Received: 30 July 2023 | Revised: 14 March 2024 | Accepted: 7 May 2024

DOI: 10.1002/sim.10116

RESEARCH ARTICLE

Statistics in Medicine WILEY

# Structured learning in time-dependent Cox models

**Guanbo Wang**[1] | **Yi Lian**[2] | **Archer Y. Yang**[3,4] | **Robert W. Platt**[5] | **Rui Wang**[6,7] | **Sylvie Perreault**[8] | **Marc Dorais**[9] | **Mireille E. Schnitzer**[8,10]

**Correspondence**
Guanbo Wang, Department of Epidemiology, Harvard T.H. Chan School of Public Health, 677 Huntington Ave, Boston, MA 02115, USA.
Email: gwang@hsph.harvard.edu

Archer Y. Yang, Department of Mathematics and Statistics, McGill University, Montreal, QC, Canada.
Email: archer.yang@mcgill.ca

**Funding information**
Fonds de Recherche du Québec - Nature et Technologies, Grant/Award Number: FRQNT-327788; Natural Sciences and Engineering Research Council of Canada, Grant/Award Number: RGPIN-2016-05174; Heart and Stroke Foundation of Canada, Grant/Award Number: G-17-0018326; Canadian Institutes of Health Research, Grant/Award Number: FDN-143297; Fonds de Recherche du Québec - Santé, Grant/Award Number: FRQS-272161; Réseau Québécois de Recherche sur les Médicaments; Canada Research Chair in Causal Inference and Machine Learning in Health Science; NSERC Discovery Grant.

Cox models with time-dependent coefficients and covariates are widely used in survival analysis. In high-dimensional settings, sparse regularization techniques are employed for variable selection, but existing methods for time-dependent Cox models lack flexibility in enforcing specific sparsity patterns (ie, covariate structures). We propose a flexible framework for variable selection in time-dependent Cox models, accommodating complex selection rules. Our method can adapt to arbitrary grouping structures, including interaction selection, temporal, spatial, tree, and directed acyclic graph structures. It achieves accurate estimation with low false alarm rates. We develop the `sox` package, implementing a network flow algorithm for efficiently solving models with complex covariate structures. `sox` offers a user-friendly interface for specifying grouping structures and delivers fast computation. Through examples, including a case study on identifying predictors of time to all-cause death in atrial fibrillation patients, we demonstrate the practical application of our method with specific selection rules.

**KEYWORDS**
grouping structures, high-dimensional data, network flow algorithm, structured sparse regularization, structured variable selection, survival analysis, time-dependent Cox models

## 1 | INTRODUCTION

The Cox model[1] is a well-established statistical model widely used for survival data analysis. Incorporating time-dependent covariates and coefficients in the Cox model offers more flexibility in representing associations between covariates and the hazard of the event of interest. Examples of time-varying covariates include medication usage[2] and disease status.[3] Integrating time-varying coefficients into the Cox model is particularly relevant in cases where the relationship between covariates and the outcome of interest changes over time.

In many real-world applications of time-dependent Cox models, the number of covariates can be very large, potentially exceeding the number of observations in the data. To address the challenges of model overfitting and perform variable selection in such high-dimensional settings, sparse regularization techniques can be employed. These techniques help remove redundant covariates from the model and improve estimation/prediction accuracy. For example, LASSO and

Guanbo Wang and Yi Lian are co-first author.

For affiliations refer to page 3180.

SCAD regularization methods have been extensively studied for Cox models.[4,5] In the context of time-dependent covariates and coefficients, some variable selection methods in the Cox model have been proposed.[6,7] However, these methods only select variables individually and do not enforce specific sparsity patterns on the covariates.

Investigators often have prior knowledge about the structure of potential model covariates, which imposes certain restrictions on how covariates should be included in the model. For example, strong heredity states that "if the interaction term is selected, then the main terms should also be selected."[8] To incorporate such information, which we refer to as "selection rules," a penalty can be applied to a weighted sum of the norms of group variable coefficients. Different specifications of grouping structures correspond to different selection rules. Complicated selection rules usually require that the groups are overlapped. However, many existing variable selection methods, such as the group LASSO and the sparse group LASSO, do not allow overlapped groups and thus cannot handle complex selection rules. In the context of Cox models without time-varying covariates or coefficients, various methods have been proposed to incorporate specific types of selection rules. For example, Wang et al[9] introduced methods to incorporate strong heredity in interaction selection, while Simon et al[10] and Wang et al[11] developed sparse group LASSO techniques. Additionally, Dang et al[12] extended the latent overlapping group LASSO[13] to the Cox model, which requires specifying latent variables.

While the method mentioned above may adhere to certain selection rules, it faces challenges in scaling to high-dimensional settings with complex grouping structures due to its built-in algorithms. For example, the method is not well-suited for scenarios where multi-layer groups (such as tree and graph structures) exhibit significant overlap and the sparsity level is low.[14] Additionally, none of the aforementioned methods can perform structural variable selection for time-dependent Cox models. The absence of such a method and software may hinder investigators from fully leveraging their prior knowledge about the structure of potential covariates when analyzing (time-dependent) Cox models. This could lead to compromised prediction accuracy and yield misleading and uninterpretable results regarding variable selection.

We contribute to this field of research in several ways. First, we propose the first application of the structured sparsity-inducing penalty[15] to time-dependent Cox models. Our method can easily adapt to arbitrary grouping structures, allowing for the incorporation of highly complex selection rules. This flexibility enables the inclusion of various structures such as interaction selection, temporal, spatial, tree, and directed acyclic graph structures. Our estimator demonstrates low false alarm rates and high estimation accuracy.

Second, to reduce the computational burden caused by selecting time-dependent covariates with complex grouping structures required by our method, we develop a network flow algorithm that efficiently and effectively solves models with complex covariate structures. To ensure optimal efficiency, we implement this algorithm as an `R` package called `sox`, (stands for structured learning for time-dependent Cox),[16] which is available on CRAN. Our software leverages established `SPAMS` packages and provides users with a user-friendly interface to specify arbitrary grouping structures. It has a `C++` core and offers fast computational speed and reliable performance.

Finally, we provide examples that illustrate how to specify grouping structures to respect complex selection rules in practical scenarios. In particular, in a case study, we apply our developed method to identify significant predictors associated with the time to death by any cause among hospitalized patients with atrial fibrillation. In this analysis, we incorporate eight selection rules and demonstrate the rationale behind specifying the corresponding grouping structure.

The rest of the article is organized as follows. In Section 2, we introduce the proposed method. Section 3 illustrates the algorithm for `sox`, followed by implementation details in Section 4. We then present the results of simulation studies to compare our method to unstructured variable selection in both low and high dimensional settings in Section 5, and present the application of `sox` in the case study in Section 6. We conclude with a discussion in Section 7.

## 2 | MODEL SPECIFICATION AND STRUCTURED PENALIZATION

Consider individual failure times $T_i$ and censoring times $C_i$ indexed by $i = 1, \ldots, n$. We can observe only the time to either failure or censoring, whichever comes first, that is, $U_i = \min(T_i, C_i)$ with censoring indicator $\delta_i = I(T_i \leq C_i)$. We also consider a possibly time-varying, $p$-vector-valued covariate process $\mathbf{X}_i(t) = (X_{i1}(t), \ldots, X_{ip}(t))^\top$. We assume noninformative censoring, that is, upon conditioning on $\mathbf{X}_i(t)$, $C_i$ is independent of $T_i$. Therefore the observed data associated with $n$ individuals are $n$ triplets $\{U_i, \delta_i, \mathbf{X}_i(t)\}$ for $i = 1, \ldots, n$, which we assume to be independently drawn from a common distribution. Denote by $h(t|\cdot)$ the covariate-conditional hazard function; we assume that $h_i(t) = h_0(t) \cdot \exp\{\mathbf{X}_i(t)^\top \boldsymbol{\beta}\}$, where $h_0(\cdot)$ is an unspecified baseline hazard function. We begin by considering the scenario in which the coefficient vector, denoted as $\boldsymbol{\beta} \in \mathbb{R}^p$, is time-invariant. Subsequently, we will explore the case of time-varying coefficients.

To handle tied-events, we define an index $\ell = 1, \ldots, L$ for the ordered unique follow-up times in the dataset, and an ordered list $t_1 < t_2 < \cdots < t_L$ of unique time-to-event realizations. The number of tied events occurring at the $\ell$th distinct survival time is denoted by $d_\ell$. We can further define two index sets, $D_\ell$ and $R_\ell$, representing the subjects whose event occurred at time $t_\ell$ and were at risk at time $t_j$, respectively. Using the Breslow approximation to accommodate tied events,[17] the negative log partial likelihood can be approximated as

$$f(\boldsymbol{\beta}) \approx -\sum_{\ell=1}^{L}\left(\left\{\sum_{i\in D_\ell}\mathbf{X}_i(t_\ell)^\top\right\}\boldsymbol{\beta} - d_\ell \log\left[\sum_{i\in R_\ell}\exp\{\mathbf{X}_i(t_\ell)^\top\boldsymbol{\beta}\}\right]\right). \tag{1}$$

Up until now, we assume the coefficients $\boldsymbol{\beta}$ are time-invariant, but there are several ways to accommodate the case where $\boldsymbol{\beta}$ depends on time $t$. For example, for $j = 1, \ldots, p$, let $\beta_j(t)$ be $\sum_{m=1}^{M} I(T_m \leqslant t < T_{m+1})\beta_{jm}$ (with specified time intervals $[T_m, T_{m+1}), m = 1, \ldots, M$) or $a_j + b_j \log(t)$, as functions of $t$. More flexibly, let $\beta_j(t) = \sum_{m=1}^{M}\theta_{jm}\phi_m(t)$, where $\phi_m(\cdot)$ is a set of B-spline basis functions for approximating the function $\beta_j(t)$. The problem of estimating $\beta_j(t)$ is then transformed to the problem of estimating $a_j, b_j, \boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jm})^\top$, or $\boldsymbol{\theta}_j = (\theta_{j1}, \ldots, \theta_{jM})^\top$. For the sake of simplicity, we adopt the notation of time-invariant coefficients, as given in Equation (1), as the primary framework in this article. But we note that Cox models with time-dependent coefficients can be viewed as a specific instance in the broader context. For a detailed explanation, please refer to Example 3.

To incorporate a selection rule into selecting time-dependent covariates, one approach is to enforce the collective selection of groups of covariates. However, some previous approaches have imposed restrictions on the grouping structure,[10,18] such as the prohibition of overlap between groups. In contrast, this work adopts a flexible approach by imposing no constraints on the grouping structure, allowing for the consideration of a wider range of selection rules.[15] Let $\mathbb{V}(t) = \{X_1(t), X_2(t), \ldots, X_p(t)\}$ denote the set containing all the covariates (for the brevity, we will use $\mathbb{V}$ throughout the article). Suppose there are $K$ pre-defined groups of these covariates, and let us define the grouping structure as $\mathbb{G} = \mathbb{g}_k, k = 1 \ldots, K$, where $\mathbb{g}_k$ represents a group—a non-empty subset of $\mathbb{V}$—and the union of all $\mathbb{g}_k$'s is equal to $\mathbb{V}$. It is worth noting that the groups can overlap, meaning that $\mathbb{g}_j \cap \mathbb{g}_k$ may not be empty for $j \neq k$. To denote a vector of the same length as $\boldsymbol{\beta}$, with non-zero entries corresponding to the covariates in $\mathbb{g}$ and zero entries elsewhere, we use $\boldsymbol{\beta}_{|\mathbb{g}}$.

To select variables according to the pre-defined groups to achieve structural selection in time-dependent Cox models, we solve the following problem

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) + \lambda\Omega(\boldsymbol{\beta}), \quad \Omega(\boldsymbol{\beta}) = \sum_{\mathbb{g}\in\mathbb{G}}\omega_{\mathbb{g}}\left\|\boldsymbol{\beta}_{|\mathbb{g}}\right\|_d. \tag{2}$$

Here, $f(\boldsymbol{\beta})$ is a convex differentiable function as defined in Equation (1). The weight $\omega_{\mathbb{g}}$ is a positive user-defined value associated with the group $\mathbb{g}$, and $\Omega(\boldsymbol{\beta})$ represents the weighted sum of sparsity-inducing $\ell_d$ norms ($d = 2$ or $\infty$) applied to groups of coefficients $\boldsymbol{\beta}_{|\mathbb{g}}$, where $\mathbb{g} \in \mathbb{G}$. The choice of norm can be either the $\ell_\infty$ norm (which corresponds to the maximum absolute value of $\boldsymbol{\beta}_{|\mathbb{g}}$) or the $\ell_2$ norm. Both norms serve the purpose of encouraging the collective selection of a group of variables. The choice of $d = 2$ or $d = \infty$ has been studied, and they produce similar results regarding prediction accuracy and selection consistency.[19-21] Since the choice of norm is not the main focus of this article, we primarily focus on the $\ell_\infty$ norm.

By employing this type of penalization, each group of variables can be excluded from the model as a group, thereby promoting sparsity. The specifications of the grouping structure $\mathbb{G}$, determining the membership of variables in each group $\mathbb{g}$, leads to different sets of variables that can be selected (ie, the complement of the union of the groups). This allows for incorporating various a priori knowledge or structures exhibited in the real data. Importantly, we allow for the inclusion of highly overlapped groups, enabling the inclusion of a wide range of structures.

To operationalize the structures in a mathematical and explicit manner, we initially translate them into selection rules, which represent the dependencies among variables. Subsequently, we specify the grouping structure $\mathbb{G}$ to adhere to these selection rules. This approach allows us to articulate and identify the structures to be incorporated effectively. While we do not delve into the detailed explanation in this article, interested readers can find further information in existing literature.[22,23]

Various types of selection rules can be followed by the structured sparsity-inducing penalty, such as strong heredity, temporal and spatial structures, and rules that require tree or graph grouping structures. We next provide five detailed examples of such selection rules and their related grouping structure specifications. More examples of selection rules can be found in References 15 and 24.

**Example 1** (Coefficients interpretability). Consider a cohort study in which patients intermittently receive a drug (treatment) at different dose levels. The objective is to investigate the association between dose level and time-to-event. Let $X_1(t)$ and $X_2(t)$ represent indicators for the patient receiving treatment (either high or low dose) at time $t$ and receiving the high dose treatment, respectively. A crucial selection rule in this scenario is "if $X_2(t)$ is selected, then $X_1(t)$ must also be selected." This is because if $X_2(t)$ is selected without $X_1(t)$, then the coefficients of $X_2(t)$ would be the contrast between taking high-dose treatment vs taking low-dose treatment combined with not taking the treatment, which is not of interest. Incorporating such selection rules guarantees the interpretability of the selected model's coefficients. To satisfy this selection rule, the grouping structure $\mathbb{G}$ is specified as $\{\{X_2(t)\}, \{X_1(t), X_2(t)\}\}$. It is worth noting that even this simple selection rule necessitates an overlapping grouping structure.

**Example 2** (Strong heredity). As mentioned earlier, caution is required when dealing with interaction terms in variable selection. Consider a study involving covariates $X_1(t), X_2(t)$ and their interaction $X_3(t) = X_1(t) \times X_2(t)$. The strong heredity[8] states that "If the interaction term is selected, then all its main terms must also be selected." We specify the grouping structure as $\mathbb{G} = \{\{X_3(t)\}, \{X_1(t), X_3(t)\}, \{X_2(t), X_3(t)\}\}$. Incorporating such selection rules can enhance the interpretability of the model, improve statistical power, and simplify experimental designs.[25] In our simulation studies, we present additional examples of grouping structures for more complex scenarios. For instance, in Section 5.1, we demonstrate the grouping structure for the case involving interactions between two categorical variables and a continuous variable. In Section 5.2, we showcase the strategy for setting the grouping structure when high-dimensional main terms and interactions are included.

**Example 3** (Incorporating temporal structure). In studies involving patients with dementia, it is common to categorize them into four phases: mild, moderate, moderately severe, and severe cognitive decline.[26] Consider a cohort study focusing on patients diagnosed with mild cognitive decline and examining the association between blood pressure $X(t)$ at each phase and the time to severe cognitive decline. We define $T_1$ as the time of diagnosis for moderate cognitive decline and $T_2$ as the time for moderately severe cognitive decline. To capture the temporal aspect, we include the time-dependent covariates $Z_1(t) = I(t < T_1)X(t)$, $Z_2(t) = I(T_1 \leqslant t < T_2)X(t)$ and $Z_3(t) = I(t \geqslant T_2)X(t)$ in the Cox model. Suppose we have a priori knowledge that if the blood pressure in a previous phase is associated with the outcome, the blood pressure in the later phase should also be associated. This implies the following selection rules: "if $Z_1(t)$ is selected, then $Z_2(t)$ and $Z_3(t)$ should also be selected" and "if $Z_2(t)$ is selected, then $Z_3(t)$ should be selected." By incorporating such selection rules, we can accommodate the temporal structure of the time-dependent covariate. The corresponding grouping structure is $\mathbb{G} = \{\{Z_1(t)\}, \{Z_1(t), Z_2(t)\}, \{Z_1(t), Z_2(t), Z_3(t)\}\}$. This example illustrates the use of step functions to model time-dependent associations.

More flexibly, consider the Cox model with both time-dependent covariates and coefficients, $h\{t|\mathbf{X}(t)\} = h_0(t) \cdot \exp\{\mathbf{X}(t)^\top \boldsymbol{\beta}(t)\}$, where $\boldsymbol{\beta}(t) = \{\beta_1(t), \ldots, \beta_j(t)\}^\top$ with $\beta_j(t) = \sum_{m=1}^{M} \theta_{j,m}\phi_m(t)$, for $j = 1, \ldots, p$. Rewriting, we have: $h\{t|\mathbf{Z}(t)\} = h_0(t) \cdot \exp\{\mathbf{Z}(t)^\top \boldsymbol{\theta}\}$, where $Z_{j,m}(t) = X_j(t)\phi_m(t)$, for example,

$$\mathbf{Z}(t) = (\underbrace{X_1(t)\phi_1(t), \ldots, X_1(t)\phi_M(t)}_{\mathbf{Z}_{1,\cdot}(t)}, \ldots, \underbrace{X_p(t)\phi_1(t), \ldots, X_p(t)\phi_M(t)}_{\mathbf{Z}_{p,\cdot}(t)})^\top = (\mathbf{Z}_{1,\cdot}(t), \ldots, \mathbf{Z}_{p,\cdot}(t))^\top,$$

which has $p \times M$ elements. Thus, in the context of predictor identification, selecting $X_j$ is equivalent to selecting $\mathbf{Z}_{j,\cdot}(t)$. In the context of estimation, estimating $\boldsymbol{\beta}(t)$ equates to estimating the $p \times M$ length vector $\boldsymbol{\theta} = (\boldsymbol{\theta}_{1,\cdot}, \ldots, \boldsymbol{\theta}_{p,\cdot})$, where $\boldsymbol{\theta}_{j,\cdot} = \{\theta_{j,1}, \ldots, \theta_{j,M}\}$. Suppose the selection rule to be respected is "all the basis functions related to one variable should be selected collectively" or "$\mathbf{Z}_{j,\cdot}(t) = \{Z_{j,1}(t), \ldots, Z_{j,M}(t)\}, \forall j = 1, \ldots, p$ should be selected collectively." By incorporating such selection rules, we can avoid the selection discrepancy among basis functions while accommodating the time-dependent coefficient feature in Cox models. We can define $\mathbb{V} = \{Z_{j,m}(t), j = 1, \ldots, p, m = 1 \ldots, M\}$ and $\mathbb{G} = \{\mathbf{Z}_{j\cdot}(t), j = 1 \ldots, p\}$, and the selection rule can be respected.

**Example 4** (Incorporating spatial structure). Suppose that high-dimensional voxel signals over time in patients' brains are collected in functional magnetic resonance imaging (fMRI) data. The goal is to identify the regions (of voxels) and individual voxels associated with the time to cocaine relapse[27] or Alzheimer's disease

progression.[28] Variable selection conducted to analyze such data should be informed by the three-dimensional grid structure (such as localized clusters on the brain).[29] Here we take a simplified example for illustration. Suppose there are three contiguous voxels; the intensities of the signals are denoted $X_1(t)$, $X_2(t)$, and $X_3(t)$. Consider the case where two parcels (clusters) $\{X_1(t), X_2(t)\}$ and $\{X_1(t), X_2(t), X_3(t)\}$ are hierarchically constructed,[30] in which only neighboring voxels can be merged together. Larger parcels can be regarded as potential regions of interest. To incorporate such a structure, we first define $X_4(t)$ and $X_5(t)$ as the average of the two parcels. Then we set selection rules as "if either $X_1(t)$ or $X_2(t)$ is selected, then select $X_4(t)$," and "if either $X_3(t)$ or $X_4(t)$ is selected, then select $X_5(t)$" to encode the hierarchy and promote the parcel selection.[31] The corresponding $\mathbb{G}$ is $\{\{X_1(t)\}, \{X_2(t)\}\{X_3(t)\}, \{X_1(t), X_2(t), X_4(t)\}, \{X_1(t), X_2(t), X_3(t), X_4(t), X_5(t)\}\}$. The above example represents the case where two parcels are nested. Incorporating more complex structures, such as multiple overlapped parcels nesting in different larger parcels, is also possible.

**Example 5** (Tree and directed acyclic graph grouping structures). In certain real-world scenarios, more complex structures of covariates, such as trees[21] and directed acyclic graphs,[15] can be incorporated into variable selection. These structures allow for more intricate relationships among variables to be taken into account. For example, in our model, covariates can be represented as nodes in a tree, and users can specify that a variable is selected only if all its ancestors in the tree are already selected. Moreover, the framework can be further extended to include directed cyclic graphs, which have been found useful in hierarchical variable selection. These enhancements provide additional flexibility in capturing complex relationships among variables.

In the next section, we present an efficient algorithm for solving the objective function (2) for time-dependent Cox models, allowing for the incorporation of selection rules that can be followed by the structured sparsity-inducing penalty.

## 3 | PROXIMAL GRADIENT WITH NETWORK FLOW ALGORITHM

In this section, we illustrate the use of a proximal gradient algorithm[32] with network flow to solve (2) with a structural penalty.

Since $\Omega(\boldsymbol{\beta})$ is not differentiable on its entire support, the optimization of the penalized likelihood requires the proximal method. The proximal method[33] has been successfully applied in various research areas, including signal processing[34] and machine learning.[35]

To address the computational challenge posed by the non-smooth component in the objective function, the proximal method updates estimates that remain close to the gradient update for the differentiable function $f(\boldsymbol{\beta})$, while also minimizing the non-differentiable penalty term.[36] This approach enjoys a linear convergence rate.[37,38] More specifically, the updated value of $\boldsymbol{\beta}$ in each iteration of the proximal gradient algorithm, denoted as $\boldsymbol{\beta}^+$, is obtained by minimizing the following approximated problem. Here, the loss function $f$ is approximated by a quadratic function:

$$\begin{aligned}\boldsymbol{\beta}^+ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}}\, f(\tilde{\boldsymbol{\beta}}) + (\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}})\nabla f(\tilde{\boldsymbol{\beta}}) + \frac{1}{2q}\|\boldsymbol{\beta} - \tilde{\boldsymbol{\beta}}\|_2^2 + \lambda\Omega(\boldsymbol{\beta}) \\ &= \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2q}\|\boldsymbol{\beta} - \{\tilde{\boldsymbol{\beta}} - q\nabla f(\tilde{\boldsymbol{\beta}})\}\|_2^2 + \lambda\Omega(\boldsymbol{\beta}),\end{aligned}$$

where $\tilde{\boldsymbol{\beta}}$ is the value of $\boldsymbol{\beta}$ from the previous iteration, and $t$ is the step size of the update. By defining $\boldsymbol{u} = \tilde{\boldsymbol{\beta}} - q\nabla f(\tilde{\boldsymbol{\beta}})$, the above problem can be further written as

$$\boldsymbol{\beta}^+ = \operatorname{prox}_{q\lambda\Omega}\{\tilde{\boldsymbol{\beta}} - q\nabla f(\tilde{\boldsymbol{\beta}})\} \;:=\; \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{\beta} - \boldsymbol{u}\|_2^2 + q\lambda\sum_{\mathrm{g}\in\mathbb{G}}\omega_{\mathrm{g}}\left\|\boldsymbol{\beta}_{|\mathrm{g}}\right\|_\infty. \tag{3}$$

In many cases, the proximal operator can be computed in a closed form, leading to efficient computations. However, for more complex structures such as nested groups (eg, tree structures) or general directed acyclic graphs (eg, Example 1 and the first one in Example 3), the closed-form proximal operator may not exist. In the case of a tree structure, the proximal operator can still be efficiently computed using its dual form in a blockwise coordinate ascent fashion.[21] However, dealing with general directed acyclic graphs presents a greater challenge. To address this issue, Marial et al[24] converted

the dual form of the proximal operator into a quadratic min-cost flow problem, enabling efficient computations in the context of such graphs.

Define the dual variables $\boldsymbol{\xi}_{|\mathbb{g}}$, which satisfies $\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}_{|\mathbb{g}} = \boldsymbol{\beta}$.

According to Lemma 2 of Marial et al,[24] the dual of problem (3) is

$$\min_{\boldsymbol{\xi}} \frac{1}{2}\left\| \boldsymbol{u} - \sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}_{|\mathbb{g}} \right\|_2^2, \text{ s.t. } \forall \mathbb{g} \in \mathbb{G}, \left\| \boldsymbol{\xi}_{|\mathbb{g}} \right\|_1 \leqslant \lambda\omega_{\mathbb{g}} \text{ and } \boldsymbol{\xi}_{|\mathbb{g},j} = 0 \text{ if } j \notin \mathbb{g}, \tag{4}$$

where $\omega_{\mathbb{g}}$ is the same as in (3). The proof can be found in Jenatton et al.[21,39] The dual problem can be transformed into a quadratic min-cost flow problem.[40] This conversion allows us to efficiently solve the problem using a network flow algorithm based on the mini-cut theorem.[41] More details can be found in Web Appendix A. The algorithm converges in a finite and polynomial number of operations, providing an effective solution to the dual problem.

The network flow algorithm, commonly used in graph models,[42] has found widespread application in various machine learning domains, such as in image processing.[43] Because of the special form of the constrain in our problem, we are able to present a more efficient version of the algorithm,[24] referred to as Algorithm 1, for solving (4). The central computation in the algorithm is the evaluation of $\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}_{|\mathbb{g}}$, accomplished by the `computeFlow` function. The details of the algorithm's steps are presented in Web Appendix B.

**Algorithm 1.** Solving (4)

**Inputs:** The estimate in the $k$th step $\boldsymbol{\beta}^k \in \mathbb{R}^p$, step size $q$, $\mathbb{V}$, $\mathbb{G}$, $\omega_{\mathbb{g}}$, $\lambda$.Set $\boldsymbol{\xi} = 0$.
Compute $\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}_{|\mathbb{g}} \leftarrow$ `computeFlow`$(\mathbb{V}, \mathbb{G})$.
**return** $\boldsymbol{\beta}^k - q\nabla f(\boldsymbol{\beta}^k) - \sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}_{|\mathbb{g}}$
**Function** `computeFlow`$(\mathbb{V}, \mathbb{G})$
Projection: $\boldsymbol{\gamma} \leftarrow \text{argmin}_{\boldsymbol{\gamma}} \sum_{j:X_j\in\mathbb{V}} \frac{1}{2t}(\beta_j^k - q\nabla f(\beta_j^k) - \gamma_j)^2$ s.t. $\sum_{j:X_j\in\mathbb{V}} \gamma_j \leqslant \lambda \sum_{\mathbb{g}\in\mathbb{G}} \omega_{\mathbb{g}}$
Updating: $(\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}})_{X_j\in\mathbb{V}} \leftarrow \text{argmax}_{(\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}})_{X_j\in\mathbb{V}}} \sum_{X_j\in\mathbb{V}} \sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}}$ s.t. $\sum_{X_j\in\mathbb{g}} \boldsymbol{\xi}^j_{|\mathbb{g}} \leqslant \lambda\omega_{\mathbb{g}}$
Recursion:
**if** $\exists X_j \in \mathbb{V}$ *s.t.* , $\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}} \neq \gamma_j$ **then**
  Denote $\mathbb{V}^* = \{X_j \in \mathbb{V} : \sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}} = \gamma_j\}$, and $\mathbb{g}^* = \{\mathbb{g} \in \mathbb{G} : \sum_{X_j\in\mathbb{g}} \boldsymbol{\xi}^j_{|\mathbb{g}} < \lambda\omega_{\mathbb{g}}\}$
  $(\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}})_{X_j\in\mathbb{V}^*} \leftarrow$ `computeFlow`$(\mathbb{V}^*, \mathbb{G}^*)$
  $(\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}})_{X_j\in\mathbb{V}\setminus\mathbb{V}^*} \leftarrow$ `computeFlow`$(\mathbb{V} \setminus \mathbb{V}^*, \mathbb{G} \setminus \mathbb{G}^*)$
**end**
**return** $(\sum_{X_j\in\mathbb{g}} \boldsymbol{\xi}^j_{|\mathbb{g}})_{X_j\in\mathbb{V}}$

## 4 | IMPLEMENTATION DETAILS

We provide an efficient and user-friendly `R` package `sox`, which is available on CRAN (https://cran.r-project.org/package=sox). The statistical software is implemented in `C++` with the incorporation of programs adapted from well-established software packages including the `survival` ,[44] `glmnet`[45,46] and `SPAMS`[24] to ensure optimal efficiency. It provides users with a convenient interface to specify the grouping structure relevant to their specific data analysis task. In addition, the `sox` features built-in solution path and cross-validation functions with their corresponding visualization tools to facilitate model tuning and diagnostics.

### 4.1 | Details of implementing the max flow algorithm

We utilize the max flow algorithm, as proposed by Goldberg and Tarjan,[47] for the efficient execution of the updating step within the `computeFlow` process (Table 1). To our knowledge, it remains unmatched in terms of speed and effectiveness

**TABLE 1** Corresponding inputs of the max flow algorithm.

| Type | From | To | Flow $f$ | Capacity $c$ |
|---|---|---|---|---|
| 1 | $s_1$ | $\mathrm{g}_k \in \mathbb{G}$ | $\sum_{X_j \in \mathbb{V}} \sum_{\mathrm{g} \in \mathbb{G}} \xi^j_{\mid \mathrm{g}}$ | $\lambda \omega_{\mathrm{g}}$ |
| 2 | $\mathrm{g}_k$ | $X_j \in \mathrm{g}_k$ | $\xi^j_{\mid \mathrm{g}}$ | $\infty$ |
| 3 | $X_j \in \mathbb{V}$ | $s_2$ | $(\sum_{\mathrm{g} \in \mathbb{G}} \xi^j_{\mid \mathrm{g}})_{X_j \in \mathbb{V}}$ | $\infty$ |

for solving max-flow problems. This algorithm has been effectively integrated into our package `sox`, using the `SPAMS` packages with a `C++` core. In the following section, we provide a concise introduction to this algorithm.

The algorithm first initializes and relabels the distance, and then pushes excess from the vertex whose flow equals the capacity on each edge and can reach the sink to vertices that have a shorter estimated distance to the sink $s_2$, with the goal of getting as much excess as possible to $s_2$. When a vertex cannot reach the sink with a positive excess, the algorithm pushes such excess in the opposite direction. In each update, the value of the flow function is changed. Eventually, all vertices other than the source and sink have zero excess while each arc respects the capacity. At this point, the flow is a maximum flow, and thus, the value of the flow function can be computed as the updated value in the updating step in Algorithm 1. More details are given in Web Appendix C.

## 4.2 | Backtracking line search

The proposed algorithm for solving the dual of the proximal operator includes the step size $q$, which enables the incorporation of backtracking line search. The algorithm, presented in Algorithm 2, allows us to solve (3) using proximal gradient descent with backtracking line search.[48] Backtracking line search is an optimization technique that helps determine the appropriate step size. It begins with a predefined step size for updating along the search direction and iteratively shrinks the step size (ie, "backtracks") until the decrease in the loss function corresponds reasonably to the expected decrease based on the local gradient of the loss function. This technique enhances the convergence speed of the algorithm.

**Algorithm 2.** Solving (3) using proximal gradient descent with backtracking line search

**Inputs:** $X_i(t)$, $T_i$, $\delta_i$, $\mathbb{V}$, $\mathbb{G}$, $\omega_{\mathrm{g}}$, $\lambda$, convergence threshold $r$, shrinkage rate $\alpha < 1$, step size $q$. Set $\boldsymbol{\beta}^0 = \mathbf{0}, k = 0$.

**repeat**

  $\boldsymbol{\beta}^+ \leftarrow \mathrm{prox}_{q\lambda\Omega}\left(\boldsymbol{\beta}^k - q\nabla f(\boldsymbol{\beta}^k)\right)$ ▷ call Algorithm 1

  **if** $f(\boldsymbol{\beta}^+) \leqslant f(\boldsymbol{\beta}^k) + \nabla f(\boldsymbol{\beta}^k)^{\intercal}(\boldsymbol{\beta}^+ - \boldsymbol{\beta}^k) + \frac{1}{2q}\left\|\boldsymbol{\beta}^+ - \boldsymbol{\beta}^k\right\|_2^2$ **then**

    $k = k + 1$; $\boldsymbol{\beta}^{k+1} \leftarrow \boldsymbol{\beta}^+$

    **exit;**

  **else**

    $q \leftarrow \alpha q$

  **end**

**until** $\left\|\boldsymbol{\beta}^k - \boldsymbol{\beta}^{k-1}\right\|_1 < r$;

**return** $\hat{\boldsymbol{\beta}} \leftarrow \boldsymbol{\beta}^{k+1}$

In our implementation, the step size shrinkage rate $\alpha$ is a parameter in the backtracking line-search that controls the rate at which the step size is reduced during each iteration of line-search until an appropriate step size is found. The proper step size should satisfy the line-search criteria, ensuring that the update with this step size leads to a sufficient decrease in the objective function. If $\alpha$ is too large, the step size might not reduce sufficiently during each iteration of the line search, which can cause the line search algorithm to take more iterations to find an appropriate step size. On the other hand, if the shrinkage factor is too small, it may lead to an over-reduction of the step size, resulting in a small update and slow convergence of the algorithm. In our implementation, we have chosen the commonly used default value of $\alpha = 0.5$ as the shrinkage rate. This choice has proven to be effective for all computations in our simulations and real data analysis.

### 4.3 | Cross-validation

We employ cross-validation to select the appropriate value of $\lambda$. The average cross-validated error (CV-E) is utilized for this purpose. Consider performing $L$-fold cross-validation, where we denote $\hat{\boldsymbol{\beta}}^{-l}$ as the estimate obtained from the remaining $L$-1 folds (training set). The error of the $l$th fold (test set) is defined as $2(P-Q)/R$, where $P$ is the log partial likelihood evaluated at $\hat{\boldsymbol{\beta}}^{-l}$ using the entire dataset, $Q$ is the log partial likelihood evaluated at $\hat{\boldsymbol{\beta}}^{-l}$ using the training set, and $R$ is the number of events in the test set.

We opt for using the error defined above instead of the negative log partial likelihood evaluated at $\hat{\boldsymbol{\beta}}^{-l}$ using the test set because it efficiently leverages the risk set, resulting in greater stability when the number of events in each test set is small. The CV-E serves as a metric for parameter tuning. Additionally, to account for the outcome balance among randomly formed test sets, we divide the deviance $2(P-Q)$ by $R$.

### 4.4 | A note on the "One Standard Error Rule"

Different values of $\lambda$ correspond to different models in the regularization framework. The selection of the appropriate $\lambda$ value is achieved through cross-validation.

When the objective is to identify the model with the lowest prediction error, we choose the value of $\lambda$ that yields the lowest CV-E. This approach is known as the *min* rule.[45] However, if the goal is to recover the sparsity pattern, meaning to select the set of variables that closely resembles the true model's variable set, an alternative rule called the "one-standard-error-rule" (*1se* rule) is recommended.[49] The *1se* rule selects the most parsimonious model whose prediction error is within one standard error of the minimum CV-E. By applying the *1se* rule, we prioritize models that are more sparse while still maintaining reasonable prediction accuracy.

If the time-dependent covariates are internal covariates,[50] using the time-dependent Cox model for prediction may not be appropriate. However, when the objective is predictor identification, we recommend applying the *1se* rule.

## 5 | SIMULATION

We conduct several simulation studies. We evaluate our method in both low- and high-dimensional settings and test its performance in terms of the ability to strictly respect complex selection rules, selection consistency, estimation and prediction accuracy. We also compare our methods with the LASSO, the sparse group LASSO, and the latent overlapping group LASSO. In addition, we report the computation time, evaluate the cross-validation stability, and give insights into the influence of effect of group size, the amount of overlap, and sparsity levels on the performance of our method. All simulations employ 10-fold cross-validation to evaluate the performance. The simulations were conducted using `R` version 4.0.5.[51] The `R` code for simulation is available at https://github.com/Guanbo-W/sox_sim.

### 5.1 | Categorical interaction selection under the time-dependent, low-dimensional setting

In the simulation, we generate data with three main terms, two of which are categorical variables, and two interactions between categorical variables. The three independent variables, $A(t)$, $B(t)$, and $C(t)$, are generated with values that randomly change over time in a piece-wise constant fashion. We consider 50 time points at which the values of any variable can potentially change, with each variable being held constant for a random duration between 5 and 10 time points. To simplify the notation, we use shorthand representations such as $A$ to denote $A(t)$ (similarly for $B$ and $C$). The categorical variables $A$ and $C$ are three-level variables represented by two dummy variables, denoted as $A_1$, $A_2$, $C_1$, and $C_2$, respectively. The variable $B$ is continuous.

Hence, we consider the covariates and functions of covariates as follows $\mathbf{X} = \{A_1, A_2, B, A_1B, A_2B, C_1, C_2, C_1B, C_2B\}$. The below steps outline the procedure for generating $A$ and $C$. Step 1, with replacement, sample 10 integers from {1, 2, 3} with equal probability to represent the three categories of the variables; Step 2, for each sampled value, repeats the value $R_i$ times, where $R_i$ is sampled from $\{5, 6, \ldots, 10\}$ with equal probabilities. Then, concatenate these repeated values

together, resulting in a single vector with a length between 50 and 100; Step 3, take the first 50 elements as the values of the categorical variable. $B$ is generated similarly, with the only difference being that in Step 1, we generate a series of numbers from a standard normal distribution. The time-to-event outcome is generated using a permutation algorithm implemented in the `R` function `PermAlgo`.[52] The event times are dependent on the time-dependent potential predictors $\mathbf{X}$ according to the Cox model, where $\boldsymbol{\beta}^{9\times1}$ represents the vector of log hazard ratios of the predictors. The generated data included approximately 50% random censoring, meaning that for about half of the observations, the event time is unknown due to censoring. Among those not censored, the median event time occurred at approximately time 25. Two scenarios were evaluated:

Scenario 1: Only $A_1$ and $A_2$ are predictive of the outcome; their coefficients are set as log(3). This represents a sparse structure.

Scenario 2: There are five true predictors ($A_1, A_2, B, A_1B, A_2B$) that were predictive of the outcome; all of their coefficients are set as log(3). This corresponds to a less sparse structure.

To enforce selection rules strong heredity (selection rule 1) and "the binary indicators representing a categorical variable are selected collectively" (selection rule 2), we defined a grouping structure as $\{\{A_1, A_2, A_1B, A_2B\}, \{B, A_1B, A_2B, C_1B, C_2B\}, \{A_1B, A_2B\}, \{C_1, C_2, C_1B, C_2B\}, \{C_1B, C_2B\}\}$. This grouping structure ensures that the dummy variables representing a categorical variable are selected collectively, and if an interaction term is selected, the corresponding main terms are also selected. Detailed information on how to determine the grouping structure is provided in Web Appendix D.

We compare our method with the $\ell_1$-penalized Cox models with time-dependent covariates (`CoxL`, Cox LASSO) using the `glmnet` package in R. The `glmnet` package provides an implementation of the (time-dependent) Cox model using $\ell_1$ regularization. We use this implementation as a baseline for comparison with our method.

The performance of the two methods was evaluated using several measures, as presented in Table 2. For each simulated dataset, these measures were calculated individually and then averaged to obtain the overall performance statistics. The weight $\omega_g$ for each group in the penalization term is set to one. We use a convergence criterion of $10^{-5}$, which is based on the sum of the absolute differences between the estimates from the two steps. In the process of model selection, we compare both the *min* and *1se* rules for choosing the tuning parameter $\lambda$. The results are given in Table 3.

Our `sox` method always followed both of the rules as designed. However, Cox LASSO with the *min* rule fails to respect the categorical selection rule in 20%-40% of the cases, and it violates the strong heredity rule in 80% of the cases in certain settings, even with increased sample sizes. It is important to note that in scenario 2, Cox LASSO had a higher probability of breaking the rules. When applying the *min* rule, both methods achieved a perfect missing rate. Additionally, it is observed that `sox` converged faster than Cox LASSO. However, when applying the *1se* rule, the missing rate of `sox` converged faster compared to Cox LASSO. This confirms that `sox` had a higher chance of successfully selecting the variables that should be selected, even with a relatively small sample size.

For both methods, the false alarm rate was much worse when using the *min* rule, indicating that the methods select a significant number of noise variables. Therefore, in a low-dimensional setting, if the objective is to recover the sparsity pattern, it is advisable to avoid applying the *min* rule. However, it is worth noting that the false alarm rate of `sox` with

**TABLE 2** Measures of comparison in the the simulation studies.

| # | Measurements |
|---|---|
| (1) | Missing rate (MR): the percentage of variables not selected among the true predictors. |
| (2) | False alarm rate (FAR): the percentage of selected variables among the noise variables. |
| (3) | Rule 1 Satisfaction (R1S): whether the selected model satisfied strong heredity. |
| (4) | Rule 2 Satisfaction (R2S): whether the resulting selected model satisfied selection rule 2, that dummy indicators of the same variable are always selected together. |
| (5) | C index (RCI): the C index of the selected model. |
| (6) | Mean-squared error (MSE): the mean of the squared differences between each coefficient in the data generating mechanism and its estimate, that is, the $\ell_2$-norm of the difference between the coefficient vector and its estimate. |
| (7) | Cross-validated error (CV-E): the cross-validated error defined in Section 4. |

**TABLE 3** Simulation results of Section 5.1.

| Scenario | 1 (A≠0) | | | | 2 (A, B, AB≠0) | | | |
|---|---|---|---|---|---|---|---|---|
| **Method** | sox.1se | sox.min | CoxL.1se | CoxL.min | sox.1se | sox.min | CoxL.1se | CoxL.min |
| | $N = 100$ | | | | | | | |
| MR | 0.82 | 0.02 | 0.85 | 0.26 | 0.08 | 0.00 | 0.41 | 0.16 |
| FAR | 0.02 | 0.65 | 0.03 | 0.40 | 0.04 | 0.53 | 0.06 | 0.49 |
| R1S | 1.00 | 1.00 | 0.88 | 0.70 | 1.00 | 1.00 | 0.84 | 0.72 |
| R2S | 1.00 | 1.00 | 0.75 | 0.38 | 1.00 | 1.00 | 0.39 | 0.35 |
| RCI | 0.53 | 0.67 | 0.54 | 0.64 | 0.91 | 0.92 | 0.91 | 0.92 |
| MSE | 0.24 | 0.06 | 0.27 | 0.14 | 0.27 | 0.10 | 0.41 | 0.30 |
| CV-E | 6.87 | 6.74 | 6.68 | 6.59 | 4.70 | 4.36 | 4.74 | 4.33 |
| | $N = 500$ | | | | | | | |
| MR | 0.12 | 0.00 | 0.32 | 0.00 | 0.00 | 0.00 | 0.36 | 0.04 |
| FAR | 0.02 | 0.70 | 0.01 | 0.54 | 0.00 | 0.15 | 0.05 | 0.59 |
| R1S | 1.00 | 1.00 | 0.90 | 0.57 | 1.00 | 1.00 | 0.94 | 0.78 |
| R2S | 1.00 | 1.00 | 0.79 | 0.22 | 1.00 | 1.00 | 0.45 | 0.62 |
| RCI | 0.60 | 0.63 | 0.58 | 0.63 | 0.91 | 0.91 | 0.91 | 0.91 |
| MSE | 0.14 | 0.02 | 0.22 | 0.03 | 0.14 | 0.04 | 0.31 | 0.07 |
| CV-E | 6.79 | 6.70 | 6.81 | 6.68 | 4.38 | 4.26 | 4.40 | 4.26 |
| | $N = 1000$ | | | | | | | |
| MR | 0.00 | 0.00 | 0.04 | 0.00 | 0.00 | 0.00 | 0.20 | 0.02 |
| FAR | 0.02 | 0.73 | 0.02 | 0.58 | 0.00 | 0.08 | 0.04 | 0.66 |
| R1S | 1.00 | 1.00 | 0.97 | 0.70 | 1.00 | 1.00 | 0.90 | 0.82 |
| R2S | 1.00 | 1.00 | 0.94 | 0.31 | 1.00 | 1.00 | 0.53 | 0.65 |
| RCI | 0.61 | 0.62 | 0.60 | 0.62 | 0.91 | 0.91 | 0.91 | 0.91 |
| MSE | 0.10 | 0.01 | 0.15 | 0.02 | 0.12 | 0.02 | 0.29 | 0.08 |
| CV-E | 6.76 | 6.68 | 6.76 | 6.67 | 4.33 | 4.25 | 4.36 | 4.24 |
| | $N = 2000$ | | | | | | | |
| MR | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.08 | 0.00 |
| FAR | 0.00 | 0.06 | 0.00 | 0.54 | 0.00 | 0.00 | 0.04 | 0.57 |
| R1S | 1.00 | 1.00 | 1.00 | 0.60 | 1.00 | 1.00 | 0.93 | 0.76 |
| R2S | 1.00 | 1.00 | 0.99 | 0.20 | 1.00 | 1.00 | 0.79 | 0.57 |
| RCI | 0.61 | 0.61 | 0.61 | 0.61 | 0.91 | 0.91 | 0.91 | 0.91 |
| MSE | 0.06 | 0.00 | 0.10 | 0.01 | 0.12 | 0.05 | 0.23 | 0.01 |
| CV-E | 6.71 | 6.66 | 6.72 | 6.66 | 4.30 | 4.23 | 4.31 | 4.23 |

Abbreviations: `1se`, applying the 1se rule; Cox LASSO, unstructured $\ell_1$ penalty (`glmnet` with cox); CV-E, cross-validated error; FAR, false alarm rate; `min`, applying the min rule; MR, missing rate; MSE, mean-squared error; R1S, Rule 1 Satisfaction; R2S, Rule 2 Satisfaction; RCI, the C index of the model with the selected variables; `sox`, our method.

the *1se* rule converged to 0 relatively quickly with increasing sample sizes. The MSE of `sox` was significantly smaller and converged faster compared to Cox LASSO under either rule, indicating that `sox` achieved better estimation performance. The cross-validated errors and prediction accuracy of both methods are similar across all the settings.

Overall, the simulation results verify that `sox` can more effectively recover the sparsity pattern and provide better estimation when incorporating the correct selection rules.

## 5.2 | Interaction selection under time-dependent, high-dimensional setting

We follow the design outlined in She et al[53] to perform interaction selection. In this model, our goal is to identify significant two-way interactions from all the potential ones while adhering to the strong heredity selection rule. We generate time-dependent predictors denoted as $X = (X^{\text{main}}, X^{\text{inter}})$. Within $X^{\text{main}} = (X_1, \dots, X_{p'})$, each main term is generated following a standard normal distribution, similar to the low-dimensional case (eg, $B(t)$). We introduce four time-varying points, with each variable held constant for two or three time-points. Subsequently, we create $X^{\text{inter}} = (X_1X_2, \dots, X_1X_{p'}, X_2X_3, \dots, X_{p'-1}X_{p'})$. The true coefficient vector is denoted as

$$\boldsymbol{\beta} = (\beta_1, \dots, \beta_{p'}, \beta_{1,2}, \dots, \beta_{1,p'}, \beta_{2,3}, \dots, \beta_{p'-1,p'})^\top,$$

which has a length of $p' + \binom{p'}{2}$, matching the dimension of the combined predictors in $X$.

We generate time-to-event outcomes using the `R` package `PermAlgo`.[52] For all simulations with different combinations of $n$ and $p$, we set the nine coefficients of the main terms $\beta_j, \forall j = 1, \dots, 9$ as 0.4 and the nine coefficients of interaction terms $\beta_{j,j'} = 0.3$ for $(j, j') = (1,2), (1,3), (1,7), (1,8), (1,9), (4,5), (4,6), (7,8)$ and $(7,9)$. All other coefficients are set to zero.

To enforce the strong heredity selection rule, we define the grouping structure as follows:

$$\mathbb{g}_1 = \{X_1, X_1X_2, \dots, X_1X_{p'}\}, \dots, \mathbb{g}_{p'} = \{X_{p'}, X_1X_{p'}, \dots, X_{p'-1}X_{p'}\}, \quad \mathbb{g}_{p'+1} = \{X_1X_2\}, \dots, \mathbb{g}_{p'+\binom{p'}{2}} = \{X_{p'-1}X_{p'}\}.$$

We consider six different combinations of $(n, p)$ with $n = (400, 800)$ and $p = p' + \binom{p'}{2} = (210, 465, 820)$, where $p' = (20, 30, 40)$ represents the number of main terms. Since `SGL` and `grpCox` do not support the time-dependent model, we only compare the performance of `sox` with the LASSO regularized Cox model (both using the *min* rule). The optimal value of $\lambda$ was selected from a sequence of candidate values through 10-fold cross-validation. We also applied adaptive penalty weights to obtain debiased estimates (`db`). Specifically, the adaptive regularization weights were calculated as the inverse of the original `sox` or the LASSO Cox estimates $\omega_{\mathbb{g}} = 1/\max(|\hat{\boldsymbol{\beta}}_{\mathbb{g}}|, 10^{-16})$.

We assess selection consistency using the same metrics as in Section 5.1. The results of these evaluations are presented in Table 4. In summary, `sox` outperforms Cox LASSO in several aspects. First, `sox` strictly adheres to the strong heredity selection rule, which is not achieved by Cox LASSO. Second, `sox` exhibits lower missing rates than Cox LASSO when $n = 400$ and comparable missing rates when $n = 800$. Debiased `sox` has the lowest false alarm rates in most cases. This can be attributed to the enforcement of the selection rule, where the elimination of an interaction term from the model is triggered not only by the term itself but also by the absence of either of its main terms. Furthermore, `sox` demonstrates higher estimation accuracy, as evidenced by the lower mean-squared errors.

When incorporating the strong heredity selection rule, the algorithm forces the selection of the two main terms when an interaction is selected, which would increase the false alarm rate if the selected interaction term is a noisy variable. Therefore, the false alarm rate of `sox` can be slightly higher than the methods without incorporating the selection rule. However, by employing our method, we can achieve a lower missing rate and more importantly, an interpretable prediction model.

Similar conclusions can be made when the *1se* rule is applied. The results are given in Web Appendix E.

## 5.3 | Comparison with existing sparse group lasso methods

In this section, we compare `sox` with two existing packages, `SGL`[54] and `grpCox`,[55] both of which implement the sparse group lasso for the time-fixed Cox model. Specifically, `grpCox` achieves within-group sparsity by utilizing a latent

**TABLE 4** Simulation results of Section 5.2.

| **Method** | sox | sox.db | CoxL | CoxL.db | sox | sox.db | CoxL | CoxL.db |
|---|---|---|---|---|---|---|---|---|
| $p = 210$ | $n = 400$ | | | | $n = 800$ | | | |
| MR | 0.04 | 0.06 | 0.05 | 0.07 | 0.02 | 0.04 | 0.01 | 0.01 |
| FAR | 0.27 | 0.11 | 0.21 | 0.18 | 0.14 | 0.03 | 0.20 | 0.19 |
| R1S | 1.00 | 1.00 | 0.87 | 0.89 | 1.00 | 1.00 | 0.88 | 0.89 |
| RCI | 0.88 | 0.86 | 0.89 | 0.88 | 0.85 | 0.83 | 0.85 | 0.85 |
| MSE* | 4.97 | 3.91 | 5.97 | 4.86 | 4.37 | 3.88 | 5.21 | 4.11 |
| CV-E | 1.74 | 1.70 | 1.76 | 1.60 | 1.68 | 1.67 | 1.69 | 1.61 |
| $p = 465$ | $n = 400$ | | | | $n = 800$ | | | |
| MR | 0.06 | 0.08 | 0.11 | 0.12 | 0.01 | 0.04 | 0.01 | 0.01 |
| FAR | 0.17 | 0.10 | 0.11 | 0.10 | 0.12 | 0.04 | 0.12 | 0.11 |
| R1S | 1.00 | 1.00 | 0.91 | 0.92 | 1.00 | 1.00 | 0.90 | 0.91 |
| RCI | 0.91 | 0.87 | 0.91 | 0.91 | 0.86 | 0.84 | 0.87 | 0.87 |
| MSE* | 2.45 | 1.77 | 2.95 | 2.50 | 2.63 | 2.55 | 3.14 | 2.58 |
| CV-E | 1.76 | 1.66 | 1.79 | 1.52 | 2.04 | 1.66 | 2.46 | 1.94 |
| $p = 820$ | $n = 400$ | | | | $n = 800$ | | | |
| MR | 0.05 | 0.07 | 0.14 | 0.15 | 0.02 | 0.04 | 0.03 | 0.03 |
| FAR | 0.14 | 0.08 | 0.07 | 0.06 | 0.10 | 0.05 | 0.08 | 0.08 |
| R1S | 1.00 | 1.00 | 0.94 | 0.94 | 1.00 | 1.00 | 0.93 | 0.93 |
| RCI | 0.93 | 0.90 | 0.92 | 0.92 | 0.88 | 0.86 | 0.89 | 0.89 |
| MSE* | 1.45 | 0.98 | 1.78 | 1.57 | 1.19 | 0.92 | 1.47 | 1.21 |
| CV-E | 1.79 | 1.67 | 1.84 | 1.48 | 1.70 | 1.67 | 1.73 | 1.53 |

*Note*: In the tuning process, "lambda.min" is used. Results are averaged over 20 independent replications.
Abbreviations: .db, with additional debiasing procedure; CoxL, unstructured $\ell_1$ penalty (`glmnet` with "cox" family); CV-E, cross-validated error; FAR, false alarm rate; JDR, joint detection rate; MR, missing rate; MSE, mean-squared error (*values are multiplied by $10^{-3}$); R1S, rule 1 satisfaction; RCI, the C index of the model with the selected variables; sox, our method.

group LASSO approach.[13,56] More introduction on the methods implemented in these two packages are given in Web Appendix F.

Following Simon et al,[10] we simulate a covariate matrix $\mathbf{X}$ with dimensions $n = 100$ and $p = 200$. The columns of $\mathbf{X}$ are independently generated from a standard Gaussian distribution. The variables $X_1, \ldots, X_{200}$ are divided into 10 groups, each containing 20 variables.

We consider three different cases of true coefficients

$$\begin{aligned} &\text{Case 1:} \quad \boldsymbol{\beta} = (\boldsymbol{\beta}_s^\top, 0, \ldots, 0)^\top, \\ &\text{Case 2:} \quad \boldsymbol{\beta} = (\boldsymbol{\beta}_s^\top, \boldsymbol{\beta}_s^\top, 0, \ldots, 0)^\top, \\ &\text{Case 3:} \quad \boldsymbol{\beta} = (\boldsymbol{\beta}_s^\top, \boldsymbol{\beta}_s^\top, \boldsymbol{\beta}_s^\top, 0, \ldots, 0)^\top, \end{aligned}$$

where

$$\boldsymbol{\beta}_s = (0.1, 0.2, 0.3, 0.4, 0.5, \underbrace{0, \ldots, 0}_{15})^\top \in \mathbb{R}^{20}.$$

We generate the time-to-event outcome using the `R` package `coxed`,[57] which employs uses duration-based simulation methods. The generated event times depends on fixed predictors according to the proportional hazards model $h(t|\mathbf{X}) = h_0(t)\exp(\boldsymbol{\beta}^\top \mathbf{X})$.

To fit the sparse group LASSO-regularized Cox model in `sox`, we specify the following grouping structures:

$$\begin{aligned} &\mathrm{g}_1 = \{X_1\}, \quad \cdots \quad \mathrm{g}_{20} = \{X_{20}\}, \quad \mathrm{g}_{201} = \{X_1, \ldots, X_{20}\}, \\ &\mathrm{g}_{21} = \{X_{21}\}, \quad \cdots \quad \mathrm{g}_{40} = \{X_{40}\}, \quad \mathrm{g}_{202} = \{X_{21}, \ldots, X_{40}\}, \\ &\qquad \vdots \qquad\qquad\qquad \vdots \qquad\qquad\qquad \vdots \\ &\mathrm{g}_{181} = \{X_{181}\}, \quad \cdots \quad \mathrm{g}_{200} = \{X_{200}\}, \quad \mathrm{g}_{210} = \{X_{181}, \ldots, X_{200}\}, \end{aligned}$$

where each group $\mathrm{g}_1, \ldots, \mathrm{g}_{200}$ contains only a single covariate with the corresponding index, and each group $\mathrm{g}_{201}, \ldots, \mathrm{g}_{210}$ contains 20 covariates. The groups adhere to a nested structure, for example, $\mathrm{g}_1, \ldots, \mathrm{g}_{200}$ are nested within group $\mathrm{g}_{201}$ and $\mathrm{g}_{21}, \ldots, \mathrm{g}_{400}$ within $\mathrm{g}_{202}$, and so on. The grouping structure is built for `grpCox` and `SGL`. For `SGL`, the mixing parameter (of the $\ell_1$ and $\ell_2$ component) in `SGL` is set to 0.5. For a fair comparison, we ensure that the amount of regularization applied by `sox` is effectively the same as in `SGL`. Specifically, for groups $\mathrm{g}_1, \ldots, \mathrm{g}_{200}$, we set the regularization weight to 0.5, and for groups $\mathrm{g}_{201}, \ldots, \mathrm{g}_{210}$, the regularization weight is set to $\sqrt{20} \times 0.5$, where 20 is the group size. In contrast, `grpCox` does not support custom regularization weights, so we use the package defaults.

We perform 10-fold cross-validation to select the optimal regularization coefficient $\lambda$ and evaluate the performance of the three methods. To ensure a fair comparison, we ensure that all methods use the same $\lambda$ sequence and the same train-validation split. In the case of `grpCox`, we employ the default settings for model tuning.

We summarize the performance statistics in Table 5. In summary, `sox` and `SGL` perform similarly in fitting SGL-regularized time-fixed Cox models. In fact, the two sets of estimates are very close, as shown in the sample solution path in Web Appendix G. On the other hand, `grpCox` exhibits a considerably more conservative approach to variable selection, as evidenced by MR and FAR. Additionally, the mean squared errors (MSE) of `grpCox` estimates are also higher.

Furthermore, we perform additional simulations that compare `sox` and `SGL` under some simulation settings where `SGL` failed to adhere to the selection rule. The details are provided in Web Appendix H.

## 5.4 | Additional simulations

Here, we present the additional simulations.

### 5.4.1 | Timing

We test the computational speed of `sox`. We adopt the simulation setting from Section 5.2. We report the computation time for solving 10-fold cross-validation on the same $\lambda$ sequence of length 30. We find that for a complex case where the sample size is $n = 800$ and the number of variables is $p = 820$, a comprehensive analysis can be completed in approximately 10 minutes. In contrast, for a simpler scenario with a sample size of $n = 400$ and $p = 210$ variables, the analysis requires less than 30 s to finish. More details are given in Web Appendix I.

### 5.4.2 | Stability of cross-validation

To evaluate the stability of the CV of `sox`, we conduct additional simulations using the simulation setting from Section 5.2. We simulate a single set of data ($n = 400, p = 210$) and performed 10-fold CV twenty times. To demonstrate the stability

**TABLE 5** Simulation results of Section 5.3.

| | sox | SGL | grpCox | sox | SGL | grpCox | sox | SGL | grpCox |
|---|---|---|---|---|---|---|---|---|---|
| | **Case 1** | | | **Case 2** | | | **Case 3** | | |
| MR | 0.27 | 0.31 | 0.71 | 0.34 | 0.30 | 0.79 | 0.36 | 0.46 | 0.82 |
| FAR | 0.13 | 0.12 | 0.02 | 0.17 | 0.18 | 0.02 | 0.20 | 0.16 | 0.02 |
| MSE* | 2.04 | 2.02 | 2.21 | 4.38 | 4.19 | 4.77 | 7.28 | 7.25 | 7.49 |

*Note*: MSE: $\|\boldsymbol{\beta} - \hat{\boldsymbol{\beta}}\|_2^2/p$, values are multiplied by $10^{-3}$. In the tuning process, "lambda.min" is used. Results are averaged over 20 independent replications. Abbreviations: `grpCox`, the latent overlapping group LASSO; `SGL`, the sparse group LASSO; `sox`, our method.

of the CV of sox, we report the average CV error and MSE (resulting from the final model chosen by each repeated CV procedure), and their corresponding standard errors. We compare these results to those generated by glmnet. The mean CV error (with standard error) was 1.82 (0.03) for sox and 1.87 (0.03) for glmnet. The mean MSE $\times 10^{-3}$ (with standard error) was 8 (0.4) for sox and 9 (0.4) for glmnet. The results show that the CV procedure in sox is stable.

### 5.4.3 | Comparison of different group sizes, the amount of overlap, and the sparsity levels

We conduct an additional simulation to compare the effect of different group sizes, the amount of overlap, and the sparsity levels on the performance of our method. We demonstrate that depending on the true data-generating mechanism, these factors may influence the performance of sox. The details are given in Web Appendix J.

## 6 | CASE STUDY: TIME-DEPENDED PREDICTOR IDENTIFICATION FOR TIME TO DEATH BY ANY CAUSE AMONG PATIENTS HOSPITALIZED FOR ATRIAL FIBRILLATION

### 6.1 | Background

Atrial fibrillation (AF) is a medical condition characterized by an irregular heartbeat. Patients with AF are at a higher risk of experiencing cardiovascular complications and mortality.[58] Therefore, it is important to identify predictors that contribute to these outcomes and develop a predictive model that can help understand the disease and support clinical decision-making.

In treating AF, most patients are prescribed oral anticoagulants (OAC) for long-term management. OAC includes medications such as warfarin and direct OAC (DOAC), the latter including Dabigatran, Apixaban, and Rivaroxaban, each of which can be taken as high or low dose. However, the use of OAC in AF patients is often intermittent due to contraindications, adverse effects, and the need for surgeries.[59] Furthermore, the use of other medications and changes in disease conditions may vary over time among AF patients. Taking into account such time-varying information can be beneficial in identifying predictors of clinical outcomes in AF patients.[60]

### 6.2 | Data

In this study, we utilize the sox method to identify significant baseline and time-varying predictors associated with the time to all-cause death among patients hospitalized for AF who initiated OAC between 2010 and 2017 in the province of Quebec, Canada. The data used in this study were obtained from a larger dataset[61] and represent 36 381 patients who were followed up for a period of 365 days. To ensure the validity of the analysis, we applied specific inclusion and exclusion criteria to the dataset, which are detailed in Web Appendix K. Among the included patients, a total of 4384 individuals experienced the event of interest (ie, death by any cause) during the follow-up period, accounting for approximately 12.05% of the population.

In this study, we selected a total of 24 candidate predictors based on previous research findings[62-65] and data availability. Out of these predictors, 7 are baseline (time-invariant) covariates. The baseline covariates include age, sex, medical scores, comorbidities (eight conditions), OAC use information, concomitant medication use (four drugs), and the interactions between each concomitant medication and DOAC. Additionally, we define five time-dependent indicator covariates to capture OAC use: DOAC, Apixaban, Dabigatran, OAC, and High-dose-DOAC.

The definitions and summary statistics for the covariates are provided in Web Appendices L and M, respectively. It is worth noting that the time-dependent covariates, especially those related to OAC use, exhibited significant changes over time. This highlights the importance of considering the time-varying nature of these covariates, as neglecting their changes may introduce substantial bias in covariates selection and coefficient estimation.

### 6.3 | Analysis

To assess the associations between the covariates and the time to the event of interest, we conduct both univariate and multivariate time-dependent Cox models. The crude hazard ratios from the univariate models and the adjusted

**TABLE 6** Selection rules for the case study.

| # | Selection rule |
|---|---|
| 1 | If Apixaban is selected, then select DOAC |
| 2 | If Dabigatran is selected, then select DOAC |
| 3 | If DOAC is selected, then select OAC |
| 4 | If High-dose-DOAC is selected, then select DOAC |
| 5 … 8 | If the interaction of DOAC and a concomitant medication is selected, then both DOAC and the concomitant medication are selected |

hazard ratios from the multivariate models, along with their corresponding 95% confidence intervals, are presented in Web Appendix N.

For this data analysis, we establish selection rules (including strong heredity and the rules for coefficients interpretability) to ensure the interpretability of the model. The specific selection rules are outlined in Table 6. It aim to capture the following associations: (1) the use of OAC vs non-use; (2) the use of DOAC compared to warfarin; (3) the use of Apixaban compared to Rivaroxaban; (4) the use of Dabigatran compared to Rivaroxaban; (5) the use of high-dose-DOAC compared to low-dose-DOAC; (6) the simultaneous use of concomitant medication and DOAC.

The selection rules ensure that the aforementioned comparisons are estimable and correspond to the coefficients of the selected variables if selected. Further explanation and rationale for these selection rules can be found in Web Appendix O. To respect these selection rules simultaneously, we identified the appropriate graph-structured grouping structure, which is provided in Web Appendix P, following the approach described in References 23,66.

### 6.4 | Results

First, we applied both the `sox` method and Cox LASSO to select variables from the data. We then used the unpenalized time-dependent Cox model to estimate the hazard ratios for the selected covariates. Additionally, we report the estimates with 95% confidence intervals when all the covariates are included in the model. The results are given in Table 7. Our proposed method, `sox`, identified 15 predictors that are associated with the outcome. Comparing Cox LASSO with `sox`, we observe the following differences: (1) DOAC was not selected by Cox LASSO, which affects the interpretation of the coefficient of Apixaban; (2) the interaction of DOAC and Statin was selected without Statin in Cox LASSO, violating the strong heredity assumption; (3) predictors such as high-dose-DOAC and Beta-Blockers were not selected by Cox LASSO. Additionally, our method achieves a slightly higher concordance index compared to the unstructured penalization, though slightly lower than the full model. These results demonstrate the advantages of utilizing `sox` over Cox LASSO, highlighting the benefits of incorporating prior knowledge about the underlying structure of the data. To further illustrate the results, we visualize the impact of time-dependent covariates on the survival probability. See more details in Web Appendix Q.

## 7 | DISCUSSION

In both low and high-dimension settings, incorporating a priori knowledge of covariate structures can achieve robust and interpretable models. In survival models, especially when the covariates and coefficients are time-dependent, no clear guidance and methods are available for the incorporation of such prior information. In this article, we introduced `sox`, a novel structured sparsity-inducing penalty for the time-dependent Cox model. The method can accommodate a wide range of a priori knowledge about the data structure in the form of restrictions on covariate inclusion. We empirically showed that incorporating correct selection rules can improve model selection performance and accuracy of estimation. The developed algorithm converges fast and is able to handle high-dimensional data, which can be implemented in the developed `R` package. Through examples, simulations, and the case study, we also explored how to set appropriate selection rules and the corresponding grouping structures for the developed method in practice, which provided users with guidance on the implementation of the methods in their own application.

**TABLE 7** The estimated hazard ratios and 95% confidence intervals of each covariate from various methods.

| Covariate | sox.refit | CoxL.refit | Cox |
|---|---|---|---|
| C-Index | 0.8034 | 0.7991 | 0.8077 |
| Age ($\geqslant$ 75) | 1.84 | 1.80 | 1.85 (1.70, 2.03) |
| Sex (female/male) | - | - | 0.96 (0.90, 1.03) |
| *Comorbidities/medical score* | | | |
| $CHA_2DS_2VAS_c$ ($\geqslant$ 3) | 0.84 | - | 0.90 (0.80, 1.02) |
| Diabetes | - | - | 1.08 (1.01, 1.15) |
| COPD/asthma | 1.51 | 1.49 | 1.49 (1.41, 1.59) |
| Hypertension | - | - | 0.89 (0.81, 0.97) |
| Malignant cancer | 1.56 | 1.58 | 1.55 (1.46, 1.65) |
| Stroke | - | - | 0.95 (0.88, 1.02) |
| Chronic kidney disease | 2.40 | 2.34 | 2.39 (2.23, 2.55) |
| Heart disease | 2.55 | 2.36 | 2.56 (2.31, 2.83) |
| Major bleeding | 1.71 | 1.69 | 1.72 (1.61, 1.83) |
| *OAC use* | | | |
| DOAC | 0.89 | - | 1.04 (0.76, 1.44) |
| Apixaban | 0.94 | 0.94 | 0.86 (0.67, 1.12) |
| Dabigatran | - | - | 0.80 (0.55, 1.17) |
| OAC | 0.17 | 0.16 | 0.17 (0.15, 0.19) |
| High-dose-DOAC | 0.91 | - | 0.87 (0.69, 1.11) |
| *Concomitant medication use* | | | |
| Antiplatelets | - | - | 1.10 (1.03, 1.19) |
| NSAIDs | - | - | 1.58 (1.34, 1.86) |
| Statin | 0.65 | - | 0.63 (0.59, 0.67) |
| Beta-Blockers | 1.04 | - | 1.03 (0.97, 1.10) |
| *Potential drug-drug interaction* | | | |
| DOAC: Antiplatelets | - | - | 0.67 (0.50, 0.90) |
| DOAC: NSAIDs | - | - | 0.76 (0.43, 1.37) |
| DOAC: Statin | 1.05 | 0.66 | 1.15 (0.90, 1.48) |
| DOAC: Beta-Blockers | 0.83 | - | 0.86 (0.68, 1.10) |

Abbreviations: C-Index, concordance index; `Cox`, the standard time dependent-Cox model with all covariates included; `CoxL.refit`, unstructured $\ell_1$ penalty; `sox.refit`, our method.

We would also like to emphasize that our work primarily focuses on scenarios where practitioners have prior knowledge of the definition of each variable and its relationship with others, aiming to produce an interpretable prediction model. For instance, variables may be defined as interactions of other variables to ensure the interpretability of the resulting coefficients, necessitating the use of strong heredity (as demonstrated in various examples in Web Appendix A and the case study).

In practical situations where the definition of a variable or its relationship with other variables is unclear, we recommend not including the variable in a selection rule or performing sensitivity analysis. In such cases, it is prudent to use more conservative methods like the LASSO. Incorporating uncertain or incorrect selection rules can potentially degrade the performance of the resulting model.

There are several avenues for future work. For instance, one could relax the assumption that the hazard is dependent on the current value of the covariates, assume event times follow a parametric distribution by using accelerated failure time models,[67] generalize to different penalty types (for example, minimax concave penalty,[68] or smoothly clipped absolute deviation[69]), and investigate the impact of applying different weighting schemes. In addition, it would be beneficial to integrate structured variable selection into causal inference, especially when the confounders or effect modifiers are high-dimensional.[70-79]

## AFFILIATIONS

[1]Department of Epidemiology, Harvard T.H. Chan School of Public Health, Boston, Massachusetts, USA

[2]Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, Philadelphia, Pennsylvania, USA

[3]Department of Mathematics and Statistics, McGill University, Montreal, Quebec, Canada

[4]Mila Québec AI Institute, Montreal, Quebec, Canada

[5]Department of Epidemiology, Biostatistics and Occupational Health, McGill University, Montreal, Quebec, Canada

[6]Department of Population Medicine, Harvard Pilgrim Health Care Institute and Harvard Medical School, Boston, Massachusetts, USA

[7]Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, Massachusetts, USA

[8]Faculté de Pharmacie, Université de Montréal, Montreal, Quebec, Canada

[9]StatSciences Inc., Notre-Dame-de-l'Île-Perrot, Quebec, Canada

[10]Département de Médecine Sociale et Préventive, Université de Montréal, Montreal, Quebec, Canada

## ACKNOWLEDGEMENTS

We would like to thank the Régie de l'Assurance Maladie du Québec (RAMQ) and Ministry of Health and Social Services (MSSS) (Quebec, Canada) for providing assistance in handling the data and the Commission d'accès à l'information for authorizing the study. G. Wang is supported by the Fonds de Recherche du Québec Santé (FRQS-272161). A. Y. Yang is supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant RGPIN-2016-05174 and Fonds de Recherche du Québec Nature et Technologies Team Grant (FRQNT-327788). M. E. Schnitzer is supported by a Tier 2 Canada Research Chair in Causal Inference and Machine Learning in Health Science and an NSERC Discovery Grant. R. W. Platt is supported by CIHR Foundation Grant (FDN-143297). The study was supported by the Heart and Stroke Foundation of Canada (G-17-0018326) and the Réseau Québécois de Recherche sur les Médicaments (RQRM). Please refer to the following https://www.heartandstroke.ca/ and http://www.frqs.gouv.qc.ca/en/.

## CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest.

## DATA AVAILABILITY STATEMENT

The `R` package developed for the proposed method is available at https://cran.r-project.org/package=sox. The `R` code for simulation is available at https://github.com/Guanbo-W/sox_sim. The data that support the findings of this study are available from RAMQ, MSSS and the Commission d'accès à l'information. Restrictions apply to the availability of these data, which were used under license for this study. Data are available from the author(s) with the permission of RAMQ, MSSS and the Commission d'accès à l'information.

## ORCID

*Guanbo Wang* https://orcid.org/0000-0002-5446-4543
*Robert W. Platt* https://orcid.org/0000-0002-5981-8443
*Rui Wang* https://orcid.org/0000-0001-5007-193X
*Mireille E. Schnitzer* https://orcid.org/0000-0001-8049-9646

## REFERENCES

1. Reid N, Cox D. *Analysis of Survival Data*. Boca Raton, FL: Chapman and Hall/CRC; 2018.
2. Holcomb JB, del Junco DJ, Fox EE, et al. The prospective, observational, multicenter, major trauma transfusion (PROMMTT) study: comparative effectiveness of a time-varying treatment with competing risks. *JAMA Surg*. 2013;148(2):127-136.

3. Weisz G, Généreux P, Iñiguez A, et al. Ranolazine in patients with incomplete revascularisation after percutaneous coronary intervention (RIVER-PCI): a multicentre, randomised, double-blind, placebo-controlled trial. *Lancet*. 2016;387(10014):136-145.
4. Huang J, Sun T, Ying Z, Yu Y, Zhang CH. Oracle inequalities for the lasso in the Cox model. *Ann Stat*. 2013;41(3):1142.
5. Fan J, Li R. Variable selection for Cox's proportional hazards model and frailty model. *Ann Stat*. 2002;30:74-99.
6. Beretta A, Heuchenne C. Variable selection in proportional hazards cure model with time-varying covariates, application to US bank failures. *J Appl Stat*. 2019;46(9):1529-1549.
7. Yan J, Huang J. Model selection for Cox models with time-varying coefficients. *Biometrics*. 2012;68(2):419-428.
8. Lim M, Hastie T. Learning interactions via hierarchical group-lasso regularization. *J Comput Graph Stat*. 2015;24(3):627-654.
9. Wang L, Shen J, Thall PF. A modified adaptive lasso for identifying interactions in the Cox model with the heredity constraint. *Stat Probab Lett*. 2014;93:126-133.
10. Simon N, Friedman J, Hastie T, Tibshirani R. A sparse-group lasso. *J Comput Graph Stat*. 2013;22(2):231-245.
11. Wang S, Nan B, Zhu N, Zhu J. Hierarchically penalized Cox regression with grouped variables. *Biometrika*. 2009;96(2):307-322.
12. Dang X, Huang S, Qian X. Penalized Cox's proportional hazards model for high-dimensional survival data with grouped predictors. *Stat Comput*. 2021;31:1-27.
13. Obozinski G, Jacob L, Vert JP. Group lasso with overlaps: the latent group lasso approach. arXiv preprint arXiv:1110.0413, 2011.
14. Villa S, Rosasco L, Mosci S, Verri A. Proximal methods for the latent group lasso penalty. *Comput Optim Appl*. 2014;58:381-407.
15. Jenatton R, Audibert JY, Bach F. Structured variable selection with sparsity-inducing norms. *J Mach Learn Res*. 2011;12:2777-2824.
16. Lian Y, Wang G, Yang AY, et al. sox: structured learning in time-dependent Cox models. R package version 1.0. 2023.
17. Breslow N. Covariance analysis of censored survival data. *Biometrics*. 1974;30:89-99.
18. Kim J, Sohn I, Jung SH, Kim S, Park C. Analysis of survival data with group lasso. *Commun Stat Simul Comput*. 2012;41(9):1593-1605.
19. Vogt JE, Roth V. The group-lasso: $\ell 1,\infty$ regularization versus $\ell 1$, 2 regularization. *Joint Pattern Recognition Symposium*. Berlin: Springer; 2010:252-261.
20. Vogt JE, Roth V. A complete analysis of the l1,p group-lasso. *Proceedings of the 29th International Coference on International Conference on Machine Learning ICML'12*. Madison, WI: Omnipress; 2012:1091-1098.
21. Jenatton R, Mairal J, Obozinski G, Bach F. Proximal methods for hierarchical sparse coding. *J Mach Learn Res*. 2011;12:2297-2334.
22. Yan X, Bien J. Hierarchical sparse modeling: a choice of two group lasso formulations. *Stat Sci*. 2017;32(4):531-560.
23. Wang G, Schnitzer M, Chen T, Wang R, Platt RW. A general framework for formulating structured variable selection. Transactions on Machine Learning Research; 2024.
24. Mairal J, Jenatton R, Obozinski G, Bach F. Convex and network flow optimization for structured sparsity. *J Mach Learn Res*. 2011;12(9):2681-2720.
25. Haris A, Witten D, Simon N. Convex modeling of interactions with strong heredity. *J Comput Graph Stat*. 2016;25(4):981-1004. doi:10.1080/10618600.2015.1067217
26. Reisberg B, Ferris SH, de Leon MJ, Crook T. The global deterioration scale for assessment of primary degenerative dementia. *Am J Psychiatry*. 1982;139:1136-1139.
27. Zhai T, Gu H, Yang Y. Cox regression based modeling of functional connectivity and treatment outcome for relapse prediction and disease subtyping in substance use disorder. *Front Neurosci*. 2021;15:768602.
28. Lee YB, Lee J, Tak S, et al. Sparse SPM: group sparse-dictionary learning in SPM framework for resting-state functional connectivity MRI analysis. *Neuroimage*. 2016;125:1032-1045.
29. Chklovskii DB, Koulakov AA. Maps in the brain: what can we learn from them? *Annu Rev Neurosci*. 2004;27:369-392.
30. Ward JH Jr. Hierarchical grouping to optimize an objective function. *J Am Stat Assoc*. 1963;58(301):236-244.
31. Jenatton R, Gramfort A, Michel V, et al. Multiscale mining of fMRI data with hierarchical structured sparsity. *SIAM J Imaging Sci*. 2012;5(3):835-856.
32. Boyd S, Vandenberghe L. *Convex Optimization*. Cambridge, UK: Cambridge University Press; 2004.
33. Moreau JJ. Fonctions convexes duales et points proximaux dans un espace hilbertien. *C R Hebd Seances Acad Sci*. 1962;255:2897-2899.
34. Combettes PL, Pesquet JC. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Berlin: Springer; 2011:185-212.
35. Bach F. Structured sparsity-inducing norms through submodular functions. *Proceedings of the 23rd International Conference on Neural Information Processing Systems*. Red Hook, NY: Curran Associates Inc.; 2010:118-126.
36. Beck A, Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J Imaging Sci*. 2009;2(1):183-202.
37. Nesterov Y. Gradient methods for minimizing composite objective function. CORE Discussion paper #2007#76. 2007.
38. Beck A. *First-Order Methods in Optimization*. Vol 25. Philadelphia, PA: SIAM; 2017.
39. Jenatton R, Mairal J, Obozinski G, Bach FR. Proximal methods for sparse hierarchical dictionary learning. *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Madison, WI: Omnipress; 2010:487-494.
40. Bertsekas D. *Network Optimization: Continuous and Discrete Models*. Vol 8. Nashua, NH: Athena Scientific; 1998.
41. Ford LR, Fulkerson DR. Maximal flow through a network. *Can J Math*. 1956;8:399-404.
42. Babenko M, Goldberg AV. Experimental evaluation of a parametric flow algorithm. Technical report MSR-TR-2006-77. 2006.
43. Mairal J, Bach F, Ponce J. Sparse modeling for image and vision processing. arXiv preprint arXiv:1411.3230, 2014.

44. Therneau TM. A package for survival analysis in R. R package version 3.2-10. 2021.
45. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. *J Stat Softw*. 2010;33(1):1.
46. Simon N, Friedman J, Hastie T, Tibshirani R. Regularization paths for Cox's proportional hazards model via coordinate descent. *J Stat Softw*. 2011;39(5):1.
47. Goldberg AV, Tarjan RE. A new approach to the maximum-flow problem. *J ACM*. 1988;35(4):921-940.
48. Bertsekas DP. Nonlinear programming. *J Oper Res Soc*. 1997;48(3):334.
49. Chen Y, Yang Y. The one standard error rule for model selection: does it work? *Stats*. 2021;4(4):868-892.
50. Zhang Z, Reinikainen J, Adeleke KA, Pieterse ME, Groothuis-Oudshoorn CGM. Time-varying covariates and coefficients in Cox regression models. *Ann Transl Med*. 2018;6(7):121.
51. R Core Team. R: a language and environment for statistical computing; 2021.
52. Sylvestre MP, Evans T, MacKenzie T, Abrahamowicz M. PermAlgo: permutational algorithm to generate event times conditional on a covariate matrix including time-dependent covariates. R package version 1.1. 2010.
53. She Y, Wang Z, Jiang H. Group regularized estimation under structural hierarchy. *J Am Stat Assoc*. 2018;113(521):445-454.
54. Simon N, Friedman J, Hastie T, Tibshirani R. SGL: fit a GLM (or Cox model) with a combination of lasso and group lasso regularization. R package version 1.3. 2019.
55. Dang X. grpCox: penalized Cox model for high-dimensional data with grouped predictors. R package version 1.0.1. 2020.
56. Jacob L, Obozinski G, Vert JP. Group lasso with overlap and graph lasso. *Proceedings of the 26th Annual International Conference on Machine Learning*. New York: ACM; 2009:433-440.
57. Kropko J, Harden J. Coxed: duration-based quantities of interest for the Cox proportional hazards model. R package version 0.3.3. 2020.
58. Lip GY, Tse HF. Management of atrial fibrillation. *Lancet*. 2007;370(9587):604-618.
59. Angiolillo DJ, Bhatt DL, Cannon CP, et al. Antithrombotic therapy in patients with atrial fibrillation treated with oral anticoagulation undergoing percutaneous coronary intervention: a north American perspective: 2021 update. *Circulation*. 2021;143(6):583-596.
60. Claxton JS, MacLehose RF, Lutsey PL, et al. A new model to predict major bleeding in patients with atrial fibrillation using warfarin or direct oral anticoagulants. *PLoS One*. 2018;13(9):e0203599.
61. Perreault S, de Denus S, White-Guay B, et al. Oral anticoagulant prescription trends, profile use, and determinants of adherence in patients with atrial fibrillation. *Pharmacotherapy*. 2020;40(1):40-54.
62. Fauchier L, Samson A, Chaize G, et al. Cause of death in patients with atrial fibrillation admitted to French hospitals in 2012: a nationwide database study. *Open Heart*. 2015;2(1):e000290.
63. Fauchier L, Villejoubert O, Clementy N, et al. Causes of death and influencing factors in patients with atrial fibrillation. *Am J Med*. 2016;129(12):1278-1287.
64. Lee E, Choi EK, Han KD, et al. Mortality and causes of death in patients with atrial fibrillation: a nationwide population-based study. *PLoS One*. 2018;13(12):e0209687.
65. Harb SC, Wang TKM, Nemer D, et al. CHA2DS2-VASc score stratifies mortality risk in patients with and without atrial fibrillation. *Open Heart*. 2021;8(2):e001794.
66. Wang G, Perreault S, Platt RW, Wang R, Dorais M, Schnitzer ME. Integrating complex selection rules into the latent overlapping group lasso for constructing coherent prediction models. arXiv preprint arXiv:2206.05337, 2023.
67. Kalbfleisch JD, Prentice RL. *The Statistical Analysis of Failure Time Data*. Vol 360. Hoboken, NJ: John Wiley & Sons; 2011.
68. Zhang CH. Nearly unbiased variable selection under minimax concave penalty. *Ann Stat*. 2010;38(2):894-942.
69. Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J Am Stat Assoc*. 2001;96(456):1348-1360.
70. Siddique AA, Schnitzer ME, Bahamyirou A, et al. Causal inference with multiple concurrent medications: a comparison of methods and an application in multidrug-resistant tuberculosis. *Stat Methods Med Res*. 2019;28(12):3534-3549.
71. Wang G, Schnitzer ME, Menzies D, Viiklepp P, Holtz TH, Benedetti A. Estimating treatment importance in multidrug-resistant tuberculosis using targeted learning: an observational individual patient data network meta-analysis. *Biometrics*. 2020;76(3):1007-1016.
72. Liu Y, Schnitzer ME, Wang G, et al. Modeling treatment effect modification in multidrug-resistant tuberculosis in an individual patient data meta-analysis. *Stat Methods Med Res*. 2022;31(4):689-705.
73. Wang G, Poulin-Costello M, Pang H, et al. Evaluating hybrid controls methodology in early-phase oncology trials: a simulation study based on the MORPHEUS-UC trial. *Pharm Stat*. 2023;23(1):31-45. doi:10.1002/pst.2336
74. Wang G. Review 1: "antibiotic prescribing in remote versus face-to-face consultations for acute respiratory infections in English primary care: an observational study using TMLE". *Rapid Reviews Infectious Diseases*. Cambridge, MA: The MIT Press; 2023.
75. Bouchard A, Bourdeau F, Roger J, et al. Predictive factors of detectable viral load in HIV-infected patients. *AIDS Res Hum Retroviruses*. 2022;38(7):552-560.
76. Jaman A, Wang G, Ertefaie A, et al. Penalized G-estimation for effect modifier selection in the structural nested mean models for repeated outcomes. arXiv preprint arXiv:2402.00154, 2024.
77. Wang G, Levis A, Steingrimsson J, Dahabreh I. Efficient estimation of subgroup treatment effects using multi-source data. arXiv preprint 2402.02684, 2024.
78. Wang G, Levis A, Steingrimsson J, Dahabreh I. Causal inference under transportability assumptions for conditional relative effect measures. arXiv preprint 2402.02702, 2024.

79. Wang G, Heagerty PJ, Dahabreh IJ. Using effect scores to characterize heterogeneity of treatment effects. *JAMA*. 2024; 331(14):1225-1226. doi:10.1001/jama.2024.3376

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.

**How to cite this article:** Wang G, Lian Y, Yang AY, et al. Structured learning in time-dependent Cox models. *Statistics in Medicine*. 2024;43(17):3164-3183. doi: 10.1002/sim.10116

# Supporting Information for

## *"Structured Learning in Time-dependent Cox Models"*

## by

**Guanbo Wang[†*1] Yi Lian[†2], Archer Y. Yang[*3,4], Robert W. Platt[5],**

**Rui Wang[6,7], Sylvie Perreault[8], Marc Dorais[9], and Mireille E. Schnitzer[8,10]**

[1]Department of Epidemiology, Harvard T.H. Chan School of Public Health, MA, U.S.A.

[2]Department of Biostatistics, Epidemiology and Informatics, University of Pennsylvania, PA, U.S.A.

[3]Department of Mathematics and Statistics, McGill University, QC, Canada

[4]Mila - Québec AI Institute, QC, Canada

[5]Department of Epidemiology, Biostatistics and Occupational Health, McGill University, QC, Canada

[6]Department of Population Medicine, Harvard Pilgrim Health Care Institute and Harvard Medical School, MA, USA

[7]Department of Biostatistics, Harvard T. H. Chan School of Public Health, MA, USA

[8]Faculté de pharmacie, Université de Montréal, QC, Canada

[9]StatSciences Inc., Notre-Dame-de-l'Île-Perrot, QC, Canada

[10]Département de médecine sociale et préventive, Université de Montréal, QC, Canada

## Web Appendix A More about quadratic min-cost flow problem, network flow algorithm, and the mini-cut theorem.

The quadratic minimum-cost flow problem is a fundamental optimization challenge that arises in various fields, including transportation, network design, and resource allocation (Cohen et al., 2014). In this problem, the goal is to determine the most cost-effective way to send flow through a network, subject to capacity constraints, while minimizing the overall cost (Skutella, 2013). Unlike the linear minimum-cost flow problem, which assumes linear cost functions, the quadratic variant incorporates quadratic cost functions, making it more expressive and capturing nonlinear relationships between flow and cost (Magdon-Ismail and Atiya, 2016). The quadratic terms often represent additional costs associated with flow, such as congestion or utilization-dependent expenses (Dadush et al., 2019). Solving the quadratic minimum-cost flow problem involves finding the flow rates that minimize the total cost, taking into account both linear and quadratic cost components, thereby optimizing resource utilization and minimizing operational expenses (Gabow et al., 2020).

Network flow algorithms (Ford and Fulkerson, 1956; Ahuja et al., 1993; Goldberg and Tarjan, 1988; Gallo et al., 1993) offer powerful solutions for addressing the quadratic minimum-cost flow problem by leveraging their ability to efficiently handle flow optimization in networks with nonlinear cost functions. These algorithms, such as the push-relabel algorithm and its variants, have been extended to accommodate quadratic cost functions, allowing for the optimization of resource allocation while minimizing operational expenses (Gabow et al., 2020). By incorporating quadratic terms into the cost functions, these algorithms can capture complex relationships between flow rates and associated costs, such as congestion or utilization-dependent expenses (Dadush et al., 2019). The adaptation of network flow algorithms to handle quadratic cost functions enables the determination of the most cost-effective flow distribution through a network, subject to capacity constraints, thereby

optimizing resource utilization (Magdon-Ismail and Atiya, 2016). Moreover, these algorithms provide efficient solutions to the quadratic minimum-cost flow problem, even in large-scale networks, making them valuable tools in various applications, including transportation, telecommunications, and supply chain management (Cohen et al., 2014).

Network flow algorithms utilize the mini-cut theorem as a foundational concept to efficiently compute flows in networks. The mini-cut theorem states that in any directed graph with a source and a sink, there exists a minimum cut separating the source from the sink, where the capacity of the cut equals the maximum flow from the source to the sink (Ford and Fulkerson, 1956). This theorem is pivotal in algorithms such as the Ford-Fulkerson algorithm and its variants, which iteratively augment flow along augmenting paths until no more paths can be found, effectively determining the maximum flow in the network (Goldberg and Tarjan, 1988). By leveraging the mini-cut theorem, these algorithms identify critical edges whose removal would disrupt the flow from the source to the sink, guiding their search for optimal flow solutions. Furthermore, the theorem provides insights into the relationship between flow capacities and network connectivity, enabling the development of efficient algorithms for flow optimization (Ahuja et al., 1993).

## Web Appendix B Steps of `computeFlow`

Table S1 shows the steps of `computeFlow`.

## Web Appendix C More implementation details on cross-validation and one-standard-error-rule

The algorithm has a worst-case complexity of $O(|V|^2|E|^{1/2})$ (Cherkassky and Goldberg, 1997), and is well-suited for efficient distributed and parallel implementations. Consider a canonical graph where each node $v \in V$ can be a source $s_1$, and sink $s_2$, a single variable ($X_j \in \mathbb{V}$), or a set of variables ($\mathbb{g}_k \in \mathbb{G}$), that is, $V = \{s_1, s_2\} \cup \mathbb{V} \cup \mathbb{G}$. In addition, there

| Step | Details |
|---|---|
| Projection step | solve a relaxed version of (4) to calculate $\boldsymbol{\gamma}$, which is the lower bound of $\frac{1}{2t}\left\|\left\{\tilde{\boldsymbol{\beta}} - t\nabla f(\tilde{\boldsymbol{\beta}})\right\} - \boldsymbol{\gamma}\right\|_2^2$, and also satisfies $\sum_j \gamma_j \leqslant \lambda \sum_{\mathbb{g}\in\mathbb{G}} \omega_{|\mathbb{g}}$. The value of $\boldsymbol{\gamma}$ is the projection of the vectors $\boldsymbol{\xi}_{|\mathbb{g}}$. |
| Updating step | update $(\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}})_{X_j\in\mathbb{V}}$ by maximizing $\sum_{X_j\in\mathbb{V}} \sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}}$, while keeping $\sum_{X_j\in\mathbb{g}} \boldsymbol{\xi}^j_{|\mathbb{g}} \leqslant \lambda\omega_{\mathbb{g}}$. By doing so, we can ensure that the constraint in (4) holds. This can be done by the max flow algorithm. Details of the implementation can be found in Section 4. |
| Recursion step/divide and conquer | According to the mini-cut theorems (Ford and Fulkerson, 1956), define $\mathbb{V}^* = \{X_j \in \mathbb{V} : \sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}} = \gamma_j\}$, and $\mathbb{G}^* = \{\mathbb{g} \in \mathbb{G} : \sum_{X_j\in\mathbb{g}} \boldsymbol{\xi}^j_{|\mathbb{g}} < \lambda\omega_{\mathbb{g}}\}$. Then apply steps 1 and 2 to $(\mathbb{V}^*, \mathbb{G}^*)$ and their respective complements until $(\sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\xi}^j_{|\mathbb{g}})_{X_j\in\mathbb{V}}$ (obtained from step 2) matches $\boldsymbol{\gamma}$ (obtained from step 1). |

Table S1: Steps of `computeFlow`

is an arc $e \in E \subseteq V \times V$ from $s_1$, $\mathbb{g}_k$, and $X_j$ to $\mathbb{g}_k$, $X_j$, and $s_2$ respectively. From one vertex $v$ to another $w$, each arc has attributes such as non-negative functions of flow $f(v,w)$, which equals $-f(w,v))$, capacity $c(v,w) \geqslant f(v,w)$, the flow excess $h(v) = \sum_{u\in V} f(u,v) \geqslant 0, \forall v \in \{V - \{s_1\}\}$, and the residual capacity $r(v,w) = c(v,w) - f(v,w)$. See Table 1 for the definitions of those functions. Therefore, the updating step in Algorithm 1 can be formulated as "finding the maximum value of the flow while ensuring that the flow on each arc does not exceed its capacity".

There are two basic operations in the max flow algorithm. One is *push*, which pushes the excess from $v$ to $w$ by $\min\{h(v), r(v,w)\}$ when $h(v) > 0$ and $r(v,w) > 0$. The other is *relabel*, which estimates the distance from a vertex $v$ to the sink $s_2$. Define the distance as $d(v)$, where $d(s_1) = |V|$. Relabeling updates the $d(v)$ to $\min\{d(w) + 1 | r(v,w) > 0, d(v) < d(w)\}$.

## Web Appendix D Grouping structure identification in the simulation

The developed methods (similar to the overlapping group Lasso in (Wang et al., 2024)) can enforce a number of groups of variable coefficients to be 0 with a certain level of penalization. The remaining variables are said to be selected.

For the ease of notation, we use $A$ to represent $A(t)$. Consider the selection rule "if $\{A_1B, A_2B\}$ is selected, then $\{A_1, A_2, B\}$ must be selected". Suppose for now all candidate variables are $\mathbb{V} = \{A_1, A_2, B, A_1B, A_2B\}$, which are the variables that are involved in this rule. According to Table 2 in (Wang et al., 2024), the selection dictionary (all permissible subsets of covariates that respect the selection dependency) is

$$\mathbb{D} = \{\emptyset, \{A_1, A_2\}, \{B\}, \{A_1, A_2, B\}, \{A_1, A_2, B, A_1B, A_2B\}\}.$$

Based on Theorem 5 in (Wang et al., 2024), we need to create groups whose complements (and their combinations) are equal to $\mathbb{D} \setminus \mathbb{V}$. We thus postulate three groups: $\{A_1, A_2, A_1B, A_2B\}$ , $\{B, A_1B, A_2B\}$ and $\{A_1B, A_2B\}$, which satisfy the requirement. Similarly, to respect the selection dependency "if $\{C_1B, C_2B\}$ is selected, then $\{C_1, C_2, B\}$ must be selected", we postulate another three groups $\{C_1, C_2$ , $C_1B$ , $C_2B\}$ , $\{B, C_1B, C_2B\}$ and $\{C_1B, C_2B\}$.

However, the two rules share a same variable $B$: if either $\{A_1B, A_2B\}$ or $\{C_1B, C_2B\}$ is selected, then $B$ must be selected. To satisfy this requirement, we need to merge the two groups $\{B, A_1B, A_2B\}$ and $\{B, C_1B, C_2B\}$ into one group $\{A_1B$ , $A_2B$ , $B$ , $C_1B$ , $C_2B\}$ to prevent the occurrence of rule-breaking combinations for example, $\{C_1B, C_2B\}$ being selected without $B$.

We also need to respect another selection rule: the dummy variables for a categorical variable need to be selected collectively. The categorical interaction variables $AB$ and $BC$ are already being selected collectively because of the above selection dependencies. However, additional groups for $\{A_1, A_2\}$ are unnecessary as this would make it possible to select $\{A_1B, A_2B\}$

without $A$. In addition, with the above groups, $A_1,$ and $A_2$ would never be selected individually because they are always in a same group.

Therefore, we have 5 defined groups listed below

$$\mathbb{g}_1 = \{A_1, A_2, A_1B, A_2B\}, \mathbb{g}_2 = \{B, A_1B, A_2B, C_1B, C_2B\},$$

$$\mathbb{g}_3 = \{A_1B, A_2B\}, \mathbb{g}_4 = \{C_1, C_2, C_1B, C_2B\}, \mathbb{g}_5 = \{C_1B, C_2B\}.$$

Our resulting selection dictionary is: $\{\emptyset$ , $\{B\}$ , $\{C_1, C_2\}$ , $\{B, C_1, C_2\}$ , $\{B, C_1, C_2, C_1B, C_2B\}$ , $\{A_1, A_2\}$ , $\{A_1B, A_2B, B\}$ , $\{A_1, A_2, C_1$ , $C_2\}$ , $\{A_1, A_2, A_1B, A_2B, B\}$ , $\{A_1, A_2, B, C_1, C_2\}$ , $\{A_1, A_2, B, C_1, C_2, A_1B, A_2B\}$ , $\{A_1, A_2, B$ , $C_1, C_2, C_1B, C_2B\}$ , $\{A_1, A_2$ , $B, A_1B, A_2B$ , $C_1, C_2, C_1B, C_2B\}\}$. The code to derived the selection dictionary using `R` is available at `https://github.com/Guanbo-W/sox_sim`. One can verify the correctness of the derived selection dictionary using Theorem 5 in (Wang et al., 2024).

## Web Appendix E Additional simulation results for Section 5.2

See the results in Table S2

## Web Appendix F Time-fixed sparse group Lasso and the latent overlapping group Lasso as a special case of the proposed method

There is no structured variable selection available for time-dependent Cox models. Within time-fixed Cox models, structured variable selection such as sparse group Lasso and the latent overlapping group Lasso are available to perform structured variable selection.

Our proposed method solves

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) + \lambda \sum_{\mathbb{g} \in \mathbb{G}} \omega_{\mathbb{g}} \left\| \boldsymbol{\beta}_{|\mathbb{g}} \right\|, \tag{S1}$$

where $\mathbb{g}_i$ and $\mathbb{g}_j, i \neq j$ can be overlapped. The norm can be $\ell_2$ or $\ell_\infty$ norm. In this work, we use $\ell_\infty$ norm. These two norms have similar performance (Jenatton et al., 2011). Because

| Method | sox | sox.db | CoxL | CoxL.db | sox | sox.db | CoxL | CoxL.db |
|---|---|---|---|---|---|---|---|---|
| $p = 210$ | | | $n = 400$ | | | | $n = 800$ | |
| JDR | 0.15 | 0.05 | 0.00 | 0.00 | 0.45 | 0.00 | 0.40 | 0.05 |
| MR | 0.17 | 0.28 | 0.30 | 0.37 | 0.05 | 0.15 | 0.05 | 0.13 |
| FAR | 0.04 | 0.00 | 0.02 | 0.01 | 0.02 | 0.00 | 0.03 | 0.01 |
| R1S | 1.00 | 1.00 | 0.98 | 0.99 | 1.00 | 1.00 | 0.98 | 0.99 |
| RCI | 0.83 | 0.81 | 0.82 | 0.81 | 0.83 | 0.82 | 0.83 | 0.82 |
| MSE* | 7.01 | 6.39 | 9.17 | 6.93 | 5.79 | 5.76 | 6.48 | 5.70 |
| CV-E | 1.82 | 1.78 | 1.87 | 1.77 | 1.73 | 1.73 | 1.75 | 1.71 |
| $p = 465$ | | | $n = 400$ | | | | $n = 800$ | |
| MR | 0.18 | 0.33 | 0.36 | 0.40 | 0.04 | 0.13 | 0.07 | 0.13 |
| FAR | 0.02 | 0.00 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 | 0.00 |
| R1S | 1.00 | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 |
| RCI | 0.84 | 0.81 | 0.83 | 0.81 | 0.83 | 0.82 | 0.83 | 0.82 |
| MSE* | 3.22 | 2.85 | 3.71 | 3.00 | 2.63 | 2.55 | 3.14 | 2.58 |
| CV-E | 1.83 | 1.78 | 1.87 | 1.74 | 1.71 | 1.71 | 1.76 | 1.70 |
| $p = 820$ | | | $n = 400$ | | | | $n = 800$ | |
| MR | 0.20 | 0.31 | 0.40 | 0.45 | 0.04 | 0.18 | 0.10 | 0.16 |
| FAR | 0.02 | 0.00 | 0.01 | 0.01 | 0.01 | 0.00 | 0.01 | 0.00 |
| R1S | 1.00 | 1.00 | 0.99 | 0.99 | 1.00 | 1.00 | 0.99 | 0.99 |
| RCI | 0.84 | 0.80 | 0.82 | 0.81 | 0.83 | 0.82 | 0.84 | 0.83 |
| MSE* | 1.89 | 1.69 | 2.27 | 1.87 | 1.49 | 1.46 | 1.78 | 1.40 |
| CV-E | 1.82 | 1.78 | 1.90 | 1.74 | 1.73 | 1.73 | 1.78 | 1.69 |

Table S2: Simulation results of the high-dimensional case. In the tuning process, "lambda.1se" is used. Results are averaged over 20 independent replications. CoxL: unstructured $\ell_1$ penalty (`glmnet` with `"cox"` family); sox: our method; .db: with additional debiasing procedure. JDR: joint detection rate, MR: missing rate, FAR: false alarm rate, R1S: rule 1 satisfaction, RCI: the C index of the model with the selected variables, MSE: mean-squared error (*values are multiplied by $10^{-3}$), CV-E: cross-validated error.

of the nature of overlapping groups, many selection rules can be respected by solving this optimization problem, such as strong heredity.

The latent overlapping group Lasso solves

$$\min_{\boldsymbol{\beta},\boldsymbol{\gamma}} f(\boldsymbol{\beta}) + \lambda \sum_{\mathbb{g}\in\mathbb{G}} \omega_{\mathbb{g}} \left\|\boldsymbol{\gamma}_{|\mathbb{g}}\right\|_2, \quad \text{s.t.} \boldsymbol{\beta} = \sum_{\mathbb{g}\in\mathbb{G}} \boldsymbol{\gamma}_{|\mathbb{g}}, \tag{S2}$$

where $\mathbb{g}_i$ and $\mathbb{g}_j, i \neq j$ can be overlapped. Similar types of selection rules can be incorporated by the latent overlapping group Lasso. However, it is not scalable to high-dimensional settings with complex grouping structures due to the built-in algorithms. For instance, the method is

not well studied when multi-layer groups (such as tree and graph structures) have significant overlap and the sparsity level is low.

Sparse group Lasso solves

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) + (1-\alpha)\lambda \sum_{\mathbb{g}\in\mathbb{G}} \omega_{\mathbb{g}} \left\|\boldsymbol{\beta}_{|\mathbb{g}}\right\|_2 + \alpha\lambda \left\|\boldsymbol{\beta}_{|\mathbb{g}}\right\|_1, \tag{S3}$$

where $\mathbb{g}_i$ and $\mathbb{g}_j, i \neq j$ cannot be overlapped. Only one type of selection rule can be incorporated by using sparse group Lasso, "select a number of variable in each of the non-overlapped groups," where the number is from zero to the number of variables in each group. The number cannot be pre-specified and is data-driven.

## Web Appendix G Sample solution path

See Figure S1 for the sample solution path.

## Web Appendix H Additional simulation: Comparison with existing sparse group lasso methods (additional simulations)

In this simulation, we compare our method, the sparse group LASSO, and the LASSO, implemented by `sox`, `SGL`, and `glmnet` respectively. We aim to show that `sox` can respect certain selection rules that `SGL` or `glmnet` cannot. We follow a similar setting as the one in Section 5.2 within the cases of ($n = 400/800, p = 210$). Since `SGL` cannot handle time-dependent Cox models, we generate time-fixed covariates.

The sparse group LASSO, due to its inability to accommodate overlapping groups, does not adhere to the principle of strong heredity. To define its group structure, we have organized the groups in a specific manner: each of the 10 groups contains two consecutive main terms along with their interaction (we have 20 main terms, so 10 such groups are specified, each containing three variables). Additionally, the 180 remaining interactions are each treated as individual groups. This configuration results in a total of 190 distinct groups.

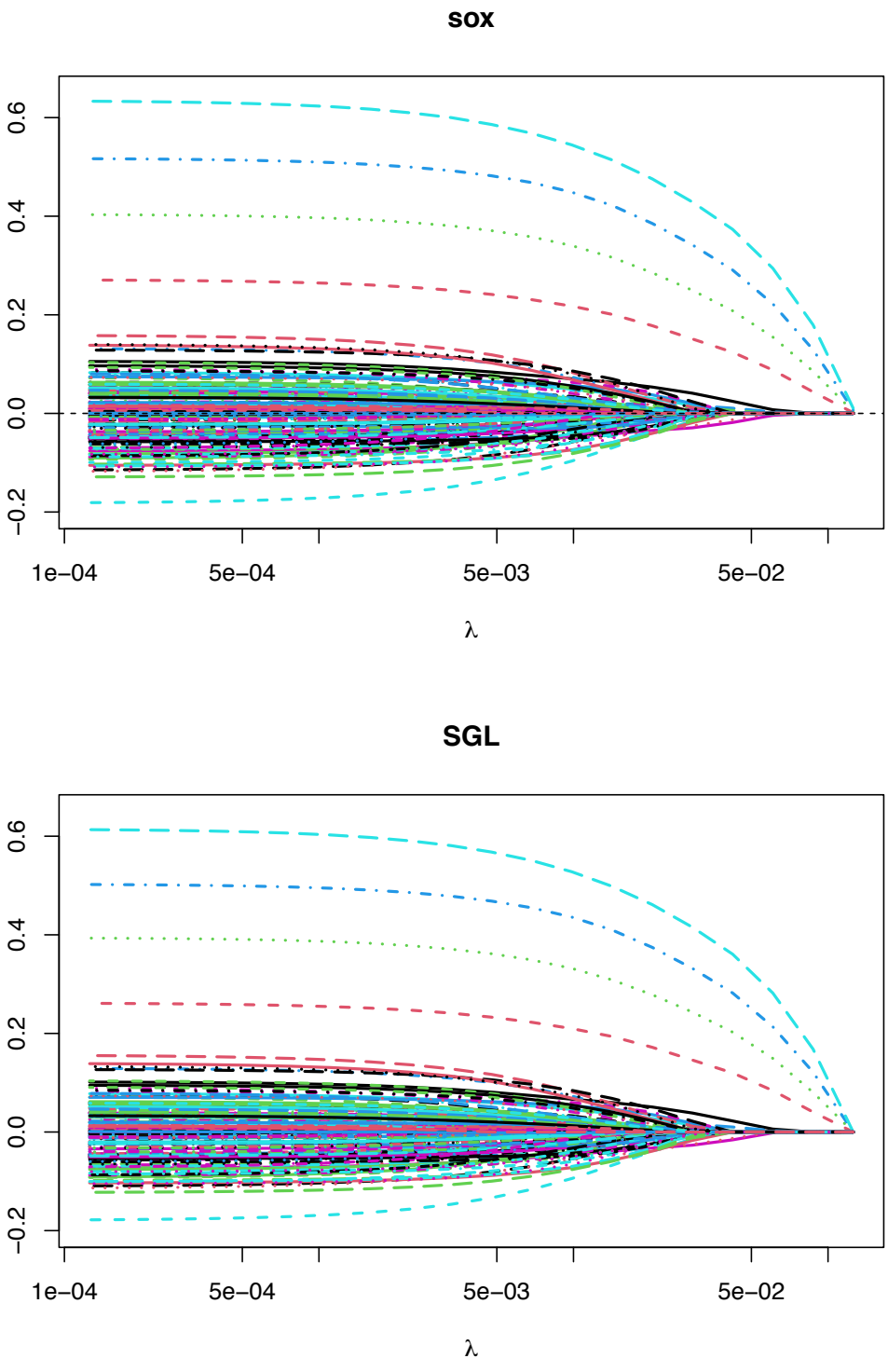


**Figure S1**: Sample solution paths from `sox` and `SGL` using the same data and the same $\lambda$ sequence.

| Method | sox | sox.db | CoxL | CoxL.db | SGL | sox | sox.db | CoxL | CoxL.db | SGL |
|---|---|---|---|---|---|---|---|---|---|---|
| $p = 210$ | | | $n = 400$ | | | | | $n = 800$ | | |
| MR | 0.04 | 0.05 | 0.06 | 0.08 | 0.02 | 0.00 | 0.01 | 0.00 | 0.00 | 0.18 |
| FAR | 0.32 | 0.22 | 0.17 | 0.14 | 0.76 | 0.34 | 0.24 | 0.22 | 0.18 | 0.40 |
| R1S | 1.00 | 1.00 | 0.89 | 0.91 | 0.81 | 1.00 | 1.00 | 0.87 | 0.90 | 0.87 |
| RCI | 0.84 | 0.84 | 0.84 | 0.84 | 0.88 | 0.82 | 0.82 | 0.82 | 0.82 | 0.72 |
| MSE* | 3.41 | 2.37 | 3.77 | 3.18 | 3727 | 2.21 | 1.50 | 2.37 | 1.80 | 3242 |
| CV-E | 5.51 | 5.34 | 5.52 | 5.30 | 48.02 | 5.48 | 5.38 | 5.48 | 5.36 | 53.14 |

Table S3: Simulation results of interaction selection under the time-fixed, high-dimensional setting. In the tuning process, "lambda.1se" is used. Results are averaged over 20 independent replications. CoxL: unstructured $\ell_1$ penalty (`glmnet` with `"cox"` family); sox: our method; SGL: the sparse group LASSO; .db: with additional debiasing procedure. JDR: joint detection rate, MR: missing rate, FAR: false alarm rate, R1S: rule 1 satisfaction, RCI: the C index of the model with the selected variables, MSE: mean-squared error (*values are multiplied by $10^{-3}$), CV-E: cross-validated error.

The results, presented in Table S3, show that `sox` outperforms `glmnet`, consistent with the findings in Section 5.2. The inferior performance of `glmnet` is attributed to its inability to incorporate selection rules. In this simulation, a flawed grouping structure was applied to the sparse group LASSO, leading it to adhere to selection rules not aligned with the actual data generation mechanism. This situation parallels Bayesian analysis, where an incorrect prior leads to suboptimal outcomes, which explains why `SGL` demonstrates the least effective performance in this context.

## Web Appendix I Additional simulation: Timing results.

Additionally, we also tested the computational speed of `sox`. We adopted the simulation setting from Section 5.2. For all $(n, p)$ pairs, we reported the computation time (in seconds) for solving 10-fold cross-validation on the same $\lambda$ sequence of length 30 in Table S4. The computation time of `sox.db` does not include the necessary procedures to acquire the initial estimates (`sox` in our case) used to calculate the regularization weights. Our findings indicate that for a complex case where the sample size is $n = 800$ and the number of variables

| | $n = 400$ | | $n = 800$ | |
|---|---|---|---|---|
| | sox | sox.db | sox | sox.db |
| $p = 210$ | 26.80 | 26.47 | 54.13 | 53.93 |
| $p = 465$ | 121.39 | 114.12 | 217.95 | 217.11 |
| $p = 820$ | 391.26 | 359.85 | 704.67 | 698.86 |

Table S4: Timing results of the high-dimensional case. Results are averaged over 20 independent replications.

is $p = 820$, a comprehensive analysis can be completed in approximately 10 minutes. In contrast, for a simpler scenario with a sample size of $n = 400$ and $p = 210$ variables, the analysis requires less than 30 seconds to finish.

## Web Appendix J Additional simulation: Comparison of different group sizes, the amount of overlap, and the sparsity levels

In this section, we delved into how group size, the amount of overlap between groups, and sparsity level influence the performance of our method. We adopt a low-dimensional setting $(n = 400, p = 25)$ time-dependent (with four time-points, and each variable held constant for two or three times-points). Each variable is independently generated by the standard Gaussian distribution. We investigate seven grouping structures with different group sizes and amount of overlap, the details of which are summarized in Table S5.

The results of the simulation are detailed in Table S6. A closer look at settings 1, 3, and 2, which feature groups of 10 variables each with similar sparsity but different overlap sizes (8, 5, and 2, respectively), reveals a decrease in false alarm rates. This trend is attributed to the fact that smaller overlaps reduce the probability of mistakenly selecting variables from both groups, thereby lowering the chances of misidentifying noise as a significant signal. Although setting 1 shows a slightly higher sparsity level, it does not substantially affect the outcome.

In contrast, settings 4, 3, and 5, which have identical overlap and sparsity levels but varying group sizes (13, 10, and 7, respectively), also show a declining trend in false alarm rates. This

| | Variable index | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Setting** | **True Coefficient** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0** | **0.4** | **0.4** | **0.4** | **0** | **0** | **0.4** | **0** | **0** | **0.4** | **0** | **0** | **0** |
| **1** | **Group size 10 Overlap size 8** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| **2** | **Group size 10 Overlap size 2** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| **3** | **Group size 10 Overlap size 5** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| **4** | **Group size 13 Overlap size 5** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| **5** | **Group size 7* Overlap size 5** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| **6** | **Group size 7** Overlap size 5** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
| **7** | **Group size 7*** Overlap size 5** | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 |

Table S5: True coefficients and grouping structures for the simulation to evaluate the effects of group sizes and the amount of overlaps. In all seven settings, there is a group 1 in red and a group 2 in blue with their overlaps in purple. For the variables that are in neither groups 1 or 2, each belongs to a group of size one.

is because the selection of variables 14 and 15 may result in the selection of the variables in group 2. When group 2 has more variables, there is an increased risk of erroneously selecting them.

Regarding settings 5 to 7, where each group consists of 7 variables with an overlap size of 5 and varying sparsity levels for group 2 (0.7, 0.9, and complete sparsity or 1), settings 5 and 6 display similar false alarm rates. This is due to the possibility of a single signal in a group triggering the erroneous selection of all variables in that group. However, with complete sparsity, the false alarm rate tends to zero. Notably, setting 6 exhibits a significantly higher missing rate, possibly because a lone signal in group 2 (variable 14) is sometimes not strong enough for selection, leading to its omission.

These findings demonstrate that group size, overlap, and sparsity levels significantly influence the performance of `sox`. Nonetheless, caution should be exercised in generalizing these results, as variations in the data-generating mechanism can alter these conclusions.

| Setting | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Overlap size | 8 | 2 | 5 | 5 | 5 | 5 | 5 |
| Group size | 10 | 10 | 10 | 13 | 7 | 7 | 7 |
| SL* of group 2 | 0.8 | 0.7 | 0.7 | 0.7 | 0.7 | 0.9 | 1 |
| SL* of group 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| MR | 0.01 | 0.00 | 0.01 | 0.01 | 0.00 | 0.11 | 0.00 |
| FAR | 0.36 | 0.30 | 0.31 | 0.46 | 0.17 | 0.17 | 0.01 |
| RCI | 0.76 | 0.76 | 0.76 | 0.76 | 0.76 | 0.75 | 0.76 |
| MSE** | 1.64 | 1.57 | 1.61 | 1.66 | 1.51 | 1.76 | 1.55 |
| CV-E | 1.90 | 1.89 | 1.89 | 1.89 | 1.88 | 1.91 | 1.88 |

Table S6: Simulation results for different group sizes and amount of overlap. * Sparsity Level; ** MSEs are multiplied by $10^{-2}$.

## Web Appendix K Inclusion and exclusion criteria

Figure S2 shows the inclusion and exclusion criteria for the patients of the study cohort.

## Web Appendix L Covariate definitions

The 7 baseline covariates are 1) Age ($\geqslant 75/ < 75$), 2) Sex(female/male), 3) CHA2DS2VASc ($\geqslant 3/ < 3$), 4) Diabetes, 5) COPD/asthma, 6) Hypertension and 7) Malignant cancer. All other covariates are time-dependent covariates.

Heart disease: including 1) valvular heart disease, 2) peripheral vascular disease, 3) cardiovascular disease, 4) chronic heart failure, and 5) myocardial infarction.

DOAC: it is 1 if the patient uses DOAC, 0 if the patient is taking warfarin or not taking any OAC at the time $t$.

OAC: it is 1 if the patient uses OAC, 0 if the patient is not taking any OAC at the time $t$.

High-dose-DOAC: it is 1 if the patient uses high-dose-DOAC, 0 if the patient is taking low-dose-DOAC, or warfarin or not taking any OAC at the time $t$.

Antiplatelets: ASA low dose ($\geqslant$ 80 mg and $\leqslant$ 260 mg), dipyridamole or clopidrogel or ticlopidine or prasugrel or ticagrelor.

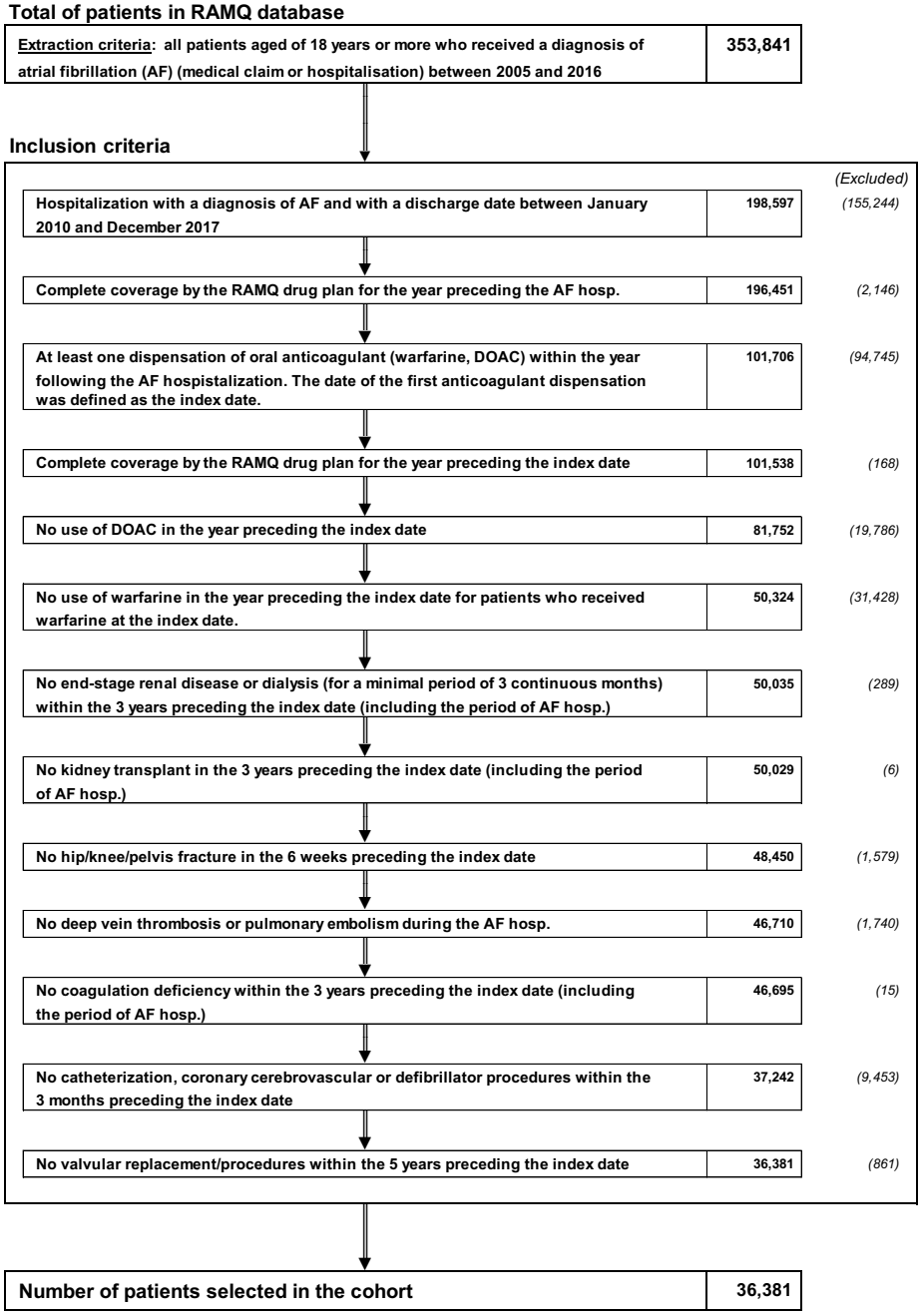


**Figure S2**: The inclusion and exclusion criteria for the patients of the study cohort. (AF: atrial fibrillation; OAC: oral anticoagulants).

Statin: Statin or other lipid lowering drugs.

NASIDs: Nonsteroidal anti-inflammatory drugs.

## Web Appendix M Summary statistics of all the covariates

Table S7 gives the summary statistics of all the covariates stratified by if the subject experienced the event (death) during the follow-up. All the covariates in this analysis are binary variables. The first two columns (% at cohort entry) show the summary statistics of the covariates at the time of cohort entry. The second two columns (% of value changed) show, for each covariate, the percentage of patients who experienced situation change during the follow-up. The third two columns (% mean over population and time) show, for each covariate, the average value of the covariates across all patients during the follow-up (the mean, over the population, of the values of the covariates during the follow-up).

## Web Appendix N Analysis using the time-dependent Cox model

Table S8 provides the crude (univariate) and adjusted hazard ratios from simple and multivariate time-dependent Cox models for death, respectively, and 95% confidence intervals using the covariates in the analysis.

## Web Appendix O Rational of the selection rules

Selection rules 1 and 2: rule 1 is needed since when DOAC is in the model, and if Apixaban is selected, then the interpretation of the coefficient of Apixaban is the contrast of Apixaban and Rivaroxaban. If High-dose-DOAC is also in the model, the interpretation would be the contrast (e.g. log hazard ratio) of low-dose-Apixaban versus low-dose Rivaroxaban. However, without DOAC, the coefficient of Apixaban would represent a contrast against warfarin and Rivaroxaban combined, which is less interpretable. The same rationale applies to Dabigatran in rule 2.

Selection rule 3: it is needed because when OAC is in the mode, and if DOAC is selected, then the interpretation of the coefficient of DOAC is the contrast of DOAC and warfarin.

| Covariate | % at cohort entry | | % of value changed | | % mean | |
|---|---|---|---|---|---|---|
| Age ($\geqslant 75$) | 67 | 82 | 0 | 0 | 66 | 82 |
| Sex(female/male) | 54 | 52 | 0 | 0 | 54 | 52 |
| **Comorbidities/Medical score** | | | | | | |
| $CHA_2DS_2VAS_c$ ($\geqslant 3$) | 80 | 89 | 0 | 0 | 79 | 88 |
| Diabetes | 34 | 39 | 0 | 0 | 34 | 39 |
| COPD/asthma | 35 | 51 | 0 | 0 | 35 | 49 |
| Hypertension | 81 | 84 | 0 | 0 | 81 | 84 |
| Malignant cancer | 23 | 36 | 0 | 0 | 23 | 36 |
| Stroke | 19 | 16 | 3 | 5 | 21 | 18 |
| Chronic kidney disease | 33 | 53 | 6 | 14 | 36 | 58 |
| Heart disease | 66 | 80 | 7 | 9 | 70 | 83 |
| Major bleeding | 28 | 38 | 9 | 18 | 33 | 45 |
| **OAC use** | | | | | | |
| DOAC | 61 | 51 | 58 | 47 | 54 | 38 |
| Apixaban | 31 | 29 | 26 | 25 | 27 | 22 |
| Dabigatran | 11 | 7 | 13 | 8 | 10 | 5 |
| OAC | 100 | 100 | 91 | 94 | 83 | 70 |
| High-dose-DOAC | 39 | 23 | 41 | 23 | 34 | 17 |
| **Concomitant medication use** | | | | | | |
| Antiplatelets | 52 | 59 | 43 | 34 | 24 | 37 |
| NSAIDs | 7 | 5 | 11 | 6 | 3 | 3 |
| Statin | 54 | 52 | 11 | 11 | 53 | 49 |
| Beta-Blockers | 65 | 63 | 18 | 12 | 62 | 62 |
| **Potential drug-drug interaction** | | | | | | |
| DOAC: Antiplatelets | 27 | 25 | 30 | 27 | 10 | 11 |
| DOAC: NSAIDs | 4 | 3 | 7 | 4 | 2 | 1 |
| DOAC: Statin | 30 | 23 | 33 | 24 | 29 | 18 |
| DOAC: Beta-Blockers | 38 | 30 | 40 | 32 | 34 | 22 |

Table S7: Summary statistics of the baseline and time-dependent covariates

However, without OAC, the coefficient of DOAC would represent a contrast against DOAC and warfarin or taking none of OAC combined, which is less interpretable.

Selection rule 4: it is needed since when DOAC is in the model, and if High-dose-DOAC is selected, then the interpretation of the coefficient of High-dose-DOAC is the contrast of high-dose-DOAC versus low-dose-DOAC, which is of interest. If Apixaban and Dabigatran are also in the model, the coefficient of High-dose-DOAC represents the contrast between high-dose-Rivaroxaban versus low-dose-Rivaroxaban. However, without DOAC in the model,

| Covariate | Crude HR | | Adjusted HR | |
|---|---|---|---|---|
| | Estimate | CI | Estimate | CI |
| Age ($\geqslant 75$) | 2.25 | (2.09, 2.44) | 1.89 | (1.73, 2.06) |
| Sex(female/male) | 0.92 | (0.86, 0.97) | 0.98 | (0.92, 1.04) |
| **Comorbidities/Medical score** | | | | |
| CHA2DS2VASc ($\geqslant 3$) | 2.00 | (1.82, 2.20) | 1.01 | (0.90, 1.14) |
| Diabetes | 1.23 | (1.16, 1.31) | 1.08 | (1.02, 1.15) |
| COPD/asthma | 1.84 | (1.74, 1.96) | 1.49 | (1.40, 1.58) |
| Hypertension | 1.20 | (1.10, 1.30) | 0.91 | (0.83, 0.99) |
| Malignant cancer | 1.79 | (1.68, 1.90) | 1.45 | (1.36, 1.54) |
| Stroke | 1.07 | (1.00, 1.15) | 1.00 | (0.92, 1.07) |
| Chronic kidney disease | 3.50 | (3.29, 3.73) | 2.10 | (1.96, 2.25) |
| Heart disease | 3.42 | (3.11, 3.76) | 2.41 | (2.18, 2.67) |
| Major bleeding | 2.55 | (2.40, 2.70) | 1.38 | (1.29, 1.47) |
| **OAC use** | | | | |
| DOAC | 0.06 | (0.06, 0.07) | 1.60 | (1.16, 2.21) |
| Apixaban | 0.11 | (0.09, 0.13) | 0.88 | (0.68, 1.14) |
| Dabigatran | 0.09 | (0.07, 0.12) | 0.77 | (0.53, 1.13) |
| OAC | 0.03 | (0.02, 0.03) | 0.84 | (0.66, 1.06) |
| High-dose-DOAC | 0.07 | (0.06, 0.08) | 0.80 | (0.63, 1.01) |
| **Concomitant medication use** | | | | |
| Antiplatelets | 1.37 | (1.28, 1.47) | 0.80 | (0.74, 0.86) |
| NSAIDs | 1.14 | (0.97, 1.33) | 1.50 | (1.27, 1.76) |
| Statin | 0.70 | (0.66, 0.75) | 0.82 | (0.76, 0.87) |
| Beta-Blockers | 0.92 | (0.87, 0.98) | 1.37 | (1.28, 1.47) |
| **Potential drug-drug interaction** | | | | |
| DOAC: Antiplatelets | 0.11 | (0.09, 0.15) | 1.09 | (0.81, 1.47) |
| DOAC: NSAIDs | 0.15 | (0.09, 0.26) | 0.99 | (0.56, 1.78) |
| DOAC: Statin | 0.08 | (0.07, 0.09) | 0.90 | (0.70, 1.15) |
| DOAC: Beta-Blockers | 0.08 | (0.09, 0.10) | 0.60 | (0.47, 0.77) |

Table S8: Crude (univariate) and adjusted hazard ratios from simple and multivariate time-dependent Cox models for death, respectively, and 95% confidence intervals using the covariates in the analysis.

these relevant interpretations would be lost.

## Web Appendix P Grouping structure of the data analysis

In our method, we need to specify the grouping structure to respect the selection rules.

Thirteen variables are included in the 8 selection rules. For the convenience of grouping

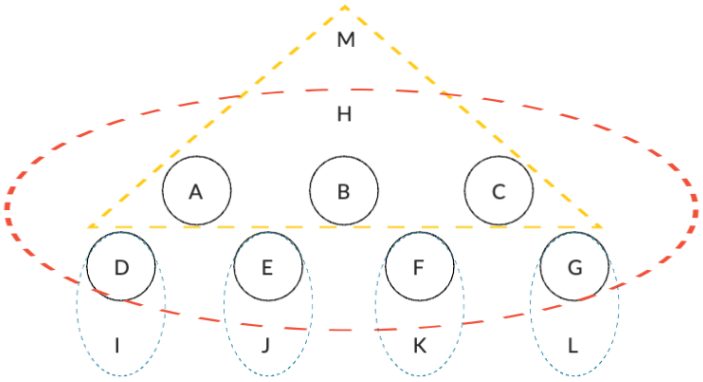


**Figure S3**: Grouping structure plot for the 13 groups

structure specification, we denote the 13 variables as such:

A: Apixaban, B: Dabigatran, C: High-dose-DOAC, D: DOAC: Antiplatelets, E: DOAC: NSAIDs, F: DOAC: Statin, G: DOAC: Beta-Blockers, H: DOAC, I: Antiplatelets, J: NSAIDs, K: Statin, L: Beta-Blockers, M: OAC.

According to (Wang et al., 2024, 2023). The grouping structure that relevant to these 13 variables should be

$$\mathcal{g}_1 = \{A\}, \mathcal{g}_2 = \{B\}, \mathcal{g}_3 = \{C\}, \mathcal{g}_4 = \{D\}, \mathcal{g}_5 = \{E\}, \mathcal{g}_6 = \{F\}, \mathcal{g}_7 = \{G\}, \mathcal{g}_8 = \{D, I\},$$

$$\mathcal{g}_9 = \{E, J\}, \mathcal{g}_{10} = \{F, K\}, \mathcal{g}_{11} = \{G, L\}, \mathcal{g}_{12} = \{A - H\}, \mathcal{g}_{13} = \{A, B, C, H, M\},$$

13 groups in total. For the remaining 11 variables, each of them has one individual group. Therefore, we have 24 groups in total. For the 13 groups, we plot the grouping structure in Figure S3. We can see that it presents a graph structure. Multiple groups are overlapped with and nested in other groups.

To encode the grouping structure into our developed software, we need to specify two matrices, both of which are 24 by 24 matrices. The first matrix has 1 in the positions (4, 8), (5, 9), (6, 10), (7, 11), (1, 12), (2, 12), (3, 12), (4, 12), (5, 12), (6, 12), (7, 12), (1, 13) (2, 13), (3, 13), the rest are 0. The second matrix has 1 in the positions $(i, i), i = 1, \ldots, 7, 13, \ldots, 24$,

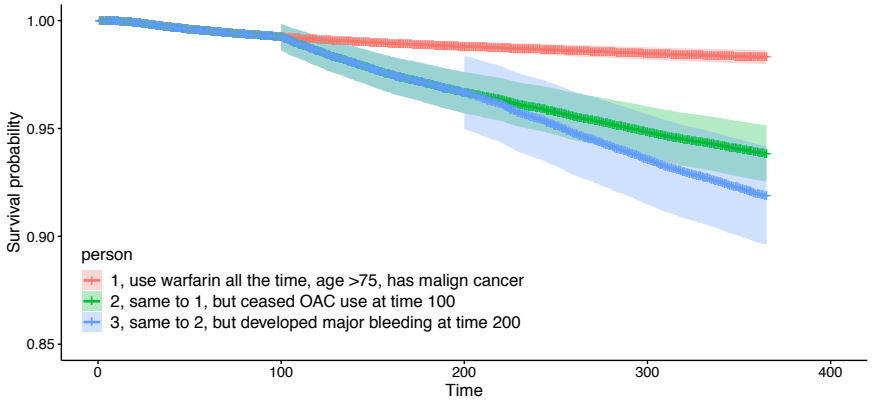


**Figure S4**: Estimated survival curves of three typical persons.

(9, 8), (10, 9), (11, 10), (12, 11), (8, 12), (8, 13), the rest are 0. For details, please see the help file in the R package.

## Web Appendix Q Visualization of the analysis

We artificially create three hypothetical patients' disease progression. Suppose person 1, age $\geqslant$ 75, who only used the drug warfarin, had only malign cancer among all the disease variables included in the data. Person 2 has the same profile as person 1 except they ceased warfarin at time 100 during the follow-up, while other statuses stayed the same. Person 3 developed major bleeding at time 200, and all other statuses were the same as person 2. Figure S4 shows the survival curves of the three people, estimated by the time-dependent Cox model using the covariates that were selected by our method. As we can see, person 1 (in red) has the highest estimated survival probability. The survival probability of person 2 drops immediately after the cease of warfarin. Similarly, the survival probability drops significantly at time 200 due to the major bleeding. Note that Figure S4 only intends to show, as an example, how the survival probability can vary according to time-dependent covariates, rather than predicting the survival probability of the hypothetical cases. All the covariates involved in the study are internal covariates that relate to the outcome in the

sense that the covariates can be measured only among the patients who are still at risk of the event (alive).

## References


Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network flows: theory, algorithms, and applications*. Prentice Hall.

Cherkassky, B. V. and Goldberg, A. V. (1997). On implementing the push—relabel method for the maximum flow problem. *Algorithmica* **19,** 390–410.

Cohen, J., Dadush, D. N., and Rothvoß, T. (2014). Dynamic minimum-cost flow algorithms. *SIAM Journal on Computing* **43,** 1042–1064.

Dadush, D. N., Gabow, H. N., and Rothvoß, T. (2019). A faster scaling algorithm for minimum-cost flow. *Mathematics of Operations Research* **44,** 501–527.

Ford, L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Canadian journal of Mathematics* **8,** 399–404.

Gabow, H. N., Hu, X., and Makarychev, K. (2020). An $o(m^{3/2} \log^2 m)$ algorithm for quadratic minimum-cost flows. *Mathematics of Operations Research* **45,** 125–145.

Gallo, G., Grigoriadis, M. D., and Tarjan, R. E. (1993). A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing* **22,** 131–146.

Goldberg, A. V. and Tarjan, R. E. (1988). A new approach to the maximum-flow problem. *Journal of the ACM (JACM)* **35,** 921–940.

Jenatton, R., Audibert, J.-Y., and Bach, F. (2011). Structured variable selection with sparsity-inducing norms. *The Journal of Machine Learning Research* **12,** 2777–2824.

Magdon-Ismail, M. and Atiya, A. F. (2016). Online learning for quadratic minimum-cost flows. *Operations Research Letters* **44,** 1–7.

Skutella, M. (2013). Minimum-cost flow problems. In *Encyclopedia of Operations Research and Management Science*, pages 952–956. Springer.

Wang, G., Perreault, S., Platt, R. W., Wang, R., Dorais, M., and Schnitzer, M. E. (2023). Integrating complex selection rules into the latent overlapping group lasso for constructing coherent prediction models. *arXiv preprint arXiv:2206.05337* .

Wang, G., Schnitzer, M., Chen, T., Wang, R., and Platt, R. W. (2024). A general framework for formulating structured variable selection. *Transactions on Machine Learning Research* .