GPU-ACCELERATED BRANCH-AND-BOUND AND CONDITIONAL SCREENING FOR SPARSE LOGISTIC REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Cardinality-constrained logistic regression selects a small set of predictors, but optimizing over feature subsets requires many closely related continuous solves. We develop a branch-and-bound framework that batches ridge-logistic subproblems on a GPU and uses gradient residuals to obtain valid lower bounds from inexact solves. A complementary analysis of conditional greedy screening gives a finite-sample support-containment guarantee with a controlled number of competitive irrelevant selections. The analysis accommodates signals that become detectable only after conditioning and separates the statistical screening guarantee from the deterministic bounds used for pruning. Synthetic experiments show faster support recovery for conditional screening than for marginal screening in a correlated suppressor design. On two classification datasets, the GPU implementation attains smaller terminal optimality gaps than the tested CPU baselines and lower batch runtimes for moderate and large batch sizes. These results support batched GPU computation for exact sparse classification.

1 INTRODUCTION

Sparse classifiers restrict prediction to a small set of covariates, making the fitted model easier to inspect and reducing the number of inputs required at deployment. Logistic regression provides a natural setting for studying this trade-off: the loss is smooth, yet direct control of the number of selected features introduces a discrete optimization problem. Convex penalties and local-search methods offer efficient approximations, while mixed-integer optimization can additionally quantify the remaining global optimality gap (Friedman et al., 2010; Dedieu et al., 2021; Hazimeh et al., 2023).

Let $\boldsymbol{X} \in \mathbb{R}^{n\times p}$ have rows $\boldsymbol{x}_i^\top$, and let $\boldsymbol{y} \in \{0,1\}^n$. For a ridge parameter $\lambda > 0$ and a sparsity budget $k \in [p] := \{1,\dots,p\}$, we study

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^p:\ \|\boldsymbol{\theta}\|_0\le k} f_n(\boldsymbol{\theta}), \qquad f_n(\boldsymbol{\theta}) = \frac{1}{n}\sum_{i=1}^{n} \log\Big(1 + e^{(1-2y_i)\boldsymbol{x}_i^\top\boldsymbol{\theta}}\Big) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2. \tag{1}$$

The cardinality constraint controls support size directly, without an additional ℓ_1 penalty. The estimator is still shrunk by the ridge term. For every fixed support, the objective is λ-strongly convex and has a unique minimizer. The computational difficulty lies in searching over supports, not in the individual continuous problems. We formulate the model without an intercept for notational simplicity. An unpenalized intercept can be profiled out: partial minimization over the intercept preserves convexity, so the profiled objective remains λ-strongly convex in $\boldsymbol{\theta}$ after adding the ridge term; Appendix E gives the corresponding certificate.

Branch-and-bound repeatedly solves problems that share the same design matrix and differ mainly in their permitted features. Treating these problems as unrelated CPU tasks leaves this structure underused. We instead batch their coefficients and feature masks, converting repeated matrix–vector operations into matrix–matrix operations on a GPU. Fast subproblem evaluation also makes it less expensive to compare candidate branches. This approach requires a careful distinction between an

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

approximate primal objective and a valid node lower bound: the former is not, on its own, suitable for pruning.

1.1 Related work

Sparse estimation and classification. The lasso and elastic net promote sparsity through convex regularization (Tibshirani, 1996; Zou & Hastie, 2005); coordinate descent efficiently computes generalized-linear-model regularization paths (Friedman et al., 2010). Adaptive and folded-concave penalties address aspects of shrinkage and selection (Zou, 2006; Fan & Li, 2001; Zhang, 2010). Coordinate descent combined with combinatorial local search also scales ℓ_0-regularized learning to large problems (Hazimeh & Mazumder, 2020; Hazimeh et al., 2023). Adaptive best-subset search for generalized linear models is also available through the `abess` framework (Zhu et al., 2022). These methods solve different formulations or provide local rather than global guarantees for (1).

Screening and greedy sparse logistic methods. Sure independence screening (SIS) ranks marginal utilities to reduce the candidate feature set (Fan & Lv, 2008; Fan & Song, 2010). Iterative screening already addresses signals that become visible after conditioning (Fan et al., 2009). Greedy sparse optimization has also been studied directly for logistic models. Logit-GOMP extends orthogonal matching pursuit to logistic regression and gives nonasymptotic variable-identification conditions (Lozano et al., 2011). GraSP treats general sparsity-constrained objectives under restricted-curvature conditions and includes ridge-logistic examples (Bahmani et al., 2013); gradient hard-thresholding pursuit develops related guarantees for sparsity-constrained convex optimization (Yuan et al., 2014); and the greedy projected gradient–Newton method studies support identification and local convergence for sparse logistic regression (Wang et al., 2020). Our contribution is not the introduction of conditional refitting. We analyze a particular ridge-logistic gradient rule and quantify its support hitting time in terms of a common exceptional set of competitive irrelevant variables. Unlike recovery or optimization guarantees for a fixed sparsity level, our result bounds the first time at which the monotone forward path contains the full ridge-population support, while allowing up to M_n irrelevant additions before containment. The result explicitly separates sampling error from population score separation.

Exact sparse optimization. Modern mixed-integer formulations have renewed interest in best subset selection (Bertsimas et al., 2016). Logistic feature selection and sparse classification have been studied using mixed-integer optimization and integrality generation (Sato et al., 2016; Dedieu et al., 2021); first-order branch-and-bound methods further improve scalability for sparse regression (Hazimeh et al., 2022). Recent specialized methods include OKRidge for optimal sparse ridge regression (Liu et al., 2023), the distributed sparse-convex MINLP framework SCOT (Olama et al., 2023), and a perspective-relaxation first-order method for certifying optimal sparse GLMs (Liu et al., 2025). Concurrent GPU work accelerates perspective-relaxation solves (Liu et al., 2026), batches branch-and-bound nodes for optimal sparse GLMs (Liu & Lodi, 2026), and develops GPU nonlinear branch-and-bound for sparse linear models (Meng et al., 2026). Our method instead uses a lightweight masked ridge-logistic relaxation whose strong convexity converts any inexact iterate into a valid residual-corrected lower bound, and it pairs this exact search with the conditional-screening hitting-time analysis above. Our experiments compare the GPU branch-and-bound implementation with CPU branch-and-bound and MOSEK baselines.

1.2 Contributions and scope

We give a finite-sample conditional screening theorem with an explicit allowance for irrelevant selections, formulate a branch-and-bound procedure whose pruning remains valid with inexact masked solves, and develop a batched GPU solver. We use conditional screening to guide refitting and feasible-candidate construction, while deterministic optimization bounds provide global certification. This separation lets the statistical screening result strengthen incumbent construction while preserving exact branch-and-bound certification. Detailed assumptions, proofs, algorithmic safeguards, and experiment summaries appear in the appendices.

Notation. For $A \subseteq [p]$, $\boldsymbol{v}_A$ is a subvector, $\boldsymbol{M}_{A,B}$ is a submatrix, and $A^c = [p] \setminus A$. We write $\mathrm{supp}(\boldsymbol{v})$ for the support, $\|\cdot\|_2$ for the Euclidean or spectral norm, and $\|\cdot\|_{\psi_2}$ for the sub-Gaussian

norm. Inequalities between symmetric matrices use the positive-semidefinite order. All logarithms are natural.

2 CONDITIONAL GREEDY SCREENING

2.1 POPULATION TARGET AND ASSUMPTIONS

Let $(\boldsymbol{x}_i, y_i)_{i=1}^n$ be independent copies of $(\tilde{\boldsymbol{x}}, \tilde{y})$, and define

$$f(\boldsymbol{\theta}) = \mathbb{E}\Big[\log\Big(1 + e^{(1-2\tilde{y})\tilde{\boldsymbol{x}}^\top \boldsymbol{\theta}}\Big)\Big] + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2, \qquad \boldsymbol{\theta}^\star = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} f(\boldsymbol{\theta}), \quad S^\star = \mathrm{supp}(\boldsymbol{\theta}^\star). \tag{2}$$

In this section, $k = |S^\star| \geq 1$; the optimization budget in later sections is chosen independently. The target is the minimizer of the *ridge-regularized population risk*, not automatically the coefficient vector of a correctly specified unpenalized logistic model. Sparsity of this target is an assumption and need not hold for an arbitrary correlated design.

For any $A \subseteq [p]$, let $\boldsymbol{\theta}^A$ and $\hat{\boldsymbol{\theta}}^A$ minimize f and f_n, respectively, over vectors supported on A. For $j \notin A$, define

$$G_j(A) = |\nabla_j f(\boldsymbol{\theta}^A)|, \qquad \hat{G}_j(A) = |\nabla_j f_n(\hat{\boldsymbol{\theta}}^A)|. \tag{3}$$

The greedy rule refits after each selection:

$$A_0 = \varnothing, \qquad j_{t+1} \in \arg\max_{j\notin A_t} \hat{G}_j(A_t), \qquad A_{t+1} = A_t \cup \{j_{t+1}\}, \quad 0 \leq t < d_n. \tag{4}$$

Ties are broken by the smallest index. We use the following sufficient conditions.

(D1) Signal size. For constants $c_\beta, B_\theta > 0$, $\min_{j\in S^\star} |\theta_j^\star| \geq c_\beta$ and $\|\boldsymbol{\theta}^\star\|_2 \leq B_\theta$.

(D2) Sparse directional regularity. For every $B \subseteq [p]$ with $|B| \leq \min\{p, d_n + k\}$ and every unit vector $\boldsymbol{u} \in \mathbb{R}^{|B|}$,

$$\|\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_B\|_{\psi_2} \leq K_x, \qquad \lambda_{\max}\big(\mathbb{E}[\tilde{\boldsymbol{x}}_B \tilde{\boldsymbol{x}}_B^\top]\big) \leq C_\Sigma, \tag{5}$$

where $K_x, C_\Sigma > 0$ do not depend on n, p, B. The directional bound, rather than only coordinatewise tail bounds, is used in the uniform concentration argument.

(D3) A common competition set. For each $j \notin S^\star$, define

$$\Delta_j^{(n)} = \sup_{\substack{|A|\leq d_n,\ A\supseteq S^\star \\ j\notin A}} \sup_{t\in[0,1]} \big\|\nabla^2_{j,A\cup S^\star} f\big(\boldsymbol{\theta}^A + t(\boldsymbol{\theta}^\star - \boldsymbol{\theta}^A)\big)\big\|_2 . \tag{6}$$

For a deterministic threshold $\mu_n \geq 0$, let $\mathcal{N}_n = \{j \notin S^\star : \Delta_j^{(n)} > \mu_n\}$ and $M_n := |\mathcal{N}_n|$. A supremum over an empty index set is zero.

(D4) Budget and sample size. The budget satisfies $k + M_n \leq d_n \leq p$. For a failure probability $\delta \in (0, 1/2)$, set

$$H_n(\delta) = (d_n + k)\log(ep) + \log(8/\delta), \qquad n \geq C_0 H_n(\delta), \tag{7}$$

where C_0 depends only on K_x, C_Σ.

Condition (D3) is uniform over all admissible active sets, which covers the data-dependent path in (4). It is a structural restriction, not a consequence of sub-Gaussian tails. The same exceptional set must work across the path; bounding the number of competitors separately at each iteration would not control the total number of false selections. Fixed constants in (D1) also imply $k \leq B_\theta^2/c_\beta^2$. Growing-support extensions require corresponding signal-dependent constants.

2.2 A SUPPORT HITTING-TIME GUARANTEE

Lemma 1 (Uniform conditional-score concentration)**.** *Under (D2) and* (7)*, a constant C_1, depending only on K_x, C_Σ, λ, gives*

$$\mathbb{P}\left\{ \sup_{|A|\le d_n} \max_{j\notin A} |\hat{G}_j(A) - G_j(A)| \le r_n(\delta) \right\} \ge 1-\delta, \qquad r_n(\delta) = C_1\sqrt{\frac{H_n(\delta)}{n}}. \tag{8}$$

The maximum over no remaining coordinates is zero.

The proof in Appendix A.2 controls restricted gradients with finite nets, bounds estimation error using strong convexity, and then transfers that error to unselected coordinates through the empirical cross-Hessian. A union bound is taken over every admissible active set, so no independence between the selected set and the data is assumed.

For $U = S^\star \setminus A \neq \varnothing$, the population bounds are

$$\max_{j\in U} G_j(A) \ge \lambda c_\beta, \tag{9}$$

$$\|\boldsymbol{\theta}^A - \boldsymbol{\theta}^\star\|_2 \le C_\theta \|\boldsymbol{\theta}^\star_U\|_2, \qquad \max_{j\in (S^\star)^c\setminus(\mathcal{N}_n\cup A)} G_j(A) \le C_\theta B_\theta \mu_n, \tag{10}$$

where $C_\theta = 1 + C_\Sigma/(4\lambda)$. Lemmas 5–7 give the details. The signal bound uses a Schur complement of the integrated Hessian. The noise bound uses the weighted cross-Hessian in (6); ordinary covariance alone is not a valid replacement.

Theorem 2 (Screening with a bounded number of mistakes)**.** *Suppose (D1)–(D4) hold. If*

$$C_\theta B_\theta \mu_n + 2r_n(\delta) < \lambda c_\beta, \tag{11}$$

then $\mathbb{P}\{S^\star \subseteq A_{k+M_n}\} \ge 1-\delta$. Before support containment, every irrelevant selection belongs to $\mathcal{N}_n$ on the event in (8)*.*

To see the mechanism, a missing signal has empirical score at least $\lambda c_\beta - r_n(\delta)$, whereas a nonexceptional irrelevant coordinate has score at most $C_\theta B_\theta \mu_n + r_n(\delta)$. The strict margin excludes the latter from the maximizers. There are only M_n exceptional irrelevant coordinates, so at most M_n such selections can delay the k signals. Appendix A.6 supplies the full argument.

Taking $\delta_n = 1/n$ and $(d_n + k)\log(ep) = o(n)$ gives $r_n(\delta_n) \to 0$. The threshold μ_n need not vanish: any sequence satisfying $C_\theta B_\theta \mu_n \le \lambda c_\beta - \eta$ eventually for some $\eta > 0$, together with $k + M_n \le d_n$, yields the same weak sure-screening conclusion. Thus μ_n may be chosen as large as the score margin permits, which can reduce the exceptional-set size M_n. Appendix A.6 gives the asymptotic statement and discusses its dependence on λ. The block example in Appendix B gives an explicit upper bound on M_n under an additional weighted-Hessian condition.

2.3 COMPARISON WITH MARGINAL SCREENING

SIS uses univariate utilities, while (4) updates its scores after a multivariate refit. The theorem requires at least one detectable missing signal at each unfinished step, allowing signals to become detectable after conditioning rather than requiring marginal detectability of every signal.

The synthetic comparison measures the number of selected variables needed to contain the generating support. Across 30 replicates of an independent design, both methods select all four signals in four steps. In a correlated suppressor design with three signals, the reported mean is 3.4 for greedy screening and 19.2 for SIS; the observed maxima are 6 and 69. Figure 1 also shows medians and ranges. These are descriptive results, not confidence intervals. Appendix F.1 gives the designs and complete summary statistics. For both synthetic designs, the generating support equals the support of the ridge-population target in (2); Appendix F.1 verifies the correlated suppressor case using Gaussian Stein's identity. Thus the reported hitting times also measure containment of $S^\star$ for these two designs.

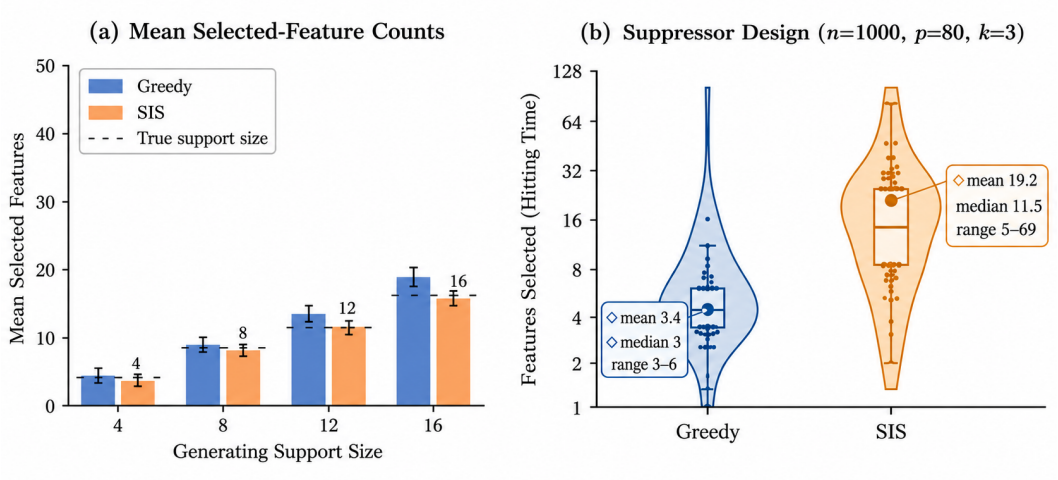


Figure 1: Support hitting times over 30 independent replicates; smaller values indicate faster containment of the generating support. (a) Mean hitting times for the independent and correlated designs; dashed lines mark the corresponding support sizes. (b) For the correlated suppressor design, vertical segments show the observed minimum–maximum ranges, filled circles denote sample means, and open diamonds denote medians. The vertical axis in panel (b) is logarithmic. Only aggregate statistics reported in Appendix F.1 are displayed; the range segments are descriptive summaries rather than confidence intervals.

3 Branch-and-Bound with Valid Inexact Bounds

3.1 Node Representation and Masked Relaxation

A node $N = (\mathcal{F}^+, \mathcal{F}^-)$ fixes feature selectors to one on $\mathcal{F}^+$ and zero on $\mathcal{F}^-$. A selected feature consumes a budget slot but its coefficient may still be zero. Nodes with overlapping sets or $|\mathcal{F}^+| > k$ are infeasible. Branching on a free coordinate i creates $(\mathcal{F}^+ \cup \{i\}, \mathcal{F}^-)$ and $(\mathcal{F}^+, \mathcal{F}^- \cup \{i\})$.

For a valid node, define its permitted set

$$D(N) = \begin{cases} \mathcal{F}^+, & |\mathcal{F}^+| = k, \\ [p] \setminus \mathcal{F}^-, & |\mathcal{F}^+| < k, \end{cases} \qquad \phi_0(N) = \min_{\mathrm{supp}(\boldsymbol{\theta}) \subseteq D(N)} f_n(\boldsymbol{\theta}). \tag{12}$$

This is the masked relaxation solved by the batched gradient method. At a nonterminal node it drops the remaining cardinality restriction. It is not the continuous big-M relaxation, which retains a coupled selector-budget constraint. Appendix C.1 gives the big-M formulation and compares the two bounds.

Permitted sets do not expand along a branch: an exclusion removes a permitted coordinate, while an inclusion leaves the permitted set unchanged until the selected-feature budget is reached, at which point it shrinks to the selected coordinates. Hence exact relaxation values cannot decrease. We therefore rank candidate exclusions by their evaluated lower bounds, as a strong-branching heuristic. At a node with $|\mathcal{F}^+| < k$, the batched candidate pool contains up to $k - |\mathcal{F}^+|$ free coordinates selected by the brancher's memoryless ranking; all corresponding exclusion relaxations are evaluated in one batch, and the largest valid exclusion bound determines the branch variable. A feasible candidate is obtained by retaining $\mathcal{F}^+$, selecting at most $k - |\mathcal{F}^+|$ free coordinates, and refitting on that set. Conditional greedy scores from Section 2 may be used to choose the free coordinates in this completion step. This choice improves incumbent construction while the exact search continues to cover all valid nodes and features. Such a candidate supplies an upper bound even when its refit is unfinished.

3.2 RESIDUAL CERTIFICATES AND PRUNING

An unfinished relaxation solve returns an objective value *above* its optimum. Using that value as a lower bound can incorrectly prune the optimal support. Strong convexity supplies the required correction.

Proposition 3 (A lower bound from a masked gradient)**.** *For a valid node N, let $\boldsymbol{\vartheta}$ be supported on $D = D(N)$ and let $\boldsymbol{g}_D = \nabla_D f_n(\boldsymbol{\vartheta})$. Then*

$$\ell_N(\boldsymbol{\vartheta}) := \max\left\{0,\ f_n(\boldsymbol{\vartheta}) - \frac{\|\boldsymbol{g}_D\|_2^2}{2\lambda}\right\} \le \phi_0(N) \le v(N), \tag{13}$$

where $v(N)$ is the optimal node value. Moreover,

$$0 \le f_n(\boldsymbol{\vartheta}) - \phi_0(N) \le \frac{\|\boldsymbol{g}_D\|_2^2}{2\lambda} \le \frac{|D|\varepsilon_g^2}{2\lambda} \quad \text{if } \|\boldsymbol{g}_D\|_\infty \le \varepsilon_g. \tag{14}$$

Appendix C.2 proves the result by minimizing the quadratic lower model on the permitted coordinates. The formula is a standard strong-convexity consequence; its role here is to make inexact GPU solves usable for global bounding. A child may inherit the maximum of its own certificate and its parent's stored lower bound. Pruning uses these certificates, not raw loss values.

Algorithm 1 maintains the incumbent, the open-node queue, and a lower bound for retired nodes. The last quantity keeps the global certificate valid when nodes are retired at a positive gap tolerance; Appendix D proves this invariant. Feasible candidates remain available throughout the search, including the zero vector at initialization. No feature is permanently removed using the population screening theorem. All statements about valid bounds are in exact arithmetic; Appendix E describes the additional error allowances needed for floating-point certification.

Algorithm 1 Branch-and-bound with valid inexact certificates

Require: $\boldsymbol{X}, \boldsymbol{y}, \lambda > 0$, budget k, absolute gap tolerance $\varepsilon > 0$.
Ensure: A feasible incumbent $\boldsymbol{\theta}^{\mathrm{inc}}$ and valid bounds $LB \le \mathrm{OPT} \le UB$.

$\boldsymbol{\theta}^{\mathrm{inc}} \leftarrow \mathbf{0}$; $UB \leftarrow \log 2$; $L_{\mathrm{closed}} \leftarrow +\infty$.
$Q \leftarrow \{(\varnothing, \varnothing)\}$ with root lower bound 0.
$LB \leftarrow 0$.
while $Q \neq \varnothing$ and $UB - LB > \varepsilon$ **do**
    Select a node N with minimum stored lower bound in Q.
    Compute a masked iterate and certificate by Algorithm 2; raise ℓ_N to the maximum of its old and new bounds.
    Form and evaluate a node-feasible sparse candidate; update $UB, \boldsymbol{\theta}^{\mathrm{inc}}$ if it improves the incumbent.
    if $\ell_N \ge UB - \varepsilon$ **then**
        Retire N: remove it from Q and set $L_{\mathrm{closed}} \leftarrow \min\{L_{\mathrm{closed}}, \ell_N\}$.
    else if $|D(N)| \le k$ **then**
        If $D(N) = \varnothing$, use $\boldsymbol{\vartheta} = \mathbf{0}$; otherwise refine with $\varepsilon_g \le \sqrt{2\lambda\varepsilon/|D(N)|}$ until $f_n(\boldsymbol{\vartheta}) - \ell_N(\boldsymbol{\vartheta}) \le \varepsilon$.
        Update the incumbent and stored bound; retire N and update L_{closed}.
    else
        Form the candidate pool described in Section 3.1; batch all candidate-exclusion solves and select the coordinate with largest valid exclusion bound (smallest index for ties).
        Replace N in Q by both valid children. Each child inherits ℓ_N and any larger valid evaluated bound.
    end if
    $LB \leftarrow \min\{UB, L_{\mathrm{closed}}, \min_{N' \in Q} \ell_{N'}\}$, with an empty minimum equal to $+\infty$.
end while
return $\boldsymbol{\theta}^{\mathrm{inc}}, LB, UB$.

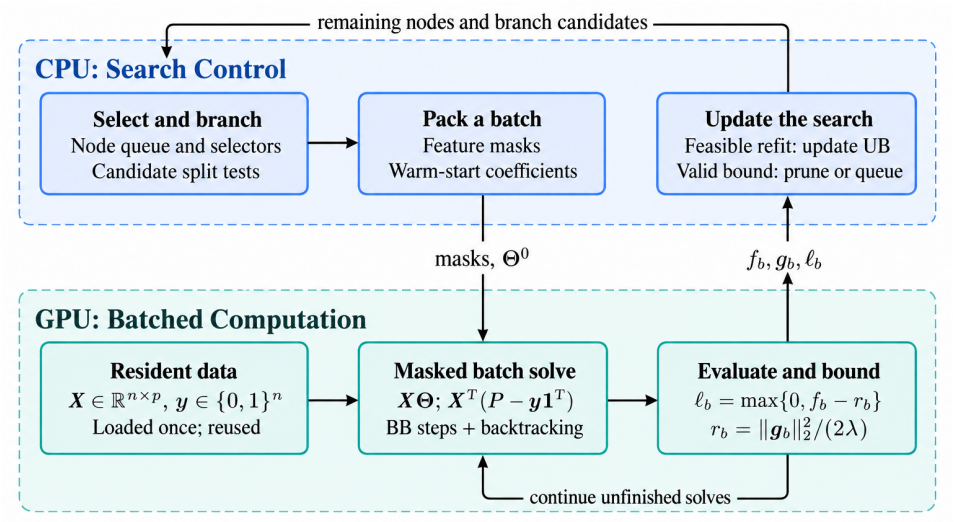


Figure 2: CPU–GPU workflow. The CPU manages selectors, candidate branches, the incumbent, and the node queue. The GPU reuses the design matrix, performs masked batched updates, and evaluates objective values and residual-based lower bounds. Only feasible sparse candidates update the upper bound; only valid lower bounds support pruning.

4 Batched GPU computation

For a batch of B node relaxations, store coefficients as columns of $\boldsymbol{\Theta} \in \mathbb{R}^{p\times B}$ and permitted-coordinate indicators in $\boldsymbol{M} \in \{0,1\}^{p\times B}$. A masked gradient evaluation is

$$\boldsymbol{G} = \left[\frac{1}{n}\boldsymbol{X}^\top\left\{\sigma(\boldsymbol{X}\boldsymbol{\Theta}) - \boldsymbol{y}\mathbf{1}_B^\top\right\} + \lambda\boldsymbol{\Theta}\right] \odot \boldsymbol{M}, \qquad \sigma(u) = (1+e^{-u})^{-1}. \tag{15}$$

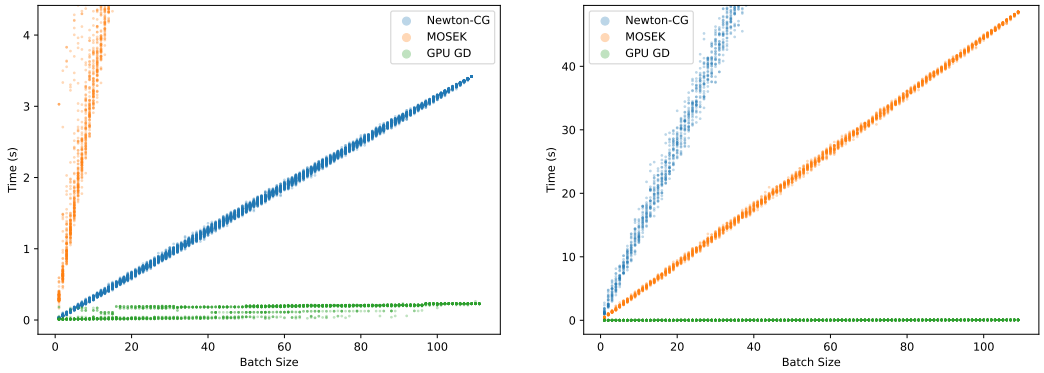
The sigmoid acts entrywise; Figure 2 writes $\boldsymbol{P} = \sigma(\boldsymbol{X}\boldsymbol{\Theta})$. The dominant operations are $\boldsymbol{X}\boldsymbol{\Theta}$ and $\boldsymbol{X}^\top\boldsymbol{R}$, with $\boldsymbol{X}, \boldsymbol{y}$ reused on the device. An iteration requires $O(npB)$ arithmetic and $O(np+nB+pB)$ dense storage. Larger batches can improve utilization, but also increase memory use and may contain models with different convergence rates.

Each active column uses a Barzilai–Borwein proposal (Barzilai & Borwein, 1988), formed from successive coefficient and gradient differences. We specify clipping and per-column backtracking in Algorithm 2 to ensure a descent safeguard. Converged columns are frozen. A changed mask resets the step history. Losses and gradients are reevaluated at the returned coefficients before applying (13); stale gradients are not used for certification.

Figure 2 separates data movement from search decisions. The CPU sends masks and warm starts, while the GPU carries out blocks of updates and returns objectives, certificates, and convergence flags. Periodic checks limit synchronization. A terminal node is refined until its local objective–bound gap is small enough for the global stopping tolerance. This design accelerates the continuous workload without changing which supports must remain covered by the search.

5 Experiments

We examine three questions: whether conditional refitting helps in a suppressor design, whether batched subproblem evaluation improves optimization progress, and how the aggressiveness of greedy pre-screening affects the tractability of large solves. The first screening comparison appears in Figure 1. The optimization study uses Myocardial Infarction Complications ($n = 1700, p = 111$) and Madelon ($n = 2000, p = 500$) from the UCI repository (Golovenkin et al., 2020; Guyon, 2004). Analysis with further datasets can be found in Appendix. Experiments are performed with $k \in \{4, 6, 8, 10\}$ and a 1000-second cutoff. The GPU implementation uses CuPy (Okuta et al., 2017) and CUDA on an NVIDIA RTX 6000. CPU experiments run on Georgia Tech's PACE `cpu-small`



(a) Myocardial.

(b) Madelon.

Figure 3: Runtime of warm-started subproblem batches. The vertical axis measures seconds per batch, not per node. The GPU implementation provides the lowest runtimes for moderate and large batch sizes in both datasets.

cluster (Partnership for an Advanced Computing Environment, 2017). The CPU brancher uses a largest-coefficient rule, whereas the GPU version uses candidate-branch evaluation. The code is available at `https://tinyurl.com/48jchvfj`.

5.1 Subproblem runtime

We first examine the effect of batched GPU subproblem evaluation on computational performance. Figure 3 compares batches warm-started from a related model with one additional feature. Newton–CG denotes the CPU Newton–CG implementation from scikit-learn (Pedregosa et al., 2011), MOSEK denotes the same convex ridge-logistic subproblems solved through the MOSEK Optimizer API (MOSEK ApS, 2025), and GPU GD denotes the batched gradient solver in Section 4. The GPU curves show lower batch runtimes than the plotted CPU alternatives for moderate and large batches. At the larger Myocardial batch sizes, the plotted separation exceeds an order of magnitude. The advantage grows with batch size as matrix–matrix operations make fuller use of GPU parallelism.

5.2 End-to-end search performance

We next compare the resulting end-to-end search performance of the CPU B&B, MOSEK, GPU, and pre-screened GPU implementations. Table 1 summarizes terminal reported gaps. On Myocardial, the GPU implementation reports gaps rounded to 0.00% at 74 and 227 seconds for $k = 4$ and 6, respectively; both CPU baselines reach the time limit with positive gaps. For $k = 8$, the reported GPU gap is 0.01% at 957 seconds. Pre-screening to $8k$ retains the optimal upper bound while providing significant speedup at all 3 sparsity levels.

Madelon is more challenging at the tested sparsity levels: all three exact implementations reach the 1000-second cutoff without converging. Nevertheless, the GPU implementation reports smaller terminal gaps at every tested value of k, ranging from 12.61% to 19.82%, compared with the two CPU baselines. Thus, the reduction in subproblem runtime observed in Figure 3 translates into improved end-to-end search progress within a fixed computational budget. Pre-screening to $2k$ maintained the current best feasible upper bound (before the time limit) at all 3 sparsity levels.

Figure 4 further examines the tradeoff between screening aggressiveness and total runtime by varying the screening multiplier over $M \in \{2, 4, 6, 8\}$. More aggressive screening generally reduces runtime by shrinking the subsequent B&B search, while larger multipliers retain more candidate variables and increase computational cost. This effect is most pronounced on Madelon, where larger screening levels approach or exceed the 1000-second cutoff for larger k. Additional screening results for upper bounds, lower bounds, and larger datasets are reported in Appendix **??**.

Table 1: Reported terminal gaps under a 1000-second cutoff. CPU baselines use the full cutoff in every row. Gap percentages are reported solver outputs; smaller is better. GPU times below 1000 seconds denote termination before the cutoff.

Dataset	k	Solver	Bound Gap	UB	Time (s)
Myocardial Infarction Complications	4	CPU B&B	11.52%	0.6753	1000
		MOSEK	2.75%	0.6271	1000
		GPU GD B&B	0.00%	0.6271	74
		Pre-screened (8k)	0.00%	0.6271	0.62
	6	CPU B&B	10.13%	0.6623	1000
		MOSEK	3.84%	0.6171	1000
		GPU GD B&B	0.00%	0.6167	227
		Pre-screened (8k)	0.00%	0.6167	7.02
	8	CPU B&B	8.70%	0.6334	1000
		MOSEK	4.47%	0.6092	1000
		GPU GD B&B	0.01%	0.6088	957
		Pre-screened (8k)	0.00%	0.6088	78.9
Madelon	4	CPU B&B	24.54%	0.6909	1000
		MOSEK	21.29%	0.6597	1000
		GPU GD B&B	12.61%	0.6578	1000
		Pre-screened (2k)	0.00%	0.6578	0.94
	6	CPU B&B	24.86%	0.6907	1000
		MOSEK	21.29%	0.6583	1000
		GPU GD B&B	18.64%	0.6548	1000
		Pre-screened (2k)	0.00%	0.6548	0.39
	8	CPU B&B	24.96%	0.6894	1000
		MOSEK	21.26%	0.6558	1000
		GPU GD B&B	19.82%	0.6513	1000
		Pre-screened (2k)	0.00%	0.6513	2.10

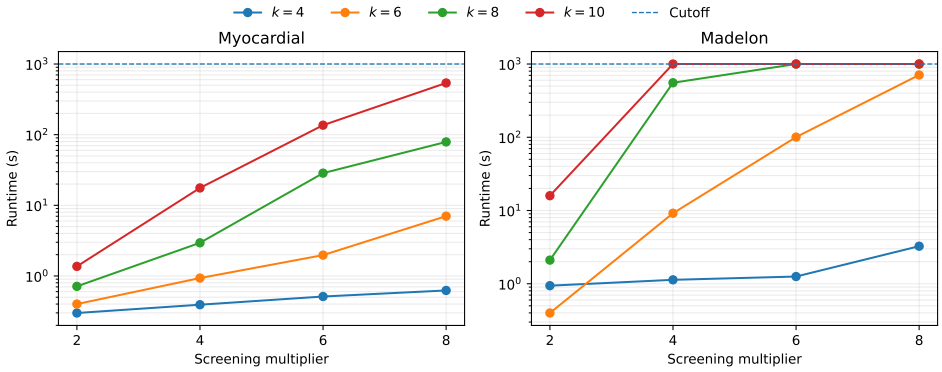


Figure 4: Runtime of greedy pre-screened GPU accelerated B&B for $k = \{4, 6, 8, 10\}$ pre-screened to $M \times k$ features where $M \in \{2, 4, 6, 8\}$. The vertical axis measures total runtime on a log-scale, while the horizontal axis measures the screening multiplier M.

6 Conclusion

We developed a GPU-compatible branch-and-bound framework for cardinality-constrained ridge-logistic regression and a conditional screening analysis that permits a bounded number of irrelevant selections. The optimization method separates approximate primal values from residual-corrected lower bounds, while batching exploits the shared algebra of node relaxations. The experiments show faster support recovery under correlated suppressors, lower batched subproblem runtimes, and smaller terminal optimality gaps than the tested CPU baselines. Together, the statistical and computational results support GPU-batched branch-and-bound as an effective approach to exact sparse logistic regression.

AI USE STATEMENT

Generative AI assisted this revision with editing and manuscript organization. Generative AI also assisted with finding datasets and coding for numerical results.

ETHICS STATEMENT

The study concerns optimization methods and uses public benchmark data, including a medical dataset. No clinical deployment, diagnostic validity, or causal interpretation is claimed. A small selected feature set does not by itself ensure calibrated, unbiased, or clinically appropriate predictions. Any downstream use would require separate validation, attention to data provenance and privacy, and assessment of group-specific performance.

REPRODUCIBILITY STATEMENT

Appendices A–E provide detailed assumptions, proofs, and safeguarded pseudocode. Appendix F specifies the simulation designs and reports complete aggregate tables. The accompanying source package includes the figure sources and aggregate experimental data used in the manuscript.

REFERENCES

Sohail Bahmani, Bhiksha Raj, and Petros T. Boufounos. Greedy sparsity-constrained optimization. *Journal of Machine Learning Research*, 14(25):807–841, 2013.

Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8(1):141–148, 1988. doi: 10.1093/imanum/8.1.141.

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

Antoine Dedieu, Hussein Hazimeh, and Rahul Mazumder. Learning sparse classifiers: Continuous and mixed integer optimization perspectives. *Journal of Machine Learning Research*, 22(135): 1–47, 2021.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911, 2008.

Jianqing Fan and Rui Song. Sure independence screening in generalized linear models with NP-dimensionality. *The Annals of Statistics*, 38(6):3567–3604, 2010.

Jianqing Fan, Richard Samworth, and Yichao Wu. Ultrahigh Dimensional Feature Selection: Beyond the Linear Model. *Journal of Machine Learning Research*, 10:2013–2038, 2009.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

S. Golovenkin, V. Shulman, D. Rossiev, P. Shesternya, S. Nikulina, Y. Orlova, and V. Voino-Yasenetsky. Myocardial infarction complications [dataset]. UCI Machine Learning Repository, 2020.

Isabelle Guyon. Madelon. UCI Machine Learning Repository, 2004.

Isabelle Guyon, Steve Gunn, Asa Ben-Hur, and Gideon Dror. Gisette. UCI Machine Learning Repository, 2004. DOI: https://doi.org/10.24432/C5HP5B.

Hussein Hazimeh and Rahul Mazumder. Fast best subset selection: Coordinate descent and local combinatorial optimization algorithms. *Operations Research*, 68(5):1517–1537, 2020.

Hussein Hazimeh, Rahul Mazumder, and Ali Saab. Sparse regression at scale: Branch-and-bound rooted in first-order optimization. *Mathematical Programming*, 196(1–2):347–388, 2022.

Hussein Hazimeh, Rahul Mazumder, and Tim Nonet. L0Learn: A scalable package for sparse learning using l_0 regularization. *Journal of Machine Learning Research*, 24(205):1–8, 2023.

Aaron P. Heath, Vincent Ferretti, Sumit Agrawal, et al. The nci genomic data commons. *Nature Genetics*, 53:257–262, 2021. doi: 10.1038/s41588-021-00791-5.

Jiachang Liu and Andrea Lodi. From sequential nodes to GPU batches: Parallel branch and bound for optimal k-sparse GLMs, 2026.

Jiachang Liu, Sam Rosen, Chudi Zhong, and Cynthia Rudin. OKRidge: Scalable optimal k-sparse ridge regression. In *Advances in Neural Information Processing Systems*, volume 36, pp. 41076–41258, 2023.

Jiachang Liu, Soroosh Shafiee, and Andrea Lodi. Scalable first-order method for certifying optimal k-sparse GLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39455–39481. PMLR, 2025. URL https://proceedings.mlr.press/v267/liu25bk.html.

Jiachang Liu, Andrea Lodi, and Soroosh Shafiee. GPU-friendly and linearly convergent first-order methods for certifying optimal k-sparse GLMs, 2026.

Aurelie Lozano, Grzegorz Swirszcz, and Naoki Abe. Group orthogonal matching pursuit for logistic regression. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pp. 452–460. PMLR, 2011.

Xiang Meng, Ryan Lucas, and Rahul Mazumder. A GPU-accelerated nonlinear branch-and-bound framework for sparse linear models, 2026.

MOSEK ApS. *The MOSEK Python Optimizer API Manual, Version 11.0*, 2025. URL https://docs.mosek.com/11.0/pythonapi/index.html.

Ryosuke Okuta, Yuya Unno, Daisuke Nishino, Shohei Hido, and Crissman Loomis. CuPy: A NumPy-compatible library for NVIDIA GPU calculations. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS)*, 2017. URL http://learningsys.org/nips17/assets/papers/paper_16.pdf.

Alireza Olama, Eduardo Camponogara, and Jan Kronqvist. Sparse convex optimization toolkit: A mixed-integer framework. *Optimization Methods and Software*, 38(6):1269–1295, 2023. doi: 10.1080/10556788.2023.2222429.

Partnership for an Advanced Computing Environment. Partnership for an advanced computing environment (PACE). Georgia Institute of Technology, 2017. URL https://pace.gatech.edu/.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12: 2825–2830, 2011.

Toshiki Sato, Yuichi Takano, Ryuhei Miyashiro, and Akiko Yoshise. Feature subset selection for logistic regression via mixed integer optimization. *Computational Optimization and Applications*, 64(3):865–880, 2016.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996. doi: 10.1111/j.2517-6161.1996.tb02080.x.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge University Press, 2018.

Martin J. Wainwright. *High-dimensional statistics: A non-asymptotic viewpoint*, volume 48. Cambridge University Press, 2019.

Rui Wang, Naihua Xiu, and Chao Zhang. Greedy projected gradient–newton method for sparse logistic regression. *IEEE Transactions on Neural Networks and Learning Systems*, 31(2):527–538, 2020. doi: 10.1109/TNNLS.2019.2905261.

Xiaotong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit for sparsity-constrained optimization. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 127–135. PMLR, 2014.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2):894–942, 2010.

Jin Zhu, Xueqin Wang, Liyuan Hu, Junhao Huang, Kangkang Jiang, Yanhang Zhang, Shiyun Lin, and Junxian Zhu. abess: A fast best-subset selection library in python and r. *Journal of Machine Learning Research*, 23(202):1–7, 2022. URL `https://www.jmlr.org/papers/v23/21-1060.html`.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x.

A DETAILED SCREENING PROOFS

This appendix provides complete arguments for the population bounds and the finite-sample screening theorem. Constants denoted by c, C may change between displays, but depend only on the parameters indicated in the corresponding statement. The assumptions apply to a triangular array when p or d_n changes with n. The ridge parameter is fixed and positive unless a statement explicitly tracks its dependence.

A.1 POPULATION CURVATURE AND SPARSE SECOND MOMENTS

Lemma 4. *Under (D2), for every $\boldsymbol{\theta} \in \mathbb{R}^p$ and every $B \subseteq [p]$ with $|B| \le d_n + k$,*

$$\lambda \boldsymbol{I}_{|B|} \preceq \nabla^2_{B,B} f(\boldsymbol{\theta}) \preceq (\lambda + C_\Sigma/4)\boldsymbol{I}_{|B|}. \tag{16}$$

For $\delta \in (0, 1/2)$, if $n \ge C_0 H_n(\delta)$ with a sufficiently large $C_0 = C_0(K_x, C_\Sigma)$, then with probability at least $1 - \delta/4$,

$$\sup_{\substack{B \subseteq [p] \\ |B| \le d_n + k}} \lambda_{\max}\left(\frac{1}{n}\boldsymbol{X}_B^\top \boldsymbol{X}_B\right) \le 2C_\Sigma. \tag{17}$$

Proof. Write $\sigma(u) = (1 + e^{-u})^{-1}$ and $w(u) = \sigma(u)\{1 - \sigma(u)\}$. The loss equals $\log(1 + e^u) - yu$ for $y \in \{0, 1\}$. Its second derivative is $w(u)$, independent of y, with $0 \le w(u) \le 1/4$. Finite second moments permit differentiation under expectation. Thus

$$\nabla^2 f(\boldsymbol{\theta}) = \mathbb{E}[w(\tilde{\boldsymbol{x}}^\top \boldsymbol{\theta})\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}^\top] + \lambda \boldsymbol{I}_p.$$

For any B and $\boldsymbol{u} \in \mathbb{R}^{|B|}$,

$$0 \le \boldsymbol{u}^\top \{\nabla^2_{B,B} f(\boldsymbol{\theta}) - \lambda \boldsymbol{I}_{|B|}\}\boldsymbol{u} = \mathbb{E}[w(\tilde{\boldsymbol{x}}^\top \boldsymbol{\theta})(\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_B)^2] \le \tfrac{1}{4}\mathbb{E}[(\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_B)^2] \le \tfrac{C_\Sigma}{4}\|\boldsymbol{u}\|_2^2.$$

This proves (16).

For the empirical claim, set $m = \min\{p, d_n + k\}$ and $\boldsymbol{\Sigma}_B = \mathbb{E}[\tilde{\boldsymbol{x}}_B \tilde{\boldsymbol{x}}_B^\top]$. For a fixed B of size $s \ge 1$, choose a $1/4$-net $\mathcal{V}_B$ of the unit sphere in $\mathbb{R}^s$, with $|\mathcal{V}_B| \le 9^s$. Such a net follows by taking a maximal separated set and comparing volumes of disjoint Euclidean balls. For any symmetric matrix $\boldsymbol{Q}$, a unit vector $\boldsymbol{v}$ can be approximated by $\boldsymbol{u} \in \mathcal{V}_B$ with $\|\boldsymbol{v} - \boldsymbol{u}\|_2 \le 1/4$. The identity

$$\boldsymbol{v}^\top \boldsymbol{Q}\boldsymbol{v} - \boldsymbol{u}^\top \boldsymbol{Q}\boldsymbol{u} = (\boldsymbol{v} - \boldsymbol{u})^\top \boldsymbol{Q}\boldsymbol{v} + \boldsymbol{u}^\top \boldsymbol{Q}(\boldsymbol{v} - \boldsymbol{u})$$

implies an absolute difference of at most $\|\boldsymbol{Q}\|_2/2$. Taking the supremum over $\boldsymbol{v}$ gives

$$\|\boldsymbol{Q}\|_2 \le 2 \max_{\boldsymbol{u} \in \mathcal{V}_B} |\boldsymbol{u}^\top \boldsymbol{Q}\boldsymbol{u}|. \tag{18}$$

For fixed $B, \boldsymbol{u}$, condition (D2) implies that $(\boldsymbol{u}^\top \boldsymbol{x}_{i,B})^2 - \mathbb{E}[(\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_B)^2]$ is a mean-zero sub-exponential variable with norm at most CK_x^2. Bernstein's inequality therefore gives, for every $t > 0$,

$$\mathbb{P}\left\{\left|\frac{1}{n}\sum_{i=1}^n (\boldsymbol{u}^\top \boldsymbol{x}_{i,B})^2 - \mathbb{E}[(\boldsymbol{u}^\top \tilde{\boldsymbol{x}}_B)^2]\right| > t\right\} \le 2\exp\left[-cn \min\left\{\frac{t^2}{K_x^4}, \frac{t}{K_x^2}\right\}\right]. \tag{19}$$

These scalar concentration and net bounds are standard; see Vershynin (2018); Wainwright (2019). No independence among coordinates is needed.

There are at most $(p + 1)^m$ subsets of size at most m: encode a subset by its ordered elements and pad its list to length m with zeroes. The total number of net points over these subsets is at most $9^m(p + 1)^m$. Its logarithm is at most $Cm \log(ep)$. Apply (19) to all these points and use (18). Increasing a constant if needed, the resulting event has probability at least $1 - \delta/4$ and satisfies

$$\sup_{|B| \le m} \left\|\frac{1}{n}\boldsymbol{X}_B^\top \boldsymbol{X}_B - \boldsymbol{\Sigma}_B\right\|_2 \le CK_x^2 \left\{\sqrt{\frac{H_n(\delta)}{n}} + \frac{H_n(\delta)}{n}\right\}. \tag{20}$$

Choose C_0 so large that the right-hand side is at most C_Σ whenever $n \ge C_0 H_n(\delta)$. Weyl's inequality, or the variational definition of the largest eigenvalue, now gives

$$\lambda_{\max}(\boldsymbol{X}_B^\top \boldsymbol{X}_B/n) \le \lambda_{\max}(\boldsymbol{\Sigma}_B) + \|\boldsymbol{X}_B^\top \boldsymbol{X}_B/n - \boldsymbol{\Sigma}_B\|_2 \le 2C_\Sigma$$

uniformly over B. The empty set contributes nothing. □

A.2 UNIFORM CONDITIONAL-SCORE CONCENTRATION

Lemma 1 (Uniform conditional-score concentration)**.** *Under (D2) and* (7)*, a constant C_1, depending only on K_x, C_Σ, λ, gives*

$$\mathbb{P}\left\{\sup_{|A|\le d_n}\max_{j\notin A}|\hat{G}_j(A)-G_j(A)|\le r_n(\delta)\right\}\ge 1-\delta,\qquad r_n(\delta)=C_1\sqrt{\frac{H_n(\delta)}{n}}. \tag{8}$$

The maximum over no remaining coordinates is zero.

Proof. We prove three simultaneous bounds: restricted population-point gradients, scalar population-point gradients, and an empirical cross-Hessian bound. The population optimizer $\boldsymbol{\theta}^A$ is deterministic for each fixed A.

Step 1. Let $a=|A|\ge 1$, and write

$$\boldsymbol{Z}_{i,A}=\{\sigma(\boldsymbol{x}_i^\top\boldsymbol{\theta}^A)-y_i\}\boldsymbol{x}_{i,A},\qquad \boldsymbol{R}_A=\frac{1}{n}\sum_{i=1}^n(\boldsymbol{Z}_{i,A}-\mathbb{E}\boldsymbol{Z}_{i,A}).$$

The deterministic ridge terms cancel in the difference between sample and population gradients, so

$$\boldsymbol{R}_A=\nabla_A f_n(\boldsymbol{\theta}^A)-\nabla_A f(\boldsymbol{\theta}^A).$$

For any unit vector $\boldsymbol{u}\in\mathbb{R}^a$, the residual has absolute value at most one, and hence

$$|\boldsymbol{u}^\top\boldsymbol{Z}_{i,A}|\le|\boldsymbol{u}^\top\boldsymbol{x}_{i,A}|.$$

Condition (D2) implies $\|\boldsymbol{u}^\top\boldsymbol{Z}_{i,A}\|_{\psi_2}\le K_x$, regardless of dependence between the residual and the covariates. Centering multiplies this norm by at most an absolute constant. Independence across observations then yields

$$\mathbb{P}\{|\boldsymbol{u}^\top\boldsymbol{R}_A|>t\}\le 2\exp(-cnt^2/K_x^2),\qquad t>0. \tag{21}$$

Take a $1/2$-net $\mathcal{U}_A$ with at most 5^a elements. If $\boldsymbol{v}=\boldsymbol{R}_A/\|\boldsymbol{R}_A\|_2$ when $\boldsymbol{R}_A\neq 0$, choose $\boldsymbol{u}\in\mathcal{U}_A$ within $1/2$ of $\boldsymbol{v}$. Then $\boldsymbol{u}^\top\boldsymbol{R}_A\ge\|\boldsymbol{R}_A\|_2/2$. Thus

$$\|\boldsymbol{R}_A\|_2\le 2\max_{\boldsymbol{u}\in\mathcal{U}_A}|\boldsymbol{u}^\top\boldsymbol{R}_A|.$$

The total number of pairs $(A,\boldsymbol{u})$ with $|A|\le d_n$ is at most $5^{d_n}(p+1)^{d_n}$. Applying (21) to these pairs and choosing a sufficiently large C_R gives, with probability at least $1-\delta/4$,

$$\sup_{|A|\le d_n}\|\boldsymbol{R}_A\|_2\le R_n:=C_R K_x\sqrt{H_n(\delta)/n}. \tag{22}$$

The claim is immediate for $A=\varnothing$.

Step 2. The first-order conditions are $\nabla_A f(\boldsymbol{\theta}^A)=\mathbf{0}$ and $\nabla_A f_n(\hat{\boldsymbol{\theta}}^A)=\mathbf{0}$. Define

$$\bar{\boldsymbol{H}}^{(n)}_{A,A}=\int_0^1\nabla^2_{A,A}f_n\Big(\boldsymbol{\theta}^A+t(\hat{\boldsymbol{\theta}}^A-\boldsymbol{\theta}^A)\Big)\,dt.$$

The fundamental theorem of calculus gives

$$\mathbf{0}-\nabla_A f_n(\boldsymbol{\theta}^A)=\bar{\boldsymbol{H}}^{(n)}_{A,A}(\hat{\boldsymbol{\theta}}^A_A-\boldsymbol{\theta}^A_A).$$

Every empirical Hessian is at least $\lambda\boldsymbol{I}$, so this matrix is invertible and its inverse has norm at most $1/\lambda$. On (22),

$$\sup_{|A|\le d_n}\|\hat{\boldsymbol{\theta}}^A-\boldsymbol{\theta}^A\|_2\le R_n/\lambda. \tag{23}$$

This uses strong convexity of the empirical problem; a lower restricted eigenvalue condition on the design is not needed.

Step 3. For fixed A and $j \notin A$, write

$$T_{A,j} = \nabla_j f_n(\boldsymbol{\theta}^A) - \nabla_j f(\boldsymbol{\theta}^A).$$

This is an average of independent centered variables of the form $\{\sigma(\boldsymbol{x}_i^\top \boldsymbol{\theta}^A) - y_i\}x_{ij}$. Their sub-Gaussian norms are at most CK_x. There are at most $p(p+1)^{d_n}$ admissible pairs (A, j). The same scalar bound and a union bound give, with probability at least $1 - \delta/4$,

$$\sup_{|A|\le d_n} \max_{j\notin A} |T_{A,j}| \le T_n := C_T K_x \sqrt{H_n(\delta)/n}. \tag{24}$$

The entropy $\log p$ for the additional coordinate is covered by the $k\log(ep)$ term in H_n, since $k \ge 1$.

Step 4. Fix $j \notin A$ and put $B = A \cup \{j\}$. Its size is at most $d_n + 1 \le d_n + k$. For every $\boldsymbol{\theta}$, let $\boldsymbol{W}(\boldsymbol{\theta}) = \mathrm{diag}\{w(\boldsymbol{x}_i^\top \boldsymbol{\theta})\}_{i=1}^n$. Then

$$\mathbf{0} \preceq \frac{1}{n}\boldsymbol{X}_B^\top \boldsymbol{W}(\boldsymbol{\theta})\boldsymbol{X}_B \preceq \frac{1}{4n}\boldsymbol{X}_B^\top \boldsymbol{X}_B.$$

On (17), the operator norm of this weighted matrix is at most $C_\Sigma/2$. Its (j, A) block is precisely $\nabla^2_{j,A} f_n(\boldsymbol{\theta})$, as the ridge matrix has zero off-diagonal blocks. The norm of a submatrix is bounded by the norm of the full matrix: coordinate-selection matrices on the left and right have norm at most one. Therefore

$$\sup_{\boldsymbol{\theta}\in\mathbb{R}^p} \sup_{\substack{|A|\le d_n \\ j\notin A}} \|\nabla^2_{j,A} f_n(\boldsymbol{\theta})\|_2 \le C_\Sigma/2. \tag{25}$$

Integrating along the segment between $\boldsymbol{\theta}^A$ and $\hat{\boldsymbol{\theta}}^A$ and applying (23) now gives

$$|\nabla_j f_n(\hat{\boldsymbol{\theta}}^A) - \nabla_j f_n(\boldsymbol{\theta}^A)| \le \frac{C_\Sigma R_n}{2\lambda}.$$

Step 5. The reverse triangle inequality and (24) imply

$$\begin{aligned} |\hat{G}_j(A) - G_j(A)| &\le |\nabla_j f_n(\hat{\boldsymbol{\theta}}^A) - \nabla_j f(\boldsymbol{\theta}^A)| \\ &\le |\nabla_j f_n(\hat{\boldsymbol{\theta}}^A) - \nabla_j f_n(\boldsymbol{\theta}^A)| + |T_{A,j}| \\ &\le C_\Sigma R_n/(2\lambda) + T_n. \end{aligned}$$

The three events used in Steps 1, 3, and 4 fail with total probability at most $3\delta/4 \le \delta$. Choose $C_1 \ge C_\Sigma C_R K_x/(2\lambda) + C_T K_x$. The last display is then at most $r_n(\delta)$ uniformly over all A, j, proving (8). □

A.3 A MISSING SIGNAL HAS A NONZERO CONDITIONAL SCORE

Lemma 5. *Under (D1)–(D2), every A with $|A| \le d_n$ and $U = S^\star \setminus A \ne \varnothing$ satisfies* $\max_{j\in U} G_j(A) \ge \lambda c_\beta$.

Proof. Set

$$\bar{\boldsymbol{H}} = \int_0^1 \nabla^2 f\big(\boldsymbol{\theta}^A + t(\boldsymbol{\theta}^\star - \boldsymbol{\theta}^A)\big)\, dt.$$

The first-order conditions are $\nabla_A f(\boldsymbol{\theta}^A) = \mathbf{0}$ and $\nabla f(\boldsymbol{\theta}^\star) = \mathbf{0}$. Since $\boldsymbol{\theta}^\star - \boldsymbol{\theta}^A$ is supported on $A \cup U$, the integrated gradient identity on A reads

$$\mathbf{0} = \bar{\boldsymbol{H}}_{A,A}(\boldsymbol{\theta}^\star_A - \boldsymbol{\theta}^A_A) + \bar{\boldsymbol{H}}_{A,U}\boldsymbol{\theta}^\star_U.$$

When $A \ne \varnothing$, $\bar{\boldsymbol{H}}_{A,A} \succeq \lambda \boldsymbol{I}$ is invertible, so

$$\boldsymbol{\theta}^\star_A - \boldsymbol{\theta}^A_A = -\bar{\boldsymbol{H}}^{-1}_{A,A}\bar{\boldsymbol{H}}_{A,U}\boldsymbol{\theta}^\star_U. \tag{26}$$

On U, the same identity and substitution give

$$-\nabla_U f(\boldsymbol{\theta}^A) = \Big(\bar{\boldsymbol{H}}_{U,U} - \bar{\boldsymbol{H}}_{U,A}\bar{\boldsymbol{H}}^{-1}_{A,A}\bar{\boldsymbol{H}}_{A,U}\Big)\boldsymbol{\theta}^\star_U =: \boldsymbol{S}\boldsymbol{\theta}^\star_U.$$

We verify the lower bound on this Schur complement. For any $\boldsymbol{u} \in \mathbb{R}^{|U|}$, completing the square in $\boldsymbol{v}$ yields

$$\boldsymbol{u}^\top \boldsymbol{S}\boldsymbol{u} = \min_{\boldsymbol{v}\in\mathbb{R}^{|A|}} \begin{pmatrix}\boldsymbol{v}\\ \boldsymbol{u}\end{pmatrix}^\top \bar{\boldsymbol{H}}_{A\cup U, A\cup U} \begin{pmatrix}\boldsymbol{v}\\ \boldsymbol{u}\end{pmatrix} \geq \min_{\boldsymbol{v}} \lambda(\|\boldsymbol{v}\|_2^2 + \|\boldsymbol{u}\|_2^2) = \lambda\|\boldsymbol{u}\|_2^2.$$

Hence $\boldsymbol{S} \succeq \lambda \boldsymbol{I}$. If $A = \varnothing$, the same claim holds directly with $\boldsymbol{S} = \bar{\boldsymbol{H}}_{U,U}$; there is no inverse of an empty block to evaluate. The spectral theorem gives $\|\boldsymbol{S}\boldsymbol{\theta}^\star_U\|_2 \geq \lambda\|\boldsymbol{\theta}^\star_U\|_2$. Finally,

$$\max_{j\in U} G_j(A) \geq \frac{\|\nabla_U f(\boldsymbol{\theta}^A)\|_2}{\sqrt{|U|}} \geq \lambda \frac{\|\boldsymbol{\theta}^\star_U\|_2}{\sqrt{|U|}} \geq \lambda \min_{j\in U} |\theta^\star_j| \geq \lambda c_\beta.$$

□

A.4 Error from omitting signal coordinates

Lemma 6. *Under (D1)–(D2), every A with $|A| \leq d_n$ and $U = S^\star \setminus A \neq \varnothing$ satisfies*

$$\|\boldsymbol{\theta}^A - \boldsymbol{\theta}^\star\|_2 \leq C_\theta \|\boldsymbol{\theta}^\star_U\|_2 \leq C_\theta B_\theta, \qquad C_\theta = 1 + C_\Sigma/(4\lambda).$$

Proof. Use the integrated Hessian from the preceding proof. For $A \neq \varnothing$, (26) gives

$$\|\boldsymbol{\theta}^\star_A - \boldsymbol{\theta}^A_A\|_2 \leq \|\bar{\boldsymbol{H}}^{-1}_{A,A}\|_2 \|\bar{\boldsymbol{H}}_{A,U}\|_2 \|\boldsymbol{\theta}^\star_U\|_2.$$

The inverse norm is at most $1/\lambda$. To bound the off-diagonal block, set $B = A \cup U$. For every point on the interpolation segment,

$$\boldsymbol{Q}_B(\boldsymbol{\theta}) = \mathbb{E}[w(\tilde{\boldsymbol{x}}^\top\boldsymbol{\theta})\tilde{\boldsymbol{x}}_B\tilde{\boldsymbol{x}}_B^\top] \quad \text{satisfies} \quad \boldsymbol{0} \preceq \boldsymbol{Q}_B(\boldsymbol{\theta}) \preceq \tfrac{1}{4}\boldsymbol{\Sigma}_B.$$

Since $|B| \leq d_n + k$, $\|\boldsymbol{Q}_B(\boldsymbol{\theta})\|_2 \leq C_\Sigma/4$. The ridge term has no (A,U) contribution, since $A \cap U = \varnothing$. Selecting the (A,U) block and integrating therefore yields $\|\bar{\boldsymbol{H}}_{A,U}\|_2 \leq C_\Sigma/4$. Thus

$$\|\boldsymbol{\theta}^\star_A - \boldsymbol{\theta}^A_A\|_2 \leq \frac{C_\Sigma}{4\lambda}\|\boldsymbol{\theta}^\star_U\|_2.$$

On U, $\boldsymbol{\theta}^A_U = \boldsymbol{0}$, and outside $A \cup U$ both vectors vanish. The triangle inequality gives

$$\|\boldsymbol{\theta}^A - \boldsymbol{\theta}^\star\|_2 \leq \|\boldsymbol{\theta}^A_A - \boldsymbol{\theta}^\star_A\|_2 + \|\boldsymbol{\theta}^\star_U\|_2 \leq C_\theta\|\boldsymbol{\theta}^\star_U\|_2.$$

For $A = \varnothing$ the first term is zero, so the same inequality holds. The final claim uses (D1). □

A.5 Scores outside the exceptional set

Lemma 7. *Under (D1)–(D3), for every A with $|A| \leq d_n$ and $A \not\supseteq S^\star$,*

$$\max_{j\in(S^\star)^c\setminus(\mathcal{N}_n\cup A)} G_j(A) \leq C_\theta B_\theta \mu_n.$$

An empty maximum is zero. For fixed C_θ, B_θ, the bound tends to zero when $\mu_n \to 0$.

Proof. If there is no admissible j, the claim follows from the convention. Otherwise fix $j \notin S^\star \cup \mathcal{N}_n \cup A$. Along $\boldsymbol{\theta}(t) = \boldsymbol{\theta}^A + t(\boldsymbol{\theta}^\star - \boldsymbol{\theta}^A)$, $\nabla_j f(\boldsymbol{\theta}^\star) = 0$ and the fundamental theorem of calculus give

$$-\nabla_j f(\boldsymbol{\theta}^A) = \int_0^1 \nabla^2_{j,A\cup S^\star} f(\boldsymbol{\theta}(t))(\boldsymbol{\theta}^\star - \boldsymbol{\theta}^A)_{A\cup S^\star}\, dt.$$

The coefficient difference is supported on $A \cup S^\star$. By the integral triangle inequality, the operator-norm inequality, and the definition of $\Delta_j^{(n)}$,

$$\begin{aligned} G_j(A) &\leq \int_0^1 \|\nabla^2_{j,A\cup S^\star} f(\boldsymbol{\theta}(t))\|_2 \|\boldsymbol{\theta}^\star - \boldsymbol{\theta}^A\|_2\, dt \\ &\leq \Delta_j^{(n)}\|\boldsymbol{\theta}^\star - \boldsymbol{\theta}^A\|_2 \leq \mu_n C_\theta B_\theta. \end{aligned}$$

The last step uses $j \notin \mathcal{N}_n$ and Lemma 6. The bound is uniform in both indices. □

A.6 SUPPORT CONTAINMENT WITH A BOUNDED DELAY

Theorem 2 (Screening with a bounded number of mistakes)**.** *Suppose (D1)–(D4) hold. If*

$$C_\theta B_\theta \mu_n + 2r_n(\delta) < \lambda c_\beta, \tag{11}$$

then $\mathbb{P}\{S^\star \subseteq A_{k+M_n}\} \geq 1-\delta$. *Before support containment, every irrelevant selection belongs to* $\mathcal{N}_n$ *on the event in* (8)*.*

Proof. Let $\mathcal{E}$ be the event in (8), whose probability is at least $1-\delta$. Fix a step $t < k + M_n$ before containment, and put $U_t = S^\star \setminus A_t \neq \varnothing$. Since $|A_t| = t \leq d_n$, the population lemmas and $\mathcal{E}$ imply

$$\max_{j\in U_t} \hat{G}_j(A_t) \geq \lambda c_\beta - r_n(\delta), \qquad \max_{j\in (S^\star)^c\setminus(\mathcal{N}_n\cup A_t)} \hat{G}_j(A_t) \leq C_\theta B_\theta \mu_n + r_n(\delta).$$

Condition (11) makes the first bound strictly larger than the second. Therefore an empirical maximizer cannot be a nonexceptional irrelevant coordinate. Ties among signals and exceptional coordinates are harmless; the stated deterministic rule fixes the path when needed.

Suppose, for contradiction, that $S^\star \not\subseteq A_{k+M_n}$ on $\mathcal{E}$. The path selects distinct coordinates, one per step, and has not reached containment at any earlier step. Each selected irrelevant coordinate must then be in $\mathcal{N}_n$, which has size M_n. Among the $k + M_n$ selected coordinates, at least k must belong to $S^\star$. Since $|S^\star| = k$, all of $S^\star$ has been selected, a contradiction. Thus containment holds on $\mathcal{E}$, and its probability is at least $1-\delta$. □

Asymptotic specialization. For $n \geq 3$, take $\delta_n = 1/n$. If $(d_n + k)\log(ep) = o(n)$, then $H_n(\delta_n)/n \to 0$, since $\log(8n)/n \to 0$. Hence $r_n(\delta_n) \to 0$ and (7) holds eventually. The condition $\mu_n \to 0$ is sufficient but not necessary. If there is an $\eta > 0$ such that

$$C_\theta B_\theta \mu_n \leq \lambda c_\beta - \eta$$

eventually, then $2r_n(\delta_n) < \eta$ eventually and the margin (11) follows. Together with $k + M_n \leq d_n$, Theorem 2 gives $\mathbb{P}\{S^\star \subseteq A_{k+M_n}\} \geq 1 - 1/n \to 1$. For fixed $c_\beta, B_\theta, C_\Sigma$, the largest admissible scale suggested by this margin is

$$\mu_n < \frac{\lambda c_\beta}{C_\theta B_\theta} = \frac{4\lambda^2 c_\beta}{B_\theta(4\lambda + C_\Sigma)}.$$

Thus, when these signal quantities are held fixed, the admissible threshold is of order λ^2 as $\lambda \downarrow 0$; in turn, a smaller threshold can enlarge M_n. If the ridge-population target changes with λ, then c_β and B_θ should be tracked as well. When $M_n = 0$, containment at step k and $|A_k| = |S^\star| = k$ imply $A_k = S^\star$ on the same event.

Inexact refits. The proof also handles approximate conditional scores. Suppose a refit $\tilde{\boldsymbol{\theta}}^A$ is supported on A and satisfies $\|\nabla_A f_n(\tilde{\boldsymbol{\theta}}^A)\|_2 \leq \nu_n$ uniformly over the refits used. Strong convexity gives $\|\tilde{\boldsymbol{\theta}}^A - \hat{\boldsymbol{\theta}}^A\|_2 \leq \nu_n/\lambda$. On the covariance event, (25) bounds the additional score error by $C_\Sigma\nu_n/(2\lambda)$. Replacing $r_n(\delta)$ in the separation margin by $r_n(\delta) + C_\Sigma\nu_n/(2\lambda)$ yields the same counting argument. This extension requires a uniform residual tolerance; an unspecified optimization stopping rule is insufficient.

B DESIGN EXAMPLES AND THE COMPETITION COUNT

The general theorem controls competition through a weighted Hessian. This section explains what an ordinary covariance description can and cannot imply.

B.1 A CONDITIONAL BLOCK-COVARIANCE EXAMPLE

Partition $(S^\star)^c = \mathcal{C}_n \cup \mathcal{R}_n$, where $|\mathcal{C}_n| = m_n$, and consider a centered covariate vector with unit marginal variances and covariance

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{I}_k & \rho_n \mathbf{1}_k \mathbf{1}_{m_n}^\top & \mathbf{0} \\ \rho_n \mathbf{1}_{m_n}\mathbf{1}_k^\top & \boldsymbol{I}_{m_n} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{I}_{|\mathcal{R}_n|} \end{pmatrix}. \tag{27}$$

Thus $\mathbf{\Sigma}_{j,S^\star} = \rho_n \mathbf{1}_k^\top$ for $j \in \mathcal{C}_n$ and zero for $j \in \mathcal{R}_n$. The off-diagonal rank-one block has singular value $|\rho_n|\sqrt{km_n}$. The nontrivial eigenvalues of the upper-left block are $1 \pm |\rho_n|\sqrt{km_n}$, so a necessary and sufficient condition for (27) to be positive semidefinite is

$$|\rho_n|\sqrt{km_n} \le 1. \tag{28}$$

When this holds, its largest eigenvalue is at most two. A centered Gaussian vector with this covariance satisfies (D2) with absolute constants. These observations verify the design regularity, but not sparsity of the ridge target or the next condition.

(D5) Weighted cross-Hessian control. For a constant $\gamma > 0$ and the same budget d_n used in (6), assume

$$\Delta_j^{(n)} \le \gamma \|\mathbf{\Sigma}_{j,S^\star}\|_2, \qquad j \notin S^\star. \tag{29}$$

This is an additional hypothesis on the population problem. It is not implied by the covariance pattern, even though logistic weights are bounded. In particular, the signed expectation $\mathbb{E}[x_j \boldsymbol{x}_B]$ need not control the weighted expectation $\mathbb{E}[w(\boldsymbol{x}^\top \boldsymbol{\theta}) x_j \boldsymbol{x}_B]$: cancellation in the former can disappear in the latter. We make no claim that (D5) follows from Gaussianity alone.

Define

$$\bar{M}_n = \#\{j \notin S^\star : \gamma \|\mathbf{\Sigma}_{j,S^\star}\|_2 > \mu_n\} = m_n \, \mathbf{1}\{\gamma\sqrt{k}|\rho_n| > \mu_n\}. \tag{30}$$

Proposition 8. *Under* (27) *and (D5),* $\mathcal{N}_n \subseteq \{j \notin S^\star : \gamma \|\mathbf{\Sigma}_{j,S^\star}\|_2 > \mu_n\}$*, so* $M_n \le \bar{M}_n$*. In particular,* $M_n \le m_n$ *above the threshold in* (30)*, and* $M_n = 0$ *at or below it.*

Proof. If $j \in \mathcal{N}_n$, then $\Delta_j^{(n)} > \mu_n$. By (D5), $\gamma \|\mathbf{\Sigma}_{j,S^\star}\|_2 \ge \Delta_j^{(n)} > \mu_n$. This proves the set inclusion. Taking cardinalities gives $M_n \le \bar{M}_n$, since M_n was defined as $|\mathcal{N}_n|$, not as an arbitrary larger upper bound. Under (27), the norm is $\sqrt{k}|\rho_n|$ on $\mathcal{C}_n$ and zero on $\mathcal{R}_n$, which proves the displayed formula and both cases. □

Corollary 9. *Assume (D1), (D2), (D5),* (27)*, the sample-size condition* (7)*, and the margin* (11)*. If* $d_n \ge d_n^{\mathrm{blk}} := k + \bar{M}_n$*, then* $\mathbb{P}\{S^\star \subseteq A_{d_n^{\mathrm{blk}}}\} \ge 1 - \delta$*. In the zero-competition case,* $\mathbb{P}\{A_k = S^\star\} \ge 1 - \delta$*.*

Proof. The proposition implies $k + M_n \le k + \bar{M}_n \le d_n$, so the budget requirement in (D4) is satisfied. Theorem 2 gives containment at step $k + M_n$ with probability at least $1 - \delta$. Active sets are nested, so containment persists through step $k + \bar{M}_n$. If $\bar{M}_n = 0$, both sets at step k have size k, and containment is equality. □

The threshold μ_n is an analysis device, not a parameter used by the greedy rule. The bound $\bar{M}_n = m_n$ does not assert $M_n = m_n$ or that every correlated variable is actually selected. Also, the covariance feasibility condition (28) prevents ρ_n and m_n from being chosen independently at arbitrary sizes.

B.2 A FULLY SPECIFIED ZERO-COMPETITION EXAMPLE

Consider $\tilde{\boldsymbol{x}} \sim N(\mathbf{0}, \boldsymbol{I}_p)$ and $\tilde{y} \mid \tilde{\boldsymbol{x}} \sim \mathrm{Bernoulli}\{\sigma(\tilde{\boldsymbol{x}}^\top \boldsymbol{\beta})\}$, where $\boldsymbol{\beta} \ne \mathbf{0}$ has a fixed nonempty support S and fixed nonzero coefficients. Fix $\lambda > 0$.

We first verify that the ridge target has support S. Decompose any coefficient vector as $\boldsymbol{\theta} = t\boldsymbol{\beta} + \boldsymbol{v}$ with $\boldsymbol{v}^\top \boldsymbol{\beta} = 0$. Gaussianity makes $V = \tilde{\boldsymbol{x}}^\top \boldsymbol{v}$ independent of $Z = \tilde{\boldsymbol{x}}^\top \boldsymbol{\beta}$ and of $\tilde{y}$ conditional on Z, with mean zero. Conditional Jensen's inequality for the convex logistic loss gives

$$\mathbb{E}[\log(1 + e^{tZ+V}) - \tilde{y}(tZ + V) \mid Z, \tilde{y}] \ge \log(1 + e^{tZ}) - \tilde{y}\, tZ.$$

The ridge penalty increases by $\lambda\|\boldsymbol{v}\|_2^2/2$, so a minimizer must have $\boldsymbol{v} = \mathbf{0}$. The derivative of the risk restricted to $t\boldsymbol{\beta}$ is

$$h(t) = \mathbb{E}[\{\sigma(tZ) - \sigma(Z)\}Z] + \lambda t \|\boldsymbol{\beta}\|_2^2.$$

Its derivative is $\mathbb{E}[w(tZ)Z^2] + \lambda\|\boldsymbol{\beta}\|_2^2 > 0$. Also, $h(0) < 0$, since $(\sigma(Z) - 1/2)Z > 0$ almost surely except at $Z = 0$, and $h(1) = \lambda\|\boldsymbol{\beta}\|_2^2 > 0$. There is a unique $t_\lambda \in (0, 1)$ with $h(t_\lambda) = 0$. Thus $\boldsymbol{\theta}^\star = t_\lambda \boldsymbol{\beta}$, $S^\star = S$, and (D1) holds with $c_\beta = t_\lambda \min_{j \in S} |\beta_j|$ and $B_\theta = t_\lambda \|\boldsymbol{\beta}\|_2$.

For an arbitrary active set A, the coordinates in $A \setminus S$ are independent of $(\tilde{\boldsymbol{x}}_S, \tilde{y})$ and have mean zero. Applying the same conditional Jensen argument removes their coefficients from the restricted population fit. Hence $\boldsymbol{\theta}^A$ is supported on $A \cap S$. Every interpolation point between $\boldsymbol{\theta}^A$ and $\boldsymbol{\theta}^\star$ is supported on S. For $j \notin S \cup A$, the coordinate $\tilde{x}_j$ is independent of all coordinates appearing in the weight or in $\tilde{\boldsymbol{x}}_{A\cup S}$, and is centered. It follows that

$$\nabla^2_{j,A\cup S} f(\boldsymbol{\theta}(t)) = \mathbb{E}[w(\tilde{\boldsymbol{x}}^\top \boldsymbol{\theta}(t)) \tilde{x}_j \tilde{\boldsymbol{x}}_{A\cup S}^\top] = \mathbf{0}.$$

The ridge cross-block is also zero. Thus all $\Delta_j^{(n)}$ outside S vanish, $M_n = 0$ for any $\mu_n \geq 0$, and the theorem yields recovery in $k = |S|$ steps once its sampling margin holds. This example verifies the assumptions for a concrete data-generating model; it does not claim that arbitrary correlated logistic models have the same property.

C Optimization bounds and their proofs

C.1 Big-M and masked relaxations

The zero vector is feasible for (1), with value $\log 2$. Every minimizer has objective at most $\log 2$. Since the logistic loss is nonnegative,

$$\frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 \leq f_n(\boldsymbol{\theta}) \leq \log 2 \quad \Longrightarrow \quad |\theta_j| \leq R_\lambda := \sqrt{2\log 2/\lambda} \quad \text{for all } j.$$

The same argument applies to every valid selector node: selected coordinates may have zero coefficients, so the zero vector remains feasible. Thus $M_j = R_\lambda$ preserves an optimum at each valid node. More refined valid coefficient bounds may also be used.

A continuous big-M relaxation is

$$\begin{aligned}
\phi_M(N) = \min_{\boldsymbol{\theta}\in\mathbb{R}^p,\, \boldsymbol{z}\in[0,1]^p} \quad & f_n(\boldsymbol{\theta}) && \text{(31a)}\\
\text{subject to} \quad & -M_j z_j \leq \theta_j \leq M_j z_j, \quad j \in [p], && \text{(31b)}\\
& \sum_{j=1}^p z_j \leq k, && \text{(31c)}\\
& z_j = 1 \quad (j \in \mathcal{F}^+), \qquad z_j = 0 \quad (j \in \mathcal{F}^-). && \text{(31d)}
\end{aligned}$$

Replacing $\boldsymbol{z} \in [0,1]^p$ by $\boldsymbol{z} \in \{0,1\}^p$ gives the exact bounded node formulation. Its optimum equals $v(N)$ for the bounds above. A feasible big-M point has $\theta_j = 0$ on $\mathcal{F}^-$. If $|\mathcal{F}^+| = k$, the budget also forces $z_j = 0$ outside $\mathcal{F}^+$, so its coefficients are supported on $D(N)$ in both cases of (12). Hence

$$\phi_0(N) \leq \phi_M(N) \leq v(N). \tag{32}$$

The first inequality may be strict. A masked gradient update alone does not enforce the coupled big-M constraints or solve that relaxation.

Property 10. *For any valid parent and valid child, $\phi_0(\textit{parent}) \leq \phi_0(\textit{child})$. The same inequality holds for ϕ_M.*

Proof. An exclusion removes a coordinate from the permitted set. An inclusion leaves that set unchanged until the selected budget reaches k, at which point the permitted set shrinks to the selected coordinates. Thus $D(\text{child}) \subseteq D(\text{parent})$ in every case, and minimizing over the smaller masked feasible region cannot lower the objective. A big-M child fixes one additional selector relative to its parent, so its feasible region is also a subset. This proves both claims. An infeasible child can be assigned value $+\infty$. □

This monotonicity concerns exact relaxation optima. Independently computed inexact certificates need not be monotone. Replacing a child's certificate by the maximum of that certificate and the inherited parent bound restores monotonic stored bounds without compromising validity.

C.2 A CERTIFIED LOWER BOUND FROM AN UNFINISHED SOLVE

Proposition 3 (A lower bound from a masked gradient)**.** *For a valid node N, let $\boldsymbol{\vartheta}$ be supported on $D = D(N)$ and let $\boldsymbol{g}_D = \nabla_D f_n(\boldsymbol{\vartheta})$. Then*

$$\ell_N(\boldsymbol{\vartheta}) := \max\left\{0,\ f_n(\boldsymbol{\vartheta}) - \frac{\|\boldsymbol{g}_D\|_2^2}{2\lambda}\right\} \le \phi_0(N) \le v(N), \tag{13}$$

where $v(N)$ is the optimal node value. Moreover,

$$0 \le f_n(\boldsymbol{\vartheta}) - \phi_0(N) \le \frac{\|\boldsymbol{g}_D\|_2^2}{2\lambda} \le \frac{|D|\varepsilon_g^2}{2\lambda} \quad \text{if } \|\boldsymbol{g}_D\|_\infty \le \varepsilon_g. \tag{14}$$

Proof. Let $D = D(N)$, and identify a supported vector with its coordinates in $\mathbb{R}^{|D|}$. For every supported $\boldsymbol{u}$, strong convexity implies

$$f_n(\boldsymbol{u}) \ge f_n(\boldsymbol{\vartheta}) + \boldsymbol{g}_D^\top(\boldsymbol{u}_D - \boldsymbol{\vartheta}_D) + \frac{\lambda}{2}\|\boldsymbol{u}_D - \boldsymbol{\vartheta}_D\|_2^2.$$

Completing the square in $\boldsymbol{h} = \boldsymbol{u}_D - \boldsymbol{\vartheta}_D$ gives

$$\boldsymbol{g}_D^\top \boldsymbol{h} + \frac{\lambda}{2}\|\boldsymbol{h}\|_2^2 = \frac{\lambda}{2}\|\boldsymbol{h} + \boldsymbol{g}_D/\lambda\|_2^2 - \frac{\|\boldsymbol{g}_D\|_2^2}{2\lambda}.$$

The minimum over $\boldsymbol{h} \in \mathbb{R}^{|D|}$ is attained at $-\boldsymbol{g}_D/\lambda$. Taking the infimum over $\boldsymbol{u}$ yields

$$\phi_0(N) \ge f_n(\boldsymbol{\vartheta}) - \frac{\|\boldsymbol{g}_D\|_2^2}{2\lambda}.$$

Nonnegativity of the objective gives $\phi_0(N) \ge 0$, proving the first inequality in (13). Every feasible node solution is supported on $D(N)$, so $\phi_0(N) \le v(N)$.

Since $\boldsymbol{\vartheta}$ is itself feasible for the masked problem, $f_n(\boldsymbol{\vartheta}) \ge \phi_0(N)$. Rearranging the preceding lower bound proves the objective-error bound in (14). If $\|\boldsymbol{g}_D\|_\infty \le \varepsilon_g$, then $\|\boldsymbol{g}_D\|_2^2 = \sum_{j\in D} g_j^2 \le |D|\varepsilon_g^2$, which proves the last inequality. For $D = \varnothing$, the masked problem has only the zero vector; the empty gradient has norm zero and all statements hold with equality. □

Why the mask matters. The certificate uses the gradient on D, not a demand that the full gradient vanish. Excluded-coordinate derivatives may remain nonzero at a masked optimum. Using the full gradient in the quadratic correction would still give a possibly weaker bound, but a full-gradient stopping test could fail indefinitely at a valid constrained optimum.

From gradient tolerances to objective tolerances. For a desired local objective tolerance $\varepsilon_{\text{node}} > 0$ and $D \neq \varnothing$, it suffices to choose $\varepsilon_g \le \sqrt{2\lambda\varepsilon_{\text{node}}/|D|}$. Alternatively, test the returned difference $f_n(\boldsymbol{\vartheta}) - \ell_N(\boldsymbol{\vartheta})$ directly. A single numerical gradient tolerance has different objective implications at different mask sizes.

D COMPLETE ALGORITHMIC SPECIFICATION

The following algorithms specify valid bounding, stopping, and safeguarded continuous updates. The mathematical guarantees follow directly from the stated residual certificates and stopping rules.

D.1 BRANCH-AND-BOUND AND GLOBAL GAP BOOKKEEPING

A node-feasible candidate can be formed by selecting a set S that contains $\mathcal{F}^+$, avoids $\mathcal{F}^-$, and has $|S| \le k$, then zeroing all other coefficients and optionally refitting on S. The selection rule affects solution quality, not feasibility. A greedy conditional fit may be used here without removing any feature from the full search.

A selected node remains in the covered search state until it is retired or replaced. At an external time limit, any in-flight node is returned to the queue with its last valid bound before the final global bounds are computed. An iteration limit for a continuous solve does not authorize pruning: the returned residual certificate remains the only available lower bound.

Proposition 11. *In exact arithmetic, Algorithm 1 maintains $LB \le \mathrm{OPT} \le UB$. If it returns after the gap test or after exhausting the queue, the incumbent is ε-optimal. With no external time limit and convergent continuous refinement, the search terminates after finitely many branch operations and refinements to the required positive tolerances.*

Proof. Initially, the root covers every feasible selector assignment and its bound 0 is valid. The zero incumbent is feasible, giving $UB = \log 2$. A branch partitions a parent's feasible selector assignments into its inclusion and exclusion children; infeasible children contain no assignment and need not be stored. The remaining children inherit a valid parent bound, and any added certificate is valid by Proposition 3. The maximum of valid lower bounds is valid.

At any time, the root's assignments are covered by open nodes and retired nodes. The minimum of the bounds over all these nodes is therefore at most the global optimum. The scalar L_{closed} stores precisely the smallest bound among retired nodes, so the global-bound update gives a valid lower bound. Including UB in that minimum keeps $LB \le UB$ even when all assignments are already dominated. Every incumbent update evaluates a feasible sparse vector, so $UB \ge \mathrm{OPT}$ always holds.

Suppose a node is retired by the bound test. Its lower bound is at least the incumbent at that time minus ε. Incumbents only decrease, so its bound remains at least the final $UB - \varepsilon$. At a terminal masked node, every supported vector is node-feasible: its permitted set contains $\mathcal{F}^+$ and has size at most k. After terminal refinement, let $v = f_n(\boldsymbol{\vartheta})$ and ℓ be its certificate. We have $\ell \ge v - \varepsilon$. Updating the incumbent gives $UB_{\mathrm{new}} \le v$, hence $\ell \ge UB_{\mathrm{new}} - \varepsilon$, so terminal retirement has the same property. If the queue is exhausted, all retired bounds are at least $UB - \varepsilon$, and the stated formula yields $UB - LB \le \varepsilon$. If the loop stops by the explicit gap test, that inequality already holds. In either case, $f_n(\boldsymbol{\theta}^{\mathrm{inc}}) - \mathrm{OPT} \le UB - LB \le \varepsilon$.

Each branch fixes an additional previously free selector. There are at most $2^{p+1} - 1$ nodes in the complete binary search tree. Each nonterminal visit either retires or branches. A terminal visit is finished by refinement to a fixed positive local gap tolerance. Under convergent refinement, this takes finitely many iterations. The finite tree and these finite refinements prove termination; this statement is not a polynomial-time bound. □

D.2 Safeguarded batched gradient descent

Backtracking can be evaluated for all columns still requiring a shorter step using the same batched kernels. The algorithm permits a column to be frozen only for its current mask; reassigning a column to a new node resets its history and convergence status. Backtracking can be evaluated for all columns still requiring a shorter step using the same batched kernels. The algorithm permits a column to be frozen only for its current mask; reassigning a column to a new node resets its history and convergence status. The final reevaluation is required even when the loop ends at the maximum iteration count. The final reevaluation is required even when the loop ends at the maximum iteration count.

For completeness, each masked objective has a global Lipschitz gradient constant no larger than $L = \lambda + \|\boldsymbol{X}\|_2^2/(4n)$. The descent lemma gives

$$f_n(\boldsymbol{\theta} - \eta \boldsymbol{g}) \le f_n(\boldsymbol{\theta}) - \eta(1 - L\eta/2)\|\boldsymbol{g}\|_2^2.$$

Thus every step with $\eta \le 2(1-c)/L$ satisfies the Armijo condition. Backtracking terminates, and each accepted positive step is bounded below by $\underline{\eta} = \min\{\eta_{\min}, 2\tau(1-c)/L\} > 0$. The accepted decrease is at least $c\underline{\eta}\|\boldsymbol{g}\|_2^2$. Summing over iterations and using a lower bound on the objective shows that the squared gradient norms are summable, hence tend to zero. Proposition 3 then shows that the local objective–bound gap tends to zero. This proves the convergence needed for terminal refinement in Algorithm 1. It does not apply to arbitrary unsafeguarded step sequences.

E Numerical safeguards and implementation conventions

Stable loss and derivative evaluation. For $u = \boldsymbol{x}_i^\top \boldsymbol{\theta}$ and $y_i \in \{0, 1\}$, evaluate the logistic loss as $\max\{u, 0\} + \log(1 + e^{-|u|}) - y_i u$, or use an equivalent stable softplus operation. Evaluate the

Algorithm 2 Batched masked gradient descent with safeguarded Barzilai–Borwein steps

Require: Masks $\boldsymbol{M} \in \{0,1\}^{p\times B}$, warm starts $\boldsymbol{\Theta}$, maximum updates T, check interval $J = 10$, tolerances $\varepsilon_{g,b}$; $0 < \eta_{\min} \le \eta_0 \le \eta_{\max}$, $c \in (0, 1/2)$, $\tau \in (0, 1)$.
Ensure: Returned coefficients, losses, masked gradients, and lower bounds for all columns.
$\boldsymbol{\Theta} \leftarrow \boldsymbol{\Theta} \odot \boldsymbol{M}$; evaluate losses and masked gradients by (15); initialize all columns as active.
for $t = 0, \dots, T-1$ **do**
  for each active column b, in parallel **do**
    if a valid previous pair exists for the current mask **then**
      $\boldsymbol{s}_b \leftarrow \boldsymbol{\theta}_b - \boldsymbol{\theta}_b^{\mathrm{old}}$; $\boldsymbol{q}_b \leftarrow \boldsymbol{g}_b - \boldsymbol{g}_b^{\mathrm{old}}$.
      Propose $\eta_b \leftarrow \mathrm{clip}(\|\boldsymbol{s}_b\|_2^2/(\boldsymbol{s}_b^\top \boldsymbol{q}_b), \eta_{\min}, \eta_{\max})$ if the quotient and denominator are positive and finite; otherwise use η_0.
    else
      $\eta_b \leftarrow \eta_0$.
    end if
    Set $\boldsymbol{v}_b \leftarrow (\boldsymbol{\theta}_b - \eta_b \boldsymbol{g}_b) \odot \boldsymbol{M}_b$.
    while $f_n(\boldsymbol{v}_b) > f_n(\boldsymbol{\theta}_b) - c\eta_b \|\boldsymbol{g}_b\|_2^2$ **do**
      $\eta_b \leftarrow \tau\eta_b$; recompute $\boldsymbol{v}_b$.
    end while
    Save the old coefficient/gradient pair; set $\boldsymbol{\theta}_b \leftarrow \boldsymbol{v}_b$.
  end for
  Evaluate current masked gradients for the next update.
  if $t+1$ is a multiple of J **then**
    Freeze each column with $\|\boldsymbol{g}_b\|_\infty \le \varepsilon_{g,b}$; stop if all columns are frozen.
  end if
end for
Reevaluate $v_b = f_n(\boldsymbol{\theta}_b)$ and $\boldsymbol{g}_b$ at the returned coefficients for every b.
$\ell_b \leftarrow \max\{0, v_b - \|\boldsymbol{g}_b\|_2^2/(2\lambda)\}$; **return** $\boldsymbol{\Theta}, \{v_b, \boldsymbol{g}_b, \ell_b\}_{b=1}^B$.

sigmoid with a sign-dependent formula to avoid overflow. Masked-out coefficients are explicitly set to zero after updates. Gradient and loss evaluations use the same averaged-loss normalization as (1).

Floating-point lower and upper bounds. The analytic certificate is exact in real arithmetic. Suppose numerical evaluations satisfy $|\widehat{f} - f_n(\boldsymbol{\vartheta})| \le e_f$ and $\|\widehat{\boldsymbol{g}}_D - \nabla_D f_n(\boldsymbol{\vartheta})\|_2 \le e_g$. Then

$$\max\left\{0, \widehat{f} - e_f - \frac{(\|\widehat{\boldsymbol{g}}_D\|_2 + e_g)^2}{2\lambda}\right\} \le \phi_0(N). \tag{33}$$

Indeed, $f_n(\boldsymbol{\vartheta}) \ge \widehat{f} - e_f$ and the gradient triangle inequality gives $\|\nabla_D f_n(\boldsymbol{\vartheta})\|_2 \le \|\widehat{\boldsymbol{g}}_D\|_2 + e_g$; substitute both into the exact certificate. Arithmetic in the final expression must also be enclosed conservatively. For a feasible sparse candidate, $\widehat{f} + e_f$ is a corresponding upper bound. Directed rounding, interval evaluation, or explicit verified arithmetic-error bounds can support these enclosures. Merely using double precision or subtracting an unexplained small constant is not a proof of a machine-valid certificate.

Intercepts. An unpenalized intercept can be incorporated by profiling it out rather than placing it in the ridge-regularized parameter block. Let

$$\bar{f}_n(\boldsymbol{\theta}) = \min_{b\in\mathbb{R}} \left\{ \frac{1}{n}\sum_{i=1}^n \log\bigl(1 + e^{(1-2y_i)(b+\boldsymbol{x}_i^\top\boldsymbol{\theta})}\bigr) + \frac{\lambda}{2}\|\boldsymbol{\theta}\|_2^2 \right\}.$$

When both response classes are present, the one-dimensional intercept minimizer is finite and unique for every $\boldsymbol{\theta}$. Partial minimization of the jointly convex logistic term preserves convexity, so $\bar{f}_n$ is λ-strongly convex in $\boldsymbol{\theta}$. By the envelope theorem, its gradient is the coefficient gradient evaluated at the profiled intercept. Proposition 3 therefore applies verbatim to the profiled objective using the masked coefficient gradient. This keeps the intercept unpenalized while retaining the residual certificate in the sparse coefficient coordinates.

Timing and stopping. GPU timing must include a device synchronization at the timing boundary. A matched-solver comparison should specify whether data transfers, warm starts, branching evaluations, and initialization are included. Relative gaps require a stated normalization and consistent treatment of rounded values. Algorithm 1 uses an absolute gap for its mathematical stopping rule, while the benchmark tables report percentage gaps for comparison across methods.

F Experimental details and complete summary statistics

F.1 Synthetic screening designs

Let $S_\beta = \mathrm{supp}(\boldsymbol{\beta})$ denote the generating support in this experiment. For the greedy sequence and the top-j marginal screened set $\widehat{S}_j^{\mathrm{SIS}}$, respectively, define

$$T_{\mathrm{greedy}} = \min\{t \in [p] : S_\beta \subseteq A_t\}, \qquad T_{\mathrm{SIS}} = \min\{j \in [p] : S_\beta \subseteq \widehat{S}_j^{\mathrm{SIS}}\}.$$

Both quantities count selected features. Their minimum possible value is $|S_\beta|$. They measure support containment, not the number of continuous solver updates or computational time.

Independent signals. Generate $n = 1200$ observations with $p = 100$ independent $N(0, 1)$ covariates. Set $S_\beta = \{1, 2, 3, 4\}$ and $(\beta_1, \beta_2, \beta_3, \beta_4) = (1.3, 1.1, 0.9, 0.8)$, with all other coefficients zero. Generate responses independently conditional on the covariates from $y_i \sim \mathrm{Bernoulli}\{\sigma(\boldsymbol{x}_i^\top \boldsymbol{\beta})\}$.

Correlated suppressors. Generate $n = 1000$ observations with $p = 80$ covariates. For each observation, draw independent standard normals Z_1, Z_2, Z_3 and set $X_1 = Z_1$, $X_2 = Z_2$, and $X_3 = 0.9Z_1 + \sqrt{1 - 0.9^2}\, Z_3$. The remaining coordinates are independent standard normal noise, independent of these three. Set $S_\beta = \{1, 2, 3\}$ and $(\beta_1, \beta_2, \beta_3) = (2, 1, -2)$, with other coefficients zero, and generate logistic responses as above. The opposite signs on two strongly correlated coordinates weaken their marginal association with the response.

Screening rules and settings. SIS ranks the absolute univariate logistic-regression coefficients. Greedy screening uses (4), refitting the restricted ridge-logistic model with $\lambda = 0.1$ at every step. There are 30 independent datasets per design, and support hitting times are recorded without truncation.

Table 2: Complete screening summaries. The range is the observed minimum and maximum over 30 replicates, not a confidence interval.

Design	Method	Mean	Median	Minimum	Maximum
Independent signals	Greedy	4	4	4	4
	SIS	4	4	4	4
Correlated suppressors	Greedy	3.4	3	3	6
	SIS	19.2	11.5	5	69

Figure 1 displays the aggregate summaries in Table 2: means, medians, and observed ranges. The independent-design support equality for a ridge target follows from the Gaussian argument above. For completeness, the correlated suppressor design also has $S_\beta = S^\star$. Let $\rho = 0.9$, $\boldsymbol{u} = (1, 0, -1)^\top$, and write $\boldsymbol{\beta} = 2\boldsymbol{u} + \boldsymbol{e}_2$. On the first three coordinates,

$$\boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0 & \rho \\ 0 & 1 & 0 \\ \rho & 0 & 1 \end{pmatrix}, \qquad \boldsymbol{\Sigma}\boldsymbol{u} = (1 - \rho)\boldsymbol{u}, \qquad \boldsymbol{\Sigma}\boldsymbol{e}_2 = \boldsymbol{e}_2.$$

For $\boldsymbol{x} \sim N(0, \boldsymbol{\Sigma})$ and $q(\boldsymbol{a}) := \mathbb{E}[\sigma'(\boldsymbol{x}^\top \boldsymbol{a})] > 0$, Gaussian Stein's identity gives $\mathbb{E}[\boldsymbol{x}\, \sigma(\boldsymbol{x}^\top \boldsymbol{a})] = q(\boldsymbol{a})\boldsymbol{\Sigma}\boldsymbol{a}$. The population ridge first-order condition is therefore

$$\{q(\boldsymbol{\theta}^\star)\boldsymbol{\Sigma} + \lambda \boldsymbol{I}\}\boldsymbol{\theta}^\star = q(\boldsymbol{\beta})\boldsymbol{\Sigma}\boldsymbol{\beta}.$$

Since $\boldsymbol{u}$ and $\boldsymbol{e}_2$ are eigenvectors of $\boldsymbol{\Sigma}$,

$$\boldsymbol{\theta}^\star = q(\boldsymbol{\beta}) \left[\frac{2(1 - \rho)}{q(\boldsymbol{\theta}^\star)(1 - \rho) + \lambda} \boldsymbol{u} + \frac{1}{q(\boldsymbol{\theta}^\star) + \lambda} \boldsymbol{e}_2 \right].$$

Both displayed coefficients are nonzero, while every noise coordinate has zero population coefficient by independence and symmetry. Hence $\mathrm{supp}(\boldsymbol{\theta}^\star) = \{1, 2, 3\} = S_\beta$.

F.2 Optimization datasets and full table

The optimization study uses Myocardial Infarction Complications ($n = 1700, p = 111$), Madelon ($n = 2000, p = 500$), and Gisette ($n = 6000, p = 5000$) from the UCI repository (Guyon et al., 2004) as well as a sampled dataset of the first 2000 columns of gene expression data ($n = 1815, p = 2000$) for TCGA-BRCA and TCGA-LUAD obtained from the National Cancer Institute Genetic Data Commons (GDC) using the STAR-FPKM expression datasets accessed through the GDC Hub via UCSC Xena (Heath et al., 2021). The main text identifies the GPU and computing framework. Table 3 reports the full-feature branch-and-bound results, including iteration counts, percentage gaps, and runtimes for all tested sparsity budgets.

F.3 Pre-screening bounds and runtimes

Figure 5 reports runtimes for differing levels of pre-screening on the Myocardial, Madelon, Gisette, and TCGA datasets. Figure 6 reports the lower bounds attained for differing levels of pre-screening on those same datasets.

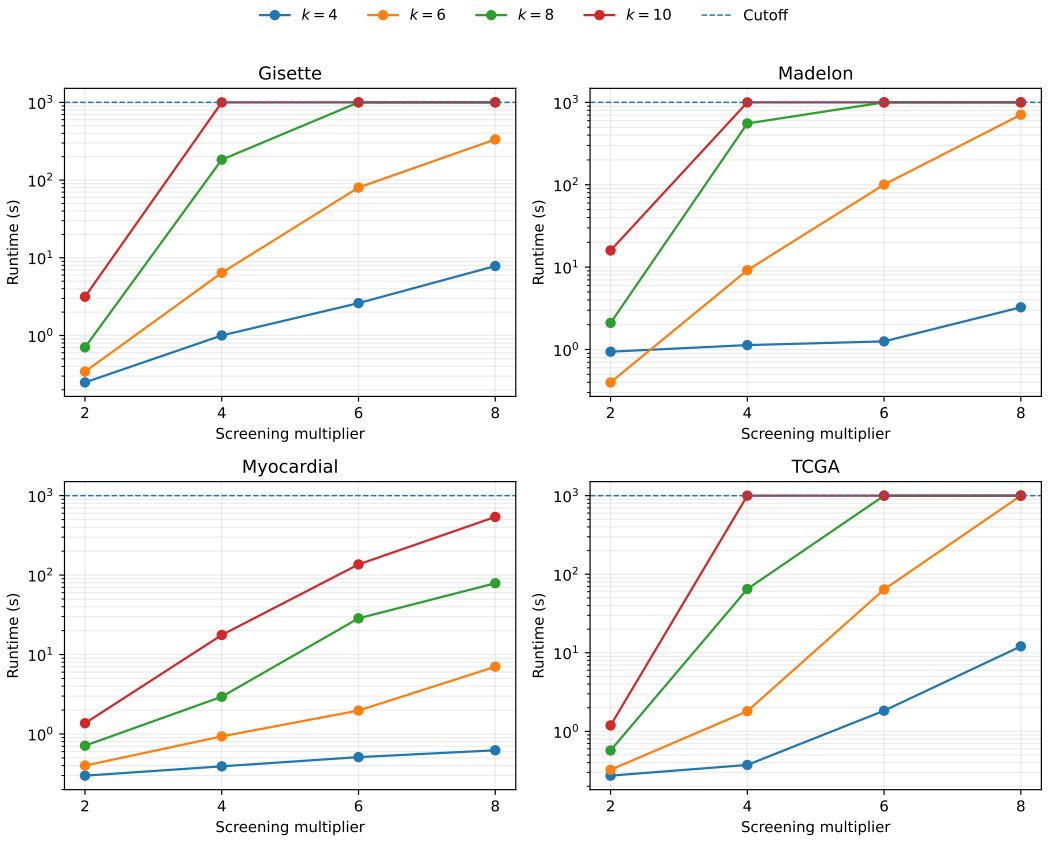


Figure 5: Runtime of greedy pre-screened GPU accelerated B&B for $k = \{4, 6, 8, 10\}$ pre-screened to $M \times k$ features where $M \in \{2, 4, 6, 8\}$. The vertical axis measures total runtime on a log-scale, while the horizontal axis measures the screening multiplier M.

Table 3: Full branch-and-bound records from the supplied master. Time is in seconds. Iterations are reported as recorded by each implementation and are not assumed to be equivalent units of work across solvers.

Dataset	k	Solver	Iterations	Gap (%)	Time (s)
Myocardial	4	CPU B&B	1389	11.52	1000
		MOSEK	1992	2.75	1000
		GPU B&B	142	0.00	74
	6	CPU B&B	1489	10.13	1000
		MOSEK	1951	3.84	1000
		GPU B&B	1826	0.00	227
	8	CPU B&B	1660	8.70	1000
		MOSEK	1954	4.47	1000
		GPU B&B	14344	0.01	957
	10	CPU B&B	1630	7.82	1000
		MOSEK	1935	3.83	1000
		GPU B&B	20289	1.37	1000
Madelon	4	CPU B&B	1926	24.54	1000
		MOSEK	1020	21.29	1000
		GPU B&B	5716	12.61	1000
	6	CPU B&B	1912	24.86	1000
		MOSEK	1019	21.29	1000
		GPU B&B	6517	18.64	1000
	8	CPU B&B	1930	24.96	1000
		MOSEK	1044	21.26	1000
		GPU B&B	7146	19.82	1000
	10	CPU B&B	1959	24.98	1000
		MOSEK	1068	21.26	1000
		GPU B&B	7461	19.20	1000
GISETTE	4	CPU B&B	1	91.09	1000
		MOSEK	5	100.00	1000
		GPU B&B	2055	83.74	1019
	6	CPU B&B	1	99.99	1000
		MOSEK	5	100.00	1000
		GPU B&B	2226	90.05	1017
	8	CPU B&B	1	99.99	1000
		MOSEK	5	100.00	1000
		GPU B&B	2127	94.57	1009
	10	CPU B&B	1	99.99	1000
		MOSEK	5	100.00	1000
		GPU B&B	2227	96.02	1021
TCGA	4	CPU B&B	1	97.62	1000
		MOSEK	107	99.91	1000
		GPU B&B	7455	97.11	1004
	6	CPU B&B	1	99.99	1000
		MOSEK	103	99.89	1000
	8	CPU B&B	1	99.99	1000
		MOSEK	103	99.89	1000
		GPU B&B	8327	99.20	1008
	10	CPU B&B	1	99.99	1000
		MOSEK	83	99.69	1000
		GPU B&B	8427	99.19	1007

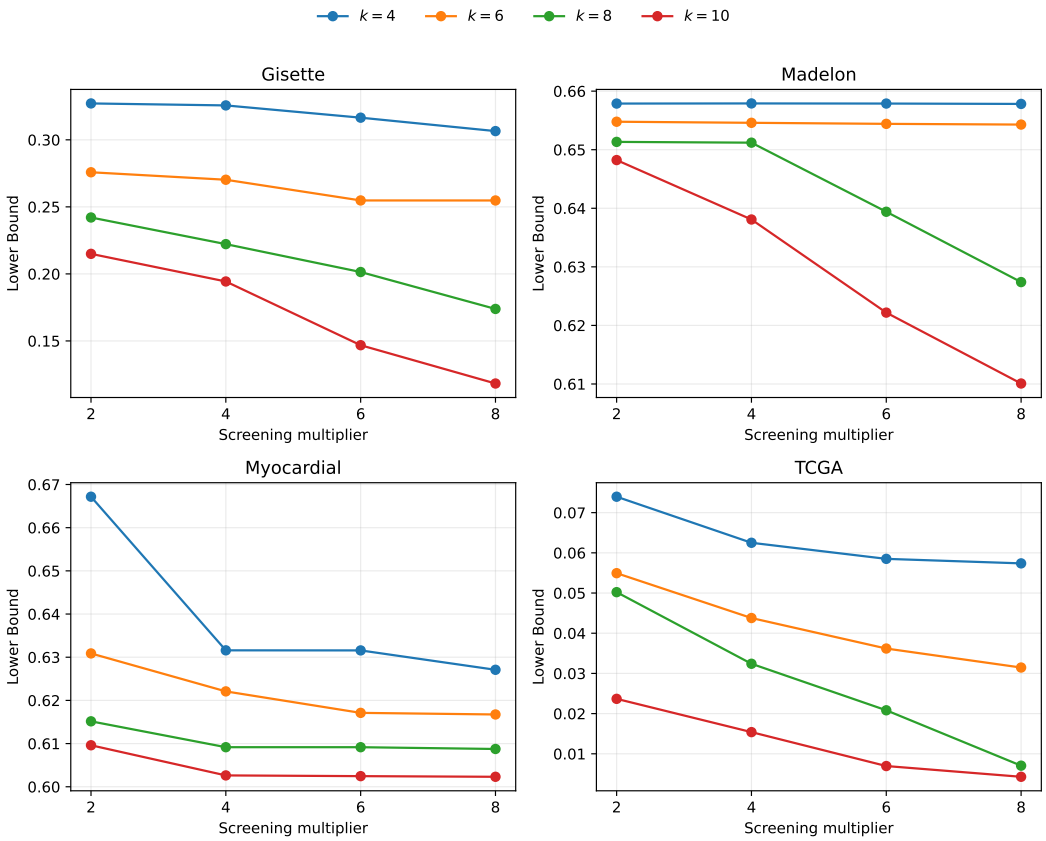


Figure 6: Lower bounds at completion of greedy pre-screened GPU accelerated B&B for $k = \{4, 6, 8, 10\}$ pre-screened to $M \times k$ features where $M \in \{2, 4, 6, 8\}$.