Tabular Foundation Models Can Do Survival Analysis

Anonymous authors
Paper under double-blind review

Abstract

While tabular foundation models have achieved remarkable success in classification and regression, adapting them to model time-to-event outcomes for survival analysis is non-trivial due to right-censoring, where data observations may end before the event of interest occurs. We utilize a classification-based framework that reformulates both static and dynamic survival analysis as a series of binary classification problems by discretizing event times. Censored observations are naturally handled as examples with missing labels at certain time points. This classification formulation enables existing tabular foundation models (TFMs) to perform survival analysis through in-context learning without explicit training. In contrast to classical approaches that use binary classifiers to model discrete-time hazards, our approach directly models cumulative failure probabilities, which we find empirically to be more robust to the number of discretization bins by avoiding multiplicative accumulation of per-bin errors. We prove that under standard censoring assumptions, minimizing our binary classification loss recovers the true survival probabilities as the training set size increases. We demonstrate through evaluation across 48 real-world datasets (43 static and 5 dynamic) that off-the-shelf TFMs with this classification formulation outperform classical and deep learning baselines on average over multiple survival metrics.

1 Introduction

Survival analysis or time-to-event analysis models *when* an event of interest will occur, rather than just *whether* it will happen or not (Wang et al., 2019). Applications of survival analysis span many domains, including healthcare (e.g., time to disease recurrence or mortality), social sciences (e.g., duration of unemployment), engineering (e.g., time to equipment failure), and finance/business (e.g., time to default or customer churn). A defining feature of survival analysis that differentiates it from standard regression is *censoring*, where the true event time is only partially observed (Klein & Moeschberger, 2006). Under right-censoring, the exact time an event of interest occurs may be unknown because the observation period may end before the event happens. For example, consider a 5-year clinical drug trial where mortality is the main outcome; the mortality event may be right-censored if the individual drops out early or remains alive through the 5-year observation period. This incomplete event time information distinguishes survival analysis from standard regression tasks, where the target is always observed.

Despite its broad applicability, survival analysis often remains challenging due to (a) right-censoring, (b) small dataset sizes, e.g., a several hundred samples or fewer (Drysdale, 2022), and (c) covariates that may be irregularly sampled over time (Wang et al., 2019). While foundation models have been developed for survival analysis in specialized medical tasks (Steinberg et al., 2024; Vu et al., 2025), for general-purpose survival analysis, it remains standard to use specialized models fit from scratch on a single dataset.

Recently, tabular foundation models (TFMs) (Hollmann et al., 2023; Grinsztajn et al., 2025; Zhang et al., 2025; Gardner et al., 2024) have achieved remarkable success in classification and regression tasks, often outperforming methods like XGBoost, especially in small-data regimes. TFMs are usually transformer-based models, pretrained on a diverse collection of synthetic or real-world datasets, that are able to perform in-context learning at inference time without parameter updates. While this paradigm offers a promising direction for survival analysis, adapting TFMs to this setting is non-trivial. Standard regression architectures cannot easily account for censoring, a challenge that historically necessitated the development of specialized

variants for survival analysis, e.g., Random Survival Forest (Ishwaran et al., 2008). Additionally, while TFMs are primarily evaluated on classification accuracy or mean squared error, survival modeling requires well-calibrated probabilities that accurately characterize risk dynamics over time.

In this work, we cast survival analysis as a series of binary classification problems that involve modeling the cumulative failure probability. This allows survival models to be optimized using a standard binary cross-entropy loss and furthermore enables TFM classifiers to perform survival analysis out-of-the-box via in-context learning, leveraging pretrained representations to achieve high performance without requiring additional model training or hyperparameter tuning. Our approach discretizes event times, a common practice in the survival analysis literature (Efron, 1988; Craig et al., 2025; Allison, 1982; Singer & Willett, 1993; Lee et al., 2019; Yu et al., 2011; Maystre & Russo, 2022), and is applicable to both static and dynamic survival modeling. Our contributions are as follows:

(1) Classification formulation of survival analysis that models cumulative failure probabilities directly. In this work, we show that the exact classification formulation used with TFMs can significantly impact performance. We find empirically that our cumulative failure modeling formulation outperforms other classification formulations with TFMs—including formulations that model the discrete hazard hazard rate (Efron, 1988; Allison, 1982; Craig et al., 2025; Zhong & Tibshirani, 2019) and a multi-class classification formulations used in other work (Qi et al., 2026). Compared to concurrent work developing survival-specific foundation models (Seletkov et al., 2026; Qi et al., 2026), our formulation is flexible in that changing the number or placement of the discretization bins does not require retraining the model (see Section 3.2 for a detailed discussion).

(2) Consistent model guarantee as the training set grows. We prove that minimizing our binary cross-entropy loss recovers the true underlying survival probabilities as the training sample size grows. This consistency result holds under a conditional independent censoring assumption that is common in the survival analysis literature (Uno et al., 2011; Kalbfleisch & Prentice, 2002). We also show empirically that classification models with lower binary cross-entropy loss have more accurate survival probabilities (Figure 3).

(3) Empirical evaluation on a suite of real-world survival analysis datasets. We evaluate on 43 static and 5 dynamic real-world survival analysis datasets from *SurvSet* (Drysdale, 2022). Across multiple common survival metrics, TFMs using our classification formulation outperform hazard-based and multi-class formulations with TFMs and a range of classical and deep learning-based survival models on average.

2 Problem statement

We consider both *static* and *dynamic* survival analysis (Chen et al., 2024; Van Houwelingen & Putter, 2011). While the static survival analysis models event times given baseline features collected at the start of observation, dynamic models update predictions as more information becomes available.

2.1 Static setting

Each dataset consists of a collection of n i.i.d. data tuples $\{(X_{i,0}, T_i, C_i, \delta_i)\}_{i=1}^n$. Here, $X_{i,0} \in \mathbb{R}^d$ is a feature vector consisting of baseline covariates given at the start of the observation period. $T_i > 0$ represents the (latent) event time and $C_i > 0$ is the censoring time. We use $\delta_i \triangleq \mathbb{1}(T_i \leq C_i)$ as an indicator of whether the event is observed or censored. Note, $\delta_i = 1$ if the event is observed and $\delta_i = 0$ if censored.

Our primary objective is to estimate the survival probability $S(t \mid X_{i,0})$, which represents the probability that the event occurs after time t:

$$S(t \mid X_{i,0}) \triangleq \mathbb{P}\left(T_i > t \mid X_{i,0}\right). \tag{1}$$

Note that $S(t \mid X_{i,0})$ is a monotonically decreasing function of t with $S(0 \mid X_{i,0}) = 1$ and $\lim_{t\to\infty} S(t \mid X_{i,0}) = 0$. Next, we introduce a conditionally independent censoring assumption. Note, without this assumption, the survival probabilities are provably not identifiable from observed data (Tsiatis, 1975).

Assumption 2.1 (Conditionally Independent Censoring)**.** The event time T_i and censoring time C_i are conditionally independent given $X_{i,0}$, i.e., $T_i \perp\!\!\!\perp C_i \mid X_{i,0}$.

2.2 Dynamic setting

In the dynamic setting, the baseline covariates $X_{i,0}$ is replaced by a sequence of observations revealed over time that could be observed at irregular intervals. We denote this sequence of covariates as $\mathcal{X}_i \triangleq \{X_{i,t} : t \in \mathcal{T}_i\}$, where $\mathcal{T}_i \subset [0, \infty)$ represents the observation times. The history of covariates up to time t is defined as

$$\mathcal{H}_{i,t} \triangleq \big\{(s, X_{i,s}) : s \in \mathcal{T}_i, s \leq t\big\}. \tag{2}$$

The dataset has n i.i.d. tuples $\{(\mathcal{X}_i, T_i, C_i, \delta_i)\}_{i=1}^n$, where the definitions of T_i, C_i, and δ_i follow Section 2.1.

The modeling task is to estimate the conditional survival probability for a future horizon $\Delta > 0$, given the information available at time t:

$$S(t + \Delta \mid T_i > t, \mathcal{H}_{i,t}) \triangleq \mathbb{P}\left(T_i > t + \Delta \mid T_i > t, \mathcal{H}_{i,t}\right). \tag{3}$$

We use a dynamic analogue of the conditionally independent censoring condition from the static setting (Assumption 2.1), which ensures the dynamic survival probabilities (3) are identifiable from observed data.

Assumption 2.2 (Dynamic Conditionally Independent Censoring)**.** For any $t \geq 0$, the event time T_i and censoring time C_i are conditionally independent given the history and survival up to time t, i.e., $T_i \perp\!\!\!\perp C_i \mid (\mathcal{H}_{i,t}, T_i > t)$.

3 Related work

3.1 Overview of static and dynamic survival modeling approaches

The Cox Proportional Hazards (CoxPH) model (Cox, 1972) and Random Survival Forests (Ishwaran et al., 2008) are the most widely utilized approaches in applied static survival analysis (Wang et al., 2019). Deep learning-based static survival models largely fall into two categories: (1) *Deep extensions of (semi-)parametric survival models* (Wiegrebe et al., 2024; Kvamme et al., 2019; Nagpal et al., 2021), e.g., DeepSurv (Katzman et al., 2018), which replaces the linear risk function in CoxPH with a neural network, and (2) *Discrete-time models* that partition the time horizon into intervals before modeling event probabilities (Lee et al., 2018; Gensheimer & Narasimhan, 2019; Chi et al., 2021); see Section 3.2 for classification-based approaches.

The predominant dynamic survival modeling approaches fall into three categories: (1) *Landmarking*, which fits a static survival model at multiple timepoints (Van Houwelingen, 2007). (2) *Deep sequence or Joint modeling*, which jointly models longitudinal observations and event times (Tsiatis & Davidian, 2004; Ibrahim et al., 2010); this includes classical statistical approaches that use linear mixed-effects models and CoxPH, as well as deep-learning variants that use recurrent neural networks with survival objectives (Giunchiglia et al., 2018; Ren et al., 2019). Neural ODEs have also been used to explicitly handle irregularly sampled data (Tang et al., 2022; Moon et al., 2022). (3) *Discrete-time models*, that extend the static discrete-time models to longitudinal data (Lee et al., 2019; Kvamme & Borgan, 2021; Thorsen-Meyer et al., 2022).

3.2 Classification models for survival analysis

Multiple papers have formulated survival analysis as a classification problem. One of the first approaches was developed by Efron (1988), who used a logistic regression model to approximate the discrete hazard rate $\lambda(t_k \mid X_{i,0}) \triangleq \mathbb{P}\big(T_i \in (t_k, t_{k+1}] \mid X_{i,0}, T_i > t_k\big)$, i.e., the probability that an event occurs in the kth interval given that it has not occurred by the start of the interval. The logistic regression model is fit by maximizing the likelihood of the observed data, which factors into a function of the hazard rate; note, the likelihood of the observed data includes both censored and uncensored examples. Singer & Willett (1993) extended this to log-odds models. This general paradigm of using binary classifiers to model the hazard rate has been termed *Survival Stacking* (Craig et al., 2025; Zhong & Tibshirani, 2019). The survival probability is recovered as the product of a function of the hazard rate: $S(t_k \mid X_{i,0}) = \prod_{j=1}^{k-1} \big\{1 - \lambda(t_j \mid X_{i,0})\big\}$.

In our experiments, we evaluate survival stacking with off-the-shelf tabular foundation models and find that the discrete-hazard classification approach is competitive for small numbers of discretization bins K, but

often degrades as K increases (see Section 5.2 and Figure 2 for details). We hypothesize this sensitivity is due to the product-based construction of survival probabilities, under which small errors in the estimated per-bin hazards can accumulate over time. This is structurally similar to the error accumulation observed in iterated multi-step forecasting, where later predictions depend on a sequence of earlier estimated quantities (Marcellino et al., 2006). These observations motivate our formulation, which avoids multiplicative computing errors, by using TFMs to directly estimate cumulative failure probabilities at discretized time points.

Our formulation is also related to Yu et al. (2011), which similarly represents survival prediction using K binary outcomes over discretized time points with the same label structure that we present in Section 4.1. However, Yu et al. (2011) propose multi-task logistic regression with dependencies across time points, trained using a conditional-random-field-regularized objective optimized via expectation maximization. In contrast, our formulation uses a simple binary cross-entropy loss that can be applied directly to off-the-shelf tabular foundation models through in-context learning. The empirical scope also differs: Yu et al. (2011) evaluate on three large static datasets, whereas we study both static and dynamic survival analysis across a broader collection of real-world datasets. Subsequent work has extended the Yu et al. (2011) framework to neural survival models trained from scratch (Fotso, 2018) and to models that learn a multi-task logistic regression model on top of representations from a pretrained tabular foundation models (Pham et al., 2026).

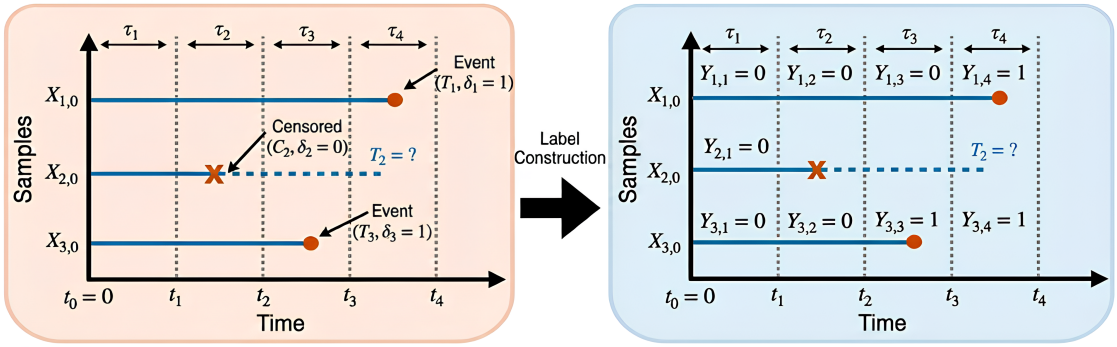
Beyond binary classification models, a a variety of approaches have been proposed to learn survival probabilities by training a neural network to predict which of K time intervals an event occurs in, including DeepHit, Dynamic DeepHit, and TabSurv (Lee et al., 2018; 2019; Kirpichenko et al., 2026). For the DeepHit models, when the event is censored, the models are trained to maximize the probability that the event occurred in any of the bins following the censoring time. We compare to both DeepHit and Dynamic DeepHit in our simulations but find they significantly underperform our TFM-based approach. We hypothesize this is because DeepHit and Dynamic DeepHit require training recurrent neural networks from scratch, which is less data efficient; these models were originally evaluated on larger datasets (800-10,000 samples), while the datasets we evaluate on are much smaller (median size of ≈ 500).

3.3 Concurrent work on in-context learning models for survival analysis

After the initial release of our manuscript in January 2026, two related and independent efforts were released in May 2026: Survival In-Context (SIC) (Seletkov et al., 2026) and SurvivalPFN (Qi et al., 2026). Both develop *survival-specific* tabular foundation models by pretraining transformers from scratch on large-scale synthetic survival tasks. SIC uses the DeepHit training loss (Lee et al., 2018), while SurvivalPFN uses a cross-entropy objective and provides a censoring event indicator separately as an input to the model.[1] These models are complementary to our work: they incorporate survival structure directly into the pretraining procedure, whereas we ask whether existing general-purpose TFMs can be used for survival analysis without survival-specific pretraining, fine-tuning, or architectural changes.

This distinction leads to several practical differences. Because survival-specific foundation models are trained with a fixed survival output representation, they are tied to the discretization scheme used during pretraining; changing the number or placement of time bins requires modifying or retraining the model. Specifically, SIC uses $K = 10$ bins and SurvivalPFN uses $K = 1024$ bins. In contrast, our method varies the number of time bins K by reprocessing the survival data while keeping the underlying TFM fixed. Our formulation also extends naturally to *dynamic* survival analysis with time-varying covariates by re-querying the classifier at each landmark time, whereas SIC and SurvivalPFN, focus on static survival prediction. Moreover, SurvivalPFN uses of a censoring event indicator as an input to the model appears less directly compatible with dynamic landmarking, where censoring status depends on the landmark time and prediction horizon. Our contribution is distinct in showing that survival-analysis capabilities can be elicited from off-the-shelf TFMs through a simple binary classification reformulation. Note that we were not able to directly empirically compare to these three pre-trained models as they are currently not publicly available.

[1] In our experiments we assessed a multi-class classification formulation akin to that of SurvivalPFN with off-the-shelf TFMs but found that they underperform our cumulative failure formulation we propose; for details see Remark 4.1 and Table 1.



Original Data	Classification Data			
$(X_{1,0}, T_1, C_1, \delta_1 = 1)$	$((X_{1,0}, t_1), Y_{1,1} = 0, C_1)$	$((X_{1,0}, t_2), Y_{1,2} = 0, C_1)$	$((X_{1,0}, t_3), Y_{1,3} = 0, C_1)$	$((X_{1,0}, t_4), Y_{1,4} = 1, C_1)$
$(X_{2,0}, T_2, C_2, \delta_2 = 0)$	$((X_{2,0}, t_1), Y_{2,1} = 0, C_2)$			
$(X_{3,0}, T_3, C_3, \delta_3 = 1)$	$((X_{3,0}, t_1), Y_{3,1} = 0, C_3)$	$((X_{3,0}, t_2), Y_{3,2} = 0, C_3)$	$((X_{3,0}, t_3), Y_{3,3} = 1, C_3)$	$((X_{3,0}, t_4), Y_{3,4} = 1, C_3)$

Figure 1: **Transforming of survival data into a binary classification dataset (static setting).** The left panel illustrates the original time-to-event data $(X_{i,0}, T_i, C_i, \delta_i)$. For second example, the event $(T_2 =?)$ is censored at C_2. The right panel displays the discretization of time into 5 intervals $\{\tau_k\}_{k=1}^{5}$, where binary labels $Y_{i,k} = \mathbb{1}(T_i \leq t_k)$ are defined at each timestep t_k for $k = 1, 2, 3, 4$. The table (bottom) shows the resulting classification dataset: each example is transformed into at most 4 tuples, where labels are only constructed for time steps preceding the censoring time $(t_k < C_i)$.

4 Methodology: Time-to-event modeling as supervised learning

We frame survival modeling under right-censoring as a series of binary classification tasks. By reducing survival modeling to a standard classification task, we can leverage the tabular foundation models and enable survival modeling via in-context learning without additional survival-specific training or hyperparameter tuning. Our approach involves transforming each time-to-event data tuple into a sequence of binary labels indicating whether the event has occurred by the end of each respective time interval. We first describe our temporal discretization approach, as it underpins the classification formulations for both the static (Section 4.1) and dynamic (Section 4.2) settings.

Temporal discretization. To transform the continuous survival task into a tractable classification problem, we partition the time horizon into K discrete intervals $\{\tau_k\}_{k=1}^{K}$. Given the boundaries $0 = t_0 < t_1 < t_2 < \cdots < t_{K-1}$, we define the intervals as $\tau_k = (t_{k-1}, t_k]$ for $k < K$, with a final open-ended interval $\tau_K = (t_{K-1}, \infty)$. For convenience, let $t_K \triangleq \infty$. In practice, we determine the boundaries $\{t_k\}_{k=1}^{K-1}$ using the quantiles of the observed event times in the training set to ensure a balanced distribution of events across intervals. This time discretization approach is well-established in the survival analysis literature (Efron, 1988; Craig et al., 2025; Allison, 1982; Singer & Willett, 1993; Lee et al., 2019; Yu et al., 2011; Maystre & Russo, 2022). An alternative censoring-aware binning that uses quantiles of the Kaplan–Meier estimate of the marginal survival function (Kvamme & Borgan, 2021) yields nearly identical performance on average; see Appendix D.5.

4.1 Static setting

Constructing the supervised learning dataset. To transform survival modeling into a supervised learning problem, we define binary labels $\{Y_{i,k}\}_{k=1}^{K-1}$ where

$$Y_{i,k} \triangleq \mathbb{1}(T_i \leq t_k). \tag{4}$$

Due to right-censoring, $Y_{i,k}$ is only observed when $t_k < C_i$. Note that by construction $T_i \in \tau_k$ for some interval τ_k where $k \in [1\colon K-1]$, since $T_i < \infty$. As depicted in Figure 1, the original time-to-event data tuple $(X_{i,0}, T_i, C_i, \delta_i)$ is expanded into $K-1$ data tuples for binary classification: $\big\{\big((X_{i,0}, t_k), Y_{i,k}, C_i\big)\big\}_{k=1}^{K-1}$. The censoring time C_i is used to derive whether the label $Y_{i,k}$ is observed or missing for the time interval τ_k.

Model training. We train a classifier $\hat{p}$ to minimize a empirical binary cross-entropy loss; below, the indicator $\mathbb{1}(t_k < C_i)$ restricts the objective to observed labels $Y_{i,k}$:

$$\hat{\ell}_{\text{static}}(\hat{p}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \sum_{k=1}^{K-1} \mathbb{1}(t_k < C_i) \cdot \text{BCE}\big(\hat{p}(X_{i,0}, t_k), Y_{i,k}\big) \tag{5}$$

where $\text{BCE}\big(p, Y\big) \triangleq -\{Y \log p + (1-Y)\log(1-p)\}$. By only evaluating the loss only for pairs (i, t_k) where $t_k < C_i$, we restrict training strictly to observed labels $Y_{i,k}$. While simply ignoring unobserved data might appear to be a heuristic handling of censoring, we formally show in Section 4.3 that minimizing this loss recovers the true underlying survival probabilities as $n \to \infty$.

The cross-entropy loss from (5) can be minimized using standard classification models, such as XGBoost or multi-layer perceptrons. Furthermore, this framework allows us to "fit" $\hat{p}$ via in-context learning using tabular foundation models for classification (Hollmann et al., 2023; Zhang et al., 2025), where $\hat{p}$ is computed by conditioning on the set of tuples $\big((X_{i,0}, t_k), Y_{i,k}\big)$ for which $t_k < C_i$.

Inference. Given a binary classifier $\hat{p}$, we form our estimate of the survival probability $S(t_k \mid X_{i,0})$ as follows:

$$\hat{S}(t_k \mid X_{i,0}) \triangleq 1 - \hat{p}(X_{i,0}, t_k). \tag{6}$$

To ensure the survival probabilities are monotonically decreasing in t_k, by clipping $1 - \hat{p}(X_{i,0}, t_k)$ if it is larger than $1 - \hat{p}(X_{i,0}, t_{k-1})$. See Appendix C.1.1 for details.

Remark 4.1 (Multi-Class Classification Formulations). Besides the cumulative-failure prediction formulation we present here and the discrete hazard formulation discussed in Section 3.2, one could also consider multi-class classification formulations. For example, the survival foundation model SurvivalPFN (Qi et al., 2026) is pre-trained to directly predict $\min(T_i, C_i)$ given the baseline covariates and censoring event indicator, $(X_{i,0}, \delta_i)$; at inference time the survival probability is obtained by inputting $(X, \delta = 1)$ into the model. In Section 5.2, we empirically compare XGBoost classifiers and off-the-shelf TFMs using this multi-class classification formulation and find that it significantly underperforms our cumulative-failure prediction formulation. We hypothesize that the cumulative-failure prediction formulation may be easier to learn from finite data because it decomposes survival prediction into a sequence of lower-complexity binary prediction problems: whether the event has occurred by a given time horizon. This avoids requiring the model to distinguish among many possible event-time bins, while still recovering survival probabilities.

4.2 Dynamic setting

We adopt the same temporal discretization and binary label definition, $Y_{i,k} \triangleq \mathbb{1}(T_i \leq t_k)$, as in the static setting. In the dynamic setting, however, the model must account for time-varying information. At a given timestep t_k, the model is provided with the covariate history $\mathcal{H}_{i,t_k} \triangleq \big\{(s, X_{i,s}) : s \in \mathcal{T}_i, s \leq t_k\big\}$ and the knowledge that the subject has survived up to t_k (i.e., $Y_{i,1} = \cdots = Y_{i,k} = 0$). Consequently, each original time-to-event data tuple $(\mathcal{X}_i, T_i, C_i, \delta_i)$ is transformed into training examples: $\big((\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}), Y_{i,k+\Delta}, C_i\big)$, for each $k \in \{0, \ldots, K-2\}$ and $\Delta \in \{1, \ldots, K-k-1\}$, provided $t_k < C_i$. We use $Y_{i,1:k} \triangleq (Y_{i,1}, \ldots, Y_{i,k})$. Same as in the static setting, C_i is used to derive whether the $Y_{i,k+\Delta}$ is observed or missing.

Model training. We train a binary classifier $\hat{p}$ to minimize the dynamic empirical binary cross-entropy loss:

$$\hat{\ell}_{\text{dynamic}}(\hat{p}) \triangleq \frac{1}{n} \sum_{i=1}^{n} \sum_{k=0}^{K-2} \sum_{\Delta=1}^{K-k-1} \mathbb{1}(t_k < T_i)\mathbb{1}(t_{k+\Delta} < C_i) \cdot \text{BCE}\big(\hat{p}(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}), Y_{i,k+\Delta}\big). \tag{7}$$

We include the indicator $\mathbb{1}(t_k < T_i)$ to ensure that we only train on examples for which the event is at risk, i.e., if the event occurred before time t_k, we are not using $\mathcal{H}_{i,t_k}$ to do further prediction. Similar to the static setting, the indicator $\mathbb{1}(t_{k+\Delta} < C_i)$ to account for censoring effectively truncates the label sequence $Y_{i,1:K}$; our formulation allows us to utilize all information prior to that truncation time.

Practically, for training these classifiers across different history lengths ($\mathcal{H}_{i,t_k}$ for different values of k), we featurize the inputs into a fixed dimension d vector $\phi_k(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}) \in \mathbb{R}^d$ for a vector-valued

functions ϕ_k. For our classification models, we use vectors $\phi(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta})$ as input and $Y_{i,t_{k+\Delta}}$ as the label. See Appendix C.2.1 for details on how we choose ϕ_k.

Inference. Given a classifier $\hat{p}$ we estimate the dynamic survival probability $S(t_{k+\Delta} \mid T_i > t_k, \mathcal{H}_{i,t_k})$ using:

$$\hat{S}(t_{k+\Delta} \mid T_i > t_k, \mathcal{H}_{i,t_k}) \triangleq 1 - \hat{p}(\mathcal{H}_{i,k}, Y_{i,1:k}, t_k, t_{k+\Delta}). \tag{8}$$

We clip the survival probabilities to ensure that they are monotonically decreasing in $t_{k+\Delta}$ (Appendix C.2.1).

4.3 Theory: Consistent survival models via loss minimization

We now show formally that minimizing the losses from (5) and (7) for the static and dynamic settings, respectively, yields consistent survival models as the number of training samples increases ($n \to \infty$). This means that despite the presence of censoring, perfect binary classifiers necessarily recover the true underlying survival probabilities. The result follows from the fact that binary cross-entropy is minimized by the true conditional event probabilities, together with conditionally independent censoring (Assumptions 2.1 and 2.2), and establishes that the proposed classification objectives remain statistically aligned with the underlying survival targets despite censoring.

Static setting. For infinite training data ($n \to \infty$), the empirical loss from (5) equals the population loss

$$\ell_{\text{static}}(p) \triangleq \mathbb{E}\Big[\sum_{k=1}^{K} \mathbb{1}(C_i > t_k) \cdot \text{BCE}\big(p(X_{i,0}, t_k), Y_{i,k}\big)\Big]. \tag{9}$$

In Theorem 4.2 below, we show that a binary classifier p minimizes the loss $\ell(p)$ if and only if it accurately models the ground-truth survival probabilities.

Theorem 4.2 (Consistent static survival model via loss minimization)**.** *Let p be any binary prediction model for $Y_{i,k}$ given $(X_{i,0}, t_k)$. Under Assumption 2.1 (Conditionally Independent Censoring), the population loss $\ell_{\text{static}}(p)$ is minimized if and only if, for each $k \in \{1, \ldots, K-1\}$,*

$$p(x, t_k) = \mathbb{P}(Y_{i,k} = 1 \mid X_{i,0} = x) = 1 - \mathbb{P}(T_i > t_k \mid X_{i,0} = x)$$

for all x in the support of $X_{i,0}$ satisfying the positivity condition $\mathbb{P}(C_i \geq t_k \mid X_{i,0} = x) > 0$.

The equality $\mathbb{P}(Y_{i,k} = 1 \mid X_{i,0}) = 1 - \mathbb{P}(T_i > t_k \mid X_{i,0})$ above holds by the definition $Y_{i,k} \triangleq \mathbb{1}(T_i \leq t_k)$. See Appendix A.1 for the proof of Theorem 4.2.

The significance of Theorem 4.2 is that under censoring Assumption 2.1, we can in principle recover the true survival probabilities $\mathbb{P}(T_i > t_k \mid X_{i,0})$ via loss minimization, given sufficient data. *This shows that survival analysis under temporal discretization can be reduced to a supervised, binary classification problem in a theoretically grounded manner; our empirical contribution is to demonstrate that this reduction unlocks practical use of modern tabular foundation models for survival analysis.*

Dynamic setting. Analogously to the static setting, in the dynamic setting, we can show that for infinite training data ($n \to \infty$), minimizing the loss from (7) will necessarily lead the classifier to learn the true underlying survival probabilities. This result is very similar to Theorem 4.2; see Appendix A.2.

5 Experimental results

Datasets. We use datasets from *SurvSet*, a collection of real-world, right-censored time-to-event datasets (Drysdale, 2022). These datasets span multiple domains, for example, clinical studies in biomedical research, labor datasets related to unemployment duration, as well as engineering reliability datasets concerning time to system or component failure. Out of the total 69 static and 8 dynamic datasets in *SurvSet*, we utilize 43 static and 5 dynamic datasets. We excluded datasets that (i) contained fewer than two unique event times in the test set, or (ii) had a Kaplan-Meier estimate of the censoring probability $\mathbb{P}(C_i > t_k \mid X_{i,0}) = 0$ or $\mathbb{P}(C_i > t_{k+\Delta} \mid T_i > t_k, \mathcal{H}_{i,t_k}) = 0$ for any t_k. This second criterion ensures a positive probability of not being

censored; without it, the survival distribution is non-identifiable in regions lacking support, and the inverse weighting used in standard evaluation metrics leads to unstable, biased estimates (Uno et al., 2011).

Across the 43 static datasets, the median sample size is 461 (IQR: 228-1494). The dimension of covariates $X_{i,0}$ ranged from 4 to 161 (median: 8, mean: 15.2). The censoring rate in these datasets is heterogeneous ranging from from 6.6% to 94.4% (median: 56.4%). The dynamic datasets vary in scale, from the $4,603$ samples in *oldmort* to smaller ones with 103 to 556 samples. Across these 5 datasets, the number of covariates ranges from 4 to 7, and the censoring rate range is 27.2%-66.2%. See Appendix D.1 for more details on the datasets.

Models. In both the static and dynamic settings, consider methods based on two tabular foundation models MITRA (Zhang et al., 2025) and TabPFN-2.5 (Grinsztajn et al., 2025),[2] as well as an XGBoost model. For our approach, which models the cumulative failure probability, we abbreviate using "CF". We also compare to classification approaches that model the discrete-time hazard rate (Efron, 1988; Allison, 1982; Kvamme & Borgan, 2021; Craig et al., 2025), as discussed in Section 3.2; we abbreviate these methods using "DH". Finally, we use "MC" to abbreviate the multi-class classification approach inspired by SurvivalPFN (Qi et al., 2026); see Remark 4.1 for additional details. In the **static setting**, we compare to classical survival models CoxPH (Cox, 1972) and Random Survival Forest (RSF) (Ishwaran et al., 2008), as well as neural survival methods DeepSurv (Katzman et al., 2018), a neural-network extension of the CoxPH model, and DeepHit (Lee et al., 2018). In the **dynamic setting**, we compare to landmarked versions of the CoxPH and RSF models (Van Houwelingen, 2007). We also compare to two popular dynamic survival models: Dynamic DeepHit (Lee et al., 2019), and Joint Modeling (Ibrahim et al., 2010), which simultaneously models the longitudinal trajectory of time-varying covariates with a linear mixed-effects model and the time-to-event process with a CoxPH model.

5.1 Evaluation metrics

We evaluate the survival models using standard metrics for discrimination and calibration. Specifically, we assess the ability to rank-order survival times using the (time-dependent) *C-index* and the *Integrated AUC*. To quantify calibration and overall predictive accuracy, we use the *Integrated Brier Score*. **We now discuss these metrics in the static setting; we discuss dynamic analogues, specific estimators, and implementation details in Appendix B.** To ensure consistency under conditionally independent censoring (Assumption 2.1), we use Inverse-Probability-of-Censoring Weighted (IPCW) estimators for all three metrics.

Concordance index (C-Index). The C-Index quantifies a survival model's discriminative ability by calculating the probability that, for a randomly selected pair of observations, the model assigns a higher risk to the observation that experiences the event first (Harrell et al., 1982; Uno et al., 2011). A higher C-index indicates greater discriminative ability. In the static setting, this is given by

$$C_{\text{Index}} \triangleq \mathbb{P}\big(\hat{r}(X_{i,0}) > \hat{r}(X_{j,0}) \mid T_i < T_j\big), \tag{10}$$

where $\hat{r}$ is a risk scoring model (higher values imply earlier predicted events). For our classification-based survival models, we construct the risk score as follows:

$$\hat{r}(x) = \frac{1}{K-1} \sum_{k=1}^{K-1} \Big\{1 - \hat{S}(t_k \mid x)\Big\}. \tag{11}$$

Above, the risk score $\hat{r}(x)$ can be interpreted as the average probability that an event has occurred across all time steps $t_1, \ldots, t_{K-1}$. This aggregate measure ensures that observations with consistently lower survival curves are assigned higher risk, even if their survival at a single specific time point is similar. For our baseline models, we discuss how we construct risk scores in Appendix C.

Integrated AUC. The time-dependent AUC (Hung & Chiang, 2010)assesses a model's ability to differentiate between examples that survive up to some time t versus those that do not:

$$\text{AUC}(t) \triangleq \mathbb{P}\big(\hat{r}(X_{i,0}) > \hat{r}(X_{j,0}) \mid T_i \leq t, T_j > t\big), \tag{12}$$

[2]We also assessed TabPFN-3 (Grinsztajn et al., 2026), but found it underperformed TabPFN-2.5; see Appendix D.6.

where $\hat{r}$ is a risk scoring model. See (11) for the risk score we use for our classification-based survival models. The Integrated AUC takes a weighted average of $\text{AUC}(t)$ over time. Specifically, in the static setting we use $\overline{\text{AUC}} \triangleq \frac{\sum_{k=1}^{K-1} \text{AUC}(t_k) w_{t_k}}{\sum_{k=1}^{K-1} w_{t_k}}$, where $w_{t_k} \triangleq S(t_{k-1}) - S(t_k)$ for $S(t_k) \triangleq \mathbb{P}(T_i > t_k)$ is the marginal survival probability (Antolini et al., 2005; Lambert & Chevret, 2016). This weights the AUC by the proportion of events occurring in each interval.

Integrated Brier Score (IBS). The Brier Score is a mean-squared error of the estimated survival probability at time t and the true event indicator at time t (Graf et al., 1999; Kvamme & Borgan, 2023):

$$\text{BS}(t) \triangleq \mathbb{E}\left[\left\{\mathbb{1}(T_i > t) - \hat{S}(t \mid X_{i,0})\right\}^2\right]. \tag{13}$$

Low Brier Scores imply the model's survival probabilities are accurate and well-calibrated (in contrast to C-index and Integrated AUC, which only assess accurate ranking). The Integrated Brier Score averages the $\text{BS}(t)$ over time: $\overline{\text{BS}} \triangleq \frac{1}{t_{K-1} - t_1} \int_{t_1}^{t_{K-1}} \text{BS}(t)\, dt$.

Table 1: **Average performance comparison in the static setting.** Our classification-based survival models, especially MITRA-CF and XGBoost-CF, achieve the strongest overall performance across the 43 benchmark datasets. Results are averaged over four time-discretization granularities ($K \in \{5, 10, 15, 20\}$) using quantiles of observed event times. Values are reported as mean ± standard error. The best and second-best performers are highlighted in dark blue and light blue, respectively.

Model	C-Index			Integrated AUC			Integrated Brier Score			Avg. Rank ↓	Avg. ELO ↑
	Value ↑	Rank ↓	ELO ↑	Value ↑	Rank ↓	ELO ↑	Value ↓	Rank ↓	ELO ↑		
Classification-based Models											
MITRA-CF	$0.674_{\pm 0.017}$	$\mathbf{5.2}_{\pm 0.4}$	**1047**	$\mathbf{0.729}_{\pm 0.016}$	$\mathbf{5.0}_{\pm 0.4}$	**1051**	$\mathbf{0.164}_{\pm 0.007}$	$\mathbf{3.9}_{\pm 0.3}$	**1077**	$\mathbf{4.7}_{\pm 0.3}$	**1058**
TabPFN-CF	$0.635_{\pm 0.020}$	$8.5_{\pm 0.5}$	964	$0.691_{\pm 0.017}$	$8.5_{\pm 0.5}$	963	$0.180_{\pm 0.009}$	$7.0_{\pm 0.5}$	1000	$8.0_{\pm 0.4}$	976
XGBoost-CF	$\mathbf{0.688}_{\pm 0.017}$	$\mathbf{5.2}_{\pm 0.5}$	1046	$0.726_{\pm 0.017}$	$5.3_{\pm 0.5}$	1044	$0.166_{\pm 0.008}$	$4.0_{\pm 0.4}$	1076	$4.9_{\pm 0.4}$	1055
Multiclass Variants											
MITRA-MC	$0.626_{\pm 0.014}$	$9.0_{\pm 0.5}$	950	$0.685_{\pm 0.016}$	$8.2_{\pm 0.6}$	971	$0.289_{\pm 0.014}$	$10.9_{\pm 0.2}$	903	$9.4_{\pm 0.4}$	941
TabPFN-MC	$0.638_{\pm 0.018}$	$7.6_{\pm 0.6}$	985	$0.694_{\pm 0.019}$	$7.2_{\pm 0.6}$	996	$0.322_{\pm 0.019}$	$11.6_{\pm 0.3}$	883	$8.8_{\pm 0.4}$	955
XGBoost-MC	$0.634_{\pm 0.016}$	$8.9_{\pm 0.5}$	951	$0.679_{\pm 0.018}$	$8.6_{\pm 0.5}$	956	$0.305_{\pm 0.017}$	$11.2_{\pm 0.3}$	894	$9.6_{\pm 0.4}$	934
Discrete-time Hazard Variants											
MITRA-DH	$0.677_{\pm 0.018}$	$\mathbf{5.2}_{\pm 0.4}$	1044	$0.728_{\pm 0.016}$	$5.2_{\pm 0.4}$	1047	$0.171_{\pm 0.007}$	$5.4_{\pm 0.3}$	1039	$5.3_{\pm 0.3}$	1043
TabPFN-DH	$0.657_{\pm 0.020}$	$5.3_{\pm 0.6}$	1044	$0.723_{\pm 0.015}$	$\mathbf{5.0}_{\pm 0.5}$	**1051**	$0.191_{\pm 0.012}$	$6.2_{\pm 0.6}$	1018	$5.5_{\pm 0.5}$	1038
XGBoost-DH	$0.687_{\pm 0.018}$	$5.5_{\pm 0.5}$	1036	$0.721_{\pm 0.017}$	$5.9_{\pm 0.5}$	1026	$0.167_{\pm 0.008}$	$4.3_{\pm 0.4}$	1065	$5.2_{\pm 0.4}$	1042
Classical & Neural Survival Baselines											
CoxPH	$0.682_{\pm 0.017}$	$\mathbf{5.2}_{\pm 0.6}$	1043	$0.721_{\pm 0.016}$	$5.6_{\pm 0.6}$	1032	$0.169_{\pm 0.008}$	$4.9_{\pm 0.5}$	1054	$5.2_{\pm 0.5}$	1043
RSF	$0.666_{\pm 0.021}$	$6.7_{\pm 0.5}$	1007	$0.717_{\pm 0.017}$	$7.0_{\pm 0.5}$	1000	$0.167_{\pm 0.006}$	$5.2_{\pm 0.4}$	1046	$6.3_{\pm 0.4}$	1018
DeepHit	$0.585_{\pm 0.015}$	$11.0_{\pm 0.4}$	900	$0.612_{\pm 0.017}$	$11.2_{\pm 0.3}$	894	$0.211_{\pm 0.008}$	$9.4_{\pm 0.4}$	943	$10.5_{\pm 0.3}$	912
DeepSurv	$0.650_{\pm 0.017}$	$7.7_{\pm 0.5}$	984	$0.685_{\pm 0.017}$	$8.3_{\pm 0.5}$	969	$0.179_{\pm 0.008}$	$7.0_{\pm 0.4}$	1001	$7.7_{\pm 0.4}$	985

5.2 Static results

Aggregate performance trends (Table 1). Our evaluation across 43 datasets demonstrates that classification-based survival models consistently outperform existing baselines. The strong performance of XGBoost-CF highlights the benefits of our classification formulation, while MITRA-CF having the best overall performance demonstrates the advantage of leveraging prior knowledge in TFMs. The discrete-hazard (DH) and multi-class (MC) variants of the same classifiers underperform the direct variants on every aggregate metric, illustrating that the choice of formulation matters. Our Table 1 results are averaged over four different time-discretization granularities $K \in \{5, 10, 15, 20\}$. Overall, we find that our approach is relatively robust to the choice of K; see Appendix D.4 and Figure 2 for additional details. We also notice that CoxPH outperforms both DeepHit and DeepSurv, likely because deep survival models trained from scratch struggle in the small-sample regimes. These results suggest that reformulating survival as classification allows general-purpose foundation models to leverage their inductive biases more effectively.

Heterogeneity in gains across datasets (Figure 2). While the aggregate gains in Table 1 are modest in median, the per-dataset distribution is heterogeneous. For each of the 43 static datasets d, we compute the relative gain of MITRA-CF over the best non-TFM baseline: $G_d \triangleq \frac{m_d^{\text{TFM}} - m_d^{\text{best-baseline}}}{m_d^{\text{best-baseline}}}$, where m is the metric

of interest (C-index, Integrated AUC, or IBS, with signs adjusted so that larger G_d corresponds to greater improvement for the TFM). *We find that the gains in the top* 30% *of datasets shows a* 5–9% *improvement and the top* 10% *show* $>$ 20% *improvement over the best non-TFM baseline.*

Furthermore, our results in Figure 2 also illustrate the benefits of the cumulative failure modeling approach over the discrete hazard modeling approach, especially as the number of discretization intervals K increases. Notice especially for the median dataset (subfigure a), that the discrete hazard (DH) modeling approach has gains over the baseline methods that are monotonically decreasing as K increases. In contrast, the

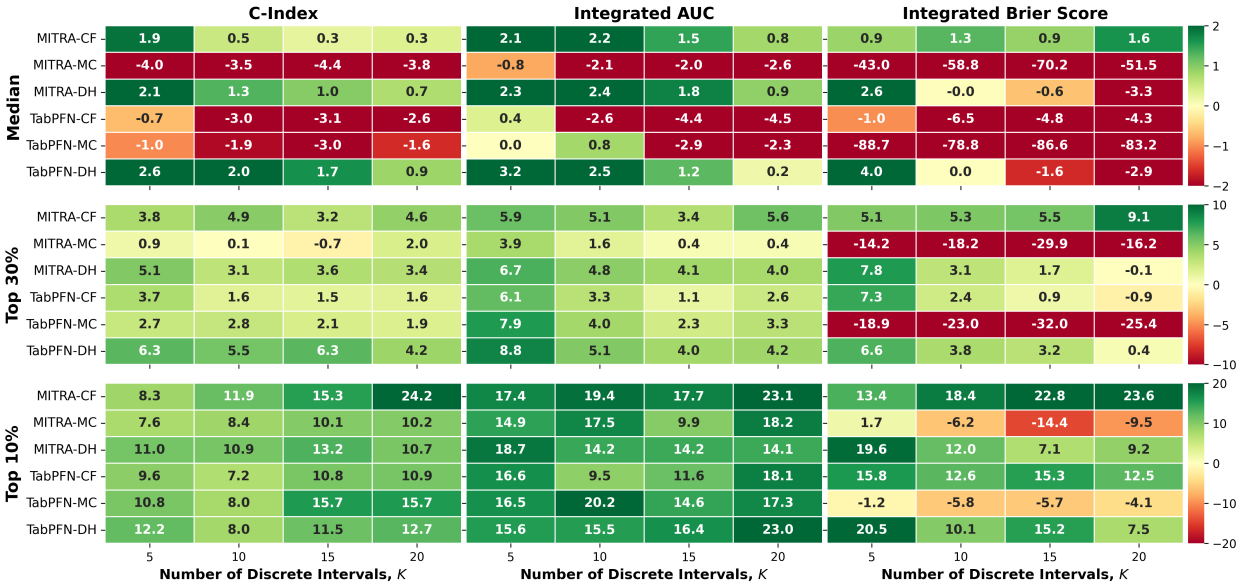


Figure 2: **Relative gain G_d of MITRA-CF over the best non-TFM baseline across** 43 **static datasets.** The median gain (top) is modest; however, the top 30% of datasets (middle) show 5–9% improvement, and the top 10% (bottom) show $>$ 20% improvement. The best non-TFM baseline is chosen per dataset by validation performance (c-index) among {CoxPH, RSF, DeepHit, DeepSurv, XGBoost, XGBoost-DH, XGBoost-MC}.

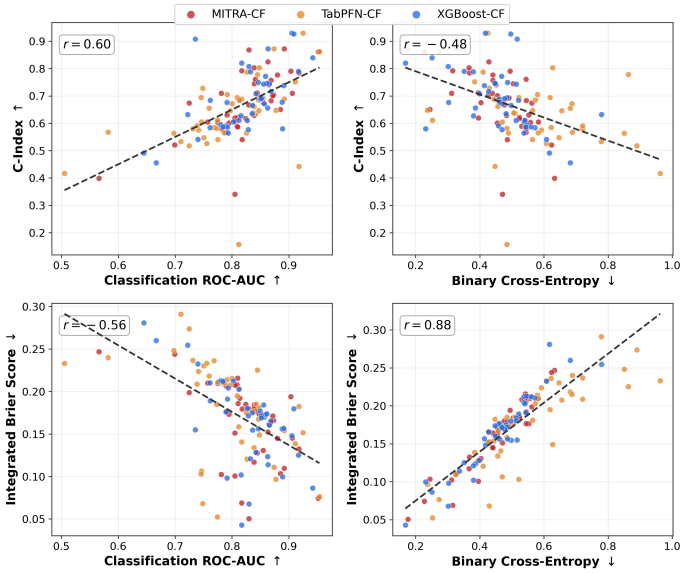


Figure 3: **Correlation between classification and survival metrics.** Each point represents a classification model evaluated on 1 of the 43 datasets (test split, averaged over $K \in \{5, 10, 15, 20\}$). We see a strong correlation ($r = 0.89$) between binary cross entropy loss and integrated Brier score. This empirically supports Theorem 4.2: a model that minimizes the binary cross-entropy loss recovers accurate survival probabilities.

cumulative failure (CF) approaches (especially with MITRA) have performance that has relatively stable or even increasing gains over baseline methods as K increases.

Correlation between classification and survival metrics (Figure 3). A natural question is whether improvements in the binary classification objective from Section 4 translate into improvements in standard survival metrics. To investigate this, we examine the relationship between *classification metrics*—binary cross-entropy loss and ROC-AUC—and *survival metrics*—C-Index and IBS—across the 43 static datasets. We find only a moderate correlation between the classification metrics and C-Index, reflecting the fact that our binary objective does not explicitly optimize for ranking. In contrast, binary cross-entropy loss is strongly correlated with Integrated Brier Score ($r = 0.89$), supporting the theoretical result that minimizing the proposed binary cross-entropy loss targets accurate survival probability estimates (Theorem 4.2).

Practically, these results suggest that model selection can be performed directly in the classification space: among candidate classifiers, including TFMs and non-TFM models, choose the one with the lowest validation binary cross-entropy loss. This provides a simple, survival-aware selection criterion and empirically supports the reduction of survival probability modeling to binary classification formalized in Theorem 4.2.

5.3 Dynamic results

Aggregate performance trends (Table 2). In the dynamic setting, the tabular foundation model MITRA-CF is the best overall under both $K = 5$ and $K = 8$; in particular, it is the best on Integrated AUC and IBS, and achieves the highest average C-Index. While the Joint Model has historically been a competitive method for dynamic modeling, MITRA-CF achieves the best raw values in all metrics. Additionally, MITRA-CF's superior IBS suggests that it captures complex temporal patterns more effectively than Joint Models, which

Table 2: **Average performance comparison in the dynamic setting.** Results are aggregated across 5 dynamic datasets at two discretization granularities ($K = 5$ and $K = 8$). Within each K, results average over prediction times t_1, t_2, t_3. MITRA-CF achieves the best overall performance under both granularities, leading in average rank and ELO; the Joint Model is the strongest non-TFM baseline. *Values* represent means and standard errors; the best and second-best performers are highlighted in dark blue and light blue.

Model	C-Index			Integrated AUC			Integrated Brier Score			Avg. Rank ↓	Avg. ELO ↑
	Value ↑	Rank ↓	ELO ↑	Value ↑	Rank ↓	ELO ↑	Value ↓	Rank ↓	ELO ↑		
$K = 5$											
Classification-based Models											
MITRA-CF	$\mathbf{0.650}_{\pm 0.043}$	$5.0_{\pm 1.4}$	1011	$\mathbf{0.693}_{\pm 0.022}$	$3.2_{\pm 0.5}$	1058	$\mathbf{0.099}_{\pm 0.018}$	$\mathbf{2.2}_{\pm 0.6}$	**1088**	**3.5**	**1052**
TabPFN-CF	$0.567_{\pm 0.082}$	$6.0_{\pm 1.7}$	982	$0.632_{\pm 0.048}$	$6.4_{\pm 1.7}$	971	$0.121_{\pm 0.020}$	$6.8_{\pm 0.7}$	961	6.4	971
XGBoost-CF	$0.532_{\pm 0.047}$	$6.6_{\pm 0.8}$	972	$0.600_{\pm 0.042}$	$6.4_{\pm 1.6}$	976	$0.123_{\pm 0.023}$	$6.6_{\pm 0.9}$	968	6.5	972
Discrete-time Hazard Variants											
MITRA-DH	$0.631_{\pm 0.033}$	$\mathbf{4.4}_{\pm 0.7}$	**1033**	$0.689_{\pm 0.022}$	$\mathbf{2.8}_{\pm 0.9}$	**1070**	$0.107_{\pm 0.023}$	$3.8_{\pm 1.0}$	1048	3.7	1050
TabPFN-DH	$0.549_{\pm 0.059}$	$5.2_{\pm 1.2}$	1010	$0.620_{\pm 0.041}$	$5.8_{\pm 1.1}$	993	$0.110_{\pm 0.018}$	$4.6_{\pm 0.9}$	1021	5.2	1008
XGBoost-DH	$0.514_{\pm 0.057}$	$6.8_{\pm 1.4}$	962	$0.594_{\pm 0.038}$	$6.8_{\pm 1.0}$	964	$0.138_{\pm 0.029}$	$7.4_{\pm 1.4}$	947	7.0	958
Classical & Neural Survival Baselines											
Landmark Cox	$0.615_{\pm 0.027}$	$5.2_{\pm 1.8}$	1008	$0.611_{\pm 0.077}$	$5.2_{\pm 1.7}$	1009	$0.130_{\pm 0.031}$	$5.0_{\pm 1.6}$	1019	5.1	1012
Landmark RSF	$0.603_{\pm 0.032}$	$5.4_{\pm 1.7}$	1004	$0.605_{\pm 0.066}$	$5.8_{\pm 1.0}$	993	$0.104_{\pm 0.020}$	$4.2_{\pm 0.6}$	1037	5.1	1011
Dynamic DeepHit	$0.590_{\pm 0.032}$	$6.0_{\pm 1.0}$	987	$0.544_{\pm 0.053}$	$7.8_{\pm 0.6}$	941	$0.219_{\pm 0.026}$	$9.8_{\pm 0.2}$	884	7.9	937
Joint Model	$0.647_{\pm 0.051}$	$\mathbf{4.4}_{\pm 1.5}$	1030	$0.642_{\pm 0.041}$	$4.8_{\pm 1.5}$	1023	$0.111_{\pm 0.021}$	$4.6_{\pm 1.3}$	1027	4.6	1027
$K = 8$											
Classification-based Models											
MITRA-CF	$\mathbf{0.676}_{\pm 0.045}$	$\mathbf{4.4}_{\pm 1.5}$	**1026**	$\mathbf{0.636}_{\pm 0.048}$	$\mathbf{3.8}_{\pm 1.2}$	**1043**	$0.102_{\pm 0.021}$	$3.6_{\pm 0.5}$	1048	**3.9**	**1039**
TabPFN-CF	$0.589_{\pm 0.075}$	$5.0_{\pm 1.3}$	1013	$0.575_{\pm 0.047}$	$5.2_{\pm 1.7}$	1009	$0.137_{\pm 0.025}$	$7.2_{\pm 1.4}$	949	5.8	990
XGBoost-CF	$0.592_{\pm 0.070}$	$5.4_{\pm 1.0}$	1001	$0.598_{\pm 0.042}$	$4.6_{\pm 0.9}$	1026	$0.114_{\pm 0.020}$	$5.8_{\pm 0.7}$	993	5.3	1007
Discrete-time Hazard Variants											
MITRA-DH	$0.630_{\pm 0.041}$	$5.0_{\pm 1.3}$	1011	$0.582_{\pm 0.092}$	$4.6_{\pm 1.2}$	1023	$0.117_{\pm 0.036}$	$3.4_{\pm 1.7}$	1054	4.3	1029
TabPFN-DH	$0.578_{\pm 0.079}$	$5.0_{\pm 1.9}$	1014	$0.583_{\pm 0.030}$	$6.4_{\pm 1.7}$	974	$0.142_{\pm 0.026}$	$8.0_{\pm 0.7}$	929	6.5	972
XGBoost-DH	$0.535_{\pm 0.037}$	$7.0_{\pm 1.1}$	958	$0.546_{\pm 0.066}$	$7.6_{\pm 0.9}$	941	$0.130_{\pm 0.031}$	$7.0_{\pm 0.5}$	960	7.2	953
Classical & Neural Survival Baselines											
Landmark Cox	$0.644_{\pm 0.022}$	$5.4_{\pm 1.3}$	1007	$0.575_{\pm 0.071}$	$5.6_{\pm 1.1}$	997	$0.113_{\pm 0.016}$	$4.6_{\pm 1.5}$	1029	5.2	1011
Landmark RSF	$0.625_{\pm 0.015}$	$4.8_{\pm 1.2}$	1021	$0.582_{\pm 0.075}$	$5.4_{\pm 1.1}$	1001	$\mathbf{0.094}_{\pm 0.017}$	$\mathbf{2.2}_{\pm 0.6}$	**1090**	4.1	1037
Dynamic DeepHit	$0.528_{\pm 0.075}$	$8.2_{\pm 0.9}$	930	$0.504_{\pm 0.092}$	$7.2_{\pm 1.7}$	957	$0.165_{\pm 0.028}$	$8.6_{\pm 0.4}$	919	8.0	935
Joint Model	$0.668_{\pm 0.053}$	$4.8_{\pm 1.5}$	1019	$0.594_{\pm 0.043}$	$4.6_{\pm 1.3}$	1028	$0.102_{\pm 0.015}$	$4.6_{\pm 1.1}$	1029	4.7	1025

model the relationship between longitudinal inputs and risk using a specific pre-specified functional form. As in the static setting, the DH variants of the TFMs underperform their direct counterparts, consistent with our K-sensitivity analysis: dynamic survival reconstruction also accumulates multiplicative errors over the prediction horizon.

The performance gap between MITRA-CF and XGBoost-CF widens in the dynamic setting. While XGBoost-CF is competitive in the static setting, its significant decline in average rank compared to MITRA-CF suggests that the temporal prior knowledge embedded in the TFM is particularly well-suited to modeling the complexities of longitudinal data. These results suggest our classification-based TFM is an effective framework for dynamic survival analysis.

Model performance over time (Figure 4). The timepoint-wise trends in Figure 4 show that MITRA-CF's performance lead is most pronounced at t_2, across all three metrics. In C-Index, MITRA-CF has the best performance at t_1 and t_2, but it is surpassed by the Joint Model and Landmark CoxPH at t_3. MITRA-CF consistently has superior performance in Integrated AUC over time, and is better than or highly competitive in IBS, particularly when the other neural model, Dynamic DeepHit, struggles.

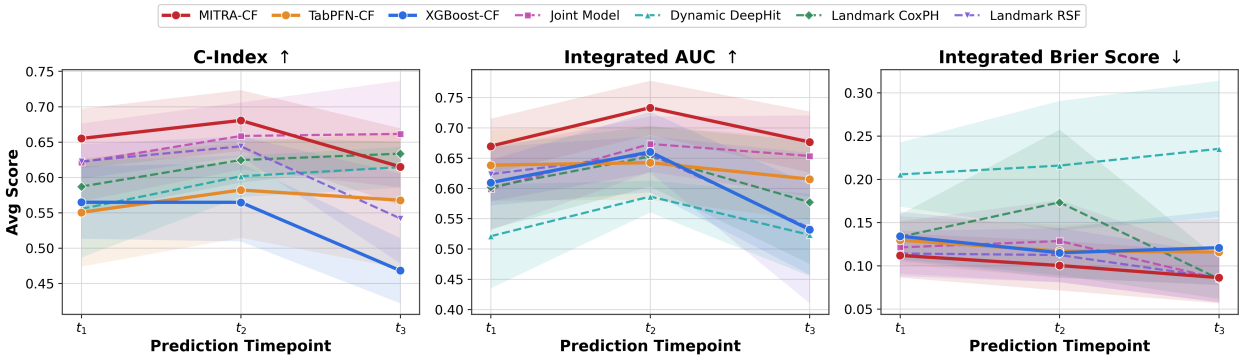


Figure 4: **Performance over time in the dynamic setting.** Timepoint-wise performance across the five longitudinal datasets at timepoints t_1, t_2, t_3 ($K = 5$ discretization bins). MITRA-CF consistently outperforms both traditional landmarking baselines, joint modeling, and dynamic neural network-based survival models across most timepoints for all metrics. Notably, MITRA-CF achieves a great lead over all other models on all three metrics at t_2. Baseline models are represented by dashed lines.

6 Discussion

In this work, we demonstrate the utility of a classification-based formulation of survival analysis, specifically as it allows us to leverage pretrained TFMs. Beyond its strong modeling performance, this approach is also computationally efficient. Using a single NVIDIA Tesla A100 GPU (80GB), our static TFM experiments (over 43 datasets and 4 time discretization granularities K for both MITRA-CF and TabPFN-CF) took ≈ 30 minutes. Similarly, the dynamic TFM experiments across 5 datasets took ≈ 5 minutes.

Limitations. Casting continuous event times into K discrete bins introduces an unavoidable loss of temporal resolution, particularly when event times are concentrated or highly non-uniformly distributed. Increasing K mitigates this resolution loss but raises computational cost. Additionally, our missing-label treatment of censored observations is theoretically justified only under conditionally independent censoring (Assumptions 2.1 and 2.2). Informative censoring (where censoring time depends on the latent event time given covariates) and competing risks (where multiple event types can occur) fall outside the regime covered by Theorem 4.2. A natural future direction is to investigate approaches to handle dependent censoring.

Future work. Directions for future work include (1) exploring fine-tuning TFMs explicitly for survival analysis, e.g., to explicitly optimize for ranking or C-Index; (2) investigating ways to leverage TFMs for continuous-time survival modeling rather than relying on discretization; (3) extending our approach to handle informative censoring; and (4) leveraging TFMs for treatment effect analysis in time-to-event outcomes, a common use case for survival analysis (Hosmer et al., 2008).

References

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

Paul D Allison. Discrete-time methods for the analysis of event histories. *Sociological methodology*, 13:61–98, 1982.

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in medicine*, 24(24):3927–3944, 2005.

Norman Breslow. Covariance analysis of censored survival data. *Biometrics*, pp. 89–99, 1974.

George H Chen et al. An introduction to deep survival analysis models for predicting time-to-event outcomes. *Foundations and Trends® in Machine Learning*, 17(6):921–1100, 2024.

Shengqiang Chi, Yu Tian, Feng Wang, Yu Wang, Ming Chen, and Jingsong Li. Deep semisupervised multitask learning model and its interpretability for survival analysis. *IEEE journal of biomedical and health informatics*, 25(8):3185–3196, 2021.

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

Erin Craig, Chenyang Zhong, and Robert Tibshirani. A review of survival stacking: a method to cast survival regression analysis as a classification problem. *The International Journal of Biostatistics*, 2025.

Erik Drysdale. Survset: An open-source time-to-event dataset repository. *arXiv preprint arXiv:2203.03094*, 2022.

Bradley Efron. Logistic regression, survival analysis, and the kaplan-meier curve. *Journal of the American statistical Association*, 83(402):414–425, 1988.

Stephane Fotso. Deep neural networks for survival analysis based on a multi-task framework. *arXiv preprint arXiv:1801.05512*, 2018.

Josh Gardner, Juan C Perdomo, and Ludwig Schmidt. Large scale transfer learning for tabular data via language modeling. *Advances in Neural Information Processing Systems*, 37:45155–45205, 2024.

Michael F Gensheimer and Balasubramanian Narasimhan. A scalable discrete-time survival model for neural networks. *PeerJ*, 7:e6257, 2019.

Eleonora Giunchiglia, Anton Nemchenko, and Mihaela van der Schaar. Rnn-surv: A deep recurrent model for survival analysis. In *International conference on artificial neural networks*, pp. 23–32. Springer, 2018.

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.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, et al. Tabpfn-2.5: Advancing the state of the art in tabular foundation models. *arXiv preprint arXiv:2511.08667*, 2025.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Mihir Manium, Shi Bin Hoo, Magnus Bühler, Anurag Garg, et al. Tabpfn-3: Technical report. *arXiv preprint arXiv:2605.13986*, 2026.

Frank E Harrell, Robert M Califf, David B Pryor, Kerry L Lee, and Robert A Rosati. Evaluating the yield of medical tests. *Jama*, 247(18):2543–2546, 1982.

Noah Hollmann, Samuel Müller, Katharina Eggensperger, and Frank Hutter. TabPFN: A transformer that solves small tabular classification problems in a second. In *International Conference on Learning Representations (ICLR)*, 2023.

David W Hosmer, Stanley Lemeshow, and Susanne May. Applied survival analysis. *Wiley Series in Probability and Statistics*, pp. 60, 2008.

Hung Hung and Chin-Tsang Chiang. Estimation methods for time-dependent auc models with survival data. *Canadian Journal of Statistics*, 38(1):8–26, 2010.

Joseph G Ibrahim, Haitao Chu, and Liddy M Chen. Basic concepts and methods for joint models of longitudinal and survival data. *Journal of clinical oncology*, 28(16):2796–2801, 2010.

Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. *Annals of Applied Statistics*, 2008.

John D Kalbfleisch and Ross L Prentice. *The statistical analysis of failure time data*. John Wiley & Sons, 2002.

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

Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC medical research methodology*, 18(1):24, 2018.

Stanislav Kirpichenko, Andrei Konstantinov, and Lev Utkin. TabSurv: Adapting modern tabular neural networks to survival analysis. *arXiv preprint arXiv:2605.03944*, 2026.

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

Håvard Kvamme and Ørnulf Borgan. Continuous and discrete-time survival prediction with neural networks. *Lifetime data analysis*, 27(4):710–736, 2021.

Håvard Kvamme and Ørnulf Borgan. The brier score under administrative censoring: Problems and a solution. *Journal of Machine Learning Research*, 24(2):1–26, 2023.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of machine learning research*, 20(129):1–30, 2019.

Jérôme Lambert and Sylvie Chevret. Summary measure of discrimination in survival models based on cumulative/dynamic time-dependent roc curves. *Statistical methods in medical research*, 25(5):2088–2102, 2016.

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

Changhee Lee, Jinsung Yoon, and Mihaela Van Der Schaar. 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, 2019.

Roderick JA Little and Donald B Rubin. *Statistical analysis with missing data*. John Wiley & Sons, 2019.

Massimiliano Marcellino, James H. Stock, and Mark W. Watson. A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series. *Journal of Econometrics*, 135(1–2): 499–526, 2006.

Lucas Maystre and Daniel Russo. Temporally-consistent survival analysis. *Advances in Neural Information Processing Systems*, 35:10671–10683, 2022.

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*, pp. 800–827. PMLR, 2022.

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

Wayne Nelson. Theory and applications of hazard plotting for censored failure data. *Technometrics*, 14(4): 945–966, 1972.

Minh-Khoi Pham, Luca Cotugno, Alina Sîrbu, Tai Tan Mai, Martin Crane, and Marija Bezbradica. Tabular foundation models for clinical survival analysis via survival-aware adaptation. *arXiv preprint arXiv:2606.12006*, 2026. Accepted at AIiH 2026.

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, Vahid Balazadeh, Michael Cooper, Russell Greiner, and Rahul G. Krishnan. SurvivalPFN: Amortizing survival prediction via in-context Bayesian inference. *arXiv preprint arXiv:2605.15488*, 2026.

Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu, and Yong Yu. Deep recurrent survival analysis. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4798–4805, 2019.

Dmitrii Seletkov, Paul Hager, Georgios Kaissis, Rickmer Braren, Daniel Rueckert, and Raphael Rehms. Survival in-context: Amortized Bayesian survival analysis via prior-fitted networks. *arXiv preprint arXiv:2603.29475*, 2026.

Judith D Singer and John B Willett. It's about time: Using discrete-time survival analysis to study duration and the timing of events. *Journal of educational statistics*, 18(2):155–195, 1993.

Ethan Steinberg, Jason Alan Fries, Yizhe Xu, and Nigam Shah. MOTOR: A time-to-event foundation model for structured medical records. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=NialiwI2V6`.

Weijing Tang, Jiaqi Ma, Qiaozhu Mei, and Ji Zhu. Soden: a scalable continuous-time survival model through ordinary differential equation networks. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Hans-Christian Thorsen-Meyer, Davide Placido, Benjamin Skov Kaas-Hansen, Anna P Nielsen, Theis Lange, Annelaura B Nielsen, Palle Toft, Jens Schierbeck, Thomas Strøm, Piotr J Chmura, et al. Discrete-time survival analysis in the critically ill: a deep learning approach using heterogeneous data. *NPJ digital medicine*, 5(1):142, 2022.

Anastasios Tsiatis. A nonidentifiability aspect of the problem of competing risks. *Proceedings of the National Academy of Sciences*, 72(1):20–22, 1975.

Anastasios A Tsiatis and Marie Davidian. Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica*, pp. 809–834, 2004.

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

Hans Van Houwelingen and Hein Putter. *Dynamic prediction in clinical survival analysis*. CRC Press, 2011.

Hans C Van Houwelingen. Dynamic prediction by landmarking in event history analysis. *Scandinavian Journal of Statistics*, 34(1):70–85, 2007.

Tuyen Vu, Ha X Tran, Xiaomei Li, Lin Liu, Jiuyong Li, Jia Tina Du, and Thuc D Le. Tabular foundation model for breast cancer prognosis using gene expression data. *medRxiv*, pp. 2025–10, 2025.

Ping Wang, Yan Li, and Chandan K Reddy. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)*, 51(6):1–36, 2019.

Simon Wiegrebe, Philipp Kopper, Raphael Sonabend, Bernd Bischl, and Andreas Bender. Deep learning for survival analysis: a review. *Artificial Intelligence Review*, 57(3):65, 2024.

Chun-Nam Yu, Russell Greiner, Hsiu-Chin Lin, and Vickie Baracos. Learning patient-specific cancer survival distributions as a sequence of dependent regressors. *Advances in neural information processing systems*, 24, 2011.

Xiyuan Zhang, Danielle C Maddix, Junming Yin, Nick Erickson, Abdul Fatir Ansari, Boran Han, Shuai Zhang, Leman Akoglu, Christos Faloutsos, Michael W Mahoney, et al. Mitra: Mixed synthetic priors for enhancing tabular foundation models. *39th Conference on Neural Information Processing Systems (NeurIPS 2025)*, 2025.

Chenyang Zhong and Robert Tibshirani. Survival analysis as a classification problem. *arXiv preprint arXiv:1909.11171*, 2019.

Table of Contents for Appendices

A Theoretical Results **17**

B Metrics **19**

C Modeling Details **22**

D Additional Experimental Details and Results **27**

A Theoretical Results

A.1 Consistent static survival model

Theorem 4.2 (Consistent static survival model via loss minimization). *Let p be any binary prediction model for $Y_{i,k}$ given $(X_{i,0}, t_k)$. Under Assumption 2.1 (Conditionally Independent Censoring), the population loss $\ell_{\text{static}}(p)$ is minimized if and only if, for each $k \in \{1, \ldots, K-1\}$,*

$$p(x, t_k) = \mathbb{P}(Y_{i,k} = 1 \mid X_{i,0} = x) = \mathbb{P}(T_i \leq t_k \mid X_{i,0} = x)$$

for all x in the support of $X_{i,0}$ satisfying the positivity condition $\mathbb{P}(C_i \geq t_k \mid X_{i,0} = x) > 0$. Note, by convention, we define $0 \log 0 \triangleq 0$ and $x \log 0 = -\infty$ for $x > 0$.

Proof. By the definition of $\ell_{\text{static}}(p)$ from (9),

$$\begin{aligned}
\ell_{\text{static}}(p) &= -\mathbb{E}\Bigg[\sum_{k=1}^{K-1} \mathbb{1}(C_i > t_k) \cdot \text{BCE}\big(p(X_{i,0}, t_k), Y_{i,k}\big)\Bigg] \underbrace{=}_{(a)} -\sum_{k=1}^{K-1} \mathbb{E}\left[\mathbb{1}(C_i > t_k) \cdot \text{BCE}\big(p(X_{i,0}, t_k), Y_{i,k}\big)\right] \\
&\underbrace{=}_{(b)} -\sum_{k=1}^{K-1} \mathbb{E}\Bigg[\mathbb{E}\left[\mathbb{1}(C_i > t_k) \cdot \text{BCE}\big(p(X_{i,0}, t_k), Y_{i,k}\big) \mid X_{i,0}\right]\Bigg] \\
&\underbrace{=}_{(c)} -\sum_{k=1}^{K-1} \mathbb{E}\Bigg[\mathbb{P}(C_i > t_k \mid X_{i,0}) \cdot \mathbb{E}\left[\text{BCE}\big(p(X_{i,0}, t_k), Y_{i,k}\big) \mid X_{i,0}\right]\Bigg] \\
&\underbrace{=}_{(d)} -\sum_{k=1}^{K-1} \mathbb{E}\Bigg[\mathbb{P}(C_i > t_k \mid X_{i,0}) \cdot \mathbb{E}\left[Y_{i,k} \log p(X_{i,0}, t_k) + (1 - Y_{i,k}) \log\big(1 - p(X_{i,0}, t_k)\big) \mid X_{i,0}\right]\Bigg] \\
&\underbrace{=}_{(e)} -\sum_{k=1}^{K-1} \mathbb{E}\Bigg[\mathbb{P}(C_i > t_k \mid X_{i,0})\Big\{\mathbb{P}(Y_{i,k} = 1 \mid X_{i,0}) \log p(X_{i,0}, t_k) + \big\{1 - \mathbb{P}(Y_{i,k} = 1 \mid X_{i,0})\big\} \log\big(1 - p(X_{i,0}, t_k)\big)\Big\}\Bigg]
\end{aligned} \tag{14}$$

Above, equality (a) holds by linearity of expectations. Equality (b) holds by the law of iterated expectations. Equality (c) holds because $C_i \perp\!\!\!\perp Y_{i,k} \mid X_{i,0}$ since $Y_{i,k} \triangleq \mathbb{1}_{T_{i,k} \leq t_k}$ and $C_i \perp\!\!\!\perp T_i \mid X_{i,0}$ by Assumption 2.1. Equality (d) holds by the definition of BCE from Section 4.1. Equality (e) holds because since $Y_{i,k}$ is binary, $\mathbb{E}[Y_{i,k} \mid X_{i,0}] = \mathbb{P}(Y_{i,k} = 1 \mid X_{i,0})$.

We will now use the result from (14) to prove the if and only if statement of the theorem. Note that by Lemma A.1,

$$\mathbb{P}(Y_{i,k} = 1 \mid X_{i,0} = x) = \text{argmin}_{b \in (0,1)} \; -\mathbb{P}(Y_{i,k} = 1 \mid X_{i,0} = x) \log b - \big\{1 - \mathbb{P}(Y_{i,k} = 1 \mid X_{i,0} = x)\big\} \log\big(1 - b\big)$$

and the minimizer is unique. Thus, p is a minimizer of the loss $\ell_{\text{static}}(p)$ if and only if $p(x, t_k) = \mathbb{P}(Y_{i,k} = 1 \mid X_{i,0} = x)$ for all x in the support of $X_{i,0}$ for which $\mathbb{P}(C_i > t_k \mid X_{i,0} = x) > 0$. □

Lemma A.1. *Fix $q \in [0,1]$ and let $b \in [0,1]$. The function $f(b) = -q \log b - (1-q)\log(1-b)$ is uniquely minimized at $b = q$.*

Proof. In the edge case that $q = 0$, $f(b)$ is minimized at $b = 0$. Also, when $q = 1$, $f(b)$ is minimized at $b = 1$. Now suppose $q \in (0,1)$. Note that $f'(b) = -\frac{q}{b} + \frac{1-q}{1-b}$. When $f'(b) = 0$, we get that $b = q$. This must be the global minimizer of $f(b)$ because the second derivative of $f(b)$ is strictly positive: $f''(b) = \frac{q}{b^2} + \frac{1-q}{(1-b)^2} > 0$. □

A.2 Consistent dynamic survival model

Analogously to the static setting, in the dynamic setting, we can show that as the amount of training samples $n \to \infty$, minimizing the loss from (7) will necessarily lead the classifier to learn the true underlying survival probabilities. Below, we define $\ell_{\text{dynamic}}(p)$, which is the population version of the loss from (7).

$$\ell_{\text{dynamic}}(p) \triangleq \mathbb{E}\Bigg[\sum_{k=0}^{K-2}\sum_{\Delta=1}^{K-k-1} \mathbb{1}(t_k < T_i)\cdot \mathbb{1}(t_{k+\Delta} < C_i)\cdot \text{BCE}\big(p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_{k+\Delta}), Y_{i,k+\Delta}\big)\Bigg]. \tag{15}$$

Theorem A.2 formalizes how minimizing the loss $\ell_{\text{dynamic}}(p)$ is equivalent to learning the underlying survival probabilities, despite censoring, given that the dynamic censoring Assumption 2.2 holds.

Theorem A.2 (Consistent dynamic survival model via loss minimization)**.** *Let p be any binary prediction model for $Y_{i,k+\Delta}$ given $(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta})$). Under Assumption 2.2 (Dynamic Conditionally Independent Censoring), the population loss $\ell_{\text{dynamic}}(p)$ is minimized if and only if, for each $k \in \{1,\dots,K-2\}$ and $\Delta \in \{1,\dots,K-k-1\}$,*

$$p(h, y, t_k, t_{k+\Delta}) = \mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y) = 1 - \mathbb{P}(T_i > t_{k+\Delta} \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y)$$

for all $h, y_{1:k}$ in the support of $\mathcal{H}_{i,t_k}, Y_{i,1:k}$ that (i) satisfy the positivity condition $\mathbb{P}(C_i \geq t_{k+\Delta} \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y) > 0$ and (ii) have not had the event occur by t_k, i.e., $y_1 = \cdots = y_k = 0$. Note, by convention, we define $0 \log 0 \triangleq 0$ and $x \log 0 = -\infty$ for $x > 0$.

Proof. By the definition of $\ell_{\text{dynamic}}(p)$ from (15),

$$\begin{aligned}
\ell_{\text{dynamic}}(p) &= -\mathbb{E}\Bigg[\sum_{k=0}^{K-2}\sum_{\Delta=1}^{K-k-1} \mathbb{1}(T_i > t_k)\cdot \mathbb{1}(C_i > t_{k+\Delta})\cdot \text{BCE}\big(p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}), Y_{i,k+\Delta}\big)\Bigg] \\
&\underbrace{=}_{(a)} -\sum_{k=0}^{K-2}\sum_{\Delta=1}^{K-k-1} \mathbb{E}\left[\mathbb{1}(T_i > t_k)\cdot \mathbb{1}(C_i > t_{k+\Delta})\cdot \text{BCE}\big(p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}), Y_{i,k+\Delta}\big)\right] \\
&\underbrace{=}_{(b)} -\sum_{k=0}^{K-2}\sum_{\Delta=1}^{K-k-1} \mathbb{E}\Bigg[\mathbb{1}(T_i > t_k)\cdot \mathbb{E}\left[\mathbb{1}(C_i > t_{k+\Delta})\cdot \text{BCE}\big(p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}), Y_{i,k+\Delta}\big) \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}\right]\Bigg] \\
&\underbrace{=}_{(c)} -\sum_{k=0}^{K-2}\sum_{\Delta=1}^{K-k-1} \mathbb{E}\Bigg[\mathbb{1}(T_i > t_k)\cdot \mathbb{P}\left(C_i > t_{k+\Delta} \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}\right)\cdot \mathbb{E}\left[\text{BCE}\big(p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}), Y_{i,k+\Delta}\big) \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}\right]\Bigg] \\
&\underbrace{=}_{(d)} -\sum_{k=0}^{K-2}\sum_{\Delta=1}^{K-k-1} \mathbb{E}\Bigg[\mathbb{1}(T_i > t_k)\cdot \mathbb{P}\left(C_i > t_{k+\Delta} \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}\right) \\
&\qquad \times \mathbb{E}\big[Y_{i,k+\Delta}\log p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}) + (1 - Y_{i,k+\Delta})\log\big(1 - p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta})\big) \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}\big]\Bigg] \\
&\underbrace{=}_{(e)} -\sum_{k=0}^{K-2}\sum_{\Delta=1}^{K-k-1} \mathbb{E}\Bigg[\mathbb{1}(T_i > t_k)\cdot \Bigg\{\mathbb{P}\left(C_i > t_{k+\Delta} \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}\right)\cdot \mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k}, Y_{i,1:k})\log p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta}) \\
&\qquad + \big\{1 - \mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k}, Y_{i,1:k})\big\}\log\big(1 - p(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta})\big) \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}\Bigg]\Bigg\}\Bigg]
\end{aligned} \tag{16}$$

Above, equality (a) holds by linearity of expectations. Equality (b) holds by the law of iterated expectations and since $\mathbb{1}(T_i > t_k)$ is known given $\mathcal{H}_{i,t_k}, Y_{i,1:k}$. Equality (c) holds because $C_i \perp\!\!\!\perp Y_{i,k+\Delta} \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}$ since $Y_{i,k+\Delta} \triangleq \mathbb{1}_{T_{i,k+\Delta} \leq t_k}$ and $T_i \perp\!\!\!\perp C_i \mid (\mathcal{H}_{i,t}, T_i > t)$ by Assumption 2.2. Equality (d) holds because since $Y_{i,k+\Delta}$ is binary, $\mathbb{E}[Y_{i,k+\Delta} \mid \mathcal{H}_{i,t_k}, Y_{i,1:k}] = \mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k}, Y_{i,1:k})$.

We will now use the result from (16) to prove the if and only if statement of the theorem. Note that by Lemma A.1,

$$\mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y) = \operatorname{argmin}_{b\in[0,1]} \big\{ \mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y) \log b \\ + \big\{ 1 - \mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y) \big\} \log \big(1 - b\big) \big\},$$

and the minimizer is unique. Thus, p is a minimizer of the loss $\ell_{\text{dynamic}}(p)$ if and only if $p(\mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y, t_k, t_{k+\Delta}) = \mathbb{P}(Y_{i,k+\Delta} = 1 \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y)$ if and only if for all h, y in the support of $\mathcal{H}_{i,t_k}, Y_{i,1:k}$ for which $\mathbb{P}(C_i > t_k \mid \mathcal{H}_{i,t_k} = h, Y_{i,1:k} = y) > 0$ and for which the event has not occurred by t_k, i.e., $y_1 = \cdots = y_k = 0$. □

B Metrics

This appendix provides formal definitions, corresponding estimators and implementation details for the evaluation metrics used in the *static setting* and *dynamic setting*. All survival metrics are implemented using the `sksurv` library (Pölsterl, 2020). We also briefly explain how we calculate the ELO rating in Section B.3. See our codebase for additional details.

B.1 Static Metrics

B.1.1 Concordance Index (C-Index)

The concordance index (C-Index) is a measure of a model's ability to accurately rank individuals by their risk of experiencing an event (Harrell et al., 1982; Uno et al., 2011). We say a pair of samples is concordant if the individual with a shorter time-to-event has a higher risk score assigned by a model.

Recall from display (10), the C-Index measures the following in the static setting

$$C_{\text{Index}} \triangleq \mathbb{P}(\hat{r}(X_{i,0}) > \hat{r}(X_{j,0}) \mid T_i < T_j)\,. \tag{17}$$

where $\hat{r}$ is a risk scoring model. A common approach to correct for censoring is by using Inverse-Probability-of-Censoring-Weights (IPCW), which provides an unbiased and consistent estimator, under certain assumptions, including Assumption 2.1 (Uno et al., 2011). We first consider a truncated version of display (17):

$$C_{\text{Index}} \triangleq \mathbb{P}\left(\hat{r}(X_{i,0}) > \hat{r}(X_{j,0}) \,|\, T_i < T_j,\ T_i \leq T_{\max}\right), \tag{18}$$

where we set $T_{\max}$ to be the maximum observed timepoint in the train set. Truncation is used to ensure that the predicted probability of not being censored remains positive, which is required when using IPCW weighting. Following Uno et al. (2011), an IPCW estimator of display (18) is

$$\hat{C}_{\text{Index}} = \frac{\sum_{i\neq j} \hat{G}(T_i)^{-2} \delta_i \mathbb{1}\big(T_i < T_j, T_i \leq T_{\max}\big) \mathbb{1}\big(\hat{r}(X_{i,0}) > \hat{r}(X_{j,0})\big)}{\sum_{i\neq j} \delta_i \hat{G}(T_i)^{-2} \mathbb{1}\big(T_i < T_j, T_i \leq T_{\max}\big)},$$

where $G(t)$ denotes the population censoring function at time t. Formally,

$$G(t) \triangleq \mathbb{P}(C_i > t), \tag{19}$$

and $\hat{G}(t)$ is estimated using the Kaplan-Meier estimator (Kaplan & Meier, 1958; Pölsterl, 2020).

B.1.2 Integrated AUC

The AUC (area under the ROC curve) at time t is a measure of how well a model can distinguish individuals who experience an event by time t from those who do not. The weighted mean of these AUCs tells us about the overall discriminative ability of a survival model.

Recall from display (12), the time-dependent AUC measures the following in the static setting

$$\mathrm{AUC}(t) \triangleq \mathbb{P}(\hat{r}(X_{i,0}) > \hat{r}(X_{j,0}) \,|\, T_i \leq t,\ T_j > t)\,. \tag{20}$$

A consistent IPCW estimate (Hung & Chiang, 2010) of display (20) is given by

$$\widehat{\mathrm{AUC}}(t) = \frac{\sum_{i \neq j} \hat{G}(T_i)^{-1}\, \delta_i\, \mathbb{1}\big(T_i \leq t,\ T_j > t\big)\, \mathbb{1}\big(\hat{r}(X_{i,0}) > \hat{r}(X_{j,0})\big)}{\sum_{i \neq j} \hat{G}(T_i)^{-1}\, \delta_i\, \mathbb{1}\big(T_i \leq t,\ T_j > t\big)}, \tag{21}$$

where $G(t)$ is the censoring function at time t; see display (19). We evaluate display (21) at t_k for $k \in \{1, \ldots, K-1\}$, and compute a weighted average of the resulting AUCs, using the estimated survival function as weights

$$\overline{\mathrm{AUC}} = \frac{\sum_{k'=1}^{K-1} \widehat{\mathrm{AUC}}(t_{k'})\, w_{t_{k'}}}{\sum_{k'=1}^{K-1} w_{t_{k'}}},$$

where $w_{t_k} = \hat{S}(t_{k-1}) - \hat{S}(t_k)$, and $\hat{S}(t)$ is the Kaplan-Meier estimator of the marginal survival function: $S(t) \triangleq \mathbb{P}(T_i > t)$ (Kaplan & Meier, 1958).

B.1.3 Integrated Brier Score (IBS)

The Integrated Brier Score quantifies a model's calibration and accuracy by averaging over time, the squared error between the predicted survival probabilities and the observed survival outcomes. Recall from display (13), the time-dependent Brier score at time t is defined as the mean squared error between the binary outcome $\mathbb{1}(T_i > t)$ and the predicted survival function

$$\mathrm{BS}(t) \triangleq \mathbb{E}\Big[\big(\mathbb{1}(T_i > t) - \hat{S}(t \mid X_{i,0})\big)^2\Big]. \tag{22}$$

A standard IPCW estimator of display (22) from Kvamme & Borgan (2023) is

$$\widehat{\mathrm{BS}}(t) = \frac{1}{n} \sum_{i=1}^{n} \left[\frac{\mathbb{1}(T_i \leq t,\, \delta_i = 1)}{\hat{G}(T_i)} \big(0 - \hat{S}(t \mid X_{i,0})\big)^2 + \frac{\mathbb{1}(T_i > t)}{\hat{G}(t)} \big(1 - \hat{S}(t \mid X_{i,0})\big)^2 \right], \tag{23}$$

where $G(t)$ is the censoring function at time t, defined in display (19). We estimate the Integrated Brier Score by evaluating display (23) at time points t_k, $k \in \{1, \ldots, K-1\}$, and then integrating these values over time

$$\overline{\mathrm{BS}} = \frac{1}{t_{K-1} - t_1} \int_{t_1}^{t_{K-1}} \widehat{\mathrm{BS}}(t)\, dt.$$

We use the trapezoidal rule to approximate the integral.

B.2 Dynamic Metrics

Risk scores. A risk score is a real-valued prediction of an individual's risk, where higher values correspond to a greater risk of experiencing the event. We denote the risk scoring model by $\hat{r}$. In the static setting, $\hat{r}(X_{i,0})$ denotes a risk score for individual i given baseline covariates $X_{i,0}$. In the dynamic setting, $\hat{r}(\mathcal{H}_{i,t_k})$ denotes the time-dependent risk score for individual i at time t_k, computed using the history available up to time t_k.

B.2.1 Concordance Index (C-Index)

In the dynamic setting, the C-Index at time t is defined as

$$C_{\text{Index}}(t) \triangleq \mathbb{P}(\hat{r}(\mathcal{H}_{i,t}) > \hat{r}(\mathcal{H}_{j,t}) \mid t < T_i < T_j)\,, \tag{24}$$

where the risk scores are computed using history available up to t. Analogous to the static setting (see Appendix B.1.1), we can define a truncated, IPCW estimator of display (24)

$$\hat{C}_{\text{Index}}(t) = \frac{\sum_{i \neq j} \hat{G}(T_i \mid t)^{-2} \delta_i \mathbb{1}\big(t < T_i < T_j,\ T_i \leq T_{\max}\big) \mathbb{1}\big(\hat{r}(\mathcal{H}_{i,t}) > \hat{r}(\mathcal{H}_{j,t})\big)}{\sum_{i \neq j} \delta_i \hat{G}(T_i \mid t)^{-2} \mathbb{1}\big(t < T_i < T_j,\ T_i \leq T_{\max}\big)},$$

where $G(t + \Delta \mid t)$ denotes the conditional censoring function, or formally

$$G(t + \Delta \mid t) \triangleq \mathbb{P}(C_i > t + \Delta \mid C_i > t), \tag{25}$$

which is the probability of remaining uncensored up to time $t + \Delta$ given being uncensored at time t. Similar to the static setting, $\hat{G}(t + \Delta \mid t)$ is estimated using the Kaplan-Meier estimator (Kaplan & Meier, 1958; Pölsterl, 2020).

B.2.2 Integrated AUC

In the dynamic setting, the time dependent AUC at time $t + \Delta$, given survival up to t is given by

$$\text{AUC}(t + \Delta \mid t) \triangleq \mathbb{P}\left(\hat{r}(\mathcal{H}_{i,t}) > \hat{r}(\mathcal{H}_{j,t}) \mid t < T_i \leq t + \Delta < T_j\right) \tag{26}$$

Analogous to the static setting (see Appendix B.1.2), we form the following IPCW estimate of display (26)

$$\widehat{\text{AUC}}(t + \Delta \mid t) = \frac{\sum_{i \neq j} \hat{G}(T_i \mid t)^{-1}\, \delta_i\, \mathbb{1}\big(t < T_i \leq t + \Delta < T_j\big)\, \mathbb{1}\big(\hat{r}(\mathcal{H}_{i,t}) > \hat{r}(\mathcal{H}_{j,t})\big)}{\sum_{i \neq j} \hat{G}(T_i \mid t)^{-1}\, \delta_i\, \mathbb{1}\big(t < T_i \leq t + \Delta < T_j\big)},$$

where $G(t + \Delta \mid t)$ is the conditional censoring function; see display (25).

At each conditioning time t_k, we select the future evaluation timepoints $t_{k+\Delta}$ from the discretized horizons. For example, if the time grid is $t_1, \ldots, t_4$ and we condition on survival up to t_2, we compute the time dependent AUC for $\text{AUC}(t_3 \mid t_2)$ and $\text{AUC}(t_4 \mid t_2)$, which assess the accuracy of the predicted survival probabilities at times t_3 and t_4 given information available up to t_2.

Under our discretized setting, we estimate the Integrated AUC at t_k by

$$\overline{\text{AUC}}(t_k) = \frac{\sum_{\Delta=1}^{K-k-1} \widehat{\text{AUC}}(t_{k+\Delta} \mid t_k)\, w_{t_{k+\Delta}}}{\sum_{\Delta=1}^{K-k-1} w_{t_{k+\Delta}}},$$

where $w_{t_{k+\Delta}} = \hat{S}(t_{k+\Delta-1} \mid t_k) - \hat{S}(t_{k+\Delta} \mid t_k)$, and $\hat{S}(t + \Delta \mid t)$ is the Kaplan–Meier estimator of the population survival function at $t + \Delta$, given survival up to t, i.e. $S(t + \Delta \mid t) \triangleq \mathbb{P}(T_i > t + \Delta \mid T_i > t)$.

B.2.3 Integrated Brier Score (IBS)

In the dynamic setting, the time dependent Brier Score at time $t + \Delta$, given survival up to t can be defined as

$$\text{BS}(t + \Delta \mid t) \triangleq \mathbb{E}\Big[\Big(\mathbb{1}\{T_i > t + \Delta\} - \hat{S}(t + \Delta \mid T_i > t,\ \mathcal{H}_{i,t})\Big)^2\Big]. \tag{27}$$

An analogous estimator for display (27) is

$$\widehat{\text{BS}}(t + \Delta \mid t) = \frac{1}{n} \sum_{i=1}^{n} \Bigg[\frac{\mathbb{1}(t < T_i \leq t + \Delta,\, \delta_i = 1)}{\hat{G}(T_i \mid t)} \big(0 - \hat{S}(t + \Delta \mid T_i > t,\ \mathcal{H}_{i,t})\big)^2 + \frac{\mathbb{1}(T_i > t + \Delta)}{\hat{G}(t + \Delta \mid t)} \big(1 - \hat{S}(t + \Delta \mid T_i > t,\ \mathcal{H}_{i,t})\big)^2 \Bigg],$$

where $G(t + \Delta \mid t)$ is the conditional censoring function, as in display (25).

Similar to the dynamic Integrated AUC, for each conditioning time t_k, we select the future evaluation timepoints $t_{k+\Delta}$ from the discretized horizons. Under this discretized setting, the Integrated Brier Score at time t_k can be estimated by

$$\overline{\mathrm{BS}}(t_k) = \frac{1}{t_{K-1} - t_{k+1}} \int_{t_{k+1}}^{t_{K-1}} \widehat{\mathrm{BS}}(t \mid t_k)\, dt.$$

B.3 ELO Rating

The Elo rating system is a method for ranking competitors based on the outcomes of pairwise comparisons. Each model is assigned a numerical rating that reflects its relative performance, with higher values indicating better performance. All models start from the same initial rating (which we set to 1000), and their ratings are updated after each comparison depending on whether a model performs better or worse than expected given its current rating. We implement Elo using the `elote` Python package[3] with the default $\tilde{K}$-factor of 32, which controls how strongly each comparison influences the ratings.

Static setting. In the static setting, performance is reported for multiple values of time discretization granularities $K \in \{5, 10, 15, 20\}$. We first average each model's metric values across all K values within each dataset. Using these aggregated values, we build one Elo arena per dataset and compare all models pairwise. The final static Elo score is the average of a model's Elo ratings across datasets.

Dynamic setting. In the dynamic setting, performance is reported at multiple time points across $K \in \{5, 8\}$. We aggregate each model's metric values over time points t_1, t_2, t_3 within each dataset and across both granularities. Using these aggregated values, we again build one Elo arena per dataset and compare all models pairwise. The final dynamic Elo score is the average of a model's Elo ratings across datasets.

C Modeling Details

C.1 Static Models

C.1.1 Classification-based models

Subsampling. The context windows of tabular foundation models are generally limited. To manage this, after constructing the supervised learning dataset as described in Section 4.1, we randomly subsample the training set to the maximum number of rows supported by each model when needed ($10,000$ for MITRA and $50,000$ for TabPFN); no such limit is imposed for XGBoost[4].

Clipping survival probabilities. To ensure our survival probabilities are non-increasing in time, we apply a post-processing step that enforces monotonicity over time. Specifically, we refine our definition of $\hat{S}(t_k \mid X_{i,0})$ from display (6) as follows:

$$\hat{S}(t_k \mid X_{i,0}) \triangleq \min_{j \leq k} \; 1 - \hat{p}(X_{i,0}, t_k).$$

This ensures that $\hat{S}(t_1 \mid X_{i,0}) \geq \hat{S}(t_2 \mid X_{i,0}) \geq \cdots \geq \hat{S}(t_{K-1} \mid X_{i,0})$. Our definition of the risk score from display (11) uses these clipped survival probabilities.

C.1.2 Alternative classification formulations

To isolate the effect of our *direct* formulation $\mathbb{P}(T_i > t_k \mid X_{i,0})$ from the effect of the underlying classifier, we additionally evaluate *discrete-time hazard* (DH) and *multi-class* (MC) variants of all three classifiers: MITRA-DH/TabPFN-DH/XGBoost-DH and MITRA-MC/TabPFN-MC/XGBoost-MC. These variants share the same data preprocessing, time discretization, subsampling, and hyperparameter selection procedures as their direct counterparts (Section C.1.1); the only difference is the prediction target and the inference formula. We evaluate the DH and MC variants at the same $K \in \{5, 10, 15, 20\}$ as the direct variants.

[3] `https://pypi.org/project/elote/`
[4] `https://pypi.org/project/xgboost/`

Discrete-time hazard (DH) classification formulation. This formulation is based on the following works: Efron (1988); Allison (1982); Craig et al. (2025); Zhong & Tibshirani (2019). For each example i and each bin index $k \in \{1, \ldots, K-1\}$, we form a training tuple

$$\big((X_{i,0}, t_k),\ Z_{i,k}\big), \quad \text{where } Z_{i,k} \triangleq \mathbb{1}\big(T_i \in (t_{k-1}, t_k]\big),$$

i.e., the indicator that the event occurred in the kth interval among subjects at risk at t_{k-1}. We train the classifier to predict $Z_{i,k}$ given $(X_{i,0}, t_k)$ using a binary cross-entropy loss to obtain $\hat{\lambda}(t_k \mid X_{i,0}) \triangleq \mathbb{P}(T_i \in (t_{k-1}, t_k] \mid X_{i,0}, T_i > t_{k-1})$, i.e., an estimate of the discrete-time hazard.

Given the estimated hazards, we reconstruct the survival probability via the classical multiplicative product

$$\hat{S}(t_k \mid X_{i,0}) = \prod_{j=1}^{k} \big(1 - \hat{\lambda}(t_j \mid X_{i,0})\big), \qquad k = 1, \ldots, K-1.$$

The risk score is defined analogously to the direct formulation, replacing the directly modeled survival probability with the reconstructed one in display (11).

Multi-class (MC) classification formulation. This formulation is based on that used in Qi et al. (2026). For each example i, we form a training tuple

$$\big((X_{i,0}, \delta_i),\ Q_i\big), \quad \text{where } Q_i \triangleq k \quad \text{if } \min(C_i, T_i) \in (t_{k-1}, t_k].$$

We train the classifier $\hat{p}(Q_i = k \mid X_{i,0}, \delta_i)$ to predict Q_i given $(X_{i,0}, \delta_i)$ using a multi-class cross-entropy loss. At inference time, estimate the survival probability as follows (note that we do not require knowing the true δ_i to form survival probabilities, we always input $\delta = 1$ into the model):

$$\hat{S}(t_k \mid X_i) = 1 - \sum_{j=1}^{k} \hat{p}(Q_i = j \mid X_{i,0}, \delta = 1).$$

The risk score is defined analogously to the direct formulation, replacing the directly modeled survival probability with the reconstructed one in display (11).

C.1.3 Cox Proportional Hazards (CoxPH)

We use the `sksurv` package to implement the CoxPH model.[5] The CoxPH model relates the baseline covariates $X_{i,0}$ to the hazard of an event at time t using $\lambda(t \mid X_{i,0}) = \lambda_0(t) \exp((X_{i,0})^\top \beta)$, where β is the vector of regression coefficients, and $\lambda_0(t)$ is the baseline hazard function, a population-level function defined as the hazard for the baseline covariate $X = \mathbf{0}$.

$$\lambda_0(t) \triangleq \lambda(t \mid X_{i,0} = \mathbf{0}). \tag{28}$$

The coefficients β are estimated by maximizing the Cox partial likelihood function (Cox, 1972). Let $\hat{\beta}$ denote the fitted coefficients, the risk score for an individual i is defined as the inner product between the covariates and the coefficients:

$$\hat{r}(X_{i,0}) = X_{i,0}^\top \hat{\beta}.$$

The corresponding survival function is

$$\hat{S}(t \mid X_{i,0}) = \hat{S}_0(t)^{\exp(X_{i,0}^\top \hat{\beta})},$$

where $\hat{S}_0(t)$ is the estimated baseline survival function, i.e., the survival function for the reference covariate value $X_{i,0} = \mathbf{0}$. Formally, the baseline survival function is

$$S_0(t) \triangleq \mathbb{P}(T > t \mid X_{i,0} = \mathbf{0}), \tag{29}$$

and $\hat{S}_0(t)$ is estimated using the Breslow estimator (Breslow, 1974).

[5] `https://scikit-survival.readthedocs.io/en/stable/api/generated/sksurv.linear_model.CoxPHSurvivalAnalysis.html`

C.1.4 Random Survival Forests (RSF)

We use the `sksurv` RSF implementation.[6] RSF is a nonparametric ensemble model that extends random forests to right-censored survival data by learning an ensemble of tree-based cumulative hazard functions (Ishwaran et al., 2008). An RSF consists of an ensemble of B survival trees. Each tree partitions the feature space into terminal nodes, and for each terminal node a cumulative hazard function is estimated from the training samples in that node using the Nelson-Aalen estimator (Nelson, 1972; Aalen, 1978). For an individual i, let $\ell_b(X_{i,0})$ denote the terminal node in tree b to which $X_{i,0}$ is assigned. The tree-level cumulative hazard is given by $\hat{H}_b(t \mid X_{i,0}) = \hat{H}_{b,\ell_b(X_{i,0})}(t)$. The forest-level cumulative hazard is obtained by averaging over trees: $\hat{H}(t \mid X_{i,0}) = \frac{1}{B}\sum_{b=1}^{B} \hat{H}_b(t \mid X_{i,0})$. The risk score for an individual i can be computed as follows:

$$\hat{r}(X_{i,0}) = \sum_{t' \in \mathcal{T}} \hat{H}(t' \mid X_{i,0}),$$

where $\mathcal{T}$ is a set that contains all the distinct event times observed in the training data. The corresponding survival function is computed from the forest-level cumulative hazard as

$$\hat{S}(t \mid X_{i,0}) = \exp\big(-\hat{H}(t \mid X_{i,0})\big).$$

C.1.5 DeepSurv

We use the `pycox` implementation of DeepSurv.[7] DeepSurv is a neural-network extension of the CoxPH model in which the linear risk function is replaced by a nonlinear function learned by a deep neural network (Katzman et al., 2018). In DeepSurv, the hazard for an individual with covariates $X_{i,0}$ is $\lambda(t \mid X_{i,0}) = \lambda_0(t)\exp\big(f_\theta(X_{i,0})\big)$, where $\lambda_0(t)$ is the baseline hazard defined in display (28) and $f_\theta(X_{i,0})$ is a neural network with parameters θ that replaces the linear term $(X_{i,0})^\top\beta$ in CoxPH. The network weights θ are then learned by maximizing the Cox partial likelihood. After training, the risk score for an individual i is

$$\hat{r}(X_{i,0}) = f_\theta(X_{i,0}),$$

and the corresponding survival function is

$$\hat{S}(t \mid X_{i,0}) = \hat{S}_0(t)^{\exp(f_\theta(X_{i,0}))},$$

where $\hat{S}_0(t)$ is the estimated baseline survival function defined in display (29); we use the Breslow estimator (Breslow, 1974).

C.1.6 DeepHit

We use the `pycox` implementation of DeepHit.[8] DeepHit is a neural-network–based, nonparametric survival model that directly learns the discrete-time distribution of event times without assuming proportional hazards (Lee et al., 2018). We apply a discretization setting similar to our classifier-based models as described in Section 4. For an input $X_{i,0}$, the network outputs a probability mass function over a discretized time with K ordered time bins. Specifically, DeepHit produces estimates

$$p_{\theta,k}(X_{i,0}) \approx \mathbb{P}(T_i \in \tau_k \mid X_{i,0}), \qquad k = 1, \ldots, K,$$

where $\tau_k = (t_{k-1}, t_k]$ are time bins induced by cut points $0 = t_0 < \cdots < t_K = \infty$. We obtain the risk score by computing the negative expected bin index under the predicted distribution,

$$\hat{r}(X_{i,0}) = -\sum_{k=1}^{K} k\, p_{\theta,k}(X_{i,0}).$$

The survival probability at t_k is then given by

$$\hat{S}(t_k \mid X_{i,0}) = 1 - \sum_{j=1}^{k} p_{\theta,j}(X_{i,0}).$$

[6] `https://scikit-survival.readthedocs.io/en/stable/api/generated/sksurv.ensemble.RandomSurvivalForest.html`
[7] `https://github.com/havakv/pycox`
[8] `https://github.com/havakv/pycox`

C.2 Dynamic Models

Last-Observation-Carried-Forward (LOCF). We employ a slightly modified version of the last-observation-carried-forward (LOCF) method in several of the dynamic models to represent the information available up to a particular time (Little & Rubin, 2019). Recall from display (3) that dynamic survival prediction focuses on the conditional survival probability for a future horizon $\Delta > 0$, given the information available at time t. Since the covariates are usually observed at irregular intervals, their covariate values are generally not available exactly at time t. We therefore apply the LOCF method, where the time-dependent covariate at time t is imputed by its most recent observed value prior to t.

Recall from Section 2.2 that the observed covariate history for individual i up to time t is $\mathcal{H}_{i,t} \triangleq \{(s, X_{i,s}) : s \in \mathcal{T}_i,\ s \leq t\}$, where $\mathcal{T}_i$ denotes the set of observation times for individual i. Define the most recent observation time prior to t by

$$s_i(t) \triangleq \arg \max_{s \in \mathcal{T}_i :\, s \leq t} s.$$

The LOCF representation of the covariate history at time t is then

$$X_{i,t}^{\text{LOCF}} \ \triangleq\ X_{i,s_i(t)}. \tag{30}$$

C.2.1 Classification-Based Models

Representation of history ϕ during training. Recall from Section 4.2, the classification-based dynamic models (MITRA, TabPFN, and XGBoost) take the vector-valued function $\phi_k(\mathcal{H}_{i,t_k}, Y_{i,1:k}, t_k, t_{k+\Delta})$ as input, and predict the binary label $Y_{i,t_{k+\Delta}}$ during training. Let $\bar{X}_{i,t_k}$ be the running means of the covariates computed up to t_k. We consider the following feature representations:

- Base representation: $\phi_k(\mathcal{H}_{i,t_k},\, Y_{i,1:k},\, t_k,\, t_{k+\Delta}) = \Big[X_{i,t_k}^{\text{LOCF}},\ \bar{X}_{i,t_k},\ t_k,\ t_{k+\Delta}\Big]$
- Representation incorporating the time elapsed since the most recent observation prior to t_k: $\phi_k(\mathcal{H}_{i,t_k},\, Y_{i,1:k},\, t_k,\, t_{k+\Delta}) = \Big[X_{i,t_k}^{\text{LOCF}},\ \bar{X}_{i,t_k},\ t_k,\ t_{k+\Delta},\ t_k - s_i(t_k)\Big]$
- Representation incorporating the info (i.e. the index) of the prediction horizon: $\phi_k(\mathcal{H}_{i,t_k},\, Y_{i,1:k},\, t_k,\, t_{k+\Delta}) = \Big[X_{i,t_k}^{\text{LOCF}},\ \bar{X}_{i,t_k},\ t_k,\ t_{k+\Delta},\ k+\Delta\Big]$
- Full Representation: $\phi_k(\mathcal{H}_{i,t_k},\, Y_{i,1:k},\, t_k,\, t_{k+\Delta}) = \Big[X_{i,t_k}^{\text{LOCF}},\ \bar{X}_{i,t_k},\ t_k,\ t_{k+\Delta},\ t_k - s_i(t_k),\ k+\Delta\Big]$

See Appendix D.3 on how we select which representation to use.

Clipping Survival Probabilities. Since $\hat{S}(t_{k+\Delta} \mid T_i > t_k, \mathcal{H}_{i,t_k})$ must be non-increasing in Δ, we apply a similar monotonicity post-processing described in Appendix C.1.1. We refine display (8) by the following

$$\hat{S}(t_{k+\Delta} \mid T_i > t_k,\ \mathcal{H}_{i,t_k}) \triangleq \min_{1 \leq \ell \leq \Delta} 1 - \hat{p}(\mathcal{H}_{i,k}, Y_{i,1:k}, t_k, t_{k+\ell}), \qquad \Delta = 1, \ldots, K-1-k.$$

This ensures that $\hat{S}(t_{k+1} \mid t_k,\ \mathcal{H}_{i,t_k}) \geq \hat{S}(t_{k+2} \mid t_k,\ \mathcal{H}_{i,t_k}) \geq \cdots \geq \hat{S}(t_{K-1} \mid t_k,\ \mathcal{H}_{i,t_k})$.

The time-dependent risk score is defined by averaging the predicted future cumulative risks over the remaining time horizon. For individual i at time t_k, the risk score is

$$\hat{r}(\mathcal{H}_{i,t_k}) = \frac{1}{K-1-k} \sum_{\Delta=1}^{K-1-k} \Big(1 - \hat{S}(t_{k+\Delta} \mid T_i > t_k,\ \mathcal{H}_{i,t_k})\Big).$$

C.2.2 Landmark CoxPH

Since the Cox Proportional Hazards (CoxPH) model requires static covariates, we adopt the landmarking approach using LOCF (Little & Rubin, 2019). Again, we use the `sksurv` package to implement the CoxPH model. For each t_k, the covariate history is represented by

$$\phi_{\text{landmark}}(\mathcal{H}_{i,t_k}) \triangleq (X_{i,t_k}^{\text{LOCF}},\ \bar{X}_{i,t_k}) \tag{31}$$

where we also included the running means of the covariates computed up to t_k, $\bar{X}_{i,t_k}$. For each t_k, $k \in \{1, \ldots, K-2\}$, we fit a separate CoxPH model using only individuals who are still event-free at t_k. Each landmark CoxPH model yields regression coefficient estimates $\hat{\beta}_{t_k}$. Analagous to the static setting of the CoxPH model (see Appendix C.1.3), the time-dependent risk score for individual i at t_k can be computed as

$$\hat{r}(\mathcal{H}_{i,t_k}) = \phi_{\text{landmark}}(\mathcal{H}_{i,t_k})^\top \hat{\beta}_{t_k}.$$

Survival predictions over a horizon $\Delta > 0$ are obtained conditionally on survival up to t_k:

$$\hat{S}(t_{k+\Delta} \mid T_i > t_k,\ \mathcal{H}_{i,t_k}) = \hat{S}_0\left(t_{k+\Delta} \mid t_k\right)^{\exp(\hat{r}(\mathcal{H}_{i,t_k}))}.$$

where $\hat{S}_0(t_{k+\Delta} \mid t_k)$ is the estimated baseline survival function, i.e., the survival function for a reference individual with zero covariates. Formally, the baseline survival function conditional on time t_k is

$$S_0(t_{k+\Delta} \mid t_k) \triangleq \mathbb{P}\left(T > t_{k+\Delta} \mid T > t_k,\ \phi_{\text{landmark}}(\mathcal{H}_{i,t_k}) = 0\right), \tag{32}$$

and $\hat{S}_0(t_{k+\Delta} \mid t_k)$ is estimated using the Breslow estimator (Breslow, 1974).

C.2.3 Landmark Random Survival Forest

Similar to the landmark CoxPH model, at each landmark time t_k, $k \in \{1, \ldots, K-2\}$, we fit a separate random survival forest (RSF) using only individuals who are event-free at t_k, $\phi_{\text{landmark}}(\mathcal{H}_{i,t_k})$ defined in display (31) as a representation of the history. Again, we use the `sksurv` RSF implementation. Each landmark-specific forest estimates a cumulative hazard function for $t \geq t_k$, $\hat{H}_{t_k}(t \mid T_i > t_k,\ \mathcal{H}_{i,t_k})$. The time-dependent risk score for individual i at landmark time t_k is defined as

$$\hat{r}(\mathcal{H}_{i,t_k}) = \sum_{t' \in \mathcal{T}} \hat{H}_{t_k}(t' \mid T_i > t_k,\ \mathcal{H}_{i,t_k}),$$

where $\mathcal{T}$ is a set that contains all the distinct event times observed in the training data. The corresponding survival probability over a horizon Δ, conditional on survival up to the landmark time, is given by

$$\hat{S}(t_{k+\Delta} \mid T_i > t_k,\ \mathcal{H}_{i,t_k}) = \exp\Big(-\hat{H}_{t_k}(t_{k+\Delta} \mid T_i > t_k,\ \mathcal{H}_{i,t_k})\Big).$$

C.2.4 Dynamic DeepHit

Dynamic DeepHit (Lee et al., 2019) extends DeepHit to the dynamic setting by modeling the conditional distribution of future event times given an individual's longitudinal history. We use the following public implementation: `https://github.com/Jeanselme/DynamicDeepHit/tree/main`

We use a fixed discretization with K time bins that aligns with the time discretization approach described in Section 4. The implementation of Dynamic DeepHit takes as input the longitudinal history $\mathcal{H}_{i,t_k}$ and outputs the estimates of the cumulative incidence distribution,

$$\hat{p}_{t_k,k+\Delta}(\mathcal{H}_{i,t_k}) \approx \mathbb{P}(T_i \leq t_{k+\Delta} \,|\, T_i > t_k,\ \mathcal{H}_{i,t_k}), \qquad \Delta = 1, \ldots, K-k-1.$$

We define the time-dependent risk score at time t_k as the predicted probability of experiencing the event in the next interval,

$$\hat{r}(\mathcal{H}_{i,t_k}) = \hat{p}_{t_k,k+1}(\mathcal{H}_{i,t_k}).$$

The conditional survival probability at $t_{k+\Delta}$ given survival up to t_k is

$$\hat{S}(t_{k+\Delta} \mid T_i > t_k,\ \mathcal{H}_{i,t_k}) = 1 - \hat{p}_{t_k,k+\Delta}(\mathcal{H}_{i,t_k}).$$

C.2.5 Joint Model

The joint model links a longitudinal process and a survival outcome through shared subject-specific random effects (Ibrahim et al., 2010). We implement the joint model using the `JMbayes2` package.[9] We model the longitudinal process using a linear mixed-effects model, while the survival outcome is modeled using a CoxPH model.

Given survival up to a time t_k and the observed longitudinal history $\mathcal{H}_{i,t_k}$ the joint model yields dynamic survival predictions at future times $t_{k+\Delta} > t_k$. At each t_k, the predicted event probability at a future time point $t_{k+\Delta}$ is denoted using $\hat{p}_{t_k,k+\Delta}(\mathcal{H}_{i,t_k})$, which can be obtained by calling the `predict()` function in the `JMbayes2` package. We define the time-dependent risk score at t_k as the predicted probability the event occurs at the next interval

$$\hat{r}(\mathcal{H}_{i,t_k}) = \hat{p}_{t_k,k+1}(\mathcal{H}_{i,t_k}),$$

and the conditional survival probability at $t_{k+\Delta}$ can be computed using

$$\hat{S}(t_{k+\Delta} \mid T_i > t_k,\ \mathcal{H}_{i,t_k}) = 1 - \hat{p}_{t_k,k+\Delta}(\mathcal{H}_{i,t_k}).$$

D Additional Experimental Details and Results

D.1 Datasets

Static. Across 43 static datasets (Drysdale, 2022), the median number of unique samples was 461 (range 26–6805). The number of covariates ranged from 3 to 160. The censoring rate ranged from 6.6% to 94.4%.

Dynamic. Across 5 dynamic datasets (Drysdale, 2022), the median number of unique samples was 467 (range 103–4603). The number of covariates ranged from 4 to 7. The patient-wise censoring rate ranged from 27.2% to 66.2%.

D.2 Data Preprocessing

We split each dataset into training, validation, and test sets (70:15:15) using a stratified procedure on the event indicator to preserve the event–censoring proportion across splits. Categorical covariates were then transformed via one-hot encoding, fitted on the training set, and the resulting feature matrices for the validation and test sets were aligned to the training design matrix by adding any missing dummy columns and filling them with zeros. Missing covariate values were handled using a `SimpleImputer`[10] fit on the training data only and subsequently applied to the validation and test sets, ensuring that no information from the held-out splits informed preprocessing.

D.3 Hyperparameters

Model selection approach. For each dataset, we select hyperparameters using the best performing model in terms of *C-Index* (IPCW estimator) on the validation split.

Static search spaces For the baselines, we perform a small grid search:

- **CoxPH:** ℓ_2 regularization strength $\alpha \in \{10^{-6}, 10^{-3}, 10^{-1}\}$.
- **RSF:** number of trees $n_{\text{estimators}} \in \{50, 100, 200\}$, min split $\in \{2, 5, 10, 20\}$.
- **DeepSurv:** dropout $\in \{0, 0.1\}$, hidden layers $\in \{(128, 32), (256, 32), (512, 32)\}$, learning rate $\in \{10^{-3}, 10^{-4}, 10^{-5}\}$.
- **DeepHit:** dropout $\in \{0, 0.1\}$, hidden layers $\in \{(128, 32), (256, 32), (512, 32)\}$, learning rate $\in \{10^{-3}, 10^{-4}, 10^{-5}\}$.

[9] `https://cran.r-project.org/web/packages/JMbayes2/index.html`
[10] `https://scikit-learn.org/stable/modules/generated/sklearn.impute.SimpleImputer.html`

Table 3: Summary statistics for each static survival dataset.

No.	Dataset	No. of covariates	No. of unique samples	Censoring (%)
1	Aids2	4	2839	38.0
2	Dialysis	4	6805	76.4
3	Framingham	7	4699	68.7
4	GBSG2	8	686	56.4
5	Melanoma	5	205	72.2
6	TRACE	6	1878	49.0
7	UnempDur	6	3241	38.7
8	Unemployment	5	452	43.4
9	actg	11	1151	91.7
10	breast	4	100	74.0
11	burn	11	154	68.8
12	cancer	8	228	27.6
13	cgd	10	128	65.6
14	colon	9	929	51.3
15	cost	13	518	22.0
16	d.oropha.rec	12	192	27.6
17	dataDIVAT1	5	5943	83.6
18	dataDIVAT2	4	1837	68.3
19	dataDIVAT3	7	4267	94.4
20	dataOvarian1	160	912	40.4
21	diabetes	4	197	60.7
22	e1684	3	284	31.0
23	follic	5	541	35.7
24	glioma	4	37	37.8
25	hepatoCellular	43	227	57.3
26	mgus	9	241	6.6
27	nki70	75	144	66.7
28	nwtco	7	4028	85.8
29	ova	5	358	25.7
30	ovarian	4	26	53.8
31	pbc	6	312	59.9
32	phpl04K8a	21	442	46.6
33	prostate	15	502	29.5
34	rdata	4	1040	47.4
35	retinopathy	7	197	60.7
36	scania	5	1931	43.8
37	smarto	26	3873	88.1
38	stagec	7	146	63.0
39	uis	8	628	19.1
40	veteran	6	137	6.6
41	vlbw	22	617	82.7
42	whas500	16	461	61.8
43	zinc	13	431	81.2

Table 4: Summary statistics for each dynamic survival dataset.

No.	Dataset	No. of covariates	No. of unique samples	Censoring (%)
1	aids	5	467	59.7
2	csl	6	446	39.5
3	epileptic	5	556	66.2
4	heart	4	103	27.2
5	oldmort	7	4603	57.2

Note that we do not define any hyperparameters for the classifier models in the static setting.

Dynamic search spaces For the baselines, we perform a small grid search:

- **Landmark CoxPH:** ℓ_2 regularization strength $\alpha \in \{10^{-6}, 10^{-3}, 10^{-1}\}$.
- **Landmark RSF:** number of trees $n_{\text{estimators}} \in \{100, 200, 500, 1000\}$, max depth $\in \{3, 10, 50, \texttt{None}\}$, min split $\in \{10, 50\}$.
- **Dynamic DeepHit:** RNN hidden units $\in \{50, 100\}$, fully connected hidden layers $\in \{(100, 100), (100, 100, 100)\}$, loss parameter $\sigma \in \{1, 3\}$, learning rate $\in \{10^{-4}, 10^{-3}, 5 \times 10^{-3}, 10^{-2}\}$.
- **Joint model:** standard deviation of the Normal prior on the association parameter, $\alpha \sim \mathcal{N}(0, \sigma_\alpha^2)$ with $\sigma_\alpha \in \{10^{-6}, 10^{-3}, 10^{-1}\}$.

Recall from Appendix C.2.1 the representations of the features ϕ. For all classifier-based dynamic models (MITRA, TabPFN, and XGBoost), we tune how temporal information is encoded in the input features using two binary hyperparameters:

- `quantile_info`: whether the index of the prediction horizon is included as an additional covariate;
- `time_since`: whether the time elapsed since the most recent observation prior to the timepoint t_k is included as a covariate.

Each binary flag takes values in $\{0, 1\}$, so there are four combinations of feature representations. For XGBoost, we additionally tune the tree ensemble parameters: `n_estimators` $\in \{100, 200\}$, `learning_rate` $\in \{0.1, 0.3\}$, and `max_depth` $\in \{3, 6\}$.

D.4 Sensitivity to discretization granularity K

We assess the sensitivity of our results for the number of discretization bins $K \in \{5, 10, 15, 20\}$. In Figure 5, we show results for the best performing model in each class. In Figure 6, we show results for different TFM-based approaches.

Why the DH formulation degrades at large K. The DH reconstruction $S(t_k \mid X_{i,0}) = \prod_{j=1}^{k-1}(1 - \hat{\lambda}(t_j \mid X_{i,0}))$ multiplies $K-1$ per-bin estimates. Even small per-bin estimation errors compound multiplicatively across these factors, so the error in $\hat{S}(t_k \mid X_{i,0})$ grows with K. This is structurally analogous to the error-accumulation problem documented for iterated multistep forecasting in macroeconomics (Marcellino et al., 2006).

Model-specific behavior. The interaction between the choice of TFM and the choice of formulation is non-uniform (Figure 6). For MITRA, the direct formulation is preferred at larger K; for TabPFN, the DH variant can be competitive at small K but degrades sharply at large K, particularly in IBS. We hypothesize this reflects different pretraining inductive biases of the two TFMs and, given the lack of survival-specific pretraining, leaves open the possibility that a TFM directly pretrained for survival would close this gap.

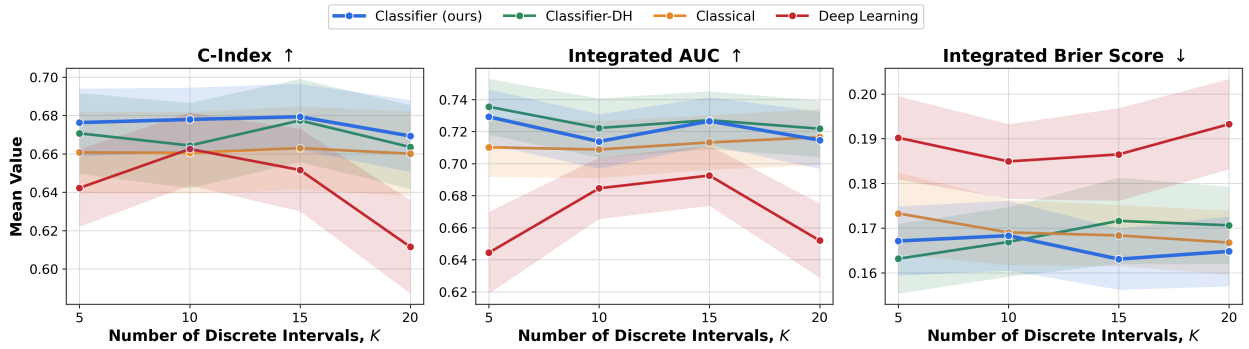


Figure 5: **Static model performance over discretization granularities $K \in \{5, 10, 15, 20\}$ by model group.** We group methods into classical (CoxPH, RSF), deep learning (DeepSurv, DeepHit), our cumulative failure formulation (MITRA-CF, TabPFN-CF, XGBoost-CF), and the discrete-time hazard variants (MITRA-DH, TabPFN-DH, XGBoost-DH). Within each group we pick the best-performing model by validation C-index. Our cumulative failure models are among the best across all three metrics, and is particularly strong on IBS at larger K—compared to the *DH* models that degrade with larger K.

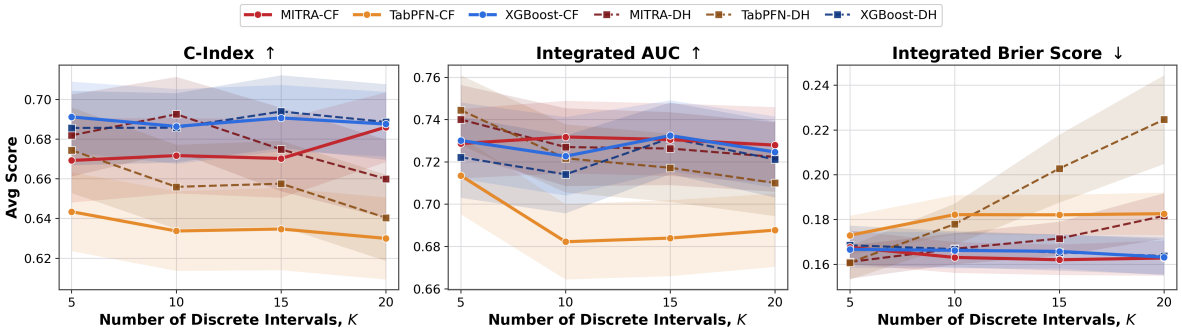


Figure 6: **Tabular foundation model performance over discretization granularities K (per-model).** MITRA-CF (our direct formulation) is the strongest TFM at larger K. At smaller K, MITRA-DH and TabPFN-DH can be competitive on C-index or Integrated AUC, but TabPFN-DH degrades sharply as K grows, especially in IBS, consistent with multiplicative error compounding in the survival reconstruction $S(t_k) = \prod_j (1 - \lambda(t_j))$.

D.5 Censoring-aware binning via Kaplan–Meier quantiles

Throughout the main paper, we set the bin boundaries $\{t_k\}_{k=1}^{K-1}$ as quantiles of the *observed* event times in the training set. Under heavy censoring, observed events are a non-random subset of all events, which could in principle bias the chosen discretization. In this appendix section, we evaluate an alternative *censoring-aware* binning strategy. Following Kvamme & Borgan (2021), we consider an alternative binning where the boundaries are chosen as equi-spaced quantiles of the Kaplan–Meier (KM) estimate of the marginal survival function $S(t) = \mathbb{P}(T_i > t)$ computed on the training data. This places more bin boundaries in regions of high event density even when those regions are partially obscured by censoring, and is the standard alternative in the discrete-time survival literature.

Table 5 reports the aggregate static performance under the KM-quantile binning, with the same metrics, baselines, and $K \in \{5, 10, 15, 20\}$ as Table 1 in the main paper. Additionally, Figures 7 and 8 depict results for different discretization granularities K. Overall, the qualitative picture is unchanged under this alternative binning approach: MITRA-CF remains the best overall, the cumulative failure approach outperforms discrete hazard modeling for each classifier model type. We conclude that our classification formulation is *robust to the choice of binning rule*, despite heavy and heterogeneous censoring across the 43 datasets.

Table 5: **Average performance comparison in the static setting (discretization using quantiles of the Kaplan–Meier curve).** Values represent means and standard errors aggregated over 43 datasets and 4 time-discretization granularities ($K \in \{5, 10, 15, 20\}$). The best and second-best performers are highlighted in dark blue and light blue, respectively.

Model	C-index			Integrated AUC			Integrated Brier Score			Avg. Rank ↓	Avg. ELO ↑
	Value ↑	Rank ↓	ELO ↑	Value ↑	Rank ↓	ELO ↑	Value ↓	Rank ↓	ELO ↑		
Classification-based Models											
MITRA-CF	$0.674_{\pm 0.017}$	$\mathbf{4.1}_{\pm 0.3}$	**1039**	$\mathbf{0.736}_{\pm 0.015}$	$\mathbf{3.5}_{\pm 0.3}$	**1058**	$\mathbf{0.162}_{\pm 0.007}$	$\mathbf{3.7}_{\pm 0.3}$	**1051**	$\mathbf{3.7}_{\pm 0.3}$	**1049**
TabPFN-CF	$0.635_{\pm 0.020}$	$7.0_{\pm 0.4}$	961	$0.706_{\pm 0.018}$	$6.6_{\pm 0.4}$	970	$0.177_{\pm 0.009}$	$7.1_{\pm 0.4}$	958	$6.9_{\pm 0.3}$	963
XGBoost-CF	$0.681_{\pm 0.015}$	$4.7_{\pm 0.4}$	1022	$0.725_{\pm 0.015}$	$4.6_{\pm 0.4}$	1023	$0.164_{\pm 0.008}$	$4.2_{\pm 0.4}$	1036	$4.5_{\pm 0.3}$	1027
Discrete-time Hazard Variants											
MITRA-DH	$0.671_{\pm 0.016}$	$4.6_{\pm 0.3}$	1023	$0.732_{\pm 0.017}$	$4.7_{\pm 0.3}$	1023	$0.167_{\pm 0.007}$	$5.0_{\pm 0.3}$	1013	$4.8_{\pm 0.3}$	1020
TabPFN-DH	$0.659_{\pm 0.020}$	$4.8_{\pm 0.4}$	1021	$0.733_{\pm 0.016}$	$3.9_{\pm 0.4}$	1044	$0.179_{\pm 0.011}$	$5.5_{\pm 0.5}$	1002	$4.7_{\pm 0.4}$	1022
XGBoost-DH	$0.681_{\pm 0.017}$	$4.7_{\pm 0.4}$	1021	$0.724_{\pm 0.017}$	$5.3_{\pm 0.3}$	1003	$0.165_{\pm 0.008}$	$4.3_{\pm 0.4}$	1032	$4.8_{\pm 0.3}$	1019
Classical & Neural Survival Baselines											
CPH	$0.682_{\pm 0.017}$	$4.5_{\pm 0.5}$	1025	$0.726_{\pm 0.017}$	$4.5_{\pm 0.5}$	1023	$0.165_{\pm 0.008}$	$4.7_{\pm 0.4}$	1020	$4.6_{\pm 0.4}$	1023
RSF	$\mathbf{0.684}_{\pm 0.018}$	$5.3_{\pm 0.4}$	1004	$0.719_{\pm 0.017}$	$5.7_{\pm 0.4}$	994	$0.166_{\pm 0.007}$	$5.1_{\pm 0.4}$	1010	$5.4_{\pm 0.3}$	1003
DeepHit	$0.607_{\pm 0.012}$	$8.8_{\pm 0.3}$	910	$0.629_{\pm 0.013}$	$8.9_{\pm 0.3}$	909	$0.206_{\pm 0.008}$	$8.7_{\pm 0.3}$	914	$8.8_{\pm 0.3}$	911
DeepSurv	$0.649_{\pm 0.018}$	$6.5_{\pm 0.4}$	974	$0.687_{\pm 0.017}$	$7.3_{\pm 0.3}$	954	$0.178_{\pm 0.008}$	$6.9_{\pm 0.3}$	964	$6.9_{\pm 0.3}$	964

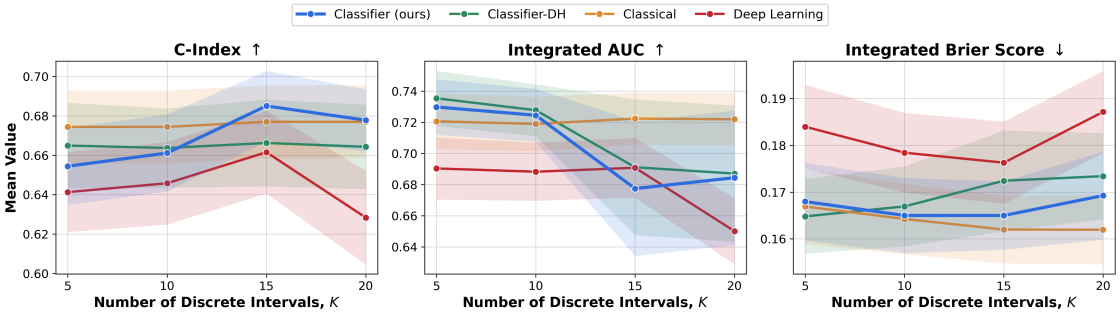


Figure 7: **Static model performance over discretization granularities $K \in \{5, 10, 15, 20\}$ by model group (Kaplan–Meier-quantile binning).** Same grouping and metrics as Figure 5, except that bin boundaries follow KM-quantile binning instead of observed-event-time quantiles. The qualitative ranking is unchanged: our cumulative failure classification based models is among the best across all metrics, and particularly strong on IBS at larger K.

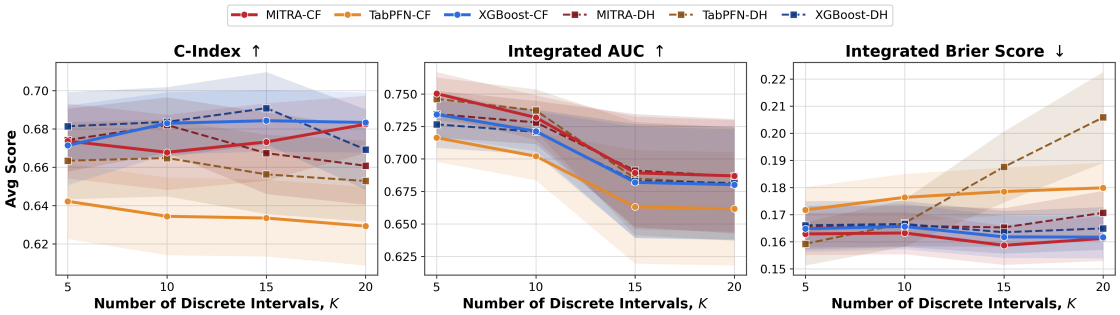


Figure 8: **Tabular foundation model performance over discretization granularities K (Kaplan–Meier-quantile binning).** MITRA-CF remains the strongest TFM at larger K, and TabPFN-DH degrades as K grows, which qualitatively matches the results from Figure 6 in Appendix D.4.

D.6 TabPFN v3

Throughout the paper, all experiments primarily use TabPFN v2.5 to maintain consistency with the main experimental setup. Following the public release of TabPFN v3, we additionally report TabPFN v3 results in Table 6 for completeness.

Table 6: **Average performance comparison.** Results are averaged over four discretization granularities ($K \in \{5, 10, 15, 20\}$). Values denote mean ± standard error. The best and second-best performers are highlighted in dark blue and light blue, respectively.

Model	C-Index			Integrated AUC			Integrated Brier Score			Avg. Rank ↓	Avg. ELO ↑
	Value ↑	Rank ↓	ELO ↑	Value ↑	Rank ↓	ELO ↑	Value ↓	Rank ↓	ELO ↑		
Classification-based Models											
MITRA-CF	$0.674_{\pm 0.017}$	$\mathbf{5.7}_{\pm 0.5}$	**1055**	$\mathbf{0.729}_{\pm 0.016}$	$\mathbf{5.6}_{\pm 0.5}$	1058	$\mathbf{0.164}_{\pm 0.007}$	$\mathbf{4.1}_{\pm 0.4}$	**1092**	$\mathbf{5.2}_{\pm 0.4}$	**1068**
TabPFN-CF	$0.635_{\pm 0.020}$	$9.5_{\pm 0.6}$	965	$0.691_{\pm 0.017}$	$9.6_{\pm 0.6}$	961	$0.180_{\pm 0.009}$	$7.7_{\pm 0.5}$	1006	$8.9_{\pm 0.5}$	977
TabPFNv3-CF	$0.624_{\pm 0.019}$	$10.3_{\pm 0.5}$	945	$0.679_{\pm 0.016}$	$10.8_{\pm 0.5}$	931	$0.193_{\pm 0.009}$	$9.9_{\pm 0.5}$	957	$10.3_{\pm 0.4}$	944
XGBoost-CF	$\mathbf{0.688}_{\pm 0.017}$	$5.8_{\pm 0.6}$	1054	$0.726_{\pm 0.017}$	$5.9_{\pm 0.5}$	1052	$0.166_{\pm 0.008}$	$4.4_{\pm 0.4}$	1089	$5.4_{\pm 0.4}$	1065
Multi-class Variants											
MITRA-MC	$0.626_{\pm 0.014}$	$10.3_{\pm 0.6}$	942	$0.685_{\pm 0.016}$	$9.3_{\pm 0.7}$	970	$0.289_{\pm 0.014}$	$12.3_{\pm 0.3}$	896	$10.6_{\pm 0.4}$	936
TabPFN-MC	$0.638_{\pm 0.018}$	$8.7_{\pm 0.7}$	983	$0.694_{\pm 0.019}$	$8.0_{\pm 0.7}$	999	$0.322_{\pm 0.019}$	$13.1_{\pm 0.4}$	874	$10.0_{\pm 0.5}$	952
XGBoost-MC	$0.634_{\pm 0.016}$	$10.0_{\pm 0.6}$	948	$0.679_{\pm 0.018}$	$9.6_{\pm 0.6}$	958	$0.305_{\pm 0.017}$	$12.7_{\pm 0.4}$	887	$10.8_{\pm 0.4}$	931
Discrete-time Hazard Variants											
MITRA-DH	$0.677_{\pm 0.018}$	$5.9_{\pm 0.4}$	1049	$0.728_{\pm 0.016}$	$5.8_{\pm 0.5}$	1055	$0.171_{\pm 0.007}$	$5.8_{\pm 0.4}$	1050	$5.8_{\pm 0.3}$	1051
TabPFN-DH	$0.657_{\pm 0.020}$	$5.8_{\pm 0.7}$	1054	$0.723_{\pm 0.015}$	$\mathbf{5.6}_{\pm 0.6}$	**1060**	$0.191_{\pm 0.012}$	$6.8_{\pm 0.7}$	1026	$6.1_{\pm 0.5}$	1047
TabPFNv3-DH	$0.650_{\pm 0.020}$	$7.1_{\pm 0.7}$	1025	$0.708_{\pm 0.016}$	$6.8_{\pm 0.7}$	1032	$0.227_{\pm 0.017}$	$9.1_{\pm 0.7}$	975	$7.6_{\pm 0.6}$	1011
XGBoost-DH	$0.687_{\pm 0.018}$	$6.2_{\pm 0.5}$	1043	$0.721_{\pm 0.017}$	$6.6_{\pm 0.5}$	1033	$0.167_{\pm 0.008}$	$4.7_{\pm 0.4}$	1078	$5.8_{\pm 0.4}$	1051
Classical & Neural Survival Baselines											
CPH	$0.682_{\pm 0.017}$	$5.9_{\pm 0.7}$	1049	$0.721_{\pm 0.016}$	$6.3_{\pm 0.7}$	1038	$0.169_{\pm 0.008}$	$5.3_{\pm 0.5}$	1065	$5.8_{\pm 0.6}$	1051
RSF	$0.666_{\pm 0.021}$	$7.6_{\pm 0.6}$	1010	$0.717_{\pm 0.017}$	$7.9_{\pm 0.6}$	1001	$0.167_{\pm 0.006}$	$5.7_{\pm 0.5}$	1054	$7.1_{\pm 0.5}$	1022
DeepHit	$0.585_{\pm 0.015}$	$12.5_{\pm 0.4}$	892	$0.612_{\pm 0.017}$	$12.7_{\pm 0.4}$	886	$0.211_{\pm 0.008}$	$10.6_{\pm 0.5}$	942	$11.9_{\pm 0.4}$	907
DeepSurv	$0.650_{\pm 0.017}$	$8.7_{\pm 0.5}$	985	$0.685_{\pm 0.017}$	$9.5_{\pm 0.6}$	966	$0.179_{\pm 0.008}$	$7.7_{\pm 0.4}$	1008	$8.6_{\pm 0.5}$	986