THE LOSS OR THE ARCHITECTURE? WHAT MATTERS IN SURVIVAL ANALYSIS WITH COMPETING RISKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Often, *eg* in healthcare and predictive maintenance, what matters is which of several competing causes will strike first, and with what probability: survival analysis with competing risks. Corresponding deep learning methods are developed as bundles: a new architecture together with a new training loss, tested on one to five datasets against a handful of baselines. Thus, a state-of-the-art result cannot be attributed to the architecture, the loss, or the evaluation protocol. The coupling is not accidental: with one incidence curve per cause, a network's head is tied to the losses expressible on it. We vary the two axes independently, three losses, two of them strictly proper, on four backbones, across two modalities that tease the axes apart: 20 tabular and 3 pathology-image datasets. One loss wins in both: modelling cumulative incidence directly under censoring weighting. It brings twice as much as benefits a good choice of backbone on competing-risks data, and no more than it on single-event data: the loss is decisive for competing risks. The backbone matters too, and must match the modality: tabular networks lead on tables, pretrained vision encoders on images. The two choices compound rather than substitute. Proper scoring rules reveal these dynamics; the concordance index (C-index), which is not proper, does not separate models: neither axis moves it, not even tuning on the loss built to improve it. This is an avenue of progress: composed, the two give a model nobody has tried, RealMLP or TabM trained with the cumulative-incidence loss, which takes the best Brier score in our benchmark.

1 INTRODUCTION: COMPETING RISKS, AND WHERE THE GAINS COME FROM

Time-to-event prediction arises when the quantity of interest is a delay: how long until a patient relapses, a borrower defaults, a component fails. What makes it more than regression is right-censoring: When follow-up ends, many subjects have simply not had the event yet, and all we know is that their time exceeds it. Discarding them biases the model towards those who failed early; counting them as event-free biases it the other way. Survival models instead use what a censored subject does tell us, that the event came after a known time. They call for dedicated losses.

Competing risks adds a second twist: the event comes in several mutually exclusive kinds, and only the first is ever seen, since a patient who dies of heart disease cannot later die of cancer. Treating the other causes as further censoring and fitting one model per cause is tempting, but it answers the risk in a world where those causes do not exist and so overstates the risk that matters. What predictions need is the cumulative incidence function (CIF) of each cause: the probability that it strikes before a given time and before any competing cause. Deciding whether to treat a patient or service a machine builds on that absolute probability, more than finding who is most at risk.

Machine-learning research has produced many competing-risks methods, from semi-parametric estimators to random forests and neural networks (Sec. 3), and recent ones frequently introduce a new architecture and a new training loss at once (Nagpal et al., 2021; Danks & Yau, 2022; Wang & Sun, 2022; Jeanselme et al., 2023). For example, Lee et al. (2018) pair a custom architecture with a likelihood-plus-ranking loss. Because the two axes are never varied separately, architecture and loss are confounded by construction, and a state-of-the-art result cannot be attributed to one or the other. The evidence base is also thin: Table 1 shows methods are introduced on one to five real competing-risks datasets and assessed with one to three metrics, not enough to separate models well. Often, the metric is concordance alone (C-index), which ranks individuals by risk and may miss

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.

differences in predicted probabilities, yet those absolute probabilities drive decisions in medicine (Van Calster et al., 2019). The effect of proper scoring rules has been isolated in standard survival analysis on simple MLPs (Yanagisawa, 2023), but competing risks has seen no such study.

We therefore build a benchmark that decouples the training loss from the backbone. Three losses are trained on shared neural backbones: a competing-risks likelihood over cause-specific event probabilities (HazardLikelihood), the hybrid ranking-plus-likelihood loss of DeepHit (HLRanking), and the cumulative-risk negative log-likelihood (CRNLL). The first and third are strictly proper scoring rules under their respective assumptions, meaning that only the true predictive distribution minimizes them; the ranking term makes HLRanking improper. We evaluate them on 10 competing-risks and 10 single-event tabular datasets (including SEER (SEER, 2025), MIMIC (Johnson et al., 2016; 2024), eICU (Pollard et al., 2018), and METABRIC (Mangravite, 2013)) with a randomized hyperparameter search for each training scenario, and report probabilistic accuracy, discrimination, and calibration.

Table 1: **Competing-risks methods are introduced on very thin evidence.** Datasets, baselines and metrics used by the papers proposing methods, against this study.

Publication	# Real Life CR Datasets	# baselines	# CR metrics
Ishwaran et al. 2014	5	5	2
Lee et al. 2018	1	3	1
Wang & Sun 2022	1	6	1
Jeanselme et al. 2023	3	5	3
Alberge et al. 2025	1	7	3
Ours	**10** tabular **3** vision	**9 baselines +** **12 loss×arch.**	**6**

We then adapted the whole comparison on pathology images, with pretrained vision backbones (He et al., 2016; Huang et al., 2017; Krizhevsky et al., 2012). This change of modality introduces new variation. Indeed, for tables gains from the backbones are small, so the architecture axis is narrow, while the loss reshapes what the whole network is fit to. The second modality, vision, tells the two axes apart cleanly, because it moves one of them a long way while leaving the other alone: we use the same three losses, whereas the backbone explores architecture dedicated to vision (pretrained encoder) as well as a general one, an MLP. The ordering of the losses survives the move, while the backbone that wins does not: pretrained encoders lead on images, tabular networks on tables.

Contributions

A controlled loss × architecture matrix: Each of three losses trained on each of four backbones, with the same search space and the same 150-iteration budget in every cell, so the two axes can be read off independently; and nine methods with their own fixed losses serve as baselines.
An evidence base large enough to separate two effects: 10 competing-risks and 10 single-event tabular datasets plus three built from pathology images (an order of magnitude more than the studies of Table 1) scored on metrics spanning accuracy, discrimination and calibration.
The loss decides in competing risks, and properness does not suffice: Modeling cumulative incidence directly under censoring weighting improves strongly on both hazard-based losses on tabular data, and still leads on every frozen vision encoder. It brings twice as much benefits as a good backbone in competing-risks settings, while in single-event settings the two contribute equally to the performance. Two strictly proper losses, CRNLL and HazardLikelihood, nonetheless sit far apart, so properness is needed to trust a metric but not sufficient to make a loss competitive. The one improper ingredient we test, DeepHit's ranking term, is down-weighted by tuning.
The backbone must match the modality: RealMLP and TabM lead on tables, pretrained encoders on images, and the MLP under-performs on images whatever the loss.
Ranking metrics cannot see any of this: Nothing measurably moves the C-index, not even tuning on the loss built to optimize it. A study reporting concordance alone, which Table 1 shows is common practice, would have concluded that none of these choices matters.
A stronger model, obtained by composition: Pairing a recent tabular architecture with the cumulative-incidence loss, which no prior work has done, beats every published neural competing-risks method on the Brier score, and meet or exceed results of SurvivalBoost, the tree-based method in our benchmark that builds on the same loss.

Scope We compare losses and backbones for competing-risks prediction rather than proposing new architecture. Some losses impose a parametric form on the CIF (eg Deep Survival Machines, Nagpal et al., 2021); they cannot be added onto an arbitrary backbone and so enter as fixed-loss baselines rather than as cells of the matrix. Time-varying covariates and recurrent events are out of scope.

2 SETTING: COMPETING RISKS, LOSSES, AND HOW TO SCORE THEM

In the following, we note random variables in capitalized letters, and observations are in lowercase.

2.1 COMPETING RISKS: ONE INCIDENCE CURVE PER CAUSE

We follow the standard formulation of the competing-risks problem (Pintilie, 2006), with $K > 1$ types of event. Let $T^\star \in \mathbb{R}_+$ be a continuous random variable representing the time to the first event, and $\Delta^\star \in \{1, \dots, K\}$ be a categorical variable denoting the specific event. We also face right-censoring, denoted C. The final time observation is $T = \min(T^\star, C)$. For a given sample, the observed triplet $(\mathbf{x}, (t, \delta))$, where $\mathbf{x} \in \mathbb{R}^d$ denotes the d covariates, t the time observed and δ indicates the event type (with $\delta = 0$ representing censoring). The primary quantity of interest is the Cumulative Incidence Function (CIF) for each cause k, defined as $F_k(t, \mathbf{x}) = \mathbb{P}(T^\star \leq t, \Delta^\star = k \mid \mathbf{X} = \mathbf{x})$. It can be modelled directly (Aalen & Johansen, 1978; Lee et al., 2018; Alberge et al., 2025), or through the cause-specific hazard or density, its derivative (Fine & Gray, 1999; Kvamme & Borgan, 2019; Wang & Sun, 2022; Jeanselme et al., 2023). Appendix E gives the computations.

2.2 THREE LOSSES THAT ANY BACKBONE CAN CARRY, TWO OF THEM PROPER

We consider losses imposing no strong parametric prior on the CIF, so they can be moved across backbones. Three qualify: two strictly proper scoring rules and the composite loss of Lee et al. (2018).

HazardLikelihood Loss The most widely adopted loss is the competing-risks negative log-likelihood, a strictly proper scoring rule (Guan, 2021) used in several works (Fine & Gray, 1999; Jeanselme et al., 2024). It is also what the SurvTRACE repository implements (Wang & Sun, 2022), although the original manuscript discusses alternatives.

Definition 2.1 (Hazards Likelihood)**.** The log-likelihood in competing risks setting can be defined as:

$$\mathcal{L}_{HL} \triangleq -\sum_{i=1}^{N} \left[\mathbb{1}_{\delta_i \neq 0} \cdot \log\left(\hat{f}_k(t_i | \mathbf{x}_i) \right) + \mathbb{1}_{\delta_i = 0} \cdot \log\left(1 - \sum_{k=1}^{K} \hat{F}_k\left(t_i \mid \mathbf{x}_i\right) \right) \right], \tag{1}$$

where $\delta_i \in \{0, 1, \dots, K\}$ is the event indicator (with 0 denoting right-censoring), $\hat{f}_{\delta_i}$ is the predicted cause-specific density for the realized event, and $\hat{F}_k$ is the predicted CIF for cause k. As shown by Guan (2021), this loss is a proper scoring rule.

HazardLikelihoodRanking Loss The loss used to train DeepHit (Lee et al., 2018), which we call HazardLikelihoodRanking (HLRanking), adds a ranking term to that likelihood. Following its pycox implementation (Kvamme et al., 2019), $\mathcal{L}_{HLR} = \alpha\, \mathcal{L}_{HL} + (1 - \alpha)\, \mathcal{L}_R$ with a hyperparameter $\alpha \in [0, 1]$, where $\mathcal{L}_{HL}$ is the negative log-likelihood of Def. 2.1 and $\mathcal{L}_R$ is a ranking loss (Def. 2.2).

Definition 2.2 (Ranking Loss)**.** Let $A_{k,i,j} \triangleq \mathbb{1}\left(\delta_i = k, t_i < t_j\right)$ be an indicator that patient i experiences event k before patient j experiences any event or is censored. It is defined as:

$$\mathcal{L}_R \triangleq \sum_{k=1}^{K} \sum_{i \neq j} A_{k,i,j} \cdot \eta\left(\hat{F}_k\left(t_i \mid \mathbf{x}_i\right), \hat{F}_k\left(t_i \mid \mathbf{x}_j\right) \right) \tag{2}$$

where $\eta(x, y)$ is a convex penalty function (e.g., exponential) that penalizes incorrect risk orderings.

The ranking penalty renders HLRanking improper. Note however that tuning on IBS selects α values which downplay the ranking penalty and pulls HLRanking towards $\mathcal{L}_{HL}$ (Appendix Tab. C.3).

CompetingRisksNegativeLogLikelihood Loss The following loss is designed to directly optimize the CIFs, and can be seen as the negative log-likelihood, re-weighted by the censoring distribution using Inverse Probability of Censoring Weighting (IPCW), which up-weights the subjects still under observation so that they stand in for those already lost to follow-up. This is a strictly proper scoring rule (Alberge et al., 2025) under the assumption that the censoring weights are correctly specified.

Definition 2.3 (CompetingRisksNegativeLog-Likelihood (CRNLL)). $\forall \tau, (\mathbf{x}, t, \delta) \sim \mathcal{D}$,

$$\mathcal{L}_{CRNLL,\tau}\left(\left(\hat{F}_1(\tau \mid \mathbf{x}), \ldots, \hat{F}_K(\tau \mid \mathbf{x}), \hat{S}(\tau \mid \mathbf{x})\right), (t, \delta)\right) \triangleq \frac{1}{n}\sum_{i=1}^{n}\left[\sum_{k=1}^{K}\frac{\mathbb{1}_{t_i \leq \tau, \delta_i = k}\log\left(\hat{F}_k(\tau \mid \mathbf{x}_i)\right)}{G^\star(t_i \mid \mathbf{x}_i)} + \frac{\mathbb{1}_{t_i > \tau}\log\left(\hat{S}(\tau \mid \mathbf{x}_i)\right)}{G^\star(\tau \mid \mathbf{x}_i)}\right] \quad (3)$$

where $G^\star$ is the complement of the CDF of the censoring distribution, i.e. the probability of not being censored for a given sample until the time t_i or τ.

2.3 Metrics: probabilistic accuracy, discrimination, calibration

Competing-risks predictions must be right about both which event occurs first and when, under right-censoring. Following Qi et al. (2024b), we study six evaluation metrics along three axes, capturing different aspects of predictive performance; Appendix B defines each one of them.

Probabilistic accuracy: the Integrated Brier Score (Graf et al., 1999), the time-integrated squared error between the predicted cumulative incidence and the observed outcome, and the Integrated Negative Log-Likelihood, its log-loss analogue; both are strictly proper scoring rules under the assumptions discussed next. *Discrimination:* the competing-risks C-index (Harrell Jr et al., 1996; Antolini et al., 2005; Uno et al., 2011; Wolbers et al., 2009), measuring whether patients are ranked in the right order, which we evaluate at several time horizons since the ranking may change in time; and the cumulative/dynamic time-dependent AUC (Uno et al., 2007; Blanche et al., 2013). *Calibration:* the competing-risks extensions of D-calibration (Haider et al., 2020) and of Aalen–Johansen calibration (Haider et al., 2020; D'Agostino & Nam, 2003) introduced by Alberge et al. (2026), which detect systematic over- or under-estimation of a cause's cumulative incidence.

Prioritizing the IBS/INLL We focus on the Integrated Brier Score rather than the C-index, for two reasons. First, the IBS is a strictly proper scoring rule (Gneiting & Raftery, 2007; Rindt et al., 2022; Yanagisawa, 2023) under two assumptions: censoring is independent of covariates and that the censoring distribution is correctly estimated by the inverse probability of Censoring Weighting (IPCW). It is then minimized if and only if the predicted distribution matches the true underlying distribution.The C-index, in contrast, has drawn criticism (Rindt et al., 2022; Lillelund et al., 2026): being a pure ranking metric it is blind to the magnitude of the predicted risks, and Rindt et al. (2022) show it can score an incorrect distribution above the oracle, making it improper.

3 Related work: many methods, few controlled comparisons

Benchmarks Standard survival analysis has seen several benchmarks in the past few years (Yanagisawa, 2023; Jeanselme et al., 2024; Wang et al., 2024; Mesinovic & Zhu, 2025; Noroozizadeh et al., 2026), supported by libraries packaging datasets (Mesinovic & Zhu, 2025; Kvamme et al., 2019; Nagpal et al., 2022; Drysdale, 2022) or metrics (Qi et al., 2024a; Pölsterl, 2020), and reviewed by Wiegrebe et al. (2024). Closest to us, Burk et al. (2026) neutrally compare 19 single-event models over 34 datasets, tuning on both a ranking measure and a scoring rule as we do, and Kopper et al. (2024) vary the scoring rule used for *training* single-event models, reporting only minor differences between rules. Competing risks has been studied far less, for want of easily obtained datasets: SurvTRACE (Wang & Sun, 2022), for instance, is benchmarked on a single real dataset. Reviews (Djangang et al., 2025) and benchmarks (Van Geloven et al., 2022; Bolós Sierra et al., 2026) have begun to fill the gap, though on few datasets and without varying the loss.

Modern Methods Methods range from the Aalen–Johansen estimator (Aalen & Johansen, 1978) to support vector machines (Van Belle et al., 2011), Gaussian processes (Aala & Van Der Shaar, 2017), trees (Ishwaran et al., 2014; Bellot & Schaar, 2018), neural networks (Lee et al., 2018; Nagpal et al., 2021; Katzman et al., 2018) and transformers (Wang & Sun, 2022; Ries & Seth, 2025). Deep competing-risks methods each introduce a network together with a likelihood-type loss of their own (Lee et al., 2018; Nagpal et al., 2021; Danks & Yau, 2022; Jeanselme et al., 2023); none compares losses across backbones, which is the comparison we run.

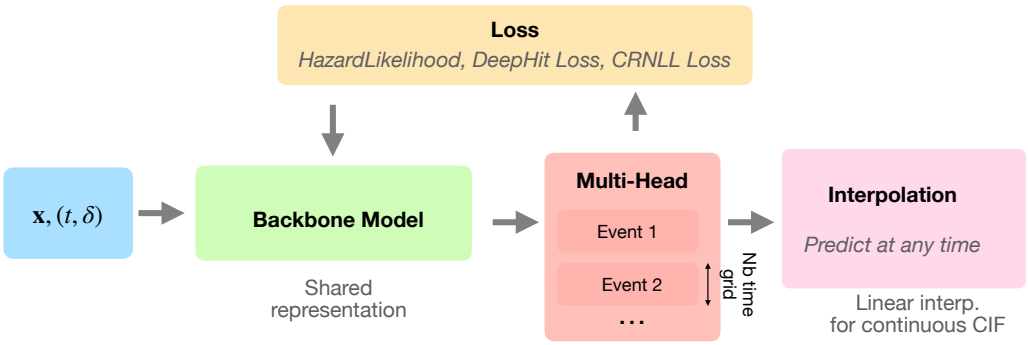


Figure 1: **Any backbone becomes a competing-risks model with two changes.** One prediction head per cause, plus a survival head for losses that model the cumulative incidence directly, and a discretized time grid at the output. Everything upstream is left as the original architecture, which is what lets us vary loss and backbone independently.

4 Building blocks: one adaptation that fits every backbone

We adapt a neural network backbone to the CR setting with the minimum changes, shown in Fig. 1.

Multi-Head Architecture: Competing risks need one incidence curve per cause. The backbone becomes a shared representation learner branching into one head per curve. Hazard-based losses use K heads, and the overall survival function follows by integrating the hazards over time, as in DeepHit and its pycox implementation (Kvamme et al., 2019). For the CRNLL loss, which models the CIF directly, we use $K + 1$ heads followed by a softmax activation to ensure valid, normalized probability distributions across all events for each sample. Because this additional survival head is a parameterization choice rather than a property of the CRNLL loss itself, we run an ablation isolating the two: CRNLL trained with the extra $(K + 1)$-th head against CRNLL trained with only K heads (renormalized without a dedicated survival output), on all four backbones. Removing the head shifts neither IBS nor calibration systematically on any backbone, so what we report is an effect of the loss, not of the output parameterization (Appendix D.6).

Time-Grid Discretizations and Continuous Inference: Training uses a discrete time grid built from observed event times, excluding censored ones; its resolution is a tuned hyperparameter (20 to 60 points, Appendix Tab. C.2). At inference we interpolate linearly between grid points.

Specificity for each loss For CRNLL, we use the marginal Kaplan & Meier (1958) estimator for the training-time IPCW, rather than a covariate-conditional Cox estimator, which a sensitivity analysis (Sec. C.3) found fragile: it converges on only a dataset-dependent subset of our benchmark, and whenever its IBS differs from the Kaplan-Meier one, it is always worse, never better.

Backbone model We train every loss on four backbones, each adapted as in Fig. 1: a plain 3-layer MLP, which is DeepHit's own architecture (Lee et al., 2018) and which we call MLP throughout; RealMLP (Holzmüller et al., 2024), an MLP tuned for tabular data; TabM (Gorishniy et al., 2025), which imitates an ensemble of MLPs; and the SurvTRACE transformer (Wang & Sun, 2022). RealMLP and TabM are the two best non-foundation models on TabArena (Erickson et al., 2025).

Vision backbones The adaptation of Fig. 1 is not specific to tabular data, so we apply it unchanged to image inputs. We use three standard pre-trained architectures (AlexNet (Krizhevsky et al., 2012), MobileNetV2 (Howard et al., 2017) and ResNet50 (He et al., 2016)) from `torchvision` (TorchVision maintainers and contributors, 2016) with their default weights. To isolate the effect of the competing-risks adaptation from that of representation learning, we freeze the backbone parameters and use the networks as fixed feature extractors, optimizing only the heads. We also carry the tabular MLP over unchanged, which tests whether a backbone has to match the data.

5 EXPERIMENTS: 23 DATASETS, EQUAL TUNING BUDGET IN EVERY CELL

5.1 DATASETS: 10 COMPETING-RISKS, 10 SINGLE-EVENT, 3 IMAGE

Tabular data We benchmark on 10 competing-risks tabular datasets with two or three competing events and, since a competing-risks model handles standard survival analysis as the case $K = 1$ while single-event datasets are far more plentiful, on 10 single-event datasets as well, matching the two counts. They serve as a contrast as much as a supplement: the same losses and backbones on a problem without competing causes isolate what the competing-risks structure itself contributes (Appendix D.1). All have more than 1,000 samples. This collection is the larger of the two because most competing-risks data is tabular, clinical registries and cohort studies above all. Table 2 summarises them; Appendix A gives for each one how it was obtained and preprocessed together with its full characteristics, and Appendix A.2 states the research question and censoring definition adopted for MIMIC-III/IV (Johnson et al., 2024; 2016) and eICU (Pollard et al., 2018).

Image data Beyond the tabular collection, we assembled three competing-risks datasets from TCGA whole-slide pathology images (Lu et al., 2021; Kandoth et al., 2013), preprocessed following Liu et al. (2024), giving 635 images. Combining these with the clinical records from cbioportal[1], we defined two competing events on each of them: death from cancer and from other causes.

5.2 TABULAR BASELINES: NINE METHODS WITH THEIR OWN LOSSES

The four backbones described above (MLP, RealMLP, TabM, SurvTRACE) are each trained under all three losses of Sec. 2.2, forming our controlled loss $\times$ architecture matrix; the hyperparameter grids are in Appendix Tab. C.2. We also include baseline models with different learning paradigms, each trained with its own original, fixed loss rather than as part of that matrix.

Aalen–Johansen (Aalen & Johansen, 1978): the covariate-free, population-level estimator of the CIFs; the reference of all our comparisons.

Fine and Gray (Fine & Gray, 1999): a linear model of the subdistribution hazard, under proportional hazards.

Random Survival Forests (Ishwaran et al., 2014): random forests adapted to competing risks.

SurvivalBoost (Alberge et al., 2025): gradient-boosted trees trained with the CRNLL loss and a feedback loop on the censoring model.

DeepHit (Lee et al., 2018): a neural network trained with the HLRanking loss (Def. 2.2).

Deep Survival Machines (DSM) (Nagpal et al., 2021): a mixture of parametric distributions (e.g. Weibull) for the CIFs, fit by maximum likelihood.

[1] https://www.cbioportal.org

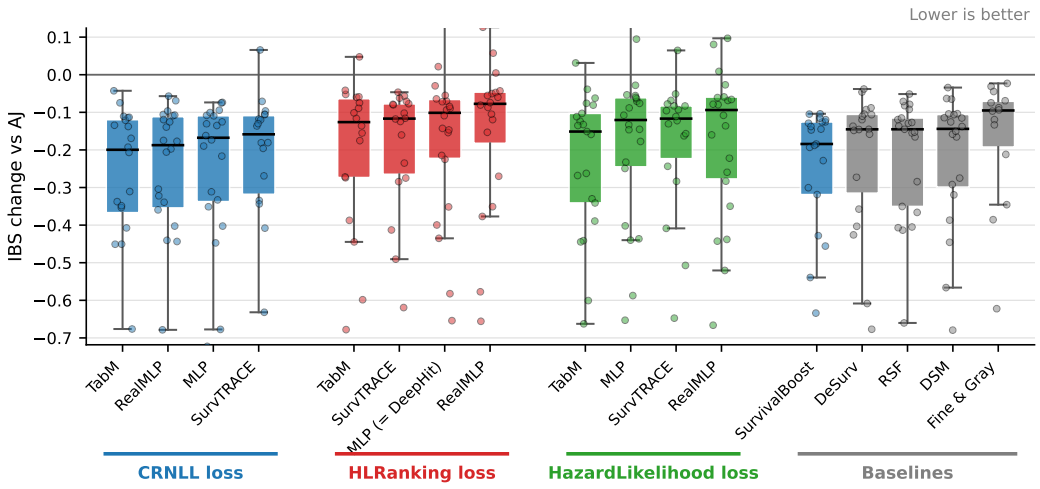
Table 2: **Our 20 tabular survival datasets, spanning three orders of magnitude in sample size and censoring rates from 19% to 95%.** The three image datasets are described in Sec. 5.1. Appendix A gives the full characteristics and the access procedure for each one.

Competing Risks

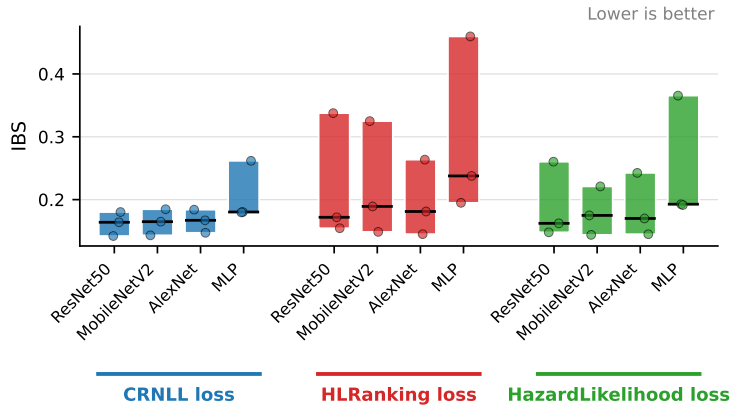
Dataset	Samples	Cens.
Framingham	4,240	58%
Metabric (CR)	3,237	33%
MIMIC IV	65,366	58%
MIMIC III	38,511	54%
SEER Breast	476,746	63%
SEER Digestion	887,552	19%
PBC	1,945	55%
Loan	1,652,304	41%
eICU	106,196	95%
TCGA	10,540	70%

Survival Analysis

Dataset	Samples	Cens.
Flchain	6,524	70%
Hdfail	52,422	95%
GBSG	2,232	43%
Metabric	1,904	42%
MSK 2021	25,659	60%
MSK 2026	40,755	57%
NWTCO	4,028	85%
Smarto	3,873	88%
Support	8,873	32%
ENS Challenge	3,173	50%



(a) **20 tabular datasets**: IBS gain over Aalen–Johansen, higher is better. Four backbones per loss, plus nine baselines carrying their own fixed losses. One box per model, one point per dataset, box spanning the interquartile range. SurvivalBoost is a non-differentiable baseline that uses the CNRLL loss.



(b) **3 TCGA pathology-image datasets**: raw IBS, lower is better. Three frozen pretrained encoders and the tabular MLP, no external baselines.

Figure 2: **A competitive competing-risks model needs a backbone matched to the modality *and* a loss that models cumulative incidence under censoring; neither substitutes for the other.** Both panels are grouped and coloured by loss, with the backbone named beneath each box. In both modalities CRNLL comes first within every backbone. What changes is which backbone wins: on tables the backbones built for tables, RealMLP and TabM, lead the plain MLP, while on images the pretrained encoders lead it by a wide margin and the MLP is also the least stable. The two panels have different backbones, baselines and vertical axes, and are not to be compared cell by cell.

DeSurv (Danks & Yau, 2022): each cause's CDF as the integral of a neural network, valid by construction, fit by maximum likelihood.

Neural Fine-Gray (Jeanselme et al., 2023): Fine and Gray with monotonic neural networks, relaxing proportional hazards while keeping valid CIFs.

SurvTRACE (Wang & Sun, 2022): a transformer whose published code implements only the 'naive' loss, i.e. HazardLikelihood (Def. 2.1).

5.3 Evaluation strategy: two hyperparameter searches per cell

Each dataset is randomly split into three folds (train + validation sets: 70%, and test set: 30%) and each split is capped at 50,000 samples. Each method is trained using `RandomizedSearchCV`

from scikit-learn (Pedregosa et al., 2011) with 150 iterations. Selecting on validation IBS alone would leave models undertuned for the C-index, which is one measure of our benchmark. We therefore run the search *twice* in every cell (pair of loss & architecture), once tuning for IBS and once tuning for C-index, and evaluate the two selections separately. This matters most for HLRanking, whose weight α between likelihood and ranking and ranking temperature σ are searched with the other hyperparameters (Tab. C.2): the C-index search is free to lean on the ranking term. IBS is reported as a gain over the Aalen–Johansen estimator, $1 - \mathrm{IBS}/\mathrm{IBS}_{\mathrm{AJ}}$, putting datasets on a common scale. After choosing the hyperparameters, we evaluate the models on 5 train/test splits.

6 Results: the loss and the backbone each matter

Trends across the two modalities Figure 2 gives results for tabular (top panel) and image (bottom panel) data, for IBS. The two panels have different backbones and different baselines. But they show common trends. They agree on the loss: in both modalities, and within every backbone, modeling the cumulative incidence directly under censoring weighting comes first. Where they differ is in which backbone wins, and this difference is itself informative. On tables the backbones designed for tables lead, RealMLP and TabM ahead of the plain MLP; on images the pretrained encoders lead and that same MLP is last, and least stable. Neither choice stands in for the other: a good loss on a mismatched backbone, and a matched backbone with a weaker loss, are both beaten by the pair.

Tabular data: the loss is decisive with competing risks We compare the effect of loss versus backbone: the spread across losses holding the backbone fixed, and the converse (Appendix D.1). Here we observe different trends in competing-risks versus single-event survival. On the 10 competing-risks datasets the loss effect is 0.053 against 0.027 for the backbone, so the loss moves the Brier score about twice as much. On the 10 single-event datasets the order reverses, 0.072 against 0.086, the two axes being comparable and the backbone marginally ahead. Pooled across the two families of datasets, the effects average out (Tab. D.1). The loss therefore matters most in competing risks settings; modeling jointly the several mutually-exclusive causes to censoring makes getting it right decisive. Table D.4 shows the same contrast cell by cell: on competing risks CRNLL leads the nearest hazard-based loss by a wide margin on every backbone, whereas on single-event data the three losses fall closer together, in line with Kopper et al. (2024). On competing risks, swapping DeepHit's loss for CRNLL on DeepHit's own MLP more than doubles its median IBS gain, from 0.12 to 0.26.

Figure 2a shows the distribution, across the 20 datasets, of the IBS gain over the Aalen–Johansen estimator, the covariate-free empirical incidence curve; the competing-risks datasets alone (Fig. D.7) and the most heavily censored ones (Fig. D.8) put the losses in the same order, with wider margins. By IBS, CRNLL comes first among the losses: it beats each hazard-based loss on all 20 datasets (median over backbones; sign test $p < 10^{-6}$) and in about 90% of (dataset, backbone) cells, while the two hazard-based losses tie. That tie is not a coincidence: tuning down-weights the ranking term (Sec. 2.2), so HLRanking resembles HazardLikelihood. Properness does not account for the ordering either, since CRNLL and HazardLikelihood are both strictly proper yet sit far apart: what sets CRNLL apart is that it models the cumulative incidence directly and reweights for censoring, rather than using cause-specific hazards. Properness matters for a metric to give a sharp picture (Appendix D.2), but does not suffice to make a loss competitive. Once the loss is fixed, architecture still matters, as the gap between RealMLP and MLP within the CRNLL group shows. On tabular data the loss is crucial, though it needs a *tabular* backbone.

Pathology images: the backbone decides, the loss still matters Here the twelve cells are the three losses on three pretrained vision encoders and on the same MLP (Fig. 2b). There are no external baselines as competing risks has been less studied for images. The backbone now separates the models far more than on tables: every pretrained encoder beats the MLP under every loss, and the MLP's spread across datasets is several times wider, so a tabular-style backbone on images is not merely worse but unreliable. The ordering of the losses is nevertheless the one we measured on tables, CRNLL first within each backbone, then HazardLikelihood, then HLRanking, including for the MLP. It holds although the encoders are frozen: acting on the heads alone, the loss still orders the models.

C-index Neither loss nor architecture has visible effect on the C-index: every method performs alike (Fig. D.9a). Even HLRanking, built for this metric, gains only marginally at the 25th percentile and nothing at the 50th or 75th. All quantiles (Fig. D.9) and the integrated AUC (Fig. D.4) agree.

Does tuning HLRanking on its own target metric change the picture? Appendix Tab. D.5 compares HLRanking's own IBS and C-index (50th quantile) under the two selection criteria, paired at the (dataset, backbone) level. Pooled over the four backbones, selecting on C-index improves the C-index by a median of $+0.0018$ and degrades IBS by $+0.0117$: the gain is small and it is paid for dearly. This is the same finding seen from the other side. Left free, the search down-weights the ranking term; forced to favour concordance, it up-weights it (median $1 - \alpha$ from 0.41 to 0.71, Tab. C.3) and buys a negligible amount of C-index at a real cost in Brier score.

Calibration gives little signal Neither calibration criterion separates the losses usefully, and the two disagree: CRNLL is best calibrated on AJ-calibration (Fig. D.5) and trails the hazard-based losses on D-calibration (Fig. D.6). The likely reason is that there are multiple ways of defining calibration for competing risks, all imperfectly capturing errors to true probability, and all difficult to estimate. We report both AJ and D calibration for completeness but draw no conclusion from them.

7 Conclusion: backbone for the modality, loss for the censoring

Limitations *Not pure competing-risks.* Half of the 20 tabular datasets are single-event ($K = 1$). The ordering of the losses is the same on either family (Fig. D.7), but the effect sizes are not, and we report the two apart wherever that matters (Appendix D.1). *Sample size.* Splits are capped at 50,000 training samples, so the largest datasets never exercise the large-data regime.

Combining loss and backbone for the win We presented a large-scale benchmark for survival analysis with competing risks, spanning loan default, long-term medical follow-up and ICU outcomes. On tabular data, the cumulative-incidence loss (eq. 3) is the best of the three losses on every backbone and, pooling backbones, on all 20 datasets; it is worth about twice the backbone where competing causes are present, against a comparable weight on single-event data. Proper losses help single-event models (Yanagisawa, 2023), but properness alone does not explain this, since the hazard likelihood is proper too: what matters is modeling the cumulative incidence directly under censoring weighting. The same loss still leads on pathology images, so its advantage is not an artifact of the tabular regime. The backbone, in turn, has to match the modality: the MLP that is competitive on tables is the worst and least stable model on images. A practitioner should thus pick the backbone for the data and the loss for the censoring. Adapting the loss to censoring may seem obvious, yet recent deep competing-risks methods have invested in dedicated architectures and output parameterizations (Lee et al., 2018; Nagpal et al., 2021; Danks & Yau, 2022; Wang & Sun, 2022; Jeanselme et al., 2023; Ries & Seth, 2025), perhaps because several incidence curves seem to call for a dedicated network, and none builds on recent tabular architectures. Our results argue for undoing this coupling: one head per cause on a generic backbone suffices, and the gains lie in the loss. Pairing RealMLP or TabM, built for tabular prediction, with the cumulative-incidence loss beats every published neural competing-risks model on the Brier score. Advances in backbones will keep arriving from outside this field; they transfer only once paired with a loss that suits the task, and establishing whether they do calls for rigorous, many-dataset benchmarks such as this one. We release its code, dataset loaders and search spaces, so that a new backbone or loss can be scored under the same protocol.

Only strictly proper scoring rules cleanly separated the models Of our six evaluation metrics (Appendix D.2), only the two strictly proper scoring rules gave clear signals: the Brier score and the log-likelihood order the models consistently and alike. The ranking metrics, C-index and AUC, move for nothing, not even for the loss built for concordance, tuned on it; the calibration criteria disagree. Theory agrees: the C-index is not proper (Rindt et al., 2022; Lillelund et al., 2026) and the AUC not strictly so, so gaining on either need not mean a better distribution. We recommend comparing models on a strictly proper scoring rule, keeping the rest as diagnostics. Applications that need only a ranking lose nothing by it: our best loss is competitive on the C-index anyway.

AI USE STATEMENT

We used generative AI tools (Claude) to assist with debugging and drafting analysis/plotting scripts for the benchmark codebase, and to assist with drafting, editing, and proofreading parts of this manuscript. We proofed and manually reworked all the manuscript. We did not use generative AI tools to generate research ideas, to run experiments unsupervised, or to write results we did not independently verify against the underlying score files. All AI-assisted code and text were reviewed and verified by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses de-identified clinical, financial, and engineering datasets, several of which (MIMIC-III/IV, eICU, SEER) require credentialed access and a signed data use agreement, and others of which (Kaggle Loan, ENS Challenge) require registration; none were redistributed as part of this submission. Appendix A.1 lists the access procedure, and where applicable the licensing/redistribution status, for every dataset used. We do not foresee a specific harmful application of this benchmark beyond the general risks of clinical risk-prediction models (e.g. miscalibrated absolute-risk estimates being used for decision-making); our results in fact argue for evaluating and selecting such models on strictly proper scoring rules, which reward accurate absolute risks, rather than on discrimination alone, which we believe mitigates rather than adds to this risk.

REPRODUCIBILITY STATEMENT

An anonymized version of the full benchmark codebase – loss implementations, backbone architectures, hyperparameter search spaces, evaluation logic, and dataset preprocessing/splitting – is available at https://anonymous.4open.science/r/loss_survival_analysis-4D3B/README.md and will be released on GitHub after publication. Section A and Appendix A.1 detail how each dataset was obtained and preprocessed, including access requirements for credentialed sources. Appendix E gives the full derivations for every loss and calibration metric used in the main text, and Tab. C.2 reports the hyperparameter search space and compute budget for every model.

REFERENCES

A. Aala and M Van Der Shaar. Deep multi-task gaussian processes for survival analysis with competing risks. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Odd O. Aalen and SOren Johansen. An Empirical Transition Matrix for Non-Homogeneous Markov Chains Based on Censored Observations. *Scandinavian Journal of Statistics*, 5(3):141–150, 1978. ISSN 0303-6898. Publisher: [Board of the Foundation of the Scandinavian Journal of Statistics, Wiley].

Julie Alberge, Vincent Maladière, Olivier Grisel, Judith Abécassis, and Gaël Varoquaux. Survival Models: Proper Scoring Rule and Stochastic Optimization with Competing Risks. *AIStats*, 2025.

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

Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for survival data. *Statistics in Medicine*, 24(24):3927–3944, December 2005. ISSN 0277-6715, 1097-0258.

S. S. Arslan and E. Zeydan. Hard drive failure data. Kaggle, 2019. URL https://www.kaggle.com/datasets/ezeydan/hard-drive-failure-data.

Alexis Bellot and Mihaela Schaar. Tree-based Bayesian Mixture Model for Competing Risks. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pp. 910–918. PMLR, March 2018. ISSN: 2640-3498.

Paul Blanche, Jean-François Dartigues, and Hélène Jacqmin-Gadda. Estimating and comparing time-dependent areas under receiver operating characteristic curves for censored event times with competing risks. *Statistics in Medicine*, 2013.

Begoña Bolós Sierra, Colin McLean, Sarah Friedrich-Welz, Peter S. Hall, and Catalina A. Vallejos. A reproducible and extensible framework for benchmarking competing risks survival models. *arXiv preprint arXiv:2608.00271*, 2026.

Lukas Burk, John Zobolas, Bernd Bischl, Andreas Bender, Marvin N. Wright, and Raphael Sonabend. A large-scale neutral comparison study of survival models on low-dimensional data. *arXiv preprint arXiv:2406.04098*, 2026.

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

R.B. D'Agostino and Byung-Ho Nam. Evaluation of the Performance of Survival Analysis Models: Discrimination and Calibration Measures. In *Handbook of Statistics*, volume 23, pp. 1–25. Elsevier, 2003. ISBN 978-0-444-50079-3.

Dominic Danks and Christopher Yau. Derivative-based neural modelling of cumulative distribution functions for survival analysis. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*. PMLR, 2022.

Angela Dispenzieri, Jerry A Katzmann, Robert A Kyle, Dirk R Larson, J Benson, C Mullaney, and et al. Detection of monoclonal gammopathy of undetermined significance (mgus) in serum free light chain assays. *Blood*, 116(12):2117–2120, 2010.

Paul M Djangang, Summer S Han, and Nilotpal Sanyal. Comparative review of modern competing risk methods in high-dimensional settings. *arXiv preprint arXiv:2503.12824*, 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 Mutalik Desai, David Salinas, and Frank Hutter. TabArena: A living benchmark for machine learning on tabular data. In *Advances in Neural Information Processing Systems 38, Datasets and Benchmarks Track*, 2025. arXiv:2506.16791.

Jason P. Fine and Robert J. Gray. A Proportional Hazards Model for the Subdistribution of a Competing Risk. *Journal of the American Statistical Association*, 94(446):496–509, June 1999. ISSN 0162-1459, 1537-274X.

Thomas R Fleming and David P Harrington. *Counting Processes and Survival Analysis*. Wiley, New York, 1991. ISBN 978-0471522188.

Tilmann Gneiting and Adrian E Raftery. Strictly Proper Scoring Rules, Prediction, and Estimation. *Journal of the American Statistical Association*, 102(477):359–378, March 2007. ISSN 0162-1459, 1537-274X.

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

Erika Graf, Claudia Schmoor, Willi Sauerbrei, and Martin Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in Medicine*, 18(17-18):2529–2545, September 1999. ISSN 0277-6715, 1097-0258.

Zoe Guan. A proper scoring rule for validation of competing risks models. *arXiv preprint arXiv:2104.01000*, 2021.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *Journal of Machine Learning Research*, 21(85):1–63, 2020.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

David Holzmüller, Léo Grinsztajn, and Ingo Steinwart. Better by default: Strong pre-tuned mlps and boosted trees on tabular data. *Advances in Neural Information Processing Systems*, 37: 26577–26658, 2024.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 2261–2269. IEEE, 2017.

Hemant Ishwaran, Thomas A. Gerds, Udaya B. Kogalur, Richard D. Moore, Stephen J. Gange, and Bryan M. Lau. Random survival forests for competing risks. *Biostatistics*, 15(4):757–773, October 2014. ISSN 1465-4644.

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

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

A. Johnson, T. Pollard, and R. Mark. MIMIC-III Clinical Database (version 1.4). *PhysioNet*, 2016.

A. Johnson, L. Bulgarelli, T. Pollard, B. Gow, B. Moody, S. Horng, L. A. Celi, and R. Mark. MIMIC-IV (version 3.1). *PhysioNet*, 2024.

Cyriac Kandoth, Michael D McLellan, Fabio Vandin, Kai Ye, Beifang Niu, Charles Lu, Mingchao Xie, Qunyuan Zhang, Joshua F McMichael, Matthew A Wyczalkowski, et al. Mutational landscape and significance across 12 major cancer types. *Nature*, 502(7471):333–339, 2013.

E. L. Kaplan and Paul Meier. Nonparametric Estimation from Incomplete Observations. *Journal of the American Statistical Association*, 53(282):457–481, June 1958. ISSN 0162-1459, 1537-274X.

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.

Philipp Kopper, David Rügamer, Raphael Sonabend, Bernd Bischl, and Andreas Bender. On training survival models with scoring rules. *arXiv preprint arXiv:2403.13150*, 2024.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pp. 1097–1105. 2012.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-Event Prediction with Neural Networks and Cox Regression. *Journal of Machine Learning Research*, 20:1–30, 2019.

Håvard Kvamme and Ornulf Borgan. Continuous and Discrete-Time Survival Prediction with Neural Networks, October 2019. arXiv:1910.06724 [cs, stat].

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar. DeepHit: A Deep Learning Approach to Survival Analysis With Competing Risks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468, 2159-5399.

Christian Marius Lillelund, Shi-ang Qi, Russell Greiner, and Christian Fischer Pedersen. Position: Stop chasing the c-index when evaluating survival analysis models. In *Proceedings of the 43rd International Conference on Machine Learning (ICML), Position Papers*, 2026. arXiv:2506.02075.

Pei Liu, Luping Ji, Xinyu Zhang, and Feng Ye. Pseudo-bag mixup augmentation for multiple instance learning-based whole slide image classification. *IEEE Transactions on Medical Imaging*, 43(5): 1841–1852, 2024.

Ming Y Lu, Drew FK Williamson, Tiffany Y Chen, Richard J Chen, Matteo Barbieri, and Faisal Mahmood. Data-efficient and weakly supervised computational pathology on whole-slide images. *Nature biomedical engineering*, 5(6):555–570, 2021.

Lara Mangravite. Metabric data for use in independent research, 2013.

Munib Mesinovic and Tingting Zhu. Survbench: A standardised preprocessing pipeline for multi-modal electronic health record survival analysis. *arXiv 2511.11935*, 2025.

Chirag Nagpal, Xinyu Rachel 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. doi: 10.1109/JBHI.2021.3052441.

Chirag Nagpal, William Potosnak, and Aleksandar Dubrawski. auton-survival: An open-source package for regression, counter-factual estimation, evaluation and phenotyping with censored time-to-event data. In *Machine Learning for Healthcare Conference*, pp. 585–608, 2022.

Cancer Genome Atlas Research Network, John N. Weinstein, Eric A. Collisson, Gordon B. Mills, Kenna R. Mills Shaw, Brad A. Ozenberger, Kyle Ellrott, Ilya Shmulevich, Chris Sander, and Joshua M. Stuart. The cancer genome atlas pan-cancer analysis project. *Nature Genetics*, 45(10): 1113–1120, 2013.

Shahriar Noroozizadeh, Xiaobin Shen, Jeremy C. Weiss, and George H. Chen. SurvHTE-Bench: A benchmark for heterogeneous treatment effect estimation in survival analysis. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2603.05483.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Melania Pintilie. *Competing Risks: A Practical Perspective*. Wiley, Chichester, England, 2006. ISBN 978-0-470-87068-6.

Tom J. Pollard, Alistair E. W. Johnson, Jesse D. Raffa, Leo A. Celi, Roger G. Mark, and Omar Badawi. The eICU Collaborative Research Database, a freely available multi-center database for critical care research. *Scientific Data*, 5(1):180178, September 2018. ISSN 2052-4463.

Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

Shi-ang Qi, Weijie Sun, and Russell Greiner. SurvivalEVAL: A Comprehensive Open-Source Python Package for Evaluating Individual Survival Distributions. *Proceedings of the AAAI Symposium Series*, 2(1):453–457, 2024a.

Shi-ang Qi, Yakun Yu, and Russell Greiner. Toward conditional distribution calibration in survival prediction. In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2024b. arXiv:2410.20579.

Maxmillan Ries and Sohan Seth. Tracer: Transformer-based competing risk analysis with longitudinal covariates, 2025. URL https://arxiv.org/abs/2512.18129.

David Rindt, Robert Hu, David Steinsaltz, and Dino Sejdinovic. Survival regression with proper scoring rules and monotonic neural networks. In *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 151 of *Proceedings of Machine Learning Research*. PMLR, 2022.

Oscar M Rueda, Stephen-John Sammut, Jose A Seoane, Suet-Feung Chin, Jennifer L Caswell-Jin, Maurizio Callari, Rajbir Batra, Bernard Pereira, Alejandra Bruna, H Raza Ali, et al. Dynamics of breast-cancer relapse reveal late-recurring er-positive genomic subgroups. *Nature*, 567(7748): 399–404, 2019.

SEER. SEER database. https://seer.cancer.gov/causespecific, 2025.

TorchVision maintainers and contributors. TorchVision: PyTorch's computer vision library. https://github.com/pytorch/vision, 2016.

Connie W Tsao and Ramachandran S Vasan. The framingham heart study: past, present and future. *International Journal of Epidemiology*, 44(6):1763–1766, 12 2015.

Hajime Uno, Tianxi Cai, Lu Tian, and L. J. Wei. Evaluating prediction rules for t-year survivors with censored regression models. *Journal of the American Statistical Association*, 102(478):527–537, 2007.

Hajime Uno, Tianxi Cai, Michael J. Pencina, Ralph B. D'Agostino, and L. J. Wei. On the C-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in Medicine*, 30(10):1105–1117, May 2011. ISSN 0277-6715, 1097-0258.

Vanya Van Belle, Kristiaan Pelckmans, Sabine Van Huffel, and Johan A.K. Suykens. Support vector methods for survival analysis: a comparison between ranking and regression approaches. *Artificial Intelligence in Medicine*, 53(2):107–118, October 2011. ISSN 09333657.

Ben Van Calster, David J McLernon, Maarten Van Smeden, Laure Wynants, Ewout W Steyerberg, and Topic Group 'Evaluating diagnostic tests and prediction models' of the STRATOS initiative. Calibration: the achilles heel of predictive analytics. *BMC medicine*, 17(1):230, 2019.

Nan Van Geloven, Daniele Giardiello, Edouard F Bonneville, Lucy Teece, Chava L Ramspek, Maarten Van Smeden, Kym I E Snell, Ben Van Calster, Maja Pohar-Perme, Richard D Riley, Hein Putter, and Ewout Steyerberg. Validation of prediction models in the presence of competing risks: a guide through modern methods. *BMJ*, pp. e069249, May 2022. ISSN 1756-1833.

Zifeng Wang and Jimeng Sun. SurvTRACE: Transformers for Survival Analysis with Competing Events. In *Proceedings of the 13th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pp. 1–9, August 2022. arXiv:2110.00855 [cs, stat].

Ziwen Wang, Jin Wee Lee, Tanujit Chakraborty, Yilin Ning, Mingxuan Liu, Feng Xie, Marcus Eng Hock Ong, and Nan Liu. Survival modeling using deep learning, machine learning and statistical methods: A comparative analysis for predicting mortality after hospital admission. *arxiv 2403.06999*, 2024.

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.

Marcel Wolbers, Michael T. Koller, Jacqueline C. M. Witteman, and Ewout W. Steyerberg. Prognostic Models With Competing Risks: Methods and Application to Coronary Risk Prediction. *Epidemiology*, 20(4):555–561, July 2009. ISSN 1044-3983.

Hiroki Yanagisawa. Proper scoring rules for survival analysis. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202 of *Proceedings of Machine Learning Research*. PMLR, 2023.

Appendices

We give detailed appendices to be precise on all definitions and help reproducibility.

A Details on Datasets **15**
A.1 Access and preprocessing . . . 15
A.2 Study on MIMIC III-IV, and eICU . . . 18
A.3 Statistics summarizing the datasets . . . 19

B Theory: Definitions of the evaluation metrics **19**
B.1 Metrics controlling probabilistic prediction metrics: IBS and INLL . . . 19
B.2 Discrimination metrics: C-index and time-dependent AUC . . . 20
B.3 Calibration metrics . . . 21

C Details on the experimental methods **23**
C.1 Competing Risks libraries . . . 23
C.2 Compute and Hyper parameter search . . . 23
C.3 Training-time IPCW sensitivity: Kaplan-Meier vs. Cox vs. stratified . . . 25

D Detailed Experimental Results **26**
D.1 Computing effect sizes: How much the loss matters, and where . . . 26
D.2 Comparing the different model-selection metrics . . . 27
D.3 Dataset stratification: multiple events or not, amount of censoring . . . 31
D.4 C-indexes at different time points . . . 33
D.5 HLRanking: tuning on its own target metric . . . 33
D.6 CRNLL survival-head ablation . . . 33

E Background theory: Link between the hazard function and the CIF function **35**

A Details on Datasets

A.1 Access and preprocessing

The following instructions cover the 20 tabular datasets in Table A.1. Paths are relative to the `benchmark/` directory of the accompanying code. The dataset-specific loaders are in `_dataset.py`, with the PBC preparation in `auton_datasets.py`. Each loader returns covariates and a target with columns `duration` and `event`; event code 0 denotes right-censoring. For prepared CSV files, obtaining the source data and constructing the cohort are separate steps.

Shared tabular preparation. The custom CSV loaders, the competing-risks METABRIC loader, and the SurvSet loader apply `skrub.TableVectorizer`, separate the target columns from the transformed covariates, and fill remaining missing covariate values with column means. They cast covariates to `float32` and events to integers. Rows with missing targets are removed where specified below. The pycox loaders and the SEER breast loader use their own column typing, described in their entries. The dataset-specific operations below precede the benchmark split and model fitting.

Framingham (Tsao & Vasan, 2015). The local `datasets/framingham.csv` has the same 4,240 rows and column names as the `cvdd` teaching dataset documented by `riskCommunicator`.[2] To obtain that public teaching dataset, install `riskCommunicator` from CRAN, load `data(cvdd, package="riskCommunicator")`, and export it with `write.csv(cvdd, "framingham.csv")`. This source already restricts observations to the first examination and excludes prevalent coronary heart disease. Our `load_framingham` uses `timeout` as the duration in days, codes CVD as event 1 and death without CVD as event 2, drops `logpdays`, `cvd_dth`,

[2] https://search.r-project.org/CRAN/refmans/riskCommunicator/html/cvdd.html

bmicat, and CURSMOKE, and removes missing targets before vectorization. The retained columns are listed in the loader.

METABRIC with competing events (Rueda et al., 2019). Open the supplementary information for Rueda et al. (2019)[3] and obtain 41586_2019_1007_MOESM7_ESM.txt. Place it at the location specified by PATH_METABRIC. The loader extracts diagnosis year from Date.Of.Diagnosis, normalizes column names, and retains 16 clinical covariates. It removes rows missing any of Death, T, TLR, TDR, LR, or DR. Event 1 denotes local or distant relapse, with duration initially set to min(TLR, TDR). Death is event 2 when Death=1 and T is no greater than that duration; the duration is then replaced by T. Other observations retain event code 0. We then apply the shared tabular preparation.

MIMIC-IV (Johnson et al., 2024). Open the PhysioNet page for version 3.1.[4] Create a PhysioNet account, obtain credentialed access, complete the required research training, and sign the dataset's data use agreement. Download the hospital patients, admissions, and labevents tables and the ICU icustays table. Construct the cohort using Section A.2 and save the prepared table as datasets/mimic_cohort.csv. The load_mimic function removes missing targets and applies the shared preparation to the remaining columns.

MIMIC-III (Johnson et al., 2016). Use the PhysioNet page for version 1.4[5] and complete its access requirements. Download PATIENTS, ADMISSIONS, ICUSTAYS, and LABEVENTS. Apply the cohort procedure described in Section A.2, using the MIMIC-III schema, and save datasets/mimiciii_cohort.csv. The load_mimiciii function removes missing targets and selects diagnosis, age, and the 50 laboratory variables explicitly listed in the code before vectorization and mean imputation.

eICU (Pollard et al., 2018). Request access to version 2.0 on PhysioNet.[6] The account and approved training used for MIMIC can be reused, but the eICU data use agreement must also be signed. Download the patient and lab tables and construct the cohort described in Section A.2. Save datasets/eicu_cohort.csv. The load_eicu function removes missing targets and retains gender, admission height and weight, cleaned age, and the laboratory columns listed in the loader before applying the shared preparation.

SEER breast cancer (SEER, 2025). Request SEER data access through the National Cancer Institute and follow the instructions for SEER*Stat.[7] Use the hazardous export tutorial[8] to obtain the tab-separated file expected by PATH_SEER, preserving its 29-column order. We call load_seer with survtrace_preprocessing=True. This removes records with the specified unavailable cause-specific classification or death before recommended surgery, recodes selected unknown values, and groups rare categories as implemented in hazardous/data/_seer.py. We retain 14 categorical and four numerical predictors. Duration is Survival months; events are breast cancer death, heart disease death, and other causes of death, coded 1, 2, and 3, respectively. Alive patients are censored.

SEER digestive cancers (SEER, 2025). Use the same SEER access procedure and export the digestive-cancer cohort in SEER*Stat, including survival months, cause of death, and clinical covariates. The benchmark reads the prepared file datasets/seer_cohort.csv through load_seer_digestion, removes missing event or duration, and applies the shared preparation to all remaining columns. Reproducing the raw-data extraction requires the original SEER*Stat session, including the database release, site filters, and event recoding; these are not recorded in this loader.

[3] https://doi.org/10.1038/s41586-019-1007-8
[4] https://physionet.org/content/mimiciv/3.1/
[5] https://physionet.org/content/mimiciii/1.4/
[6] https://physionet.org/content/eicu-crd/2.0/
[7] https://seer.cancer.gov/data/access.html
[8] https://soda-inria.github.io/hazardous/downloading_seer.html

PBC (Fleming & Harrington, 1991). Obtain `pbc2.csv` from the `auton-survival` dataset collection[9] and place it in `datasets/`. Use the benchmark's `load_pbc`, which calls the local `load_dataset_auton("PBC")`. The local helper keeps the repeated observations as separate rows, dummy-encodes seven categorical variables, retains seven laboratory variables, and derives age as `age + years`. Duration is `years - year`; transplantation is event 1 and death is event 2. It mean-imputes and standardizes the covariates. The outer loader removes missing targets and applies the shared preparation. Thus the sample count refers to observations, not distinct patients.

Loan. Register on Kaggle and download the Lending Club archive.[10] The benchmark uses a derived `datasets/loan_cohort.csv`. The `load_loan` function removes duplicate interest-rate and installment columns and `sub_grade`, converts the loan term from text to months, and converts employment length to a number, coding "10+ years" as 10 and "< 1 year" as 0. It renames `time_to_event` to `duration` and applies the shared preparation. The prepared file also includes four macroeconomic covariates. Its construction script is needed to reproduce the loan-status event mapping, time origin, censoring date, and macroeconomic joins; these operations are not performed by the loader.

TCGA (Network et al., 2013). Clinical TCGA data can be obtained through cBioPortal.[11] Our tabular loader expects the prepared `datasets/tcga.csv`, including the competing-event columns `DSS_cr` and `DSS.time.cr`. It renames these to `event` and `duration`, removes missing targets, and excludes patient barcodes, vital status, follow-up and death dates, cause of death, new-tumor-event descriptors, treatment outcome, margin status, and residual tumor. It then applies the shared preparation. A generic cBioPortal download does not establish how the two competing-event columns were derived: the original clinical export and endpoint-construction script are needed to reproduce this step. The results are based on data generated by the TCGA Research Network.[12]

FLCHAIN (Dispenzieri et al., 2010). Install `pycox`[13] and call `pycox.datasets.flchain.read_df(processed=False)`. Our `get_split_flchain` removes rows missing creatinine, drops `chapter` and `rownames`, and encodes sex as a male indicator. It treats `sample.yr` and `flc.grp` as categorical and casts the other columns to `float32`. The target uses `futime` for duration and `death` for the event indicator.

Hdfail (Arslan & Zeydan, 2019). Install `SurvSet`[14] and obtain the data with `SurvLoader().load_dataset("hdfail")["df"]` from `SurvSet.data`. The benchmark calls `load_survset("hdfail")`: it renames `time` to `duration`, drops rows missing a target and removes `pid`, then vectorizes and mean-imputes the remaining covariates. It retains the supplied failure indicator and follow-up times.

GBSG (Katzman et al., 2018). Obtain the packaged benchmark with `pycox.datasets.gbsg.read_df()`. Our `get_split_gbsg` retains its supplied `event` and `duration`, treats all seven covariates `x0`–`x6` as numerical, and casts them to `float64`. Use this packaged version to match the 2,232-row benchmark.

METABRIC, single event (Mangravite, 2013). Call `pycox.datasets.metabric.read_df()`. The benchmark retains the supplied overall-survival target and nine covariates, casting `x4`, `x5`, `x6`, and `x7` as categorical and the remaining five as `float64`. This is the 1,904-row pycox dataset; the relapse and death construction described above belongs to the separate competing-events dataset.

[9] https://github.com/autonlab/auton-survival/tree/master/auton_survival/datasets

[10] https://www.kaggle.com/datasets/wordsforthewise/lending-club

[11] https://www.cbioportal.org

[12] https://www.cancer.gov/tcga

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

[14] https://github.com/ErikinBC/SurvSet

MSK 2021. Open the cBioPortal study `msk_met_2021`[15] and download the clinical-data table as `datasets/msk_met_2021_clinical_data.tsv`. Use the clinical export with human-readable column names expected by `load_msk`. The loader converts `NA` to missing values, uses overall survival in months as duration, maps `0:LIVING` to censoring and `1:DECEASED` to event 1, and removes missing targets. The 14 retained covariates are listed in `get_split_msk_2021`; they include age at sequencing, cancer type, gene panel, mutation count, and sample characteristics. We then apply the shared preparation.

MSK 2026. Download the clinical-data table for cBioPortal study `msk_impact_50k_2026`[16] as `datasets/msk_impact_50k_2026_clinical_data.tsv`. We use the same overall-survival coding and missing-target removal as for MSK 2021. The 15 covariates in `get_split_msk_2026` include age at diagnosis, cancer type, tumor mutation burden, ploidy, purity, genomic alteration fraction, gene panel, ancestry, MSI score, mutation count, coverage, sex, FACETS quality control, and HLA genotype availability. The shared preparation follows this selection.

NWTCO. Call `pycox.datasets.nwtco.read_df(processed=False)`. The benchmark uses `edrel` as duration and `rel` as the relapse indicator. It creates `instit_2 = instit - 1`, `histol_2 = histol - 1`, and `study_4 = study - 3`; it drops the original three columns, `rownames`, and `seqno`. Stage is categorical, and the other covariates are cast to `float32`.

Smarto. Use `SurvLoader().load_dataset("smarto")["df"]` from `SurvSet.data`. The benchmark's `load_survset("smarto")` preserves the supplied event indicator, renames `time` to `duration`, removes missing targets and `pid`, and applies the shared preparation to the remaining covariates.

SUPPORT (Katzman et al., 2018). Call `pycox.datasets.support.read_df()` to obtain the packaged 8,873-row benchmark. We retain its supplied survival target and 14 covariates. The loader casts `x1`–`x6` as categorical and `x0`, `x7`–`x13` as `float64`.

ENS Challenge. Create an account on Challenge Data and download the labeled training files for QRT's "Overall Survival Prediction for patients diagnosed with Myeloid Leukemia."[17] The source provides clinical measurements, molecular records, and survival targets. Our `load_challenge_ens` reads `datasets/challenge_ens_cohort.csv`, removes missing targets, and applies the shared preparation. The prepared table contains center, six blood or marrow measurements, and the derived columns `num_mutated_genes`, `nb_of_chromosom`, and `sex_chromosome`. The original preparation script is needed to reproduce the molecular aggregation and cytogenetic parsing that created these columns.

A.2 Study on MIMIC III-IV, and eICU

For those ICU datasets, we applied the same methodology.

A.2.1 Cohort Selection and Inclusion Criteria

We constructed our cohort from the MIMIC-IV database. To ensure a focused analysis on critical care outcomes, we applied the following inclusion criteria:

- **ICU Admission:** Only patients with at least one recorded ICU stay (`icustays`) were included.
- **Adult Population:** Patients were filtered to include only those ≥ 18 years of age at the time of their anchor year.
- **Index Stay Selection:** We randomly sampled a single hospital admission (`hadm_id`) for each unique patient (`subject_id`).

[15] https://www.cbioportal.org/study/summary?id=msk_met_2021
[16] https://www.cbioportal.org/study/summary?id=msk_impact_50k_2026
[17] https://challengedata.ens.fr/challenges/162

A.2.2 Feature Engineering: Aggregated Clinical Labs

Due to the high dimensionality and irregular sampling of laboratory measurements, we processed the `labevents` table using a memory-efficient chunking strategy:

- **Dimensionality Reduction:** We identified the top 50 most frequent lab tests to represent the physiological state during the first day of the stay.
- **Aggregation:** For the selected index stay, all lab observations recorded between `admittime` and `admittime` + 1 day were aggregated. We computed the mean values for each test, resulting in a fixed-length feature vector per patient.

A.2.3 Target Definition: Competing Risks Framework

We treat the post-discharge outcome as a competing risks problem. Let T be the time to the first event following hospital discharge. Because MIMIC-IV dates are shifted for de-identification, we calculated all durations in relative days from the index discharge time.

We define three mutually exclusive outcomes:

- **Rehospitalization (Event Type 1):** Defined as the duration between the current index `admittime` + 1 day and the `admittime` of the subsequent hospital visit.
- **Death (Event Type 2):** Defined as the duration between the index `admittime` + 1 day and the date of death (dod) recorded in the patients table.
- **Administrative Censoring (Event Type 0):** To handle the lack of a global end-date, we applied individual-level administrative censoring. A patient is censored at the time of their last known interaction with the hospital system (the maximum discharge time across all their records).

A.3 Statistics summarizing the datasets

Table A.1 reports the dataset characteristics after the dataset-specific preparation described in Section A.1. For loaders using `TableVectorizer`, the feature count is measured after vectorization. The benchmark code is available at https://anonymous.4open.science/r/loss_survival_analysis/README.md and will be on GitHub after publication.

B Theory: Definitions of the evaluation metrics

This appendix gives the precise definition of every metric used in Sec. 2.3. Throughout, the model outputs a predicted CIF $\hat{F}_k(t \mid \mathbf{x})$ for each cause $k \in \{1, \dots, K\}$ and a predicted overall survival function $\hat{S}(t \mid \mathbf{x}) = 1 - \sum_{k=1}^{K} \hat{F}_k(t \mid \mathbf{x})$. Metrics are computed on a held-out test set $\mathcal{D}_{\text{test}} = \{(\mathbf{x}_i, t_i, \delta_i)\}_{i=1}^{n}$, with $\delta_i \in \{0, 1, \dots, K\}$ and $\delta_i = 0$ denoting right-censoring. We write $\hat{G}(u) = \hat{\mathbb{P}}(C > u)$ for the Kaplan-Meier estimate of the censoring survival function, fitted on $\mathcal{D}_{\text{test}}$ by swapping the event indicator ($\delta_i = 0$ becomes the "event"). This evaluation-time $\hat{G}$ is the marginal Kaplan-Meier estimator for every model and every metric; it is distinct from the training-time censoring weights $G^\star$ of the CRNLL loss (Eq. 3, Sec. C.3). Time integrals are approximated on a fixed grid $0 = \tau_0 < \tau_1 < \cdots < \tau_M = t_{\max}$ of empirical event-time quantiles, shared by all models on a given dataset, with $t_{\max}$ the 99th percentile of the observed times. All time-dependent scores below are also reported at the 25th, 50th and 75th percentiles of the observed event times ("C-index 25", etc.).

B.1 Metrics controlling probabilistic prediction metrics: IBS and INLL

For cause k and horizon t, let $y_{i,k}(t) = \mathbb{1}(t_i \le t,\ \delta_i = k)$ be the binary label "subject i has experienced cause k by time t", and let the IPCW weight be

$$\hat{w}_i(t) \;=\; \frac{\mathbb{1}(t_i \le t,\ \delta_i \neq 0)}{\hat{G}(t_i)} \;+\; \frac{\mathbb{1}(t_i > t)}{\hat{G}(t)}. \tag{4}$$

Table A.1: **Characteristics of the 20 tabular datasets.** The CR column indicates whether the dataset has competing events. The number of features is counted after applying skrub's TableVectorizer. The Github column indicates whether the dataset is directly available from GitHub; Sec. A.1 explains how we obtained the others.

Dataset name	CR	Samples	Nb features	Censoring rate	Github
Framingham	✓	4,240	24	58%	∼
Metabric (competing events)	✓	3,237	73	33%	✗
MIMIC IV	✓	65,366	172	58%	✗
MIMIC III	✓	38,511	81	54%	✗
SEER Breast Cancer	✓	476,746	18	63%	✗
SEER Digestion Cancer	✓	887,552	304	19%	✗
PBC	✓	1,945	25	55%	auton-datasets
Loan	✓	1,652,304	25	41%	✗
eICU	✓	106,196	58	95 %	✗
TCGA	✓	10,540	151	70%	✗
Flchain	✗	6,524	8	70%	pycox
Hdfail	✗	52,422	38	95%	SurvSet
GBSG	✗	2,232	7	43%	pycox
Metabric	✗	1,904	9	42%	pycox
MSK 2021	✗	25,659	86	60%	✗
MSK 2026	✗	40,755	56	57%	✗
NWTCO	✗	4,028	6	85%	pycox
smarto	✗	3,873	39	88%	SurvSet
Support	✗	8,873	14	32%	pycox
ENS Challenge	✗	3,173	42	50%	✗

Subjects censored before t ($t_i \le t,\ \delta_i = 0$) receive weight 0; the weights of the others compensate for this missing mass (Graf et al., 1999).

Definition B.1 (Integrated Brier Score (IBS))**.** The time-dependent Brier score and its integrated, cause-averaged version are

$$\mathrm{BS}_k(t) = \frac{1}{n}\sum_{i=1}^{n} \hat{w}_i(t)\,\big(y_{i,k}(t) - \hat{F}_k(t \mid \mathbf{x}_i)\big)^2, \tag{5}$$

$$\mathrm{IBS}_k = \frac{1}{t_{\max}}\int_0^{t_{\max}} \mathrm{BS}_k(t)\,\mathrm{dt}, \qquad \mathrm{IBS} = \frac{1}{\mathrm{K}}\sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{IBS}_{\mathrm{k}}. \tag{6}$$

Definition B.2 (Integrated Negative Log-Likelihood (INLL))**.** INLL replaces the squared error of Def. B.1 by the Bernoulli log-loss, keeping the same weights $\hat{w}_i(t)$ and the same time integration:

$$\mathrm{NLL}_k(t) = -\frac{1}{n}\sum_{i=1}^{n} \hat{w}_i(t)\Big[y_{i,k}(t)\log \hat{F}_k(t \mid \mathbf{x}_i) + \big(1 - y_{i,k}(t)\big)\log\big(1 - \hat{F}_k(t \mid \mathbf{x}_i)\big)\Big], \tag{7}$$

$$\mathrm{INLL}_k = \frac{1}{t_{\max}}\int_0^{t_{\max}} \mathrm{NLL}_k(t)\,\mathrm{dt}, \qquad \mathrm{INLL} = \frac{1}{\mathrm{K}}\sum_{\mathrm{k}=1}^{\mathrm{K}} \mathrm{INLL}_{\mathrm{k}}. \tag{8}$$

Predicted probabilities are clipped to $[\varepsilon, 1-\varepsilon]$ with $\varepsilon = 10^{-8}$ before taking logs.

Strictly proper scores These two metrics are *strictly proper* scoring rules: Both are minimized (in expectation) *if and only if* $\hat{F}_k = F_k$ for all k, under the assumption that $\hat{G}$ is correctly specified (Gneiting & Raftery, 2007; Rindt et al., 2022). They should be read as lower is better.

B.2 Discrimination metrics: C-index and time-dependent AUC

Definition B.3 (Time-dependent competing-risks C-index)**.** Following Wolbers et al. (2009); Antolini et al. (2005), for cause k and horizon t we order pairs (i, j) by the predicted incidence $\hat{F}_k(t \mid \mathbf{x})$. A

pair is *comparable* for cause k if i has experienced cause k by t and j is known to be at risk of a later or competing cause-k event:

$$A_{ij}^k = \mathbb{1}(\delta_i = k,\ t_i \le t)\ \Big[\mathbb{1}(t_j > t_i)\ +\ \mathbb{1}\big(t_j \le t_i,\ \delta_j \notin \{0, k\}\big)\Big]. \tag{9}$$

The pair is *concordant* if $\hat{F}_k(t \mid \mathbf{x}_i) > \hat{F}_k(t \mid \mathbf{x}_j)$, and

$$C_k(t) = \frac{\sum_{i \neq j} A_{ij}^k \Big[\mathbb{1}\big(\hat{F}_k(t \mid \mathbf{x}_i) > \hat{F}_k(t \mid \mathbf{x}_j)\big) + \frac{1}{2}\,\mathbb{1}\big(\hat{F}_k(t \mid \mathbf{x}_i) = \hat{F}_k(t \mid \mathbf{x}_j)\big)\Big]}{\sum_{i \neq j} A_{ij}^k}. \tag{10}$$

Comparable pairs are IPCW-weighted by $1/\hat{G}(t_i)$ to correct for censoring (Uno et al., 2011). $C_k(t) = 0.5$ is random ranking, $C_k(t) = 1$ is perfect ranking; higher is better.

Definition B.4 (Time-dependent (cumulative/dynamic) AUC)**.** Following Uno et al. (2007); Blanche et al. (2013), at horizon t the *cases* for cause k are $\{i : t_i \le t,\ \delta_i = k\}$ and the *controls* are $\{j : t_j > t\}$. With IPCW weights $\omega_i = 1/\hat{G}(t_i)$ on the cases,

$$\widehat{\mathrm{AUC}}_k(t) = \frac{\sum_i \sum_j \omega_i\, \mathbb{1}(t_i \le t,\ \delta_i = k)\, \mathbb{1}(t_j > t) \Big[\mathbb{1}\big(\hat{F}_k(t \mid \mathbf{x}_i) > \hat{F}_k(t \mid \mathbf{x}_j)\big) + \frac{1}{2}\mathbb{1}(\text{tie})\Big]}{\Big(\sum_i \omega_i\, \mathbb{1}(t_i \le t,\ \delta_i = k)\Big)\Big(\sum_j \mathbb{1}(t_j > t)\Big)}. \tag{11}$$

It estimates $\mathbb{P}\big(\hat{F}_k(t \mid \mathbf{X}_i) > \hat{F}_k(t \mid \mathbf{X}_j) \,\big|\, T_i^\star \le t, \Delta_i^\star = k,\ T_j^\star > t\big)$. Higher is better.

Both discrimination metrics are invariant to any strictly monotone transformation of the predicted incidences, hence insensitive to calibration.

B.3 Calibration metrics

We use the two competing-risks calibration metrics of Alberge et al. (2026), both computed with $\alpha = 2$ on a held-out calibration split $\mathcal{D}_{cal}$:

AJ-calibration (plug-in / Aalen-Johansen calibration): the time-integrated absolute gap between the average predicted CIF $\frac{1}{|\mathcal{D}_{cal}|}\sum_i \hat{F}_k(\tau \mid \mathbf{x}_i)$ and a marginal Aalen–Johansen estimate (Aalen & Johansen, 1978) of $\mathbb{P}(T^\star \le \tau, \Delta^\star = k)$ on the same split, summed over causes. It targets marginal (population-level) calibration. Full definition in Def. B.5. **AJ calibration.** We compute the PI (plug-in) calibration with the Aalen–Johansen estimator (Aalen & Johansen, 1978) as the PI estimator, and $\alpha = 2$. As the Aalen–Johansen estimator is marginal, it predicts, for a given time τ and a given event k, $\mathrm{AJ}_k(\tau) \approx \mathbb{P}(T^\star \le \tau, \Delta^\star = k)$. The metric is computed onto a calibration set called $\mathcal{D}_{cal}$, and each sample $x_i \in \mathcal{X}_{cal}$.

Definition B.5 (Plug-in calibration)**.**

Plug-in-cal$_K^\alpha$-calibration:

$$\mathrm{PI}_{cal}^\alpha = \sum_{k=1}^{K} \mathrm{PI}_k^\alpha$$

with

Plug-in-k^α calibration:

$$\mathrm{PI}_k^\alpha = \left(\int_0^{t_{max}} \mathrm{PI}_k(\tau)^\alpha \mathrm{d}\tau\right)^{1/\alpha}$$

where

Plug-in-cal$_k(\tau)$ calibration:

$$\mathrm{PI}_k(\tau) = \left|\hat{F}^k_{\mathrm{PI}(\mathcal{D}_{cal})}(\tau) - \frac{1}{|\mathcal{D}_{cal}|}\sum_{\mathbf{x}_i \in \mathcal{X}_{cal}} (\hat{F}_k(\tau|\mathbf{x}_i))\right|$$

CR-D-calibration a competing-risks extension of D-calibration (Haider et al., 2020): the normalized predicted incidence $\hat{F}_k(t_i \mid \mathbf{x}_i)/\hat{F}_k(\infty \mid \mathbf{x}_i)$ should be uniform on $[0, 1]$ for subjects experiencing cause k. For each $\rho \in [0, 1]$ the observed bin mass $\hat{b}^k_{[0,\rho]}$ (with a censoring correction) is compared to ρ, and the deviations are integrated over ρ and summed over causes. It targets conditional calibration. Full definition in Def. B.6.

Definition B.6 (CR $\hat{D}_\alpha^{CR}$-calibration)**.** For a fixed $\rho \in [0,1]$, and event type k:

$$\hat{b}^k_{[0,\rho]} = \frac{\operatorname{card}\left\{ i \;\middle|\; \delta_i = k, \hat{F}_{k/\infty}(t_i \mid \mathbf{x}_i) \in [0,\rho] \right\}}{\sum_i \{\hat{F}_k(\infty \mid \mathbf{x}_i)\}} + \frac{\sum_i \left\{ \frac{\hat{F}_k(\infty|\mathbf{x}_i)\rho - \hat{F}_k(t_i|\mathbf{x}_i)}{\hat{S}(t_i|\mathbf{x}_i)} \;\middle|\; \begin{array}{l} \delta_i = 0 \\ \hat{F}_{k/\infty}(t_i \mid \mathbf{x}_i) \in [0,\rho]) \end{array} \right\}}{\sum_i \{\hat{F}_k(\infty \mid \mathbf{x}_i)\}}$$

Where the first term counts individuals who experienced event k and whose normalized prediction falls in $[0,\rho]$, scaled by the total predicted risk. The second term accounts for censored patients, estimating their contribution. Then, we can define an estimator of the CR D_α^{CR}-calibration as:

$\hat{D}_\alpha^{CR}$-cal estimator $$\hat{D}_\alpha^{CR} \overset{\text{def}}{=} \sum_{k=1}^{K} \left(\int_0^1 \left| \hat{b}^k_{[0,\rho]} - \rho \right|^\alpha d\rho \right)^{\frac{1}{\alpha}}$$

For both, a value of 0 is perfect calibration and lower is better; in the main text we report the gain over the marginal Aalen–Johansen estimator (Aalen & Johansen, 1978).

C Details on the experimental methods

C.1 Competing Risks libraries

Table C.1 presents the libraries used to evaluate or download the datasets.

Table C.1: **Open-source libraries used in this benchmark.**

Package	Language	used for
pycox	Python	Downloading datasets, DeepHit Architecture
lifelines	Python	Aalen–Johansen estimator (Aalen & Johansen, 1978)
auton-survival	Python	Downloading Datasets
hazardous	Python	Computing CR metrics (IBS, C-index, calibrations), SurvivalBoost
cbioportal	R	Downloading Datasets
cmprsks	R	Fine and Gray estimator (Fine & Gray, 1999), RSF

C.2 Compute and Hyper parameter search

Compute To run the experiments, we had two setups, one for the CPU methods (Fine & Gray, 1999; Ishwaran et al., 2014; Alberge et al., 2025) and one for the GPUs ones. The CPUs methods were ran onto 2x Intel Xeon E5-2660 v2 @ 2.20GHz with 40 cores and 256Go RAM. The GPUs ones were ran onto 2x Nvidia V100 32G, with the 2x AMD EPYC 7502 @ 2.50GHz processor, 128 cores, and 512G RAM.

Table C.2: Hyper-parameter search space for every backbone and baseline. Each cell is searched with `RandomizedSearchCV` for 150 iterations over 3 stratified splits. `randint`/`loguniform`/`uniform` denote the `scipy.stats` distributions; a list denotes a categorical choice. The ranking weight α and temperature σ are searched only for the HLRanking loss, which is the only loss that uses them.

model	parameter	range
	Backbones (each trained under all three losses)	
MLP	batch size	128, 256
	dropout	None, 0.1, 0.2, 0.3
	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]
	weight decay	`loguniform`(1e-05, 0.001)
	alpha †	`uniform`[0, 1]
	sigma †	`loguniform`(0.01, 10)
RealMLP	base lr	`loguniform`(1e-05, 0.001)
	batch size	128, 256
	beta1	`uniform`[0.1, 0.99]
	beta2	`uniform`[0.1, 0.99]
	hidden sizes	[128, 128, 128], [256, 256, 256], [64, 64, 64], [128, 256, 128], [256, 128, 256], [64, 128, 64], [128, 64, 128], [256, 64, 256], [64, 256, 64]
	n epochs	`randint`(256, 1024)
	nb output times	40, 60, 80, 100
	weight decay	`loguniform`(1e-05, 0.001)
	alpha †	`uniform`[0, 1]
	sigma †	`loguniform`(0.01, 10)
TabM	batch size	128, 256
	d block	64, 128, 256
	epochs	`randint`(256, 1024)

model	parameter	range
	k	4, 8, 16
	lr	loguniform(1e-05, 0.001)
	n blocks	1, 2, 3, 4
	nb output times	40, 60, 80, 100
	weight decay	loguniform(1e-05, 0.001)
	alpha †	uniform[0, 1]
	sigma †	loguniform(0.01, 10)
SurvTRACE	attention probs dropout prob	uniform[0, 0.3]
	batch size	64, 128, 256
	patience	randint(3, 20)
	hidden dropout prob	uniform[0, 0.3]
	hidden size	16, 32, 64, 128
	initializer range	loguniform(0.001, 0.1)
	intermediate size	64, 128, 256
	lr	loguniform(1e-05, 0.01)
	max epochs	randint(256, 1024)
	nb output times	20, 40, 60, 80, 100
	num attention heads	1, 2, 4, 8
	num hidden layers	1, 2, 3, 4
	betas	(0.9, 0.999), (0.9, 0.95), (0.95, 0.999), (0.8, 0.999)
	weight decay	loguniform(1e-06, 0.01)
	alpha †	uniform[0, 1]
	sigma †	loguniform(0.01, 10)
Baselines (each trained with its own fixed loss)		
Aalen–Johansen (Aalen & Johansen, 1978)	no hyperparameters (population-level estimator)	
Fine and Gray (Fine & Gray, 1999)	max iter	randint(10, 100)
	max samples	50000
	n components	5, 10, 20, 30, 50
	tol	loguniform(1e-08, 0.0001)
Random Survival Forests	max depth	None, 4, 8, 12, 16, 24
	max features	sqrt, log2, 0.3, 0.5, 0.8, None
	min samples leaf	randint(1, 25)
	min samples split	randint(2, 30)
	n estimators	randint(100, 1000)
SurvivalBoost	hard zero fraction	uniform[0, 0.5]
	learning rate	loguniform(0.01, 0.3)
	max depth	randint(2, 10)
	max leaf nodes	15, 31, 63, 127
	min samples leaf	randint(5, 100)
	n horizons per observation	randint(1, 5)
	n iter	randint(50, 200)
	n iter before feedback	randint(5, 50)
	n time grid steps	50, 100, 200
SurvTRACE	attention probs dropout prob	uniform[0, 0.3]
	batch size	64, 128, 256
	patience	randint(3, 20)
	hidden dropout prob	uniform[0, 0.3]
	hidden size	16, 32, 64, 128
	initializer range	loguniform(0.001, 0.1)
	intermediate size	64, 128, 256
	lr	loguniform(1e-05, 0.01)
	max epochs	randint(50, 500)
	num attention heads	1, 2, 4, 8
	num hidden layers	1, 2, 3, 4
	betas	(0.9, 0.999), (0.9, 0.95), (0.95, 0.999), (0.8, 0.999)

model	parameter	range
	weight decay	loguniform(1e-06, 0.01)
	quantile horizons	evenly spaced quantiles, $n \in \{3, 10, 20, 40, 60\}$
Deep Survival Machines	batch size	128, 256
	discount	0.5, 0.75, 1.0
	distribution	Weibull, LogNormal
	epochs	randint(50, 300)
	k	3, 4, 6, 8
	layers	None, [64], [128], [64, 64], [128, 128], [256, 256]
	lr	loguniform(0.0001, 0.01)
DeSurv	batch size	128, 256
	epochs	randint(100, 500)
	hidden dim	32, 64, 128, 256
	lr	loguniform(0.0001, 0.01)
Neural Fine-Gray	batch size	128, 256, 512
	dropout	uniform[0, 0.5]
	epochs	randint(100, 1000)
	layers	(50, 50), (100, 100), (100, 100, 100), (200, 200), (50, 100, 50), (200, 100, 200)
	layers surv	(50,), (100,), (100, 100), (200,)
	lr	loguniform(0.0001, 0.01)
	patience max	randint(5, 30)
	weight decay	loguniform(1e-05, 0.01)
DQS	batch size	128, 256
	epochs	randint(100, 500)
	hidden dim	64, 128, 256
	lr	loguniform(0.0001, 0.01)
	n bins	20, 50, 100

Selected ranking weight of HLRanking Table C.3 reports the weight α of $\mathcal{L}_{HLR} = \alpha\,\mathcal{L}_{HL} + (1-\alpha)\,\mathcal{L}_R$ (Sec. 2.2) selected by the search above (uniform$[0,1]$) for HLRanking, for each backbone and for both selection criteria. *Conventions.* DeepHit (Lee et al., 2018) writes the loss as $\mathcal{L}_{HL} + \alpha_{\mathrm{DH}}\,\mathcal{L}_R$ with $\alpha_{\mathrm{DH}} \geq 0$ on the ranking term. We use the pycox implementation (Kvamme et al., 2019) in every backbone, which instead takes the convex combination above: α weights the likelihood, $\alpha = 1$ is pure likelihood and $\alpha = 0$ pure ranking. Up to an overall scale absorbed by the learning rate, the two are related by $\alpha_{\mathrm{DH}} = (1-\alpha)/\alpha$, so a *small* α here means a *large* ranking weight. *What the selected values imply.* Tuned on IBS, the median α is 0.59 over all backbones ($\alpha_{\mathrm{DH}} \approx 0.7$): the likelihood carries more weight than the ranking term, which is down-weighted but not switched off. Tuned on the C-index, the median drops to 0.29 ($\alpha_{\mathrm{DH}} \approx 2.4$) and the ranking term dominates, on RealMLP almost entirely (median $\alpha \approx 0.003$). Each criterion thus pulls the ranking weight in the direction that favours it, which is the trade-off Tab. D.5 measures.

C.3 Training-time IPCW sensitivity: Kaplan-Meier vs. Cox vs. stratified

To justify the marginal Kaplan-Meier default used for the CRNLL loss's training-time IPCW (Sec. 2.2), we retrained CRNLL under three training-time censoring-weighting schemes – marginal Kaplan-Meier (Kaplan & Meier, 1958), Cox (Cox, 1972), and a covariate-stratified Kaplan-Meier (a piecewise-constant-in-covariates middle ground) – on all four backbones, holding the evaluation-side IPCW fixed at marginal Kaplan-Meier throughout so that only the training-time weighting changes.

Cox is markedly less reliable than the other two schemes. Where Kaplan-Meier scores all 94 (dataset, backbone) cells and the stratified estimator scores 93/94, Cox produces a usable fit on only 35/94 (63% missing, due to non-convergence). On the 35 cells where all three schemes completed, splitting at a 20% relative IBS gain over Kaplan-Meier gives a sharp split: 24 cells are stable (median gain $+0.0\%$ for Cox, $+0.4\%$ for stratified), while on the remaining 11 cells Cox's IBS is on average

Table C.3: **Ranking weight α selected for HLRanking**, by selection criterion and backbone: median, quartiles and range.

Selection	Backbone	Median	Q_1	Q_3	Min	Max
IBS	MLP	0.56	0.56	0.82	0.20	0.95
	RealMLP	0.66	0.52	0.82	0.09	0.98
	TabM	0.58	0.20	0.73	0.04	0.90
	SurvTRACE	0.66	0.46	0.74	0.15	0.99
	All backbones	0.59	0.42	0.79	0.04	0.99
C-index	MLP	0.46	0.20	0.55	0.03	0.83
	RealMLP	0.00	0.00	0.66	0.00	0.89
	TabM	0.20	0.03	0.67	0.01	1.00
	SurvTRACE	0.29	0.22	0.52	0.06	0.97
	All backbones	0.29	0.03	0.63	0.00	1.00

$+139.4\%$ worse than Kaplan-Meier's – and in every one of these 11 cells Cox is worse, never better – while the stratified estimator stays within a couple of percent of Kaplan-Meier even on this same subset (Table C.4). Cox is also slower to fit: a median $1.75\times$ and up to $6.7\times$ the fitting time of Kaplan-Meier, on the cells where both complete. We found no consistent relationship between a dataset's censoring rate and whether Cox is stable on it: the blown-up cells span censoring rates from 35% to 98%, and some of the most heavily censored datasets in the benchmark (e.g. nwtco, smarto) are among the most stable for Cox.

Table C.4: **A Cox censoring model is unreliable here, while a covariate-stratified one reproduces the marginal estimate.** CRNLL training-time IPCW, Cox against Kaplan-Meier against stratified Kaplan-Meier, on the 35 (dataset, backbone) cells where all three schemes produced a usable fit, split at $|\text{IBS gain over Kaplan-Meier}| = 20\%$.

	n cells	median Cox vs. KM	median stratified vs. KM
Stable	24	$+0.0\%$	$+0.4\%$
Blown up ($> 20\%$ worse)	11	$+139.4\%$	$+0.2\%$

We read this as supporting the marginal-Kaplan-Meier default used throughout the paper: a stable, covariate-adjusted alternative (stratified Kaplan-Meier) reproduces it almost everywhere it is estimable, while Cox's instability under this benchmark is a property of that estimator on these datasets, not evidence that a covariate-conditional censoring model would change our conclusions if it were reliably estimable.

D Detailed Experimental Results

D.1 Computing effect sizes: How much the loss matters, and where

Section 6 compares the importance of the two axes loss/backbone with three numbers, obtained from the per-(dataset, backbone, loss) mean IBS gains that also produce Fig. 2a. Restricted to the 12 cells of the loss $\times$ backbone matrix, so that both axes are free, and to the 20 tabular datasets of Table 2. The *loss effect* is, for each (backbone, dataset) pair, the spread over the three losses, $\max - \min$; the *backbone effect* is, for each (loss, dataset) pair, the spread over the four backbones. Both are in IBS-gain units and are paired within a dataset, which is what makes them comparable. Coverage is 229 of the 240 cells, and a spread is taken only where every member of its group is present, so the two effects rest on $n = 75$ and $n = 49$ groups respectively. Table D.1 details the corresponding estimates.

Table D.1: **The loss separates the models twice as much as the backbone on competing-risks data, and slightly less than the backbone on single-event data.** Median spread in IBS gain, with the interquartile range across groups. Pooling the two families averages two opposite regimes and hides the effect, which is why the main text reports them apart.

Datasets	Loss effect		Backbone effect		Ratio
	median	IQR	median	IQR	
Competing risks	0.053	[0.020, 0.084]	0.027	[0.021, 0.062]	1.93
Single event	0.072	[0.045, 0.132]	0.086	[0.039, 0.208]	0.83
All 20	0.065	[0.030, 0.100]	0.059	[0.023, 0.136]	1.10

Table D.2: **Per-dataset view behind Tab. D.1.** For each dataset, the median spread over the three losses and over the four backbones, and which loss attains the best median gain. The loss spread exceeds the backbone spread on most competing-risks datasets and on fewer of the single-event ones.

Dataset	Type	Loss spread	Backbone spread	Best loss
framingham	CR	0.036	0.059	CRNLL
metabric_competing	CR	0.068	0.022	CRNLL
mimic	CR	0.012	–	CRNLL
mimiciii	CR	0.011	0.020	CRNLL
seer	CR	0.043	0.021	CRNLL
seer_digestion	CR	0.029	–	CRNLL
pbc	CR	0.100	0.060	CRNLL
loan	CR	0.038	0.033	CRNLL
eicu	CR	0.298	0.213	CRNLL
tcga	CR	0.103	0.079	CRNLL
flchain	single	0.067	0.197	CRNLL
hdfail	single	0.110	0.146	CRNLL
gbsg	single	0.126	0.218	CRNLL
metabric	single	0.052	0.028	CRNLL
msk_2021	single	0.066	0.055	CRNLL
msk_2026	single	0.064	0.052	CRNLL
nwtco	single	0.155	3.136	CRNLL
smarto	single	0.137	0.256	CRNLL
support	single	0.078	0.120	CRNLL
challenge_ens	single	0.041	0.067	CRNLL

D.2 Comparing the different model-selection metrics

Here we give results for the six model-selection metrics of Sec. 2.3 on the tabular data.

The six metrics fall into two groups: strictly proper scoring rules and others. Results of whether they efficiently tell the models apart are related to their properness. The two proper scoring rules agree: the Brier score (Fig. 2a) and the integrated negative log-likelihood (Fig. D.2) both put the CRNLL group ahead of the hazard-based ones on every backbone. The two ranking metrics behave similarly but do not distinguish models or losses: neither the C-index (Fig. D.3) nor the integrated AUC (Fig. D.4) separates any model from any other, the boxes overlapping almost completely across all twelve cells and the nine baselines. Note that AUC is proper (it is minimized by the true generative mechanism), but not strictly proper (the true generative mechanism is not the *only* minimizer. Experimental results show that its integration over time is gives more muddied signal that than a strictly proper scoring rule (IBS or INLL). We discuss calibration metrics below.

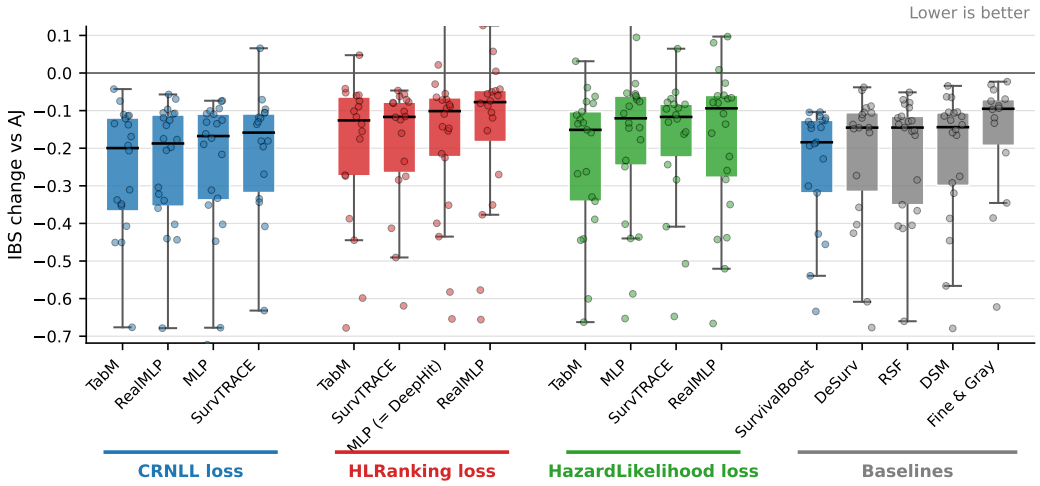


Figure D.1: **IBS results.** (reproducing Fig. 2a of the main text for ease of read) IBS gain over the Aalen–Johansen estimator on all 20 tabular datasets.

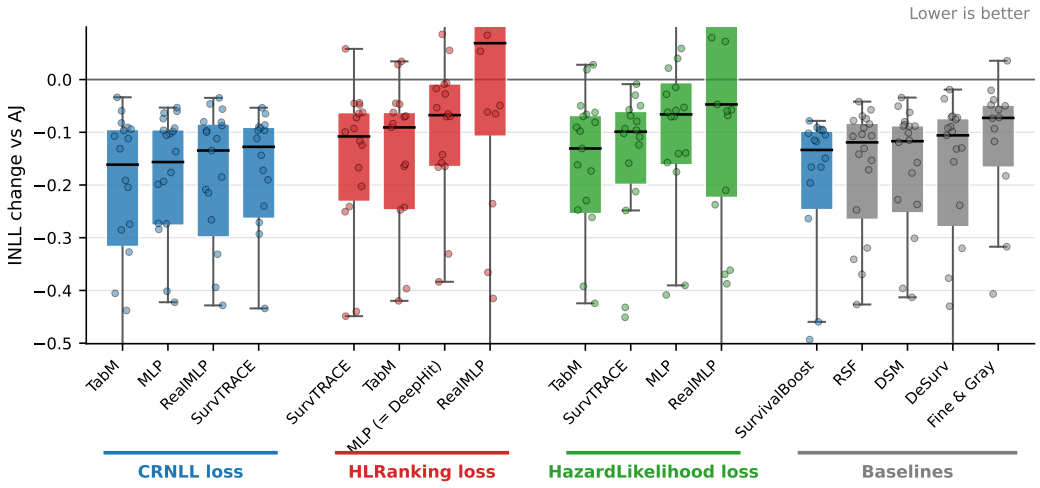


Figure D.2: **The second proper scoring rule orders the losses as the Brier score does.** Integrated negative log-likelihood against the Aalen–Johansen estimator. CRNLL is lowest on every backbone.

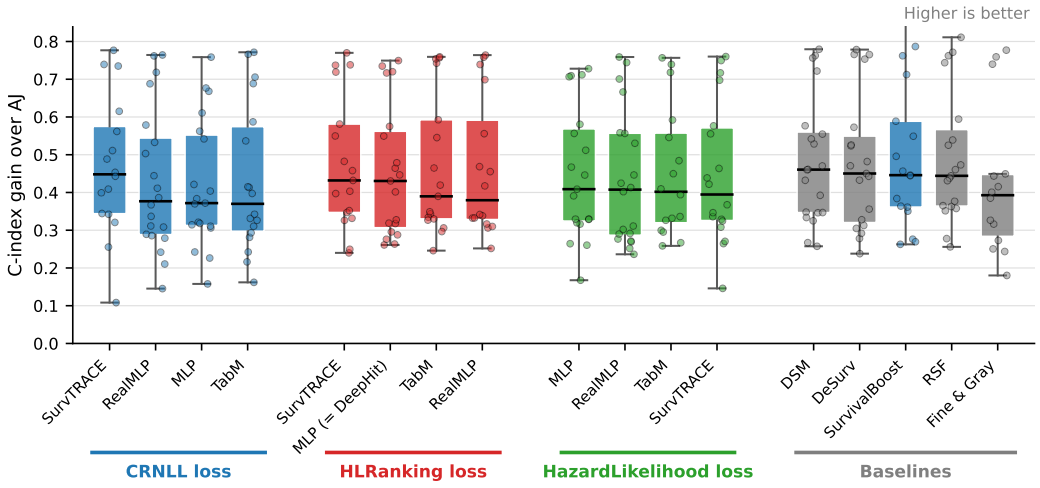


Figure D.3: **The C-index does not separate well the models.** Gain in C-index over the Aalen–Johansen estimator, higher is better. The boxes overlap across losses, backbones and baselines alike.

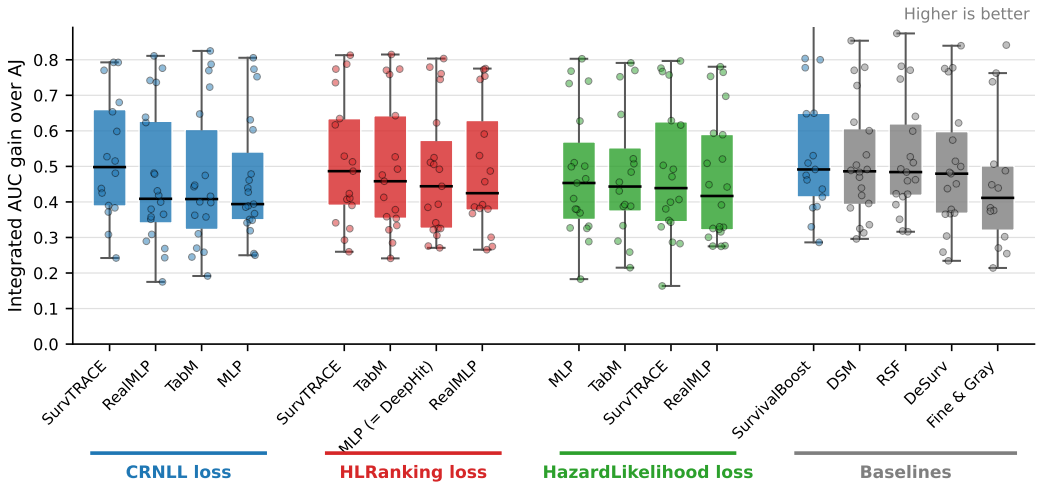


Figure D.4: **Integrated AUC also gives a muddied picture.** Gain in integrated time-dependent AUC over the Aalen–Johansen estimator, higher is better. As for the C-index, models are not well separated.

Calibration We detail below the two calibration criteria. While calibration is an important topic, it comes with different definitions in the case of survival analysis (whether single-event or competing risks), which are difficult to estimate:

AJ-calibration The AJ-calibration (Definition B.5) measures the area between a model's marginal predictions (i.e., the mean predicted cumulative incidence across samples) and the Aalen-Johansen estimator; a model is considered perfectly calibrated when this quantity is near zero (Alberge et al., 2026). The boxplots confirm that models trained with the CRNLL loss are the best calibrated overall, while those trained with the HLRanking loss exhibit the largest calibration errors.

D-calibration We additionally report the D-calibration gain over the Aalen-Johansen estimator (Aalen & Johansen, 1978). Despite the large variance across datasets, a tendency emerges: proper scoring rule (PSR) based losses, and the HazardLikelihood loss in particular, yield better-calibrated models. As with AJ-calibration, a smaller D-calibration value indicates better calibration (Alberge et al., 2026).

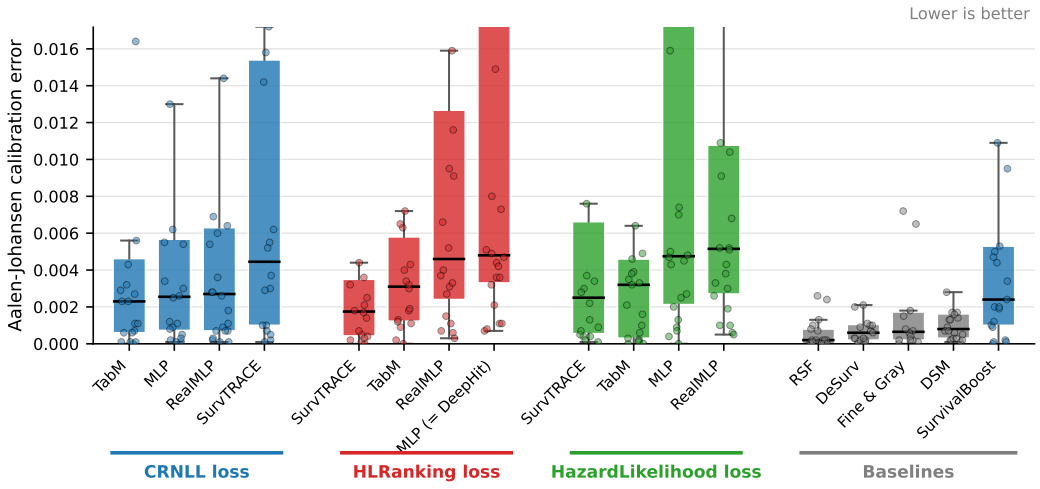


Figure D.5: **On raw marginal calibration too, CRNLL is the best calibrated and HLRanking the worst.** Area between the Aalen–Johansen estimator (Aalen & Johansen, 1978) and one model.

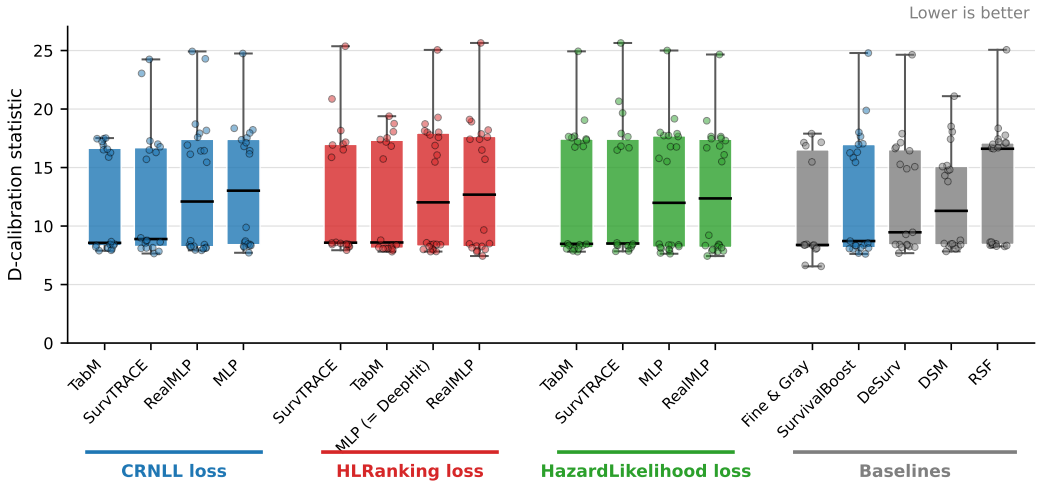


Figure D.6: **On D-calibration the three losses overlap heavily, the spread across datasets being far larger than any difference between them.** Raw D-calibration statistic

The intention behind the use of calibration metrics is typically to judge how far a model is from good probabilities. In terms of pure model selection (comparing multiple models), our results suggest that strictly proper scoring rules should probably be preferred.

Is the C-index flatter on competing risks than on single-event data? We tested whether the C-index separates models less on the 10 competing-risks datasets than on the 10 single-event ones, reusing the spreads of Sec. D.1 on the C-index gain at the median time (Tab. D.3). It does not. The spread across losses, across backbones, or across all cells of a dataset is about the same in both families, and a one-sided Mann–Whitney test of a smaller spread on competing risks is far from significant ($p \geq 0.43$). Relative to the IBS spread on the same group, the C-index even moves more on competing risks, mostly because the IBS itself separates the models less there. Nor does the C-index recover the IBS ordering in either family: the per-dataset Spearman correlation between the two rankings of the cells has median -0.23 on competing risks and -0.06 on single-event data ($p = 0.52$). With 7 to 10 datasets per family these tests have little power, but they give no sign that the C-index is less informative in one family: it is equally flat in both.

Table D.3: **The C-index separates models about as little on competing risks (CR) as on single-event data.** Median spread in C-index gain at the median time, with interquartile range, over the groups of Sec. D.1; p: one-sided Mann–Whitney test of a smaller spread on CR. Right: the same spread divided by the IBS spread of the same group.

	C-index spread			C-index / IBS spread	
Spread across	CR	single event	p	CR	single event
the 3 losses	0.032 [0.015, 0.067]	0.035 [0.015, 0.068]	0.43	0.63 [0.46, 1.25]	0.50 [0.16, 1.05]
the 4 backbones	0.054 [0.026, 0.113]	0.038 [0.024, 0.075]	0.69	1.09 [0.76, 4.09]	0.44 [0.14, 1.40]
all cells of a dataset	0.113 [0.061, 0.182]	0.082 [0.048, 0.133]	0.67	1.11 [0.62, 1.76]	0.60 [0.18, 0.74]

D.3 Dataset stratification: multiple events or not, amount of censoring

Here we study how results vary along different dataset characteristics, namely competing risks versus single-event survival analysis, as well as amount of censoring.

Multiple events or not Figure D.7 gives the per-dataset IBS gain over the Aalen–Johansen estimator (Aalen & Johansen, 1978) for each competing risks dataset. Results folow the trend observed in Fig. 2a, but the effect is even more marked in pure competing-risk settings: the CRNLL loss consistently achieves the largest gains across datasets and architectures and the HLRanking that struggles the most.

Table D.4: **The advantage of the cumulative-incidence loss is far wider on competing-risks data.** Median IBS gain over Aalen–Johansen for each cell of the matrix, best backbone per row in bold. On competing risks CRNLL leads the nearest hazard-based loss by a large margin on every backbone; on single-event data the three losses fall much closer together.

	Competing risks (10 datasets)				Single event (10 datasets)			
Loss	MLP	RealMLP	TabM	SurvTRACE	MLP	RealMLP	TabM	SurvTRACE
CRNLL	0.261	0.250	**0.351**	0.258	0.146	0.150	**0.152**	0.125
HazardLikelihood	0.117	0.198	**0.268**	0.120	0.084	0.043	**0.128**	0.117
HLRanking	0.120	0.093	**0.180**	0.120	0.063	0.049	**0.126**	0.117

Amount of censoring Heavy censoring is the regime in which competing-risks models are hardest to fit and in which the choice of a loss adapted to the censoring in competing risk is most important, so we also isolate the most heavily censored datasets of the collection (Fig. D.8). The figure shows a widening of the benefits observed for CRNLL on the full dataset.

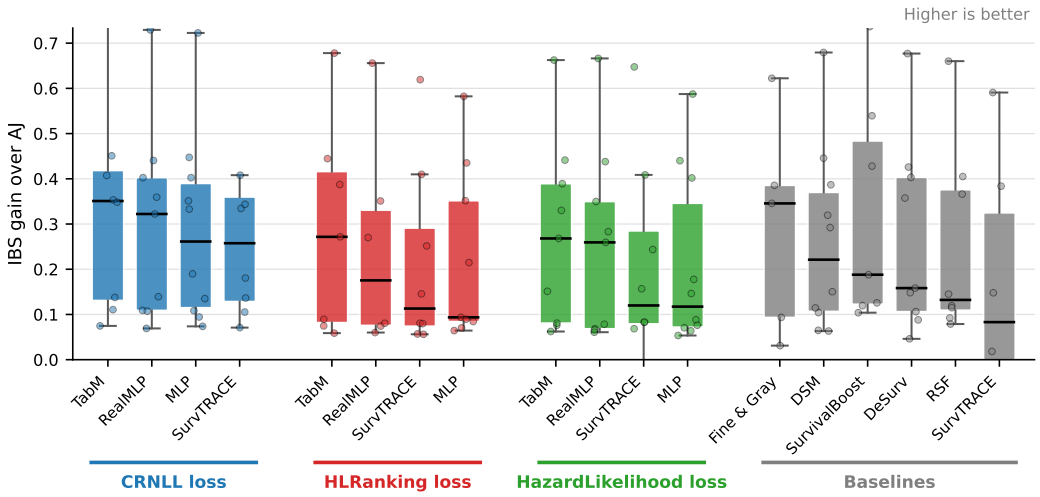


Figure D.7: **The ordering is unchanged on the competing-risks datasets alone.** IBS gain over the Aalen–Johansen estimator, restricted to the 10 competing-risks datasets. Layout as in Fig. 2a.

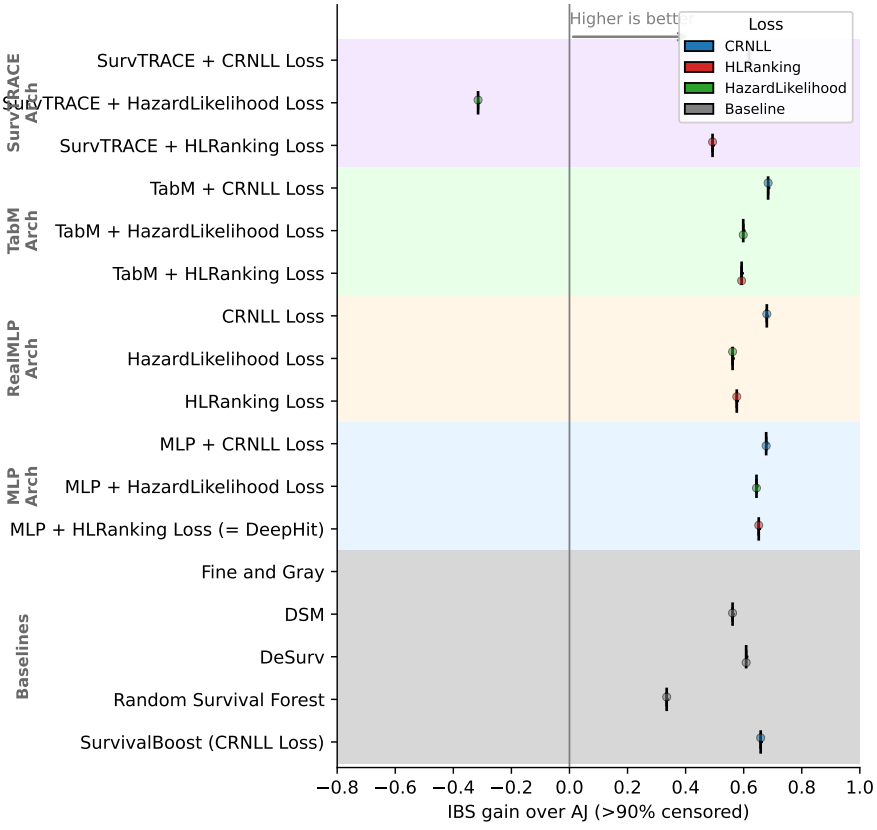


Figure D.8: **Under heavy censoring the advantage of the CRNLL loss persists and widens.** IBS gain over the Aalen–Johansen estimator on the datasets censored above 90%, grouped by backbone so that the three losses can be compared within each block; higher is better. CRNLL is ahead of both hazard-based losses on every backbone, while those two remain close to each other, as in Fig. 2a. Two contrasts with the full collection are worth noting: the random survival forest, which is competitive overall, falls well behind the neural models trained with CRNLL here, and one cell (SurvTRACE with the HazardLikelihood loss) degrades to worse than the Aalen–Johansen reference.

D.4 C-INDEXES AT DIFFERENT TIME POINTS

The C-index is a ranking metric evaluated at a specific time point. The main text summarises the 25th-quantile result; every C-index figure lives here. We show the gain over the Aalen-Johansen estimator for all three quantiles (25th, 50th, 75th). While a marginal advantage of the HLRanking loss is visible at the 25th quantile (Fig. D.9a), no clear trend emerges at the 50th or 75th quantiles (Figs. D.9b, D.9c). Here we see that even restricting C-index at given time points does not lead it to bring light on relative benefits of either architectures or loss functions.

D.5 HLRANKING: TUNING ON ITS OWN TARGET METRIC

The HLRanking results in the main text select hyperparameters on validation IBS, which leaves open the possibility that HLRanking's ranking term is simply undertuned for the metric it is designed to optimize. Table D.5 reports, for each of the four backbones, HLRanking's own IBS and C-index (50th quantile) under the two selection criteria of Sec. 2.2 (IBS-selected vs. C-index-selected), paired at the (dataset, backbone) level.

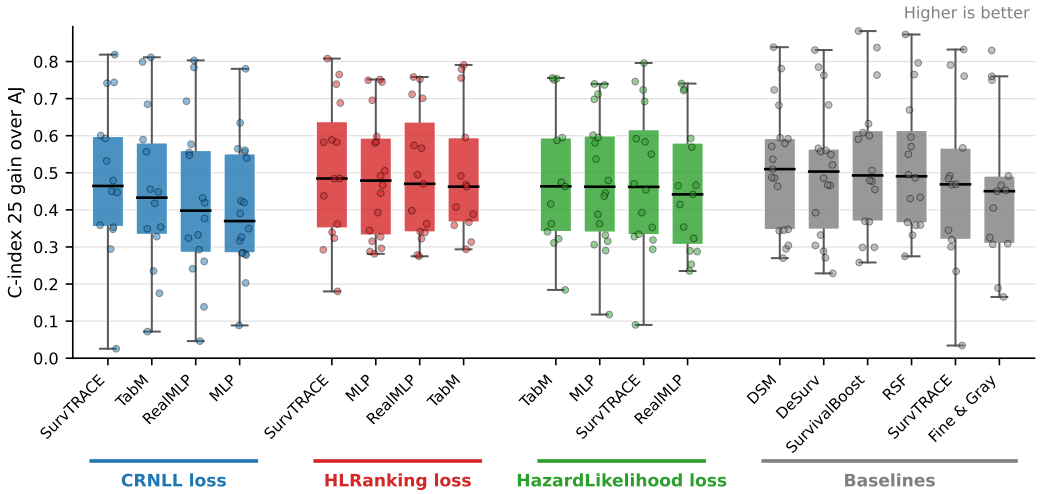
Table D.5: **Tuning HLRanking on the C-index buys almost no C-index and costs a great deal of IBS.** IBS-selected against C-index-selected hyperparameters, paired per (dataset, backbone), post alpha/sigma fix (Sec. 2.2). n = number of datasets with both selections available for that backbone. Median Δ = C-index-selected minus IBS-selected.

Backbone	n	med. Δ IBS	IBS worse	med. Δ C-idx	C-idx better
MLP	22	+0.0069	17/22	+0.0018	13/22
RealMLP	19	+0.0150	14/19	+0.0018	12/19
TabM	20	+0.0142	17/20	+0.0020	13/20
SurvTRACE	13	+0.0082	9/13	+0.0006	7/13
Pooled	74	+0.0117	57/74 (77%)	+0.0018	45/74 (61%)

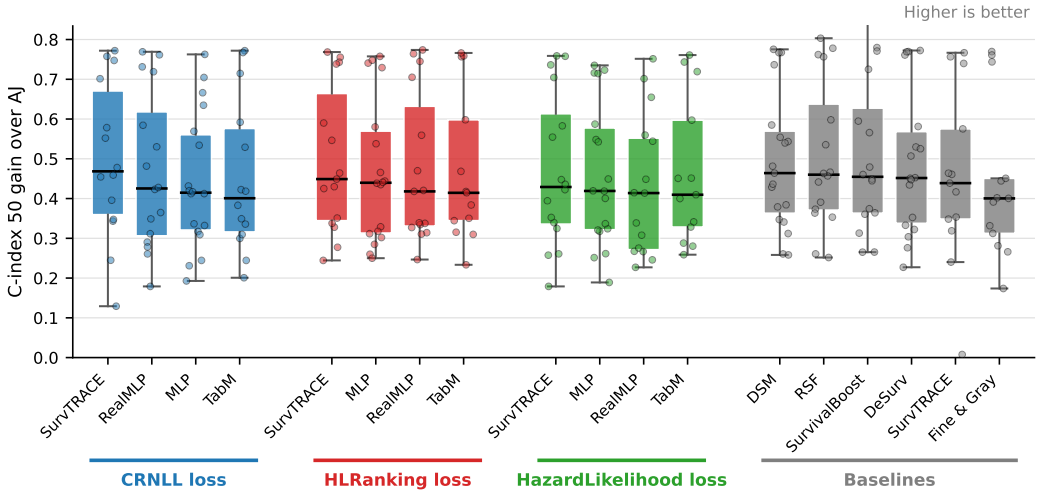
D.6 CRNLL SURVIVAL-HEAD ABLATION

The CRNLL loss uses $K + 1$ heads (Sec. 2.2), one per competing event plus a dedicated survival head, whereas the hazard-based losses use only K heads. Loss and output parameterization are therefore two choices that vary together, and an advantage we attribute to the loss could in principle come from the extra output instead. We separate them by training CRNLL twice on each of the four backbones (MLP, RealMLP, TabM, SurvTRACE), once with the $(K + 1)$-th head and once with only K heads renormalized without a dedicated survival output, and comparing the two on IBS and on both calibration criteria.

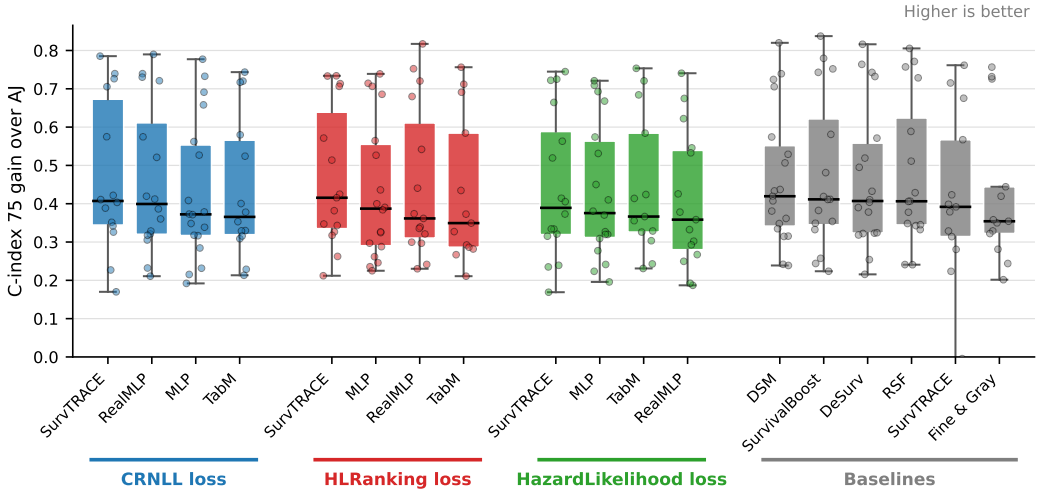
The extra head is not what makes CRNLL competitive. On IBS (Fig. D.10) the two variants overlap on every backbone: their medians differ by far less than the spread across datasets, and the sign of the difference is not even consistent, RealMLP slightly favouring the extra head and SurvTRACE slightly favouring its removal. Both calibration criteria behave the same way (Fig. D.11), with the paired distributions essentially indistinguishable per backbone. Since CRNLL keeps its gain once the extra output is removed, the advantage reported in Sec. 6 follows from the loss rather than from the output parameterization.



(a) **t = 25th percentile.**



(b) **t = 50th percentile.**



(c) **t = 75th percentile.**

Figure D.9: **C-index at given times separates neither the loss nor the backbone** Gain in C-index over the Aalen–Johansen estimator for different time horizons. Layout as in Fig. 2a.

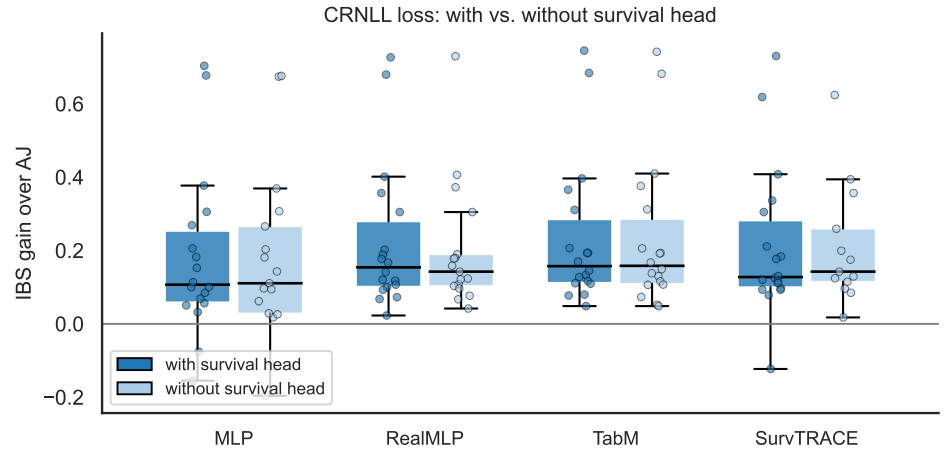


Figure D.10: **The CRNLL advantage does not come from its extra survival head.** IBS gain over the Aalen–Johansen estimator for CRNLL trained with the dedicated $(K+1)$-th survival head (dark) and with K heads renormalized without one (light), paired per backbone; higher is better. Each point is one dataset, the box spans the interquartile range and the whiskers reach 1.5 IQR. The two variants overlap on all four backbones and neither is systematically ahead, so "loss" and "output parameterization" are separable here and it is the loss that carries the effect.

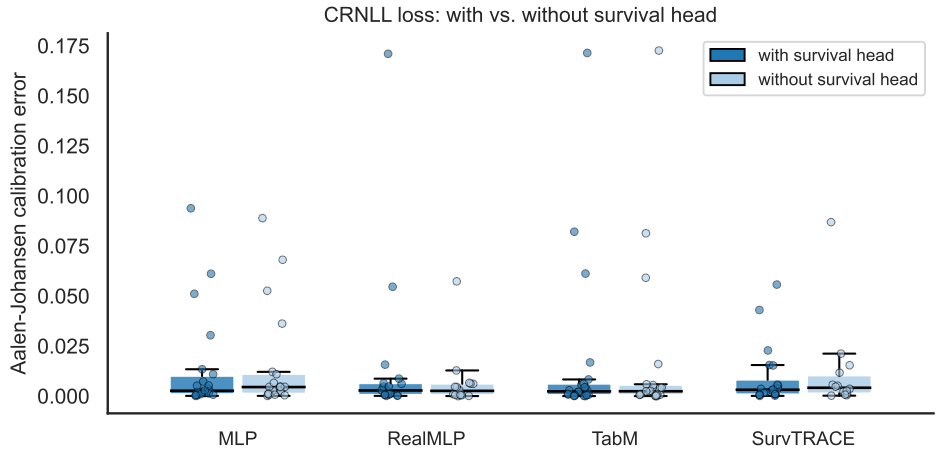
E Background theory: Link between the hazard function and the CIF function

Here, we do the derivation to explain the link of the hazard function λ_k of the event of interest k, the survival function S and the CIF of the event k F_k, and the density of the CIF (f_k).

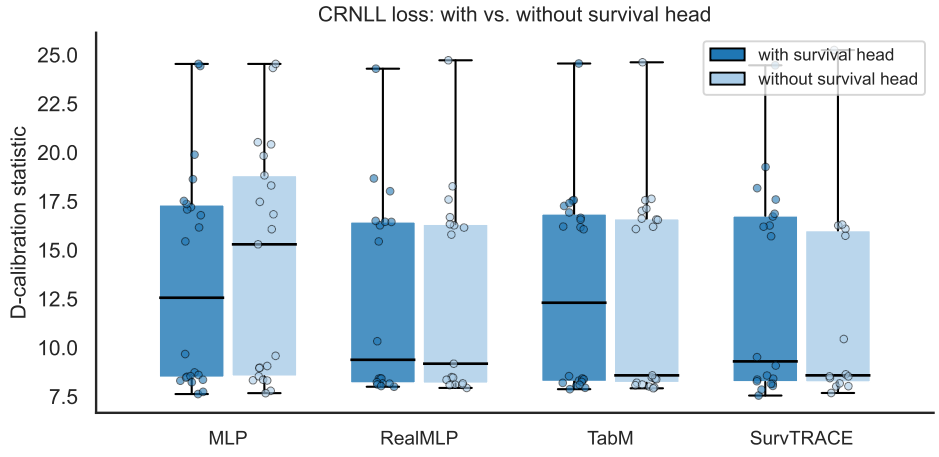
$$\mathbb{P}(T^\star \in [t, t+dt], \Delta^\star = k | T^\star > t, \mathbf{X} = \mathbf{x})\, dt = \frac{\mathbb{P}(T^\star \in [t, t+dt], \Delta^\star = k | \mathbf{X} = \mathbf{x})\, dt}{\mathbb{P}(T^\star > t | \mathbf{X} = \mathbf{x})} \tag{12}$$

$$\implies \lambda_k(t|\mathbf{x}) = \frac{f_k(t|\mathbf{x})}{S(t|\mathbf{x})} \tag{13}$$

$$\implies F_k(t|\mathbf{x}) = \int_0^t \lambda_k(t|\mathbf{x}) S(t|\mathbf{x})\, du \tag{14}$$



(a) AJ-calibration error (lower is better).



(b) D-calibration statistic (lower is better).

Figure D.11: **The survival head does not change calibration either.** Same ablation and same layout as Fig. D.10, scored on the two competing-risks calibration criteria of Sec. 2.3. On AJ-calibration both variants concentrate near zero on every backbone, with the same handful of poorly calibrated datasets in each. D-calibration is far more dispersed, but the paired boxes still overlap heavily and the better median alternates between the two variants across backbones, which is what one expects when the parameterization is immaterial rather than beneficial.