SURVICL: A FOUNDATION MODEL FOR IN-CONTEXT SURVIVAL ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Tabular foundation models (TFMs) now outperform gradient-boosted trees in classification and regression. Yet in survival analysis, which predicts when an event occurs from censored data, often on small clinical cohorts, we find that no existing TFM beats the half-century-old Cox model. We address this issue by introducing SurvICL, a TFM that predicts survival curves in-context. SurvICL is pretrained on purely synthetic datasets generated from a novel flexible proportional-odds prior that outperforms other prior types in our ablation study. Unlike most other survival models, SurvICL is trained with a plain quantile regression loss, which is possible because its synthetic pre-training data generator can provide uncensored event times. Additionally, we introduce the largest survival benchmark to date spanning 103 datasets. On the 63 datasets held out from all design decisions, SurvICL outperforms 20 tree-based, deep, and foundation-model baselines without any tuning: it wins 82.9% of pairwise comparisons in integrated Brier score, against 71.3% for the runner-up, the Cox model, and is the first model to rank above Cox on small datasets. Our code is available at https://anonymous.4open.science/r/survicl-FFC4.

1 INTRODUCTION

Many decisions hinge on *when* an event will happen: how long will a patient survive after a cancer diagnosis? When will a customer churn? When will a machine fail? Survival analysis answers such questions by predicting, for an individual with features $\boldsymbol{x} \in \mathbb{R}^d$, the *survival function* $S(t \mid \boldsymbol{x}) = \mathbb{P}(T^\star > t \mid \boldsymbol{x})$, that is, the probability that the event has not occurred by time t (Kleinbaum and Klein, 2012; Machin et al., 2006; Wiegrebe et al., 2024; Jeanselme et al., 2024). What sets survival analysis apart from regression, aside from strictly positive predictions, is *censoring*. A patient who leaves a clinical study alive after three years does not reveal their time of death, only that it exceeds three years. Discarding such partial observations biases predictions, and treating them as observed events is plainly wrong: a survival model must learn from both observed and censored events. In addition, survival predictions feed decisions (such as choosing a treatment given a five-year survival probability) and must therefore be *calibrated* probabilities rather than mere rankings of individuals (Goldstein et al., 2020; Lillelund et al., 2025; Alberge et al., 2026).

Survival analysis is often a small-data problem. Many time-to-event datasets come from clinical studies, where observing events requires following patients for years, and cohorts of a few hundred individuals are the norm (Indrayan and Mishra, 2021; O'Donnell et al., 2025): in all the data we collect for this work, more than 70% of the datasets have fewer than 1,000 samples. In this regime, the flexibility of modern machine learning is a liability. Half a century after its introduction, the Cox proportional hazards model (Cox, 1972) remains the workhorse of practitioners, and for good reason: on the small datasets of our benchmark, none of the gradient-boosted trees, deep survival networks, or existing survival foundation models outperforms it.

Tabular foundation models (TFMs) seem very well suited to this regime. Pre-trained on many synthetic datasets, models such as TabPFN and TabICL predict in-context and now outperform gradient-boosted trees for classification and regression, in particular on small datasets (Hollmann et al., 2023; 2025; Grinsztajn et al., 2025; Qu et al., 2025; 2026; Zhang et al., 2025). This recipe has been extended beyond standard supervised learning to causal effect estimation (Robertson et al., 2025; Balazadeh et al., 2025; Ma et al., 2025b). Survival analysis has not yet benefited from this progress.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

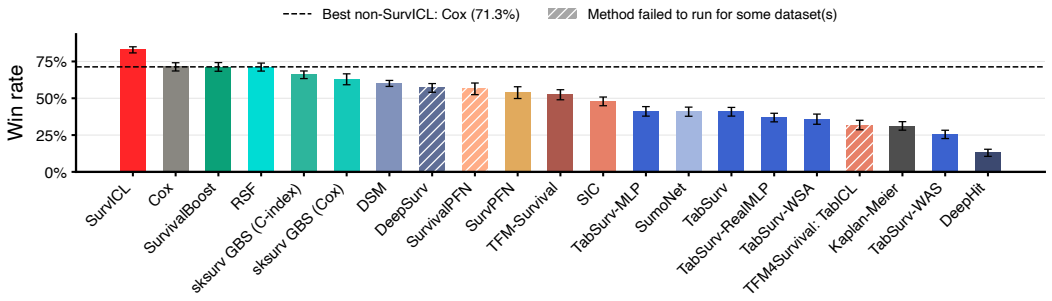


Figure 1: **SurvICL is the best survival model on 63 held-out datasets.** Pairwise win rate in integrated Brier score (IBS): the fraction of (dataset, competitor) pairs on which a method achieves a lower IBS. SurvICL wins 82.9% of its comparisons, against 71.3% for the runner-up, the classical Cox model. All existing survival TFMs trail behind the Cox model.

Recent survival TFMs either recast survival as classification over a time grid for off-the-shelf TFMs (Kim et al., 2026), or are pre-trained on synthetic survival data (Seletkov et al., 2026; Böhm et al., 2026; Qi et al., 2026) but do not beat the Cox model or random survival forests (Figure 1).

Building a High-Performing TFM for Survival Analysis. We introduce SurvICL, a tabular foundation model for in-context survival analysis, built on the TabICLv2 backbone (Qu et al., 2026). Similar to SurvivalPFN (Qi et al., 2026), it handles censoring in the data generation rather than the loss. During pre-training, SurvICL sees censored datasets in its context, but is supervised with the latent, uncensored event times of the simulator: it learns to correct for censoring in-context, so that a plain quantile-regression loss suffices. SurvICL predicts for each individual a continuous quantile function of the event time, from which we can directly infer a monotone survival function.

Scaling Evaluation of Survival Models. We evaluate SurvICL in the largest benchmark to date for survival analysis: 103 real-world datasets from healthcare, genomics, economics and engineering, with up to 52,000 samples and 15,000 features, against 20 baselines spanning classical, tree-based, deep and foundation models. For comparison, survival papers at machine-learning venues typically report results on 5 to 20 datasets, and the largest evaluations of survival foundation models span 48 to 81 datasets (Kim et al., 2026; Lyu et al., 2026; Pham et al., 2026a; Qi et al., 2026). All design choices were made on 40 development datasets, and all results are reported on 63 held-out datasets never used during development. On these held-out datasets, SurvICL is the best method both in prediction accuracy and in discrimination (Figure 1): it wins 82.9% of pairwise comparisons for the integrated Brier score (IBS), against 71.3% for the runner-up, the Cox model, and in 77.7% for the C-index, against 71.6% for random survival forests (see Figure 12). On small datasets, where no previous method surpasses the Cox model, SurvICL is the first to rank above it. On the largest datasets, gradient boosting still maintains a slight edge. Importantly, these gains do not come at the expense of computational efficiency. Across predictive metrics and dataset-size regimes, SurvICL lies on or close to the performance-runtime Pareto frontier (Appendix D). Thus, SurvICL combines strong predictive performance with efficient out-of-the-box inference. We commit to releasing code and model in an open-source license.

Contributions. Our contributions are threefold:

1. **We design a foundation model for survival analysis** that, using in-context learning, outperforms survival-analysis models (classical, tree-based, deep, all three existing survival foundation models as well as an adaptation of a classification foundation model), on integrated Brier score (IBS) and C-index.
2. **We identify key ingredients** of in-context survival analysis through extensive ablations: (i) a **proportional-odds (PO) prior** with a flexible, multi-modal baseline hazard, which outperforms Cox, accelerated failure time, and existing TFM priors; and (ii) **event-only targets**, rather than also learning the censoring distribution.
3. **An extensive benchmark** of 103 survival datasets with a strict separation between development and held-out datasets, covering high-dimensional genomics cohorts ($p > n$) and large registries, and comparing 21 methods on four metrics.

2 Related Work

Deep Learning for Survival Analysis. The toolbox of survival analysis has been considerably extended in the last years to include deep architectures (Katzman et al., 2018; Lee et al., 2018; Kvamme et al., 2019) that can accommodate a wide variety of data and event types, including time series, competing events, images and text (Zhu et al., 2016; Thorsen-Meyer et al., 2022; Bleistein et al., 2024; Wiegrebe et al., 2024; Alberge et al., 2024; Jeanselme et al., 2024). Our work builds on this research by tailoring deep transformers and training procedures for tabular data, which is the most common form of survival data.

Deep Learning for Tabular Data. Gradient-based decision tree models have long dominated benchmarks on tabular data (Gorishniy et al., 2021; Grinsztajn et al., 2022; Ye et al., 2024). Tabular foundation models have recently challenged their superiority and sparked a new line of research that focused on designing tailored architectures and training procedures (Hollmann et al., 2023; 2025; Qu et al., 2025; 2026; Kim et al., 2025; Ding et al., 2025; Zhang et al., 2025). We build on these architectural advances and adapt them to calibrated survival prediction. Once pre-trained, our method performs in-context inference without fine-tuning or weight updates.

Tabular Foundation Models for Survival Analysis is a burgeoning field of research. Efforts follow three distinct strategies. *(i) Reducing survival to a task that generic TFMs already solve.* Kim et al. (2026) frame survival prediction as a series of binary classification problems over a time grid, so that any TFM that handles classification can be deployed for survival analysis. Vieyra (2026) instead plugs an off-the-shelf TFM regressor into a log-normal accelerated failure time model and impute censored times iteratively, while SurvFM (Lyu et al., 2026) converts censored follow-up into pseudo-observations of the restricted mean survival time (RMST), which are then regressed in-context by unmodified TFMs. These approaches inherit the strengths of generic TFMs, but their backbone has never seen censored data: they either depend on a time discretization, impose a parametric form on the event-time distribution, or only output a summary statistic (the RMST) rather than a full survival function. *(ii) Adapting generic TFMs with a survival head.* Pham et al. (2026b;a) train classical survival heads (Cox, DeepHit, multi-task logistic regression) on top of frozen or fine-tuned representations of TabPFN, TabDPT (Ma et al., 2025a), and TabICL, and systematically compare these adaptation interfaces on 74 datasets. They find that zero-shot reformulations work best on small datasets, whereas trained heads take over as the sample size grows. This line of work matches the common practice of fitting shallow survival models on top of deep foundation models, and gives up the pre-training + in-context learning recipe: the head must be fitted, and possibly tuned, on each new dataset. *(iii) Pre-training on synthetic survival data.* Closest to our work, Seletkov et al. (2026), Böhm et al. (2026) and Qi et al. (2026) pre-train a tabular foundation model on synthetic survival data, following previous work in the field of tabular foundation models (Hollmann et al., 2023; Qu et al., 2025; 2026). All three models predict on a discretized time grid (see Table 1), while SurvICL outputs a continuous quantile function of the event time. Finally, foundation models for time-to-event prediction have also been pre-trained on large real-world electronic health records (Steinberg et al., 2024); such models are tied to a specific data modality and coding system, whereas we target generic tabular data through a synthetic prior.

3 Survival Analysis Settings: Censored Targets, Small Samples

3.1 Censored Data and Survival Models

Survival Data. In survival analysis, data comes in the form of i.i.d. samples $\mathcal{D}_n = \{(\boldsymbol{x}_i, t_i, \delta_i)\}_{i=1}^n$. For each instance i, $\boldsymbol{x}_i \in \mathbb{R}^d$ denotes the feature vector and $t_i > 0$ is a positive valued time. Unlike standard supervised learning, the target is subject to *right-censoring*, meaning that their true event time is not necessarily observed but might be censored, in which case we only observe the censoring time $c_i > 0$. This proxy time yields weak information on the true event time, since it will necessarily happen *after* censoring. We denote by $t_i^\star$ the possibly unobserved true time of event, such that

$$t_i := \min\{t_i^\star, c_i\} \quad \text{and} \quad \delta_i := \mathbb{1}_{t_i^\star \leq c_i}.$$

Predicting with Survival Models. Our goal is to approximate the survival function

$$S(t \,|\, \boldsymbol{x}_i) := \mathbb{P}(T^\star > t \,|\, X = \boldsymbol{x}_i)$$

at any time $t > 0$. This stands in contrast to classification or regression, where the model's goal is to output a value $\widehat{y}_i$ which approximates the binary or continuous outcome y_i. The survival function can be used to predict the time of event $\hat{t}_i$ for individual i as the first time it crosses the threshold 0.5.

Hazard-based Models. Following the seminal work of Cox (1972), an extensive family of models parametrized the hazard of the survival process, the time-wise derivative of the event probability:

$$\text{Hazard:} \qquad h(t \mid \boldsymbol{x}_i) := \lim_{\varepsilon \to 0} \frac{\mathbb{P}(T^\star \in [t, t+\varepsilon) \mid X = \boldsymbol{x}_i, T^\star \geq t)}{\varepsilon} \qquad = -\partial_t \log S(t \mid \boldsymbol{x}_i)$$

and trains it via (partial) likelihood maximization (Friedman, 1982; Fornili et al., 2013; Katzman et al., 2018; Kvamme et al., 2019; Kvamme and Borgan, 2021; Bleistein et al., 2024). Once the hazard is learned, it can be used to predict the survival function using numerical integration.

Discrete-Time Models. On the other side, some models bin time to simplify the computation of the model's likelihood and frame survival analysis as a classification task (Yu et al., 2011; Fotso, 2018; Lee et al., 2018; Kim et al., 2026). More precisely, in their approach, time is discretized into $B+1$ bins $([s_k, s_{k+1}[)_{k=0}^{B}$ for $0 = s_0 < \cdots < s_B = +\infty$. The tuple $(\boldsymbol{x}_i, t_i, \delta_i)$ is then expanded into a series of points by creating a series of binary outcomes $y_{i,k} := \mathbb{1}_{t_i \leq s_k}$ and considering the expanded dataset $\{((\boldsymbol{x}_i, s_k), y_{i,k})_k\}_{i=1}^{n}$. For individuals who experience the event, B points are created, while for censored individuals only the bins prior to the censoring time are added to the expanded dataset. This approach is the dominant approach in survival TFMs (see Table 1). While flexible and easy to train, unconstrained formulations of these models can output non-monotonic survival functions, requiring post-hoc workarounds such as clipping probabilities (Kim et al., 2026).

3.2 Design goals for a Survival TFM

Performance on Small Datasets. Small datasets are ubiquitous in survival analysis (O'Donnell et al., 2025), notably because time-to-event data is expensive and time-consuming to collect (to analyze event times that span over N years, one first needs to wait for N years) and because datasets often originate from healthcare (Indrayan and Mishra, 2021). State-of-the-art models perform well on large samples ($n \geq 2000$), but struggle on small datasets (Zhao et al., 2026).

Out-of-the-Box Calibration. In survival analysis, accurate probability estimates are crucial for decision-making (Goldstein et al., 2020; Alberge et al., 2026). Yet, models are often trained and evaluated chasing the C-index, which has been criticized as it does not assess calibration and is not a proper metric (not minimized for true probabilities, Lillelund et al., 2025). We train SurvICL to provide calibrated predictions without hyperparameter tuning.

4 SurvICL Architecture: Quantiles Supervised by Latent Times

The TabICLv2 Backbone. Our architecture inherits the major components of TabICLv2 (Qu et al., 2026). In particular, we use *repeated feature grouping*, which bags groups of features before tokenizing them and *target-aware embedding*. We adapt the original architecture of Qu et al. (2026) to handle triplets as inputs, which correspond to features, censoring indicators and observed times. More details on TabICLv2 and our modifications can be found in Appendix A.1.

Continuous Quantile Head. To obtain calibrated and accurate predictions, we use a multi-quantile pinball loss following TabICLv2. Given an input $\boldsymbol{x} \in \mathbb{R}^d$, our model outputs $\hat{\boldsymbol{q}}(\boldsymbol{x}) = (\hat{q}_\tau(\boldsymbol{x}))_{\tau \in \mathcal{T}} \in [0, +\infty)^{M-1}$, whose entries are the predicted quantiles of the event time at the levels $\mathcal{T} = \{1/M, 2/M, \ldots, (M-1)/M\}$. We then compute

$$\mathcal{L}(\hat{\boldsymbol{q}}(\boldsymbol{x}), t^\star) := \frac{1}{M-1} \sum_{\tau \in \mathcal{T}} \ell_\tau\big(\hat{q}_\tau(\boldsymbol{x}), t^\star\big) \text{ where } \ell_\tau(\hat{q}, t^\star) := \begin{cases} \tau\,(t^\star - \hat{q}) & \text{if } t^\star \geq \hat{q} \\ (1-\tau)\,(\hat{q} - t^\star) & \text{if } t^\star < \hat{q} \end{cases}$$

where $t^\star$ is the latent event time drawn from the prior (see below). We use $M = 1000$, i.e., 999 quantile levels as in TabICLv2. To ensure monotonicity, we sort the predicted quantiles except during pre-training. Experiments with other numbers of percentiles are reported in Figure 4. This loss is one of the major design choices that drive the capabilities of our model, and distinguishes it from previous models, which use classification losses or the hybrid loss of Lee et al. (2018).

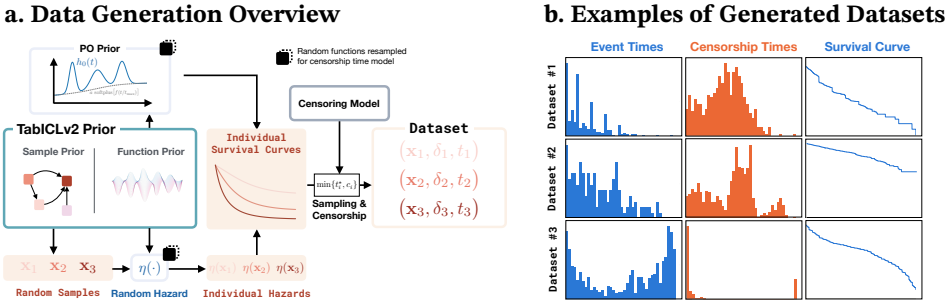


Figure 2: **(a)** Overview over our synthetic data prior: building on Qu et al. (2026), we construct a pipeline to simulate survival datasets by generating an event time and censoring time distribution. **(b)** Three example datasets drawn from our prior, which illustrate the high diversity in observed event time and censoring time distributions.

Latent Time Supervision. The target $t^\star$ of the pinball loss is never observed in real data: it is the latent event time generated by our prior (Section 5). During pre-training, the context only contains censored observations $(\boldsymbol{x}_i, t_i, \delta_i)$, while the loss is computed on the uncensored $t_i^\star$ of the query points. The model thus learns to infer the event-time distribution from censored data in-context. The survival-specific knowledge lives in the prior, so that the above regression loss is valid: we need neither a censored likelihood, nor inverse-probability-of-censoring weights, nor a discretization of time. SurvivalPFN (Qi et al., 2026) also supervises with latent times, but over a discretized time grid and with censoring times as an additional target, which we find to be detrimental (Section 6).

Mean-Duration Scaling. We rescale time, to cater for the wide variety of time range across datasets. Our strategy is to divide survival times by the mean observed duration.

Event-Only Targets. In survival analysis, censoring times are often considered to be hurdles to efficient estimation. Interestingly, Qi et al. (2026) use the censored times as an alternative target: their model learns to approximate two distributions (one for censored times and one for event times) by predicting the event and censoring times during pre-training. We ablate this mixed prediction task and show that it degrades the performance of our model.

5 Pretraining: Synthetic Data Priors with Proportional-Odds

A key novelty of our work is our synthetic data prior, illustrated in Figure 2.

Proportional-Odds (PO) Prior. We start by sampling numerical and categorical features $(\boldsymbol{x}_i)_{i=1}^n$ using the structural causal prior of Qu et al. (2026). The number of nodes (from 2 to 32) and features (from 1 to 100) span a wide range to allow for diverse datasets. We then sample survival times from a novel proportional-odds prior. Like the Cox model, it combines a baseline with an individual effect $\eta(\boldsymbol{x}_i)$, but on the odds rather than the hazard scale: hazard ratios then vary over time and may vanish in the long run, a common pattern in clinical data that Cox priors cannot generate. For the baseline hazard, we set

$$h_0(t) = a\ \mathrm{softplus}\big[f(t/t_{\max})\big] + \sum_{k=1}^{K} A_k \exp\left(-\frac{(t-\mu_k)^2}{2\sigma_k^2}\right),$$

with a, $(A_k)_{k=1}^K$, $(\mu_k)_{k=1}^K$ and $(\sigma_k)_{k=1}^K$ positive values, and $t_{\max} > 0$. The parameter K is taken randomly between 1 and 5. This parametrization adds a random number of positive Gaussian bumps at random times, each with its own width and height, directly on top of the smooth trend. We then integrate this baseline hazard to obtain $S_0(t) = \exp\big(-\int_0^t h_0(u)\,du\big)$. To generate the individual specific part of the hazard function, we apply a random function $\eta(\boldsymbol{x}_i)$ to the features as in Qu et al. (2026). We then define the individual-specific survival function as

$$S(t \mid \boldsymbol{x}_i) := \frac{1}{\frac{1-S_0(t)}{S_0(t)}\exp(\eta(\boldsymbol{x}_i)) + 1}.$$

Setting $\eta(\boldsymbol{x}_i) = 0$ recovers the baseline hazard, while positive values increase the odds of experiencing the event by any time. This prior draws on a long line of work on mixture models in survival analysis (McLachlan and McGiffin, 1994; De Angelis et al., 1999; Li et al., 2024). We sample event times by inverting the cumulative distribution function, with $U \sim \mathcal{U}([0,1])$:

$$t_i^\star = H_0^{-1}\Big(\log\left[1 + \exp(-\eta(\boldsymbol{x}_i))\,(U^{-1} - 1)\right]\Big)$$

with $H_0(t) := -\log S_0(t)$. We draw 5 times per individual to speed up convergence.

Generating Censoring Times. We add two sources of censoring to our model. First, in 50% of datasets, we apply a shared end-of-study cutoff to a randomly sampled set of individuals. Secondly, we generate a concurrent censoring time distribution by sampling another pair of baseline and individual specific hazards and combining them as for the event time distribution. We then sample from this second distribution, and set the observed time to be the minimum between the two. Lastly, to introduce tied times, we move some individual times to a common time grid with low probability.

Filtering. We remove datasets where more than 95% of samples are censored, as well as *non-predictable* datasets, adapting the filter of Qu et al. (2026): a dataset is non-predictable if predicting the mean survival time achieves a lower IBS (Section 6) than an ExtraTrees-based survival model (Geurts et al., 2006). Every pretraining dataset thus carries exploitable signal.

Training. During training, we split every generated dataset into a context (train) set $\mathcal{D}_c$ of size n_c and a query (test) set $\mathcal{D}_q$ of size n_ℓ with $n = n_c + n_\ell$. We then compute and take the gradients of

$$\frac{1}{n_\ell} \sum_{i \in \mathcal{D}_q} \mathcal{L}\big(f_\theta(\boldsymbol{x}_i, \mathcal{D}_c), t_i^\star\big),$$

where $f_\theta(\boldsymbol{x}_i, \mathcal{D}_c)$ denotes the quantiles predicted from the context $\mathcal{D}_c$; the loss is computed on the query set only. The context size varies during pre-training. We adopt the pre-training setup from Qu et al. (2026) and only use stage 1 for 200K steps, which takes around 72 hours on two H100 GPUs.

6 Large Scale Evaluation on 63 Datasets

6.1 Survival Metrics

We evaluate our models on four different metrics to overcome the pitfalls of single-metric evaluation (Lillelund et al., 2025). In the following, $N \geq 1$ denotes the size of the evaluation dataset. All metrics are computed using the `SurvEval` package (Qi et al., 2023)

Integrated Brier Score (IBS). The IBS is computed by integrating the Brier score at time $t > 0$, which is equal for an individual to the difference between the predicted survival curve and the observed binary survival indicator function, over time. It is hence equal to

$$\frac{1}{N_{test} t_{\max}} \sum_{i=1}^{N} \int_0^{t_{\max}} \mathrm{BS}_i(t)dt \quad \text{where} \quad \mathrm{BS}_i(t) := \frac{\widehat{S}(t \,|\, \boldsymbol{x}_i)^2 \mathbb{1}_{t_i \leq t, \delta_i = 1}}{\widehat{G}(t_i)} + \frac{(1 - \widehat{S}(t \,|\, \boldsymbol{x}_i))^2 \mathbb{1}_{t_i > t}}{\widehat{G}(t)}.$$

Unlike the C-index, the IBS measures the accuracy of predicted probabilities, not just rankings.

C-index at the 50th Quantile. We evaluate discrimination using an IPCW concordance index truncated at the median observed training follow-up time $t_{50} = \mathrm{Quantile}_{0.5}\{t_i^{\text{train}}\}$, where t_i^{train} includes both event and censoring times. For each test individual, we use the predicted risk $r_i = 1 - \widehat{S}(t_{50} \,|\, \boldsymbol{x}_i)$. Concordance is computed over comparable pairs with an observed event before t_{50} and weighted by $\widehat{G}(t_i)^{-2}$, where $\widehat{G}$ is the Kaplan–Meier estimate of the censoring survival function fitted on the training data. This corresponds to a truncated IPCW concordance estimator.

MAE-Hinge. We use absolute error for event times and a one-sided hinge error for censored times:

$$\mathrm{MAE}_{\mathrm{Hinge}} = \frac{1}{n} \sum_{i=1}^{n} \left[\delta_i |t_i - \hat{t}_i| + (1 - \delta_i) \max(0, t_i - \hat{t}_i)\right],$$

where δ_i indicates an observed event and $\hat{t}_i$ is the predicted median survival time. For censored individuals, only predictions earlier than the censoring time are penalized.

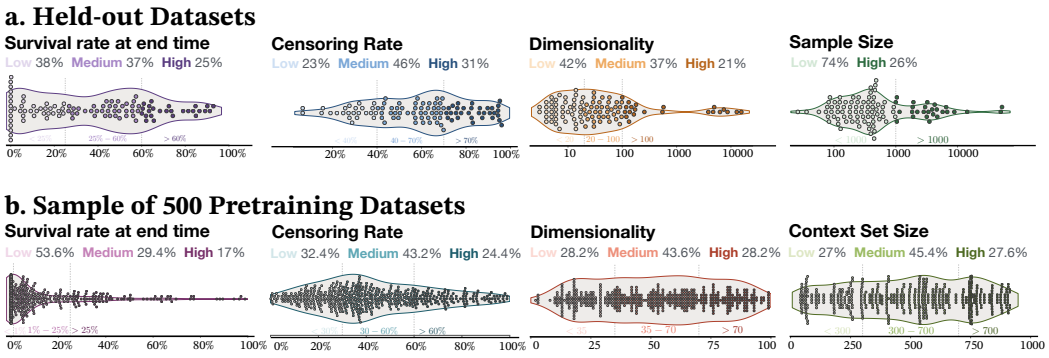


Figure 3: Overview of **(a)** the held-out datasets and **(b)** a sample of 500 pre-training datasets. In (b), the sample size is replaced by the context (train) set size (see Section 5).

KM-Calibration. We assess marginal calibration with the normalized integrated squared error, in $[0, 1]$, between the average predicted survival curve $\frac{1}{N_{test}} \sum_i \widehat{S}(t \mid \boldsymbol{x}_i)$ and the Kaplan–Meier curve of the test set (Chapfuwa et al., 2020).

6.2 Baselines: From Cox to Foundation Models

All models are described in more detail in Appendix B.1, we refer to Table 3 for a synthetic overview.

Shallow models. We include Cox model (Cox, 1972) and KM estimator (Kaplan and Meier, 1958) as baselines. The latter does not use covariates, and hence any model is expected to outperform it.

Deep Networks. We include five deep networks tailored for survival analysis (Lee et al., 2018; Katzman et al., 2018; Nagpal et al., 2021; Rindt et al., 2022; Kirpichenko et al., 2026). These models are fitted on each dataset and use various modeling techniques (from Bayesian prior to a MLP that predicts the survival function at different times).

Tree-Based Models. Following the seminal work of Ishwaran et al. (2008) on random survival forests (RSF) and Grinsztajn et al. (2022), a body of work has adapted tree-based methods to survival analysis. On top of RSF, we evaluate SurvivalBoost (Alberge et al., 2024) and alternative gradient-boosted tree-based methods implemented in `sksurv` (Pölsterl, 2020).

Tabular Foundation Models (TFMs). We include four open-weight tabular foundation models in our benchmark: SurvPFN (Böhm et al., 2026), SurvivalPFN (Qi et al., 2026), Survival-IC (SIC) (Seletkov et al., 2026) and TFM4Survival (Kim et al., 2026).

6.3 Datasets and Hold-Out Protocol

In total, we consider 103 datasets with varying properties, see Figure 3. The datasets are partitioned into 40 development datasets used during model development and 63 held-out datasets at random. For development, we source 2 PyCox (Kvamme et al., 2019), 25 SurvSet (Drysdale, 2022), one from a challenge and 12 cBioPortal (Cerami et al., 2012; Gao et al., 2013; De Bruijn et al., 2023) datasets. For held-out datasets, we use 6 from Pycox, 39 from SurvSet and 18 datasets from cBioPortal.

Protocol. All design choices of SurvICL (prior, target, and checkpoint) were made using the 40 development datasets; the 63 held-out datasets were kept for the final evaluation. We evaluated each method over ten event-stratified 80/20 train–test splits (test sets capped at 4,096 observations). Baseline hyperparameters were selected using a budget of up to 50 random search steps per dataset, minimizing the integrated Brier score on an event-stratified 80/20 split within the seed-0 training partition. The selected configuration was reused across all ten evaluation splits, with each estimator refitted on the corresponding training split. Methods without tunable hyperparameters were evaluated directly. Search spaces and method-specific budgets are detailed in Appendix B.1. Cox regression uses an ℓ_2 penalty to regularize estimation, including when $p > n$.

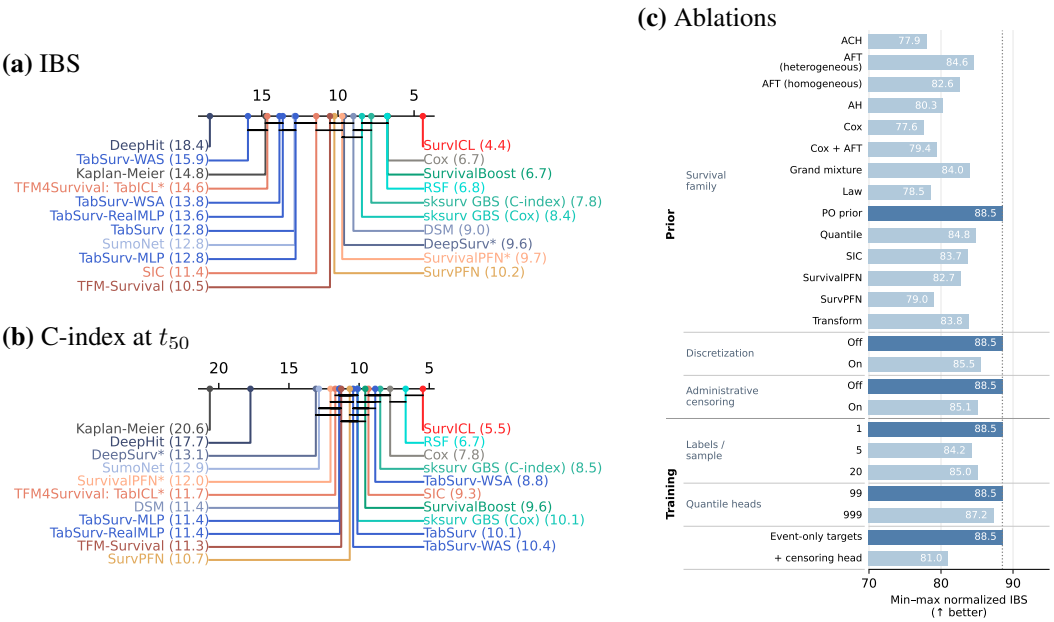


Figure 4: **(a, b)** Critical difference diagrams of average ranks on all held-out datasets (lower is better), for IBS and C-index at the median horizon t_{50}; methods joined by a bar are not significantly different. **(c)** Ablations of prior and training choices, by min–max normalized IBS (higher is better).

SurvICL uses each dataset's training split as context, with at most 4,096 labeled observations selected by event-stratified subsampling, without fine-tuning or dataset-specific hyperparameter tuning. Benchmark results for all methods and metrics are in Appendix D. Baseline runtimes measure fitting and survival-curve prediction, excluding hyperparameter search and metric computation.

6.4 Results: SurvICL Is Best Overall

SurvICL is the best survival model overall. On the 63 held-out datasets, SurvICL wins 82.9% of pairwise comparisons in IBS, against 71.3% for the runner-up, the Cox model (Figure 1), and the difference is statistically significant (Figure 4a, ranks and winrates are related by an affine transform). It also achieves the highest win rate in C-index, 77.7% against 71.6% for random survival forests (Figures 4b and 12). All existing survival TFMs trail behind classical methods. That the runners-up are the Cox model and random survival forests, with few parameters and strong inductive biases, underlines how hard survival analysis is for flexible models, which overfit small censored datasets.

SurvICL is the first model to rank above Cox on small datasets. 69.8% of our held-out datasets have fewer than 1,000 samples, making this regime central and not a niche. SurvICL is the first to rank above Cox in IBS (win rate 84.9% against 77.2%, Figure 23) and it matches Cox in C-index, although these differences are not statistically significant (Figures 22 and 26).

On large datasets, boosting remains strongest. On held-out datasets with more than 1,000 samples, SurvivalBoost is clearly the best method in IBS (Figure 37, 86.9% winrate). SurvICL ranks second in average IBS and first in C-index 50th quantile, ahead of TabSurv-WSA, RSF, and all other TFMs, although it wins fewer pairwise comparisons than on small datasets (Figure 38).

Performance-runtime trade-off. SurvICL lies on the Pareto frontier across C-index and IBS for all dataset-size regimes (Appendix D), and on the frontier for MAE in the small regime and all held-out datasets. SurvICL combines strong predictive performance with competitive runtime.

Calibration and time accuracy. SurvICL is among the best-calibrated methods: its average KM-calibration rank (6.0) trails only the KM estimator (4.3), the Cox model (5.0), DeepSurv (5.6) and sksurv GBS-Cox (5.9) (Figure 15). Both KM and the Breslow baseline hazard of the Cox model enforce calibration by construction. In MAE, SurvICL is close to Survival-IC and slightly behind SurvivalPFN, without statistically significant difference (Figure 18).

Model	Loss	Continuous Time	Prior	Prediction Target
SurvPFN	NLL + ranking	✓	Weibull	Observed times
SurvivalPFN	NLL	✗	Random MLP	Latent event + censoring times
Survival-IC	DeepHit Loss	✗	EH Model	Observed times
TFM4Survival	BCE	✗	✗	Binary survival labels
SurvICL (ours)	Pinball Loss	✓	PO prior	Latent event times

Table 1: Major differences between existing survival TFMs and SurvICL. "Latent" targets are uncensored times generated by the prior, unobserved in real data.

What makes SurvICL work. Figure 4c ablates design choices on development datasets with a shallower model trained for 10K steps, cf. Appendix C for more details. The prior has a substantial impact on performance: by keeping the exact same architecture and only changing the generation process, our *proportional-odds prior* outperforms Cox (77.6) and mixed priors (84.0). Compared to the prior of the other TFMs, our prior yields better results than SurvPFN (79.0), SurvivalPFN (82.7), and SIC (83.7). This demonstrates that our prior is highly effective, while adding a censoring-time prediction head actually reduces performance (81.0). Administrative censoring and time discretization have limited effects at this scale; we retain them to represent study-end censoring and finite temporal resolution. The final model uses 999 quantiles for finer resolution, despite a small advantage for 99 in the ablation (88.5 versus 87.2).

7 Discussion and Conclusion

Designing Efficient Survival TFMs. SurvICL differs from existing survival TFMs along three axes (Table 1). *Output and supervision.* Unlike previous survival TFMs that rely on discrete time bins such as TFM4Survival (Kim et al., 2026) and SurvivalPFN (Qi et al., 2026), which use classification or likelihood losses over time bins, SurvPFN (Böhm et al., 2026) uses a censored likelihood with a ranking term over 1,000 bins, and Survival-IC (Seletkov et al., 2026) uses the DeepHit loss (Lee et al., 2018) over 10 bins. SurvICL is the first to output a continuous quantile function. Quantile outputs are not new per se: TabICLv2 (Qu et al., 2026) uses a pinball loss for regression, and censored quantile regression (Portnoy, 2003; Pearce et al., 2022) modifies this loss to account for censoring. With latent-time supervision, no such modification is needed, as censoring is handled by the prior. SurvivalPFN also supervises with latent times, but over a time grid and with censoring times as an additional target, which our ablations show to be detrimental. *Prior.* SurvivalPFN draws times from random noised neural networks with several censoring mechanisms, SurvPFN draws Weibull times from a risk score and a dataset-specific shape, and Survival-IC mixes an extended hazard and a discrete-time model, with covariate-independent and administrative censoring. Our PO prior instead combines a flexible, multi-modal baseline hazard with a random risk function, for both event and censoring times; in our ablations, it outperforms the SurvPFN prior as well as Cox and AFT priors. *Backbone and evaluation.* Like Survival-IC, built on TabICL (Qu et al., 2025), we reuse an existing architecture, TabICLv2, which is not our contribution. Finally, we provide, to our knowledge, the first comparison of all four survival TFMs, on 63 datasets.

Implications. On the small cohorts that dominate clinical research, SurvICL is the first model to rank above the Cox model, the default of practitioners for half a century, without any training or tuning, while remaining among the best-calibrated methods. This makes foundation models a credible default for survival analysis.

Limitations and outlook. On the largest datasets, gradient boosting remains slightly stronger, calling for longer contexts or light fine-tuning. Baseline hyperparameters are tuned on the first split and reused across splits, as in Ye et al. (2024); Gorishniy et al. (2025); as tuning data then overlaps other splits' test sets, this protocol, if anything, favors the baselines. SurvICL addresses right-censored, single-event data; competing risks and time-varying covariates are natural extensions of the prior. We release our code and models to foster research on foundation models for time-to-event data.

AI USE STATEMENT

In this work, we used generative AI tools to generate synthetic data sets, design or provide feedback on research methodology or experiments, implement methods, draft preliminary plots and provide feedback on research methodology or experiments. We have not used generative AI tools to help develop theoretical models or conceptual frameworks, formulate mathematical claims, assist with translation, clean and reformat datasets, support qualitative and thematic data analysis, and interpret results. The categories provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses are not applicable to this work.

Additionally, we used generative AI tools to create or modify scientific figures or images, suggest experimental parameters, create or edit software code, and to brainstorm and source and search for information.

We have reviewed all AI-assisted work. The paper has been proofread by all authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To ensure full reproducibility of our methodology and empirical findings, we have provided a complete, anonymized codebase for this submission at: https://anonymous.4open.science/r/survicl-FFC4/ and the final checkpoint of the model can be found at https://figshare.com/s/7d9b0dab09cbb84a2da8.
We commit to releasing the codebase, dataset preprocessing files, and model checkpoints under an open-source license upon publication.

REFERENCES

Julie Alberge, Vincent Maladière, Olivier Grisel, Judith Abécassis, and Gaël Varoquaux. Survival models: Proper scoring rule and stochastic optimization with competing risks. *arXiv preprint arXiv:2410.16765*, 2024.

Julie Alberge, Tristan Haugomat, Gaël Varoquaux, and Judith Abécassis. On the calibration of survival models with competing risks. *arXiv preprint arXiv:2602.00194*, 2026.

Vahid Balazadeh, Hamidreza Kamkari, Valentin Thomas, Benson Li, Junwei Ma, Jesse C. Cresswell, and Rahul G. Krishnan. CausalPFN: Amortized causal effect estimation via in-context learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Linus Bleistein, Van-Tuan Nguyen, Adeline Fermanian, and Agathe Guilloux. Dynamical survival analysis with controlled latent states. *arXiv preprint arXiv:2401.17077*, 2024.

Samuel Böhm, Lennart Purucker, Frank Hutter, and Pascal Schlosser. Survpfn: Towards foundation models for survival predictions. *arXiv preprint arXiv:2606.04564*, 2026.

Ethan Cerami, Jianjiong Gao, Ugur Dogrusoz, Benjamin E Gross, Selcuk Onur Sumer, Bülent Arman Aksoy, Anders Jacobsen, Caitlin J Byrne, Michael L Heuer, Erik Larsson, et al. The cbio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer discovery*, 2(5):401–404, 2012.

Paidamoyo Chapfuwa, Chenyang Tao, Chunyuan Li, Irfan Khan, Karen J. Chandross, Michael J. Pencina, Lawrence Carin, and Ricardo Henao. Calibration and uncertainty in neural time-to-event modeling. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

David R Cox. Regression models and life-tables. *Journal of the royal statistical society: Series B (methodological)*, 34(2):187–202, 1972.

R De Angelis, R Capocaccia, T Hakulinen, B Soderman, and A Verdecchia. Mixture models for cancer survival analysis: application to population-based data with covariates. *Statistics in medicine*, 18(4):441–454, 1999.

Ino De Bruijn, Ritika Kundra, Brooke Mastrogiacomo, Thinh Ngoc Tran, Luke Sikina, Tali Mazor, Xiang Li, Angelica Ochoa, Gaofei Zhao, Bryan Lai, et al. Analysis and visualization of longitudinal genomic and clinical data from the aacr project genie biopharma collaborative in cbioportal. *Cancer research*, 83(23):3861–3867, 2023.

Jiayuan Ding, Jianhui Lin, Shiyu Jiang, Yixin Wang, Ziyang Miao, Zhaoyu Fang, Jiliang Tang, Min Li, and Xiaojie Qiu. Tabula: A tabular self-supervised foundation model for single-cell transcriptomics. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

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

Nick Erickson, Lennart Purucker, Andrej Tschalzev, David Holzmüller, Prateek Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. *Advances in Neural Information Processing Systems*, 38, 2025.

Marco Fornili, Federico Ambrogi, Patrizia Boracchi, and Elia Biganzoli. Piecewise exponential artificial neural networks (peann) for modeling hazard function with right censored data. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 125–136. Springer, 2013.

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

Michael Friedman. Piecewise exponential models for survival data with covariates. *The Annals of Statistics*, 10(1):101–113, 1982.

Jianjiong Gao, Bülent Arman Aksoy, Ugur Dogrusoz, Gideon Dresdner, Benjamin Gross, S Onur Sumer, Yichao Sun, Anders Jacobsen, Rileen Sinha, Erik Larsson, et al. Integrative analysis of complex cancer genomics and clinical profiles using the cbioportal. *Science signaling*, 6(269): pl1–pl1, 2013.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

Mark Goldstein, Xintian Han, Aahlad Puli, Adler Perotte, and Rajesh Ranganath. X-cal: Explicit calibration for survival analysis. *Advances in neural information processing systems*, 33:18296–18307, 2020.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. *Advances in neural information processing systems*, 34:18932–18943, 2021.

Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. TabM: Advancing tabular deep learning with parameter-efficient ensembling. In *International Conference on Learning Representations*, 2025.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35: 507–520, 2022.

Léo Grinsztajn, Klemens Flöge, Oscar Key, Felix Birkel, Philipp Jund, Brendan Roof, Benjamin Jäger, Dominik Safaric, Simone Alessi, Adrian Hayler, Mihir Manium, Rosen Yu, Felix Jablonski, Shi Bin Hoo, Anurag Garg, Jake Robertson, Magnus Bühler, Vladyslav Moroshan, Lennart Purucker, Clara Cornu, Lilly Charlotte Wehrhahn, Alessandro Bonetto, Bernhard Schölkopf, Sauraj Gambhir, Noah Hollmann, and Frank Hutter. Tabpfn-2.5: Advancing the state of the art in tabular foundation models, 2025. URL https://arxiv.org/abs/2511.08667.

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*, 2023.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 01 2025. doi: 10.1038/s41586-024-08328-6. URL https://www.nature.com/articles/s41586-024-08328-6.

A Indrayan and A Mishra. The importance of small samples in medical research. *Journal of Postgraduate Medicine*, 67(4):219–223, 2021.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *The Annals of Applied Statistics*, 2(3):841 – 860, 2008. doi: 10.1214/08-AOAS169. URL https://doi.org/10.1214/08-AOAS169.

Vincent Jeanselme, Nikita Agarwal, and Chen Wang. Review of language models for survival analysis. In *AAAI 2024 Spring Symposium on Clinical Foundation Models*, 2024.

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.

Da In Kim, Wei Siang Lai, and Kelly W Zhang. Tabular foundation models can do survival analysis. *arXiv preprint arXiv:2601.22259*, 2026.

Myung Jun Kim, Félix Lefebvre, Gaëtan Brison, Alexandre Perez-Lebel, and Gaël Varoquaux. Table foundation models: on knowledge pre-training for tabular learning. *arXiv preprint arXiv:2505.14415*, 2025.

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

David G. Kleinbaum and Mitchel Klein. *Survival Analysis: A Self-Learning Text*. Statistics for Biology and Health. Springer, New York, NY, 2012.

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

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.

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.

Yunzhe Li, Juhee Lee, and Athanasios Kottas. Bayesian nonparametric erlang mixture modeling for survival analysis. *Computational Statistics & Data Analysis*, 191:107874, 2024.

Christian Marius Lillelund, Shi-ang Qi, Russell Greiner, and Christian Fischer Pedersen. Stop chasing the c-index: This is how we should evaluate our survival models. *arXiv preprint arXiv:2506.02075*, 2025.

Yue Lyu, Steven H. Lin, Xuelin Huang, and Ziyi Li. SurvFM enables tabular foundation models for right-censored survival prediction. *arXiv preprint arXiv:2607.09577*, 2026.

Junwei Ma, Valentin Thomas, Rasa Hosseinzadeh, Alex Labach, Jesse Cresswell, Keyvan Golestan, Guangwei Yu, Anthony L Caterini, and Maks Volkovs. TabDPT: Scaling tabular foundation models on real data. In *Advances in Neural Information Processing Systems*, 2025a.

Yuchen Ma, Dennis Frauen, Emil Javurek, and Stefan Feuerriegel. Foundation models for causal inference via prior-data fitted networks. *arXiv preprint arXiv:2506.10914*, 2025b.

David Machin, Yin Bun Cheung, and Mahesh Parmar. *Survival analysis: a practical approach*. John Wiley & Sons, 2006.

Geoff J McLachlan and David C McGiffin. On the role of finite mixture models in survival analysis. *Statistical methods in medical research*, 3(3):211–226, 1994.

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

Autumn O'Donnell, Michael Cronin, Shirin Moghaddam, and Eric Wolsztynski. A systematic review on machine learning techniques for survival analysis in cancer. *Cancer Medicine*, 14(22): e71375, 2025.

Tim Pearce, Jong-Hyeon Jeong, Yichen Jia, and Jun Zhu. Censored quantile regression neural networks for distribution-free survival analysis. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Minh-Khoi Pham, Luca Cotugno, D. Cernei, Alina Sîrbu, S. Masi, G. Prencipe, A. Pingitore, P. Landi, Tai Tan Mai, Martin Crane, and Marija Bezbradica. Adaptation interfaces for in-context tabular foundation models in time-to-event prediction. *arXiv preprint arXiv:2609.04901*, 2026a.

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. In *International Conference on AI in Healthcare (AIiH)*, 2026b. arXiv:2606.12006.

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.

Stephen Portnoy. Censored regression quantiles. *Journal of the American Statistical Association*, 98(464):1001–1012, 2003.

Shi-ang Qi, Weijie Sun, and Russell Greiner. Survivaleval: A comprehensive open-source python package for evaluating individual survival distributions. In *Proceedings of the AAAI Symposium Series*, volume 2, pages 453–457, 2023.

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.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabicl: A tabular foundation model for in-context learning on large data. *arXiv preprint arXiv:2502.05564*, 2025.

Jingang Qu, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. Tabiclv2: A better, faster, scalable, and open tabular foundation model. *arXiv preprint arXiv:2602.11139*, 2026.

David Rindt, Robert Hu, David Steinsaltz, and Dino Sejdinovic. Survival regression with proper scoring rules and monotonic neural networks. In *International Conference on Artificial Intelligence and Statistics*, 2022.

Jake Robertson, Arik Reuter, Siyuan Guo, Noah Hollmann, Frank Hutter, and Bernhard Schölkopf. Do-pfn: In-context learning for causal effect estimation. *arXiv preprint arXiv:2506.06039*, 2025.

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.

Ethan Steinberg, Jason Fries, Yizhe Xu, and Nigam Shah. MOTOR: A time-to-event foundation model for structured medical records. In *International Conference on Learning Representations (ICLR)*, 2024.

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.

Mariana Vargas Vieyra. Staying alive: Uncensored survival analysis with tabular foundation models. *arXiv preprint arXiv:2606.03689*, 2026.

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.

Han-Jia Ye, Si-Yang Liu, Hao-Run Cai, Qi-Le Zhou, and De-Chuan Zhan. A closer look at deep learning methods on tabular datasets. *arXiv preprint arXiv:2407.00956*, 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. *arXiv preprint arXiv:2510.21204*, 2025.

Yonghao Zhao, Changtao Li, Chi Shu, Qingbin Wu, Hong Li, Chuan Xu, Tianrui Li, Ziqiang Wang, Zhipeng Luo, and Yazhou He. Tackling small sample survival analysis via transfer learning: A study of colorectal cancer prognosis. *Artificial Intelligence in Medicine*, page 103426, 2026.

Xinliang Zhu, Jiawen Yao, and Junzhou Huang. Deep convolutional neural network for survival analysis with pathological images. In *2016 IEEE international conference on bioinformatics and biomedicine (BIBM)*, pages 544–547. IEEE, 2016.

Appendices

Contents

A SurvICL model details 16
A.1 Architecture . . . 16
A.2 Pretraining . . . 17
A.3 Example of generated datasets . . . 17
B Experimental Details 18
B.1 Details on Baselines . . . 18
B.2 Hyperparameter tuning . . . 20
B.3 Issues for some methods . . . 22
B.4 Details on Datasets . . . 22
C Ablations 25
D Other Results 26
D.1 All held-out datasets . . . 26
D.2 Held-out datasets with less than 1000 samples . . . 34
D.3 Held-out datasets with more than 1000 samples . . . 42

A SURVICL MODEL DETAILS

A.1 ARCHITECTURE

We build SurvICL on top of TabICLv2 (Qu et al., 2026), which is at the time of writing the leading fully open-source model on the TabArena benchmark (Erickson et al., 2025). While refer to the original paper for the details, the key components of the TabICLv2 architecture can be summarized as follows:

- **Preprocessing**: First, some light preprocessing is applied, including ordinal encoding of categorical columns. After the preprocessing, the model receives a fully numerical table.
- **Target-aware embedding**: The original covariates $(x_{ij}) \in \mathbb{R}^{n\times m}$ are transformed into embeddings $(z_{ij}) \in \mathbb{R}^{n\times m\times 128}$ containing one embedding vector $z_{ij} \in \mathbb{R}^{128}$ per cell (i, j). Each embedding vector is obtained from an affine transformation applied to three cells in the same row ("repeated feature grouping"). An affine transformation of the target y_i is added for the context set rows.
- **Column-wise embedding**: Each column j of the resulting embeddings is processed independently using three induced self-attention blocks that allow interactions between different rows in the same column, resulting in a complexity of $O(nm)$, which is linear in the number of rows, unlike a full Transformer. Crucially, the attention is designed to only retrieve information from the training set, meaning that predictions on test samples are independent of the other test samples.
- **Row-wise interaction**: Next, three full transformer blocks allow different columns within a row to interact. Before these blocks, four extra columns with fixed values are appended, and after the blocks, the values of these four columns are concatenated and serve as row embeddings, while the other columns are discarded.
- **ICL transformer**: After adding another embedding of the targets, the embeddings of each row interact via 12 regular transformer layers, again making sure that test samples cannot access other test samples.
- **Head**: Finally, a multilayer perceptron (MLP) head transforms the output embeddings into 999 quantile predictions (for regression).

Overall, the TabICLv2 architecture has a runtime complexity of $O(n^2 + nm^2)$, and SurvICL preserves this complexity since it only modifies the $O(nm)$ preprocessing and embedding parts. In addition, like other TFMs, we use ensembling, where the predictions of eight forward passes with different preprocessings and column permutations are averaged.

Building on TabICLv2, we introduce the following adaptations to handle survival data:

- **Time normalization**: We normalize observed times by their mean over the context set, yielding $\tilde{t}_i = t_i/\bar{t}$, where $\bar{t}_{\text{context}}$ is the mean duration of the observed times. This makes the time representation invariant to the choice of measurement units. During pre-training, both context and query sets are normalized and the mean $\bar{t}$ is taken across context and query set. During inference (after pre-training), the mean for $\bar{t}$ is taken across the context set, and the predicted quantiles are transformed back to the original time scale.
- **Censoring-aware target embedding**: We replace the scalar target embedding with a joint representation of the observed time and event indicator. Specifically, we sum the embedding of the continuous time $\tilde{t}_i$ and a learned embedding of the binary indicator δ_i. This representation is added to every cell embedding in context row i, allowing the column-wise encoder to distinguish observed events from censored observations.
- **Survival information at the ICL stage**: We also encode $(\tilde{t}_i, \delta_i)$ using a separate pair of time and event-indicator embeddings whose sum matches the dimension of the row representation. This embedding is added to context row representations before the ICL transformer. At both embedding stages, survival outcomes are supplied only for context rows.
- **Survival prediction**: The MLP head predicts quantiles of the conditional event-time distribution, using a softplus transformation to ensure that the quantiles are positive. After

restoring the original time scale and forcing the predicted quantiles to be monotonous by sorting them, we obtain an estimated cumulative distribution function by linear interpolation between quantile levels. The survival function is then given by

$$\widehat{S}(t \mid x) = 1 - \widehat{F}(t \mid x).$$

Table 2: Backbone architecture and prior for SurvICL.

Backbone	
Column-wise transformer	3 blocks, 8 heads, 128 inducing points, QSSMax attention
Row-wise transformer	3 blocks, 8 heads, 4 CLS tokens
In-context (ICL) transformer	12 blocks, 8 heads, QSSMax attention
Max. sequence length (rows)	1024
Max. features per dataset	100 (min. 1)
Regression head	Pinball / quantile, 999 quantile levels
Target embedding	Sum
Duration rescaling	Mean-duration
Label draws per row	5
Training steps	200,000
Learning rate	0.0008
Synthetic prior	
Prior generator	GraphSCM (`graph_scm_survival`)
Structural family	Proportional odds (`generation_type po`)
Survival parameter range	Wide
Loss	Pinball loss on death-time quantiles

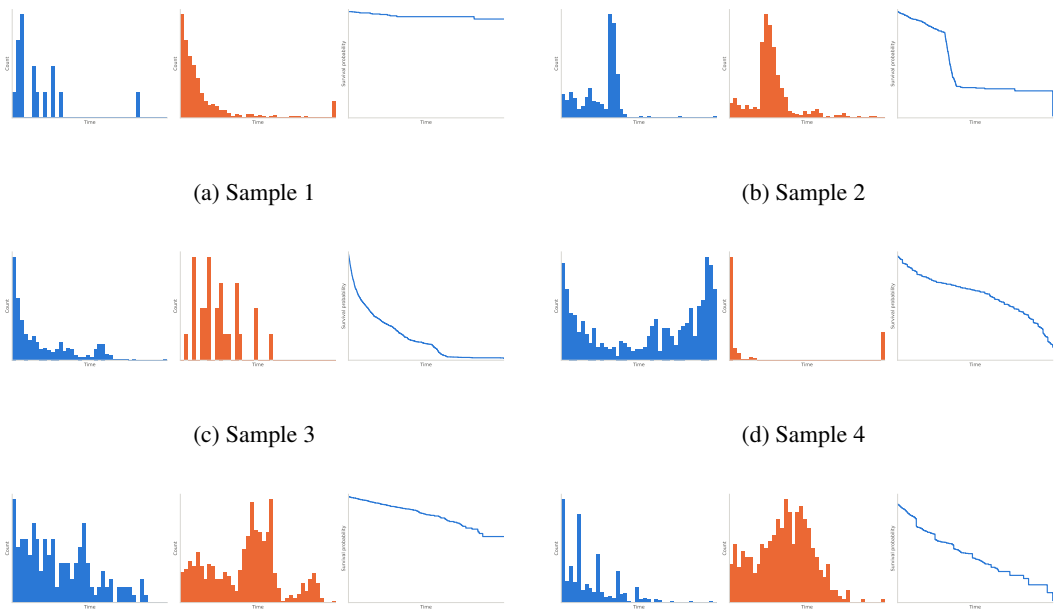
A.2 Pretraining

We follow many of the pretraining settings from TabICLv2 but make the following modifications:

- We omit the second and third stages of pretraining, which scale to larger datasets, since the second stage did not show improvements on our benchmark.
- For computational reasons, we only train for 200K steps instead of 500K. Moreover, our pretraining consisted of two pretrainings for 100K steps each, meaning that the learning-rate schedule consisted of two 100K-steps schedules.
- We generate pretraining datasets with 1024 samples each, of which a random fraction in $[0.03, 0.9]$ is used as context (train) and the rest as query (test) samples. We lowered the lower bound from 0.3 to 0.03 to cover the smaller datasets found in our benchmark.
- We pretrain fully in 32-bit floats to avoid potential instabilities.

A.3 Example of generated datasets

Figure 5 presents a diverse set of examples sampled from our final Proportional Odds (PO) prior. These datasets incorporate both administrative censoring and time discretization. In each subfigure: left and middle panels are histograms showing the observed events and censoring events over time while the right panel is the resulting Kaplan-Meier survival probability curve.



(a) Sample 1 (b) Sample 2

(c) Sample 3 (d) Sample 4

(e) Sample 5 (f) Sample 6

Figure 5: Six generated datasets using our final PO prior, incorporating administrative censoring and time discretization. For each sample, we show the histogram of observed event (left), the censoring events (center), and the Kaplan-Meier curve (right).

B Experimental Details

B.1 Details on Baselines

B.1.1 Usual methods

Those two methods are the most used in the medical setting.

Kaplan-Meier Estimator (Kaplan and Meier, 1958): A non-parametric statistic used to estimate the marginal survival probability from observed lifetime data. Because it does not take individual covariates into account, it serves as the most basic population-level baseline for survival functions and is consistent.

Cox Model (Cox Proportional Hazards) (Cox, 1972): The standard semiparametric workhorse of survival analysis. It models the hazard rate as a product of a time-varying baseline hazard and an exponential linear combination of the covariates. It relies on the strong assumption of proportional hazards (the hazard ratio between any two individuals is constant over time).

Type	Model Name	Reference
TFM	**SurvICL**	This work
	SurvPFN	Böhm et al. 2026
	SurvivalPFN	Qi et al. 2026
	Survival-IC	Seletkov et al. 2026
	TFM4Survival	Kim et al. 2026
Tree Based	**SurvivalBoost**	Alberge et al. 2024
	GBC	Pölsterl 2020
	RSF	Ishwaran et al. 2008
Deep Neural Net	TabSurv	Kirpichenko et al. 2026
	SumoNet	Rindt et al. 2022
	DSM	Nagpal et al. 2021
	DeepSurv	Katzman et al. 2018
	DeepHit	Lee et al. 2018
Shallow Model	Cox	Cox 1972
	Kaplan-Meier	Kaplan and Meier 1958

Table 3: Overview of methods we consider in our experimental section.

B.1.2 Tree-Based Models

These models adapt decision trees and boosting trees to handle right-censored data, remaining highly competitive on tabular datasets.

RSF (Random Survival Forests) (Ishwaran et al., 2008): An extension of the Random Forest algorithm for time-to-event data. It constructs an ensemble of survival trees using specialized splitting criteria (like the log-rank test) to handle censored observations.

GBS (Gradient Boosting for Survival) (Pölsterl, 2020): A boosting approach implemented in `scikit-survival` that sequentially builds regression trees. In our evaluation, we include two variants corresponding to different optimization targets: **sksurv GBS (Cox)**, which minimizes the negative log partial likelihood of the Cox proportional hazards model, and **sksurv GBS (C-index)**, which directly optimizes a smoothed concordance index.

SurvivalBoost (Alberge et al., 2024): A modern, highly optimized gradient boosting framework specifically tailored for survival analysis, optimizing a proper scoring rule.

B.1.3 Deep Neural Networks

These methods leverage multi-layer architectures to capture complex, non-linear relationships between covariates and survival times, often dropping the proportional hazards assumption.

DeepHit (Lee et al., 2018): A discrete-time survival network that optimizes a loss that combines a likelihood and a ranking metric over an MLP, usable in survival analysis and competing events.

DeepSurv (Katzman et al., 2018): It uses a Multi-Layer Perceptron (MLP) to replace the linear combination of covariates in the standard Cox model, allowing it to learn complex, non-linear interactions while still optimizing the Cox partial likelihood.

DSM (Deep Survival Machines) (Nagpal et al., 2021): DSM uses a neural network to model the survival function as a mixture of parametric distributions (such as Weibull or Log-Normal), providing well-calibrated survival estimates.

SumoNet (Rindt et al., 2022): SurvivalMonotonic-net utilizes a partially monotonic neural network to directly optimize the right-censored log-likelihood.

TabSurv (Kirpichenko et al., 2026): A specialized deep learning framework that adapts modern tabular neural networks to time-to-event prediction. Because this framework supports multiple representation backbones, we evaluate several architectural variants that appear throughout our results: the default **TabSurv**, **TabSurv-MLP** (using a standard Multi-Layer Perceptron), **TabSurv-RealMLP** (utilizing the more robust RealMLP tabular backbone), as well as **TabSurv-WAS** and **TabSurv-WSA** (which incorporate specialized attention and sparse embedding mechanisms designed specifically for tabular features).

B.1.4 Tabular Foundation Models

This category represents the most modern class of baselines: tabular foundation models trained on synthetic tabular data. They generalize to new datasets in a zero-shot or in-context manner.

SurvPFN: (Böhm et al., 2026) SurvPFN is a survival-specific Prior-Data Fitted Network (PFN) based on the architecture of TabPFN v2, pretrained on synthetic censored survival tasks (sampling from different distributions such as Weibull).

SurvivalPFN (Qi et al., 2026): SurvivalPFN performs amortized Bayesian survival inference by using the observed covariates, event times, and censoring indicators as in-context examples to directly predict posterior event-time distributions for new individuals based on the CausalPFN architecture.

Survival-IC (Seletkov et al., 2026): Survival In-Context (SIC) is a prior-fitted foundation model pretrained on synthetic survival datasets to approximate Bayesian posterior predictive inference and based on the architecture of TabICL.

TFM4Survival (Kim et al., 2026): TFM4Survival adapts pretrained tabular foundation models such as TabICL to censored time-to-event data through a sampling-based formulation, allowing existing TFMs to perform survival prediction without survival-specific pretraining.

B.2 HYPERPARAMETER TUNING

Table 4 shows a description of the hyperparameter search spaces for each method. These spaces are used with a budget of 50 iterations.

Table 4: Hyper-parameter search space for every baseline in our survival benchmark. Each cell is searched with `RandomizedSearchCV` for 50 iterations on a single stratified 80/20 split of the seed-0 training fold, scored by the integrated Brier score (IBS). `randint`/`loguniform` denote the `scipy.stats` distributions; a list denotes a categorical choice; a lone value is held fixed (not searched).

model	parameter	range
	Others	
Kaplan–Meier	no hyperparameters (population-level estimator)	
Cox (lifelines, ridge-penalized)	penalizer	`loguniform`(1e-04, 10)
	Tree-based methods	
Random Survival Forest	max features	sqrt, log2, 0.3, 0.5, 1.0
	min samples leaf	`randint`(1, 30)
	min samples split	`randint`(2, 20)
	n estimators	`randint`(100, 500)
sksurv GBS (Cox)	dropout rate	0.0, 0.1, 0.2
	learning rate	`loguniform`(1e-03, 0.3)
	loss	coxph
	n estimators	`randint`(50, 500)
	subsample	0.7, 0.85, 1.0
sksurv GBS (C-index)	dropout rate	0.0, 0.1, 0.2
	learning rate	`loguniform`(1e-03, 0.3)
	loss	coxph
	max depth	`randint`(2, 8)
	min samples leaf	`randint`(1, 20)
	n estimators	`randint`(50, 400)
	subsample	0.7, 0.85, 1.0
SurvivalBoost	learning rate	`loguniform`(0.01, 0.1)
	max depth	`randint`(2, 10)
	n horizons per observation	`randint`(1, 5)
	n iter	`randint`(5, 50)
	n iter before feedback	`randint`(5, 50)
	Neural network methods	
DeepHit	alpha	0.2, 0.5, 0.9
	batch size	128, 256, 512
	epochs	`randint`(256, 1024)
	lr	`loguniform`(1e-05, 0.001)
	nb output times	40, 60, 80, 100
	num nodes shared	[128, 128], [256, 256], [64, 64]
	sigma	0.1, 0.5, 0.9
	weight decay	`loguniform`(1e-05, 0.001)
DeepSurv	activation	ReLU
	batch norm	True, False
	batch size	128, 256, 512
	dropout	0.0, 0.1, 0.2, 0.4
	early stopping	True (patience 20)
	hidden dim	16, 32, 64, 128
	lr	`loguniform`(1e-05, 0.001)
	n epochs	64, 128, 256, 512
	n layers	1, 2, 3
	output bias	False

model	parameter	range
	weight decay	loguniform(1e-05, 0.1)
Deep Survival Machines	batch size	128, 256, 512
	distribution	Weibull, LogNormal
	iters	randint(50, 200)
	k	randint(2, 7)
	layers	[100], [100, 100], [50, 50], [100, 100, 100]
	learning rate	loguniform(1e-04, 0.01)
	temperature	0.5, 1.0, 2.0
SuMo-Net	batch size	128, 256, 512
	dropout	0.0, 0.1, 0.25
	layers	[64, 64], [100, 100], [32, 32, 32]
	layers surv	[64], [100], [100, 100]
	lr	loguniform(1e-04, 0.01)
	n iter	randint(200, 1500)
	weight decay	loguniform(1e-05, 0.01)
TabSurv-LAS	activation	ReLU, SiLU, SELU
	batch size	32, 64, 96
	d block	128, 256, 512
	dropout	loguniform(0.01, 0.1)
	emb dim rtdl	8, 12, 16
	k	8, 16, 32
	llh radius	1, 2, 3, 4, 5
	lr	loguniform(1e-04, 0.005)
	n blocks	randint(1, 4)
	rtdl activation	True, False
	rtdl n bins	32, 48, 64
TabSurv-MLP	activation	ReLU, SiLU, SELU
	batch size	32, 64, 96
	emb dim rtdl	8, 12, 16
	hidden dim	128, 256, 512
	layer n	randint(1, 5)
	llh radius	1, 2, 3, 4, 5
	lr	loguniform(1e-04, 0.005)
	normalization	False, True
	rtdl activation	True, False
	rtdl n bins	32, 48, 64
TabSurv-RealMLP	activation	Mish, SELU
	batch size	32, 64, 96
	emb dim rtdl	8, 12, 16
	frequency scale	loguniform(0.01, 1.0)
	hidden dim	128, 256, 512
	llh radius	1, 2, 3, 4, 5
	lr	loguniform(1e-04, 0.005)
	n frequencies	32, 48, 64
	n hidden layers	randint(1, 5)
TabSurv-WAS	activation	ReLU, SELU
	batch size	32, 64, 96
	d block	64, 128, 256
	emb dim rtdl	8, 12, 16
	k	8, 16, 32
	llh radius	1, 3, 5
	lr	loguniform(1e-04, 0.005)
	n blocks	randint(1, 4)
	rtdl activation	True, False
	rtdl n bins	32, 48, 64
TabSurv-WSA	activation	ReLU, SELU
	batch size	32, 64, 96
	d block	128, 256, 512
	emb dim rtdl	8, 12, 16

model	parameter	range
	k	8, 16, 32
	llh radius	1, 3, 5
	lr	`loguniform`(1e-04, 0.005)
	n blocks	`randint`(1, 4)
	rtdl activation	True, False
	rtdl n bins	32, 48, 64
SurvICL and tabular foundation models		
SIC	no hyperparameters (frozen DeepHit10 checkpoint)	
SurvivalPFN	calibrate	False, True
	max context size	1024, 2048, 4096, 8192
SurvPFN	max context size	1024, 2048, 4096
	max features	5, 10
TFM-Survival	max train rows	4096, 8192
	n intervals	`randint`(5, 20)
Surv. Classif.: TabICL	n intervals	`randint`(5, 40)

B.3 Issues for some methods

Some baselines failed to complete their hyperparameter search or final fit on a small subset of datasets, for reasons intrinsic to the method rather than to the benchmark protocol (e.g., a training run that diverged numerically, or a step of the method's own pipeline that is only defined for datasets larger than the fold at hand). We assign the method a trivially achievable score on each dataset on which it did not complete: an IBS of 1 (the maximum attainable value in expectation if G is estimated perfectly) or a concordance index of 0.5 (no better than chance), matching the metric being ranked. All critical-difference diagrams and pairwise win-rate figures are therefore always computed over the full 63-dataset held-out cohort. Methods with at least one imputed dataset are marked with an asterisk in the critical-difference diagrams and with a hatched bar in the win-rate figures; Table 5 lists the affected (method, dataset) pairs and their cause.

Table 5: Baseline (method, dataset) pairs that created issues

Method	**Dataset(s)**	**Cause**
DeepSurv	*divorce*, *actg*, *FRTCS*	Training loss diverged (became non-finite) during optimization on these three small datasets.
TFM4Surival: TabICL	*sac3*, *DLBCL*, *gse1992*	Out-of-memory issue on these larger/high-dimensional datasets.
SurvivalPFN	*AML_Bull*, *gse1992*, *vdv*	The method's internal dimensionality-reduction step (a truncated SVD) requires more components than the smaller of the fold's row or feature count provides, and errors when that bound is violated.
TabSurv	*DBCD*, *cesc tcga, kirc tcga, lgg tcga*	The method's obtained a MAE of value in-fty.

B.4 Details on Datasets

An overview of the development and held-out datasets is shown in Tables 6 and 7.

name	source	n samples	n features	censoring frac	km survival at tmax
Framingham	survset	4699	18	68.7%	60.5 %
MCLcleaned	survset	92	574	30.4 %	0.0 %
Melanoma	survset	205	5	72.2 %	64.5 %
NSBCD	survset	115	549	67.0 %	57.4 %
Rossi	survset	432	29	73.6 %	73.6 %
Unemployment	survset	452	8	43.4 %	5.8 %
acath	survset	3504	3	33.4 %	0.0 %
aids	survset	467	5	59.7 %	49.2 %
brca tcga pan can atlas 2018	cbioportal	980	137	95.2 %	60.1 %
cgd	survset	128	25	65.6 %	47.6 %
challenge ens	custom	3173	42	49.6 %	5.9 %
chop	survset	414	3833	60.1 %	39.5 %
cost	survset	518	13	22.0 %	22.0 %
crc orion 2024	cbioportal	74	131	66.2 %	30.4 %
csl	survset	446	6	39.5 %	17.3 %
d.oropha.rec	survset	192	30	27.6 %	16.5 %
dataDIVAT3	survset	4267	17	94.4 %	84.7 %
dataOvarian1	survset	912	163	40.4 %	11.9 %
epileptic	survset	605	5	63.5 %	46.3 %
flchain	pycox	6524	8	69.9 %	65.6 %
gse3143	survset	158	8664	68.4 %	41.9 %
kirc tcga pan can atlas 2018	cbioportal	369	53	92.7 %	86.5 %
kirp tcga pan can atlas 2018	cbioportal	251	56	96.0 %	92.9 %
luad tcga pan can atlas 2018	cbioportal	392	98	82.4 %	68.4 %
lusc tcga pan can atlas 2018	cbioportal	481	100	57.2 %	0.0 %
micro.censure	survset	117	118	77.8 %	42.4 %
msk 2026	cbioportal	40755	56	57.0 %	31.7 %
oldmort	survset	4603	17	57.2 %	0.2 %
ov tcga pan can atlas 2018	cbioportal	427	34	48.2 %	6.5 %
ova	survset	358	13	25.7 %	22.2 %
ovarian	survset	26	4	53.8 %	49.7 %
pancan pcawg 2020	cbioportal	361	255	62.9 %	0.0 %
rdata	survset	1040	7	47.4 %	38.1 %
sarc tcga pan can atlas 2018	cbioportal	181	147	87.3 %	74.3 %
smarto	pycox	3873	39	88.1 %	72.5 %
ucs tcga pan can atlas 2018	cbioportal	57	11	38.6 %	9.2 %
uvm tcga pan can atlas 2018	cbioportal	80	6	71.2 %	38.5 %
vlbw	survset	617	55	82.7 %	0.0 %
whas500	survset	461	18	61.8 %	0.0 %
zinc	survset	431	23	81.2 %	79.0 %

Table 6: Overview of development datasets.

name	source	n samples	n features	censoring frac	km survival at tmax
AML Bull	survset	116	6283	42.2 %	28.3 %
Aids2	survset	2839	14	38.0 %	5.8 %
Bergamaschi	survset	82	10	65.9 %	58.3 %
DBCD	survset	295	4919	73.2 %	61.0 %
DLBCL	survset	240	7399	42.5 %	26.8 %
Dialysis	survset	6805	37	76.4 %	58.7 %
FRTCS	survset	697	5	89.7 %	89.1 %
LeukSurv	survset	1043	30	15.7 %	4.4 %
Pbc3	survset	349	21	82.5 %	63.4 %
TRACE	survset	1878	6	49.0 %	43.9 %
UnempDur	survset	3241	6	38.7 %	10.5 %
acc tcga pan can atlas 2018	cbioportal	91	39	63.7 %	49.9 %
actg	survset	1151	20	91.7 %	90.1 %
blca tcga pan can atlas 2018	cbioportal	301	84	76.1 %	62.0 %
breast	survset	100	4	74.0 %	71.4 %
burn	survset	154	14	68.8 %	48.0 %
cancer	survset	228	30	27.6 %	5.0 %
cesc tcga pan can atlas 2018	cbioportal	241	97	95.0 %	91.3 %
chol tcga pan can atlas 2018	cbioportal	36	53	50.0 %	19.5 %
coadread tcga pan can atlas 2018	cbioportal	543	133	86.2 %	62.1 %
colon	survset	929	14	51.3 %	45.5 %
crc hta8 htan 2024	cbioportal	83	91	69.9 %	24.4 %
dataDIVAT2	survset	1837	4	68.3 %	31.1 %
diabetes	survset	394	4	60.7 %	53.0 %
difg glass	cbioportal	661	179	24.8 %	0.0 %
divorce	survset	3371	5	69.4 %	55.7 %
e1684	survset	284	3	31.0 %	28.3 %
esca tcga pan can atlas 2018	cbioportal	182	90	58.2 %	10.7 %
gbm tcga pan can atlas 2018	cbioportal	277	30	31.8 %	0.0 %
gbsg	pycox	2232	7	43.2 %	36.9 %
grace	survset	1000	5	67.6 %	63.5 %
gse1992	survset	124	15545	71.8 %	45.4 %
hdfail	pycox	52422	38	94.5 %	56.1 %
heart	survset	103	4	27.2 %	15.2 %
heartvalve	survset	256	27	78.9 %	53.0 %
hepatoCellular	survset	227	43	57.3 %	51.2 %
hnsc tcga pan can atlas 2018	cbioportal	364	119	83.0 %	44.6 %
lgg tcga pan can atlas 2018	cbioportal	422	63	91.9 %	43.3 %
lihc tcga pan can atlas 2018	cbioportal	371	70	64.7 %	18.2 %
meso tcga pan can atlas 2018	cbioportal	86	58	15.1 %	0.0 %
metabric	pycox	1904	9	42.1 %	0.0 %
mgus	survset	241	13	6.6 %	1.8 %
msk 2021	cbioportal	25659	86	60.4 %	34.3 %
nki70	survset	144	77	66.7 %	48.0 %
nwtco	pycox	4028	6	85.8 %	84.9 %
paad tcga pan can atlas 2018	cbioportal	184	75	46.2 %	13.7 %
pbc	survset	312	6	59.9 %	34.1 %
pharmacoSmoking	survset	125	19	28.8 %	28.8 %
phpl04K8a	survset	442	21	46.6 %	0.0 %
prostate	survset	502	28	29.5 %	23.8 %
prostateSurvival	survset	14294	8	94.4 %	81.4 %
rhc	survset	5735	79	35.1 %	0.0 %
sac3	pycox	50000	45	37.3 %	20.2 %
scania	survset	1931	9	43.8 %	15.9 %
skcm tcga pan can atlas 2018	cbioportal	433	172	50.3 %	5.7 %
stad tcga pan can atlas 2018	cbioportal	355	112	75.5 %	64.4 %
stagec	survset	146	22	63.0 %	55.8 %
support	pycox	8873	14	32.0 %	24.1 %
ucec tcga pan can atlas 2018	cbioportal	486	58	90.7 %	83.6 %
uis	survset	628	17	19.1 %	16.6 %
vdv	survset	78	4705	56.4 %	56.4 %
veteran	survset	137	9	6.6 %	0.0 %
wpbc	survset	198	32	76.3 %	65.3 %

Table 7: Overview of held-out datasets.

C ABLATIONS

To isolate the impact of specific design choices, we conduct ablation studies using a shared `nanotabiclexsurvival` backbone trained for 10,000 steps. Each ablation alters exactly one variable against a control arm trained on the Proportional Odds (`po`) prior as detailed in the following table. One arm can be trained in 2h on 2 A100.

Table 8: Backbone architecture shared by every arm.

Backbone (`nanotabiclexsurvival`)	
Column-wise transformer	2 blocks, 4 heads, 128 inducing points, SSMax attention
Row-wise transformer	2 blocks, 4 heads, 4 CLS tokens
In-context (ICL) transformer	4 blocks, 4 heads, SSMax attention
Regression head	Pinball / quantile, 99 quantile levels
Target embedding	Sum
Duration rescaling (control)	Mean-duration
Label draws per row (control)	1
Learning rate	0.0015
Synthetic prior	
Prior generator	GraphSCM (`graph_scm_survival`)
Loss	Pinball loss on death-time quantiles

C.0.1 ABLATION DESIGN

Survival-family axis. Table 9 lists the generative equation of every family tested.

Table 9: Structural survival families compared on the family axis. H, O, A and h denote the cumulative hazard, cumulative odds, Aranda-Ordaz/Box-Cox transform, and hazard of $T \mid x$; a 0 subscript denotes the shared random baseline.

Family	**Arm(s)**	**Generative structure**
Proportional hazards	`cox`	$H(t \mid x) = e^{\eta} H_0(t)$
Proportional odds	`po`	$O(t \mid x) = e^{\eta} O_0(t)$, $O = (1-S)/S$
Transformation family	`transform`	$A_\rho(t \mid x) = e^{\eta} A_\rho(S_0(t))$, $\rho \sim [0.05, 3]$ per dataset (interpolates PH at $\rho \to 0$ and PO at $\rho = 1$)
Additive hazards	`ah`	$h(t \mid x) = h_0(t) + c \cdot \tanh(\eta)$, clamped at 0
Accelerated hazards	`ach`	$h(t \mid x) = h_0(t\, e^{\eta})$ (curves leave S=1 together)
AFT, homoscedastic	`aft-homo`	$\log T = \mu(x) + (\log T_0 - \mathbb{E} \log T_0)$
AFT, heteroscedastic	`aft-hetero`	as above with a per-subject log-time spread, producing crossing survival curves
Quantile model	`quantile`	$\log Q_p(T \mid x) = \log Q_{0,p} + \eta_p(x)$, 3 anchor levels
Classical parametric laws	`law`	Weibull / gamma / exponential / log-logistic mixture ("all"), proportional hazards
Simplified (reduced)	`simple`	single-law, non-mixture, single label draw
Cox + AFT mixture	`cox-aft`	per-dataset random choice of `cox` or `aft`
Grand mixture	`grand-mix`	per-dataset random choice among `cox`, `aft`, `po`, `ah`, `quantile`
SurvPFN prior	`survpfn`	closed-form Weibull PH/non-PH times, ported from SurvPFN's own pretraining prior
SurvivalPFN prior	`survivalpfn`	nonparametric hazard shape via a monotone Bernstein-polynomial map, ported from SurvivalPFN's prior
SIC prior	`sic`	extended-hazards law with a per-dataset hazard-family draw, ported from SIC's pretraining recipe

Recipe axes. Seven arms each flip exactly one flag off the control, holding `generation_type=po` fixed (Table 10). Two of the axes (discretization, administrative censoring) are already off in the control, so we ablate by turning them *on* rather than removing them – the same one-knob-at-a-time principle, applied in the direction the control recipe permits.

Table 10: Recipe axes ablated against the `po` control, one flag at a time.

Axis	Control	Variant(s)
Discretization	Off	On (10% per-dataset chance of coarsening times to a grid)
Administrative censoring	Off	On (50% per-dataset chance of a study-end cutoff)
Label draws per row	1	5, 20
Quantile head width	99 levels	999 levels
Auxiliary censoring head	Event-only targets	+ censoring-distribution head (train-time only, dropped at inference)

C.0.2 Evaluation protocol

Arms are evaluated at their final checkpoint on frozen, held-out validation datasets. To account for varying dataset scales and censoring rates, raw metrics are min–max normalized per dataset (100 = best arm, 0 = worst arm) before averaging. The primary ranking metric is the mean Integrated Brier Score (IBS).

D Other Results

D.1 All held-out datasets

D.1.1 IBS

Results for the IBS on held-out datasets are shown in Figures 6, 7, 8, 9.

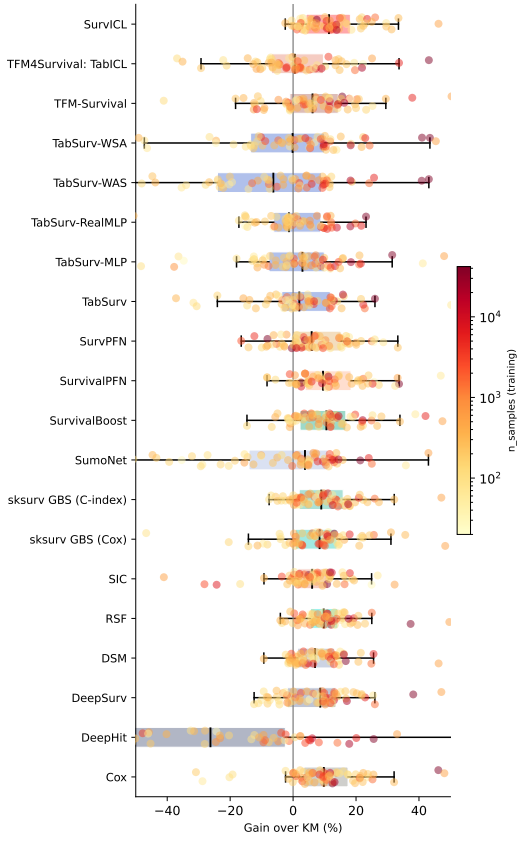


Figure 6: **IBS on all held-out datasets, compared to the IBS score obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

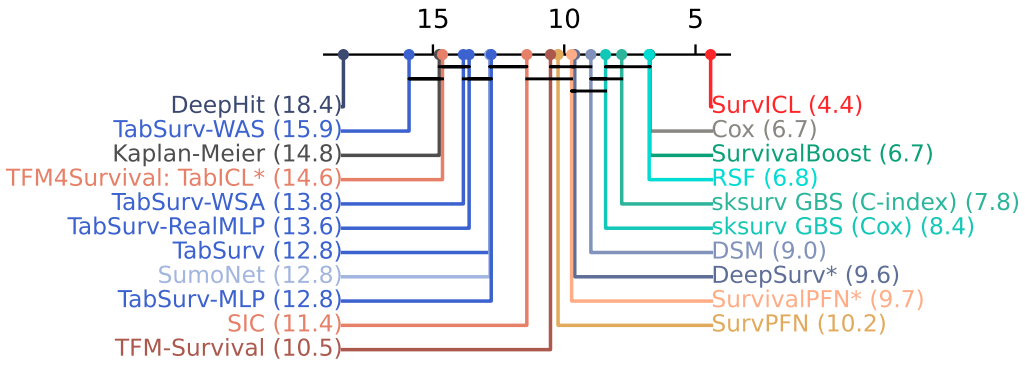


Figure 7: **Critical difference diagram of average ranks on all held-out datasets (lower is better) for IBS.** Methods joined by a bar are not significantly different

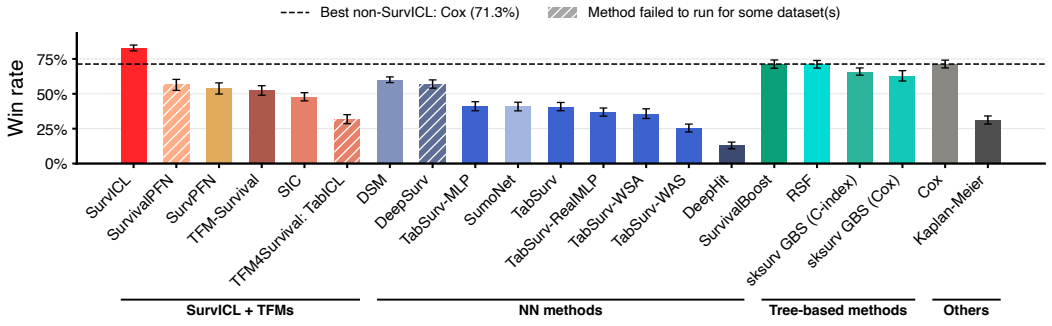


Figure 8: **Pairwise win rate IBS all held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a lower IBS. The scores are the same as in Figure 1, but shown with a different grouping.

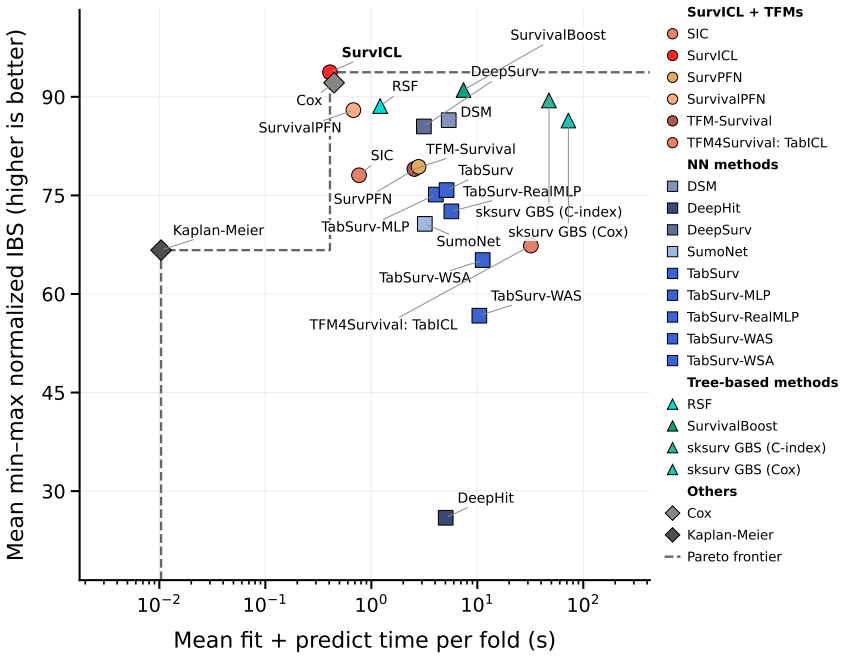


Figure 9: **Pareto front of mean normalized IBS (higher is better) and fit + predict time.** We report average performances over all held-out datasets. IBS was normalized to provide comparability across all datasets.

D.1.2 C-INDEX50

Results for the C-index on held-out datasets are shown in Figures 10, 11, 12, 13.

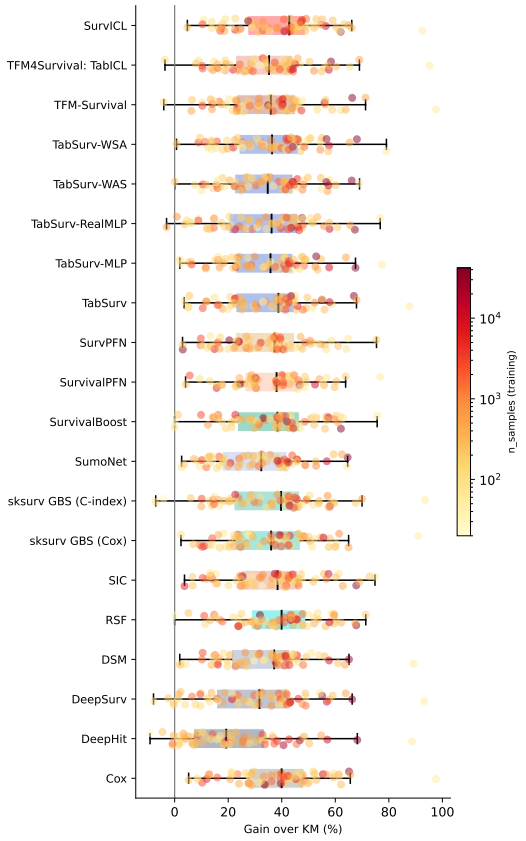


Figure 10: **C-index on the 50th quantile on all held-out datasets, compared to the score obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

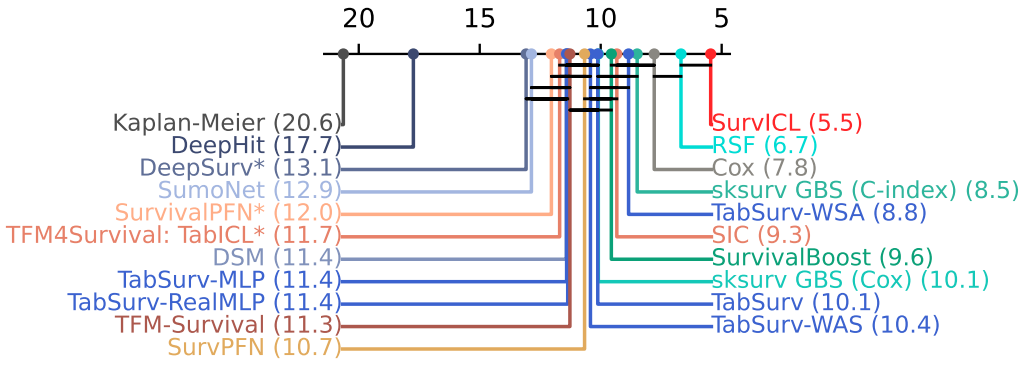


Figure 11: **Critical difference diagram of average ranks on all held-out datasets (lower is better) for the C-index on the 50th quantile.** Methods joined by a bar are not significantly different.

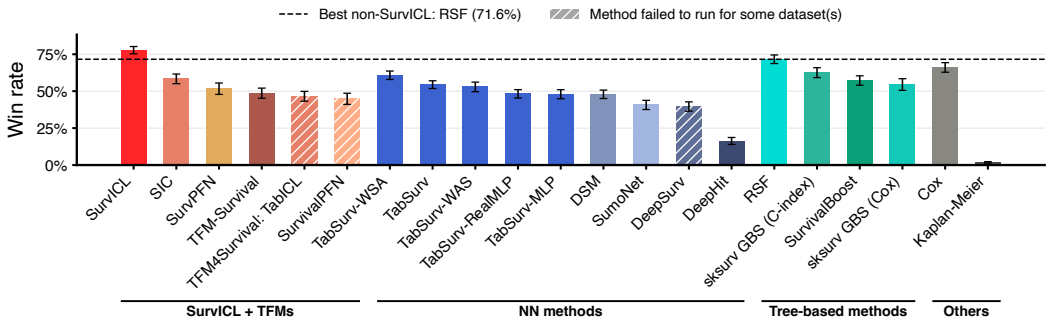


Figure 12: **Pairwise win rate in the C-index on the 50th quantile on all held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a higher C-index.

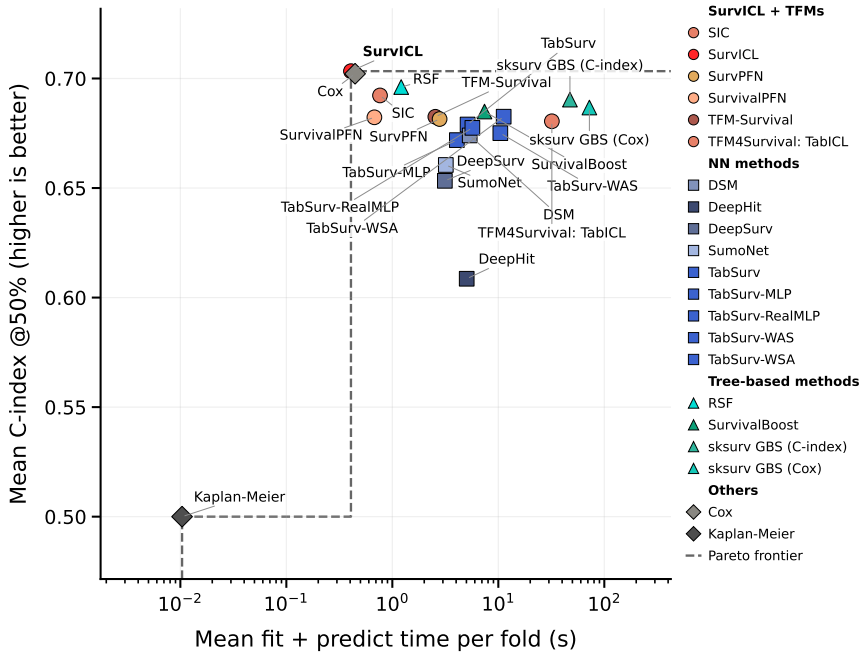


Figure 13: **Pareto front of mean C-index on the 50th quantile (higher is better) and fit + predict time.** We report average performances over all held-out datasets.

D.1.3 KM-CALIBRATION

Results for the KM-calibration on held-out datasets are shown in Figures 14, 15, 16.

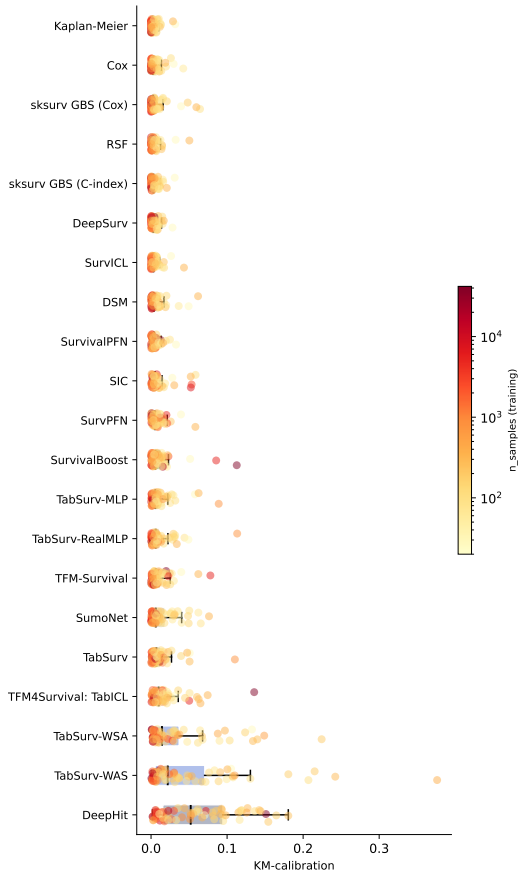


Figure 14: **KM calibration on all held-out datasets (lower is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

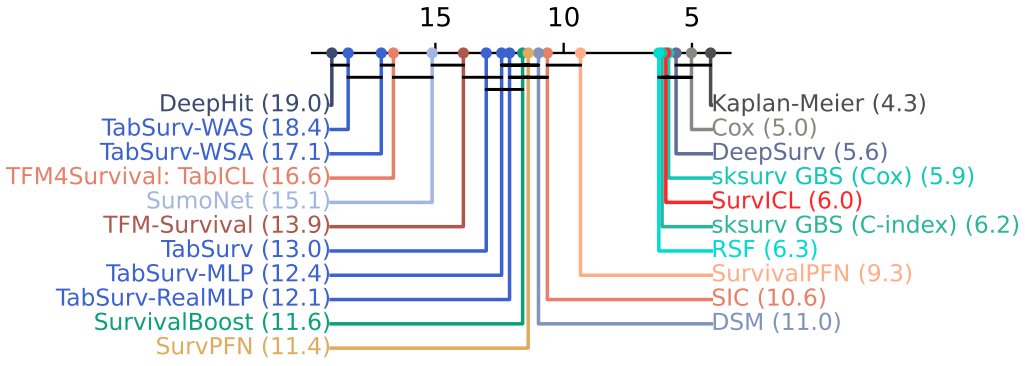


Figure 15: **Critical difference diagram of average ranks on all held-out datasets (lower is better) for the KM calibration.** Methods joined by a bar are not significantly different.

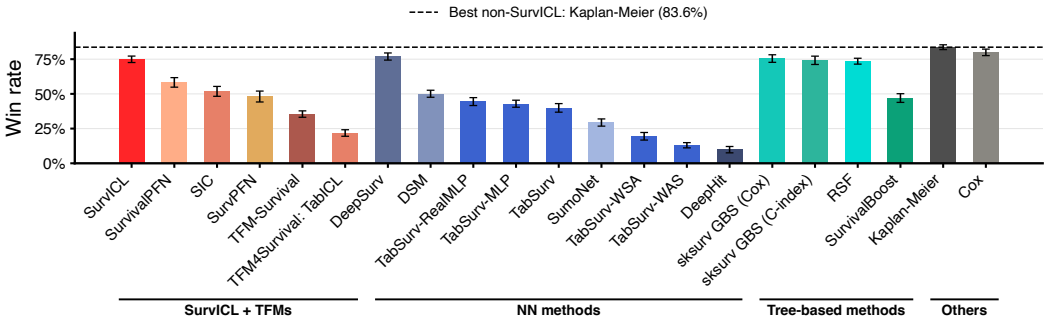


Figure 16: **Pairwise win rate in KM calibration on all held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a better KM calibration.

D.1.4 MAE

Results for the MAE on held-out datasets are shown in Figures 17, 18, 19, 20.

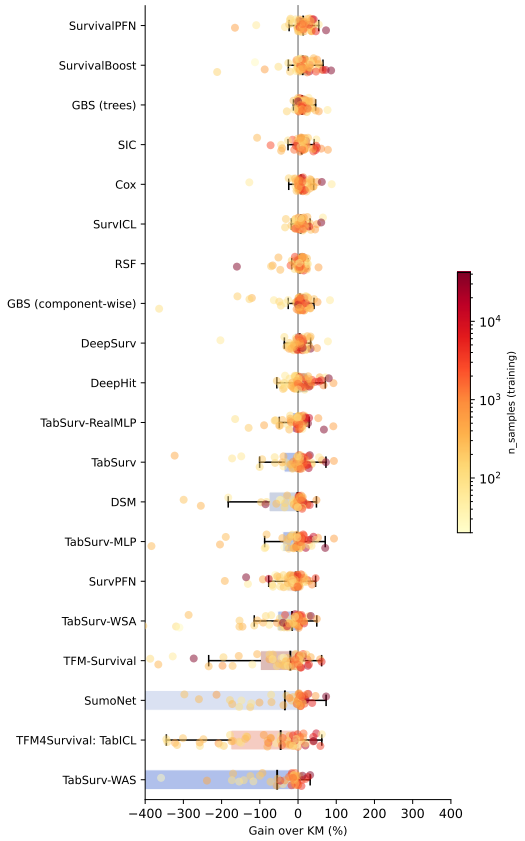


Figure 17: **MAE on all held-out datasets, compared to the MAE obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

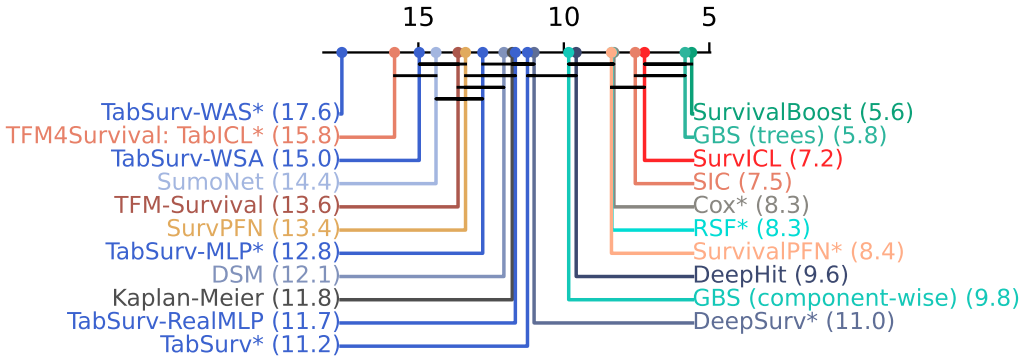


Figure 18: **Critical difference diagram of average ranks on all held-out datasets (lower is better) for the MAE.** Methods joined by a bar are not significantly different

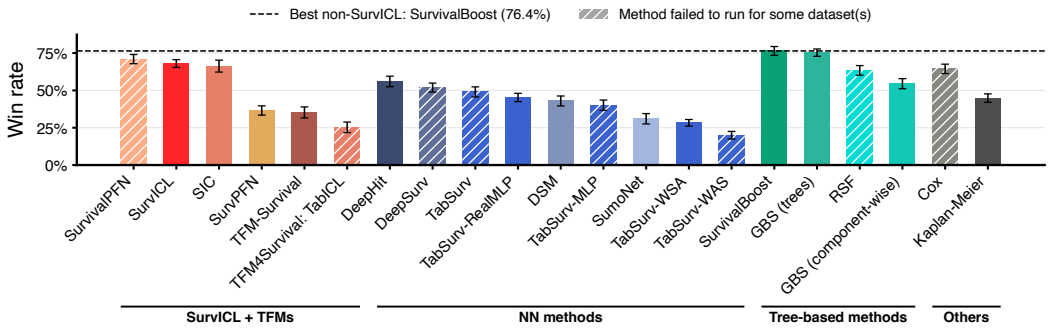


Figure 19: **Pairwise win rate in MAE on all held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a lower MAE.

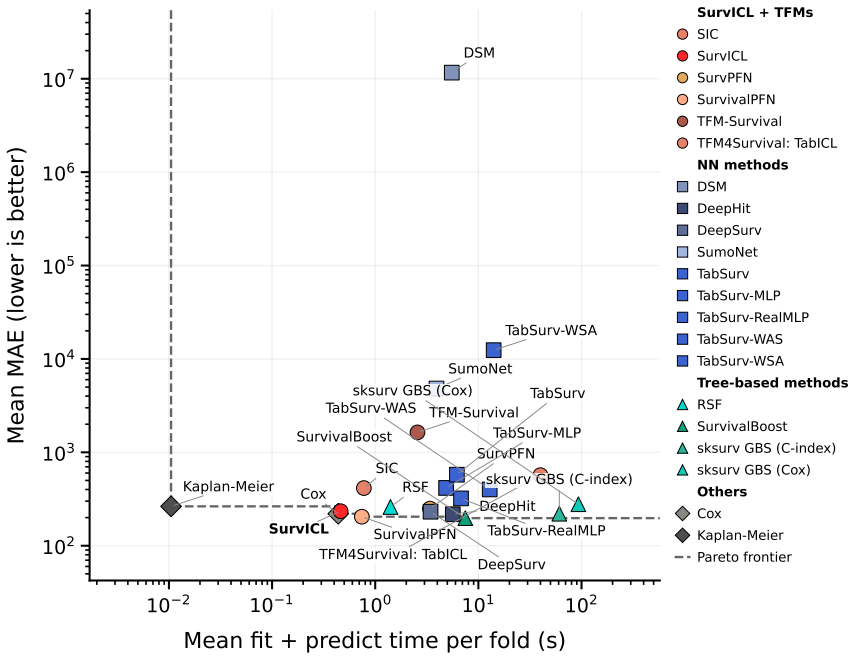


Figure 20: **Pareto front of mean MAE (lower is better) and fit + predict time.** We report average performances over all held-out datasets.

D.2 Held-out datasets with less than 1000 samples

D.2.1 IBS

Results for the IBS on small held-out datasets are shown in Figures 21, 22, 23, 24.

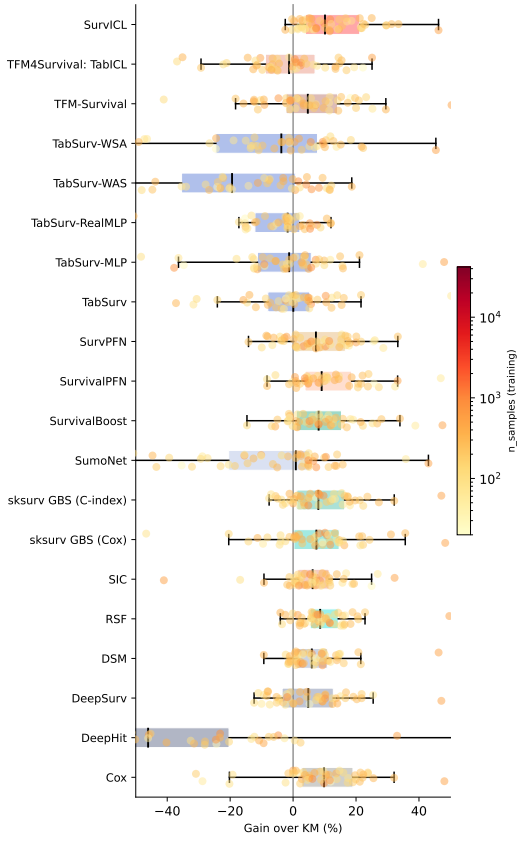


Figure 21: **IBS on the small held-out datasets, compared to the IBS score obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

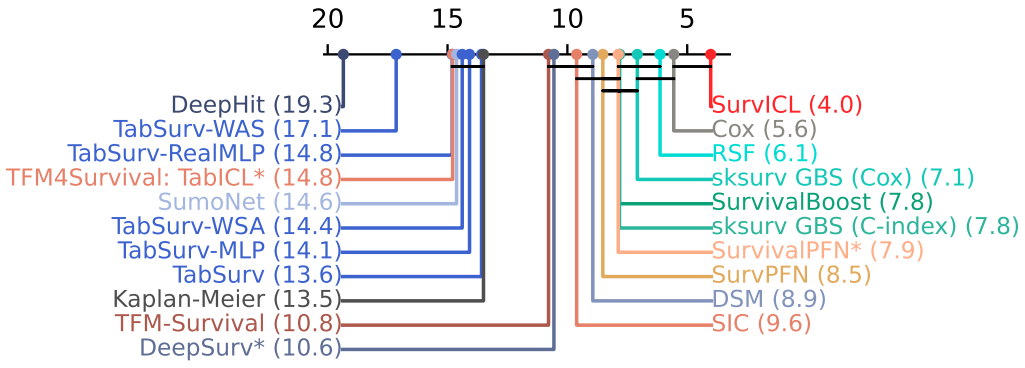


Figure 22: **Critical difference diagram of average ranks on the small held-out datasets (lower is better) for the IBS.** Methods joined by a bar are not significantly different.

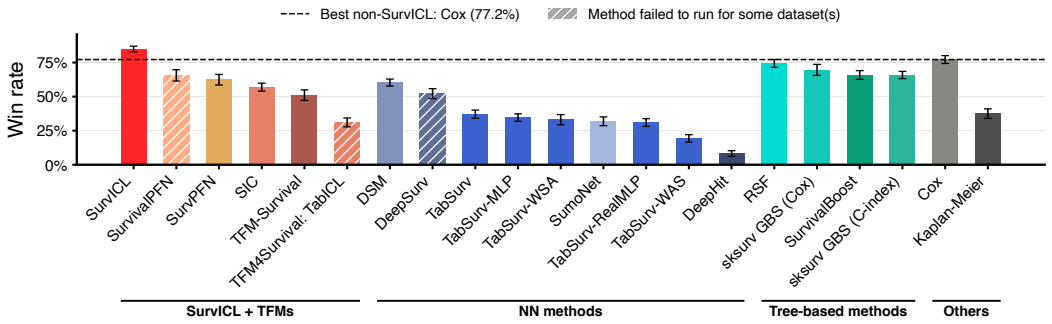


Figure 23: **Pairwise win rate in IBS on the small held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a lower IBS.

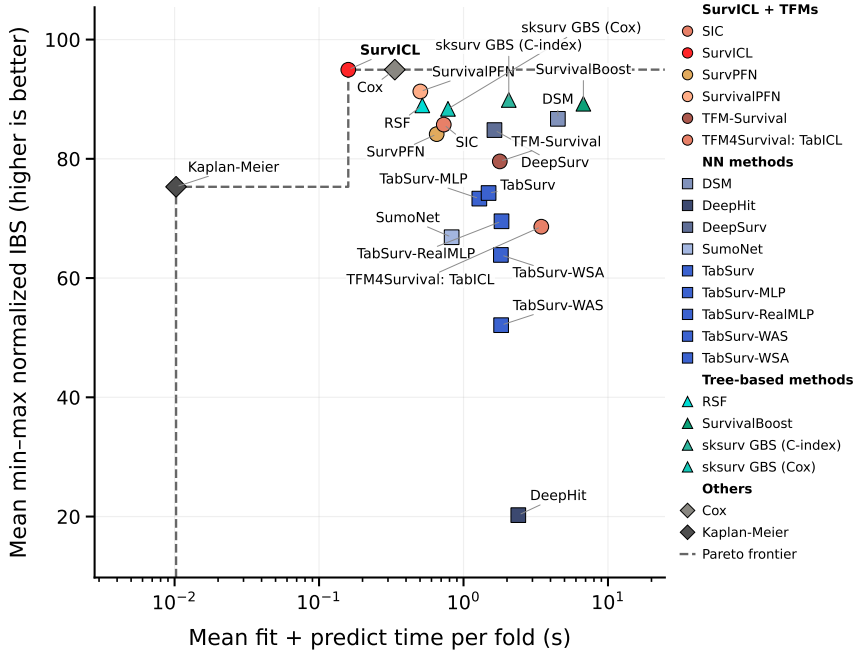


Figure 24: **Pareto front of mean normalized IBS (higher is better) and fit + predict time.** We report average performances over the small held-out datasets. IBS was normalized to provide comparability across all datasets.

D.2.2 C-INDEX50

Results for the C-index at the 50th quantile on small held-out datasets are shown in Figures 25, 26, 27, 28.

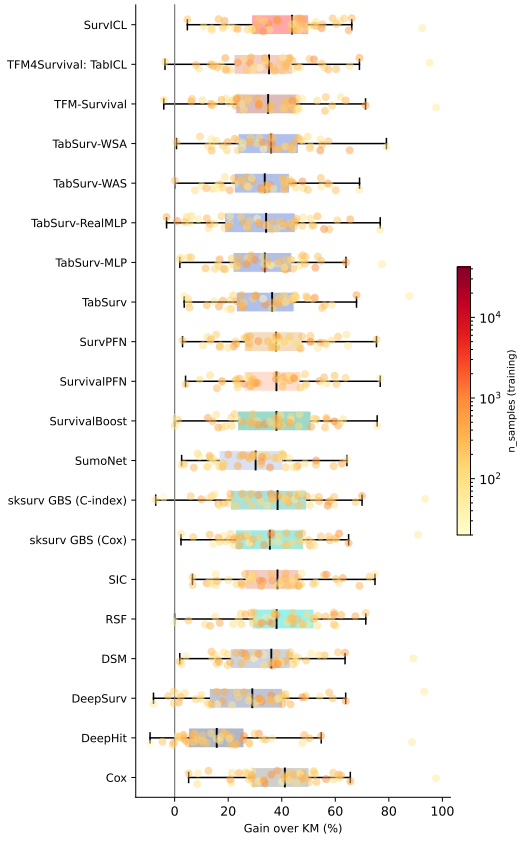


Figure 25: **C-index at the 50th quantile on the small held-out datasets, compared to the C-index score obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

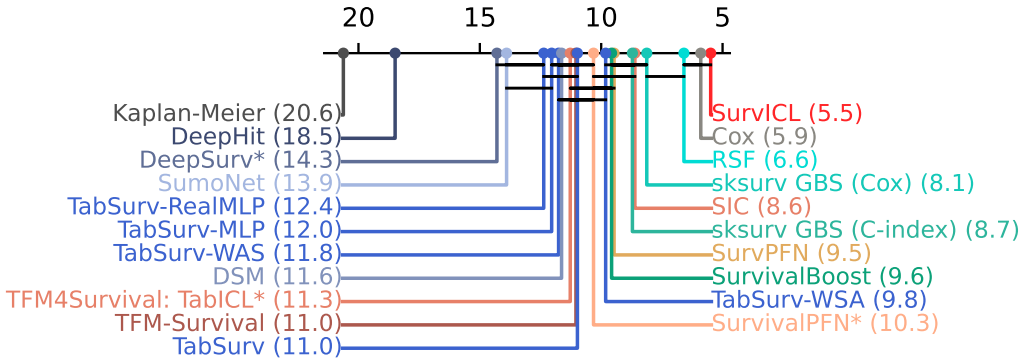


Figure 26: **Critical difference diagram of average ranks on the small held-out datasets (lower is better) for the C-index at the 50th quantile.** Methods joined by a bar are not significantly different.

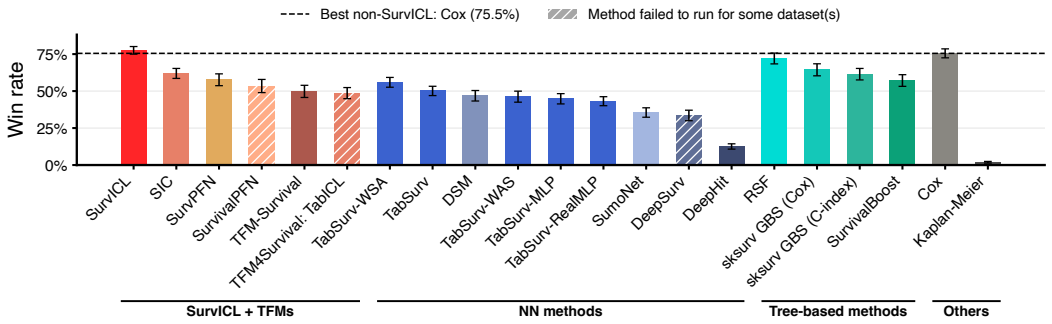


Figure 27: **Pairwise win rate in the C-index at the 50th quantile on the small held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a higher C-index.

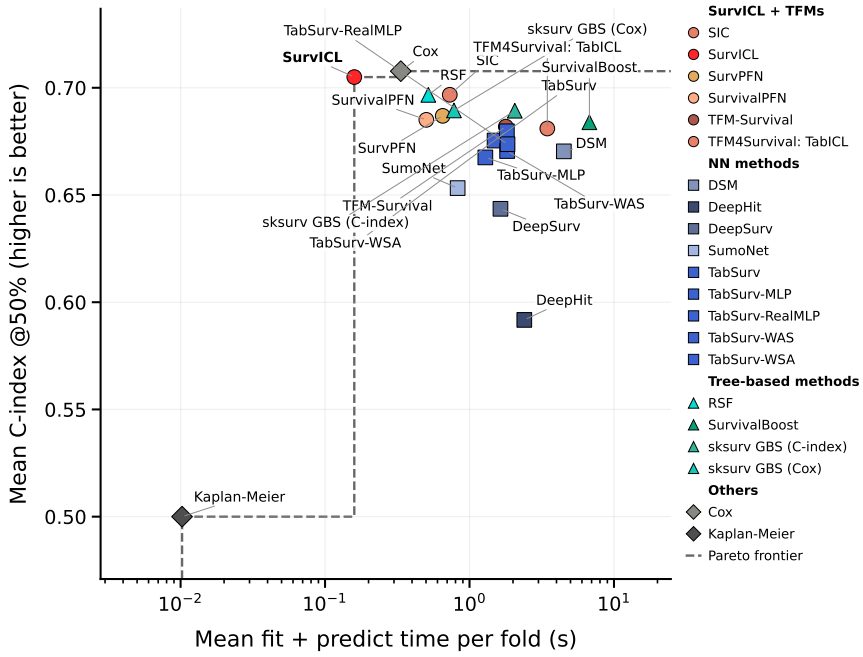


Figure 28: **Pareto front of mean C-index on the 50th quantile (higher is better) and fit + predict time.** We report average performances over the small held-out datasets.

D.2.3 KM-CALIBRATION

Results for the KM-calibration on small held-out datasets are shown in Figures 29, 30, 31.

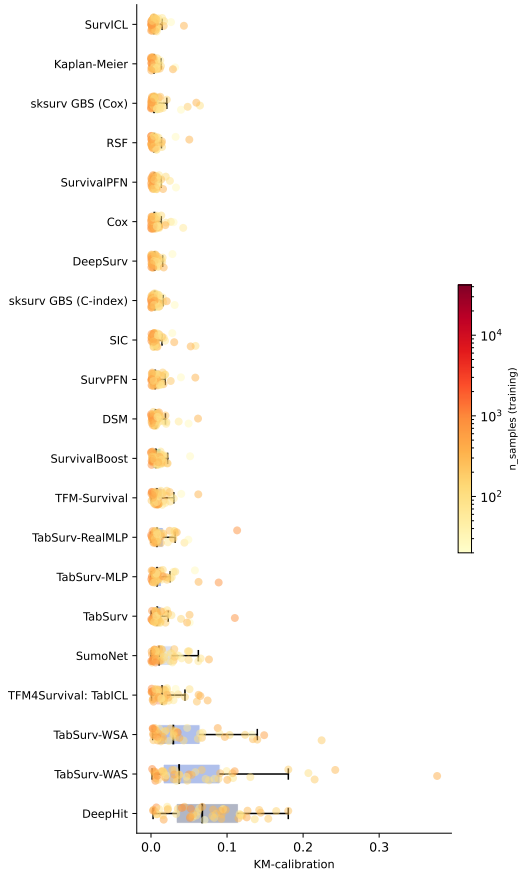


Figure 29: **KM calibration on the small held-out datasets (lower is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

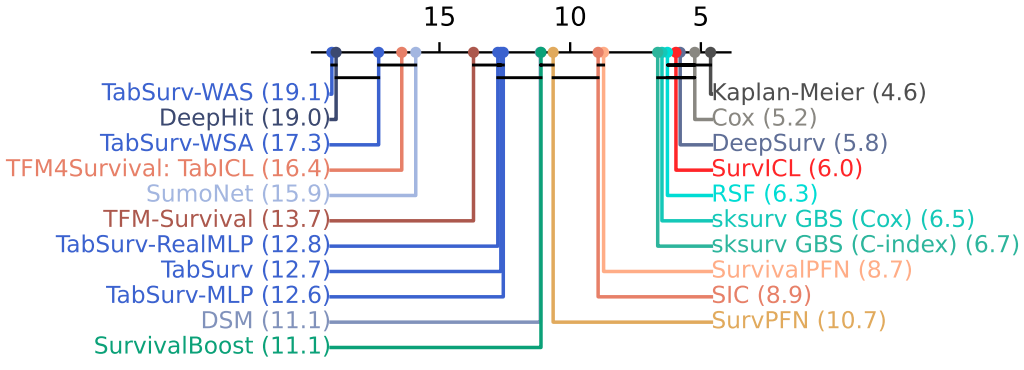


Figure 30: **Critical difference diagram of average ranks on the small held-out datasets (lower is better) for the KM calibration.** Methods joined by a bar are not significantly different.

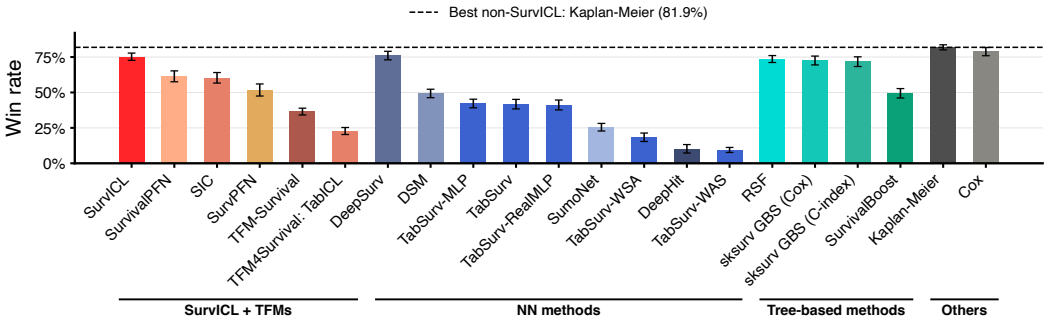


Figure 31: **Pairwise win rate in the KM-calibration on the small held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a better KM-calibration.

D.2.4 MAE

Results for the MAE on small held-out datasets are shown in Figures 32, 33, 34, 35.

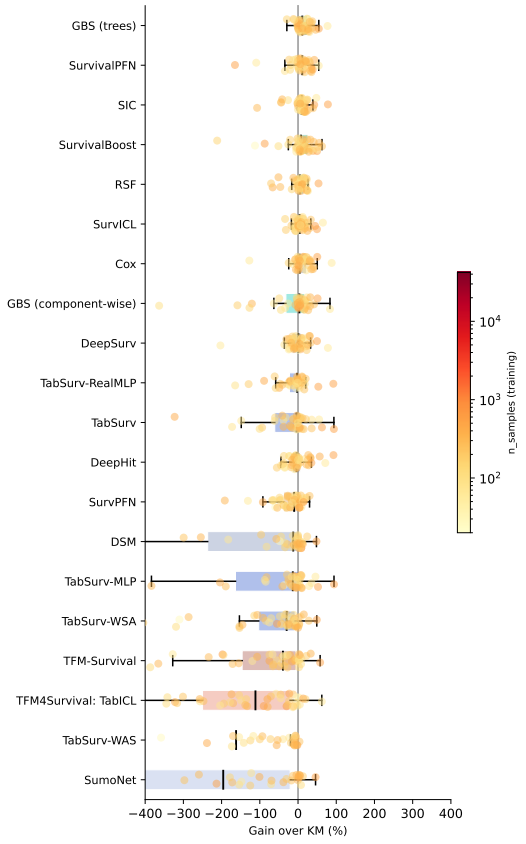


Figure 32: **MAE on the small held-out datasets, compared to the MAE obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

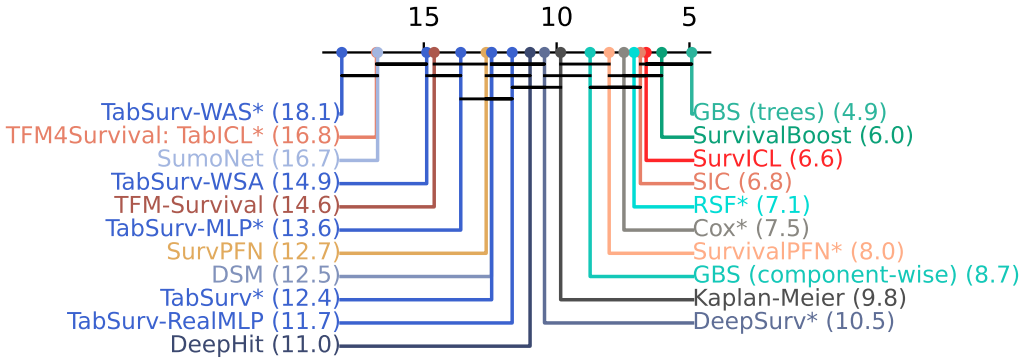


Figure 33: **Critical difference diagram of average ranks on the small held-out datasets (lower is better) for the MAE.** Methods joined by a bar are not significantly different.

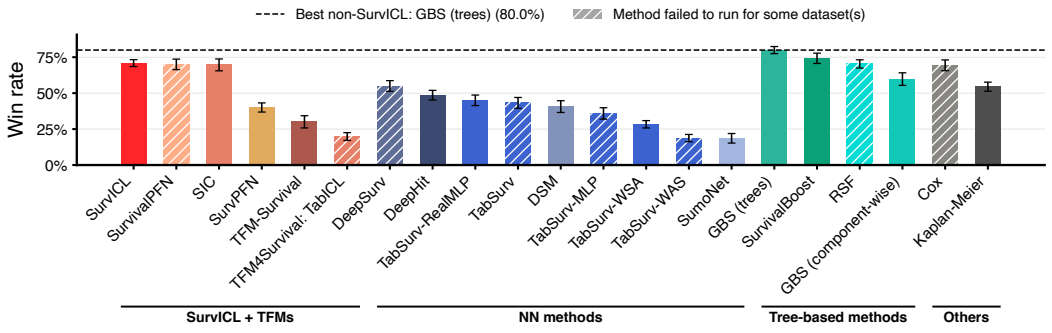


Figure 34: **Pairwise win rate in the MAE on the small held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a lower MAE.

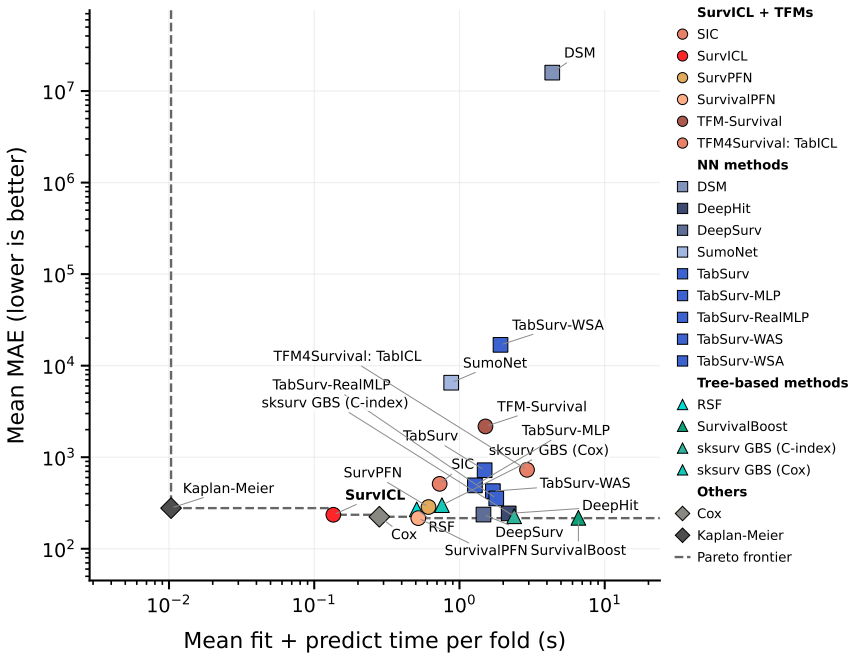


Figure 35: **Pareto front of mean MAE (lower is better) and fit + predict time.** We report average performances over the small held-out datasets.

D.3 Held-out datasets with more than 1000 samples

D.3.1 IBS

Results for the IBS on large held-out datasets are shown in Figures 36, 37, 38, 39.

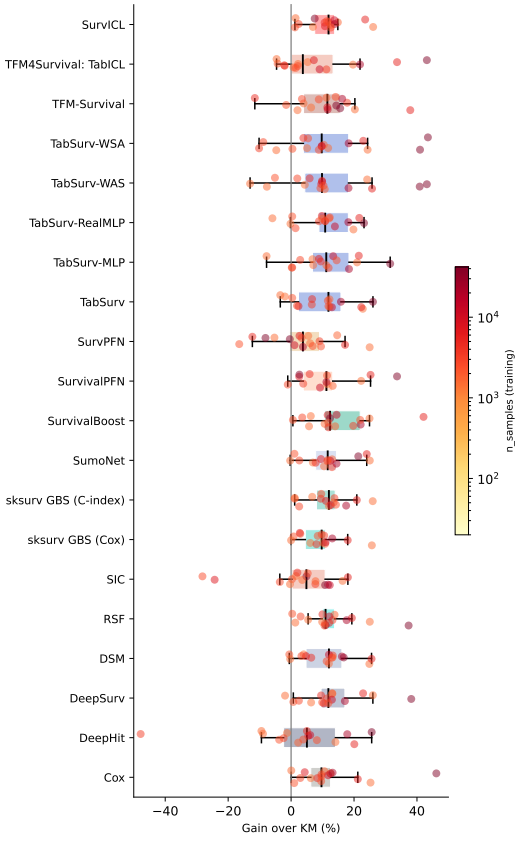


Figure 36: **IBS on the large held-out datasets, compared to the IBS obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

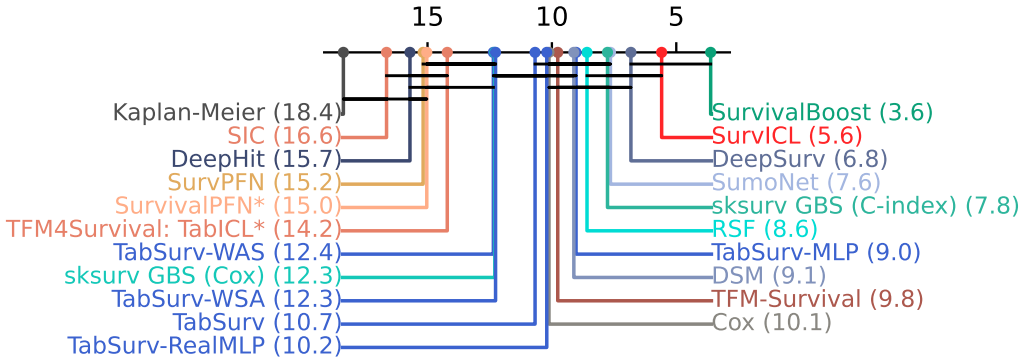


Figure 37: **Critical difference diagram of average ranks on the large held-out datasets (lower is better) for the IBS.** Methods joined by a bar are not significantly different.

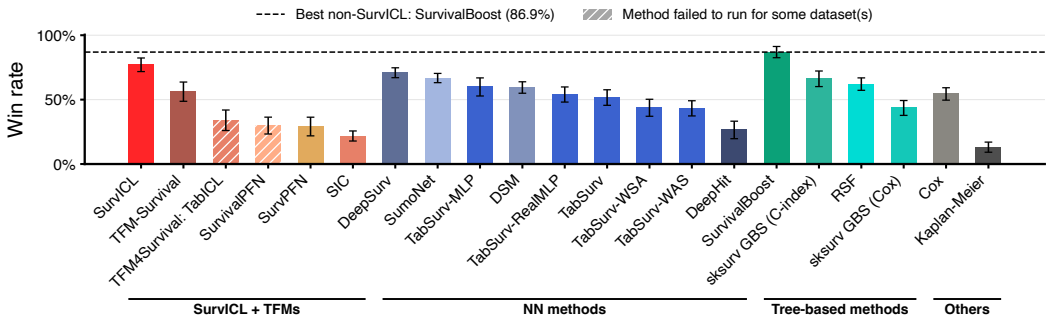


Figure 38: **Pairwise win rate in the IBS on the large held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a lower IBS.

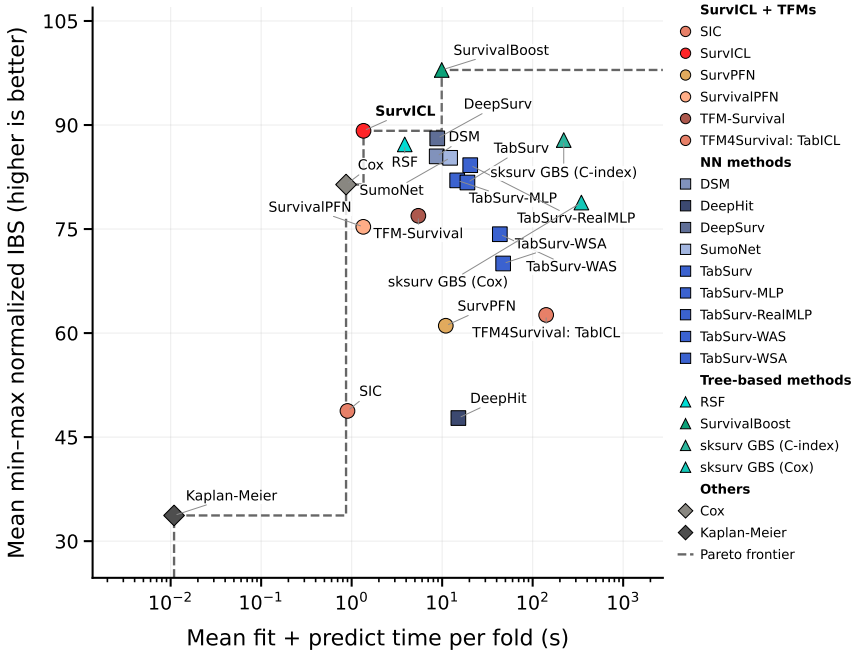


Figure 39: **Pareto front of mean normalized IBS (higher is better) and fit + predict time.** We report average performances over the large held-out datasets. IBS was normalized to provide comparability across all datasets.

D.3.2 C-INDEX

Results for the C-index on large held-out datasets are shown in Figures 40, 41, 42, 43.

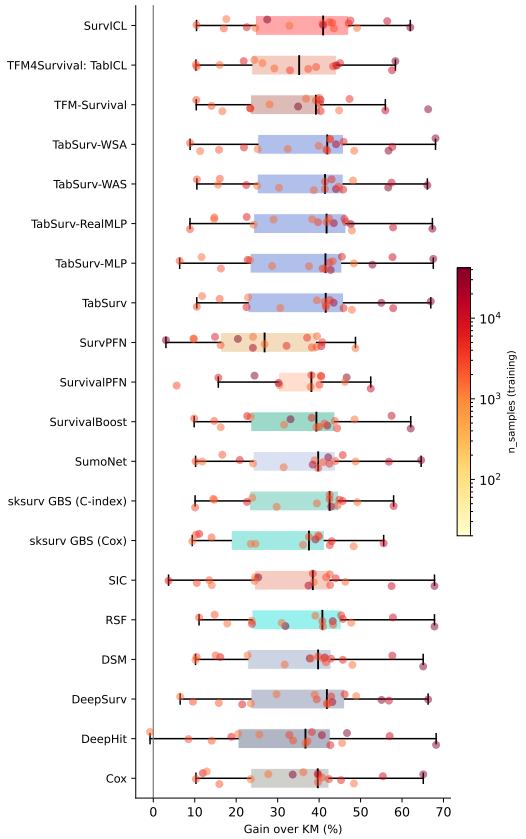


Figure 40: **C-index at the 50th quantile on the large held-out datasets, compared to the C-index obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

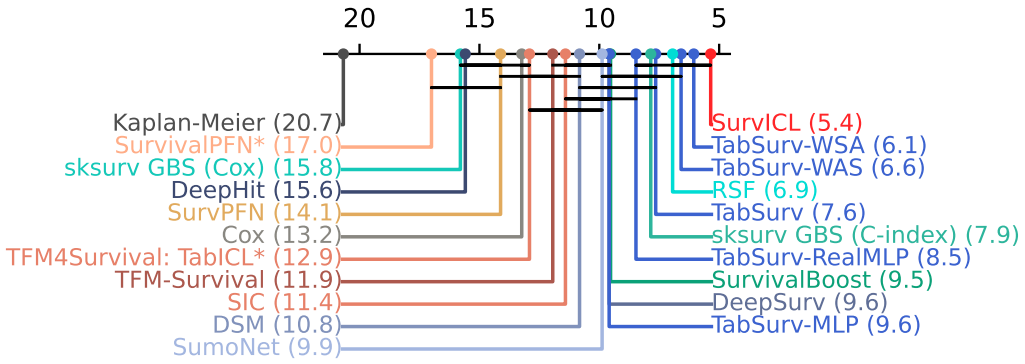


Figure 41: **Critical difference diagram of average ranks on the large held-out datasets (lower is better) for the C-index at the 50th quantile.** Methods joined by a bar are not significantly different.

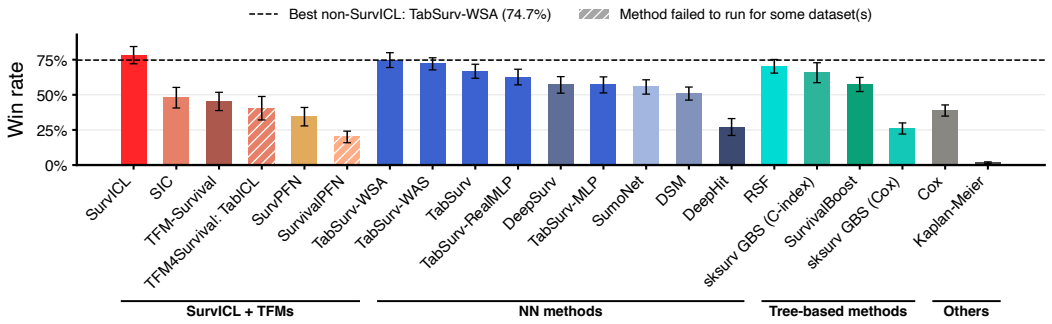


Figure 42: **Pairwise win rate in the C-index at the 50th quantile on the large held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a higher C-index.

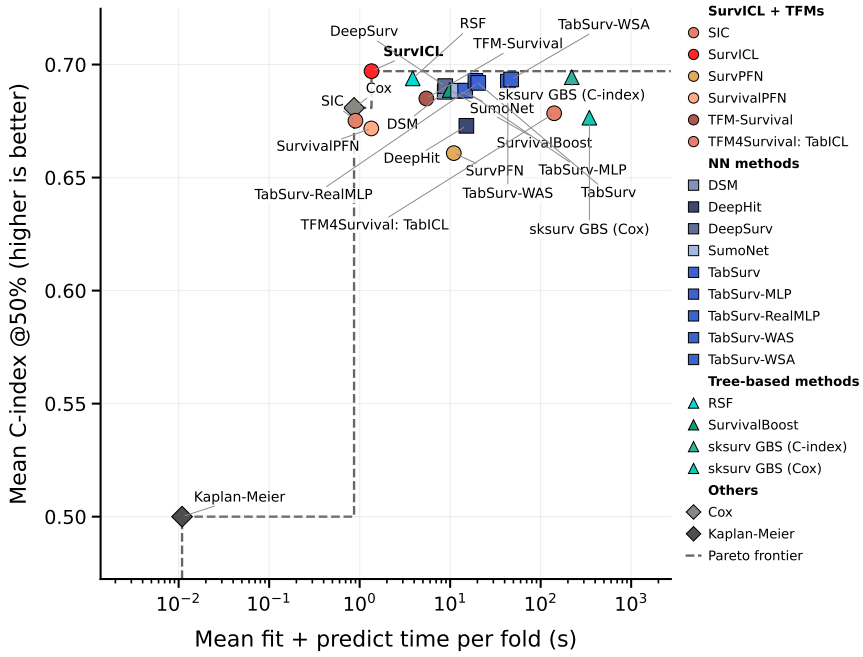


Figure 43: **Pareto front of mean C-index on the 50th quantile (higher is better) and fit + predict time.** We report average performances over the large held-out datasets.

D.3.3 KM-CALIBRATION

Results for the KM-calibration on large held-out datasets are shown in Figures 44, 45, 46.

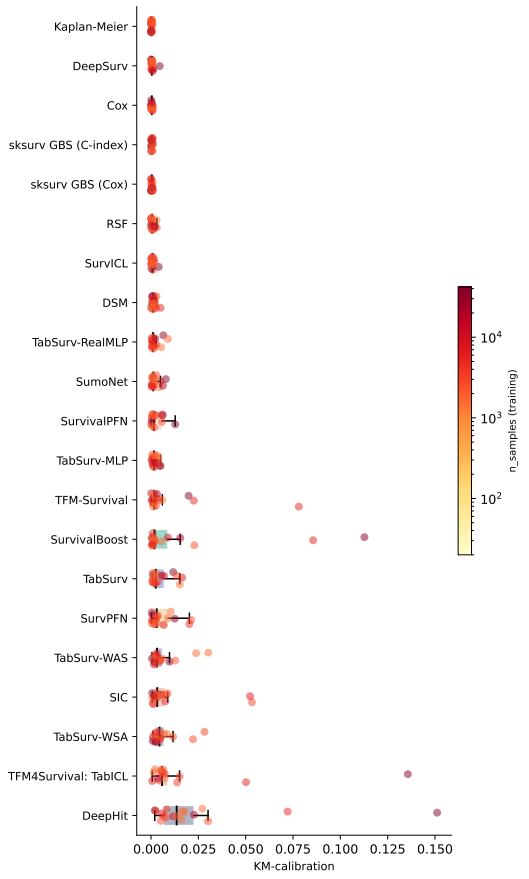


Figure 44: **KM calibration on the large held-out datasets (lower is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

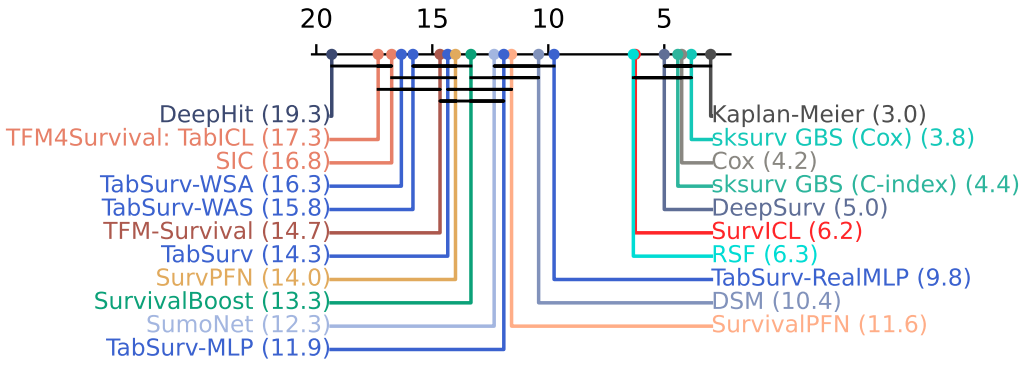


Figure 45: **Critical difference diagram of average ranks on the large held-out datasets (lower is better) for the KM calibration.** Methods joined by a bar are not significantly different.

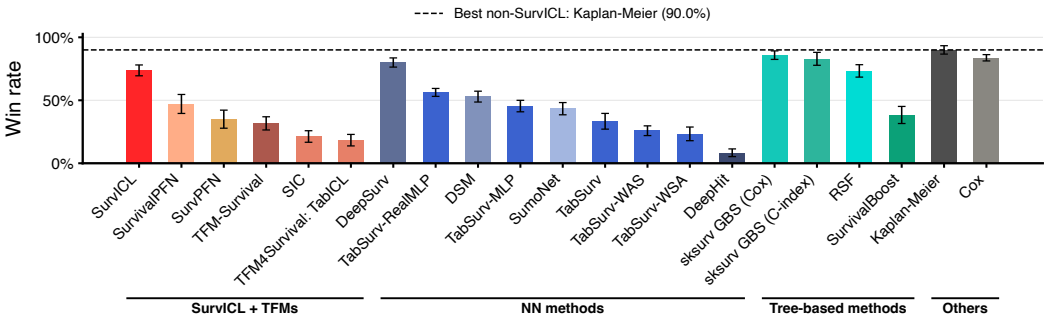


Figure 46: **Pairwise win rate in the KM calibration on the large held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a better KM calibration.

D.3.4 MAE

Results for the MAE on large held-out datasets are shown in Figures 47, 48, 49.

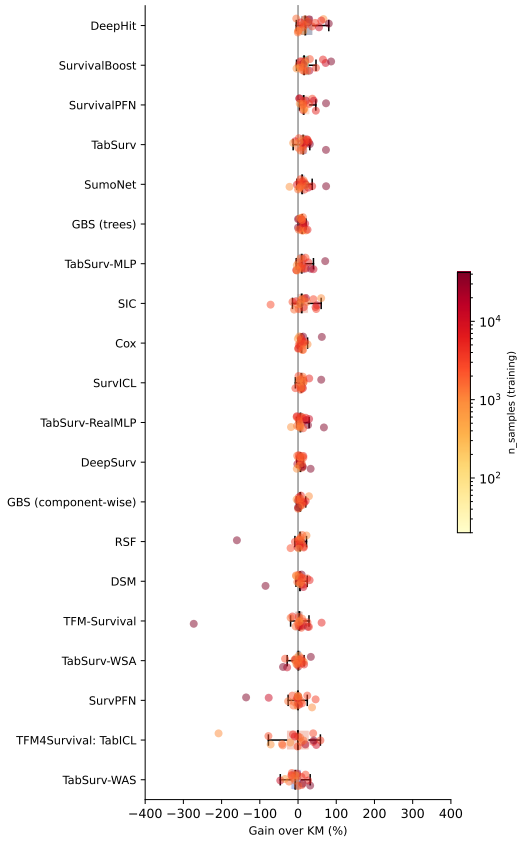


Figure 47: **MAE on the large held-out datasets, compared to the MAE obtained with a Kaplan-Meier estimator (higher is better).** The points show individual dataset performances, and a boxplot summarizes the overall distribution. The dot color represents the dataset size.

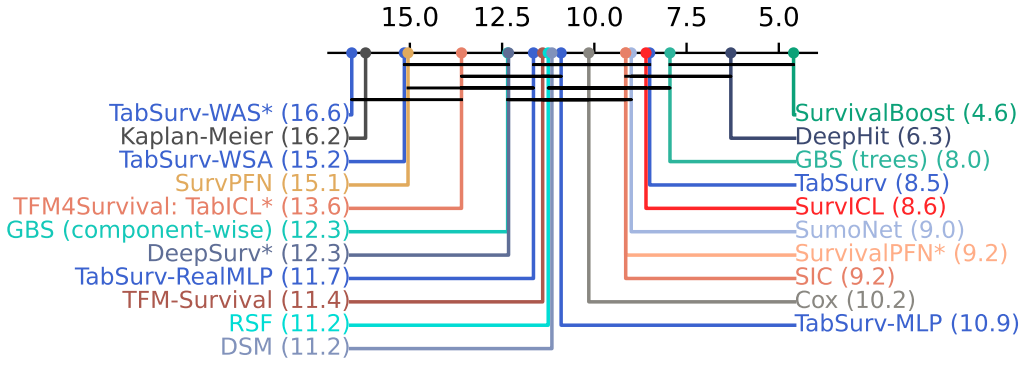


Figure 48: **Critical difference diagram of average ranks on the large held-out datasets (lower is better) for the MAE.** Methods joined by a bar are not significantly different.

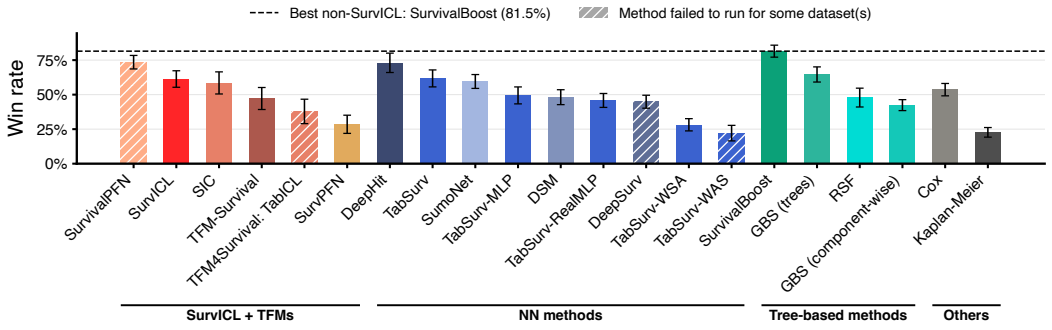


Figure 49: **Pairwise win rate in the MAE on the large held-out datasets:** the fraction of (dataset, competitor) pairs on which a method achieves a lower MAE.

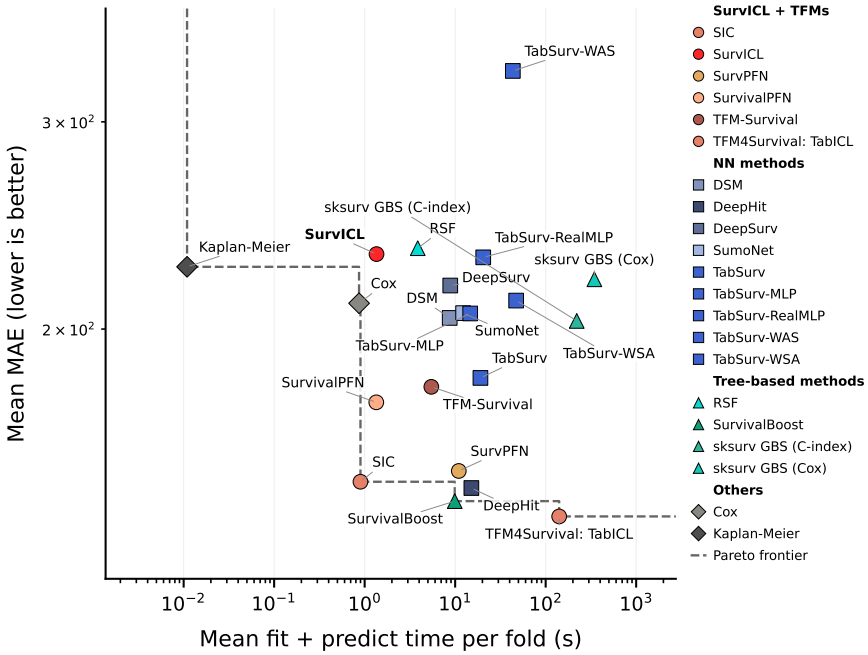


Figure 50: **Pareto front of mean MAE (lower is better) and fit + predict time.** We report average performances over the large held-out datasets.