# ADHAM: Additive Deep Hazard Analysis Mixtures for Interpretable Survival Regression

**Mert Ketenci** MK4130@COLUMBIA.EDU
*Department of Computer Science*
*Columbia University*
*New York, NY, USA*

**Vincent Jeanselme** VJ2292@CUMC.COLUMBIA.EDU
*Department of Biomedical Informatics*
*Columbia University*
*New York, NY, USA*

**Harry Reyes Nieva** HR2479@CUMC.COLUMBIA.EDU
*Department of Biomedical Informatics*
*Columbia University*
*New York, NY, USA*

**Shalmali Joshi** SJ3261@CUMC.COLUMBIA.EDU
*Department of Biomedical Informatics*
*Columbia University*
*New York, NY, USA*

**Noémie Elhadad** NOEMIE.ELHADAD@COLUMBIA.EDU
*Department of Biomedical Informatics*
*Columbia University*
*New York, NY, USA*

## Abstract

Survival analysis is a fundamental tool for modeling time-to-event outcomes in healthcare. Recent advances have introduced flexible neural network approaches for improved predictive performance. However, most of these models do not provide interpretable insights into the association between exposures and the modeled outcomes, a critical requirement for decision-making in clinical practice. To address this limitation, we propose Additive Deep Hazard Analysis Mixtures (ADHAM)[1], an interpretable additive survival model. ADHAM assumes a conditional latent structure that defines subgroups, each characterized by a combination of covariate-specific hazard functions. To select the number of subgroups, we introduce a post-training refinement that reduces the number of equivalent latent subgroups by merging similar groups. We perform comprehensive studies to demonstrate ADHAM's interpretability at the population, subgroup, and individual levels. Extensive experiments on real-world datasets show that ADHAM provides novel insights into the association between exposures and outcomes. Further, ADHAM remains on par with existing state-of-the-art survival baselines in terms of predictive performance, offering a scalable and interpretable approach to time-to-event prediction in healthcare.

1. Code available at github.com/ketencimert/adham.

## 1. Introduction

Survival analysis, a subfield of machine learning (ML) and statistics, focuses on modeling time-to-event outcomes (e.g., disease progression, hospital readmission, or relapse). In medical settings, the event of interest is often not observed for all patients due to study end, loss to follow-up, or withdrawal, resulting in censored data. This characteristic sets survival analysis apart from regression (Clark et al., 2003; Singh and Mukhopadhyay, 2011). To address this challenge, survival analysis leverages the survival function $S(t \mid \mathbf{x}) = p(T > t \mid \mathbf{x})$—which corresponds to the probability that an individual does not experience the event of interest past time $t$, for model learning (Haider et al., 2020). Survival analysis plays a critical role in healthcare, supporting both clinical decision-making and outcome evaluation in trials (Viganò et al., 2000; Fleming and Lin, 2000; Cole et al., 2001; Faucett et al., 2002; Hagar et al., 2014; Perotte et al., 2015; Panahiazar et al., 2015; Morita et al., 2009).

Despite growing interest in ML for healthcare, clinical adoption of ML survival models remains limited (Abdullah et al., 2021; Lu et al., 2023). A key limitation of existing methodologies is the lack of interpretability of model predictions (Shortliffe and Sepúlveda, 2018; Tonekaboni et al., 2019). Clinicians are unlikely to trust or act on model outputs without a clear understanding of how and why a prediction was made, especially when guiding high-stakes decisions such as individualized care or the development of treatment guidelines (Amann et al., 2020). This challenge is further complicated in survival analysis, where interpretability must account not only for which covariates induce the predictions, but also for how their influence evolves. Unlike standard ML tasks, survival interpretability requires continuous-time explanations that show how risk changes longitudinally.

The granularity of interpretability in clinical risk models can be broadly categorized into three levels (Ahmad et al., 2018): (1) Population-level interpretability, which captures how covariates influence outcomes across the entire cohort and helps identify global risk trends (Lou et al., 2013); (2) Subgroup-level interpretability, which uncovers patterns within latent groups of patients who share similar progression profiles and helps with tailored interventions (Bhavnani et al., 2022); and (3) Individual-level interpretability, which provides personalized insights into how specific covariates relate to a given patient's risk over time (Krzyziński et al., 2023). For clinical relevance, models should ideally provide explanations at the population, subgroup, and individual levels. Yet, methods that support all three levels remain largely unexplored.

The predominant class of interpretable models in survival analysis extends the classical Cox Proportional Hazards (CoxPH) framework (Cox, 1972), often through the incorporation of Generalized Additive Models (GAMs), a flexible approach that represents the hazard function as a sum of smooth, potentially nonlinear transformations of individual covariates (Jiang, 2022; Peroni et al., 2022). However, a key limitation of additive modeling lies in concurvity (Siems et al., 2023; Kovács, 2024), a form of multicollinearity where correlated features obscure each other's effects, entangling their individual contributions. This phenomenon can lead to explanations that misalign with established physiological mechanisms or clinical expectations (Ramsay et al., 2003; Tan et al., 2024). For instance, GAMs showed limited clinical interpretability when applied to the Glasgow Coma Score (GCS), as correlations between its components (e.g., verbal, eye response) can distort the model's

explanations (Hegselmann et al., 2020). In addition, GAMs do not provide subgroup-level interpretability, an essential feature for identifying clinically meaningful heterogeneity and informing treatment guidelines (Bhavnani et al., 2022).

To overcome these challenges, we introduce Deep Additive Hazard Mixtures (ADHAM), which combines additive hazard modeling with mixture density networks. ADHAM represents the hazard function as a mixture of subgroup-specific hazard functions. Unlike former additive models, which assume a uniform sum of covariate effects (Jiang, 2022; Peroni et al., 2022; Xu and Guo, 2023), ADHAM learns a weighted combination to combine the hazard associated with each additive component. Importantly, ADHAM decouples the learning of hazard functions from subgroup assignment weights and trains each additive term separately to break the pairwise covariate correlations to address the concurvity problem. This leads to population-level hazard shapes that purely capture the covariate-specific trends in data.

A key challenge in mixture models is how to select the number of subgroups a priori. The true number of latent subgroups is typically unknown, which leads practitioners to often train multiple models under varying numbers of groups. We address this with a principled, post-training model selection strategy. This allows ADHAM to adapt its subgroup size *a posteriori*, and eliminates the need to train multiple models. Our results show that ADHAM achieves performance comparable to state-of-the-art additive models while offering improved interpretability.

Our contributions are as follows:

1. **Multi-level interpretability:** ADHAM provides explanations at the population, subgroup, and individual-levels within a single survival modeling framework. To the best of our knowledge, it is the only survival analysis model that can provide all three levels of explanation at once (Section 3.2).

2. **Overcoming concurvity through decoupled training:** We propose a training strategy for ADHAM that separates the learning of covariate-specific hazard functions from subgroup assignment. This approach mitigates concurvity by ensuring each hazard component is learned independently. Empirically, this leads to stable and reliable interpretability across training runs (Section 3.3).

3. **Efficient model selection:** ADHAM includes a principled *post-training* method to estimate and reduce redundant subgroups. To the best of our knowledge, ADHAM is the only survival analysis method that avoids training multiple models and allows for model selection post-training (Section 3.4).

4. **Competitive predictive performance:** ADHAM achieves predictive performance comparable to state-of-the-art interpretable survival models across multiple benchmarks (Section 5).

### Generalizable Insights about Machine Learning in the Context of Healthcare

In healthcare, ML models should not only exhibit strong predictive performance but also interpretability. This is especially true in survival analysis, where understanding the risk

at a given time is as important as surfacing associated predictors with fidelity. Clinicians must see how the risk evolves, how different covariates contribute to the risk, and how these patterns vary across the population, within subgroups, and for individual patients. Most ML models fall short in these areas, either acting as black-boxes, tackling a subset of these interpretability levels, or offering explanations that break down when the covariates are correlated. Additionally, models that attempt to identify subgroups typically require the true number of subgroups to be specified in advance, which introduces a challenging model selection problem —often requiring training multiple models, which places an extra burden on practitioners. To address these issues, we introduce ADHAM, a survival analysis model that enables clear explanations at multiple levels, breaks covariate correlation during training to prevent concurvity, and facilitates easy post-training model selection.

## 2. Related Work

**Time-to-event models.** Traditional non-parametric survival analysis techniques such as Kaplan-Meier and Nelson-Aalen estimators model population-level survival functions (Kaplan and Meier, 1958; Nelson, 1969; Aalen, 1978). These methods do not consider individual patient covariates and thus do not provide insights on the impact of exposure on outcome (Haider et al., 2020). To address this limitation, the foundational semi-parametric Cox Proportional Hazards (CoxPH) (Cox, 1972) models the shift induced by individual covariates on a non-parametric population survival. The estimation of the impact of covariates on outcomes offers tailored risk assessments. DeepSurv (Katzman et al., 2018) extends this method by replacing the parametric relation between covariates and outcomes with a neural network, resulting in improved predictive performance.

However, to obtain a closed-form likelihood, these approaches rely on the assumption, known as the proportional hazards (PH) assumption (Grambsch and Therneau, 1994; Hess, 1995), that the hazard function of any patient (1) differs from the population average by a constant factor and (2) follows the same trajectory through time. Avoiding this assumption, Cox-Time uses time-dependent conditional neural hazard functions (Kvamme et al., 2019). Despite its flexibility, this approach introduces bias in parameter estimation. Recently, Deep Hazard Analysis (DHA) introduced an unbiased estimator of the likelihood, allowing unbiased and flexible non-PH survival estimation (Ketenci et al., 2023).

In another attempt to avoid the PH assumption, previous works aimed to estimate the time-to-event distribution instead of the hazard function. For instance, DeepHit approaches survival as a classification approach by learning probability mass functions over discrete time intervals (Lee et al., 2018). This discretisation may, however, result in inexact likelihood estimates. Alternatively, recent works have proposed monotonic neural networks for estimating this distribution without approximation or discretization of the likelihood (Rindt et al., 2022; Jeanselme et al., 2023).

Closer to our work, Mixture Density Networks (MDNs) estimate the complete time-to-event distribution as a mixture of base distributions. MDNs define a mixture of flexible probability density functions where both the density and mixture weights are modeled by neural networks. Given enough mixture components, MDNs can model any probability density (Bishop and Nasrabadi, 2006). Recent approaches to MDNs use neural networks for group and time-to-event distributions. For example, Deep Survival Machines (DSM)

uses a mixture of parametric distributions (such as Log-normal and Weibull) (Nagpal et al., 2021a). Survival Mixture Density Networks (Survival MDNs) assume a mixture of Gaussian distributions and maximize the associated likelihood after marginalizing latent subgroup assignments (Han et al., 2022).

**Interpretable time-to-event models.** Existing interpretable survival models for clinical risk prediction integrate CoxPH's framework with Generalized Additive Models (GAMs). For example, CoxNAM and TimeNAM use Neural Additive Models (NAMs) to model the hazard rate in Cox-Time (Kvamme et al., 2019; Agarwal et al., 2021; Jiang, 2022; Utkin et al., 2022; Peroni et al., 2022; Xu and Guo, 2023). These models offer interpretability by defining a separate per-covariate function. This allows clinicians to visualize how individual covariates contribute to risk at different time points.

Other lines of work utilize post-hoc interpretability tools. As an example, the SurvLIME explanation method estimates the cumulative hazard function of a black-box model by fitting CoxPH's regression model (Kovalev et al., 2020). SurvSHAP(t) uses KernelSHAP to decompose the survival function to its Shapley values at each time step $t$ (Lundberg and Lee, 2017; Krzyziński et al., 2023). While effective, these ad-hoc interpretability methods become impractical for large datasets and extended time horizons, as they must be re-executed for every covariate-time combination.

## 3. ADHAM: Additive Deep Hazard Analysis Mixtures

### 3.1. Background: Generalized Additive Models

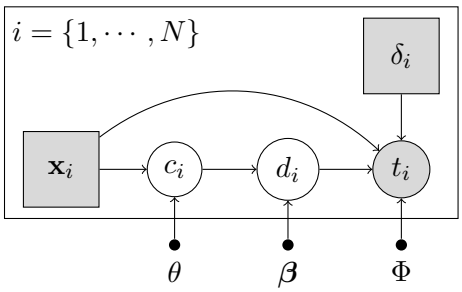
GAMs are a flexible class of statistical models that extend generalized linear models by allowing for non-linear relationships between the covariates, denoted by a $D$-dimensional vector $\mathbf{x} \in \mathbb{R}^D$, and the outcome. A $k^{\text{th}}$ order GAM, $g$, is defined as:

$$g(\mathbf{x}) = \sum_{u \subseteq [D] || u| \leq k} g_u(\mathbf{x}_u), \tag{1}$$

where $[D] = \{1, 2, \cdots, D\}$. Each $g_u(\mathbf{x}_u)$ denotes the contribution of a covariate subset $u$ to the overall prediction. In this work, we adopt the additive decomposition in Equation 1 to model the hazard function considering $k = 1$. Unlike traditional GAMs, however, we introduce a weighted sum formulation in which the influence of each component $g_u$ is modulated by patient-specific subgroup assignments, thereby enabling subgroup-specific additive effects.

### 3.2. Model

Consider a time-to-event dataset with $N$ points, $\mathcal{D} = \{(\mathbf{x}_i, t_i, \delta_i)\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^D$ denotes the vector of covariates associated with patient $i$, $t_i \in \mathbb{R}^+$ is the recorded event or censoring time, and $\delta_i \in \{0, 1\}$ indicates whether the event occurred ($\delta_i = 1$) or was right-censored ($\delta_i = 0$). With these notations, we assume a latent subgroup membership $c$ dependent on covariates $\mathbf{x}$, where covariate-weights are group-specific. We describe the data-generating process in Figure 1:




For $i = \{1, \cdots, N\}$:

- Draw time-to-event $t_i \sim p(t \mid \mathbf{x}_i, \delta_i; \theta, \boldsymbol{\beta}, \Phi)$ with marginal hazard function $\sum_{c=1}^{C} \sum_{d=1}^{D} p(d \mid c; \boldsymbol{\beta}) p(c \mid \mathbf{x}_i; \theta) \lambda(t \mid x_{id}; \phi_d)$ where, $c \mid \mathbf{x}_i \sim \text{Cat}\left(f_\theta(\mathbf{x}_i)\right)$, $d \mid c \sim \text{Cat}\left(\boldsymbol{\beta}_c\right)$, and $\Phi = \{\phi_d\}_{d=1}^{D}$

Figure 1: Plate notation and data generating process of ADHAM. $f_\theta$ is subgroup assignment network, $\beta_c$ are subgroup-specific feature importance values, and $\lambda(t \mid x_{id}; \phi_d)$ is the population-level hazard curve.

The subgroup assignment network $f_\theta(\mathbf{x})$ and associated weight vector $\boldsymbol{\beta}_c$ lie on $C$- and $D$-dimensional simplexes, respectively[2]. Each patient is assigned to a latent subgroup $c$, with probability $f_{\theta_c}(\mathbf{x})$ characterized by weights on each covariate-specific hazard, $\beta_{dc}$. Marginal hazard is then defined as the group-specific weighted sum of covariate-specific hazard functions. We summarize all notations with a table in Appendix A and illustrate the methodology with a flow chart in Appendix B.

In the following paragraphs, we describe how each term in ADHAM's marginal hazard function links to different levels of interpretability.

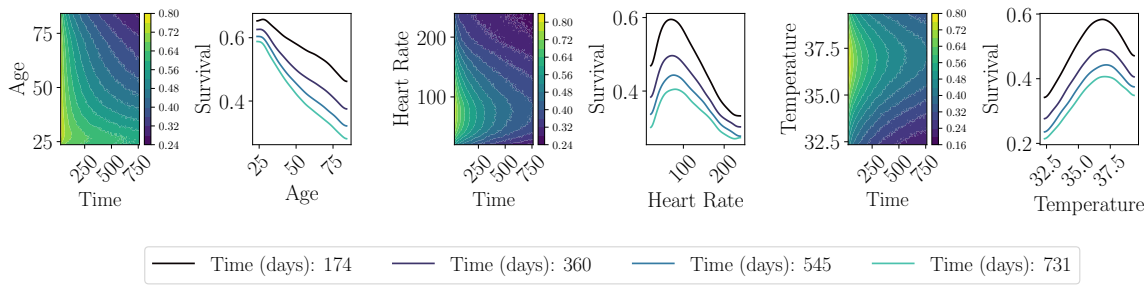



Figure 2: Covariate-specific *population-level* survival functions, $\lambda(t \mid x_{id}; \phi_d)$, of ADHAM trained on **SUPPORT** dataset. ADHAM captures well-known physiological trends in heart rate and temperature. In particular, normal ranges (e.g., 36 - 37.5 °C for temperature and 60 - 100 bpm for heart rate (Tan et al., 2024)) have better survival odds. Similarly, the survival probability decreases with age. In Appendix E, we compare ADHAM's population-level survival functions to those of TimeNAM's and demonstrate that our results are consistent.

**Multi-level interpretability.** We start with the *population-level* hazard, defined as:

$$\lambda(t \mid x_d; \phi_d). \tag{2}$$

This quantity represents the shared hazard as a function of the covariate $d$. It is the same across the population for the same $x_d$ values. One can either use this quantity directly or calculate the population-level survival function using it. In Figure 6, we illustrate

2. This is ensured by softmax function.

population-level survival functions, derived using Equation (2), as also discussed in Appendix D.

We weigh each hazard function using subgroup-level weights $\boldsymbol{\beta} \in [0,1]^{C\times D}$, where $\sum_d p(d \mid c;\boldsymbol{\beta}) = \sum_d \beta_{dc} = 1$. Note that, each $\boldsymbol{\beta}_c$ vector informs about *subgroup-level* covariate importances (e.g., if covariate $d$ is important for subgroup $c$, then $\beta_{dc}$ is high):

$$p(d \mid c;\boldsymbol{\beta})\lambda(t \mid x_d;\phi_d) = \beta_{dc}\lambda(t \mid x_d;\phi_d). \tag{3}$$

We use $\boldsymbol{\beta}$ for *subgroup-level* interpretability and illustrate it, in Figure 3, along with subgroup statistics.

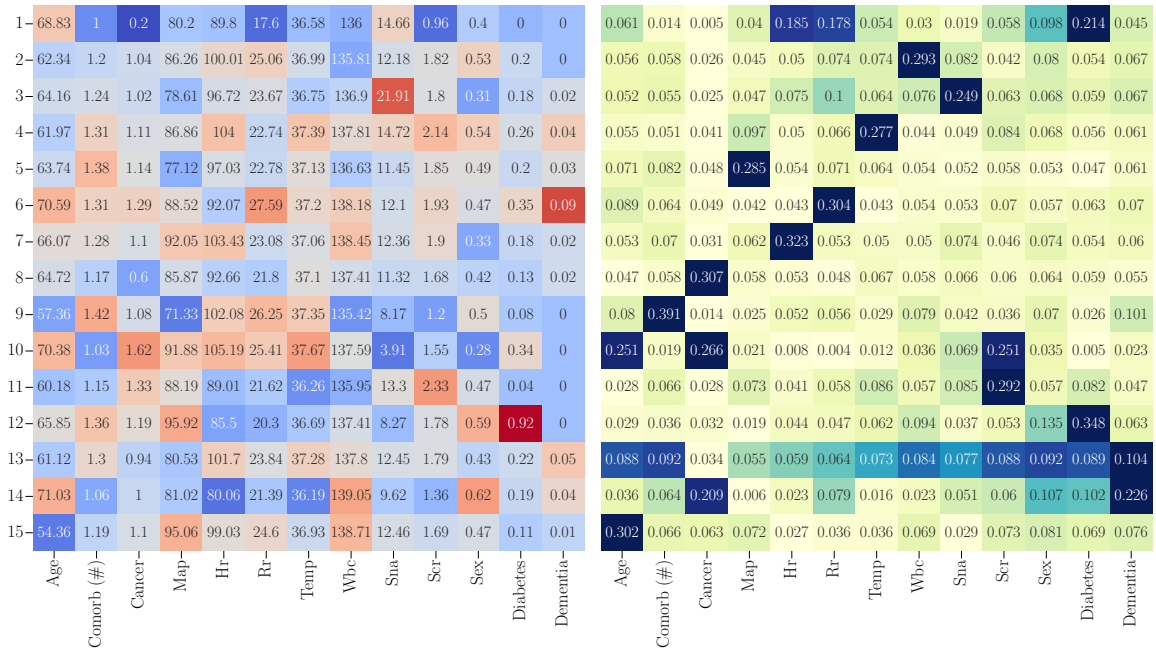



Figure 3: *Subgroup-level* interpretability of **SUPPORT** dataset. Subgroup-specific average measurements and corresponding $\beta_{dc}$ values are provided in **left** and **right**, respectively. Each row describes a subgroup. In the left heatmap, warmer colors indicate values above population average, while cooler colors reflect below-average measurements, with each cell including the corresponding numeric value. See Appendix F.2 for covariate details. In the right heatmap, darker shades denote higher covariate importance, indicating which covariates drive the hazard in each subgroup. ADHAM consistently identifies covariates that deviate from normal ranges via $\boldsymbol{\beta}$ matrix. For instance, Subgroup 1 shows low heart and respiratory rates, both heavily weighted. Subgroup 2 is marked by low WBC and elevated heart and respiratory rates. Subgroup 4 is dominated by high temperature, while Subgroup 10 highlights age and cancer. Overall, ADHAM effectively identifies clinically relevant risk factors across subgroups.

For a given patient $\mathbf{x}$, we compute its group assignment via $p(c \mid \mathbf{x};\theta)$ and define the marginal hazard rate as a sum over *individual-level* hazard functions, each describing a contribution to the marginal patient hazard:

$$\lambda(t \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \Phi) = \sum_{d=1}^{D} \sum_{c=1}^{C} p(d \mid c; \boldsymbol{\beta}) p(c \mid \mathbf{x}; \theta) \lambda(t \mid x_d; \phi_d) \tag{4}$$

$$= \sum_{d=1}^{D} \underbrace{p(d \mid \mathbf{x}; \theta, \beta) \lambda(t \mid x_d; \phi_d)}_{\textit{individual-level hazard function}} \tag{5}$$

$$= \sum_{d=1}^{D} \lambda(t \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \phi_d), \tag{6}$$

where $p(d \mid \mathbf{x}; \theta, \beta) = \sum_{c=1}^{C} p(d \mid c; \boldsymbol{\beta}) p(c \mid \mathbf{x}; \theta) = \sum_{c=1}^{C} \beta_{dc} f_{\theta_c}(\mathbf{x})$ is a function of all covariates $\mathbf{x}$ and introduces patient-specific covariate interactions into the picture. Note that, $\sum_{c=1}^{C} f_{\theta_c}(\mathbf{x}) = 1$. We use $\lambda(t \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \phi_d)$ for *individual-level* interpretability, as demonstrated in Figure 4.

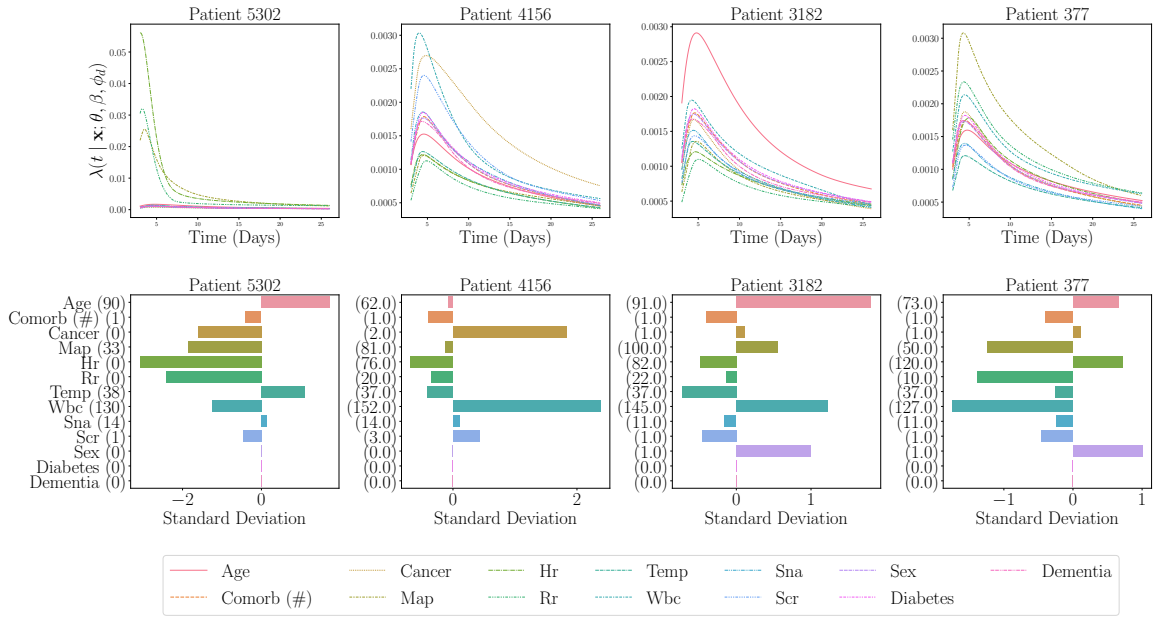



Figure 4: *Individual-level* hazard functions and corresponding input values for four patients. For the first patient, ADHAM identifies an immediate risk driven by low heart rate (Hr), respiratory rate (Rr), and mean arterial pressure (Map), matching the patient's high short-term risk. For Patient 4156, the model captures both short-term risk from elevated white blood cell (Wbc) count—suggesting possible infection—and long-term risk from advanced-stage cancer. Patient 3182, an older individual with moderately abnormal covariates, shows a hazard profile dominated by age. For Patient 377, ADHAM points to low Map and high Wbc as the key contributors to their risk, which are outside of normal range.

In this section, we focused on the hazard function. Similarly, one can study the different levels of interpretability from the survival function perspective, as presented in Appendix D.

### 3.3. Parameter Estimation

Given the marginal patient hazard, the corresponding probability density is expressed as:

$$p(t \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \Phi) = \lambda(t \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \Phi) \exp\left\{ - \int_0^t \lambda(s \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \Phi), \mathrm{d}s \right\}, \tag{7}$$

and the corresponding log-likelihood over the dataset $\mathcal{D}$ is:

$$\ell(\mathbf{T}_N | \mathbf{X}_N, \Delta_N; \theta, \boldsymbol{\beta}, \Phi) = \log \prod_{i=1}^{N} \lambda(t_i \mid \mathbf{x}_i; \theta, \boldsymbol{\beta}, \Phi)^{\delta_i} \exp\left\{ - \int_0^{t_i} \lambda(s \mid \mathbf{x}_i; \theta, \boldsymbol{\beta}, \Phi) \, \mathrm{d}s \right\}, \tag{8}$$

where $\mathbf{T}_N = \{t_i\}_{i=1}^N$, $\mathbf{X}_N = \{\mathbf{x}_i\}_{i=1}^N$, and $\Delta_N = \{\delta_i\}_{i=1}^N$.

We use neural networks to model $f_\theta(\mathbf{x})$ and $\lambda(t \mid x_d, \phi_d)$. We ensure positive $\lambda(t \mid x_d, \phi_d)$ by applying the softplus(.) function. A computationally efficient and unbiased stochastic approximation to Equation (8), using $L$ data samples and $M$ importance samples over time-to-event outcomes, has been proposed by Ketenci et al. (2023):

$$\tilde{\ell}(\mathbf{T}_L | \mathbf{X}_L, \Delta_L, \tilde{\mathbf{T}}_{LM}; \theta, \boldsymbol{\beta}, \Phi) = \frac{N}{L} \sum_{i=1}^{L} \left( \delta_i \log \lambda(t_i \mid \mathbf{x}_i; \theta, \boldsymbol{\beta}, \Phi) - \frac{t_i}{M} \sum_{j=1}^{M} \lambda(\tilde{t}_{ij} \mid \mathbf{x}_i; \theta, \boldsymbol{\beta}, \Phi) \right),$$

where $\tilde{\mathbf{T}}_{LM} = \{\{\tilde{t}_{ij}\}_{j=1}^M\}_{i=1}^L$, and $\tilde{t}_{ij} \sim U(0, t_i)$. Unlike former numerical integration-based estimations, such as quadrature, this Monte Carlo (MC) estimate is unbiased (Gregoire and Valentine, 1995).

**Overcoming concurvity through decoupled training.** Empirically, we observe that jointly optimizing $\{\theta, \boldsymbol{\beta}, \Phi\}$ can hinder interpretability due to two key factors: (1) concurvity, where correlated covariates distort each other's contributions, leading to unreliable and unstable population-level hazard functions; and (2) ill-conditioning, where the flexibility of the subgroup assignment network allows individual-level hazard functions, $\lambda(t \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \phi_d)$, to behave as arbitrary, unconstrained functions over all covariates—particularly as the number of subgroups $C$ increases. Together, these effects would reduce the model's interpretability.

Our goal is to ensure that ADHAM remains interpretable and stable regardless of concurvity and choice of $C$. To mitigate this issue, we disentangle the learning of covariate-specific hazard functions from the identification of subgroups as described in Algorithm 1. In particular, we train each covariate-specific population-level hazard function and mixture assignment network separately by maximizing:

$$\tilde{\ell}_d(\mathbf{T}_L | \mathbf{X}_L, \Delta, \tilde{\mathbf{T}}_{LM}; \phi_d) = \frac{N}{L} \sum_{i=1}^{L} \left( \delta_i \log \lambda(t_i \mid x_{id}; \phi_d) - \frac{t_i}{M} \sum_{j=1}^{M} \lambda(\tilde{t}_{ij} \mid x_{id}; \phi_d) \right), \tag{9}$$

with respect to $\phi_d$, and $\tilde{\ell}(\mathbf{T}_L | \mathbf{X}_L, \Delta_L, \tilde{\mathbf{T}}_{LM}; \theta, \boldsymbol{\beta}, \Phi)$ with respect to $\{\theta, \beta\}$. Independently maximizing the population-level hazard log-likelihood $\tilde{\ell}_d$ for each $d \in \{1, 2, \cdots, D\}$

helps eliminate covariate correlation during training by ensuring that each population-level hazard function is learned in isolation. Subsequently, optimizing the overall log-likelihood $\tilde{\ell}(\mathbf{T}_L|\mathbf{X}_L, \Delta_L, \tilde{\mathbf{T}}_{LM}; \theta, \boldsymbol{\beta}, \Phi)$ with respect to $\theta, \boldsymbol{\beta}$ learns a distribution $p(d \mid \mathbf{x}; \theta, \boldsymbol{\beta})$ that re-weights the fixed hazard curves to explain the data likelihood best. This two-step approach prevents the mixing weights from influencing the population-level hazard functions, avoiding covariate-specific curves from degenerating into arbitrary functions of all covariates. The full training procedure is outlined in Algorithm 1.

**Algorithm 1** Mini-batch stochastic gradient descent learning of ADHAM parameters. We use overall model log-likelihood as our early stopping criteria.

**def** `fit_adham`$(\mathcal{D}, \theta, \boldsymbol{\beta}, \Phi)$:
  $\theta, \boldsymbol{\beta}, \Phi \leftarrow$ Initialize neural network parameters
  **while** $\tilde{\ell}(\mathbf{T}_L|\mathbf{X}_L, \Delta_L, \tilde{\mathbf{T}}_{LM}; \theta, \boldsymbol{\beta}, \Phi)$ has not converged **do**
    $\{\mathbf{X}_L, \mathbf{T}_L, \Delta_L\} \leftarrow$ Sample $L$ data from $\mathcal{D}$
    $\tilde{\mathbf{T}}_{LM} \leftarrow$ Sample $M$ importance samples from $U(0, \mathbf{T})$
    # 1. Fit Hazard Functions
    **for** $d = \{1, 2, \cdots, D\}$ **do**
      $g_d \leftarrow \nabla_{\phi_d} \tilde{\ell}_d(\mathbf{T}_L|\mathbf{X}_L, \Delta, \tilde{\mathbf{T}}_{LM}; \phi_d)$
    **end for**
    $\Phi \leftarrow$ Update using gradients $\{g_d\}_{d=1}^{D}$
    # 2. Fit Mixture Components
    $g \leftarrow \nabla_{\theta,\beta} \tilde{\ell}(\mathbf{T}_L|\mathbf{X}_L, \Delta_L, \tilde{\mathbf{T}}_{LM}; \theta, \boldsymbol{\beta}, \Phi)$
    $\theta, \beta \leftarrow$ Update using gradients $g$
  **end while**
  **Output:** $\theta, \boldsymbol{\beta}, \Phi$

**Regularization of ADHAM.** To (i) encourage sparsity in subgroup assignments and (ii) promote broader exploration of covariate relevance during training, we subtract two regularization terms from the objective function $\tilde{l}$. While sparsity in covariate weights may eventually reflect meaningful signals, regularization ensures that this structure emerges through learning rather than an early optimization bias:

$$\begin{aligned}&\tilde{\mathcal{R}}(\mathbf{X}_L; \theta, \beta)\\ &= \frac{1}{L(L-1)} \sum_{i=1}^{L} \sum_{\substack{j=1 \\ j \neq i}}^{L} \sum_{c=1}^{C} p(c \mid \mathbf{x}_i; \theta) p(c \mid \mathbf{x}_j; \theta) + \frac{1}{L} \sum_{i=1}^{L} \sum_{d=1}^{D} p(d \mid \mathbf{x}_i; \theta, \beta) \log p(d \mid \mathbf{x}_i; \theta, \beta).\end{aligned} \tag{10}$$

The first term is known as an orthogonal output regularization (Brock et al., 2016; Bansal et al., 2018), which promotes diversity in subgroup assignments by encouraging the model to assign different data points to distinct subgroups with high confidence. The second term is known as an entropy-regularization term (Mnih et al., 2016; Tang et al., 2023), which encourages ADHAM to consider a wider set of covariates during training, *helping to avoid*

*premature convergence* to narrow, locally optimal solutions. We conduct experiments both with and without incorporating regularization terms into the ADHAM objective, to study the effect empirically, as discussed in Section 5.

### 3.4. Model Selection via Subgroup Refinement

Determining the optimal number of heterogeneous subgroups is often difficult and usually involves training and evaluating several models. ADHAM circumvents this challenge through a post-training refinement strategy that takes advantage of two key properties: (1) each population-level hazard function is tied to a single covariate $x_d$, and (2) its contribution is scaled by a subgroup-specific constant $\beta_{dc}$.

**Algorithm 2** Pseudo-algorithm for model selection given the subgroup groups $\mathcal{C}$, covariate importance matrix $\boldsymbol{\beta}$, and a threshold, $h$. In practice, we use efficient implementations provided by `fcluster` and `linkage` modules of SciPY (Virtanen et al., 2020).

**def** `combine_clusters`$(\mathcal{C}, \boldsymbol{\beta}, h)$:
  $\mathcal{C}^\star \leftarrow$ Initialize empty set $\{\}$
  $\rho \leftarrow$ Initialize correlation matrix $\frac{\beta\beta^\top}{\sqrt{(\mathrm{tr}\{\beta\beta^\top\})(\mathrm{tr}\{\beta\beta^\top\})^\top}}$ # $C \times C$ correlation matrix
  **while** $\max \rho \geq h$ **do** # While there are no correlated subgroups left
    **for** $c \in \mathcal{C}$ **do**
      **if** $\max \rho_c > h$ **then**
        $c^\star \leftarrow \mathrm{argmax}\, \rho_c$ # $\rho_c$ is $C \times 1$ row vector
        $\mathcal{C}^\star \leftarrow$ Combine groups $\mathcal{C}^\star \cup \{c, c^\star\}$ # $\mathcal{C}^\star$ is a set of (2 cardinality) sets
        $\rho_{cc^\star} \leftarrow$ Update entry to $-\infty$ # So that **while** does not run forever
      **end if**
    **end for**
  **end while**
  **while** $\bigcap_{c^\star \in \mathcal{C}^\star} c^\star \neq \{\}$ **do** # While there are groups to be merged
    **for** $\{c_1^\star, c_2^\star\} \in \mathcal{C}^\star \times \mathcal{C}^\star$ **do**
      **if** $c_1^\star \cap c_2^\star \neq \{\}$ **then** # If there are common elements then merge
        $\mathcal{C}^\star \leftarrow \mathcal{C}^\star \setminus c_1^\star$ # Subtract $c_1^\star$
        $\mathcal{C}^\star \leftarrow \mathcal{C}^\star \setminus c_2^\star$ # Subtract $c_2^\star$
        $\mathcal{C}^\star \leftarrow \mathcal{C}^\star \cup \{c_1^\star \cup c_2^\star\}$ # Add their union back
      **end if**
    **end for**
  **end while**
  **Output:** $\mathcal{C}^\star$ # Refined subgroups

As such, if two subgroups $c_1$ and $c_2$ have identical covariate importance vectors, i.e., $\boldsymbol{\beta}_{c_1} = \boldsymbol{\beta}_{c_2}$, they can be combined without affecting the model's data log-likelihood and overall performance. A formal proof is provided in Appendix C.1. Note that, this property is not a byproduct of the parameter estimation procedure but the model design. In practice, we evaluate the similarity between subgroups by calculating their pairwise correlation: $\rho_{c_1 c_2} = \frac{\sum_{d=1}^{D} \beta_{c_1 d}\beta_{c_2 d}}{\sqrt{\left(\sum_{d=1}^{D} \beta_{c_1 d}^2\right)\left(\sum_{d=1}^{D} \beta_{c_2 d}^2\right)}}$. subgroups with correlation above a predefined threshold $h$ are

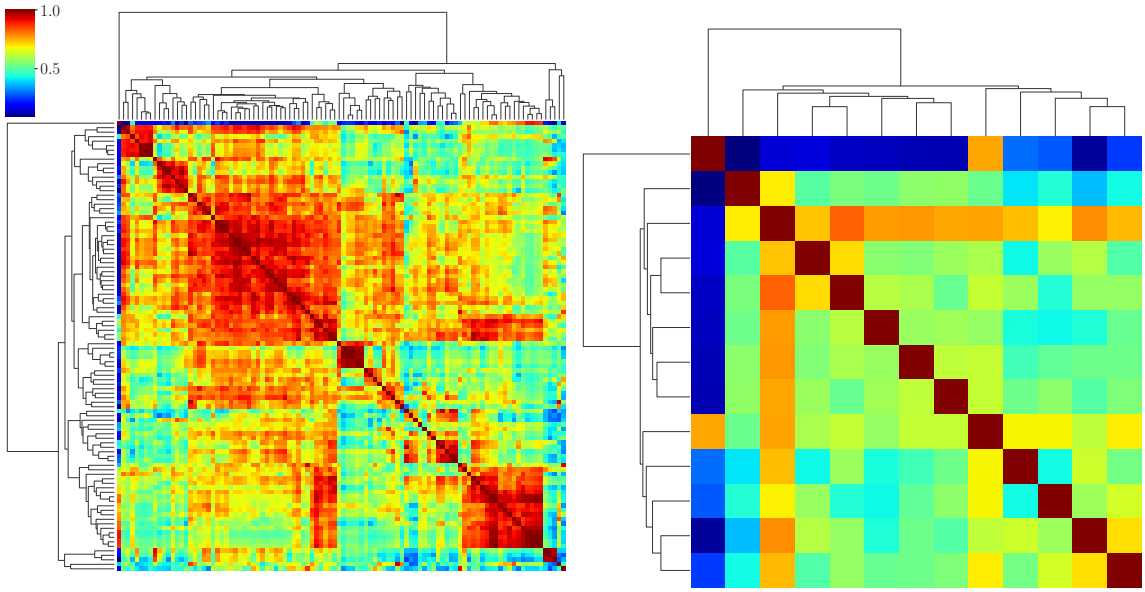



Figure 5: Example correlation matrix $\rho$ before (left) and after (right) subgroup refinement on the **SUPPORT** dataset. Rows and columns are ordered using `sns.clustermap` to group similar subgroups. Each cell shows the correlation between two groups (e.g., $\rho_{c_1 c_2}$), with warmer colors indicating higher similarity. Before refinement, many subgroups have overlapping covariate importance patterns, suggesting repetition. After refinement, correlations decrease and subgroup profiles become more distinct, each emphasizing different sets of covariates.

then merged using a bottom-up agglomerative strategy (Müllner, 2011), as described in Algorithm 2. We refer to this procedure as *subgroup refinement*. We illustrate an example correlation matrix $\rho$ in Figure 5. In practice, ADHAM can initially be trained with a large number of subgroups $C$, which can later be reduced through this refinement process if needed. We demonstrate an example on **SUPPORT** dataset in Table 2.

### 3.5. Predictions

Making survival predictions with ADHAM requires calculating the following estimate:

$$S(t|\mathbf{x};\theta,\boldsymbol{\beta},\Phi) = \exp\left\{-\int_0^t \lambda(s \mid \mathbf{x};\theta,\boldsymbol{\beta},\Phi),\mathrm{d}s\right\} \tag{11}$$

This integral can be calculated by Monte Carlo samples, or numerical integration (Ketenci et al., 2023; Kvamme et al., 2019). In our experiments, we use the estimation method proposed by Ketenci et al. (2023).

## 4. Experimental Setup

### 4.1. Datasets

We experiment on two standard benchmark datasets, **SUPPORT** and **FLCHAIN**, widely used in survival analysis, and a real-world clinical dataset of patients with chronic kidney disease (**CKD**), tracking time to acute kidney injury based on electronic health records. Please see Appendix F.1 for a detailed description of the datasets.

### 4.2. Baseline Models

We consider ten well-established baselines for survival analysis and state-of-the-art models: CoxPH (Cox, 1972), DeepSurv (Katzman et al., 2018), RSF (Ishwaran et al., 2008), Cox-Time (Kvamme et al., 2019), TimeNAM & TimeNA2M (Peroni et al., 2022), DSM (Nagpal et al., 2021a), DCM (Nagpal et al., 2021b), DeepHit (Lee et al., 2018), and DHA (Ketenci et al., 2023). Please see Appendix F.3 for a description of these models.

### 4.3. Evaluation Metrics

We follow the same evaluation setup as Li et al. (2023); Nagpal et al. (2022, 2021a,b); Jeanselme et al. (2022, 2023); Wang and Sun (2022); Lee et al. (2019); Ketenci et al. (2023) and compare the average C-Index, Brier Score, and AUROC statistics over different time horizons to assess each model's predictive performance. We assess the performance of ADHAM and baseline models using both discrimination and calibration performance metrics across three time horizons, as often used at deployment. The details of evaluation metrics are described in Appendix F.5.

### 4.4. Empirical Setup

We perform 5-fold cross-validation across all models and datasets. At each training step, we leave 20% of the data out and divide the remaining data into training and validation sets by 70% and 30%, respectively. For a fair comparison, we use fixed random seeds. This ensures that the training, validation, and test sets seen by ADHAM and baseline models are identical. We standardize the datasets by subtracting the mean and dividing by the standard deviation of the covariates. We use Adam optimizer for neural models (Kingma and Ba, 2014). Each model is trained for 4000 epochs. The best model is saved based on the validation loss during training (early stopping), and evaluations are done over the held-out test set, which the models do not see during training. Hyperparameter settings are available at Appendix F.4.

## 5. Results

**Interpretability of ADHAM.** We illustrate ADHAM's *population-level interpretability* on the **SUPPORT** dataset in Figure 6, showing that it reliably captures meaningful trends for key measurements. These patterns are a useful tool to understand and debug models (Caruana et al., 2015).

| Models | SUPPORT Dataset | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $25^{\text{th}}$ Quantile | | | $50^{\text{th}}$ Quantile | | | $75^{\text{th}}$ Quantile | | |
| | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ |
| ADHAM ($\mathcal{R}$, $h = 1$, $C = 100$) | 0.660 | 0.144 | 0.666 | 0.630 | 0.222 | 0.644 | 0.620 | 0.247 | 0.658 |
| ADHAM ($\mathcal{R}$, $h = 0.8$, $C = [24, 40]$) | 0.660 | 0.145 | 0.666 | 0.629 | 0.223 | 0.644 | 0.620 | 0.247 | 0.656 |
| ADHAM ($\mathcal{R}$, $h = 0.75$, $C = [17, 30]$) | 0.660 | 0.145 | 0.666 | 0.629 | 0.223 | 0.644 | 0.619 | 0.247 | 0.656 |
| ADHAM ($\mathcal{R}$, $h = 0.7$, $C = [14, 23]$) | 0.660 | 0.145 | 0.666 | 0.629 | 0.223 | 0.644 | 0.619 | 0.247 | 0.656 |
| ADHAM ($\mathcal{R}$, $h = 0.65$, $C = [11, 19]$) | 0.659 | 0.145 | 0.665 | 0.629 | 0.223 | 0.644 | 0.619 | 0.247 | 0.656 |
| ADHAM | 0.613 | 0.142 | 0.616 | 0.588 | 0.219 | 0.602 | 0.588 | 0.237 | 0.639 |

Table 1: Performance of ADHAM on the SUPPORT dataset as a function of the subgroup refinement threshold $h$. Larger values of $h$ impose stricter criteria for subgroup merging—that is, merging occurs only when the corresponding $\beta_c$ parameters are identical. We observe that model performance remains largely stable across varying values of $h$, indicating robustness to the choice of refinement threshold.

| Models | **SUPPORT** Dataset | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $25^{\text{th}}$ Quantile | | | $50^{\text{th}}$ Quantile | | | $75^{\text{th}}$ Quantile | | |
| | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ |
| DeepSurv | 0.605 | **0.143** | 0.608 | 0.598 | **0.217** | 0.618 | 0.609 | **0.232** | **0.659** |
| RSF | **0.662** | **0.139** | **0.668** | **0.624** | **0.214** | **0.641** | **0.614** | **0.230**$^\star$ | **0.668**$^\star$ |
| DeepHit | **0.652** | **0.141** | **0.659** | 0.613 | 0.220 | **0.633** | 0.605 | **0.236** | **0.661** |
| Cox-Time | 0.625 | **0.141** | 0.631 | **0.614** | **0.214** | **0.635** | **0.616** | **0.230**$^\star$ | **0.663** |
| DSM | 0.633 | **0.141** | 0.639 | **0.621** | **0.215** | **0.638** | 0.605 | 0.254 | 0.653 |
| DCM | **0.657** | **0.138**$^\star$ | **0.663** | **0.620** | **0.213** | **0.638** | 0.603 | **0.234** | 0.652 |
| DHA | **0.663**$^\star$ | **0.138**$^\star$ | **0.672**$^\star$ | **0.631**$^\star$ | **0.211**$^\star$ | **0.650**$^\star$ | **0.615** | **0.231** | **0.663** |
| CoxPH | 0.553 | 0.262 | 0.558 | 0.567 | 0.222 | 0.588 | 0.590 | 0.351 | 0.649 |
| TIMENAM | 0.621 | **0.142** | 0.627 | 0.612 | **0.215** | 0.633 | **0.615** | **0.230**$^\star$ | **0.666** |
| TIMENA2M | **0.643** | **0.139** | 0.650 | **0.618** | **0.216** | **0.636** | **0.610** | 0.238 | 0.654 |
| **ADHAM** ($\mathcal{R}$) | **0.660** | **0.144** | **0.666** | **0.630** | 0.222 | **0.644** | **0.620**$^\star$ | 0.247 | **0.658** |
| **ADHAM** | 0.613 | **0.142** | 0.616 | 0.588 | 0.219 | 0.602 | 0.588 | 0.237 | 0.639 |

Table 2: Results on the **SUPPORT** dataset. The models in gray region are interpretable. Best values are denoted by $^\star$, and results that are statistically close to the best values are shown in **bold**. Best performing interpretable values are underlined. The standard error of the sample mean (SEM) values are provided in Table 6.

| Models | **CKD** Dataset | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $25^{\text{th}}$ Quantile | | | $50^{\text{th}}$ Quantile | | | $75^{\text{th}}$ Quantile | | |
| | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ |
| DeepSurv | **0.620** | **0.089** | 0.635 | 0.611 | **0.164** | 0.653 | **0.608** | **0.217** | **0.684** |
| RSF | **0.628** | **0.088**$^\star$ | **0.647** | **0.621** | **0.164** | **0.665** | **0.611** | **0.216**$^\star$ | **0.701** |
| DeepHit | **0.642**$^\star$ | **0.088**$^\star$ | **0.658**$^\star$ | **0.630**$^\star$ | **0.161**$^\star$ | **0.673**$^\star$ | **0.617**$^\star$ | **0.216**$^\star$ | **0.698** |
| Cox-Time | **0.615** | **0.090** | 0.626 | 0.611 | **0.165** | 0.651 | **0.609** | **0.219** | **0.684** |
| DSM | **0.630** | **0.091** | **0.644** | **0.619** | 0.182 | **0.669** | **0.607** | 0.244 | **0.702**$^\star$ |
| DHA | **0.623** | **0.089** | **0.636** | **0.609** | **0.164** | **0.651** | **0.606** | **0.217** | **0.690** |
| CoxPH | 0.593 | **0.089** | 0.608 | 0.582 | **0.168** | 0.615 | 0.579 | **0.224** | 0.658 |
| TIMENAM | **0.622** | **0.089** | **0.639** | **0.610** | **0.168** | **0.651** | **0.597** | **0.217** | **0.688** |
| TIMENA2M | **0.625** | 0.099 | **0.639** | 0.612 | 0.192 | 0.646 | 0.591 | 0.261 | 0.647 |
| **ADHAM** ($\mathcal{R}$) | **0.620** | **0.090** | **0.632** | **0.618** | **0.170** | **0.652** | **0.607** | 0.236 | **0.682** |
| **ADHAM** | 0.603 | **0.089** | 0.615 | 0.597 | **0.167** | 0.630 | 0.592 | **0.226** | 0.657 |

Table 3: Results on the **CKD** dataset. The models in the gray region are interpretable. Best values are denoted by $^\star$, and results that are statistically close to the best values are shown in **bold**. Best performing interpretable values are underlined. The standard error of the sample mean (SEM) values are provided in Table 7.

| Models | **FLCHAIN** Dataset | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $25^{\text{th}}$ Quantile | | | $50^{\text{th}}$ Quantile | | | $75^{\text{th}}$ Quantile | | |
| | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ |
| DeepSurv | **0.788** | **0.060** | **0.799** | **0.792** | **0.100** | **0.815** | **0.788** | **0.126** | **0.823** |
| RSF | **0.801** | **0.059** | **0.812** | **0.798** | **0.099** | **0.821** | **0.793** | **0.124** | **0.828** |
| DeepHit | **0.794** | **0.061** | **0.805** | **0.795** | **0.101** | **0.818** | **0.792** | **0.127** | **0.827** |
| Cox-Time | **0.798** | **0.065** | **0.810** | **0.797** | 0.118 | **0.820** | **0.793** | 0.160 | **0.828** |
| DSM | 0.760 | 0.065 | 0.770 | 0.766 | 0.118 | 0.786 | 0.768 | 0.159 | 0.800 |
| DCM | **0.790** | **0.059** | **0.801** | **0.788** | **0.100** | 0.810 | **0.782** | **0.128** | 0.814 |
| DHA | **0.801** | **0.058**$^\star$ | **0.813** | **0.800**$^\star$ | **0.097**$^\star$ | **0.824**$^\star$ | **0.795**$^\star$ | **0.122**$^\star$ | **0.830**$^\star$ |
| CoxPH | **0.789** | 0.103 | **0.801** | **0.794** | 0.098 | **0.817** | **0.791** | 0.168 | **0.826** |
| TIMENAM | **0.796** | **0.060** | **0.807** | **0.796** | **0.104** | **0.818** | **0.791** | **0.134** | **0.826** |
| TIMENA2M | **0.802**$^\star$ | 0.065 | **0.814**$^\star$ | **0.797** | 0.118 | **0.819** | **0.791** | 0.160 | **0.828** |
| **ADHAM** ($\mathcal{R}$) | **0.798** | **0.064** | **0.809** | **0.795** | 0.114 | **0.818** | **0.791** | 0.154 | **0.825** |
| **ADHAM** | 0.776 | **0.060** | 0.786 | 0.782 | **0.102** | 0.804 | 0.779 | 0.135 | 0.810 |

Table 4: Results on the **FLCHAIN** dataset. The models in the gray region are interpretable. Best values are denoted by $^\star$, and results that are statistically close to the best values are shown in **bold**. Best performing interpretable values are underlined.The standard error of the sample mean (SEM) values are provided in Table 8.

Figure 3 illustrates ADHAM's *subgroup-level interpretability*, by uncovering heterogeneous patient subgroups along with their associated covariate importance profiles. We observe that ADHAM groups patients based on abnormal measurement patterns, which are captured through the subgroup-specific weight matrix $\boldsymbol{\beta}$. In Figure 4, we present patient-specific hazard functions, where we once again observe that ADHAM assigns importance to most critical covariates.

**Performance of ADHAM.** We present performance results in Tables 2, 3, and 4. The best average results are denoted by $^\star$, values that are statistically close to the best results (with $p = 0.05$) with respect to two-sided Welch's t-test are denoted with **bold**[3], and best interpretable methods are underlined. DHA achieves the best performance across most evaluation metrics, though it functions as a black-box model. Among the interpretable approaches—CoxPH, TIMENAM, TIMENA2M, and ADHAM—ADHAM with regularization ranks highest in 9 metrics, matching TIMENA2M, followed by TIMENAM with 8 and CoxPH with 2, across various datasets. We also find that regularization benefits ADHAM, particularly in ranking-based metrics, albeit with a minor trade-off in calibration accuracy. Furthermore, in Table 1—on the **SUPPORT** dataset, we observe that ADHAM's model selection strategy, where we alter $h = 1$ to $h = 0.65$, results in a minor performance loss while substantially reducing the number of subgroups (e.g., from 100 to 11). Overall, ADHAM demonstrates similar performance to the existing state-of-the-art interpretable survival models. To evaluate statistical significance, the standard errors are provided in Appendix G.

3. Note that the difference in two results may be statistically insignificant due to (1) the mean over fold and (2) the (corrected) sample standard deviation of runs.

## 6. Discussion and Limitations

ADHAM offers a unified framework for interpretability, delivering individualized, subgroup-level, and population-wide risk explanations. Our results show that ADHAM performs on par with leading interpretable survival models. Nonetheless, several limitations remain.

**Causal interpretation.** ADHAM is a predictive, not a causal model. While the proposed tool provides insights into the learned relation between covariates and survival outcomes, practitioners should not interpret these observational correlations as causation.

**Sensitivity to dataset biases.** As with any data-driven model, ADHAM reflects the characteristics and potential biases present in the training data. These biases may affect both predictions and interpretations.

**Regularization trade-offs.** While regularization improves ADHAM's ranking performance, particularly in identifying high-risk patients, our results show a slight reduction in calibration accuracy. Future work could explore ways to balance this trade-off more effectively.

**Applicability to other data types.** ADHAM is tailored for structured, tabular, and time-series data. Adapting it to handle other modalities, such as imaging or text, is a promising area for future exploration.

**Competing risks.** The current formulation of ADHAM is limited to single-risk scenarios, and adapting it to competing risk frameworks is an open problem with important ramifications (Jeanselme et al., 2025).

## 7. Conclusion

In this paper, we introduce ADHAM, a novel survival analysis model that integrates deep additive hazard functions with a mixture-based structure to provide interpretable predictions at the population, subgroup, and individual levels. By decoupling hazard and mixture learning, ADHAM mitigates common interpretability challenges such as concurvity and provides patient-specific risk explanations. Furthermore, we propose a post-training refinement mechanism for selecting the number of subgroups *a posteriori*. ADHAM achieves competitive predictive performance while offering practical interpretability of covariates that support clinical understanding and decision-making, positioning it as a useful tool for real-world applications in healthcare.

## Acknowledgments

Mert Ketenci acknowledges this research is supported by NHLBI award R01HL148248.

## References

Odd Aalen. Nonparametric inference for a family of counting processes. *The Annals of Statistics*, pages 701–726, 1978.

Talal AA Abdullah, Mohd Soperi Mohd Zahid, and Waleed Ali. A review of interpretable ml in healthcare: taxonomy, applications, challenges, and future directions. *Symmetry*, 13(12):2439, 2021.

Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

Muhammad Aurangzeb Ahmad, Carly Eckert, and Ankur Teredesai. Interpretable machine learning in healthcare. In *Proceedings of the 2018 ACM international conference on bioinformatics, computational biology, and health informatics*, pages 559–560, 2018.

Julia Amann, Alessandro Blasimme, Effy Vayena, Dietmar Frey, Vince I Madai, and Precise4Q Consortium. Explainability for artificial intelligence in healthcare: a multidisciplinary perspective. *BMC medical informatics and decision making*, 20:1–9, 2020.

Nitin Bansal, Xiaohan Chen, and Zhangyang Wang. Can we gain more from orthogonality regularizations in training deep networks? *Advances in Neural Information Processing Systems*, 31, 2018.

Suresh K Bhavnani, Weibin Zhang, Shyam Visweswaran, Mukaila Raji, and Yong-Fang Kuo. A framework for modeling and interpreting patient subgroups applied to hospital readmission: visual analytical approach. *JMIR Medical Informatics*, 10(12):e37239, 2022.

Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*, volume 4. Springer, 2006.

Glenn W Brier et al. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016.

Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1721–1730, 2015.

Taane G Clark, Michael J Bradburn, Sharon B Love, and Douglas G Altman. Survival analysis part i: basic concepts and first analyses. *British journal of cancer*, 89(2):232–238, 2003.

Bernard F Cole, Richard D Gelber, Shari Gelber, Alan S Coates, and Aron Goldhirsch. Polychemotherapy for early breast cancer: an overview of the randomised clinical trials with quality-adjusted survival analysis. *The Lancet*, 358(9278):277–286, 2001.

David R Cox. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)*, 34(2):187–202, 1972.

Angela Dispenzieri, Jerry A Katzmann, Robert A Kyle, Dirk R Larson, Terry M Therneau, Colin L Colby, Raynell J Clark, Graham P Mead, Shaji Kumar, L Joseph Melton III, et al. Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. In *Mayo Clinic Proceedings*, volume 87, pages 517–523. Elsevier, 2012.

Cheryl L Faucett, Nathaniel Schenker, and Jeremy MG Taylor. Survival analysis using auxiliary variables via multiple imputation, with application to aids clinical trial data. *Biometrics*, 58(1):37–47, 2002.

Thomas R Fleming and DY Lin. Survival analysis in clinical trials: past developments and future directions. *Biometrics*, 56(4):971–983, 2000.

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

Patricia M Grambsch and Terry M Therneau. Proportional hazards tests and diagnostics based on weighted residuals. *Biometrika*, 81(3):515–526, 1994.

Timothy G Gregoire and Harry T Valentine. A sampling strategy to estimate the area and perimeter of irregularly shaped planar regions. *Forest science*, 41(3):470–476, 1995.

Yolanda Hagar, David Albers, Rimma Pivovarov, Herbert Chase, Vanja Dukic, and Noémie Elhadad. Survival analysis with electronic health record data: Experiments with chronic kidney disease. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 7 (5):385–403, 2014.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *J. Mach. Learn. Res.*, 21(85):1–63, 2020.

Xintian Han, Mark Goldstein, and Rajesh Ranganath. Survival mixture density networks. *arXiv preprint arXiv:2208.10759*, 2022.

Stefan Hegselmann, Thomas Volkert, Hendrik Ohlenburg, Antje Gottschalk, Martin Dugas, and Christian Ertmer. An evaluation of the doctor-interpretability of generalized additive models with interactions. In *Machine Learning for Healthcare Conference*, pages 46–79. PMLR, 2020.

Kenneth R Hess. Graphical methods for assessing violations of the proportional hazards assumption in cox regression. *Statistics in medicine*, 14(15):1707–1723, 1995.

Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *The annals of applied statistics*, 2(3):841–860, 2008.

Vincent Jeanselme, Brian Tom, and Jessica Barrett. Neural survival clustering: Non-parametric mixture of neural networks for survival clustering. In *Conference on Health, Inference, and Learning*, pages 92–102. PMLR, 2022.

Vincent Jeanselme, Chang Ho Yoon, Brian Tom, and Jessica Barrett. Neural fine-gray: Monotonic neural networks for competing risks. In *Conference on Health, Inference, and Learning*, pages 379–392. PMLR, 2023.

Vincent Jeanselme, Brian Tom, and Jessica Barrett. Competing risks: Impact on risk estimation and algorithmic fairness. *arXiv preprint arXiv:2508.05435*, 2025.

Zhenjie Jiang. Coxnams: Interpretable deep learning model for survival analysis. Master's thesis, ETH Zurich, 2022.

Edward L Kaplan and Paul Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1): 1–12, 2018.

Mert Ketenci, Shreyas Bhave, Noemie Elhadad, and Adler Perotte. Maximum likelihood estimation of flexible survival densities with importance sampling. In *Machine Learning for Healthcare Conference*, pages 360–380. PMLR, 2023.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

William A Knaus, Frank E Harrell, Joanne Lynn, Lee Goldman, Russell S Phillips, Alfred F Connors, Neal V Dawson, William J Fulkerson, Robert M Califf, Norman Desbiens, et al. The support prognostic model: Objective estimates of survival for seriously ill hospitalized adults. *Annals of internal medicine*, 122(3):191–203, 1995.

László Kovács. Feature selection algorithms in generalized additive models under concurvity. *Computational Statistics*, 39(2):461–493, 2024.

Maxim S Kovalev, Lev V Utkin, and Ernest M Kasimov. Survlime: A method for explaining machine learning survival models. *Knowledge-Based Systems*, 203:106164, 2020.

Mateusz Krzyziński, Mikołaj Spytek, Hubert Baniecki, and Przemysław Biecek. Survshap (t): time-dependent explanations of machine learning survival models. *Knowledge-Based Systems*, 262:110234, 2023.

Håvard Kvamme. havakv/pycox: Survival analysis with pytorch. https://github.com/havakv/pycox, 11 2022.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of Machine Learning Research*, 2019.

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Changhee Lee, William Zame, Ahmed Alaa, and Mihaela Schaar. Temporal quilting for survival analysis. In *The 22nd international conference on artificial intelligence and statistics*, pages 596–605. PMLR, 2019.

Yang Li, Dongzuo Liang, Shuangge Ma, and Chenjin Ma. Spatio-temporally smoothed deep survival neural network. *Journal of Biomedical Informatics*, 137:104255, 2023.

Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate Intelligible Models with Pairwise Interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 623–631, 2013.

Sheng-Chieh Lu, Christine L Swisher, Caroline Chung, David Jaffray, and Chris Sidey-Gibbons. On the importance of interpretable machine learning predictions to inform clinical decision making in oncology. *Frontiers in Oncology*, 13:1129380, 2023.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937. PmLR, 2016.

Satoshi Morita, Isamu Okamoto, Kunihiko Kobayashi, Koichi Yamazaki, Hajime Asahina, Akira Inoue, Koichi Hagiwara, Noriaki Sunaga, Noriko Yanagitani, Toyoaki Hida, et al. Combined survival analysis of prospective clinical trials of gefitinib for non–small cell lung cancer with egfr mutations. *Clinical Cancer Research*, 15(13):4493–4498, 2009.

Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*, 2011.

Chirag Nagpal, Xinyu Li, and Artur Dubrawski. Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. *IEEE Journal of Biomedical and Health Informatics*, 25(8):3163–3175, 2021a.

Chirag Nagpal, Steve Yadlowsky, Negar Rostamzadeh, and Katherine Heller. Deep cox mixtures for survival regression. In *Machine Learning for Healthcare Conference*, pages 674–708. PMLR, 2021b.

Chirag Nagpal, Mononito Goswami, Keith Dufendach, and Artur Dubrawski. Counterfactual phenotyping with censored time-to-events. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022.

Wayne Nelson. Hazard plotting for incomplete failure data. *Journal of Quality Technology*, 1(1):27–52, 1969.

Maryam Panahiazar, Vahid Taslimitehrani, Naveen Pereira, and Jyotishman Pathak. Using ehrs and machine learning for heart failure survival analysis. *Studies in health technology and informatics*, 216:40, 2015.

Matthew Peroni, Marharyta Kurban, Sun Young Yang, Young Sun Kim, Hae Yeon Kang, and Ji Hyun Song. Extending the neural additive model for survival analysis with ehr data. *arXiv preprint arXiv:2211.07814*, 2022.

Adler Perotte, Rajesh Ranganath, Jamie S Hirsch, David Blei, and Noémie Elhadad. Risk prediction for chronic kidney disease progression using heterogeneous electronic health record data and time series analysis. *Journal of the American Medical Informatics Association*, 22(4):872–880, 2015.

Shi-ang Qi, Neeraj Kumar, Mahtab Farrokh, Weijie Sun, Li-Hao Kuan, Rajesh Ranganath, Ricardo Henao, and Russell Greiner. An effective meaningful way to evaluate survival models. *arXiv preprint arXiv:2306.01196*, 2023.

Timothy O Ramsay, Richard T Burnett, and Daniel Krewski. The effect of concurvity in generalized additive models linking mortality to ambient particulate matter. *Epidemiology*, 14(1):18–23, 2003.

David Rindt, Robert Hu, David Steinsaltz, and Dino Sejdinovic. Survival regression with proper scoring rules and monotonic neural networks. In *International conference on artificial intelligence and statistics*, pages 1190–1205. PMLR, 2022.

Edward H Shortliffe and Martin J Sepúlveda. Clinical decision support in the era of artificial intelligence. *Jama*, 320(21):2199–2200, 2018.

Julien Siems, Konstantin Ditschuneit, Winfried Ripken, Alma Lindborg, Maximilian Schambach, Johannes Otterbach, and Martin Genzel. Curve your enthusiasm: concurvity regularization in differentiable generalized additive models. *Advances in Neural Information Processing Systems*, 36:19029–19057, 2023.

Ritesh Singh and Keshab Mukhopadhyay. Survival analysis in clinical trials: Basics and must know areas. *Perspectives in clinical research*, 2(4):145–148, 2011.

Daniel J Tan, Jiayang Chen, Yirui Zhou, Jaryl Shen Quan Ong, Richmond Jing Xuan Sin, Thach V Bui, Anokhi Amit Mehta, Mengling Feng, and Kay Choong See. Association of body temperature and mortality in critically ill patients: an observational study using two large databases. *European Journal of Medical Research*, 29(1):33, 2024.

Liyao Tang, Zhe Chen, Shanshan Zhao, Chaoyue Wang, and Dacheng Tao. All points matter: entropy-regularized distribution alignment for weakly-supervised 3d segmentation. *Advances in Neural Information Processing Systems*, 36:78657–78673, 2023.

Sana Tonekaboni, Shalmali Joshi, Melissa D McCradden, and Anna Goldenberg. What clinicians want: contextualizing explainable machine learning for clinical end use. In *Machine learning for healthcare conference*, pages 359–380. PMLR, 2019.

Hajime Uno, Tianxi Cai, Michael J Pencina, Ralph B D'Agostino, and Lee-Jen Wei. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*, 30(10):1105–1117, 2011.

Lev V Utkin, Egor D Satyukov, and Andrei V Konstantinov. Survnam: The machine learning survival model explanation. *Neural Networks*, 147:81–102, 2022.

Antonio Viganò, Marlene Dorgan, Jeanette Buckingham, Eduardo Bruera, and Maria E Suarez-Almazor. Survival prediction in terminal cancer patients: a systematic review of the medical literature. *Palliative Medicine*, 14(5):363–374, 2000.

Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

Zifeng Wang and Jimeng Sun. Survtrace: Transformers for survival analysis with competing events. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 1–9, 2022.

Liangchen Xu and Chonghui Guo. Coxnam: An interpretable deep survival analysis model. *Expert Systems with Applications*, page 120218, 2023.

## Appendix A. Notation Table

| Symbol | Meaning |
|---|---|
| $N$ | Dataset size |
| $D$ | Feature dimensionality |
| $C$ | Subgroup size |
| $\mathcal{D}$ | Empirical dataset |
| $\tilde{\mathbf{T}}_{LM}$ | Importance time samples to approximate loglikelihood objective |
| $i$ | Data instance index |
| $d$ | Feature index |
| $h$ | Correlation threshold to merge subgroups |
| $\mathbf{x}$ | Observed covariates |
| $\delta$ | Censoring index |
| $t$ | Recorded time-to-ecent or censoring time |
| $\phi_d$ | Population-level hazard network parameter |
| $\theta$ | Subgroup assignment network parameter |
| $\Phi$ | $\Phi = \{\phi_d\}_{d=1}^{D}$ |

Table 5: Notation Table

## Appendix B. Flow chart

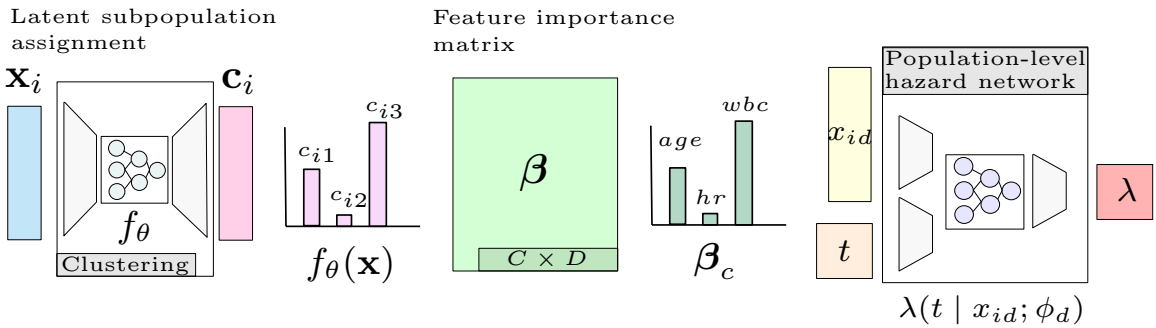



Figure 6: Flowchart of ADHAM. The covariates are mapped to latent subgroups, which are used to index the rows of $\beta$. The population-level hazard curves are calculated via $\lambda(t \mid x_{id}; \phi_d)$, and the marginal hazard is calculated by marginalizing the population-level curve using subgroup-level weights, and instance-specific subgroup assignments.

## Appendix C. Model Selection

### C.1. Merging Same subgroups, with $\boldsymbol{\beta}_{c_1} = \boldsymbol{\beta}_{c_2}$, Retains the Data Generating Log-likelihood and Predictive Performance

In this section, we show that our model selection procedure, outlined in Algorithm 2, does not change the data log-likelihood and predictive performance if $\boldsymbol{\beta}_{c_1} = \boldsymbol{\beta}_{c_2}$.

**Proposition 1** *if $\boldsymbol{\beta}_{c_1} = \boldsymbol{\beta}_{c_2}$, for $c_1$ and $c_2 \in \mathcal{C} = \{1, 2, \cdots, C\}$, then grouping $c_1$ and $c_2$ into a new group $c^\star$, where $p(c^\star \mid \mathbf{x}; \theta) = p(c_1 \mid \mathbf{x}; \theta) + p(c_2 \mid \mathbf{x}; \theta)$ does not change the data generating log-likelihood and risk predictions.*

**Proof** Consider an input patient $\mathbf{x}$ and a model in which two groups, $c_1$ and $c_2$, share identical covariate importance vectors, i.e., $\boldsymbol{\beta}_{c_1} = \boldsymbol{\beta}_{c_2}$. To complete the proof, it suffices to show that their combined contribution to the marginal hazard function remains unchanged when merged into a single group $c^\star$, with assignment probability $p(c^\star \mid \mathbf{x}; \theta) = p(c_1 \mid \mathbf{x}; \theta) + p(c_2 \mid \mathbf{x}; \theta)$:

$$
\begin{aligned}
&\sum_{c=1}^{C} \sum_{d=1}^{D} \beta_{dc} p(c \mid \mathbf{x}; \theta) \lambda(t \mid x_{id}; \phi_d) \\
&= \sum_{\substack{c \neq c_1 \\ c \neq c_2}} \sum_{d=1}^{D} \beta_{dc} p(c \mid \mathbf{x}; \theta) \lambda(t \mid x_{id}; \phi_d) + \sum_{c \in \{c_1, c_2\}} \sum_{d=1}^{D} \beta_{dc} p(c \mid \mathbf{x}; \theta) \lambda(t \mid x_{id}; \phi_d) \\
&= \sum_{\substack{c \neq c_1 \\ c \neq c_2}} \sum_{d=1}^{D} \beta_{dc} p(c \mid \mathbf{x}; \theta) \lambda(t \mid x_{id}; \phi_d) + (p(c_1 \mid \mathbf{x}; \theta) + p(c_2 \mid \mathbf{x}; \theta)) \sum_{d=1}^{D} \beta_{dc^\star} \lambda(t \mid x_{id}; \phi_d) \\
&= \sum_{\substack{c \neq c_1 \\ c \neq c_2}} \sum_{d=1}^{D} \beta_{dc} p(c \mid \mathbf{x}; \theta) \lambda(t \mid x_{id}; \phi_d) + p(c^\star \mid \mathbf{x}; \theta) \sum_{d=1}^{D} \beta_{dc^\star} \lambda(t \mid x_{id}; \phi_d) \\
&= \sum_{c \in \mathcal{C}^\star} \sum_{d=1}^{D} p(c \mid \mathbf{x}; \theta) \beta_{dc^\star} \lambda(t \mid x_{id}; \phi_d). \qquad (12)
\end{aligned}
$$

The resulting set is updated to $\mathcal{C}^\star = c^\star \cup \mathcal{C} \setminus \{c_1, c_2\}$, where $\mathcal{C} := \{1, 2, \ldots, C\}$. Since the marginal hazard function is the same, the model data log-likelihood and predictive performance do not change. ■

In practice, it is uncommon for $\boldsymbol{\beta}_{c_1}$ and $\boldsymbol{\beta}_{c_2}$ to be exactly identical. However, as outlined in Algorithm 2, we can define a similarity measure and apply a threshold to merge subgroups, using a performance metric to guide this process—as demonstrated in the following section.

## Appendix D. Multi-level Interpretability of Survival Function

Population-level survival functions can be computed by:

$$S(t \mid x_d; \phi_d) = \exp\left\{-\int_0^t \lambda(s \mid x_d; \phi_d)\,\mathrm{d}s\right\} \tag{13}$$

This function outputs a probability value, between 0 and 1, and is the same across the population for the same $x_d$ value.

Individual-level survival function is the composition of population-level survival functions exponentiated by patient-specific weights, $p(d \mid \mathbf{x}; \theta, \boldsymbol{\beta})$:

$$S(t \mid \mathbf{x}; \theta, \boldsymbol{\beta}, \Phi) = \exp\left\{-\sum_{d=1}^{D}\left(\sum_{c=1}^{C} p(d \mid c; \boldsymbol{\beta}) p(c \mid \mathbf{x}; \theta) \lambda(t \mid x_d; \phi_d)\,\mathrm{d}s\right)\right\} \tag{14}$$

$$= \prod_{c=1}^{C}\left(\prod_{d=1}^{D}\left(\exp\left\{-\lambda(s \mid x_d; \phi_d)\,\mathrm{d}s\right\}\right)^{\beta_{dc}}\right)^{f_{\theta_c}(\mathbf{x})} \tag{15}$$

$$= \prod_{d=1}^{D}\exp\left\{-\lambda(s \mid x_d; \phi_d)\,\mathrm{d}s\right\}^{p(d|\mathbf{x};\theta,\boldsymbol{\beta})} \tag{16}$$

Note that, $\exp\{-\lambda(s \mid x_d; \phi_d)\,\mathrm{d}s\}$ is the same across the population for the same $x_d$. $\beta_{dc}$ adjusts the latter for subgroup (i.e., for a given subgroup $c$ the survival function differs by some exponent $\beta_{dc}$). Finally, $f_{\theta_c}(\mathbf{x})$ modulates the subgroup weights given a patient covariates.

## Appendix E. Population Level Curves by Different Runs

In this section, we compare the population-level survival functions of TimeNAM (**top**) and ADHAM (**bottom**) on **SUPPORT** dataset over different runs (each with identical random seeds).[4] Higher values imply longer time-to-event (i.e., lower risk of observing the event within time $t$). Survival functions for TimeNAM and ADHAM both capture the well-established trend of increasing risk with age (i.e., survival probability decreases with age). ADHAM also captures the well-known physiological trends in heart rate and temperature. In particular, ADHAM highlights the values linked to normal ranges (e.g., 36 - 37.5 °C for temperature and 60 - 100 bpm for heart rate (Tan et al., 2024)), where risk values are lower, while it is harder to observe this for TimeNAM. *The results are consistent for different runs.*

**Run 1.**

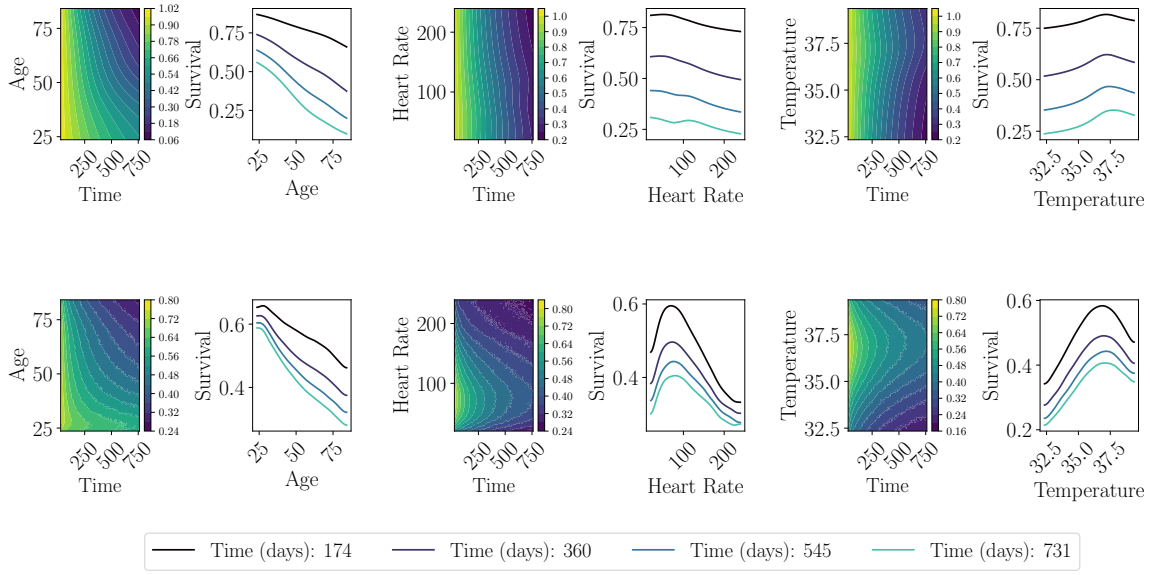



Figure 7: Covariate-specific *population-level* risk functions of TimeNAM (**top**) and ADHAM (**bottom**) trained on **SUPPORT** dataset on run 1.

4. While it is possible to compare individual curves, we focus on population-level interpretability over different model runs as it provides a higher degree of information across the entire dataset rather than select patients.

**Run 2.**

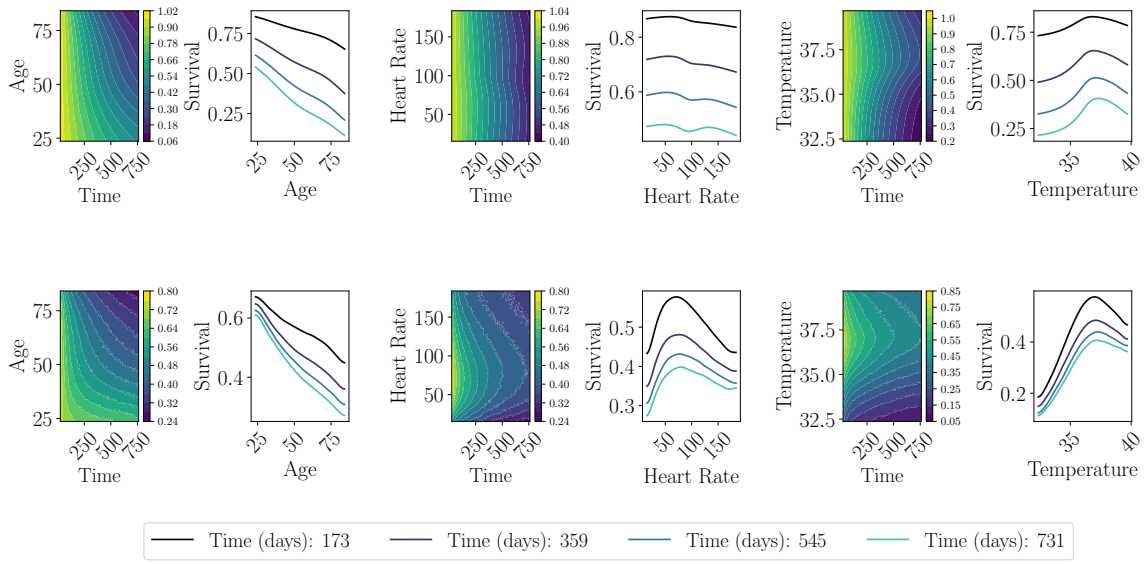



Figure 8: Covariate-specific *population-level* risk functions of TimeNAM (**top**) and ADHAM (**bottom**) trained on **SUPPORT** dataset on run 2.

**Run 3.**

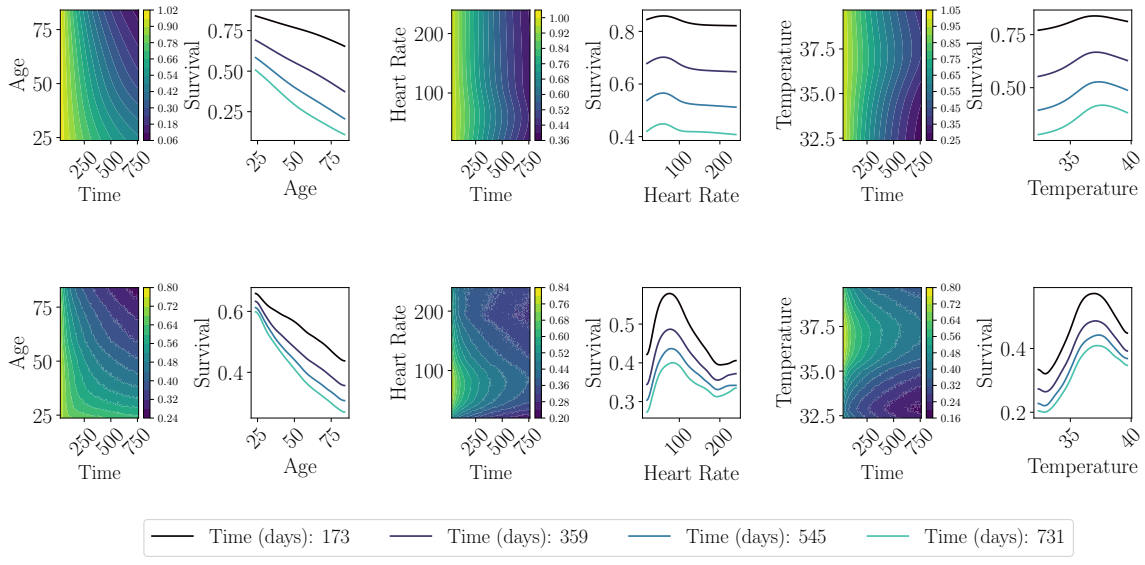



Figure 9: Covariate-specific *population-level* risk functions of TimeNAM (**top**) and ADHAM (**bottom**) trained on **SUPPORT** dataset on run 3.

**Run 4.**

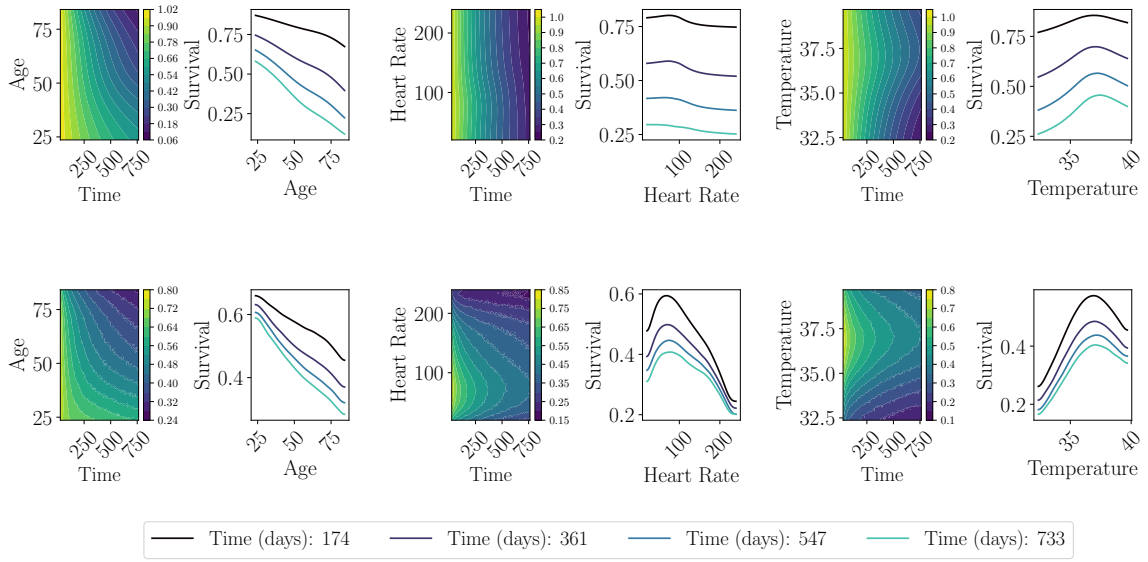



Figure 10: Covariate-specific *population-level* risk functions of TimeNAM (**top**) and ADHAM (**bottom**) trained on **SUPPORT** dataset on run 4.

**Run 5.**

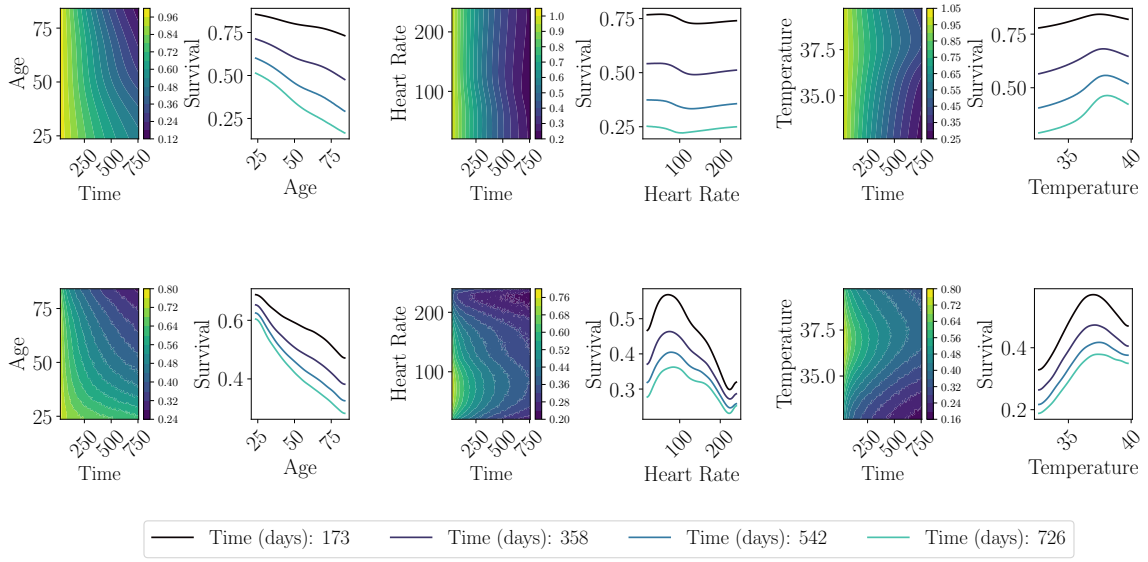



Figure 11: Covariate-specific *population-level* risk functions of TimeNAM (**top**) and ADHAM (**bottom**) trained on **SUPPORT** dataset on run 5.

## Appendix F. Experimental Setup Details

In this section, we explain our datasets in detail.

### F.1. Dataset Details

**SUPPORT.** The Study to Understand Prognoses Preferences Outcomes and Risks of Treatment dataset (Knaus et al., 1995). After preprocessing with the `PyCox` library (Kvamme, 2022), there are 8,873 patients and 23 covariates with a median follow-up of 231 days and a censoring rate of 31.9%.

**FLCHAIN.** Data collected from a controlled trial in Olmsted County, Minnesota, comprised of assays of serum free light chain (FLCHAIN) and mortality data (Dispenzieri et al., 2012). There are 6524 patients with 16 covariates with a median follow-up of 4,303 days and a censoring rate of 70% after preprocessing with `PyCox`.

**CKD.** Electronic health record data from a large urban hospital is used for this dataset. It comprises a cohort of patients with incident chronic kidney disease (CKD) where the event of interest is in-hospital diagnosis of acute kidney injury. The dataset contains 10,173 patients and 33 covariates. The median follow-up days and censoring rate are 67 days and 64%, respectively. This dataset was used in (Ketenci et al., 2023), one of our comparator models.

### F.2. SUPPORT Dataset Abbreviations

Here, we explain the abbreviations used in the **SUPPORT** dataset for Figures 6, 3, and 4.

Feature Maps
- Age – Age
- Number Of Comorbidities – Comorbidities (#)
- Presence Of Cancer – Cancer
- Mean Arterial Blood Pressure – Map
- Heart Rate – Hr
- Respiration Rate – Rr
- Temperature – Temp
- White Blood Cell Count – Wbc
- Serum'S Sodium – Sna
- Serum'S Creatinine – Scr
- Sex – Sex
- Presence Of Diabetes – Diabetes
- Presence Of Dementia – Dementia

### F.3. Baseline Model Details

**CoxPH.** The semi-parametric Cox proportional hazards model (Cox, 1972). Parameter learning is carried out by optimizing the partial log-likelihood.

**DeepSurv.** A semi-parametric survival model that improves the Cox Proportional Hazards model by employing neural networks, thereby establishing a non-linear proportional hazards function (Katzman et al., 2018).

**Random Survival Forests (RSF).** An extension of random forests that fits multiple trees to survival data by bagging and using the cumulative hazard function computed by the Nelson-Aalen estimator (Ishwaran et al., 2008).

**DeepHit.** A discrete-time survival model parameterized by neural networks with a softmax output layer. DeepHit uses cross-entropy loss combined with a ranking loss (Lee et al., 2018).

**Cox-Time.** A semi-parametric method that extends Cox analysis beyond proportional hazards. Cox-Time uses neural networks to parameterize the hazard function. Parameter estimation is done by optimizing a biased approximation of the partial log-likelihood (Kvamme et al., 2019).

**TimeNAM & TimeNA2M.** TimeNA2M extends Cox-Time by using a neural additive hazard function that explicitly captures both main effects and pairwise (second-order) interactions between covariates (Agarwal et al., 2021; Peroni et al., 2022).

**Deep Survival Machines (DSM).** A parametric survival model that uses a mixture of Weibull and log-normal distributions. Mixture assignments and time-to-event distributions are parameterized by neural networks conditioned on covariates. Parameter estimation is done by optimizing the ELBO, where the expectation is taken with respect to the conditional model prior (Nagpal et al., 2021a).

**Deep Cox Mixtures (DCM).** An semi-parametric extension of DSM where each time-to-event density is modeled by DeepSurv. DCM assumes that the hazards within subgroups is proportional. Parameter estimation is done by the Expectation-Maximization (EM) algorithm and fitting polynomial splines to baseline hazards (Nagpal et al., 2021b).

**Deep Hazard Analysis (DHA).** A fully parametric survival analysis approach that directly models the hazard function. The intractable cumulative hazard integral is handled by importance sampling and learning is performed by exact log-likelihood optimization (Ketenci et al., 2023).

### F.4. Hyperparameter Details

All models have an equal training length of 4000 epochs. We pick the best-performing model with respect to their validation loss. The hyper-parameter spaces of each benchmark model are listed below.

**CoxPH.**

'alpha': [0, 1e-3, 1e-2, 1e-1],

**DeepSurv.**

'lr' : [5e-4, 1e-3],
'batch_size': [256, 512, 1024],
'weight_decay': [0, 1e-8, 1e-6, 1e-3, 1e-1],
'nodes_':[128, 256, 512],
'layers_': [2, 3],
'dropout': [0, 1e-1, 2e-1, 4e-1, 5e-1],

**RSF.**

'max_depth' : [None, 5],
'n_estimators' : [50, 100, 150, 200, 150],
'max_covariates' : [50, 75, sqrt(d), d//2, d],
'min_samples_split' : [10, 150, 200, 250],

'max_depth':None means that the expansion continues until all leaves are pure.

**DSM.**

'k ': [3, 4, 6],
'distribution' : ['Weibull', 'LogNormal'],
'learning_rate' : [1e-4, 5e-4, 1e-3],
'nodes_' : [48, 64, 96, 256],
'hidden_layers_': [1, 2, 3],
'discount': [1/3, 3/4, 1],
'batch_size': [128, 256],

**DCM.**

'k' : [3, 4, 6],
'nodes_' : [48, 64, 96, 256],
'hidden_layers_': [1, 2, 3],
'batch_size': [128, 256],
'use_activation': [True, False],

**Deep-Hit.**

'lr' : [5e-4, 1e-3],
'batch_size': [256, 512, 1024],
'weight_decay': [0, 1e-8, 1e-6, 1e-3, 1e-1],
'nodes_':[128, 256, 512],

'hidden_layers_': [2, 3],
'dropout': [0, 1e-1, 2e-1, 4e-1, 5e-1],
'alpha': [1e-1, 2e-1, 4e-1, 8e-1, 1],
'sigma' : [1e-1, 2.5e-1, 4e-1, 8e-1, 1, 2, 10],
'num_durations' : [10, 50, 100],

**Cox-Time.**

'lr' : [5e-4, 1e-3],
'batch_size': [256, 512, 1024],
'weight_decay': [0, 1e-8, 1e-6, 1e-3, 1e-1],
'nodes_':[128, 256, 512],
'hidden_layers_': [1,2,3],
'dropout': [0, 1e-1, 2e-1, 4e-1, 5e-1],
'lambda': [0, 1e-3, 1e-2, 1e-1],
'log_duration' : [True, False],

**TimeNAM & TimeNA2M.**

'lr' : [5e-4, 1e-3],
'batch_size': [256, 512, 1024],
'weight_decay': [0, 1e-8, 1e-6, 1e-3, 1e-1],
'nodes_':[128, 256, 512],
'hidden_layers_': [1,2,3],
'dropout': [0, 1e-1, 2e-1, 4e-1, 5e-1],
'lambda': [0, 1e-3, 1e-2, 1e-1],
'log_duration' : [True, False],

**DHA.**

'lr' : 1e-3,
'batch_size': 512,
'imps_size': 64,
'layer_norm' : True,
'weight_decay':0,
'nodes_': 100
'layers_': 3,
'dropout': 0.15,
'act': elu,

**ADHAM.**

'lr' : 1e-3,
'batch_size': 512,
'imps_size': 64,
'layer_norm' : True,
'weight_decay':0,
'nodes_': 100
'layers_': 3,
'dropout': [0, 0.15],
'act': elu,
'n_mixtures': 100,
'add_const': [True, False],

For DHA and ADHAM, we use the architecture A1 defined in (Ketenci et al., 2023).

### F.5. Evaluation Metric Details

In this section, we explain the evaluation metrics.

**Concordance Index (C-Index).** The C-Index measures how well a model ranks individuals with respect to their risk. We use the inverse probability of censoring weighting (IPCW)-based C-Index proposed by Uno et al. (2011), which evaluates the agreement between predicted and observed orderings of events, with a time cutoff $t$. It is defined as:

$$\mathrm{C}(t) = p(S(t|\mathbf{x}_i) < S(t|\mathbf{x}_j) \mid \delta_i = 1, t_i < t_j, t_i < t). \tag{17}$$

Larger C-Index is better.

**Brier Score (BS).** The Brier Score (Brier et al., 1950) measures the calibration of predicted survival probabilities. It is defined as the expected squared difference between the predicted survival probability and the event occurrence indicator. To handle censored data, we use the extension of the Brier Score proposed by (Graf et al., 1999), which employs inverse probability of censoring weighting (IPCW)[5]. The definition is:

$$\mathrm{BS}(t) = \mathbb{E}_{t_i, \mathbf{x}_i \sim \mathcal{D}} \left\{ (I_{t_i > t} - S(t|\mathbf{x}_i))^2 \right\}. \tag{18}$$

Smaller BS is better.

**Area Under the Receiver Operating Characteristic (AUROC).** The time-dependent AUROC at a fixed cutoff time measures the model's ability to distinguish between individuals who experience the event by time $t$ and those who do not. It is defined as the probability that the predicted risk (i.e., one minus the survival probability) is higher for an individual who experiences the event before or at time $t$ than for one who survives beyond $t$:

$$\mathrm{AUC}(t) = p(S(t|\mathbf{x}_i) \leq S(t|\mathbf{x}_j) \mid t_i \leq t, t_j > t). \tag{19}$$

Larger AUROC is better.

5. Although, IPCW can fail to provide accurate estimates for censored individuals when there are no comparable individuals who experienced the event afterward, it is a widely used method to correct for censoring (Qi et al., 2023).

## Appendix G. The Standard Error of the Sample Mean

In this section, we demonstrate the standard error results.

| **SUPPORT** Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | $25^{\text{th}}$ Quantile | | | $50^{\text{th}}$ Quantile | | | $75^{\text{th}}$ Quantile | | |
| | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ |
| DeepSurv | 0.008 | 0.002 | 0.009 | 0.004 | 0.002 | 0.005 | 0.004 | 0.001 | 0.005 |
| RSF | 0.005 | 0.003 | 0.005 | 0.003 | 0.001 | 0.004 | 0.002 | 0.002 | 0.004 |
| DeepHit | 0.009 | 0.003 | 0.008 | 0.005 | 0.003 | 0.006 | 0.005 | 0.002 | 0.006 |
| Cox-Time | 0.013 | 0.002 | 0.012 | 0.007 | 0.002 | 0.008 | 0.004 | 0.001 | 0.004 |
| DSM | 0.006 | 0.002 | 0.004 | 0.005 | 0.002 | 0.006 | 0.004 | 0.002 | 0.005 |
| DCM | 0.009 | 0.003 | 0.008 | 0.006 | 0.002 | 0.006 | 0.005 | 0.002 | 0.005 |
| DHA | 0.008 | 0.003 | 0.007 | 0.005 | 0.002 | 0.006 | 0.005 | 0.002 | 0.006 |
| CoxPH | 0.007 | 0.002 | 0.007 | 0.003 | 0.002 | 0.003 | 0.002 | 0.001 | 0.004 |
| TIMENAM | 0.008 | 0.002 | 0.008 | 0.003 | 0.002 | 0.003 | 0.002 | 0.001 | 0.003 |
| TIMENA2M | 0.009 | 0.003 | 0.009 | 0.004 | 0.002 | 0.004 | 0.004 | 0.001 | 0.003 |
| ADHAM ($\mathcal{R}$) | 0.006 | 0.002 | 0.004 | 0.004 | 0.002 | 0.006 | 0.003 | 0.000 | 0.003 |
| ADHAM | 0.008 | 0.003 | 0.005 | 0.004 | 0.001 | 0.006 | 0.003 | 0.001 | 0.004 |

Table 6: Standard error of the sample mean (SEM) results on the **SUPPORT** dataset.

| **CKD** Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | $25^{\text{th}}$ Quantile | | | $50^{\text{th}}$ Quantile | | | $75^{\text{th}}$ Quantile | | |
| | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ |
| DeepSurv | 0.005 | 0.004 | 0.006 | 0.005 | 0.004 | 0.006 | 0.007 | 0.004 | 0.007 |
| RSF | 0.011 | 0.002 | 0.012 | 0.004 | 0.004 | 0.006 | 0.004 | 0.005 | 0.009 |
| DeepHit | 0.010 | 0.003 | 0.010 | 0.004 | 0.004 | 0.005 | 0.007 | 0.005 | 0.009 |
| Cox-Time | 0.005 | 0.004 | 0.008 | 0.005 | 0.005 | 0.007 | 0.005 | 0.006 | 0.008 |
| DSM | 0.009 | 0.004 | 0.008 | 0.007 | 0.003 | 0.008 | 0.006 | 0.003 | 0.008 |
| DHA | 0.008 | 0.004 | 0.008 | 0.008 | 0.004 | 0.009 | 0.005 | 0.004 | 0.008 |
| CoxPH | 0.006 | 0.002 | 0.007 | 0.006 | 0.004 | 0.009 | 0.003 | 0.004 | 0.006 |
| TIMENAM | 0.008 | 0.002 | 0.008 | 0.008 | 0.004 | 0.009 | 0.006 | 0.004 | 0.010 |
| TIMENA2M | 0.011 | 0.003 | 0.011 | 0.005 | 0.006 | 0.006 | 0.005 | 0.008 | 0.008 |
| ADHAM ($\mathcal{R}$) | 0.016 | 0.004 | 0.019 | 0.01 | 0.005 | 0.011 | 0.006 | 0.005 | 0.007 |
| ADHAM | 0.008 | 0.004 | 0.008 | 0.007 | 0.004 | 0.009 | 0.007 | 0.004 | 0.010 |

Table 7: Standard error of the sample mean (SEM) results on the **CKD** dataset.

| **FLCHAIN** Dataset | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Models | $25^{\text{th}}$ Quantile | | | $50^{\text{th}}$ Quantile | | | $75^{\text{th}}$ Quantile | | |
| | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ | C-Index ↑ | BS ↓ | AUROC ↑ |
| DeepSurv | 0.004 | 0.001 | 0.004 | 0.005 | 0.001 | 0.005 | 0.004 | 0.002 | 0.005 |
| RSF | 0.005 | 0.001 | 0.006 | 0.005 | 0.002 | 0.006 | 0.003 | 0.001 | 0.004 |
| DeepHit | 0.004 | 0.002 | 0.004 | 0.007 | 0.002 | 0.007 | 0.005 | 0.002 | 0.006 |
| Cox-Time | 0.004 | 0.004 | 0.005 | 0.006 | 0.012 | 0.007 | 0.004 | 0.024 | 0.005 |
| DSM | 0.008 | 0.001 | 0.009 | 0.010 | 0.002 | 0.011 | 0.005 | 0.001 | 0.006 |
| DCM | 0.003 | 0.002 | 0.003 | 0.009 | 0.004 | 0.010 | 0.007 | 0.003 | 0.008 |
| DHA | 0.003 | 0.002 | 0.004 | 0.007 | 0.002 | 0.008 | 0.004 | 0.001 | 0.005 |
| CoxPH | 0.005 | 0.001 | 0.005 | 0.006 | 0.002 | 0.007 | 0.003 | 0.002 | 0.003 |
| TIMENAM | 0.004 | 0.001 | 0.004 | 0.006 | 0.003 | 0.007 | 0.004 | 0.006 | 0.005 |
| TIMENA2M | 0.006 | 0.001 | 0.007 | 0.007 | 0.006 | 0.008 | 0.003 | 0.015 | 0.004 |
| ADHAM ($\mathcal{R}$) | 0.006 | 0.002 | 0.007 | 0.005 | 0.003 | 0.005 | 0.005 | 0.003 | 0.007 |
| ADHAM | 0.008 | 0.002 | 0.008 | 0.008 | 0.002 | 0.008 | 0.006 | 0.003 | 0.007 |

Table 8: Standard error of the sample mean (SEM) results on the **FLCHAIN** dataset.