Safe Screening Rules for Group SLOPE

Runxue Bao[1](✉), Quanchao Lu[2], and Yanfu Zhang[3]

[1] University of Pittsburgh, Pittsburgh, PA 15260, United States
runxue.bao@pitt.edu
[2] Georgia Institute of Technology, Atlanta, GA 30332, United States
qlu43@gatech.edu
[3] The College of William and Mary, Williamsburg, VA 23185, United States
yzhang105@wm.edu

Abstract. Variable selection is a challenging problem in high-dimensional sparse learning, especially when group structures exist. Group SLOPE performs well for the adaptive selection of groups of predictors. However, the block non-separable group effects in Group SLOPE make existing methods either invalid or inefficient. Consequently, Group SLOPE tends to incur significant computational costs and memory usage in practical high-dimensional scenarios. To overcome this issue, we introduce a safe screening rule tailored for the Group SLOPE model, which efficiently identifies inactive groups with zero coefficients by addressing the block non-separable group effects. By excluding these inactive groups during training, we achieve considerable gains in computational efficiency and memory usage. Importantly, the proposed screening rule can be seamlessly integrated into existing solvers for both batch and stochastic algorithms. Theoretically, we establish that our screening rule can be safely employed with existing optimization algorithms, ensuring the same results as the original approaches. Experimental results confirm that our method effectively detects inactive feature groups and significantly boosts computational efficiency without compromising accuracy.

Keywords: Safe Screening Rules · Group SLOPE · Feature Selection.

1 Introduction

Group structures are ubiquitous in many high-dimensional problems with massive correlated and superfluous features. To obtain more stable and interpretable models with better prediction performance, many sparse learning models with grouping structures were proposed and achieved great success in many real-world applications. Group Lasso [39] and its variants, including Sparse Group Lasso [30], composite absolute penalties [41], tree Lasso [22], and Overlapping Group Lasso [18, 19], are the most popular ones for group feature selection, which encourages structured sparsity with the prior information of feature group structures.

In this paper, we focus on the adaptive group feature selection model - Group SLOPE [10, 15, 16]. Let design matrix $X = [x_1, \ldots, x_n]^\top \in \mathbb{R}^{n\times d}$ have n observations and d variables and $y \in \mathbb{R}^n$ denote the measurement vector. Given I is a partition of the set

Table 1: Representative safe screening algorithms.

Problem	Reference	Group-wise	Inseparability	Group Effects	Dynamic
Lasso	[13]	✗	✗	✗	Singly
Logistic Regression	[34]	✗	✗	✗	✗
Proximal Weighted Lasso	[28]	✗	✗	✗	Singly
SLOPE	[23]	✗	✓	✗	Singly
Group Lasso	[9]	✓	✗	✗	Singly
Sparse Group Lasso	[32]	✓	✗	✗	Singly
Tree Structured Group Lasso	[33]	✓	✗	✗	✗
Sparse-group Lasso	[26]	✓	✗	✗	Singly
Group SLOPE	Ours	✓	✓	✓	Doubly

$\{1, \ldots, d\}$ and W is a diagonal matrix with $W_{i,i} := w_i$ for $i = 1, \ldots, m$, $X_{I_i} \in \mathbb{R}^{n \times |I_i|}$ denotes a partition of the matrix X and $[\![\beta]\!]_{X,I} := \left(\|X_{I_1}\beta_{I_1}\|_2, \ldots, \|X_{I_m}\beta_{I_m}\|_2\right)^\intercal$ denotes the group effects, Group SLOPE can be formulated as follows:

$$\min_{\beta} P_\lambda(\beta) := \frac{1}{2}\|y - X\beta\|_2^2 + J_\lambda(W[\![\beta]\!]_{X,I}), \tag{1}$$

where $\beta \in \mathbb{R}^d$ denotes the unknown coefficient vector and $\lambda = [\lambda_1, \ldots, \lambda_m]$ is a non-negative regularization parameter vector of m non-increasing weights that $\lambda_1 \geq \ldots \geq \lambda_m$. The term $J_\lambda(b)$ denotes the ordered weighted l_1-norm as $J_\lambda(b) = \sum_{i=1}^m \lambda_i |b|_{[i]}$ where $b_{[1]} \geq \ldots \geq b_{[m]}$ are the ordered terms.

Group SLOPE penalty $J_\lambda(W[\![b]\!]_{X,I})$ adaptively penalizes the group effects based on the magnitude. Thus, Group SLOPE can simultaneously encourage the group effects to be equal and sparse, which is helpful to denoise and improve the prediction. [10] provided both nice empirical results and theoretical analysis for Group SLOPE on the adaptive selection of groups of predictors. In general, Group SLOPE can achieve the exact minimax estimation without any knowledge of coefficients sparsity and control the group false discovery rate at a specific level [10, 15]. The attractive properties above, which do not simultaneously exist in other models such as Group Lasso and SLOPE [8], had made Group SLOPE an effective method for the analysis in the high-dimensional setting [10, 15, 16]. Please note that Group SLOPE includes a broad set of sparse learning models. For example, Group SLOPE reduces to Group Lasso when $\lambda_1 = \ldots = \lambda_m$ and $w_i = \sqrt{|I_i|}$. Group SLOPE reduces to SLOPE when each group only has one variable and X is standardized to have unit column norms. Besides, Group SLOPE certainly includes Lasso, weighted Lasso [7] and L_∞-norm regression. Also, Group SLOPE can be easily extended to the logistic loss for classification tasks.

From an optimization perspective, the block-nonseparable group effects render the coordinate descent algorithm for Group Lasso ineffective. To address the computational challenges of Group SLOPE, proximal gradient methods have been introduced [10]. However, these methods encounter significant computational and memory challenges, particularly in high-dimensional settings. This is primarily because the algorithm processes all data points at each iteration, even when group coefficients are zero.

Various screening rules have been developed to speed up the training of sparse learning models by eliminating inactive features. [24] introduced a static safe screening rule for l_1-regularized problems, which eliminates features before the optimization process. Relaxing the constraints of the safe rule, [31] introduced a strong rule for Lasso that employs heuristic strategies through an active set method. However, this approach may mistakenly discard features. Additionally, the sequential safe rule presented in [35, 37] relies on the exact dual optimal solution, making it potentially time-consuming. More recently, [13] proposed a dynamic screening rule for Lasso, which is applied throughout the learning process, based on the duality gap, offering provable safety and improved speed compared to previous rules. This has led to the development of many dynamic screening rules for various sparse learning models [2–5, 26, 28, 29], all aimed at enhancing training efficiency. Thus, improving the efficiency of solving Group SLOPE via dynamic screening rules becomes both important and promising. Moreover, by reducing the number of model parameters, such techniques can also enhance inference performance, similar to the benefits observed with pruning methods [14, 17, 25, 36].

The goal of this research is to expedite the training process of Group SLOPE models through the application of safe screening techniques, enabling the secure exclusion of inactive groups whose parameters are guaranteed to be zeros during training. Table 1 outlines various representative safe screening algorithms, highlighting that such rules have been developed to boost the efficiency of training algorithms across numerous sparse learning models. However, the complex penalty structure of Group SLOPE, characterized by its block-nonseparable nature in relation to group effects, has so far hindered the development of safe screening rules for this model. The challenges can be summarized as follows: Firstly, unlike other models that penalize coefficients directly, Group SLOPE penalizes group effects $\|X\beta\|_2$, which does not directly enforce coefficient sparsity. Secondly, while other models are restricted to either feature-wise or separable group-wise penalties, Group SLOPE introduces the first non-separable group-wise feature selection method, with all hyperparameters for each group remaining unfixed during the training process—unlike in models such as Group Lasso or Sparse Group Lasso, where hyperparameters are predetermined before optimization.

In response to these challenges, this paper introduces a doubly dynamic safe screening rule tailored to the general Group SLOPE models. This represents, to our knowledge, the first safe screening rule specifically designed for adaptive group feature selection models. In high-dimensional settings where many groups have zero-valued coefficients, our screening rule efficiently identifies and excludes these inactive groups, thereby accelerating the original algorithms. Our approach begins by decoupling the design matrix to manage the block non-separable group effects. We then establish a doubly dynamic screening rule featuring a decreasing left bound and an increasing right bound, resulting in an expanding safe region. Crucially, the proposed screening rule is solver-independent and can be seamlessly integrated into existing iterative algorithms. Empirical evaluations on four benchmark datasets confirm that our approach yields significant computational advantages.

2 Safe Screening Rules for Group SLOPE

In this section, we first decouple the over-complex group effect penalty and then propose a doubly dynamic safe screening rule for Group SLOPE.

2.1 Decoupling the Group Effect Penalty

Different from other models, Group SLOPE penalizes group effects $\|X\beta\|_2$ directly. To propose a screening rule for Group SLOPE, we first derive an equivalent formulation of (1), which decomposes the design matrix as an orthogonal matrix and a corresponding full-row rank matrix. Specifically, by representing $X_{I_i} = U_i R_i$ where U_i is any matrix with $|I_i|$ orthogonal columns and R_i is the corresponding full-row rank matrix, we can obtain:

$$X\beta = \sum_{i=1}^{m} X_{I_i}\beta_{I_i} = \sum_{i=1}^{m} U_i R_i \beta_{I_i} = \widetilde{X}\eta, \tag{2}$$

$$\|X_{I_i}\beta_{I_i}\|_2 = \|R_i\beta_{I_i}\|_2 = \|\eta_{I_i}\|_2, \tag{3}$$

where $\widetilde{X}_{I_i} = U_i$ and $\eta_{I_i} := R_i\beta_{I_i}$ for $i = 1, \ldots, m$. Thus, the decoupled version of Problem (1) can be equivalently presented as:

$$\min_{\eta} \frac{1}{2}\|y - \widetilde{X}\eta\|_2^2 + J_\lambda(W[\![\eta]\!]_{\mathbb{I}}), \tag{4}$$

where $[\![\eta]\!]_{\mathbb{I}} := \big(\|\eta_{I_1}\|_2, \|\eta_{I_2}\|_2, \ldots, \|\eta_{I_m}\|_2\big)^\top$.

Considering the diagonal matrix W with $W_{i,i} = w_i$ for $i = 1, \ldots, m$, define Z as a diagonal matrix with $Z_{i,i} := 1/w_j$ for $i \in I_j$ where $i = 1, \ldots, d$ and $j = 1, \ldots, m$, we have $J_\lambda(W[\![\eta]\!]_{\mathbb{I}}) = J_\lambda([\![Z^{-1}\eta]\!]_{\mathbb{I}})$. Further, defining $b = Z^{-1}\eta$, we have $\eta = Zb$. Thus, denoting $\hat{X} = \widetilde{X}Z$, we can formulate (4) as

$$\begin{aligned}
&\min_{\eta} \frac{1}{2}\|y - \widetilde{X}\eta\|_2^2 + J_\lambda(W[\![\eta]\!]_{\mathbb{I}}) \\
&= \min_{b} \frac{1}{2}\|y - \widetilde{X}\eta\|_2^2 + J_\lambda([\![Z^{-1}\eta]\!]_{\mathbb{I}}) \\
&= \min_{b} \frac{1}{2}\|y - \widetilde{X}Zb\|_2^2 + J_\lambda([\![b]\!]_{\mathbb{I}}) \\
&= \min_{b} \frac{1}{2}\|y - \hat{X}b\|_2^2 + J_\lambda([\![b]\!]_{\mathbb{I}}).
\end{aligned} \tag{5}$$

That is to say, by the equivalent transformation above, the next step to achieve our aim is to propose a safe screening rule for Problem (5).

2.2 Dual Formulation and Screening Test

Problem (5) can be formulated as follows:

$$b = \operatorname*{arg\,min}_{b \in \mathbb{R}^d} P_\lambda(b) := F(b) + J_\lambda([\![b]\!]_{\mathbb{I}}), \tag{6}$$

where loss $F(b) = \sum_{i=1}^{n} f_i(x_i^\top b)$, $f_i : \mathbb{R} \to \mathbb{R}_+$ is the squared loss. Generally, (6) is convex, non-smooth, and non-separable.

We initiate the derivation of the screening test by reformulating the primal objective (6) into its dual. Leveraging insights from the dualization of l_1-regularized models as outlined in [21], the resulting dual problem takes the form:

$$\begin{aligned}
&\min_b F(b) + J_\lambda([\![b]\!]_{\mathbb{I}}) \\
&= \min_b \sum_{i=1}^{n} f_i(x_i^\top b) + J_\lambda([\![b]\!]_{\mathbb{I}}) \\
&= \min_b \sum_{i=1}^{n} f_i^{**}(x_i^\top b) + \sum_{i=1}^{m} \lambda_i \|b_{I_{[i]}}\|_2 \\
&= \min_b \sum_{i=1}^{n} \max_{\theta_i} [\beta x_i \theta_i - f_i^*(\theta_i)] + \sum_{i=1}^{m} \lambda_i \|b_{I_{[i]}}\|_2 \\
&= \min_b \max_\theta - \sum_{i=1}^{n} f_i^*(\theta_i) + \beta^\top X^\top \theta + \sum_{i=1}^{m} \lambda_i \|b_{I_{[i]}}\|_2 \\
&= \max_\theta - \sum_{i=1}^{n} f_i^*(\theta_i) + \min_b \beta^\top X^\top \theta + \sum_{i=1}^{m} \lambda_i \|b_{I_{[i]}}\|_2 \\
&= \max_{\theta \in \Delta} D(\theta) := \sum_{i=1}^{n} -f_i^*(\theta_i),
\end{aligned} \tag{7}$$

where $\theta \in \mathbb{R}^n$ is the solution of the dual problem. Note f_i^* is the convex conjugate of function f_i as

$$f_i^*(\theta_i) = \max_{z_i \in \mathbb{R}} \theta_i z_i - f_i(z_i). \tag{8}$$

Let us define $\widetilde{\theta} := \left(\|X_{I_1}^\top \theta\|_2, \ldots, \|X_{I_m}^\top \theta\|_2\right)^\mathsf{T}$. Under this definition, the constraint $\theta \in \Delta$ in (7) can be equivalently expressed as $\sum_{j \le i} \widetilde{\theta}_{[j]} \le \sum_{j \le i} \lambda_j$ for all $i = 1, \ldots, m$. We next apply the optimality condition associated with the minimization part in the penultimate expression of (7):

$$\min_b \beta^\top X^\top \theta + \sum_{i=1}^{m} \lambda_i \|b_{I_{[i]}}\|_2. \tag{9}$$

This optimality condition naturally leads to the constraint structure previously introduced in (7), thereby finalizing the dual reformulation of (6).

Leveraging the Fermat rule [6], we obtain

$$-X^\top \theta^* \in \partial J_\lambda([\![b]\!]_{\mathbb{I}}), \tag{10}$$

where θ^* denotes the dual optimum solution and $\partial J_\lambda([\![b]\!]_{\mathbb{I}})$ represents the subdifferential of the regularizer $J_\lambda([\![b]\!]_{\mathbb{I}})$.

Let $\widetilde{\mathcal{A}}^*$ be the index corresponding to inactive groups at optimality. The conditions for the partition $\mathcal{A}^*$ and $\widetilde{\mathcal{A}}^*$ of problems (9) can be separately expressed as:

$$-X_{I_{\mathcal{A}^*}}^\top \theta^* \in \partial J_{\lambda_{\mathcal{A}^*}}([\![b]\!]_{\mathbb{I}}), \tag{11}$$

$$-X_{I_{\widetilde{\mathcal{A}}^*}}^\top \theta^* \in \partial J_{\lambda_{\widetilde{\mathcal{A}}^*}}([\![b]\!]_{\mathbb{I}}). \tag{12}$$

For any group $i \in \mathcal{A}^*$, we have $b_{I_i}^* \neq 0$, it holds that

$$\|X_{I_i}^\top \theta^*\|_2 \in [\min_{j \in \mathcal{A}^*} \lambda_j, \max_{j \in \mathcal{A}^*} \lambda_j]. \tag{13}$$

Assuming both primal and dual optimal solutions are available, one can derive a safe screening rule for each group based on the following condition:

$$\|X_{I_i}^\top \theta^*\|_2 < \lambda_{|\mathcal{A}^*|} \Longrightarrow b_{I_i}^* = 0, \tag{14}$$

which implies that such a group can be safely discarded without affecting the final solution. This enables subsequent training stages to proceed with a significantly reduced parameter space, leading to faster training while preserving accuracy.

Nevertheless, the main difficulty lies in the fact that the screening conditions (14) necessitate prior knowledge of both the dual optimum and the order structure of the primal optimum, which can only be obtained after the full training process has been completed. As a result, these screening conditions cannot be utilized to enhance optimization during the training phase.

Therefore, our objective is to devise a screening rule capable of identifying as many inactive variables (i.e., those with coefficients that should be zero) as possible, using the screening test (14) without knowing the dual optimum or the order structure of the primal optimum during the optimization process. To this end, we can formulate safe screening rules by defining a screening region that is as large as possible, characterized by smaller lower bounds and larger upper bounds derived from the screening conditions (14).

2.3 Upper Bound for the Left Term

It is worth noting that the lower bound of the screening region corresponds to the upper bound of $\|X_{I_i}^\top \theta^*\|_2$. To this end, we focus on deriving a tight upper estimate for $\|X_{I_i}^\top \theta^*\|_2$ by monitoring the intermediate duality gap $G(b, \theta)$ throughout the training iterations.

Utilizing the triangle inequality, we have:

$$\|X_{I_i}^\top \theta^*\|_2 \leq \|X_{I_i}^\top \theta\|_2 + \|X_{I_i}\|_2 \|\theta - \theta^*\|_2. \tag{15}$$

Since each $f_i^*(\theta_i)$ in the dual is known to be strongly convex (see Proposition 3.2 in [21]), the overall dual objective $D(\theta) := \sum_{i=1}^n -f_i^*(\theta_i)$ inherits strong concavity *w.r.t.* θ. As a direct implication, we obtain the following upper bound:

$$D(\theta) \leq D(\theta^*) - \mathrm{tr}(\nabla D(\theta^*)^\top (\theta^* - \theta)) - \frac{1}{2}\|\theta - \theta^*\|_2^2. \tag{16}$$

This inequality enables us to derive a bound on the distance between any feasible dual iterate θ and the optimal dual solution θ^* based on the first-order condition summarized in Corollary 1

Corollary 1. *For any dual feasible point θ, the following estimate holds:*

$$\|\theta - \theta^*\|_2 \leq \sqrt{2G(b,\theta)}, \tag{17}$$

where $G(b,\theta) = P(b) - D(\theta)$ denotes the intermediate duality gap at training.

Proof. We begin by applying the first-order optimality condition to the strongly concave dual objective $D(\theta)$, yielding:

$$\mathrm{tr}(\nabla D(\theta^*)^\top(\theta^* - \theta)) \geq 0. \tag{18}$$

Combining this with inequality (16), we obtain:

$$\|\theta - \theta^*\|_2 \leq \sqrt{D(\theta^*) - D(\theta))}. \tag{19}$$

Under the assumption of strong duality, which ensures $P(b) \geq D(\theta^*)$, we replace the intractable term with a computable surrogate:

$$\|\theta - \theta^*\|_2 \leq \sqrt{2(P(b) - D(\theta))}. \tag{20}$$

This concludes the proof.

Based on the upper bound derived in Corollary 1, we substitute the quantity $\|\theta - \theta^*\|_2$ in the right-hand side of Inequality (15). This leads to an improved safe screening condition given by:

$$\|X_{I_i}^\top \theta\|_2 + \|X_{I_i}\|_2 \sqrt{2G(b,\theta)} < \lambda_{|\mathcal{A}^*|} \Rightarrow b^*_{I_i} = 0. \tag{21}$$

The duality gap $G(b,\theta)$ can be efficiently evaluated using the primal-dual variables b and θ, both of which are directly available during each iteration of standard proximal gradient methods.

As training proceeds and the gap between the primal and dual objectives narrows, the upper estimate of $\|X_{I_i}^\top \theta^*\|_2$ is reduced accordingly. This results in a progressively tighter screening threshold over time.

2.4 Lower bound for the Right Term

In contrast, the upper limit of the screening region aligns with the minimal value of $\lambda_{|\mathcal{A}^*|}$. Therefore, our objective in this section is to derive a sharp lower estimate for this critical quantity.

To effectively compute this bound amidst numerous unspecified hyperparameters, we design an iterative scheme that tackles the challenges introduced by the non-separability of the penalty term. This is achieved by exploiting the unknown order structure embedded in the primal solution of (14). In its general form, our screening criterion can be formulated as:

$$\|X_{I_i}^\top \theta\|_2 + \|X_{I_i}\|_2 \sqrt{2G(b,\theta)} < \lambda_{|\mathcal{A}|} \Rightarrow b^*_{I_i} = 0. \tag{22}$$

At the initial stage, we assume all m groups are active, and hence the screening is applied with respect to λ_m:

$$\|X_{I_i}^\top \theta\|_2 + \|X_{I_i}\|_2 \sqrt{2G(b,\theta)} < \lambda_m \Rightarrow b_{I_i}^* = 0. \tag{23}$$

As training proceeds and only m_k groups remain active, the set $\mathcal{A}$ is updated to size m_k. The remaining $m - m_k$ groups can then be assigned any permutation of $\lambda_{m_k+1}, \ldots, \lambda_m$—the smallest parameters—without affecting the final result. This reveals that the ranks of the $m - m_k$ zero-valued coefficients are deterministically among the lowest values in the λ sequence. Accordingly, the screening test is adapted and evaluated at λ_{m_k}:

$$\|X_{I_i}^\top \theta\|_2 + \|X_{I_i}\|_2 \sqrt{2G(b,\theta)} < \lambda_{m_k} \Rightarrow b_{I_i}^* = 0, \tag{24}$$

yielding an updated active group set $\mathcal{A}$ of size m_k', where $m_k' \leq m_k$. The $m_k - m_k'$ newly deactivated groups are again assigned the next smallest unused λ values.

This iterative refinement continues until convergence—i.e., when the active set stabilizes. Crucially, each iteration involves only a single hyperparameter, ensuring computational efficiency even when Group SLOPE involves a large number of tuning parameters.

Throughout this process, as the active set $\mathcal{A}$ shrinks and due to the monotonicity of λ, the corresponding $\lambda_{|\mathcal{A}|}$ increases. This raises the lower bound of $\lambda_{|\mathcal{A}^*|}$ and in turn, enlarges the screening threshold.

In summary, by jointly updating the upper and lower bounds across iterations, the screening region continuously expands, enabling more inactive groups to be excluded and improving overall algorithmic efficiency.

For the computation of the screening rule, the dual f_i^* for Group SLOPE can also be calculated as $f_i^*(\theta_i) = \frac{1}{2}\theta_i^2 + \theta_i y_i$. Using this screening rule, inactive feature groups can be eliminated during the training process.

3 Proposed Algorithms

In this section, we begin by applying the safe screening rules to proximal gradient algorithms, specifically focusing on the APGD algorithm for batch settings and the SPGD algorithm for stochastic settings. We then delve into the theoretical analysis of our unified safe screening rules, highlighting their properties in terms of safeness, convergence, and screening capability.

3.1 Algorithms

Coordinate descent and block coordinate descent methods are efficient for solving Lasso and Group Lasso problems. However, due to the challenges posed by the non-separable penalty, these algorithms are highly efficient but not practical for solving Group SLOPE. To address this issue, accelerated proximal gradient descent (APGD) methods have been proposed, as seen in works like [10, 16].

Since Group SLOPE is generally used in high-dimensional settings, all the proximal algorithms mentioned above face significant computational and memory challenges when dealing with large feature sizes. Therefore, accelerating the training of Group SLOPE through the use of screening techniques for proximal algorithms becomes both important and promising.

For the APGD algorithm in batch settings, our approach involves repeatedly performing the screening test and updating the active set $\mathcal{A}$. If $\mathcal{A}$ is updated during this iteration, we set the step size $t_k = t_1$. From this point onward, the procedure mirrors that of the original APGD algorithm, using the current active set $\mathcal{A}$. This process is detailed in Algorithm 1.

Algorithm 1 APGD Algorithm with Our Safe Screening Rules

Input: $b^0, \hat{b}^1 = b^0, t_1 = 1$
for $k = 1, 2, \ldots$ **do**
  repeat
    Apply the safe screening from (22)
    Update the active group set $\mathcal{A}$
  until $\mathcal{A}$ remains unchanged
  if $\mathcal{A}$ was updated **then**
    $t_k = t_1$
  end if
  $b^k = prox_{t_k,\lambda}(\hat{b}^k - t_k \nabla F(\hat{b}^k))$
  $t_{k+1} = \frac{1}{2}(1 + \sqrt{1 + 4t_k^2})$
  $\hat{b}^{k+1} = b^k + \frac{t_k - 1}{t_{k+1}}(b^k - b^{k-1})$
end for
Output: Coefficient b

Moreover, as each update in Algorithm 1 relies on all samples, the per-iteration cost of the APGD algorithm can be substantial in large-scale learning because it necessitates full gradient computations. To mitigate this, the stochastic proximal gradient descent (SPGD) algorithm, as introduced in [38] and building on [20], serves as an efficient alternative in the stochastic setting, requiring only mini-batch gradient calculations.

In applying our screening rule to the SPGD algorithm for stochastic settings, we similarly repeat the screening test and update $\mathcal{A}$ in the outer loop before proceeding with the standard SPGD algorithm steps using the newly obtained active set. This procedure is outlined in Algorithm 2.

Interestingly, the duality gap, which represents the main time-consuming aspect of our screening rule, has already been computed in the original APGD and SPGD algorithms. Furthermore, as inactive variables are continually screened during optimization, and given that the active set size for iteration k is d_k, the computational complexity of the screening rule for this iteration is only $O(d_k)$, which is even less than the complexity of the original stopping criterion evaluation $O(d)$. Consequently, the complexity $O(d_k(n + \log d_k))$ and $O(d_k(n + Tl + T \log d_k))$ for each iteration of the APGD and SPGD algorithms, respectively, can be reduced to $O(d_k)$ for analysis purposes.

Algorithm 2 SPGD Algorithm with Our Safe Screening Rules

Input: b^0, l.
for $k = 1, 2, \ldots$ **do**
  repeat
    Apply the safe screening from (22)
    Update the active group set $\mathcal{A}$
  until $\mathcal{A}$ remains unchanged
  $b = b^{k-1}$
  $\tilde{v} = \nabla F(b)$
  $\tilde{b}^0 = b$
  for $t = 1, 2, \ldots, T$ **do**
    Pick mini-batch $I_t \subseteq X$ of size l
    $v_t = (\nabla F_{I_t}(\tilde{b}^{t-1}) - \nabla F_{I_t}(b))/l + \tilde{v}$
    $\tilde{b}^t = prox_{\gamma,\lambda}(\tilde{b}^{t-1} - \gamma v_t)$
  end for
  $b^k = \tilde{b}^T$
end for
Output: Coefficient b

We further examine the overall complexity of our algorithms, focusing on both the per-iteration cost and the number of iterations. For per-iteration cost, if the algorithm has d_k active variables at iteration k, our Algorithm 1 requires only $O(d_k(n+\log d_k))$, which is less than the original APGD algorithm's complexity of $O(d(n + \log d))$. Regarding the number of iterations, since the optimal solution for the inactive features screened at iteration k must be zeros, removing these inactive features beforehand either keeps the objective function the same or decreases it. Thus, our Algorithm 1 will converge to the same stopping criterion with at most the same (and usually fewer) iterations compared to the original APGD algorithm. With fewer or the same iterations and lower per-iteration costs, our proposed algorithm is more efficient. Similarly, our Algorithm 2 requires $O(d_k(n + Tl + T \log d_k))$ for the main loop k, whereas the original SPGD algorithm's complexity is $O(d(n + Tl + T \log d))$. Since our algorithm also requires fewer or the same iterations and lower per-iteration costs, the overall complexity is reduced compared to the original SPGD algorithm.

More precisely, the computational advantage of our methods hinges on the sparsity of the final model. The training process gains more from the screening rule when dealing with sparser models. In cases where $n < d$, the final model becomes very sparse, leading to $d_k \ll d$ during training. Consequently, our method is particularly well-suited for high-dimensional settings. It is evident that the proposed algorithms consistently outperform the original ones in terms of speed. Additionally, it's important to note that our method also applies to datasets where $n > d$. Assuming the presence of sparsity, our screening rule will effectively identify inactive features, enabling our algorithms to determine the final active set in a finite number of iterations. As a result, we still achieve $d_k < d$ during training, and the per-iteration cost of our algorithm remains lower than the original one. Therefore, with fewer or an equivalent number of iterations and reduced per-iteration costs, the proposed algorithms remain faster than the original ones for $n > d$.

The following part provides the theoretical analysis of our screening rules, highlighting their safeness, convergence, and screening ability.

Property 1. (Safeness) The proposed screening rule retains all relevant groups throughout the entire optimization trajectory of Group SLOPE, irrespective of the specific iterative method employed.

Property 2. (Convergence) Our screening rule can be seamlessly embedded into a wide range of iterative algorithms, such as APGD, SPGD, and their derivatives, without disrupting convergence guarantees.

Theorem 1. *Let i denote any group that belongs to the final active set $\mathcal{A}^*$. Then $\|X_{I_i}^\top \theta^*\|_2 \in [\min_{j\in\mathcal{A}^*} \lambda_j, \max_{j\in\mathcal{A}^*} \lambda_j]$. As algorithm Ψ converges, there exists a finite iteration number $K_0 \in \mathbb{N}$ s.t. $\forall k \geq K_0$, any group $i \notin \mathcal{A}^*$ will be successfully discarded by the screening rule.*

Proof. From the strong concavity of the dual problem, it follows that the optimal dual variable θ^* is unique. Moreover, θ converges to θ^* as b converges to b^*. Thus, for any given $\epsilon > 0$, one can find an index K_0 s.t. $\forall k \geq K_0$: $\|\theta^k - \theta^*\|_2 \leq \epsilon$, $\sqrt{2G(b^k, \theta^k)} \leq \epsilon$. Then, for any group $i \notin \mathcal{A}^*$, we can bound the screening condition as:

$$\begin{aligned} &\|X_{I_i}^\top \theta^k\|_2 + \|X_{I_i}\|_2 \sqrt{2G(b^k, \theta^k)} \\ \leq\ &\|X_{I_i}\|_2 \|\theta^k - \theta^*\|_2 + \|X_{I_i}^\top \theta^*\|_2 + \|X_{I_i}\|_2 \sqrt{2G(b^k, \theta^k)} \\ \leq\ &2\|X_{I_i}\|_2 \epsilon + \|X_{I_i}^\top \theta^*\|_2. \end{aligned} \tag{25}$$

Since $i \notin \mathcal{A}^*$ implies $\lambda_{|\mathcal{A}^*|} - \|X_{I_i}^\top \theta^*\|_2 > 0$, choosing $\epsilon < \frac{\lambda_{|\mathcal{A}^*|} - \|X_{I_i}^\top \theta^*\|_2}{2\|X_{I_i}\|_2}$, ensures $\|X_{I_i}^\top \theta^k\|_2 + \|X_{I_i}\|_2 \sqrt{2G(b^k, \theta^k)} < \lambda_{|\mathcal{A}^*|}$, which triggers our screening condition.

Theorem 1 highlights the strong screening performance of the proposed rules. As the iterative solver progresses and the duality gap narrows, the rule becomes increasingly effective: the upper bound on the left-hand side tightens, while the right-hand side's lower bound increases. This progressively improves the chances of filtering out inactive groups. Ultimately, every group $i \notin \mathcal{A}^*$ will be accurately screened and discarded in a finite number of iterations.

4 Experiments

In this section, we outline our experimental setup and subsequently present the results and discussions.

4.1 Experimental Setup

Design of Experiments We empirically evaluated our method on real-world benchmark datasets under the Group SLOPE framework, highlighting its computational advantages and its capability to reliably discard irrelevant groups.

Table 2: The descriptions of benchmark datasets used in our experiments.

Dataset	**Sample size**	**Attribute**
Duke Breast Cancer (DBC)	44	7129
Colon Cancer (CC)	62	2000
IndoorLoc (IL)	21048	520
SenseIT Vehicle (SV)	78823	100

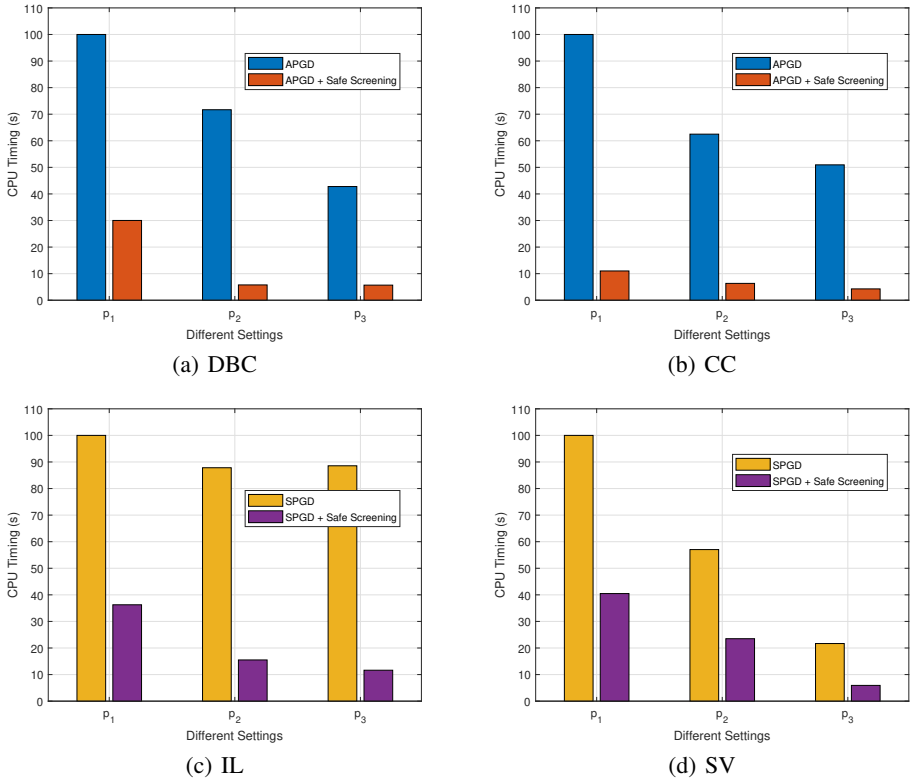
To assess the efficiency of our algorithms in reducing computation time, we compared the runtime of our proposed algorithms against other competitive algorithms for solving Group SLOPE under various conditions. Given that the APGD algorithm is well-suited for scenarios where $n \ll p$ in the batch setting, and the SPGD algorithm is tailored for large-scale learning where n is large in stochastic settings, we evaluated the runtime across both batch and stochastic setups using different datasets. The algorithms compared in batch and stochastic settings are summarized as follows:

- Batch setting
  - APGD: Accelerated proximal gradient descent algorithms as presented in [10, 16].
  - APGD + Screening: Accelerated proximal gradient descent algorithms enhanced with our safe screening rules.
- Stochastic setting
  - SPGD: Stochastic proximal gradient descent algorithm adopted from [38].
  - SPGD + Screening: Stochastic proximal gradient descent algorithm integrated with our safe screening rules.

To further confirm the effectiveness of our algorithms in filtering inactive variables, we evaluated the screening rate at each iteration of the algorithms with our screening rules applied to Group SLOPE, tested in both batch and stochastic setups across various datasets during the training process.

Datasets Table 2 provides an overview of the benchmark datasets utilized in our experiments. Duke Breast Cancer, Colon Cancer, and SenseIT Vehicle datasets are from the LIBSVM repository [11], which can be accessed at `https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. IndoorLoc dataset is obtained from the UCI benchmark repository [12], available at `https://archive.ics.uci.edu/ml/datasets.php`. IndoorLoc dataset includes 2 tasks: IndoorLoc Latitude and IndoorLoc Longitude.

Implementation Details We implemented all the algorithms using MATLAB and compared the average CPU time across different algorithms on a 2.70 GHz machine over 5 trials. We adhered to the basic setup described in [10] and set the tolerance for the duality gap and the dual infeasibility to 10^{-6}. To ensure fairness in comparisons,



(a) DBC (b) CC

(c) IL (d) SV

Fig. 1: Running time of the algorithms without and with safe screening for Group SLOPE.

the experimental configurations for Algorithm 1 and 2 followed the original APGD and SPGD algorithms, maintaining consistent hyperparameters across all setups. In the stochastic setting, the mini-batch size and the number of inner loop iterations were set to 30 or 40, depending on the dataset. The step size γ was selected from a range of 10^{-8} to 10^{-5}. Initially, the APGD algorithms exhibited a large duality gap, offering minimal benefit from our screening rule, so we first ran the algorithms without screening and later applied our screening rule with a warm start. For ease of comparison, the CPU time of each algorithm is presented as a percentage relative to the runtime of the first configuration for each dataset.

The OSCAR hyperparameter setting, which is commonly used (see [1, 27, 40, 42]), was applied in all our experiments:

$$\lambda_i = \alpha_1 + \alpha_2(m - i), \tag{26}$$

where $\alpha_1 = p_i \|X^\top y\|_\infty$ and $\alpha_2 = \alpha_1/d$ for Group SLOPE. For a fair comparison, the factor p_i is used to control sparsity. In our experiments, we set $p_i = i * e^{-\tau}$, $i = 1, 2, 3$.

For Group SLOPE, batch algorithms were applied to the Duke Breast Cancer and Colon Cancer datasets, while stochastic algorithms were run on the SenseIT and Indoor-

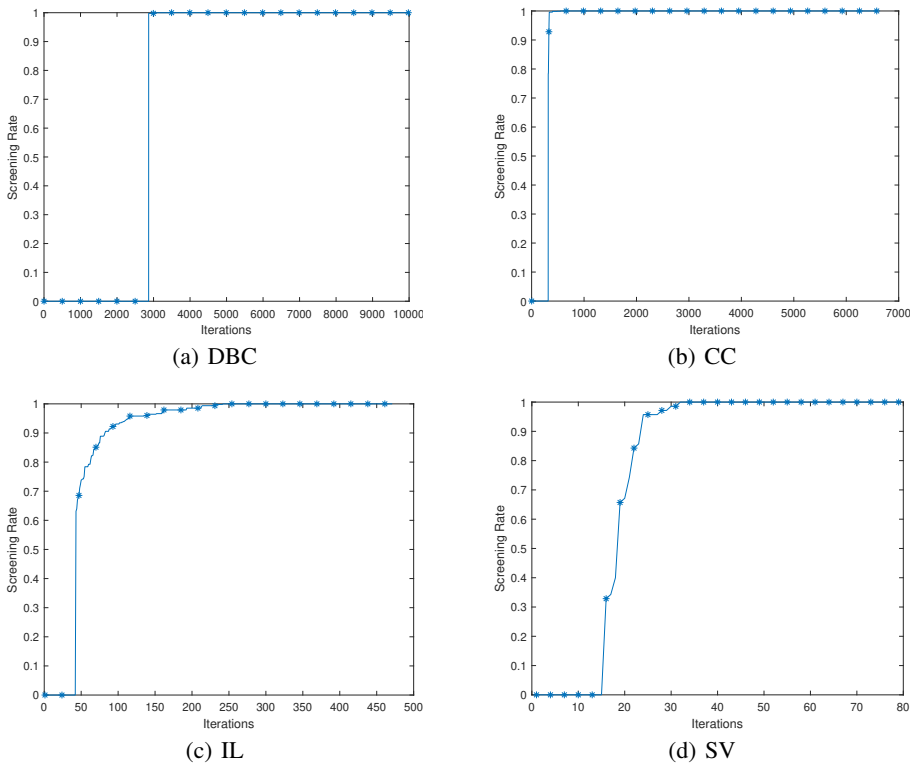


Fig. 2: Screening rate of our screening rule in both batch and stochastic settings for Group SLOPE.

Loc Longitude datasets, with τ set to 2 for IndoorLoc Longitude and 3 for the other datasets. We duplicate each feature i to form the feature group I_i with size $|I_i| \sim U(1, s)$ where U is the discrete uniform distribution and choose $s = 10$ for Duke Breast Cancer and Colon Cancer datasets and $s = 40$ for IndoorLoc Latitude and SenseIT Vehicle datasets.

To assess the screening rate of our algorithms, it was calculated as the proportion of inactive partitions of groups screened by our method to the total number of inactive groups at the optimal solution. We used the p_1 setting for this evaluation.

4.2 Experimental Results and Discussions

Figures 1(a)–(d) present the average runtime comparisons of the proposed algorithms with and without the safe screening technique applied to Group SLOPE, under both batch and stochastic optimization frameworks across multiple experimental settings. In cases where the sample size is much smaller than the number of features ($n \ll d$), the APGD method achieves acceleration ratios between 3× and 14× when integrated with our screening approach. For large-scale problems, incorporating the screening rule

into the SPGD variant leads to speed improvements ranging from 2.5× to 8× compared to its unscreened counterpart. These observations consistently validate the substantial efficiency gains attained by augmenting Group SLOPE solvers—both batch and stochastic—with our screening mechanism. The primary contributor to this acceleration is the effective early-stage exclusion of inactive feature groups, which lowers the computational complexity throughout training. Notably, the improvements become even more pronounced as the data dimensionality increases and sparsity intensifies.

Figures 2(a)-(d) depict the screening rates of our algorithms in both batch and stochastic settings for Group SLOPE, highlighting the effectiveness and characteristics of our screening rule. The data support the conclusion that our algorithm can successfully eliminate most inactive feature groups at an early stage, converge on the final active set, and ultimately screen nearly all inactive features within a finite number of iterations. This effectiveness is due to the tight upper and lower bounds of the screening test's left and right terms, respectively, which allow for more efficient screening of inactive variables as the optimization algorithm progresses. Specifically, as the algorithm converges, the duality gap narrows, leading to a continuously decreasing upper bound for the left term of the screening test, while the iterative strategy increasingly solidifies the order structure of variables, thus continuously increasing the lower bound for the right term.

5 Conclusion

In this paper, we introduced a safe variable screening rule for Group SLOPE, addressing the challenges posed by the non-separable group effects. This approach significantly speeds up the training process by eliminating unnecessary computations for inactive variables. Our screening rule is uniquely dynamic, featuring a decreasing left term through tracking the intermediate duality gap, and an increasing right term by iteratively assessing the order of the primal solution, considering the unknown order structure. Importantly, the proposed rules are seamlessly integrable into existing iterative optimization methods, applicable in both batch and stochastic settings, such as the APGD and SPGD algorithms. We have theoretically proven that our screening rule remains safe when applied to these algorithms, ensuring no loss in accuracy. Extensive empirical results on real-world benchmark datasets demonstrate that our algorithms provide substantial computational benefits while maintaining accuracy in both batch and stochastic learning contexts by effectively screening out inactive variables.

Acknowledgement

This work is supported in part by the National Science Foundation (NSF) grant IIS-2451436 and Commonwealth Cyber Initiative grant HC-4Q24-059.

References

1. Bao, R., Gu, B., Huang, H.: Efficient approximate solution path algorithm for order weight l_1-norm with accuracy guarantee. In: 2019 IEEE International Conference on Data Mining (ICDM). pp. 958–963. IEEE (2019)

2. Bao, R., Gu, B., Huang, H.: Fast oscar and owl regression via safe screening rules. In: International conference on machine learning. pp. 653–663. PMLR (2020)
3. Bao, R., Gu, B., Huang, H.: An accelerated doubly stochastic gradient method with faster explicit model identification. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management. pp. 57–66 (2022)
4. Bao, R., Lu, Q., Zhang, Y.: Safe screening rules for group owl models. arXiv preprint arXiv:2504.03152 (2025)
5. Bao, R., Wu, X., Xian, W., Huang, H.: Doubly sparse asynchronous learning. In: The 31st International Joint Conference on Artificial Intelligence (IJCAI 2022) (2022)
6. Bauschke, H.H., Combettes, P.L., et al.: Convex analysis and monotone operator theory in Hilbert spaces, vol. 408. Springer (2011)
7. Bergersen, L.C., Glad, I.K., Lyng, H.: Weighted lasso with data integration. Statistical applications in genetics and molecular biology **10**(1) (2011)
8. Bogdan, M., Van Den Berg, E., Sabatti, C., Su, W., Candès, E.J.: Slope—adaptive variable selection via convex optimization. The annals of applied statistics **9**(3), 667–698 (2015)
9. Bonnefoy, A., Emiya, V., Ralaivola, L., Gribonval, R.: Dynamic screening: Accelerating first-order algorithms for the lasso and group-lasso. IEEE Transactions on Signal Processing **63**(19), 5121–5132 (2015)
10. Brzyski, D., Gossmann, A., Su, W., Bogdan, M.: Group slope–adaptive selection of groups of predictors. Journal of the American Statistical Association **114**(525), 419–433 (2019)
11. Chang, C.C., Lin, C.J.: Libsvm: A library for support vector machines. ACM transactions on intelligent systems and technology (TIST) **2**(3), 1–27 (2011)
12. Dua, D., Graff, C.: UCI machine learning repository (2017)
13. Fercoq, O., Gramfort, A., Salmon, J.: Mind the duality gap: safer rules for the lasso. In: International Conference on Machine Learning. pp. 333–342 (2015)
14. Frankle, J., Carbin, M.: The lottery ticket hypothesis: Finding sparse, trainable neural networks. In: International Conference on Learning Representations (2018)
15. Gossmann, A., Cao, S., Brzyski, D., Zhao, L.J., Deng, H.W., Wang, Y.P.: A sparse regression method for group-wise feature selection with false discovery rate control. IEEE/ACM transactions on computational biology and bioinformatics **15**(4), 1066–1078 (2017)
16. Gossmann, A., Cao, S., Wang, Y.P.: Identification of significant genetic variants via slope, and its extension to group slope. In: Proceedings of the 6th ACM Conference on Bioinformatics, Computational Biology and Health Informatics. pp. 232–240 (2015)
17. Han, S., Pool, J., Tran, J., Dally, W.: Learning both weights and connections for efficient neural network. Advances in neural information processing systems **28** (2015)
18. Jacob, L., Obozinski, G., Vert, J.P.: Group lasso with overlap and graph lasso. In: Proceedings of the 26th annual international conference on machine learning. pp. 433–440 (2009)
19. Jenatton, R., Audibert, J.Y., Bach, F.: Structured variable selection with sparsity-inducing norms. The Journal of Machine Learning Research **12**, 2777–2824 (2011)
20. Johnson, R., Zhang, T.: Accelerating stochastic gradient descent using predictive variance reduction. In: Advances in neural information processing systems. pp. 315–323 (2013)
21. Johnson, T., Guestrin, C.: Blitz: A principled meta-algorithm for scaling sparse optimization. In: International Conference on Machine Learning. pp. 1171–1179 (2015)
22. Kim, S., Xing, E.P.: Tree-guided group lasso for multi-task regression with structured sparsity. In: Proceedings of the 27th International Conference on International Conference on Machine Learning. pp. 543–550 (2010)
23. Larsson, J., Bogdan, M., Wallin, J.: The strong screening rule for slope. Advances in neural information processing systems **33**, 14592–14603 (2020)
24. Laurent El Ghaoui, Vivian Viallon, T.R.: Safe feature elimination in sparse supervised learning. Pacific Journal of Optimization **8**, 667–698 (2012)

25. Lu, L., Wang, Z., Bao, R., Wang, M., Li, F., Wu, Y., Jiang, W., Xu, J., Wang, Y., Gao, S.: All-in-one tuning and structural pruning for domain-specific llms. arXiv preprint arXiv:2412.14426 (2024)
26. Ndiaye, E., Fercoq, O., Gramfort, A., Salmon, J.: Gap safe screening rules for sparse-group lasso. In: Advances in Neural Information Processing Systems. pp. 388–396 (2016)
27. Oswal, U., Cox, C., Lambon-Ralph, M., Rogers, T., Nowak, R.: Representational similarity learning with application to brain networks. In: International Conference on Machine Learning. pp. 1041–1049 (2016)
28. Rakotomamonjy, A., Gasso, G., Salmon, J.: Screening rules for lasso with non-convex sparse regularizers. In: International Conference on Machine Learning. pp. 5341–5350 (2019)
29. Shibagaki, A., Karasuyama, M., Hatano, K., Takeuchi, I.: Simultaneous safe screening of features and samples in doubly sparse modeling. In: International Conference on Machine Learning. pp. 1577–1586 (2016)
30. Simon, N., Friedman, J., Hastie, T., Tibshirani, R.: A sparse-group lasso. Journal of computational and graphical statistics **22**(2), 231–245 (2013)
31. Tibshirani, R., Bien, J., Friedman, J., Hastie, T., Simon, N., Taylor, J., Tibshirani, R.J.: Strong rules for discarding predictors in lasso-type problems. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **74**(2), 245–266 (2012)
32. Wang, J., Ye, J.: Two-layer feature reduction for sparse-group lasso via decomposition of convex sets. In: Advances in Neural Information Processing Systems. pp. 2132–2140 (2014)
33. Wang, J., Ye, J.: Multi-layer feature reduction for tree structured group lasso via hierarchical projection. In: Advances in Neural Information Processing Systems. pp. 1279–1287 (2015)
34. Wang, J., Zhou, J., Liu, J., Wonka, P., Ye, J.: A safe screening rule for sparse logistic regression. In: Advances in neural information processing systems. pp. 1053–1061 (2014)
35. Wang, J., Zhou, J., Wonka, P., Ye, J.: Lasso screening rules via dual polytope projection. In: Advances in neural information processing systems. pp. 1070–1078 (2013)
36. Wu, X., Gao, S., Zhang, Z., Li, Z., Bao, R., Zhang, Y., Wang, X., Huang, H.: Auto-train-once: Controller network guided automatic network pruning from scratch. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16163–16173 (2024)
37. Xiang, Z.J., Wang, Y., Ramadge, P.J.: Screening tests for lasso problems. IEEE transactions on pattern analysis and machine intelligence **39**(5), 1008–1027 (2016)
38. Xiao, L., Zhang, T.: A proximal stochastic gradient method with progressive variance reduction. SIAM Journal on Optimization **24**(4), 2057–2075 (2014)
39. Yuan, M., Lin, Y.: Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology) **68**(1), 49–67 (2006)
40. Zhang, D., Wang, H., Figueiredo, M., Balzano, L.: Learning to share: Simultaneous parameter tying and sparsification in deep learning (2018)
41. Zhao, P., Rocha, G., Yu, B.: The composite absolute penalties family for grouped and hierarchical variable selection. The Annals of Statistics **37**(6A), 3468–3497 (2009)
42. Zhong, L.W., Kwok, J.T.: Efficient sparse modeling with automatic feature grouping. IEEE transactions on neural networks and learning systems **23**(9), 1436–1447 (2012)