
Differentiable sorting for censored time-to-event data.

Andre Vauvelle^{1,2*}, Benjamin Wild^{3*}, Roland Eils³, Spiros Denaxas¹
University College London¹, BenevolentAI², Berlin Institute of Health³
{andre.vauvelle.19,s.denaxas}@ucl.ac.uk
{benjamin.wild, roland.eils}@bih-charite.de

Abstract

1 Survival analysis is a crucial semi-supervised task in machine learning with signifi-
2 cant real-world applications, especially in healthcare. The most common approach
3 to survival analysis, Cox’s partial likelihood, can be interpreted as a ranking model
4 optimized on a lower bound of the concordance index. We follow these connec-
5 tions further, with listwise ranking losses that allow for a relaxation of the pairwise
6 independence assumption. Given the inherent transitivity of ranking, we explore
7 differentiable sorting networks as a means to introduce a stronger transitive in-
8 ductive bias during optimization. Despite their potential, current differentiable
9 sorting methods cannot account for censoring, a crucial aspect of many real-world
10 datasets. We propose a novel method, DiffSurv, to overcome this limitation by
11 extending differentiable sorting methods to handle censored tasks. DiffSurv pre-
12 dicted matrices of *possible* permutations that accommodate the label uncertainty
13 introduced by censored samples. Our experiments reveal that DiffSurv outperforms
14 established baselines in various simulated and real-world risk prediction scenarios.
15 Furthermore, we demonstrate the algorithmic advantages of DiffSurv by presenting
16 a novel method for top-k risk prediction that surpasses current methods. In conclu-
17 sion, DiffSurv not only provides a novel framework for survival analysis through
18 differentiable sorting, but also significantly impacts real-world applications by im-
19 proving risk stratification and offering a methodological foundation for developing
20 predictive models in healthcare and beyond.

21 1 Introduction

22 Survival analysis plays a pivotal role in many realworld machine learning applications, spanning
23 fields such as reliability engineering, marketing, and insurance, with a particularly significant impact
24 in healthcare. The goal of survival analysis is to predict the time until the occurrence of an event of
25 interest, such as death, based on a set of covariates. In clinical studies, these include demographic
26 variables such as sex and age, but may also encompass more complex data modalities such as medical
27 images.

28 The concept of censoring is a distinguishing characteristic that sets survival analysis apart from
29 conventional machine learning approaches. Particularly prevalent in observational datasets, it refers
30 to situations where event times remain unobserved because a patient might not have undergone the
31 event by the time of data collection. This can be due to a variety of reasons such as the study period
32 ending before all events of interest have occurred or subjects leaving the study.

33 Overlooking censoring can skew predictions towards the censoring event, rather than the event of
34 interest. This bias becomes particularly noticeable when the study’s endpoint can be inferred from the
35 observed covariates - age being a notable example. In such cases, the predicted event times are likely

*Equal contribution

36 to biased towards the censoring event time, thereby neglecting the actual event of interest [Kvamme
37 and Borgan, 2019].

38 The Cox Proportional Hazards model is widely used for handling censored data in survival analysis
39 [Cox, 1972]. The model optimizes a partial likelihood function over ranked data, considering only
40 the order of events, not their exact time of occurrence. As such, Cox’s partial likelihood serves as a
41 ranking loss, learning from the order of patients based on their hazard of experiencing an event, not
42 their exact survival time.

43 Raykar et al. [2007] showed that Cox’s partial likelihood (CPL) and ranking losses can be directly
44 equated, with both providing lower bounds to the concordance index, the primary evaluation metric
45 used in survival analysis. Both losses are foundational to many survival deep learning methodologies
46 like DeepSurv Katzman et al. [2018] and DeepHit Lee et al. [2018].

47 However, this relation operates under the assumption of pairwise independence. This simplification,
48 while practical, can de-emphasize the transitive properties inherent in survival data. As shown by
49 Goldstein and Langholz [1992], larger risk set sizes can lead to more efficient estimators, suggesting
50 potential benefits in considering listwise ranking losses Cao et al. [2007]. These losses optimize over
51 lists of values rather than individual pairs, thereby better capturing the transitive dynamics of the data.
52 Despite the similarities, listwise losses have remained largely unexplored within the field of survival
53 analysis. This could be partly due to an uncertainty around how to handle censoring.

54 We propose a new approach that takes advantage of recent developments in continuous relaxations of
55 sorting operations, allowing end-to-end training of neural networks with ordering supervision [Grover
56 et al., 2019, Blondel et al., 2020, Petersen et al., 2021]. This method incorporates a sorting algorithm
57 into the network architecture, where the order of the samples is known, but their exact values are
58 unsupervised. With this, we introduce *DiffSurv*, an extension of differentiable sorting methods that
59 enables end-to-end training of survival ranking models with censored data.

60 Briefly, our contributions are summarised:

- 61 • Our primary contribution is the extension of differentiable sorting methods to account for
62 censoring by introducing the concept of possible permutation matrices. (Section 3.1)
- 63 • We empirically demonstrate that our new differentiable sorting method matches or improves
64 risk ranking performance across multiple semi-simulated and real-worlds censored datasets.
65 (Section 4)
- 66 • We investigate the role of transitivity in survival analysis and show, through experiments
67 with semi-simulated data, that differentiable sorting networks can benefit from this inherent
68 property of the data. (Section 4)
- 69 • We demonstrate that differentiable sorting of censored data enables the development of new
70 methods with practical applications, using the example of end-to-end learning for top-k risk
71 stratification. (Section 3.2)

72 2 Survival Analysis and its Relation to Ranking

73 A dataset with censored event times is summarized as $\mathcal{D} = \{t_i, \mathbf{x}_i, \delta_i\}_{i=1}^N$, where N is the total
74 number of patients. For a patient i , the time-to-event t_i is the minimum of the true survival time
75 t_i^* and the censoring time c_i^* , with δ_i indicating whether an event ($t_i^* \leq c_i^*$, $\delta_i = 1$) or censoring
76 ($t_i^* > c_i^*$, $\delta_i = 0$) was observed. Covariates are $\mathbf{x}_i \in \mathbb{R}^d$ representing a d -dimensional vector but the
77 methods discussed here also generalise to higher dimensional tensors such as image data.

78 The widely-used method for addressing censoring in survival analysis is the Cox Partial Likelihood
79 (CPL) model, introduced by Cox [1972]. The CPL is designed to maximize the following general
80 form:

$$\mathcal{L}(\theta) := \prod_{i:\delta_i=1} \frac{f_{\theta}(\mathbf{x}_i)}{\sum_{j:t_j>t_i} f_{\theta}(\mathbf{x}_j)}, \quad (1)$$

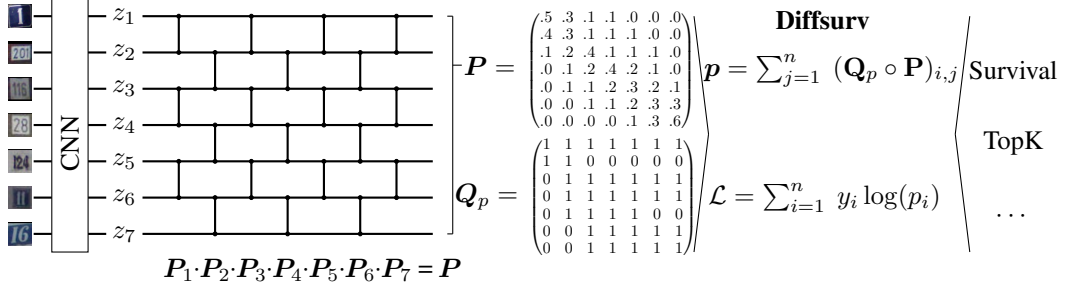


Figure 1: Differentiable Sorting for Censored Time-to-Event Data. Inputs, in this case SVHN images, are transformed into scalar values through a neural network. A differentiable permutation matrix, P , is computed using sorting networks. The model can be optimized for downstream tasks, such as risk stratification and top-k highest risk prediction, by using the matrix Q_p of possible permutations based on the observed events and censoring.

81 where f_θ is the hazard function, a score prediction function estimating the probability of an event at a
 82 particular time, given input features \mathbf{x}_i . The product only includes uncensored patients, whereas the
 83 denominator term also includes censored patients with $t_j > t_i$.

84 Reflecting the structure of survival data, the Cox Partial Likelihood (CPL) model compares individuals
 85 still "at risk" at each time point, similar to a nested case-control study. This directly shapes the
 86 likelihood equation in CPL, with the numerator representing the hazard function for the event-
 87 experiencing individual, and the denominator summing over all individuals still at risk.

88 Extensions of the Cox model, like [Katzman et al., 2018] and [Kvamme et al., 2019], have modified
 89 $f_\theta = \exp(\theta \cdot \mathbf{x}_i)$ to relax the linear covariate interaction and proportional hazards assumptions.
 90 Introducing neural networks h_θ to handle the non-linearity, adjusting f_θ to be $f_\theta = \exp(h_\theta(\mathbf{x}_i))$, and
 91 to manage non-proportional hazards, they set $f_\theta = \exp(h_\theta(\mathbf{x}_i, t_i))$.

92 2.1 Pairwise Independence

93 Both of these previous works note that the risk set $\mathcal{R} = \{j : t_j > t_i\}$ is intractable for deep learning
 94 applications as it considers all comparable patients. To mitigate memory constraints, we can sample
 95 a fixed-size risk set, denoted as $\tilde{\mathcal{R}}$, such that $|\tilde{\mathcal{R}}| = n < N$. Kvamme et al. [2019] go further, arguing
 96 it is reasonable to take a constant sample size of 1 and include the individual i in the risk set (such
 97 that $n = 2$). This leads to the simplified loss of the form

$$\mathcal{L}(\theta) = \prod_{i:\delta_i=1} \frac{f_\theta(\mathbf{x}_i)}{f_\theta(\mathbf{x}_i) + f_\theta(\mathbf{x}_{J(i)})}, \text{ where } J(i) \in \mathcal{R} \setminus \{i\}. \quad (2)$$

98 Further, take the mean log partial likelihood to be

$$\text{loss} = \frac{1}{n_e} \sum_{i:\delta_i=1} \log(1 + \exp[h_\theta(\mathbf{x}_{J(i)}) - h_\theta(\mathbf{x}_i)]), \text{ where } J(i) \in \mathcal{R} \setminus \{i\}, \quad (3)$$

99 where n_e is the number of non-censored events. In this simplified form, it can be seen that the partial
 100 likelihood only considers the pairwise relative ordering or ranking of survival times.

101 The concordance index or c-index Harrell et al. [1982] is a commonly used as an evaluation for survival
 102 analysis methods and is a generalization of the Area Under the Receiver Operating Characteristic
 103 Curve (AUROC) that handles right-censored data. It is defined as

$$\text{c-index} := \frac{1}{n} \sum_{i:\delta_i=1} \mathbb{1}(f(\mathbf{x}_i) < f(\mathbf{x}_j)), j \in \mathcal{R} \setminus \{i\}. \quad (4)$$

104 Raykar et al. [2007] first showed that the Cox's partial likelihood is approximately equivalent to
 105 maximizing the concordance index or c-index and that closer bounds can be found by minimizing the

106 general ranking loss, with acceptable pairs $\mathcal{A} = \{(i, j) : \delta_i = 1 \wedge t_j > t_i\}$ and

$$\text{ranking-loss} := \frac{1}{|\mathcal{A}|} \sum_{(i,j) \in \mathcal{A}} \phi(f_\theta(\mathbf{x}_i) - f_\theta(\mathbf{x}_j)), \quad (5)$$

107 where ϕ is a function that relaxes the non-differentiable $\mathbb{1}$ of the c-index. From Equation 3, it can be
 108 seen that $\phi : x \rightarrow -\log(1 + \exp(-x)) = \log(\sigma(x))$. Here, we have shown that the simplifications to
 109 the partial likelihood made by Kvamme et al. [2019] are equivalent to using the log-sigmoid ranking
 110 loss.

111 The key difference between ranking and partial likelihood losses comes when considering the
 112 assumption that it is reasonable to take a constant sample size of 2 (one pair in the risk set) in the
 113 partial likelihood. This effectively introduces the assumption that each pair (i, j) is independent of
 114 any other pair. However, this assumption seems puzzling given the inherent transitivity of ranking (if
 115 $i > j$ and $j > k$ then $i > k$).

116 2.2 Listwise Ranking and Differentiable Sorting Networks

117 [Cao et al., 2007] first proposed the notion of a *listwise* ranking loss, arguing its benefits in situations
 118 where the entire order of items is of importance. This approach treats the ranking problem as a
 119 permutation problem, with the aim of learning a model that can provide the optimal permutation of
 120 an entire list of items, rather than considering independent pairs. Indeed, this idea is closely related
 121 to the partial likelihood from Cox’s original work. As pointed out by Wang and Yang [2022], the
 122 Top-1 probability method originally proposed by Cao et al. [2007] and extended to ListMLE by Xia
 123 et al. [2008] takes the same form as CPL.

124 More recent works closely related to listwise ranking have begun to explore combining traditional
 125 sorting algorithms with differentiable sorting functions [Petersen et al., 2021]. Sorting networks are
 126 a family of sorting algorithms that consist of two basic components: wires and conditional swaps
 127 [Batcher, 1968, Knuth, 1998]. Wires carry values to be compared at conditional swaps. If one value
 128 is bigger than the other then the values carried forward are swapped around. This allows construction
 129 of *sorting networks* that can provably sort a list of values. Conditional swaps are exactly the min
 130 and max operators that ensure that with inputs $\{a, b\}$ and outputs $a^* \leq b^*$, $a^* = \min(a, b)$ and
 131 $b^* = \max(a, b)$. Examples of odd even and bitonic networks are shown in Appendix C.

132 In order to train models based on ordering information alone, differences between predicted and true
 133 orderings must be backpropagated through sorting algorithms. However, they often require the use of
 134 non-differentiable max and min operators. These are analogous to the non-differentiable indicator
 135 function that was discussed earlier in the c-index equation 4. Differentiable sorting methods, similar
 136 to ranking losses, rely on approximating these operators with smooth alternatives [Grover et al.,
 137 2019].

138 Petersen et al. [2021] propose combining traditional sorting networks and differentiable sorting
 139 functions. Consider that when a asymptotically approaches b , the transition point where it surpasses
 140 b is non-continuous and therefore non-differentiable. Just as previously shown in the ranking loss,
 141 such operations can be made differentiable using the logistic relaxation

$$\min_\sigma(a, b) = a \cdot \sigma(b - a) + b \cdot \sigma(a - b) \text{ and } \max_\sigma(a, b) = a \cdot \sigma(a - b) + b \cdot \sigma(b - a). \quad (6)$$

142 If an inverse temperature parameter $\beta > 0$ is introduced such that $\sigma : x \rightarrow \frac{1}{1 + e^{-\beta x}}$, then as $\beta \rightarrow \infty$
 143 the functions tend to the exact min and max functions. Other relaxations of the step function can
 144 also be considered, Petersen et al. [2022a] show that the Cauchy distribution preserves monotonicity
 145 which is desirable for optimization. Given this, we use the Cauchy distribution as our relaxation for
 146 all experiments, where $\sigma : x \rightarrow \frac{1}{\pi} \arctan(\beta x) + \frac{1}{2}$.

147 For an input list to be ordered, each layer of the sorting network can be considered an independent
 148 permutation matrix P_l with elements given by

$$P_{l,ii} = P_{l,jj} = \sigma(a_j - a_i) \text{ and } P_{l,ij} = P_{l,ji} = 1 - \sigma(a_j - a_i), \quad (7)$$

149 where a signifies intermediate values being compared. The first layer is input with $z_i = h_\theta(\mathbf{x}_i)$,
 150 each vector of covariates or images being processed independently by the same neural network. The

151 indices being compared at each layer are determined by the sorting network and the final predicted
 152 probability matrix is the product of each layer of sorting operations,

$$P = \left(\prod_{l=1}^n P_l^\top \right)^\top. \quad (8)$$

153 Where P , is the final doubly-stochastic permutation matrix, doubly-stochastic meaning that the rows
 154 and columns both sum to 1. It is possible to interpret each element P_{ij} of the predicted permutation
 155 matrix as the predicted probability of permuting from a randomly assigned rank i to a true rank
 156 j . Finally, we can define a loss by minimizing the cross-entropy between the ground truth orders
 157 represented by true permutation matrix Q and predicted permutation matrix P as

$$\mathcal{L} := \sum_{c=1}^n \left(\frac{1}{n} \text{CrossEntropy}(P_c, Q_c) \right), \quad (9)$$

158 where P_c and Q_c denote the c -th columns of their respective matrices.

159 2.3 Differentiable Sorting Networks Relation to Ranking and Partial Likelihood

160 It is possible to directly relate differentiable sorting networks with ranking losses and partial likelihood.
 161 Expanding out the cross entropy loss, we find

$$\mathcal{L} = \sum_{c=1}^n \left(\frac{1}{n} \sum_{i=1}^n q_{ic} \log(p_{ic}) \right), \quad (10)$$

162 where $q_{ic} = 1$ only when i is the true rank otherwise 0. Each p_{ic} is always a function of the difference
 163 in pairs of inputs x_i and x_j . This is complicated by the products of intermediate values a introduced
 164 by the sorting network but denoted as

$$p_{ic} = \prod_{(a_i, a_j) \in \mathcal{P}_l: l=1}^n \sigma(a_i - a_j) \quad (11)$$

165 where \mathcal{P}_l to denotes the set of comparisons to be made at each layer of the sorting network. With
 166 $n = 2$ and $\beta = 1$, a sorting network only requires a single relaxed conditional swap and the loss
 167 returns to the same recognisable log-sigmoid ranking loss in Equation 5, and Cox negative log partial
 168 likelihood in Equation 3.

169 3 Methods

170 3.1 DiffSurv: Handling Censoring with Possible Permutation Matrices

171 For risk sets of size 2, given proper case-control sampling, it will always be possible to define a single
 172 ground truth permutation matrix Q . However, when venturing to higher risk set sizes, differentiable
 173 sorting methods can no longer handle censoring since there is not a single ground truth permutation
 174 matrix Q . We cannot determine the exact rank of patients who are censored before another who
 175 experienced an event. It is only possible to know the range of possible ranks to which a patient should
 176 belong. In Figure 2, we provide an illustration demonstrating the possible ranks for a number of
 177 censored and uncensored events.

178 Though we no longer have access to a single permutation matrix, we may instead consider the
 179 set of all possible permutation matrices, $\mathcal{Q} = \{Q_1, Q_1, \dots, Q_\kappa\}$. In the best case, all values are
 180 uncensored and $|\mathcal{Q}| = 1$ and in the worse case, when all patients are censored $|\mathcal{Q}| = n!$. Our primary
 181 contribution is to extend differentiable sorting methods to censored ranks by discriminating between
 182 possible and impossible permutations.

183 We introduce a more computationally tractable representation of \mathcal{Q} by defining the *possible permuta-*
 184 *tion matrix*, Q_p , which is the element-wise maximum of every permutation in \mathcal{Q} ,

$$Q_{pij} = \max\{Q_{1ij}, Q_{2ij}, \dots, Q_{\kappa ij}\}. \quad (12)$$

185 For survival analysis, it is possible to determine Q_p in linear time given a sorted list of event times t_i
 186 and event indicators δ_i . We will consider higher ranks to correspond with a smaller time-to-event.
 187 Let us consider two scenarios:

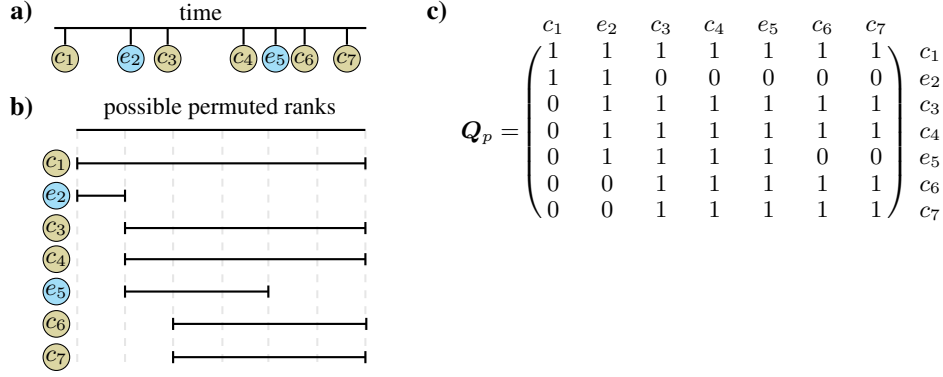


Figure 2: For an example case (a) with two events (\circ , e_1 and e_5) and multiple censored samples (\circ , c_1, c_3, c_4, c_6, c_7) the uncertainty in the possible permuted rankings (b) due to censoring is taken into account to derive the possible permutation matrix Q_p (c).

- 188 1. For a right-censored individual i (i.e., $\delta_i = 0$), the possible ranks must be lower than the
 189 ranks of preceding uncensored events. We can express the set of possible ranks \mathcal{R}_i as:

$$\mathcal{R}_i = \{r \mid r < \text{rank}(j), \forall j : \delta_j = 1, t_j < t_i\} \quad (13)$$

190 This set includes all ranks r that are less than the rank of any uncensored patient j with an
 191 event time t_j preceding the censoring time t_i of the individual i .

- 192 2. For an uncensored individual i (i.e., $\delta_i = 1$), the possible rank must be lower than all
 193 preceding uncensored events and higher than the ranks of all subsequent events. We can
 194 define the set of possible ranks as:

$$\mathcal{R}_i = \{r \mid r < \text{rank}(j), \forall j : \delta_j = 1, t_j < t_i\} \cap \{r \mid r > \text{rank}(j), \forall j : t_j > t_i\} \quad (14)$$

195 This set includes all ranks r that satisfy both conditions: being less than the rank of any
 196 preceding uncensored patient j with $t_j < t_i$ and greater than the rank of any subsequent
 197 patient j with $t_j > t_i$.

198 With these observations, it's straightforward to construct Q_p . If it's feasible for patient i to permute to
 199 rank j , i.e. $j \in \mathcal{R}_i$, then $Q_{pij} = 1$, otherwise $Q_{pij} = 0$. See Figure 2 (c) for a visual representation
 200 of Q_p .

201 Given the possible permutation matrix Q_p and the predicted permutation matrix P , the vector of
 202 probabilities p of a value being ranked within the set of possible ranks can be computed. Although
 203 the ground truth probabilities are unknown, the range of possible ranks is known, and the model can
 204 be optimized to maximize the sum of the predicted probabilities of all possible ranks for each sample.
 205 Noted here as the column-sum of the element-wise product \circ , between Q_p and P .

$$p = \sum_{j=1}^n (Q_p \circ P)_{i,j}. \quad (15)$$

206 The binary cross-entropy loss can then be easily applied

$$\mathcal{L} = \sum_{i=1}^n -y_i \log(p_i) - (1 - y_i) \log(1 - p_i) \quad (16)$$

207 where y_i indicates whether set of predicted ranks is possible or impossible.

208 Equation 15 accounts for the potential challenges of incorporating right-censored samples. The binary
 209 cross-entropy remains identical whether the model predicts uniform probability for all possible ranks
 210 or concentrates the probability mass on a single rank possible rank.

211 The introduction of the possible permutation matrix can be used in conjunction with any differentiable
 212 sorting method that outputs a doubly-stochastic permutation matrix. This includes methods such

213 as SinkhornSort from Cuturi et al. [2019]. Though, in this paper, we will restrict our focus to the
 214 discussed differentiable sorting networks. We refer to the use of differentiable sorting networks and
 215 the possible permutation matrix as *DiffSurv*.

216 3.2 Top-K risk prediction

217 Finally, we demonstrate how the algorithmic supervision of sorting algorithms enables the develop-
 218 ment of novel methods in survival analysis, using the example of top-k risk prediction. In practical
 219 settings, it is often not necessary to rank all samples correctly. Rather, it is essential to identify the
 220 samples with the highest risk, such as by a healthcare provider, to prioritize care and interventions.

221 With *DiffSurv*, top-k risk prediction is straightforward to implement by modifying the loss such that
 222 the negative log-likelihood of predicted top-k ranks in \mathbf{P} is maximised for individuals with a possible
 223 permutation to *any* top-k rank according to \mathbf{Q}_p .

224 First, let's denote \mathcal{T}_k as the set of values with a possible permutation to a top-k rank, derived from the
 225 ground truth possible permutation matrix \mathbf{Q}_p :

$$\mathcal{T}_k = \{i \mid \sum_{j=1}^k \mathbf{Q}_{pij} > 0\} \quad (17)$$

226 Importantly, due to the uncertainty introduced by censoring, the set of individuals with a possible
 227 permutation to a top k rank \mathcal{T}_k can be arbitrarily large. For example, in case all individuals are
 228 censored, \mathcal{T}_k is the set of all individuals. Then, the top-k loss is described as:

$$\mathcal{L}_{\text{top-k}} = - \sum_{i \in \mathcal{T}_k} \log \left(\sum_{j=1}^k \mathbf{P}_{ij} \right). \quad (18)$$

229 This loss is minimized when the model correctly predicts a top-k rank for the indices in \mathcal{T}_k . This
 230 represents the individuals with possible permutations to the top-k highest risk ranks. Importantly,
 231 this loss function is optimized for the identification of potential top-k high-risk individuals, without
 232 considering the specific order within these top-k ranks. To establish a baseline for comparison with
 233 *DiffSurv*'s top-k risk prediction, we also introduce two variants of the Cox Partial Likelihood method.
 234 In the first variant, we adjust the likelihood term so that the product considers only the set of patients
 235 who have a potential permutation to a top-k rank, according to the matrix of possible permutations
 236 \mathbf{Q}_p :

$$\mathcal{L}_{\text{CPL}_I} = \prod_{i: i \in \mathcal{T}_k} \frac{f_{\theta}(\mathbf{x}_i)}{\sum_{j: T_j > T_i} f_{\theta}(\mathbf{x}_j)} \quad (19)$$

237 In the second variant, we further limit the set of patients to those who have both experienced an event
 238 and have a possible permutation to a top-k rank:

$$\mathcal{L}_{\text{CPL}_{II}} = \prod_{i: \delta_i = 1 \wedge i \in \mathcal{T}_k} \frac{f_{\theta}(\mathbf{x}_i)}{\sum_{j: T_j > T_i} f_{\theta}(\mathbf{x}_j)}. \quad (20)$$

239 Note that the denominator term is unchanged in both variants and considers only comparable pairs
 240 and includes censored patients $T_j > T_i$. Evaluation of top-k risk prediction is also complicated by
 241 the uncertainty due to censoring. For *DiffSurv* and both variants of the Cox Partial Likelihood, we
 242 can first define the set of individuals predicted to be within the top-k highest risk:

$$\mathcal{P}_k = \{i \mid \text{rank}(f_{\theta}(\mathbf{x}_i)) \geq k\} \quad (21)$$

243 We can then define the fraction of how many of these individuals are in the set of possible top-k
 244 highest ranks \mathcal{T}_k to evaluate the top-k risk prediction performance:

$$\text{top-k-score} = \frac{|\mathcal{P}_k \cap \mathcal{T}_k|}{|\mathcal{P}_k|} \quad (22)$$

245 4 Experiments

246 In our experiments, we aim to assess the performance of *DiffSurv* and compare it against the con-
 247 ventional Cox Partial Likelihood (CPL) methods. Initially, we focus on confirming the importance

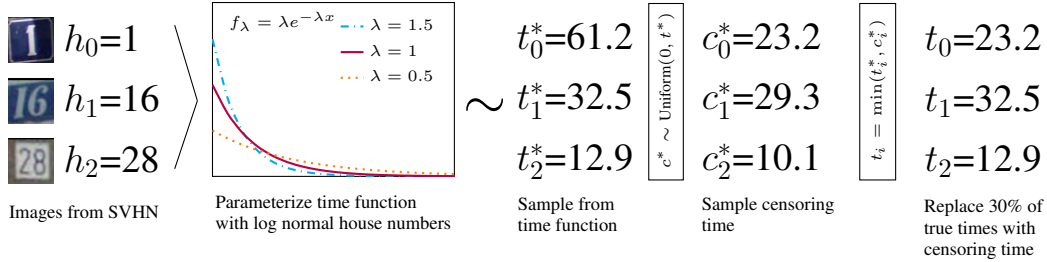


Figure 3: Visual abstract of the survSVHN dataset.

248 of taking a listwise approach and evaluating the ability of differentiable sorting networks to better
 249 capture the inherent transitivity in semi-simulated data. Subsequently, we extend our analysis to
 250 compare DiffSurv and its top-k extension across multiple publicly available real-world datasets.

251 **Baselines:** We compare *DiffSurv* primarily against Cox’s Partial Likelihood, using the Ranked List
 252 implementation from *pycox* [Kvamme et al., 2019]. We include Efron and Breslow estimates of CPL
 253 from Yang et al. [2022] for survSVHN. For smaller datasets, we add non-deep learning baselines:
 254 Lifelines’ Cox Regression [Davidson-Pilon, 2019] and *skSurv*’s Random Survival Forests [Pölsterl,
 255 2020]. We do not compare with DeepHit [Lee et al., 2018] since we do not model non-proportional
 256 hazards. For an extended discussion, see Appendix A.

257 **Network Architectures:** For both CPL and *DiffSurv*, we use a fixed neural network architecture
 258 depending on the dataset. Small datasets utilize a single-layer neural network, survSVHN uses a
 259 ConvNet architecture as in Petersen et al. [2021], and MIMIC IV CXR uses EfficientNet-B0 [Tan
 260 and Le, 2020].

261 **Training and Evaluation:** We employ AdamW for optimization. Validation approach varies:
 262 for smaller datasets, we apply nested 5-fold cross-validation, while for imaging datasets we use
 263 train:val:test splits. We performed hyperparameter tuning for learning rate, weight decay, batch
 264 size, and risk set size. In the case of imaging datasets, we maintained fixed values for learning rate
 265 and weight decay. As in Petersen et al. [2021], we determine steepness as a function of the risk
 266 set size n , $\beta = 2n$ for odd-even and $\beta = (\log_2 n)(1 + \log_2 n)$ for bitonic. The type of sorting
 267 network can either be bitonic or odd-even and is determined during hyperparameter tuning. Further
 268 details on the experimental setup, including compute time, are provided in Appendix B and at
 269 <https://github.com/andre-vauvelle/diffSurv>.

270 **Semi-synthetic survSVHN:** Based on the Street View House Numbers (SVHN) dataset [Netzer et al.,
 271 2011], we simulate survival times akin to survMNIST Pölsterl [2019]. The increased complexity
 272 of SVHN over MNIST offers a testbed which is better able to discern the performance differences
 273 between methods. Each house number parameterizes a beta-exponential time function for survival
 274 times. Risk parameters or hazards λ_i are calculated as the logarithm of house numbers, standardized
 275 and scaled for a mean survival time of 30. We introduce censoring by randomly selecting 30% of
 276 house numbers and replacing true times with values sampled uniformly between $(0, t_i]$ (See Figure 3).

277 We can examine the implications of inherent transitivity within the data. Instead of parameterizing
 278 a time function based on unique hazards derived from house numbers, we group λ_i into distinct
 279 hazard quantiles. Each quantile encompasses a set of house numbers associated with a similar hazard
 280 level. We then calculate the transitivity ratio, defined as $\frac{\# \text{ of transitive triplets}}{\# \text{ of triplets}}$, where a sampled triplet is
 281 considered transitive if $(\lambda_i > \lambda_j > \lambda_k)$.

282 This methodology provides us with a means to control the degree of transitivity in our data. At one
 283 extreme, we might categorize data into only two groups, representing the lower and upper halves of
 284 house numbers, which results in a transitivity ratio of 0. At the other extreme, each house number
 285 could constitute its own unique category (indicated by ∞), leading to high transitivity.

286 Our results, summarized in Table 1, align with the expectations laid out in Section 2.3: both *DiffSurv*
 287 and CPL methods perform similarly when the risk set size is at its minimum ($n = 2$). However,
 288 with the expansion of the risk set size, the performance of the two methods diverges, with *DiffSurv*
 289 consistently outperforming CPL. Table 2 sheds light on a potential reason for this divergence. As

Table 1: Results for training on survSVHN with increasing risk set size. Mean (standard deviation) c-index over 5 trails with different seeds. [†] When $n = 2$ both methods are equivalent to the ranking loss to up continuous relaxation of swap operation.

Risk set size	2 [†]	4	8	16	32
Diffsurv	.918 (.003)	.934 (.002)	.940 (.001)	.943 (.002)	.941 (.002)
Cox Partial Likelihood	.913 (.002)	.925 (.002)	.931 (.002)	.933 (.002)	.930 (.003)

Table 2: Results for training on survSVHN while increasing transitivity. Metric is c-index. Mean performance over 3 trails with different seeds. **Bold** indicates significant improvement (t-test, $p \leq 0.01$). Restricted to a fixed batch size and risk set size of 32.

Number of Quantiles	2	4	8	16	32	64	128	∞
Transitivity Ratio	.0	.374	.657	.819	.908	.954	.975	.991
Diffsurv: Bitonic	.643	.803	.882	.922	.933	.939	.939	.939
Diffsurv: Odd Even	.646	.802	.883	.923	.935	.939	.940	.941
CPL: Ranked List	.651	.803	.880	.909	.916	.920	.921	.920
CPL: Efron	.647	.801	.871	.898	.904	.905	.909	.908
CPL: Breslow	.648	.801	.871	.898	.904	.909	.907	.910

290 the number of quantiles is increased, thereby enhancing the degree of transitivity within the data,
 291 Diffsurv-based methods start to surpass CPL methods. This finding underscores the role of transitivity
 292 in survival data and validates Diffsurv’s effectiveness in encapsulating this inherent property. Despite
 293 the strong performance of Diffsurv, the C-index for the ground truth risks is 0.980, which is still far
 294 above 0.943 for Diffsurv, highlighting the challenging nature of the survSVHN dataset.

295 **Real-world datasets:** We assess our methods on several public datasets: Four small, popular real-
 296 world survival datasets (FLCHAIN, NWTCO, SUPPORT, METABRIC) [Kvamme et al., 2019] and
 297 the MIMIC IV Chest X-Ray dataset (CXR) with death as the event [Johnson et al., 2019]. Further
 298 details in Appendix B.1.

299 The results presented in Table 3 demonstrate that Diffsurv achieves equal to or better performance
 300 on all datasets analyzed. Additionally, when Diffsurv is optimized for predicting the top 10% of
 301 highest-risk individuals, it matches or outperforms Cox’s partial likelihood on the real-world datasets.

302 5 Conclusion

303 Diffsurv introduces a new perspective in survival analysis with censored data, highlighting the
 304 relations between survival analysis and the listwise ranking. Our experiments show the effectiveness
 305 of differentiable sorting methods for improving survival analysis predictions. Notably, Diffsurv
 306 matches or surpasses the performance of the established CPL methods across all examined datasets.

307 Crucially, Diffsurv sheds light on the importance of transitivity in ranking and survival data, revealing
 308 that methods sensitive to this inherent property, such as Diffsurv, show improved performance over
 309 those that are not. This insight underscores the value of a listwise approach in dealing with survival
 310 data and encourages further exploration for methods that promote a transitive inductive bias.

311 Moreover, Diffsurv provides a foundation for the development of innovative methods, including the
 312 top-k risk stratification method introduced in this work. Beyond survival analysis, the introduction of
 313 the possible permutations carries potential for other tasks that involve ranking based on limited order
 314 information. The utilization of specialized sorting networks, such as splitter selection networks as in
 315 Petersen et al. [2022b], could further leverage partial order information.

316 Though promising, this work is not without limitations. Future research could focus on extending
 317 its applicability to non-proportional hazards and understanding the impact of ties. Moreover, in-
 318 vestigating how well it can recover survival functions using approaches like Breslow’s estimator
 319 and evaluating with Brier scores would provide valuable insights into its potential and limitations.

Table 3: Results for real-world and semi-synthetic datasets. Mean (standard deviation). Survival metric is c-index, Top 10% metric is top-k-score. **Bold** indicates significant improvement (t-test, $p \leq 0.01$).

	FLCHAIN	NWTCO	SUPPORT	METABRIC	MIMIC IV CXR	survSVHN
Size	6,524	4,028	8,873	1,904	377,110	248,823
Censored Proportion	69.9%	85.8%	32.0%	42.1%	60.9%	30.0%
Survival						
Cox Regression	.750 (.083)	.692 (.021)	.598 (.010)	.628 (.013)	-	-
Random Survival Forest	.789 (.011)	.691 (.024)	.614 (.009)	.641 (.012)	-	-
Cox Partial Likelihood	.794 (.013)	.709 (.015)	.642 (.006)	.698 (.011)	.760 (.002)	.933 (.002)
DiffSurv	.793 (.009)	.703 (.026)	.645 (.002)	.684 (.011)	.763 (.001)	.943 (.002)
Top 10% prediction						
Cox Partial Likelihood	.460 (.013)	.390 (.068)	.280 (.023)	.249 (.065)	.390 (.010)	-
CPL-TopK (Variant I)	.469 (.007)	.413 (.061)	.479 (.016)	.527 (.083)	.408 (.008)	-
CPL-TopK (Variant II)	.460 (.009)	.413 (.054)	.479 (.035)	.487 (.058)	.406 (.006)	-
DiffSurv	.452 (.011)	.395 (.082)	.296 (.015)	.331 (.102)	.412 (.002)	-
DiffSurv-TopK	.482 (.019)	.421 (.065)	.508 (.027)	.533 (.092)	.412 (.009)	-

320 Furthermore, it is important to note that DiffSurv is a survival ranking method and thus can not be
321 used to directly estimate the expected duration until an event occurs.

322 Overall, DiffSurv constitutes a meaningful advancement in survival analysis, showcasing its significant
323 potential for enhancing risk prediction in real-world use cases. It not only demonstrates promising
324 performance improvements, but also introduces new directions for future research, thereby making a
325 valuable contribution to the field.

326 6 Acknowledgements

327 We extend our gratitude to Felix Peterson, Leon Sixt, and Samuel Holt for their insightful discussions
328 and invaluable feedback, which significantly contributed to the quality of this work. We also thank
329 the anonymous reviewers for their thoughtful feedback and fruitful discussions, which have been
330 instrumental in enhancing the manuscript.

331 **References**

- 332 Laura Antolini, Patrizia Boracchi, and Elia Biganzoli. A time-dependent discrimination index for
333 survival data. *Statistics in Medicine*, 24(24):3927–3944, 2005. ISSN 1097-0258. doi: 10.1002/sim.
334 2427. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.2427>. _eprint:
335 <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.2427>.
- 336 K. E. Batchler. Sorting networks and their applications. In *Proceedings of the April 30–May 2, 1968,*
337 *spring joint computer conference on - AFIPS '68 (Spring)*, page 307, Atlantic City, New Jersey,
338 1968. ACM Press. doi: 10.1145/1468075.1468121. URL <http://portal.acm.org/citation.cfm?doid=1468075.1468121>.
- 340 Mathieu Blondel, Olivier Teboul, Quentin Berthet, and Josip Djolonga. Fast Differentiable Sorting
341 and Ranking, June 2020. URL <http://arxiv.org/abs/2002.08871>. arXiv:2002.08871 [cs,
342 stat].
- 343 Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise
344 approach to listwise approach. In *Proceedings of the 24th international conference on Machine*
345 *learning*, pages 129–136, Corvallis Oregon USA, June 2007. ACM. ISBN 978-1-59593-793-3. doi:
346 10.1145/1273496.1273513. URL <https://dl.acm.org/doi/10.1145/1273496.1273513>.
- 347 D. R. Cox. Regression Models and Life-Tables. *Journal of the Royal Statistical Society. Series*
348 *B (Methodological)*, 34(2):187–220, 1972. ISSN 0035-9246. URL <https://www.jstor.org/stable/2985181>. Publisher: [Royal Statistical Society, Wiley].
- 350 Marco Cuturi, Olivier Teboul, and Jean-Philippe Vert. Differentiable Ranking and Sorting using
351 Optimal Transport. In *Advances in Neural Information Processing Systems*, volume 32. Curran
352 Associates, Inc., 2019. URL https://papers.nips.cc/paper_files/paper/2019/hash/d8c24ca8f23c562a5600876ca2a550ce-Abstract.html.
- 354 Cameron Davidson-Pilon. lifelines: survival analysis in Python. *Journal of Open Source Software*,
355 4(40):1317, August 2019. ISSN 2475-9066. doi: 10.21105/joss.01317. URL <https://joss.theoj.org/papers/10.21105/joss.01317>.
- 357 Larry Goldstein and Bryan Langholz. Asymptotic Theory for Nested Case-Control Sam-
358 pling in the Cox Regression Model. *The Annals of Statistics*, 20(4):1903–1928, De-
359 cember 1992. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1176348895. URL
360 [https://projecteuclid.org/journals/annals-of-statistics/volume-20/](https://projecteuclid.org/journals/annals-of-statistics/volume-20/issue-4/Asymptotic-Theory-for-Nested-Case-Control-Sampling-in-the-Cox/10.1214/aos/1176348895.full)
361 [issue-4/Asymptotic-Theory-for-Nested-Case-Control-Sampling-in-the-Cox/](https://projecteuclid.org/journals/annals-of-statistics/volume-20/issue-4/Asymptotic-Theory-for-Nested-Case-Control-Sampling-in-the-Cox/10.1214/aos/1176348895.full)
362 [10.1214/aos/1176348895.full](https://projecteuclid.org/journals/annals-of-statistics/volume-20/issue-4/Asymptotic-Theory-for-Nested-Case-Control-Sampling-in-the-Cox/10.1214/aos/1176348895.full). Publisher: Institute of Mathematical Statistics.
- 363 Aditya Grover, Eric Wang, Aaron Zweig, and Stefano Ermon. Stochastic Optimization of Sorting
364 Networks via Continuous Relaxations. *arXiv:1903.08850 [cs, stat]*, April 2019. URL <http://arxiv.org/abs/1903.08850>. arXiv: 1903.08850.
- 366 Frank E. Harrell, Jr, Robert M. Califf, David B. Pryor, Kerry L. Lee, and Robert A. Rosati.
367 Evaluating the Yield of Medical Tests. *JAMA*, 247(18):2543–2546, May 1982. ISSN 0098-
368 7484. doi: 10.1001/jama.1982.03320430047030. URL <https://doi.org/10.1001/jama.1982.03320430047030>.
- 370 Alistair E. W. Johnson, Tom J. Pollard, Seth J. Berkowitz, Nathaniel R. Greenbaum, Matthew P.
371 Lungren, Chih-ying Deng, Roger G. Mark, and Steven Horng. MIMIC-CXR, a de-identified
372 publicly available database of chest radiographs with free-text reports. *Scientific Data*, 6(1):
373 317, December 2019. ISSN 2052-4463. doi: 10.1038/s41597-019-0322-0. URL <https://www.nature.com/articles/s41597-019-0322-0>. Number: 1 Publisher: Nature Publishing
374 Group.
375
- 376 Jared L. Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yu-
377 val Kluger. DeepSurv: personalized treatment recommender system using a Cox proportional
378 hazards deep neural network. *BMC Medical Research Methodology*, 18(1):24, February 2018.
379 ISSN 1471-2288. doi: 10.1186/s12874-018-0482-1. URL <https://doi.org/10.1186/s12874-018-0482-1>.
- 380

- 381 Donald E Knuth. The Art of Computer Programming. *Addison Wesley Longman Publishing Co., Inc.*,
382 Volume 3: (2nd Ed.) Sorting and Searching, 1998.
- 383 Håvard Kvamme and Ørnulf Borgan. The Brier Score under Administrative Censoring: Problems and
384 Solutions. Technical Report arXiv:1912.08581, arXiv, December 2019. URL <http://arxiv.org/abs/1912.08581>. arXiv:1912.08581 [cs, stat] type: article.
- 386 Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-Event Prediction with Neural Networks
387 and Cox Regression. *arXiv:1907.00825 [cs, stat]*, September 2019. URL <http://arxiv.org/abs/1907.00825>. arXiv: 1907.00825.
- 389 Changhee Lee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar. DeepHit: A Deep Learn-
390 ing Approach to Survival Analysis with Competing Risks. *Proceedings of the AAAI Conference on*
391 *Artificial Intelligence*, page 8, 2018.
- 392 Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading
393 digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learn-*
394 *ing and Unsupervised Feature Learning 2011*, 2011. URL http://ufldl.stanford.edu/housenumbers/nips2011_housenumbers.pdf.
- 396 Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Differentiable Sorting Networks
397 for Scalable Sorting and Ranking Supervision. *arXiv:2105.04019 [cs]*, July 2021. URL <http://arxiv.org/abs/2105.04019>. arXiv: 2105.04019.
- 399 Felix Petersen, Christian Borgelt, Hilde Kuehne, and Oliver Deussen. Monotonic Differentiable
400 Sorting Networks. *arXiv:2203.09630 [cs, stat]*, March 2022a. URL <http://arxiv.org/abs/2203.09630>. arXiv: 2203.09630.
- 402 Felix Petersen, Hilde Kuehne, Christian Borgelt, and Oliver Deussen. Differentiable Top-k
403 Classification Learning, June 2022b. URL <http://arxiv.org/abs/2206.07290>. Number:
404 arXiv:2206.07290 arXiv:2206.07290 [cs].
- 405 Sebastian Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn.
406 *Journal of Machine Learning Research*, 21(212):1–6, 2020. URL <http://jmlr.org/papers/v21/20-729.html>.
- 408 Sebastian Pölsterl. Survival Analysis for Deep Learning, July 2019. URL <https://k-d-w.org/blog/2019/07/survival-analysis-for-deep-learning/>.
- 410 Vikas C Raykar, Harald Steck, Balaji Krishnapuram, Cary Dehing-oberije, and Philippe Lambin. On
411 Ranking in Survival Analysis: Bounds on the Concordance Index. In *Advances in Neural Infor-*
412 *mation Processing Systems*, volume 20. Curran Associates, Inc., 2007. URL <https://papers.nips.cc/paper/2007/hash/33e8075e9970de0cfea955afd4644bb2-Abstract.html>.
- 414 Mingxing Tan and Quoc V. Le. EfficientNet: Rethinking Model Scaling for Convolutional Neural
415 Networks, September 2020. URL <http://arxiv.org/abs/1905.11946>. arXiv:1905.11946
416 [cs, stat].
- 417 Bokun Wang and Tianbao Yang. Finite-Sum Coupled Compositional Stochastic Optimization:
418 Theory and Applications, September 2022. URL <http://arxiv.org/abs/2202.12396>.
419 arXiv:2202.12396 [cs, math].
- 420 Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise Approach to Learning
421 to Rank - Theory and Algorithm. *Proceedings of the 25th International Conference on Machine*
422 *learning*, 2008.
- 423 Xuelin Yang, Louis Abraham, Sejin Kim, Petr Smirnov, Feng Ruan, Benjamin Haibe-Kains, and
424 Robert Tibshirani. FastCPH: Efficient Survival Analysis for Neural Networks, August 2022. URL
425 <http://arxiv.org/abs/2208.09793>. arXiv:2208.09793 [cs, stat].

426 A Non-proportional Hazards

427 Our current implementation of DiffSurv operates under the proportional hazards assumption. While
428 this may not fully capture the intricacies of some survival analysis problems—particularly those
429 involving non-proportional hazards—it does not necessarily limit the model’s effectiveness in sce-
430 narios where the goal is to assess cumulative risk from a fixed index date or from the date of an
431 imaging study. This aligns with the necessity of time-dependent modifications to the C-index for
432 non-proportional models as indicated by Antolini et al. [2005].

433 If we are primarily interested in understanding the cumulative hazard of an event occurring rather
434 than tracking changes in the hazard over time, the assumption of proportional hazards becomes less
435 pivotal. As such, DiffSurv and CPL remain valuable tools for these cases.

436 Despite the current limitation of DiffSurv to proportional hazards, it is conceivable that an extension
437 to accommodate non-proportional hazards could be developed, similar to adaptations made for the
438 CPL method.

439 For instance, as we briefly mentioned earlier, continuous-time extensions of partial likelihood can be
440 used to enable non-proportional hazards Kvamme et al. [2019]. Implemented by directly modeling
441 temporal covariates as $f_\theta = \exp(h_\theta(x_i, T_i))$.

442 Another class of methods focuses on discretizing the time-to-event variable and modeling the
443 probability mass function (PMF) of event times. For instance, the DeepHit model Lee et al. [2018]
444 employs a neural network architecture to learn the relationships between input features and discretized
445 time-to-event outcomes. Time discretization facilitates modeling of non-proportional hazards but
446 introduces two significant challenges: 1) sensitivity to the choice of time intervals, which can affect
447 the model’s accuracy and interpretability, and 2) increased computational complexity, as predictions
448 must be made for each time interval. These models can be computationally expensive, especially
449 for deep learning-based models like DeepHit, making them less suitable for high-dimensional and
450 large-scale datasets, such as the imaging dataset used in this study.

451 Several future work proposals arise from these observations. First, differentiable sorting could explore
452 the approach of directly modeling temporal covariates, resulting in a time-parameterized predicted
453 permutation matrix. Second, extending DiffSurv to discrete time could be achieved by parameterizing
454 a predicted permutation matrix for each time discretization.

455 B Training and evaluation

456 B.1 Datasets and Preprocessing

457 As in Goldstein and Langholz [1992] and Kvamme et al. [2019], we ensure that each risk set contains
458 a valid risk set by sampling controls for a given case. Each batch consists of a number of risk sets
459 such that the input data has shape (batch size, risk set size, covariate shape).

460 We provide an additional description of each small realworld dataset:

- 461 • **FLCHAIN dataset:** A dataset containing information on patients with monoclonal gam-
462 mopathy of undetermined significance (MGUS), focusing on serum free light chain (FLC)
463 levels to study their prognostic significance in predicting disease progression. Number of
464 covariates: 8.
- 465 • **NWTCO dataset:** A dataset from a series of clinical trials on the treatment and outcomes
466 of children with Wilms’ tumor, a type of kidney cancer, aiming to improve understanding of
467 tumor biology and optimize treatment strategies. Number of covariates: 9.
- 468 • **SUPPORT dataset:** A dataset from a multi-center study investigating the prognosis and
469 treatment preferences of seriously ill hospitalized adults, with the goal of improving end-of-
470 life care and informing decision-making processes. Number of covariates: 22.
- 471 • **METABRIC dataset:** A dataset comprising genomic and clinical data on breast cancer
472 patients, focused on uncovering novel molecular subtypes for more precise prognostication
473 and personalized treatment strategies. Number of covariates: 9.

474 Covariate preprocessing follows [Kvamme et al., 2019], and includes standardising continuous
475 variables and one-hot encoding categorical variables.

476 **MIMIC IV CXR:** For the survival task, we extract death events from the MIMIC IV dataset. This is
477 done by merging the data on "subject_id", "study_id", and "dicom_id" with the patient table from
478 MIMIC IV, and on "subject_id" with the admission table. For patients without a recorded date of
479 death, censoring dates are determined as 1 year after the last recorded discharge date for each
480 patient. We exclude 29,345 images without any matches in the MIMIC IV patient table, 19,337
481 images taken after the latest found discharge date and 55 images taken after a recorded date of death.
482 Time to event is calculated as the number of days from the image study date to either date of death or
483 the censoring date. For MIMIC IV CXR, images undergo several standard transformations: a random
484 horizontal flip and a 15-degree rotation, resizing to 230 x 230 pixels, a 224 x 224 pixel center crop,
485 and conversion to grayscale with three output channels. The data is then transformed into tensors and
486 normalized using ImageNet’s mean and standard deviation values. Finally, the train:val:test split of
487 8:1:1 is done at the patient level ensuring no images from a patient in the test set was found in the
488 training data.

489 **survSVHN:** In this semi-synthetic dataset, we sample survival times using the beta-exponential
490 distribution. The beta distribution used to sample from the exponential uses a fixed value of 500
491 for both shape parameters. We follow Petersen et al. [2021] by cropping the centered multi-digit
492 numbers with a boundary of 30%, resizing it to a resolution of 64x64, and then selecting 54 x 54
493 pixels at a random location. For survSVHN the train:val:test split is provided by Netzer et al. [2011]
494 as is 230,755:5,000:13,068.

495 B.2 Model Architecture and Hyperparameters

496 For the smaller real-world datasets, the hazard function f_θ is a small fixed Multi-layer Perceptron
497 network with 1 hidden layer and 64 hidden nodes. We also apply a fixed dropout rate of 0.1. Learning
498 rate, weight decay, batch set size and risk set size were found using a grid search across the possible
499 values in Table 4.

Table 4: Hyperparameter values for small real-world datasets.

Hyperparameter	Values
Learning rate	[0.1, 0.01, 0.001, 1e-4]
Weight decay	[0.1, 0.01, 0.001, 1e-4, 1e-5, 0]
(Batch size, risk set size)	[(32, 8), (16, 16), (8, 32), (4, 64), (1, 256)]

500 For imaging datasets, we fix learning rate and weight decay for both CPL and DiffSurv. For both
501 survSVHN and MIMIC IV CXR, we use a fixed learning rate of 10^{-4} and weight decay of 10^{-5} . We
502 also used early stopping with a patience of 20 epochs and a maximum of 100,000 training steps.

503 **survSVHN:** As per Petersen et al. [2021], the model consists of four convolutional layers (with a 5x5
504 kernel size and 32, 64, 128, 256 filters), each followed by ReLU and max-pooling (2x2 stride). The
505 architecture concludes with a fully connected layer of 64 units, another ReLU, and a one-unit output
506 layer.

507 **MIMIC IV CXR:** Here, we use EfficientNet-B0 with an added linear layer for single output. We
508 first train the linear prediction layer alone for the initial 2,000 steps. After this, we continue training,
509 this time including both the EfficientNet-B0 and the linear prediction layer.

510 For the results in Table 1, we keep a fixed batch size of 100. We also provide a comparison where the
511 number of values is fixed in each batch in Table 9.

512 Note that during evaluation, the sorting network is not used since we only need to evaluate the ranks
513 of the trained risk scores. Similarly, case-control sampling is not used. We measure the ranking
514 performance of the models using the concordance index.

515 Further implementation details and the best hyperparameters for each dataset are provided at anon@
516 git.com.

517 B.3 Compute Requirements

518 Experiments on smaller real-world datasets are compact enough to facilitate effective training on a
519 CPU, with each variant, including the CPH baselines, completing per experiment in less than 20 min-
520 utes. However, the larger imaging datasets require more significant computational power. In the most
521 demanding case, the MIMIC IV CXR experiments, run on an 11GB NVIDIA GeForce GTX 1080 Ti,
522 took roughly 18.5 hours per experiment. Both DiffSurv and CPH methods exhibited comparable run
523 times; however, the Bitonic variant was the fastest, with a lead of approximately 6 minutes. All neural
524 network baselines were implemented using PyTorch and PyTorch Lightning. Although measures
525 were taken to reduce compute time and complexity, such as using half-precision and distributed data
526 parallel (DDP) training strategies, the overall training times are far from optimized.

527 To understand the runtime differences of DiffSurv and Cox Partial Likelihood variants in isolation,
528 we also run an additional experiment. For each method, we compute and time over 100 trials for a
529 forward and backward pass on a NVIDIA GeForce GTX 1080 Ti using randomly generated logits.
530 For DiffSurv, this includes computing the predicted permutation matrix with differentiable sorting
531 networks and applying the masking (Equation 15) and binary cross entropy (Equation 16). The
532 possible permutation matrix generation is possible to precompute off GPU on the dataloader, so
533 is not included in the timing. In Table 5, the DiffSurv methods, particularly Bitonic, consistently
534 outperform CPL methods in compute time almost across all batch sizes and risk set sizes. As risk set
535 size increases, CPL methods exhibit a decreasing trend in compute time, while DiffSurv’s Odd-Even
536 method experiences a notable rise, especially from risk set sizes of 32 to 128.

537 It is worth noting that CPL methods are currently computed over batches using a simple for loop, as
538 the present implementations do not support batch parallel computation. However, there’s potential
539 for further optimization. For similar batch sizes and risk set sizes, we noted very similar overall
540 convergence times with DiffSurv Bitonic variant being marginally faster than other methods. In the
541 context of full model runs, the difference between DiffSurv and CPL in terms of training times is
542 minimal; it is the model architecture that have a dominant effect on compute time.

Table 5: Isolated compute time for different methods, with various batch sizes and risk set sizes over 100 trials. Mean time and 95% confidence intervals are provided in milliseconds.

Method	Batch Size, Risk Set Size			
	512, 2	128, 8	32, 32	8, 128
DiffSurv: Odd-Even	12.37 ± 0.15	25.63 ± 0.09	107.71 ± 0.38	305.91 ± 3.66
DiffSurv: Bitonic	3.85 ± 0.03	17.29 ± 0.08	55.49 ± 0.47	92.32 ± 0.80
CPL: Breslow	853.32 ± 11.39	281.22 ± 1.20	94.08 ± 0.15	25.45 ± 0.46
CPL: Efron	1729.87 ± 5.54	494.83 ± 4.27	164.73 ± 1.40	51.69 ± 0.76
CPL: Ranked List	719.42 ± 4.99	225.69 ± 0.94	74.36 ± 0.18	18.19 ± 0.31

543 C Sorting Networks

544 There are multiple different types of sorting networks each with varying complexity. The ability to
545 implement networks with the divide-and-conquer paradigm allows for sorting networks that scale
546 more efficiently. Examples for Odd-Even and Bitonic sorting networks with $n = 8$ are shown in
547 Figure 4. The latter allows construction of networks with size complexity $\mathcal{O}(n \log^2 n)$ versus the
548 $\mathcal{O}(n^2)$ in Odd-Even networks.

549 It is worth emphasising these are not neural networks. They are called "networks" because they are
550 typically represented as diagrams that show how the items are compared and swapped as they are
551 being sorted. Differentiable sorting networks do not introduce any additional parameters that need to
552 be updated during optimization.

553 D Calibration of Predicted Permutations

554 Model calibration in survival analysis models is essential for ensuring that the predicted probabilities
555 of outcomes align closely with the true probabilities. An improperly calibrated model may lead

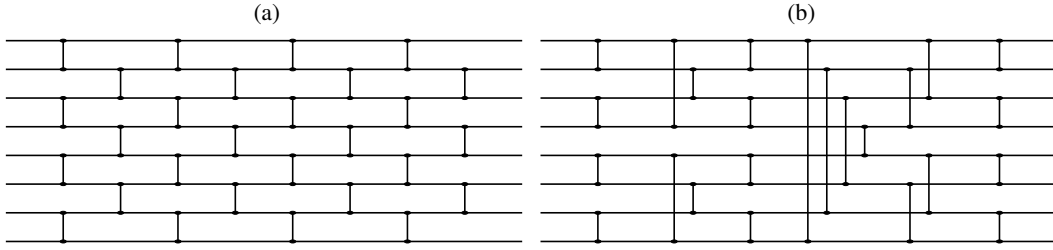


Figure 4: Example sorting networks of size 8; (a) Odd-Even, (b) Bitonic.

Table 6: Calibration of rank probabilities from predicted permutation matrices on survSVHN, keeping the number of events per batch equal. Mean (and standard deviation) Brier scores over 5 trails with different seeds. Corresponds with C-index results in Table 9.

Method	Batch Size, Risk Set Size			
	512, 2	128, 8	32, 32	8, 128
CPL: Breslow	0.081 (0.002)	0.181 (0.001)	0.056 (0.000)	0.014 (0.000)
CPL: Efron	0.081 (0.002)	0.180 (0.001)	0.055 (0.000)	0.014 (0.000)
CPL: Ranked_List	0.066 (0.001)	0.185 (0.001)	0.053 (0.000)	0.012 (0.000)
DiffSurv: Bitonic	0.059 (0.001)	0.165 (0.001)	0.043 (0.000)	0.010 (0.000)
DiffSurv: Odd-Even	0.056 (0.001)	0.158 (0.001)	0.039 (0.000)	0.009 (0.000)

556 to incorrect risk assessments and treatment decisions, potentially resulting in suboptimal patient
 557 care and even adverse clinical consequences. We focus on the calibration of predicted individual
 558 rankings, as demonstrated in Figure 5 and Table 6. Specifically, we qualitatively illustrate in Figure 5
 559 that for a model with a risk set size of 8, both discrete predicted ranks and ranking probabilities are
 560 accurately calibrated for the DiffSurv approach. To perform a quantitative comparison with baseline
 561 methods, we need to derive ranking probabilities for the CPL model. Based on the assumptions
 562 in Raykar et al. [2007], we assume that the probability of correct pairwise ordering for the CPL
 563 adheres to the logistic function. We thus compute permutation matrices using differential sorting
 564 networks, employing predicted partial log hazards as inputs and the logistic sigmoid function as the
 565 differentiable sorting operator. By subsequently calculating Brier scores for the rank probabilities in
 566 the predicted permutation matrices survSVHN, we analyze various combinations of batch size and
 567 risk set size. Our findings show that the DiffSurv models consistently exhibit the lowest Brier scores
 568 across all settings (refer to Table 6).

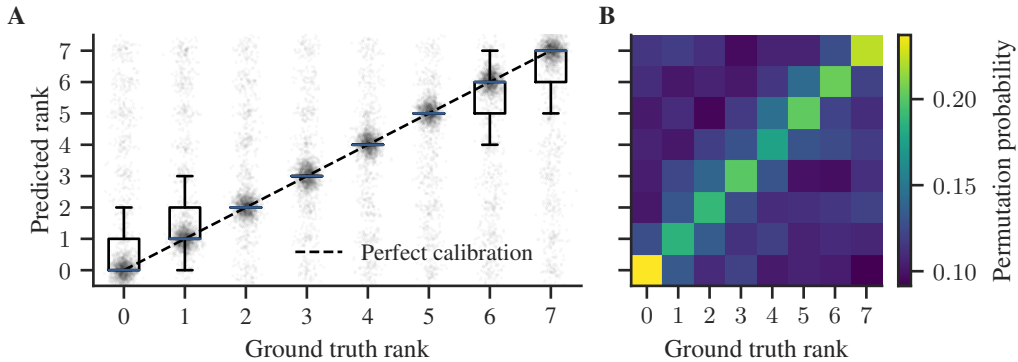


Figure 5: Calibration visualization of the DiffSurv (Bitonic) predicted ranking for a group of 8 subjects on the survSVHN dataset.

569 **E Effect of top-k loss variants on predictive performance**

570 The Top-K risk prediction variants for DiffSurv and the Cox Partial Likelihood, as introduced in
 571 Section 3.2, inherently bias the model to recognize only the top-k individuals with the highest
 572 risk. In this section, we explore the extent to which these specific variants influence the predictive
 573 performance of the models across the entire set of individuals. These results are shown in Table 7.

Table 7: C-index for real-world and semi-synthetic datasets for Top-k loss variants

	FLCHAIN	NWTCO	SUPPORT	METABRIC	MIMIC IV CXR
Top 10% prediction					
Cox Partial Likelihood	.796 (.010)	.725 (.014)	.650 (.008)	.691 (.013)	.760 (.002)
CPL-TopK (Variant I)	.794 (.011)	.715 (.011)	.602 (.003)	.644 (.016)	.751 (.002)
CPL-TopK (Variant II)	.798 (.011)	.722 (.016)	.600 (.010)	.658 (.008)	.753 (.002)
DiffSurv	.794 (.011)	.696 (.025)	.634 (.008)	.649 (.016)	.761 (.002)
DiffSurv-TopK	.783 (.011)	.689 (.022)	.596 (.005)	.639 (.023)	.754 (.001)

574 **F Additional Results**

575 In Table 8 and Table 9, additional results for the MIMIC IV CXR and survSVHN imaging datasets
 576 are provided. Here, we maintain a constant total number of samples in each batch, which means that
 577 an increase in risk set size is compensated by a decrease in batch size. These results offer further
 578 understanding of the balance required between these two variables. We observed, while larger risk
 579 set sizes generally improve performance for both DiffSurv and CPH, the benefits tend to taper off as
 580 training can become more unstable and noisy with smaller batch sizes.

Table 8: Additional Results for MIMIC IV CXR. Mean and standard deviation of the C-index for different methods and batch risk set sizes. Bold indicates a significantly higher result with t-test and $p \leq 0.01$.[†] Most significant across all Batch Size, Risk Set Sizes.

Method	Batch Size, Risk Set Size			
	64, 2	4, 32	16, 8	1, 128
DiffSurv: Bitonic	0.761 (0.001)	0.761 (0.000)	0.763[†] (0.001)	0.761 (0.002)
DiffSurv: Odd-Even	0.761 (0.002)	0.756 (0.002)	0.761 (0.001)	0.749 (0.001)
CPL: Ranked List	0.760 (0.002)	0.755 (0.002)	0.758 (0.003)	0.755 (0.002)

Table 9: Additional results for survSVNH keeping the number of events per batch equal. Mean (and standard deviation) over 5 trials with different seeds. Metric is C-index. Bold indicates a significantly higher result with t-test and $p \leq 0.01$.

Method	Batch Size, Risk Set Size			
	512, 2	128, 8	32, 32	8, 128
DiffSurv: Odd-Even	0.934 (0.001)	0.940 (0.001)	0.941 (0.001)	0.933 (0.002)
DiffSurv: Bitonic	0.931 (0.001)	0.942 (0.001)	0.940 (0.00166)	0.928 (0.001)
CPL: Breslow	0.905 (0.001)	0.897 (0.001)	0.910 (0.002)	0.919 (0.001)
CPL Efron	0.904 (0.002)	0.898 (0.002)	0.909 (0.003)	0.918 (0.003)
CPL: Ranked List	0.921 (0.001)	0.922 (0.003)	0.921 (0.001)	0.917 (0.003)