IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 March 2024, accepted 1 April 2024, date of publication 8 April 2024, date of current version 16 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3386093

RESEARCH ARTICLE

Evaluation of Sparse Proximal Multi-Task Learning for Genome-Wide Prediction

YUHUA FAN, ILKKA LAUNONEN, MIKKO J. SILLANPÄÄ, AND PATRIK WALDMANN
Research Unit of Mathematical Sciences, University of Oulu, 90014 Oulu, Finland

Corresponding author: Patrik Waldmann (Patrik.Waldmann@oulu.fi)

This work was supported by the University of Oulu and the Research Council of Finland Profi5/HiDyn funding for Mathematics and AI: Data Insight for High Dimensional Dynamics under Grant 326291.

ABSTRACT Multi-task learning (MTL) is a learning paradigm whose aim is to leverage information shared across related tasks to improve the generalization of models. Motivated by the success of proximal optimization algorithms and single-task learning regression models, sparse proximal multi-task learning (SPMTL) for genome-wide prediction (GWP) should be explored. This study investigates proximal gradient descent splitting algorithms with five non-smooth sparsity-inducing norm regularizers, including the novel $L_{2,\frac{1}{2}}$ norm for GWP. Additionally, two popular methods based on Markov chain Monte Carlo (MCMC) are examined. To improve the computational efficiency, parallel Bayesian optimization strategy is employed for efficient hyperparameter tuning. Evaluation is conducted on three different real-world genomic datasets from mice, pigs and wheat, each associated with two, five, and four traits, respectively. Performance is assessed using mean squared error (MSE) and correlation coefficient between predicted and observed trait values in test sets. Experimental results reveal that the $L_{2,\frac{1}{2}}$ regularizer consistently achieves the best out-of-sample prediction across all datasets, demonstrating the effectiveness of SPMTL in leveraging shared information for improved GWP accuracy. Furthermore, the influence of different regularizers on sparsity and other properties of the SPMTL model are also explored.

INDEX TERMS Multi-task learning, genome-wide prediction, regularization, proximal algorithm, sparsity, Bayesian optimization.

I. INTRODUCTION

Genome-wide prediction (GWP) is a well-established technique where genomic markers covering the whole genome are used for selection of superior candidates in animal and plant breeding [1], [2]. Its applications extend to human genetics, where GWP facilities phenotype prediction and disease risk score estimation [3]. The genomic prediction task, which estimates simultaneously the effect of all markers, requires phenotyped and genotyped individuals in a training population. The idea is to predict phenotypes based solely on genomic marker information. However, genomic data often presents a high-dimensional challenge: thousands to millions of markers measured on a smaller number of individuals. Furthermore, due to the availability of affordable genome-wide dense marker maps, the use of marker information in practical animal and plant breeding programs is significantly increasing. This means that regularization methods have become important in GWP [4].

GWP methods traditionally focus on single phenotypic traits. However, data often encompass multiple traits that exhibit genetic correlations. To exploit these correlations and potentially improve prediction accuracy, methods have been developed using maximum-likelihood [5] and Bayesian models [6], [7]. Additionally, Lasso and its variants, such as Elastic net and adaptive Lasso, have been explored for both quantitative trait loci (QTL) mapping and genomic selection in multi-trait settings [8], [9]. Calus and Veerkamp [10] compared three models for multi-trait genomic selection (MT-GS) and demonstrated that the degree of genetic correlation significantly impacts the advantage of MT-GS over single-trait genomic selection (ST-GS). Subsequently, several Bayesian

The associate editor coordinating the review of this manuscript and approving it for publication was Xinyu Du.

multi-trait methods have been developed, including BayesCπ and BayesDπ [11], BayesA and multi-trait genomic best linear unbiased prediction (GBLUP) [12], the Bayesian multivariate antedependence model [13] and multiple trait Bayesian spike-and-slab variable selection [14]. Recently, multi-trait Bayesian Lasso approaches for genome analysis and prediction have also been proposed [9], [15]. These multi-trait methods often outperform single-trait prediction methods. However, a major limitation of Bayesian regression models lies in their reliance on Markov chain Monte Carlo (MCMC) methods, which can become computationally expensive for large datasets with numerous regression parameters.

Proximal algorithms have established themselves as powerful tools for tackling large-scale optimization problems due to their ability to efficiently solve specific subproblems within the overall optimization process [16]. These subproblems, often easier to solve than the original problem, are iteratively addressed, guiding the algorithm towards the optimal solution. Proximal algorithms have emerged in several different fields, and consequently they are known by different names [17], [18]. Traditionally, proximal algorithms rely on a two-operator splitting scheme, as exemplified by the LASSO formulation which combines the Euclidean loss with an L_1-norm penalty [19]. However, recent advancements have focused on extending their applicability to multi-objective functions - those containing the sum of more than two separable convex functions. While minimizing such functions presents greater challenges, convergence guarantees have been established under specific conditions [20], [21].

Sparse proximal multi-task learning (SPMTL) leverages two key concepts: sparse proximal algorithms and shared feature selection. By jointly modeling related tasks, SPMTL capitalizes on the computational efficiency of proximal regularization with sparsity-inducing norms and the inherent relationships between tasks. This approach can enhance predictive accuracy, promote efficient variable selection, and improve model interpretability [22]. Regularized multi-task learning (MTL), an alternative approach, aims to learn robust and generalizable representations across tasks, leading to stronger complementary information and a reduced risk of overfitting in individual tasks [23], [24].

In this work, we treat the multitrait GWP problem as a MTL problem. Leveraging sparsity-inducing norms, we extend five models for SPMTL regression: Lasso [19], group Lasso [25], sparse group Lasso [26], nuclear norm regularizer [27], and the $L_{2,\frac{1}{2}}$ regularization [28] in the GWP context. These models were chosen for this study because (1) each model's regularizer has a unique way to efficiently combine information together over tasks, (2) they induce sparsity (i.e. feature selection) in the predictor vectors, and (3) they can all be implemented efficiently using proximal methods while preserving stability and generalization. For the first three models, we employ the alternating direction method of multipliers (ADMM) [18] with a constant step size and a pre-calculated inverse matrix to improve computational efficiency. For the nuclear norm and $L_{2,\frac{1}{2}}$ regularization, a contractive Peaceman-Rachford splitting method (PRSM) [27] is utilized to find the estimated coefficient matrix. We evaluate the proposed approaches on three real-world datasets. To expedite hyperparameter tuning, a parallel Bayesian optimization strategy is employed.

II. SPARSE PROXIMAL MULTI-TASK LEARNING MODELS

In this section, we introduce five sparse proximal multi-task learning (SPMTL) models which are formed through the imposition of different sparsity-inducing norm (penalty function). These models can promote the selection of shared features that influence multiple traits during joint prediction. Proximal gradient descent with proximal mapping is employed to efficiently solve for the optimal model parameters.

A. PRELIMINARIES

Given the genomic data (input) $X \in \mathbb{R}^{N\times J}$ for J markers, let $Y \in \mathbb{R}^{N\times K}$ denote the matrix for K traits (output) collected over N samples. Assume that the k-th column of Y for the k-th task is generated from a linear model as follows

$$\mathbf{y}_k = \mathbf{X}\beta_k + \epsilon_k, \qquad \forall k = \{1, \cdots, K\} \tag{1}$$

where $\beta_k = [\beta_{1k}, \cdots, \beta_{Jk}]^T$ is the regression coefficient vector for the k-th task (trait) and ϵ_k is Gaussian noise.

For a multi-task regression model, let $\mathbf{B} = [\beta_1, \cdots, \beta_K] \in \mathbb{R}^{J\times K}$ represent the matrix of regression coefficients for K tasks. Let $\mathcal{L}(\mathbf{B}; \mathbf{X})$ represent a real-valued cost function which is the negative log-likelihood of data, where the minimum is obtained at an interior point $\hat{\mathbf{B}}$ that satisfies the zero gradient equation $\nabla\mathcal{L}(\hat{\mathbf{B}}) = 0$ (i.e. the maximum likelihood solution). Then the MTL problem can be formulated as the optimization problem

$$\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} f(\mathbf{B}) \equiv \underbrace{\frac{1}{2}\|\mathbf{Y} - \mathbf{X}\mathbf{B}\|_F^2}_{\mathcal{L}(\mathbf{B};\mathbf{X})} + \Omega(\mathbf{B}), \tag{2}$$

where $\|\cdot\|_F$ denotes the Frobenius matrix norm and $\Omega(\mathbf{B})$ is a given sparsity-inducing norm. Minimizing the squared Frobenius norm is equivalent to minimize the sum of L_2 norms of the vectors. For different sparsity-inducing norms, the squared Frobenius norm in equation (2) is replaced by the sum of L_2 norms. The sparsity-inducing norm, $\Omega(\mathbf{B})$, promotes sparsity by penalizing large coefficients, driving them towards zero. Different choices of $\Omega(\mathbf{B})$ can lead to various sparsity patterns in the estimated coefficient matrix (denoted by $\hat{\mathbf{B}}$). The optimization process balances the goodness-of-fit (captured by the Frobenius norm) with sparsity (enforced by $\Omega(\mathbf{B})$), leading to models that potentially select informative features while discarding irrelevant ones.

B. PROXIMAL GRADIENT DESCENT FOR LASSO

GWP often leverages Lasso regression for its feature selection properties [19]. By penalizing the absolute values

of the coefficients (L_1 regularization), Lasso drives many coefficients to zero, effectively selecting a limited set of relevant features for prediction. This is particularly valuable in genomics, where identifying a small subset of genetic markers associated with a trait is crucial from a large pool of potential candidates. Additionally, L_1 regularization helps mitigate overfitting compared to ordinary least squares by constraining model complexity and reducing sensitivity to noise in the data. In the context of MTL, a single L_1 penalty is typically applied across all tasks, promoting the shared information. Incorporating the L_1-norm penalty function into the MTL optimization, problem (2) can be formulated as

$$\begin{aligned}\hat{\mathbf{B}} &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\mathcal{L}(\mathbf{B}) + \Omega(\mathbf{B})] \\ &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\underbrace{\frac{1}{2}\|\mathbf{Y} - X\mathbf{B}\|_2^2}_{\mathcal{L}(\mathbf{B})} + \underbrace{\lambda\|\mathbf{B}\|_1}_{g(\mathbf{B})}],\end{aligned} \tag{3}$$

where $\lambda \geq 0$ is a regularization parameter to control the sparsity of the coefficient matrix. The notation $\|\cdot\|_1$ stands for the L_1-norm which performs variable selection, and therefore produces a sparse coefficient matrix $\hat{\mathbf{B}}$ where some entries are zero. The alternating direction method of multipliers (ADMM) was proposed as a variant of augmented Lagrangian multiplier methods [29]. Instead of using a smoothing term to approximate the non-smooth L_1 term, ADMM takes advantage of the structure of the L_1 norm. Hence, it is easier to obtain closed-form solutions to the subproblems at each iteration [30]. Following the ADMM approach for the optimization problem presented above, the solution in equation (3) can be expressed as

$$\arg\min_{\mathbf{B},\mathbf{Z}} \mathcal{L}(\mathbf{B}) + g(\mathbf{Z}) \qquad s.t. \quad \mathbf{B} - \mathbf{Z} = 0. \tag{4}$$

According to the optimality condition [20], steps for updating the minimization of $\mathbf{B}$, $\mathbf{Z}$ and dual variable $\mathbf{W}$ are

$$\begin{cases} \mathbf{B}^{(k+1)} = \mathbf{prox}_{f,\rho}(\mathbf{Z}^{(k)} - \mathbf{W}^{(k)}) \\ \mathbf{Z}^{(k+1)} = \mathbf{prox}_{g,\lambda}(\mathbf{B}^{(k+1)} + \mathbf{W}^{(k)}) \\ \mathbf{W}^{(k+1)} = \mathbf{W}^{(k)} + \mathbf{B}^{(k+1)} - \mathbf{Z}^{(k+1)} \end{cases} \tag{5}$$

where $\mathbf{prox}_{f,\rho}$ is the proximal operator for f at parameter ρ and $\mathbf{prox}_{g,\lambda}$ is the proximal operator for g at parameter λ. The update of $\mathbf{B}$ using $\mathbf{prox}_{f,\rho}$ in equation (5) can be derived as

$$\begin{aligned}\mathbf{B}^{(k+1)} &= \arg\min_{\mathbf{B}^{(k)}} [\frac{1}{2}\left\|\mathbf{Y} - \mathbf{X}\mathbf{B}^{(k)}\right\|_2^2 \\ &\quad + \frac{\rho}{2}\left\|\mathbf{B}^{(k)} - \mathbf{Z}^{(k)} + \mathbf{W}^{(k)}\right\|_2^2] \\ &= (\mathbf{X}^T\mathbf{X} + \rho\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{Y} + \rho(\mathbf{Z}^{(k)} - \mathbf{W}^{(k)})).\end{aligned} \tag{6}$$

The matrix $\mathbf{X}^T\mathbf{X} + \rho\mathbf{I}$ in equation (6) is always invertible, regardless of how the matrix $\mathbf{X}$ is designed. When $\mathbf{X}$ is a large-scale data set, computation of the inverse matrix related to $\mathbf{X}^T\mathbf{X}$ in the iteration process will be time-consuming. Parameter $\rho > 0$ also affects the convergence rate of ADMM. By choosing ρ as a constant and pre-calculate the inverse matrix computational efficiency can be improved. Let $\rho = \frac{1}{\|\mathbf{X}\|_{op}^2}$, where $\|\cdot\|_{op}$ is an operator norm (i.e. the maximum singular value of $\mathbf{X}$ on the training set). The updating of $\mathbf{Z}$ through $\mathbf{prox}_{g,\lambda}$ can be represented as

$$\begin{aligned}\mathbf{Z}^{(k+1)} &= \arg\min_{\mathbf{Z}^{(k)}} \lambda\left\|\mathbf{Z}^{(k)}\right\|_1 + \frac{\rho}{2}\left\|\mathbf{B}^{(k+1)} - \mathbf{Z}^{(k)} + \mathbf{W}^{(k)}\right\|_2^2 \\ &= \mathcal{S}_\lambda(\mathbf{B}^{(k+1)} + \mathbf{W}^{(k)}),\end{aligned} \tag{7}$$

where $\mathcal{S}_\lambda(\mathbf{Z}) = (\mathbf{Z}-\lambda)_+ - (-\mathbf{Z}-\lambda)_+$ is the soft thresholding operator.

C. PROXIMAL GRADIENT DESCENT FOR GROUP LASSO

In the Lasso regression model, each variable is treated individually. However, variables often exhibit natural grouping structures in many scenarios. The group Lasso addresses this by leveraging structured sparsity through a penalty term formulated as the ratio of norms L_1/L_2 [25], [31]. This approach encourages sparsity at the group level, driving all variables within a group to be either zero (excluded) or non-zero (included) simultaneously. This property of group Lasso, inducing sparsity in learned coefficients, is particularly valuable for prediction tasks in genomics, where high-dimensional and sparse features are prevalent [32].

Moreover, the group Lasso regularizer facilitates the incorporation of prior knowledge about the data structure by grouping related features together and applying regularization jointly to these groups. This approach enhances both interpretability and prediction performance by encouraging shared sparsity patterns among related tasks. Assuming that the group structure, denoted by $\mathscr{L} = \{l_1, \cdots, l_{|\mathscr{L}|}\}$, is known for all tasks, the group Lasso estimator for any variable index $j \in \{1, \cdots, J\}$ and group $l \in \mathscr{L}$ is defined as

$$\begin{aligned}\hat{\mathbf{B}} &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\mathcal{L}(\mathbf{B}) + \Omega(\mathbf{B})] \\ &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\underbrace{\frac{1}{2}\|\mathbf{Y} - X\mathbf{B}\|_2^2}_{\mathcal{L}(\mathbf{B})} + \underbrace{\lambda\sum_{j=1}^{J}\left\|\mathbf{B}_j\right\|_2}_{g(\mathbf{B})}],\end{aligned} \tag{8}$$

where $\lambda \geq 0$ represents a tuning parameter and $\mathbf{B}_j = (\mathbf{B}_{1,j}, \cdots, \mathbf{B}_{K,j})$ is a vector of length K, where $j \in \{1, \cdots, J\}$ and K is the number of tasks. The group Lasso estimator in equation (8) can also be expressed in the form (4). The coefficient matrix $\mathbf{B}$ can be updated with $\mathbf{prox}_{f,\rho}$ in a similar manner as described in equation (6) for the Lasso estimator. To update $\mathbf{Z}$, the proximal operator of group Lasso can be formulated as

$$\mathbf{prox}_{g,\lambda} = [(1 - \frac{\lambda}{\left\|\mathbf{Z}_j\right\|})_+\mathbf{Z}_j]. \tag{9}$$

The group soft thresholding method can be defined as

$$\mathcal{GS}_\lambda(\mathbf{Z}_j) = \begin{cases} 0 & , \quad \left\|\mathbf{Z}_j\right\| \leqslant \lambda, \\ \frac{\|\mathbf{Z}_j\|_2 - \lambda}{\|\mathbf{Z}_j\|_2}\mathbf{Z}_j, & \text{otherwise.} \end{cases} \tag{10}$$

The iterative update scheme employed for the Lasso problem, where updates for matrices $\mathbf{B}^{(k+1)}$, $\mathbf{Z}^{(k+1)}$ and $\mathbf{W}^{(k+1)}$ occur sequentially, can be applied to the group Lasso problem. This approach demonstrably converges to the group Lasso regression estimator $\hat{\mathbf{B}}$ in equation (8).

D. PROXIMAL GRADIENT DESCENT FOR SPARSE GROUP LASSO

Building upon Lasso and group Lasso, sparse group Lasso offers benefits of both by imposing both individual and group-wise penalties on regression coefficients. This promotes sparsity at two levels: individual features and group of features. Consequently, it selects the most relevant individual predictors and groups of predictors for the task [26]. By combining the L_1-norm with the L_1/L_2 norm ratio, sparse group Lasso achieves two distinct types of sparsity: group-level and non-group level. This allows for both joint learning across related tasks through sparsity in task-specific coefficients and some level of trait-specific variation. The mathematical formulation of the sparse group Lasso estimator for the multi-task regression model is given by:

$$\begin{aligned}\hat{\mathbf{B}} &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\mathcal{L}(\mathbf{B}) + \Omega(\mathbf{B})] \\ &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\underbrace{\frac{1}{2}\|\mathbf{Y} - X\mathbf{B}\|_2^2}_{\mathcal{L}(\mathbf{B})} + \underbrace{\lambda_1 \sum_{j=1}^{J} \|\mathbf{B}_j\|_2}_{\text{group Lasso}} \\ &\quad + \underbrace{\lambda_2 \|\mathbf{B}\|_1}_{\text{ordinary Lasso}}],\end{aligned} \tag{11}$$

where the penalty functions are combined into $g(\mathbf{B}) = \varphi(\mathbf{B}) + \phi(\mathbf{B}) = \lambda_1 \sum_{j=1}^{J} \|\mathbf{B}_j\|_2 + \lambda_2 \|\mathbf{B}\|_1$ with the tuning parameters $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$.

Sparse group lasso leverages proximal gradient descent for coefficient updates. The coefficient matrix $\mathbf{B}$ can be updated with $\mathbf{prox}_{f,\rho}$, which is the same proximal operator as in the Lasso. The update of $\mathbf{Z}$ involves decomposing the proximal map of sparse group Lasso into the composition of proximal mappings $\mathbf{prox}_{\varphi}(\cdot)$ and $\mathbf{prox}_{\phi}(\cdot)$. Hence, the first step is to apply the proximal operator of group Lasso, followed by the proximal mapping of ordinary Lasso. This approach is advantageous in domains where tasks share underlying structures or features, but individual traits may not involve all genes. It can lead to more interpretable and generalizable models by capturing both shared and task-specific effects.

E. PROXIMAL GRADIENT DESCENT FOR THE NUCLEAR NORM REGULARIZER

Nuclear norm regularization, also known as trace norm or matrix norm regularization, is primarily used for low-rank matrix recovery or matrix completion tasks. It promotes low rank solutions by penalizing the singular values of the parameter matrix, driving many towards to zero and effectively reducing its rank [33]. Theoretical analyses based on Rademacher complexity demonstrate that trace norm regularization can be advantageous compared to group structured norms, particularly when dealing with limited samples per task or a large number of tasks [34], [35].

Similar to the approach outlined in [33], the nuclear norm regularized estimator was utilized to learn across multiple tasks with a limited number of observations. Introducing the nuclear norm on the space of $N \times J$ matrices, the solution to the corresponding convex optimization problem (2) takes the form

$$\begin{aligned}\hat{\mathbf{B}} &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\mathcal{L}(\mathbf{B}) + \Omega(\mathbf{B})] \\ &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\underbrace{\frac{1}{2}\|\mathbf{Y} - X\mathbf{B}\|_2^2}_{\mathcal{L}(\mathbf{B})} + \underbrace{\lambda \|\mathbf{B}\|_*}_{g(\mathbf{B})}].\end{aligned} \tag{12}$$

The solution of equation (12) can be expressed using the same form as in equation (4), representing a canonical convex minimization problem with linear constraints and a separable objective function [36], [37]. In comparison to ADMM, the Peaceman-Rachford Splitting Method (PRSM) also functions as a splitting version of the Uzawa method [38] and is applicable for solving problems with a nuclear norm penalty [39]. PRSM exhibits less robustness and converges under more restrictive assumptions than ADMM [40]. Observing the absence of strict contraction in the iterative sequence of PRSM's iterative sequence, He et al. [41] highlighted that PRSM with an undetermined relaxation factor might establish a worst-case convergence rate due to this property. Let $\mathbf{L} \in \mathbb{R}^{J\times K}$ denote the matrix of Lagrangian multipliers. Given matrices $\mathbf{Z}$, $\mathbf{B}$ and $\mathbf{L}$, employing the strict contraction proximal PRSM from [27], the iterative equations for solving the nuclear norm of SPMTL are as follows

$$\begin{cases}\mathbf{Z}^{(k+1)} = \mathbf{prox}_{f,\gamma}(\mathbf{B}^{(k)} + \mathbf{L}^{(k)}), \\ \mathbf{L}^{(k+\frac{1}{2})} = \mathbf{L}^{(k)} - \alpha\gamma(\mathbf{Z}^{(k+1)} - \mathbf{B}^{(k)}), \\ \mathbf{B}^{(k+1)} = \mathbf{prox}_{g,\lambda}(\mathbf{Z}^{(k+1)} - \mathbf{L}^{(k+\frac{1}{2})}), \\ \mathbf{L}^{(k+1)} = \mathbf{L}^{(k+\frac{1}{2})} - \alpha\gamma(\mathbf{Z}^{(k+1)} - \mathbf{B}^{(k+1)}),\end{cases} \tag{13}$$

where α is a relaxation factor that impacts the convergence speed. This factor is also associated with the penalty parameter γ during Lagrange multiplier updates. A key distinction between proximal PRSM and ADMM lies in the inclusion of an intermediate update for the multipliers, denoted by $\mathbf{L}^{(k+\frac{1}{2})}$. This intermediate step ensures balanced treatment of matrices $\mathbf{Z}$ and $\mathbf{B}$, leading to a strictly contractive iteration sequence that converges to the solution of the original optimization problem. As evident from the outlined steps, the core proximal computations involve solving two subproblems, one for $\mathbf{B}$ and another for $\mathbf{Z}$. The initial update step for $\mathbf{Z}$ can be reformulated as

$$\mathbf{Z}^{(k+1)} = (\mathbf{X}^T\mathbf{X} + \rho\mathbf{I})^{-1}(\mathbf{X}^T\mathbf{Y} + \gamma\mathbf{B}^{(k)} + \mathbf{L}^{(k)}). \tag{14}$$

Given the singular soft-threshold operator [42], $\mathbf{B}$ can be updated as

$$\mathbf{B}^{(k+1)} = \mathcal{S}_\tau(\mathbf{Z}^{(k+1)} - \frac{1}{\gamma}\mathbf{L}^{(k+\frac{1}{2})}), \tag{15}$$

where $\tau = \frac{\lambda}{\gamma}$. With the value of $\mathbf{Z}^{(k+1)}$ computed, we can update the Lagrangian multiplier $\mathbf{L}^{(k+\frac{1}{2})}$ first, and then compute the updated $\mathbf{B}^{(k+1)}$. The goal is to estimate the coefficient matrix $\hat{\mathbf{B}}$, which is assumed to have a nearly low-rank structure, that is, the singular values of $\hat{\mathbf{B}}$ will decay fast enough for $\hat{\mathbf{B}}$ to be well approximated by a low-rank matrix. Given the estimated coefficient matrix $\hat{\mathbf{B}}$ of rank r with the SVD $\hat{\mathbf{B}} = \mathbf{U}\Sigma\mathbf{V}^T$, where $\Sigma = diag(\{\sigma_i\}_{1\leqslant i\leqslant r})$, and $\mathbf{U}$ and $\mathbf{V}$ are orthogonal matrices. For $\tau \geq 0$, the soft-thresholding operator $\mathcal{S}_\tau$ can be defined as

$$\mathcal{S}_\tau(\hat{\mathbf{B}}) = \mathbf{U}\mathcal{S}_\tau(\Sigma)\mathbf{V}^T,\ \mathcal{S}_\tau(\Sigma) = diag(\{(\sigma_i - \tau)_+\}). \quad (16)$$

By applying a soft-thresholding rule to the singular values of $\hat{\mathbf{B}}$, we can effectively shrink the regression coefficients towards zero. This process continues until convergence or until the maximum number of iterations is reached. The computation costs of the inverse of the matrix $\mathbf{X}^T\mathbf{X}$ and the singular value decomposition of $\hat{\mathbf{B}}$ can be very high. We pre-calculate $(\alpha I + \gamma\mathbf{X}^T\mathbf{X})^{-1}$ once before the iterations start. We set the parameter γ as 1.0 and optimize the relaxation factor α.

F. PROXIMAL GRADIENT DESCENT FOR $L_{2,\frac{1}{2}}$ REGULARIZATION

Generalized regularization methods become particularly valuable when explanatory variables exhibit a group structure. Examples include the group smoothly clipped absolute deviation (SCAD) penalty [43] and the group minimax concave penalty (MCP) models [44]. $L_{p,q}$ regularization (where $p \geq 1$ and $0 \leq q \leq 1$) offers an extension to existing group sparse optimization techniques in two ways. First, it incorporates lower-order regularization, encompassing the L_q regularization problem ($q < 1$). Second, it generalizes to group sparse optimization, including the special case of $L_{2,1}$ regularization. For the L_q regularization problem ($0 < q < 1$), prior work has shown that $L_{\frac{1}{2}}$ regularization admits fast solutions via the proximal gradient method [45], [46]. In terms of application, numerical experiments on both simulated data and real data in gene transcriptional regulation indicate that $L_{p,\frac{1}{2}}$ regularization is superior among $L_{p,q}$ regularizations for $q \in [0, 1]$. It specifically outperforms $L_{p,1}$ and $L_{p,0}$ in both accuracy and robustness [28], [47], [48].

Multi-task learning often involves related tasks that share similar underlying features or parameters. To exploit these relationships and enhance model interpretability and generalizability, $L_{2,\frac{1}{2}}$ group regularization is utilized to promote sparsity within and across tasks, leading to the selection of a common feature subset that is relevant to all tasks. Additionally, the non-convex nature of $L_{2,\frac{1}{2}}$ norm offers more flexibility in modeling complex relationships between tasks compared to convex regularizers like the nuclear norm [28], [49]. Following the group Lasso formulation, we leverage the same group structure and notations to achieve sparsity of the input data $\mathbf{X}$ using the $L_{2,\frac{1}{2}}$ norm. The estimator for the multi-task regression problem with $L_{2,\frac{1}{2}}$ regularization can be expressed as

$$\begin{aligned}\hat{\mathbf{B}} &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\mathcal{L}(\mathbf{B}) + \Omega(\mathbf{B})] \\ &= \arg\min_{\mathbf{B}\in\mathbb{R}^{J\times K}} [\underbrace{\frac{1}{2}\|\mathbf{Y} - X\mathbf{B}\|_2^2}_{\mathcal{L}(\mathbf{B})} + \underbrace{\lambda\sum_{j=1}^{J}\left\|\mathbf{B}_j\right\|_{2,\frac{1}{2}}}_{g(\mathbf{B})}]. \end{aligned} \quad (17)$$

The $L_{2,\frac{1}{2}}$ norm introduces a non-convex and non-smooth optimization problem, posing challenges for a quick and efficient solution. A strictly contractive PRSM [27] is employed to address the $L_{2,\frac{1}{2}}$ regularization problem, assuming that any non-zero group of a local minimum is active. The iterative equations align with those used for the nuclear norm regularizer in equations (13). With an initialization value, $\mathbf{Z}^{(k+1)}$ can be updated using the same step as in (14), and $(\alpha I + \gamma\mathbf{X}^T\mathbf{X})^{-1}$ can also be pre-calculated before the iterations start. $\mathbf{B}^{(k+1)}$ can be updated as

$$\mathbf{B}^{(k+1)} = \mathbf{prox}_{g,\lambda,\nu}(\mathbf{Z}^{(k+1)} - \frac{1}{\gamma}\mathbf{L}^{(k+\frac{1}{2})}), \quad (18)$$

where we, for ease of notation, write $\tilde{\mathbf{B}} = \mathbf{Z}^{(k+1)} - \frac{1}{\gamma}\mathbf{L}^{(k+\frac{1}{2})}$, and then define the proximal operator as

$$\begin{cases} \dfrac{16\left\|\tilde{\mathbf{B}}\right\|_2^{\frac{3}{2}}\cos^3\left(\frac{\pi}{3} - \frac{\varpi(\tilde{\mathbf{B}})}{3}\right)}{3\sqrt{3}\nu\lambda + 16\left\|\tilde{\mathbf{B}}\right\|_2^{\frac{3}{2}}\cos^3\left(\frac{\pi}{3} - \frac{\varpi(\tilde{\mathbf{B}})}{3}\right)}\tilde{\mathbf{B}}, \\ \quad \left\|\tilde{\mathbf{B}}\right\|_2 \geq \frac{3}{2}(\nu\lambda)^{\frac{2}{3}}, \\ 0, \quad \left\|\tilde{\mathbf{B}}\right\|_2 < \frac{3}{2}(\nu\lambda)^{\frac{2}{3}}, \end{cases} \quad (19)$$

where ν is a constant and $\varpi(\tilde{\mathbf{B}}) = \arccos(\frac{\nu\lambda}{4}(\frac{3}{\left\|\tilde{\mathbf{B}}\right\|_2})^{\frac{3}{2}})$ [28]. The Lagrange multipliers $\mathbf{L}^{(k+\frac{1}{2})}$ and $\mathbf{L}^{(k+1)}$ are updated following the equations (13). In this study, $B_j = 0$ signifies a zero group, indicating that all elements within the group are zero, while $B_j \neq 0$ means a non-zero group, implying the presence of non-zero elements in the group. The parameter ν within the model is set to $\frac{1}{2}$ to verify the relaxation factor for a higher convergence rate.

III. PARALLEL BAYESIAN OPTIMIZATION IN SPMTL

Bayesian optimization (BO) stands out as an iterative optimization technique that employs probabilistic modeling to identify the optimal set of hyperparameters. In this study, BO is performed in a parallel fashion to improve execution time. The objective is to minimize the function $f(\lambda)$ according to

$$\lambda_{opt} \in \arg\min_{\lambda\in\Gamma} f(\lambda), \quad (20)$$

where λ represents the regularized parameter for the MTL models and Γ denotes the full set of potential parameters. The objective is to find an optimal parameter λ_{opt} that yields

the smallest test MSE across each sparse proximal multi-task learning model.

BO conducts an iterative search for λ_{opt} using an acquisition function to balance exploration and exploitation. Exploration involves searching near promising observations, while exploitation involves exploring new regions. Various acquisition functions have been developed [50]. The expected improvement (EI) is as a popular and easily implemented function that is defined as [51]

$$\mathbf{EI}_{\mathbf{f}(\lambda^*)}[\lambda|\mathcal{D}] = \int_{-\infty}^{f(\lambda^*)} (f(\lambda^*) - f(\lambda)) p(f(\lambda)|\lambda, \mathcal{D}) df(\lambda), \tag{21}$$

where $\mathcal{D} = \{\lambda_n, f(\lambda_n)\}_{n=1}^{|\mathcal{D}|}$ is a set of parameter observations, and $|\mathcal{D}|$ denotes the size of set $\mathcal{D}$. The function value $f(\lambda^*)$ acts as a control parameter that must be specified in the algorithm. To compute the acquisition functions, the distribution $p(f(\lambda)|\lambda, \mathcal{D})$ needs to be modeled. A common choice is Gaussian process regression (GPR), while other choices include Random Forests as in SMAC and Kernel Density Estimators (KDEs) as in TPE [52], [53].

In the optimization of loss functions, a tree-structured configuration space is often utilized. The Tree Parzen Estimator (TPE) provides a unique perspective on modeling and optimizing functions. Unlike GPR, which directly models $p(f(\lambda)|\lambda)$, TPE models $p(\lambda|f(\lambda))$ and $p(f(\lambda))$ based on observations. It comes from the fact that a tree-structured search space includes some conditional parameters and the Parzen estimators which are kernel density estimators (KDEs). TPE estimates $p(f(\lambda)|\lambda, \mathcal{D})$ using the assumption

$$p(\lambda|f(\lambda), \mathcal{D}) = \begin{cases} p(\lambda|\mathcal{D}^{(\psi)}), & \text{if } f(\lambda) \le f(\lambda^{\varsigma}), \\ p(\lambda|\mathcal{D}^{(\zeta)}), & \text{if } f(\lambda) > f(\lambda^{\varsigma}). \end{cases} \tag{22}$$

Here, the top-quantile ς can be computed at each iteration based on the number of observations $|\mathcal{D}|$, and $f(\lambda^{\varsigma})$ is the top-ς-quantile objective value in the set of observations $\mathcal{D}$. Assume that $\mathcal{D}$ has been sorted by $f(\lambda_n)$ in ascending order. Then the "good group" and the "bad group" are obtained as $\mathcal{D}^{(\psi)} = \{(\lambda_n, f(\lambda_n)\}_{n=1}^{|\mathcal{D}|^{\psi}}$ and $\mathcal{D}^{(\zeta)} = \{(\lambda_n, f(\lambda_n)\}_{n=|\mathcal{D}|^{\psi}+1}^{|\mathcal{D}|}$, respectively. The KDEs in the above equation can be obtained via

$$\begin{aligned} p(\lambda|\mathcal{D}^{(\psi)}) &= w_0^{(\psi)} p_0(\lambda) + \sum_{n=1}^{|\mathcal{D}|^{\psi}} w_n \mathcal{K}(\lambda, \lambda_n|b^{(\psi)}), \\ p(\lambda|\mathcal{D}^{(\zeta)}) &= w_0^{(\zeta)} p_0(\lambda) + \sum_{n=|\mathcal{D}|^{\psi}+1}^{|\mathcal{D}|} w_n \mathcal{K}(\lambda, \lambda_n|b^{(\zeta)}), \end{aligned} \tag{23}$$

where the weights $\{w_n \in [0, 1]\}_{n=1}^{|\mathcal{D}|}$ are determined at each iteration and the summations of the weights for $p(\lambda|\mathcal{D}^{(\psi)})$ and $p(\lambda|\mathcal{D}^{(\zeta)})$ are 1. Note that $\mathcal{K}$ is a kernel density function with bandwidths $b^{(\psi)}$ and $b^{(\zeta)}$, and p_0 is a non-informative prior [53]. Finally, the acquisition function can be reformulated as

$$\mathbb{P}(f(\lambda) \le f(\lambda^{\varsigma})|\lambda, \mathcal{D}) \overset{\text{rank}}{\simeq} \varrho(\lambda|\mathcal{D}) = \frac{p(\lambda|\mathcal{D}^{(\psi)})}{p(\lambda|\mathcal{D}^{(\zeta)})}, \tag{24}$$

where $\varrho(\lambda|\mathcal{D})$ is the density ratio, which is used to judge the promise of a parameter configuration. The next evaluation point is selected by maximizing the ratio $\varrho(\lambda|\mathcal{D})$. In each iteration of the BO, the parameters of the TPE will be calculated first and then the configuration with the best acquisition function value based on the samples from the KDE will be picked to build a better group $\mathcal{D}^{(\psi)}$. This procedure can guarantee the global convergence of the TPE [54].

Parallel computing is employed to evaluate multiple hyper-parameter configurations concurrently where each evaluation corresponds to a potential sample in the probabilistic models. The objective function $f(\lambda)$ in equation (20) represents the mean test MSE across all cross-validation folds

$$f(\lambda) = \frac{1}{M} \sum_{m=1}^{M} \mathbf{MSE}_m, \tag{25}$$

where M is the number of cross-validation folds and $\mathbf{MSE}_m$ represents the test MSE obtained from fold m. The hyperparameter optimization process terminates if the improvement in the best observed objective value satisfies $|f(\lambda^*) - f(\lambda)| < \xi$. A parallel TPE BO implementation avoids generating new sample points too aggressively by utilizing a classification method over each fold to build surrogate models. This strategy ensures effective global exploration, preventing convergence to local optima.

IV. EXPERIMENTS

A. DATA

The performances of SPMTL models were evaluated on three publicly available GWP datasets varying with respect to the number of individuals, the number of traits and the number of markers, as well as diverse genetic architectures underlying each trait. These datasets were chosen especially to cover a diverse range of applications and ensure a comprehensive assessment of the methods' performance across various scenarios.

1) MICE DATA

The mice data originates from the Wellcome Trust and has previously been used for whole-genome regression by [55] and [56]. The data is included in the BGLR package in the R language and comprises genotypes and phenotypes of 1814 mice [57]. Each mouse was genotyped at 10,346 single nucleotide polymorphisms (SNPs), which were coded as 0, 1 and 2. The dataset includes two traits: body length (BL) and body mass index (BMI) [57].

2) PIG DATA

The MTL investigation was extended to another public dataset sourced from pigs [58]. This dataset comprises 3,534

TABLE 1. Mice data. A mean summary of best performing multi-task learning models over 5-folds cross validation for two traits with optimal set of hyperparameters (the best-performing values are highlighted in boldface).

MTL method	MSE	Pearson Correlation	CPU time (s)	Convergence rate	Relaxation factor	BO Iterations	Number of non-zeros
Lasso with ADMM	0.220	0.611	186	271	-	29	**184**
Group Lasso with ADMM	0.207	0.69	120	293	-	25	394
Sparse group Lasso with ADMM	0.23	0.603	1231	110	-	**21**	297
Nuclear norm regularization with PRSM	0.223	0.608	634	**91**	1.3	30	rank:30
$L_{2,\frac{1}{2}}$ regularization with PRSM	**0.151**	**0.702**	**113**	110	1.3	27	289
Bayesian ridge regression	0.264	0.499	551	-	-	-	-
Spike-and-slab variable selection	0.262	0.512	1278	-	-	-	-

TABLE 2. Pig data. A mean summary of best performing multi-task learning models over 5-folds cross validation for five traits with optimal set of hyperparameters (the best-performing values are highlighted in boldface).

MTL method	MSE	Pearson Correlation	CPU time (s)	Convergence rate	Relaxation factor	BO iterations	Number of non-zeros
Lasso with ADMM	0.207	0.62	329	320	-	26	**2789**
Group Lasso with ADMM	0.185	0.703	320	334	-	24	7351
Sparse group Lasso with ADMM	0.221	0.642	2017	179	-	**23**	7668
Nuclear norm regularization with PRSM	0.221	0.635	954	**92**	1.3	28	rank:53
$L_{2,\frac{1}{2}}$ regularization with PRSM	**0.16**	**0.71**	**310**	135	1.2	27	5335
Bayesian ridge regression	0.275	0.571	7300	-	-	-	-
Spike-and-slab variable selection	0.304	0.554	8207	-	-	-	-

TABLE 3. Wheat data. A mean summary of best performing multi-task learning models over 5-folds cross validation for four traits with optimal set of hyperparameters (the best-performing values are highlighted in boldface).

MTL method	MSE	Pearson Correlation	CPU time (s)	Convergence rate	Relaxation factor	BO Iterations	Number of non-zeros
Lasso with ADMM	0.220	0.611	87	140	-	25	**79**
Group Lasso with ADMM	0.205	0.612	81	161	-	23	205
Sparse group Lasso with ADMM	0.23	0.603	87	110	-	**20**	327
Nuclear norm regularization with PRSM	0.223	0.608	100	**88**	1.2	24	rank:17
$L_{2,\frac{1}{2}}$ regularization with PRSM	**0.18**	**0.62**	**71**	95	1.3	24	116
Bayesian ridge regression	0.264	0.499	160	-	-	-	-
Spike-and-slab variable selection	0.229	0.512	140	-	-	-	-

individuals with high-density genotypes and phenotypes of five anonymized traits. After removing the missing data, a total of 2,314 samples were retained where each containing 52,843 SNP markers. The anonymization process for this data involved randomizing the SNP map order and recording only SNP identifiers.

3) WHEAT DATASET

The wheat data set originates from CIMMYT's Global Wheat Program and is also a part of the BGLR package [57]. It consists of 599 historical CIMMYT wheat lines that we treat as individuals. Each individual has four traits and genotypes available from 1447 Diversity Array Technology (DArT) markers [59]. As a quality control, all the markers with a minor allele frequency below 0.05 were eliminated, and any missing genotypes were imputed using samples from the marginal distribution of marker genotypes. Following these procedures, the dataset was reduced to 1279 DArT markers.

B. IMPLEMENTATION DETAILS

1) PARALLEL BAYESIAN OPTIMIZATION

The capabilities of JAX allow for conducting parallel BO across each fold [60]. Initial trials with the sparse group Lasso informed the selection of hyperparameter search bounds for BO. The lower bounds for λ_1 and λ_2 were set to 0.01 and 0.1, whereas the upper bounds were set to 30 and 10, respectively. For the other sparse proximal regularization methods, the lower and upper bounds of regularization parameter in BO were chosen as 0.001 and 100, respectively. The data sets were randomly divided into 5-fold cross-validation sets. To find the best parameters, BO was conducted for a maximum of 100 iterations and models were executed in parallel across each fold, with the test MSE calculated independently. Termination of the BO process was done by tracking improvement in model performance over the iterations. Convergence was reached when the improvement in test MSE fell below 1e-5 for five consecutive iterations. To maintain exploration diversity, the TPE method capitalized on inherent model stochasticity while integrating new recommendations [61]. The Hyperopt implementation [62] faciliated the BO process, leveraging JAX's high-performance computing capabilities [60]. The code for all algorithms is publicly available at: https://github.com/angelYHF/SPMTL-for-regression-models. Parallel computations were executed on an RTX 4090 with 60 cores.

2) OTHER SETTINGS

Two benchmark methods were chosen for comparison: multi-task Bayesian ridge regression model (also known as

Bayesian GBLUP) and the Bayesian multiple trait spike-and-slab variable selection model [14]. Both are popular choices for GWP in quantitative genetics and were implemented using the BGLR package [57]. For these methods, test predictions were imputed by the Gibbs sampling algorithm. The number of MCMC iterations was set to 6000, with the first 1000 MCMC rounds discarded as burn-in. To ensure numerical stability and interpretability of the model, all response and explanatory variables are centered to have zero mean before the optimization. This eliminates the need for an intercept term in the regression model. Average test MSE and Pearson correlation coefficient (r) were calculated across traits for each dataset for the evaluation.

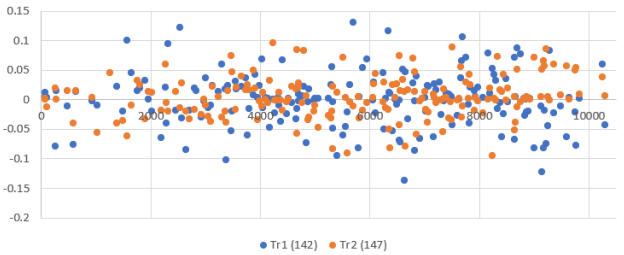
V. RESULTS AND DISCUSSION

A. EVALUATION OF MULTI-TASK PREDICTION

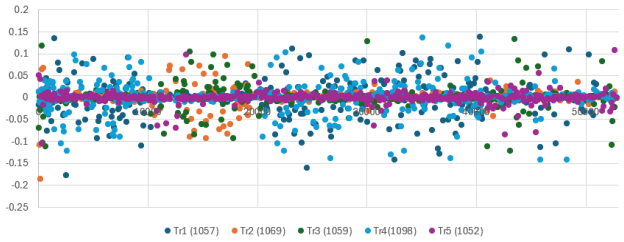
Performance evaluation was conducted using 5-fold cross-validation for all methods, employing their respective optimal hyperparameters. Among the evaluated methods, the $L_{2,\frac{1}{2}}$ norm achieved the best out-of-sample prediction performance across all datasets. It exhibited the smallest test MSE and the highest correlation coefficient r on the mice data (MSE $=$ $0.151, r = 0.702$), pig data (MSE $=$ $0.160, r = 0.71$), and wheat data (MSE $=$ $0.180, r = 0.620$). The group Lasso also yielded competitive results on all datasets (mice: MSE $=$ $0.207, r = 0.690$; pig: MSE $=$ $0.185, r =$ 0.703; wheat: MSE $=$ $0.205, r = 0.612$). The Lasso, sparse group Lasso and nuclear norm regularizer produced similar but somewhat worse prediction results on all data sets. However, multi-task Bayesian ridge regression and spike-and-slab variable selection models underperformed compared to the sparse proximal regression MTL models. These findings suggest that group-based sparse proximal methods are adept at capturing shared genetic signals across multiple traits, leading to improved prediction accuracy and generalizability.

B. SPARSITY

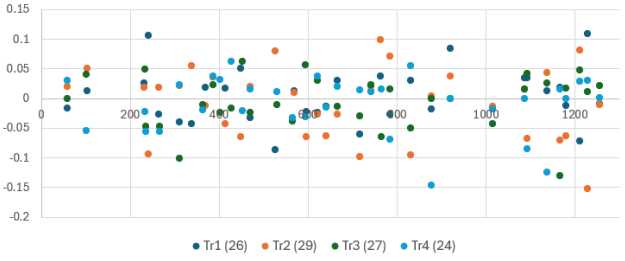
The number of non-zero coefficients identified by different regularization methods varied across datasets (Tables 1-3). Lasso achieved the sparsest solution, selecting 184, 2789, and 79 non-zero coefficients for mice, pig, and wheat data, respectively. Further on, the group Lasso produced 394 non-zero elements on the mice data, whereas for the pig and wheat datasets the method resulted in 7351 and 205 non-zero elements in the coefficient matrix, respectively. The sparse group Lasso showed a sparsity behavior similar to the group Lasso across the datasets, employing group effects to identify 297, 7668, and 327 variables on the mice, pig, and wheat datasets, respectively. Moreover, the $L_{2,\frac{1}{2}}$ procedure selected 289, 5335, and 116 regression coefficients, respectively. Regarding the nuclear norm, the model achieved a rank($\hat{\mathbf{B}}$) of 30, 53, and 17 on the mice, pig and wheat data, respectively. Compared to the group Lasso and the sparse group Lasso, the $L_{2,\frac{1}{2}}$ regularizer exhibits a tendency to select fewer groups.



(a) Mice data.



(b) Pig data.



(c) Wheat data.

FIGURE 1. Selected coefficients using SPMTL with the $L_{2,\frac{1}{2}}$ regularization norm on the three datasets with trait specific values in parentheses (zero values in the coefficient matrix have been removed).

However, the Lasso clearly displays the sparsest pattern, which is not surprising since all the group Lasso methods have an L_2 norm within groups that perform no variable selection. Figure 1 visualizes the selected coefficients for the three datasets with the $L_{2,\frac{1}{2}}$ method. The trait-specific values depicted in each subfigure present the number of selected features using $L_{2,\frac{1}{2}}$ regularization norm with PRSM on three datasets.

C. CONVERGENCE OF PROXIMAL ALGORITHMS

1) ADMM VS PRSM

This study compared the convergence rates of proximal gradient descent splitting with randomized shuffling (PRSM) and alternating direction method of multipliers (ADMM) for solving SPMTL with different regularizers. The PRSM algorithm was employed to address the nuclear norm regularizer and $L_{2,\frac{1}{2}}$ norm regularizer, while ADMM was utilized to solve three other Lasso-type MTL models. The nuclear norm regularizer exhibited the highest convergence rate, completing in 91, 92 and 88 iterations on the datasets (Tables 1-3). In the case of $L_{2,\frac{1}{2}}$ regularization, convergence was reached in 110, 135 and 95 iterations across the three

datasets. Regarding the group Lasso model, slightly higher numbers of iterations were observed compared to the Lasso, plausibly due to the non-smoothness introduced by group-level sparsity. For the sparse group Lasso, the ADMM algorithm achieved rapid convergence despite incorporating interactions between individual and group sparsities. The number of iterations required to achieve convergence on pig, mice and wheat datasets were 179, 110 and 110 respectively. Our results suggest that PRSM generally leads to faster convergence compared to ADMM for the tested regularizers and datasets.

2) CONVERGENCE OF BO

The analysis of Tables 1-3 reveals consistent convergence patterns for Bayesian optimization (BO) across various regression models and datasets. Among the five tested SPMTL methods with BO, Sparse Group Lasso achieved the fastest convergence, reaching completion in 21, 23, and 20 iterations for mice, pig, and wheat datasets, respectively. Group Lasso also demonstrated fast convergence, requiring only 25, 24, and 23 iterations for the same datasets. Likewise, the $L_{2,\frac{1}{2}}$ regularization needed the same number of iterations (27) on the mice and the pig datasets and 24 iterations on the wheat dataset. The nuclear norm regularizer exhibited slightly higher iterations (30, 28, and 24). The observed variation in BO iterations can be partially attributed to the inherent stochasticity of the method. The evaluation search is crucial for BO to accumulate knowledge about the parameter space, facilitating effective exploration of relevant hyperparameters. While some variation exists between regularization methods, the overall differences in BO iterations were surprisingly small. This suggests that BO is a robust and effective hyperparameter optimization technique for SPMTL.

3) RELAXATION FACTOR IN PRSM

The relaxation factor α in the Peaceman-Rachford splitting method is a parameter that scales the updates in each iteration. Tuning of this factor is crucial as it significantly impacts convergence speed. We explored a rang of values 0.9 to 1.4 based on some initial trails. In the case of the mice data with the nuclear norm regularizer, the convergence rate was 95 for $\alpha \leqslant 1.1$ and then decreased to 91 for $\alpha \in \{1.2, 1.3\}$, before rising to 93 for $\alpha = 1.4$. Similar trends were observed on the pig data, where the highest convergence rate of 92 was achieved when $\alpha \in \{1.2, 1.3\}$. Nearly the same trends were observed on the wheat data, where the highest convergence rate of 88 was achieved when $\alpha = 1.2$. For the $L_{2,\frac{1}{2}}$ regularization method on the mice data, the convergence rate was at 137 at $\alpha = 0.9$, increased to 140 for $\alpha \in \{1.0, 1.1\}$, and then decreased to 110 for $\alpha \in \{1.2, 1.3\}$. On the pig data, the convergence rate was 137 for $\alpha \leqslant 1.1$, then decreased from 136 for $\alpha = 1.3$ to 135 for $\alpha = 1.2$. Finally, on the wheat data, the convergence rate was 112 for $\alpha = 0.9$, increased to 123 for $\alpha \in \{1.0, 1.1\}$, and then decreased to 95 when $\alpha = 1.2$. The best relaxation factors in PRSM are shown in Table 1 to Table 3. It is important to note that a relaxation factor greater than 1 accelerates convergence in our study, but its selection should be carefully evaluated in order to prevent any algorithmic instability.

D. CPU TIME

We employed a parallel Bayesian optimization (BO) strategy to efficiently tune the hyperparameters of various regularization techniques across the three datasets. Tables 1-3 summarize the execution times in seconds. The results demonstrate significant differences in execution times, with parallel BO substantially reducing the search time compared to MCMC-based algorithms. Leveraging parallel computing resources significantly accelerates the optimization process. The group Lasso required 120 seconds on the mice data, 320 seconds on the pig data, and 81 seconds on the wheat data. Regarding the $L_{2,\frac{1}{2}}$ regularization, the CPU times were 113, 310 and 71 seconds over three datasets, respectively. In the case of Lasso, the CPU times were 186, 329, 87 seconds on the three datasets, respectively. It is evident that the execution times for the other methods were larger, emphasizing the computational demands associated with MTL in genome-wide prediction. In addition, the group Lasso exhibits more variability possibly due to the non-differentiability of the penalty term at zero. The $L_{2,\frac{1}{2}}$ regularization method gave more stable solutions and was less computationally demanding. It also proved to be more robust to outliers. These results highlight the computational efficiency gains achieved by parallel BO and the varying demands of different regularization techniques in MTL for GWP.

VI. CONCLUSION

This study investigates the statistical and computational properties of sparse proximal multi-task learning (SPMTL) for genome-wide prediction (GWP). We evaluate five proximal regularization methods, including Lasso, group Lasso, sparse group Lasso, nuclear norm, and the novel $L_{2,\frac{1}{2}}$ norm. The results demonstrate the efficacy of SPMTL in performing joint predictions across correlated traits while promoting informative genetic variant selection. Additionally, we compared these methods with multi-task Bayesian ridge regression and spike-and-slab analyses. Results indicate the efficacy and enhancement of SPMTL in genome-wide prediction. Notably, the $L_{2,\frac{1}{2}}$ regularization achieved superior prediction accuracy across all three datasets, highlighting its potential for leveraging shared genetic information. Furthermore, the nuclear norm exhibited the fastest convergence, while the Lasso selected the fewest predictors. A parallel Bayesian optimization strategy significantly enhances computational efficiency in hyperparameter tuning.

Despite our rather comprehensive evaluation of SPMTL for GWP, some limitations warrant consideration. Firstly, although the pig dataset is relatively large, comprising

2,314 individuals with 52,843 genomic markers, its size is still limited compared to datasets containing hundreds of thousands to millions of genetic variants. Secondly, SPMTL methods may struggle to efficiently handle ultra high-dimensional data, leading to increased computational complexity and scalability issues. SPMTL models are prone to overfitting, particularly when the tasks are highly correlated. Careful regularization and cross-validation are imperative to ensure that these models generalize well to unseen data. Finally, while this study investigates five different proximal regularization methods, they are all strictly linear. Further investigation is therefore required for improved feature selection, reduced overfitting, enhanced prediction accuracy and more efficient parameter tuning.

REFERENCES

[1] T. H. E. Meuwissen, B. J. Hayes, and M. E. Goddard, "Prediction of total genetic value using genome-wide dense marker maps," *Genetics*, vol. 157, no. 4, pp. 1819–1829, Apr. 2001.

[2] S. H. Lee, J. H. J. van der Werf, B. J. Hayes, M. E. Goddard, and P. M. Visscher, "Predicting unobserved phenotypes for complex traits from whole-genome SNP data," *PLoS Genet.*, vol. 4, no. 10, Oct. 2008, Art. no. e1000231.

[3] N. R. Wray, M. E. Goddard, and P. M. Visscher, "Prediction of individual genetic risk to disease from genome-wide association studies," *Genome Res.*, vol. 17, no. 10, pp. 1520–1528, Oct. 2007.

[4] P. Waldmann, G. Mészáros, B. Gredler, C. Fuerst, and J. Sölkner, "Evaluation of the lasso and the elastic net in genome-wide association studies," *Frontiers Genet.*, vol. 4, p. 270, Jan. 2013.

[5] C. Jiang and Z. B. Zeng, "Multiple trait analysis of genetic mapping for quantitative trait loci," *Genetics*, vol. 140, no. 3, pp. 1111–1127, Jul. 1995.

[6] S. Banerjee, B. S. Yandell, and N. Yi, "Bayesian quantitative trait loci mapping for multiple traits," *Genetics*, vol. 179, no. 4, pp. 2275–2289, Aug. 2008.

[7] C. Xu, X. Wang, Z. Li, and S. Xu, "Mapping QTL for multiple traits using Bayesian statistics," *Genet. Res.*, vol. 91, no. 1, pp. 23–37, Feb. 2009.

[8] Z. Li and M. J. Sillanpää, "Overview of LASSO-related penalized regression methods for quantitative trait mapping and genomic selection," *Theor. Appl. Genet.*, vol. 125, no. 3, pp. 419–435, Aug. 2012.

[9] D. Gianola and R. L. Fernando, "A multiple-trait Bayesian lasso for genome-enabled analysis and prediction of complex traits," *Genetics*, vol. 214, no. 2, pp. 305–331, Feb. 2020.

[10] M. P. Calus and R. F. Veerkamp, "Accuracy of multi-trait genomic selection using different methods," *Genet. Selection Evol.*, vol. 43, no. 1, pp. 1–14, Dec. 2011.

[11] D. Habier, R. L. Fernando, K. Kizilkaya, and D. J. Garrick, "Extension of the Bayesian alphabet for genomic selection," *BMC Bioinf.*, vol. 12, no. 1, pp. 1–12, Dec. 2011.

[12] C. Gondro, J. Van der Werf, and B. Hayes, *Genome-wide Association Studies and Genomic Prediction*. New York, NY, USA: Springer, 2013.

[13] J. Jiang, Q. Zhang, L. Ma, J. Li, Z. Wang, and J.-F. Liu, "Joint prediction of multiple quantitative traits using a Bayesian multivariate antedependence model," *Heredity*, vol. 115, no. 1, pp. 29–36, Jul. 2015.

[14] H. Ishwaran and U. Kogalur, "Spikeslab: Prediction and variable selection using spike and slab regression," *R J.*, vol. 2, no. 2, p. 68, 2010.

[15] H. Cheng, K. Kizilkaya, J. Zeng, D. Garrick, and R. Fernando, "Genomic prediction from multiple-trait Bayesian regression methods using mixture priors," *Genetics*, vol. 209, no. 1, pp. 89–103, May 2018.

[16] N. Parikh and S. Boyd, "Proximal algorithms," *Found. Trends Optim.*, vol. 1, no. 3, pp. 127–239, 2013.

[17] N. G. Polson, J. G. Scott, and B. T. Willard, "Proximal algorithms in statistics and machine learning," *Stat. Sci.*, vol. 30, no. 4, pp. 559–581, Nov. 2015.

[18] A. Beck, *First-order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.

[19] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[20] Y. Wang, W. Yin, and J. Zeng, "Global convergence of ADMM in nonconvex nonsmooth optimization," *J. Sci. Comput.*, vol. 78, no. 1, pp. 29–63, Jan. 2019.

[21] T. Evgeniou and M. Pontil, "Regularized multi-task learning," in *Proc. 10th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2004, pp. 109–117.

[22] X. Chen, Q. Lin, S. Kim, J. Pena, J. G. Carbonell, and E. P. Xing, "An efficient proximal-gradient method for single and multi-task regression with structured sparsity," *Stat*, vol. 1050, p. 26, Jan. 2010.

[23] R. Caruana, "Multitask learning," *Mach. Learn.*, vol. 28, no. 1, pp. 41–75, 1997.

[24] L. Chen, C. Li, S. Miller, and F. Schenkel, "Multi-population genomic prediction using a multi-task Bayesian learning model," *BMC Genet.*, vol. 15, no. 1, p. 53, 2014.

[25] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *J. Roy. Stat. Soc. B, Stat. Methodol.*, vol. 68, no. 1, pp. 49–67, Feb. 2006.

[26] S. Noah, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graph. Stat.*, vol. 22, no. 2, pp. 231–245, 2013.

[27] J. Fan, W. Wang, and Z. Zhu, "A shrinkage principle for heavy-tailed data: High-dimensional robust low-rank matrix recovery," *Ann. Statist.*, vol. 49, no. 3, pp. 1239–1266, Jun. 2021.

[28] Y. Hu, C. Li, K. Meng, J. Qin, and X. Yang, "Group sparse optimization via $\ell_{p,q}$ regularization," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 960–1011, 2017.

[29] D. Gabay and B. Mercier, "A dual algorithm for the solution of nonlinear variational problems via finite element approximation," *Comput. Math. Appl.*, vol. 2, no. 1, pp. 17–40, 1976.

[30] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[31] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{p,q}$-norm minimization," 2012, *arXiv:1205.2631*.

[32] A. Nouira and C.-A. Azencott, "Multitask group lasso for genome wide association studies in diverse populations," in *Proc. Biocomputing*, Dec. 2021, pp. 163–174.

[33] E. Boursier, M. Konobeev, and N. Flammarion, "Trace norm regularization for multi-task learning with scarce data," in *Proc. Conf. Learn. Theory*, 2022, pp. 1303–1327.

[34] N. Yousefi, Y. Lei, M. Kloft, M. Mollaghasemi, and G. C. Anagnostopoulos, "Local Rademacher complexity-based learning guarantees for multi-task learning," *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 1385–1431, 2018.

[35] T. K. Pong, P. Tseng, S. Ji, and J. Ye, "Trace norm regularization: Reformulations, algorithms, and multi-task learning," *SIAM J. Optim.*, vol. 20, no. 6, pp. 3465–3489, Jan. 2010.

[36] J. Douglas and H. H. Rachford, "On the numerical solution of heat conduction problems in two and three space variables," *Trans. Amer. Math. Soc.*, vol. 82, no. 2, pp. 421–439, 1956.

[37] P. L. Lions and B. Mercier, "Splitting algorithms for the sum of two nonlinear operators," *SIAM J. Numer. Anal.*, vol. 16, no. 6, pp. 964–979, Dec. 1979.

[38] H. Uzawa, "Iterative methods for concave programming," *Stud. Linear Nonlinear Program.*, vol. 6, pp. 154–165, Dec. 1958.

[39] D. Gabay, *Applications of the Method of Multipliers to Variational Inequalities*. Amsterdam, The Netherlands: North Holland, 1983.

[40] X. Li and X. Yuan, "A proximal strictly contractive Peaceman-Rachford splitting method for convex programming with applications to imaging," *SIAM J. Imag. Sci.*, vol. 8, no. 2, pp. 1332–1365, Jan. 2015.

[41] B. He, H. Liu, Z. Wang, and X. Yuan, "A strictly contractive Peaceman-Rachford splitting method for convex programming," *SIAM J. Optim.*, vol. 24, no. 3, pp. 1011–1040, Jan. 2014.

[42] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1956–1982, Jan. 2010.

[43] L. Wang, H. Li, and J. Z. Huang, "Variable selection in nonparametric varying-coefficient models for analysis of repeated measurements," *J. Amer. Stat. Assoc.*, vol. 103, no. 484, pp. 1556–1569, Dec. 2008.

[44] J. Huang, P. Breheny, and S. Ma, "A selective review of group selection in high-dimensional models," *Stat. Sci.*, vol. 27, no. 4, pp. 481–499, Nov. 2012.

[45] R. Chartrand and V. Staneva, "Restricted isometry properties and nonconvex compressive sensing," *Inverse Problems*, vol. 24, no. 3, Jun. 2008, Art. no. 035020.

[46] Z. Xu, X. Chang, F. Xu, and H. Zhang, "$L_{1/2}$ regularization: A thresholding representation theory and a fast solver," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 7, pp. 1013–1027, May 2012.
[47] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4680–4688, Jul. 2011.
[48] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4689–4708, Jul. 2011.
[49] Y. Zhang, C. Wei, and X. Liu, "Group logistic regression models with $\ell_{p,q}$ regularization," *Mathematics*, vol. 10, no. 13, p. 2227, Jun. 2022.
[50] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 24, 2011, pp. 2546–2544.
[51] D. R. Jones, M. Schonlau, and W. J. Welch, "Efficient global optimization of expensive black-box functions," *J. Global Optim.*, vol. 13, no. 4, pp. 455–492, Dec. 1998.
[52] H. J. Kushner, "A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise," *J. Basic Eng.*, vol. 86, no. 1, pp. 97–106, Mar. 1964.
[53] S. Watanabe, "Tree-structured Parzen estimator: Understanding its algorithm components and their roles for better empirical performance," 2023, *arXiv:2304.11127*.
[54] S. Watanabe, N. Awad, M. Onishi, and F. Hutter, "Speeding up multi-objective hyperparameter optimization by task similarity-based meta-learning for the tree-structured Parzen estimator," 2022, *arXiv:2212.06751*.
[55] A. Legarra, C. Robert-Granié, E. Manfredi, and J.-M. Elsen, "Performance of genomic selection in mice," *Genetics*, vol. 180, no. 1, pp. 611–618, Sep. 2008.
[56] H. Okut, D. Gianola, G. J. M. Rosa, and K. A. Weigel, "Prediction of body mass index in mice using dense molecular markers and a regularized neural network," *Genet. Res.*, vol. 93, no. 3, pp. 189–201, Jun. 2011.
[57] P. Pérez and G. de los Campos, "Genome-wide regression and prediction with the BGLR statistical package," *Genetics*, vol. 198, no. 2, pp. 483–495, Oct. 2014.
[58] M. A. Cleveland, J. M. Hickey, and S. Forni, "A common dataset for genomic analysis of livestock populations," *G3 Genes|Genomes|Genetics*, vol. 2, no. 4, pp. 429–435, Apr. 2012.
[59] C. McLaren, L. Ramos, C. Lopez, and W. Eusebio, "Applications of the genealogy management system," Centro Internacional de Mejoramiento de Maiz y Trigo, Mexico, Mexico, Tech. Rep. 5, pp. 1–56, 2000.
[60] R. Frostig, M. J. Johnson, and C. Leary, "Compiling machine learning programs via high-level tracing," *Syst. Mach. Learn.*, vol. 4, no. 9, pp. 1–3, 2018.
[61] P. I. Frazier, "Bayesian optimization," in *Recent Advances in Optimization and Modeling of Contemporary Problems*. Arizona: INFORMS, 2018, pp. 255–278.
[62] J. Bergstra, D. Yamins, and D. Cox, "Hyperopt: A Python library for optimizing the hyperparameters of machine learning algorithms," in *Proc. Python Sci. Conf.*, 2013, p. 20.

YUHUA FAN received the Ph.D. degree from the School of Automation Sciences and Electrical Engineering, Beihang University, Beijing, China, in 2013. She is currently a Postdoctoral researcher in statistics with the Research Unit for Mathematical Sciences, University of Oulu. Her research interests include high dimensional data analysis, statistical machine learning, and machine learning theory and algorithmic.

ILKKA LAUNONEN received the Ph.D. degree in applied mathematics and statistics from the University of Oulu (UO), in 2016. He is currently a Coordinator of the UO and Research Council of Finland funded PROFI5 HiDyn Program, which acts as a multidisciplinary hub for data science and artificial intelligence to strengthen all UO scientific profiling areas and the 6G flagship. His research interests include Bayesian statistics and time series analysis, with applications in climate science and machine learning with health applications.

MIKKO J. SILLANPÄÄ is a Full Professor of statistics with the Research Unit of Mathematical Sciences, University of Oulu. He has a long and acknowledged experience in Bayesian statistics and computational methods in high-dimensional problems in biology, medicine, and other fields. His research interests include variable selection, functional data analysis, and predictive analytics. He is the Director of the University of Oulu HiDyn Program (2019–2025) funded by the Academy of Finland PROFI5: multidisciplinary data science and artificial intelligence hub strengthening all of University of Oulu's profiling areas and the 6G flagship.

PATRIK WALDMANN is an Associate Professor of statistics with the Research Unit for Mathematical Sciences, University of Oulu. His main research areas include high-dimensional data analysis, statistical machine learning, Bayesian statistics, statistical genetics and bioinformatics, and precision modeling in agriculture and medicine.