# 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.

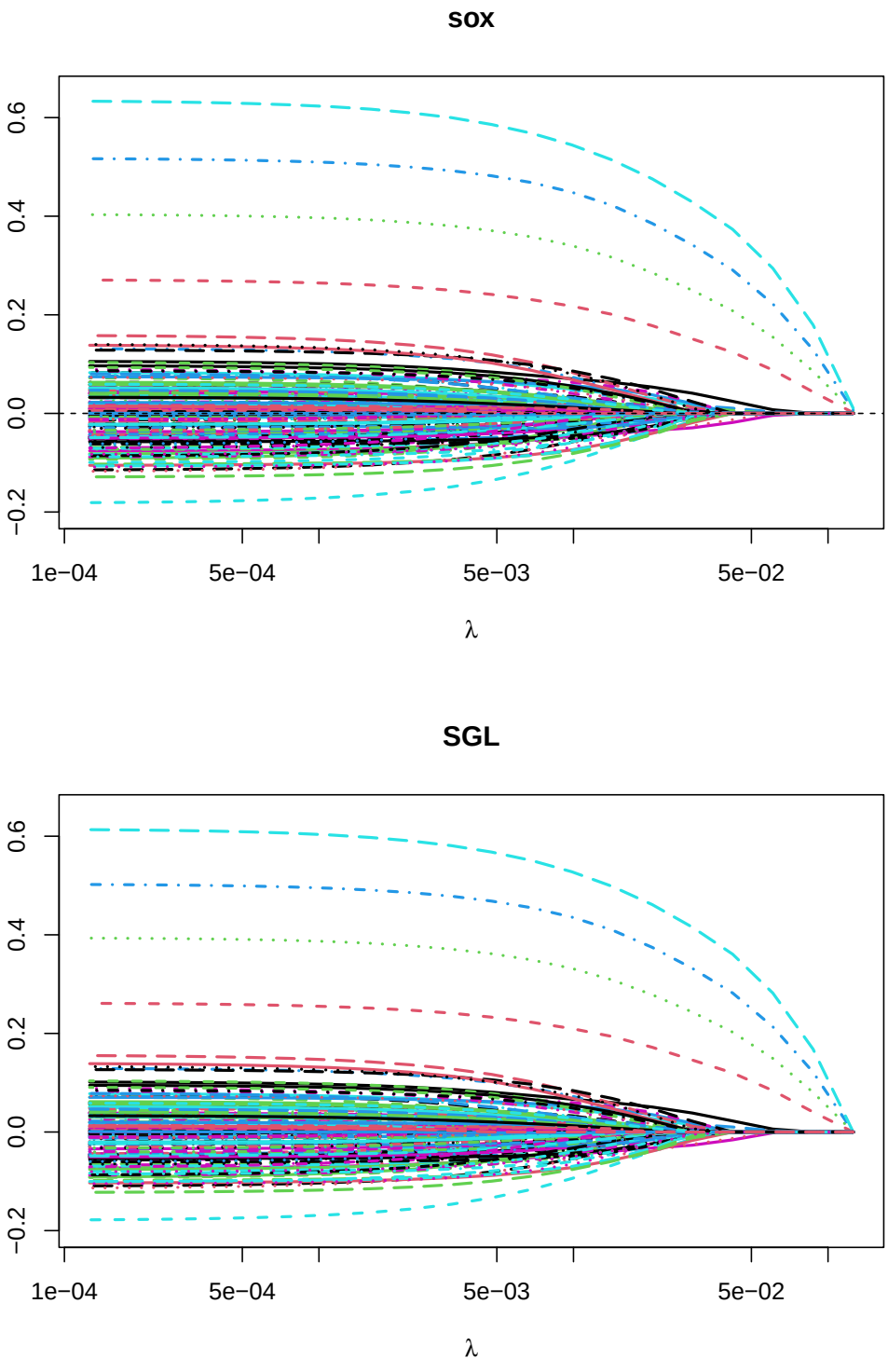


**Figure S1**: Sample solution paths from `sox` and `SGL` using the same data and the same $\lambda$ sequence.

| 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.

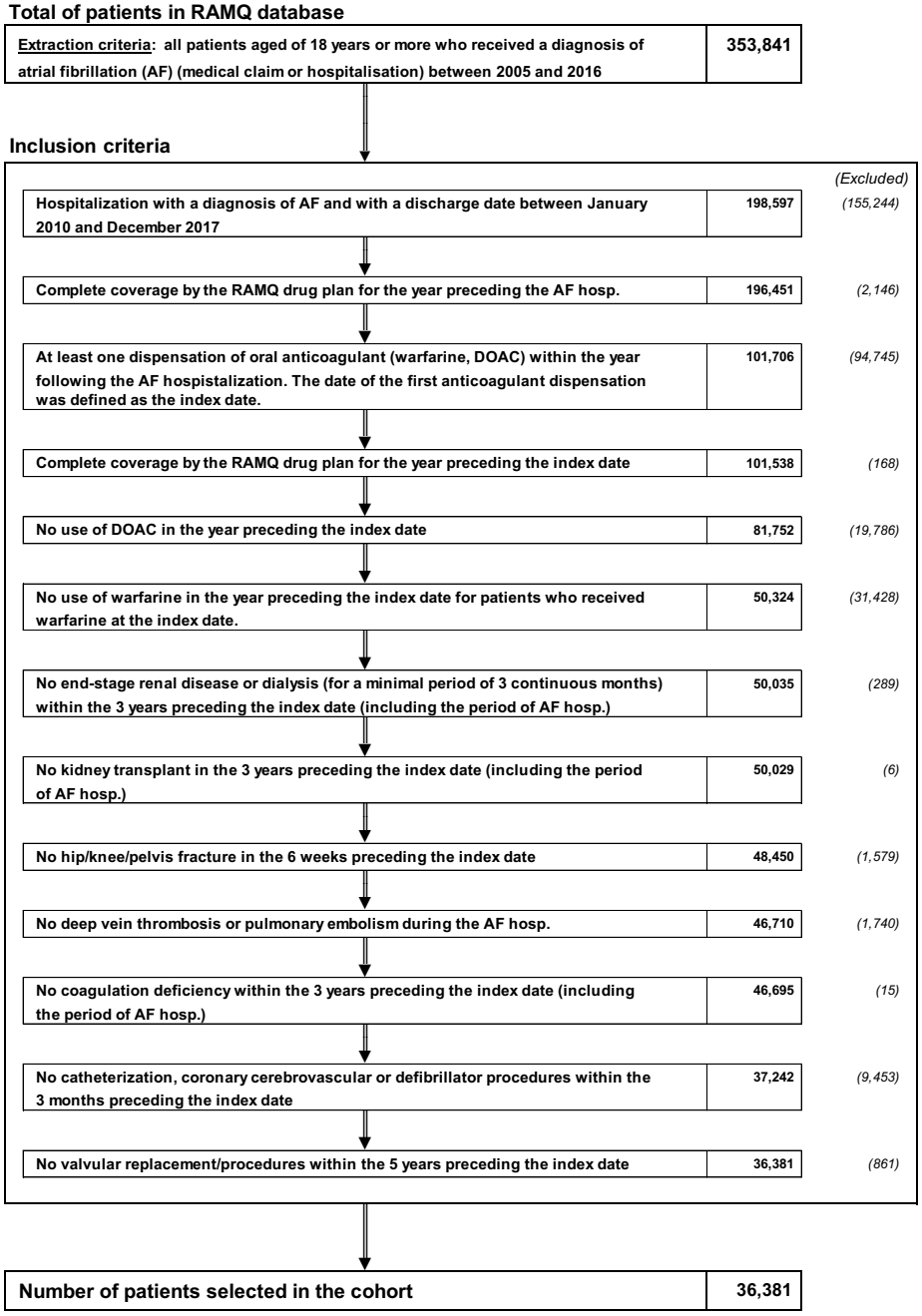


**Figure S2**: The inclusion and exclusion criteria for the patients of the study cohort. (AF: atrial fibrillation; OAC: oral anticoagulants).

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

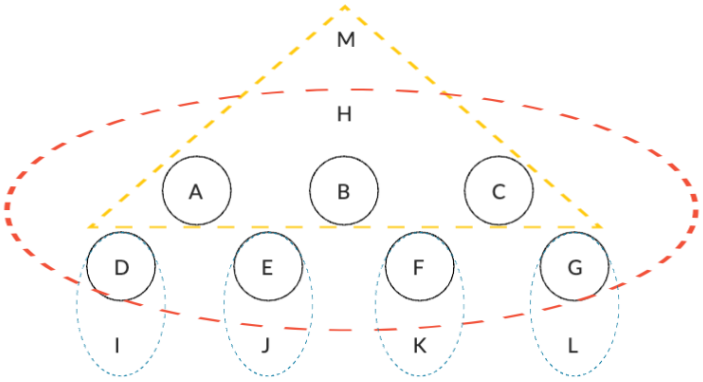


**Figure S3**: Grouping structure plot for the 13 groups

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 $(i, i), i = 1, \ldots, 7, 13, \ldots, 24$,

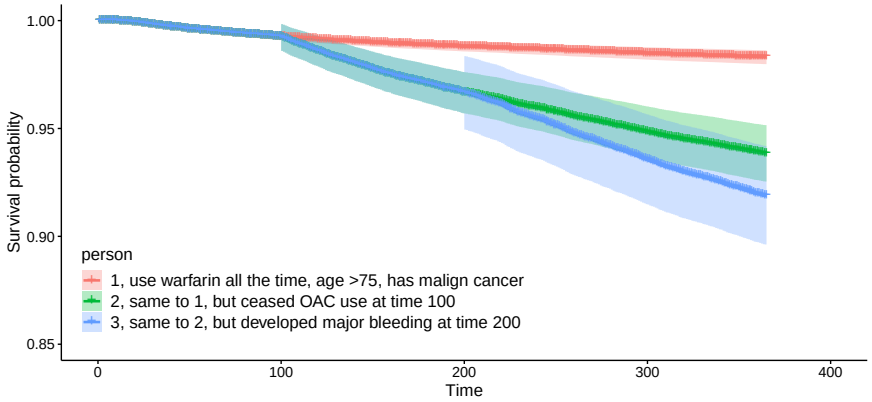


**Figure S4**: Estimated survival curves of three typical persons.

(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* .