Distributionally Robust Survival Analysis: A Novel Fairness Loss Without Demographics

Shu Hu* SHUHU@CMU.EDU
George H. Chen*† GEORGECHEN@CMU.EDU
Heinz College of Information Systems and Public Policy, Carnegie Mellon University

Abstract

We propose a general approach for training survival analysis models that minimizes a worst-case error across *all* subpopulations that are large enough (occurring with at least a user-specified minimum probability). This approach uses a training loss function that does not know any demographic information to treat as sensitive. Despite this, we demonstrate that our proposed approach often scores better on recently established fairness metrics (without a significant drop in prediction accuracy) compared to various baselines, including ones which directly use sensitive demographic information in their training loss. Our code is available at: https://github.com/discovershu/DRO_COX

Keywords: survival analysis, fairness, distributionally robust optimization

1. Introduction

One of the recent advances for encouraging fairness in machine learning models is to minimize a worst-case error over *all* subpopulations that are large enough (e.g., Hashimoto et al. 2018; Duchi and Namkoong 2021; Li et al. 2021; Duchi et al. 2022; Hu et al. 2022a). In particular, a modeler specifies a probability threshold α of a minority subpopulation occurring. The goal is to ensure that all subpopulations with at least occurrence probability α have low error whereas we make no promises for subpopulations occurring with probability less than α. The modeler need not provide a list of subpopulations to account for. This problem is tractable to solve in practice and is called *distributionally robust optimization* (DRO).

We emphasize that curating a list of all subpopulations to account for can be challenging in practice for numerous 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 such sensitive attributes in a training loss function.

Our main contribution in this paper is to show how to apply DRO to survival analysis. The key technical challenge is that existing DRO theory assumes that the overall training loss can be separated across individuals so that any individual's loss term does not

* equal contribution
† corresponding author

depend on other individuals. This assumption does not hold for many survival analysis loss functions, including that of the popular Cox proportional hazards model (Cox, 1972), due to pairwise comparisons from ranking or similarity score evaluations (e.g., Steck et al. 2007; Lee et al. 2018; Chen 2020; Wu et al. 2021). We use a sample splitting approach to address this technical challenge. We specifically show how to use DRO with the Cox model and its deep neural network variant (Faraggi and Simon, 1995; Katzman et al., 2018). On three standard survival analysis datasets that have been previously used for research on fairness, our approach often outperforms various baseline methods in terms of existing fairness metrics that focus on user-specified sensitive attributes, including baselines with training loss functions that directly use these sensitive attributes (whereas ours does not). 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). This tradeoff in accuracy vs fairness can be tuned by the user. For ease of presentation, we apply DRO only to classical and deep Cox models, but the ideas we use readily extend to other survival models as well.

2. Background

We review the standard survival analysis setup in Section 2.1, classical and neural network variants of the Cox proportional hazards model in Section 2.2, and existing work on fair survival analysis in Section 2.3. We defer explaining the basics of DRO to Section 3 when we simultaneously explain how we apply DRO to survival analysis.

2.1. Survival Analysis Setup

Survival analysis aims to model the amount of time that will elapse before a critical event of interest happens. Classically, this critical event is death (i.e., we model time until different individuals are deceased), but the critical event need not be death and could instead be, for example, discharge from a hospital, or awakening from a coma.

We assume that we have training data $\{(X_i, Y_i, \delta_i)\}_{i=1}^n$, where the i-th training patient has feature vector $X_i \in \mathcal{X}$, 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 patient), 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 patient, i.e., the true time until event is unknown but we know that it is at least Y_i. In more detail, each training data point (X_i, Y_i, δ_i) is assumed to be generated from the following procedure:

1. Sample feature vector X_i from a feature vector 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 $F(\cdot|x)$.

A standard prediction task is to estimate the probability that a patient with feature vector x survives beyond time t. Formally, this is defined as the survival function

$$S(t|x) := \mathbb{P}(T > t|X = x) = 1 - F(t|x).$$

We explain how to estimate $S(\cdot|x)$ using variants of the Cox model next.

2.2. Classical and Deep Cox Models

The Cox proportional hazards model (Cox, 1972) estimates a transformed version of the survival function $S(\cdot|x)$ called the *hazard function*, given by $h(t|x) := -\frac{\partial}{\partial t} \log S(t|x)$; from negating both sides of this equation, integrating over time, and exponentiating, we get $S(t|x) = \exp(-\int_0^t h(u|x)du)$. Thus, if we have an estimate of $h(\cdot|x)$, then we can readily estimate the survival function $S(\cdot|x)$.

The Cox model assumes that the hazard function has the factorization

$$h(t|x) = h_0(t)\exp(f(x;\theta)), \tag{1}$$

where h_0 is called the baseline hazard function (h_0 maps a nonnegative time $t \geq 0$ to a nonnegative number), and $f(\cdot;\theta)$ is the so-called log partial hazard function ($f(x;\theta)$ could be thought of as assigning a real-valued "risk score" to feature vector x: when $f(x;\theta)$ is higher, then x has a higher risk of the critical event happening); note that θ refers to the parameters of f.

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. In either case, the standard approach for learning a Cox model is to first learn $f(\cdot;\theta)$ (i.e., learn the parameters θ) by minimizing the negative log partial likelihood:

$$\mathcal{L}_{\text{average}}(\theta) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(\theta), \tag{2}$$

where the i-th patient's loss is

$$\ell_i(\theta) := -\delta_i \Big[f(X_i;\theta) - \log \sum_{\substack{j=1,\ldots,n \\ \text{s.t. } Y_j \geq Y_i}} \exp(f(X_j;\theta)) \Big]. \tag{3}$$

If the i-th patient is censored ($\delta_i = 0$), then $\ell_i(\theta) = 0$. Thus, the loss $\mathcal{L}_{\text{average}}(\theta)$ weights *uncensored* training patients equally. After learning $f(\cdot;\theta)$, we then estimate h_0; as this step is not essential to our exposition, we explain it in Appendix A, along with details on constructing the final estimate of $S(\cdot|x)$.

2.3. Fair Survival Analysis

Despite many advances in survival analysis methodology in recent years (e.g., see the survey by Wang et al. (2019)), very few of these advances focus on fairness (Keya et al., 2021; Zhang and Weiss, 2022; Sonabend et al., 2022; Rahman and Purushotham, 2022). From a practical standpoint, asking for a survival analysis model to be fair is not different from asking for any other machine learning model to be fair in that if the model is used to assist high-stakes decision making (e.g., helping clinicians decide on personalized treatments, improving how hospitals allocate resources for different patients), then accounting for some notion of fairness could be an important design consideration.

To this end, 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 to the conventional loss function stated in equation (2). 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) with the idea that intersections of protected groups could be vulnerable to additional harms.

However, a major limitation of the notions of fairness defined by Keya et al. (2021) for

survival analysis is that they focus on predicted model outputs and do not actually use any of the label information (the observed time Y_i and event indicator δ_i variables). 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. (2021) would be asking for the predicted outcomes of the two different 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 observed times and censoring 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/groups to account for) or are simply adding a loss term that encourages smoothness in the model outputs (the individual fairness metrics by Keya et al. (2021) and Rahman and Purushotham (2022) are simply encouraging the predicted model output to be Lipschitz continuous; for details, see Appendix B). In contrast, our proposed approach does not require the user to specify any sensitive demographic attributes in the training loss function, and is not simply encouraging the model output to be Lipschitz continuous.

3. DRO for Survival Analysis

We now present our proposed method that applies distributionally robust optimization (DRO) to survival analysis. DRO uses a worst-case average error over "large enough" subpopulations. Note that there are now a number of DRO variants (e.g., Hashimoto et al. 2018; Sagawa et al. 2020; Duchi and Namkoong 2021; Duchi et al. 2022). We use the one by Hashimoto et al. (2018).

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 $\mathbb{P} := \sum_{k=1}^K \pi_k \mathbb{P}_k$, 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.

We would like to minimize the risk

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

where $\widetilde{\ell}$ 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 nor how many such groups there are. However, 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 3.1, how to empirically minimize the upper bound in Section 3.2, and finally how to choose the loss $\widetilde{\ell}$ in Section 3.3. Note that the material in Sections 3.1 and 3.2 is not novel; these sections translate the DRO formulation by Hashimoto et al. (2018) to survival analysis. On the other hand, Section 3.3 is novel and focuses on a technical complication in applying DRO to survival analysis.

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:

$$\mathcal{R}_{\text{DRO}}(\theta; r) := \sup_{\mathbb{Q}\in\mathcal{B}_r(\mathbb{P})} \mathbb{E}_{(X,Y,\delta)\sim\mathbb{Q}}[\widetilde{\ell}(\theta; X, Y, \delta)]. \tag{4}$$

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 is at most r from distribution $\mathbb{P}$. Formally,*

$$\mathcal{B}_r(\mathbb{P}) := \{\textit{dist. } \mathbb{Q} \mid \mathbb{Q} \ll \mathbb{P}, D_{\chi^2}(\mathbb{Q}\|\mathbb{P}) \le r\},$$

where the notation "$\mathbb{Q} \ll \mathbb{P}$" roughly means that $\mathbb{Q}$ has the same (or smaller) support as $\mathbb{P}$.[1] *Meanwhile, $D_{\chi^2}(\mathbb{Q}\|\mathbb{P}) := \int(\frac{d\mathbb{Q}}{d\mathbb{P}} - 1)^2 d\mathbb{P}$.*

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 $\min_{k=1,\dots,K}\pi_k$).

Proposition 2 *(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., $\alpha \le \min_{k=1,\dots,K}\pi_k$). Then $\mathcal{R}_{DRO}(\theta; r_{\max}) \ge \mathcal{R}_{\max}(\theta)$, where $r_{\max} := (\frac{1}{\alpha} - 1)^2$.*

In other words, if we have a guess for $\alpha \in (0, \min_{k=1,\dots,K}\pi_k]$, 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 $\pi_1, \dots, \pi_K$ 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 least 0.1.

3.2. Empirical DRO Risk

The next issue is how to minimize the risk $\mathcal{R}_{\text{DRO}}(\theta; r_{\max})$. This risk appears challenging to evaluate since it involves a 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 3 *(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., $[a]_+ := \max\{a, 0\}$ for any $a \in \mathbb{R}$), and $C := \sqrt{2(\frac{1}{\alpha} - 1)^2 + 1}$. Then*

$$\mathcal{R}_{DRO}(\theta; r_{\max}) = \inf_{\eta\in\mathbb{R}} \left\{ C\sqrt{\mathbb{E}_{(X,Y,\delta)\sim\mathbb{P}}\left[[\widetilde{\ell}(\theta; X, Y, \delta) - \eta]_+^2\right]} + \eta \right\}. \tag{5}$$

The right-hand side of equation (5) could be interpreted as follows. Suppose that we have

1. The measure-theoretic definition of "$\mathbb{Q} \ll \mathbb{P}$" is that $\mathbb{Q}$ is absolutely continuous with respect to $\mathbb{P}$.

achieved the optimal value η^*. Then the loss from a patient will be ignored if it is less than η^* (due to the ReLU function). Thus, only the patients with losses above η^* are considered for learning the survival model.

Note that as we vary the model parameters θ, the different patients' losses change. Thus, as a function of θ, the DRO risk $\mathcal{R}_{\text{DRO}}(\theta; r_{\max})$ dynamically adjusts which patients to focus on, always prioritizing the patients with the highest loss values (again, we only consider the patients 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 (5) with an empirical average to arrive at the following optimization problem:

$$\min_{\theta\in\Theta,\eta\in\mathbb{R}} \mathcal{L}_{\text{DRO}}(\theta, \eta), \tag{6}$$

where Θ denotes the feasible set of the model parameters, and we define the empirical loss

$$\mathcal{L}_{\text{DRO}}(\theta, \eta) := C\sqrt{\frac{1}{n}\sum_{i=1}^{n}[\widetilde{\ell}(\theta; X_i, Y_i, \delta_i) - \eta]_+^2} + \eta. \tag{7}$$

Numerical optimization. The optimization problem in equation (6) 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 $\mathcal{L}_{\text{DRO}}(\theta, \eta)$. To do this, we use binary search to find the global optimum of η since $\mathcal{L}_{\text{DRO}}(\theta, \eta)$ is a convex function with respect to η.
- We fix η and update θ by minimizing $\mathcal{L}_{\text{DRO}}(\theta, \eta)$ (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 Appendix C.

3.3. Choosing the Individual Loss $\widetilde{\ell}$

The technical difficulty in applying DRO to the Cox model is somewhat subtle. In our description of DRO so far, the loss $\widetilde{\ell}$ that is mentioned depends only on model parameters θ and a single data point (X, Y, δ). In contrast, for the Cox model, the i-th patient's loss $\ell_i(\theta)$ as described in equation (3) can actually depend on multiple training patients. The reason is that inside the log term of equation (3), there is a sum over all training patients $j = 1, \ldots, n$ whose observed time Y_j is at least Y_i. Thus, replacing $\widetilde{\ell}(\theta; X_i, Y_i, \delta_i)$ in equation (7) with the Cox individual loss $\ell_i(\theta)$ actually invalidates the theory we have covered thus far. However, our experiments later will reveal that this replacement works very well in practice. We call this method DRO-COX.

A theoretically sound DRO method for Cox models. We show how to define the individual loss $\widetilde{\ell}$ so that it complies with existing DRO theory. To achieve this, we use sample splitting and an approximation of the Cox individual loss. We divide the training patients into two sets $\mathcal{D}_1 \subset \{1, \ldots, n\}$ and $\mathcal{D}_2 := \{1, \ldots, n\} \setminus \mathcal{D}_1$ of sizes $n_1 := |\mathcal{D}_1|$ and $n_2 := |\mathcal{D}_2| = n - n_1$. The high-level idea is that we only compute an approximation of the Cox individual loss $\ell_i(\theta)$ for $i \in \mathcal{D}_1$ (so that the empirical average in the DRO loss $\mathcal{L}_{\text{DRO}}(\theta, \eta)$ is modified to only be over the training patients in $\mathcal{D}_1$). Meanwhile, each $\ell_i(\theta)$ for $i \in \mathcal{D}_1$ is modified so that the sum inside the log term only depends on the i-th patient and patients in $\mathcal{D}_2$.

In more detail, we approximate $\ell_i(\theta)$ for $i \in \mathcal{D}_1$ with the new individual loss

$$\widetilde{\ell}_{\text{split}}(\theta; X_i, Y_i, \delta_i, \mathcal{D}_2) := -\delta_i\big[f(X_i; \theta) - \log\big(\Phi(\theta; X_i, Y_i, \mathcal{D}_2)\big)\big], \tag{8}$$

where

$$\Phi(\theta; X_i, Y_i, \mathcal{D}_2) := \exp(f(X_i;\theta)) + \sum_{j\in\mathcal{D}_2 \text{ s.t. } Y_j \geq Y_i} \exp(f(X_j;\theta)).$$

This loss is no longer equal to the Cox individual loss $\ell_i(\theta)$ because the log term is computed only using the i-th training patient and patients in $\mathcal{D}_2$. Importantly, treating the training patients in $\mathcal{D}_2$ as fixed, then the individual loss $\widetilde{\ell}_{\text{split}}(\theta; X_i, Y_i, \delta_i, \mathcal{D}_2)$ only depends on θ and the data point (X_i, Y_i, δ_i). 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.

We next modify the empirical DRO loss $\mathcal{L}_{\text{DRO}}(\theta, \eta)$ given in equation (7) so that the empirical average (inside the square root) is only computed using training patients in $\mathcal{D}_1$, and we set $\widetilde{\ell}$ equal to $\widetilde{\ell}_{\text{split}}$. In particular, we replace $\mathcal{L}_{\text{DRO}}(\theta, \eta)$ with the loss

$$\mathcal{L}_{\text{DRO-split}}(\theta, \eta, \mathcal{D}_1, \mathcal{D}_2) := C\sqrt{\frac{1}{|\mathcal{D}_1|}\sum_{i\in\mathcal{D}_1}[\widetilde{\ell}_{\text{split}}(\theta; X_i, Y_i, \delta_i, \mathcal{D}_2) - \eta]_+^2} + \eta. \tag{9}$$

Although minimizing $\mathcal{L}_{\text{DRO-split}}(\theta, \eta, \mathcal{D}_1, \mathcal{D}_2)$ is compliant with DRO theory, it uses data less effectively since at most n_1 patients (rather than n) are used to compute the empirical average (note that only uncensored patients have nonzero loss), and for these patients, at most n_2+1 points are used to compute the sum inside each of their log terms.

A simple way to more effectively use the data is to change optimization problem (6) to instead minimize the sum of two losses: the first is $\mathcal{L}_{\text{DRO-split}}(\theta, \eta, \mathcal{D}_1, \mathcal{D}_2)$, and the second is $\mathcal{L}_{\text{DRO-split}}(\theta, \eta', \mathcal{D}_2, \mathcal{D}_1)$, i.e., the latter loss swaps the roles of $\mathcal{D}_1$ and $\mathcal{D}_2$ and also η is replaced with a different η' (the two losses do not share the same η). The iterative optimization procedure in Section 3.2 can still be applied except where each iteration now consists of three steps: updating η, η', and θ. We refer to this method as DRO-COX (SPLIT); we provide pseudocode for it in Appendix C.

4. Experiments

We compare DRO-COX and DRO-COX (SPLIT) against various baselines using a similar experimental setup as Keya et al. (2021).

Datasets. We use three standard, publicly available survival analysis datasets that have been used in fair survival analysis research:

- 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 regard binary encoded age (age≤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 regard binary encoded age (age≤65 and age>65), race (white and non-white), and gender (women and men) as sensitive attributes.
- The **SEER** dataset (Teng, 2019) of breast cancer patients is obtained from the 2017 November update of the Sureillance, Epidemiology, and End Results (SEER) program of the National Cancer Institute. The dataset is on female patients with breast cancer diagnosed in 2006-2010. We regard binary encoded age (age≤65 and age>65) and race (white and non-white) as sensitive attributes.

Basic characteristics of these datasets are reported in Table 1. For all datasets, we first use a random 80%/20% train/test split to hold out a test set that will be the same across experimental repeats. Then we repeat

Table 1: Basic dataset characteristics.

	FLC	SUPPORT	SEER
# samples	7,874	9,105	4,024
# features	6 (9*)	14 (19*)	13
Censoring rate	0.725	0.319	0.847
Sensitive attributes	age, gender	age, race, gender	age, race

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

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.

Evaluation metrics. We use accuracy and, separately, fairness metrics. The accuracy metrics we use are (a) concordance index (abbreviated as "c-index", higher is better) (Harrell et al., 1982), (b) time-dependent AUC (AUC, higher is better) (Chambless and Diao, 2006), (c) log partial likelihood (LPL, higher is better), and (d) integrated IPCW Brier Score (IBS, lower is better) (Graf et al., 1999). Note that the *negative* LPL averaged across data points is precisely given by equation (2) (all methods we consider are variants of Cox models).

As our experimental setup is largely based on that of Keya et al. (2021), we use the fairness metrics that they had defined: individual fairness (F_I), group fairness (F_G), and intersectional fairness ($\mathrm{F}_\cap$). We also include a summary fairness metric $\mathrm{F}_A = (\mathrm{F}_I + \mathrm{F}_G + \mathrm{F}_\cap)/3$. As we pointed out in Section 2.3, the fairness metrics by Keya et al. (2021) do not actually account for accuracy. We thus also include the concordance imparity (CI) fairness metric by Zhang and Weiss (2022) that is based on accuracy. For all fairness metrics, lower is better. Definitions of these fairness metrics are in Appendix B.

Note that the fairness metrics F_G and CI require us to specify groups. For the FLC dataset, we separately use (binary encoded) age and gender (i.e., we first run experiments using only age in evaluating F_G and CI; 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. Note that since F_A depends on F_G, the F_A metric also changes when we switch the sensitive attribute used for F_G and CI. Meanwhile, the intersectional fairness metric $\mathrm{F}_\cap$ is meant for when multiple sensitive attributes are specified. Per dataset, we use all sensitive attributes specified in Table 1 to evaluate $\mathrm{F}_\cap$.

Methods evaluated. For simplicity, all models evaluated are Cox models, either assuming the **linear** setting (the log partial hazard function is $f(x;\theta) = \theta^T x$) or the **nonlinear** setting in which f is a multilayer perceptron (MLP). When DRO-COX or DRO-COX (SPLIT) are used in the latter case, we add the prefix "Deep" in tables for clarity.

For baselines, the unregularized linear Cox model (Cox, 1972) is denoted as "Cox" in our tables, 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), then 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)".[2] Note that group fairness regularization (suffixes "$_G$(Keya et al.)" and "$_G$(R&P)") uses the same groups that test set F_G and CI fairness metrics use.

Hyperparameter grids for all methods (including our DRO-COX variants) are in Appendix D, where we also provide information on the compute environment that we used. 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-index from the classical Cox model (for the linear setting) or DeepSurv (for the nonlinear setting) while minimizing the validation set CI fairness metric (Keya et al. used their own fairness metrics though instead of CI).

Experimental results. We report the test set evaluation metrics for FLC (using age to evaluate F_G and CI) in Table 2, SUPPORT (gender) in Table 3, and SEER (race) in Table 4. Experimental results using other sensitive attributes for the datasets have similar trends and are in Appendix E. From these tables, we have the following observations:

- Among linear methods, DRO-COX 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 deep DRO-COX variant.
- The performance difference (in terms of both accuracy and fairness) between DRO-COX and DRO-COX (SPLIT) is not clear cut; sometimes one performs better 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 validation set c-index), we also tried instead minimizing the validation set F_A metric and found similar results: our DRO-COX variants end up consistently outperforming all the baselines on F_A (and also often achieves competitive CI metric scores) without a large accuracy drop. 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 $\mathcal{L}_{\text{DRO-split}}(\theta, \eta, \mathcal{D}_1, \mathcal{D}_2)$ and $\mathcal{L}_{\text{DRO-split}}(\theta, \eta', \mathcal{D}_2, \mathcal{D}_1)$ (i.e., if it only used one of these), then it performs worse. For details on these experiments, see Appendix E.

2. 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 2: 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.

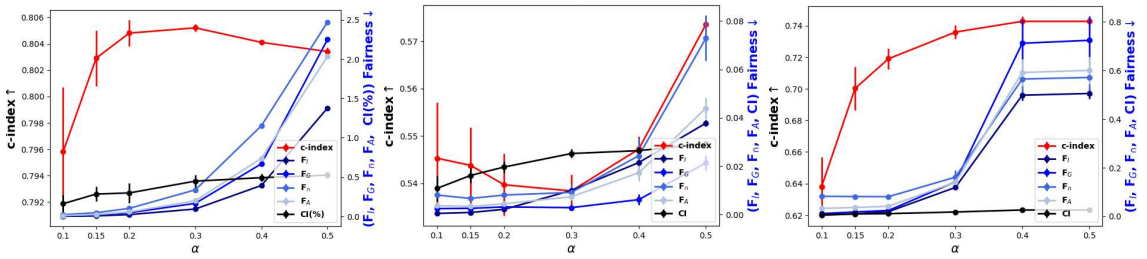
	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	0.8032 (0.0002)	0.8176 (0.0005)	**-6.3724 (0.0011)**	0.1739 (0.0004)	1.8787 (0.0304)	3.0282 (0.0469)	2.8355 (0.0297)	2.5808 (0.0332)	0.5350 (0.0413)
	Cox_I(Keya et al.)	0.7937 (0.0068)	0.8179 (0.0067)	-6.6044 (0.0721)	0.1414 (0.0073)	0.4493 (0.1217)	0.7623 (0.1995)	1.2045 (0.2701)	0.8054 (0.1966)	0.5400 (0.3270)
	Cox_I(R&P)	**0.8034 (0.0007)**	0.8188 (0.0008)	-6.4111 (0.0203)	0.1636 (0.0030)	1.0920 (0.1391)	1.7990 (0.2242)	2.1828 (0.1620)	1.6913 (0.1747)	0.4330 (0.1196)
	Cox_G(Keya et al.)	0.7974 (0.0117)	**0.8196 (0.0063)**	-6.6869 (0.0693)	0.1492 (0.0077)	1.0495 (0.6647)	1.1802 (0.6893)	1.7940 (0.7234)	1.3412 (0.6880)	0.3410 (0.3011)
	Cox_G(R&P)	0.8027 (0.0005)	0.8172 (0.0011)	-6.3921 (0.0095)	0.1676 (0.0012)	1.3130 (0.0801)	2.1601 (0.1296)	2.3984 (0.0903)	1.9571 (0.0992)	0.4950 (0.1517)
	$Cox_\cap$(Keya et al.)	0.7870 (0.0029)	0.8148 (0.0017)	-6.7272 (0.0048)	**0.1400 (0.0005)**	0.2921 (0.0056)	**0.4156 (0.0220)**	**0.6827 (0.0291)**	**0.4635 (0.0180)**	1.0790 (0.1098)
	DRO-COX	0.7959 (0.0036)	0.8149 (0.0020)	-6.7630 (0.2113)	0.1408 (0.0050)	**0.2793 (0.1818)**	0.4694 (0.3016)	0.7880 (0.5011)	0.5122 (0.3282)	**0.0510 (0.0401)**
	DRO-COX (SPLIT)	0.8017 (0.0017)	0.8221 (0.0014)	-6.6593 (0.2604)	0.1658 (0.0098)	0.5611 (0.3588)	0.8935 (0.5652)	1.2734 (0.7880)	0.9094 (0.5705)	0.0840 (0.0594)
Nonlinear	DeepSurv	0.8070 (0.0014)	0.8247 (0.0026)	**-6.3552 (0.0052)**	0.1767 (0.0018)	2.9691 (1.2481)	4.6647 (1.9185)	2.8800 (0.0531)	3.5046 (1.0506)	0.2940 (0.2147)
	$DeepSurv_I$(Keya et al.)	0.7884 (0.0070)	0.8134 (0.0109)	-6.6416 (0.1399)	0.1441 (0.0130)	0.1510 (0.1062)	0.2425 (0.1675)	1.1281 (0.5625)	0.5072 (0.1335)	0.3700 (0.2523)
	$DeepSurv_I$(R&P)	0.8071 (0.0041)	0.8254 (0.0049)	-6.3824 (0.0743)	0.1729 (0.0093)	**0.0713 (0.1204)**	0.1167 (0.1785)	2.5089 (0.4270)	0.8990 (0.0643)	0.1870 (0.1117)
	$DeepSurv_G$(Keya et al.)	0.7990 (0.0120)	0.8189 (0.0108)	-6.4954 (0.1924)	0.4190 (0.2487)	0.1604 (0.3575)	**0.1600 (0.3249)**	1.0645 (0.6657)	0.4617 (0.4381)	0.2490 (0.1646)
	$DeepSurv_G$(R&P)	**0.8073 (0.0036)**	0.8255 (0.0049)	-6.3786 (0.0687)	0.1731 (0.0087)	0.2376 (0.2349)	0.3749 (0.3587)	2.6416 (0.4063)	1.0847 (0.1954)	0.2290 (0.1344)
	$DeepSurv_\cap$(Keya et al.)	0.7751 (0.0018)	0.7893 (0.0022)	-6.8458 (0.0031)	**0.1357 (0.0002)**	0.1688 (0.0035)	0.2412 (0.0051)	**0.4633 (0.0106)**	**0.2911 (0.0062**	0.4300 (0.1091)
	Deep DRO-COX	0.8068 (0.0024)	**0.8259 (0.0031)**	-6.4698 (0.1069)	0.1595 (0.0135)	1.3709 (1.1919)	2.1481 (1.8343)	1.8712 (0.6223)	1.7967 (1.1697)	**0.0730 (0.0822)**
	Deep DRO-COX (SPLIT)	0.7650 (0.0024)	0.7744 (0.0022)	-6.8071 (0.0091)	0.1703 (0.0002)	0.4480 (0.1050)	0.5327 (0.0706)	0.7762 (0.0992)	0.5856 (0.0914)	2.8000 (0.1450)

Table 3: Test set scores on the SUPPORT (gender) dataset, in the same format as Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	0.6025 (0.0005)	0.6163 (0.0010)	**-6.8761 (0.0010)**	0.2304 (0.0015)	0.2113 (0.0093)	0.0439 (0.0052)	0.4490 (0.0322)	0.2347 (0.0127)	1.4300 (0.0654)
	Cox_I(Keya et al.)	0.5881 (0.0114)	0.5998 (0.0142)	-6.9387 (0.0202)	**0.2157 (0.0060)**	0.0382 (0.0320)	0.0076 (0.0057)	0.0938 (0.0700)	0.0465 (0.0353)	0.9650 (0.6126)
	Cox_I(R&P)	0.6019 (0.0019)	0.6159 (0.0029)	-6.8798 (0.0029)	0.2282 (0.0013)	0.1814 (0.0127)	0.0383 (0.0139)	0.3841 (0.0240)	0.2013 (0.0126)	1.4190 (0.1002)
	Cox_G(Keya et al.)	**0.6030 (0.0007)**	**0.6177 (0.0011)**	-6.8772 (0.0017)	0.2297 (0.0018)	0.2016 (0.0117)	0.0032 (0.0016)	0.4012 (0.0163)	0.2020 (0.0071)	1.4190 (0.0632)
	Cox_G(R&P)	0.6018 (0.0017)	0.6156 (0.0027)	-6.8779 (0.0023)	0.2295 (0.0009)	0.1993 (0.0048)	0.0430 (0.0157)	0.4239 (0.0281)	0.2221 (0.0128)	1.4340 (0.1039)
	$Cox_\cap$(Keya et al.)	0.5715 (0.0062)	0.5718 (0.0081)	-6.9078 (0.0062)	0.2275 (0.0016)	0.1334 (0.0129)	0.0092 (0.0037)	0.0743 (0.0108)	0.0723 (0.0077)	1.1270 (0.2457)
	DRO-COX	0.5734 (0.0019)	0.6083 (0.0023)	-6.9388 (0.0007)	0.2210 (0.0010)	0.0378 (0.0013)	**0.0022 (0.0015)**	0.0731 (0.0094)	0.0377 (0.0033)	0.4350 (0.0674)
	DRO-COX (SPLIT)	0.5725 (0.0075)	0.6056 (0.0092)	-6.9410 (0.0057)	0.4264 (0.1667)	**0.0285 (0.0188)**	0.0041 (0.0028)	**0.0594 (0.0366)**	**0.0307 (0.0190)**	**0.3410 (0.1781)**
Nonlinear	DeepSurv	0.6108 (0.0029)	**0.6327 (0.0045)**	**-6.8754 (0.0040)**	0.2417 (0.0016)	0.4072 (0.0369)	0.0570 (0.0180)	0.4244 (0.0573)	0.2962 (0.0283)	1.6220 (0.3303)
	$DeepSurv_I$(Keya et al.)	0.5984 (0.0124)	0.6164 (0.0150)	-6.9130 (0.0220)	0.2376 (0.0182)	**0.0179 (0.0249)**	0.0059 (0.0076)	0.5688 (0.3030)	0.1975 (0.0930)	1.3280 (0.7670)
	$DeepSurv_I$(R&P)	0.6104 (0.0076)	0.6315 (0.0115)	-6.8761 (0.0132)	0.2379 (0.0079)	0.0544 (0.0468)	0.0132 (0.0141)	0.4997 (0.1722)	0.1891 (0.0435)	1.6490 (0.2368)
	$DeepSurv_G$(Keya et al.)	0.5982 (0.0109)	0.6176 (0.0144)	-6.9121 (0.0278)	0.2436 (0.0121)	0.1131 (0.0718)	**0.0047 (0.0036)**	0.3972 (0.1017)	0.1717 (0.0375)	1.6540 (0.3892)
	$DeepSurv_G$(R&P)	**0.6110 (0.0057)**	0.6325 (0.0089)	-6.8766 (0.0117)	0.2406 (0.0068)	0.0452 (0.0476)	0.0113 (0.0144)	0.5246 (0.1217)	0.1937 (0.0320)	1.6250 (0.1931)
	$DeepSurv_\cap$(Keya et al.)	0.6015 (0.0069)	0.6190 (0.0100)	-6.8794 (0.0055)	0.2378 (0.0053)	0.2465 (0.0424)	0.0053 (0.0032)	**0.0745 (0.0263)**	0.1088 (0.0213)	1.4110 (0.2129)
	Deep DRO-COX	0.5829 (0.0067)	0.6237 (0.0111)	-6.9253 (0.0025)	**0.2240 (0.0010)**	0.1109 (0.0377)	0.0058 (0.0021)	0.0816 (0.0095)	**0.0661 (0.0141)**	**1.2600 (0.4412)**
	Deep DRO-COX (SPLIT)	0.5448 (0.0015)	0.5625 (0.0021)	-6.9555 (0.0012)	0.6390 (0.0005)	0.1605 (0.0030)	0.0071 (0.0024)	0.1754 (0.0062)	0.1143 (0.0031)	2.1690 (0.0727)

Table 4: Test set scores on the SEER (race) dataset, in the same format as Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.7409 (0.0016)**	**0.7624 (0.0017)**	**-5.9427 (0.0034)**	0.0964 (0.0008)	0.6105 (0.0307)	0.1183 (0.0645)	0.6750 (0.0630)	0.4679 (0.0437)	1.1880 (0.1742)
	Cox_I(Keya et al.)	0.7143 (0.0300)	0.7367 (0.0303)	-6.1602 (0.0728)	0.0894 (0.0014)	0.1968 (0.0831)	0.0944 (0.0770)	0.3768 (0.1736)	0.2227 (0.0951)	1.5500 (1.5117)
	Cox_I(R&P)	0.7353 (0.0195)	0.7559 (0.0220)	-6.0046 (0.0616)	0.0919 (0.0012)	0.3655 (0.0720)	0.0707 (0.0567)	0.4031 (0.0698)	0.2798 (0.0396)	1.5390 (0.5256)
	Cox_G(Keya et al.)	0.7308 (0.0227)	0.7508 (0.0259)	-5.9867 (0.0813)	0.0951 (0.0024)	0.5345 (0.1282)	0.3053 (0.1003)	0.8003 (0.2040)	0.5467 (0.1429)	1.4360 (0.3377)
	Cox_G(R&P)	0.7339 (0.0196)	0.7548 (0.0222)	-5.9960 (0.0616)	0.0925 (0.0012)	0.3985 (0.0679)	0.1022 (0.0835)	0.4374 (0.0565)	0.3127 (0.0272)	1.7580 (0.5936)
	$\text{Cox}_\cap$(Keya et al.)	0.7280 (0.0237)	0.7516 (0.0260)	-6.0043 (0.0884)	0.0951 (0.0016)	0.5183 (0.0941)	0.2781 (0.0799)	0.6519 (0.2615)	0.4828 (0.1433)	0.9240 (0.4045)
	DRO-COX	0.7283 (0.0054)	0.7494 (0.0054)	-6.2140 (0.0653)	**0.0880 (0.0005)**	**0.0791 (0.0514)**	**0.0113 (0.0162)**	**0.1317 (0.0523)**	**0.0740 (0.0370)**	**0.2300 (0.2219)**
	DRO-COX (SPLIT)	0.7202 (0.0137)	0.7404 (0.0120)	-6.1406 (0.1324)	0.1020 (0.0315)	0.2150 (0.1876)	0.0392 (0.0491)	0.3115 (0.2198)	0.1885 (0.1484)	0.1840 (0.1930)
Nonlinear	DeepSurv	**0.7488 (0.0103)**	**0.7729 (0.0105)**	**-5.9582 (0.0538)**	0.0966 (0.0047)	0.3686 (0.0959)	0.1178 (0.0771)	0.4734 (0.1203)	0.3199 (0.0776)	0.4270 (0.4259)
	DeepSurv_I(Keya et al.)	0.7100 (0.0236)	0.7340 (0.0237)	-6.1830 (0.1620)	0.0995 (0.0085)	**0.0564 (0.0515)**	**0.0238 (0.0198)**	1.0618 (0.7650)	0.3807 (0.2370)	3.0400 (1.2074)
	DeepSurv_I(R&P)	0.7353 (0.0104)	0.7579 (0.0103)	-6.0247 (0.0938)	0.0944 (0.0057)	0.1146 (0.0559)	0.0297 (0.0250)	0.6595 (0.4998)	0.2679 (0.1465)	0.5220 (0.2891)
	DeepSurv_G(Keya et al.)	0.7299 (0.0224)	0.7540 (0.0210)	-6.0622 (0.1643)	0.0972 (0.0081)	0.2667 (0.1408)	0.0898 (0.0325)	0.6532 (0.4237)	0.3366 (0.1456)	0.7070 (0.8405)
	DeepSurv_G(R&P)	0.7368 (0.0114)	0.7594 (0.0109)	-6.0146 (0.0916)	0.0942 (0.0052)	0.1283 (0.0593)	0.0324 (0.0281)	0.6080 (0.3616)	0.2562 (0.0999)	0.5600 (0.4160)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7344 (0.0112)	0.7613 (0.0098)	-6.0001 (0.0791)	0.0958 (0.0047)	0.4034 (0.1813)	0.1209 (0.0566)	0.4576 (0.1760)	0.3273 (0.1143)	0.4920 (0.4089)
	Deep DRO-COX	0.7305 (0.0216)	0.7521 (0.0271)	-6.0667 (0.1196)	**0.0913 (0.0041)**	0.2206 (0.1239)	0.0498 (0.0556)	**0.3004 (0.1191)**	**0.1903 (0.0932)**	**0.0910 (0.0461)**
	Deep DRO-COX (SPLIT)	0.6980 (0.0024)	0.7264 (0.0027	-6.1182 (0.0221)	0.1023 (0.0003)	0.3781 (0.0565)	0.1945 (0.0386)	0.4135 (0.0405)	0.3287 (0.0419)	0.3480 (0.2794)



(*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 three datasets.

5. Discussion

We have shown how to apply DRO to Cox models in a manner that is compliant with existing DRO theory (DRO-COX (SPLIT)) and in a manner that is heuristic (DRO-COX). Importantly, how we applied DRO to Cox models works with other survival models as well. The key idea is to write the overall loss in terms of individual losses, which in turn could be used in a DRO framework. An open question is whether we could derive a theoretically sound DRO-COX variant that does not require sample splitting. This same technical challenge would arise in working with other survival models that use pairwise comparisons between patients. When a parametric survival model is used in which each patient's loss does not depend on other patients, we point out that existing DRO machinery directly works; a strategy such as sample splitting would be unnecessary. We defer a thorough evaluation of DRO applied to more survival models to future work.

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.

References

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.

Lloyd E Chambless and Guoqing Diao. Estimation of time-dependent area under the ROC curve for long-term risk prediction. *Statistics in Medicine*, 25(20):3474–3486, 2006.

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.

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.

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.

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.

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, 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.

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.

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.

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.

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.

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

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 Fasching, 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.

Appendix A. Estimating the Baseline Hazard and 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, where $j = 1, \ldots, m$. Then we compute the following estimate of h_0 at the j-th time step:

$$\widehat{h}_{0,j} := \frac{d_j}{\sum_{i=1}^{n} \mathbf{1}\{Y_i \geq t_j\} \exp(f(x_i;\widehat{\theta}))}.$$

After estimating the baseline hazard function, estimating the survival function is straightforward. Recall that $S(t|x) = \exp\Big(-\int_0^t h(u|x)du\Big)$. Then combining this equation with the factorization (1), we get

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

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

$$\widehat{H}_0(t) := \sum_{j=1}^{m} \mathbf{1}\{t_j \leq t\}\widehat{h}_{0,j}.$$

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 survival function estimate $\widehat{S}(t|x) := \exp(-\widehat{H}_0(t)\exp(f(x;\widehat{\theta})))$.

Appendix B. Fairness Metrics

In this paper, we use the individual, group, and intersectional fairness metrics defined by Keya et al. (2021) and also the concordance imparity (CI) metric by Zhang and Weiss (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* $\widetilde{h}(x) := \exp(f(x;\theta))$; this is the same as the hazard function given in equation (1) 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 indi-

viduals (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

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

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

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

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

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

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

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

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 $\mathcal{S} := \mathcal{G}_1 \times \mathcal{G}_2 \times \cdots \mathcal{G}_J$. 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

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

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:

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

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 B.1; 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.

Scale Issues with F_I and F_G

We point out that the F_I and F_G 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 $\widetilde{h}(x) = \exp(f(x;\theta))$. 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 and F_G 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.

Appendix C. Pseudocode for Our Proposed Methods

We provide pseudocode for DRO-COX and DRO-COX (SPLIT) in Algorithm C.1 and Algorithm C.2, respectively.

Algorithm B.1 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 $\leftarrow \max_{a,a' \in \mathcal{A} \text{ s.t. } a \neq a'} |\mathbf{CF}(a) - \mathbf{CF}(a')|$

Algorithm C.1 DRO-COX

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 $l = 0$ *to* max_iterations **do**
  for $i = 1$ *to* n **do**
    Set $u_i \leftarrow \ell_i(\widehat{\theta}_l)$ using equation (3).
  end
  Set $\widehat{\eta}$ to be the value of $\eta \in \mathbb{R}$ that minimizes $\mathcal{L}_{\text{DRO}}(\widehat{\theta}_l, \eta)$ as given in equation (7), where in the empirical average, the i-th individual's loss is set to be the variable u_i computed above. This minimization is solved using binary search.
  Set $\widehat{\theta}_{l+1} \leftarrow \widehat{\theta}_l - \xi \cdot \nabla_\theta \mathcal{L}_{DRO}(\widehat{\theta}_l, \widehat{\eta})$.
end
return $\widehat{\theta} \leftarrow \widehat{\theta}_{\text{max\_iterations}+1}$

Algorithm C.2 DRO-COX (SPLIT)

Input: A training dataset $\{(X_i, Y_i, \delta_i)\}_{i=1}^{n}$, minimum subpopulation probability hyperparameter α, 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 $l = 0$ *to* max_iterations **do**
  for $i \in \mathcal{D}_1$ **do**
    Set $u_i \leftarrow \widetilde{\ell}_{\text{split}}(\widehat{\theta}_l; X_i, Y_i, \delta_i, \mathcal{D}_2)$ with equation (8).
  end
  Set $\widehat{\eta}$ to be the value of $\eta \in \mathbb{R}$ that minimizes $\mathcal{L}_{\text{DRO-split}}(\widehat{\theta}_l, \eta, \mathcal{D}_1, \mathcal{D}_2)$ as given in equation (9), where in the empirical average, the i-th individual's loss is set to be the variable u_i computed above. This minimization is solved using binary search.
  for $i \in \mathcal{D}_2$ **do**
    Set $v_i \leftarrow \widetilde{\ell}_{\text{split}}(\widehat{\theta}_l; X_i, Y_i, \delta_i, \mathcal{D}_1)$ with equation (8).
  end
  Set $\widehat{\eta}'$ to be the value of $\eta' \in \mathbb{R}$ that minimizes $\mathcal{L}_{\text{DRO-split}}(\widehat{\theta}_l, \eta', \mathcal{D}_2, \mathcal{D}_1)$ as given in equation (9), where in the empirical average, the i-th individual's loss is set to be the variable v_i computed above. This minimization is solved using binary search.
  Set $\widehat{\theta}_{l+1} \leftarrow \widehat{\theta}_l - \xi \cdot \big(\nabla_\theta \mathcal{L}_{\text{DRO-split}}(\widehat{\theta}_l, \widehat{\eta}, \mathcal{D}_1, \mathcal{D}_2) + \nabla_\theta \mathcal{L}_{\text{DRO-split}}(\widehat{\theta}_l, \widehat{\eta}', \mathcal{D}_2, \mathcal{D}_1)\big)$.
end
return $\widehat{\theta} \leftarrow \widehat{\theta}_{\text{max\_iterations}+1}$

Appendix D. Hyperparameter Tuning and Compute Environment Details

Hyperparameters. 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. We tune on the following hyperparameter grid:

- learning rate: 0.01, 0.001, 0.0001
- 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 (for DRO-COX/DRO-COX (SPLIT) variants): 0.1, 0.15, 0.2, 0.3, 0.4, 0.5

In addition, for DRO-COX (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).

Following Keya et al. (2021), the final hyperparameter setting per dataset and per method is determined based on a preset rule that allows up to a 5% degradation in the validation set c-index from the classical Cox model (for the linear setting) or DeepSurv (for the nonlinear setting) while minimizing the validation set CI fairness metric.

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). As a reminder, we did not train using a GPU.

Appendix E. Additional Experiments

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

Hyperparameter tuning based on F_A instead of CI. The previous experimental results are based on hyperparameters chosen by minimizing the validation set CI fairness metric (while tolerating a small degradation in c-index). If instead of focusing on the CI fairness metric, we used F_A instead, then we get the results in Tables E.5, E.6, E.7, E.8, E.9, E.10, and E.11. In particular, our experimental findings from before remain the same except now our DRO-COX and DRO-COX (SPLIT) variants consistently achieve the best F_A scores (while often also scoring well on other fairness metrics) without too large of a drop in accuracy.

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 in evaluating F_G and CI) in Table E.12. 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 $\mathcal{L}_{\text{DRO-split}}(\theta, \eta, \mathcal{D}_1, \mathcal{D}_2)$ and $\mathcal{L}_{\text{DRO-split}}(\theta, \eta', \mathcal{D}_2, \mathcal{D}_1)$. Towards the end of Section 3.3, we said that an ap-

proach 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) that only minimizes $\mathcal{L}_{\text{DRO-split}}(\theta, \eta, \mathcal{D}_1, \mathcal{D}_2)$ 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 to evaluate F_G and CI. We report the resulting test set performance metrics in Table E.13. 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.

Table E.1: Test set scores on the FLC (gender) dataset, in the same format as Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.8032**	0.8176	**-6.3724**	0.1739	1.8787	0.5421	2.8355	1.7521	0.8610
		(0.0002)	(0.0005)	**(0.0011)**	(0.0004)	(0.0304)	(0.0299)	(0.0297)	(0.0266)	(0.0197)
	Cox_I(Keya et al.)	0.7932	0.8175	-6.6845	0.1368	0.3234	0.0273	0.9325	0.4277	1.6750
		(0.0083)	(0.0078)	(0.0786)	(0.0052)	(0.1098)	(0.0214)	(0.2562)	(0.1249)	(0.7969)
	Cox_I(R&P)	0.8028	0.8192	-6.4480	0.1588	0.8721	0.1921	1.9009	0.9884	0.8050
		(0.0012)	(0.0008)	(0.0149)	(0.0029)	(0.1203)	(0.0250)	(0.1098)	(0.0758	(0.0978)
	Cox_G(Keya et al.)	0.8011	**0.8205**	-6.4556	0.1619	1.1257	0.2749	2.0284	1.1430	0.7020
		(0.0015)	**(0.0015)**	(0.0544)	(0.0077)	(0.4887)	(0.2117)	(0.5407)	(0.4134)	(0.1081)
	Cox_G(R&P)	0.8023	0.8185	-6.4152	0.1646	1.1454	0.2616	2.2027	1.2032	0.7850
		(0.0009)	(0.0006	(0.0186)	(0.0026)	(0.1371)	(0.0716)	(0.1604)	(0.1175)	(0.0826)
	$\text{Cox}_\cap$(Keya et al.)	0.7868	0.8147	-6.7277	0.1400	0.2922	**0.0206**	0.6836	0.3321	0.4830
		(0.0018)	(0.0009)	(0.0043)	(0.0005)	(0.0067)	**(0.0100)**	(0.0207)	(0.0093)	(0.1020)
	DRO-COX	0.7605	0.7890	-6.9232	**0.1350**	**0.1168**	0.0324	**0.3397**	**0.1630**	0.3040
		(0.0096)	(0.0065)	(0.0240)	**(0.0003)**	**(0.0151)**	(0.0064)	**(0.0477)**	**(0.0209)**	(0.1569)
	DRO-COX (SPLIT)	0.7592	0.7900	-6.9098	0.1700	0.1342	0.0386	0.3744	0.1824	**0.1820**
		(0.0076)	(0.0055)	(0.0138)	(0.0004)	(0.0113)	(0.0067)	(0.0303)	(0.0145)	**(0.0795)**
Nonlinear	DeepSurv	**0.8070**	0.8247	**-6.3552**	0.1767	2.9691	0.7721	2.8800	2.2070	1.0760
		(0.0014)	(0.0026)	**(0.0052)**	(0.0018)	(1.2481)	(0.3225)	(0.0531)	(0.5149)	(0.1702)
	DeepSurv_I(Keya et al.)	0.7916	0.8175	-6.5516	0.1548	0.0839	0.0085	1.6023	0.5649	1.4610
		(0.0121)	(0.0120)	(0.1650)	(0.0176)	(0.0980)	(0.0120)	(0.7389)	(0.2187)	(0.7342)
	DeepSurv_I(R&P)	0.8067	**0.8252**	-6.3848	0.1729	0.0866	0.0316	2.4328	0.8504	1.0640
		(0.0041)	**(0.0053)**	(0.0769)	(0.0093)	(0.1374)	(0.0563)	(0.4052)	(0.0977)	(0.1408)
	DeepSurv_G(Keya et al.)	0.7964	0.8115	-6.5574	0.1576	0.4691	0.1348	1.8107	0.8048	0.9420
		(0.0117)	(0.0149)	(0.2021)	(0.0196)	(0.4532)	(0.1476)	(1.0544)	(0.4733)	(0.2229)
	DeepSurv_G(R&P)	0.8059	0.8242	-6.4062	0.1699	0.1877	0.0700	2.4171	0.8916	1.0750
		(0.0045)	(0.0057)	(0.0944)	(0.0118)	(0.1998)	(0.0872)	(0.5159)	(0.1079)	(0.1204)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7804	0.7965	-6.8087	0.1399	0.4481	0.1130	0.6093	0.3901	0.8440
		(0.0119)	(0.0158)	(0.0769)	(0.0086)	(0.5628)	(0.2517)	(0.3812)	(0.3790)	(0.2581)
	Deep DRO-COX	0.7699	0.7878	-6.9773	**0.1336**	**0.0661**	**0.0209**	**0.2362**	**0.1077**	**0.4870**
		(0.0147)	(0.0163)	(0.0474)	**(0.0004)**	**(0.0271)**	**(0.0105)**	**(0.1005)**	**(0.0454)**	**(0.2540)**
	Deep DRO-COX (SPLIT)	0.7650	0.7744	-6.8071	0.1703	0.4480	0.1991	0.7762	0.4744	0.5290
		(0.0024)	(0.0022)	(0.0091)	(0.0002)	(0.1050)	(0.0963)	(0.0992)	(0.1000)	(0.0908)

Table E.2: Test set scores on the SUPPORT (age) dataset, in the same format as Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.6025**	0.6163	**-6.8761**	0.2304	0.2113	0.1528	0.4490	0.2710	2.2240
		(0.0005)	(0.0010)	**(0.0010)**	(0.0015)	(0.0093)	(0.0059)	(0.0322)	(0.0128)	(0.1078)
	Cox_I(Keya et al.)	0.5820	0.5919	-6.9530	**0.2153**	0.0117	0.0117	0.0375	0.0203	1.3120
		(0.0116)	(0.0160)	(0.0122)	**(0.0076)**	(0.0229)	(0.0189)	(0.0561)	(0.0326)	(0.7623)
	Cox_I(R&P)	0.6020	**0.6163**	-6.8792	0.2285	0.1865	0.1321	0.3798	0.2329	2.1120
		(0.0010)	**(0.0018)**	(0.0018)	(0.0014)	(0.0122)	(0.0154)	(0.0461)	(0.0233)	(0.2653)
	Cox_G(Keya et al.)	0.5875	0.5963	-6.8870	0.2315	0.1925	0.0100	0.1981	0.1335	2.2030
		(0.0013)	(0.0020)	(0.0016)	(0.0014)	(0.0077)	(0.0038)	(0.0243	(0.0080)	(0.0986)
	Cox_G(R&P)	0.6018	0.6159	-6.8780	0.2296	0.2039	0.1577	0.4352	0.2656	2.1210
		(0.0008)	(0.0020)	(0.0014)	(0.0013)	(0.0089)	(0.0136)	(0.0385)	(0.0186)	(0.2863)
	$\text{Cox}_\cap$(Keya et al.)	0.5664	0.5663	-6.9132	0.2273	0.1291	0.0090	0.0688	0.0689	2.8030
		(0.0061)	(0.0078)	(0.0064)	(0.0016)	(0.0115)	(0.0038)	(0.0144)	(0.0091)	(0.2551)
	DRO-COX	0.5722	0.6068	-6.9399	0.2210	0.0340	0.0191	0.0654	0.0394	1.8310
		(0.0031)	(0.0041)	(0.0031)	(0.0010)	(0.0113)	(0.0060)	(0.0212)	(0.0126)	(0.2546)
	DRO-COX (SPLIT)	0.5501	0.5765	-6.9496	0.2253	**0.0002**	**0.0022**	**0.0075**	**0.0033**	**0.8520**
		(0.0104)	(0.0125)	(0.0002)	(0.0049)	**(0.0002)**	**(0.0010)**	**(0.0016)**	**(0.0008)**	**(0.4874)**
Nonlinear	DeepSurv	**0.6108**	**0.6327**	**-6.8754**	0.2417	0.4072	0.1897	0.4244	0.3404	2.1170
		(0.0029)	**(0.0045)**	**(0.0040)**	(0.0016)	(0.0369)	(0.0235)	(0.0573)	(0.0312)	(0.2107)
	DeepSurv_I(Keya et al.)	0.5950	0.6111	-6.9169	0.2316	**0.0234**	0.0186	0.4338	0.1586	1.6330
		(0.0116)	(0.0178)	(0.0195)	(0.0188)	**(0.0279)**	(0.0203)	(0.2972)	(0.0887)	(0.5036)
	DeepSurv_I(R&P)	0.6036	0.6223	-6.8859	0.2323	0.0752	0.0559	0.4616	0.1975	2.1030
		(0.0075)	(0.0116)	(0.0075)	(0.0083)	(0.0600)	(0.0429)	(0.2017)	(0.0424)	(0.2650)
	DeepSurv_G(Keya et al.)	0.5869	0.6043	-6.9159	0.2372	0.0738	**0.0052**	0.2937	0.1243	1.6760
		(0.0122)	(0.0155)	(0.0149)	(0.0131)	(0.0652)	**(0.0053)**	(0.2008)	(0.0521)	(0.4326)
	DeepSurv_G(R&P)	0.6039	0.6226	-6.8834	0.2322	0.0952	0.0738	0.4635	0.2108	2.1660
		(0.0089)	(0.0136)	(0.0096)	(0.0075)	(0.0590)	(0.0473)	(0.1841)	(0.0451)	(0.3318)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.5979	0.6131	-6.8813	0.2345	0.2182	0.0133	**0.0559**	0.0958	2.4300
		(0.0063)	(0.0090)	(0.0057)	(0.0036)	(0.0307)	(0.0038)	**(0.0150)**	(0.0135)	(0.2338)
	Deep DRO-COX	0.5833	0.6251	-6.9270	**0.2231**	0.0779	0.0278	0.0738	**0.0598**	0.7590
		(0.0088)	(0.0137)	(0.0053)	**(0.0015)**	(0.0153)	(0.0042)	(0.0100)	**(0.0068)**	(0.3395)
	Deep DRO-COX (SPLIT)	0.5448	0.5625	-6.9555	0.6390	0.1605	0.0442	0.1754	0.1267	**0.5710**
		(0.0015)	(0.0021)	(0.0012)	(0.0005)	(0.0030)	(0.0056)	(0.0062)	(0.0034)	**(0.1022)**

Table E.3: Test set scores on the SUPPORT (race) dataset, in the same format as Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.6025**	0.6163	**-6.8761**	0.2304	0.2113	0.0297	0.4490	0.2300	1.4160
		(0.0005)	(0.0010)	**(0.0010)**	(0.0015)	(0.0093)	(0.0130)	(0.0322)	(0.0085)	(0.0696)
	Cox_I(Keya et al.)	0.5905	0.6007	-6.9390	**0.2161**	0.0381	0.0148	0.0910	0.0479	1.1230
		(0.0086)	(0.0108)	(0.0214)	**(0.0054)**	(0.0383)	(0.0197)	(0.0846)	(0.0459)	(0.6621)
	Cox_I(R&P)	0.6024	**0.6165**	-6.8788	0.2287	0.1899	0.0273	0.3952	0.2042	1.3320
		(0.0010)	**(0.0020)**	(0.0024)	(0.0015)	(0.0115)	(0.0139)	(0.0358)	(0.0154)	(0.1742)
	Cox_G(Keya et al.)	0.6013	0.6149	-6.8797	0.2282	0.1883	**0.0051**	0.4723	0.2219	1.3610
		(0.0008)	(0.0011)	(0.0018)	(0.0017)	(0.0099)	**(0.0037)**	(0.0200)	(0.0080)	(0.0647)
	Cox_G(R&P)	0.6024	0.6165	-6.8780	0.2294	0.1981	0.0259	0.4163	0.2134	1.3350
		(0.0010)	(0.0022)	(0.0020)	(0.0013)	(0.0090)	(0.0149)	(0.0418)	(0.0163)	(0.1889)
	$\text{Cox}_\cap$(Keya et al.)	0.5681	0.5683	-6.9114	0.2271	0.1286	0.0167	0.0704	0.0719	1.4020
		(0.0079)	(0.0086)	(0.0086)	(0.0018)	(0.0135)	(0.0058)	(0.0162)	(0.0109)	(0.1743)
	DRO-COX	0.5735	0.6085	-6.9388	0.2210	**0.0378**	0.0099	**0.0730**	**0.0402**	0.4640
		(0.0018)	(0.0023)	(0.0007)	(0.0010)	**(0.0013)**	(0.0026)	**(0.0094)**	**(0.0041)**	(0.0790)
	DRO-COX (SPLIT)	0.5762	0.6107	-6.9374	0.5122	0.0412	0.0105	0.0809	0.0442	**0.3640**
		(0.0032)	(0.0026)	(0.0010)	(0.1007)	(0.0023)	(0.0044)	(0.0142)	(0.0051)	**(0.1975)**
Nonlinear	DeepSurv	**0.6108**	**0.6327**	**-6.8754**	0.2417	0.4072	0.0297	0.4244	0.2871	1.7440
		(0.0029)	**(0.0045)**	**(0.0040)**	(0.0016)	(0.0369)	(0.0121)	(0.0573)	(0.0251)	(0.2649)
	DeepSurv_I(Keya et al.)	0.5927	0.6078	-6.9212	0.2316	**0.0228**	0.0071	0.4557	0.1619	1.0380
		(0.0082)	(0.0104)	(0.0148)	(0.0166)	**(0.0241)**	(0.0100)	(0.3385)	(0.1051)	(0.5996)
	DeepSurv_I(R&P)	0.6078	0.6283	-6.8805	0.2374	0.0528	0.0144	0.4615	0.1762	1.6470
		(0.0067)	(0.0096)	(0.0106)	(0.0090)	(0.0530)	(0.0186)	(0.1031)	(0.0125)	(0.3917)
	DeepSurv_G(Keya et al.)	0.5941	0.6113	-6.9055	0.2369	0.1048	**0.0037**	0.3912	0.1666	1.2780
		(0.0145)	(0.0194)	(0.0219)	(0.0117)	(0.0782)	**(0.0022)**	(0.1044)	(0.0395)	(0.3894)
	DeepSurv_G(R&P)	0.6108	0.6327	-6.8776	0.2396	0.0505	0.0119	0.4820	0.1815	1.5720
		(0.0076)	(0.0112)	(0.0131)	(0.0086)	(0.0537)	(0.0193)	(0.0799)	(0.0128)	(0.2968)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.5992	0.6151	-6.8805	0.2357	0.2316	0.0269	**0.0687**	0.1091	1.4230
		(0.0072)	(0.0101)	(0.0066)	(0.0042)	(0.0459)	(0.0068)	**(0.0191)**	(0.0215)	(0.4286)
	Deep DRO-COX	0.5798	0.6193	-6.9278	**0.2234**	0.0898	0.0047	0.0777	**0.0574**	**0.7900**
		(0.0101)	(0.0166)	(0.0052)	**(0.0017)**	(0.0349)	(0.0034)	(0.0085)	**(0.0137)**	**(0.4283)**
	Deep DRO-COX (SPLIT)	0.5448	0.5625	-6.9555	0.6390	0.1605	0.0071	0.1754	0.1143	2.1690
		(0.0015)	(0.0021)	(0.0012)	(0.0005)	(0.0030)	(0.0024)	(0.0062)	(0.0031)	(0.0727)

Table E.4: Test set scores on the SEER (age) dataset, in the same format as Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	0.7409	0.7624	**-5.9427**	0.0964	0.6105	0.9037	0.6750	0.7297	2.5640
		(0.0016)	(0.0017)	**(0.0034)**	(0.0008)	(0.0307)	(0.1308)	(0.0630)	(0.0656)	(0.3531)
	Cox_I(Keya et al.)	0.7118	0.7345	-6.1539	0.0896	0.2045	0.2900	0.3802	0.2916	1.4310
		(0.0277)	(0.0287)	(0.0716)	(0.0014)	(0.0587)	(0.1183)	(0.1431)	(0.0994)	(1.2077)
	Cox_I(R&P)	**0.7425**	**0.7644**	-5.9829	0.0920	0.3761	0.4582	0.4125	0.4156	2.6950
		(0.0037)	**(0.0037)**	(0.0188)	(0.0010)	(0.0516)	(0.1339)	(0.0930)	(0.0907)	(0.4455)
	Cox_G(Keya et al.)	0.7263	0.7503	-5.9968	0.0946	0.5094	**0.0480**	0.5041	0.3538	3.2960
		(0.0261)	(0.0284)	(0.0868)	(0.0021)	(0.1122)	**(0.0390)**	(0.1042)	(0.0662)	(1.0958)
	Cox_G(R&P)	0.7401	0.7609	-5.9733	0.0930	0.4278	0.7888	0.6221	0.6129	3.0080
		(0.0041)	(0.0041)	(0.0193)	(0.0010)	(0.0488)	(0.0564)	(0.0377	(0.0320)	(0.3603)
	$\text{Cox}_\cap$(Keya et al.)	0.7323	0.7565	-5.9867	0.0954	0.5376	0.0599	0.7183	0.4386	3.0430
		(0.0179)	(0.0199)	(0.0679)	(0.0014)	(0.0738)	(0.0709)	(0.2117)	(0.1044)	(0.9716)
	DRO-COX	0.6975	0.7152	-6.2201	**0.0885**	**0.1127**	0.1267	**0.2013**	**0.1469**	**0.3600**
		(0.0125)	(0.0165)	(0.0547)	**(0.0007)**	**(0.0681)**	(0.0718)	**(0.0873)**	**(0.0752)**	**(0.4190)**
	DRO-COX (SPLIT)	0.7026	0.7210	-6.5385	0.0955	8.8123	16.6719	0.7097	8.7313	0.4560
		(0.0130)	(0.0167)	(0.7038)	(0.0051)	(18.6473)	(33.4878)	(0.9603)	(17.1007)	(0.3810)
Nonlinear	DeepSurv	**0.7488**	**0.7729**	**-5.9582**	0.0966	0.3686	0.3530	0.4734	0.3983	2.0450
		(0.0103)	**(0.0105)**	**(0.0538)**	(0.0047)	(0.0959)	(0.0898)	(0.1203)	(0.0900)	(0.8414)
	DeepSurv_I(Keya et al.)	0.7126	0.7363	-6.1757	0.0985	**0.0688**	0.0697	0.8667	0.3351	2.2580
		(0.0224)	(0.0217)	(0.1692)	(0.0094)	**(0.0378)**	(0.0431)	(0.5610)	(0.1711)	(2.1372)
	DeepSurv_I(R&P)	0.7375	0.7603	-6.0034	0.0947	0.1074	0.1226	0.6459	0.2920	2.4260
		(0.0114)	(0.0113)	(0.0947)	(0.0046)	(0.0535)	(0.0618)	(0.4236)	(0.1150)	(1.9244)
	DeepSurv_G(Keya et al.)	0.7324	0.7587	-6.0709	0.0991	0.2526	**0.0210**	0.6425	0.3053	2.7790
		(0.0234)	(0.0250)	(0.2042)	(0.0066)	(0.1423)	**(0.0138)**	(0.2721)	(0.0755)	(2.0384)
	DeepSurv_G(R&P)	0.7382	0.7603	-5.9906	0.0964	0.1149	0.1639	0.7670	0.3486	2.6600
		(0.0107)	(0.0113)	(0.0850)	(0.0047)	(0.0678)	(0.1067)	(0.3615)	(0.0935)	(1.8288)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7303	0.7579	-6.0238	0.0955	0.4060	0.0355	0.3632	0.2682	2.0270
		(0.0134)	(0.0119)	(0.0795)	(0.0057)	(0.2202)	(0.0299)	(0.1883)	(0.1138)	(1.8428)
	Deep DRO-COX	0.7178	0.7390	-6.2029	**0.0878**	0.0763	0.0996	**0.1772**	**0.1177**	**0.3040**
		(0.0157)	(0.0133)	(0.0365)	**(0.0004)**	(0.0195)	(0.0445)	**(0.0206)**	**(0.0230)**	**(0.2281)**
	Deep DRO-COX (SPLIT)	0.6834	0.7105	-6.1861	0.1023	0.2288	0.0174	0.3279	0.1913	1.3880
		(0.0128)	(0.0132)	(0.0412)	(0.0004)	(0.0884)	(0.0209)	(0.0517)	(0.0514)	(0.6979)

Table E.5: Test set scores on the FLC (age) dataset when hyperparameter tuning is based on F_A. The format of this table is the same that of Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.8032** (**0.0002**)	0.8176 (0.0005)	**-6.3724** (**0.0011**)	0.1739 (0.0004)	1.8787 (0.0304)	3.0282 (0.0469)	2.8355 (0.0297)	2.5808 (0.0332)	0.5350 (0.0413)
	Cox_I(Keya et al.)	0.7852 (0.0220)	0.8071 (0.0240)	-6.7714 (0.0360)	0.1333 (0.0034)	0.2233 (0.0161)	0.3809 (0.0317)	0.7013 (0.0406)	0.4352 (0.0222)	0.6500 (0.6830)
	Cox_I(R&P)	0.8009 (0.0004)	**0.8182** (**0.0005**)	-6.4775 (0.0041)	0.1564 (0.0006)	0.7615 (0.0223)	1.2662 (0.0342)	1.7325 (0.0318)	1.2534 (0.0285)	0.1430 (0.0372)
	Cox_G(Keya et al.)	0.7862 (0.0133)	0.8134 (0.0077)	-6.7099 (0.0883)	0.1413 (0.0035)	0.3323 (0.1033)	0.4855 (0.1858)	0.9801 (0.2431)	0.5993 (0.1768)	0.5360 (0.3888)
	Cox_G(R&P)	0.8012 (0.0004)	0.8181 (0.0005)	-6.4299 (0.0024)	0.1637 (0.0006)	1.0730 (0.0282)	1.7702 (0.0422)	2.0727 (0.0299)	1.6386 (0.0320)	**0.1420** (**0.0483**)
	$\text{Cox}_\cap$(Keya et al.)	0.7866 (0.0014)	0.8147 (0.0008)	-6.7276 (0.0043)	0.1400 (0.0005)	0.2923 (0.0067)	0.4118 (0.0168)	0.6794 (0.0206)	0.4611 (0.0133)	1.0670 (0.1285)
	DRO-COX	0.7958 (0.0049)	0.8169 (0.0035)	-7.0859 (0.0009)	**0.1330** (**0.0002**)	**0.0015** (**0.0004**)	**0.0086** (**0.0007**)	**0.0227** (**0.0030**)	**0.0110** (**0.0013**)	0.1620 (0.1132)
	DRO-COX (SPLIT)	0.7963 (0.0045)	0.8168 (0.0030)	-7.0856 (0.0009)	0.1390 (0.0008)	0.0016 (0.0004)	0.0089 (0.0008)	0.0232 (0.0031)	0.0113 (0.0013)	0.2340 (0.1237)
Nonlinear	DeepSurv	**0.8070** (**0.0014**)	**0.8247** (**0.0026**)	**-6.3552** (**0.0052**)	0.1767 (0.0018)	2.9691 (1.2481)	4.6647 (1.9185)	2.8800 (0.0531)	3.5046 (1.0506)	0.2940 (0.2147)
	DeepSurv_I(Keya et al.)	0.7844 (0.0104)	0.8090 (0.0103)	-6.7225 (0.1247)	0.1373 (0.0121)	0.1533 (0.0530)	0.2511 (0.0884)	0.7680 (0.3014)	0.3908 (0.0640)	0.3990 (0.2614)
	DeepSurv_I(R&P)	0.8017 (0.0063)	0.8197 (0.0077)	-6.5098 (0.1202)	0.1570 (0.0146)	0.2705 (0.2162)	0.3973 (0.3113)	1.7136 (0.5151)	0.7938 (0.0194)	0.1670 (0.0804)
	DeepSurv_G(Keya et al.)	0.8019 (0.0117)	0.8220 (0.0099)	-6.4162 (0.1373)	0.5376 (0.2037)	**0.0187** (**0.0560**)	**0.0226** (**0.0649**)	0.8407 (0.1512)	0.2940 (0.0574)	**0.2480** (**0.1638**)
	DeepSurv_G(R&P)	0.8025 (0.0055)	0.8198 (0.0071)	-6.4924 (0.1125)	0.1586 (0.0139)	0.3746 (0.2788)	0.5386 (0.3891)	1.9252 (0.5881)	0.9462 (0.0461)	0.2760 (0.0732)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7751 (0.0018)	0.7893 (0.0022)	-6.8458 (0.0031)	0.1357 (0.0002)	0.1688 (0.0035)	0.2412 (0.0051)	0.4633 (0.0106)	0.2911 (0.0062)	0.4300 (0.1091)
	Deep DRO-COX	0.7726 (0.0137)	0.7917 (0.0148)	-7.0406 (0.0118)	**0.1331** (**0.0002**)	0.0269 (0.0055)	0.0447 (0.0093)	**0.1031** (**0.0213**)	**0.0582** (**0.0118**)	2.3680 (0.5542)
	Deep DRO-COX (SPLIT)	0.7629 (0.0064)	0.7719 (0.0076)	-6.8131 (0.0199)	0.1703 (0.0002)	0.4347 (0.1214)	0.5184 (0.0922)	0.7508 (0.1417)	0.5680 (0.1176)	2.8490 (0.2435)

Table E.6: Test set scores on the FLC (gender) dataset when hyperparameter tuning is based on F_A. The format of this table is the same that of Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.8032** (**0.0002**)	0.8176 (0.0005)	**-6.3724** (**0.0011**)	0.1739 (0.0004)	1.8787 (0.0304)	0.5421 (0.0299)	2.8355 (0.0297)	1.7521 (0.0266)	0.8610 (0.0197)
	Cox_I(Keya et al.)	0.7918 (0.0078)	0.8158 (0.0068)	-6.7585 (0.0202)	0.1335 (0.0035)	0.2282 (0.0164)	0.0190 (0.0111)	0.6974 (0.0372)	0.3149 (0.0194)	1.6720 (0.8430)
	Cox_I(R&P)	0.8009 (0.0004)	0.8182 (0.0005)	-6.4775 (0.0041)	0.1564 (0.0006)	0.7615 (0.0223)	0.1411 (0.0149)	1.7325 (0.0318)	0.8784 (0.0213)	0.6950 (0.0246)
	Cox_G(Keya et al.)	0.8002 (0.0004)	**0.8215** (**0.0004**)	-6.4914 (0.0028)	0.1568 (0.0004)	0.8051 (0.0124)	0.1358 (0.0129)	1.6709 (0.0194)	0.8706 (0.0136)	0.6310 (0.0070)
	Cox_G(R&P)	0.8011 (0.0004)	0.8185 (0.0005)	-6.4437 (0.0029)	0.1613 (0.0005)	0.9634 (0.0248)	0.1845 (0.0211)	1.9781 (0.0322)	1.0420 (0.0235)	0.7120 (0.0166)
	$\text{Cox}_\cap$(Keya et al.)	0.7880 (0.0017)	0.8153 (0.0009)	-6.7275 (0.0039)	0.1403 (0.0005)	0.2925 (0.0054)	0.0182 (0.0083)	0.6967 (0.0178)	0.3358 (0.0066)	**0.4340** (**0.0898**)
	DRO-COX	0.7958 (0.0049)	0.8169 (0.0035)	-7.0859 (0.0009)	**0.1330** (**0.0002**)	**0.0015** (**0.0004**)	0.0034 (0.0010)	**0.0227** (**0.0030**)	**0.0092** (**0.0014**)	1.0780 (0.0739)
	DRO-COX (SPLIT)	0.7963 (0.0045)	0.8168 (0.0030)	-7.0856 (0.0009)	0.1390 (0.0008)	0.0016 (0.0004)	**0.0033** (**0.0010**)	0.0232 (0.0031)	0.0094 (0.0015)	1.0250 (0.1376)
Nonlinear	DeepSurv	**0.8070** (**0.0014**)	0.8247 (0.0026)	**-6.3552** (**0.0052**)	0.1767 (0.0018)	2.9691 (1.2481)	0.7721 (0.3225)	2.8800 (0.0531)	2.2070 (0.5149)	1.0760 (0.1702)
	DeepSurv_I(Keya et al.)	0.7825 (0.0068)	0.8066 (0.0073)	-6.7680 (0.0093)	0.1337 (0.0034)	0.1653 (0.0049)	0.0158 (0.0119)	0.6577 (0.0266)	0.2796 (0.0136)	1.4790 (0.7038)
	DeepSurv_I(R&P)	0.7965 (0.0003)	0.8134 (0.0004)	-6.6064 (0.0050)	0.1453 (0.0002)	0.4452 (0.0183)	0.1675 (0.0206)	1.2888 (0.0275)	0.6338 (0.0217)	1.0730 (0.0168)
	DeepSurv_G(Keya et al.)	0.7844 (0.0008)	0.7964 (0.0013)	-6.7635 (0.0015)	0.1378 (0.0002)	0.2658 (0.0112)	0.0727 (0.0146)	0.7370 (0.0258)	0.3585 (0.0172)	0.7450 (0.0102)
	DeepSurv_G(R&P)	0.7974 (0.0003)	0.8140 (0.0003)	-6.5926 (0.0067)	0.1468 (0.0002)	0.5502 (0.0252)	0.2312 (0.0291)	1.4168 (0.0338)	0.7328 (0.0289)	1.0630 (0.0179)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7751 (0.0018)	0.7893 (0.0022)	-6.8458 (0.0031)	0.1357 (0.0002)	0.1688 (0.0035)	0.0252 (0.0037)	0.4633 (0.0106)	0.2191 (0.0058)	0.7400 (0.0671)
	Deep DRO-COX	0.7747 (0.0128)	0.7938 (0.0142)	-7.0412 (0.0114)	**0.1331** (**0.0002**)	**0.0267** (**0.0055**)	**0.0098** (**0.0033**)	**0.1028** (**0.0211**)	**0.0464** (**0.0095**)	1.2430 (0.9204)
	Deep DRO-COX (SPLIT)	0.7629 (0.0064)	0.7719 (0.0076)	-6.8131 (0.0199)	0.1703 (0.0002)	0.4347 (0.1214)	0.1877 (0.1098)	0.7508 (0.1417)	0.4577 (0.1231)	**0.5970** (**0.2383**)

Table E.7: Test set scores on the SUPPORT (age) dataset when hyperparameter tuning is based on F_A. The format of this table is the same that of Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.6025** (**0.0005**)	0.6163 (0.0010)	**-6.8761** (**0.0010**)	**0.2304** (**0.0015**)	0.2113 (0.0093)	0.1528 (0.0059)	0.4490 (0.0322)	0.2710 (0.0128)	2.2240 (0.1078)
	Cox_I(Keya et al.)	0.5831 (0.0100)	0.5932 (0.0143)	-6.9594 (0.0054)	**0.2147** (**0.0063**)	0.0011 (0.0001)	0.0038 (0.0010)	0.0180 (0.0036)	0.0076 (0.0015)	1.4280 (0.8245)
	Cox_I(R&P)	**0.6027** (**0.0010**)	**0.6173** (**0.0012**)	-6.8797 (0.0016)	0.2277 (0.0011)	0.1754 (0.0048)	0.1198 (0.0075)	0.3553 (0.0299)	0.2168 (0.0118)	2.1240 (0.2217)
	Cox_G(Keya et al.)	0.5863 (0.0008)	0.5948 (0.0014)	-6.8903 (0.0009)	0.2292 (0.0010)	0.1686 (0.0034)	0.0096 (0.0024)	0.1810 (0.0195)	0.1197 (0.0064)	2.2160 (0.0981)
	Cox_G(R&P)	0.6023 (0.0008)	0.6167 (0.0014)	-6.8777 (0.0015)	0.2291 (0.0011)	0.1976 (0.0051)	0.1522 (0.0095)	0.4260 (0.0318)	0.2586 (0.0130)	2.1330 (0.2387)
	$\text{Cox}_\cap$(Keya et al.)	0.5631 (0.0070)	0.5620 (0.0084)	-6.9163 (0.0069)	0.2264 (0.0017)	0.1183 (0.0124)	0.0092 (0.0039)	0.0604 (0.0144)	0.0627 (0.0096)	2.8350 (0.2498)
	DRO-COX	0.5438 (0.0080)	0.5754 (0.0080)	-6.9496 (0.0003)	0.2213 (0.0010)	0.0009 (0.0002)	0.0026 (0.0005)	0.0067 (0.0010)	0.0034 (0.0004)	1.6190 (0.3069)
	DRO-COX (SPLIT)	0.5501 (0.0104)	0.5765 (0.0125)	-6.9496 (0.0002)	0.2253 (0.0049)	**0.0002** (**0.0002**)	**0.0022** (**0.0010**)	**0.0075** (**0.0016**)	**0.0033** (**0.0008**)	**0.8520** (**0.4874**)
Nonlinear	DeepSurv	0.6108 (0.0029)	**0.6327** (**0.0045**)	**-6.8754** (**0.0040**)	0.2417 (0.0016)	0.4072 (0.0369)	0.1897 (0.0235)	0.4244 (0.0573)	0.3404 (0.0312)	2.1170 (0.2107)
	DeepSurv_I(Keya et al.)	0.5827 (0.0111)	0.5940 (0.0114)	-6.9337 (0.0214)	**0.2177** (**0.0083**)	0.0277 (0.0138)	0.0212 (0.0096)	0.2499 (0.1275)	0.0996 (0.0432)	1.3570 (0.5989)
	DeepSurv_I(R&P)	0.6019 (0.0055)	0.6197 (0.0081)	-6.8865 (0.0085)	0.2319 (0.0083)	0.0791 (0.0534)	0.0572 (0.0379)	0.4050 (0.1156)	0.1804 (0.0104)	2.0060 (0.2204)
	DeepSurv_G(Keya et al.)	0.5825 (0.0113)	0.5978 (0.0129)	-6.9086 (0.0177)	0.2293 (0.0080)	0.1098 (0.0492)	**0.0050** (**0.0036**)	0.1620 (0.0722)	0.0923 (0.0244)	1.6950 (0.3622)
	DeepSurv_G(R&P)	**0.6127** (**0.0043**)	0.6359 (0.0064)	-6.8792 (0.0086)	0.2438 (0.0041)	**0.0084** (**0.0011**)	0.0073 (0.0011)	0.5414 (0.1291)	0.1857 (0.0434)	2.0580 (0.3551)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.5912 (0.0012)	0.6037 (0.0022)	-6.8876 (0.0015)	0.2309 (0.0011)	0.1867 (0.0071)	0.0134 (0.0033)	0.0903 (0.0079)	0.0968 (0.0040)	2.4750 (0.1695)
	Deep DRO-COX	0.5833 (0.0088)	0.6251 (0.0137)	-6.9270 (0.0053)	0.2231 (0.0015)	0.0779 (0.0153)	0.0278 (0.0042)	**0.0738** (**0.0100**)	**0.0598** (**0.0068**)	0.7590 (0.3395)
	Deep DRO-COX (SPLIT)	0.5448 (0.0015)	0.5625 (0.0021)	-6.9555 (0.0012)	0.6390 (0.0005)	0.1605 (0.0030)	0.0442 (0.0056)	0.1754 (0.0062)	0.1267 (0.0034)	**0.5710** (**0.1022**)

Table E.8: Test set scores on the SUPPORT (race) dataset when hyperparameter tuning is based on F_A. The format of this table is the same that of Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	0.6025 (0.0005)	0.6163 (0.0010)	**-6.8761** (**0.0010**)	0.2304 (0.0015)	0.2113 (0.0093)	0.0297 (0.0130)	0.4490 (0.0322)	0.2300 (0.0085)	1.4160 (0.0696)
	Cox_I(Keya et al.)	0.5831 (0.0100)	0.5932 (0.0143)	-6.9594 (0.0054)	**0.2147** (**0.0063**)	**0.0001** (**0.0001**)	0.0023 (0.0016)	0.0080 (0.0036)	0.0034 (0.0017)	1.2580 (0.5410)
	Cox_I(R&P)	**0.6027** (**0.0010**)	**0.6173** (**0.0012**)	-6.8797 (0.0016)	0.2277 (0.0011)	0.1754 (0.0048)	0.0230 (0.0109)	0.3553 (0.0299)	0.1846 (0.0081)	1.3150 (0.1813)
	Cox_G(Keya et al.)	0.6011 (0.0006)	0.6147 (0.0011)	-6.8802 (0.0009)	0.2279 (0.0009)	0.1846 (0.0030)	0.0046 (0.0032)	0.4692 (0.0178)	0.2195 (0.0054)	1.3610 (0.0650)
	Cox_G(R&P)	0.6026 (0.0009)	0.6171 (0.0013)	-6.8781 (0.0016)	0.2286 (0.0011)	0.1884 (0.0048)	0.0221 (0.0123)	0.3898 (0.0300)	0.2001 (0.0084)	1.3240 (0.1902)
	$\text{Cox}_\cap$(Keya et al.)	0.5631 (0.0070)	0.5620 (0.0084)	-6.9163 (0.0069)	0.2264 (0.0017)	0.1183 (0.0124)	0.0154 (0.0056)	0.0604 (0.0144)	0.0647 (0.0101)	1.3670 (0.1406)
	DRO-COX	0.5438 (0.0080)	0.5754 (0.0080)	-6.9496 (0.0003)	0.2213 (0.0010)	0.0009 (0.0002)	**0.0011** (**0.0004**)	**0.0067** (**0.0010**)	**0.0029** (**0.0004**)	**0.2110** (**0.1652**)
	DRO-COX (SPLIT)	0.5501 (0.0104)	0.5765 (0.0125)	-6.9496 (0.0002)	0.2253 (0.0049)	0.0002 (0.0002)	0.0011 (0.0005)	0.0075 (0.0016)	0.0029 (0.0005)	0.2530 (0.2378)
Nonlinear	DeepSurv	**0.6108** (**0.0029**)	**0.6327** (**0.0045**)	**-6.8754** (**0.0040**)	0.2417 (0.0016)	0.4072 (0.0369)	0.0297 (0.0121)	0.4244 (0.0573)	0.2871 (0.0251)	1.7440 (0.2649)
	DeepSurv_I(Keya et al.)	0.5827 (0.0111)	0.5940 (0.0114)	-6.9337 (0.0214)	**0.2177** (**0.0083**)	**0.0277** (**0.0138**)	0.0077 (0.0073)	0.1299 (0.1275)	0.0551 (0.0429)	**0.9270** (**0.4994**)
	DeepSurv_I(R&P)	0.6020 (0.0065)	0.6195 (0.0090)	-6.8846 (0.0109)	0.2294 (0.0065)	0.0979 (0.0390)	0.0298 (0.0184)	0.3598 (0.0879)	0.1625 (0.0138)	1.4500 (0.2005)
	DeepSurv_G(Keya et al.)	0.5798 (0.0086)	0.5911 (0.0113)	-6.9148 (0.0049)	0.2260 (0.0075)	0.0888 (0.0166)	**0.0021** (**0.0014**)	0.2776 (0.0710)	0.1228 (0.0212)	0.9990 (0.2159)
	DeepSurv_G(R&P)	0.6055 (0.0073)	0.6252 (0.0114)	-6.8861 (0.0092)	0.2366 (0.0097)	0.0567 (0.0591)	0.0194 (0.0208)	0.4703 (0.1506)	0.1821 (0.0382)	1.5130 (0.3131)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.5912 (0.0012)	0.6037 (0.0022)	-6.8876 (0.0015)	0.2309 (0.0011)	0.1867 (0.0071)	0.0223 (0.0036)	0.0903 (0.0079)	0.0998 (0.0039)	1.1590 (0.1338)
	Deep DRO-COX	0.5833 (0.0088)	0.6251 (0.0137)	-6.9270 (0.0053)	0.2231 (0.0015)	0.0779 (0.0153)	0.0054 (0.0025)	**0.0738** (**0.0100**)	**0.0524** (**0.0052**)	1.6590 (0.3733)
	Deep DRO-COX (SPLIT)	0.5448 (0.0015)	0.5625 (0.0021)	-6.9555 (0.0012)	0.6390 (0.0005)	0.1605 (0.0030)	0.0071 (0.0024)	0.1754 (0.0062)	0.1143 (0.0031)	2.1690 (0.0727)

Table E.9: Test set scores on the SUPPORT (gender) dataset when hyperparameter tuning is based on F_A. The format of this table is the same that of Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	0.6025 (0.0005)	0.6163 (0.0010)	**-6.8761 (0.0010)**	0.2304 (0.0015)	0.2113 (0.0093)	0.0439 (0.0052)	0.4490 (0.0322)	0.2347 (0.0127)	1.4300 (0.0654)
	Cox_I(Keya et al.)	0.5831 (0.0100)	0.5932 (0.0143)	-6.9594 (0.0054)	**0.2147 (0.0063)**	**0.0001 (0.0001)**	0.0012 (0.0004)	0.0081 (0.0036)	0.0031 (0.0013)	1.1110 (0.7139)
	Cox_I(R&P)	**0.6026 (0.0011)**	**0.6173 (0.0012)**	-6.8800 (0.0021)	0.2277 (0.0011)	0.1759 (0.0047)	0.0280 (0.0132)	0.3591 (0.0282)	0.1877 (0.0117)	1.3800 (0.1137)
	Cox_G(Keya et al.)	0.6024 (0.0006)	0.6171 (0.0010)	-6.8791 (0.0009)	0.2284 (0.0010)	0.1890 (0.0027)	0.0031 (0.0017)	0.3927 (0.0150)	0.1949 (0.0049)	1.4360 (0.0674)
	Cox_G(R&P)	0.6025 (0.0009)	0.6170 (0.0014)	-6.8779 (0.0020)	0.2291 (0.0011)	0.1953 (0.0052)	0.0340 (0.0148)	0.4029 (0.0284)	0.2107 (0.0125)	1.3960 (0.1148)
	$\text{Cox}_\cap$(Keya et al.)	0.5631 (0.0070)	0.5620 (0.0084)	-6.9163 (0.0069)	0.2264 (0.0017)	0.1183 (0.0124)	0.0076 (0.0023)	0.0604 (0.0144)	0.0621 (0.0089)	0.8650 (0.2958)
	DRO-COX	0.5438 (0.0080)	0.5754 (0.0080)	-6.9496 (0.0003)	0.2213 (0.0010)	0.0009 (0.0002)	**0.0011 (0.0004)**	**0.0067 (0.0010)**	**0.0029 (0.0004)**	**0.2110 (0.1652)**
	DRO-COX (SPLIT)	0.5501 (0.0104)	0.5765 (0.0125)	-6.9496 (0.0002)	0.2253 (0.0049)	0.0002 (0.0002)	0.0011 (0.0005)	0.0075 (0.0016)	0.0029 (0.0005)	0.2530 (0.2378)
Nonlinear	DeepSurv	0.6108 (0.0029)	0.6327 (0.0045)	**-6.8754 (0.0040)**	0.2417 (0.0016)	0.4072 (0.0369)	0.0570 (0.0180)	0.4244 (0.0573)	0.2962 (0.0283)	1.6220 (0.3303)
	DeepSurv_I(Keya et al.)	0.5827 (0.0111)	0.5940 (0.0114)	-6.9337 (0.0214)	**0.2177 (0.0083)**	**0.0277 (0.0138)**	0.0089 (0.0045)	0.1599 (0.1275)	0.0655 (0.0427)	**1.3560 (0.7501)**
	DeepSurv_I(R&P)	0.5988 (0.0037)	0.6152 (0.0055)	-6.8900 (0.0051)	0.2271 (0.0052)	0.1097 (0.0359)	0.0333 (0.0111)	0.3396 (0.0843)	0.1609 (0.0130)	1.7860 (0.1170)
	DeepSurv_G(Keya et al.)	0.5850 (0.0089)	0.5992 (0.0122)	-6.9154 (0.0022)	0.2287 (0.0105)	0.0883 (0.0145)	**0.0026 (0.0022)**	0.2509 (0.0918)	0.1139 (0.0264)	1.8730 (0.6485)
	DeepSurv_G(R&P)	0.6053 (0.0074)	0.6251 (0.0116)	-6.8834 (0.0095)	0.2358 (0.0090)	0.0653 (0.0590)	0.0195 (0.0204)	0.4398 (0.0799)	0.1749 (0.0124)	1.5780 (0.2560)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.5912 (0.0012)	0.6037 (0.0022)	-6.8876 (0.0015)	0.2309 (0.0011)	0.1867 (0.0071)	0.0029 (0.0017)	0.0903 (0.0079)	0.0933 (0.0033)	1.5390 (0.1303)
	Deep DRO-COX	0.5833 (0.0088)	0.6251 (0.0137)	-6.9270 (0.0053)	0.2231 (0.0015)	0.0779 (0.0153)	0.0054 (0.0025)	**0.0738 (0.0100)**	**0.0524 (0.0052)**	1.6590 (0.3733)
	Deep DRO-COX (SPLIT)	0.5448 (0.0015)	0.5625 (0.0021)	-6.9555 (0.0012)	0.6390 (0.0005)	0.1605 (0.0030)	0.0071 (0.0024)	0.1754 (0.0062)	0.1143 (0.0031)	2.1690 (0.0727)

Table E.10: Test set scores on the SEER (age) dataset when hyperparameter tuning is based on F_A. The format of this table is the same that of Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.7409 (0.0016)**	**0.7624 (0.0017)**	**-5.9427 (0.0034)**	0.0964 (0.0008)	0.6105 (0.0307)	0.9037 (0.1308)	0.6750 (0.0630)	0.7297 (0.0656)	2.5640 (0.3531)
	Cox_I(Keya et al.)	0.7229 (0.0177)	0.7473 (0.0207)	-6.1913 (0.0250)	0.0885 (0.0009)	0.1299 (0.0177)	0.1312 (0.0546)	0.1824 (0.0765)	0.1479 (0.0432)	1.4070 (1.0039)
	Cox_I(R&P)	0.7293 (0.0279)	0.7496 (0.0315)	-6.0515 (0.0681)	0.0902 (0.0005)	0.2738 (0.0303)	0.2978 (0.0528)	0.3235 (0.0510)	0.2984 (0.0291)	2.3570 (0.9789)
	Cox_G(Keya et al.)	0.7192 (0.0293)	0.7429 (0.0319)	-6.0195 (0.0962)	0.0941 (0.0023)	0.4794 (0.1267)	0.0425 (0.0339)	0.4876 (0.1023)	0.3365 (0.0685)	2.9950 (1.3371)
	Cox_G(R&P)	0.7124 (0.0317)	0.7293 (0.0355)	-6.0682 (0.0920)	0.0913 (0.0012)	0.3188 (0.0771)	0.5823 (0.2186)	0.5447 (0.0853)	0.4819 (0.1248)	2.1690 (1.4271)
	$\text{Cox}_\cap$(Keya et al.)	0.7143 (0.0288)	0.7365 (0.0320)	-6.0508 (0.1053)	0.0942 (0.0020)	0.4658 (0.1188)	0.0498 (0.0434)	0.5179 (0.3066)	0.3445 (0.1445)	2.7750 (0.9141)
	DRO-COX	0.7058 (0.0165)	0.7288 (0.0173)	-6.2914 (0.0391)	**0.0877 (0.0004)**	**0.0207 (0.0296)**	**0.0292 (0.0313)**	**0.0871 (0.0184)**	**0.0456 (0.0262)**	**0.8650 (0.4212)**
	DRO-COX (SPLIT)	0.7040 (0.0181)	0.7274 (0.0165)	-6.2806 (0.0723)	0.0889 (0.0035)	0.0357 (0.0746)	0.0356 (0.1103)	0.1161 (0.0990)	0.0625 (0.0946)	1.2320 (0.6788)
Nonlinear	DeepSurv	**0.7488 (0.0103)**	**0.7729 (0.0105)**	**-5.9582 (0.0538)**	0.0966 (0.0047)	0.3686 (0.0959)	0.3530 (0.0898)	0.4734 (0.1203)	0.3983 (0.0900)	2.0450 (0.8414)
	DeepSurv_I(Keya et al.)	0.7003 (0.0272)	0.7222 (0.0309)	-6.1838 (0.1060)	0.0925 (0.0076)	0.0807 (0.0267)	0.0694 (0.0345)	0.4122 (0.3883)	0.1875 (0.1238)	3.3150 (2.5298)
	DeepSurv_I(R&P)	0.7304 (0.0104)	0.7535 (0.0097)	-6.0678 (0.0928)	0.0906 (0.0036)	0.1236 (0.0122)	0.1236 (0.0229)	0.3603 (0.1503)	0.2025 (0.0541)	1.3020 (0.6967)
	DeepSurv_G(Keya et al.)	0.7300 (0.0070)	0.7571 (0.0069)	-6.0645 (0.0507)	0.0899 (0.0026)	0.2102 (0.0568)	0.0142 (0.0085)	0.3246 (0.0881)	0.1830 (0.0501)	1.3520 (0.3187)
	DeepSurv_G(R&P)	0.7298 (0.0110)	0.7528 (0.0105)	-6.0705 (0.0817)	0.0914 (0.0051)	0.1250 (0.0341)	0.1376 (0.0428)	0.4288 (0.2521)	0.2305 (0.0743)	1.5580 (1.0975)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7151 (0.0040)	0.7444 (0.0042)	-6.1133 (0.0086)	0.0889 (0.0004)	0.1630 (0.0086)	**0.0117 (0.0081)**	0.2326 (0.0229)	0.1358 (0.0114)	**0.5870 (0.2585)**
	Deep DRO-COX	0.6889 (0.0277)	0.7029 (0.0335)	-6.2345 (0.0602)	**0.0878 (0.0007)**	**0.0559 (0.0326)**	0.0511 (0.0254)	**0.1303 (0.0298)**	**0.0791 (0.0217)**	2.8040 (1.0324)
	Deep DRO-COX (SPLIT)	0.6829 (0.0142)	0.7099 (0.0149)	-6.1870 (0.0426)	0.1023 (0.0004)	0.2276 (0.0901)	0.0157 (0.0212)	0.3261 (0.0554)	0.1898 (0.0537)	1.4970 (0.6481)

Table E.11: Test set scores on the SEER (race) dataset when hyperparameter tuning is based on F_A. The format of this table is the same that of Table 2.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	Cox	**0.7409 (0.0016)**	**0.7624 (0.0017)**	**-5.9427 (0.0034)**	0.0964 (0.0008)	0.6105 (0.0307)	0.1183 (0.0645)	0.6750 (0.0630)	0.4679 (0.0437)	1.1880 (0.1742)
	Cox_I(Keya et al.)	0.7224 (0.0176)	0.7471 (0.0208)	-6.1914 (0.0252)	0.0885 (0.0009)	0.1292 (0.0181)	0.0318 (0.0280)	0.1828 (0.0760)	0.1146 (0.0309)	2.1470 (1.6144)
	Cox_I(R&P)	0.7365 (0.0224)	0.7580 (0.0252)	-6.0328 (0.0551)	0.0903 (0.0005)	0.2833 (0.0285)	0.0582 (0.0664)	0.3279 (0.0506)	0.2231 (0.0292)	1.6720 (0.7682)
	Cox_G(Keya et al.)	0.7065 (0.0287)	0.7231 (0.0328)	-6.0720 (0.1023)	0.0933 (0.0030)	0.4196 (0.1634)	0.2148 (0.1305)	0.6131 (0.2525)	0.4158 (0.1817)	1.2700 (0.2641)
	Cox_G(R&P)	0.7277 (0.0279)	0.7484 (0.0315)	-6.0357 (0.0751)	0.0908 (0.0006)	0.3139 (0.0444)	0.1489 (0.0848)	0.3735 (0.0483)	0.2788 (0.0281)	2.3540 (0.8776)
	$\text{Cox}_\cap$(Keya et al.)	0.7191 (0.0260)	0.7419 (0.0290)	-6.0343 (0.0959)	0.0944 (0.0019)	0.4810 (0.1123)	0.2417 (0.0834)	0.5645 (0.2882)	0.4291 (0.1591)	0.8320 (0.4299)
	DRO-COX	0.7055 (0.0165)	0.7286 (0.0173)	-6.2915 (0.0392)	**0.0877 (0.0004)**	**0.0207 (0.0296)**	**0.0085 (0.0050)**	**0.0871 (0.0184)**	**0.0387 (0.0162)**	0.7090 (0.5854)
	DRO-COX (SPLIT)	0.7037 (0.0180)	0.7271 (0.0163)	-6.2806 (0.0723)	0.0889 (0.0035)	0.0357 (0.0746)	0.0181 (0.0237)	0.1161 (0.0990)	0.0566 (0.0656)	**0.6500 (0.6580)**
Nonlinear	DeepSurv	**0.7488 (0.0103)**	**0.7729 (0.0105)**	**-5.9582 (0.0538)**	0.0966 (0.0047)	0.3686 (0.0959)	0.1178 (0.0771)	0.4734 (0.1203)	0.3199 (0.0776)	0.4270 (0.4259)
	DeepSurv_I(Keya et al.)	0.7003 (0.0272)	0.7222 (0.0309)	-6.1838 (0.1060)	0.0925 (0.0076)	0.0807 (0.0267)	0.0205 (0.0118)	0.4122 (0.3883)	0.1711 (0.1221)	2.7770 (1.3398)
	DeepSurv_I(R&P)	0.7304 (0.0104)	0.7535 (0.0097)	-6.0678 (0.0928)	0.0906 (0.0036)	0.1236 (0.0122)	0.0407 (0.0207)	0.3603 (0.1503)	0.1749 (0.0422)	0.6010 (0.2886)
	DeepSurv_G(Keya et al.)	0.7249 (0.0084)	0.7506 (0.0071)	-6.0926 (0.0716)	0.0913 (0.0059)	0.1934 (0.0355)	0.0742 (0.0074)	0.3280 (0.0664)	0.1985 (0.0099)	0.5820 (0.2068)
	DeepSurv_G(R&P)	0.7300 (0.0106)	0.7533 (0.0101)	-6.0621 (0.0918)	0.0907 (0.0036)	0.1352 (0.0068)	0.0491 (0.0219)	0.3681 (0.1400)	0.1841 (0.0401)	0.6130 (0.2758)
	$\text{DeepSurv}_\cap$(Keya et al.)	0.7151 (0.0040)	0.7444 (0.0042)	-6.1133 (0.0086)	0.0889 (0.0004)	0.1630 (0.0086)	0.0579 (0.0069)	0.2326 (0.0229)	0.1512 (0.0111)	**0.2050 (0.0829)**
	Deep DRO-COX	0.6888 (0.0206)	0.7014 (0.0240)	-6.2531 (0.0305)	**0.0876 (0.0004)**	**0.0456 (0.0177)**	**0.0094 (0.0081)**	**0.1375 (0.0186)**	**0.0641 (0.0117)**	1.3000 (0.9742)
	Deep DRO-COX (SPLIT)	0.6829 (0.0142)	0.7099 (0.0149)	-6.1870 (0.0426)	0.1023 (0.0004)	0.2276 (0.0901)	0.1441 (0.0543)	0.3261 (0.0554)	0.2326 (0.0652)	0.9190 (0.4446)

Table E.12: 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 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-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	$0.1n$	0.7813 (0.0181)	0.8023 (0.0174)	-7.0822 (0.0069)	0.1415 (0.0063)	0.0035 (0.0030)	0.0118 (0.0062)	0.0292 (0.0138)	0.0148 (0.0075)	0.5670 (0.3535)
	$0.2n$	0.7955 (0.0053)	0.8156 (0.0036)	-7.0835 (0.0044)	0.1403 (0.0031)	0.0026 (0.0020)	0.0105 (0.0032)	0.0251 (0.0107)	0.0127 (0.0053)	0.3810 (0.2621)
	$0.3n$	0.7976 (0.0027)	0.8181 (0.0026)	-7.0844 (0.0033)	0.1398 (0.0025)	0.0021 (0.0015)	0.0099 (0.0025)	0.0254 (0.0074)	0.0124 (0.0038)	0.2150 (0.1907)
	$0.4n$	0.7969 (0.0040)	0.8174 (0.0024)	-7.0852 (0.0019)	0.1393 (0.0014)	0.0018 (0.0008)	0.0093 (0.0014)	0.0240 (0.0043)	0.0117 (0.0021)	0.2910 (0.1280)
	$0.5n$	0.7963 (0.0045)	0.8168 (0.0030)	-7.0856 (0.0009)	0.1390 (0.0008)	0.0016 (0.0004)	0.0089 (0.0008)	0.0232 (0.0031)	0.0113 (0.0013)	0.2340 (0.1237)
Nonlinear	$0.1n$	0.7619 (0.0068)	0.7707 (0.0079)	-6.8103 (0.0186)	0.1703 (0.0002)	0.3923 (0.0514)	0.4897 (0.0509)	0.6953 (0.0889)	0.5258 (0.0624)	2.8090 (0.1807)
	$0.2n$	0.7621 (0.0069)	0.7710 (0.0082)	-6.8114 (0.0186)	0.1703 (0.0002)	0.4167 (0.0676)	0.5056 (0.0591)	0.7299 (0.1075)	0.5507 (0.0773)	2.8030 (0.2261)
	$0.3n$	0.7627 (0.0067)	0.7719 (0.0080)	-6.8115 (0.0182)	0.1703 (0.0002)	0.4321 (0.0755)	0.5164 (0.0645)	0.7525 (0.1159)	0.5670 (0.0849)	2.7770 (0.2414)
	$0.4n$	0.7627 (0.0061)	0.7719 (0.0073)	-6.8123 (0.0180)	0.1703 (0.0002)	0.4414 (0.1018)	0.5236 (0.0798)	0.7578 (0.1316)	0.5742 (0.1037)	2.7930 (0.2281)
	$0.5n$	0.7629 (0.0064)	0.7719 (0.0076)	-6.8131 (0.0199)	0.1703 (0.0002)	0.4347 (0.1214)	0.5184 (0.0922)	0.7508 (0.1417)	0.5680 (0.1176)	2.8490 (0.2435)

Table E.13: 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 2 except without any cells highlighted in green as we are not comparing against baselines by previous authors.

	Methods	Accuracy Metrics				Fairness Metrics				
		c-index↑	AUC↑	LPL↑	IBS↓	F_I↓	F_G↓	$F_\cap$↓	F_A↓	CI(%)↓
Linear	DRO-COX (SPLIT, ONE SIDE)	0.7809 (0.0092)	0.8031 (0.0101)	-7.0862 (0.0029)	**0.1330 (0.0002)**	0.0017 (0.0011)	0.0090 (0.0024)	0.0232 (0.0079)	0.0113 (0.0037)	0.4420 (0.3206)
	DRO-COX (SPLIT)	**0.7963 (0.0045)**	**0.8168 (0.0030)**	**-7.0856 (0.0009)**	0.1390 (0.0008)	**0.0016 (0.0004)**	**0.0089 (0.0008)**	**0.0232 (0.0031)**	**0.0113 (0.0013)**	**0.2340 (0.1237)**
Non-linear	Deep DRO-COX (SPLIT, ONE SIDE)	0.7625 (0.0062)	0.7715 (0.0075)	-6.8133 (0.0208)	**0.1371 (0.0008)**	0.4402 (0.1369)	0.5232 (0.1039)	0.7533 (0.1514)	0.5722 (0.1298)	**2.8000 (0.2671)**
	Deep DRO-COX (SPLIT)	**0.7629 (0.0064)**	**0.7719 (0.0076)**	**-6.8131 (0.0199)**	0.1703 (0.0002)	**0.4347 (0.1214)**	**0.5184 (0.0922)**	**0.7508 (0.1417)**	**0.5680 (0.1176)**	2.8490 (0.2435)