SurvTRACE: Transformers for Survival Analysis with Competing Events

Zifeng Wang and Jimeng Sun*

University of Illinois, Urbana-Champaign
zifengw2@illinois.edu, jimeng@illinois.edu

Abstract

In medicine, survival analysis studies the time duration to events of interest such as mortality. One major challenge is how to deal with multiple competing events (e.g., multiple disease diagnoses). In this work, we propose a transformer-based model that does not make the assumption for the underlying survival distribution and is capable of handling competing events, namely SurvTRACE. We account for the implicit *confounders* in the observational setting in multi-events scenarios, which causes selection bias as the predicted survival probability is influenced by irrelevant factors. To sufficiently utilize the survival data to train transformers from scratch, multiple auxiliary tasks are designed for multi-task learning. The model hence learns a strong shared representation from all these tasks and in turn serves for better survival analysis. We further demonstrate how to inspect the covariate relevance and importance through interpretable attention mechanisms of SurvTRACE, which suffices to great potential in enhancing clinical trial design and new treatment development. Experiments on METABRIC, SUPPORT, and SEER data with 470k patients validate the all-around superiority of our method.

1 Introduction

Time-to-event analysis, or survival analysis, studies the *probability* of event occurrence and the *timing* of the event with broad applications, including medicine (Friedman et al. 2015), reliability engineering (Peña and Hollander 2004), business analysis (Morrison 2004). It also allows us to handle *censored data*, i.e., we do not observe the occurrence of events due to the early stop of follow-ups, which has to be removed if we opt to use standard regression models.

Apart from data censoring, there are often multiple *competing risks* which can lead to the incidence of events. For instance, it is prevalent that the elder who carries malignant cancer also suffers from other chronic diseases like diabetes. Dealing with competing risks is more difficult than with a single event, thus much less explored in the literature. These works try to alleviate the performance loss due to the strong assumption that each event is independent. However, the *selection bias* in competing events survival data was seldom mentioned. Consider the targets $S(0)$ and $S(1)$ indicate the predicted survival probability for event 0 and event 1, respectively; $E \in \{0, 1\}$ means which event is actually observed; and X is the measured covariates of patients. Our estimated survival probability is unbiased only when $\{S(0), S(1)\} \perp\!\!\!\perp E$, i.e. exchangeable (Little and Rubin 2019). In the observational setting of survival data, as shown by Fig. 1, once an event was observed, others become *counterfactuals*. In this case, the model performance is influenced by the event incidence and is biased towards those events which happen more frequently, or *common events*. Likewise, the model will have much worse performance on *rare events*. In this sense, it is imperative to debias survival models such that it satisfies ignorability, i.e. $\{S(0), S(1)\} \perp\!\!\!\perp E \mid X$, by counterfactual learning (Schnabel et al. 2016).

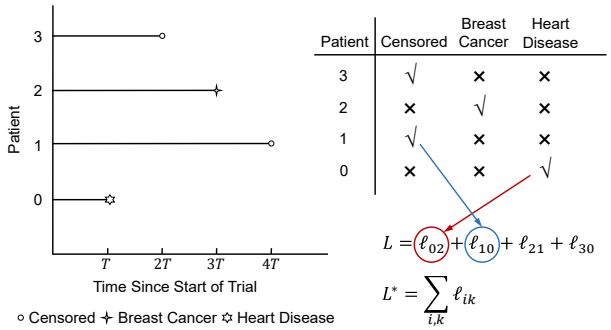


Figure 1: **Left**: Follow-up experience of 4 patients. **Right**: Patient-event table that shows event/censoring occurrence. ✓ indicates observed and ✗ is unobserved. Only the observed events can be used for training with the naive loss L and the true loss L^* is inaccessible.

On the other hand, deep learning (DL) was proved useful in enhancing survival analysis recently (Luck et al. 2017; Lee et al. 2018; Nagpal, Li, and Dubrawski 2021). However, they still suffer from insufficient training over rare events (Castañeda and Gerritse 2010). One reason is these models only use the basic multi-layer perceptron (MLP) with handcrafted features. The other challenge is that the current public survival data is either too small or too imbalanced to train strong DL models. That is, how to leverage strong DL models to boost survival analysis with limited survival data remains a challenge.

In this work, we propose SurvTRACE, which stands for **Surv**ival analysis using **TRA**ansformers with **C**ompeting

*Corresponding author.

Events. SurvTRACE is enabled by the following technical contributions:

1. **Debiasing competing events analysis using counterfactual learning.** We develop a learning method based on inverse propensity score (IPS) (Little and Rubin 2019) that remedies selection biases in survival data with competing risks. This method guarantees unbiased evaluation and learning of survival analysis models hence outperforms methods that ignore selection bias, especially on the analysis for rare events.
2. **Automatic feature engineering with attentive encoders.** We study how to automatically learn high-order interactions between covariates through attention (Devlin et al. 2019), to avoid manual feature engineering. Meanwhile, we inspect how the learned attention scores demonstrate relevance between covariates as well as show interpretability for the prediction results.
3. **Multi-task learning with a shared backbone.** We design multiple auxiliary tasks to make the best of limited survival data to train SurvTRACE from scratch. A shared representation is learned from all tasks then serves as a strong backbone for downstream tasks. This fashion strengthens the prediction accuracy for all events by sharing common knowledge.

We evaluate SurvTRACE on two open survival datasets and a large-scale dataset with 470k patients, where it outperforms the state-of-the-art baselines significantly. Software code implemented by PyTorch (Paszke et al. 2019) is available at https://github.com/RyanWangZf/SurvTRACE.

2 Model Architecture

In this section, we elaborate on the architecture and inference of SurvTRACE.

2.1 Problem Formulation

We assume our survival data consists of three parts $\mathcal{D} = \{(\mathbf{x}_i, t_i, e_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^D$ is D baseline covariates associated with the i-th patient; $t_i \in \mathbb{R}$ is the time at which an event of interest takes place or the time when the sample is censored; and e_i shows whether t_i is the event occurance or censored time. For single event scenarios, e_i is a binary indicator; For competing events scenarios, $e_i \in \{0, 1, \ldots, K_E\}$ tells which event happens at time t_i. When $e_i = 0$, the patient is said to be *right-censored* (i.e., no event is recorded at the end of the study for patient i).

The goal of survival analysis is to estimate the hazard and survival function. The survival function signifies the alive probability for one patient at time t, as $S(t) \triangleq \Pr(T > t)$. The hazard function is defined by

$$\lambda(t) \triangleq \lim_{\Delta t \to 0} \frac{\Pr(t \le T < t + \Delta t \mid T \ge t)}{\Delta t}, \tag{1}$$

which corresponds to the probability of death at time t given that the patient has survived up to that point. Likewise, we denote the probability mass function (PMF) of event time by $g(t) = \Pr(T = t)$.

2.2 SurvTRACE: Main Architecture

As illustrated in Fig. 2, SurvTRACE includes a baseline covariates embedding module, a deep-stacked attentive encoder module, and the alignment and subnetwork prediction module. Next we will present the technical details of these modules and then introduce how our model is used for training and inference.

Input & Embedding Module The raw baseline covariates describe the characteristics of patients. These covariates can be separated into two types: categorical and numerical. We set D_c and D_n as the number of categorical and numerical covariates, respectively. We denote the number of covariates as $D = D_c + D_n$. Categorical covariates are usually transformed into one-hot vectors before entering into the survival model. Here, we represent them in a d_e-dimensional space through an embedding matrix $\mathbf{V}_c \in \mathbb{R}^{D_c \times d_e}$ as

$$\mathbf{t}_i^m = \mathbf{V}_c \mathbf{x}_i^m, \tag{2}$$

where $\mathbf{x}_i \in \mathbb{R}^{D_c}$ is a one-hot vector for the category field m and $\mathbf{t}_i^m$ is the yielded embedding.

To allow interactions between numerical and categorical covariates, we represent numerical covariates in a low-dimensional space by

$$\mathbf{t}_i^j = \mathbf{v}_j x_i^j, \tag{3}$$

where x_i^j is a scalar for the j-th numerical field, $\mathbf{V}_n \in \mathbb{R}^{D_n \times d_e}$ is the embedding matrix for numerical features and $\mathbf{v}_j$ is the j-th row of $\mathbf{V}_n$. With both two types of embeddings at hand, we can concatenate them to obtain the representation $\mathbf{t}_i$ for all raw input covariates of the i-th patient, such that

$$\mathbf{t}_i = \mathbf{t}_i^1 \otimes \mathbf{t}_i^2 \cdots \otimes \mathbf{t}_i^D, \tag{4}$$

where $\otimes$ denotes concatenation operation.

Encoder Module To enable sufficient interactions between covariate embeddings, we use multi-head self-attention. The key-value attention mechanism allows the model to learn combinatorial interactions automatically.

In detail, the key idea is to obtain the each j-th field processed covariate embedding $\widetilde{\mathbf{t}}_i^j$ through a combination of other embeddings weighted by their correlation, as

$$\widetilde{\mathbf{t}}_i^j = \sum_{k=1}^{D} \alpha_{j,k} (\mathbf{W}_{\text{value}} \mathbf{t}_i^k), \tag{5}$$

where $\alpha_{j,k}$ signifies the the relevance score between covariate j and k; $\mathbf{W}_{\text{value}}$ is a learnable weight matrix to transform the raw embeddings to the same space. We omit the subscript i of $\mathbf{t}$ for avoiding clutter notations. $\alpha_{j,k}$ is produced by the softmax outputs of attention function ψ as

$$\alpha_{j,k} = \frac{\exp(\psi(\mathbf{t}^j, \mathbf{t}^k))}{\sum_{k'=1}^{D} \exp(\psi(\mathbf{t}^j, \mathbf{t}^{k'}))}, \tag{6}$$

where the attention function $\psi(\cdot, \cdot)$ could be an arbitrary function that maps two embeddings to a real-value output. Here, we set it as

$$\psi(\mathbf{t}^j, \mathbf{t}^k) = \langle \mathbf{W}_{\text{query}} \mathbf{t}^j, \mathbf{W}_{\text{key}} \mathbf{t}^k \rangle, \tag{7}$$

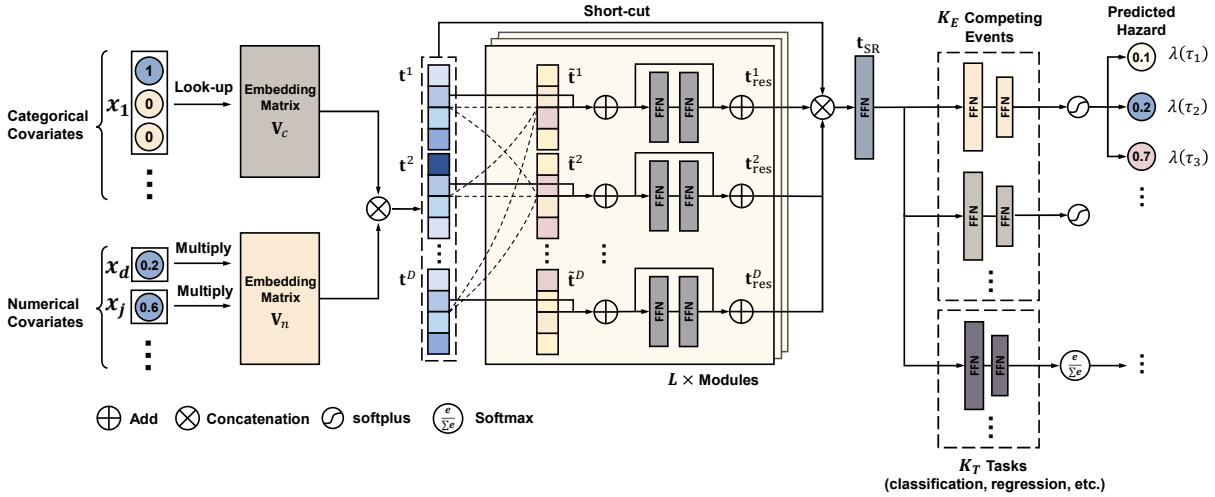


Figure 2: SurvTRACE architecture. The raw numerical and discrete covariates are encoded through two embedding matrices $\mathbf{V}_c$ and $\mathbf{V}_n$ separately. Two types of embeddings $(\mathbf{t}^1, \ldots, \mathbf{t}^D)$ are concatenated before going into the attentive encoder layers, where covariate embeddings interact to form high-order combinatorial embeddings $\widetilde{\mathbf{t}}^1, \ldots, \widetilde{\mathbf{t}}^D$. The yielded representations $\mathbf{t}_{\text{SR}}$ are shared across all task-specific sub-networks for downstream tasks. For survival analysis, hazard ratio $\lambda(\tau)$ of each discrete duration index τ is predicted.

which is an inner product of the transformed embedding by two weight matrices: $\mathbf{W}_{\text{query}}$ and $\mathbf{W}_{\text{key}}$.

The attention in Eq. (5) can be further enhanced by introducing H multi-heads to yield a series of diverse processed embeddings $\{\widetilde{\mathbf{t}}^{1,j} \ldots \widetilde{\mathbf{t}}^{H,j}\}$. Then the embedding for the field j is obtained by concatenation of $\widetilde{\mathbf{t}}^{h,j}$ from all heads

$$\widetilde{\mathbf{t}}^{j} = \widetilde{\mathbf{t}}^{1,j} \otimes \widetilde{\mathbf{t}}^{1,j} \ldots \otimes \widetilde{\mathbf{t}}^{H,j}. \tag{8}$$

Likewise, we have H pairs of attention parameters as $\{\mathbf{W}^h_{\text{query}}, \mathbf{W}^h_{\text{value}}, \mathbf{W}^h_{\text{key}}\}_{h=1}^{H}$.

To reserve the information of the raw input embeddings, a residual connection is added to yield the final embedding

$$\mathbf{t}^j_{\text{res}} = \text{SELU}(\mathbf{W}_{\text{res}}\widetilde{\mathbf{t}}^j + \mathbf{t}^j). \tag{9}$$

$\text{SELU}(\cdot)$ denotes the Scaled Exponential Linear Unit (SELU) activation function (Klambauer et al. 2017). The output embedding of the transformer layer is then obtained by another l_1-layer feed-forward network (FFN) with residual connection

$$\hat{\mathbf{t}}^j = \text{SELU}(\mathbf{W}^{(l_1)}_{\text{FFN}}(\cdots \mathbf{W}^{(2)}_{\text{FFN}}(\text{SELU}(\mathbf{W}^{(1)}_{\text{FFN}}\mathbf{t}^j))) + \mathbf{t}^j_{\text{res}}). \tag{10}$$

In a nutshell, from the first transformer, we transform the raw embedding $\mathbf{t}^j$ to the attentive embedding $\hat{\mathbf{t}}^j$. To encourage further interactions between covariates to get high-order combinatorial embeddings, we can stack l_2 transformers such that

$$\mathbf{t}^j \to \hat{\mathbf{t}}^{(1,j)} \to \cdots \hat{\mathbf{t}}^{(l_2,j)}, \tag{11}$$

hence $\hat{\mathbf{t}} = \hat{\mathbf{t}}^{(l_2,1)} \otimes \cdots \otimes \hat{\mathbf{t}}^{(l_2,D)}$ is the final representation for the patient generated by the stacked transformer encoder.

Shared Representation & Sub-networks Module SurvTRACE builds a shared representation $\mathbf{t}_{\text{SR}}$ from the encoder and serves for all downstream tasks, which enables the model to learn generalizable knowledge across all tasks and gets better performance for each task. Upon obtaining the representation $\hat{\mathbf{t}}$, we design a shortcut to concatenate it with $\mathbf{t}$, therefore utilize an alignment layer to transform them to the same space

$$\mathbf{t}_{\text{SR}} = \text{SELU}(\mathbf{W}_{\text{SR}}(\hat{\mathbf{t}} \otimes \mathbf{t})). \tag{12}$$

Based on the shared representation $\mathbf{t}_{\text{SR}}$, we can design many sub-networks for downstream tasks. These tasks can be split into two buckets: major tasks (survival analysis) and auxiliary tasks. Next, we will elaborate on how to deal with these tasks with different sub-network designs.

3 Task Design for Learning

3.1 Task I: Single-Event Survival Analysis

The ultimate goal of survival analysis is to estimate the survival function for individual patients, which is the PMF of survival time distribution. To make continuous-time hazard rate prediction feasible for neural networks, we parameterize the discrete-time hazard rate of events by a sub-network. Consider a time point set $\mathbb{T} = \{\tau_1, \ldots, \tau_m\}$ where τ_m is the pre-defined maximum follow-up time horizon. The discrete index set $\kappa(t) = \{1, \ldots, m\}$. In this scenario, the hazard rate at time τ_j is defined by

$$\lambda(\tau_j) = \Pr(T = \tau_j \mid T > \tau_{j-1}), \tag{13}$$

hence each corresponds to an output node of the sub-network. Likewise, we write the censored time as $T_C \in \mathbb{T}$.

First, let us take the single-event ($e \in \{0, 1\}$) survival analysis as the case. Assuming T and T_C are independent, we can write the likelihood function by

$$\begin{aligned}\Pr(T = t, E = e) = &[\Pr(T = t)\Pr(T_C \geq t)]^e \\ &\times [\Pr(T > t)\Pr(T_C = t)]^{1-e}.\end{aligned} \tag{14}$$

We can omit the terms which are only determined by censored time distribution (e.g., the PMF of censored time), then denote $S(t)$ and $g(t)$ by discrete hazard rate $\lambda(t)$ as

$$g(\tau_j) = \lambda(\tau_j)S(\tau_{j-1}),\ S(\tau_j) = [1-\lambda(\tau_j)]S(\tau_{j-1}). \quad (15)$$

With the assumption of constant hazard within each interval $[\tau_{j-1}, \tau_j]$, the piecewise constant hazard (PCH) loss (Kvamme and Borgan 2019) is defined by

$$\ell_i = -e_i \log \lambda(t_i) + \lambda(t_i)\rho(t_i) + \sum_{j=1}^{\kappa(t_i)-1} \exp[-\lambda(\tau_j)], \quad (16)$$

where $\rho(t)$ is the proportion of interval $\kappa(t)$ as time t as $\rho(t) = (t - \tau_{\kappa(t)-1})/(\tau_{\kappa(t)} - \tau_{\kappa(t)-1})$. With the shared representation $\mathbf{t}_{\text{SR}}$, the output network outputs the hazard rate prediction as

$$\lambda(t) = \log\left[1 + \exp[f(\kappa(t)|\mathbf{t}_{\text{SR}})]\right], \quad (17)$$

which is used for training by PCH loss in Eq. (16).

3.2 Task II: Debiasing Competing Events Survival Analysis

In competing events scenarios, e is no longer a binary indicator. Instead, we have K_E competing events as $e \in \{1, \ldots, K_E\}$. Denote $\mathbb{1}_{ik} = \mathbb{1}\{e_i = k\}$ and ℓ_{ik} is ℓ_i given $e_i = k$, a naive adaptation from Eq. (16) for competing events is to take

$$L_{\text{naive}} \triangleq \frac{1}{\sum_{i,k} \mathbb{1}_{ik}} \sum_{i,k} \mathbb{1}_{ik}\ell_{ik}. \quad (18)$$

The hazard prediction is performed by the attached cause-specific (CS) sub-networks as $\lambda_k(t) = \log[1 + \exp[f_k(\kappa(t)|\mathbf{t}_{\text{SR}})]]$ for $k = 1, \ldots, K_E$.

However, L_{naive} assumes events are independent but in reality ar often biased by the common events. This bias exaggerates with more imbalanced event distribution, which causes poor performance. Unfortunately, imbalanced event distribution is common in the real world. For instance, the 15% events in SEER data used (Lee et al. 2018) are breast cancer, while only 1% are cardiovascular diseases. To resolve it, we leverage the inverse propensity score (IPS) technique for debiasing. Denote $\pi_{ik} = \Pr(e_i = k|\phi, \mathbf{x})$ as the estimate of $\Pr(e_i = k)$, a.k.a propensity score, we derive a novel IPS-based PCH loss as

$$L_{\text{IPS}} \triangleq \frac{1}{nK_E} \sum_{i,k} \frac{\mathbb{1}_{ik}\ell_{ik}}{\pi_{ik}}. \quad (19)$$

Here, $\phi(\mathbf{x})$ is a logistic regression model

$$\pi_{ik} = \phi(\mathbf{x}_i) \triangleq \sigma(\mathbf{w}^\top \mathbf{x}_i + \beta), \quad (20)$$

where β denotes the offset; $\sigma(\cdot)$ is sigmoid function. Please refer to Appendix A for the proof of why L_{IPS} is unbiased with further explanation.

3.3 Auxiliary Tasks: Multi-task Learning

Multi-task Learning (MTL) puts the model to learn from multiple related tasks with shared representations, thus enabling the model to generalize better on the targeted task. Besides survival analysis, we design two auxiliary tasks for enhancing the representation learning: mortality prediction (MP) and length-of-stay prediction (LS).

For the mortality prediction task, we urge the model to learn to predict if there will be an event happening ($\delta = 1$ if $e > 0$) during the whole time

$$L_{\text{MP}} = -\frac{1}{n}\sum_i \left[\delta_i \log \hat{y}_i + (1-\delta_i)\log(1-\hat{y}_i)\right], \quad (21)$$

where $\hat{y}_i$ is predicted by the task-specific (TS) sub-network.

For the length-of-stay prediction task, the model predicts how much time the event happens or becomes censored after the initial observation

$$L_{\text{LS}} = \frac{1}{n}\sum_i (\hat{t}_i - t_i)^2, \quad (22)$$

where $\hat{t}_i$ is the predicted event time. Afterwards, we can write the final loss function by

$$\mathcal{L} = L_{\text{IPS}} + \gamma_1 L_{\text{MP}} + \gamma_2 L_{\text{LS}}. \quad (23)$$

Two hyperparameters γ_1 and γ_2 can be set to 1 initially and then annealed in the following training.

4 Experiment

In this section, we resort to focus on the following four research questions:

- **RQ1.** Does high-order covariates interaction with transformers encourage better performance?
- **RQ2.** How much does selection bias harm the competing events survival analysis?
- **RQ3.** Does MTL help learn a stronger encoder of `SurvTRACE` for survival analysis?
- **RQ4.** What is the insight `SurvTRACE` can offer by its interpretable function?

We will first present the setups then discuss each of the RQs one by one.

4.1 Experimental Setup

Datasets For single event survival analysis, we evaluate models on two real-world medical datasets: Study to Understand Prognoses Preferences Outcomes and Risks of Treatment (**SUPPORT**) (Knaus et al. 1995) and Molecular Taxonomy of Breast Cancer International Consortium (**METABRIC**) (Curtis et al. 2012). For competing events, we collect and proceed with the data from Surveillance, Epidemiology, and End Results Program (**SEER**)[1].

SUPPORT. It is a multicenter study designed to examine outcomes and clinical decision-making for seriously ill hospitalized patients. The version we utilize comes from the pycox[2] package (Kvamme and Borgan 2019) following the preprocessing steps in (Katzman et al. 2018).

METABRIC. This data uses gene and protein expression profiles to determine new breast cancer subgroups to help

[1] https://seer.cancer.gov/

[2] https://github.com/havakv/pycox

Dataset		No. Events	No. Censored	No. Covariates	Event Duration			Censoring Time		
				(real, categorical)	min	max	mean	min	max	mean
METABRIC		1,103 (57.9%)	801 (42.1%)	9 (5, 4)	0.1	355.2	99.9	0	337	159.5
SUPPORT		6,036 (68.0%)	2,837 (32.0%)	14 (8, 6)	3	1944	205.4	344	2029	1059.8
SEER	**BC**	87,495 (18.4%)	367,702 (77.1%)	18 (4, 14)	1	121	40.2	1	121	74.7
	HD	21,549 (4.5%)			1	121	53.4			

Table 1: Descriptive statistics of datasets. BC and HD are shorthands of breast cancer and heart diseases, respectively.

Algorithms	METABRIC			SUPPORT		
	25%	50%	75%	25%	50%	75%
CPH	0.628(0.024)	0.627(0.020)	0.632(0.016)	0.549(0.017)	0.564(0.004)	0.586(0.005)
DeepSurv	0.660(0.028)	0.648(0.022)	0.644(0.018)	0.594(0.013)	0.591(0.007)	0.605(0.006)
DeepHit	0.712(0.026)	0.657(0.023)	0.603(0.014)	0.650(0.009)	0.602(0.013)	0.574(0.009)
RSF	0.698(0.029)	0.658(0.022)	0.630(0.017)	0.660(0.005)	0.621(0.006)	0.602(0.006)
PC-Hazard	0.713(0.024)	0.680(0.017)	0.644(0.017)	0.652(0.011)	0.620(0.008)	0.607(0.008)
DSM	0.707(0.023)	0.663(0.014)	0.636(0.017)	0.640(0.007)	0.609(0.007)	0.596(0.008)
SurvTRACE w/o MTL	0.722(0.022)	0.686(0.010)	0.649(0.017)	0.665(0.008)	0.630(0.006)	0.614(0.005)
SurvTRACE	**0.728(0.019)**	**0.690(0.013)**	**0.655(0.013)**	**0.670(0.008)**	**0.633(0.006)**	**0.617(0.004)**

Table 2: C^{td} for **METABRIC** and **SUPPORT** datasets at different quantiles of event times; Values in the bracket show the standard deviation of performances of 10 runs.

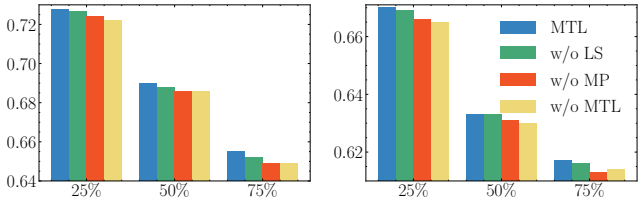
physicians provide better treatment. We utilize the data from pycox as well.

SEER. This is an authoritative source for cancer statistics in the US. We select breast cancer patients registered from 2004 to 2014, with the follow-up period restricted to 10 years. Among all these patients, we select who also suffer from heart diseases, which renders a large-scale dataset with 476,746 patients. Therefore, we treat breast cancer and heart diseases as two competing events. We include 18 covariates, including age, race, gender, diagnostic confirmation, morphology information (primary site, laterality, histologic type, etc.), tumor information (size, type, number, etc.), and surgery information. We fill missing values with the mean of numerical covariates and mode of categorical covariates. The statistics of all datasets are available in Table 1.

Evaluation Metrics We make use of time-dependent concordance index (C^{td}) (Antolini, Boracchi, and Biganzoli 2005) for the performance evaluation of k-th event, as

$$C^{\text{td}}(\tau, k) = \Pr\{S_k(\tau|\mathbf{x}_i) > S_k(\tau|\mathbf{x}_j) \mid e_i = k, t_i < t_j, t_i \leq \tau, k > 0\}. \quad (24)$$

Here, $S_k(t|\mathbf{x}_i)$ is the predicted survival function considering the k-th event at the truncation time τ. We adjust the estimate with an inverse probability of censoring weighted (IPCW) estimate to obtain an unbiased estimate following (Uno et al. 2011). Since C^{td} at different time horizons indicate how models capture the possible changes in risk over time, we follow (Nagpal, Li, and Dubrawski 2021) to report C^{td} at different truncated time quantiles of 25%, 50%, and 75%.



(a) On **METABRIC**. (b) On **SUPPORT**.

Figure 3: Ablation study of MTL for SurvTRACE on **METABRIC** and **SUPPORT**. We compare the full MTL model with models without length-of-stay (LS) prediction, mortality prediction (MP), and without MTL, respectively. x-axis and y-axis are time horizons and C^{td}.

Baselines We pick the following baselines for comparison: Cox Proportional Hazards (CPH) (Cox 1972), Random Survival Forests (RSF) (Ishwaran et al. 2008), DeepSurv (Katzman et al. 2018), DeepHit (Lee et al. 2018), Piecewise Constant Hazard (PC-Hazard) (Kvamme and Borgan 2019), and Deep Survival Machines (DSM) (Nagpal, Li, and Dubrawski 2021).

For competing event survival analysis, we pick DeepHit and DSM, which apply to these cases. We also utilize cause-specific CPH (CS-CPH) and cause-specific PC-Hazard (CS-PC-Hazard) by assuming the independence of events (Haller, Schmidt, and Ulm 2013). These CS-based methods learn from each competing event separately by treating others as censored.

Implementation We use Adam (Kingma and Ba 2014) as the optimizer to train SurvTRACE in all experiments, with learning rate in $\{1e^{-4}, 1e^{-3}\}$, weight decay in

Algorithms	25%		50%		75%	
	HD	Breast Cancer	HD	Breast Cancer	HD	Breast Cancer
CS-CPH	N/A	0.828(1.5e-3)	N/A	0.799(1.1e-3)	N/A	0.781(7e-4)
CS-PC-Hazard	0.774(5.1e-3)	0.895(1.7e-3)	0.769(3.3e-3)	0.875(1.6e-3)	0.766(3.9e-3)	0.858(4e-4)
DeepHit	0.763(1.6e-2)	0.896(2.2e-3)	0.748(1.5e-2)	0.875(2.7e-3)	0.724(1.13-2)	0.853(1.6e-3)
DSM	0.765(4.6e-3)	0.895(1.2e-3)	0.761(3.9e-3)	0.873(2.0e-3)	0.750(2.3e-3)	0.856(1.2e-3)
SurvTRACE w/o IPS	0.789(6.3e-3)	0.902(1.2e-3)	0.780(5.0e-3)	0.882(1.3e-3)	0.768(2.7e-3)	0.864(9e-4)
SurvTRACE w/o MTL	0.793(6.4e-3)	0.903(1.1e-3)	0.784(5.4e-3)	0.881(1.5e-3)	0.768(3.1e-3)	0.863(5e-4)
SurvTRACE	**0.797(6.2e-3)**	**0.904(1.2e-3)**	**0.788(5.5e-3)**	**0.883(1.2e-3)**	**0.775(3.1e-3)**	**0.866(5e-4)**

Table 3: C^{td} for competing risks on **SEER** dataset; Values in the bracket show the standard deviation of 10 runs.

$\{1e^{-3}, 1e^{-4}, 0\}$. The number of transformer layers is chosen from $\{2, 3, 4\}$, the embedding size is selected from $\{8, 16\}$, the intermediate layer size is picked from $\{32, 64\}$, and the number of attention heads is set from $\{1, 2, 4\}$. The cause-specific and task-specific sub-networks are MLPs with one or two layers with the same intermediate size as the transformers and ReLU activation.

We use 30% data as the test set, 10% data as the validation set, and 60% as the training set. We report the mean and standard deviation of metrics with 10 multiple runs on different train/validation/test splits.

4.2 Performance Comparison in Single Event (RQ1)

We compare SurvTRACE with baselines on single event datasets. Results are reported in Table 2, where the best-performing method is shown in bold. We find that SurvTRACE without MTL consistently outperforms all baselines across two datasets in terms of C^{td} under all time horizons. The reasons for this improvement are multi-facet: (1) SurvTRACE leverages multi-head attention to build rich interactions among covariates and dynamically adjust to different inputs; (2) The stacked attentive encoders provide higher-order covariate conjunctions, thus mining more complicated patterns from the survival data. Furthermore, SurvTRACE gets better results when engaged with MTL.

We also observe that all methods experienced performance deterioration on longer horizons like 75%. As more patients are involved in the evaluation, it is harder for models to predict the orders of event time of all patients. Nevertheless, SurvTRACE performs better across for all horizons with the relative improvement of $2.1\%, 1.5\%, 1.7\%$ on METABRIC and $1.5\%, 1.9\%, 1.6\%$ on SUPPORT over the best baselines.

4.3 Study of SurvTRACE in Competing Events (RQ2)

We compare SurvTRACE with baselines on **SEER** data. Results are reported in Table 3. N/A in the table means the method does not converge for the corresponding event analysis. Through experiments, we find:

- In competing events scenarios, the performance for rare events is much worse than for common events while SurvTRACE works better on HD and BC than baselines. We credit it to (1) IPS offers an unbiased estimate of objectives hence assisting in balancing predictions for events. In contrast, without IPS, SurvTRACE has around 0.1 reduction of C^{td} on HD on the 25% horizon; (2) MTL further enhances the generalizability of representations thus yielding a better performance for the target task.
- Except for CS-CPH, other baselines have similar performance for both HD and BC. SurvTRACE wins over baselines by a significant margin: it reaches 0.797, 0.788, 0.775 for HD on each time horizon, respectively, which are $3.0\%, 2.5\%, 1.2\%$ better than the best baselines. This demonstrates the usability of the cutting-edge deep learning techniques for gaining improvement for survival analysis. More importantly, comparing to the reduced version, SurvTRACE with MTL and IPS gets the best performance, which signifies the need of considering selection bias in competing events and using MTL to make the best of the data.

4.4 In-depth Analysis (RQ3 & RQ4)

Multi-task Learning We analyze the utility that multi-task learning can add, results are shown on the bottom row of Tables 2 & 3. It illustrates that MTL suffices to improve the training across all datasets. Specifically, on SEER, SurvTRACE achieves much better performance than SurvTRACE on the rare event HD. We owe this to the auxiliary mortality prediction and time prediction tasks which enhance representation learning for transformers, thus utilizing the survival data more sufficiently.

To study the contribution of each task, we conduct an ablation study, shown by Fig. 3. We identify that both auxiliary tasks strengthen the model performance, as the model w/o MTL has the worst C^{td}. It validates the effectiveness of MTL in enhancing representation learning of SurvTRACE. Note that the MP task renders more gain than the LS task, which shows that the MP task is more relevant to survival analysis. It tells that designing auxiliary tasks similar to major task benefit the model more.

Interpretability We investigate the attention scores between different covariates. Samples of two patients are shown in Fig. 4. The first patient (on the left) takes hormone treatment. Likewise, the HT indicator shows a sig-

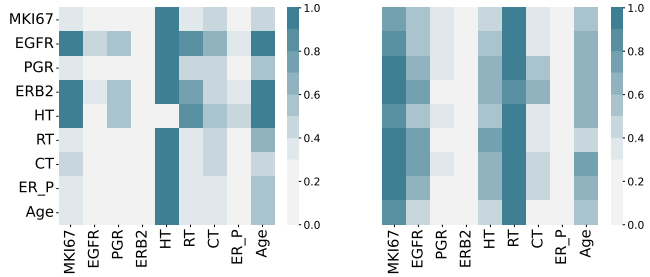
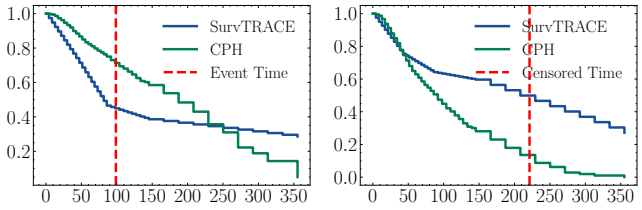


Figure 4: Visualization of attention scores between covariates for two patients; MKI67, EGFR, PGR, ERB2, and ER, are gene biomarkers; HT: Hormone treatment; RT: Radiotherapy; CT: Chemotherapy; ER_P: ER Positive.



(a) Predicted survival function for **uncensored** data.

(b) Predicted survival function for **censored** data.

Figure 5: Predicted survival function (by interpolation) by SurvTRACE and CPH, respectively. x-axis shows the duration time points; y-axis shows the probability. Dotted lines stand on the point where the event/censoring happens.

nificant correlation to almost all the rest covariates, which makes it an important factor for predicting the outcomes. The second patient (on the right) takes radiotherapy, and we observe the same degree of saliency in the case. For both two patients, the treatment indicator is deeply correlated to their age, pointing out the influence of age on the effectiveness of specific therapies.

An interesting finding is that the ER-positive indicator seems to have a low effect on other factors. About 85% of all breast cancers are Estrogen Receptor (ER) positive, which means most patients have the same value for this term. The model downweighs the attention over it because it does not offer much additional information for discriminating survival probability across patients. Moreover, we identify the MKI67 biomarker plays a significant role in other factors, including EGFR and ERB2.

Case Study To better visualize the superiority of SurvTRACE, we plot examples of predicted survival functions by SurvTRACE and CPH, respectively (Fig. 5). We identify that for uncensored data, our model senses the event happening, and the predicted probability decreases sharply. On the contrary, CPH fails to capture the signal of events and gives relatively even hazard prediction across the whole period. On the other hand, for the censored data, SurvTRACE also succeeds in maintaining a high survival probability before the censored time point.

5 Related Work

The thriving need for survival analysis encourages a plethora of statistical methods. For instance, the Cox proportional hazards model (CPH) (Cox 1972) which is multivariate linear regression. To enhance CPH, multi-task learning (Li et al. 2016a), transfer learning (Li et al. 2016b), active learning (Vinzamuri, Li, and Reddy 2014) were used. On the other hand, new models advanced by machine learning (ML), e.g., survival support vector machine (Van Belle et al. 2007; Pölsterl, Navab, and Katouzian 2015), random survival forests (RSF) (Ishwaran et al. 2008), gradient boosting (Wang et al. 2020a), were proposed. There were also attempts using neural networks for learning representations of covariates (Luck et al. 2017; Katzman et al. 2018; Lee et al. 2018; Ren et al. 2019; Nagpal, Li, and Dubrawski 2021). However, these NN-based methods do not fully exploit the power of NNs as they only use simple multi-layer perceptron, which is inherently limited in its learning capacity. More importantly, few of them are interpretable such that it is unclear what the black-box model learns and how we gain insight from the predictions for applications, e.g., identifying risk factors (Koene et al. 2016) and guiding design of clinical trials (Liu et al. 2021). Readers may refer to (Wang, Li, and Reddy 2019) for a survey of ML-based survival analysis. Different from the above, our model utilizes the cutting-edge transformers as the backbone that also provides the means of interpretability.

In competing events scenarios, many works assume each event is independent and handle each event separately by setting others to be censored (Ranganath et al. 2016; Chapfuwa et al. 2018; Nagpal, Li, and Dubrawski 2021). The existing competing event analysis methods (Lee et al. 2018; Bellot and Schaar 2018a,b; Lee, Yoon, and Schaar 2019; Tjandra, He, and Wiens 2021; Rahman et al. 2021) try to weaken the event independent assumption but do not consider bias and imbalance in survival data. Selection bias in multi-event data, shown by Fig. 1, causes naive multi-event loss a biased estimate of the true loss when the occurrences of events are covariate-dependent and the rare events are under-represented (Wang et al. 2020b). This bias was hardly discussed in the survival analysis literature, which makes the most difference of our work from the others.

6 Conclusion

In summary, we propose a multi-task transformer-based survival analysis network, namely SurvTRACE, which can handle both censored data and competing risks. Specifically, we take the implicit bias in censoring survival data into account and propose to debias through counterfactual learning. We also design two auxiliary tasks to utilize limited survival data for representation learning of SurvTRACE. According to the visualization of the attention module engaged in SurvTRACE, we can provide a case-by-case explanation for each individual. Our future work will further take time-varying covariates and multimodal data into consideration for enhancing survival analysis.

References

Antolini, L.; Boracchi, P.; and Biganzoli, E. 2005. A time-dependent discrimination index for survival data. *Statistics in Medicine*, 24(24): 3927–3944.

Bellot, A.; and Schaar, M. 2018a. Multitask boosting for survival analysis with competing risks. *Advances in Neural Information Processing Systems*, 31: 1390–1399.

Bellot, A.; and Schaar, M. 2018b. Tree-based Bayesian mixture model for competing risks. In *International Conference on Artificial Intelligence and Statistics*, 910–918. PMLR.

Castañeda, J.; and Gerritse, B. 2010. Appraisal of several methods to model time to multiple events per subject: modelling time to hospitalizations and death. *Revista Colombiana de Estadística*, 33(1): 43–61.

Chapfuwa, P.; Tao, C.; Li, C.; Page, C.; Goldstein, B.; Duke, L. C.; and Henao, R. 2018. Adversarial time-to-event modeling. In *International Conference on Machine Learning*, 735–744. PMLR.

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

Curtis, C.; Shah, S. P.; Chin, S.-F.; Turashvili, G.; Rueda, O. M.; Dunning, M. J.; Speed, D.; Lynch, A. G.; Samarajiwa, S.; Yuan, Y.; et al. 2012. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature*, 486(7403): 346–352.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, 4171–4186.

Friedman, L. M.; Furberg, C. D.; DeMets, D. L.; Reboussin, D. M.; and Granger, C. B. 2015. *Fundamentals of clinical trials*. Springer.

Haller, B.; Schmidt, G.; and Ulm, K. 2013. Applying competing risks regression models: an overview. *Lifetime Data Analysis*, 19(1): 33–58.

Ishwaran, H.; Kogalur, U. B.; Blackstone, E. H.; and Lauer, M. S. 2008. Random survival forests. *The Annals of Applied Statistics*, 2(3): 841–860.

Katzman, J. L.; Shaham, U.; Cloninger, A.; Bates, J.; Jiang, T.; and Kluger, Y. 2018. DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1): 1–12.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klambauer, G.; Unterthiner, T.; Mayr, A.; and Hochreiter, S. 2017. Self-normalizing neural networks. In *International Conference on Neural Information Processing Systems*, 972–981.

Knaus, W. A.; Harrell, F. E.; Lynn, J.; Goldman, L.; Phillips, R. S.; Connors, A. F.; Dawson, N. V.; Fulkerson, W. J.; Califf, R. M.; Desbiens, N.; et al. 1995. The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults. *Annals of Internal Medicine*, 122(3): 191–203.

Koene, R. J.; Prizment, A. E.; Blaes, A.; and Konety, S. H. 2016. Shared risk factors in cardiovascular disease and cancer. *Circulation*, 133(11): 1104–1114.

Kvamme, H.; and Borgan, Ø. 2019. Continuous and discrete-time survival prediction with neural networks. *arXiv preprint arXiv:1910.06724*.

Lee, C.; Yoon, J.; and Schaar, M. 2019. Dynamic-Deephit: A deep learning approach for dynamic survival analysis with competing risks based on longitudinal data. *IEEE Transactions on Biomedical Engineering*, 67(1): 122–133.

Lee, C.; Zame, W. R.; Yoon, J.; and Schaar, M. 2018. DeepHit: A deep learning approach to survival analysis with competing risks. In *AAAI Conference on Artificial Intelligence*.

Li, Y.; Wang, J.; Ye, J.; and Reddy, C. K. 2016a. A multi-task learning formulation for survival analysis. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1715–1724.

Li, Y.; Wang, L.; Wang, J.; Ye, J.; and Reddy, C. K. 2016b. Transfer learning for survival analysis via efficient l2, 1-norm regularized cox regression. In *IEEE International Conference on Data Mining*, 231–240. IEEE.

Little, R. J.; and Rubin, D. B. 2019. *Statistical analysis with missing data*, volume 793. John Wiley & Sons.

Liu, R.; Rizzo, S.; Whipple, S.; Pal, N.; Pineda, A. L.; Lu, M.; Arnieri, B.; Lu, Y.; Capra, W.; Copping, R.; et al. 2021. Evaluating eligibility criteria of oncology trials using real-world data and AI. *Nature*, 592(7855): 629–633.

Luck, M.; Sylvain, T.; Cardinal, H.; Lodi, A.; and Bengio, Y. 2017. Deep learning for patient-specific kidney graft survival analysis. *arXiv preprint arXiv:1705.10245*.

Morrison, J. 2004. Introduction to survival analysis in business. *Journal of Business Forecasting Methods and Systems*, 23(1): 18–22.

Nagpal, C.; Li, X. R.; and Dubrawski, A. 2021. Deep survival machines: Fully parametric survival regression and representation learning for censored data with competing risks. *IEEE Journal of Biomedical and Health Informatics*.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32: 8026–8037.

Peña, E. A.; and Hollander, M. 2004. Models for recurrent events in reliability and survival analysis. In *Mathematical reliability: An expository perspective*, 105–123. Springer.

Pölsterl, S.; Navab, N.; and Katouzian, A. 2015. Fast training of support vector machines for survival analysis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 243–259. Springer.

Rahman, M. M.; Matsuo, K.; Matsuzaki, S.; and Purushotham, S. 2021. DeepPseudo: Pseudo value based deep learning models for competing risk analysis. In *AAAI Conference on Artificial Intelligence*, volume 35, 479–487.

Ranganath, R.; Perotte, A.; Elhadad, N.; and Blei, D. 2016. Deep survival analysis. In *Machine Learning for Healthcare Conference*, 101–114. PMLR.

Ren, K.; Qin, J.; Zheng, L.; Yang, Z.; Zhang, W.; Qiu, L.; and Yu, Y. 2019. Deep recurrent survival analysis. In *AAAI Conference on Artificial Intelligence*, volume 33, 4798–4805.

Schnabel, T.; Swaminathan, A.; Singh, A.; Chandak, N.; and Joachims, T. 2016. Recommendations as treatments: Debiasing learning and evaluation. In *International Conference on Machine Learning*, 1670–1679. PMLR.

Tjandra, D.; He, Y.; and Wiens, J. 2021. A hierarchical approach to multi-event survival analysis. In *AAAI Conference on Artificial Intelligence*, volume 35, 591–599.

Uno, H.; Cai, T.; Pencina, M. J.; D'Agostino, R. B.; and Wei, L.-J. 2011. On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in Medicine*, 30(10): 1105–1117.

Van Belle, V.; Pelckmans, K.; Suykens, J.; and Van Huffel, S. 2007. Support vector machines for survival analysis. In *International Conference on Computational Intelligence in Medicine and Healthcare*, 1–8.

Vinzamuri, B.; Li, Y.; and Reddy, C. K. 2014. Active learning based survival regression for censored data. In *ACM International Conference on Information and Knowledge Management*, 241–250.

Wang, P.; Li, Y.; and Reddy, C. K. 2019. Machine learning for survival analysis: A survey. *ACM Computing Surveys*, 51(6): 1–36.

Wang, X.; Pakbin, A.; Mortazavi, B.; Zhao, H.; and Lee, D. 2020a. BoXHED: Boosted eXact hazard estimator with dynamic covariates. In *International Conference on Machine Learning*, 9973–9982. PMLR.

Wang, Z.; Chen, X.; Wen, R.; Huang, S.-L.; Kuruoglu, E.; and Zheng, Y. 2020b. Information theoretic counterfactual learning from missing-not-at-random feedback. *Advances in Neural Information Processing Systems*, 33: 1854–1864.

A Unbiased Estimate using IPS-based Loss

Proposition 1 (IPS Loss is Unbiased Estimate of True Loss)**.** *The proposed inverse propensity score based logistic hazard loss L_{IPS} is the unbiased estimate of true risk L^*, as*

$$L^* \triangleq \frac{1}{nK_E} \sum_{i,k} \ell_{ik}. \tag{1}$$

Proof. Let's first see why the naive LH loss is biased with selection bias censoring. The naive LH loss is defined by

$$L_{\text{naive}} = \frac{1}{\sum_{i,k} \mathbb{1}_{ik}} \sum_{i,k} \mathbb{1}_{ik} \ell_{ik}. \tag{2}$$

If we take the expectation of it on the event indicator e, we shall get

$$\begin{aligned} \mathbb{E}_e[L_{\text{naive}}] &= \frac{1}{\sum_{i,k} \mathbb{1}_{ik}} \sum_{i,k} \mathbb{E}_e[\mathbb{1}_{ik}] \ell_{ik} \\ &= \frac{1}{\sum_{i,k} \mathbb{1}_{ik}} \sum_{i,k} \Pr(e_i = k) \ell_{ik} \\ &\neq L^*. \end{aligned} \tag{3}$$

This is because the censoring of an event is not at random, and the probability of event occurrences depends on the patient's characteristics, which are so-called *confounders* under the context of counterfactual learning. Therefore, we build a new estimator L_{IPS} as

$$\mathbb{E}_e[L_{\text{IPS}}] = \frac{1}{nK_E} \sum_{i,k} \frac{\mathbb{E}_e[\mathbb{1}_{ik}] \ell_{ik}}{\pi_{ik}} \tag{4}$$

$$= \frac{1}{nK_E} \sum_{i,k} \frac{\Pr(e_i = k) \ell_{ik}}{\pi_{ik}} \tag{5}$$

$$= \frac{1}{nK_E} \sum_{i,k} \ell_{ik} \tag{6}$$

$$= L^*. \tag{7}$$

In detail, we require an IPS estimator to obtain π_{ik} due to the *observational* setting: the patients are part of the assignment mechanism that generates the observational matrix. In other words, the appearance of events is covariate-dependent. Hence, we ought to estimate the propensity score π from the observational matrix ourselves, as done by Eq. (20). □