Fairness in Survival Analysis with Distributionally Robust Optimization

Shu Hu*[1] and George H. Chen*†[2]

[1]*Department of Computer Information and Graphics Technology, Indiana University-Purdue University Indianapolis*
[2]*Heinz College of Information Systems and Public Policy, Carnegie Mellon University*

Abstract

We propose a general approach for encouraging fairness in survival analysis models that is based on minimizing a worst-case error across *all* subpopulations that are "large enough" (occurring with at least a user-specified probability threshold). This approach can be used to convert a wide variety of existing survival analysis models into ones that simultaneously encourage fairness, *without* requiring the user to specify which attributes or features to treat as sensitive in the training loss function. From a technical standpoint, our approach applies recent methodological developments of *distributionally robust optimization* (DRO) to survival analysis. The complication is that existing DRO theory uses a training loss function that decomposes across contributions of individual data points, i.e., any term that shows up in the loss function depends only on a single training point. This decomposition does not hold for commonly used survival loss functions, including for the standard Cox proportional hazards model, its deep neural network variants, and many other recently developed survival analysis models that use loss functions involving ranking or similarity score calculations. We address this technical hurdle using a sample splitting strategy. We demonstrate our DRO approach by using it to create fair versions of a diverse set of existing survival analysis models including the classical Cox model (and its deep neural network variant DeepSurv), the discrete-time model DeepHit, and the neural ODE model SODEN. For all of these models, we show that our DRO variants often score better on recently established fairness metrics (without incurring a significant drop in accuracy) compared to existing survival analysis fairness regularization techniques, including ones which directly use sensitive demographic information in their training loss functions.

Our code is available at: `https://github.com/discovershu/DRO_COX`.

1 Introduction

Survival analysis aims to model time durations before a critical event happens. Examples of such critical events include a patient dying, a convicted criminal reoffending, or a customer cancelling a subscription service. Predicting such time durations accurately could help plan patient treatments, make bail decisions, or target subscription pricing promotions. If a survival analysis model is to be used in high-stakes decision making, fairness could be an important design criterion. For example, in the case of making bail decisions with the help of predicted time durations until a criminal reoffends, we may want a survival analysis model that produces these predictions to be similarly accurate across different races.

One of the major recent advances in encouraging fairness for machine learning models is to minimize a worst-case error over *all* subpopulations that are "large enough" (e.g., Hashimoto et al.

*Equal contribution
†Corresponding author: `georgechen@cmu.edu`

2018, Duchi and Namkoong 2021, Li et al. 2021, Duchi et al. 2022, Hu et al. 2022a). In particular, a modeler specifies a minimum probability threshold α. The goal then is to ensure that all subpopulations that occur with probability at least α have low average error, whereas we make no promises for subpopulations that occur with probability less than α. The modeler need not provide a list of subpopulations to account for. This problem can be tractably solved in practice and is called *distributionally robust optimization* (DRO).

We emphasize that curating a list of all subpopulations to account for can be challenging for various reasons. For example, one major challenge is *intersectionality*: subpopulations that a machine learning model yields the worst accuracy scores for can be defined by complex intersections of sensitive attributes (e.g., age, race, gender) [Buolamwini and Gebru, 2018]. Some of these attributes might require discretization (e.g., dividing age into bins), for which choosing the "best" discretization strategy might not be straightforward. Moreover, if there is a large number of features and we suspect that the sensitive attributes (encoded by specific features) could possibly be correlated with other features (not flagged as sensitive), there is a question of whether these other features should also be accounted for in a listing of what the sensitive attributes are. DRO provides a theoretically sound alternative to having to specify which attributes to treat as sensitive in a training loss function.

Our main contribution in this paper is to show how to apply DRO to survival analysis. Specifically, we propose a general strategy for converting a wide variety of survival analysis models into ones that simultaneously encourage fairness. Our strategy supports all survival analysis models we are aware of that minimize a loss function (details on the general form of survival analysis models that our approach supports are in Section 3).

The key technical challenge is that existing DRO theory assumes that the overall training loss is the sum of individual loss terms, where each such term only depends on a single data point. This assumption fails to hold for commonly used survival analysis loss functions—including that of the standard Cox proportional hazards model [Cox, 1972]—that involve pairwise comparisons from ranking or similarity score evaluations (e.g., Steck et al. 2007, Lee et al. 2018, Chen 2020, Wu et al. 2021). In particular, there are loss terms that arise that incorporate information from multiple data points at once. We propose a sample splitting approach to address this technical challenge. We point out that there are also parametric survival analysis models with loss functions that directly adhere to existing DRO theory (e.g., parametric accelerated failure time models [Klein and Moeschberger, 2003, Chapter 12] or, as a more exotic example, the recently proposed neural ordinary differential equation (ODE) model called SODEN [Tang et al., 2022b]); such models can trivially be modified to use DRO without the sample splitting approach that we propose.

We specifically show how to derive DRO variants of the standard Cox model [Cox, 1972] (and its deep neural network variant DeepSurv [Faraggi and Simon, 1995, Katzman et al., 2018]), the discrete-time DeepHit model [Lee et al., 2018], and the neural ODE model called SODEN [Tang et al., 2022b]. Again, we emphasize that our strategy for converting an existing survival analysis model to its DRO variant is fairly general and is not limited to only the few models that we showcase as illustrative examples.

On three standard datasets that have been previously used for research on fair survival analysis, we show that our DRO modification often outperforms various baseline fairness regularization techniques in terms of existing fairness metrics that focus on user-specified sensitive attributes. Most of these baselines require the user to specify which attributes to treat as sensitive attributes within the added regularization term. As with other fairness methods recently developed for survival analysis (e.g., Keya et al. [2021], Rahman and Purushotham [2022]), our approach also results in a drop in accuracy (compared to using a loss that does not encourage fairness). Note that our paper does not aim to find which survival model is the most accurate or the most fair. In fact, per survival model, there is in general a tradeoff between accuracy and fairness that can be tuned by the modeler. We show how to visualize this tradeoff using a plot inspired by an ROC curve.

Related work on fair survival analysis Despite many recent advances in survival analysis methodology (see, for instance, the survey by Wang et al. [2019]), very few of these advances study fairness [Keya et al., 2021, Zhang and Weiss, 2022, Sonabend et al., 2022, Rahman and Purushotham, 2022].

We provide an overview of these existing papers, and we discuss how they differ from our work.

Keya et al. [2021] adapted existing fairness definitions to the survival analysis setting and showed how to encourage different notions of fairness by adding fairness regularization terms. Specifically, Keya et al. [2021] came up with individual [Dwork et al., 2012], group [Dwork et al., 2012], and intersectional [Foulds et al., 2020] fairness definitions specialized to Cox models. Keya et al. define individual fairness in terms of model predictions being similar for similar individuals, and group fairness in terms of different user-specified groups having similar average predicted outcomes. Intersectional fairness further considers subgroups defined by intersections of protected groups (e.g., individuals of a specific race and simultaneously a specific gender). However, a major limitation of the notions of fairness defined by Keya et al. is that they focus on predicted model outputs and do not actually use any of the ground truth label information. For example, if one uses age as a sensitive attribute and suppose we discretize age into two groups, then the notion of group fairness by Keya et al. would ask for the predicted outcomes of the two age groups to be similar, which for healthcare problems often does not make sense (since age is often highly predictive of different health outcomes). Instead, in such a scenario, a more desirable notion of fairness is that the model's *accuracy* for the different age groups be similar.

To account for model accuracy, Zhang and Weiss [2022] introduced a fairness metric called *concordance imparity* that computes a quantity similar to the standard survival analysis accuracy metric of *concordance index* [Harrell et al., 1982] for different groups and then looks at the worst-case difference between any two groups' accuracy scores. Meanwhile, Rahman and Purushotham [2022] directly modified the fairness definitions of Keya et al. [2021] to account for ground truth label information, and also generalized these definitions to survival models beyond Cox models.

Separately, Sonabend et al. [2022] empirically explored how well existing survival analysis accuracy and calibration metrics measure bias by synthetically modifying datasets (e.g., undersampling disadvantaged groups). However, they do not propose any new fairness metric or survival model that encourages fairness.

The papers mentioned above that propose new methods for learning fair survival models all either require user-specified demographic information to treat as sensitive (possibly as a list of subpopulations or groups to account for) or are simply adding a regularization term that encourages smoothness in the model outputs (the individual fairness regularization by Keya et al. [2021] and Rahman and Purushotham [2022] are directly related to encouraging Lipschitz continuity; for details, see Appendix B). In contrast, our proposed DRO approach does not require the user to indicate which attributes to treat as sensitive in the training loss function, and is not simply encouraging the model output to be Lipschitz continuous.

Bibliographical note This paper significantly extends our previous conference paper [Hu and Chen, 2022] in methodological development and in experiments. For methodological development, whereas our conference paper only considered Cox models, we show in this journal paper version how to convert a much wider class of survival analysis models into their DRO variants that encourage fairness. In fact, this wider class of models consists of all survival models we are aware of that are learned by minimizing an overall loss function. For experiments, we demonstrate our conversion strategy on not only Cox models but also on DeepHit and SODEN models. Our experiments are overall more extensive, and the SEER dataset we now use is much larger ($\sim$28k data points in this version vs $\sim$4k in the conference paper). Lastly, we also add a new visualization for seeing the tradeoff between accuracy and fairness across multiple models within a single plot.

Outline The rest of the paper is organized as follows. We provide background on survival analysis, existing research on fairness in survival analysis, and DRO in Section 2. We then present our strategy for converting a wide family of existing survival analysis models into their corresponding DRO variants that encourage fairness in Section 3. We conduct experiments to compare DRO variants of Cox, DeepHit, and SODEN models to their original non-DRO variants as well as to variants of these models that encourage fairness using non-DRO baseline regularization strategies. We conclude the paper in Section 5.

2 Background

We begin by reviewing the basic survival analysis problem setup in Section 2.1 and then provide three examples of survival analysis models (Cox, DeepHit, and SODEN) in Section 2.2. We then review DRO in Section 2.3. Throughout the paper, we frequently use the notation $[\ell] \triangleq \{1, 2, \ldots, \ell\}$ for any positive integer ℓ.

2.1 Survival Analysis Setup

Survival analysis aims to model the amount of time that will elapse before a critical event of interest happens. We assume that we have training data $\{(X_i, Y_i, \delta_i)\}_{i=1}^n$, where training data point $i \in [n]$ has raw input $X_i \in \mathcal{X}$, observed duration $Y_i \geq 0$, and event indicator $\delta_i \in \{0, 1\}$. If $\delta_i = 1$ (i.e., the critical event of interest happened for the i-th data point), then Y_i is the time until the event happens. Otherwise, if $\delta_i = 0$, then Y_i is the time until censoring for the i-th point, i.e., the true time until event is unknown but we know that it is at least Y_i. The raw input space $\mathcal{X}$ could be any input space supported by standard neural network software (e.g., tabular data, images, time series).

Each training data point (X_i, Y_i, δ_i) is assumed to be generated as follows:

1. Sample raw input X_i from a raw input distribution $\mathbb{P}_X$.
2. Sample nonnegative time duration T_i (this is the true time until the critical event happens) from a conditional distribution $\mathbb{P}_{T|X=X_i}$.
3. Sample nonnegative time duration C_i (this is the true time until the data point is censored) from a conditional distribution $\mathbb{P}_{C|X=X_i}$.
4. If $T_i \leq C_i$ (the critical event happens before censoring), then set $Y_i = T_i$ and $\delta_i = 1$. Otherwise, set $Y_i = C_i$ and $\delta_i = 0$.

Distributions $\mathbb{P}_X$, $\mathbb{P}_{T|X}$, and $\mathbb{P}_{C|X}$ are shared across data points and are unknown. We assume that the random variables T_i and C_i are independent given X_i. We denote the CDF of distribution $\mathbb{P}_{T|X=x}$ as $F(\cdot|x)$.

Prediction A standard prediction task is to estimate the probability that a data point with raw input $x \in \mathcal{X}$ survives beyond time t. Formally, this is defined as the *conditional survival function*

$$S(t|x) \triangleq \mathbb{P}(T > t|X = x) = 1 - F(t|x) \qquad \text{for } t \geq 0. \tag{2.1}$$

Importantly, for raw input x, we are predicting an entire probability distribution (since $S(\cdot|x)$ encodes the same information as the CDF $F(\cdot|x)$).

Some survival analysis models, such as the Cox proportional hazards model [Cox, 1972], estimate a transformed version of $S(\cdot|x)$ called the *hazard function*, given by

$$h(t|x) \triangleq -\frac{\partial}{\partial t} \log S(t|x) \qquad \text{for } t \geq 0. \tag{2.2}$$

From negating both sides of this equation, integrating over time, and exponentiating, we get $S(t|x) = \exp(-\int_0^t h(u|x)\mathrm{d}u)$. Thus, if we have an estimate of $h(\cdot|x)$, then we can readily estimate the conditional survival function $S(\cdot|x)$.

2.2 Examples of Survival Analysis Models

We now review three examples of survival analysis models (Cox, DeepHit, and SODEN) that can be modified to encourage fairness using DRO. In reviewing these models, we focus on aspects most relevant to our exposition later for how to convert these models into their DRO variants. For all three examples, we denote the neural network to be learned as $f(\cdot; \theta)$, where θ denotes the parameters of the neural network. The domain and range of f depends on the specific survival model we look at. Meanwhile, the architecture of f is up to the modeler to specify, where standard strategies could be used (e.g., if the raw inputs are tabular data, then a multilayer perceptron could be used; if the raw inputs are images, a convolutional neural network could be used; etc).

2.2.1 Classical and Deep Cox Models

The classical Cox model assumes that the hazard function has the factorization

$$h(t|x) = h_0(t)\exp(f(x;\theta)), \tag{2.3}$$

where h_0 is called the baseline hazard function (h_0 maps a nonnegative time $t \geq 0$ to a nonnegative number), and neural network $f(\cdot;\theta)$ maps a raw input from $\mathcal{X}$ to a single real number (i.e., $f(\cdot;\theta)$ has domain $\mathcal{X}$ and range $\mathbb{R}$). In particular, $f(x;\theta)$ models the so-called *log partial hazard function* and could be thought of as assigning a real-valued "risk score" to raw input $x \in \mathcal{X}$: when $f(x;\theta)$ is higher, then x has a higher risk of the critical event happening, so that the survival time of x will tend to be lower.

The original Cox model [Cox, 1972] defines f to be a dot product: $f(x;\theta) = \theta^T x$, where θ and x are in the same Euclidean vector space. More recently, researchers replaced f with a neural network [Faraggi and Simon, 1995, Katzman et al., 2018], resulting in a method called DeepSurv (which could be viewed as a generalization of the original Cox model in that the classical definition $f(x;\theta) = \theta^T x$ is a simple neural network consisting of a linear layer with no bias and no nonlinear activation). In either case, the standard approach for learning a Cox model is to first learn the neural network parameters θ by minimizing the negative log partial likelihood:

$$L^{\text{Cox}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} L_i^{\text{Cox}}(\theta), \tag{2.4}$$

where the i-th data point's loss is

$$L_i^{\text{Cox}}(\theta) \triangleq -\delta_i\left[f(X_i;\theta) - \log \sum_{j\in[n] \text{ s.t. } Y_j \geq Y_i} \exp(f(X_j;\theta))\right]. \tag{2.5}$$

If the i-th data point is censored (i.e., $\delta_i = 0$), then $L_i^{\text{Cox}}(\theta) = 0$. Thus, the overall loss $L^{\text{Cox}}(\theta)$ could be viewed as weighting the *uncensored* training points equally. After learning θ, we then estimate h_0; as this step is not essential to our exposition, we explain it in Appendix A.1, along with details on constructing the final estimate of $S(\cdot|x)$.

We remark that the factorization in equation (2.3) is referred to as the *proportional hazards assumption*: regardless of what the input x is, the hazard function $h(\cdot|x)$ is always proportional to the baseline hazard function h_0. A consequence of this assumption is that the resulting conditional survival function $S(\cdot|x)$ is heavily constrained in terms of its shape. In particular, regardless of what x is, $S(t|x)$ must be a power of the function $S_0(t) \triangleq \exp(-\int_0^t h_0(u)\mathrm{d}u)$ (for details, see Appendix A.2). The next two survival analysis models that we describe do not have this assumption and can more flexibly estimate $S(\cdot|x)$.

2.2.2 DeepHit

A wide class of survival analysis models directly estimate (some transformed version of) the conditional survival function $S(\cdot|x)$ along a discretized time grid, without requiring the proportional hazards assumption. The time grid itself is up to the modeler to choose and can depend on the observed time Y_i and event indicator δ_i variables in the training data. For example, we could use a uniformly-spaced time grid between the minimum and maximum observed times (for some user-specified number of discretized time steps), or we could have the time grid consist of all unique times in the training data in which the critical event happened (in fact, this how the classical Kaplan-Meier estimator [Kaplan and Meier, 1958] discretizes time). Some other time grids are discussed by Kvamme and Borgan [2021].

An example of a model that uses a discretized time grid is DeepHit [Lee et al., 2018]. Note that DeepHit supports the so-called *competing risks* setting where there are multiple critical events of interest. For simplicity, we review DeepHit where we only present the case where there is a

single critical event of interest, which reduces the problem setup to the same one we specified in Section 2.1.

Let $t_1 < t_2 < \cdots < t_m$ denote the m discretized time points based on some user-specified grid. We assume that these are the only time points in which the critical event or censoring could happen (if a critical event or censoring happens at some other time point, we quantize it to one of these m time points). Then DeepHit parameterizes the following conditional probability mass function using a neural network:

$$\mathbb{P}(T = t_j | X = x) = f_j(x; \theta) \qquad \text{for } j \in [m], \tag{2.6}$$

where neural network $f(x;\theta) = \big(f_1(x;\theta), f_2(x;\theta), \ldots, f_m(x;\theta)\big) \in [0,1]^m$ has parameters θ and maps a raw input $x \in \mathcal{X}$ to a probability distribution over the m time steps. In other words, the domain of $f(\cdot;\theta)$ is $\mathcal{X}$ and the range of $f(\cdot;\theta)$ is the probability simplex $\Delta^m \triangleq \big\{z \in \mathbb{R}^m : z_j \geq 0 \text{ for all } j \in [m] \text{ and } \sum_{j=1}^m z_j = 1\big\}$. For example, when working with tabular data, f could be a multilayer perceptron, where the last linear layer outputs m numbers and has softmax activation.

Because of the parameterization in equation (2.6), we can write the conditional survival function $S(t|x)$ at any discrete time point t_j in terms of the neural network $f(\cdot;\theta)$:

$$S_j(x;\theta) \triangleq S(t_j|x) = \mathbb{P}(T > t_j | X = x) = \sum_{\ell=j+1}^{m} f_\ell(x;\theta) \quad \text{for } j \in [m].$$

To learn θ, DeepHit uses the sum of two loss terms, corresponding to a negative log likelihood term and, separately, a ranking loss term. In what follows, we use the notation $\kappa(Y_i) \in [m]$ to denote the time step index corresponding to the i-th training point's observed time Y_i (i.e., Y_i gets quantized to integer time step $\kappa(Y_i)$). Then the overall DeepHit loss is

$$\begin{aligned} L^{\text{DeepHit}}(\theta) \triangleq \beta \cdot \overbrace{\frac{1}{n}\sum_{i=1}^{n}\big[-\delta_i \log(f_{\kappa(Y_i)}(X_i;\theta)) - (1-\delta_i)\log(S_{\kappa(Y_i)}(X_i;\theta))\big]}^{\text{negative log likelihood loss term}} \\ + (1-\beta) \cdot \underbrace{\frac{1}{n^2}\sum_{i=1}^{n}\delta_i \sum_{j \in [n] \text{ s.t. } \kappa(Y_j) > \kappa(Y_i)} \exp\left(\frac{S_{\kappa(Y_i)}(X_i;\theta) - S_{\kappa(Y_i)}(X_j;\theta)}{\sigma}\right)}_{\text{ranking loss term}}, \end{aligned} \tag{2.7}$$

where $\beta \in [0,1]$ and $\sigma > 0$ are hyperparameters. Note that this formulation of the overall loss follows the implementation of DeepHit by Kvamme et al. [2019] in the now-standard `pycox` software package and is slightly different from the original formulation by Lee et al. [2018] (the only difference is in the weights used to combine the two main loss terms).

For how we convert DeepHit into its DRO variant later, it will be helpful to rewrite the DeepHit loss in terms of individual losses:

$$L^{\text{DeepHit}}(\theta) \triangleq \frac{1}{n}\sum_{i=1}^{n} L_i^{\text{DeepHit}}(\theta), \tag{2.8}$$

where the i-th individual loss is

$$\begin{aligned} L_i^{\text{DeepHit}}(\theta) = \beta \cdot \big[-\delta_i \log(f_{\kappa(Y_i)}(X_i;\theta)) - (1-\delta_i)\log(S_{\kappa(Y_i)}(X_i;\theta))\big] \\ + (1-\beta) \cdot \frac{1}{n} \cdot \delta_i \sum_{j \in [n] \text{ s.t. } \kappa(Y_j) > \kappa(Y_i)} \exp\left(\frac{S_{\kappa(Y_i)}(X_i;\theta) - S_{\kappa(Y_i)}(X_j;\theta)}{\sigma}\right). \end{aligned} \tag{2.9}$$

2.2.3 SODEN

Recently, a number of researchers have considered a differential-equation approach to setting up a survival analysis model that can avoid the proportional hazards assumption while also not requiring the modeler to explicitly specify a discrete time grid [Groha et al., 2020, Moon et al., 2022, Tang

et al., 2022a,b]. We review one such model called SODEN (Survival model through Ordinary Differential Equation Networks), proposed by Tang et al. [2022b]. Note that we review a special case that is easier to describe and that corresponds to our survival analysis problem setup in Section 2.1, where survival times are all nonnegative.

In what follows, we denote $H(t|x) \triangleq -\log S(t|x)$. From how we defined the hazard function $h(t|x)$ in equation (2.2), we have $h(t|x) = \frac{\partial}{\partial t} H(t|x)$, so $H(t|x) = \int_0^t h(u|x)\mathrm{d}u$; this integral expression reveals why $H(t|x)$ is commonly called the *cumulative hazard function*.

SODEN uses a neural network $f(\cdot;\theta)$ to parameterize the hazard function as the solution to an ordinary differential equation (ODE):

$$\begin{cases} \frac{\partial}{\partial t} H(t|x) = h(t|x) = f\big((t, H(t|x), x); \theta\big) & \text{for } t > 0, \\ H(0|x) = 0 \qquad \text{(initial condition at time 0)}, \end{cases} \tag{2.10}$$

where the neural network $f(\cdot;\theta)$ has domain $[0,\infty) \times [0,\infty) \times \mathcal{X}$ and range $\mathbb{R}$. Specifically $f(\cdot;\theta)$ takes as input time $t \geq 0$, a cumulative hazard value $H(t|x)$ (which is nonnegative), and a raw input $x \in \mathcal{X}$, and $f(\cdot;\theta)$ outputs a single real number that is $h(t|x)$. For example, $f(\cdot;\theta)$ could concatenate all its inputs to form a single vector of numbers that is then treated as the input to a multilayer perceptron, where the final linear layer outputs a single number and has softplus activation (to ensure that the output is always positive). The initial condition follows from the fact that $H(0|x) = \int_0^0 h(u|x)\mathrm{d}u = 0$.

Learning neural networks in terms of ODEs (as in equation (2.10)) is possible thanks to the landmark paper by Chen et al. [2018]. Importantly, using any user-specified ODE solver, given any raw input $x \in \mathcal{X}$ and neural network parameters θ, we can numerically solve the ODE in equation (2.10) (going from time 0 to any user-specified time $t > 0$) to obtain an estimate for $H(t|x)$. In particular, a major result of Chen et al. [2018] is that the loss function we use to train the neural network can contain terms involving $h(t|x) = f((t, H(t|x), x); \theta)$ and $H(t|x)$; backpropagation is possible with the help of any ODE solver.

To train the SODEN model, Tang et al. [2022b] use the overall loss function

$$L^{\text{SODEN}}(\theta) \triangleq \frac{1}{n} \sum_{i=1}^{n} L_i^{\text{SODEN}}(\theta), \tag{2.11}$$

where the i-th individual loss is

$$L_i^{\text{SODEN}}(\theta) = -\delta_i \log f\big((Y_i, H(Y_i|X_i), X_i); \theta\big) + H(Y_i|X_i). \tag{2.12}$$

Note that the overall loss (2.11) is just a negative log likelihood expression, so that minimizing this loss corresponds to solving a maximum likelihood problem.

2.3 Distributionally Robust Optimization (DRO)

DRO uses a worst-case average error over "large enough" subpopulations. Note that there are now a number of different versions of DRO (e.g., Hashimoto et al. 2018, Sagawa et al. 2020, Duchi and Namkoong 2021, Duchi et al. 2022). We specifically use the one by Hashimoto et al. [2018]. Even though existing literature on DRO does not consider survival analysis to the best of our knowledge, we intentionally review DRO here using survival analysis notation that we have introduced in Section 2.1. In fact, existing DRO theory actually works with many existing survival analysis loss functions already, without modification. In particular, survival analysis models for which each data point's individual loss does not depend on any other data points could trivially use existing DRO machinery. Examples of such survival analysis models include DeepHit when $\beta = 1$ (see equation (2.9)), SODEN (see equation (2.12)), as well as exponential, Weibull, log-logistic, log-normal, and generalized Gamma accelerated failure time models [Klein and Moeschberger, 2003, Chapter 12].

Problem setup Let $\mathbb{P}$ denote the joint distribution over each data point (X_i, Y_i, δ_i). This joint distribution corresponds to the generative procedure described in Section 2.1. We assume that there are K groups that comprise $\mathbb{P}$. In particular, $\mathbb{P}$ is a mixture of K distributions $\mathbb{P} \triangleq \sum_{k=1}^{K} \pi_k \mathbb{P}_k$, where the k-th group occurs with probability $\pi_k \in (0,1)$ and has associated distribution $\mathbb{P}_k$. Moreover, $\sum_{k=1}^{K} \pi_k = 1$. We assume that we do not know $\{(\pi_k, \mathbb{P}_k)\}_{k=1}^{K}$, nor do we know K. This setting, for instance, handles the case where we do not exhaustively know all subpopulations to consider. The smallest minority group corresponds to whichever group has the smallest π_k value. A simple special case would be when $K = 2$, where data are drawn from either a minority group or a majority group.

We would like to minimize the risk

$$\mathcal{R}_{\max}(\theta) \triangleq \max_{k=1,\ldots,K} \mathbb{E}_{(X,Y,\delta)\sim\mathbb{P}_k}[L_{\text{indiv}}(\theta; X, Y, \delta)],$$

where L_{indiv} is a loss function that depends only on the parameters θ (for a survival analysis model that we aim to learn) and on a single data point (X, Y, δ). However, minimizing $\mathcal{R}_{\max}(\theta)$ is not possible as we do not know any of the latent groups. Nevertheless, it turns out that there is an optimization problem that we can tractably solve that minimizes an empirical version of an upper bound on $\mathcal{R}_{\max}(\theta)$. We explain what the upper bound is in Section 2.3.1 and how to empirically minimize the upper bound in Section 2.3.2.

2.3.1 Upper Bound on the Risk $\mathcal{R}_{\max}(\theta)$ Using DRO

For a set of distributions $\mathcal{B}_r(\mathbb{P})$ to be defined shortly, we consider minimizing the following alternative risk instead:

$$\mathcal{R}_{\text{DRO}}(\theta; r) \triangleq \sup_{\mathbb{Q}\in\mathcal{B}_r(\mathbb{P})} \mathbb{E}_{(X,Y,\delta)\sim\mathbb{Q}}[L_{\text{indiv}}(\theta; X, Y, \delta)]. \tag{2.13}$$

This is the worst-case expected loss when we sample from any distribution in $\mathcal{B}_r(\mathbb{P})$.

The definition for $\mathcal{B}_r(\mathbb{P})$ is somewhat technical; we first give its precise definition and then state how to choose r so that $\mathcal{R}_{\text{DRO}}(\theta; r)$ is an upper bound on $\mathcal{R}_{\max}(\theta)$. Importantly, we will be able to efficiently minimize an empirical version of $\mathcal{R}_{\text{DRO}}(\theta; r)$.

Definition 1. *The set $\mathcal{B}_r(\mathbb{P})$ consists of all distributions $\mathbb{Q}$ that have the same (or smaller) support as $\mathbb{P}$ and have χ^2-divergence at most r from distribution $\mathbb{P}$. Formally,*

$$\mathcal{B}_r(\mathbb{P}) \triangleq \{\textit{distribution } \mathbb{Q} \textit{ such that } \mathbb{Q} \ll \mathbb{P} \textit{ and } D_{\chi^2}(\mathbb{Q}\|\mathbb{P}) \leq r\},$$

where, using standard measure theory notation, "$\mathbb{Q} \ll \mathbb{P}$" means that $\mathbb{Q}$ is absolutely continuous with respect to $\mathbb{P}$, and $D_{\chi^2}(\mathbb{Q}\|\mathbb{P}) \triangleq \int (\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}\mathbb{P}} - 1)^2 \mathrm{d}\mathbb{P}$.

Working with $\mathcal{B}_r(\mathbb{P})$ turns out to be straightforward so long as we have a lower bound on the smallest group's probability (i.e., a lower bound on $\min_{k=1,\ldots,K} \pi_k$).

Proposition 1. *(Directly follows from Proposition 2 of Hashimoto et al. [2018]) Suppose that we have a lower bound $\alpha > 0$ on the K latent groups' probabilities of occurring (i.e., $\alpha \leq \min_{k=1,\ldots,K} \pi_k$). Then $\mathcal{R}_{DRO}(\theta; r_{\max}) \geq \mathcal{R}_{\max}(\theta)$, where $r_{\max} \triangleq (\frac{1}{\alpha} - 1)^2$.*

In other words, if we have a guess for $\alpha \in (0, \min_{k=1,\ldots,K} \pi_k]$, then it suffices to choose r for $\mathcal{B}_r(\mathbb{P})$ to be $r_{\max} = (\frac{1}{\alpha} - 1)^2$. Furthermore, the risk $\mathcal{R}_{\text{DRO}}(\theta; r_{\max})$ is an upper bound on $\mathcal{R}_{\max}(\theta)$. In practice, $\alpha \in (0,1)$ is a user-specified hyperparameter since we do not know $\pi_1, \ldots, \pi_K$ nor K. Choosing α to be smaller means that we want to ensure that groups with smaller probabilities of occurring also have low expected loss. For example, setting $\alpha = 0.1$ means that the "rarest" group that we want to ensure low expected loss for occurs with probability at least 0.1.

Algorithm 1: DRO

Input: A training dataset $\{(X_i, Y_i, \delta_i)\}_{i=1}^n$, minimum subpopulation probability hyperparameter α, learning rate ξ, max_iterations

Output: Survival model parameters $\widehat{\theta}$

Obtain initial survival model parameters $\widehat{\theta}_0$ (e.g., using default PyTorch parameter initialization).

for $\ell = 0$ *to* max_iterations **do**

  for $i = 1$ *to* n **do**

    Set $u_i \leftarrow L_{\text{indiv}}(\widehat{\theta}_\ell; X_i, Y_i, \delta_i)$.

  end

  Set $\widehat{\eta} \leftarrow \arg\min_{\eta \in \mathbb{R}} \left\{ \left(\sqrt{2(\frac{1}{\alpha} - 1)^2 + 1} \right) \sqrt{\frac{1}{n} \sum_{i=1}^n [u_i - \eta]_+^2} + \eta \right\}$, where this minimization is solved using binary search. (This step directly corresponds to minimizing $L_{\text{DRO}}(\widehat{\theta}_\ell, \eta)$ as given in equation (2.15).)

  Set $\widehat{\theta}_{\ell+1} \leftarrow \widehat{\theta}_\ell - \xi \cdot \nabla_\theta L_{\text{DRO}}(\widehat{\theta}_\ell, \widehat{\eta})$.

end

return $\widehat{\theta} \leftarrow \widehat{\theta}_{\text{max\_iterations}+1}$

2.3.2 Empirical DRO Risk

The next issue is how to minimize the risk $\mathcal{R}_{\text{DRO}}(\theta; r_{\max})$, which at a first glance might appear daunting due to the supremum over all distributions in $\mathcal{B}_{r_{\max}}(\mathbb{P})$. However, a fundamental theoretical result from DRO literature is that $\mathcal{R}_{\text{DRO}}(\theta; r_{\max})$ can be written in a form that is amenable to computation.

Proposition 2. *(Lemma 1 in Duchi and Namkoong [2021]) Suppose $\widehat{\ell}(\theta; X, Y, \delta)$ is upper semi-continuous with respect to θ. Let $[\cdot]_+$ denote the ReLU function (i.e., $[a]_+ \triangleq \max\{a, 0\}$ for any $a \in \mathbb{R}$), and $C_\alpha \triangleq \sqrt{2(\frac{1}{\alpha} - 1)^2 + 1}$. Then*

$$\mathcal{R}_{DRO}(\theta; r_{\max}) = \inf_{\eta \in \mathbb{R}} \left\{ C_\alpha \sqrt{\mathbb{E}_{(X,Y,\delta) \sim \mathbb{P}} \left[[L_{indiv}(\theta; X, Y, \delta) - \eta]_+^2 \right]} + \eta \right\}. \tag{2.14}$$

The right-hand side of equation (2.14) could be interpreted as follows. Suppose that we have achieved the optimal value η^*. Then the loss from a data point will be ignored if it is less than η^* (due to the ReLU function). Thus, only the data points with losses above η^* are considered for learning the survival model.

Note that as we vary the model parameters θ, the different data points' losses change. Thus, as a function of θ, the DRO risk $\mathcal{R}_{\text{DRO}}(\theta; r_{\max})$ dynamically adjusts which data points to focus on, always prioritizing the points with the highest loss values (again, we only consider the points with a loss greater than the optimal value of η).

We can readily minimize an empirical version of $\mathcal{R}_{\text{DRO}}(\theta; r_{\max})$. Specifically, we replace the expectation on the right-hand side of equation (2.14) with an empirical average to arrive at the following optimization problem:

$$\min_{\theta \in \Theta, \eta \in \mathbb{R}} \Bigg(\underbrace{C_\alpha \sqrt{\frac{1}{n} \sum_{i=1}^n [L_{\text{indiv}}(\theta; X_i, Y_i, \delta_i) - \eta]_+^2} + \eta}_{\triangleq L_{\text{DRO}}(\theta, \eta)} \Bigg), \tag{2.15}$$

where Θ denotes the feasible set of the model parameters.

Numerical optimization The optimization problem in equation (2.15) can be solved with an iterative gradient descent approach [Hu et al., 2020, 2021, 2022b]. Specifically, we first initialize the model parameters θ. Then, following Hashimoto et al. [2018], we alternate between two steps:

- We fix θ and update η by finding the value of η that minimizes $L_{\text{DRO}}(\theta, \eta)$. To do this, we use binary search to find the global optimum of η since $L_{\text{DRO}}(\theta, \eta)$ is a convex function with respect to η.

- We fix η and update θ by minimizing $L_{\text{DRO}}(\theta, \eta)$ (e.g., using gradient descent).

We stop iterating after user-specified stopping criteria are reached (e.g., maximum number of iterations reached, early stopping due to no improvement in a validation metric after a pre-specified number of epochs). The pseudocode can be found in Algorithm 1.

3 Converting Existing Survival Analysis Models into DRO Variants

Throughout this section, we assume that the training points $\{(X_i, Y_i, \delta_i)\}_{i=1}^n$ are generated by the procedure stated in Section 2.1. We describe the general class of survival models that we can convert into DRO variants in Section 3.1. For some models (such as SODEN), the existing DRO approach stated in Section 2.3 directly works without modification. For other survival models (such as Cox models), existing DRO theory does not work as advertised and we propose a sample splitting approach in Section 3.2 to obtain an approximate loss to minimize that does comply with DRO theory.

3.1 Class of Survival Models Convertible Into DRO Variants

Our technique for converting a survival model into its DRO variant works with any survival model that minimizes a loss of the form

$$L_{\text{average}}(\theta) = \frac{1}{n} \sum_{i=1}^{n} L_i(\theta; \mathcal{A}_i), \tag{3.1}$$

where the i-th loss term L_i depends on training point $i \in [n]$ as well as possibly other training points $\mathcal{A}_i \subseteq [n] \setminus \{i\}$. We refer to $\mathcal{A}_i$ as the *adjacency set* for the i-th training point, where $\mathcal{A}_i$ can be empty. Meanwhile, for any subset $\mathcal{I} \subseteq [n] \setminus \{i\}$, we assume that the function $L_i(\theta; \mathcal{I})$ is well-defined ($\mathcal{I}$ need not equal $\mathcal{A}_i$). To provide some intuition for the adjacency set $\mathcal{A}_i$ and the individual loss $L_i(\theta; \mathcal{I})$, we state what these are for the survival models we presented in Section 2.2.

Example 1 (Cox models)**.** *For the i-th training point, define the adjacency set*

$$\mathcal{A}_i \triangleq \{j \in [n] \setminus \{i\} \text{ such that } Y_j \geq Y_i\}, \tag{3.2}$$

and define the individual loss function

$$L_i(\theta; \mathcal{I}) \triangleq -\delta_i \left[f(X_i; \theta) - \log \left(\exp(f(X_i; \theta)) + \sum_{j \in \mathcal{I}} \exp(f(X_j; \theta)) \right) \right].$$

One can verify that plugging in these choices for $\mathcal{A}_i$ and L_i into equation (3.1) *yields the Cox loss from equation* (2.4)*.*

Example 2 (DeepHit)**.** *We first consider when hyperparameter $\beta \in [0, 1)$. For the i-th training point, define the adjacency set*

$$\mathcal{A}_i \triangleq \{j \in [n] \setminus \{i\} \text{ such that } \kappa(Y_j) > \kappa(Y_i)\}, \tag{3.3}$$

and define the individual loss function

$$\begin{aligned} L_i(\theta; \mathcal{I}) = \beta \cdot \big[-\delta_i \log(f_{\kappa(Y_i)}(X_i; \theta)) - (1 - \delta_i) \log(S_{\kappa(Y_i)}(X_i; \theta)) \big] \\ + (1 - \beta) \cdot \frac{1}{n} \cdot \delta_i \sum_{j \in \mathcal{I}} \exp\Big(\frac{S_{\kappa(Y_i)}(X_i; \theta) - S_{\kappa(Y_i)}(X_j; \theta)}{\sigma} \Big). \end{aligned} \tag{3.4}$$

One can verify that plugging in these choices for $\mathcal{A}_i$ and L_i into equation (3.1) *yields the DeepHit loss from equation* (2.8)*.*

Next, we consider when hyperparameter $\beta = 1$. In this case, we can still use $L_i(\theta; \mathcal{I})$ as defined in equation (3.4)*, where we note that the second term becomes 0; thus, $L_i(\theta; \mathcal{I})$ actually no longer depends on $\mathcal{I}$. In fact, when $\beta = 1$, it suffices to set $\mathcal{A}_i = \varnothing$ for all $i \in [n]$. Conceptually, what is happening is that the ranking loss disappears from the overall DeepHit loss* (2.7)*, and this ranking loss is the only loss term that introduces coupling between different data points.*

If $\mathcal{A}_i = \varnothing$ for all $i \in [n]$, then we can directly use the existing DRO optimization (2.15); the overall loss decouples across the different data points so we do not run into issues where multiple data points get "coupled".

Example 3 (SODEN)**.** *For the SODEN model, the overall loss function* (2.11) *actually has no coupling across training points, so it suffices to set $\mathcal{A}_i = \varnothing$ for all $i \in [n]$, and have $L_i(\theta; \mathcal{I}) = L_i^{SODEN}(\theta)$ using equation* (2.12) *(so $L_i(\theta; \mathcal{I})$ does not actually depend on $\mathcal{I}$).*

3.2 Applying DRO When Adjacency Sets Can be Nonempty

We now discuss how to use DRO when $\mathcal{A}_i$ is not guaranteed to be empty (as in the case of the Cox model or of DeepHit when $\beta \in [0, 1)$).

Heuristic approach To convert a survival analysis model that minimizes the loss (3.1) into one that uses DRO, a heuristic approach that does not comply with existing DRO theory would be to solve the DRO optimization problem (2.15), ignoring the fact that the individual loss terms are not guaranteed to depend only on a single data point each. To be clear, existing DRO theory effectively requires that the sets $\mathcal{A}_i$ are all empty. As a preview of our experimental results, we mention that this heuristic approach actually works well in practice but we lack any justification as to why it should be expected to work well.

Sample splitting approach We now propose a sample splitting approach that creates an approximate loss function that complies with existing DRO theory. We divide the training data into two sets $\mathcal{D}_1 \subset [n]$ and $\mathcal{D}_2 \triangleq [n] \setminus \mathcal{D}_1$ of sizes $n_1 \triangleq |\mathcal{D}_1|$ and $n_2 \triangleq |\mathcal{D}_2| = n - n_1$. The basic idea is that we treat the data points in $\mathcal{D}_2$ as fixed, and then define a DRO loss only over data points in $\mathcal{D}_1$. For each $i \in \mathcal{D}_1$, we replace its original individual loss $L_i(\theta; \mathcal{A}_i)$ with an approximate version $L_i(\theta; \mathcal{A}_i \cap \mathcal{D}_2)$ that only depends on the i-th point along with points in $\mathcal{D}_2$. Specifically, we minimize the new DRO loss function

$$L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2) \triangleq C_\alpha \sqrt{\frac{1}{|\mathcal{D}_1|} \sum_{i \in \mathcal{D}_1} [L_i(\theta; \mathcal{A}_i \cap D_2) - \eta]_+^2} + \eta. \tag{3.5}$$

The key observation is that conditioned on the points in $\mathcal{D}_2$, the loss terms $L_i(\theta; \mathcal{A}_i \cap D_2)$ appear i.i.d. across $i \in \mathcal{D}_1$ and the i-th loss only depends on the i-th data point (and possibly points in $\mathcal{D}_2$ which are treated as fixed). Hence, the original DRO theory applies. Note that our sample splitting strategy is somewhat inspired by the "case control" strategy by Kvamme et al. [2019], where instead of using the full Cox loss, they approximate each individual data point's loss (which could depend on many other data points) to only depend on a single other data point.

Although minimizing $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$ is compliant with DRO theory, it uses data "less effectively" since at most n_1 data points (rather than n) are used to compute the empirical average inside the square root in equation (3.5) (as compared to the empirical average inside the square root of $L_{\text{DRO}}(\theta, \eta)$ in equation (2.15)); as reminder, some individual loss terms might actually be zero (in the case of the Cox model, individual loss terms are zero for censored data). Moreover, in the new split loss $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$, each individual loss within the empirical average is computed using only a subset of each individual's original adjacency set (for each $i \in \mathcal{D}_1$, we approximate individual loss $L_i(\theta; \mathcal{A}_i)$ by $L_i(\theta; \mathcal{A}_i \cap \mathcal{D}_2)$).

To "more effectively" use data, we use the following simple strategy. Whereas the loss $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$ treats $\mathcal{D}_2$ as fixed and computes an average over $\mathcal{D}_1$, we also do the opposite: we treat $\mathcal{D}_1$

Algorithm 2: DRO (SPLIT)

Input: A training dataset $\{(X_i, Y_i, \delta_i)\}_{i=1}^n$, minimum subpopulation probability hyperparameter α, subset size n_1, learning rate ξ, max_iterations

Output: Survival model parameters $\widehat{\theta}$

Obtain initial survival model parameters $\widehat{\theta}_0$ (e.g., using default PyTorch parameter initialization).
Set $\mathcal{D}_1 \leftarrow \{1, 2, \ldots, n_1\}$ and $\mathcal{D}_2 \leftarrow \{n_1 + 1, \ldots, n\}$.
for $\ell = 0$ *to* max_iterations **do**
  for $i \in \mathcal{D}_1$ **do**
    Set $u_i \leftarrow L_i(\widehat{\theta}_\ell; \mathcal{A}_i \cap D_2)$.
  end
  Set $\widehat{\eta} \leftarrow \arg\min_{\eta \in \mathbb{R}} \left\{ \left(\sqrt{2(\frac{1}{\alpha} - 1)^2 + 1} \right) \sqrt{\frac{1}{n_1} \sum_{i \in \mathcal{D}_1} [u_i - \eta]_+^2} + \eta \right\}$, where this minimization is solved using binary search.
  for $i \in \mathcal{D}_2$ **do**
    Set $v_i \leftarrow L_i(\widehat{\theta}_\ell; \mathcal{A}_i \cap D_1)$.
  end
  Set $\widehat{\eta}' \leftarrow \arg\min_{\eta' \in \mathbb{R}} \left\{ \left(\sqrt{2(\frac{1}{\alpha} - 1)^2 + 1} \right) \sqrt{\frac{1}{n - n_1} \sum_{i \in \mathcal{D}_2} [v_i - \eta']_+^2} + \eta' \right\}$, where this minimization is solved using binary search.
  Set $\widehat{\theta}_{\ell+1} \leftarrow \widehat{\theta}_\ell - \xi \cdot (\nabla_\theta L_{\text{DRO}}^{\text{split}}(\widehat{\theta}_\ell, \widehat{\eta}, \mathcal{D}_1 \mid \mathcal{D}_2) + \nabla_\theta L_{\text{DRO}}^{\text{split}}(\widehat{\theta}_\ell, \widehat{\eta}', \mathcal{D}_2 \mid \mathcal{D}_1))$.
end
return $\widehat{\theta} \leftarrow \widehat{\theta}_{\text{max\_iterations}+1}$

as fixed and compute an average over $\mathcal{D}_2$, which would corresponds precisely to using the loss $L_{\text{DRO}}^{\text{split}}(\theta, \eta', \mathcal{D}_2 \mid \mathcal{D}_1)$; note that we use a different variable η' than the variable η used in $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$. Overall, we minimize the loss

$$L_{\text{DRO}}^{\text{split}}(\theta, \eta, \eta') \triangleq L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2) + L_{\text{DRO}}^{\text{split}}(\theta, \eta', \mathcal{D}_2 \mid \mathcal{D}_1)$$

via coordinate descent, alternating between the following steps:

- Treating η' and θ as fixed, we update η by finding the value of η that minimizes $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \eta')$. This amounts to solving $\min_{\eta \in \mathbb{R}} L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$ using binary search (since $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$ is convex w.r.t. η).
- Treating η and θ as fixed, we update η' by finding the value of η' that minimizes $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \eta')$. This amounts to solving $\min_{\eta' \in \mathbb{R}} L_{\text{DRO}}^{\text{split}}(\theta, \eta', \mathcal{D}_2 \mid \mathcal{D}_1)$ using binary search.
- Treating η and η' as fixed, we update θ by minimizing $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \eta')$ (e.g., using gradient descent).

We provide the pseudocode in Algorithm 2.

4 Experiments

To see how well our proposed DRO conversion strategy works in practice, we now conduct extensive experiments to evaluate the accuracy and fairness of DRO variants of different survival models compared to the original versions of these models, as well as to versions of these models modified to encourage fairness using existing fairness regularizers. We describe the datasets we use in Section 4.1, the experimental setup in Section 4.2, the evaluation metrics in Section 4.3, and the models evaluated in Section 4.4. We then present our experimental results in Section 4.5. Lastly, we show how to compare across multiple models using a plot inspired by ROC curves in Section 4.6.

4.1 Datasets

We use three standard, publicly available survival analysis datasets:

Table 4.1: Basic dataset characteristics.

	FLC	SUPPORT	SEER
# samples	7,874	9,105	28,018
# features	6 (9*)	14 (19*)	11
Censoring rate	0.725	0.319	0.654
Sensitive attributes	age, gender	age, race, gender	age, race

* indicates the number before preprocessing (preprocessing removes some features)

- The **FLC** dataset [Dispenzieri et al., 2012] is from a study on the relationship between serum free light chain (FLC) and mortality of Olmsted County residents aged 50 or higher. We treat discretized age (age$\leq$65 and age$>$65) and gender (women and men) as sensitive attributes.
- The **SUPPORT** dataset [Knaus et al., 1995] is from a study at Vanderbilt University on understanding prognoses, preferences, outcomes, and risks of treatment by analyzing survival times of severely ill hospitalized patients. We treat discretized age (age$\leq$65 and age$>$65), race (white and non-white), and gender (women and men) as sensitive attributes.
- The **SEER** dataset is on breast cancer patients from the Sureillance, Epidemiology, and End Results (SEER) program of the National Cancer Institute. We collected this dataset using the data extraction software from the official SEER program of the National Cancer Institute. We used 11 covariates that also appear in an existing snapshot of the SEER dataset [Teng, 2019] that only contained 4024 data points. We also treat discretized age (age$\leq$65 and age$>$65) and race (white and non-white) as sensitive attributes.

These datasets have appeared in existing fair survival analysis research (e.g., Keya et al. 2021, Rahman and Purushotham 2022, Zhang and Weiss 2022) although not always with all three of these appearing within the same paper. Basic characteristics of these datasets are reported in Table 4.1.

4.2 Experimental Setup

For all models, we first use a random 80%/20% train/test split to hold out a test set that will be the same across experimental repeats for all datasets. Then we repeat the following basic experiment 10 times: (1) We hold out 20% of the training data to treat as a validation set, which is used to tune hyperparameters. (2) We then compute evaluation metrics across the same test set. We describe the evaluation metrics and how hyperparameter tuning works shortly. When we report our experimental results, we provide the mean and standard deviation of each metric across the 10 experimental repeats. More hyperparameter settings can be found in Appendix C.

4.3 Evaluation Metrics

For accuracy metrics, we use Time-dependent concordance index (C^{td}, higher is better) [Antolini et al., 2005] and Integrated IPCW Brier Score (IBS, lower is better) [Graf et al., 1999]. For fairness metrics, we use the concordance imparity (CI) fairness metric by [Zhang and Weiss, 2022], Censoring-based individual fairness (F_{CI}) [Rahman and Purushotham, 2022], and Censoring-based group fairness (F_{CG}) [Rahman and Purushotham, 2022]. For these fairness metrics, lower is better. Definitions of these fairness metrics are in Appendix B.

Note that the fairness metrics CI and F_{CG} require us to specify groups. For the FLC dataset, we use (discretized) age and, separately, gender (i.e., we first run experiments using only age in evaluating CI and F_{CG}; we then re-run experiments using gender instead of age). For the SUPPORT dataset, we separately use gender, age, and race. For the SEER dataset, we separately use race and age.

4.4 Models Evaluated

Working off our running examples from Section 2.2, we consider Cox models (classical and deep), DeepHit, and SODEN. For each of these, we compare the original model with its DRO variants using our conversion strategy (the heuristic approach and also the sample splitting approach stated in Section 3.2; for SODEN, there is no need to do sample splitting and the heuristic approach is actually exact). We also try versions of the original models modified to encourage fairness using existing fairness regularizers.

Cox models We separately experiment on the classical **linear** setting (the log partial hazard function is $f(x;\theta) = \theta^T x$) or the "deep" **nonlinear** setting in which f is a multilayer perceptron (MLP). In the linear case, we denote the heuristic DRO variant as DRO-COX and the sample splitting DRO variant as DRO-COX (SPLIT). For the nonlinear case, we add the prefix "Deep" to these names for clarity.

In terms of baselines, we use the unregularized linear Cox model [Cox, 1972] (denoted as "Cox"), whereas the unregularized nonlinear Cox model [Katzman et al., 2018] is denoted as "DeepSurv". The rest of our baselines are all regularized versions of either the standard Cox or DeepSurv models, using different fairness regularization terms. When we use individual, group, or intersectional regularization terms by Keya et al. [2021] (we discuss these in Appendix B), we add the suffix "$_I$(Keya et al.)", "$_G$(Keya et al.)", or "$_\cap$(Keya et al.)" respectively to a model name; for example, "DeepSurv$_G$(Keya et al.)" corresponds to DeepSurv with group fairness regularization by Keya et al. [2021]. When we use the individual or group fairness regularization terms that account for observed times and censoring information [Rahman and Purushotham, 2022], we instead use the suffix "$_I$(R&P)" or "$_G$(R&P)".[1] Note that group fairness regularization (suffixes "$_G$(Keya et al.)" and "$_G$(R&P)") uses the same groups that test set CI and F_{CG} fairness metrics use.

In terms of hyperparameter tuning, we use the strategy by Keya et al. [2021]: the final hyperparameter setting used per dataset and per method is determined based on a preset rule in practice that allows up to a 5% degradation in the validation set C^{td} from the classical Cox model (for the linear setting) or DeepSurv (for the nonlinear setting) while minimizing the validation set CI fairness metric or F_{CG} fairness metric (see Appendix D).

DeepHit and SODEN For DeepHit [Tang et al., 2022b], we denote its heuristic DRO variant as DRO-DEEPHIT and its sample splitting DRO variant as DRO-DEEPHIT (SPLIT). For SODEN [Tang et al., 2022b], there is only one DRO variant to consider which we denote as DRO-SODEN.

In terms of baselines, we consider the original DeepHit and SODEN models that do not account for fairness. We further adapt the group-based fairness regularization that accounts for censoring from Rahman and Purushotham [2022] to each of DeepHit and SODEN separately as additional baselines (DEEPHIT$_G$(R&P) and SODEN$_G$(R&P)).

The hyperparameter setting used per dataset and per method is also determined based on a preset rule in practice that allows up to a 5% degradation in the validation set C^{td} from the original model (that does not encourage fairness) while minimizing the validation set CI fairness metric or F_{CG} fairness metric. Hyperparameter grids for all methods are in Appendix C, where we also provide information on the compute environment that we used.

4.5 Experimental Results

Cox models We compare DRO-COX and DRO-COX (SPLIT) against various baselines using a similar experimental setup as Keya et al. [2021]. Specifically, we report the test set evaluation metrics for FLC (using age to evaluate CI and F_{CG}) in Table 4.2, SUPPORT (gender) in Table 4.3, and SEER

[1] Rahman and Purushotham [2022] did not propose an intersectional fairness regularizer and technically did not try regularized versions of Cox models using their fairness definitions. However, it is straightforward to adapt their individual and group fairness definitions as regularization terms for a Cox model, especially as their work is directly modifying definitions by Keya et al. [2021].

Table 4.2: Cox model test set accuracy and fairness metrics on the FLC (age) dataset. We report mean and standard deviation (in parentheses) across 10 experimental repeats (each repeat holds out a different 20% of the training data as a validation set for hyperparameter tuning; the test set is the same across experimental repeats). Higher is better for metrics with "↑", while lower is better for metrics with "↓". The best results are shown in bold for linear and, separately, nonlinear models. When one of our methods outperforms all baselines (in linear and, separately, nonlinear models), we highlight the corresponding cell in green.

	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS↓	CI(%)↓	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS↓	CI(%)↓	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	Cox	0.8032 (0.0002)	0.1739 (0.0004)	0.5350 (0.0413)	0.3608 (0.0045)	0.0744 (0.0011)	**0.8032 (0.0002)**	0.1739 (0.0004)	0.4479 (0.0919)	0.3608 (0.0045)	0.0744 (0.0011)
	Cox_I(Keya et al.)	0.7937 (0.0068)	0.1414 (0.0073)	0.5400 (0.3270)	0.1081 (0.0254)	0.0210 (0.0051)	0.7923 (0.0074)	0.1334 (0.0034)	0.2280 (0.2540)	0.0551 (0.0048)	0.0100 (0.0013)
	Cox_I(R&P)	**0.8034 (0.0007)**	0.1636 (0.0030)	0.4330 (0.1196)	0.2291 (0.0256)	0.0452 (0.0053)	0.8020 (0.0015)	0.1565 (0.0008)	0.2219 (0.1426)	0.1688 (0.0050)	0.0326 (0.0012)
	Cox_G(Keya et al.)	0.7974 (0.0117)	0.1492 (0.0077)	0.3410 (0.3011)	0.2465 (0.1576)	0.1201 (0.0905)	0.7862 (0.0133)	0.1413 (0.0035)	0.3026 (0.3611)	0.0693 (0.0248)	0.0168 (0.0070)
	Cox_G(R&P)	0.8027 (0.0005)	0.1676 (0.0012)	0.4950 (0.1517)	0.2718 (0.0130)	0.0539 (0.0026)	0.8012 (0.0004)	0.1637 (0.0006)	0.1871 (0.0567)	0.2323 (0.0046)	0.0455 (0.0011)
	$Cox_\cap$(Keya et al.)	0.7870 (0.0029)	0.1400 (0.0005)	1.0790 (0.1098)	0.0615 (0.0012)	0.0153 (0.0013)	0.7875 (0.0021)	0.1402 (0.0004)	0.5904 (0.5340)	0.0618 (0.0014)	0.0149 (0.0017)
	DRO-COX	0.7959 (0.0036)	0.1408 (0.0050)	**0.0510 (0.0401)**	0.0631 (0.0413)	0.0116 (0.0076)	0.7958 (0.0049)	**0.1330 (0.0002)**	**0.0810 (0.1139)**	**0.0000 (0.0000)**	**0.0000 (0.0000)**
	DRO-COX (SPLIT)	0.7964 (0.0045))	**0.1389 (0.0008)**	0.1175 (0.1482)	**0.0000 (0.0000)**	**0.0000 (0.0000)**	0.7964 (0.0045))	**0.1389 (0.0008)**	0.1175 (0.1482)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
Nonlinear	DeepSurv	0.8070 (0.0014)	0.1767 (0.0018)	0.2940 (0.2147)	0.5788 (0.2493)	0.1314 (0.0606)	0.8070 (0.0014)	0.1767 (0.0018)	0.2940 (0.2147)	0.5788 (0.2493)	0.1314 (0.0606)
	$DeepSurv_I$(Keya et al.)	0.7884 (0.0070)	0.1441 (0.0130)	0.3700 (0.2523)	0.0355 (0.0266)	0.0082 (0.0067)	0.7994 (0.0069)	0.1672 (0.0051)	0.3155 (0.4907)	0.0001 (0.0002)	0.0000 (0.0001)
	$DeepSurv_I$(R&P)	0.8071 (0.0041)	0.1729 (0.0093)	0.1870 (0.1117)	0.0138 (0.0264)	0.0036 (0.0073)	0.8084 (0.0021)	0.1757 (0.0029)	**0.0912 (0.1975)**	0.0004 (0.0002)	0.0001 (0.0001)
	$DeepSurv_G$(Keya et al.)	0.7990 (0.0120)	0.4190 (0.2487)	0.2490 (0.1646)	0.0446 (0.1048)	0.0240 (0.0609)	0.8061 (0.0020)	0.4713 (0.2142)	0.1350 (0.2092)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
	$DeepSurv_G$(R&P)	**0.8073 (0.0036)**	0.1731 (0.0087)	0.2290 (0.1344)	0.0472 (0.0488)	0.0119 (0.0133)	**0.8092 (0.0017)**	0.1764 (0.0022)	0.0955 (0.1085)	0.0060 (0.0022)	0.0014 (0.0005)
	$DeepSurv_\cap$(Keya et al.)	0.7751 (0.0018)	**0.1357 (0.0002)**	0.4300 (0.1091)	0.0394 (0.0010)	0.0125 (0.0005)	0.7751 (0.0018)	0.1357 (0.0002)	0.2347 (0.2100)	0.0394 (0.0010)	0.0125 (0.0005)
	Deep DRO-COX	0.8068 (0.0024)	0.1595 (0.0135)	**0.0730 (0.0822)**	0.2865 (0.2425)	0.0642 (0.0573)	0.7781 (0.0091)	**0.1331 (0.0002)**	1.2177 (1.2368)	0.0054 (0.0012)	0.0012 (0.0002)
	Deep DRO-COX (SPLIT)	0.7784 (0.0092)	0.1647 (0.0037)	1.1632 (1.1853)	**0.0054 (0.0013)**	**0.0011 (0.0002)**	0.7784 (0.0092)	0.1647 (0.0037)	1.1632 (1.1853)	0.0054 (0.0013)	0.0011 (0.0002)

Table 4.3: Cox model test set scores on the SUPPORT (gender) dataset, in the same format as Table 4.2.

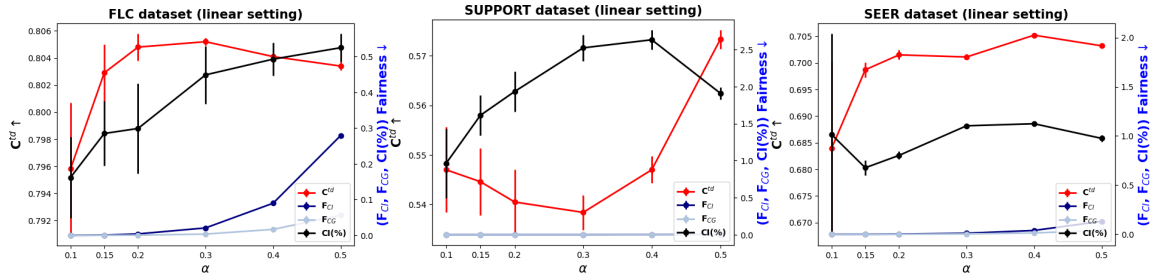
	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS↓	CI(%)↓	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS↓	CI(%)↓	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	Cox	0.6025 (0.0005)	0.2304 (0.0015)	1.4300 (0.0654)	0.0207 (0.0008)	0.0105 (0.0004)	**0.6025 (0.0005)**	0.2304 (0.0015)	1.4300 (0.0654)	0.0207 (0.0008)	0.0105 (0.0004)
	Cox_I(Keya et al.)	0.5881 (0.0114)	**0.2157 (0.0060)**	0.9650 (0.6126)	0.0028 (0.0026)	0.0014 (0.0013)	0.5829 (0.0099)	**0.2147 (0.0063)**	0.5665 (0.7451)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
	Cox_I(R&P)	0.6019 (0.0019)	0.2282 (0.0013)	1.4190 (0.1002)	0.0177 (0.0012)	0.0090 (0.0006)	0.6024 (0.0008)	0.2276 (0.0011)	0.6989 (0.6852)	0.0169 (0.0005)	0.0086 (0.0002)
	Cox_G(Keya et al.)	**0.6030 (0.0007)**	0.2297 (0.0018)	1.4190 (0.0632)	0.0197 (0.0011)	0.0101 (0.0005)	0.6024 (0.0006)	0.2284 (0.0009)	0.7273 (0.7103)	0.0185 (0.0003)	0.0095 (0.0001)
	Cox_G(R&P)	0.6018 (0.0017)	0.2295 (0.0009)	1.4340 (0.1039)	0.0196 (0.0005)	0.0100 (0.0002)	0.6026 (0.0008)	0.2290 (0.0012)	0.7054 (0.6896)	0.0188 (0.0005)	0.0096 (0.0002)
	$Cox_\cap$(Keya et al.)	0.5715 (0.0062)	0.2275 (0.0016)	1.1270 (0.2457)	0.0131 (0.0013)	0.0067 (0.0006)	0.5631 (0.0070)	0.2264 (0.0017)	0.4383 (0.4752)	0.0117 (0.0012)	0.0059 (0.0006)
	DRO-COX	0.5734 (0.0019)	0.2210 (0.0010)	0.4350 (0.0674)	0.0028 (0.0001)	0.0015 (0.0001)	0.5641 (0.0105)	0.2211 (0.0010)	**0.1930 (0.2308)**	0.0019 (0.0012)	0.0010 (0.0006)
	DRO-COX (SPLIT)	0.5701 (0.0056)	0.4569 (0.1314)	**0.1941 (0.2088)**	**0.0023 (0.0011)**	**0.0012 (0.0006)**	0.5701 (0.0056)	0.4570 (0.1314)	**0.1941 (0.2088)**	0.0023 (0.0011)	0.0012 (0.0006)
Nonlinear	DeepSurv	0.6108 (0.0029)	0.2417 (0.0016)	1.6220 (0.3303)	0.0453 (0.0041)	0.0233 (0.0021)	0.6108 (0.0029)	0.2417 (0.0016)	1.6220 (0.3303)	0.0453 (0.0041)	0.0233 (0.0021)
	$DeepSurv_I$(Keya et al.)	0.5984 (0.0124)	0.2376 (0.0182)	1.3280 (0.7670)	**0.0012 (0.0020)**	**0.0006 (0.0011)**	0.6031 (0.0059)	0.2459 (0.0102)	**0.5795 (0.8413)**	**0.0000 (0.0000)**	**0.0000 (0.0000)**
	$DeepSurv_I$(R&P)	0.6104 (0.0076)	0.2379 (0.0079)	1.6490 (0.2368)	0.0049 (0.0046)	0.0026 (0.0024)	**0.6124 (0.0048)**	0.2448 (0.0043)	0.8630 (0.8925)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
	$DeepSurv_G$(Keya et al.)	0.5982 (0.0109)	0.2436 (0.0121)	1.6540 (0.3892)	0.0111 (0.0082)	0.0057 (0.0042)	0.6034 (0.0037)	0.2499 (0.0024)	0.6218 (0.6885)	0.0046 (0.0007)	0.0024 (0.0004)
	$DeepSurv_G$(R&P)	**0.6110 (0.0057)**	0.2406 (0.0068)	1.6250 (0.1931)	0.0039 (0.0049)	0.0020 (0.0025)	0.6117 (0.0049)	0.2440 (0.0034)	0.8281 (0.8359)	0.0002 (0.0000)	0.0001 (0.0000)
	$DeepSurv_\cap$(Keya et al.)	0.6015 (0.0069)	0.2378 (0.0053)	1.4110 (0.2129)	0.0256 (0.0054)	0.0131 (0.0027)	0.5912 (0.0012)	0.2309 (0.0011)	0.7786 (0.7659)	0.0182 (0.0008)	0.0094 (0.0004)
	Deep DRO-COX	0.5829 (0.0067)	**0.2240 (0.0010)**	1.2600 (0.4412)	0.0117 (0.0049)	0.0061 (0.0025)	0.5754 (0.0120)	**0.2227 (0.0011)**	0.7807 (0.8404)	0.0065 (0.0023)	0.0034 (0.0012)
	Deep DRO-COX (SPLIT)	0.5772 (0.0093)	0.6387 (0.0007)	**0.7799 (0.8410)**	0.0069 (0.0021)	0.0036 (0.0010)	0.5772 (0.0093)	0.6387 (0.0007)	0.7799 (0.8410)	0.0069 (0.0021)	0.0036 (0.0010)

(race) in Table 4.4. Experimental results using other sensitive attributes for the datasets have similar trends and are in Appendix D. From these tables, we have the following observations:

- Among linear methods, the heuristic DRO-COX method consistently outperforms baselines in terms of the CI fairness metric (and often on the other fairness metrics too) while still achieving reasonably high accuracy scores. A similar trend holds among nonlinear methods for the heuristic deep DRO-COX variant.
- The performance difference (in terms of both accuracy and fairness) between the heuristic DRO-COX and sample-splitting-based DRO-COX (SPLIT) is not clear cut; sometimes one performs better

Table 4.4: Cox model test set scores on the SEER (race) dataset, in the same format as Table 4.2.

	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	Cox	0.7025 (0.0003)	0.2128 (0.0009)	1.3445 (1.1759)	0.1690 (0.0053)	0.1299 (0.0041)	**0.7025 (0.0003)**	0.2128 (0.0009)	1.3445 (1.1759)	0.1690 (0.0053)	0.1299 (0.0041)
Linear	Cox_I(Keya et al.)	0.6894 (0.0046)	**0.1837 (0.0027)**	1.9930 (1.0836)	0.0110 (0.0013)	0.0090 (0.0010)	0.6894 (0.0046)	**0.1837 (0.0027)**	1.9930 (1.0836)	0.0110 (0.0013)	0.0090 (0.0010)
Linear	Cox_I(R&P)	0.7022 (0.0022)	0.1974 (0.0007)	1.4192 (1.1402)	0.0823 (0.0022)	0.0638 (0.0015)	0.7023 (0.0022)	0.1973 (0.0008)	1.2311 (1.1520)	0.0822 (0.0021)	0.0636 (0.0013)
Linear	Cox_G(Keya et al.)	0.6952 (0.0146)	0.2073 (0.0049)	1.5037 (1.4153)	0.1384 (0.0256)	0.1100 (0.0215)	0.6952 (0.0146)	0.2073 (0.0049)	1.5037 (1.4153)	0.1384 (0.0256)	0.1100 (0.0215)
Linear	Cox_G(R&P)	**0.7027 (0.0022)**	0.2000 (0.0004)	1.3053 (1.2116)	0.0965 (0.0015)	0.0723 (0.0014)	0.7024 (0.0024)	0.2001 (0.0008)	1.3214 (1.2252)	0.0967 (0.0020)	0.0720 (0.0009)
Linear	$\text{Cox}_\cap$(Keya et al.)	0.6494 (0.0016)	0.1963 (0.0012)	1.4998 (1.0451)	0.0707 (0.0058)	0.0562 (0.0045)	0.6494 (0.0016)	0.1963 (0.0012)	1.4998 (1.0451)	0.0707 (0.0058)	0.0562 (0.0045)
Linear	DRO-COX	0.6927 (0.0069)	0.1868 (0.0004)	**1.1545 (1.2119)**	0.0001 (0.0001)	0.0001 (0.0001)	0.6927 (0.0069)	0.1868 (0.0004)	**1.1545 (1.2119)**	0.0001 (0.0001)	0.0001 (0.0001)
Linear	DRO-COX (SPLIT)	0.6872 (0.0047)	0.1869 (0.0004)	1.4140 (1.5085)	**0.0000 (0.0000)**	**0.0000 (0.0000)**	0.6872 (0.0047)	0.1869 (0.0004)	1.4140 (1.5085)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
Nonlinear	DeepSurv	**0.7095 (0.0014)**	0.2200 (0.0012)	1.4812 (1.1237)	0.3635 (0.1116)	0.2792 (0.0844)	**0.7095 (0.0014)**	0.2200 (0.0012)	1.4812 (1.1237)	0.3635 (0.1116)	0.2792 (0.0844)
Nonlinear	DeepSurv_I(Keya et al.)	0.6982 (0.0045)	0.2127 (0.0032)	1.7870 (0.9286)	**0.0000 (0.0000)**	**0.0000 (0.0000)**	0.6982 (0.0045)	0.2127 (0.0032)	1.7870 (0.9286)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
Nonlinear	DeepSurv_I(R&P)	0.7078 (0.0015)	0.2167 (0.0012)	1.3616 (1.2716)	0.0001 (0.0001)	0.0001 (0.0001)	0.7078 (0.0015)	0.2167 (0.0012)	1.2616 (1.2716)	0.0001 (0.0001)	0.0001 (0.0001)
Nonlinear	DeepSurv_G(Keya et al.)	0.7034 (0.0016)	0.2154 (0.0007)	1.3546 (1.2420)	0.1172 (0.0343)	0.0950 (0.0277)	0.7034 (0.0016)	0.2154 (0.0007)	1.3546 (1.2420)	0.1172 (0.0343)	0.0950 (0.0277)
Nonlinear	DeepSurv_G(R&P)	0.7079 (0.0017)	0.2167 (0.0012)	1.3392 (1.2448)	0.0013 (0.0004)	0.0011 (0.0004)	0.7079 (0.0017)	0.2167 (0.0012)	1.2392 (1.2448)	0.0013 (0.0004)	0.0011 (0.0004)
Nonlinear	$\text{DeepSurv}_\cap$(Keya et al.)	0.6537 (0.0054)	0.1998 (0.0008)	1.5587 (1.5744)	0.0694 (0.0143)	0.0558 (0.0115)	0.6537 (0.0054)	0.1998 (0.0008)	1.5587 (0.5744)	0.0694 (0.0143)	0.0558 (0.0115)
Nonlinear	Deep DRO-COX	0.6830 (0.0050)	**0.1869 (0.0004)**	1.2908 (1.3424)	0.0006 (0.0004)	0.0006 (0.0003)	0.6830 (0.0050)	**0.1869 (0.0004)**	1.2908 (1.3424)	0.0006 (0.0004)	0.0006 (0.0003)
Nonlinear	Deep DRO-COX (SPLIT)	0.6829 (0.0049)	0.1881 (0.0012)	**1.2443 (1.2960)**	0.0006 (0.0005)	0.0005 (0.0004)	0.6829 (0.0049)	0.1881 (0.0012)	**1.2443 (1.2960)**	0.0006 (0.0005)	0.0005 (0.0004)



(a) FLC (age) (b) SUPPORT (age) (c) SEER (age)

Figure 1: Effect of α on test set accuracy (c-index; higher is better) and fairness metrics (F_I, F_G, $F_\cap$, F_A, and CI; lower is better for all fairness metrics) of DRO-COX on four datasets.

than the other and vice versa. This holds for their linear variants as well as, separately, their nonlinear (deep) variants.

- As expected, the unregularized Cox and DeepSurv models often have (among) the highest accuracy scores but tend to have poor performance on fairness metrics.
- The baselines that are regularized variants of Cox and DeepSurv typically do not simultaneously achieve low scores across all fairness metrics. Even though some of these can work well with some of the metrics by Keya et al. [2021], they clearly do not work as well as our DRO-COX variants when it comes to the CI fairness metric that actually accounts for accuracy.

Effect of α. To show how α trades off between fairness and accuracy, we show results for DRO-COX in the linear setting across all datasets (using age for evaluating F_G and CI) in Figure 1, where we use c-index as the accuracy metric. It is clear that accuracy tends to increase when α increases from 0.1 to 0.3 on FLC and SEER, and from 0.3 to 0.5 on SUPPORT. However, the increase in α results in worse scores across fairness metrics.

Additional experiments. Across all methods, instead of minimizing the validation set CI fairness metric during hyperparameter tuning (tolerating a small degradation in the validation set C^{td}), we also tried instead minimizing the validation set F_{CG} metric and found similar results. We also show that our DRO-COX (SPLIT) procedure is somewhat robust to the choice of n_1 and n_2, and if DRO-COX (SPLIT) did not use both losses $L^{\text{split}}_{\text{DRO}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$ and $L^{\text{split}}_{\text{DRO}}(\theta, \eta, \mathcal{D}_2 \mid \mathcal{D}_1)$ (i.e., if it only used one of these), then it performs worse. For details on these experiments, see Appendix D.

Table 4.5: DeepHit test set scores on the FLC, SUPPORT, SEER datasets when hyperparameter tuning is based on CI and F_{CG}.

Datasets	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
FLC (age)	DeepHit	0.7937 (0.0080)	0.1560 (0.0204)	1.1950 (0.7885)	0.0108 (0.0012)	0.0038 (0.0005)	**0.7937 (0.0080)**	0.1560 (0.0204)	1.1950 (0.7885)	0.0108 (0.0012)	0.0038 (0.0005)
	DeepHit$_G$(R&P)	0.7825 (0.0237)	**0.1449 (0.0201)**	1.2340 (0.6046)	0.0097 (0.0023)	0.0038 (0.0005)	0.7446 (0.0051)	**0.1326 (0.0027)**	2.1110 (0.4770)	**0.0064 (0.0001)**	**0.0035 (0.0001)**
	DRO-DEEPHIT	**0.7956 (0.0051)**	0.1971 (0.0543)	1.0430 (0.4835)	0.0084 (0.0028)	0.0032 (0.0010)	0.7821 (0.0101)	0.2754 (0.0076)	1.0180 (0.5330)	**0.0026 (0.0005)**	**0.0012 (0.0002)**
	DRO-DEEPHIT (SPLIT)	0.7748 (0.0189)	0.2264 (0.0623)	**0.9950 (0.4556)**	**0.0067 (0.0036)**	**0.0028 (0.0011)**	0.7622 (0.0122)	0.2734 (0.0092)	**0.9270 (0.5411)**	0.0027 (0.0009)	0.0014 (0.0004)
FLC (gender)	DeepHit	0.7937 (0.0080)	0.1560 (0.0204)	0.4990 (0.3792)	0.0108 (0.0012)	0.0095 (0.0011)	**0.7937 (0.0080)**	**0.1560 (0.0204)**	**0.4990 (0.3792)**	0.0108 (0.0012)	0.0095 (0.0011)
	DeepHit$_G$(R&P)	0.7840 (0.0245)	**0.1489 (0.0212)**	0.5170 (0.3982)	0.0099 (0.0021)	0.0089 (0.0016)	**0.7937 (0.0080)**	**0.1560 (0.0204)**	**0.4990 (0.3792)**	0.0108 (0.0012)	0.0095 (0.0011)
	DRO-DEEPHIT	**0.7956 (0.0051)**	0.1971 (0.0543)	**0.4320 (0.4786)**	0.0084 (0.0028)	0.0078 (0.0023)	0.7821 (0.0101)	0.2754 (0.0076)	1.3700 (0.6702)	**0.0026 (0.0005)**	0.0028 (0.0005)
	DRO-DEEPHIT (SPLIT)	0.7748 (0.0189)	0.2264 (0.0623)	1.3100 (0.9915)	**0.0067 (0.0036)**	**0.0062 (0.0030)**	0.7622 (0.0122)	0.2734 (0.0092)	1.9350 (0.7234)	0.0027 (0.0009)	**0.0027 (0.0009)**
SUPPORT (age)	DeepHit	**0.6029 (0.0071)**	0.2151 (0.0067)	3.5910 (0.3987)	0.0055 (0.0008)	0.0049 (0.0008)	**0.6029 (0.0071)**	0.2151 (0.0067)	3.5910 (0.3987)	0.0055 (0.0008)	0.0049 (0.0008)
	DeepHit$_G$(R&P)	0.5775 (0.0050)	**0.2123 (0.0009)**	**1.1940 (0.8221)**	0.0046 (0.0006)	0.0044 (0.0005)	0.5766 (0.0033)	**0.2126 (0.0007)**	**1.0230 (0.4416)**	0.0044 (0.0002)	0.0042 (0.0002)
	DRO-DEEPHIT	0.5932 (0.0159)	0.2447 (0.0147)	2.9160 (0.8347)	**0.0014 (0.0009)**	**0.0014 (0.0009)**	0.5899 (0.0154)	0.2493 (0.0159)	3.3740 (0.6078)	**0.0007 (0.0002)**	**0.0008 (0.0002)**
	DRO-DEEPHIT (SPLIT)	0.5753 (0.0236)	0.2225 (0.0112)	2.7280 (0.9570)	0.0044 (0.0013)	0.0041 (0.0010)	0.5792 (0.0234)	0.2392 (0.0268)	3.5270 (0.7331)	0.0037 (0.0019)	0.0035 (0.0014)
SUPPORT (gender)	DeepHit	**0.6029 (0.0071)**	0.2151 (0.0067)	0.5880 (0.2895)	0.0055 (0.0008)	0.0052 (0.0008)	**0.6029 (0.0071)**	0.2151 (0.0067)	**0.5880 (0.2895)**	0.0055 (0.0008)	0.0052 (0.0008)
	DeepHit$_G$(R&P)	0.5767 (0.0034)	**0.2126 (0.0008)**	0.6960 (0.3183)	0.0044 (0.0002)	0.0043 (0.0002)	0.5773 (0.0039)	**0.2125 (0.0007)**	0.7600 (0.2994)	0.0043 (0.0002)	0.0043 (0.0001)
	DRO-DEEPHIT	0.5932 (0.0159)	0.2447 (0.0147)	1.1980 (0.6834)	**0.0014 (0.0009)**	**0.0016 (0.0009)**	0.5899 (0.0154)	0.2493 (0.0159)	1.4460 (0.4235)	**0.0007 (0.0002)**	**0.0008 (0.0003)**
	DRO-DEEPHIT (SPLIT)	0.5753 (0.0236)	0.2225 (0.0112)	**0.5160 (0.3942)**	0.0044 (0.0013)	0.0043 (0.0011)	0.5792 (0.0234)	0.2392 (0.0268)	0.7550 (0.5022)	0.0037 (0.0019)	0.0037 (0.0015)
SUPPORT (race)	DeepHit	**0.6029 (0.0071)**	0.2151 (0.0067)	1.2250 (0.4454)	0.0055 (0.0008)	0.0062 (0.0010)	**0.6029 (0.0071)**	0.2151 (0.0067)	1.2250 (0.4454)	0.0055 (0.0008)	0.0062 (0.0010)
	DeepHit$_G$(R&P)	0.5767 (0.0031)	**0.2126 (0.0008)**	**0.7290 (0.4122)**	0.0044 (0.0002)	0.0050 (0.0003)	0.5813 (0.0108)	**0.2144 (0.0041)**	**0.7400 (0.4211)**	0.0043 (0.0003)	0.0049 (0.0005)
	DRO-DEEPHIT	0.5932 (0.0159)	0.2447 (0.0147)	1.0630 (0.5174)	**0.0014 (0.0009)**	**0.0019 (0.0010)**	0.5899 (0.0154)	0.2493 (0.0159)	1.4220 (0.4302)	**0.0007 (0.0002)**	**0.0010 (0.0003)**
	DRO-DEEPHIT (SPLIT)	0.5753 (0.0236)	0.2225 (0.0112)	1.1930 (0.4449)	0.0044 (0.0013)	0.0051 (0.0013)	0.5792 (0.0234)	0.2392 (0.0268)	1.5640 (0.6744)	0.0037 (0.0019)	0.0044 (0.0019)
SEER (age)	DeepHit	**0.7156 (0.0047)**	**0.1715 (0.0038)**	1.4450 (0.2901)	0.0122 (0.0011)	0.0069 (0.0005)	**0.7156 (0.0047)**	**0.1715 (0.0038)**	1.4450 (0.2901)	0.0122 (0.0011)	0.0069 (0.0005)
	DeepHit$_G$(R&P)	0.7122 (0.0086)	0.1743 (0.0064)	1.4160 (0.2443)	0.0105 (0.0029)	0.0063 (0.0013)	0.6987 (0.0025)	0.1801 (0.0021)	2.0960 (0.4633)	0.0046 (0.0002)	0.0037 (0.0001)
	DRO-DEEPHIT	0.7112 (0.0084)	0.2794 (0.0871)	**0.7990 (0.3281)**	**0.0061 (0.0041)**	**0.0038 (0.0023)**	0.6951 (0.0051)	0.4122 (0.0304)	1.3800 (0.5574)	**0.0002 (0.0002)**	**0.0002 (0.0003)**
	DRO-DEEPHIT (SPLIT)	0.6969 (0.0211)	0.2073 (0.0464)	1.0240 (0.3449)	0.0107 (0.0016)	0.0070 (0.0006)	0.6963 (0.0224)	0.2063 (0.0419)	**1.2630 (0.7467)**	0.0098 (0.0025)	0.0064 (0.0009)
SEER (race)	DeepHit	**0.7156 (0.0047)**	**0.1715 (0.0038)**	3.2820 (0.5958)	0.0122 (0.0011)	0.0169 (0.0017)	**0.7156 (0.0047)**	**0.1715 (0.0038)**	3.2820 (0.5958)	0.0122 (0.0011)	0.0169 (0.0017)
	DeepHit$_G$(R&P)	0.7132 (0.0073)	0.1728 (0.0045)	3.1330 (0.8321)	0.0113 (0.0028)	0.0157 (0.0038)	0.6987 (0.0048)	0.1806 (0.0028)	**1.6760 (0.6385)**	0.0045 (0.0023)	0.0066 (0.0030)
	DRO-DEEPHIT	0.7112 (0.0084)	0.2794 (0.0871)	3.0120 (0.5652)	**0.0061 (0.0041)**	**0.0089 (0.0056)**	0.6951 (0.0051)	0.4122 (0.0304)	3.2520 (1.7820)	**0.0002 (0.0002)**	**0.0004 (0.0004)**
	DRO-DEEPHIT (SPLIT)	0.6969 (0.0211)	0.2073 (0.0464)	**2.7700 (0.5636)**	0.0107 (0.0016)	0.0155 (0.0016)	0.6963 (0.0224)	0.2063 (0.0419)	3.0070 (0.8355)	0.0098 (0.0025)	0.0142 (0.0031)

DeepHit We now compare DRO-DEEPHIT and DRO-DEEPHIT (SPLIT) to the original DeepHit method [Lee et al., 2018] and the regularized variant DEEPHIT$_G$(R&P). We report the test performance on all three datasets in Table 4.5. According to the results in Table 4.5, we have the following observations:

- Our DRO variants can achieve better CI performance than the original DeepHit method on most of datasets with different sensitive attributes when using a CI-based hyperparameter tuning strategy. It is also clear that DRO-DEEPHIT and DRO-DEEPHIT (SPLIT) can achieve lower values on F_{CI} and F_{CG} on all datasets when using an F_{CG}-based hyperparameter tuning strategy. These results indicate that our DRO methods can encourage fairness for DeepHit and can obtain better fairness scores than DEEPHIT$_G$(R&P).
- We find that our DRO variants outperform DeepHit on F_{CI} and F_{CG} metrics when using CI-based hyperparameter tuning. However, we find that our DRO variants cannot always achieve the best scores on the CI fairness metric when using F_{CG}-based hyperparameter tuning. We conclude that the CI metric may reflect fairness in the F_{CG} fairness metric but the reverse may not be true.
- It is hard to distinguish which method is better between DRO-DEEPHIT and DRO-DEEPHIT (SPLIT). For both methods, as expected, they have slightly lower performance than the DeepHit method on accuracy metrics. However, DRO-DEEPHIT method has the best C^{td} performance on the FLC dataset in Table 4.5.

Table 4.6: SODEN test set scores on the FLC, SUPPORT, SEER datasets when hyperparameter tuning is based on CI and F_{CG}.

Datasets	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
FLC (age)	SODEN	0.7785 (0.0175)	0.1482 (0.0138)	1.1790 (0.6098)	0.0004 (0.0009)	0.0002 (0.0003)	0.7785 (0.0175)	0.1482 (0.0138)	**1.1790 (0.6098)**	0.0004 (0.0009)	0.0002 (0.0003)
	$SODEN_G$(R&P)	0.7832 (0.0138)	0.1454 (0.0134)	0.8248 (0.6491)	0.0006 (0.0007)	0.0004 (0.0004)	0.7807 (0.0140)	0.1454 (0.0134)	1.7324 (1.2715)	0.0001 (0.0001)	**0.0000 (0.0000)**
	DRO-SODEN	**0.7857 (0.0124)**	**0.1434 (0.0141)**	**1.0401 (0.7724)**	**0.0000 (0.0000)**	**0.0000 (0.0000)**	0.7787 (0.0134)	0.1619 (0.0247)	1.4140 (0.7545)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
FLC (gender)	SODEN	0.7785 (0.0175)	0.1482 (0.0138)	1.3822 (0.6028)	**0.0004 (0.0009)**	0.0005 (0.0010)	0.7785 (0.0175)	0.1482 (0.0138)	1.3822 (0.6028)	0.0004 (0.0009)	0.0005 (0.0010)
	$SODEN_G$(R&P)	0.7824 (0.0126)	0.1496 (0.0117)	0.8252 (0.3665)	0.0005 (0.0006)	**0.0005 (0.0007)**	**0.7832 (0.0126)**	**0.1452 (0.0131)**	**1.0564 (0.4984)**	0.0001 (0.0001)	0.0001 (0.0001)
	DRO-SODEN	**0.7857 (0.0100)**	**0.1350 (0.0069)**	**0.7115 (0.3545)**	0.0008 (0.0023)	0.0008 (0.0022)	0.7811 (0.0131)	0.1592 (0.0258)	1.5226 (0.7068)	**0.0000 (0.0000)**	**0.0000 (0.0000)**
SUPPORT (age)	SODEN	0.6063 (0.0079)	**0.1960 (0.0012)**	2.5890 (0.3426)	0.0081 (0.0007)	0.0069 (0.0006)	0.6063 (0.0079)	**0.1960 (0.0012)**	2.5890 (0.3426)	0.0081 (0.0007)	0.0069 (0.0006)
	$SODEN_G$(R&P)	**0.6096 (0.0093)**	0.2118 (0.0126)	1.8446 (0.4881)	0.0052 (0.0014)	0.0048 (0.0010)	**0.6094 (0.0095)**	0.2128 (0.0094)	1.7577 (0.2762)	0.0043 (0.0007)	0.0042 (0.0007)
	DRO-SODEN	0.5802 (0.0148)	0.2054 (0.0094)	**1.5585 (0.6516)**	**0.0035 (0.0022)**	**0.0034 (0.0017)**	0.5782 (0.0147)	0.2082 (0.0088)	**1.5376 (0.6779)**	**0.0026 (0.0019)**	**0.0027 (0.0014)**
SUPPORT (gender)	SODEN	0.6063 (0.0079)	**0.1960 (0.0012)**	1.7492 (0.1505)	0.0081 (0.0007)	0.0081 (0.0007)	0.6063 (0.0079)	**0.1960 (0.0012)**	1.7492 (0.1505)	0.0081 (0.0007)	0.0081 (0.0007)
	$SODEN_G$(R&P)	**0.6066 (0.0084)**	0.2002 (0.0089)	1.5665 (0.2379)	0.0069 (0.0018)	0.0071 (0.0017)	**0.6094 (0.0074)**	0.2156 (0.0106)	**1.2016 (0.1398)**	0.0043 (0.0007)	0.0046 (0.0007)
	DRO-SODEN	0.5898 (0.0207)	0.2034 (0.0089)	**1.5474 (0.2314)**	**0.0049 (0.0033)**	**0.0050 (0.0031)**	0.5782 (0.0147)	0.2082 (0.0088)	1.5092 (0.2431)	**0.0026 (0.0019)**	**0.0029 (0.0017)**
SUPPORT (race)	SODEN	0.6063 (0.0079)	**0.1960 (0.0012)**	1.6812 (0.1965)	0.0081 (0.0007)	0.0097 (0.0009)	0.6063 (0.0079)	**0.1960 (0.0012)**	1.6812 (0.1965)	0.0081 (0.0007)	0.0097 (0.0009)
	$SODEN_G$(R&P)	**0.6084 (0.0064)**	0.2058 (0.0124)	1.6585 (0.2304)	0.0054 (0.0016)	0.0069 (0.0017)	**0.6094 (0.0074)**	0.2156 (0.0106)	1.7536 (0.4094)	0.0043 (0.0007)	0.0056 (0.0008)
	DRO-SODEN	0.5964 (0.0123)	0.1968 (0.0026)	**1.4009 (0.3518)**	**0.0059 (0.0015)**	**0.0072 (0.0018)**	0.5782 (0.0147)	0.2082 (0.0088)	**0.7343 (0.5887)**	**0.0026 (0.0019)**	**0.0036 (0.0021)**
SEER (age)	SODEN	**0.7132 (0.0017)**	**0.1550 (0.0009)**	**0.8531 (0.0940)**	0.0280 (0.0014)	0.0141 (0.0011)	**0.7132 (0.0017)**	**0.1550 (0.0009)**	**0.8531 (0.0940)**	0.0280 (0.0014)	0.0141 (0.0011)
	$SODEN_G$(R&P)	0.7131 (0.0017)	0.1556 (0.0011)	0.8541 (0.1562)	0.0277 (0.0011)	**0.0140 (0.0010)**	0.7122 (0.0009)	0.1561 (0.0012)	0.9110 (0.0948)	0.0276 (0.0013)	0.0134 (0.0008)
	DRO-SODEN	0.7026 (0.0116)	0.1757 (0.0293)	1.1275 (0.3644)	**0.0227 (0.0054)**	0.0142 (0.0014)	0.6980 (0.0108)	0.2008 (0.0367)	1.4280 (0.5242)	**0.0161 (0.0095)**	**0.0103 (0.0043)**
SEER (race)	SODEN	**0.7132 (0.0017)**	**0.1550 (0.0009)**	2.4948 (0.1341)	0.0280 (0.0014)	0.0399 (0.0019)	**0.7132 (0.0017)**	**0.1550 (0.0009)**	2.4948 (0.1341)	0.0280 (0.0014)	0.0399 (0.0019)
	$SODEN_G$(R&P)	0.7124 (0.0016)	0.1558 (0.0011)	2.4390 (0.1775)	0.0273 (0.0013)	0.0386 (0.0017)	0.7123 (0.0016)	0.1561 (0.0013)	2.4747 (0.1869)	0.0271 (0.0013)	0.0382 (0.0018)
	DRO-SODEN	0.6913 (0.0109)	0.2079 (0.0373)	**1.6398 (0.4948)**	**0.0167 (0.0057)**	**0.0265 (0.0070)**	0.6893 (0.0055)	0.2191 (0.0269)	**1.7676 (0.4458)**	**0.0126 (0.0059)**	**0.0204 (0.0092)**

SODEN We conduct experiments to compare the accuracy and fairness of SODEN and $SODEN_G$(R&P) to DRO-SODEN. Our experimental results are reported in Table 4.6 (CI-based hyperparameter tuning and F_{CG}-based hyperparameter tuning). From these results, we have the following observations:

- When tuning hyperparameters based on CI, it is clear that DRO-SODEN outperforms the other methods on the CI fairness metric for FLC and SUPPORT datasets. Meanwhile, F_{CI} and F_{CG} are also reduced by using DRO-SODEN while accuracy scores become a little lower than those of SODEN. However, we find DRO-SODEN can achieve a slightly higher C^{td} scores on the FLC dataset.
- When tuning hyperparameters based on F_{CG}, we find DRO-SODEN also can achieve better performance on F_{CG} than the corresponding values that are from the CI-based tuning since we tune hyperparameters based on this metric. In addition, DRO-SODEN can obtain the best F_{CG} and F_{CI} scores compared to the baselines.

4.6 Accuracy Fairness Tradeoff Comparison Across DRO Variants of Different Survival Models

We can compare the tradeoff between accuracy and tradeoff across different DRO variants. Specifically, for the DEEP DRO-COX, DEEP DRO-COX (SPLIT), DRO-DEEPHIT, DRO-DEEPHIT (SPLIT), and DRO-SODEN models, we test them under the nonlinear setting on FLC, SUPPORT, and SEER datasets. We evaluate the F_{CI}, F_{CG}, and C^{td} scores of all methods using different values of α from 0.1 to 1.0 and then plot accuracy vs fairness curves for each dataset, as shown in Figure 2. Note that in each plot, being closer to the lower right is considered better, corresponding to a model having an α value that achieves as low of a fairness metric score (either F_{CI} or F_{CG}) as possible (which is considered more fair) and as high of a C^{td} score as possible. From the figure, we find that the DRO-DEEPHIT method outperforms the other methods.

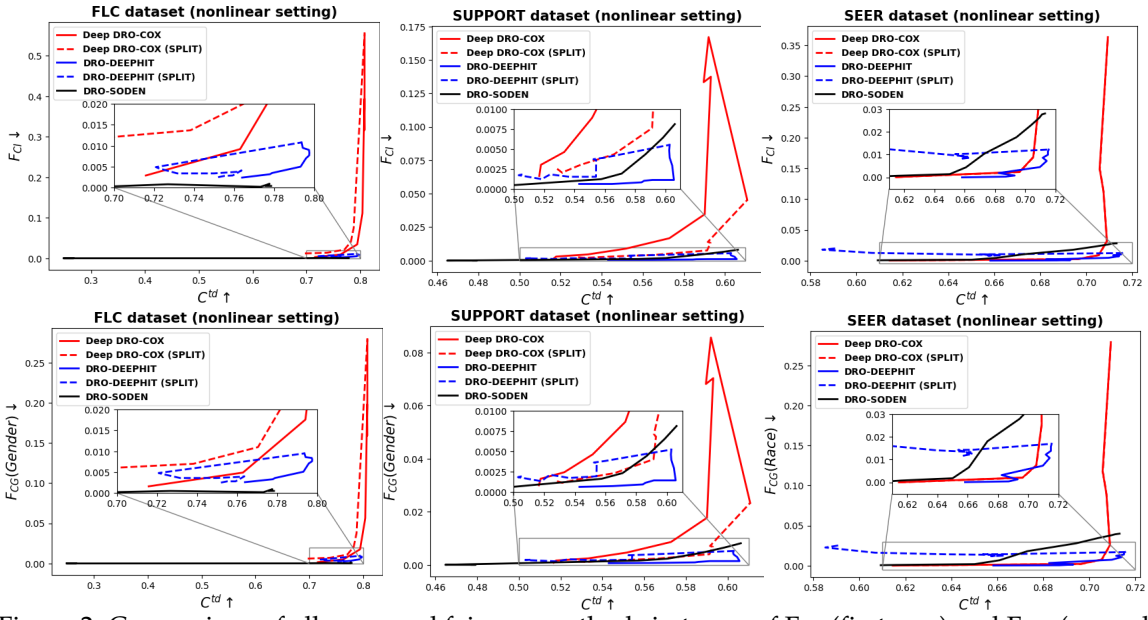


Figure 2: Comparison of all proposed fairness methods in terms of F_{CI} (first row) and F_{CG} (second row) with C^{td} on FLC, SUPPORT, and SEER2 datasets. Each line is drawn based on the various values of α (from 0.1 to 1.0). In each subfigure, the closer the curve is to the lower right corner, the better the performance.

5 Conclusion

We have shown a general strategy for converting a wide class of survival models into DRO variants that encourage fairness. The key idea is to write the overall loss in terms of individual losses, which in turn could be used in a DRO framework. When there is coupling so that an individual loss technically is not "individual" as it depends on multiple data points, we introduced a sample splitting approach that is compliant with DRO theory at the expense of approximating the loss that we aim to minimize. We also showed that the heuristic approach that ignores this coupling problem and naively runs an existing DRO algorithm (that assumes that there is no coupling) works in practice about as well as the sample splitting version. When a survival model used does not have this coupling issue (such as SODEN), then existing DRO machinery directly works; there is no need to use any sample splitting.

There are a number of open questions that remain. When the coupling issue arises, it would be interesting to determine if a theoretically sound DRO variant can be derived that does not require sample splitting. Next, in using DRO, tuning the subpopulation probability threshold α can significantly impact the results. In our experiments, we tuned α using one of two different fairness metrics (CI or F_{CG}) on a validation set. While DRO (whether heuristic or using sample splitting) itself does not require the user to specify which features to treat as sensitive in the training loss, we are effectively using some information about which features to treat as sensitive as it shows up in computing the validation set fairness metric. A open question thus arises of whether we could tune α in some other way in practice that either does not use a validation set or which does not require using a validation set fairness metric that knows which features to treat as sensitive. Lastly, we point out that it would be interesting to empirically study how converting a survival model into its DRO variant impacts other metrics aside from the accuracy or fairness metrics we considered, such as calibration metrics [Haider et al., 2020, Goldstein et al., 2020]. Ultimately, we suspect that DRO variants of survival models potentially have interesting properties that make them useful beyond encouraging fairness.

Acknowledgments This work was supported by NSF CAREER award #2047981. The authors would like to thank Tatsunori Hashimoto and the anonymous reviewers for very helpful feedback.

A More Details on Cox Models

A.1 Estimating the Baseline Hazard and Conditional Survival Function

After learning the log partial hazard function $f(\cdot;\theta)$ (or, equivalently, learning the parameters θ), a standard approach to estimating the baseline hazard function h_0 is to use the so-called Breslow method [Breslow, 1972]. In what follows, we use $\widehat{\theta}$ to denote the learned estimate of θ.

The Breslow method estimates a discretized version of h_0. Specifically, let $t_1 < t_2 < \cdots < t_m$ denote the unique times when critical event happened in the training data. Let d_j denote the number of critical events that occurred at time t_j for $j \in [m]$. Then we compute the following estimate of h_0 at the j-th time step:

$$\widehat{h}_{0,j} \triangleq \frac{d_j}{\sum_{i\in[n] \text{ s.t. } Y_i \geq t_j} \exp(f(x_i;\widehat{\theta}))} \qquad \text{for } j \in [m].$$

After estimating the baseline hazard function, estimating the survival function is straightforward. Recall that $S(t|x) = \exp\big(-\int_0^t h(u|x)\mathrm{d}u\big)$. Then combining this equation with the proportional hazards assumption (i.e., the factorization in equation (2.3)), we get

$$S(t|x) = \exp\Big(-\int_0^t h_0(u)\exp(f(x;\theta))\mathrm{d}u\Big) = \exp\Big(\Big[\underbrace{-\int_0^t h_0(u)\mathrm{d}u}_{\text{abbreviate as } H_0(t)}\Big]\exp(f(x;\theta))\Big). \tag{A.1}$$

We can estimate $H_0(t)$ via a summation in place of an integration:

$$\widehat{H}_0(t) \triangleq \sum_{j\in[m] \text{ s.t. } t_j \leq t} \widehat{h}_{0,j} \qquad \text{for } t \geq 0.$$

Thus, by plugging in $\widehat{H}_0$ in place of H_0 and $\widehat{\theta}$ in place of θ in equation (A.1), we obtain the conditional survival function estimate $\widehat{S}(t|x) \triangleq \exp(-\widehat{H}_0(t)\exp(f(x;\widehat{\theta})))$.

A.2 The Proportional Hazards Assumption and the Shape of the Conditional Survival Function

The proportional hazards assumption constrains the shape of the conditional survival function. Recall that for any two real numbers $a, b \in \mathbb{R}$, we have $\exp(a \cdot b) = (\exp(a))^b$. Then equation (A.1) (which was derived using the proportional hazard assumption) is equal to

$$S(t|x) = \exp\big(H_0(t)\exp(f(x;\theta))\big) = [\underbrace{\exp(H_0(t))}_{\triangleq S_0(t)}]^{\exp(f(x;\theta))}.$$

In other words, under the proportional hazards assumption, the conditional survival function $S(\cdot|x)$ must necessarily be a power of the so-called baseline survival function $S_0(\cdot)$.

B Fairness Metrics

In this paper, we use the individual, group, and intersectional fairness metrics defined by Keya et al. [2021], the concordance imparity (CI) metric by Zhang and Weiss [2022], and also censoring-based individual and censoring-based group fairness metrics by Rahman and Purushotham [2022]. In

what follows, since we are focusing on Cox proportional hazards models, we can take the predicted outcome for a feature vector x to be the so-called *partial hazard* $\widetilde{h}(x) \triangleq \exp(f(x;\theta))$; this is the same as the hazard function given in equation (2.3) except where we exclude the baseline hazard factor $h_0(t)$. Note that once we exclude $h_0(t)$, then $\widetilde{h}$ no longer depends on time t. We state the fairness metrics in terms of a collection of N_{test} test patients with data $(X_1^{\text{test}}, Y_1^{\text{test}}, \delta_1^{\text{test}}), \ldots, (X_{N_{\text{test}}}^{\text{test}}, Y_{N_{\text{test}}}^{\text{test}}, \delta_{N_{\text{test}}}^{\text{test}})$. Note that the fairness metrics by Keya et al. [2021] only use the test feature vectors $X_1^{\text{test}}, \ldots, X_{N_{\text{test}}}^{\text{test}}$ and ignores the test patients' observed times and event indicators. Also, at the end of this section, we point out that the individual and group fairness metrics by Keya et al. [2021] are sensitive to the scale of the log partial hazard $f(\cdot;\theta)$.

Individual fairness Roughly, Keya et al. [2021] consider a model to be fair across individuals (patients) if similar individuals have similar predicted outcomes. To operationalize this notion of fairness in the context of Cox models, Keya et al. define the individual fairness metric

$$\mathrm{F}_I \triangleq \sum_{i=1}^{N_{\text{test}}} \sum_{j=i+1}^{N_{\text{test}}} \left[|\widetilde{h}(X_i^{\text{test}}) - \widetilde{h}(X_j^{\text{test}})| - \gamma \|X_i^{\text{test}} - X_j^{\text{test}}\| \right]_+,$$

where γ is a predefined scale factor (0.01 in our experiments). As a reminder, $[\,\cdot\,]_+$ is the ReLU function (so that $[a]_+ = \max\{0, a\}$ for any $a \in \mathbb{R}$).

Note that this individual fairness metric is actually just penalizing $\widetilde{h}$ for not being Lipschitz continuous (as empirically evaluated over the test data). Specifically, $\widetilde{h}$ is defined to be γ-Lipschitz continuous if

$$|\widetilde{h}(x) - \widetilde{h}(x')| \leq \gamma \|x - x'\| \quad \text{for all } x, x' \in \mathcal{X}.$$

Meanwhile, when F_I is equal to 0, then it means that

$$|\widetilde{h}(X_i^{\text{test}}) - \widetilde{h}(X_j^{\text{test}})| \leq \gamma \|X_i^{\text{test}} - X_j^{\text{test}}\| \quad \text{for all } i, j \in \{1, \ldots, N_{\text{test}}\}.$$

As a technical remark, in the definition of F_I and also γ-Lipschitz continuity, the metric used to measure distances between feature vectors does not have to be Euclidean. For example, we can replace $\|X_i^{\text{test}} - X_j^{\text{test}}\|$ with $\rho(X_i^{\text{test}}, X_j^{\text{test}})$, where $\rho : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ is a user-specified metric.

Group fairness Next, Keya et al. [2021] consider a model is fair across a user-specified set of groups if these different groups have similar predicted outcomes. Keya et al. define the group fairness metric F_G to look at the maximum deviation of a group's average predicted outcome to the overall population's average predicted outcome. Specifically, let $\mathcal{G}$ be the user-specified set of groups to consider (for example, there could be two groups: everyone with age at most 65 years, and everyone older than 65 years), where each group $g \in \mathcal{G}$ is a subset of the test set indices $\{1, \ldots, N_{\text{test}}\}$ (so that using this notation, group g has size $|g|$); the different groups should form a partition of the test set (so that the groups are disjoint and their union is the entire test set). Then

$$\mathrm{F}_G \triangleq \max_{g \in \mathcal{G}} \left| \underbrace{\frac{1}{|g|} \sum_{i \in g} \widetilde{h}(X_i^{\text{test}})}_{\substack{\text{average predicted} \\ \text{outcome of group } g}} - \underbrace{\frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \widetilde{h}(X_i^{\text{test}})}_{\substack{\text{average predicted} \\ \text{outcome of population}}} \right|.$$

Intersectional fairness Keya et al. [2021] consider a notion of intersectional fairness that accounts for multiple sensitive attributes. For example, in the FLC dataset, we have 2 different sensitive attributes, age and gender. For each of these sensitive attributes, we can partition the test set into groups. Specifically, let $\mathcal{G}_1$ be a partition of the test set into different age groups (for example, two groups: at most 65 years old and over 65 years old), and let $\mathcal{G}_2$ be a partition of the test set into different gender groups (for example, two groups: female and male). Then intersectional

fairness looks at every intersection of age/gender groups (continuing from the previous examples, we would have four intersectional subgroups: at most 65 years old and female, at most 65 years and male, over 65 years old and female, over 65 years old and male).

The notation here is a bit more involved. The set of all intersectional subgroups of $\mathcal{G}_1$ and $\mathcal{G}_2$ is given by the Cartesian product $\mathcal{G}_1 \times \mathcal{G}_2$. Note that $s \in \mathcal{G}_1 \times \mathcal{G}_2$ means that $s = (s_1, s_2)$, where $s_1 \in \mathcal{G}_1$ and $s_2 \in \mathcal{G}_2$. More generally, if there are J sensitive attributes, corresponding to groupings $\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_J$, then the set of all intersectional subgroups would be $\mathcal{S} \triangleq \mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \mathcal{G}_J$. Now $s \in \mathcal{S}$ is a list consisting of J different subsets of test patients (i.e., $s = (s_1, s_2, \ldots, s_J)$, where $s_1 \in \mathcal{G}_1$, $\ldots$, $s_J \in \mathcal{G}_J$). The intersection of these J subsets (i.e., $\cap_{j=1}^{J} s_j \subset \{1, \ldots, N_{\text{test}}\}$) is precisely the set of test patients that intersectional subgroup s corresponds to. Then the average predicted outcome for intersectional subgroup s is

$$\widetilde{\mathbf{h}}(s) \triangleq \frac{1}{|\cap_{j=1}^{J} s_j|} \sum_{i \in \cap_{j=1}^{J} s_j} \widetilde{h}(X_i^{\text{test}}).$$

Then the intersection fairness metric $\mathrm{F}_\cap$ by Keya et al. [2021] is the worst-case log ratio of expected predicted outcomes between two intersectional subgroups:

$$\mathrm{F}_\cap \triangleq \max_{s,s' \in \mathcal{S}} \left| \log \frac{\widetilde{\mathbf{h}}(s)}{\widetilde{\mathbf{h}}(s')} \right|.$$

Concordance imparity We now describe an alternative metric for group fairness called concordance imparity (CI) that asks that a survival analysis model achieves similar prediction accuracy for different groups. For ease of exposition, we only state the CI metric by Zhang and Weiss [2022] in terms of a single sensitive attribute that has already been discretized (e.g., the attribute is already discrete or we have a pre-specified discretization rule); this special case is sufficient for our experiments. We denote the set of possible discretized values of this sensitive attribute as $\mathcal{A}$. For example, $\mathcal{A}$ could correspond to age and we could have $\mathcal{A} = \{\text{“age} \leq 65\text{”}, \text{“age} > 65\text{”}\}$, i.e., $\mathcal{A}$ consists of the different groups to consider. We refer the reader to the Zhang and Weiss's original paper for their more general definition of CI that can handle a continuous sensitive attribute via an automatic discretization strategy that they propose.

Assuming that the sensitive attribute has already been discretized into the set $\mathcal{A}$, the CI metric looks at a variant of the standard survival analysis accuracy metric of concordance index [Harrell et al., 1982] that Zhang and Weiss call the *concordance fraction* (CF), which is specific to each sensitive attribute value $a \in \mathcal{A}$. The CI metric is then defined to be the worst-case difference between the CF scores of any two $a, a' \in \mathcal{A}$ where $a \neq a'$. The pseudocode can be found in Algorithm 3; note that to keep the notation from getting clunky, we drop the superscript “test” from the test feature vectors, observed times, and event indicators in the pseudocode but we still use N_{test} to denote the number of test patients. Also, in the pseudocode, we let $A_i \in \mathcal{A}$ denote the sensitive attribute value for the i-th test patient, where we assume that A_i can directly be computed based on the i-th test patient's feature vector. For example, when age (which is not discretized) is one of the features and $\mathcal{A}$ consists of the two age groups previously stated (≤ 65 or > 65), then since we know the discretization rule used, we can readily determine which age group in $\mathcal{A}$ that any test patient is in.

Importantly, to calculate the CI metric, a way to calculate a risk score is required to compute the CF scores. How to define a risk score differs across models. For Cox models, we can take the risk score to be the log partial hazard function $f(\cdot; \theta)$. For DeepHit and SODEN models, we take the risk score to be the estimated survival probability $\widehat{S}(t|x)$ and therefore we need to replace $f(\cdot; \theta)$ with $\widehat{S}(t|x)$ before using Algorithm 3. Since different values of time t can have different estimated $\widehat{S}(t|x)$ values, we would obtain different value of the CI fairness metric for different t. We test three different values of t (the 25^{th}, 50^{th}, and 75^{th} percentile of the observed times in the test data) and use the average value for the final CI score.

Censoring-based individual fairness Individual fairness F_I does not consider censoring information that is available. Rahman and Purushotham [2022] defined a censoring-based individual fairness metric as follows:

$$\mathrm{F}_{CI} \triangleq \frac{1}{|N_c| \times |N_{uc}|} \sum_{\substack{i \in N_c, j \in N_{uc} \\ \text{s.t. } Y_j \geq Y_i}} \left[|\widetilde{h}(X_i^{\text{test}}) - \widetilde{h}(X_j^{\text{test}})| - \gamma \|X_i^{\text{test}} - X_j^{\text{test}}\| \right]_+,$$

where N_c and N_{uc} are the index sets of censored and uncensored data. Similar to in F_I, the scale factor γ is a predefined (0.01 in our experiments). This fairness metric ensures that censored patient and uncensored patients who have similar features should also have similar predictions whenever the observed time from the uncensored patient is larger than that of the censored patient.

When using DeepHit and SODEN models, since these do not use a proportional hazards assumption, we slightly modify the above definition and instead use

$$\mathrm{F}_{CI}(t) \triangleq \frac{1}{|N_c| \times |N_{uc}|} \sum_{\substack{i \in N_c, j \in N_{uc} \\ \text{s.t. } Y_j \geq Y_i}} \left[|\widehat{S}(t|X_i^{\text{test}}) - \widehat{S}(t|X_j^{\text{test}})| - \gamma \|X_i^{\text{test}} - X_j^{\text{test}}\| \right]_+,$$

where $\widehat{S}(t|X)$ is the estimated survival probability at time t for patient X. Similar to the CI fairness metric and following the settings in Rahman and Purushotham [2022], we test three different t values (25^{th}, 50^{th}, 75^{th} percentile of the observed times in the test data) and use their average value to calculate the final F_{CI} score.

Censoring-based group fairness Rahman and Purushotham [2022] also modified the F_G metric by Keya et al. [2021] to account for censoring information. Reusing notation from the definition of F_G, we now define the censoring-based group fairness metric

$$\mathrm{F}_{CG} \triangleq \frac{1}{|N_c| \times |N_{uc}|} \sum_{g \in \mathcal{G}} \sum_{\substack{i \in N_{c,g}, j \in N_{uc,g} \\ \text{s.t. } Y_j \geq Y_i}} \left[|\widetilde{h}(X_i^{\text{test}}) - \widetilde{h}(X_j^{\text{test}})| - \gamma \|X_i^{\text{test}} - X_j^{\text{test}}\| \right]_+,$$

where $N_{c,g}$ and $N_{uc,g}$ are again the index sets of censored and uncensored in group g, respectively.

Furthermore, we replace $\widetilde{h}(x)$ with $\widehat{S}(t|x)$ to define a censoring-based group fairness metric for DeepHit and SODEN models. Specifically, we have

$$\mathrm{F}_{CG}(t) \triangleq \frac{1}{|N_c| \times |N_{uc}|} \sum_{g \in \mathcal{G}} \sum_{\substack{i \in N_{c,g}, j \in N_{uc,g} \\ \text{s.t. } Y_j \geq Y_i}} \left[|\widehat{S}(t|X_i^{\text{test}}) - \widehat{S}(t|X_j^{\text{test}})| - \gamma \|X_i^{\text{test}} - X_j^{\text{test}}\| \right]_+.$$

Similar to the setting in censoring-based individual fairness, we use three different t to test the value of F_{CG} and use their average performance for the final reported score.

Scale Issues with F_I, F_G, F_{CI}, and F_{CG}

We point out that the F_I, F_G, F_{CI}, and F_{CG} fairness metrics are sensitive to the scale of the log partial hazard function $f(\cdot;\theta)$, and thus also the scale of the partial hazard $\widetilde{h}(x) = \exp(f(x;\theta))$ if they are calculated by using $\widetilde{h}(x)$. For instance, consider a standard linear Cox model with $f(x;\theta) = \theta^T x$, where the parameters θ have already been learned. Then one way to make the model appear fairer according to the F_I, F_G, F_{CI}, and F_{CG} metrics is to just scale all values in θ by any positive constant smaller than 1; doing so, the standard accuracy metric of concordance index [Harrell et al., 1982] would actually remain unchanged for the model as it only depends on the ranking of the different individuals' (log) partial hazard values. However, an accuracy score that considers each individual's survival function estimate (e.g., integrated IPCW Brier Score [Graf et al., 1999]) would be affected.

Algorithm 3: Concordance Imparity (CI) with a discrete sensitive attribute

Input: Test dataset $\{(X_i, Y_i, \delta_i)\}_{i=1}^{N_{\text{test}}}$, risk score $f(\cdot;\theta)$ (from an already trained model), set of sensitive attribute values $\mathcal{A}$ (so that each $a \in \mathcal{A}$ corresponds to a different group), $A_1, \ldots, A_{N_{\text{test}}} \in \mathcal{A}$ says which sensitive attribute value each test patient has

Output: CI score

for $a \in \mathcal{A}$ **do**
  Initialize the numerator count $\mathbf{N}(a) \leftarrow 0$ and denominator count $\mathbf{D}(a) \leftarrow 0$.
end
for $i = 1, \ldots, N_{test}$ **do**
  for $j = 1, \ldots, N_{test}$ *s.t.* $j \neq i$ **do**
    if *(*$Y_i < Y_j$ *and* $\delta_i == 0$*) or (*$Y_j < Y_i$ *and* $\delta_j == 0$*) or (*$Y_i == Y_j$ *and* $\delta_i == 0$ *and* $\delta_j == 0$*)* **then**
      continue
    else
      Set $\mathbf{D}(A_i) \leftarrow \mathbf{D}(A_i) + 1$.
    end
    if $Y_i < Y_j$ **then**
      if $f(X_i;\theta) > f(X_j;\theta)$ **then**
        Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 1$.
      else if $f(X_i;\theta) == f(X_j;\theta)$ **then**
        Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 0.5$.
      end
    else if $Y_i > Y_j$ **then**
      if $f(X_i;\theta) < f(X_j;\theta)$ **then**
        Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 1$.
      else if $f(X_i;\theta) == f(X_j;\theta)$ **then**
        Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 0.5$.
      end
    else if $Y_i == Y_j$ **then**
      if $\delta_i == 1$ *and* $\delta_j == 1$ **then**
        if $f(X_i;\theta) == f(X_j;\theta)$ **then**
          Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 1$.
        else
          Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 0.5$.
        end
      else if $\delta_i == 0$ *and* $\delta_j == 1$ *and* $f(X_i;\theta) < f(X_j;\theta)$ **then**
        Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 1$.
      else if $\delta_i == 1$ *and* $\delta_j == 0$ *and* $f(X_i;\theta) > f(X_j;\theta)$ **then**
        Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 1$.
      else
        Set $\mathbf{N}(A_i) \leftarrow \mathbf{N}(A_i) + 0.5$.
      end
    end
  end
end
for $a \in \mathcal{A}$ **do**
  Set the concordance fraction of a: $\mathbf{CF}(a) \leftarrow \frac{\mathbf{N}(a)}{\mathbf{D}(a)}$.
end
return CI $\leftarrow \max_{a,a' \in \mathcal{A} \text{ s.t. } a \neq a'} |\mathbf{CF}(a) - \mathbf{CF}(a')|$

C Hyperparameter Tuning and Compute Environment Details

Hyperparameters *Cox models*: for nonlinear Cox models, we always use a two-layer MLP with ReLU as the activation function and 24 as the number of hidden units. All models (linear and nonlinear) are trained using Adam [Kingma and Ba, 2014] in PyTorch 1.7.1 in a batch setting for 500 iterations, only using a CPU and no GPU.

DeepHit models: we use three-layer MLP with ReLU activation, batch normalization, and dropout (in 0.1). The number of hidden units is 32. The original DeepHit and DRO-DEEPHIT models are trained using Adam in PyTorch 1.7.1 in a mini-batch 256 setting for 500 epochs. However, the DRO-DEEPHIT (SPLIT) model is trained using a batch setting for 500 iterations.

SODEN models: for the FLC dataset, we use an MLP with 4 layers and 16 hidden units. For SUPPORT and SEER datasets, we use an MLP with 2 layers and 26 hidden units. In addition, RMSprop [Tieleman and Hinton, 2012] in 128 batch size with a maximum 100 epochs is used to train all models.

We tune on the following hyperparameter grid:

- To find the optimal learning rate for each Cox model, we conducted a sweep over values of 0.01, 0.001, and 0.0001. For DeepHit models, a fixed learning rate of 0.01 was used. In the case of SODEN models, the learning rates applied were 0.01, 0.002, and 0.002 for the FLC, SUPPORT, and SEER datasets, respectively.
- λ (only used for baselines; a hyperparameter that controls the tradeoff between the original Cox loss and fairness regularization term): 1, 0.7, 0.4
- α: 0.1, 0.15, 0.2, 0.3, 0.4, 0.5 for DRO-COX/DRO-COX (SPLIT) variants. 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 for DRO-DEEPHIT/DRO-DEEPHIT (SPLIT) variants and DRO-SODEN.

In addition, for DRO-COX (SPLIT) and DRO-DEEPHIT (SPLIT), we choose $n_1 = n_2 = n/2$ (rounding as needed when n is odd, so that n_1 might not equal n_2).

Compute environment All models are implemented with Python 3.8.3, and they are trained and tested on identical compute instances, each with an Intel Core i9-10900K CPU (3.70GHz with 64 GB RAM) and a Quadro RTX 4000 GPU.

D Additional Experiments

Using other sensitive attributes in evaluating CI and F_{CG} in Cox models In the main paper, we only showed test set performance metrics for FLC, SUPPORT, and SEER using age, gender, race, and race respectively in evaluating F_{CG}, and CI in Cox model settings. We now provide results using gender for FLC (Table D.1), age and separately race for SUPPORT (Tables D.2 and D.3), and age for SEER (Table D.4). Our main findings still hold for these additional results.

Effect of changing n_1 (or n_2) for DRO-COX (SPLIT) In the above experiments, we set $n_1 = n_2 = n/2$ (rounding as needed). To evaluate the sensitivity of this setting, we test the model performance using DRO-COX (SPLIT) under the linear and nonlinear settings, where we set $n_2 = 0.1n, 0.2n, 0.3n, 0.4n, 0.5n$ (corresponding to $n_1 = 0.9n, 0.8n, 0.7n, 0.6n, 0.5n$). We report the test set performance metrics for the FLC dataset (using age for evaluation) in Table D.5. From the table, we find that per metric, different settings for n_1 and n_2 lead to results that, while slightly different, are not dramatically different, i.e., the performance of DRO-COX (SPLIT) does not appear very sensitive w.r.t. the choice of n_1 and n_2.

The effect of using two losses for DRO-COX (SPLIT) rather than only one Recall that DRO-COX (SPLIT) minimizes the sum of two losses $L^{\text{split}}_{\text{DRO}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$ and $L^{\text{split}}_{\text{DRO}}(\theta, \eta, \mathcal{D}_2 \mid \mathcal{D}_1)$. Towards the end of Section 3.2, we said that an approach that only minimizes one of these losses would not use the data as effectively compared to minimizing the sum of these losses. We conducted an experiment to verify this claim, where we refer to the version of DRO-COX (SPLIT)

Table D.1: Cox model test set scores on the FLC (gender) dataset, in the same format as Table 4.2.

	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	Cox	**0.8032**	0.1739	0.8610	0.3608	0.1809	**0.8032**	0.1739	0.8610	0.3608	0.1809
		(0.0002)	(0.0004)	(0.0197)	(0.0045)	(0.0023)	**(0.0002)**	(0.0004)	(0.0197)	(0.0045)	(0.0023)
	Cox_I(Keya et al.)	0.7932	0.1368	1.6750	0.0786	0.0401	0.7859	0.1334	1.0098	0.0546	0.0277
		(0.0083)	(0.0052)	(0.7969)	(0.0266)	(0.0138)	(0.0220)	(0.0034)	(1.3509)	(0.0050)	(0.0027)
	Cox_I(R&P)	0.8028	0.1588	0.8050	0.1888	0.0954	0.8012	0.1563	0.4385	0.1680	0.0850
		(0.0012)	(0.0029)	(0.0978)	(0.0249)	(0.0127)	(0.0011)	(0.0006)	(0.2718)	(0.0040)	(0.0020)
	Cox_G(Keya et al.)	0.8011	0.1619	0.7020	0.2265	0.1141	0.8003	0.1567	0.4025	0.1699	0.0860
		(0.0015)	(0.0077)	(0.1081)	(0.0861)	(0.0427)	(0.0004)	(0.0004)	(0.2326)	(0.0022)	(0.0011)
	Cox_G(R&P)	0.8023	0.1646	0.7850	0.2410	0.1216	0.8011	0.1613	0.4604	0.2088	0.1056
		(0.0009)	(0.0026)	(0.0826)	(0.0248)	(0.0124)	(0.0004)	(0.0005)	(0.2519)	(0.0042)	(0.0021)
	$\text{Cox}_\cap$(Keya et al.)	0.7868	0.1400	0.4830	0.0616	0.0312	0.7868	0.1400	**0.2723**	0.0616	0.0312
		(0.0018)	(0.0005)	(0.1020)	(0.0015)	(0.0007)	(0.0018)	(0.0005)	**(0.2227)**	(0.0015)	(0.0007)
	DRO-COX	0.7605	**0.1350**	**0.3040**	0.0218	0.0112	0.7958	**0.1330**	0.5390	**0.0000**	**0.0000**
		(0.0096)	**(0.0003)**	**(0.1569)**	(0.0040)	(0.0020)	(0.0049)	**(0.0002)**	(0.5415)	**(0.0000)**	**(0.0000)**
	DRO-COX (SPLIT)	0.7964	0.1389	0.5060	**0.0000**	**0.0000**	0.7964	0.1389	0.5060	**0.0000**	**0.0000**
		(0.0045)	(0.0008)	(0.5152)	**(0.0000)**	**(0.0000)**	(0.0045)	(0.0008)	(0.5152)	**(0.0000)**	**(0.0000)**
Nonlinear	DeepSurv	**0.8070**	0.1767	1.0760	0.5788	0.2909	0.8070	0.1767	0.8274	0.5788	0.2909
		(0.0014)	(0.0018)	(0.1702)	(0.2493)	(0.1253)	(0.0014)	(0.0018)	(0.3277)	(0.2493)	(0.1253)
	DeepSurv_I(Keya et al.)	0.7916	0.1548	1.4610	0.0187	0.0095	0.7994	0.1673	0.7330	**0.0001**	**0.0000**
		(0.0121)	(0.0176)	(0.7342)	(0.0224)	(0.0113)	(0.0069)	(0.0051)	(0.9461)	**(0.0002)**	**(0.0001)**
	DeepSurv_I(R&P)	0.8067	0.1729	1.0640	0.0174	0.0088	0.8084	0.1757	0.5402	0.0004	0.0003
		(0.0041)	(0.0093)	(0.1408)	(0.0303)	(0.0150)	(0.0021)	(0.0029)	(0.5501)	(0.0002)	(0.0001)
	DeepSurv_G(Keya et al.)	0.7964	0.1576	0.9420	0.0962	0.0482	0.8017	0.1655	0.5413	0.0516	0.0260
		(0.0117)	(0.0196)	(0.2229)	(0.0862)	(0.0431)	(0.0114)	(0.0182)	(0.5104)	(0.0092)	(0.0046)
	DeepSurv_G(R&P)	0.8059	0.1699	1.0750	0.0386	0.0194	**0.8095**	0.1764	0.5816	0.0021	0.0011
		(0.0045)	(0.0118)	(0.1204)	(0.0438)	(0.0217)	**(0.0014)**	(0.0024)	(0.5826)	(0.0006)	(0.0003)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7804	0.1399	0.8440	0.0969	0.0497	0.7751	0.1357	**0.3897**	0.0394	0.0197
		(0.0119)	(0.0086)	(0.2581)	(0.1208)	(0.0622)	(0.0018)	(0.0002)	**(0.3535)**	(0.0010)	(0.0005)
	Deep DRO-COX	0.7699	**0.1336**	**0.4870**	0.0138	0.0071	0.7781	**0.1331**	0.4552	0.0054	0.0029
		(0.0147)	**(0.0004)**	**(0.2540)**	(0.0066)	(0.0033)	(0.0091)	**(0.0002)**	(0.4801)	(0.0012)	(0.0006)
	Deep DRO-COX (SPLIT)	0.7784	0.1647	0.5277	**0.0054**	**0.0029**	0.7784	0.1647	0.5277	0.0054	0.0029
		(0.0092)	(0.0037)	(0.5752)	**(0.0013)**	**(0.0007)**	(0.0092)	(0.0037)	(0.5752)	(0.0013)	(0.0007)

that only minimizes $L_{\text{DRO}}^{\text{split}}(\theta, \eta, \mathcal{D}_1 \mid \mathcal{D}_2)$ as DRO-COX (SPLIT, ONE SIDE). Specifically, we compare DRO-COX (SPLIT, ONE SIDE) and DRO-COX (SPLIT) under linear and nonlinear settings on the FLC dataset using age for evaluation. We report the resulting test set performance metrics in Table D.6. From the table, we find that DRO-COX (SPLIT) outperforms DRO-COX (SPLIT, ONE SIDE) on most metrics. This experimental finding supports our hypothesis that DRO-COX (SPLIT, ONE SIDE) uses data less effectively.

References

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

Norman Breslow. Discussion of the paper by David R Cox (1972), cited below. *Journal of the Royal Statistical Society, Series B*, 34:187–220, 1972.

Joy Buolamwini and Timnit Gebru. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency*, pages 77–91. PMLR, 2018.

George H Chen. Deep kernel survival analysis and subject-specific survival time prediction intervals. In *Machine Learning for Healthcare Conference*, pages 537–565. PMLR, 2020.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

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

Angela Dispenzieri, Jerry A Katzmann, Robert A Kyle, Dirk R Larson, Terry M Therneau, Colin L Colby, Raynell J Clark, Graham P Mead, Shaji Kumar, and L Joseph Melton III. Use of nonclonal serum immunoglobulin free light chains to predict overall survival in the general population. In *Mayo Clinic Proceedings*, pages 517–523. Elsevier, 2012.

Table D.2: Cox model test set scores on the SUPPORT (age) dataset, in the same format as Table 4.2.

	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	Cox	**0.6025**	0.2304	2.2240	0.0207	0.0090	**0.6025**	0.2304	2.2240	0.0207	0.0090
		(0.0005)	(0.0015)	(0.1078)	(0.0008)	(0.0005)	**(0.0005)**	(0.0015)	(0.1078)	(0.0008)	(0.0005)
	Cox_I(Keya et al.)	0.5820	**0.2153**	1.3120	**0.0007**	**0.0003**	0.5829	**0.2147**	**0.6800**	**0.0000**	**0.0000**
		(0.0116)	**(0.0076)**	(0.7623)	**(0.0018)**	**(0.0007)**	(0.0099)	**(0.0063)**	**(0.9091)**	**(0.0000)**	**(0.0000)**
	Cox_I(R&P)	0.6020	0.2285	2.1120	0.0182	0.0080	0.6018	0.2276	1.1235	0.0170	0.0075
		(0.0010)	(0.0014)	(0.2653)	(0.0013)	(0.0006)	(0.0015)	(0.0012)	(1.1232)	(0.0006)	(0.0002)
	Cox_G(Keya et al.)	0.5875	0.2315	2.2030	0.0184	0.0094	0.5862	0.2292	1.1115	0.0160	0.0082
		(0.0013)	(0.0014)	(0.0986)	(0.0007)	(0.0004)	(0.0009)	(0.0010)	(1.0976)	(0.0004)	(0.0002)
	Cox_G(R&P)	0.6018	0.2296	2.1210	0.0201	0.0086	0.6014	0.2291	1.1278	0.0195	0.0082
		(0.0008)	(0.0013)	(0.2863)	(0.0010)	(0.0005)	(0.0013)	(0.0012)	(1.1267)	(0.0007)	(0.0003)
	$\text{Cox}_\cap$(Keya et al.)	0.5664	0.2273	2.8030	0.0129	0.0066	0.5631	0.2264	1.4233	0.0117	0.0060
		(0.0061)	(0.0016)	(0.2551)	(0.0011)	(0.0006)	(0.0070)	(0.0017)	(1.4227)	(0.0012)	(0.0006)
	DRO-COX	0.5722	0.2210	1.8310	0.0025	0.0013	0.5641	0.2211	0.9255	0.0019	0.0010
		(0.0031)	(0.0010)	(0.2546)	(0.0009)	(0.0004)	(0.0105)	(0.0010)	(1.0171)	(0.0012)	(0.0006)
	DRO-COX (SPLIT)	0.5701	0.4569	**0.8631**	0.0023	0.0011	0.5701	0.4570	0.8616	0.0023	0.0011
		(0.0056)	(0.1314)	**(0.9061)**	(0.0011)	(0.0006)	(0.0056)	(0.1314)	(0.9042)	(0.0011)	(0.0006)
Nonlinear	DeepSurv	**0.6108**	0.2417	2.1170	0.0453	0.0212	**0.6108**	0.2417	2.1170	0.0453	0.0212
		(0.0029)	(0.0016)	(0.2107)	(0.0041)	(0.0020)	**(0.0029)**	(0.0016)	(0.2107)	(0.0041)	(0.0020)
	DeepSurv_I(Keya et al.)	0.5950	0.2316	1.6330	**0.0016**	**0.0007**	0.6031	0.2459	0.9475	0.0000	0.0000
		(0.0116)	(0.0188)	(0.5036)	**(0.0020)**	**(0.0009)**	(0.0059)	(0.0102)	(1.0522)	(0.0000)	(0.0000)
	DeepSurv_I(R&P)	0.6036	0.2323	2.1030	0.0075	0.0032	**0.6124**	0.2448	0.9620	**0.0000**	**0.0000**
		(0.0075)	(0.0083)	(0.2650)	(0.0061)	(0.0027)	**(0.0048)**	(0.0043)	(0.9920)	**(0.0000)**	**(0.0000)**
	DeepSurv_G(Keya et al.)	0.5869	0.2372	1.6760	0.0072	0.0037	0.5966	0.2543	0.9856	0.0001	0.0000
		(0.0122)	(0.0131)	(0.4326)	(0.0069)	(0.0036)	(0.0048)	(0.0032)	(1.0355)	(0.0002)	(0.0001)
	DeepSurv_G(R&P)	0.6039	0.2322	2.1660	0.0095	0.0040	0.6117	0.2440	1.0331	0.0002	0.0001
		(0.0089)	(0.0075)	(0.3318)	(0.0064)	(0.0027)	(0.0056)	(0.0043)	(1.0749)	(0.0001)	(0.0001)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.5979	0.2345	2.4300	0.0220	0.0113	0.5912	0.2309	1.2466	0.0182	0.0093
		(0.0063)	(0.0036)	(0.2338)	(0.0039)	(0.0020)	(0.0012)	(0.0011)	(1.2342)	(0.0008)	(0.0004)
	Deep DRO-COX	0.5833	**0.2231**	0.7590	0.0077	0.0038	0.5754	**0.2227**	**0.4152**	0.0065	0.0032
		(0.0088)	**(0.0015)**	(0.3395)	(0.0018)	(0.0009)	(0.0120)	**(0.0011)**	**(0.4798)**	(0.0023)	(0.0011)
	Deep DRO-COX (SPLIT)	0.5772	0.6387	**0.4364**	0.0069	0.0034	0.5772	0.6387	0.4364	0.0069	0.0034
		(0.0093)	(0.0007)	**(0.4875)**	(0.0021)	(0.0010)	(0.0093)	(0.0007)	(0.4875)	(0.0021)	(0.0010)

Table D.3: Cox model test set scores on the SUPPORT (race) dataset, in the same format as Table 4.2.

	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	Cox	**0.6025**	0.2304	1.4160	0.0207	0.0129	0.6025	0.2304	1.4160	0.0207	0.0129
		(0.0005)	(0.0015)	(0.0696)	(0.0008)	(0.0005)	(0.0005)	(0.0015)	(0.0696)	(0.0008)	(0.0005)
	Cox_I(Keya et al.)	0.5905	**0.2161**	1.1230	0.0030	0.0020	0.5829	**0.2147**	0.5910	**0.0000**	**0.0000**
		(0.0086)	**(0.0054)**	(0.6621)	(0.0031)	(0.0021)	(0.0099)	**(0.0063)**	(0.6975)	**(0.0000)**	**(0.0000)**
	Cox_I(R&P)	0.6024	0.2287	1.3320	0.0186	0.0116	0.6027	0.2276	0.6839	0.0169	0.0106
		(0.0010)	(0.0015)	(0.1742)	(0.0012)	(0.0007)	(0.0009)	(0.0011)	(0.6792)	(0.0005)	(0.0003)
	Cox_G(Keya et al.)	0.6013	0.2282	1.3610	0.0184	0.0115	0.6011	0.2279	0.6895	0.0181	0.0113
		(0.0008)	(0.0017)	(0.0647)	(0.0009)	(0.0006)	(0.0006)	(0.0009)	(0.6730)	(0.0003)	(0.0002)
	Cox_G(R&P)	0.6024	0.2294	1.3350	0.0194	0.0121	**0.6027**	0.2285	0.6856	0.0182	0.0114
		(0.0010)	(0.0013)	(0.1889)	(0.0010)	(0.0006)	**(0.0008)**	(0.0012)	(0.6810)	(0.0005)	(0.0003)
	$\text{Cox}_\cap$(Keya et al.)	0.5681	0.2271	1.4020	0.0127	0.0076	0.5631	0.2264	0.6893	0.0117	0.0070
		(0.0079)	(0.0018)	(0.1743)	(0.0013)	(0.0008)	(0.0070)	(0.0017)	(0.6849)	(0.0012)	(0.0007)
	DRO-COX	0.5735	0.2210	0.4640	0.0028	0.0017	0.5641	0.2211	0.3340	0.0019	0.0012
		(0.0018)	(0.0010)	(0.0790)	(0.0001)	(0.0001)	(0.0105)	(0.0010)	(0.4021)	(0.0012)	(0.0007)
	DRO-COX (SPLIT)	0.5701	0.4569	**0.3236**	**0.0023**	**0.0014**	0.5701	0.4570	**0.3231**	0.0023	0.0014
		(0.0056)	(0.1314)	**(0.3939)**	**(0.0011)**	**(0.0007)**	(0.0056)	(0.1314)	**(0.3938)**	(0.0011)	(0.0007)
Nonlinear	DeepSurv	**0.6108**	0.2417	1.7440	0.0453	0.0276	0.6108	0.2417	0.8946	0.0453	0.0276
		(0.0029)	(0.0016)	(0.2649)	(0.0041)	(0.0024)	(0.0029)	(0.0016)	(0.8698)	(0.0041)	(0.0024)
	DeepSurv_I(Keya et al.)	0.5927	0.2316	1.0380	**0.0014**	**0.0010**	0.6031	0.2459	0.6225	**0.0000**	**0.0000**
		(0.0082)	(0.0166)	(0.5996)	**(0.0017)**	**(0.0012)**	(0.0059)	(0.0102)	(0.8070)	**(0.0000)**	**(0.0000)**
	DeepSurv_I(R&P)	0.6078	0.2374	1.6470	0.0050	0.0031	0.6124	0.2448	0.8250	**0.0000**	**0.0000**
		(0.0067)	(0.0090)	(0.3917)	(0.0053)	(0.0033)	(0.0048)	(0.0043)	(0.8705)	**(0.0000)**	**(0.0000)**
	DeepSurv_G(Keya et al.)	0.5941	0.2369	1.2780	0.0104	0.0065	0.6056	0.2485	0.7256	0.0022	0.0014
		(0.0145)	(0.0117)	(0.3894)	(0.0089)	(0.0055)	(0.0044)	(0.0023)	(0.8039)	(0.0006)	(0.0003)
	DeepSurv_G(R&P)	0.6108	0.2396	1.5720	0.0046	0.0029	**0.6125**	0.2444	0.7651	0.0002	0.0001
		(0.0076)	(0.0086)	(0.2968)	(0.0057)	(0.0036)	**(0.0052)**	(0.0043)	(0.8150)	(0.0001)	(0.0000)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.5992	0.2357	1.4230	0.0236	0.0145	0.5912	0.2309	0.5886	0.0182	0.0113
		(0.0072)	(0.0042)	(0.4286)	(0.0054)	(0.0032)	(0.0012)	(0.0011)	(0.5782)	(0.0008)	(0.0005)
	Deep DRO-COX	0.5798	**0.2234**	0.7900	0.0092	0.0056	0.5754	**0.2227**	0.3602	0.0065	0.0040
		(0.0101)	**(0.0017)**	(0.4283)	(0.0043)	(0.0026)	(0.0120)	**(0.0011)**	(0.4571)	(0.0023)	(0.0014)
	Deep DRO-COX (SPLIT)	0.5772	0.6387	**0.3584**	0.0069	0.0042	0.5772	0.6387	**0.3584**	0.0069	0.0042
		(0.0093)	(0.0007)	**(0.4688)**	(0.0021)	(0.0013)	(0.0093)	(0.0007)	**(0.4688)**	(0.0021)	(0.0013)

Table D.4: Cox model test set scores on the SEER (age) dataset, in the same format as Table 4.2.

	Methods	CI-based Tuning					F_{CG}-based Tuning				
		Accuracy Metrics		Fairness Metrics			Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$	$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	Cox	**0.7025** (**0.0003**)	0.2128 (0.0009)	0.5555 (0.3870)	0.1690 (0.0053)	0.0468 (0.0009)	**0.7025** (**0.0003**)	0.2128 (0.0009)	0.5555 (0.3870)	0.1690 (0.0053	0.0468 (0.0009)
	Cox_I(Keya et al.)	0.6911 (0.0049)	0.1910 (0.0041)	0.3781 (0.4463)	0.0722 (0.0244)	0.0202 (0.0068)	0.6877 (0.0065)	**0.1838** (**0.0027**)	0.3841 (0.4965)	0.0112 (0.0016)	0.0022 (0.0007)
	Cox_I(R&P)	**0.7037** (**0.0006**)	0.2024 (0.0030)	0.5447 (0.4372)	0.1095 (0.0162)	0.0319 (0.0043)	0.7025 (0.0022)	0.1973 (0.0007)	0.5105 (0.4343)	0.0820 (0.0020)	0.0237 (0.0014)
	Cox_G(Keya et al.)	0.6517 (0.0023)	0.1986 (0.0005)	1.6654 (1.5826)	0.0838 (0.0015)	0.0431 (0.0008)	0.6517 (0.0023)	0.1986 (0.0005)	1.6654 (1.5826)	0.0838 (0.0015)	0.0431 (0.0008)
	Cox_G(R&P)	0.7028 (0.0007)	0.2086 (0.0037)	0.6010 (0.4591)	0.1450 (0.0193)	0.0382 (0.0043)	0.7007 (0.0023)	0.2043 (0.0012)	0.6109 (0.4911)	0.1229 (0.0066)	0.0283 (0.0007)
	$Cox_\cap$(Keya et al.)	0.6494 (0.0016)	0.1963 (0.0012)	1.1668 (1.0991)	0.0707 (0.0058)	0.0361 (0.0030)	0.6494 (0.0016)	0.1963 (0.0012)	1.1668 (1.0991)	0.0707 (0.0058)	0.0361 (0.0030)
	DRO-COX	0.6927 (0.0069)	**0.1868** (**0.0004**)	0.3170 (0.3762)	0.0001 (0.0001)	0.0000 (0.0000)	0.6927 (0.0069)	0.1868 (0.0004)	0.3170 (0.3762)	0.0001 (0.0001)	**0.0000** (**0.0000**)
	DRO-COX (SPLIT)	0.6872 (0.0047)	0.1869 (0.0004)	**0.2505** (**0.3292**)	**0.0000** (**0.0000**)	**0.0000** (**0.0000**)	0.6872 (0.0047)	0.1869 (0.0004)	**0.2505** (**0.3292**)	**0.0000** (**0.0000**)	**0.0000** (**0.0000**)
Nonlinear	DeepSurv	**0.7095** (**0.0014**)	0.2200 (0.0012)	0.6717 (0.3402)	0.3635 (0.1116)	0.1177 (0.0354)	**0.7095** (**0.0014**)	0.2200 (0.0012)	0.6717 (0.3402)	0.3635 (0.1116)	0.1177 (0.0354)
	$DeepSurv_I$(Keya et al.)	0.6985 (0.0041)	0.2123 (0.0035)	0.3850 (0.4457)	**0.0000** (**0.0000**)	**0.0000** (**0.0000**)	0.6982 (0.0045)	0.2127 (0.0032)	0.3820 (0.4512)	**0.0000** (**0.0000**)	**0.0000** (**0.0000**)
	$DeepSurv_I$(R&P)	0.7080 (0.0016)	0.2168 (0.0010)	0.4406 (0.4597)	0.0003 (0.0003)	0.0000 (0.0001)	0.7075 (0.0009)	0.2167 (0.0011)	0.4276 (0.4386)	0.0001 (0.0000)	**0.0000** (**0.0000**)
	$DeepSurv_G$(Keya et al.)	0.7076 (0.0015)	0.2397 (0.0822)	0.4963 (0.4986)	0.0115 (0.0055)	0.0036 (0.0022)	0.7076 (0.0015)	0.2397 (0.0822)	0.4963 (0.4986)	0.0115 (0.0055)	0.0036 (0.0022)
	$DeepSurv_G$(R&P)	0.7070 (0.0020)	0.2168 (0.0011)	0.3980 (0.4415)	0.0049 (0.0018)	0.0010 (0.0004)	0.7069 (0.0020)	0.2168 (0.0011)	0.3933 (0.4365)	0.0046 (0.0016)	0.0009 (0.0003)
	$DeepSurv_\cap$(Keya et al.)	0.6537 (0.0054)	0.1998 (0.0008)	1.0407 (0.9759)	0.0694 (0.0143)	0.0371 (0.0078)	0.6537 (0.0054)	0.1998 (0.0008)	1.0407 (0.9759)	0.0694 (0.0143)	0.0371 (0.0078)
	Deep DRO-COX	0.6830 (0.0050)	**0.1869** (**0.0004**)	**0.3628** (**0.4352**)	0.0006 (0.0004)	0.0002 (0.0001)	0.6830 (0.0050)	**0.1869** (**0.0004**)	**0.3628** (**0.4352**)	0.0006 (0.0004)	0.0002 (0.0001)
	Deep DRO-COX (SPLIT)	0.6829 (0.0049)	0.1881 (0.0012)	0.3853 (0.4475)	0.0006 (0.0005)	0.0002 (0.0001)	0.6829 (0.0049)	0.1881 (0.0012)	0.3853 (0.4475)	0.0006 (0.0005)	0.0002 (0.0001)

Table D.5: Test set scores for DRO-COX (SPLIT) on the FLC (age) dataset using $n_2 = 0.1n, 0.2n, 0.3n, 0.4n, 0.5n$ (corresponding to $n_1 = 0.9n, 0.8n, 0.7n, 0.6n, 0.5n$). The format of this table is similar to that of Table 4.2 although here we do not bold or highlight any cells, as our main finding here is that the scores are not dramatically different for the different choices for n_1 or n_2.

	n_2	Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	$0.1n$	0.7822 (0.0183)	0.1410 (0.0056)	0.2336 (0.3584)	0.0002 (0.0003)	0.0000 (0.0000)
	$0.2n$	0.7945 (0.0069)	0.1402 (0.0029)	0.1805 (0.2610)	0.0001 (0.0001)	0.0000 (0.0000)
	$0.3n$	0.7970 (0.0037)	0.1397 (0.0025)	0.1280 (0.1689)	0.0000 (0.0001)	0.0000 (0.0000)
	$0.4n$	0.7970 (0.0043)	0.1392 (0.0015)	0.1470 (0.1767)	0.0000 (0.0000)	0.0000 (0.0000)
	$0.5n$	0.7964 (0.0045)	0.1389 (0.0008)	0.1175 (0.1482)	0.0000 (0.0000)	0.0000 (0.0000)
Nonlinear	$0.1n$	0.7583 (0.0109)	0.1907 (0.0764)	1.0783 (1.3112)	0.0076 (0.0029)	0.0018 (0.0007)
	$0.2n$	0.7712 (0.0107)	0.1622 (0.0095)	1.1352 (1.2528)	0.0065 (0.0017)	0.0014 (0.0003)
	$0.3n$	0.7709 (0.0205)	0.1650 (0.0025)	1.1943 (1.2232)	0.0056 (0.0013)	0.0013 (0.0002)
	$0.4n$	0.7731 (0.0178)	0.1633 (0.0057)	1.1958 (1.2023)	0.0056 (0.0014)	0.0012 (0.0003)
	$0.5n$	0.7784 (0.0092)	0.1647 (0.0037)	1.1632 (1.1853)	0.0054 (0.0013)	0.0011 (0.0002)

Table D.6: Test set scores of DRO-COX (SPLIT, ONE SIDE) vs DRO-COX (SPLIT) on the FLC (age) dataset. The format of this table is the same that of Table 4.2 except without any cells highlighted in green as we are not comparing against baselines by previous authors.

	Methods	Accuracy Metrics		Fairness Metrics		
		$C^{td}\uparrow$	IBS$\downarrow$	CI(%)$\downarrow$	$F_{CI}\downarrow$	$F_{CG}\downarrow$
Linear	DRO-COX (SPLIT, ONE SIDE)	0.7810 (0.0109)	**0.1330** (**0.0002**)	0.2030 (0.2859)	0.0000 (0.0001)	0.0000 (0.0000)
	DRO-COX (SPLIT)	**0.7964** (**0.0045**)	0.1389 (0.0008)	**0.1175** (**0.1482**)	**0.0000** (**0.0000**)	**0.0000** (**0.0000**)
Non-linear	DEEP DRO-COX (SPLIT, ONE SIDE)	0.7554 (0.0231)	**0.1332** (**0.0002**)	**0.9530** (**1.0637**)	0.0060 (0.0019)	0.0015 (0.0005)
	Deep DRO-COX (SPLIT)	**0.7784** (**0.0092**)	0.1647 (0.0037)	1.1632 (1.1853)	**0.0054** (**0.0013**)	**0.0011** (**0.0002**)

John Duchi, Tatsunori Hashimoto, and Hongseok Namkoong. Distributionally robust losses for latent covariate mixtures. *Operations Research*, 2022.

John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. Fairness through awareness. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, pages 214–226, 2012.

David Faraggi and Richard Simon. A neural network model for survival data. *Statistics in Medicine*, 14(1):73–82, 1995.

James R Foulds, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. An intersectional definition of fairness. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1918–1921. IEEE, 2020.

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

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

Stefan Groha, Sebastian M Schmon, and Alexander Gusev. A general framework for survival analysis and multi-state modelling. *arXiv preprint arXiv:2006.04893*, 2020.

Humza Haider, Bret Hoehn, Sarah Davis, and Russell Greiner. Effective ways to build and evaluate individual survival distributions. *J. Mach. Learn. Res.*, 21(85):1–63, 2020.

Frank E Harrell, Robert M Califf, and David B Pryor. Evaluating the yield of medical tests. *Journal of the American Medical Association*, 247(18):2543–2546, 1982.

Tatsunori Hashimoto, Megha Srivastava, Hongseok Namkoong, and Percy Liang. Fairness without demographics in repeated loss minimization. In *International Conference on Machine Learning*, pages 1929–1938. PMLR, 2018.

Shu Hu and George H Chen. Distributionally robust survival analysis: A novel fairness loss without demographics. In *Machine Learning for Health*, pages 62–87. PMLR, 2022.

Shu Hu, Yiming Ying, and Siwei Lyu. Learning by minimizing the sum of ranked range. *Advances in Neural Information Processing Systems*, 2020.

Shu Hu, Lipeng Ke, Xin Wang, and Siwei Lyu. TkML-AP: Adversarial attacks to top-k multi-label learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7649–7657, 2021.

Shu Hu, Xin Wang, and Siwei Lyu. Rank-based decomposable losses in machine learning: A survey. *arXiv preprint arXiv:2207.08768*, 2022a.

Shu Hu, Yiming Ying, Xin Wang, and Siwei Lyu. Sum of ranked range loss for supervised learning. *Journal of Machine Learning Research*, 23(112):1–44, 2022b.

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

Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. *BMC Medical Research Methodology*, 18(1):1–12, 2018.

Kamrun Naher Keya, Rashidul Islam, Shimei Pan, Ian Stockwell, and James Foulds. Equitable allocation of healthcare resources with fair survival models. In *Proceedings of the 2021 SIAM International Conference on Data Mining (SDM)*, pages 190–198. SIAM, 2021.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

John P Klein and Melvin L Moeschberger. *Survival Analysis: Techniques for Censored and Truncated Data*. Springer, 2003.

William A Knaus, Frank E Harrell, Joanne Lynn, Lee Goldman, Russell S Phillips, Alfred F Connors, Neal V Dawson, William J Fulkerson, Robert M Califf, and Norman Desbiens. The SUPPORT prognostic model: Objective estimates of survival for seriously ill hospitalized adults. *Annals of Internal Medicine*, 122(3):191–203, 1995.

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

Havard 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.

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar. DeepHit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Mike Li, Hongseok Namkoong, and Shangzhou Xia. Evaluating model performance under worst-case subpopulations. *Advances in Neural Information Processing Systems*, 2021.

Intae Moon, Stefan Groha, and Alexander Gusev. Survlatent ode: A neural ode based time-to-event model with competing risks for longitudinal data improves cancer-associated venous thromboembolism (vte) prediction. In *Machine Learning for Healthcare Conference*, 2022.

Md Mahmudur Rahman and Sanjay Purushotham. Fair and interpretable models for survival analysis. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1452–1462, 2022.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations*, 2020.

Raphael Sonabend, Florian Pfisterer, Alan Mishler, Moritz Schauer, Lukas Burk, and Sebastian Vollmer. Flexible group fairness metrics for survival analysis. *arXiv preprint arXiv:2206.03256*, 2022.

Harald Steck, Balaji Krishnapuram, Cary Dehing-Oberije, Philippe Lambin, and Vikas C Raykar. On ranking in survival analysis: Bounds on the concordance index. *Advances in Neural Information Processing Systems*, 2007.

Weijing Tang, Kevin He, Gongjun Xu, and Ji Zhu. Survival analysis via ordinary differential equations. *Journal of the American Statistical Association*, pages 1–16, 2022a.

Weijing Tang, Jiaqi Ma, Qiaozhu Mei, and Ji Zhu. Soden: A scalable continuous-time survival model through ordinary differential equation networks. *J. Mach. Learn. Res.*, 23:34–1, 2022b.

Jing Teng. SEER breast cancer data. *IEEE Dataport*, 2019. URL `https://dx.doi.org/10.21227/a9qy-ph35`.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-RMSProp: Divide the gradient by a running average of its recent magnitude. *Coursera: Neural networks for machine learning*, 4(2):26–31, 2012.

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

Zhiliang Wu, Yinchong Yang, Peter A Fashing, and Volker Tresp. Uncertainty-aware time-to-event prediction using deep kernel accelerated failure time models. In *Machine Learning for Healthcare Conference*, pages 54–79. PMLR, 2021.

Wenbin Zhang and Jeremy C Weiss. Longitudinal fairness with censorship. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.