HACSurv: A Hierarchical Copula-based Approach for Survival Analysis with Dependent Competing Risks

Xin Liu
Southeast University

Weijia Zhang
The University of Newcastle

Min-Ling Zhang
Southeast University

Abstract

In survival analysis, subjects often face competing risks; for example, individuals with cancer may also suffer from heart disease or other illnesses, which can jointly influence the prognosis of risks and censoring. Traditional survival analysis methods often treat competing risks as independent and fail to accommodate the dependencies between different conditions. In this paper, we introduce HACSurv, a survival analysis method that learns Hierarchical Archimedean Copulas structures and cause-specific survival functions from data with competing risks. HACSurv employs a flexible dependency structure using hierarchical Archimedean copulas to represent the relationships between competing risks and censoring. By capturing the dependencies between risks and censoring, HACSurv achieves better survival predictions and offers insights into risk interactions. Experiments on synthetic datasets demonstrate that our method can accurately identify the complex dependency structure and precisely predict survival distributions, whereas the compared methods exhibit significant deviations between their predictions and the true distributions. Experiments on multiple real-world datasets also demonstrate that our method achieves better survival prediction compared to previous state-of-the-art methods.

1 Introduction

Survival analysis is a statistical methodology for predicting the time until an event of interest occurs. It plays a significant role in fields such as medicine Friedman et al. (2015); Yeh et al. (2016), reliability testing Peña and Hollander (2004), and finance Caselli et al. (2021); Bosco Sabuhoro et al. (2006). For example, in healthcare survival analysis has been utilized to predict the relapse/death time of cancer; in reliability engineering, it has been adopted to study the maintenance life of manufacturing equipment. The objective of survival analysis is to estimate the probability of an event happening at a specific time and the event time, providing insights into the associated risk.

Compared to standard regression problems, the key challenge in survival analysis is the handling of censoring Wang and Sun (2022); Gharari et al. (2023); Emura and Chen (2018) and competing risks Lee et al. (2018); Li et al. (2023). In healthcare, censoring occurs when a subject loses contact or experiences other risks before the event of interest. In the latter case, the situation is termed as *competing risks* where patients may have multiple diseases simultaneously, but only the occurrence time of one disease can be observed. For example, a patient with cancer may also have other comorbidities, such as heart disease Li et al. (2023).

Uncovering the dependencies between competing risks through a data-driven approach is of great practical importance, as it not only enables more accurate survival predictions but also helps answer questions such as: "Are individuals with arteriosclerosis more likely to die from pneumonia than those without a heart condition?" raised by Tsiatis (1975). Addressing such questions is essential because mutual influences among multiple diseases are common in real-world scenarios. Most existing methods for survival analysis under competing risks directly optimize the Cumulative Incidence Function (CIF) Lee et al. (2018); Nagpal et al. (2021); Jeanselme et al. (2023); these methods neglect attention to the dependency structure or are based on the assumption of independent competing risks. Moreover, existing methods are all based on the independent censoring assumption, which is often violated in practice. Additionally, these CIF-based methods cannot provide predictions of the marginal distributions, which are

also very important objective in survival analysis.

To address these limitations, *copulas* can be used as powerful statistical tools for modeling dependencies between random variables Okhrin et al. (2017). Recently, researchers in the statistical community have used copulas to model the relationships between two competing events and censoring to obtain better survival estimates Li et al. (2023). However, their methods rely on the Cox Proportional Hazards (CoxPH) assumption Cox (1972), which is a strong and often violated assumption in practice. In addition, their method requires an artificial specification of the copula type, and a wrong copula type can increase the predictive bias.

The recently introduced DCSurvival method Zhang et al. (2024), which uses the ACNet Ling et al. (2020), has demonstrated effectiveness in handling the dependencies in single-risk scenarios. However, using a single Archimedean copula can only characterize symmetric dependency relationships Okhrin et al. (2017), and thus is not sufficient for competing risks as the relationships among risks and censoring are often flexible and asymmetric Li et al. (2023); Li and Lu (2019); Lo et al. (2020). For example, a study on breast cancer survivors found that the dependency between the time to relapse/second cancer (RSC) and the time to cardiovascular disease (CVD), and their dependencies on informative censoring time are different Li et al. (2023); Davis et al. (2014)

In this paper, we introduce HACSurv, a survival analysis framework designed to learn hierarchical Archimedean copulas (HACs) and the marginal survival distributions of competing events and censoring from right-censored survival data. HACSurv uses HAC to flexibly model asymmetric dependency structures. To the best of our knowledge, HACSurv is the first data-driven survival analysis method that models the dependency between competing risks and censoring. Our contributions are summarized as follows:

- We propose *HACSurv*, a novel survival analysis method that captures the asymmetric dependency structure between competing risks and censoring using HAC. We also introduce HACSurv (Symmetry), a simplified version that employs a single Archimedean Copula to model the dependency structure which allows for end-to-end training.

- We revisit the survival prediction objective in the context of dependent competing risks and censoring. Unlike previous methods that rely solely on marginal survival functions for predictions Gharari et al. (2023); Zhang et al. (2024), our approach introduces a novel method to predict the conditional cause-specific cumulative incidence function

(CIF) for dependent competing risks.

- Our experiments on synthetic datasets demonstrate that HACSurv significantly reduces bias in predicting the marginal survival distributions. Our methods achieve state-of-the-art results in survival outcome prediction on multiple real-world datasets. Furthermore, the copulas learned by HACSurv among competing events can potentially aid practitioners in better understanding the associations between diseases.

2 Preliminaries

2.1 Survival Data and likelihood

A sample in the survival dataset $\mathcal{D} = \{(\boldsymbol{x}_i, t_i, e_i)\}_{i=1}^{N}$ typically consists of three components: (1) a D-dimensional covariate $\boldsymbol{x}$, (2) the observed event time t, and (3) an event indicator e. For a survival dataset with K competing risks, the event indicator e ranges over $\mathcal{K} = \{0, 1, \cdots, K\}$, where $e = 0$ represents censoring. In competing events scenarios, we consider that only one event can be observed, and the observed time t_i is the minimum of the potential occurrence times of all events and censoring. In the single-risk case, e is a binary indicator.

We extend the likelihood from Gharari et al. (2023) and Zhang et al. (2024) to the case with competing risks. For a sample $(\boldsymbol{x}, t, e)$ in survival data, the likelihood can be expressed as:

$$\mathcal{L} = \prod_{k=0}^{K} \left[\Pr(T_k = t, \{T_i > t\}_{i \neq k} | \boldsymbol{x}) \right]^{\mathbb{1}_{\{e=k\}}} \quad (1)$$

where $\mathbb{1}_{\{e=k\}}$ is an indicator function.

In this paper, we denote the marginal distributions for competing events or censoring time as $S_{T_k|X}(t \mid \boldsymbol{x}) = \Pr(T_k > t \mid \boldsymbol{x})$. The corresponding density function can be written as $f_{T_k|X}(t \mid \boldsymbol{x}) = -\frac{\partial S_{T_k|X}(t|\boldsymbol{x})}{\partial t}$.

If the events and censoring are assumed to be independent, Equation 1 simplifies to:

$$\mathcal{L}_{\text{Indep}} = \prod_{k=0}^{K} \left[f_k(t) \prod_{i \neq k} S_i(t) \right]^{\mathbb{1}_{\{e=k\}}} \quad (2)$$

However, in reality, events and censoring are often dependent Li and Lu (2019). In this paper, we use *copulas* to model their dependencies, under the mild assumption that the copula does not depend on the covariates $\boldsymbol{x}$.

2.2 Copulas and Archimedean Copulas

Copula is a powerful statistical tool for modeling the dependence structure between multivariate random variables, which can be viewed as a multivariate cumulative distribution function (CDF) with uniform $[0, 1]$ margins. Sklar's theorem ensures that any d-dimensional continuous CDF can be uniquely expressed as a composition of its univariate margins and a copula.

Theorem 1 (Sklar's theorem). *For a d-variate cumulative distribution function F, with j-th univariate margin F_j, the copula associated with F is a cumulative distribution function $C : [0, 1]^d \to [0, 1]$ with $U(0, 1)$ margins satisfying:*

$$F(x_1, \cdots, x_d) = C(F_1(x_1), \cdots, F_d(x_d)), \quad (3)$$

where $(x_1, \cdots, x_d) \in \mathbb{R}^d$. If F is continuous, then C is unique.

Due to its flexibility, non-parametric copulas are difficult to characterize.

Archimedean Copulas Most recent works focus on Archimedean copulas Ling et al. (2020); Ng et al. (2021); Zhang et al. (2024); Gharari et al. (2023), a widely used family of copulas that can represent different tail dependencies using a one-dimensional generator:

$$C(\mathbf{u}) = \varphi(\varphi^{-1}(u_1) + \cdots + \varphi^{-1}(u_d)) \quad (4)$$

where $\varphi : [0, \infty) \to [0, 1]$ is the generator of the Archimedean copula. To ensure that C is a valid copula, the generator must be completely monotone, i.e., $(-1)^k \varphi^{(k)} \geq 0$ for all $k \in \{0, 1, 2, \cdots\}$.

2.3 Hierarchical Archimedean Copulas

A significant limitation of Archimedean copulas, which are defined by a single generation, is their inability to model asymmetric dependency structures Okhrin et al. (2017). However, in real-world scenarios, the dependencies among events and censoring are rarely symmetric Li and Lu (2019). For example, the dependence between the time to relapse or a second cancer and cardiovascular disease may differ significantly from their dependence on censoring Li et al. (2023).

Hierarchical Archimedean Copulas (HACs) address these limitations by allowing more complex and flexible dependency structures Joe (1997).To illustrate the idea of HAC, we consider a three-dimensional example as shown in Figure 1. This approach can be easily extended to higher-dimensional cases with J inner copulas. The HAC for a three-dimensional scenario is expressed as:

$$C(u_1, u_2, u_3) = \varphi_0 \left(\varphi_0^{-1}(u_1) + \right. \\ \left. \varphi_0^{-1} \circ \varphi_1 \left(\varphi_1^{-1}(u_2) + \varphi_1^{-1}(u_3)\right)\right) \quad (5)$$

For an HAC to be a valid copula, the sufficient nesting conditions Joe (1997); McNeil (2008) are required, which state that:

- φ_j for all $j \in \{0, 1, \ldots, j\}$ are completely monotone,

- $\left(\varphi_0^{-1} \circ \varphi_j\right)'$ for $j \in \{1, \ldots, j\}$ are completely monotone.

The latter criterion, i.e., $\left(\varphi_0^{-1} \circ \varphi_j\right)'$ is completely monotone, is discussed in detail in Hering et al. (2010). In brief, for a given outer generator φ_0, the inner generator φ_j can be constructed using the Laplace exponent ψ_j of a Lévy subordinator and the outer generator φ_0, such that $\varphi_j(x) = (\varphi_0 \circ \psi_j)(x)$. The Laplace exponent ψ_j has the following expression:

$$\psi_j(x) = \mu_j x + \beta_j \left(1 - \varphi_{M_j}(x)\right) \quad (6)$$

$$= \mu_j x + \beta_j \left(1 - \int_0^\infty e^{-xs} dF_{M_j}(s)\right) \quad (7)$$

where M_1 is a positive random variable with Laplace transform φ_{M_1}, and $\mu_1 > 0$ and $\beta_1 > 0$. The derivation of ψ_1 is rather complex, so we only present the final form of ψ_1 here. The detailed mathematical derivation is provided in the supplementary material following Hering et al. (2010).

3 Methods

In order to reduce estimation bias caused by independent or symmetric dependencies, HACSurv utilizes hierarchical Archimedean copula to model the asymmetric dependency structure among competing risks and censoring, and learns the corresponding cause-specific marginal survival functions with neural network. Figure 1 illustrates the model architecture of HACSurv in the setting of 2 competing risks.

In this section, we first derive the likelihood under dependent competing risks and censoring, and introduce the first method proposed in this paper, which uses a single Archimedean copula to characterize the overall dependency of the competing risks. Next, we present how neural networks can be used to model HAC and the marginal survival distributions. The final part of this section explores how to compute the conditional
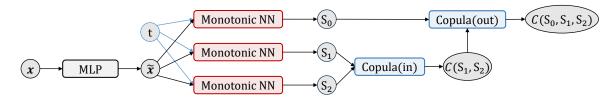
Figure 1: Overview of our HACSurv for the two competing risks scenario. We abbreviate $S_{T_k|X}$ as S_k, for $k = 0, 1, 2$. The copula (out) can be represented by the outer generator φ_0. The inner generator φ_1 corresponding to copula (in) is constructed from the Laplace exponent ψ_1 and the outer generator φ_0.

cause-specific cumulative incidence function (CIF) using HACSurv in the presence of dependencies, which is an alternative survival prediction method to directly using the marginal survival distribution.

3.1 Survival Likelihood with Dependent Competing Risks

We begin by extending the survival likelihood with dependent censoring described in Zhang et al. (2024) to the HAC-based survival likelihood with dependent competing risks. Using Sklar's theorem, we obtain the joint probability function for K competing risks:

$$\Pr(T_0 > t, \cdots, T_K > t \mid \boldsymbol{x}) = \\ \mathcal{C}\left(S_{T_0|X}(t \mid \boldsymbol{x}), \cdots, S_{T_K|X}(t \mid \boldsymbol{x})\right) \quad (8)$$

Incorporating the above equation into the survival likelihood in Equation 1 (we omit covariates $\boldsymbol{x}$ for brevity), we can get:

$$\mathcal{L} = \prod_{k=0}^{K} \left\{ f_{T_k}(t) \frac{\partial}{\partial u_k} C(u_0, \ldots, u_K) \right\}^{\mathbb{1}_{\{e=k\}}} \quad (9)$$

where each $u_i = S_{T_i}(t)$ for all $i = 0, \ldots, K$. A simple approach to represent $C(u_0, \ldots, u_K)$ is by using a single Archimedean copula according to Equation 4. This survival analysis method based on a single copula is referred to in this paper as HACSurv (Symmetry), which is an extension of DCSurvival Zhang et al. (2024) in the competing risks setting. Although it cannot capture complex asymmetric structures, it remains a practical approach. Later experiments demonstrate its superiority over the independent copula. Furthermore, a key advantage of HACSurv(Symmetry) is that it can be trained in an end-to-end manner.

A more flexible modeling approach is to represent $C(u_0, \ldots, u_K)$ using a hierarchical Archimedean copula where $\frac{\partial}{\partial u_k} C(u_0, \cdots, u_K)$ can be calculated based on the HAC structure and the chain rule of derivatives. For the example shown in Figure 1, when $k = 1$, this

can be specifically expressed as:

$$\frac{\partial}{\partial u_1} C(u_0, u_1, u_2) = \\ \frac{\partial C(u_0, u_1, u_2)}{\partial C_{in}(u_1, u_2)} \cdot \left. \frac{\partial C_{in}(u_1, u_2)}{\partial u_1} \right|_{\substack{u_0 = S_{T_0}(t) \\ u_1 = S_{T_1}(t) \\ u_2 = S_{T_2}(t)}} \quad (10)$$

where $C_{in} = \varphi_1\left(\varphi_1^{-1}(u_1) + \varphi_1^{-1}(u_2)\right)$. Under this HAC structure, C_{in} can capture diverse forms of dependency between competing events. Moreover, the outer copula can describe the dependencies between competing events and censoring.

3.2 Learning Archimedean Copula from Survival Data

As discussed in Section 2.2, we need a completely monotone generator φ to construct a valid Archimedean copula. The Bernstein-Widder characterization theorem Bernstein (1929); Widder (2015), states that any completely monotone function can be characterized by the Laplace transform of a positive random variable:

Theorem 2 (Bernstein-Widder). *A function φ is completely monotone and $\varphi(0) = 1$ if and only if φ is the Laplace transform of a positive random variable. Specifically, $\varphi(x)$ can be represented as:*

$$\varphi(x) = \int_0^\infty e^{-xs} \, dF_M(s), \quad (11)$$

where $M > 0$ is a positive random variable with the Laplace transform φ.

We represent the generator φ of Archimedean copula using the method proposed by Ng et al. (2021). Specifically, we define a generative neural network $G(\cdot; \theta)$ parameterized by θ. We let M be the output of this neural network such that samples $M \sim F_M$ are computed as $M = G(\epsilon; \theta)$, where ϵ is a source of randomness and serve as the input to the neural network. We then approximate the Laplace transform $\varphi(x)$ by its

empirical version, using L samples of M from $G(\cdot; \theta)$:

$$\varphi(x) = \int_0^\infty e^{-xs} \, dF_M(s) = \mathbb{E}[e^{-Mx}] \approx \frac{1}{L} \sum_{l=1}^{L} e^{-M_l x}, \tag{12}$$

To compute Equation 4, we need to compute the inverse of $\varphi(x)$. Following previous works Ling et al. (2020); Ng et al. (2021), we employ Newton's method to achieve this. Additionally, by differentiating with respect to $u_k = S_k(t)$, we can derive $\frac{\partial}{\partial u_k} C(u_0, \ldots, u_K)$ in Equation 9. During the optimization of the survival likelihood, the parameters of the generative neural network $G(\cdot; \theta)$ are updated via gradient descent.

3.3 Learning HAC with the Re-generation Trick

3.3.1 Determine the Structure of HAC

The ideal Hierarchical Archimedean Copula (HAC) structure should closely resemble the true dependency structure among competing events and censoring. For survival data with only one event or censoring observation time, it is theoretically not guaranteed to determine the true marginal distribution and copula without making any assumptions Tsiatis (1975). However, this unidentifiability does not make estimation pointless. As we will demonstrate in the experiments, accounting for the dependency among events substantially reduces estimation bias, despite the unidentifiable results.

In this paper, we first capture the copulas between each pair of competing events and censoring. The approach is similar to DCSurvival proposed by Zhang et al. (2024) for determining the copula between events and censoring, but we replace ACNet Ling et al. (2020) with Gen-AC Ng et al. (2021) to represent the generator φ which enhances computational efficiency. When determining the HAC structure, we follow the principle that the dependency strength of inner (lower-level) copulas should be stronger than that of outer (higher-level) copulas Okhrin et al. (2017). Thus, the basic idea for constructing the HAC structure is to group the competing events and censoring based on their dependencies: strong dependencies within groups are modeled by lower-level inner copulas, while weaker dependencies between groups are captured by higher-level outer copulas. In the experimental section, we provide examples with 2, 3, and 5 competing risks to specifically demonstrate how to determine the HAC structure.

3.3.2 Training inner generator with Re-generation Trick

For the inner generator φ_j, we define it using the composition $\varphi_j(x) = (\varphi_0 \circ \psi_j)(x)$, following the approach proposed in Ng et al. (2021). We configure the parameters μ and β in Equation 7 as trainable parameters, applying an $\exp(\cdot)$ output activation to ensure positivity. M_j is set as the output from $G(\cdot, \theta_j)$ with parameters θ_j and $\exp(\cdot)$ output activation. We proceed by computing the Laplace transform φ_{M_j} as outlined in Equation 12.

Although Ng et al. presents a method for training the inner generator with fully observed data, their approach does not apply to partially observed survival data. To address competing risks, this paper introduces a two-stage training strategy tailored for survival data:

1. The first stage involves generating a collection of bivariate data points, denoted as $\mathbf{U}$, each represented by coordinates $(U_1, U_2) \sim C$. These data points are sampled from the selected copula, identified in Section 3.3.1 as the inner copula from among several candidates. This dataset is then utilized to train the inner generator φ_j of the HAC.

2. The second stage focuses on training φ_j using Maximum Likelihood Estimation (MLE) based on the dataset generated in the first stage. To compute the MLE, we specifically require the second-order derivatives of the inner generator, φ_j''. Just as in Ling et al. (2020), we utilize PyTorch Paszke et al. (2017) for automatic differentiation.

During this process, we keep the outer generator φ_0 fixed, following Ng et al. (2021)

3.4 Survival Analysis via HACSurv

3.4.1 Learning Marginal Distributions

Once the HAC that describes the dependency structure among competing events and censoring is obtained, our goal is to learn their survival distributions. First, we set up a shared embedding network to extract features from the covariates $\boldsymbol{x}$. We use monotonic neural density estimators (MONDE) Chilinski and Silva (2020) to model $S_{k|X}$ for all k from 0 to K following DCSurvival Zhang et al. (2024) and SuMo-net Rindt et al. (2022). MONDE consists of two parts: the first part is a fully connected network that processes the covariates, and the processed covariates along with the time input t are then concatenated and fed into the second part of the network. In this second part, all weights of the successor nodes of the time input t are non-negative. The final layer output of the MONDE network is transformed into the survival function through the sigmoid function. The corresponding density functions $f_{k|X}$, can be computed using PyTorch's automatic differentiation Paszke et al. (2017).

After both the HAC and the marginals are instantiated, Equation 9 can be computed based on the HAC structure and the chain rule of differentiation, and the parameters of the marginals can be optimized using stochastic gradient descent. It is worth noting that during the estimation of the final parameters of the marginals, we freeze the parameters of the HAC.

3.4.2 Predicting Cause-Specific CIF

Previous copula-based survival analysis methods Zhang et al. (2024); Gharari et al. (2023) employ marginal survival distribution for prediction. However, as the dependency within the true copula of event and censoring increases, Gharari et al. (2023) have shown that the Integrated Brier Score (IBS) obtained from the ground-truth marginal survival function also increases, and thus IBS is not suitable for dependent censoring.

Moreover, we posit that the observed survival data represent the joint probability of the event e_i occurring at time t_i and other events not occurring at that time rather than simply marginal probability. To better account for the dependencies among competing events and censoring, we argue that predicting the conditional Cause-Specific Cumulative Incidence Function (CIF) offers a superior approach compared to predicting the marginal. To this end, we extend the original CIF Fine and Gray (1999); Lee et al. (2018) to accommodate scenarios with dependent competing risks and censoring:

$$F_{k^*}\left(t^* \mid \boldsymbol{x}^*\right) = \Pr(T_{k^*} < t^* \mid \{T_i > t^*\}_{i \neq k^*}, \boldsymbol{x}^*) \quad (13)$$

$$= 1 - \frac{C(\{u_i\}_{i=0,\ldots,K})}{C(\{u_i\}_{i \neq k^*})} \quad (14)$$

where each $u_i = S_{T_i \mid X}(t^* \mid \boldsymbol{x}^*)$ for all $i = 0, \ldots, K$.

4 Discussion on Identifiability

Tsiatis (1975) has shown that, even when survival data are generated from a ground-truth joint probability distribution (Equation 8) representing dependent competing risks and censoring, there exists an independent copula C_{Indep} and corresponding marginal survival functions $G_{T_k \mid X}(t \mid x)$ for all $k \in \{0, \ldots, K\}$ that can produce the same observed data. In fact, there are infinite combinations of copulas and corresponding marginal distributions that can represent this joint probability. Most survival methods that do not rely on copulas only optimise the cause-specific cumulative incidence function (CIF) corresponding to this joint probability, without considering the marginal distribution or assuming competing risks and censoring are mutually independent Lee et al. (2018); Nagpal et al. (2021); Danks and Yau (2022); Jeanselme et al. (2023).

However, the unidentifiability result does not render efforts to estimate the copula among competing events futile. The greater the difference between the estimated copula and the ground truth copula, the greater the disparity between the obtained and true marginal distributions. In other words, if the competing risks are dependent, existing methods will exhibit significant bias since they inherently assume an independent copula. Although it is impossible to *uniquely identify* the ground-truth copula, our method can *closely approximate* it given the available data. As we demonstrated in the experiments, under metrics that compared the estimated survival marginals to the ground truth distributions (Survival-l1) Gharari et al. (2023), HACSurv significantly reduces estimation biases when compared to methods that assume independence; under metrics that do not require access to the ground truth (e.g., C^{td}-index Antolini et al. (2005) and IBS Graf et al. (1999)), HACSurv also demonstrates significantly better results.

5 Experiments

In this section, we evaluate HAC's ability to capture the dependencies among competing risks and censoring, as well as the model's predictive ability for the marginal survival distribution and survival outcomes. Since the true dependencies are not known in the real world, we first construct a synthetic dataset with three competing risks based on a known HAC. In addition, we further compare against baselines using three real-world datasets to evaluate HACSurv's ability to predict patient survival time. We pick the following baselines for comparison: DeepHit Lee et al. (2018), DSM Nagpal et al. (2021), DeSurv Danks and Yau (2022), and NeuralFG Jeanselme et al. (2023). We also present the results of HACSurv with an independent copula and a single Archimedean copula.

Due to the complexity of the survival analysis problem, we chose whether to use the marginal survival function or CIF for prediction based on the average C^{td}-index of all risks in the validation set.

Synthetic Dataset Following the approach proposed by Gharari et al. (2023) for constructing a single-risk synthetic dataset with dependent censoring, we develop a synthetic dataset with three competing risks. The marginal distributions of competing events and censoring were modeled using a CoxPH model where the events and censoring risks are specified by Weibull distributions where the hazards are linear functions of covariates. The dependency structure is characterized by a known four-variate hierarchical Archimedean copula $C_{\varphi_0}\left(C_{\varphi_1}\left(u_1, u_2\right), C_{\varphi_2}\left(u_3, u_4\right)\right)$. The samples of the known copula were generated using the HA-
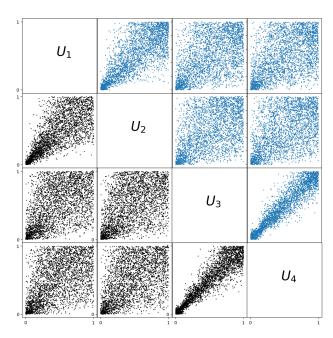
Figure 2: The blue samples are generated from the HAC learned on the synthetic dataset. The samples drawn from the ground truth copulas are in black. The results are presented in a mirrored format.

Copula Toolbox Górecki et al. (2017). Specifically, $C_{\varphi_0}, C_{\varphi_1}, C_{\varphi_2}$ are Clayton copulas with their θ parameters set to 1, 3, and 8, respectively.

Real-World Datasets *Framingham*: This is a cohort study gathering 18 longitudinal measurements on male patients over 20 years Kannel and McGee (1979). We consider death from cardiovascular disease (CVD) as Risk 2, and death from other causes as Risk 1.

SEER: The Surveillance, Epidemiology, and End Results Program (SEER) dataset Howlader et al. (2010) is an authoritative database in the U.S. that provides survival information for breast cancer patients. We selected information on breast cancer patients during 2000–2020. Among a total of 113,561 patients, the percentages of censored, died due to cardiovascular disease (Risk 1), and died due to breast cancer (Risk 2) were approximately 70.9%, 6%, and 23.1%, respectively.

MIMIC-III: We extract survival data of 2,279 patients from the MIMIC-III database Tang et al. (2021). Among them, 1,353 patients (59.37%) were right-censored; 517 patients (22.68%) died of sepsis (Risk 1); 65 patients (2.85%) died of cerebral hemorrhage (Risk 2); 238 patients (10.44%) died of acute respiratory failure (Risk 3); 62 patients (2.72%) died of subendocardial acute myocardial infarction (Risk 4); and 44 patients (1.93%) died of pneumonia (Risk 5).
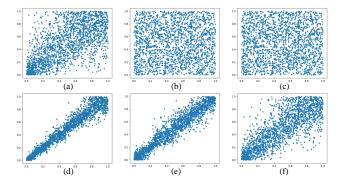

Figure 3: Copulas learned by HACSurv from Framingham and MIMIC-III dataset. (a), (b) and (c) are the copulas between Risk 1 and Risk 2, Risk 1 and censoring, and Risk 2 and censoring on the Framingham dataset. (d), (e) and (f) are the copulas between Risk 1 and Risk 3, Risk 3 and Risk 5, and Risk 2 and Risk 4 on MIMIC-III, respectively.

Table 1: Performance comparison of survival models (Survival-L_1). In all tables, HACSurv (I) and HAC-Surv (S) respectively correspond to HACSurv with an independent copula and a single symmetry copula.

Method	Risk 1	Risk 2	Risk 3
DeepHit	$0.375 \pm_{0.015}$	$0.382 \pm_{0.015}$	$0.152 \pm_{0.012}$
DSM	$0.243 \pm_{0.008}$	$0.289 \pm_{0.005}$	$0.067 \pm_{0.005}$
DeSurv	$0.362 \pm_{0.013}$	$0.371 \pm_{0.012}$	$0.133 \pm_{0.007}$
NeuralFG	$0.354 \pm_{0.012}$	$0.359 \pm_{0.012}$	$0.157 \pm_{0.004}$
HACSurv (I)	$0.204 \pm_{0.006}$	$0.230 \pm_{0.015}$	$0.067 \pm_{0.001}$
HACSurv (S)	$0.039 \pm_{0.005}$	$0.096 \pm_{0.008}$	$0.013 \pm_{0.000}$
HACSurv	$\mathbf{0.023} \pm_{0.004}$	$\mathbf{0.012} \pm_{0.002}$	$\mathbf{0.008} \pm_{0.002}$

Qualitative Results As shown in Figure 2, the dependency structure learned by HACSurv from the synthetic dataset is almost identical to the true HAC. The results demonstrate that our model can learn complex dependency structures from partially observed competing risks survival data. The results from Figure 3 on the Framingham dataset show that HACSurv found a positive dependency between Risk 1 and Risk 2, while both risks are independent of censoring.

We present some of the dependency structures discovered in the MIMIC-III. Our model found strong positive dependencies between sepsis (Risk 1) and acute respiratory failure (Risk 3), as well as between acute respiratory failure (Risk 3) and pneumonia (Risk 5), and found a moderate positive dependency between cerebral hemorrhage (Risk 2) and subendocardial acute myocardial infarction (Risk 4). This is consistent with medical knowledge, demonstrating that HACSurv may uncover the interactions between diseases from partially observed data.

Table 2: Overall Performance Comparison on Four Datasets. We report the mean of the metrics for all risks. The results for each specific risk are provided in the supplementary materials.

Metric	Model	Synthetic	Framingham	SEER	MIMIC-III
C^{td}-index	DeepHit	0.611 ± 0.008	0.699 ± 0.019	0.791 ± 0.000	0.765 ± 0.015
	DSM	0.603 ± 0.009	0.734 ± 0.016	0.781 ± 0.006	0.752 ± 0.018
	DeSurv	0.617 ± 0.007	0.674 ± 0.022	0.787 ± 0.005	0.755 ± 0.021
	NeuralFG	0.549 ± 0.019	0.746 ± 0.010	0.779 ± 0.004	0.760 ± 0.026
	HACSurv (I)	0.635 ± 0.005	$\mathbf{0.748} \pm 0.009$	0.797 ± 0.003	0.764 ± 0.018
	HACSurv (S)	0.636 ± 0.006	0.746 ± 0.013	0.797 ± 0.003	0.760 ± 0.016
	HACSurv	$\mathbf{0.643} \pm 0.007$	0.747 ± 0.012	$\mathbf{0.798} \pm 0.002$	$\mathbf{0.768} \pm 0.018$
IBS	DeepHit	0.151 ± 0.003	0.089 ± 0.003	0.074 ± 0.001	0.264 ± 0.083
	DSM	0.086 ± 0.003	0.088 ± 0.004	0.069 ± 0.000	0.226 ± 0.080
	DeSurv	0.143 ± 0.002	0.104 ± 0.004	0.068 ± 0.001	0.268 ± 0.085
	NeuralFG	0.173 ± 0.016	0.091 ± 0.003	0.068 ± 0.001	0.276 ± 0.097
	HACSurv (I)	$\mathbf{0.077} \pm 0.002$	0.092 ± 0.005	$\mathbf{0.067} \pm 0.001$	0.169 ± 0.049
	HACSurv (S)	0.078 ± 0.004	0.091 ± 0.006	$\mathbf{0.067} \pm 0.001$	$\mathbf{0.131} \pm 0.021$
	HACSurv	0.100 ± 0.008	$\mathbf{0.085} \pm 0.003$	$\mathbf{0.067} \pm 0.001$	0.163 ± 0.068

Table 3: Comparison of Training Time (per 100 epochs) and GPU Memory Usage on SEER Dataset

Method	Dim	Batch Size	Time (s)	Memory (MB)
HACSurv	2D	20000	19.5	2890
DCSurvival	2D	20000	255.36	21700
HACSurv	3D	10000	38.05	3914
DCSurvival	3D	10000	450.77	22356

Quantitative Results The results in Table 1 show that after identifying the dependency structure among events and censoring, HACSurv significantly reduces the estimation bias of the marginal survival distributions. Notably, even though only a single copula is used to capture the dependency, HACSurv (Symmetry) estimates the marginal survival distributions much better than HACSurv (Independent). DSM achieves better predictions than other existing methods due to its use of a parametric structure (Weibull) consistent with the true marginal distributions. However, since it trains and predicts based on CIF only and does not model the dependency structure, it is less effective than HACSurv.

According to Table 2, HACSurv significantly outperforms existing methods in terms of the C^{td}-index metric. It is worth mentioning that although HACSurv achieves near-perfect survival marginal predictions on the synthetic dataset, its IBS is not as good as its version based on an independent copula. This further confirms that IBS is not a strictly proper scoring rule. Nevertheless, in terms of IBS, HACSurv still achieves results superior to existing state-of-the-art methods.

Table 3 shows the computational efficiency comparison between HACSurv and DCSurvival in the case of a single copula. The training time required by HACSurv is only about 8% of that of DCSurvival, with 17%

of the GPU memory usage. Thus, in the single-risk setting, HACSurv also represents an improvement over DCSurvival.

We provide detailed experimental settings, additional qualitative results, and an analysis of the differences in predictions using marginal distributions and CIF in the supplementary materials.

6 Conclusion

In this paper, we propose HACSurv, a survival analysis method capable of modeling the dependency structure among competing risks and censoring. Our method directly uses the survival likelihood for training, without relying on the independent censoring assumption or the independent competing risks assumption. We use hierarchical Archimedean copulas (HACs) to flexibly model asymmetric dependency structures. HACSurv first learns the structure and parameters of the HAC from partially observed survival data, and then learns the marginal survival distributions. This paper also explores the prediction objective under the survival analysis setting, extending the use of survival marginal for prediction in single-risk scenarios to using conditional cause-specific CIF to better model the interactions among competing risks. Empirically, we demonstrate that HACSurv can closely approximate the dependency structures in survival data and significantly reduces survival estimation bias.

However, there are some limitations in our proposed HACSurv. Although hierarchical Archimedean copulas are capable of characterizing most dependencies in the real world, there are certain dependency structures that cannot be accurately represented by HACs. A possible future direction is to explore vine copulas that are more flexible than HAC.

References

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

Serge Bernstein. Sur les fonctions absolument monotones. *Acta Mathematica*, 52(1):1–66, 1929.

Jean Bosco Sabuhoro, Bruno Larue, and Yvan Gervais. Factors determining the success or failure of canadian establishments on foreign markets: A survival analysis approach. *The International Trade Journal*, 20(1):33–73, 2006.

Stefano Caselli, Guido Corbetta, Doriana Cucinelli, and Monica Rossolini. A survival analysis of public guaranteed loans: Does financial intermediary matter? *Journal of Financial Stability*, 54:100880, 2021.

Pawel Chilinski and Ricardo Silva. Neural likelihoods via cumulative distribution functions. In *Conference on Uncertainty in Artificial Intelligence*, pages 420–429. PMLR, 2020.

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

Dominic Danks and Christopher Yau. Derivative-based neural modelling of cumulative distribution functions for survival analysis. In *International Conference on Artificial Intelligence and Statistics*, pages 7240–7256. PMLR, 2022.

Margot K Davis, Dongdong Li, Elaine Wai, Scott Tyldesley, Christine Simmons, Christopher Baliski, and Mary L McBride. Hospital-related cardiac morbidity among survivors of breast cancer: Long-term risks and predictors. *Journal of Cardiac Failure*, 20(8):S44–S45, 2014.

Takeshi Emura and Yi-Hau Chen. *Analysis of survival data with dependent censoring: copula-based approaches*, volume 450. Springer, 2018.

Jason P Fine and Robert J Gray. A proportional hazards model for the subdistribution of a competing risk. *Journal of the American statistical association*, 94(446):496–509, 1999.

Lawrence M Friedman, Curt D Furberg, David L DeMets, David M Reboussin, and Christopher B Granger. *Fundamentals of clinical trials*. Springer, 2015.

Ali Hossein Foomani Gharari, Michael Cooper, Russell Greiner, and Rahul G Krishnan. Copula-based deep survival models for dependent censoring. In *Uncertainty in Artificial Intelligence*, pages 669–680. PMLR, 2023.

Jan Górecki, Marius Hofert, and Martin Holeňa. On structure, family and parameter estimation of hierarchical archimedean copulas. *Journal of Statistical Computation and Simulation*, 87(17):3261–3324, 2017.

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.

Christian Hering, Marius Hofert, Jan-Frederik Mai, and Matthias Scherer. Constructing hierarchical archimedean copulas with lévy subordinators. *Journal of Multivariate Analysis*, 101(6):1428–1433, 2010.

Nadia Howlader, Lynn AG Ries, Angela B Mariotto, Marsha E Reichman, Jennifer Ruhl, and Kathleen A Cronin. Improved estimates of cancer-specific survival rates from population-based data. *JNCI: Journal of the National Cancer Institute*, 102(20):1584–1598, 2010.

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

Harry Joe. *Multivariate models and multivariate dependence concepts*. CRC press, 1997.

William B Kannel and Daniel L McGee. Diabetes and cardiovascular disease: the framingham study. *Jama*, 241(19):2035–2038, 1979.

Sato Ken-Iti. *Lévy processes and infinitely divisible distributions*, volume 68. Cambridge university press, 1999.

Håvard Kvamme, Ørnulf Borgan, and Ida Scheel. Time-to-event prediction with neural networks and cox regression. *Journal of Machine Learning Research*, 20(129):1–30, 2019. URL http://jmlr.org/papers/v20/18-424.html.

Changhee Lee, William Zame, Jinsung Yoon, and Mihaela Van Der Schaar. Deephit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Dongdong Li, X Joan Hu, and Rui Wang. Evaluating association between two event times with observations subject to informative censoring. *Journal of the American Statistical Association*, 118(542):1282–1294, 2023.

Hong Li and Yang Lu. Modeling cause-of-death mortality using hierarchical archimedean copula. *Scandinavian Actuarial Journal*, 2019(3):247–272, 2019.

Chun Kai Ling, Fei Fang, and J Zico Kolter. Deep archimedean copulas. *Advances in Neural Information Processing Systems*, 33:1535–1545, 2020.

Simon MS Lo, Enno Mammen, and Ralf A Wilke. A nested copula duration model for competing risks with multiple spells. *Computational Statistics & Data Analysis*, 150:106986, 2020.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018.

Alexander J McNeil. Sampling nested archimedean copulas. *Journal of Statistical Computation and Simulation*, 78(6):567–581, 2008.

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

Yuting Ng, Ali Hasan, Khalil Elkhalil, and Vahid Tarokh. Generative archimedean copulas. In *Uncertainty in Artificial Intelligence*, pages 643–653. PMLR, 2021.

Ostap Okhrin, Alexander Ristig, and Ya-Fei Xu. Copulae in high dimensions: an introduction. *Applied quantitative finance*, pages 247–277, 2017.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Edsel A Peña and Myles Hollander. Models for recurrent events in reliability and survival analysis. *Mathematical reliability: An expository perspective*, pages 105–123, 2004.

David Rindt, Robert Hu, David Steinsaltz, and Dino Sejdinovic. Survival regression with proper scoring rules and monotonic neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1190–1205. PMLR, 2022.

Shengpu Tang, Parmida Davarmanesh, Yanmeng Song, Danai Koutra, Michael Sjoding, and Jenna Wiens. Mimic-iii and eicu-crd: Feature representation by fiddle preprocessing. *Shock (4h)*, 19(98):4–522, 2021.

Peter Tankov. *Financial modelling with jump processes*. Chapman and Hall/CRC, 2003.

Anastasios Tsiatis. A nonidentifiability aspect of the problem of competing risks. *Proceedings of the National Academy of Sciences*, 72(1):20–22, 1975.

Zifeng Wang and Jimeng Sun. Survtrace: Transformers for survival analysis with competing events. In *Proceedings of the 13th ACM international conference on bioinformatics, computational biology and health informatics*, pages 1–9, 2022.

David Vernon Widder. Laplace transform (pms-6). 2015.

Robert W Yeh, Eric A Secemsky, Dean J Kereiakes, Sharon-Lise T Normand, Anthony H Gershlick, David J Cohen, John A Spertus, Philippe Gabriel Steg, Donald E Cutlip, Michael J Rinaldi, et al. Development and validation of a prediction rule for benefit and harm of dual antiplatelet therapy beyond 1 year after percutaneous coronary intervention. *Jama*, 315(16):1735–1749, 2016.

Weijia Zhang, Chun Kai Ling, and Xuanhui Zhang. Deep copula-based survival analysis for dependent censoring with identifiability guarantees. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 20613–20621, 2024.

Supplementary Materials

A Derivation of Survival Likelihood

We omit covariates $\boldsymbol{x}$ for brevity. We start with the likelihood function:

$$\mathcal{L} = \prod_{k=0}^{K} \left[\Pr(T_k = t, \{T_i > t\}_{i \neq k})\right]^{\mathbb{1}_{\{e=k\}}} \tag{15}$$

(Initial likelihood expression)

Applying the definition of the survival function and the relationship between probability density and survival functions, we rewrite the probability as:

$$\mathcal{L} = \prod_{k=0}^{K} \left[-\frac{d}{dy} \Pr(T_k > y, \{T_i > t\}_{i \neq k}) \Big|_{y=t} \right]^{\mathbb{1}_{\{e=k\}}} \tag{16}$$

(Definition of Survival Function)

Using Sklar's Theorem to express the joint survival function in terms of a copula function $C(u_0, \ldots, u_K)$, where $u_k = S_{T_k}(y)$ and $u_i = S_{T_i}(t)$ for $i \neq k$, we have:

$$\mathcal{L} = \prod_{k=0}^{K} \left[-\frac{d}{dy} C(u_0, \ldots, u_K) \Big|_{\substack{u_k = S_{T_k}(y) \\ y=t \\ u_i = S_{T_i}(t),\, i \neq k}} \right]^{\mathbb{1}_{\{e=k\}}} \tag{17}$$

(Sklar's Theorem)

Applying the Chain Rule of differentiation, we obtain:

$$\mathcal{L} = \prod_{k=0}^{K} \left[f_{T_k}(t) \cdot \frac{\partial}{\partial u_k} C(u_0, \ldots, u_K) \Big|_{u_i = S_{T_i}(t)} \right]^{\mathbb{1}_{\{e=k\}}} \tag{18}$$

(Chain Rule of derivative)

B Derivation of the Conditional CIF

We omit covariates $\boldsymbol{x}^*$ and derive the cause-specific conditional cumulative incidence function (CIF):

$$F_{k^*}(t^*) = \Pr(T_{k^*} < t^* \mid \{T_i > t^*\}_{i \neq k^*}). \tag{19}$$

(Initial definition)

By definition of conditional probability, it can be rewritten as:

$$F_{k^*}(t^*) = \frac{\Pr(T_{k^*} < t^*, \{T_i > t^*\}_{i \neq k^*})}{\Pr(\{T_i > t^*\}_{i \neq k^*})}. \tag{20}$$

(Conditional probability)

Using Sklar's theorem, express the joint survival function in terms of copula:

$$\Pr(\{T_i > t^*\}_{i=0,\ldots,K}) = C(u_0, \ldots, u_K), \quad \Pr(\{T_i > t^*\}_{i \neq k^*}) = C(\{u_i\}_{i \neq k^*}). \tag{21}$$

(Sklar's theorem)

The joint probability $\Pr(T_{k^*} < t^*, \{T_i > t^*\}_{i \neq k^*})$ is given by:

$$\Pr(T_{k^*} < t^*, \{T_i > t^*\}_{i \neq k^*}) = C(\{u_i\}_{i \neq k^*}) - C(u_0, \ldots, u_K). \tag{22}$$

Substituting into the conditional probability formula and simplifying yields:

$$F_{k^*}(t^*) = 1 - \frac{C(\{u_i\}_{i=0,\ldots,K})}{C(\{u_i\}_{i \neq k^*})}. \tag{23}$$

where each $u_i = S_{T_i|X}(t^* \mid \boldsymbol{x}^*)$ for all $i = 0, \ldots, K$. $\hfill$ (Final simplified result)

C   Using Lévy Subordinators to Construct Inner Generators of Hierarchical Archimedean Copulas

The criterion that the derivative $\left(\varphi_0^{-1} \circ \varphi_j\right)'$ are completely monotone, is addressed in Hering et al. (2010), using Lévy Subordinators, i.e. non-decreasing Lévy processes such as the compound Poisson process, by recognizing that the Laplace transform of Lévy subordinators at a given 'time' $t \geq 0$ have the form $e^{-t\psi_j}$, where the Laplace exponent ψ_j has completely monotone derivative. For more background on Lévy processes, we refer the reader to the books Ken-Iti (1999); Tankov (2003). Here we restate the method proposed by Hering et al. (2010) and utilized in Ng et al. (2021).

For a given outer generator φ_0, a compatible inner generator φ_j can be constructed by composing the outer generator with the Laplace exponent ψ_j of a Lévy subordinator:

$$\varphi_j(x) = (\varphi_0 \circ \psi_j)(x) \tag{24}$$

where the Laplace exponent $\psi_j : [0, \infty) \to [0, \infty)$ of a Lévy subordinator has a convenient representation with drift $\mu_j \geq 0$ and Lévy measure ν_j on $(0, \infty)$ due to the Lévy-Khintchine theorem Ken-Iti (1999):

$$\psi_j(x) = \mu_j x + \int_0^\infty \left(1 - e^{-xs}\right) \nu_j(ds) \tag{25}$$

A popular Lévy subordinator is the compound Poisson process with drift $\mu_j \geq 0$, jump intensity $\beta_j > 0$, and jump size distribution determined by its Laplace transform φ_{M_j}. In this case, the Laplace exponent has the following expression:

$$\begin{aligned} \psi_j(x) &= \mu_j x + \beta_j \left(1 - \varphi_{M_j}(x)\right) \\ &= \mu_j x + \beta_j \left(1 - \int_0^\infty e^{-xs} \, dF_{M_j}(s)\right) \end{aligned} \tag{26}$$

where $M_j > 0$ is a positive random variable with Laplace transform φ_{M_j} characterizing the jump sizes of the compound Poisson process. In addition, we choose $\mu_j > 0$ to satisfy the condition $\varphi_j(\infty) = (\varphi_0 \circ \psi_j)(\infty) = 0$ such that φ_j is a valid generator of an Archimedean copula. For the sampling algorithm of HACs, we refer the reader to Hering et al. (2010).

D   ADDITIONAL EXPERIMENTS

D.1   Comparison of Marginal Survival Function and CIF

Table 4 presents the results of using marginal survival functions and CIF for predictions across four datasets. For the model based on the independent copula, there is no difference between using marginal survival functions and CIF for predictions. However, for the HAC-based model, whether marginal survival functions or CIF yield better performance varies across different datasets. We believe this may be due to the complexity of survival analysis tasks and differences in data collection across datasets. Therefore, we suggest that the final choice of whether to use the marginal survival function or CIF for prediction should be based on the results on the validation set.

Table 4: Comparison of Marginal Survival Function and CIF on Four Datasets. SF corresponds to predictions using the marginal survival function. CIF corresponds to predictions using the cause-specific conditional CIF.

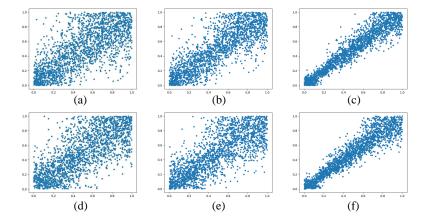| Metric	Model	Synthetic	Framingham	SEER	MIMIC-III
C^{td}-index	HACSurv + SF	0.5987 ± 0.005	0.7467 ± 0.011	0.7962 ± 0.002	$\mathbf{0.7682} \pm 0.018$
	HACSurv + CIF	$\mathbf{0.6426} \pm 0.007$	0.7415 ± 0.012	$\mathbf{0.7975} \pm 0.002$	0.7506 ± 0.017
	HACSurv (I) + SF	0.6350 ± 0.005	$\mathbf{0.7478} \pm 0.009$	0.7972 ± 0.003	0.7635 ± 0.018
	HACSurv (I) + CIF	0.6350 ± 0.005	$\mathbf{0.7478} \pm 0.009$	0.7972 ± 0.003	0.7635 ± 0.018
IBS	HACSurv + SF	0.1205 ± 0.005	$\mathbf{0.0850} \pm 0.003$	0.0669 ± 0.000	0.1629 ± 0.068
	HACSurv + CIF	0.0999 ± 0.008	0.0930 ± 0.003	0.0670 ± 0.000	$\mathbf{0.1540} \pm 0.058$
	HACSurv (I) + SF	$\mathbf{0.0773} \pm 0.002$	0.0922 ± 0.004	$\mathbf{0.0668} \pm 0.000$	0.1690 ± 0.049
	HACSurv (I) + CIF	$\mathbf{0.0773} \pm 0.002$	0.0922 ± 0.004	$\mathbf{0.0668} \pm 0.000$	0.1690 ± 0.049

Figure 4: Inner Copulas Learned by HACsurv. (a), (b) and (c) are the copulas learned in the first stage. (d), (e) and (f) are the corresponding inner copulas learned in the second stage using the Re-generation Trick

D.2 Additional Qualitative Results

As shown in Figure 4, the inner copulas learned using the Re-generation trick proposed in this paper are very close to the copulas learned in the first stage. This confirms that HACSurv can effectively learn the inner copulas of HACs.

D.3 Risk-Specific Results

According to Table 5, HACSurv achieves the best c-index metric across all risks on the synthetic dataset. HACSurv (Independent) obtained the best IBS, indicating that IBS is not a strictly proper evaluation metric. According to Table 6, NeuralFG, which also uses sumo-net as the survival marginal, achieves competitive results with HACSurv (Independent) and HACSurv (Symmetry). However, their results are inferior to HACSurv because they fail to accurately capture the dependency structure among the two risks and censoring. For the SEER dataset,

Table 5: Model Performance Comparison on Synthetic Dataset.

Model	Risk 1		Risk 2		Risk 3	
	C^{td}-index	IBS	C^{td}-index	IBS	C^{td}-index	IBS
DeepHit	0.548 ± 0.011	0.196 ± 0.003	0.618 ± 0.008	0.153 ± 0.004	0.668 ± 0.006	0.105 ± 0.002
DSM	0.552 ± 0.009	0.080 ± 0.002	0.593 ± 0.015	0.085 ± 0.004	0.665 ± 0.004	0.092 ± 0.003
DeSurv	0.534 ± 0.010	0.196 ± 0.002	0.648 ± 0.006	0.131 ± 0.003	0.670 ± 0.004	0.102 ± 0.002
NeuralFG	0.510 ± 0.004	0.211 ± 0.007	0.470 ± 0.048	0.194 ± 0.039	0.668 ± 0.004	0.115 ± 0.003
HACSurv (I)	0.590 ± 0.005	$\mathbf{0.069} \pm 0.001$	0.644 ± 0.008	$\mathbf{0.073} \pm 0.002$	$\mathbf{0.671} \pm 0.003$	$\mathbf{0.090} \pm 0.003$
HACSurv (S)	0.590 ± 0.006	0.085 ± 0.003	0.650 ± 0.008	0.085 ± 0.006	0.669 ± 0.004	0.095 ± 0.004
HACSurv	$\mathbf{0.598} \pm 0.011$	0.088 ± 0.011	$\mathbf{0.659} \pm 0.007$	0.085 ± 0.011	$\mathbf{0.671} \pm 0.004$	0.126 ± 0.003

Table 6: Model Performance Comparison on Framingham and SEER Datasets.

Dataset	Model	Risk 1		Risk 2	
		C^{td}-index	IBS	C^{td}-index	IBS
Framingham	DeepHit	0.668 ± 0.028	0.079 ± 0.003	0.730 ± 0.011	0.099 ± 0.003
	DSM	0.708 ± 0.016	0.078 ± 0.003	0.761 ± 0.017	0.097 ± 0.004
	DeSurv	0.599 ± 0.040	0.092 ± 0.006	0.748 ± 0.004	0.115 ± 0.001
	NeuralFG	0.719 ± 0.015	0.083 ± 0.003	0.774 ± 0.005	0.098 ± 0.003
	HACSurv (I)	$\mathbf{0.720} \pm 0.011$	0.081 ± 0.003	$\mathbf{0.776} \pm 0.007$	0.103 ± 0.006
	HACSurv (S)	0.718 ± 0.017	0.081 ± 0.004	0.773 ± 0.008	0.101 ± 0.006
	HACSurv	$\mathbf{0.720} \pm 0.015$	$\mathbf{0.076} \pm 0.002$	0.774 ± 0.008	$\mathbf{0.094} \pm 0.004$
SEER	DeepHit	0.746 ± 0.000	$\mathbf{0.043} \pm 0.000$	0.835 ± 0.000	0.104 ± 0.001
	DSM	0.735 ± 0.009	0.044 ± 0.000	0.827 ± 0.002	0.093 ± 0.000
	DeSurv	0.739 ± 0.003	0.045 ± 0.000	0.834 ± 0.007	$\mathbf{0.090} \pm 0.002$
	NeuralFG	0.725 ± 0.005	0.045 ± 0.000	0.833 ± 0.002	0.091 ± 0.001
	HACSurv (I)	$\mathbf{0.749} \pm 0.005$	$\mathbf{0.043} \pm 0.000$	0.846 ± 0.001	0.091 ± 0.001
	HACSurv (S)	$\mathbf{0.749} \pm 0.004$	$\mathbf{0.043} \pm 0.001$	0.846 ± 0.001	0.091 ± 0.001
	HACSurv	$\mathbf{0.749} \pm 0.003$	$\mathbf{0.043} \pm 0.000$	$\mathbf{0.847} \pm 0.002$	0.091 ± 0.000

Table 7: Model Performance Comparison on MIMIC-III Dataset.

Metric	Model	Risk 1	Risk 2	Risk 3	Risk 4	Risk 5
C^{td}-index	DeepHit	$\mathbf{0.759} \pm 0.006$	0.844 ± 0.018	0.693 ± 0.018	0.768 ± 0.013	0.760 ± 0.017
	DSM	0.724 ± 0.013	0.829 ± 0.016	0.693 ± 0.021	0.768 ± 0.016	0.745 ± 0.026
	DeSurv	0.751 ± 0.004	$\mathbf{0.859} \pm 0.028$	0.696 ± 0.017	0.742 ± 0.035	0.729 ± 0.023
	NeuralFG	$\mathbf{0.759} \pm 0.010$	0.845 ± 0.031	$\mathbf{0.711} \pm 0.015$	0.757 ± 0.031	0.730 ± 0.042
	HACSurv (I)	0.756 ± 0.010	0.851 ± 0.026	0.695 ± 0.017	0.764 ± 0.020	0.752 ± 0.018
	HACSurv (S)	0.752 ± 0.010	0.848 ± 0.017	0.680 ± 0.020	0.736 ± 0.019	$\mathbf{0.786} \pm 0.015$
	HACSurv	0.752 ± 0.008	0.856 ± 0.021	0.690 ± 0.018	$\mathbf{0.777} \pm 0.012$	0.767 ± 0.034
IBS	DeepHit	0.219 ± 0.008	0.232 ± 0.088	0.324 ± 0.073	0.269 ± 0.127	0.275 ± 0.120
	DSM	0.146 ± 0.025	0.221 ± 0.104	0.228 ± 0.020	0.255 ± 0.128	0.279 ± 0.123
	DeSurv	0.188 ± 0.027	0.245 ± 0.096	0.332 ± 0.053	0.281 ± 0.124	0.295 ± 0.128
	NeuralFG	0.255 ± 0.029	0.238 ± 0.102	0.355 ± 0.092	0.278 ± 0.143	0.253 ± 0.120
	HACSurv (I)	0.138 ± 0.034	0.130 ± 0.049	0.204 ± 0.012	0.205 ± 0.082	0.169 ± 0.067
	HACSurv (S)	$\mathbf{0.103} \pm 0.007$	$\mathbf{0.095} \pm 0.006$	$\mathbf{0.183} \pm 0.044$	0.189 ± 0.045	$\mathbf{0.084} \pm 0.004$
	HACSurv	0.137 ± 0.057	0.105 ± 0.059	0.237 ± 0.095	$\mathbf{0.151} \pm 0.064$	0.184 ± 0.081

we found that the learned dependency was relatively weak, hence the three versions of HACSurv performed similarly. As shown in Table 7 and the overall results in the main text, HACSurv achieves the best overall C-index. HACSurv (Symmetry) obtained the best IBS. Modeling the dependency structure of survival data with six dimensions (including censoring) is challenging. HACSurv (Symmetry), using an Archimedean copula to capture their dependencies, surpasses all existing methods in predicting survival outcomes.

E   Experimental Details

All experiments are conducted with a single NVIDIA RTX3090 GPU. We utilize the AdamW optimizer Loshchilov and Hutter (2018) for training. HACSurv employs the same network architecture across all datasets. Both the shared embedding network and the cause-specific embedding networks consist of two layers with 100 neurons. The monotonic neural networks are composed of three fully connected layers, each containing 100 neurons. For the inner and outer generators of HAC, we use the same network architecture as Ng et al. (2021). For more details, please refer to our code which will be available after acceptance

The split between the training and testing sets for all datasets is 8:2. 20% of the data in the training set is used as a validation set. Experiments across all datasets are conducted in five batches, using 41, 42, 43, 44, and 45 as the random seeds for splitting the datasets. For all comparative methods, a grid search is conducted to select hyperparameters. To ensure a fair comparison, all methods use the same time grid for computing metrics during testing.

For the implementation of the compared algorithms, we use the `PyCox` library Kvamme et al. (2019), which implements DeepHit. The code for NeuralFG, DeSurv, and DSM can be found at `https://github.com/Jeanselme/NeuralFineGray`.