.32	74.55	81.33	81.83	79.27
BMtLS-RG	**75.37**	**70.58**	**74.11**	75.01	**74.48**	82.06	**84.19**	**80.06**	**62.22**	**70.32**	70.12	**75.43**	76.21	**84.68**	**83.44**	**81.45**

TABLE III
REGRESSION PERFORMANCE OF TEN COMPARING ALGORITHMS ON FIVE DATASETS

Datasets	RMSE(↓)									
	BLS	FBLS	CFBLS	MSBLS	REMBLS	MTPL	VSTG	ATMTL	MWAN	BMtLS-RG
Machine	1238.8	146.40	274.19	131.13	128.77	28.850	34.542	18.231	17.207	**10.444**
CCPP	2.3542	2.8569	1.8065	2.2946	2.0506	0.6010	0.4710	0.3527	0.3749	**0.1559**
UJIndoorLoc	134.53	32.851	21.830	22.353	22.095	2.9600	8.1831	4.8649	3.3360	**1.0459**
QSAR-BCF	13.533	5.8588	6.3501	3.5014	3.4258	0.9372	0.5121	0.3388	0.3960	**0.2451**
Sarcos	0.5585	0.4410	0.5091	0.3818	0.3455	-	0.0642	0.0510	-	**0.0476**
Datasets	MAPE(↓)									
	BLS	FBLS	CFBLS	MSBLS	REMBLS	MTPL	VSTG	ATMTL	MWAN	BMtLS-RG
Machine	1283.7	518.17	897.10	617.52	657.31	144.02	153.59	130.93	122.85	**80.368**
CCPP	699.71	401.39	361.98	114.22	208.10	82.121	88.710	50.611	53.814	**32.989**
UJIndoorLoc	1002.2	534.08	461.56	359.45	310.51	196.45	105.22	70.961	104.21	**44.785**
QSAR-BCF	485.58	356.27	291.26	229.64	220.45	81.110	52.772	**21.310**	31.731	22.738
Sarcos	30.808	27.346	29.056	24.591	22.824	-	9.4252	11.421	-	**8.6483**
Datasets	Avg. Training Time (s)									
	BLS	FBLS	CFBLS	MSBLS	REMBLS	MTPL	VSTG	ATMTL	MWAN	BMtLS-RG
Machine	6.3127	16.250	**4.4531**	9.6875	10.071	79.981	95.170	450.29	645.15	8.7844
CCPP	30.500	76.718	26.140	75.234	60.697	496.65	345.56	185.24	826.11	**23.436**
UJIndoorLoc	119.35	251.43	94.546	302.96	188.75	3839.9	2620.5	988.52	4556.6	**56.334**
QSAR-BCF	7.4687	19.343	**6.0625**	11.015	18.554	566.40	195.01	116.57	1452.5	11.975
Sarcos	225.765	233.51	**221.07**	289.21	289.14	>5(h)	2753.2	1015.0	>5(h)	242.00

the BLS's lightweight architecture, enabling more efficient training time while simultaneously achieving superior results.

In summary, the BMtLS-RG model proposed in this article demonstrates superior performance to the original single-task model represented by BLS in both classification and regression tasks. Compared to existing MTL models, BMtLS-RG outperforms SOTA MTL models thanks to the advantages of the adaptive expansion of the BLS width network and the proposed group sparse mechanism.

F. Comparisons Among MTL Methods on Image Datasets

The proposed algorithm is further evaluated on two real multitask image datasets: Office-Home and ImageCLEF. These

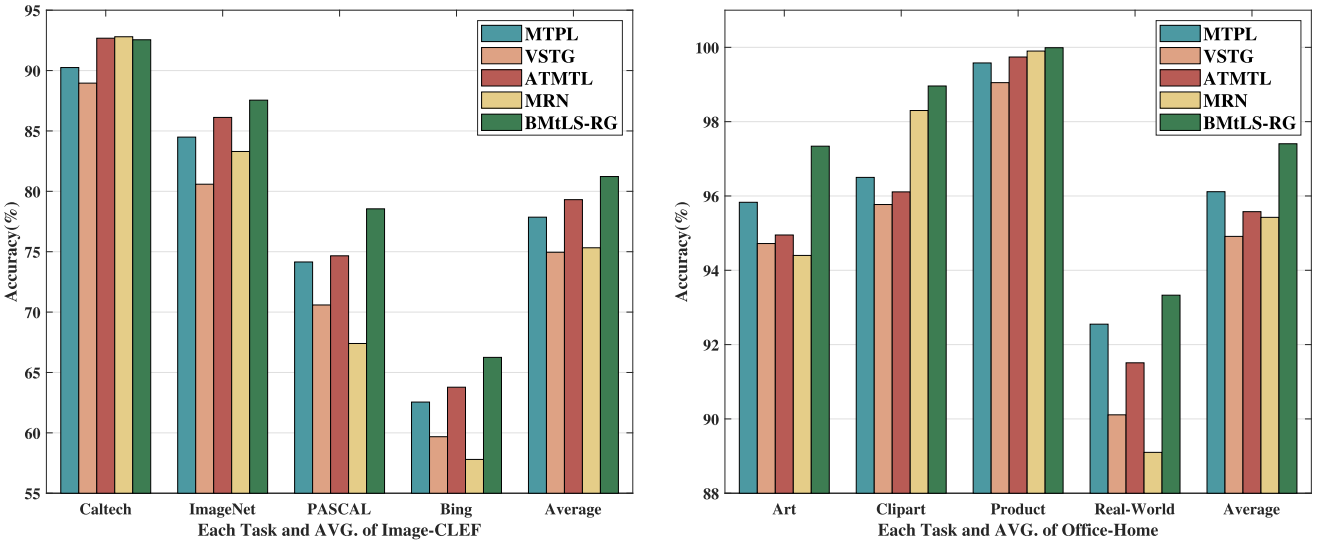


Fig. 4. Classification accuracy of different MTL methods for each task on ImageCLEF and OfficeHome. (a) Accuracy of Image-CLEF (%). (b) Accuracy of Office-Home (%).

datasets are compared with several advanced MTL algorithms to verify the effectiveness of our approach.

Office-Home dataset[3] consists of images from four different domains/tasks: artistic images, clip art, product images, and real-world images. Each task includes images belonging to 65 object categories collected in office and home settings. In total, there are approximately 15 500 images across all tasks. This dataset provides a challenging scenario for multitask learning due to the diverse domains and object categories involved.

ImageCLEF dataset[4] comprises 12 common categories shared by four tasks: Caltech-256, ImageNet ILSVRC 2012, Pascal VOC 2012, and Bing. There are around 2400 images in total across all tasks. This dataset offers another real-world MTL scenario for evaluation.

These two datasets are used to evaluate the proposed algorithm's performance compared to other SOTA MTL algorithms, i.e., MTPL, VSTG, ATMTL, and MRN. The results obtained from these experiments provide further evidence of the effectiveness of the proposed algorithm in real-world multitask image scenarios.

The classification accuracy of these comparisons on the four sub-classification tasks of the ImageCLEF and Office-Home datasets is presented in Fig. 4, which provides insights into the performance of each algorithm on different tasks. The average results of the algorithms are also shown, providing an overview of their overall performance on the dataset. The results shown in Fig. 4 demonstrate the effectiveness of the proposed BMtLS-RG algorithm in handling multitask image classification tasks. On the ImageCLEF dataset, BMtLS-RG achieves competitive performance compared to the other comparison algorithms. While it may not have the highest accuracy on the Caltech classification sub-task, it performs very closely to the top-performing algorithms, ATMTL and MRN. Notably, BMtLS-RG outperforms the other algorithms and achieves the optimal values on the ImageNet, PASCAL, and Bing tasks. Additionally, when considering the average results across all four tasks, BMtLS-RG ranks first, indicating its strong performance across the dataset. On the Office-Home dataset, BMtLS-RG demonstrates superior performance in all four sub-tasks compared to the other four comparison algorithms. This highlights the effectiveness of the BMtLS-RG algorithm in addressing multitask image classification tasks, leveraging the incremental adaptive structure of BLS combined with the proposed group sparse framework.

G. Comparisons Among Our Proposed Methods (BMtLS-RG, BMtLS-RGf, and BMtLS-RGfe) With Different Feature Nodes and Enhancement Nodes

The experimental comparisons conducted on the Car dataset for classification aim to investigate and compare the differences and robustness of the three algorithms: BMtLS-RG, BMtLS-RGf, and BMtLS-RGfe. Several evaluation indicators, including accuracy, F1, and training time, are used to evaluate the performance of the algorithms. Two sets of experiments are designed.

1) *Varying the number of feature mapping nodes*: In this experiment, the number of fixed enhancement nodes is set to 200, and the number of feature mapping nodes is increased. The feature mapping group is either 5 or 10, and each group's number of feature nodes varies from $[10, 50]$. The experimental results are presented in Fig. 5, showing the performance trends of the three algorithms on the Car dataset.
2) *Varying the number of enhanced nodes*: In this experiment, the number of fixed feature nodes is set to 200, and the number of enhanced nodes is increased. The number of enhanced nodes ranges from $[100, 1000]$, with an increment of 100. The results of this experiment are depicted in Fig. 6, illustrating the changes in performance for the three algorithms on the Car.

[3]http://hemanthdv.org/OfficeHome-Dataset
[4]http://imageclef.org/2014/adaptation

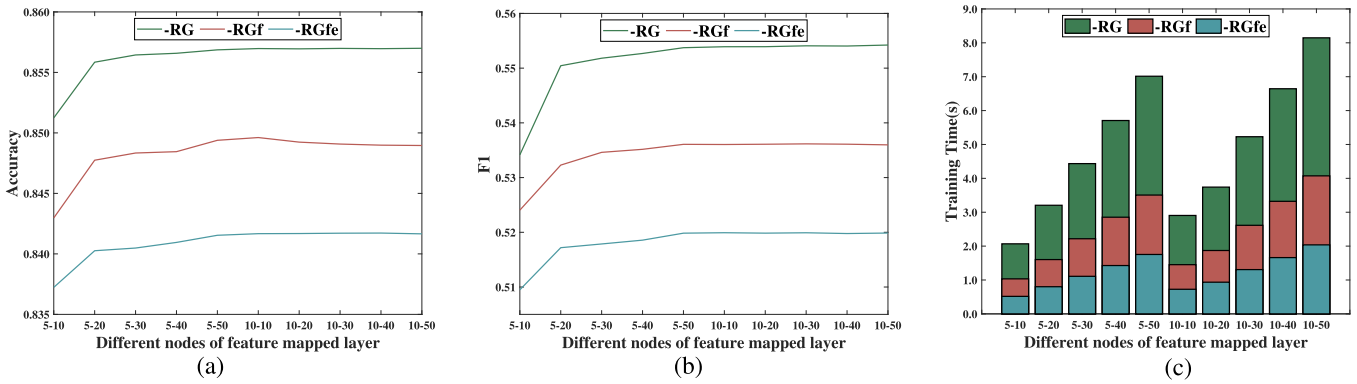


Fig. 5. Classification performance of different nodes of feature mapped layer under fixed enhanced nodes of our proposed methods on Car (fixed enhanced nodes: 200; groups of windows: {5, 10}; feature nodes of each window: {10, 20, . . . , 50}). (a) Accuracy. (b) F1. (c) Training time (s).

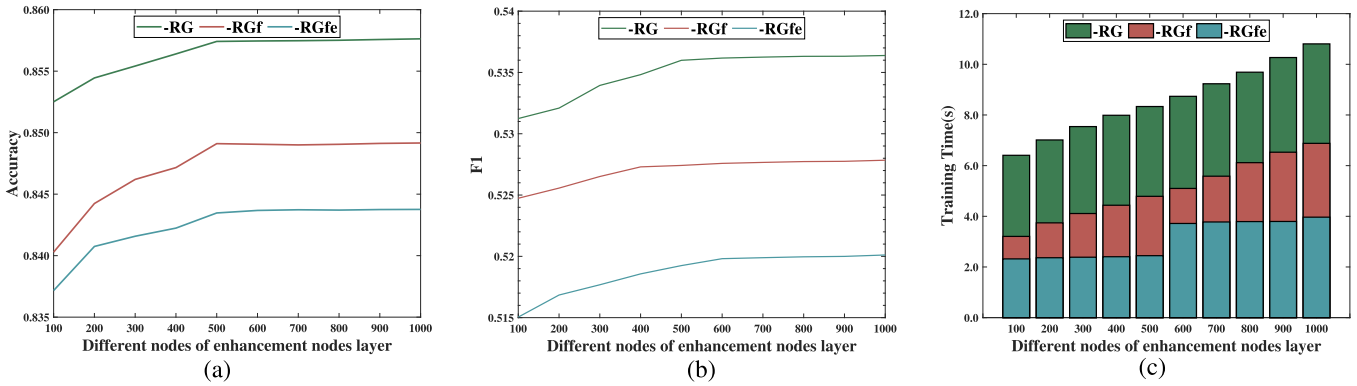


Fig. 6. Classification performance of different nodes of enhanced nodes layer under fixed feature mapped nodes of our proposed methods on Car (fixed groups of windows: 10; fixed feature nodes of each window: 20; enhanced nodes: {100, 200, . . . , 1000}). (a) Accuracy. (b) F1. (c) Training time (s).

Fig. 6 illustrates the trends of the three algorithms (BMtLS-RG, BMtLS-RGf, and BMtLS-RGfe) on the Car with varying feature mapping nodes. As the number of feature mapping nodes increases, the algorithms' performance gradually stabilizes. In terms of accuracy and F1-score, all three algorithms show a notable upward trend as the number of feature mapping nodes increases from 5 to 10 and then stabilizes. In the MTL scenario, the proposed method's accuracy and F1 score improve by about 0.6%–1% when the number of enhanced node layers increases from 100 to 500. In terms of training time, it can be observed that BMtLS-RGf and BMtLS-RGfe exhibit higher training efficiency compared to BMtLS-RG due to the parameter sharing between the feature mapping layer and the enhanced node layer. Fig. 6 illustrates the trends of the three algorithms with varying numbers of enhanced nodes. On the Car, all three algorithms exhibit an upward trend in classification performance as the number of enhanced nodes increases from 100 to 500. As the number of enhanced nodes increases, the classification performance stabilizes. These results further validate the effectiveness and stability of the proposed algorithm. However, it is essential to note that the performance trends may vary depending on the specific dataset and task. While the training time of BMtLS-RG increases with the number of enhanced nodes, its RMSE and MAPE are significantly better than those of BMtLS-RGf and BMtLS-RGfe.

To summarize, achieving relatively better results can be accomplished with 50–100 feature nodes for selecting the number of feature mapping nodes. 500–600 can lead to more effective performance for the number of enhanced nodes. Regarding parameter sharing, if the goal is to achieve optimal accuracy, BMtLS-RG (without sharing any layer) is recommended. If there is a need to balance efficiency and accuracy, BMtLS-RGf (sharing the feature mapping layer) can be chosen. Finally, BMtLS-RGfe (sharing both the feature mapping layer and enhanced node layer) achieves optimal efficiency.

H. Ablations Experiments and Robustness of BMtLS-RG

To investigate the robustness of the group sparse structure in the proposed BMtLS-RG algorithm, ablation experiments and parameter sensitivity analysis are conducted on the Land-Mine dataset (classification task) and the Sarcos dataset (regression task). These experiments aimed to assess the contribution and impact of different components or modules of the algorithm and examine the sensitivity of the algorithm to various parameter settings. In the ablation experiments, specific components or modules of the BMtLS-RG algorithm were temporarily removed or modified, and the performance of the modified algorithm was evaluated. This analysis helped determine the importance and effectiveness of the group sparse structure in improving the algorithm's performance. By comparing the results of the modified algorithms with the original BMtLS-RG, insights could be gained into the specific contributions of the group sparse framework. Additionally, parameter sensitivity analysis was performed to examine the effects of different parameter settings on the algorithm's performance. Parameters such as the regularization strength, number of feature mapping nodes, and number of enhanced

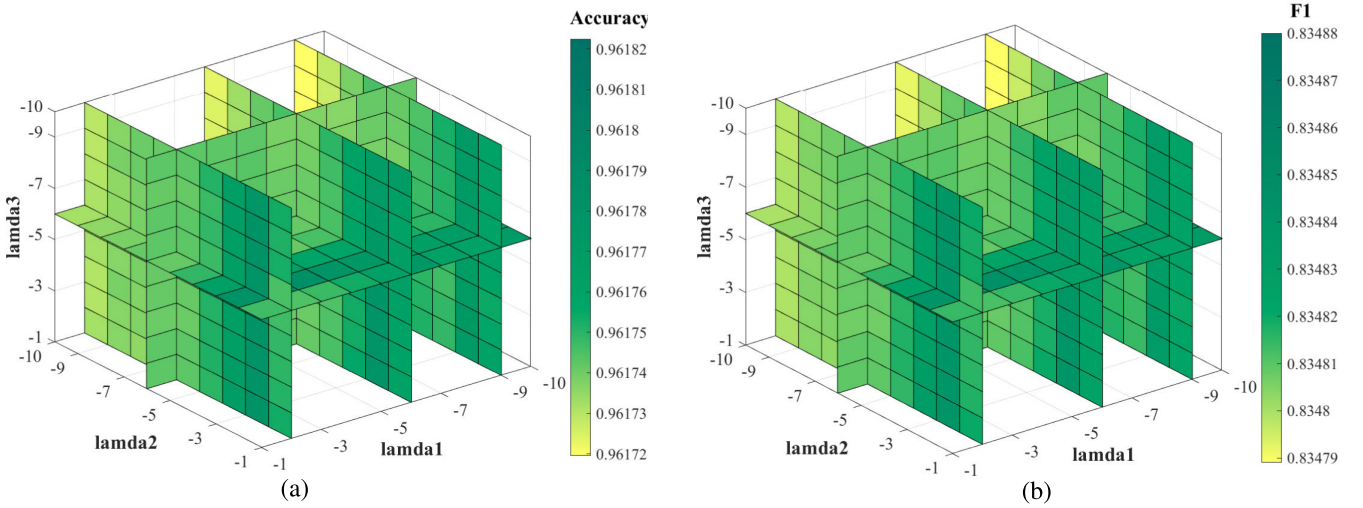


Fig. 7. Classification performance of three parameters for BMtLS-RG on Landmine (lamda1: $[2^{-1}, 2^{-2}, \ldots, 2^{-10}]$; lamda2: $[2^{-1}, 2^{-2}, \ldots, 2^{-10}]$; lamda3: $[2^{-1}, 2^{-2}, \ldots, 2^{-10}]$). (a) Accuracy(↑). (b) F1(↑).

TABLE IV
ABLATION CLASSIFICATION RESULTS OF THREE GROUP REGULARIZED TERMS FOR BMTLS-RG ON LANDMINE

$[\lambda_1, \lambda_2, \lambda_3]$	Accuracy	Precision	Recall	F1
$[0.0, 0.0, 0.0]$	0.7654	0.6528	0.6832	0.6891
$[\mathbf{0.5}, 0.0, 0.0]$	0.8257	0.6955	0.7249	0.7214
$[0.0, \mathbf{0.5}, 0.0]$	0.8556	0.7060	0.7546	0.7713
$[0.0, 0.0, \mathbf{0.5}]$	0.8356	0.7285	0.7347	0.7524
$[\mathbf{0.5}, \mathbf{0.5}, 0.0]$	0.8857	0.7511	0.7808	0.8014
$[\mathbf{0.5}, 0.0, \mathbf{0.5}]$	0.9057	0.7655	0.8049	0.8114
$[0.0, \mathbf{0.5}, \mathbf{0.5}]$	0.9256	0.8050	0.8246	0.8210
$[\mathbf{0.5}, \mathbf{0.5}, \mathbf{0.5}]$	**0.9622**	**0.8454**	**0.8449**	**0.8373**

TABLE V
ABLATION REGRESSION RESULTS OF THREE GROUP REGULARIZED TERMS FOR BMTLS-RG ON SARCOS

$[\lambda_1, \lambda_2, \lambda_3]$	RMSE	MAPE
$[0.0, 0.0, 0.0]$	0.1955	18.705
$[\mathbf{0.5}, 0.0, 0.0]$	0.1331	14.688
$[0.0, \mathbf{0.5}, 0.0]$	0.1172	13.545
$[0.0, 0.0, \mathbf{0.5}]$	0.1031	11.669
$[\mathbf{0.5}, \mathbf{0.5}, 0.0]$	0.0814	9.6857
$[\mathbf{0.5}, 0.0, \mathbf{0.5}]$	0.0937	9.8679
$[0.0, \mathbf{0.5}, \mathbf{0.5}]$	0.0711	9.6547
$[\mathbf{0.5}, \mathbf{0.5}, \mathbf{0.5}]$	**0.0471**	**8.6056**

nodes were systematically varied. The resulting performance metrics were measured and compared, such as accuracy, F1-score, RMSE, and MAPE. This analysis aimed to identify the optimal parameter settings for achieving the best performance on the LandMine and Sarcos datasets.

1) Ablations of Group Regularized Terms: A series of ablation experiments were conducted on the three group sparse modules specified in (13). Expressly, the regularization parameters λ_1, λ_2, and λ_3 were varied between 0 and 0.5. For instance, λ_1 was set to 0 while λ_2 and λ_3 were set to 0.5, denoted the BMtLS-RG algorithm without the Lasso regularization terms. Similar variations were applied to explore other scenarios. The outcomes of these experiments are presented in Tables IV and V.

Table IV presents the classification results obtained from different ablation combinations of BMtLS-RG on the LandMine dataset. It is obvious that when all group sparse terms are removed (i.e., $\lambda_1 = \lambda_2 = \lambda_3 = 0$), the performance across all four indicators is significantly compromised. Comparing these results to the scenario where all group sparse terms are included ($\lambda_1 = \lambda_2 = \lambda_3 = 0.5$), the differences in the four indicators amount to 19.68%, 19.26%, 16.17%, and 14.82%, respectively. When only one group sparse term is incorporated, λ_2 yields the optimal results. Furthermore, when combining any two group sparse terms, the combination of λ_2 and λ_3 emerges as the most effective. This suggests that the inclusion of all three group sparse terms, particularly λ_2, has a substantial impact on the classification results. Turning to Table V, which corresponds to the Sarcos dataset, it becomes evident that the advantages of BMtLS-RG are maximized when all group sparse terms are incorporated. Among the three group sparse terms, λ_3 plays a relatively significant role in the context of the regression task. Overall, the proposed BMtLS-RG algorithm, leveraging group sparse techniques, demonstrates superior performance in both classification and regression tasks.

2) Parameters Analysis of Group Regularized Terms (λ_1, λ_2, λ_3): To analyze the impact of the three group sparse parameters, λ_1, λ_2, and λ_3, we conducted a comparative analysis using ten sets of parameter values ranging from 2^{-1} to 2^{-10}. The results are visualized in Fig. 7. In the two subfigures, the x, y, and z coordinates represent the values of λ_1, λ_2, and λ_3, respectively. Meanwhile, the color map on the right depicts different evaluation metric values. In Fig. 7, for accuracy and F1, the regions closer to green represent higher values and better results. Analyzing Fig. 7, it can be observed that BMtLS-RG achieves relatively better results when λ_1 ranges from 2^{-5} to 2^{-7}, λ_2 ranges from 2^{-1} to 2^{-4}, and λ_3 ranges from 2^{-1} to 2^{-5}. Conversely, poorer results are obtained when λ_1 ranges from 2^{-9} to 2^{-10}, λ_2 ranges from 2^{-9} to 2^{-10}, and λ_3 ranges from 2^{-9} to 2^{-10}. These findings demonstrate the sensitivity of BMtLS-RG to the values of

the group sparse parameters. Selecting appropriate parameter values within the specified ranges can improve classification and regression results.

VI. Conclusion

This article has introduced a novel BLS for multitask learning called BMtLS-RG, which presents improved generality and generalization capabilities. The proposed system incorporates a task relation mechanism and a group sparse optimization mechanism to enhance the accuracy, robustness, and stability of MTL. The task relation mechanism enables effective learning of relevant information between tasks, allowing BMtLS-RG to handle multiple tasks simultaneously and improve the accuracy of MTL. It facilitates the extraction of shared knowledge and relationships among tasks, enhancing overall performance. The group sparse optimization mechanism consists of various modules, including L_1-norm regularization, task-wise sparse regularization, and group sparse regularization. This mechanism maximizes the utilization of practical information between tasks while reducing the impact of noise data from one task to another. It enhances the robustness and stability of BMtLS-RG, making it more resilient to challenging real-world scenarios. Additionally, to address practical applications with higher efficiency, two variants of BMtLS-RG have been introduced, i.e., BMtLS-RGf and BMtLS-RGfe. These variants have provided a flexibility in handling complex and diverse multitask learning scenarios, meeting the demands of high-performance applications.

Extensive experiments have been conducted to evaluate the performance of BMtLS-RG. Comparative analyses have been carried out on various datasets, including UCI and real MTL datasets, for classification and regression tasks. The results consistently demonstrate the superiority of BMtLS-RG over existing BLS STL models and SOTA MTL algorithms in terms of accuracy and training time. Furthermore, the effectiveness, stability, and robustness of BMtLS-RG are verified through ablation experiments and parameter sensitivity analysis. These experiments illustrate the significance of the group sparse terms in achieving optimal performance in classification and regression tasks. Future research directions further explore the application of BMtLS-RG in diverse MTL domains, such as autonomous driving and medical image analysis, to enhance its versatility and utility in real-world scenarios.

References

[1] A. Bouguettaya, H. Zarzour, A. Kechida, and A. M. Taberkit, "Vehicle detection from UAV imagery with deep learning: A review," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6047–6067, Nov. 2022.

[2] S. Oprea et al., "A review on deep learning techniques for video prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 2806–2826, Jun. 2022.

[3] D. Sánchez, L. Servadei, G. N. Kiprit, R. Wille, and W. Ecker, "A comprehensive survey on electronic design automation and graph neural networks: Theory and applications," *ACM Trans. Design Autom. Electron. Syst.*, vol. 28, no. 2, pp. 1–27, Mar. 2023.

[4] C. L. P. Chen and Z. Liu, "Broad learning system: An effective and efficient incremental learning system without the need for deep architecture," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 1, pp. 10–24, Jan. 2018.

[5] L. Zhang et al., "Analysis and variants of broad learning system," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 334–344, Jan. 2022.

[6] X. Gong, T. Zhang, C. L. P. Chen, and Z. Liu, "Research review for broad learning system: Algorithms, theory, and applications," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8922–8950, Sep. 2022.

[7] J. Jin, B. Geng, Y. Li, J. Liang, Y. Xiao, and C. L. P. Chen, "Flexible label-induced manifold broad learning system for multiclass recognition," *IEEE Trans. Neural Netw. Learn. Syst*.

[8] J. Jin, Y. Li, and C. L. P. Chen, "Pattern classification with corrupted labeling via robust broad learning system," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4959–4971, Oct. 2022.

[9] M. Liu, X. Chen, M. Shang, and H. Li, "A pseudoinversion-free method for weight updating in broad learning system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 2, pp. 2378–2389, Feb. 2024.

[10] K. Yang, Y. Shi, Z. Yu, Q. Yang, A. K. Sangaiah, and H. Zeng, "Stacked one-class broad learning system for intrusion detection in industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 251–260, Jan. 2023.

[11] B. Sheng, P. Li, R. Ali, and C. L. P. Chen, "Improving video temporal consistency via broad learning system," *IEEE Trans. Cybern.*, vol. 52, no. 7, pp. 6662–6675, Jul. 2022.

[12] Z. Gao, W. Dang, M. Liu, W. Guo, K. Ma, and G. Chen, "Classification of EEG signals on VEP-based BCI systems with broad learning," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 11, pp. 7143–7151, Nov. 2021.

[13] S. Issa, Q. Peng, and X. You, "Emotion classification using EEG brain signals and the broad learning system," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 12, pp. 7382–7391, Dec. 2021.

[14] M. Han, W. Li, S. Feng, T. Qiu, and C. Chen, "Maximum information exploitation using broad learning system for large-scale chaotic time-series prediction," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 6, pp. 2320–2329, Jun. 2021.

[15] J. Yi, J. Huang, W. Zhou, G. Chen, and M. Zhao, "Intergroup cascade broad learning system with optimized parameters for chaotic time series prediction," *IEEE Trans. Artif. Intell.*, vol. 3, no. 5, pp. 709–721, Oct. 2022.

[16] Y. Yang, Q. Huang, and P. Li, "Online prediction and correction control of static voltage stability index based on broad learning system," *Exp. Syst. Appl.*, vol. 199, Aug. 2022, Art. no. 117184.

[17] L. Yuan, T. Li, S. Tong, Y. Xiao, and Q. Shan, "Broad learning system approximation-based adaptive optimal control for unknown discrete-time nonlinear systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 8, pp. 5028–5038, Aug. 2022.

[18] S. Feng, C. L. P. Chen, L. Xu, and Z. Liu, "On the accuracy–complexity tradeoff of fuzzy broad learning system," *IEEE Trans. Fuzzy Syst.*, vol. 29, no. 10, pp. 2963–2974, Oct. 2021.

[19] S. Feng and C. L. P. Chen, "Fuzzy broad learning system: A novel neuro-fuzzy model for regression and classification," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 414–424, Feb. 2020.

[20] W. Chen, K. Yang, Z. Yu, and W. Zhang, "Double-kernel based class-specific broad learning system for multiclass imbalance learning," *Knowl.-Based Syst.*, vol. 253, Oct. 2022, Art. no. 109535.

[21] J. Huang, C.-M. Vong, G. Wang, W. Qian, Y. Zhou, and C. L. P. Chen, "Joint label enhancement and label distribution learning via stacked graph regularization-based polynomial fuzzy broad learning system," *IEEE Trans. Fuzzy Syst.*, vol. 31, no. 9, pp. 3290–3304, Sep. 2023.

[22] J. Huang, C.-M. Vong, C. L. P. Chen, and Y. Zhou, "Accurate and efficient large-scale multi-label learning with reduced feature broad learning system using label correlation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 12, pp. 10240–10253, Dec. 2023.

[23] W. Qian, Y. Tu, J. Huang, W. Shu, and Y.-M. Cheung, "Partial multilabel learning using noise-tolerant broad learning system with label enhancement and dimensionality reduction," *IEEE Trans. Neural Netw. Learn. Syst.*, 2024, doi: 10.1109/TNNLS.2024.3352285.

[24] T. Qiu, X. Liu, X. Zhou, W. Qu, Z. Ning, and C. L. P. Chen, "An adaptive social spammer detection model with semi-supervised broad learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 10, pp. 4622–4635, Oct. 2022.

[25] Z. Liu, Y. Zhang, Z. Ding, and X. He, "An online active broad learning approach for real-time safety assessment of dynamic systems in nonstationary environments," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 10, pp. 6714–6724, Oct. 2023.

[26] G.-Y. Chen, M. Gan, C. L. P. Chen, H.-T. Zhu, and L. Chen, "Frequency principle in broad learning system," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6983–6989, Nov. 2022.

[27] J. Jin, Z. Qin, D. Yu, Y. Li, J. Liang, and C. L. P. Chen, "Regularized discriminative broad learning system for image classification," *Knowl.-Based Syst.*, vol. 251, Sep. 2022, Art. no. 109306.

[28] J. Jin, Y. Li, T. Yang, L. Zhao, J. Duan, and C. L. P. Chen, "Discriminative group-sparsity constrained broad learning system for visual recognition," *Inf. Sci.*, vol. 576, pp. 800–818, Oct. 2021.

[29] F. Aleotti, M. Poggi, F. Tosi, and S. Mattoccia, "Learning end-to-end scene flow by distilling single tasks knowledge," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, 2020, pp. 10435–10442.

[30] J. Wang, W. Wang, S. Ren, W. Shi, and Z.-G. Hou, "Neural correlates of single-task versus cognitive-motor dual-task training," *IEEE Trans. Cognit. Develop. Syst.*, vol. 14, no. 2, pp. 532–540, Jun. 2022.

[31] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 12, pp. 5586–5609, Dec. 2022.

[32] J. Wang and L. Sun, "Multi-task personalized learning with sparse network lasso," in *Proc. 31st Int. Joint Conf. Artif. Intell.*, Jul. 2022, pp. 3516–3522.

[33] R. Mao, R. Cui, and C. L. P. Philip, "Broad learning with reinforcement learning signal feedback: Theory and applications," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 7, pp. 2952–2964, Jul. 2022.

[34] K. Doshi and Y. Yilmaz, "Multi-task learning for video surveillance with limited data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 3888–3898.

[35] Q. Ma, J. Jiang, X. Liu, and J. Ma, "Multi-task interaction learning for spatiospectral image super-resolution," *IEEE Trans. Image Process.*, vol. 31, pp. 2950–2961, 2022.

[36] T.-J. Song, J. Jeong, and J.-H. Kim, "End-to-end real-time obstacle detection network for safe self-driving via multi-task learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16318–16329, Sep. 2022.

[37] C. Zhang, F. Zhu, X. Wang, L. Sun, H. Tang, and Y. Lv, "Taxi demand prediction using parallel multi-task learning model," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 794–803, Feb. 2022.

[38] A. B. Vasudevan, D. Dai, and L. Van Gool, "Sound and visual representation learning with multiple pretraining tasks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14596–14606.

[39] B. Yang, L. Wu, J. Zhu, B. Shao, X. Lin, and T.-Y. Liu, "Multimodal sentiment analysis with two-phase multi-task learning," *IEEE/ACM Trans. Audio, Speech, Lang., Process.*, vol. 30, pp. 2015–2024, 2022.

[40] J. Cheng, J. Liu, H. Kuang, and J. Wang, "A fully automated multimodal MRI-based multi-task learning for glioma segmentation and IDH genotyping," *IEEE Trans. Med. Imag.*, vol. 41, no. 6, pp. 1520–1532, Jun. 2022.

[41] B. Zhou, X. Cai, Y. Zhang, W. Guo, and X. Yuan, "MTAAL: Multi-task adversarial active learning for medical named entity recognition and normalization," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2021, pp. 14586–14593.

[42] S. Bengio, F. C. N. Pereira, Y. Singer, and D. Strelow, "Group sparse coding," in *Proc. Adv. Neural Inf. Process. Syst.* Vancouver, BC, Canada: Curran Associates, 2009, pp. 82–89.

[43] L. Jiang and W. Zhu, "Iterative weighted group thresholding method for group sparse recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 63–76, Jan. 2021.

[44] C. Zhang, H. Li, C. Chen, Y. Qian, and X. Zhou, "Enhanced group sparse regularized nonconvex regression for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 5, pp. 2438–2452, May 2022.

[45] G. Panagopoulos, F. D. Malliaros, and M. Vazirgiannis, "Multi-task learning for influence estimation and maximization," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 9, pp. 4398–4409, Sep. 2022.

[46] X.-K. Cao, C.-D. Wang, J.-H. Lai, Q. Huang, and C. L. P. Chen, "Multiparty secure broad learning system for privacy preserving," *IEEE Trans. Cybern.*, vol. 53, no. 10, pp. 6636–6648, Oct. 2023.

[47] J.-Y. Jeong and C.-H. Jun, "Variable selection and task grouping for multi-task learning," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 1589–1598.

[48] M. Zhou, Y. Zhang, T. Liu, Y. Yang, and P. Yang, "Multi-task learning with adaptive global temporal structure for predicting Alzheimer's disease progression," in *Proc. 31st ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2022, pp. 2743–2752.

[49] C. Lian, M. Liu, L. Wang, and D. Shen, "Multi-task weakly-supervised attention network for dementia status estimation with structural MRI," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 4056–4068, Aug. 2022.

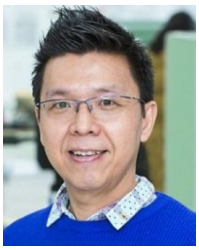
[50] M. Long, Z. Cao, J. Wang, and P. S. Yu, "Learning multiple tasks with multilinear relationship networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1594–1603.

Jintao Huang received the Ph.D. degree in computer science from the Department of Computer and Information Science, University of Macau, Macau, China, in 2023.

He is currently a Post-Doctoral Research Fellow at the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His research interests include machine learning, weak-supervised learning, and knowledge discovery.

Chuangquan Chen (Member, IEEE) received the Ph.D. degree in computer science from the University of Macau, Macau, China, in 2020.

He is currently an Associate Professor with the School of Electronics and Information Engineering, Wuyi University, Jiangmen, China. His current research interests include computational intelligence, machine learning, and affective computing.



Chi-Man Vong (Senior Member, IEEE) received the M.S. and Ph.D. degrees in software engineering from the University of Macau, Macau, China, in 2000 and 2005, respectively.

He is currently an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. His current research interests include machine learning methods and intelligent systems.

Yiu-Ming Cheung (Fellow, IEEE) received the Ph.D. degree from the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, in 2000.

He is currently the Chair Professor of artificial intelligence with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. His current research interests include machine learning, visual computing, data science, pattern recognition, multiobjective optimization, and information security.

He is a fellow of AAAS, IET, BCS, and AAIA. He is currently the Editor-in-Chief of IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE. He also serves as an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS (from 2014 to 2020), IEEE TRANSACTIONS ON COGNITIVE AND DEVELOPMENT SYSTEMS, *Pattern Recognition*, and *Neurocomputing*. More details can be found at: https://www.comp.hkbu.edu.hk/∼ymc.