Supplementary Materials for DeepPseudo: Pseudo value based deep learning models for competing risk analysis

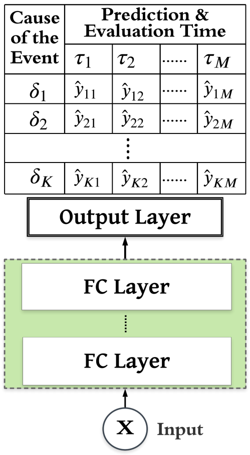
Md Mahmudur Rahman[1], **Shinya Matsuzaki**[2], **Koji Matsuo**[3], **Sanjay Purushotham**[1]

[1] Department of Information Systems, University of Maryland, Baltimore County, Baltimore, Maryland, USA
[2] Osaka University, Osaka, Japan
[3] University of Southern California, Los Angeles, California, USA
mrahman6@umbc.edu, zacky_s@gyne.med.osaka-u.ac.jp, Koji.Matsuo@med.usc.edu, psanjay@umbc.edu

In this supplementary material, we provide more details about our proposed DeepPseudo models, proofs for Lemma 1 and Theorem 1 (in the main paper), more details about the datasets, implementation, and censoring details, along with additional empirical results. We release our code at this link [1].

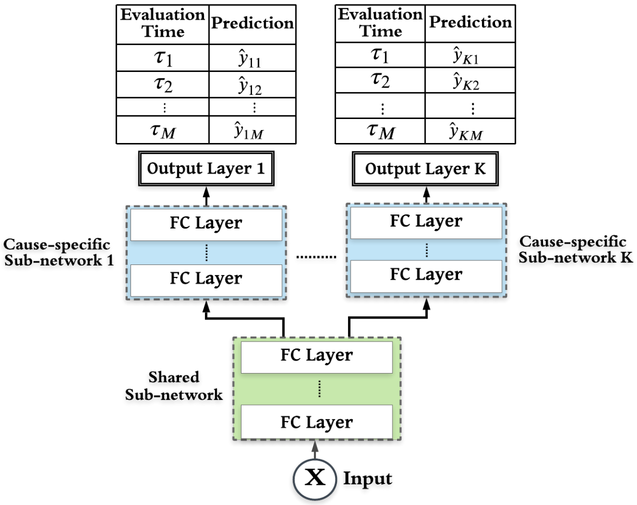
DeepPseudo models

We propose four variants of the DeepPseudo model based on how the pseudo values are calculated and how cause-specific events are modeled. In the following figures (1 to 4), we illustrate the data structure of each of the variants of the DeepPseudo model. In this paper, we consider two evaluation time points (for Marginal variants) or time intervals (for conditional variants) and two causes of the event. We can generalize for more than two causes of the event and more than two evaluation time points/intervals.



Cause of the Event	Prediction & Evaluation Time			
	τ_1	τ_2	......	τ_M
δ_1	$\hat{y}_{11}$	$\hat{y}_{12}$	......	$\hat{y}_{1M}$
δ_2	$\hat{y}_{21}$	$\hat{y}_{22}$	......	$\hat{y}_{2M}$
	⋮			
δ_K	$\hat{y}_{K1}$	$\hat{y}_{K2}$	......	$\hat{y}_{KM}$

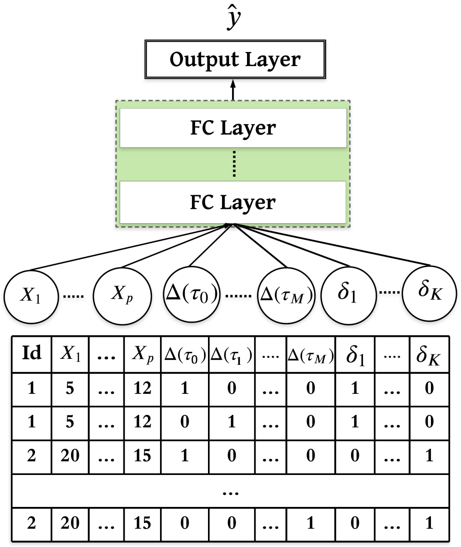
Figure 1: Marginal DeepPseudo Model



Evaluation Time	Prediction
τ_1	$\hat{y}_{11}$
τ_2	$\hat{y}_{12}$
⋮	⋮
τ_M	$\hat{y}_{1M}$

Evaluation Time	Prediction
τ_1	$\hat{y}_{K1}$
τ_2	$\hat{y}_{K2}$
⋮	⋮
τ_M	$\hat{y}_{KM}$

Figure 2: Cause-specific Marginal DeepPseudo Model



$\hat{y}$
Output Layer
FC Layer
FC Layer
X_1 ..... X_p $\Delta(\tau_0)$ ...... $\Delta(\tau_M)$ δ_1 ..... δ_K

Id	X_1	...	X_p	$\Delta(\tau_0)$	$\Delta(\tau_1)$	....	$\Delta(\tau_M)$	δ_1	....	δ_K
1	5	...	12	1	0	...	0	1	...	0
1	5	...	12	0	1	...	0	1	...	0
2	20	...	15	1	0	...	0	0	...	1
...										
2	20	...	15	0	0	...	1	0	...	1

Figure 3: Conditional DeepPseudo Model

[1] https://github.com/umbc-sanjaylab/DeepPseudo_AAAI2021

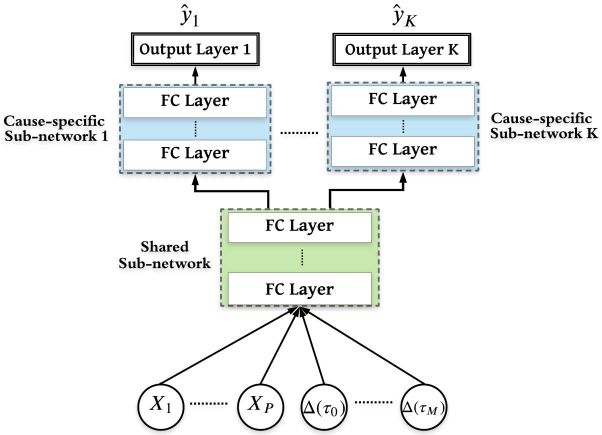


Figure 4: Cause-specific Marginal DeepPseudo Model

Table 1: Illustration of the inputs and outputs for the Marginal DeepPseudo Model

Input				**Output**		
X_1	X_2	…	X_p	**y (Pseudo values)**		
				Cause of the Event	**Evaluation Time**	
					τ_1	τ_2
15	1	…	12	**Cause 1**	-0.0049	1.0806
				Cause 2	-0.0007	-0.0046
3	0	…	9	**Cause 1**	-0.0045	-0.0163
				Cause 2	1.0727	1.0699
⋮						
11	1	…	20	**Cause 1**	-0.0049	-0.0464
				Cause 2	-0.0007	-0.0121

Table 2: Illustration of the inputs and outputs of the Cause-specific Marginal DeepPseudo Model

Input				**Output**			
X_1	X_2	…	X_p	**y (Pseudo values) - Cause 1**		**y (Pseudo values) - Cause 2**	
				τ_1	τ_2	τ_1	τ_2
15	1	…	12	-0.0049	1.0806	-0.0007	-0.0046
3	0	…	9	-0.0045	-0.0163	1.0727	1.0699
⋮							
11	1	…	20	-0.0049	-0.0464	-0.0007	-0.0121

Table 3: Illustration of the inputs and outputs for the Conditional DeepPseudo Model

Input								**Output**
X_1	X_2	…	X_p	$\Delta(\tau_0)$	$\Delta(\tau_1)$	**cause 1**	**Cause 2**	y **(Pseudo values)**
3	1	…	14	1	0	1	0	-0.0049
12	0	…	11	1	0	1	0	-0.0045
⋮								
7	1	…	24	1	0	1	0	-0.0049
3	1	…	14	0	1	1	0	1.1703
12	0	…	11	0	1	1	0	-0.0151
⋮								
7	1	…	24	0	1	1	0	-0.0369
3	1	…	14	1	0	0	1	-0.0007
12	0	…	11	1	0	0	1	1.0727
⋮								
7	1	…	24	1	0	0	1	-0.0007
3	1	…	14	0	1	0	1	-0.0057
12	0	…	11	0	1	0	1	1.0793
⋮								
7	1	…	24	0	1	0	1	-0.0130

Table 4: Illustration of the inputs and outputs for the Cause-specific Conditional DeepPseudo Model

Input						**Output**	
X_1	X_2	…	X_p	$\Delta(\tau_0)$	$\Delta(\tau_1)$	y_1 **(Pseudo values)** **Cause 1**	y_2 **(Pseudo values)** **Cause 2**
3	1	…	14	1	0	-0.0049	-0.0007
12	0	…	11	1	0	-0.0045	1.0727
⋮							
7	1	…	24	1	0	-0.0049	-0.0007
3	1	…	14	0	1	1.1703	-0.0057
12	0	…	11	0	1	-0.0151	1.0793
⋮							
7	1	…	24	0	1	-0.0369	-0.0130

Lemma 1 *The pseudo values for conditional CIFs are conditionally independent given the risk set at different time intervals.*

Proof: Let us consider two intervals $(\tau_{P-1}, \tau_P]$ and $(\tau_{Q-1}, \tau_Q]$ with risk sets R_{P-1} and R_{Q-1}. These intervals are non-overlapping from our setup (as mentioned in the main paper). Then the pseudo values for the conditional CIFs for these intervals are given by (equation (5) from the main paper):

$$\hat{F}_{ik}(\tau_P|R_{P-1}) = R_{P-1} * \hat{F}_k(\tau_P|R_{P-1}) - (R_{P-1} - 1) * \hat{F}_k^{-i}(\tau_P|R_{P-1}) \tag{1}$$

$$\hat{F}_{ik}(\tau_Q|R_{Q-1}) = R_{Q-1} * \hat{F}_k(\tau_Q|R_{Q-1}) - (R_{Q-1} - 1)* \hat{F}_k^{-i}(\tau_Q|R_{Q-1}) \quad (2)$$

To show these pseudo values are conditionally independent given risk sets, we need to show that the conditional CIFs i.e., $\hat{F}_k(\tau_P|R_{P-1})$ and $\hat{F}_k(\tau_Q|R_{Q-1})$ are independent. We note that the conditional CIFs, $\hat{F}_k(\tau_P|R_{P-1}) = P(T \leq \tau_P, \delta = k|T \geq \tau_{P-1})$ and $\hat{F}_k(\tau_Q|R_{Q-1}) = P(T \leq \tau_Q, \delta = k|T \geq \tau_{Q-1})$ are independent as the intervals are non-overlapping. Therefore, the probability of occurrence of an event at an interval given the risk set is independent to the probability of occurrence of an event at other intervals given the risk set. ■

Remark: An implication of the above lemma is that there is no within-subjects correlation between the risk set of the intervals. This is analogous to the findings in (Zhao and Feng 2019).

Theorem 1 *The pseudo values for the marginal CIF at a time point τ_M is the product of the pseudo values for conditional CIFs of the previous intervals up to time point τ_M.*

Proof: From Lemma 1, the conditional CIFs for cause k given risk set of the intervals are independent. Therefore, we can write (using Lemma 1 for multiple consecutive intervals),

$$\begin{aligned} F_k(\tau_M|T \geq \tau_0) &= P(T \leq \tau_M, \delta = k|T \geq \tau_0) \\ &= P(\tau_0 \leq T \leq \tau_1, \delta = k|T \geq \tau_0) \\ &* P(\tau_1 \leq T \leq \tau_2, \delta = k|\tau_0 \leq T \leq \tau_1) \\ &* P(\tau_2 \leq T \leq \tau_3, \delta = k|\tau_1 \leq T \leq \tau_2)........ \\ &* P(\tau_{M-1} \leq T \leq \tau_M, \delta = k|\tau_{M-2} \leq T \leq \tau_{M-1}) \end{aligned} \quad (3)$$

Pseudo values for conditional CIF for each subject i are calculated from the estimate of the conditional CIF, and these pseudo values can be approximated by independent and identically distributed variables [Lemma 2 from (Graw, Gerds, and Schumacher 2009)]. Therefore, we can estimate the pseudo values for Marginal CIF at a time point τ_M by multiplying the pseudo values for conditional CIFs of the previous intervals up to that time point τ_M. ■

Handling covariate dependent censoring

In the presence of Covariate Dependent Censoring (CDC), our DeepPseudo models use the `modified pseudo values` (Binder, Gerds, and Andersen 2014) as shown below:

$$\tilde{F}_{ik}(t) = n\tilde{F}_k(t) - (n-1)\tilde{F}_k^{-i}(t)$$

where, $\tilde{F}_k(t)$ is defined as

$$\tilde{F}_k(t) = \frac{1}{n}\sum_{i=1}^{n} \frac{N_{ik}(t)}{\hat{C}_i(\tilde{T}_{i^-}|Z_i)}$$

$N_{ik}(t)$ counts observed cause k events for subject i and $\hat{C}_i(\tilde{T}_{i^-}|Z_i)$ is the estimated censoring distribution for subject i. A standard Cox proportional hazard model $C_i(t|Z_i) = exp(-B(t)exp(\gamma Z_i))$ is used to estimate the censoring distribution, where $B(t)$ is the cumulative censoring baseline hazard and γ is the regression coefficient of the censoring distribution.

For estimating $\tilde{F}_k^{-i}(t)$, the leave-i-out estimator (obtained by eliminating i^{th} observation) for CIF, *Binder et al. 2014* proposed the following three estimation techniques.

1. Re-fitting the censoring model for n leave-i-out samples; $i = 1, 2, ..., n$, to obtain estimators $\hat{\gamma}^{(i)}$ and $\hat{B}^{(i)}(.)$ for the parameters of the censoring model.
2. Fitting the censoring model once to obtain estimators $\hat{\gamma}$ and $\hat{B}(.)$, and using the same censoring model for all subjects in the sample.
3. Re-using the censoring coefficients but re-estimating the cumulative baseline hazard for each leave-i-out sample.

We considered the second technique to calculate pseudo values (referred to as `modified pseudo values` in our paper) in the presence of covariate-dependent censoring as it requires less computation and time. The other two techniques are computationally expensive as they require fitting the censoring model (variant 1) or cumulative baseline hazard (variant 3) many times. The performance of DeepPseudo models on CDC is shown in the Tables 7 and 8 under experiments.

Experiments

Datasets

We conducted experiments on two real-world competing risk datasets (SEER and WIHS) and one synthetic data. Please refer to the main paper for the description of these datasets. In all the tables in this supplementary materials, we used SEER data with two causes of the event. Table 7 shows the descriptive statistics of the two real-world datasets.

Evaluation Metrics:

In our main paper and supplementary materials, we use two evaluation metrics for performance comparison, namely (a) the cause-specific time-dependent concordance index and (b) Brier score.

Cause-specific time-dependent concordance index: Concordance index (Gerds et al. 2013) (denoted as C-index in our paper) is used to evaluate the discriminatory ability as well as the predictive accuracy of the models. We extend the C-index metric for CRA (Cause-specific time-dependent concordance index) to evaluate the cause-specific discriminative performance at different time horizons. Consider the estimated cumulative incidence function, defined as $\hat{F}_k(t|x) = P(T \leq t, k = k|X = x)$. Then, the time-dependent C-index for cause k at time horizon t can be calculated as

$$\begin{aligned} C_k(t) =& P(\hat{F}_k(t|X_i) > \hat{F}_k(t|X_j)|\{\delta_i = k\} \cap \{T_i \leq t\} \\ & \cap \{T_i < T_j \cup \delta_j \neq k\}) \end{aligned}$$

The C-index is the ratio of the number of correct pairs (patients with less survival time has higher risk than others with more survival time) and the total number of acceptable

Table 5: Descriptive Statistics of the two Real-World Competing Risk Datasets

Dataset	No. of Observations	No. of Uncensored Cause 1 (%)	No. of Uncensored Cause 2 (%)	No. of Censored (%)	No. of Features (Real, Categorical)	Event Time (in months)			Censoring Time (in months)		
						min (Cause 1, Cause 2)	max (Cause 1, Cause 2)	mean (Cause 1, Cause 2)	min	max	mean
SEER	28366	6575 (23.18)	2381 (8.39)	19410 (68.43)	13 (1, 12)	(0, 0)	(324, 338)	(28.95, 78.28)	0	347	94.59
WIHS	1164	679 (58.33)	359 (30.84)	126 (10.82)	4 (2, 2)	(0.24, 0.24)	(128.4, 129.6)	(32.42, 25.81)	2.4	129.6	82.88

pairs, in which one patient got exposed to the cause k at a particular time while the other is uncensored without experiencing any event by that time.

Brier Score: Brier Score (Mogensen, Ishwaran, and Gerds 2012) is calculated as an adjusted mean square error used in survival analysis. Mathematically, it is given by:

$$BS_k(t) = E[(1(T_i \leq t, \delta_i = k) - \hat{F}_k(t|X_i))^2]$$

These metrics are flexible to be evaluated at different time horizons. The effect of censoring in these metrics can be adjusted using weights as defined in (Gerds et al. 2013) and (Mogensen, Ishwaran, and Gerds 2012). In our experiments, we used weighted C-index and weighted Brier scores for comparing all the CRA models.

Models Compared and Implementation details

We provide a brief description of the models compared in our paper, along with how they were implemented for our experiments.

Statistical Models

- **Cause-specific Hazard Model** (Putter, Fiocco, and Geskus 2007): The cause-specific hazard model estimates the effect of covariates on the cause-specific hazard function. We fit k independent Cox-proportional hazard models for k causes of the event, assuming competing events as right censoring. We use the `CSC` function of the `R` package `riskRegression` to implement the model. We removed linearly dependent columns from the dataset (to avoid singularity issues) before implementing this model.
- **Fine & Gray Subdistribution Hazard Model** (Fine and Gray 1999): This CRA modeling technique directly models the effect of covariates on the CIF through a subdistribution hazard function. The model also make proportional hazard assumption. We use the `FGR` function of the `R` package `riskRegression` to implement the model. We removed linearly dependent columns from the dataset (to avoid singularity issues) before using this model for comparison.
- **Pseudo values-based GEE approach (GEE)** (Klein and Andersen 2005): First, we compute the pseudo values for the CIF from the Aalen-Johansen estimator of CIF. The pseudo values are used as quantitative responses in a generalized linear model with a complementary log-log link function. A Generalized Estimating Equation (GEE) is used to estimate the model's parameter. We use the `geese` function of the `R` package `geepack` which employ the GEE to estimate the parameters.

Machine Learning Models

- **Random Survival Forest (RSF)** (Ishwaran et al. 2014): This is a fully non-parametric approach for estimating the CIF using random forest. We implement this model using the `rfsrc` function of the R package `randomForestSRC`. We can set the *cause* j for competing risk analysis, where j is the integer value indicating the event of interest. We use the default hyper-parameter settings as suggested by (Ishwaran et al. 2014) in our experiments as they provided the best results after a random search.
- **Deep Multi-Gaussian Processes (DMGP)** (Alaa and van der Schaar 2017): This approach uses a variational inference algorithm for learning the model parameters and to handle right censoring. For our experiments, we use the number of epochs as 50, the number of iterations as 100, batch size as 64 based on the validation performance and keep other hyper-parameters as in default settings.

Deep Learning Models

- **DeepHit** (Lee et al. 2018): DeepHit model uses a deep neural network to learn the distribution of survival times directly. We use the publicly available code [2] to implement the model. In our experiments, as suggested by the authors, we perform a random search to find the best hyper-parameters based on the average C-index performance for all the causes of the event on the validation dataset.
- **Our DeepPseudo models:** Please refer to the main paper about the description of our proposed models: Marginal DeepPseudo, Cause-specific Marginal DeepPseudo, Conditional DeepPseudo, and Cause-specific Conditional DeepPseudo models

Implementation details for DeepPseudo models: We create 5 sets of 5-Fold cross-validation dataset to compare model's performance using the weighted C-index and the weighted Brier score evaluation metrics. In all these sets, we stratify the folds, i.e., we maintain a constant ratio of uncensored and censored subjects in each fold. For preprocessing, we convert the categorical variables into one-hot-encoded dummy variables, and standardize the continuous valued variables (using training folds data points). We choose the best hyperparameter setting based on the average C-index as the performance metric on the validation dataset by random search over hyperparameter space. We vary the hyperparameters as follows: number of hidden layers: [2, 3, 4, 5, 6],

[2] https://github.com/chl8856/DeepHit

number of nodes in hidden layers: [32, 64, 128, 512], regularization: dropout with 0.4 or l2 with penalty weights [1.0, 0.1, 0.01, 0.001], `selu` activation function, batch size: [32, 64, 128, 512] and `Adam` optimizer (Kingma and Ba 2014) with a learning rate: [0.001, 0.0001, 0.00001]. We perform early stopping and choose the best model based on the performance on the validation data.

Implementation steps: Marginal DeepPseudo Model

- Estimate the pseudo values for marginal CIF using the `jackknife` function of `R` package `prodlim` for each cause and evaluation time point. The pseudo values will be used as the output variable in this model.
- Convert the categorical variables into one-hot-encoded dummy variables and standardize the continuous variables.
- Use the covariates as input to a feed-forward deep neural network shown in Figure 1.
- Perform 5-fold Cross validation - train the network on 3 folds, find best hyperparameters (via random search) by monitoring performance on the validation fold, and predict the pseudo values for marginal CIF using the model on test fold.

Implementation steps: Conditional DeepPseudo Model

- Divide the survival time into M intervals: $(0, \tau_1], (\tau_1, \tau_2], ..., (\tau_{M-1}, \tau_M]$.
- Compute the pseudo values for conditional CIF for each interval.
- Convert the categorical variables, time intervals and causes of the event into one-hot-encoded dummy variables and standardize the continuous variables.
- Use the covariates, time intervals $(\Delta(\tau_0), ..., \Delta(\tau_M))$ and causes of the event $(\delta_1, ..., \delta_K)$ as input in to a feed-forward deep neural network shown in Figure 3.
- Perform 5-fold Cross validation - train the network on 3 folds, find best hyperparameters (via random search) by monitoring performance on the validation fold, and predict the pseudo values for conditional CIF using the model on test fold.
- For computing the pseudo values for marginal CIF, multiply the pseudo values for conditional CIF of the previous intervals up to evaluation time τ_M.

The implementation details for the cause-specific Marginal DeepPseudo and cause-specific Conditional DeepPseudo models are similar to the Marginal and Conditional DeepPseudo models, respectively, with these models using cause-specific deep sub-networks for modeling each cause separately.

All our experiments were run on a 128GB RAM Intel Xeon dual 10-core processor with 3 GPUs. We will release our code to encourage reproducibility.

Statistical significance testing

To determine the significance of the mean difference in the C-index of the different CRA models used in Table 1 in the main paper, we first perform analysis of variance (ANOVA) test. However, the ANOVA test is unable to answer the question if there is any significant difference in the mean C-index between a pair of the algorithms. Therefore, we perform *Tukey's HSD test* (Abdi and Williams 2010), a pairwise statistical significant test, between the best DeepPseudo model (Marginal DeepPseudo for SEER and Conditional DeepPseudo for WIHS and Synthetic datasets) and other baseline models after getting overall significant difference in mean C-index among the algorithms by ANOVA test. We also tested the significance in the performance using Fisher's Least Significant Difference (LSD) test and pairwise t-test. However, we found that Tukey's HSD test is more suitable for our problem (c-indices are independent within and among models, and the models for each mean in the test are normally distributed) and hence used it for statistical significance testing in our paper.

Censoring Settings

We examine and compare our model's performance in handling censoring by evaluating it on different censoring settings. (a) *Incremental censoring*: Incrementally we add censored observations to a fixed number of uncensored observations. This helps us to study impact of censoring on uncensored observations in an increasing dataset. (b) *Induced censoring*: Starting with uncensored observations, we gradually induce censoring by changing (flipping) the label of the uncensored observations. This helps us to study the impact of increasing censoring ratio in a fixed size dataset. In our experiments on SEER dataset, for incremental censoring setup, we incrementally add 1k or 5k observations to a fixed number of uncensored observations; and for the induced censoring, we change labels for 1k uncensored observations in each setting for a fixed sized dataset. The censoring setups are described below:

Incremental Censoring

- Start with all uncensored observations (no censoring or 0k censoring).
- Generate multiple censored datasets as follows: Incrementally add 1 thousand [1k] censored observations to the uncensored observations to get 1k censored dataset, then add 1k censored observations to 1k censored dataset to get 2k censored dataset, and so on till 5k censored dataset, add another 5k censored observations to 5k censored dataset to get 10k, and add 5k more to 10k censored dataset to get 15k censored dataset.
- The number of uncensored observations remain fixed in all the settings.

We evaluate the incremental censoring performance in two ways: (a) CRA models' performance on the uncensored + censored observations in all the above settings (see table 10, and 11) (b) CRA models' performance only on the uncensored observations in all of the above settings (see table 12).

Table 6: Model Performance Comparisons using cause-specific time dependent C-index (mean and 95% confidence interval)

Dataset	Cause of the Event	Evaluation Time	Statistical Models			Machine Learning Models		Deep Learning Models				
			Cause-specific Hazard	Fine & Gray	GEE	RSF	DMGP	DeepHit	Marginal DeepPseudo	CS-Marginal DeepPseudo	Conditional DeepPseudo	CS-Conditional DeepPseudo
Seer Data	Cause 1	1 year	0.8675*** (0.8645, 0.8704)	0.8649*** (0.8618, 0.8679)	0.8675 *** (0.8646, 0.8704)	0.8703* (0.8678, 0.8729)	0.8713 (0.8683, 0.8743)	0.8761 (0.8726, 0.8796)	0.8767 (0.8736, 0.8798)	**0.8773** **(0.8743, 0.8802)**	0.8659 (0.8627, 0.8691)	0.8647 (0.8612, 0.8681)
		5 years	0.8025*** (0.8002, 0.8049)	0.8048** (0.8023, 0.8073)	0.8038 *** (0.8016, 0.8061)	0.8035*** (0.8008, 0.8063)	0.8030 *** (0.7998, 0.8062)	0.8080 (0.8051, 0.8109)	0.8122 (0.8095, 0.8148)	**0.8130** **(0.8106, 0.8155)**	0.8055 (0.8032, 0.8077)	0.8027 (0.7997, 0.8058)
	Cause 2	1 year	0.8384 (0.8273, 0.8495)	0.7787*** (0.7672, 0.7903)	0.8005 *** (0.7898, 0.8111)	0.8159*** (0.8080, 0.8239)	0.7634 *** (0.7502, 0.7767)	0.8458 (0.8351, 0.8565)	**0.8520** **(0.8431, 0.8608)**	0.8412 (0.8320, 0.8503)	0.8376 (0.8285, 0.8466)	0.8451 (0.8357, 0.8545)
		5 years	0.8027 (0.7967, 0.8087)	0.7831*** (0.7769, 0.7893)	0.7854 *** (0.7788, 0.7921)	0.7794*** (0.7735, 0.7852)	0.7684 *** (0.7633, 0.7734)	0.8028 (0.7969, 0.8087)	0.8077 (0.8016, 0.8138)	0.8089 (0.8028, 0.8150)	0.7991 (0.7907, 0.8074)	**0.8138** **(0.8078, 0.8197)**
WIHS data	Cause 1	1 year	0.7344 (0.7140, 0.7548)	0.6936 (0.6714, 0.7157)	0.7041 (0.6818, 0.7264)	0.7024 (0.6804, 0.7243)	0.7222 (0.7023, 0.7421)	0.7207 (0.6986, 0.7428)	0.7310 (0.7111, 0.7509)	**0.7342** **(0.7120, 0.7565)**	0.7318 (0.7128, 0.7509)	0.7225 (0.7003, 0.7446)
		5 years	0.6462 (0.6339, 0.6584)	0.6352 (0.6227, 0.6477)	0.6432 (0.6308, 0.6556)	0.6023*** (0.5922, 0.6124)	0.6257 * (0.6130, 0.6385)	0.6076 *** (0.5931, 0.6222)	0.6189 (0.6064, 0.6315)	0.6156 (0.6040, 6272)	**0.6536** **(0.6436, 0.6636)**	0.6381 (0.6257, 0.6505)
	Cause 2	1 year	0.6681 (0.6513, 0.6848)	0.6463*** (0.6295, 0.6631)	0.6717 (0.6526, 0.6908)	0.6909 (0.6746, 0.7072)	0.6805 (0.6663, 0.6947)	0.6802 (0.6623, 0.6982)	0.7015 (0.6855, 0.7175)	**0.7019** **(0.6858, 0.7179)**	0.6982 (0.6820, 0.7143)	0.7004 (0.6869, 0.7140)
		5 years	0.6386*** (0.6262, 0.6511)	0.6368*** (0.6244, 0.6492)	0.6609 . (0.6480, 0.6738)	0.6586* (0.6463, 0.6709)	0.6761 (0.6678, 0.6845)	0.6556 * (0.6437, 0.6674)	0.6707 (0.6589, 0.6826)	0.6681 (0.6552, 0.6811)	**0.6835** **(0.6716, 0.6953)**	0.6653 (0.6532, 0.6775)
Synthetic Data	Cause 1	1 year	0.5806*** (0.5765, 0.5846)	0.5812*** (0.5772, 0.5852)	0.5811 *** (0.5772, 0.5850)	0.6277*** (0.6230, 0.6324)	0.7508 ** (0.7479, 0.7537)	0.7532 . (0.7500, 0.7564)	0.7554 (0.7526, 0.7581)	0.7490 (0.7461, 0.7519)	**0.7606** **(0.7579, 0.7633)**	0.7529 (0.7501, 0.7556)
		5 years	0.5572*** (0.5546, 0.5599)	0.5574*** (0.5547, 0.5600)	0.5576 *** (0.5549, 0.5602)	0.5763*** (0.5729, 0.5798)	0.6761 *** (0.6726, 0.6796)	0.6824 *** (0.6784, 0.6865)	0.6707 (0.6678, 0.6736)	0.6805 (0.6774, 0.6836)	**0.7028** **(0.7000, 0.7056)**	0.6903 (0.6872, 0.6933)
	Cause 2	1 year	0.5844*** (0.5811, 0.5878)	0.5855*** (0.5821, 0.5889)	0.5849 *** (0.5817, 0.5882)	0.6241*** (0.6208, 0.6274)	0.7483 *** (0.7442, 0.7525)	0.7516 * (0.7477, 0.7555)	0.7543 (0.7513, 0.7573)	0.7487 (0.7454, 0.7520)	**0.7598** **(0.7566, 0.7631)**	0.7571 (0.7548, 0.7594)
		5 years	0.5589*** (0.5550, 0.5627)	0.5590*** (0.5552, 0.5629)	0.5587 *** (0.5549, 0.5626)	0.5725*** (0.5688, 0.5761)	0.6736 *** (0.6700, 0.6772)	0.6788 *** (0.6730, 0.6846)	0.6638 (0.6604, 0.6672)	0.6746 (0.6719, 0.6773)	**0.6989** **(0.6959, 0.7018)**	0.6895 (0.6867, 0.6922)

Tukey's HSD test - statistically significant codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1, (Read p '***' as significant at p% level of significance)

Table 7: C-Index Performance Comparison of Marginal DeepPseudo models using pseudo values calculated assuming Covariate Independent Censoring (CIC) and modified pseudo values calculated assuming Covariate Dependent Censoring (CDC) and using different activation functions in output layer

Cause of the Event	Evaluation Time	CIC (selu)	CDC (selu)	CIC (cloglog)	CDC (cloglog)
Cause 1	**1 year**	0.8750	0.8737	0.8758	0.8755
	5 years	0.8106	0.8099	0.8104	0.8102
Cause 2	**1 year**	0.8465	0.8520	0.8363	0.8329
	5 years	0.8047	0.8088	0.7998	0.8002

*CIC: Covariate Independent Censoring, CDC: Covariate Dependent Censoring

Induced Censoring

- Start with all uncensored observations (no censoring).
- We create multiple censored datasets by gradually inducing censoring to the uncensored observations by changing (flipping) the label of uncensored subjects. In particular, we use these settings to create different induced censoring datasets: induce 1 thousand (1k dataset), then induce 1 thousand more on the previous 1k dataset to get 2k induced censoring dataset and so on to get 3k, 4k and 5k induced censoring datasets.
- The number of uncensored observations decreases by 1 thousand in the above settings, but the total number of observations remain fixed in all settings.
- The ratio of the number of cause 1 and cause 2 of the event is kept fixed while inducing the censored observations.

Please refer to Table 2 in the main paper, and Tables 10,11,12 in this supplementary materials paper for the experimental results for the censoring settings.

Table 8: Brier Score Performance Comparison of Marginal DeepPseudo models using pseudo values calculated assuming Covariate Independent Censoring (CIC) and modified pseudo values calculated assuming Covariate Dependent Censoring (CDC) and using different activation functions in output layer

Cause of the Event	Evaluation Time	CIC (selu)	CDC (selu)	CIC (cloglog)	CDC (cloglog)
Cause 1	**1 year**	0.0514	0.0514	0.0538	0.0548
	5 years	0.1265	0.1291	0.1251	0.1281
Cause 2	**1 year**	0.0173	0.0173	0.0176	0.0175
	5 years	0.0665	0.0672	0.0688	0.0675

*CIC: Covariate Independent Censoring, CDC: Covariate Dependent Censoring

Results and Discussion

The model comparison results using C-index are shown in Table 6 (Table 1 in the main paper). From this table, we notice that for the SEER dataset, our Marginal DeepPseudo model showed statistically significant performance over all

Table 9: Model Performance Comparisons using Brier Score (mean and SD)

Dataset	Cause of the Event	Evaluation Time	Statistical Models			Machine Learning Models		Deep Learning Models				
			Cause-specific Hazard	Fine & Gray	GEE	RSF	DMGP	DeepHit	Marginal DeepPseudo	CS-Marginal DeepPseudo	Conditional DeepPseudo	CS-Conditional DeepPseudo
SEER Data	**Cause 1**	**1 year**	0.0509 (0.0009)	0.0486 (0.0008)	0.0512 (0.0009)	0.0519 (0.0007)	0.0651 (0.0018)	0.0598 (0.0065)	0.0511 (0.0010)	0.0512 (0.0013)	0.0507 (0.0011)	0.0539 (0.0010)
		5 years	0.1298 (0.0018)	0.1286 (0.0017)	0.1290 (0.0018)	0.1280 (0.0016)	0.1408 (0.0026)	0.1321 (0.0076)	0.1252 (0.0015)	0.1258 (0.0018)	0.1351 (0.0014)	0.1202 (0.0017)
	Cause 2	**1 year**	0.0174 (0.0004)	0.0172 (0.0004)	0.0173 (0.0004)	0.0174 (0.0004)	0.0194 (0.0006)	0.0173 (0.0003)	0.0172 (0.0004)	0.0172 (0.0004)	0.0173 (0.0004)	0.0172 (0.0004)
		5 years	0.0656 (0.0006)	0.0655 (0.0006)	0.0661 (0.0006)	0.0665 (0.0005)	0.0671 (0.0009)	0.0656 (0.0007)	0.0661 (0.0006)	0.0656 (0.0006)	0.0765 (0.0007)	0.0677 (0.0006)
WIHS Data	**Cause 1**	**1 year**	0.1453 (0.0121)	0.1445 (0.0123)	0.1432 (0.0118)	0.1505 (0.0120)	0.1430 (0.0088)	0.1695 (0.0261)	0.1476 (0.0123)	0.1469 (0.0121)	0.1474 (0.0119)	0.1483 (0.0121)
		5 years	0.2307 (0.0120)	0.2347 (0.0124)	0.2341 (0.0120)	0.2382 (0.0126)	0.2474 (0.0075)	0.2463 (0.0149)	0.2307 (0.0117)	0.2301 (0.0120)	0.5825 (0.0180)	0.2234 (0.0113)
	Cause 2	**1 year**	0.0800 (0.0039)	0.0794 (0.0043)	0.0813 (0.0034)	0.0839 (0.0036)	0.1007 (0.0030)	0.1031 (0.0178)	0.0803 (0.0043)	0.0817 (0.0039)	0.0787 (0.0042)	0.0857 (0.0042)
		5 years	0.2559 (0.0160)	0.2544 (0.0157)	0.2604 (0.0209)	0.2471 (0.0188)	0.2451 (0.0150)	0.2524 (0.0225)	0.2478 (0.0204)	0.2435 (0.0187)	0.3960 (0.0208)	0.2565 (0.0192)
Synthetic Data	**Cause 1**	**1 year**	0.2221 (0.0015)	0.2227 (0.0016)	0.2221 (0.0015)	0.2198 (0.0014)	0.2059 (0.0015)	0.2246 (0.0052)	0.2263 (0.0024)	0.2228 (0.0034)	0.2243 (0.0029)	0.2225 (0.0032)
		5 years	0.2943 (0.0065)	0.2967 (0.0066)	0.2952 (0.0065)	0.3066 (0.0063)	0.2742 (0.0075)	0.3242 (0.0408)	0.2941 (0.0083)	0.3050 (0.0101)	0.5873 (0.0111)	0.3148 (0.0129)
	Cause 2	**1 year**	0.2179 (0.0013)	0.2185 (0.0014)	0.2179 (0.0014)	0.2161 (0.0014)	0.2033 (0.0013)	0.2205 (0.0055)	0.2218 (0.0019)	0.2181 (0.0024)	0.2194 (0.0027)	0.2180 (0.0024)
		5 years	0.2992 (0.0065)	0.3015 (0.0066)	0.2992 (0.0065)	0.3117 (0.0063)	0.2748 (0.0068)	0.3277 (0.0388)	0.3018 (0.0087)	0.3097 (0.0095)	0.5871 (0.0143)	0.3203 (0.0094)

the other models except the DeepHit model in almost all the cases. All the DeepPseudo model variants performed similar or better than the DeepHit model in most cases. On the WIHS dataset, our Conditional DeepPseudo model gave a significantly better performance than all the other baseline models, especially for 5 years of evaluation time. On the Synthetic dataset, our Conditional DeepPseudo model showed statistically significant results compared to all the other models. The model performance comparison results with respect to the Brier scores are shown in table 9. A lower value of Brier scores indicates better model performance. From this Table 9, we see that brier scores for our DeepPseudo models are similar or better than other models in many cases. From these tables, we can see that our Marginal DeepPseudo model provides overall better performance than all the other variants of our DeepPseudo models.

Censoring settings: Table 10 shows the results for concordance index performance of the models for induced censoring and incremental censoring settings on SEER data. In induced censoring settings, our Marginal DeepPseudo model outperforms the other models in most cases. Among the 4 variants of our proposed DeepPseudo models, the Marginal DeepPseudo model gives better performance overall. In the incremental censoring setup, we observe that the C-index increases with the increase of censored observations for our proposed models, whereas the Brier scores decrease with increased censored observations as expected. The Marginal DeepPseudo model gives overall better performance compared to all other models (including different variants of DeepPseudo models).

In Table 11 we show the Brier scores for all the models for induced censoring and incremental censoring settings on SEER data. It is interesting to note that our Marginal DeepPseudo model outperforms the state-of-the-art DeepHit model, especially for Cause 1. On the other hand, our Marginal DeepPseudo performs better than the Fine & Gray model but achieves a similar result as RSF on almost all the cases except for the 5 year evaluation time point. We plan to conduct more investigations in the future to study the Brier score results of our models carefully.

To study how incrementally adding censored observations affects the model performance on the uncensored observations, we compared the results only on the common uncensored patients across the incrementally censored datasets. In particular, we compared our Marginal DeepPseudo and Cause-specific Marginal DeepPseudo with the DeepHit model as these models were designed to handle censoring. Table 12 shows the comparison results, and we see that our DeepPseudo models, especially the Marginal DeepPseudo model, perform significantly better than the DeepHit model with respect to both C-index and Brier Score. **Remark:** This empirically demonstrates that our model, which uses pseudo values, can handle censoring better than the DeepHit model, which uses a complex objective function to handle censoring.

DeepHit with and without ranking loss: The DeepHit model proposed using a combined loss function which consists of negative log-likelihood loss (modified to handle censoring) and cause-specific ranking loss (Lee et al. 2018) for training with censored observations. To study if the cause-specific ranking loss had any impact on the DeepHit results, we conducted additional experiments and compared it with our proposed DeepPseudo models. In table 13, we show the C-index and Brier scores comparison of the DeepPseudo models with the DeepHit model with and without the ranking loss (i.e., only negative log-likelihood loss) on the SEER and Synthetic datasets. Among the DeepHit models for the C-index metric, the DeepHit model with ranking loss performs marginally better for cause 1 of the event, whereas the DeepHit model without ranking loss performs better for cause 2 of the event. Our Marginal DeepPseudo model shows overall better performance than all other models in the SEER dataset. On the other hand, the Conditional DeepPseduo model outperforms all the other models on the Synthetic dataset. We notice that Brier score performance is better for the Cause-specific Marginal DeepPseudo model than other models in both SEER and Synthetic datasets.

Covariate Dependent Censoring: Tables 7 and 8 show the performance comparison of the Marginal DeepPseudo model that uses `pseudo values`, computed assuming covariate independent censoring and `modified pseudo values`, computed assuming covariate dependent censoring. We also use `selu` and `cloglog` activation functions in the output layer and compare the model's performance. Please note that we can use both activation functions in our models, but we will get different predictions and interpretations. Using `selu`, we will get the prediction of true pseudo values, whereas using `cloglog`, we will get a transformed prediction of the true pseudo values bounded by $[0, 1]$.

The Marginal DeepPseudo models that use `selu` activation function in the output layer give better performance (both C-index and Brier Score) than the Marginal DeepPseudo models that use `cloglog` activation function. The DeepPseudo models with modified pseudo values (CDC) and `selu` activation provided similar C-index and Brier scores as the DeepPseudo model with Jackknife pseudo values (CIC) with `selu` activation. Therefore, empirically it is evident that the DeepPseudo model's performance is less sensitive to the covariate dependent censoring.

Runtime: For similar hyperparameter settings, DeepPseudo model takes around 187 to 192 seconds for $5k$ and $1M$ model parameters, while DeepHit takes 265 to 307 seconds for the same model capacity.

References

Abdi, H.; and Williams, L. J. 2010. Tukey's honestly significant difference (HSD) test. *Encyclopedia of research design* 3: 583–585.

Alaa, A. M.; and van der Schaar, M. 2017. Deep multi-task gaussian processes for survival analysis with competing risks. In *NeurIPS*, 2326–2334. Curran Associates Inc.

Binder, N.; Gerds, T. A.; and Andersen, P. K. 2014. Pseudo-observations for competing risks with covariate dependent censoring. *Lifetime data analysis* 20(2): 303–315.

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

Gerds, T. A.; Kattan, M. W.; Schumacher, M.; and Yu, C. 2013. Estimating a time-dependent concordance index for survival prediction models with covariate dependent censoring. *Statistics in Medicine* 32(13): 2173–2184.

Graw, F.; Gerds, T. A.; and Schumacher, M. 2009. On pseudo-values for regression analysis in competing risks models. *Lifetime Data Analysis* 15(2): 241–255.

Ishwaran, H.; Gerds, T. A.; Kogalur, U. B.; Moore, R. D.; Gange, S. J.; and Lau, B. M. 2014. Random survival forests for competing risks. *Biostatistics* 15(4): 757–773.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Klein, J. P.; and Andersen, P. K. 2005. Regression modeling of competing risks data based on pseudovalues of the cumulative incidence function. *Biometrics* .

Lee, C.; Zame, W. R.; Yoon, J.; and van der Schaar, M. 2018. Deephit: A deep learning approach to survival analysis with competing risks. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Mogensen, U. B.; Ishwaran, H.; and Gerds, T. A. 2012. Evaluating random forests for survival analysis using prediction error curves. *Journal of statistical software* 50(11): 1.

Putter, H.; Fiocco, M.; and Geskus, R. B. 2007. Tutorial in biostatistics: competing risks and multi-state models. *Statistics in medicine* 26(11): 2389–2430.

Zhao, L.; and Feng, D. 2019. DNNSurv: Deep Neural Networks for Survival Analysis Using Pseudo Values. *arXiv preprint arXiv:1908.02337* .

Table 10: C-Index (mean and SD) for different censoring settings. Dataset: SEER. 0k means no censored observations, 1k corresponds to 1k censored observations in the dataset, and so on.

			Induced censoring						**Incremental censoring**							
Algorithms	**Cause of the Event**	**Evaluation Time**	**0k**	**1k**	**2k**	**3k**	**4k**	**5k**	**0k**	**1k**	**2k**	**3k**	**4k**	**5k**	**10k**	**15k**
Marginal DeepPseudo	**Cause 1**	**1 year**	0.7739 (0.0170)	**0.7748** **(0.0086)**	**0.7753** **(0.0127)**	0.7744 (0.0115)	**0.7727** **(0.0166)**	0.7654 (0.0220)	0.7764 (0.0047)	0.7909 (0.0035)	**0.8041** **(0.0099)**	0.8116 (0.0077)	0.8244 (0.0082)	0.8318 (0.0056)	**0.8569** **(0.0066)**	**0.8688** **(0.0112)**
		5 years	0.6977 (0.0088)	0.7061 (0.0091)	0.6981 (0.0041)	0.7034 (0.0100)	0.6929 (0.0106)	0.6891 (0.0136)	0.7045 (0.0191)	0.7246 (0.0073)	**0.7473** **(0.0093)**	0.7546 (0.0018)	0.7714 (0.0049)	0.7792 (0.0065)	**0.8004** **(0.0021)**	0.8099 (0.0092)
	Cause 2	**1 year**	**0.7128** **(0.0381)**	0.6985 (0.0119)	**0.7158** (0.0354)	0.6971 (0.0289)	0.6856 (0.0360)	**0.6815** (0.0222)	**0.7096** **(0.0162)**	**0.7290** **(0.0257)**	**0.7262** **(0.0310)**	**0.7365** **(0.0134)**	**0.7360** **(0.0313)**	0.7357 (0.0223)	**0.7807** **(0.0114)**	0.7844 (0.0152)
		5 years	0.6097 (0.0172)	0.5981 (0.0338)	0.6150 (0.0158)	0.6138 (0.0097)	**0.6029** **(0.0126)**	0.5918 (0.0344)	**0.6130** **(0.0269)**	0.6360 (0.0386)	**0.6727** **(0.0132)**	0.6790 (0.0180)	0.6857 (0.0205)	**0.6998** **(0.0189)**	**0.7364** **(0.0239)**	**0.7518** **(0.0097)**
CS-Marginal DeepPseudo	**Cause 1**	**1 year**	**0.7753** **(0.0170)**	0.7730 (0.0107)	0.7741 (0.0127)	0.7739 (0.0111)	0.7681 (0.0193)	0.7617 (0.0244)	0.7752 (0.0059)	**0.7925** **(0.0077)**	0.8036 (0.0058)	**0.8144** **(0.0088)**	**0.8245** **(0.0094)**	**0.8319** **(0.0055)**	0.8551 (0.0051)	0.8656 (0.0113)
		5 years	0.7035 (0.0099)	0.7055 (0.0037)	0.6908 (0.0092)	0.6906 (0.0124)	0.6762 (0.0183)	0.6820 (0.0161)	**0.7153** **(0.0039)**	**0.7363** **(0.0101)**	0.7462 (0.0081)	0.7551 (0.0083)	**0.7726** **(0.0048)**	**0.7811** **(0.0042)**	0.7999 (0.0016)	0.8099 (0.0096)
	Cause 2	**1 year**	0.6823 (0.0221)	0.7015 (0.0328)	0.6646 (0.0386)	0.6852 (0.0575)	0.6616 (0.0219)	0.6056 (0.0459)	0.7063 (0.0208)	0.6860 (0.0355)	0.7019 (0.0527)	0.7165 (0.0290)	0.7098 (0.0549)	0.7481 (0.0174)	0.7405 (0.0515)	0.7851 (0.0170)
		5 years	0.6097 (0.0176)	0.5970 (0.0183)	0.6006 (0.0237)	**0.6180** **(0.0106)**	0.5948 (0.0162)	0.5971 (0.0346)	0.6094 (0.0292)	**0.6593** **(0.0135)**	0.6708 (0.0142)	**0.6802** **(0.0173)**	0.6748 (0.0380)	0.6994 (0.0169)	0.7195 (0.0215)	0.7440 (0.0121)
Conditional DeepPseudo	**Cause 1**	**1 year**	0.7693 (0.0175)	0.7632 (0.0170)	0.7593 (0.0128)	0.7545 (0.0074)	0.7534 (0.0135)	0.7350 (0.0277)	0.7654 (0.0078)	0.7798 (0.0064)	0.7887 (0.0070)	0.8036 (0.0068)	0.8130 (0.0111)	0.8236 (0.0036)	0.8485 (0.0077)	0.8613 (0.0092)
		5 years	**0.7140** **(0.0073)**	0.7101 (0.0068)	**0.7075** **(0.0092)**	0.7004 (0.0090)	0.6932 (0.0172)	0.6815 (0.0128)	0.7115 (0.0039)	0.7349 (0.0090)	0.7461 (0.0091)	**0.7589** **(0.0046)**	0.7714 (0.0065)	0.7794 (0.0050)	0.7992 (0.0021)	0.8078 (0.0088)
	Cause 2	**1 year**	0.6838 (0.0300)	0.6832 (0.0092)	0.7063 (0.0149)	0.6748 (0.0563)	0.6718 (0.0257)	0.6397 (0.0596)	0.6986 (0.0097)	0.7080 (0.0204)	0.7218 (0.0219)	0.7088 (0.0208)	0.7178 (0.0493)	0.7286 (0.0166)	0.7501 (0.0119)	**0.7936** **(0.0134)**
		5 years	0.5950 (0.0352)	0.6000 (0.0166)	**0.6187** **(0.0274)**	0.6047 (0.0147)	0.5795 (0.0322)	0.5722 (0.0257)	0.6117 (0.0172)	0.6467 (0.0097)	0.6573 (0.0241)	0.6543 (0.0059)	**0.6862** **(0.0249)**	0.6874 (0.0202)	0.7159 (0.0219)	0.7402 (0.0102)
CS-Conditional DeepPseudo	**Cause 1**	**1 year**	0.7603 (0.0126)	0.7589 (0.0211)	0.7559 (0.0108)	0.7573 (0.0127)	0.7413 (0.0140)	0.7393 (0.0326)	0.7661 (0.0087)	0.7839 (0.0125)	0.7978 (0.0077)	0.8052 (0.0081)	0.8158 (0.0088)	0.8258 (0.0060)	0.8475 (0.0084)	0.8622 (0.0128)
		5 years	0.7069 (0.0086)	0.7010 (0.0055)	0.6942 (0.0116)	0.6825 (0.0082)	0.6853 (0.0150)	0.6802 (0.0151)	0.7069 (0.0059)	0.7341 (0.0095)	0.7420 (0.0096)	0.7572 (0.0048)	0.7676 (0.0051)	0.7807 (0.0067)	0.7971 (0.0018)	**0.8102** **(0.0104)**
	Cause 2	**1 year**	0.6909 (0.0044)	0.6915 (0.0377)	0.6760 (0.0317)	0.6885 (0.0442)	0.6700 (0.0266)	0.6451 (0.0573)	0.6835 (0.0264)	0.7231 (0.0197)	0.7057 (0.0135)	0.7354 (0.0217)	0.7186 (0.0446)	**0.7539** **(0.0236)**	0.7603 (0.0125)	0.7789 (0.0144)
		5 years	**0.6138** **(0.0149)**	**0.6128** **(0.0208)**	0.6020 (0.0235)	0.6176 (0.0177)	0.5861 (0.0089)	**0.6025** **(0.0522)**	0.6020 (0.0207)	0.6527 (0.0245)	0.6615 (0.0207)	0.6770 (0.0180)	0.6766 (0.0366)	0.6943 (0.0225)	0.7174 (0.0245)	0.7416 (0.0101)
DeepHit	**Cause 1**	**1 year**	0.7743 (0.0193)	0.7680 (0.0115)	0.7731 (0.0129)	**0.7751** **(0.0147)**	0.7707 (0.0178)	**0.7692** **(0.0171)**	**0.7790** **(0.0077)**	0.7915 (0.0061)	0.8032 (0.0070)	0.8099 (0.0058)	0.8211 (0.0101)	0.8306 (0.0080)	0.8400 (0.0105)	0.8649 (0.0114)
		5 years	0.7050 (0.0046)	0.7010 (0.0133)	0.7009 (0.0068)	0.6905 (0.0122)	0.6958 (0.0109)	0.6902 (0.0126)	0.6997 (0.0076)	0.7256 (0.0061)	0.7435 (0.0062)	0.7532 (0.0023)	0.7675 (0.0037)	0.7784 (0.0067)	0.7983 (0.0025)	0.8092 (0.0093)
	Cause 2	**1 year**	0.7001 (0.0331)	**0.7021** **(0.0127)**	0.7091 (0.0277)	**0.7101** **(0.0411)**	0.6843 (0.0353)	0.6221 (0.0802)	0.6863 (0.0254)	0.6903 (0.0355)	0.7120 (0.0144)	0.7268 (0.0183)	0.7240 (0.0527)	0.7343 (0.0254)	0.7681 (0.0186)	0.7876 (0.0145)
		5 years	0.5611 (0.0296)	0.5622 (0.0253)	0.5522 (0.0190)	0.5599 (0.0120)	0.5234 (0.0443)	0.5446 (0.0421)	0.5534 (0.0306)	0.5759 (0.0659)	0.6347 (0.0279)	0.6712 (0.0176)	0.6588 (0.0170)	0.6798 (0.0330)	0.7222 (0.0253)	0.7441 (0.0183)
Fine & Gray	**Cause 1**	**1 year**	0.7527 (0.0200	0.7567 (0.0134)	0.7575 (0.0119)	0.7602 (0.0098)	0.7590 (0.0169)	0.7568 (0.0266)	0.7537 (0.0147)	0.7727 (0.0034)	0.7842 (0.0073)	0.7953 (0.0059)	0.8058 (0.0128)	0.8161 (0.0058)	0.8447 (0.0075)	0.8596 (0.0134)
		5 years	0.7119 (0.0067)	**0.7114** **(0.0074)**	0.7073 (0.0096)	**0.7040** **(0.0079)**	**0.6977** **(0.0182)**	**0.6910** **(0.0103)**	0.7126 (0.0080)	0.7321 (0.0099)	0.7448 (0.0074)	0.7556 (0.0030)	0.7686 (0.0076)	0.7755 (0.0071)	0.7966 (0.0015)	0.8065 (0.0110)
	Cause 2	**1 year**	0.4914 (0.0193)	0.4962 (0.0185)	0.5026 (0.0347)	0.5119 (0.0490)	0.5181 (0.0442)	0.5634 (0.0871)	0.4867 (0.0162)	0.4828 (0.0140)	0.4666 (0.0388)	0.4661 (0.0501)	0.4601 (0.0586)	0.4616 (0.0356)	0.5347 (0.0467)	0.5920 (0.0496)
		5 years	0.5123 (0.0209)	0.5191 (0.0352)	0.5280 (0.0220)	0.5409 (0.0182)	0.5483 (0.0230)	0.5708 (0.0481)	0.5103 (0.0144)	0.5188 (0.0269)	0.5030 (0.0224)	0.5091 (0.0260)	0.5245 (0.0218)	0.5359 (0.0128)	0.6004 (0.0229)	0.6469 (0.0253)
RSF	**Cause 1**	**1 year**	0.7601 (0.0152)	0.7610 (0.0149)	0.7602 (0.0137)	0.7623 (0.0135)	0.7639 (0.0181)	0.7607 (0.0297)	0.7610 (0.0085)	0.7783 (0.0049)	0.7915 (0.0049)	0.8026 (0.0085)	0.8119 (0.0108)	0.8208 (0.0070)	0.8463 (0.0075)	0.8590 (0.0156)
		5 years	0.6906 (0.0080)	0.6923 (0.0075)	0.6917 (0.0082)	0.6920 (0.0055)	0.6899 (0.0135)	0.6875 (0.0158)	0.6913 (0.0086)	0.7155 (0.0090)	0.7306 (0.0046)	0.7453 (0.0037)	0.7574 (0.0063)	0.7658 (0.0050)	0.7899 (0.0020)	0.8003 (0.0116)
	Cause 2	**1 year**	0.6840 (0.0106)	0.6870 (0.0173)	0.6854 (0.0470)	0.7045 (0.0322)	**0.6873** **(0.0369)**	0.6634 (0.0603)	0.6850 (0.0175)	0.6941 (0.0270)	0.7083 (0.0309)	0.7025 (0.0310)	0.7166 (0.0278)	0.7240 (0.0332)	0.7477 (0.0229)	0.7640 (0.0209)
		5 years	0.5773 (0.0252)	0.5752 (0.0232)	0.5800 (0.0300)	0.5836 (0.0165)	0.5744 (0.0090)	0.5756 (0.0494)	0.5750 (0.0220)	0.6051 (0.0246)	0.6088 (0.0077)	0.6264 (0.0247)	0.6481 (0.0156)	0.6631 (0.0114)	0.7111 (0.0197)	0.7329 (0.0261)

Table 11: Brier scores for different censoring settings. Dataset: SEER. 0k means no censored observations, 1k corresponds to 1k censored observations in the dataset, and so on.

			Induced censoring						Incremental censoring							
Algorithms	**Cause of the Event**	**Evaluation Time**	**0k**	**1k**	**2k**	**3k**	**4k**	**5k**	**0k**	**1k**	**2k**	**3k**	**4k**	**5k**	**10k**	**15k**
Marginal DeepPseudo	**Cause 1**	**1 year**	0.1888	0.1709	0.1543	0.1374	0.1193	0.0994	0.1895	0.1717	0.1476	0.1328	0.1226	0.1084	0.0716	0.0608
		5 years	0.2781	0.2749	0.2749	0.2662	0.2568	0.2492	0.2691	0.3006	0.2847	0.2822	0.2604	0.2400	0.1821	0.1505
	Cause 2	**1 year**	0.0969	0.0865	0.0756	0.0657	0.0553	0.0445	0.0973	0.0785	0.0680	0.0581	0.0511	0.0450	0.0271	0.0216
		5 years	0.5587	0.5123	0.4654	0.4078	0.3358	0.2835	0.5610	0.4182	0.3498	0.2880	0.2431	0.2070	0.1209	0.0925
CS-Marginal DeepPseudo	**Cause 1**	**1 year**	0.1835	0.1657	0.1512	0.1337	0.1187	0.0977	0.1827	0.1595	0.1473	0.1350	0.1192	0.1107	0.0723	0.0583
		5 years	0.2791	0.2812	0.2911	0.2712	0.2914	0.2557	0.2716	0.2936	0.2875	0.2982	0.2705	0.2625	0.1872	0.1594
	Cause 2	**1 year**	0.0977	0.0877	0.0768	0.0665	0.0556	0.0449	0.0982	0.0787	0.0688	0.0590	0.0516	0.0453	0.0275	0.0218
		5 years	0.5589	0.5108	0.4592	0.3983	0.3483	0.2865	0.5584	0.4145	0.3448	0.2894	0.2429	0.2060	0.1208	0.0914
Conditional DeepPseudo	**Cause 1**	**1 year**	0.2182	0.1989	0.1785	0.1571	0.1346	0.1106	0.2182	0.1795	0.1596	0.1390	0.1233	0.1094	0.0687	0.0555
		5 years	0.6553	0.6226	0.5850	0.5388	0.4856	0.4202	0.6552	0.5508	0.4932	0.4358	0.3899	0.3445	0.2272	0.1814
	Cause 2	**1 year**	0.0987	0.0880	0.0773	0.0667	0.0559	0.0451	0.0990	0.0791	0.0692	0.0593	0.0519	0.0455	0.0276	0.0218
		5 years	0.5768	0.5254	0.4692	0.4161	0.3570	0.2980	0.5766	0.4234	0.3535	0.2928	0.2489	0.2088	0.1219	0.0928
CS-Conditional DeepPseudo	**Cause 1**	**1 year**	0.2152	0.1887	0.1714	0.1452	0.1375	0.1116	0.2326	0.1785	0.1549	0.1506	0.1333	0.1311	0.0807	0.0649
		5 years	0.258	0.2626	0.2524	0.2510	0.2473	0.2306	0.2543	0.2596	0.2550	0.2560	0.2452	0.2366	0.1693	0.1403
	Cause 2	**1 year**	0.0977	0.0873	0.0761	0.0663	0.0555	0.0449	0.0973	0.0785	0.0684	0.0591	0.0517	0.0453	0.0275	0.0218
		5 years	0.5452	0.5039	0.4491	0.3998	0.3424	0.2842	0.5495	0.4122	0.3427	0.2871	0.2439	0.2054	0.1205	0.0919
DeepHit	**Cause 1**	**1 year**	0.3154	0.3505	0.1717	0.2261	0.2021	0.2535	0.3184	0.2265	0.1941	0.1385	0.2079	0.1662	0.0847	0.0795
		5 years	0.3022	0.3273	0.2844	0.3436	0.3315	0.3425	0.2945	0.2942	0.2822	0.2513	0.2810	0.2617	0.1874	0.1624
	Cause 2	**1 year**	0.0983	0.0878	0.0768	0.0664	0.0557	0.0449	0.0979	0.0784	0.0688	0.0589	0.0518	0.0454	0.0275	0.0218
		5 years	0.5654	0.516	0.4494	0.4027	0.3447	0.2867	0.5585	0.4073	0.3460	0.2861	0.2462	0.2059	0.1204	0.0922
Fine & Gray	**Cause 1**	**1 year**	0.2607	0.2359	0.2078	0.1806	0.1554	0.1230	0.2633	0.2112	0.1832	0.1598	0.1407	0.1230	0.0750	0.0582
		5 years	0.2698	0.2773	0.2784	0.2789	0.2821	0.2646	0.2715	0.2882	0.2819	0.2715	0.2566	0.2419	0.1796	0.1456
	Cause 2	**1 year**	0.0969	0.0867	0.0762	0.0659	0.0554	0.0448	0.0969	0.0782	0.0686	0.0589	0.0516	0.0453	0.0275	0.0218
		5 years	0.5383	0.4935	0.4432	0.3951	0.3421	0.2874	0.5388	0.4040	0.3418	0.2851	0.2434	0.2051	0.1207	0.0922
RSF	**Cause 1**	**1 year**	0.1750	0.1620	0.1482	0.1327	0.1161	0.0976	0.1750	0.1515	0.1376	0.1230	0.1114	0.1011	0.0680	0.0546
		5 years	0.2169	0.2170	0.2215	0.2235	0.2237	0.2173	0.2116	0.2268	0.2257	0.2205	0.2129	0.2053	0.1644	0.1377
	Cause 2	**1 year**	0.0924	0.0830	0.0733	0.0638	0.0539	0.0438	0.0924	0.0748	0.0658	0.0567	0.0498	0.0438	0.0268	0.0213
		5 years	0.4370	0.4056	0.3689	0.3343	0.2938	0.2523	0.4371	0.3358	0.2886	0.2436	0.2104	0.1788	0.1086	0.0841

Table 12: Performance comparison for incremental censoring settings (common uncensored observations). Dataset: SEER. 1k corresponds to 1k censored observations in the dataset, and so on.

			C-index							Brier Score						
Algorithms	**Cause of the Event**	**Evaluation Time**	**1k**	**2k**	**3k**	**4k**	**5k**	**10k**	**15k**	**1k**	**2k**	**3k**	**4k**	**5k**	**10k**	**15k**
Marginal DeepPseudo	**Cause 1**	**1 year**	0.7739	0.7758	0.7723	0.7741	0.7754	0.7747	0.7700	0.1936	0.1863	0.1879	0.1924	0.1890	0.1910	0.2005
		5 years	0.7022	0.7125	0.7056	0.7086	0.7115	0.7108	0.7043	0.2491	0.2367	0.2324	0.2334	0.2294	0.2652	0.3065
	Cause 2	**1 year**	0.7105	0.6997	0.7056	0.6757	0.6748	0.6925	0.6752	0.0954	0.0938	0.0923	0.0918	0.0913	0.0878	0.0888
		5 years	0.6163	0.6426	0.6426	0.6279	0.6291	0.6342	0.6359	0.4975	0.4742	0.4443	0.4217	0.4056	0.3637	0.3619
CS-Marginal DeepPseudo	**Cause 1**	**1 year**	0.7757	0.7760	0.7753	0.7746	0.7756	0.7731	0.7670	0.1820	0.1863	0.1873	0.1845	0.1870	0.1886	0.1931
		5 years	0.7147	0.7120	0.7062	0.7097	0.7132	0.7107	0.7034	0.2415	0.2354	0.2403	0.2289	0.2315	0.2577	0.2913
	Cause 2	**1 year**	0.6724	0.6778	0.6820	0.6637	0.6926	0.6523	0.6745	0.0956	0.0951	0.0939	0.0930	0.0920	0.0896	0.0898
		5 years	0.6303	0.6369	0.6321	0.6079	0.6365	0.6245	0.6316	0.4930	0.4673	0.4466	0.4213	0.4036	0.3633	0.3572
DeepHit	**Cause 1**	**1 year**	0.7720	0.7742	0.7689	0.7690	0.7721	0.7641	0.7630	0.2493	0.2307	0.1959	0.2728	0.2448	0.2007	0.2223
		5 years	0.7017	0.7077	0.7019	0.7016	0.7078	0.7060	0.7007	0.2510	0.2483	0.2487	0.2603	0.2578	0.2762	0.3161
	Cause 2	**1 year**	0.6682	0.6822	0.6821	0.6695	0.6650	0.6661	0.6598	0.0955	0.0951	0.0938	0.0932	0.0922	0.0895	0.0900
		5 years	0.5454	0.6070	0.6175	0.6025	0.5979	0.6186	0.6124	0.4862	0.4689	0.4414	0.4273	0.4035	0.3621	0.3606

Table 13: Performance Comparison of DeepPseudo models with DeepHit model with and without ranking loss

Performance Metric	**Dataset**	**Cause of the Event**	**Evaluation Time**	**DeepHit w/o ranking loss**	**DeepHit with ranking loss**	**Marginal DeepPseudo**	**CS-Marginal DeepPseudo**	**Conditional DeepPseudo**	**CS_Conditional DeepPseudo**
C-index	**SEER**	**cause 1**	**1 year**	0.8778	0.8794	0.8740	0.8777	0.8659	0.8651
			5 years	0.8095	0.8109	0.8104	0.8116	0.8051	0.8026
		Cause 2	**1 year**	0.8525	0.8466	0.8530	0.8422	0.8372	0.8454
			5 years	0.8141	0.8053	0.8108	0.8059	0.8001	0.8118
	Synthetic	**cause 1**	**1 year**	0.7558	0.7562	0.7511	0.7517	0.7606	0.7523
			5 years	0.6971	0.6922	0.6715	0.6689	0.7033	0.6907
		Cause 2	**1 year**	0.7545	0.7524	0.7530	0.7551	0.7612	0.7574
			5 years	0.6946	0.6876	0.6714	0.6635	0.6994	0.6892
Brier Score	**SEER**	**cause 1**	**1 year**	0.0496	0.0609	0.0522	0.0522	0.0508	0.0537
			5 years	0.1211	0.1314	0.1297	0.1256	0.1351	0.1199
		Cause 2	**1 year**	0.0173	0.0173	0.0173	0.0173	0.0173	0.0172
			5 years	0.0661	0.0655	0.0660	0.0658	0.0765	0.0677
	Synthetic	**cause 1**	**1 year**	0.2209	0.2188	0.2257	0.2208	0.2252	0.2222
			5 years	0.3033	0.3088	0.3041	0.2921	0.588	0.3136
		Cause 2	**1 year**	0.2170	0.2162	0.2205	0.2165	0.2201	0.2178
			5 years	0.3074	0.3144	0.3104	0.2957	0.5875	0.3203