MENSA: A Multi-Event Network for Survival Analysis with Trajectory-based Likelihood Estimation

Christian Marius Lillelund[1,2]* CLILLELUND@UALBERTA.CA

Ali Hossein Gharari Foomani[1] HOSSEING@UALBERTA.CA

Weijie Sun[1] WEIJIE2@UALBERTA.CA

Shi-ang Qi[1] SHIANG@UALBERTA.CA

Russell Greiner[1,3] RGREINER@UALBERTA.CA

The Pooled Resource Open-Access ALS Clinical Trials Consortium (PRO-ACT)[4]

[1] *Department of Computing Science, University of Alberta, Edmonton, Canada*
[2] *Department of Electrical and Computer Engineering, Aarhus University, Aarhus, Denmark*
[3] *Alberta Machine Intelligence Institute, Edmonton, Canada*
[4] *PRO-ACT Consortium*†

Abstract

Most existing time-to-event methods focus on either single-event or competing-risks settings, leaving multi-event scenarios relatively under-explored. In many healthcare applications, for example, a patient may experience multiple clinical events, that can be non-exclusive and semi-competing. A common workaround is to train independent single-event models for such multi-event problems, but this approach fails to exploit dependencies and shared structures across events. To overcome these limitations, we propose MENSA (Multi-Event Network for Survival Analysis), a deep learning model that jointly learns flexible time-to-event distributions for multiple events, whether competing or co-occurring. In addition, we introduce a novel trajectory-based likelihood term that captures the temporal ordering between events. Across four multi-event datasets, MENSA improves predictive performance over many state-of-the-art baselines. Source code is available at https://github.com/thecml/mensa.

Keywords: Survival analysis, time-to-event prediction, density estimation, multiple events, competing risks, precision medicine

Data and Code Availability All datasets used in this study are publicly available. Appendix E.1 describes how to obtain these datasets and preprocess them accordingly.

Institutional Review Board (IRB) This study uses only publicly available datasets and does not involve any interaction with human subjects or access to identifiable private information. Therefore, IRB approval was not required.

1. Introduction

Many survival models estimate the probability that an event of interest (typically death) will occur for an individual at some future time. This can involve: a *single event* of interest, where we want to model the time until one specific event (Gareth et al., 2021, Ch. 11), *competing risks*, where the occurrence of one event excludes other events from occurring (Fine and Gray, 1999), or *multiple events*, where several events may occur in some order (Andersen and Keiding, 2002). As an example of the latter, a patient with a chronic condition may experience several medical events over the course of their disease.

In current practice, multi-event problems are often approached by training multiple single-event models (Armstrong et al., 2014; Solomon et al., 2017). However, this approach (1) fails to exploit shared covariate patterns, (2) ignores temporal or causal dependencies between events, and (3) becomes computationally inefficient as the number of events increases. Other methods mitigate these issues through shared frailty terms (Jiang and Haneuse, 2017), multi-task regularization (Li et al., 2016; Wang et al.,

* Corresponding author.

© 2025 C.M. Lillelund *et al.*

2017), or joint survival modeling (Hsieh and Wang, 2018), but these rely on restrictive assumptions about frailty, proportional hazards, or a fixed survival distribution. Recent work has explored structured transition models (Groha et al., 2020), hierarchical time-scale modeling (Tjandra et al., 2021), latent ordinary differential equations (ODEs) for estimating event-specific hazard curves (Moon et al., 2022), and transition models based on Monte Carlo sampling (Rossman et al., 2022). These offer partial solutions but typically assume predefined state transitions, scale poorly to long time horizons, cannot handle multiple events, or depend on sampling to predict an individual survival distributions (ISD) (Haider et al., 2020). To date, most research remains confined to single-event or competing-risks scenarios. Motivated by these limitations, we introduce a new and scalable deep learning model for multi-event survival analysis, called MENSA (Multi-Event Network for Survival Analysis). Our main contributions are:

- We propose a multi-state survival model based on the framework by Andersen and Keiding (2002), where each event corresponds to a transition between two states (*e.g.*, from healthy to sick). This enables prediction of how long a patient will remain in a particular state, or when the patient will experience one of multiple events. Each event-specific marginal distribution is modeled as a mixture of Weibull distributions, relaxing the proportional hazards assumption. Also, this architecture naturally extends to single-event and competing-risks settings.

- We introduce a training objective that combines the standard negative log-likelihood (NLL) loss with an additional trajectory-based term. This new term encodes known temporal relationships between events, guiding the model to predict ISDs that align with prior knowledge.

- We show empirically that MENSA offers solid discrimination and calibration performance, improving over many baseline models. MENSA is also more computationally efficient than several baselines in the multi-event setting. Two ablation studies show that training MENSA jointly across all events leads to a notable increase in the local concordance index (C-index), reflecting better within-patient event ordering, and that using the proposed trajectory-based likelihood term further improves this metric.

2. Related work

Andersen and Keiding (2002) proposed a model for multi-state processes (for example, disease progression leading to relapse and eventually death), capturing sequences of events. This framework has since become a fundamental tool for multi-state analysis (Gorfine et al., 2025). Each transition between states (events) is modeled with its own hazard function, allowing state-specific effects, and naturally incorporates competing risks, recurrent events, and absorbing states (*e.g.*, death). However, individual event probabilities must be derived from transition probabilities across states, so the event time cannot be computed directly. Moreover, the model assumes a linear relationship between covariates and the transition-specific hazard functions.

Groha et al. (2020) proposed an ODE-based neural network for multi-state survival analysis that incorporates a predefined state-transition structure. This structure is encoded using a binary transition matrix, indicating which transitions are allowed. The model then learns time-varying transition intensities by solving Kolmogorov forward equations, but only for those transitions permitted by the predefined graph. A key limitation of this approach is that it enforces a fixed sequence of events – for example, once cancer occurs, the model does not allow subsequent heart failure and vice versa. This framework requires practitioners to specify the full state-transition graph in advance, which can be restrictive when the event order is uncertain or variable.

Tjandra et al. (2021) proposed a hierarchical multi-event model that predicts event occurrences across multiple time scales, using coarse estimates (*e.g.*, monthly) to guide finer predictions (*e.g.*, daily). This approach decomposes long-horizon tasks into simpler intermediate steps, progressively refining predictions from broad to fine temporal resolutions. However, it has two main limitations. First, the hierarchical structure incurs significant parameter overhead and scales poorly with longer durations. Second, the method is optimized solely for discriminative loss. Strong discrimination performance does not necessarily translate into accurate time-to-event predictions, and vice versa (Qi et al., 2023a).

Moon et al. (2022) proposed an ODE-based model for multi-event analysis. This model maps temporal features into continuous latent trajectories and estimates cause-specific hazard functions at discrete time intervals without relying on parametric hazard

assumptions. However, the model is only designed for competing risks, supports at most two events, and assumes access to time-varying features. It cannot be directly applied on datasets with only static (baseline) covariates, which remain the most common in survival analysis. Moreover, while latent ODE-based models offer high flexibility, they face practical limitations. As noted by Chen (2024, p. 66), these models are typically slower to train and more prone to numerical instability than parametric alternatives.

Rossman et al. (2022) introduced a multi-state survival model, *PyMSM*, which estimates transition-specific hazard functions using time-dependent covariates and Monte Carlo sampling. PyMSM is well-suited for settings where individuals move through a sequence of well-defined states (*e.g.*, from illness to recovery to death), and covariates can be updated at each transition. However, its reliance on Monte Carlo sampling introduces two key limitations. First, to estimate an ISD for a patient, the model must sample many possible trajectories, which can be computationally expensive. Second, since the model only approximates the ISDs, it lacks closed-form survival or density functions, which prevents analytical comparisons (*e.g.*, hazard ratios) or mathematical guarantees (*e.g.*, proportional hazards).

3. The proposed method

3.1. Problem formulation

We consider a survival dataset with multiple events and right-censoring, consisting of N time-to-event triplets, denoted as $\mathcal{D} = \left\{\left(\boldsymbol{x}^{(i)}, \boldsymbol{t}^{(i)}, \boldsymbol{\delta}^{(i)}\right)\right\}_{i=1}^{N}$, where $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ denotes a d-dimensional vector of features (covariates), $\boldsymbol{t}^{(i)} \in \{1, 2, \ldots, t_{\max}\}^K$ is a vector of observed times for the i-th instance across K events, and $\boldsymbol{\delta}^{(i)} \in \{0, 1\}^K$ is a vector of event indicators for each event[1]. We consider each instance to have (potential) event times, $e_k^{(i)} \in \{1, 2, \ldots, t_{\max}\}$, and censoring times, $c_k^{(i)} \in \{1, 2, \ldots, t_{\max}\}$. We then have $t_k^{(i)} = e_k^{(i)}$ if $\delta_k^{(i)} = 1$, meaning that the event was observed, otherwise, $t_k^{(i)} = c_k^{(i)}$ if $\delta_k^{(i)} = 0$, meaning that the event was right-censored.

Now, as a practical example, consider a drug trial over $t \in [0, 100]$ days (see Figure 1), where a patient i with features $\boldsymbol{x}^{(i)}$ can experience four side effects: nausea (A), fatigue (B), fever (C), or headache (D). Based on $\boldsymbol{x}^{(i)}$, we want to estimate the so-called survival function $S_E(t) = \Pr(t < T_E)$[2] for each of these possible events (side effects), which is the probability that the patient has not yet experienced the event by $t \geq 0$. By estimating this function for every event, we obtain a matrix of survival functions, $\hat{\boldsymbol{S}}^{(i)} = [\boldsymbol{s}_1^{(i)}, \boldsymbol{s}_2^{(i)}, \ldots, \boldsymbol{s}_K^{(i)}]$, for the i-th patient, where K is the number of events. Finally, we can estimate the patient's likely time to each event as the point where each curve crosses the 50% (median) line.

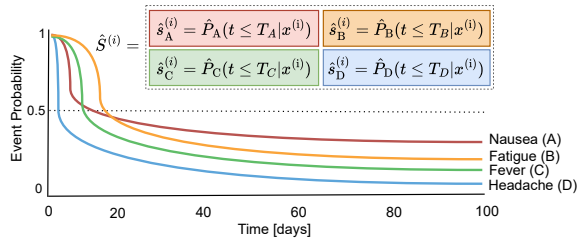

Figure 1: Example of a multi-event prediction.

3.2. Mixture model architecture

We propose a multi-state architecture (see Figure 2) in which events are modeled as transitions between discrete states – *e.g.*, from *healthy* to *sick*, or from *alive* to *dead* – based on the framework by Andersen and Keiding (2002), and further described in Appendix B. Let $\mathcal{P} = \{0, 1, \ldots, P-1\}$ denote the finite set of states, where state 0 is the initial (event-free) state. Each event $k \in \{1, \ldots, K\}$ corresponds to a transition into a state $p_k \in \mathcal{P} \setminus \{0\}$. For example, if event k represents a *relapse*, then p_k denotes the absorbing state for *relapse*. This mapping naturally supports scenarios with sequential or co-occurring events – *e.g.*, first becoming ill and then later dying.

For each instance i, we estimate a matrix of survival distributions $\boldsymbol{S}^{(i)}$: one distribution for each state $p \in \mathcal{P}$, including the initial event-free state ($p = 0$). The marginal density function $f_p(t)$ for state p, which represents the probability of entering state

1. In our terminology, we refer to *competing-risks* problems as those in which only one event can be observed per instance, and the event time is effectively a scalar: $t^{(i)}$. In contrast, *multi-event* problems involve multiple events per instance, and the event time is instead represented as a vector: $\boldsymbol{t}^{(i)}$. Cases involving *semi-competing risks* – where one event precludes another but not vice versa – fall under our definition of multi-event problems.

2. We use T_E to denote the time of event, and T_C to denote the time of censoring. The observed time is then $T = \min(T_E, T_C)$.

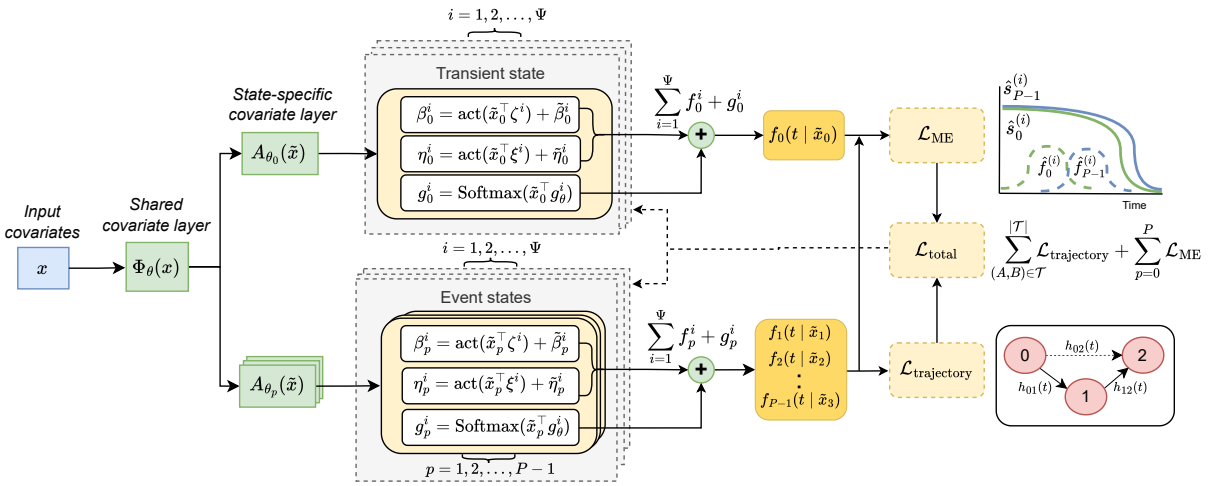

Figure 2: Overview of the MENSA architecture. The model first maps input covariates through a shared multilayer perceptron (MLP) layer, producing a common representation $\tilde{\boldsymbol{x}}$. This is passed through event-specific covariate layers, each producing $\tilde{\boldsymbol{x}}_p$ for state $p \in \mathcal{P}$. Each adapted representation then parameterizes a mixture of Weibull distributions to predict a time-to-event density function, $f_p(t)$, capturing the time of transition from the transient state (state 0) to each event state. The model is trained using a combined loss consisting of a multi-event likelihood, $\mathcal{L}_{\text{ME}}$, which captures how well the model fits event times across all events, and a trajectory likelihood, $\mathcal{L}_{\text{trajectory}}$, which enforces known temporal ordering constraints between pairs of events.

p at time point t, is modeled as a convex combination of $\Psi \in \mathbb{N}_+$ Weibull distributions. This formulation offers two key advantages. First, it relaxes the proportional hazards assumption on the underlying marginal survival distribution (proof in Appendix I), allowing the hazard function to vary over time. Second, it leverages the closed-form probability density function (PDF) and cumulative distribution function (CDF) of the Weibull distribution, respectively:

$$f(t; \beta, \eta) = \frac{\eta}{\beta}\left(\frac{t}{\beta}\right)^{\eta-1} \exp\left(-\left(\frac{t}{\beta}\right)^{\eta}\right), \quad (1)$$

$$F(t; \beta, \eta) = 1 - \exp\left(-\left(\frac{t}{\beta}\right)^{\eta}\right), \quad (2)$$

where $\beta > 0$ is the scale and $\eta > 0$ the shape parameter. We parameterize each such Weibull distribution by two covariate-dependent transformations, that is, $h_p(\boldsymbol{x}^{(i)}) = A_{\theta,p}\big(\Phi_\theta(\boldsymbol{x}^{(i)})\big)$, such that:

$$\log \beta_{\psi,p}(\boldsymbol{x}^{(i)}) = \tilde{\beta}_{\psi,p} + \text{act}\big(h_p(\boldsymbol{x}^{(i)})^\top \zeta_{\psi,p}\big), \quad (3)$$

$$\log \eta_{\psi,p}(\boldsymbol{x}^{(i)}) = \tilde{\eta}_{\psi,p} + \text{act}\big(h_p(\boldsymbol{x}^{(i)})^\top \xi_{\psi,p}\big), \quad (4)$$

where $\Phi_\theta(\cdot)$ is a shared covariate transformation, $A_{\theta,p}(\cdot)$ is a state-specific adapter applied to its output, $\text{act}(\cdot)$ denotes a non-linear activation function, $\psi \in \{1, 2, \ldots, \Psi\}$ indexes the Weibull mixture distributions, and $\zeta_{\psi,p}, \xi_{\psi,p} \in \mathbb{R}^d$ are learnable weight vectors associated with the scale and shape parameters, respectively. The terms $\tilde{\beta}_{\psi,p}$ and $\tilde{\eta}_{\psi,p}$ are learnable biases that initialize the log-scale and log-shape, respectively, for each state p. We define $\boldsymbol{\theta}_{\psi,p}(\boldsymbol{x}^{(i)}) = \big(\beta_{\psi,p}(\boldsymbol{x}^{(i)}), \eta_{\psi,p}(\boldsymbol{x}^{(i)})\big)$, and the corresponding distribution PDF and CDF for transitioning into state p at time t given covariates $\boldsymbol{x}^{(i)}$ are then given by:

$$f_p(t; \boldsymbol{x}^{(i)}) = \sum_{\psi=1}^{\Psi} g_{\psi,p}(\boldsymbol{x}^{(i)})\, f_{\psi,p}\big(t; \boldsymbol{\theta}_{\psi,p}(\boldsymbol{x}^{(i)})\big), \quad (5)$$

$$F_p(t; \boldsymbol{x}^{(i)}) = \sum_{\psi=1}^{\Psi} g_{\psi,p}(\boldsymbol{x}^{(i)})\, F_{\psi,p}\big(t; \boldsymbol{\theta}_{\psi,p}(\boldsymbol{x}^{(i)})\big), \quad (6)$$

where $g_{\psi,p}(\boldsymbol{x}^{(i)})$ are normalized mixture weights produced by a state-specific gating network, *i.e.*, a softmax over linear outputs of $h_p(\boldsymbol{x}^{(i)})$. The corresponding survival function is $S_p(t; \boldsymbol{x}^{(i)}) = 1 - F_p(t; \boldsymbol{x}^{(i)})$.

3.3. Trajectory-based likelihood

To estimate (5) and (6), we extend the standard survival likelihood (defined in Appendix C) to our multi-

Table 1: Possible scenarios for two events, A and B, with corresponding transition probabilities and log-likelihood contributions. δ_A and δ_B indicate whether each event is observed (1) or censored (0). $P_{0\to A}(t_A)$ and $P_{0\to B}(t_B)$ denote the probabilities of experiencing events A and B at times t_A and t_B, respectively; $S_A(t)$ and $S_B(t)$ are survival functions; and $f_A(t)$ and $f_B(t)$ are probability density functions. If prior knowledge dictates that event A must precede event B when both occur, the joint-event likelihood ($\delta_A = 1$, $\delta_B = 1$) incorporates the probability of surviving event B at the time of event A, as highlighted in bold.

#	δ_A	δ_B	Transition probability	Log-likelihood
1	0	0	$1 - (P_{0\to A}(t_A) + P_{0\to B}(t_B))$	$S_A(t_A) + S_B(t_B)$
2	1	0	$P_{0\to A}(t_A)$	$f_A(t_A) + S_B(t_B)$
3	0	1	$P_{0\to B}(t_B)$	$S_A(t_A) + f_B(t_B)$
4	1	1	$P_{0\to A}(t_A) * P_{0\to B}(t_B)$	$f_A(t_A) + f_B(t_B) + \boldsymbol{S_B(t_A)}$

state architecture – where each instance i may experience multiple events. However, rather than indexing events directly, we express the likelihood in terms of state transitions. Specifically, we define one indicator and time pair $(\delta_p^{(i)}, t_p^{(i)})$ per state $p \in \mathcal{P}$, where each event k corresponds to a transition into state p. The indicator $\delta_p^{(i)}$ denotes whether a transition into state p was observed, and $t_p^{(i)}$ is the corresponding observed or censored transition time. The likelihood includes the state's density $f_p(t_p^{(i)} \mid \boldsymbol{x}^{(i)})$, if observed ($\delta_p^{(i)} = 1$), and the survival function $S_p(t_p^{(i)} \mid \boldsymbol{x}^{(i)})$, if censored ($\delta_p^{(i)} = 0$).

To account for the imbalance between state transitions, we introduce non-negative weights, w_p, computed from the inverse frequency of transitions into each state using the training set, normalized so that $\sum_{p\in\mathcal{P}} w_p = |\mathcal{P}|$. Assuming conditional independence between state transitions[3], we can then define our multi-event likelihood as:

$$\mathcal{L}_{\text{ME}}(\mathcal{D}) = \sum_{i=1}^{N} \sum_{p\in\mathcal{P}} w_p \Big[\delta_p^{(i)} \log f_p\Big(t_p^{(i)} \mid \boldsymbol{x}^{(i)}\Big) + (1-\delta_p^{(i)}) \log S_p\Big(t_p^{(i)} \mid \boldsymbol{x}^{(i)}\Big)\Big], \tag{7}$$

where $f_p(\cdot)$ and $S_p(\cdot)$ denote the conditional density and survival function for state p, respectively.

To guide the model, we incorporate prior knowledge of temporal event ordering when available. As an example, consider the illness-death model (Fix and Neyman, 1951), where patients start out healthy (state 0), become ill (event A), and may eventually die (event B). This means there are four possible scenarios (see Table 1): (1) neither event occurred (both censored), (2) only A occurred (B censored), (3) only B occurred (A censored), and (4) both events occurred, with A before B. We assume that if such semi-competing events are observed, one must come before the other – *e.g.*, death cannot occur before illness. To incorporate this constraint into the optimization, we define an auxiliary likelihood term that encourages the model to assign a high survival probability for event B at the time event A occurs, *i.e.*, maximize $S_B(t_A)$. This ensures the model assigns a low probability to B when A has occurred, effectively handling semi-competing risks.

Let $\mathcal{T} = \{(A, B) \mid A, B \in \mathcal{K}, \text{A occurs before B}\}$ denote the set of known event orderings, and for each instance i, we consider all uncensored event pairs $(A, B) \in \mathcal{T}$. The trajectory likelihood is computed by summing the survival probability of event B at the time of event A, thus $S_B(T_A \mid \boldsymbol{x}^{(i)})$, weighted by the event indicators $\delta_A^{(i)}$ and $\delta_B^{(i)}$, which equal 1 if events A and B, respectively, have occurred for instance i, and 0 otherwise. The likelihood is then:

$$\mathcal{L}_{\text{trajectory}}(\mathcal{D}) = \sum_{i=1}^{N} \sum_{(A,B)\in\mathcal{T}} \delta_A^{(i)} \delta_B^{(i)} \log S_B\big(T_A \mid \boldsymbol{x}^{(i)}\big), \tag{8}$$

where $\mathcal{T}$ is the set of trajectories. Finally, we have:

$$\mathcal{L}_{\text{total}}(\mathcal{D}) = (1-\lambda)\frac{1}{N}\mathcal{L}_{\text{ME}}(\mathcal{D}) + \lambda \cdot \frac{1}{N}\mathcal{L}_{\text{trajectory}}(\mathcal{D}), \tag{9}$$

where λ is a weighting hyperparameter that controls the trade-off between the two likelihoods. The training algorithm can be found in Appendix E.4.

3. Conditional independence means that, given the covariates, the transition to one state is assumed to be independent of transitions to other states. Therefore, any relationship between states is assumed explained by the covariates.

4. Experiments

4.1. Experimental setup

Datasets. We use five datasets to test the effectiveness of our method. Table 7 in Appendix E summarizes the dataset statistics, and Appendix E.1 contains details of preprocessing steps, and flow charts of event trajectories. These datasets cover single-event, competing-risks and multi-event settings.

Baselines. We compare 11 baseline survival models: Cox proportional hazards (CoxPH) (Cox, 1972), CoxNet (Simon et al., 2011), Weibull AFT (Klein and Moeschberger, 2006), GBSA (Ridgeway, 1999), RSF (Ishwaran et al., 2008), NFG[4] (Jeanselme et al., 2023), DeepSurv (Katzman et al., 2018), DeepHit (Lee et al., 2018), Hierarchical (Tjandra et al., 2021), MTLR (Yu et al., 2011; Kim et al., 2021), and DSM (Nagpal et al., 2021). Appendix E.2 provides implementation details.

Hyperparameters. To ensure a fair comparison, all models – including baselines – were tuned using Bayesian optimization (Snoek et al., 2012) over ten iterations. For each model and dataset, we selected the hyperparameters that achieved the highest Harrell's C-index (Harrell Jr et al., 1996) on the validation set. Appendix E.3 lists the chosen parameters.

Evaluation metrics. To evaluate the discrimination performance, we report the global C-index (CI_G) (Tjandra et al., 2021), which measures how well the model can discriminate instances within an event, and the local C-index (CI_L) (Tjandra et al., 2021), which measures how well the model can discriminate events within an instance, and the time-dependent AUC, which measures discrimination at a fixed time point. We calculate the AUC by computing it at the 25th, 50th, and 75th percentiles of observed event times in the test set and then averaging these values. To evaluate calibration performance, we report the mean absolute error (MAE) using a margin loss (mMAE) (Haider et al., 2020), the integrated Brier score (IBS) (Graf et al., 1999) and the distribution calibration (D-calibration) score (Haider et al., 2020). Appendix D gives more details on the metrics.

4.2. Implementation details

Following recent literature (Katzman et al., 2018; Nagpal et al., 2021; Lillelund et al., 2025), we use a multilayer perceptron (MLP) for the backbone network architecture. This transforms the raw input features into a learned representation, which is then used to parameterize the Weibull distributions. On top of this shared representation, we add small state-specific MLPs to fine-tune representations for each prediction task. We use a single hidden layer, the ReLU6 activation function for the MLP, and a mixture of Ψ Weibull distributions. Adam is utilized for optimization (Kingma and Ba, 2017). Our model is implemented in Pytorch (Paszke et al., 2019). See Appendix E.4 for further details.

In each of 10 experiments, we follow Sechidis et al. (2011) and split each dataset into stratified training (70%), validation (10%), and test (20%) sets with random seeds 0–9. After splitting, we impute missing values using the sample mean for continuous covariates and the mode for categorical covariates, based on the training set. We apply a z-score normalization to numerical features and encode categorical features using one-hot encoding. We train MENSA with early stopping on the validation set with a patience of 20 epochs. Concerning reporting, results are first computed separately for each random seed and event. Metrics are then averaged across events within each seed, producing one value per seed per model–dataset combination. We report the mean and standard deviation of these per-seed averages.

4.3. Results and takeaways

Here, we report results on the multi-event datasets. Concerning the single-event and competing-risks settings, these results are presented in Appendix G.

Overall performance. Table 2 presents the discrimination results of our proposed model and existing baselines on four multi-event datasets. Models marked with † in the tables are significantly and practically worse than MENSA (details of the statistical testing and minimum effect of interest (MEI) thresholds are provided in Appendix F). Overall, MENSA outperforms many deep learning baselines in discrimination tasks across multiple datasets and performs comparably to strong tree-based methods such as GBSA and RSF. Unlike DeepHit or Hierarchical, MENSA avoids unstable behavior (large variance or metric collapse). For these multi-event datasets, we also assess local discrimination performance using the local C-index, which captures the accuracy of within-patient event orderings and reflects the added value of a multi-event model. In cases where MENSA out-

4. The Neural Fine Gray (NFG) model is used exclusively for the competing-risks experiments, see Appendix G.2.

Table 2: Discrimination performance (mean ± SD.) between different models on multi-event benchmark datasets, averaged over 10 experiments. For all metrics, higher (↑) is better. Model types are abbreviated as SE = single event, CR = competing risks, and ME = multiple events. Bold indicates the top-3 models. † indicates models that are significantly and practically worse than MENSA, according to paired t-tests with Holm correction ($\alpha = 0.05$) and a minimum effect of interest (MEI) threshold of 1.0.

Model	Type	MIMIC-IV ($K = 3$)			Rotterdam ($K = 3$)			PRO-ACT ($K = 4$)			EBMT ($K = 5$)		
		CI_G	CI_L	AUC	CI_G	CI_L	AUC	CI_G	CI_L	AUC	CI_G	CI_L	AUC
CoxPH	SE	$70.9_{\pm 0.52}$	$90.3_{\pm 0.48}$	$74.3_{\pm 0.71}$	$69.7_{\pm 1.16}$	$83.6_{\pm 2.09}$†	$75.1_{\pm 1.48}$	$70.6_{\pm 0.47}$	$\mathbf{76.5}_{\pm 0.55}$	$76.7_{\pm 0.91}$	$54.9_{\pm 1.04}$	$66.3_{\pm 1.16}$†	$54.6_{\pm 1.10}$
CoxNet	SE	$71.0_{\pm 0.53}$	$90.4_{\pm 0.45}$	$74.4_{\pm 0.71}$	$69.8_{\pm 1.16}$	$83.8_{\pm 2.05}$†	$75.1_{\pm 1.49}$	$70.6_{\pm 0.46}$	$76.5_{\pm 0.57}$	$76.7_{\pm 0.89}$	$\mathbf{55.1}_{\pm 1.12}$	$67.2_{\pm 1.13}$†	$54.5_{\pm 1.17}$
Weibull	SE	$71.2_{\pm 0.54}$	$\mathbf{90.8}_{\pm 0.43}$	$74.8_{\pm 0.70}$	$69.7_{\pm 1.18}$	$87.9_{\pm 1.99}$†	$75.1_{\pm 1.45}$	$70.7_{\pm 0.49}$	$\mathbf{77.1}_{\pm 0.57}$	$76.7_{\pm 0.90}$	$54.9_{\pm 1.36}$	$\mathbf{68.0}_{\pm 0.97}$	$54.2_{\pm 1.35}$
GBSA	SE	$\mathbf{71.9}_{\pm 0.43}$	$\mathbf{91.4}_{\pm 0.51}$	$\mathbf{75.2}_{\pm 0.53}$	$\mathbf{75.3}_{\pm 1.62}$	$\mathbf{90.5}_{\pm 1.05}$	$71.2_{\pm 2.02}$†	$\mathbf{72.0}_{\pm 0.31}$	$75.6_{\pm 0.66}$	$\mathbf{77.6}_{\pm 0.44}$	$\mathbf{55.3}_{\pm 0.93}$	$67.2_{\pm 0.82}$†	$\mathbf{55.4}_{\pm 1.26}$
RSF	SE	$69.1_{\pm 0.48}$†	$\mathbf{90.8}_{\pm 0.40}$	$74.6_{\pm 0.56}$	$\mathbf{70.7}_{\pm 1.15}$	$84.9_{\pm 2.19}$†	$\mathbf{77.3}_{\pm 1.74}$	$\mathbf{71.1}_{\pm 0.51}$	$75.6_{\pm 0.83}$	$\mathbf{77.5}_{\pm 0.60}$	$55.0_{\pm 0.75}$	$64.6_{\pm 0.92}$†	$\mathbf{55.6}_{\pm 1.07}$
MTLR	SE	$70.2_{\pm 0.50}$†	$88.3_{\pm 0.95}$†	$74.4_{\pm 0.71}$	$69.8_{\pm 1.16}$	$82.8_{\pm 2.37}$†	$\mathbf{75.7}_{\pm 1.85}$	$70.7_{\pm 0.53}$	$\mathbf{76.5}_{\pm 0.53}$	$77.4_{\pm 0.83}$	$53.6_{\pm 1.04}$	$61.6_{\pm 2.38}$†	$55.4_{\pm 1.19}$
DeepSurv	SE	$\mathbf{71.8}_{\pm 0.55}$	$89.9_{\pm 0.49}$	$\mathbf{75.8}_{\pm 0.68}$	$70.0_{\pm 1.24}$	$84.6_{\pm 2.93}$†	$75.4_{\pm 1.75}$	$71.0_{\pm 0.59}$	$76.0_{\pm 0.57}$	$76.7_{\pm 1.00}$	$\mathbf{55.1}_{\pm 1.02}$	$67.6_{\pm 0.82}$	$54.9_{\pm 1.22}$
DeepHit	CR	$70.9_{\pm 0.75}$	$63.1_{\pm 9.74}$†	$75.1_{\pm 0.94}$	$66.5_{\pm 1.09}$†	$\mathbf{90.2}_{\pm 1.03}$	$74.2_{\pm 1.50}$	$64.9_{\pm 1.15}$†	$65.3_{\pm 3.35}$†	$73.4_{\pm 0.73}$†	$52.4_{\pm 1.29}$†	$60.4_{\pm 2.76}$†	$55.2_{\pm 1.46}$
DSM	CR	$70.5_{\pm 0.60}$	$88.1_{\pm 1.37}$†	$74.5_{\pm 0.83}$	$\mathbf{70.6}_{\pm 1.26}$	$86.0_{\pm 2.81}$†	$\mathbf{76.4}_{\pm 1.68}$	$70.8_{\pm 0.68}$	$76.0_{\pm 0.68}$	$\mathbf{77.4}_{\pm 0.76}$	$54.8_{\pm 1.00}$	$\mathbf{68.0}_{\pm 0.99}$	$55.3_{\pm 1.36}$
Hierarch.	ME	$65.2_{\pm 1.67}$†	$85.3_{\pm 1.73}$†	$48.7_{\pm 0.51}$†	$67.8_{\pm 2.06}$	$55.7_{\pm 8.11}$†	$50.0_{\pm 0.05}$†	$70.8_{\pm 0.60}$	$73.6_{\pm 1.42}$†	$57.3_{\pm 0.89}$†	$54.7_{\pm 1.33}$	$44.5_{\pm 1.12}$†	$53.5_{\pm 1.03}$†
MENSA (Ours)	ME	$\mathbf{71.4}_{\pm 0.29}$	$90.3_{\pm 0.54}$	$\mathbf{75.2}_{\pm 0.41}$	$69.4_{\pm 1.20}$	$\mathbf{90.1}_{\pm 1.55}$	$73.8_{\pm 2.31}$	$\mathbf{70.8}_{\pm 0.45}$	$76.4_{\pm 0.61}$	$77.2_{\pm 0.60}$	$55.1_{\pm 1.37}$	$\mathbf{68.8}_{\pm 0.66}$	$\mathbf{55.7}_{\pm 1.62}$

Table 3: Calibration performance (mean ± SD.) between different models on four multi-event benchmark datasets, averaged over 10 experiments. For IBS and mMAE, lower (↓) is better. For D-calibration, higher (↑) is better. D-calibration counts the number of experiments in which the respective model was D-calibrated. Bold indicates the top-3 models. † indicates models that are significantly and practically worse than MENSA, based on paired t-tests with Holm correction ($\alpha = 0.05$) and a predefined MEI of 1.0 for IBS and mMAE.

Model	Type	MIMIC-IV ($K = 3$)			Rotterdam ($K = 3$)			PRO-ACT ($K = 4$)			EBMT ($K = 5$)		
		IBS	mMAE	D-Cal	IBS	mMAE	D-Cal	IBS	mMAE	D-Cal	IBS	mMAE	D-Cal
CoxPH	SE	$15.3_{\pm 0.51}$	$10.8_{\pm 5.77}$	30/30	$19.6_{\pm 1.53}$	$23.2_{\pm 1.15}$†	20/20	$16.6_{\pm 0.39}$	$129.0_{\pm 3.31}$†	36/40	$20.4_{\pm 0.68}$	$48.6_{\pm 0.71}$	49/50
CoxNet	SE	$15.3_{\pm 0.52}$	$8.6_{\pm 0.16}$	30/30	$19.6_{\pm 1.54}$	$23.0_{\pm 1.11}$	20/20	$16.6_{\pm 0.40}$	$128.6_{\pm 3.23}$†	36/40	$20.3_{\pm 0.68}$	$\mathbf{48.3}_{\pm 0.70}$	49/50
Weibull	SE	$15.6_{\pm 0.42}$	$11.3_{\pm 0.14}$†	30/30	$19.2_{\pm 1.64}$	$23.1_{\pm 0.78}$	15/20	$16.7_{\pm 0.35}$	$220.7_{\pm 11.24}$†	36/40	$\mathbf{20.1}_{\pm 0.98}$	$59.4_{\pm 0.86}$	18/50
GBSA	SE	$16.7_{\pm 0.63}$†	$\mathbf{6.9}_{\pm 0.07}$	29/30	$19.3_{\pm 1.44}$	$24.1_{\pm 0.74}$†	20/20	$16.7_{\pm 0.42}$	$\mathbf{120.8}_{\pm 2.14}$	35/40	$20.4_{\pm 0.75}$	$\mathbf{44.7}_{\pm 0.44}$	50/50
RSF	SE	$16.6_{\pm 0.49}$†	$\mathbf{7.5}_{\pm 0.08}$	27/30	$\mathbf{18.6}_{\pm 1.29}$	$\mathbf{22.6}_{\pm 1.05}$	18/20	$\mathbf{16.3}_{\pm 0.41}$	$170.1_{\pm 22.18}$†	39/40	$\mathbf{18.9}_{\pm 1.25}$	$68.9_{\pm 1.00}$	50/50
MTLR	SE	$\mathbf{14.9}_{\pm 0.52}$	$9.7_{\pm 0.15}$†	30/30	$\mathbf{18.1}_{\pm 1.65}$	$23.0_{\pm 0.72}$	19/20	$\mathbf{16.4}_{\pm 0.42}$	$\mathbf{126.3}_{\pm 4.56}$	40/40	$\mathbf{18.9}_{\pm 1.25}$	$68.9_{\pm 1.00}$†	50/50
DeepSurv	SE	$\mathbf{14.9}_{\pm 0.51}$	$9.1_{\pm 0.22}$	30/30	$19.4_{\pm 1.47}$	$\mathbf{22.7}_{\pm 1.02}$	18/20	$\mathbf{16.3}_{\pm 0.42}$	$139.1_{\pm 5.27}$†	37/40	$20.5_{\pm 0.73}$	$\mathbf{45.8}_{\pm 0.63}$	49/50
DeepHit	CR	$32.7_{\pm 0.98}$†	$20.0_{\pm 0.31}$†	0/30	$21.4_{\pm 2.64}$†	$27.3_{\pm 0.79}$†	18/20	$25.6_{\pm 1.21}$†	$196.5_{\pm 4.00}$†	0/40	$21.1_{\pm 3.99}$	$74.3_{\pm 0.84}$†	50/50
DSM	CR	$16.0_{\pm 0.47}$†	$17.0_{\pm 1.92}$†	22/30	$\mathbf{18.3}_{\pm 1.54}$	$23.2_{\pm 0.76}$†	15/20	$16.5_{\pm 0.35}$	$231.5_{\pm 15.51}$†	38/40	$20.7_{\pm 1.46}$	$59.6_{\pm 0.82}$	17/50
Hierarch.	ME	$51.0_{\pm 1.16}$†	$101.9_{\pm 233.61}$†	0/30	$56.0_{\pm 5.09}$†	$43.5_{\pm 0.76}$†	0/20	$53.9_{\pm 1.86}$†	$280.5_{\pm 5.47}$†	0/40	$71.8_{\pm 7.84}$†	$92.2_{\pm 0.80}$†	0/50
MENSA (Ours)	ME	$\mathbf{15.1}_{\pm 0.59}$	$\mathbf{8.3}_{\pm 0.09}$	21/30	$18.8_{\pm 1.59}$	$\mathbf{22.6}_{\pm 0.68}$	18/20	$16.6_{\pm 0.41}$	$\mathbf{124.1}_{\pm 2.14}$	39/40	$22.0_{\pm 2.33}$	$63.1_{\pm 2.88}$	10/50

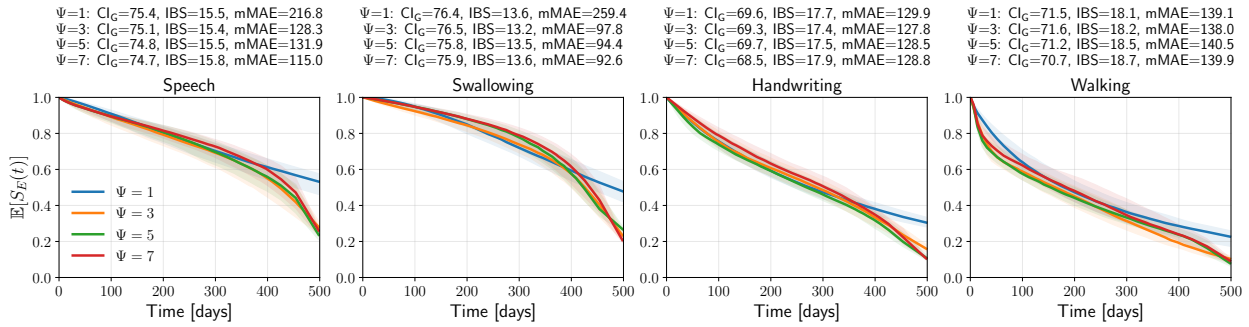

Figure 3: Population-level summaries of predicted survival functions across the PRO-ACT test set for each event. Each panel shows the median expected survival $\mathbb{E}[S_E(t)]$ with narrow (40–60th percentile) variability bands. Different colors correspond to MENSA models trained with varying numbers of Weibull distributions $\Psi \in \{1, 3, 5, 7\}$. The inset boxes report event-specific performance metrics (Global C-index, IBS, mMAE).

performs the baselines in local C-index, we attribute the improvement to its multi-state architecture and trajectory-based likelihood term. Table 3 summarizes calibration performance. MENSA maintains low IBS

Table 4: Ablation study comparing joint vs. independent training of MENSA on two multi-event datasets. Results are averaged over 10 experiments, with changes shown relative (%) to the baseline.

Dataset	Training	CI_G	CI_L	AUC	IBS	mMAE	D-Cal
Rotterdam	Joint	69.45±1.25	90.24±1.36	73.88±2.33	18.78±1.59	22.62±0.71	18/20
	Indep.	69.82±1.31 (+0.53)	87.81±1.74 (-2.69)	74.57±2.10 (+0.93)	18.84±1.67 (+0.32)	22.48±0.67 (-0.63)	20/20
PRO-ACT	Joint	70.87±0.50	76.07±0.51	77.12±1.20	16.60±0.52	124.37±3.44	36/40
	Indep.	70.98±0.64 (+0.15)	75.92±0.88 (-0.19)	77.29±0.75 (+0.23)	16.54±0.42 (-0.34)	122.78±2.33 (-1.27)	38/40

Table 5: Ablation study comparing training MENSA with/without the trajectory likelihood on two multi-event datasets. Results are averaged over 10 experiments, with changes shown relative (%) to the baseline.

Dataset	$\mathcal{L}_{trajectory}$	CI_G	CI_L	AUC	IBS	mMAE	D-Cal
Rotterdam	With	69.45±1.25	90.24±1.36	73.88±2.33	18.78±1.59	22.62±0.71	18/20
	Without	69.54±1.22 (+0.13)	89.94±1.45 (-0.34)	74.16±2.09 (+0.38)	18.79±1.72 (+0.09)	22.58±0.71 (-0.18)	18/20
EBMT	With	55.26±1.24	68.73±0.56	55.66±1.47	22.13±2.30	62.98±2.97	9/50
	Without	54.93±1.15 (-0.6)	68.39±0.56 (-0.5)	55.53±1.43 (-0.24)	22.36±2.59 (+1.04)	63.76±3.90 (+1.25)	14/50

and mMAE – often close to or better than classical survival methods (*e.g.*, CoxPH, RSF) – and exhibits comparable D-calibration to other deep learning variants (*e.g.*, DeepSurv and DSM). Compared to the Hierarchical model, our approach demonstrates consistently superior performance across all datasets.

Impact of number of distributions. Using functional decline prediction in patients with amyotrophic lateral sclerosis (ALS) as a test case (PRO-ACT dataset; four events: *Speaking*, *Swallowing*, *Handwriting*, and *Walking*; see Appendix E.1), we train MENSA to estimate event-specific population survival curves. Each curve represents the median survival (*i.e.*, event) probability $\mathbb{E}[S_E(t)]$ across all $N = 670$ patients under a given mixture configuration Ψ. Setting $\Psi = 1$ means that a single distribution models the time-to-event distribution for each event. Increasing Ψ introduces additional distributions, allowing the model to represent more complex and non-proportional risk structures. As shown in Figure 3, higher mixture complexity shifts the population survival downward across all events by the end of follow-up (about 500 days), reflecting a more pessimistic risk estimate. Quantitatively, this added flexibility lowers prediction error (mMAE: 183→118 days) and slightly improves calibration (IBS: 0.158→0.155) while maintaining stable discrimination (Global CI: ≈0.73). These results suggest that moderate mixture complexity ($\Psi \approx 3-5$) better captures disease progression in ALS without overfitting, improving time-to-event accuracy at the population level.

Computational analysis. We analyze the computational complexity of our method and several deep baselines[5]. For fairness, we configure all models with a default architecture (see Appendix E.3). The full results are presented in Appendix H. In the multi-event setting, all baselines (except the Hierarchical model) must be trained K times, whereas MENSA handles all events in a single forward pass. On a (hypothetical) dataset with 100 features and four events, MENSA requires 1.4× fewer parameters than DeepSurv, 2.5× fewer than DeepHit, 3.2× fewer than MTLR, 1.5× fewer than DSM, and 13× fewer than the Hierarchical model. Reductions in FLOP follow a similar trend, ranging from 1.5× (DeepSurv, DSM) and 2.5× (DeepHit) to 5.7× (MTLR) and 13× (Hierarchical).

Ablation studies. We perform two ablation studies evaluating (1) joint versus independent training, and (2) the trajectory-based likelihood term. Table 4 shows the results for training events jointly. Here, *jointly* refers to learning all event-specific parameters within a single shared architecture. We see in both datasets that the global metrics (*e.g.*, global C-index, AUC and mMAE) increase slightly when training independently, but the local C-index (within-patient event ordering) drops significantly in Rotterdam (-2.69% decrease) and PRO-ACT (-0.19% decrease). Table 5 presents the results for the trajectory term. In Rotterdam, it improves the local C-index with only small shifts in the global C-index, AUC and mMAE. In EBMT, the trajectory term consistently improves performance.

5. We report computational metrics only for the deep learning models; FLOPs and parameter counts are not well-defined or directly comparable for classical methods.

5. Discussion

5.1. Limitations

This work has limitations. First, MENSA assumes conditionally independent censoring, which may be violated if relevant covariates that influence both censoring and event times are unobserved. Second, as a parametric model, MENSA relies on predefined event-time distributions. While this choice enhances interpretability and stability, it can reduce flexibility and require larger datasets for reliable parameter estimation. In contrast, semiparametric approaches such as CoxPH and DeepSurv make fewer distributional assumptions and can sometimes generalize better in small-sample settings. Third, the current implementation of MENSA is limited to static baseline covariates and does not yet incorporate time-varying covariates or longitudinal measurements, which are common in many clinical applications.

5.2. Application

MENSA provides particular value in clinical contexts where the *local ordering* of events is most relevant. For example, in the PRO-ACT dataset, the goal is to predict the time to different types of functional decline in ALS patients as measured by the ALSFRS-R scale (see Appendix E.1). A model with a high local C-index can accurately rank event occurrences within a single patient, enabling clinically meaningful questions such as: *Which physical functions am I likely to lose first, and in what order?* This prioritization of local ordering is especially useful in neurodegenerative diseases like ALS, where functional decline follows distinct trajectories. Prior studies have shown, for instance, that bulbar-onset ALS typically affects speech functionality before gait and walking ability (Feldman et al., 2022). Beyond ALS, multi-event modeling is crucial in many diseases where multiple outcomes unfold over time. In chronic heart failure, for example, patients may experience repeated hospitalizations, arrhythmic events, and eventual cardiac death, each with complex temporal dependencies. In oncology (*e.g.*, the SEER cohort), treatment response, relapse, and metastasis form interdependent event pathways that conventional single-event models cannot adequately capture. Similarly, in hematologic and transplant settings (*e.g.*, the EBMT cohort), graft-versus-host disease, relapse, and non-relapse mortality compete and co-occur, requiring models that jointly estimate their risks.

Regarding interpretability, practitioners can extract feature importances for each event-specific prediction. Model-agnostic methods tailored to survival analysis, such as SurvLIME (Kovalev et al., 2020) and SurvSHAP (Krzyziński et al., 2023), can be applied to MENSA's event-specific outputs to quantify the contribution of individual covariates to each predicted event time.

5.3. Conclusion

We have presented MENSA, a new method for multi-event survival analysis. MENSA jointly learns flexible time-to-event distributions for multiple events, which may be competing or co-occurring. Across four multi-event survival datasets, MENSA demonstrated solid discrimination and calibration performance, outperforming many deep learning baselines while remaining computationally efficient in the multi-event setting. Compared to the Hierarchical survival model – the main existing deep learning approach for multi-event survival analysis – MENSA achieved superior results across all evaluation metrics. Ablation studies further showed that training all events within a shared architecture improves within-patient event ordering, as reflected by higher local C-index values, and that the proposed trajectory-based likelihood term further enhances this metric.

Acknowledgments

†Data used in the preparation of this article were obtained from the Pooled Resource Open-Access ALS Clinical Trials (PRO-ACT) Database. As such, the following organizations and individuals within the PRO-ACT Consortium contributed to the design and implementation of the PRO-ACT Database and/or provided data, but did not participate in the analysis of the data or the writing of this report: ALS Therapy Alliance, Cytokinetics, Inc., Amylyx Pharmaceuticals, Inc., Knopp Biosciences, Neuraltus Pharmaceuticals, Inc., Neurological Clinical Research Institute, MGH, Northeast ALS Consortium, Novartis, Orion Corporation, Prize4Life Israel, Regeneron Pharmaceuticals, Inc., Sanofi, Teva Pharmaceutical Industries, Ltd., The ALS Association.

This research received support from the Natural Science and Engineering Research Council of Canada (NSERC), the Canadian Institute for Advanced Research (CIFAR), and the Alberta Machine Intelligence Institute (Amii).

References

Per K. Andersen and Niels Keiding. Multi-state models for event history analysis. *Statistical Methods in Medical Research*, 11(2):91–115, 2002.

Gregory T. Armstrong, Toana Kawashima, Wendy Leisenring, Kayla Stratton, Marilyn Stovall, Melissa M. Hudson, Charles A. Sklar, Leslie L. Robison, and Kevin C. Oeffinger. Aging and risk of severe, disabling, life-threatening, and fatal events in the childhood cancer survivor study. *Journal of Clinical Oncology*, 32(12):1218, 2014.

Nazem Atassi, James Berry, Amy Shui, Neta Zach, Alexander Sherman, Ervin Sinani, Jason Walker, Igor Katsovskiy, David Schoenfeld, Merit Cudkowicz, and Melanie Leitner. The PRO-ACT database: design, initial analyses, and predictive features. *Neurology*, 83(19):1719–1725, 2014.

J. M. Cedarbaum, N. Stambler, E. Malta, C. Fuller, D. Hilt, B. Thurmond, and A. Nakanishi. The ALSFRS-R: A Revised ALS Functional Rating Scale That Incorporates Assessments of Respiratory Function. *Journal of Neurological Sciences*, 169(1-2):13–21, 1999.

George H. Chen. An Introduction to Deep Survival Analysis Models for Predicting Time-to-Event Outcomes. *Foundations and Trends in Machine Learning*, 17(6):921–1100, 2024.

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

Cameron Davidson-Pilon. lifelines: survival analysis in Python. *Journal of Open Source Software*, 4(40): 1317, 2019.

Takeshi Emura and Yi-Hau Chen. *Analysis of Survival Data with Dependent Censoring: Copula-Based Approaches*. Springer, 2018.

Eva L. Feldman, Stephen A. Goutman, Stefan Petri, Letizia Mazzini, Michael G. Savelieff, Pamela J. Shaw, and Gen Sobue. Amyotrophic Lateral Sclerosis. *The Lancet*, 400(10360):1363–1380, 2022.

Jason P. Fine and Robert J. Gray. A proportional hazards model for the subdistribution of a competing risk. *Journal of the American statistical association*, 94(446):496–509, 1999.

Evelyn Fix and Jerzy Neyman. A Simple Stochastic Model of Recovery, Relapse, Death and Loss of Patients. *Human Biology*, 23(3):205–241, Sep 1951.

James Gareth, Witten Daniela, Hastie Trevor, and Tibshirani Robert. *An introduction to statistical learning: with applications in R*. Springer, 2 edition, 2021.

Malka Gorfine, Richard J. Cook, Per Kragh Andersen, Terry M. Therneau, Pierre Joly, Hein Putter, Maja Pohar Perme, and Michal Abrahamowicz. An Overview and Recent Developments in the Analysis of Multistate Processes. *arXiv preprint arXiv:2502.06492*, 2025.

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.

Stefan Groha, Sebastian M. Schmon, and Alexander Gusev. A general framework for survival analysis and multi-state modelling. *arXiv preprint arXiv:2006.04893*, 2020.

Mehak Gupta, Brennan Gallamoza, Nicolas Cutrona, Pranjal Dhakal, Raphael Poulain, and Rahmatollah Beheshti. An Extensive Data Processing Pipeline for MIMIC-IV. In *Proceedings of the 2nd Machine Learning for Health symposium*, volume 193, pages 311–325, 2022.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective Ways to Build and Evaluate Individual Survival Distributions. *Journal of Machine Learning Research*, 21(1):1–63, 2020.

Frank E. Harrell Jr, Kerry L. Lee, and Daniel B. Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387, 1996.

Jin-Jian Hsieh and Jian-Lin Wang. Quantile residual life regression based on semi-competing risks data. *Journal of Applied Statistics*, 45(10):1770–1780, 2018.

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, Chang Ho Yoon, Brian Tom, and Jessica Barrett. Neural Fine-Gray: Monotonic neural networks for competing risks. In *Proceedings of the Conference on Health, Inference, and Learning*, volume 209, pages 379–392, 2023.

Fei Jiang and Sebastien Haneuse. A semi-parametric transformation frailty model for semi-competing risks survival data. *Scandinavian Journal of Statistics*, 44(1):112–129, 2017.

Alistair E. W. Johnson, Lucas Bulgarelli, Lu Shen, Alvin Gayles, Ayad Shammout, Steven Horng, Tom J. Pollard, Sicheng Hao, Benjamin Moody, Brian Gow, Li wei H. Lehman, Leo A. Celi, and Roger G. Mark. MIMIC-IV, a freely accessible electronic health record dataset. *Scientific Data*, 10(1): 1–9, 2023.

Jared Katzman, Uri Shaham, Jonathan Bates, Alexander Cloninger, 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.

Sejin Kim, Michal Kazmierski, and Benjamin Haibe-Kains. Deep-CR MTLR: a Multi-Modal Approach for Cancer Survival Prediction with Competing Risks. In *Proceedings of AAAI Spring Symposium on Survival Prediction - Algorithms, Challenges, and Applications*, volume 146, pages 223–231, 2021.

Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*, 2017.

John P. Klein and Melvin L. Moeschberger. *Survival analysis: techniques for censored and truncated data*. Springer Science & Business Media, 2006.

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.

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela van der Schaar. DeepHit: A Deep Learning Approach to Survival Analysis With Competing Risks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1):1145–1152, Apr. 2018.

Yan Li, Jie Wang, Jieping Ye, and Chandan K. Reddy. A Multi-Task Learning Formulation for Survival Analysis. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 1715–1724, 2016.

Christian M. Lillelund, Martin Magris, and Christian F. Pedersen. Efficient Training of Probabilistic Neural Networks for Survival Analysis. *IEEE Journal of Biomedical and Health Informatics*, 29 (9):6157–6166, 2025.

Intae Moon, Stefan Groha, and Alexander Gusev. SurvLatent ODE: A Neural ODE based time-to-event model with competing risks for longitudinal data improves cancer-associated Venous Thromboembolism (VTE) prediction. In *Machine Learning for Healthcare Conference*, pages 800–827. PMLR, 2022.

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, 2021.

Chirag Nagpal, Willa Potosnak, and Artur Dubrawski. auton-survival: an Open-Source Package for Regression, Counterfactual Estimation, Evaluation and Phenotyping with Censored Time-to-Event Data. *arXiv preprint arXiv:2204.07276*, 2022.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, volume 32, pages 8026–8037, 2019.

Sebastian Pölsterl. scikit-survival: A Library for Time-to-Event Analysis Built on Top of scikit-

learn. *Journal of Machine Learning Research*, 21 (212):1–6, 2020.

Shi-ang Qi, Neeraj Kumar, Jian-Yi Xu, Jaykumar Patel, Sambasivarao Damaraju, Grace Shen-Tu, and Russell Greiner. Personalized breast cancer onset prediction from lifestyle and health history information. *PLOS ONE*, 17(12):e0279174, 2022.

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. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 28244–28276, 2023a.

Shi-ang Qi, Weijie Sun, and Russell Greiner. SurvivalEVAL: A Comprehensive Open-Source Python Package for Evaluating Individual Survival Distributions. In *Proceedings of the 2023 AAAI Fall Symposia*, volume 2, pages 453–457, 2023b.

Greg Ridgeway. The State of Boosting. *Computing Science and Statistics*, 31:172–181, 1999.

Lynn A. Gloeckler Ries, Marsha E. Reichman, Denise Riedel Lewis, Benjamin F. Hankey, and Brenda K. Edwards. Cancer survival and incidence from the Surveillance, Epidemiology, and End Results (SEER) program. *Oncologist*, 8(6):541–552, 2003.

Hagai Rossman, Ayya Keshet, and Malka Gorfine. PyMSM: Python package for Competing Risks and Multi-State models for Survival Data. *Journal of Open Source Software*, 7(78):4566, 2022.

Patrick Royston and Douglas G. Altman. External validation of a Cox prognostic model: principles and methods. *BMC Medical Research Methodology*, 13:33, 2013.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the Stratification of Multi-label Data. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158, 2011.

Noah Simon, Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent. *Journal of Statistical Software*, 39(5), 2011.

Richard M. Simon, Edward L. Korn, Lisa M. McShane, Michael D. Radmacher, George W. Wright, and Yingdong Zhao. *Design and analysis of DNA microarray investigations*, volume 209. Springer, 2003.

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, volume 25, pages 2951—2959, 2012.

Scott D. Solomon, Adel R. Rizkala, Jianjian Gong, Wenyan Wang, Inder S. Anand, Junbo Ge, Carolyn S. P. Lam, Aldo P. Maggioni, Felipe Martinez, Milton Packer, Marc A. Pfeffer, Burkert Pieske, Margaret M. Redfield, Jean L. Rouleau, Dirk J. Van Veldhuisen, Faiez Zannad, Michael R. Zile, Akshay S. Desai, Victor C. Shi, Martin P. Lefkowitz, and John J. V. McMurray. Angiotensin receptor neprilysin inhibition in heart failure with preserved ejection fraction: rationale and design of the PARAGON-HF trial. *JACC: Heart Failure*, 5(7): 471–482, 2017.

Donna Tjandra, Yifei He, and Jenna Wiens. A Hierarchical Approach to Multi-Event Survival Analysis. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(1):591–599, 2021.

Lu Wang, Yan Li, Jiayu Zhou, Dongxiao Zhu, and Jieping Ye. Multi-task Survival Analysis. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 485–494, 2017.

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*, 2022.

Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning Patient-Specific Cancer Survival Distributions as a Sequence of Dependent Regressors. In *Advances in Neural Information Processing Systems*, volume 24, pages 1845—1853, 2011.

Appendix A. Notation

Table 6 provides a list of notations used in this work.

Table 6: Table of notation.

Symbol	Definition
$c_p^{(i)}$	Censoring time for instance i in state p
$e_p^{(i)}$	True event/transition time for instance i in state p
$t_p^{(i)}$	Observed time for instance i in state p: $\min(e_p^{(i)}, c_p^{(i)})$
$\delta_p^{(i)}$	Event indicator: $\delta_p^{(i)} = \mathbb{1}[e_p^{(i)} \leq c_p^{(i)}]$
$\mathcal{D}$	A time-to-event dataset
N	Number of individuals/instances in a dataset
d	Number of features/covariates in a dataset
$\boldsymbol{x}^{(i)}$	Covariate vector for instance i, $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$
$h_0(t)$	Baseline hazard function
$h(t \mid \boldsymbol{x}^{(i)})$	Hazard function given covariates $\boldsymbol{x}^{(i)}$
$K \in \mathbb{N}_+$	Number of distinct events
$P \in \mathbb{N}_+$	Number of distinct states
$\mathcal{K}$	Set of possible events, $\lvert\mathcal{K}\rvert = K$
$\mathcal{P}$	Set of possible states, $\lvert\mathcal{P}\rvert = P$
$\mathcal{T}$	Set of event trajectories over $\mathcal{K}$
$\mathcal{L}$	Loss/objective function used during training
L	Number of training epochs
κ	Learning rate for optimization
λ	Weighting hyperparameter for the trajectory loss
$\Psi \in \mathbb{N}_+$	Number of mixture distributions
$G(t) \in \mathcal{P}$	Latent state process (*e.g.*, Markov process) over state space $\mathcal{P}$
$f(\boldsymbol{\theta}, \boldsymbol{x}^{(i)})$	Risk or output function given parameters $\boldsymbol{\theta}$ and input $\boldsymbol{x}^{(i)}$
$S_E(t)$	Event survival function: $S_E(t) = \Pr(T_E > t)$
$f_E(t)$	Event density function: $f_E(t) = \frac{d}{dt}(1 - S_E(t))$
$F_E(t)$	Event cumulative distribution function: $F_E(t) = 1 - S_E(t)$

Appendix B. Multi-state analysis

Single event. Survival analysis typically models the time until a single event takes place (Gareth et al., 2021, Ch. 11). We can represent such problems using a two-state model, as in Figure 4.

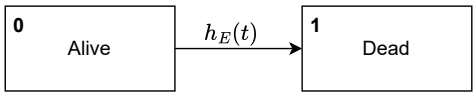

Figure 4: A single-event model with a single transition function.

In this simplest case, there is one transient state (0, *e.g.*, "alive") and one absorbing state (1, *e.g.*, "dead"). Once an instance enters an absorbing state, no further transitions are possible. The observation for a given instance will in the simplest form consist of a random variable, T_E, representing the time from a given origin ($t = 0$) to the occurrence of the event. The distribution of T_E can be characterized by the probability distribution function $F_E(t) = \Pr(T_E \leq t)$ or, equivalently, by the survival function $S_E(t) = 1 - F_E(t) = \Pr(T_E > t)$. Note that $S_E(t)$ and $F_E(t)$, respectively, correspond to the probabilities

of being in state 0 or 1 at time t. In continuous time, the distribution of T_E is commonly characterized by the *hazard rate function*:

$$h_E(t) = -d \log S_E(t)/dt = \lim_{\Delta t \to 0} \frac{\Pr(t < T_E \leq t + \Delta t | T_E > t)}{\Delta t}, \tag{10}$$

where $S_E(t)$ is the *survival function*:

$$S_E(t) = \exp\left(-\int_0^t h_E(u)\, du\right). \tag{11}$$

Here, $h_E(t)$ is the transition intensity from state 0 to 1, *i.e.*, the instantaneous risk of going from state 0 to state 1 at time t. We can extend this framework to multiple state transitions using *marked point processes* (Andersen and Keiding, 2002). Let $G(t)$, $t \in \mathbb{T}$, $\mathbb{T} = [0, t_{\max}]$ denote a stochastic process with a finite state space $\mathcal{P} = \{0, ..., P-1\}$ and with right-continuous sample paths: $G(t+) = G(t)$. The process has an initial distribution $\pi_p(0) = [\Pr(G(0) = p) \mid p \in \mathcal{P}]$. A multi-state process $G(\cdot)$ generates a history $\mathcal{G}_t$, which consists of observations of the process in the interval $[0, t_{\max}]$ – *e.g.*, the patient is healthy at the start of the study ($\mathcal{G}_0$), but becomes ill 10 days in ($\mathcal{G}_{10}$), before the study terminates after 100 days ($t_{\max} = 100$). Given this history, we can define the so-called transition probabilities between two distinct events by:

$$\Pr_{p \to q}(s, t) = \Pr(G(t) = q \mid G(s) = p, \mathcal{G}_{s^-}), \tag{12}$$

for $p, q \in \mathcal{P}, s, t \in \mathbb{T}, s \leq t$ and transition intensities by the derivatives, assuming such exist:

$$h_{p \to q}(t) = \lim_{\Delta t \to 0} \frac{\Pr_{p \to q}(t, t + \Delta t)}{\Delta t}. \tag{13}$$

Some transition intensities may be zero for all t between two states, *i.e.*, $h_{p \to q}(t) = 0 \; \forall t$ and $\forall p, q \in \mathcal{P}$. A state $p \in \mathcal{P}$ is absorbing if, for all $t \in \mathbb{T}, p \in \mathcal{P}, p \neq q, h_{p \to q}(t) = 0$; otherwise p is transient. The state probabilities $\pi_p(t) = \Pr(G(t) = p)$ are given by:

$$\pi_p(t) = \sum_{q \in \mathcal{P}} \pi^q(0) \Pr_{q \to p}(0, t). \tag{14}$$

Note that $\Pr_{p \to q}(\cdot, \cdot)$ and $h_{p \to q}(\cdot)$ depend on both the probability measure $\Pr(\cdot)$, and on the history.

Competing risks. In competing risks, multiple events (*i.e.*, absorbing states) exist, but only one event of $k = 1, \ldots, K$ different types can occur for each instance, illustrated in Figure 5.

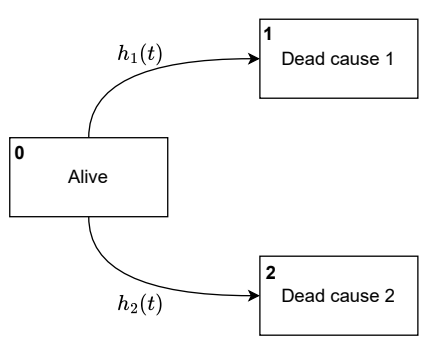

Figure 5: A competing-risks model for mortality from two causes.

In our multi-state framework, a competing-risks model has one transient state, aka state 0, and a number of absorbing states. Entering an absorbing state corresponds to experiencing event k, for $k \in \{1, \ldots, K\}$.

Thus, we have P total states and $K = P - 1$ events. In this model, if an instance is censored before experiencing any events, it simply remained in the transient state and is thus event-free. Using our state notation, the transition intensities $h_p(t)$ for $p \in \{1, \ldots, P-1\}$ are given by the cause-specific hazard function (omitting state 0):

$$h_p(t) = \lim_{\Delta t \to 0} \frac{\Pr(\text{Transition to state } p \text{ at } t + \Delta t \mid T_p \geq t)}{\Delta t}, \tag{15}$$

where T_p is the random variable corresponding to the time of the transition to state p. There are only edges originating from state 0, the only transient state of the model, *i.e.*, $h_{p \to q}(t) = 0$ for all $p \neq 0$ and all q. The transition probabilities are given by the survival function:

$$\Pr_{0 \to 0}(0, t) = S_0(t) = \Pr(T_0 > t) = \exp\left(- \int_0^t \sum_{p=1}^{P-1} h_p(u)\, du \right), \tag{16}$$

and the cumulative incidence function (CIF) for transitioning from state 0 to any state $p \neq 0$:

$$\Pr_{0 \to p}(0, t) = \int_0^t S_0(u-)\, h_p(u)\, du, \quad p = 1, \ldots, P-1. \tag{17}$$

Multiple events. In multi-event scenarios, we can observe multiple transitions (events) for each instance i, thus the transitions (or event occurrences) are not necessarily mutually exclusive, as in Figure 6.

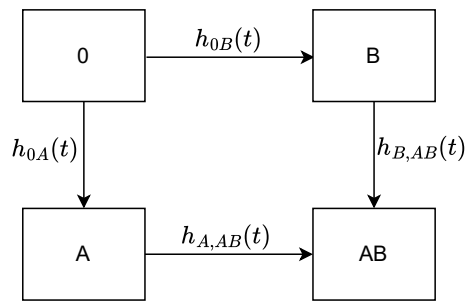

Figure 6: A multi-event model describing the joint behavior of A and B.

Such a model can, for example, describe the joint behavior of two events A and B; if $h_{0B} = h_{A,AB}$ but $h_{0A} \neq h_{B,AB}$, A is called *locally dependent* on B but B is not locally dependent on A. The temporal ordering of events enables an asymmetric notion of dependence, providing more information for causal inference than the symmetric association concepts used in cross-sectional studies.

Appendix C. Survival likelihood

In the standard survival analysis setting, considering a single event of interest, we define the survival time T_E and censoring time T_C as follows:

$$\begin{aligned} T_E = t^{(i)} \text{ and } T_C \geq t^{(i)} \quad &\text{if } \delta^{(i)} = 1, \\ T_E > t^{(i)} \text{ and } T_C = t^{(i)} \quad &\text{if } \delta^{(i)} = 0. \end{aligned}$$

Combining these cases, the likelihood for the i-th observation is:

$$\mathcal{L}^{(i)} = \Pr(T_E = t^{(i)}, T_C \geq t^{(i)} \mid x^{(i)})^{\delta^{(i)}} \Pr(T_E \geq t^{(i)}, T_C = t^{(i)} \mid x^{(i)})^{1-\delta^{(i)}} \tag{18}$$

Under the assumption of independent censoring (Emura and Chen, 2018), we have:

$$\begin{aligned}\mathcal{L}^{(i)} &= \left[\Pr(T_E = t^{(i)} \mid x^{(i)}) \Pr(T_C \geq t^{(i)} \mid x^{(i)})\right]^{\delta^{(i)}} \left[\Pr(T_E \geq t^{(i)} \mid x^{(i)}) \Pr(T_C = t^{(i)} \mid x^{(i)})\right]^{1-\delta^{(i)}} \\ &= \left[f_E(t^{(i)} \mid x^{(i)})\, S_C(t^{(i)} \mid x^{(i)})\right]^{\delta^{(i)}} \left[S_E(t^{(i)} \mid x^{(i)})\, f_C(t^{(i)} \mid x^{(i)})\right]^{1-\delta^{(i)}} \\ &= \left[f_E(t^{(i)} \mid x^{(i)})^{\delta^{(i)}} S_E(t^{(i)} \mid x^{(i)})^{1-\delta^{(i)}}\right] \left[f_C(t^{(i)} \mid x^{(i)})^{1-\delta^{(i)}} S_C(t^{(i)} \mid x^{(i)})^{\delta^{(i)}}\right] \end{aligned} \tag{19}$$

where $S_E(t \mid x^{(i)}) = \Pr(T_E \geq t \mid x^{(i)})$, $f_E(t \mid x^{(i)}) = -dS_E(t \mid x^{(i)})/dt$, $S_C(t \mid x^{(i)}) = \Pr(T_C \geq t \mid x^{(i)})$, and $f_C(t \mid x^{(i)}) = -dS_C(t \mid x^{(i)})/dt$. Under the non-informative censoring assumption, the term $f_C(t^{(i)} \mid x^{(i)})^{1-\delta^{(i)}} S_C(t^{(i)} \mid x^{(i)})^{\delta^{(i)}}$ is unrelated to the likelihood for the event times and can simply be ignored (Emura and Chen, 2018). Therefore, the likelihood function can be written as:

$$\mathcal{L} = \prod_{i=1}^{N} f_E(t^{(i)} \mid \boldsymbol{x}^{(i)})^{\delta^{(i)}} S_E(t^{(i)} \mid \boldsymbol{x}^{(i)})^{1-\delta^{(i)}}, \tag{20}$$

where $f_E(t^{(i)} \mid x^{(i)})$ is the density function of the event time and $S_E(t^{(i)} \mid x^{(i)})$ is the survival function of the event time. Note how the censoring terms $f_C(t^{(i)})$ and $S_C(t^{(i)})$ are absent in (20). If the event is observed, the likelihood contribution is the probability density of the event occurring at that time. If the observation is censored, the likelihood contribution is the probability of surviving beyond that time. The overall likelihood is the product of these contributions across all instances.

Appendix D. Evaluation metrics

Harrell's C-index: The concordance index (C-index or CI) measures the discriminative performance of a survival model by calculating the proportion of concordant pairs among all comparable pairs. A pair is considered comparable if we can determine who has the event first. It is defined as (Harrell Jr et al., 1996):

$$\text{C-index} = \frac{\sum_{i,j \in \mathcal{D}} \mathbb{1}[t^{(i)} < t^{(j)}] \cdot \mathbb{1}[\eta^{(i)} > \eta^{(j)}] \cdot \delta^{(i)}}{\sum_{i,j \in \mathcal{D}} \mathbb{1}[t^{(i)} < t^{(j)}] \cdot \delta^{(i)}} \tag{21}$$

where $\eta^{(i)}$ and $\eta^{(j)}$ represent risk scores for individuals i and j, respectively.

Global C-index: The global C-index is computed by averaging C-index scores across multiple instances for each event, and was proposed by Tjandra et al. (2021). It thus provides an overall measure of a model's discriminative performance in multiple events scenarios. This metric evaluates global consistency, *i.e.*, how well does the model discriminate instances *within* an event. It is defined as:

$$\text{Global C-index} = \frac{\sum_{k \in \mathcal{K}} \sum_{i,j \in \mathcal{D}_k} \mathbb{1}[t^{(i)} < t^{(j)}] \cdot \mathbb{1}[\eta^{(i)} > \eta^{(j)}] \cdot \delta^{(i)}}{\sum_{k \in \mathcal{K}} \sum_{i,j \in \mathcal{D}_k} \mathbb{1}[t^{(i)} < t^{(j)}] \cdot \delta^{(i)}}, \tag{22}$$

where $\mathcal{K}$ represents the set of all distinct events, and the index k corresponds to the k-th specific event.

Local C-index: The local C-index is computed by averaging C-index scores across multiple events for each instance, and was proposed by Tjandra et al. (2021). This metric evaluates local consistency, *i.e.*, how well does the model discriminate events *within* an individual. It is defined as:

$$\text{Local C-index} = \frac{\sum_{i \in N} \sum_{k_1, k_2 \in \mathcal{K}^{(i)}} \mathbb{1}[t_{k_1} < t_{k_2}] \cdot \mathbb{1}[\eta_{k_1} > \eta_{k_2}] \cdot \delta_{k_1}}{\sum_{i \in N} \sum_{k_1, k_2 \in \mathcal{K}^{(i)}} \mathbb{1}[t_{k_1} < t_{k_2}] \cdot \delta_{k_1}}, \tag{23}$$

where $\mathcal{K}^{(i)}$ represents the set of all K different events for instance i, and η_{k1} and η_{k2} denote the risk scores associated with the two events, respectively.

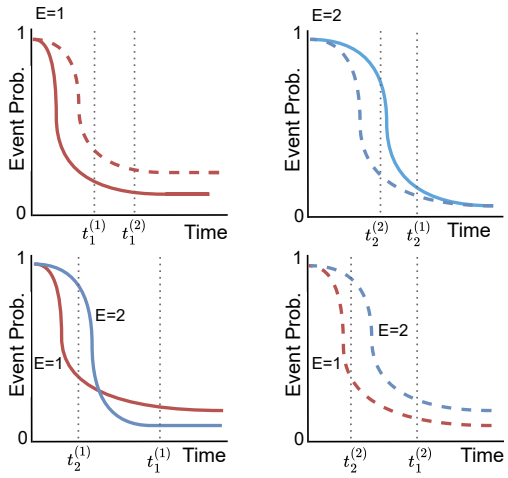

Figure 7: Global and local consistency. Red/blue lines represent survival functions $S_1(t)$ and $S_2(t)$ for event 1 and 2, respectively. Solid/dashed lines represent individuals $i = 1, 2$, respectively. Top-left: globally consistent at $t_1^{(1)}$ and $t_1^{(2)}$ since $S_1^{(1)}(t) < S_2^{(2)}(t)$ both times. Top-right: globally consistent at $t_2^{(2)}$ and $t_2^{(1)}$ since $S_2^{(2)}(t) < S_2^{(1)}(t)$ both times. Bottom-left: locally inconsistent at $t_2^{(1)}$ since $S_2^{(1)}(t) > S_1^{(1)}(t)$ and at $t_1^{(1)}$ since $S_1^{(1)}(t) > S_2^{(1)}(t)$. Bottom-right: locally inconsistent at $t_2^{(2)}$, since $S_2^{(2)}(t) > S_1^{(2)}(\mathrm{t})$.

Figure 7 shows a visual example of global and local consistency with respect to the survival function. Global consistency is satisfied between instances if the instance-specific curves align with the observed events. Local consistency is satisfied within an instance if the event-specific curves align with the observed events.

BS/IBS: The Brier Score (BS) is defined as the mean squared difference between the predicted survival curve and the Heaviside step function of the observed event. The integrated Brier Score (Graf et al., 1999) (IBS) aggregates the Brier Scores across multiple time points to provide a single measure of model performance. We use inverse probability weighting (IPCW) to handle censored events. The BS is defined as:

$$\mathrm{BS}(t^*) = \frac{1}{N}\sum_{i\in\mathcal{D}}\left[\frac{S(t^* \mid \mathbf{x}^{(i)})^2 \cdot \mathbb{1}[t^{(i)} \le t^*, \delta^{(i)} = 1]}{S_C(t^{(i)})} + \frac{(1 - S(t^* \mid \mathbf{x}^{(i)}))^2 \cdot \mathbb{1}[t^{(i)} > t^*]}{S_C(t^*)}\right], \tag{24}$$

which is the mean square error between observed survival status and survival probability at time t^*, and $S_C(t^*)$ is the non-censoring probability at time t^*. The IBS is defined as:

$$\mathrm{IBS} = \frac{1}{N}\sum_{i\in\mathcal{D}}\frac{1}{t_{\max}} \cdot \int_0^{t_{\max}} \mathrm{BS}(t)\, dt, \tag{25}$$

where $t_{\max}$ is the maximum observed time.

MAE: The mean absolute error (MAE) is the absolute difference between the predicted and actual survival times. Given an individual survival distribution, $S(t \mid \boldsymbol{x}_i) = \Pr(T > t \mid \boldsymbol{x}_i)$, we calculate the predicted survival time $\hat{t}^{(i)}$ using the median time of the predicted survival distribution (Qi et al., 2023b):

$$\hat{t}^{(i)} \;=\; \text{median}\,(S(t \mid \boldsymbol{x}_i)) = S^{-1}(\tau = 0.5 \mid \boldsymbol{x}_i), \tag{26}$$

$$\text{MAE}\,(\hat{t}_i, t_i, \delta_i = 1) \;=\; \mid t_i - \hat{t}_i \mid. \tag{27}$$

For censored individuals, we calculate the marginal MAE (mMAE), as proposed by Haider et al. (2020):

$$\text{mMAE} = \frac{1}{\sum_{i=1}^{N} \omega^{(i)}} \sum_{i=1}^{N} \omega^{(i)} \left| [(1-\delta^{(i)}) \cdot e_{\text{m}}(t^{(i)}) + \delta^{(i)} \cdot t^{(i)}] - \hat{t}^{(i)} \right|,$$
$$\text{where} \quad e_{\text{m}}(t^{(i)}) = \begin{cases} t^{(i)} + \frac{\int_{t^{(i)}}^{\infty} S_{\text{KM}}(t)dt}{S_{\text{KM}}(t)} & \text{if} \quad \delta^{(i)} = 0 \\ t^{(i)} & \text{if} \quad \delta^{(i)} = 1 \end{cases},$$
$$\text{and} \quad \omega^{(i)} = \begin{cases} 1 - S_{\text{KM}}(t^{(i)}) & \text{if} \quad \delta^{(i)} = 0 \\ 1 & \text{if} \quad \delta^{(i)} = 1 \end{cases}. \tag{28}$$

D-calibration: Distribution calibration (D-calibration) (Haider et al., 2020) measures the calibration performance of $S(t)$, expressing to what extent the predicted probabilities can be trusted. We assess this using a Pearson's chi-squared goodness-of-fit test. For any probability interval $[a, b] \in [0, 1]$, we define $D_m(a, b)$ as the group of individuals in the dataset D whose predicted probability of event is in the interval $[a, b]$ (Qi et al., 2023a). A model is D-calibrated if the proportion of individuals $|D_m(a, b)|/|D|$ is statistically similar to the amount $b - a$. In the results, we count the number of times a model is D-calibrated (*i.e.*, $p > 0.05$) for each event over 10 experiments.

Appendix E. Experimental details

E.1. Datasets and preprocessing

Table 7 shows an overview of the survival datasets used in this work.

Table 7: Overview of the raw datasets. Symbols indicate single-event ($*$), competing-risks ($\dagger$) or multi-event ($\ddagger$) task. Number of events (K) is denoted for the respective task. In SEER, "D", "BC" and "HF" denote the death event, death by breast cancer and death by heart failure. In MIMIC-IV, "A", "S", and "D" denote the ARF, shock and death events. In Rotterdam, "R" denotes relapse and "D" denotes death. In PRO-ACT, "SP", "SW", "HA" and "WA" denote the speech, swallowing, handwriting and walking events. In EBMT, "R", "AE", "RAE", "REL" and "D" denote the recovery, adverse events, recovery and adverse events, relapse, and death events.

Dataset	#Instances (N)	#Features (d)	#Events (K)	Max t	Event distribution (%)
SEER$*\dagger$	19,246	17	1/2	121	D: 45.5 BC: 12.4, HF: 45.5
MIMIC-IV$*\ddagger$	24,516	100	1/3	4,686	D: 37.2 A: 20.7, S: 19.2, D: 37.2
Rotterdam$\dagger\ddagger$	2,982	10	2/3	7,043	R: 6.5, D: 50.9 R: 50.9, D: 42.7
PRO-ACT$\ddagger$	3,220	8	4	498	SP: 37.8, SW: 31.9, HA: 50.3, WA: 60.6
EBMT$\ddagger$	2,279	6	5	6229	R: 34.4, AE: 39.8, RAE: 29, REL: 16.2, D: 23.4

SEER: The U.S. Surveillance, Epidemiology, and End Results (SEER) dataset (Ries et al., 2003) is a comprehensive collection of cancer patient data from approximately 49% of the U.S. population. It provides information about survival times for cancer patients post-diagnosis. For the single-event case, we consider the outcome of death due to breast cancer, and for the competing-risks case, we consider the outcomes of death due to breast cancer and death due to heart failure, and consider other outcomes as censored. Concerning preprocessing, we select a cohort of 19,246 newly-diagnosed patients (first year of disease) with 17 features,

which include demographics and tumor characteristics. We use the raw feature "COD to site recode" to extract the event indicator and apply some special value processing for some of the features, based on the guideline provided by Wang and Sun (2022). For categorical features, missing values are imputed using the mode, followed by one-hot encoding. The data is then standardized to ensure consistency in our analyses. The SEER cohort is available for download at www.seer.cancer.gov.

MIMIC-IV: The Medical Information Mart for Intensive Care (MIMIC-IV) dataset by Johnson et al. (2023) comprises electronic health records from patients admitted to intensive care units at the Beth Israel Deaconess Medical Center in the U.S., covering the years from 2008 to 2019. For the single-event case, we consider death since hospital admission and for the multi-event case, we consider the outcomes of acute respiratory failure (ARF), shock and death. Concerning preprocessing, we use the method described by Gupta et al. (2022) to extract 1,672 static features from the dataset, for 26,236 patients whose ages are between 60 and 65 years. To refine our feature set, we apply the unicox feature selection method (Qi et al., 2022; Simon et al., 2003), reducing the number of features to the 100 that appear to be the most relevant. In the training set, we assessed the statistical significance of each feature by calculating the p-values using lifelines's (Davidson-Pilon, 2019) CoxPHFitter summary. The top 100 features with the smallest p-values were selected. The final set of features is included in the source code. The MIMIC-IV v2.2 is available for download at https://physionet.org/content/mimiciv/2.2/.

Rotterdam: The Rotterdam dataset (Royston and Altman, 2013) includes 2,982 primary breast cancer patients from the Rotterdam tumor bank, with 1,546 having node-positive disease. The follow-up time ranged from 1 to 231 months (median 107 months). The dataset has 10 features, which include demographics, tumor characteristics, and treatment information. In the competing-risks case, we consider the outcome of recurrence-free survival time (RFS), which is defined as the time from primary surgery to the earlier of disease recurrence or death from any cause. In the multi-event case, we consider the time to death or disease recurrence, where death censors disease recurrence, but disease recurrence does not censor death. Figure 8 shows the event trajectories in the Rotterdam dataset. Concerning preprocessing, we converted the categorical size groups ('$\leq$ 20', '20-50', '$>$ 50') to their approximate median values (10, 35, and 75, respectively). The Rotterdam dataset is available at https://rdrr.io/cran/survival/man/rotterdam.html.

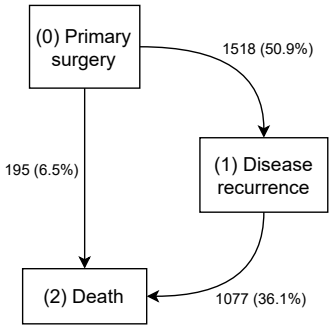

Figure 8: Event trajectories in the Rotterdam dataset (N=2,982).

PRO-ACT: The Pooled Resources Open-Access Clinical Trials (PRO-ACT)[6] database by Atassi et al. (2014) is the largest ALS dataset in the world, which includes patient demographics, lab and medical records and family history of over 11,600 ALS patients, where ALS disease state is evaluated by the ALS functional rating scale revised (ALSFRS-R) (Cedarbaum et al., 1999). The assessment contains 12 questions and a maximum score of 48, each representing physical functionality, *e.g.*, the ability to walk. Each ALSFRS-R score is between 0 and 4, where 0 represents complete inability with regard to the function, and 4

6. Data used in the preparation of this article were obtained from the Pooled Resource Open-Access ALS Clinical Trials (PRO-ACT) Database. The data available in the PRO-ACT Database have been volunteered by PRO-ACT Consortium members.

represents normal function. We use the ALSFRS-R scores to formulate four events: "Speech", "Swallowing", "Handwriting" and "Walking". We consider an event to have occurred if a patient scores 2 or lower on the ALSFRS-R scale during any subsequent follow-up visitation following their baseline visitation. The events can occur in any order and are not mutually exclusive. ALSFRS-R scores are generally monotonic decreasing, but due to the subjectivity of the assessment, scores can occasionally increase post-baseline. We use a maximum follow-up time of 500 days and exclude patients with no recorded history. The PRO-ACT dataset is available for download at https://ncri1.partners.org/ProACT.

EBMT: The European Society for Blood and Marrow Transplantation (EBMT) dataset (Rossman et al., 2022) consists of 2,279 patient records who underwent blood or marrow transplantation between 1985 and 1998. The dataset records survival times after transplantation for patients suffering from blood cancer. We consider five events in our analysis: "Recovery" (Rec), "Adverse Event" (AE), a combination of both (AE and Rec), "Relapse", and "Death". These events allowed us to evaluate their individual and combined effects on patient prognosis. The dataset includes six features belonging to four baseline covariates: donor-recipient match (where a gender mismatch is defined as a female donor and a male recipient), prophylaxis type, year of transplant, and age at transplant. Concerning preprocessing, we encode the year of transplantation as boolean variables indicating whether the transplant took place during 1990-1994 and 1995-1998, and we classify the age at the transplantation into boolean variables indicating whether the age was less than 20, between 20 and 40, or more than 40 years. The EBMT dataset is available in the PyMSM package at https://github.com/hrossman/pymsm.

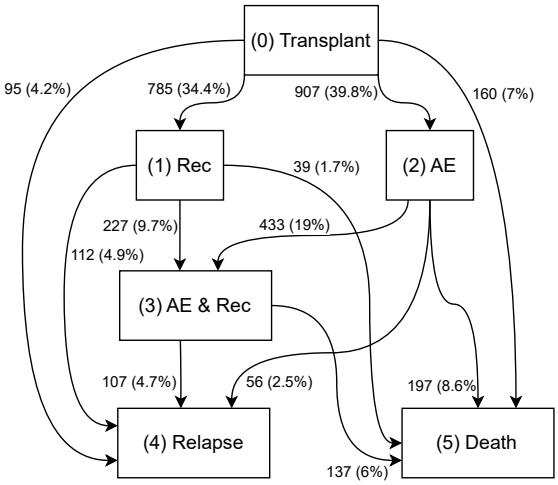

Figure 9: Event trajectories in the EBMT dataset (N=2,279).

E.2. Baselines models

CoxPH: The Cox Proportional Hazards model (CoxPH) is a widely used semiparametric model for survival analysis (Cox, 1972). It assumes that the hazard function for an individual i with covariates $\boldsymbol{x}^{(i)}$ can be expressed as $h\big(t|\boldsymbol{x}^{(i)}\big) = h_0(t)\exp\big(f\big(\boldsymbol{\theta}, \boldsymbol{x}^{(i)}\big)\big)$, where $h_0(t)$ is an unspecified baseline hazard function, and f is typically a linear function $f(\boldsymbol{\theta}, \boldsymbol{x}^{(i)}) = \boldsymbol{x}^{(i)\top}\boldsymbol{\theta}$ producing a risk score. The key assumption of CoxPH is that covariates have a multiplicative effect on the hazard and that this effect is constant over time (*i.e.*, proportional hazards). Model parameters $\boldsymbol{\theta}$ are estimated by maximizing the Cox partial log-likelihood, which does not require modeling $h_0(t)$ explicitly. The model is implemented from the *scikit-survival* package (Pölsterl, 2020).

CoxNet: CoxNet is an extension of the Cox Proportional Hazards model that incorporates Elastic Net regularization into the partial log-likelihood optimization (Simon et al., 2011). The hazard function retains

the same form as in CoxPH, $h\big(t|\boldsymbol{x}^{(i)}\big) = h_0(t)\exp\big(\boldsymbol{x}^{(i)\top}\boldsymbol{\theta}\big)$, but the parameter estimation adds a penalty term $\omega\left[\alpha\|\boldsymbol{\theta}\|_1 + \frac{1-\alpha}{2}\|\boldsymbol{\theta}\|_2^2\right]$, where $\omega > 0$ controls the overall strength of regularization, and $\alpha \in [0,1]$ controls the balance between L1 (lasso) and L2 (ridge) penalties. This approach enables feature selection (via L1) while maintaining stability in the presence of multicollinearity (via L2), making it particularly effective in high-dimensional settings. The model is implemented from the *scikit-survival* package (Pölsterl, 2020).

Weibull AFT: The Weibull Accelerated Failure Time (AFT) model is a parametric survival model in which survival times are assumed to follow a Weibull distribution whose parameters depend on covariates. In the absence of covariates, the scale parameter can be interpreted as the time at which 37% of the population has experienced the event, while the shape parameter determines whether the cumulative hazard is convex or concave, corresponding to accelerating or decelerating hazards. The cumulative hazard function has the closed form $H(t) = (t/\beta)^\eta$, where $\beta > 0$ is the scale parameter and $\eta > 0$ is the shape parameter. When covariates $\boldsymbol{x}^{(i)}$ are present, the scale parameter is modeled as $\beta^{(i)} = \exp(-\boldsymbol{x}^{(i)\top}\boldsymbol{\theta})$ and, optionally, the shape parameter may also depend on $\boldsymbol{x}^{(i)}$. Model parameters are estimated by maximizing the full likelihood. The model is implemented from the *lifelines* package (Davidson-Pilon, 2019).

GBSA: Gradient Boosted Survival Analysis (GBSA) (Ridgeway, 1999) is a shallow ensemble model that extends gradient boosting methods to survival settings by optimizing a penalized version of the Cox partial log-likelihood. Specifically, it employs an ℓ_2-norm regularization to prevent overfitting and enhance generalization. Unlike classical gradient boosting, which typically updates a single weak learner or fits all features to the negative gradient, GBSA uses an offset-based approach to flexibly update multiple candidate features at each boosting iteration. This leads to more efficient gradient approximation and model convergence. The model is implemented from the *scikit-survival* package (Pölsterl, 2020).

RSF: Random Survival Forests (RSF) (Ishwaran et al., 2008) adapt the random forest framework to survival analysis by building an ensemble of survival trees, each trained on a bootstrap sample of the data. Each tree recursively partitions the feature space using a survival-specific splitting criterion—typically the log-rank test—to maximize differences in survival outcomes between child nodes. RSF naturally handles nonlinear covariate effects and complex interactions, and it is well-suited for high-dimensional settings without requiring strong parametric assumptions. Censoring is handled natively by the splitting criterion and ensemble aggregation. The model is implemented from the *scikit-survival* package (Pölsterl, 2020).

NFG: Neural Fine-Gray (NFG) (Jeanselme et al., 2023) extends the classical sub-distribution hazard approach of Fine and Gray (1999) to a flexible neural-network framework for competing risks. The architecture employs a shared embedding network for covariates, followed by separate monotonic neural networks that directly model each instance's CIF, ensuring non-decreasing behavior over time through weight constraints. The CIFs are combined using a risk-balancing mechanism so that their sum remains less than or equal to one, preserving a valid joint probability structure. Training maximizes the exact log-likelihood derived analytically from the CIFs via automatic differentiation, avoiding numerical integration. This formulation allows the model to capture non-linear covariate effects and complex, risk-specific dynamics without assuming a parametric hazard form. The model is implemented from `https://github.com/Jeanselme/NeuralFineGray`.

DeepSurv: DeepSurv (Katzman et al., 2018) is a deep learning extension of the Cox Proportional Hazards model that replaces the linear risk function with a nonlinear neural network. The individual hazard function takes the form $h\big(t|\boldsymbol{x}^{(i)}\big) = h_0(t)\exp\big(f\big(\boldsymbol{\theta}, \boldsymbol{x}^{(i)}\big)\big)$, where $f(\boldsymbol{\theta}, \boldsymbol{x}^{(i)})$ is a feedforward neural network that maps covariates to a continuous risk score. The model is trained by numerically maximizing the Cox partial log-likelihood with respect to the network parameters $\boldsymbol{\theta}$. By capturing nonlinear interactions among covariates, DeepSurv can improve predictive performance over traditional Cox models in complex data settings. The model is implemented from `https://github.com/shi-ang/BNN-ISD/`.

DeepHit: DeepHit (Lee et al., 2018) is a deep discrete-time model designed to handle competing risks, where the occurrence of one event prevents the occurrence of another, as it assumes $\sum_{t=1}^{t_{\max}}\sum_{k=1}^{K} p_k[t] = 1$, where $t_{\max} \in \mathbb{N}_+$ is the number of discrete time bins (the maximum discretized time index) in the prediction horizon and $K \in \mathbb{N}_+$ is number of competing events (causes). DeepHit contains a shared covariate layer among all events and cause-specific subnetworks for each competing event. The model outputs a probability distribution for each discrete time bin and each competing event via a softmax layer. This architecture allows DeepHit to estimate the joint distribution of survival times and events directly without making

assumptions about the underlying stochastic process. It accounts for dependencies among events in $\boldsymbol{\theta}$ but not for individuals who experience no event by time T. The model is implemented from https://github.com/chl8856/DeepHit.

Hierarchical: The hierarchical model (Tjandra et al., 2021) is a deep discrete-time survival model, designed to handle multiple events by predicting event probabilities across multiple temporal granularities. It begins with coarse-scale predictions (*e.g.*, monthly) and progressively refines them into finer ones (*e.g.*, daily), effectively decomposing long-horizon tasks into simpler intermediate steps. Formally, the model is a multi-task network that takes input x and produces $\hat{P}$, a set of event-specific probability distributions $\hat{p}_k$, computed hierarchically by subnetworks ϕ_k. At granularity m, the probability $\hat{p}_{km}$ is derived from both $\psi_{km}(\Phi(x))$ and the preceding level $\hat{p}_{k,m-1}$, enforcing top-down consistency. During training, a composite likelihood with ranking penalties provides supervision across the event horizon, optimized only for discriminative performance using Harrell's C-index (Harrell Jr et al., 1996). The model is implemented from https://github.com/MLD3/Hierarchical_Survival_Analysis.

MTLR: The Multi-Task Logistic Regression (MTLR) model (Yu et al., 2011) is a discrete-time survival model that estimates the full survival distribution by training a sequence of dependent logistic regression models—one for each discrete time interval. Unlike models that rely on hazard functions, MTLR directly estimates the probability of survival at each time step, making it flexible and naturally suited for handling censored data. The model captures temporal dependencies between intervals by sharing parameters across the logistic regressors. MTLR has been extended to the competing-risk setting as Deep-CR MTLR (Kim et al., 2021), which replaces the logistic regressors with neural networks to model nonlinear effects and inter-event dependencies. MTLR is implemented from https://github.com/shi-ang/BNN-ISD/ and Deep-CR MTLR is implemented from https://github.com/bhklab/aaai21_survival_prediction.

DSM: Deep Survival Machines (DSM) (Nagpal et al., 2021) is a deep continuous-time survival model that estimates the survival function as a weighted mixture of k parametric survival distributions (*e.g.*, Weibull or LogNormal). Both the parameters of each distribution and their associated mixture weights are modeled as nonlinear functions of the input covariates, using a shared multilayer perceptron (MLP). This mixture-based formulation allows DSM to capture heterogeneity in the event-time distributions. The model is trained by maximizing the likelihood of observed and censored data. The model is implemented from the *auton-survival* package (Nagpal et al., 2022).

E.3. Hyperparameter settings

Tables 8-19 list the hyperparameter settings used for each model across all datasets. These values were selected via Bayesian optimization (Snoek et al., 2012) over ten iterations, adopting the hyperparameters leading to the highest Harrell's C-index in the validation set (seed = 0). When applicable, early stopping was used based on the validation loss. Note that the default configuration is used to evaluate computational complexity only.

Table 8: Hyperparameters for CoxPH.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Ties method	breslow	breslow	breslow	breslow	breslow	breslow
Regularization	0	10	10	0.01	0	0.01
Iterations	100	200	100	500	500	100
Tolerance	1e-9	1e-5	1e-5	1e-7	1e-9	1e-5

Table 9: Hyperparameters for CoxNet.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Number of alphas	100	150	150	150	50	150
Alpha min ratio	auto	0.0001	0.0001	0.0001	0.001	0.0001
L1 ratio	0.5	0.1	1	0.75	1	0.25
Tolerance	1e-7	1e-05	1e-7	1e-06	1e-7	1e-05
Max iterations	100000	100000	200000	100000	200000	200000

Table 10: Hyperparameters for Weibull AFT.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Penalizer	0	0.001	0.0001	0.1	0.001	0
L1 Ratio	0	1	1	0	0.5	0

Table 11: Hyperparameters for GBSA.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Number of estimators	100	50	800	800	800	50
Learning rate	0.1	1	1	1	1	1
Max depth	1	10	1	10	10	3
Min samples split	10	2	5	10	10	2
Min samples leaf	5	10	5	1	10	1
Max features	`sqrt`	`sqrt`	None	`sqrt`	None	None
Dropout rate	0.0	0.5	0.5	0.5	0.5	0.25
Subsample	0.8	1.0	1.0	0.25	0.25	0.25

Table 12: Hyperparameters for RSF.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Number of estimators	100	500	200	200	500	1000
Max depth	1	10	5	5	5	10
Min samples split	10	2	5	10	10	2
Min samples leaf	5	10	5	1	10	1
Max features	`sqrt`	`sqrt`	`log2`	`log2`	None	`log2`

Table 13: Hyperparameters for NFG.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Layers	[32]	[64]	[128]	–	–	–
Activation	ReLU	ReLU	ReLU	–	–	–
Survival layers	[32]	[32]	[32]	–	–	–
Dropout	0.0	0.25	0.5	–	–	–
Optimizer	Adam	Adam	Adam	–	–	–
Multihead	True	True	True	–	–	–

Table 14: Hyperparameters for DeepSurv.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Hidden size	32	128	64	32	32	64
Learning rate	0.001	0.005	0.001	0.0005	0.005	0.005
Weight decay	0.01	0.01	0.1	0.01	0	0.1
Dropout rate	0.25	0	0.5	0	0	0.5
Epochs	1000	2000	2000	1000	2000	500
Early stopping	True	True	True	True	True	True
Patience	10	10	20	10	20	20

Table 15: Hyperparameters for DeepHit.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Shared layer nodes	32	16	16	32	64	64
Individual layer nodes	32	64	64	64	128	128
Dropout rate	0.25	0.25	0.25	0.5	0.1	0.1
Learning rate	0.001	0.0001	0.005	0.0001	0.0005	0.0005
Weight decay	0.01	0.1	0	0.1	0	0.01
Alpha	0.2	0.4	0.4	0	0	0.8
Sigma	0.1	0.5	0.05	0.1	0.1	0.1
Eta multiplier	0.8	1.0	1.0	1.0	0.5	1.0

Table 16: Hyperparameters for Hierarchical.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Theta layer size	[32]	[50]	[200]	[200]	[200]	[50]
Fine bin layer size	[32, 5]	[50, 5]	[100, 10]	[150, 10]	[150, 10]	[100, 10]
Learning rate	0.001	0.0005	0.001	0.001	0.001	0.001
Regularization constant	0.01	0.01	0.01	0.01	0.01	0.01
Alpha	0.0001	1e-5	1e-5	0	0.0001	0.001
Sigma	10	10	5	20	5	5

Table 17: Hyperparameters for MTLR.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Learning rate	0.001	0.0001	0.001	0.0001	0.0001	0.0005
Weight decay	0.01	0.001	0.001	0.1	0	0.1
Dropout rate	0.25	0	0	0.1	0.1	0.25

Table 18: Hyperparameters for DSM.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Hidden layers	32	32	32	64	64	64
Learning rate	0.001	0.005	0.005	0.005	0.005	0.001
Iterations	10000	20000	10000	10000	10000	10000
Batch size	32	32	64	32	64	64
Number of dists.	3	5	1	5	5	1

Table 19: Hyperparameters for MENSA.

Hyperparameter	Default	SEER	Rotterdam	MIMIC-IV	PRO-ACT	EBMT
Hidden nodes	32	128	32	32	16	128
Learning rate	0.001	0.0005	0.001	0.001	0.0001	0.001
Weight decay	0.01	0.005	0.001	0.0001	0.001	0.0001
Dropout rate	0.25	0	0.25	0.25	0.1	0.1
Batch size	32	128	32	64	32	32
Number of distributions (Ψ)	1	3	3	1	3	3
Weighting parameter (λ)	-	-	0.25	-	-	0.5

E.4. Implementation details

To train MENSA, we optimize the joint likelihood of event times and event trajectories using stochastic gradient descent (SGD). The algorithm operates over mini-batches of survival data, computing both the multi-event likelihood and the trajectory likelihood at each iteration, and updating the shared and event-specific parameters accordingly. Algorithm 1 summarizes the training procedure, including initialization, loss computation, and parameter updates over the training epochs. All experiments were run in Python 3.9 with PyTorch 1.13.1, NumPy 1.24.3, and Pandas 1.5.3 on a workstation with an Intel Core i9-10980XE (3.00 GHz), 64 GB RAM, and an NVIDIA RTX 3090 GPU (CUDA 11.7). All datasets are publicly available, and the code for loading and preprocessing the datasets is provided in the source code repository.

Appendix F. Statistical and practical significance

To assess whether performance differences between MENSA and baseline models were statistically and practically meaningful, we conducted paired tests across all datasets and metrics. For each dataset-metric pair, we compared MENSA to every baseline model using per-seed results averaged across events.

Statistical testing. We performed two-sided paired t-tests using the per-seed metric values, treating runs with the same random seed as paired observations. The resulting p-values were adjusted for multiple comparisons using the Holm correction method ($\alpha = 0.05$). Models whose adjusted p-values fell below the

Algorithm 1 Training algorithm for MENSA.

Input: A survival dataset $\mathcal{D}$ of the form $\{(X^{(i)}, T^{(i)}_{\text{obs}}, \delta^{(i)})\}_{i=1}^{N}$; $\mathcal{M}$: the model with initialized parameters $\{\Phi_\theta, \tilde{\beta}, \tilde{\eta}, \zeta, \xi, g_\theta\}$; $\mathcal{T}$: list of trajectories; κ: learning rate; L: number of training epochs; λ: weighting hyperparameter.
Result: $\mathcal{M} = \{\hat{\Phi}_\theta, \hat{\tilde{\beta}}, \hat{\tilde{\eta}}, \hat{\zeta}, \hat{\xi}, \hat{g}_\theta\}$: trained model parameters.
Initialization: $\mathcal{M} \leftarrow \texttt{Instantiate}(\mathcal{M})$
for $i = 1, \ldots, L$ **do**
  $[\log f(T_{\text{obs}}), \log S(T_{\text{obs}})] \leftarrow \mathcal{M}(\mathcal{D})$ // $\log f$ and $\log S$ are $N \times P$
  $\mathcal{L}_{\text{ME}} \leftarrow -\sum_{p \in \mathcal{P}} w_p \big[\delta_p \log f_p(T_{\text{obs}}) + (1-\delta_p) \log S_p(T_{\text{obs}})\big]$ // Multi-event loss
  $\mathcal{L}_{\text{trajectory}} \leftarrow 0$
  for each pair $(A, B) \in \mathcal{T}$ **do**
    $\mathcal{L}_{\text{trajectory}} \mathrel{+}= \sum_i \delta_A^{(i)} \delta_B^{(i)} \big[-\log S_B(T_A^{(i)})\big]$ // Accumulate trajectory loss
  end for
  $\mathcal{L}_{\text{total}} \leftarrow (1-\lambda)\frac{1}{N}\mathcal{L}_{\text{ME}} + \lambda \frac{1}{N}\mathcal{L}_{\text{trajectory}}$ // Weighted total loss
  $\{\hat{\Phi}_\theta, \hat{\tilde{\beta}}, \hat{\tilde{\eta}}, \hat{\xi}, \hat{\zeta}, \hat{g}_\theta\}^{(i)} \leftarrow \text{SGD}\big(\{\cdot\}^{(i-1)}, \nabla \mathcal{L}_{\text{total}}, \kappa\big)$ // Update step
end for
return $\mathcal{M}$

threshold were considered significantly different from MENSA. The correction was applied independently for each dataset–metric combination.

Practical significance. In addition to statistical testing, we required a minimum effect size of interest (MEI) to ensure that reported differences were not only statistically significant but also practically meaningful. For all discrimination (CI_G, CI_L, AUC) and calibration (IBS, mMAE) metrics, the MEI threshold was set to 1.0 point on the 0-100 scale used for evaluation. For metrics where higher is better (C-indices and AUC), a baseline was considered practically worse if its mean was at least 1.0 point lower than MENSA's mean. For metrics where lower is better (IBS, mMAE), the baseline had to exceed MENSA's mean by at least 1.0 point to be considered practically worse.

Reporting. A dagger symbol (†) in the tables indicates models that were both statistically significant (Holm-corrected $p < 0.05$) and practically worse than MENSA according to the MEI criterion. D-calibration was excluded from dagger marking since it is reported as a count of calibrated experiments rather than a continuous measure. The full analysis pipeline – including scaling conventions, dataset ordering, and Holm-adjusted significance checks – is implemented in a source code repository.

Appendix G. Additional results

G.1. Single-event results

Table 20 shows the discrimination and calibration results on two single-event datasets. In the SEER dataset, MENSA demonstrates solid discriminative ability, ranking third in Harrell's C-index and second in AUC, while also achieving the second-best IBS and mMAE scores, which underscores its accuracy in survival prediction. In the MIMIC-IV dataset, MENSA shows similarly strong performance, obtaining the second-best Harrell's C-index, the best overall AUC and IBS, and the lowest mMAE among the deep learning baselines. Taken together, these findings indicate that MENSA is not only competitive with shallow single-event survival methods but also surpasses existing deep learning approaches on several key metrics. This highlights its effectiveness for both risk discrimination and time-to-event prediction in large, single-event datasets.

Table 20: Discrimination and calibration performance (mean ± SD.) between different models on single-event benchmark datasets, averaged over 10 experiments. For Harrell's CI, AUC, and D-calibration, higher (↑) is better. For IBS and mMAE, lower (↓) is better. D-calibration counts the number of experiments in which the respective model was D-calibrated.

Dataset	Model	Harrell's CI	AUC	IBS	mMAE	D-Cal
SEER ($K = 1$)	CoxPH	69.8±0.61	74.3±0.78	16.8±0.30	29.9±0.39	4/10
	CoxNet	69.8±0.62	74.3±0.79	16.8±0.30	29.8±0.37	4/10
	Weibull	69.8±0.62	74.3±0.78	16.9±0.30	34.3±0.82	10/10
	GBSA	77.1±0.53	81.7±0.62	15.1±0.28	25.6±0.33	1/10
	RSF	77.1±0.55	82.1±0.69	14.2±0.29	29.1±0.62	8/10
	MTLR	70.5±0.58	75.7±0.81	16.4±0.30	29.9±0.70	10/10
	DeepSurv	76.0±0.70	80.7±0.78	14.8±0.35	29.6±0.84	0/10
	DeepHit	75.8±0.79	80.8±0.93	14.8±0.38	26.7±0.46	2/10
	DSM	75.9±0.93	80.8±1.03	14.9±0.47	42.3±2.31	7/10
	MENSA (Ours)	76.0±0.65	81.0±0.77	14.8±0.36	27.1±0.71	8/10
MIMIC-IV ($K = 1$)	CoxPH	74.2±0.48	78.7±0.60	12.9±0.38	5.9±0.03	10/10
	CoxNet	74.3±0.48	78.8±0.61	12.8±0.38	5.9±0.04	10/10
	Weibull	74.5±0.49	79.1±0.63	13.0±0.38	5.8±0.05	10/10
	GBSA	74.9±0.43	79.5±0.56	13.9±0.39	5.9±0.05	8/10
	RSF	72.5±0.47	78.9±0.56	14.2±0.40	6.0±0.06	8/10
	MTLR	74.3±0.52	79.5±0.77	12.5±0.36	7.1±0.24	9/10
	DeepSurv	75.1±0.43	80.0±0.60	12.4±0.39	6.8±0.10	10/10
	DeepHit	74.5±0.49	79.7±0.75	20.5±0.57	11.7±0.14	0/10
	DSM	74.4±0.68	79.2±0.93	13.2±0.44	6.4±0.56	5/10
	MENSA (Ours)	75.4±0.46	80.1±0.58	12.4±0.34	6.1±0.05	4/10

G.2. Competing-risks results

Table 21 shows the discrimination and calibration results on two competing-risks datasets. In SEER, MENSA performs competitively across all metrics, combining strong discrimination with low prediction error and good calibration. In Rotterdam, MENSA also provides solid discrimination and competitive calibration, though with somewhat higher prediction error compared to the best-performing single-event baselines. However, it has the third-lowest mMAE in this dataset as well. MENSA also achieves good D-calibration results across both datasets. These findings suggest that MENSA extends robustly to competing-risk settings, offering a balanced trade-off between discrimination and calibration performance.

Appendix H. Computational analysis

Table 22 compares selected deep learning baselines and MENSA in terms of space complexity. This analysis is conducted in single-event, competing-risks and multi-event scenarios. For a fair comparison, we configure all models with a single hidden layer with 32 nodes. By nature of design, models that cannot learn events jointly have to be trained K times in the multi-event scenario. In this case, the DeepSurv model, as an example, has a $K\mathcal{O}(d)$ space complexity, where d is the number of features, assuming training is not done in parallel. This can lead to significant increases in the number of trainable parameters in high-dimensional datasets with many events, making it a less-attractive solution. MENSA does not suffer from this problem, as it can train on multiple events at once with a space complexity of $\mathcal{O}(P(\Psi + d) + d)$.

Figure 10 shows the number of trainable parameters and FLOPs (computed using the `fvcore` library[7]) as a function of the number of features. In the single-event case, MENSA uses fewer parameters and FLOPs than Hierarchical, MTLR, and DeepHit, and only slightly more than DeepSurv and DSM. In competing-risks, MENSA remains more efficient than most models except DSM, whose overhead is marginally lower. For

7. Meta Research's fvcore library, available at https://github.com/facebookresearch/fvcore (September 1, 2025).

Table 21: Discrimination and calibration performance (mean ± SD.) between different models on competing-risks benchmark datasets, averaged over 10 experiments. For global CI, local CI, and AUC, higher (↑) is better. For IBS and mMAE, lower (↓) is better. D-calibration counts the number of experiments in which the respective model was D-calibrated.

Dataset	Model	Global CI	Local CI	AUC	IBS	mMAE	D-Cal
SEER ($K=2$)	CoxPH	67.9±0.61	78.4±0.71	66.9±0.54	13.3±0.18	29.0±0.60	14/20
	CoxNet	68.8±0.61	78.2±0.60	64.8±0.62	13.4±0.18	26.0±0.71	14/20
	Weibull	67.9±0.61	78.5±0.71	66.8±0.56	13.5±0.18	54.7±0.94	20/20
	GBSA	74.8±0.54	78.5±0.70	72.7±0.67	12.4±0.15	23.8±0.68	11/20
	RSF	74.4±0.56	81.0±0.49	73.7±0.66	12.0±0.16	38.8±1.78	18/20
	NFG	72.8±0.55	80.0±0.46	68.6±0.61	12.6±0.16	185.8±11.84	7/20
	MTLR-CR	67.4±0.70	73.8±1.06	66.1±0.82	16.0±0.39	40.6±0.73	0/20
	DeepSurv	73.4±0.61	79.5±0.56	70.3±0.47	12.3±0.18	31.7±2.54	11/20
	DeepHit	71.5±0.42	78.5±0.66	68.5±0.58	12.7±0.14	135.2±15.01	8/20
	DSM	73.0±0.73	79.2±1.00	70.1±0.65	12.5±0.24	74.7±9.09	15/20
	Hierarch.	71.4±1.00	69.8±1.33	68.2±1.10	15.4±0.40	38.0±1.48	0/20
	MENSA (Ours)	72.9±0.66	78.7±0.80	69.9±0.66	12.4±0.19	26.0±0.89	17/20
Rotterdam ($K=2$)	CoxPH	69.0±1.49	86.9±1.85	78.8±1.57	13.8±2.71	252.2±84.82	20/20
	CoxNet	69.0±1.50	86.9±1.79	78.9±1.49	13.8±2.67	307.6±125.96	20/20
	Weibull	69.1±1.56	88.4±1.49	78.9±1.61	13.9±3.09	420.0±72.95	15/20
	GBSA	75.2±1.85	88.9±1.53	71.8±2.47	14.7±2.58	21.0±1.30	20/20
	RSF	69.7±1.49	88.6±1.29	78.6±1.77	13.4±2.35	224.4±58.86	20/20
	NFG	66.2±3.86	89.0±1.57	75.3±3.80	15.1±3.05	810.1±129.17	13/20
	MTLR-CR	68.4±1.81	86.2±1.69	78.3±1.73	16.0±3.89	54.3±7.57	3/20
	DeepSurv	69.1±1.62	88.3±1.93	78.7±1.78	13.9±2.53	103.4±30.50	20/20
	DeepHit	62.6±2.52	88.8±1.50	77.0±2.32	15.1±2.74	33.3±5.26	15/20
	DSM	69.9±1.40	88.8±1.37	78.2±2.01	13.5±2.47	139.3±4.09	17/20
	Hierarch.	65.2±1.72	73.4±4.57	50.0±0.06	76.1±13.31	83.4±7.46	0/20
	MENSA (Ours)	67.3±1.71	88.9±1.54	77.1±1.84	15.2±5.23	48.9±7.46	18/20

Table 22: Comparison between the proposed model and selected state-of-the-art baselines in terms of space complexity. d: number of features, K: number of events, B: number of discrete time bins, M: levels of granularity, Ψ: number of mixture distributions, P: number of states.

Model	Single-event	Competing risks	Multi-event
DeepSurv	$\mathcal{O}(d)$	$K\mathcal{O}(d)$	$K\mathcal{O}(d)$
DeepHit	$\mathcal{O}(Bd)$	$\mathcal{O}(KBd)$	$K\mathcal{O}(Bd)$
MTLR	$\mathcal{O}(Bd)$	$\mathcal{O}(KBd)$	$K\mathcal{O}(Bd)$
DSM	$\mathcal{O}(\Psi+d)$	$\mathcal{O}(K\Psi+d)$	$K\mathcal{O}(\Psi+d)$
Hierarch.	$\mathcal{O}(MB+d)$	$\mathcal{O}(MKB+d)$	$\mathcal{O}(MKB+d)$
MENSA (Ours)	$\mathcal{O}(\Psi+d)$	$\mathcal{O}(P(\Psi+d)+d)$	$\mathcal{O}(P(\Psi+d)+d)$

datasets with over 100 features, differences are negligible. In the multi-event scenario, baseline models trained separately per event require K trainings, while MENSA trains all events jointly, achieving a notably lower parameter count and FLOPs than all baselines, including the Hierarchical model (see Table 23). Overall, MENSA is the most computationally efficient among deep learning models in multi-event settings.

Appendix I. Proof of non-proportional hazards

This section addresses how the proposed method overcomes the limitations associated with the proportional hazards assumption (Cox, 1972). We begin by reviewing how a traditional Weibull distribution adheres to the proportional hazards property and then show, intuitively, why MENSA, which uses a mixture of Weibull distributions, does not meet this assumption.

Table 23: Relative reduction in parameters and FLOPs for MENSA compared to deep learning baselines in the multi-event setting ($K = 4$) with 100 features.

Model	#Params. reduction	#FLOPs reduction
DeepSurv	1.4×	1.5×
Deephit	2.5×	2.5×
MTLR	3.2×	5.7×
DSM	1.5×	1.5×
Hierarch.	13×	13×

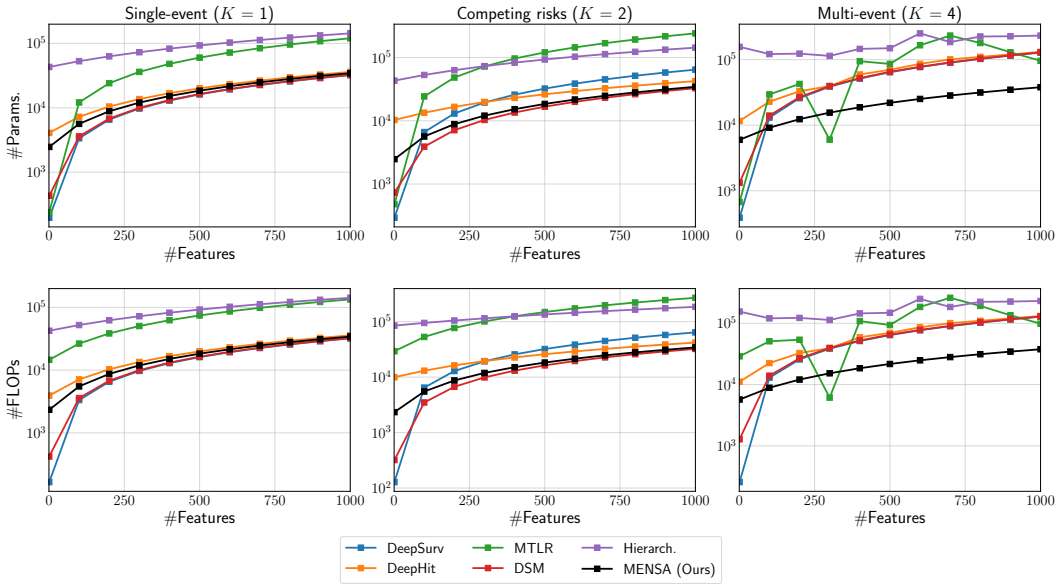

Figure 10: Number of features vs. number of trainable parameters (upper row) and number of floating-point operations (lower row) for single-event ($K = 1$), competing-risks ($K = 2$) and multi-event ($K = 4$) scenarios.

It is well-established that the Weibull distribution satisfies the proportional hazards assumption when its shape parameter, η, is independent of the covariates $\boldsymbol{x}^{(i)}$ (Klein and Moeschberger, 2006). The PDF, CDF, survival function, and hazard function for a Weibull distribution are given by:

$$f(t;\beta,\eta) = \frac{\eta}{\beta}\left(\frac{t}{\beta}\right)^{\eta-1} \exp\left(-\left(\frac{t}{\beta}\right)^{\eta}\right), \tag{29}$$

$$F(t;\beta,\eta) = \int_{\tau=0}^{t} f(\tau;\beta,\eta)\,d\tau = 1 - \exp\left(-\left(\frac{t}{\beta}\right)^{\eta}\right), \tag{30}$$

$$S(t;\beta,\eta) = 1 - F(t;\beta,\eta) = \exp\left(-\left(\frac{t}{\beta}\right)^{\eta}\right), \tag{31}$$

$$h(t;\beta,\eta) = \frac{f(t;\beta,\eta)}{S(t;\beta,\eta)} = \frac{\eta}{\beta}\left(\frac{t}{\beta}\right)^{\eta-1}. \tag{32}$$

To demonstrate that the Weibull distribution satisfies the proportional hazards assumption, we parameterize the scale parameter, β, as a function of the covariates $\boldsymbol{x}^{(i)}$:

$$\beta(\boldsymbol{x}^{(i)}) \;=\; \beta_0 \exp(-\gamma^{\top}\boldsymbol{x}^{(i)}), \tag{33}$$

where β_0 is the baseline scale parameter, and γ is a vector of regression coefficients. Substituting $\beta(\boldsymbol{x}^{(i)})$ into the hazard function yields:

$$\begin{aligned} h(t;\boldsymbol{x}^{(i)}) &= \frac{\eta}{\beta(\boldsymbol{x}^{(i)})}\left(\frac{t}{\beta(\boldsymbol{x}^{(i)})}\right)^{\eta-1} \\ &= \frac{\eta}{\beta_0\exp\left(-\gamma^\top\boldsymbol{x}^{(i)}\right)}\left(\frac{t}{\beta_0\exp\left(-\gamma^\top\boldsymbol{x}^{(i)}\right)}\right)^{\eta-1} \\ &= \frac{\eta\exp\left(\gamma^\top\boldsymbol{x}^{(i)}\right)}{\beta_0}\left(\frac{t\cdot\exp\left(\gamma^\top\boldsymbol{x}^{(i)}\right)}{\beta_0}\right)^{\eta-1} \\ &= \frac{\eta}{\beta_0^\eta}t^{\eta-1}\exp\left(\eta\gamma^\top\boldsymbol{x}^{(i)}\right). \end{aligned} \tag{34}$$

We define the baseline hazard function as:

$$h_0(t) = \frac{\eta}{\beta_0^\eta}\cdot t^{\eta-1}. \tag{35}$$

Therefore, the hazard function can be expressed as:

$$h(t;\,\boldsymbol{x}^{(i)}) = h_0(t)\exp\left(\eta\gamma^\top\boldsymbol{x}^{(i)}\right). \tag{36}$$

This formulation shows that the Weibull hazard function is the product of a time-dependent baseline hazard, $h_0(t)$, and a time-independent relative hazard, $\exp\left(\eta\gamma^\top\boldsymbol{x}^{(i)}\right)$. Hence, when the shape parameter η is constant, and only the scale parameter β varies with the covariates, the Weibull distribution satisfies the proportional hazards assumption. In contrast, both the Deep Survival Machines (DSM) (Nagpal et al., 2021) and the proposed MENSA model, which utilize a mixture of Weibull distributions, do not satisfy this assumption. This can be attributed to two reasons:

1. In DSM and MENSA, both the shape parameter η and the scale parameter β depend on the covariates $\boldsymbol{x}^{(i)}$. Substituting an $\boldsymbol{x}^{(i)}$-dependent $\eta(\boldsymbol{x}^{(i)})$ into the baseline hazard definition prevents the hazard function from being expressed in a proportional hazards form, as shown in (36), thus violating the assumption.

2. Since both DSM and MENSA model the survival distribution as a mixture of Weibulls, their hazard functions can be expressed through a weighted combination of individual Weibull distributions. Thus, given mixture weights, $g_\psi(\boldsymbol{x}^{(i)})$, the PDF (5) and CDF (6) functions, their hazard function can be described as:

$$h\left(t;\,\beta,\eta,\boldsymbol{x}^{(i)}\right) = \frac{f\left(t;\,\beta,\eta,\boldsymbol{x}^{(i)}\right)}{S\left(t;\,\beta,\eta,\boldsymbol{x}^{(i)}\right)} = \frac{\sum_{\psi=1}^{\Psi} g_\psi(\boldsymbol{x}^{(i)})f_\psi\left(t;\,\beta_\psi(\boldsymbol{x}^{(i)}),\eta_\psi(\boldsymbol{x}^{(i)})\right)}{1-\sum_{\psi=1}^{\Psi} g_\psi(\boldsymbol{x}^{(i)})F_\psi\left(t;\,\beta_\psi(\boldsymbol{x}^{(i)}),\eta_\psi(\boldsymbol{x}^{(i)})\right)}. \tag{37}$$

Because both the shape and scale parameters, as well as the mixture weights, depend on the covariates $\boldsymbol{x}^{(i)}$, and the marginal density is modeled as a mixture of Weibull distributions, the resulting hazard function is not proportional across covariates or have a simple linear relationship in t or $\log(t)$. Unlike the traditional Weibull model, mixture-based approaches, such as DSM and MENSA, deviate from the proportional hazards assumption, offering greater flexibility and potentially more accurate representations of patient risk over time.