Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Extending Deep Cox Survival Models with Case-Cohort risk set Sampling

Justine Nasejje

justinenasejje@gmail.com

University of the Witwatersrand https://orcid.org/0000-0001-8785-8808

Michael P. Falk

University of the Witwatersrand

Marvin N. Wright

University of Bremen https://orcid.org/0000-0002-8542-6291

Wende Clarence Safari

London School of Hygiene & Tropical Medicine https://orcid.org/0000-0003-4639-7552

Pierre-Philippe Dechant

University of Leeds https://orcid.org/0000-0002-4694-4010

Raeesa Manjoo Docrat

University of the Witwatersrand https://orcid.org/0000-0003-2039-0440

Rukia Nuermaimaiti

Imperial College London https://orcid.org/0000-0002-5764-9949

Research Article

Keywords: Survival analysis, deep neural networks, risk set sampling, right censored data, Cox proportional hazards, nested case-control, case-cohort, computational efficiency

Posted Date: September 17th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11057132/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Extending Deep Cox Survival Models with Case-Cohort risk set Sampling

Michael P. Falk[1†], Justine B. Nasejje[2,3*†], Marvin N. Wright[3,4], Wende Clarence Safari[5], Pierre-Philippe Dechant[6], Raeesa Manjoo Docrat[2], Rukia Nuermaimaiti[7]

[1*]School of Computer Science & Data Science, University of the Witwatersrand, 1 Jan Smuts Avenue, Johannesburg, 2000, Gauteng, South Africa.
[2*]School of Statistics and Actuarial Science, University of the Witwatersrand, 1 Jan Smuts Avenue, Johannesburg, 2000, Gauteng, South Africa.
[3]Leibniz Institute for Prevention Research and Epidemiology (BIPS), Achterstraße 30, Bremen, 28359, Germany.
[4]Faculty of Mathematics and Computer Science, University of Bremen, Bibliothekstraße 1, Bremen, 28359, Germany.
[5]Department of Health Research Services and Policy (HSRP), London School of Hygiene & Tropical Medicine, Keppel Street, London, WC1E 7HT, United Kingdom.
[6]School of Mathematics, University of Leeds, Woodhouse Lane, Leeds, LS2 9JT, United Kingdom.
[7]Department of Mathematics, Imperial College London, South Kensington Campus, London, SW7 2AZ, United Kingdom.

*Corresponding author(s). E-mail(s): justinenasejje@gmail.com;
Contributing authors: 3052851@students.wits.ac.za; wright@leibniz-bips.de; Wende.Safari@lshtm.ac.uk; matpd@leeds.ac.uk; raeesa.docrat@wits.ac.za; r.nuermaimaiti@imperial.ac.uk;
†These authors contributed equally to this work.

Abstract

Deep neural networks (DNNs) have demonstrated strong predictive performance in survival analysis when trained using the Cox partial likelihood. However,

evaluating the full risk set at each event time incurs substantial computational and memory costs, limiting scalability to large datasets. This paper evaluates an efficient training framework for Cox-based DNN survival models incorporating risk-set sampling, specifically comparing nested case-control (NCC) sampling with case-cohort (CC) sampling across full-batch and mini-batch regimes. To the best of our knowledge, this study presents the first integration and evaluation of case-cohort risk-set sampling within deep Cox proportional hazards neural networks. The proposed framework is evaluated using synthetic data and the real-world METABRIC dataset in terms of discrimination, prediction error, recovery of the known latent risk structure in simulation, and computational resource utilization. NCC consistently approximates the full risk-set baseline, with synthetic-data C-index values of approximately **0.78**, log-hazard correlations approaching **0.99**, and C-index differences on the order of $\mathbf{10^{-3}}$. Conversely, CC is more sensitive to subcohort size ($\boldsymbol{k}$), with severe down-sampling ($\boldsymbol{k = 5}$) producing a C-index reduction of approximately **23%** relative to the full-risk baseline under full-batch training. Mini-batch execution substantially reduces this degradation, allowing CC sampling with smaller absolute subcohorts to recover near-baseline C-index and Integrated Brier Score (IBS) while reducing computational resources. These findings show that risk-set sampling can provide substantial computational savings in deep Cox models, with NCC offering robust performance across sampling configurations and mini-batch CC providing an effective alternative when subcohort size is appropriately chosen, as well as a natural analytical strategy when the underlying data arise from a case-cohort study design.

Keywords: Survival analysis, deep neural networks, risk set sampling, right censored data, Cox proportional hazards, nested case-control, case-cohort, computational efficiency.

Introduction

Survival analysis is a statistical field that focuses on the time until a set events or event of interest occurs, such as failure of a machine or a product, as well as death in the medical field. Traditional methods, notably the Cox regression model [1], have long provided robust frameworks for handling right censored data where events of interest have not occurred for some subjects by a set period of a study [2]. However, as data complexity increases, there is a growing need to integrate modern machine learning techniques into survival analysis. In this study, we propose an approach that applies deep neural networks (DNNs) to enhance traditional survival analysis through risk set sampling on right censored data. Deep neural networks have gained prominence due to their ability to model complex, non-linear relationships in large datasets [3]. In survival analysis, recent methods such as those implemented in models like DeepSurv demonstrate that DNNs can capture subtle interactions that classical models overlook [4]. Our approach uses the Cox regression framework as a basis, extending its capabilities

by leveraging the flexibility of DNNs to improve predictive performance by incorporating risk set sampling, an important methodological gap especially in the era of big data, as this aspect remains relatively underexplored in the existing literature [5].
This research therefore further bridges the gap between classical survival analysis and modern deep learning approaches by leveraging risk set sampling. Sampling techniques such as nested case-control, first introduced by [5] in the context of neural networks for survival data, and the one adapted for this study, case-cohort designs, not only reduce the amount of data processed per training epoch thereby improving computational efficiency but also allow for unbiased estimation of the hazard, in accordance with the fundamental principles of time-to-event analysis [1, 3–8]. This work therefore makes the following contributions; First, we propose a novel integration of the classical case-cohort (CC) sampling design into deep neural network implementations of the Cox proportional hazards model, addressing a gap in existing deep survival learning frameworks. Secondly, we develop a training formulation that incorporates case-cohort sampling within both full-batch and mini-batch stochastic optimization, enabling scalable learning on large survival datasets. And thirdly, we provide the first systematic empirical comparison of full risk set, nested case-control (NCC), and case-cohort sampling strategies within deep Cox models, demonstrating that the proposed approach achieves comparable discrimination performance while significantly reducing computational cost. To the best of our knowledge, prior deep learning survival models have not incorporated case-cohort sampling into the Cox partial likelihood framework, nor evaluated its computational-statistical trade-offs against nested case-control risk set sampling.

Methods

Cox Proportional Hazards Model

The Cox proportional hazards model [1] expresses the instantaneous hazard for subject i at time t as

$$h(t \mid X_i) = h_0(t)\, \exp\big(\beta^\top X_i\big) \tag{1}$$

where $h_0(t)$ is an unspecified baseline hazard, X_i the feature or covariate vector and β are regression coefficients. Rather than specifying $h_0(t)$, estimation proceeds via the partial likelihood.

$$L(\beta) = \prod_{j:\,\text{event at } t_j} \frac{\exp\big(\beta^\top X_j\big)}{\sum_{i \in \mathcal{R}(t_j)} \exp\big(\beta^\top X_i\big)}. \tag{2}$$

Maximizing $\log L(\beta)$ yields consistent estimates of β even under right censoring [1]. This semi-parametric formulation forms the statistical backbone of our neural-Cox hybrid.
To train our feed-forward network end-to-end, we replace $\beta^\top X_i$ with the network output $\eta_i = f_\theta(X_i)$ and minimize the negative log–partial likelihood (log-loss):

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{j:\,\text{event at } t_j} \Big(\eta_j - \log \sum_{i \in \mathcal{R}_{\text{sample}}(t_j)} \exp(\eta_i)\Big). \tag{3}$$

Here $\mathcal{R}_{\text{sample}}(t_j)$ is the sampled risk set depending on the sampling scheme used or the full risk set ($\mathcal{R}(t_j)$) at t_j. By minimizing $\mathcal{L}(\theta)$, the network learns the representations that directly optimize time to event discrimination.

Risk Set Sampling Techniques

Computing the full partial likelihood in a Cox model involves summing over the entire risk set $\mathcal{R}(t_j)$ at each observed event time t_j, which becomes expensive when the sample size n is large. Risk set sampling offers an efficient approximation by selecting, for each event, a fixed number m of controls from those still at risk at time t_j rather than summing over every individual. This approach not only substantially reduces computational overhead but will also reduce training time of the models [7].
One of the risk-set sampling designs that has been adopted for defining loss functions in survival neural networks is the nested case–control [5]. In the nested case–control design, for each observed event at time t_j, one samples m controls at random from the corresponding risk set. A partial likelihood is then constructed using the case and the sampled controls [1, 9]. This approach has been shown to give an unbiased survival predictions, making it particularly suitable for large-scale survival analysis problems.
An alternative sampling design is the case–cohort design, in which a subcohort of size m is sampled once at baseline, forming a static subcohort. At each event time t_j, the risk set consists of all observed cases together with those members of the subcohort who remain event-free at t_j. This structure is incorporated into the partial likelihood of training the loss function[6].
In this study, we adapt the case–cohort design to construct a novel loss function for survival neural networks. We further compare its predictive performance with that of a loss function based on the nested case–control design[5].

Deep neural networks

A deep neural network (DNN) is a model that learns to approximate complex patterns by mapping inputs to outputs. These networks consist of multiple interconnected layers, each passing signals to neurons that, in turn, produce the desired output. Thus DNNs can learn highly complex, hierarchical representations of input data, making them particularly powerful for tasks where the relationship between the inputs and the outputs is a complex one. They have been adopted to many studies including survival analysis [4]. This study makes use of the Cox proportional hazards partial-likelihood loss shown in equation 4 [1, 4]. Let D denote the set of observed event indices, $\mathcal{R}(t_j)$ the risk set at time t_j, and $r_j = \eta_j$ the risk score produced by the model for subject j. The negative log-partial likelihood to be minimized is given by

$$\mathcal{L}_{\text{Cox}}(\theta) \;=\; -\sum_{j \in D} \left[r_j \;-\; \log\left(\sum_{i \in \mathcal{R}(t_j)} e^{\eta_i} \right) \right]. \tag{4}$$

In this study, we propose a deep neural network (DNN) framework for survival analysis with right censored data that incorporates the Case-Cohort risk set sampling design.

We further evaluate the performance of the Nested Case-Control and Case-Cohort sampling schemes in approximating the full risk sets. The network is trained by embedding the Cox proportional hazards partial likelihood loss, as defined in equation 4, while in-cooperating the sampling designs.

Data Simulation

To rigorously evaluate the performance of our deep survival networks under controlled risk environments, we employ a comprehensive synthetic data generation framework. This approach provides complete governance over the underlying covariate distributions, the exact functional structure of the latent hazard rates, and the severity of the right-censoring mechanism.
Each simulated subject i (for $i = 1, \ldots, N$) is characterized by a feature vector $\mathbf{x}_i$ comprised of eight distinct covariates spanning binary, continuous uniform, and continuous normal distributions:

$$X_{1i}, X_{2i}, X_{3i} \sim \text{Bernoulli}(0.5), \tag{5}$$
$$X_{4i}, X_{5i} \sim \mathcal{U}(0, 1), \tag{6}$$
$$X_{6i}, X_{7i}, X_{8i} \sim \mathcal{N}(0, 1). \tag{7}$$

These covariates are linearly combined to construct the true underlying latent log-risk profile, or linear predictor (η_i), expressed as:

$$\eta_i = \sum_{j=1}^{8} \beta_j X_{ji} = \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_8 X_{8i} \tag{8}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_8)^T$ represents the vector of true target regression coefficients.
True individual event times (T_i^{event}) are generated conditional on these covariates using an exponential regression formulation. The individual specific hazard is scaled multiplicatively by the log-risk factor $\exp(\eta_i)$ relative to a baseline hazard rate λ:

$$T_i^{\text{event}} \sim \text{Exponential}(\lambda \exp(\eta_i)) \tag{9}$$

The conditional survival probability function representing the true probability of surviving past time t given an individual's risk score is thus formulated as:

$$\mathbb{P}\big(T_i^{\text{event}} > t \mid \eta_i\big) = \exp(-\lambda \exp(\eta_i)\, t)\,, \qquad t \geq 0 \tag{10}$$

To introduce non-informative right-censoring, independent censoring times (T_i^{censor}) are drawn from a separate exponential distribution governed by a global censoring rate parameter γ:

$$T_i^{\text{censor}} \sim \text{Exponential}(\gamma) \tag{11}$$

For each subject, the observable follow-up time (t_i) recorded in the final dataset is restricted to the minimum of the true event time and the random censoring boundary:

$$t_i = \min\left(T_i^{\text{event}}, T_i^{\text{censor}}\right) \tag{12}$$

The corresponding observed event indicator ($\delta_i \in \{0, 1\}$) is defined via the indicator function:

$$\delta_i = \mathbf{1}\left\{T_i^{\text{event}} \leq T_i^{\text{censor}}\right\} \tag{13}$$

where $\delta_i = 1$ signifies an observed failure event, and $\delta_i = 0$ indicates that the observation was right-censored at time t_i.

In our baseline experimental configuration, we generate a stable sample population of $N = 6000$ individuals with a fixed baseline hazard scale of $\lambda = 0.1$. The true covariate effect vectors are parameterized as $\boldsymbol{\beta} = [1.2, -0.9, 0.0, 1.5, -0.6, 0.3, -1.2, 0.0]^T$, establishing a mathematically challenging risk structure where specific features (such as X_3 and X_8) act as non-informative controls. The censoring intensity is configured to $\gamma = 0.05$ to ensure a realistic, non-trivial right-censoring distribution across the study cohort, with algorithmic reproducibility preserved by initializing a static pseudo-random state (seed = 42).

Network architecture

The neural network takes as input the covariate vector $X \in \mathbb{R}^p$, where p is the number of predictors. The architecture consists of three hidden layers with dimensions 64, 64, and 32 units, each followed by a rectified linear unit (ReLU) activation. To improve generalization and stabilize training, we apply dropout regularization with rates $d = 0.3$ (first two layers) and $d = 0.2$ (third layer). The final layer outputs a single log–risk score,

$$\eta_i = f_{\boldsymbol{\theta}}(X_i), \tag{14}$$

which is used in the partial likelihood.

All linear layers are initialized using Kaiming normal initialization for weights and a small positive constant (0.01) for biases, following best practices for ReLU networks. The model is implemented in the PyTorch framework [5, 8]. This design balances model capacity and regularization, ensuring that the network can learn flexible non–linear effects of covariates on survival while mitigating overfitting through dropout and normalization.

Training

Model training is conducted using both mini-batch and full-batch strategies to enable a holistic comparison of performance. In the mini-batch setting, data loaders were employed to iteratively load subsets of the dataset into the feed-forward network. This approach directly influences the risk set sampling procedure, as the sampled risk sets are restricted to the mini-batch rather than the full dataset at each iteration. Conversely, full-batch training utilizes the entire dataset in each update, ensuring that the risk sets are drawn from the complete cohort.

To assess generalization performance, we adopted K-fold cross-validation. The dataset was partitioned into K folds, with each fold serving once as a validation set while the remaining $K-1$ folds were used for training. out of fold predictions were concatenated across all folds to compute the pooled concordance index, following the methodology by [5]. This procedure ensured that each subject's evaluation was based on predictions from a model that had not been trained on that subject, thereby mitigating optimistic bias. In the full batch setting, the sampled risk set coincides with the classical definition:

$$\mathcal{R}_{\text{full}}(t_j) = \{i : T_i \geq t_j\}, \tag{15}$$

that is, all individuals still at risk when subject j fails. This corresponds exactly to the Cox partial likelihood introduced by [1], and is implemented in this study by iterating over all event times. The final loss is normalized by the number of observed events (see [10]).

When training with stochastic gradient descent, only a subset of individuals is available in each batch. To remain consistent with the original loss, we redefine the sampled risk set as:

$$\mathcal{R}_{\text{mini}}(t_j) = \mathcal{R}(t_j) \cap \text{Batch}, \tag{16}$$

so that risk sets are restricted to the individuals present in the mini batch. In the full-batch setting, the sampled risk set coincides with the classical definition:

$$\mathcal{R}_{\text{full}}(t_j) = \{i : T_i \geq t_j\}. \tag{17}$$

In the nested case-control setting, the batch consists of the individual with event at time t_j together with a sampled subset of controls from the risk set:

$$\mathcal{R}_{\text{NCC}}(t_j) = \{i_j\} \cup \{\mathcal{S}_j\}, \tag{18}$$

where i_j denotes the individual experiencing the event at time t_j, and $\mathcal{S}_j \subset \mathcal{R}(t_j)$ is a sampled set of controls. In the case-cohort setting, the batch is formed by a fixed subcohort $\mathcal{C}$ sampled at baseline together with the cases:

$$\mathcal{R}_{\text{CC}}(t_j) = \{i_j\} \cup (\mathcal{C} \cap \mathcal{R}(t_j)) \ , \tag{19}$$

where $\mathcal{C}$ is the fixed subcohort and i_j is the individual having an event at time t_j. While Full and Nested Case-Control sampling strategies are well established and implemented in [4, 5], we introduce an implementation of the Case-Cohort design adapted to the mini-batch setting, in which risk sets are restricted to individuals present within each batch. As compared to the Nested-Case-Control sampler that draws controls uniformly from the subjects at risk in the current mini-batch. Where as Case-Cohort; intersects the fixed subcohort with the current mini–batch and adds the event case. This ensures that the stochastic risk sets provides strong numerical approximations of their full-batch. The mini batch implementation is provided in Repo [10].As shown , our framework extends the classical Cox partial likelihood to modern large–scale learning. In the full batch setting, it recovers the exact formulation of [1]. In the mini batch setting, it integrates sampling designs from [6] (Nested Case Control, Case Cohort)

to approximate the risk set efficiently, enabling scalable training while preserving predictive performance and empirical stability across the evaluated settings.
All models were trained using the Adam optimizer with learning rate $\alpha = 10^{-3}$, and L_2 weight-decay regularization with coefficient $\lambda = 10^{-4}$ applied to all layers. The L_2 penalty helps prevent overfitting by discouraging disproportionately large weights. Early stopping was employed with a patience of 40 epochs, based on the validation concordance index (C–index), in order to maximize generalization performance.
Two training regimes were implemented Full batch and mini batch. Full-batch training uses the entire dataset to compute a single gradient update per epoch. This was used together with the full risk set as a baseline for nested case–control, and case–cohort samplers, directly corresponding to the classical Cox model training setup. Compared to Mini batch training the training set is randomly partitioned into batches of size B, and each batch is used to update the model parameters. Within each mini batch, the risk sets are restricted to the individuals present in that batch. This setting allows stochastic optimization while maintaining consistency with the corresponding sampling design (full, Nested Case Control, case-cohort).

Evaluation metrics

Concordance index (C-index)

The concordance index (C-index) evaluates the discriminative performance of the survival models by measuring the agreement between predicted risk scores and observed event orderings. In right censored survival data, not all pairs of subjects are comparable. A pair of distinct subjects (i, j) is considered admissible if the subject with the shorter observed survival time experienced the event.
Let T_i and T_j denote the observed survival times, and $\delta_i, \delta_j \in \{0, 1\}$ be the respective event indicators (where 1 indicates an observed event and 0 indicates right-censoring). Let η_i and η_j represent the predicted log-risk scores generated by the deep survival network, where a higher risk score implies shorter expected survival. Following Harrell's formulation for censored data [11], the C-index is defined as the ratio of concordant pairs to total admissible pairs:

$$\text{C-index} = \frac{\sum_{i \neq j} \mathbf{1}[T_i < T_j] \cdot \delta_i \cdot \Big(\mathbf{1}[\eta_i > \eta_j] + 0.5 \cdot \mathbf{1}[\eta_i = \eta_j] \Big)}{\sum_{i \neq j} \mathbf{1}[T_i < T_j] \cdot \delta_i}, \tag{20}$$

where $\mathbf{1}[\cdot]$ is the indicator function. The term $0.5 \cdot \mathbf{1}[\eta_i = \eta_j]$ explicitly accounts for pairs where the model predicts identical risk scores (ties), assigning them an equal chance of correctness. A C-index of 0.5 denotes performance equivalent to random guessing, whereas a value of 1.0 indicates perfect discriminative ordering. While inverse probability of censoring weighting (IPCW) is utilized to evaluate absolute risk calibration via the Integrated Brier Score (IBS), the C-index relies on Harrell's pair-filtering mechanism to handle right-censoring within the risk set.

Out of fold concordance index (OOF C-index)

To evaluate the generalization performance of the survival frameworks without introducing data leakage, we employed an out of fold (OOF) pooling strategy as outlined by Kvamme et al. [5]. This pooling procedure avoids the high variance and bias associated with averaging independent within fold C-indices, providing a robust, single-metric evaluation of generalization fidelity across the sampled risk set frameworks [5].

Integrated Brier Score (IBS)

While the C-index captures the relative ranking performance of the survival models, it does not evaluate absolute probability calibration. To measure the accuracy of the predicted survival functionalities over time, we utilized the time-dependent Brier score adjusted for right-censoring via Inverse Probability of Censoring Weighting (IPCW), as proposed by Graf et al. [12].
Let $\hat{S}(t \mid x_i)$ represent the model's predicted probability that subject i survives past time t. The Brier score at a specific time point t is defined as:

$$BS(t) = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{\hat{S}(t \mid x_i)^2 \cdot \mathbf{1}[T_i \leq t, \delta_i = 1]}{\hat{G}(T_i)} + \frac{(1 - \hat{S}(t \mid x_i))^2 \cdot \mathbf{1}[T_i > t]}{\hat{G}(t)}\right], \tag{21}$$

where $\hat{G}(t) = P(U > t)$ represents the Kaplan-Meier estimator of the censoring distribution computed from the training data, tracking the probability of remaining uncensored at time t. The first term penalizes the model for predicting high survival probabilities for subjects who have already experienced the event, whereas the second term penalizes low predicted survival for subjects known to be alive. To obtain an overall summary of calibration performance across a designated operational horizon $[t_1, t_M]$, the Brier score is integrated across evaluated time points. The Integrated Brier Score (IBS) is formulated as:

$$\text{IBS} = \frac{1}{t_M - t_1}\int_{t_1}^{t_M} BS(t)\, dt\,. \tag{22}$$

The Brier score and IBS are non-negative measures for which lower values indicate better predictive accuracy, with a value of zero corresponding to perfect prediction.

Pooled out of fold Integrated Brier Score (Pooled IBS)

To evaluate the model survival predictions robustly across the entire study sample without overfitting, we computed a pooled out of fold IBS. For each validation fold, out of fold survival curves $\hat{S}_i^{\text{OOF}}(t \mid x_i)$ were generated using baseline cumulative hazards calculated via Breslow's method on the respective training folds. The pooled IBS was subsequently calculated using the combined out of fold survival trajectories relative to the corresponding observed survival outcomes, providing a single cross-validated assessment of absolute survival probability prediction error across our sampled frameworks.

Pearson, Spearman, and Z-score Normalization

Although classical survival performance metrics such as the C-index and Brier score quantify predictive performance under right censoring, they do not directly assess recovery of the latent risk function used to generate survival times in synthetic data. To assess whether the risk set sampling strategies preserve the underlying latent risk structure, we therefore conducted additional validation on synthetic datasets for which the ground-truth linear predictor ($\mathbf{x}_i^T\boldsymbol{\beta}$ or $f(\mathbf{x}_i)$) was explicitly known [4, 5].

Let Y_i denote the true simulated linear predictor for subject i, and let $\hat{\eta}_i^{\mathrm{OOF}}$ denote the out of fold predicted risk score generated by the neural network. To assess linear agreement and monotonic rank preservation between the true and learned risk functions, we calculated Pearson and Spearman correlation coefficients using the pooled out of fold predictions.

The Pearson correlation coefficient (r) evaluates the strength and direction of the linear association between the true and predicted risk scores [13], defined as:

$$r = \frac{\sum_{i=1}^{N}(Y_i - \bar{Y})(\hat{\eta}_i^{\mathrm{OOF}} - \bar{\hat{\eta}}^{\mathrm{OOF}})}{\sqrt{\sum_{i=1}^{N}(Y_i - \bar{Y})^2 \sum_{i=1}^{N}(\hat{\eta}_i^{\mathrm{OOF}} - \bar{\hat{\eta}}^{\mathrm{OOF}})^2}}, \tag{23}$$

where $\bar{Y}$ and $\bar{\hat{\eta}}^{\mathrm{OOF}}$ represent the sample means of the true and predicted out of fold risk scores, respectively.

Concurrently, the Spearman rank correlation coefficient (ρ) evaluates the monotonic agreement without assuming a strictly linear relationship [14]. In the absence of tied ranks, it is formulated as:

$$\rho = 1 - \frac{6\sum_{i=1}^{N} d_i^2}{N(N^2-1)}, \tag{24}$$

where $d_i = \mathrm{rank}(Y_i) - \mathrm{rank}(\hat{\eta}_i^{\mathrm{OOF}})$ represents the difference between the ranks of the true linear predictor and the model prediction for subject i.

Additionally, to facilitate comparison of the distributional shapes of the true and predicted risk scores independently of differences in location and scale, both were standardized using Z-score normalization:

$$Z_{Y_i} = \frac{Y_i - \mu_Y}{\sigma_Y}, \quad Z_{\hat{\eta}_i} = \frac{\hat{\eta}_i^{\mathrm{OOF}} - \mu_{\hat{\eta}}}{\sigma_{\hat{\eta}}}. \tag{25}$$

Kernel density estimates (KDEs) of the standardized true and out-of-fold predicted risk scores were then overlaid to provide a visual assessment of agreement in distributional shape, including features such as skewness, tail behaviour, kurtosis, and modality [15]. This latent risk-function evaluation was restricted to the synthetic experiments because the ground-truth risk function is unobservable in empirical clinical datasets. These analyses therefore provide simulation-based evidence on the extent

to which mini-batch case-cohort risk set sampling preserves the latent risk structure and ranking learned from the underlying data-generating mechanism.

Memory and CPU monitoring

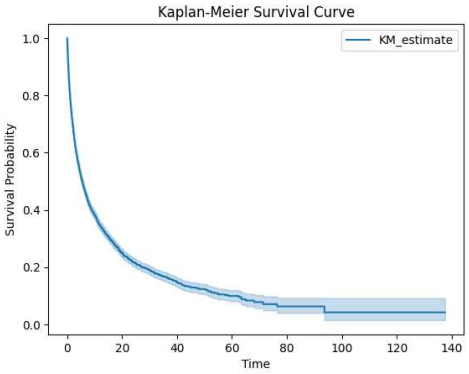
The study also monitored system resources utilization during training. This included tracking CPU usage (%) and memory consumption (MB) of the Python process over time. Monitoring resource efficiency is important in survival modeling with large datasets, particularly when comparing full–batch versus mini–batch training regimes, as the computational load differs substantially between these two approaches.
Resource monitoring was implemented using the `psutil` library, which provides fine grained access to process level statistics. A separate monitoring thread was launched in parallel to the training function, recording CPU and memory usage at fixed intervals (default: 0.5 seconds). The statistics were stored with timestamps and later converted into a `pandas` DataFrame for analysis and visualization.
The `profile_function` wrapper was used to execute a training routine while simultaneously logging system usage. At the end of execution, the function returns both the result of the training call and a DataFrame containing time series statistics of CPU and memory usage. This monitoring framework ensured that the proposed models were evaluated not only on predictive accuracy but also on their computational efficiency and scalability in real time (as the model was trained).

Results & Discussion

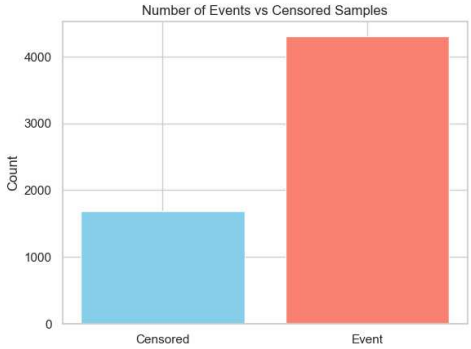
Results

Dataset characteristics

The study used both a synthetically generated dataset and a real world dataset. The real world dataset is the Molecular Taxonomy of Breast Cancer International Consortium (METABRIC), as employed in Kvamme et al. [5]. METABRIC comprises on the order of $\sim 1{,}800$ patient records with survival times, event indicators (right censoring) and a rich set of clinical and genomic covariates, whereas the simulated dataset contains 6,000 rows and therefore provides a larger, controlled sample for method validation. The Kaplan Meier curve for METABRIC (Figure 2a) shows the probability of survival shaped by heterogeneous follow-up, while the Kaplan Meier simulation data set (Figure 1a) shows a smoother, approximately exponential decline. Both curves span the full probability range from 1 to ≈ 0; however, the simulated cohort reaches low survival levels by approximately time step 140, whereas the METABRIC curve declines to comparable levels only by about time step 350. Figures 2b and 1b summarize observed versus right-censored counts for the METABRIC and simulated datasets, respectively, highlighting differences in censoring patterns that affect risk-set composition and evaluation.

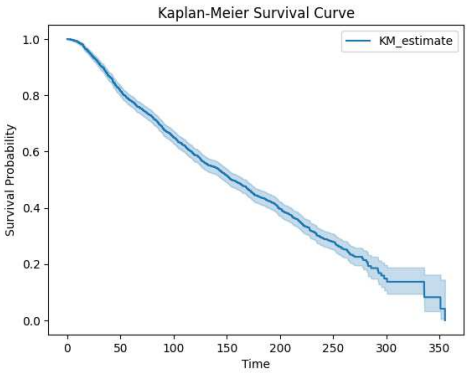


(a) Kaplan–Meier survival estimate for the simulated dataset.

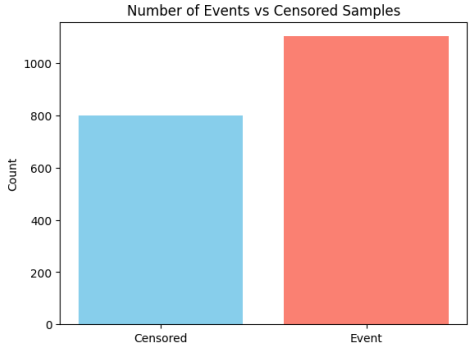


(b) Counts of observed events vs. right-censored samples for the simulated dataset.

Fig. 1: Overview of the simulated survival analysis dataset characteristics.



(a) Kaplan-Meier survival estimate for the METABRIC cohort.



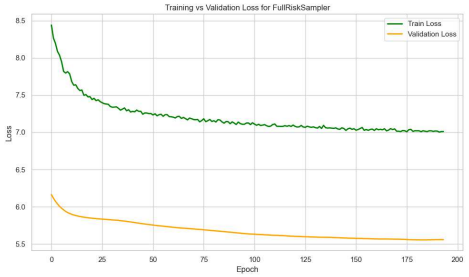
(b) Counts of observed events vs. right-censored samples for the METABRIC cohort.

Fig. 2: Overview of the METABRIC (real-world) survival analysis dataset characteristics.

Optimization and convergence (full-batch)

We trained three neural network-based Cox-type models under three risk set construction strategies: (i) Full-risk set sampling, (ii) Nested case control sampling, and (iii) Case-cohort sampling (subcohort). The Cox proportional hazards model forms the theoretical baseline for these experiments [1], and covariance/tie-handling considerations follow classical treatments [9]. Under full-batch training, all three samplers converged to stable loss values. The baseline full risk set model shows strong training convergence (training loss decreased from approximately 8.5 to 7.0). The subcohort

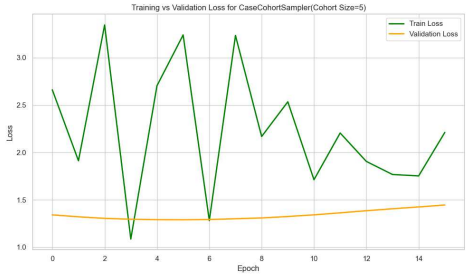
and nested case control sample techniques also converge, though the subcohort training/validation curves are noticeably more jagged and overfits during small subchorts however, when we use larger cohorts such as of 1,000 we get normal convergence as we can see in 3d. The observed noisiness in subcohort learning curves is consistent with reduced information per optimization update when only a fraction of the cohort is used in each risk set (see Figures 3a-3c), this observation is to be expected as this was predicted by case–cohort estimation theory [16].



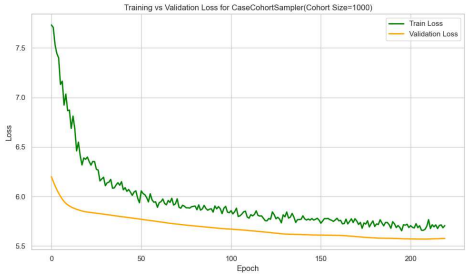
(a) Loss curve for the full-risk-set (baseline Cox) model during full-batch training.



(b) Nested case-control loss curve during full-batch training (controls = 5).



(c) Case-cohort (subcohort size = 5) loss curve during full-batch training.



(d) Case-cohort (subcohort size = 1000) loss curve during full-batch training.

Fig. 3: Comparison of loss curves across different risk set sampling strategies.

The trends observed on the simulated dataset are reinforced on the METABRIC cohort. Figure 4a and Figure 4b show that the full-risk and nested case-control strategies behave similarly to their simulated-data counterparts under full-batch training. Also as expected the METABRIC data revals the same results as the simulted data as, that models trained with **small case-cohort subcohorts** tend to overfit (see Figure 4c), whereas larger case-cohort subcohorts mitigate this overfitting (see Figure 4d). This behaviour is consistent with the increased complexity and higher covariate dimensionality of METABRIC relative to the simulated data: when the sampled subcohort is small, the model sees a limited, low-variance view of the population

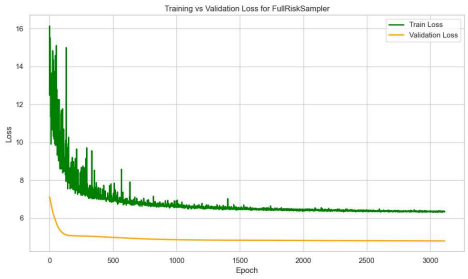
and can fit idiosyncratic noise; increasing the subcohort size increases the effective sample diversity and reduces overfitting [17, 18].
Mathematically, for a static case-cohort design (where a single subcohort is drawn once and reused across forward passes), the variance of estimators that rely on sampled risk sets decreases as the subcohort size m increases, which is consistent with classical case–cohort theory [16]. In a simplified form, this behavior follows standard sampling theory, where the variance of an estimator scales inversely with the effective sample size [17, 18]. Specifically, let $\hat{\theta}$ denote an estimator of a population parameter and let σ^2 represent the underlying population variance. The sampling contribution to the estimator variance can then be approximated as:

$$\mathrm{Var}(\hat{\theta}) \propto \frac{\sigma^2}{m}, \tag{26}$$

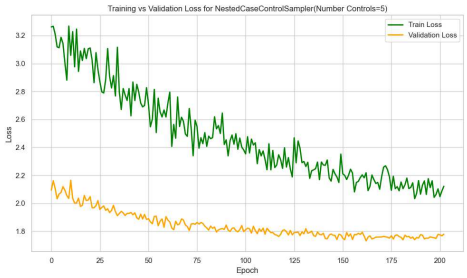
where m is the subcohort size. This relationship highlights that smaller subcohorts produce higher-variance (noisier) estimates, whereas larger subcohorts yield more stable and representative estimates of the population.
Consequently, increasing m reduces the sampling variance and stabilizes training. Because static cohorts are utilized in these experiments (the subcohort is sampled once per experiment and reused across training epochs), the effect of m is persistent throughout optimization. A small m yields consistently higher sampling variance and a greater susceptibility to overfitting, while a larger m supplies sufficient statistical representation to properly regularize the learned risk function.
When evaluating model performance across both environments, a key adjustment was made to account for the scale disparity between the synthetic dataset ($n = 6,000$) and the real-world METABRIC dataset ($n \approx 1,800$). While the simulated population allowed for larger sampling thresholds, utilizing a uniform subcohort size of $1,000$ was unfeasible for the METABRIC data, as it would encompass over half of its entire baseline population. To prevent an excessive sampling density and ensure a structurally valid comparison, an upper subcohort limit of 500 was applied exclusively to the METABRIC sampling design, as illustrated in Figure 4d.
In practice, the METABRIC empirical plots (Figures 4a–4d) demonstrate that: (i) the full-risk set and nested case-control designs (across varying control sizes) remain robust, and (ii) case-cohort performance systematically improves and overfitting diminishes as the subcohort size increases. This observation is entirely consistent with the expected $\frac{1}{m}$ dependence of sampling variance described above.

Mini-batch training behaviour

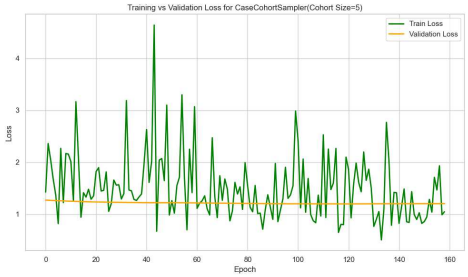
When training deep survival models under a mini-batch regime, the structural composition of the batch restricted risk sets directly dictates optimization stability and gradient quality. As illustrated in Figure 5, different risk set sampling schemes exhibit distinct convergence profiles when evaluated using a fixed batch size of 200.
For the *Full-Risk Sampler* (Figure 5a), the loss curves exhibit smooth, steady convergence, with the training loss decreasing from approximately 5.02 to 3.97 over 75 epochs. Notably, across the Full-Risk, Nested Case-Control, and larger Case-Cohort



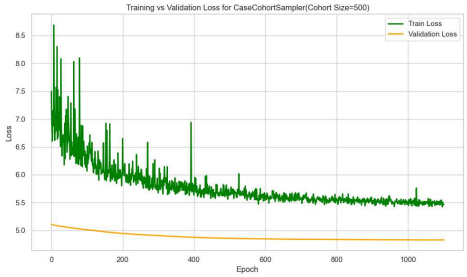
(a) Real world METABRIC loss curve for the full risk set model during full-batch training.



(b) Real world METABRIC nested case-control loss curve during full-batch training (controls = 5).



(c) Real world METABRIC case-cohort loss curve during full-batch training (subcohort size = 5).



(d) Real world METABRIC case-cohort loss curve during full batch training (subcohort size = 500).

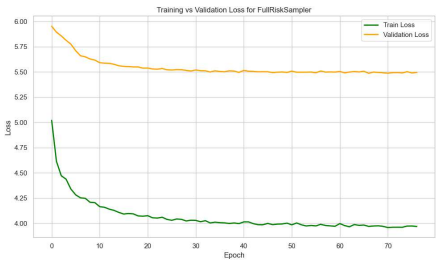
Fig. 4: Comparison of loss curves across different risk set sampling strategies on the real world METABRIC dataset under full batch training.

configurations, the validation loss remains consistently lower than the training loss. This behavior is structural: while the training loss is subject to the gradient variance of batch-restricted risk sets, the validation loss is computed over stable, unsampled risk sets, yielding a smoother and down scaled loss profile. This scaling behavior and the importance of risk-set composition match established literature on mini-batch optimization in deep survival settings [4, 7].

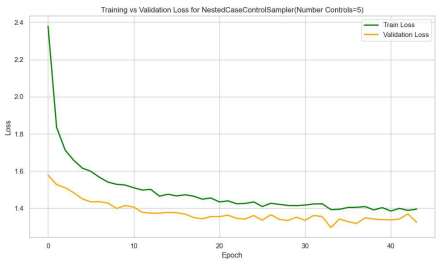
The targeted sampling strategies demonstrate the clear impact of risk-set information content per event. The *Nested Case-Control Sampler* (Figure 5b) yields an exceptionally stable and clean optimization trajectory, with the training loss smoothly decaying from 2.38 to under 1.40. This confirms that selecting a fixed number of controls ($m = 5$) per event provides high-information updates that stabilize mini-batch training.

In contrast, the *Case-Cohort Sampler* highlights a clear trade-off driven by the subcohort size. At a low subcohort size of 5 (Figure 5c), the training loss curve is highly jagged and volatile, fluctuating severely between 0.87 and 1.65. Because a subcohort size of 5 provides an impoverished representation of the risk set within a mini-batch, the optimization process suffers from high gradient variance. Compounded by the

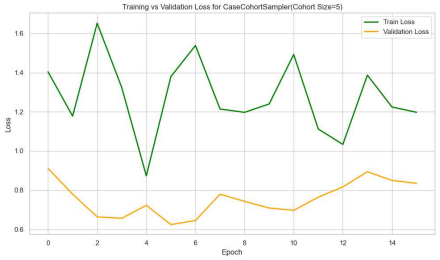
inherent stochasticity of mini-batch training, these erratic parameter updates lead to pronounced volatility across both the training and validation loss trajectories. However, when the subcohort size is increased to 100 (Figure 5d), this high variance behavior is mitigated. The larger subcohort size ensures a richer, more representative risk set per mini-batch, successfully smoothing out the gradient updates and producing a clean, stable convergence path from 4.73 down to approximately 3.37.



(a) Full-risk-set loss curve during mini-batch training (batch size = 200).



(b) Nested case-control loss curve during mini-batch training (controls = 5, batch size = 200).



(c) Case-cohort loss curve (cohort size = 5, batch size = 200).



(d) Case-cohort loss curve (cohort size = 100, batch size = 200).

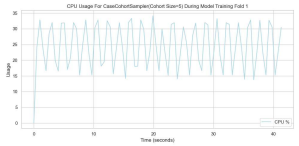
Fig. 5: Comparison of loss curves across different risk set sampling strategies under mini-batch training.

Mini-batch training on real-world data (METABRIC)

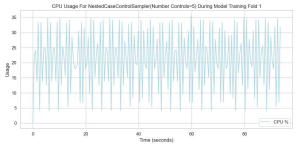
The convergence behavior and structural trade-offs observed on the simulated dataset largely carry over to the real-world METABRIC dataset, validating the effectiveness of mini-batch risk-set sampling in complex clinical settings. As shown by the supporting information.

Computational resource usage

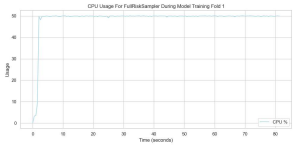
We monitored CPU and memory usage for each sampler and batching regime. Mini-batching produced a substantial reduction in peak memory usage, decreasing from roughly 6 GB (full-batch) to about 0.5 GB (mini-batch). This reduction represents the primary practical advantage of mini-batching in memory-constrained environments.
CPU usage patterns varied significantly by sampler and batching strategy. During mini-batch training, all samplers exhibited highly stable CPU utilization, consistently plateauing at approximately 50%. This stable behavior closely mimics the full-batch full-risk-set sampler, which also maintained a steady 50% utilization. In contrast, full-batch training for the other sampling techniques resulted in highly oscillatory CPU usage. Specifically, the full-batch case-cohort sampler fluctuated between roughly 14% and 34%, while the full-batch nested case-control sampler experienced even more drastic swings, persistently oscillating between approximately 4% and 35% (Figures 6a-6f). Memory usage trends were similar between samplers, though their temporal profiles differed because of how each sampler constructs and iterates over risk sets.



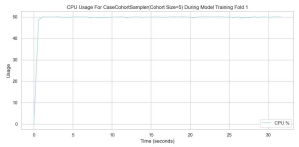
(a) CPU usage for case-cohort sampler (full-batch).



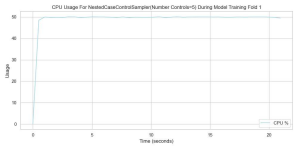
(b) CPU usage for nested case control sampler (full-batch).



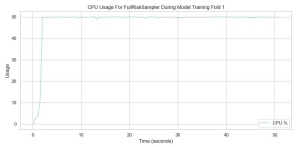
(c) CPU usage for full-risk-set sampler (full-batch).



(d) CPU usage for case-cohort sampler (mini-batch, batch size = 200).



(e) CPU usage for nested case control sampler (mini-batch, batch size = 200).



(f) CPU usage for full-risk-set sampler (mini-batch, batch size = 200).

Fig. 6: CPU resource usage comparison between full-batch (top row) and mini-batch (bottom row) training across different samplers.

Memory resource usage

We observed a substantial reduction in memory consumption when using sampling techniques compared to the full-risk-set approach during full-batch training. As shown in the supporting information. Increasing the subcohort or control size from 5 to 100 demonstrates the scalability limits of these sampling techniques. For the nested case-control sampler with 100 controls, memory usage increases substantially, reaching a

peak of approximately 12.1 GB in Fold 5—nearly approaching the consumption of the full-risk model. Similarly, expanding the case-cohort subcohort size to 100 increases its peak memory usage from 2.28 GB to roughly 7.65 GB. This indicates that while sampling reduces memory overhead, large control allocations diminish the memory benefits relative to full-batch full-risk training. We also observed that executing experiments sequentially within an interactive notebook environment (IPython/Jupyter) results in persistent memory allocations across k-folds. Later folds consistently begin from a higher baseline memory footprint, even after clearing explicit execution caches. To account for this baseline drift, per-fold consumption is evaluated by comparing the delta between the start and end memory levels of each fold, confirming that smaller sample configurations consistently require less incremental memory per fold. In contrast to full-batch optimization, mini-batch training enforces a strictly bounded memory footprint across all sampling methods and control sizes. Under mini-batch execution, expanding the subcohort or control size (k) from 5 to 100 yields virtually no increase in peak memory consumption. For example, during Fold 5 training of the nested case-control sampler, peak memory usage remains virtually identical at approximately 838 MB for $k = 5$ and 836 MB for $k = 100$. A similar pattern holds for the case-cohort sampler, where peak memory stabilizes at approximately 845 MB for $k = 5$ and 843 MB for $k = 100$. While mini-batching democratizes RAM requirements across sampling strategies, the computational cost of increasing the control size (k) manifests in wall-clock execution time rather than memory allocation. For instance, increasing k from 5 to 100 in the nested case-control sampler extends Fold 1 training time from 21 seconds to 48 seconds. Notably, the mini-batch full-risk model exhibits the lowest absolute RAM utilization ($\sim$577 MB in Fold 5), as mini-batching avoids the high-dimensional risk-set matrix calculations required during full-batch optimization.

Predictive performance and training time

Synthetic Data: Full-Batch Evaluation

We evaluated predictive performance using the cross-validated concordance index (Mean C-index $\pm$ SD), overall pooled out-of-fold C-index, and recorded training durations for each sampler. Under full-batch training, the nested case-control sampler produced C-index values that are nearly identical to the full-risk baseline across all control sizes (ranging from 0.7769 to 0.7791 against the baseline of 0.7789), indicating comparable discriminative performance without sacrificing accuracy. This consistency is visible in the cross-validated C-index in Figure 7 as well as the pooled out-of-fold C-index in Figure 8.

Conversely, the case-cohort design exhibits a substantial drop in predictive performance at very small subcohort sizes under full-batch execution. For a subcohort size of 5, the mean C-index drops to roughly 0.6021 (a $\sim$22% decrease from the baseline), accompanied by a higher Integrated Brier Score (0.1447). As shown in Figure 9, while these small case-cohort models train extremely fast (16.63s vs the baseline 142.58s), they suffer from higher variability and lower robustness. Increasing the subcohort size beyond 100 progressively recovers baseline performance, though extreme subcohort sizes (e.g., size 500) eventually exceed the baseline training duration (215.50s).

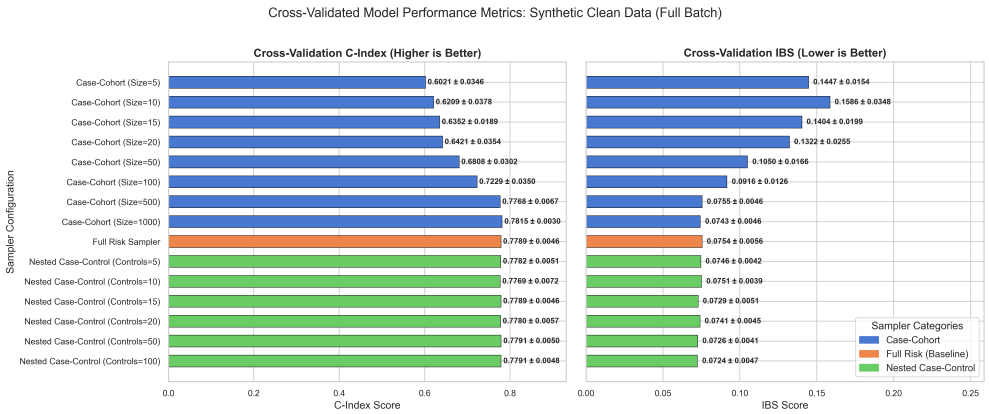


Fig. 7: Cross-validated C-index and IBS comparison across sampling techniques (synthetic clean data, full-batch).

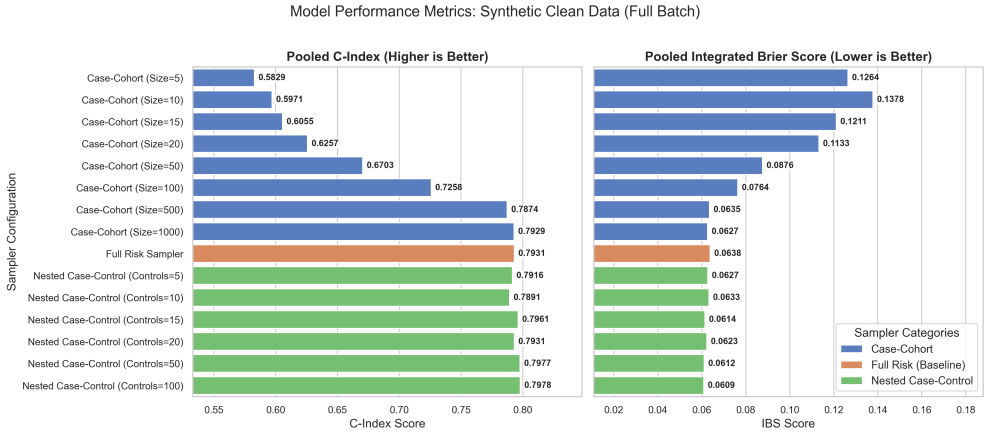


Fig. 8: Pooled out-of-fold C-index and IBS comparison across sampling techniques (synthetic clean data, full-batch).

Synthetic Data: Mini-Batch Evaluation

The performance trends under mini-batch training mirror those observed in full-batch training but with significantly muted penalties for small sample sizes. As shown in Figure 10, the nested case-control sampler again perfectly tracks the full-risk baseline ($\sim$0.7794). However, the case-cohort sampler performs much better in mini-batching at small subcohorts compared to full-batch training; a subcohort size of 5 yields a mean C-index of 0.7381 (only a $\sim$5.3% drop from baseline) and stabilizes rapidly as the size increases to 15 or higher.

This behavior highlights a clear trade-off between computational efficiency and stability. As illustrated by the standard deviation bounds in Figure 10 and the training

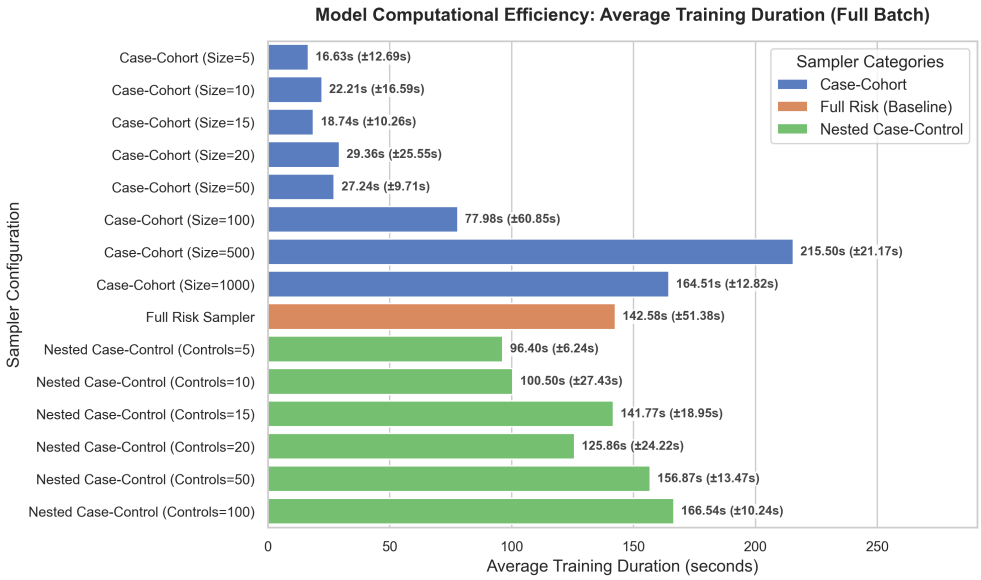


Fig. 9: Average training duration across sampling techniques (synthetic clean data, full-batch).

durations in Figure 12, smaller case-cohort sizes offer rapid convergence (∼19s) at the cost of higher validation variance. Increasing the subcohort size (e.g., to 100) or using nested case-control sampling improves validation performance, smooths variability, and achieves near-baseline C-index scores, though it pushes average training times closer to the full-risk mini-batch baseline (48.83s).

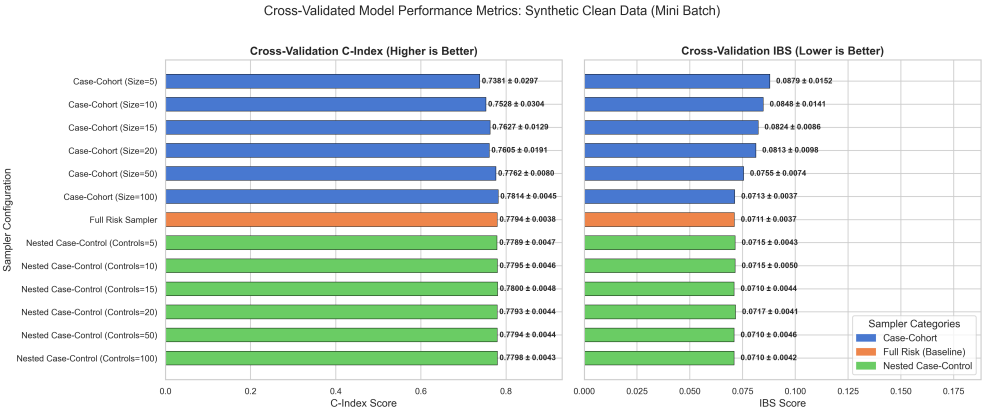


Fig. 10: Cross-validated C-index and IBS comparison across sampling techniques (synthetic clean data, mini-batch).

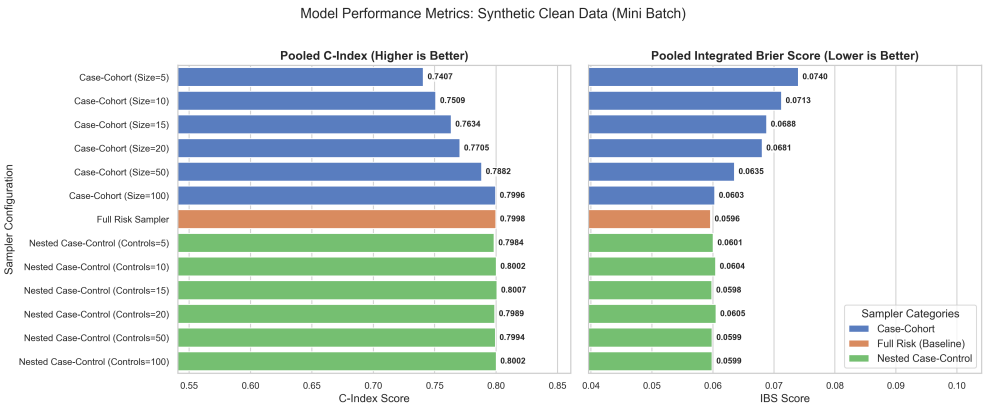


Fig. 11: Pooled out-of-fold C-index and IBS comparison across sampling techniques (synthetic clean data, mini-batch).

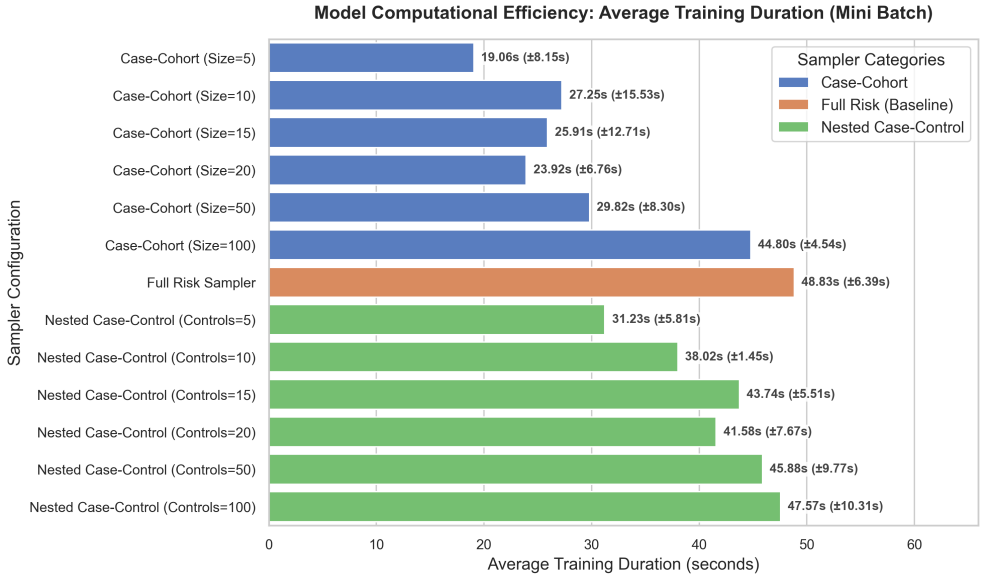


Fig. 12: Average training duration across sampling techniques (synthetic clean data, mini-batch).

.

Real-World Data: Full-Batch Evaluation

When applied to the more complex real-world dataset (METABRIC), the cross-validated metrics under full-batch training demonstrate remarkable stability across all sampling configurations. As shown in Figure 13, both case-cohort and nested

case-control methods achieve cross-validated mean C-indices tightly clustered between 0.5824 and 0.5905, closely tracking the full-risk baseline of 0.5957.
However, examining the pooled out-of-fold C-index reveals more variability (Figure 14). While most models hover between 0.55 and 0.59, an extreme case-cohort size ($k = 500$) peaks at a pooled C-index of 0.6241, suggesting that specific subcohort sizes may capture broader dataset variance when evaluated holistically. The primary advantage here remains computational efficiency: Figure 15 illustrates massive runtime reductions for smaller sampling sizes. For instance, the case-cohort sampler with a subcohort size of 15 completes training in just 12.23s, compared to 224.76s for the full-risk baseline.

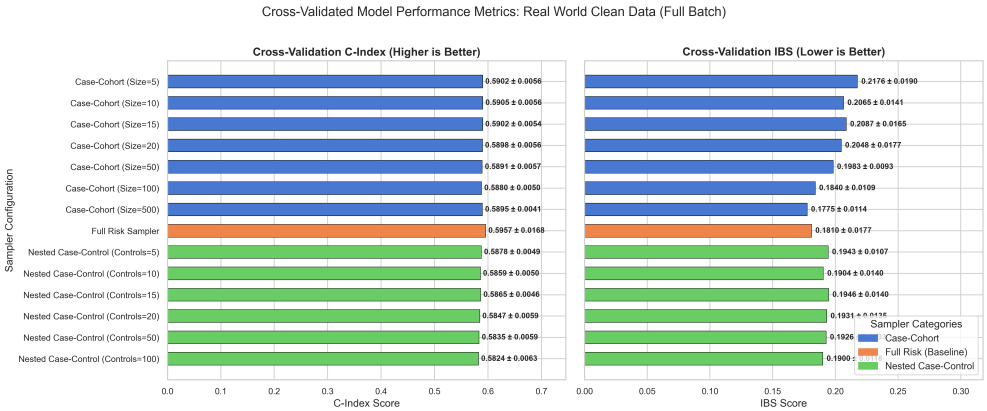


Fig. 13: Cross-validated C-index and IBS comparison across sampling techniques (real-world data, full-batch).

Real-World Data: Mini-Batch Evaluation

Mini-batching on the real-world dataset yields highly consistent cross-validated performance, with mean C-indices holding steady around ∼0.585 to 0.590 for virtually all configurations (Figure 16). Notably, however, mini-batching significantly alters the pooled C-index behavior. Figure 17 does not show the same distinct structure observed in the synthetic experiments or full-batch real-world evaluations; instead, nested case-control methods achieve the highest pooled C-index peaks under mini-batching, reaching 0.6394 for 100 controls.
This divergence is expected, as mini-batching loads small random subsets of the data at each iteration, which tends to *smooth* per-iteration metrics and can obscure gradual trends detectable under full-batch evaluation. A simple way to describe this smoothing effect is to view the observed per-iteration metric as an exponentially weighted moving average (EMA) of noisy batch estimates [19, 20]:

$$\tilde{C}_t \;=\; \alpha C_t + (1-\alpha)\tilde{C}_{t-1}, \qquad 0 < \alpha \leq 1, \tag{27}$$

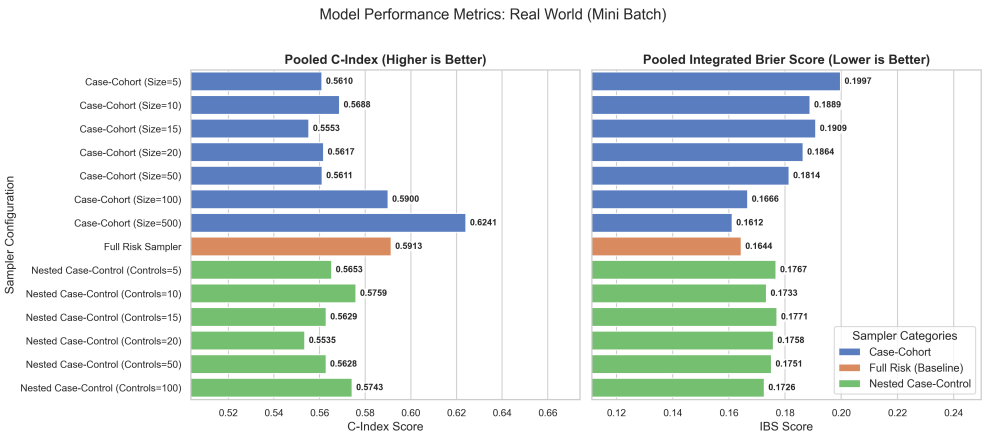


Fig. 14: Pooled out-of-fold C-index and IBS comparison across sampling techniques (real-world data, full-batch).

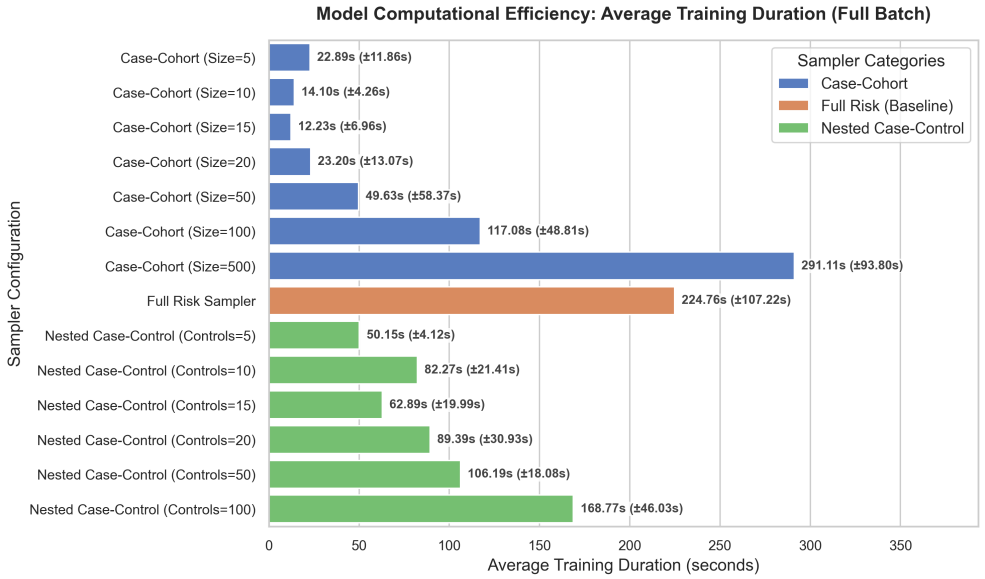


Fig. 15: Average training duration across sampling techniques (real-world data, full-batch).

where C_t is the instantaneous batch-level estimate and $\tilde{C}_t$ is the smoothed estimate. Smaller batches increase the batch-level noise in C_t, meaning the smoothed trace $\tilde{C}_t$ will show greater run-to-run variability and less clear monotonic trends. This effect, combined with the intrinsic noise of the METABRIC dataset, explains why mini-batch pooled C-index curves are less predictable than their full-batch counterparts.

Training durations under mini-batching (Figure 18) broadly follow expected patterns, offering savings at small sizes (e.g., case-cohort size 5 at 31.76s vs full risk at 82.69s). However, expanding to large sample sizes in a mini-batch regime quickly eclipses the baseline, with the nested case-control sampler ($k = 100$) taking up to 171.31s to converge.

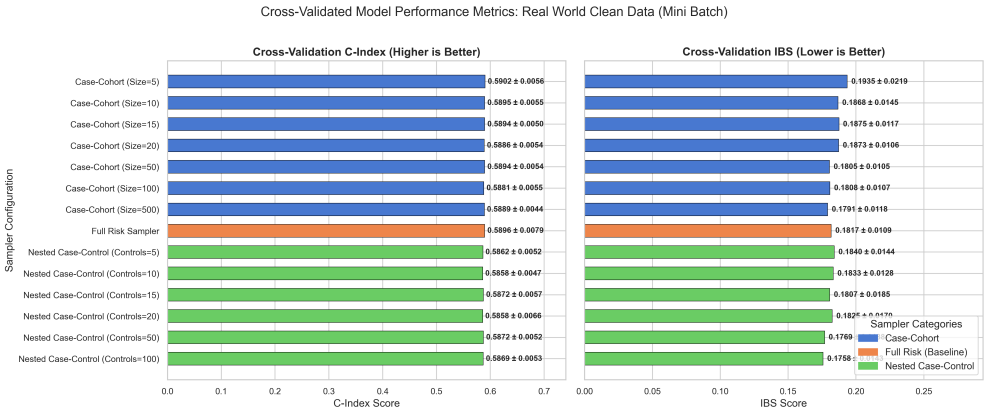


Fig. 16: Cross-validated C-index and IBS comparison across sampling techniques (real-world data, mini-batch).

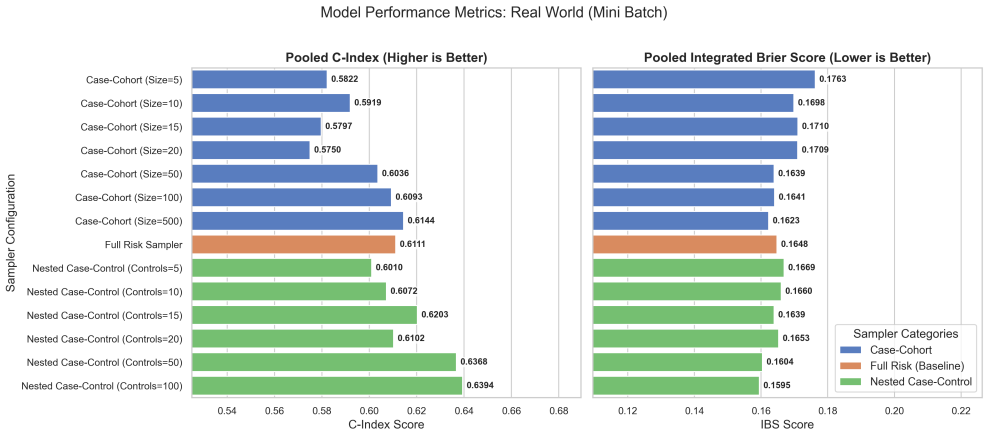


Fig. 17: Pooled out-of-fold C-index and IBS comparison across sampling techniques (real-world data, mini-batch).

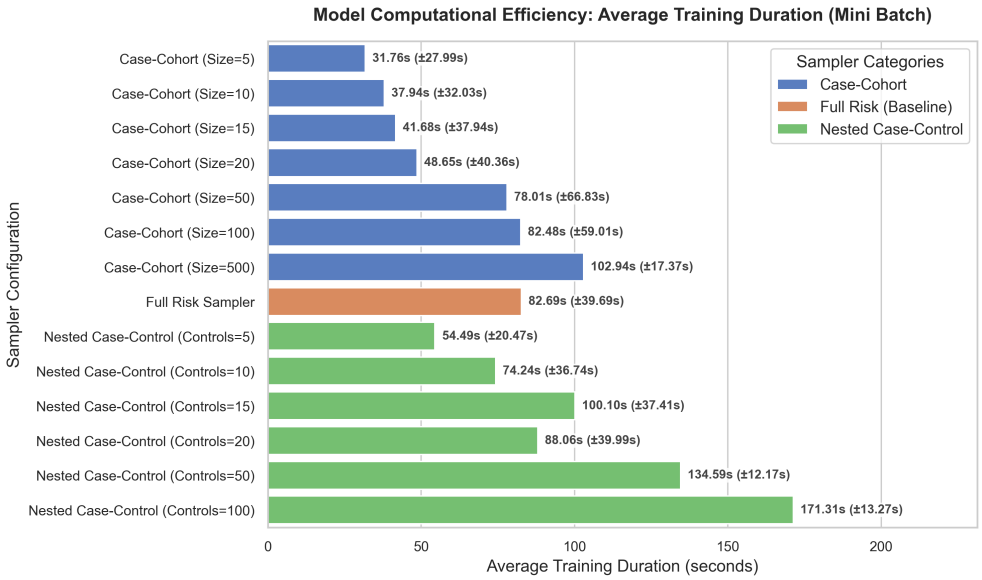


Fig. 18: Average training duration across sampling techniques (real-world data, mini-batch).

Risk-Score Associations and Comparisons

To assess how well the models estimate underlying hazard relative to the true data-generating process, we analyzed the Pearson (linear) and Spearman (monotonic) rank correlation coefficients. Because true log-partial hazard ratios (LPR) are fundamentally unobservable in clinical settings, this correlation analysis is strictly limited to the synthetic datasets where the true LPR is known. For the real-world METABRIC dataset, we instead compare the mean predicted risk scores across sampling strategies relative to the full-risk baseline.

Synthetic Data: True Risk Associations

Under full-batch execution (Figure 19), the nested case-control sampler demonstrates exceptional alignment with the full-risk baseline. Across all control sizes, it yields Pearson and Spearman coefficients heavily clustered between 0.96 and 0.98, matching the baseline's robust correlations (0.9727 Pearson, 0.9763 Spearman). Conversely, the case-cohort sampler struggles severely at small subcohort sizes. At a subcohort size of 5, the case-cohort model shows substantially weaker recovery of the underlying risk structure (0.3547 Pearson, 0.3008 Spearman), with recovery of the true risk ranking improving as the subcohort size increases and becoming substantially stronger at the larger subcohort sizes evaluated.

Mini-batching (Figure 20) significantly mitigates these penalties for the case-cohort sampler. By processing smaller, noisier subsets iteratively, the mini-batch case-cohort model of size 5 achieves a highly improved Pearson correlation of 0.8158. The nested case-control sampler under mini-batching remains virtually indistinguishable from

the baseline (≈0.99 Pearson and Spearman), further reinforcing its stability across batching regimes.

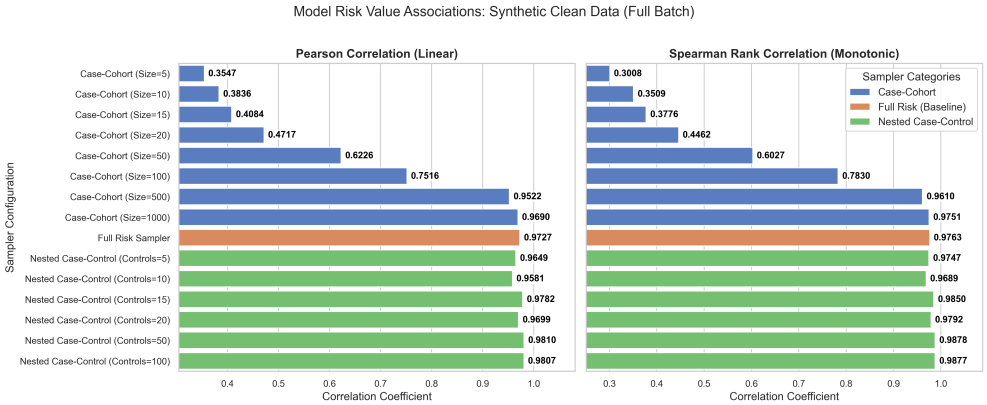


Fig. 19: Pearson and Spearman correlation between predicted and true log-partial hazards across sampling techniques (synthetic clean data, full-batch).

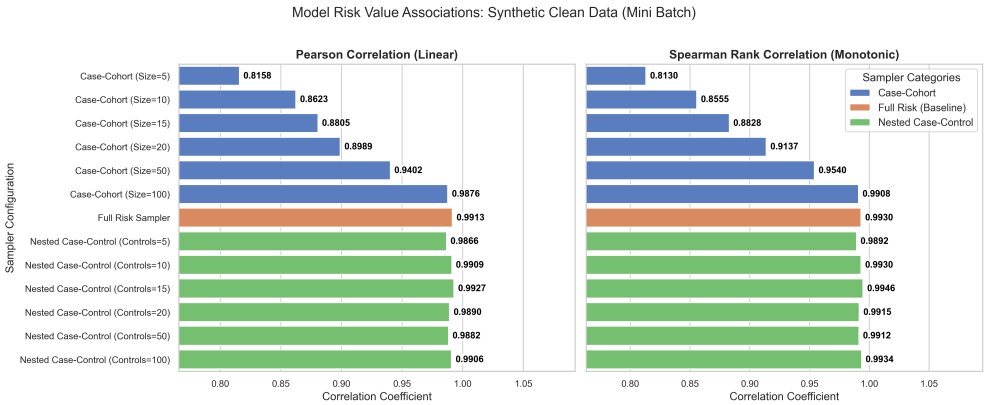


Fig. 20: Pearson and Spearman correlation between predicted and true log-partial hazards across sampling techniques (synthetic clean data, mini-batch).

Real-World Data: Predicted Risk-Score Comparison

For the real-world dataset, we examine the mean predicted risk score to understand how down-sampling scales the raw output values compared to the full-risk set (Figure 21).

The full-risk baseline yields a low mean predicted risk score under both full-batch (0.0314) and mini-batch (0.0349) regimes. The nested case-control sampler shows

differences in risk-score scale at small control sizes ($k = 5$ yields 0.4356), but the mean predicted risk score approaches the full-risk baseline as the number of controls increases, reaching 0.0544 at $k = 100$.
The case-cohort model, however, exhibits substantially larger differences in risk-score scale relative to the full-risk baseline when constrained to small subcohorts. Under full-batch training, a subcohort size of 5 yields a mean predicted risk score of 1.0635, compared with 0.0314 for the full-risk baseline. Under mini-batch training, this difference is reduced, with a mean risk score of 0.5969 at a subcohort size of 5. These results indicate greater sensitivity of the raw predicted risk-score scale to severe case-cohort down-sampling than observed for nested case-control sampling.

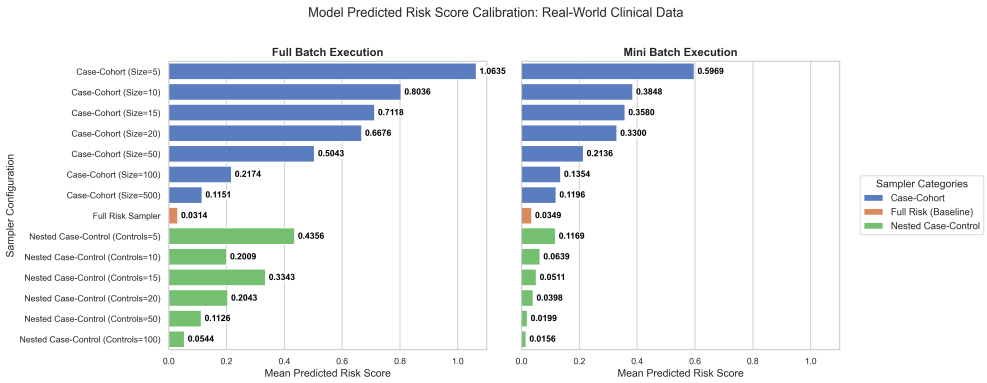


Fig. 21: Mean predicted risk-score comparison across sampling techniques for the real-world METABRIC data under full- and mini-batch training.

To understand the practical impact of down-sampling on clinical interpretability, we visualized the predicted survival curves generated by the models on the synthetic dataset. We examine therefore the full-batch execution regime, specifically comparing the full-risk baseline against the case-cohort sampler at extreme subcohort sizes ($k = 5$ and $k = 1000$). The full-risk sampler serves as the benchmark for survival prediction in this experiment. The model exhibits exceptional calibration; the predicted survival probabilities for both high- and low-risk cohorts tightly track the empirical Kaplan-Meier (KM) true risk curves as shown in the supporting information. Furthermore, the model effectively stratifies individuals by risk tier, generating proportional and visually distinct survival trajectories. When comparing individual predictions against the overall population baseline, the predicted curves show a realistic, well-distributed spread across the entire timeline. When the case-cohort subcohort size is heavily restricted to $k = 5$, the resulting survival predictions degrade significantly. As the subcohort size approaches the scale of the full dataset, the predictive capacity of the case-cohort sampler recovers. In stark contrast to the case-cohort method, the nested case-control sampler maintains remarkable structural stability and predictive accuracy even at the most extreme down-sampling levels. Even with

a minimal configuration of just 5 controls per event, the model remains highly calibrated. The predicted high- and low-risk survival probabilities track the true empirical Kaplan-Meier curves closely, without the severe underestimation seen in the equivalent case-cohort model. With the mini-batch execution regime, we first establish the performance of the full-risk sampler. As expected, the baseline model demonstrates excellent predictive fidelity. The figures as shown in the supporting information show tight calibration between the predicted risk curves and the empirical Kaplan-Meier baseline. This confirms that the baseline model's strong predictive capacity holds across different batching paradigms. When applying extreme down-sampling using the case-cohort method ($k = 5$) under a mini-batch regime, we observe a degradation in performance, though interestingly, it is notably less catastrophic than its full-batch counterpart as shown in the supporting information indicating that a subcohort size of 5 remains too restrictive to capture the full population heterogeneity. A striking dynamic emerges when marginally increasing the case-cohort size under the mini-batch regime. Unlike the full-batch scenario—which required scaling up to a massive $k = 1000$ to recover performance—the mini-batch approach allows the case-cohort sampler to effectively match the baseline with a remarkably modest size increase to just $k = 50$. This indicates a profound efficiency gain: the inherent stochasticity and frequent weight updates of the mini-batch regime allow the case-cohort method to synthesize global risk dynamics from far fewer samples, achieving full predictive capacity at a fraction of the computational cost (See supporting information for visualisation). Finally, we examine the nested case-control sampler under the mini-batch regime with a heavily restricted control set ($k = 5$). Continuing the trend observed in the full-batch experiments, the nested case-control approach proves exceptionally resilient to extreme down-sampling. This demonstrates that the nested case-control sampler successfully retains its structural integrity and baseline-equivalent performance regardless of whether it is trained in a full-batch or mini-batch environment. Additional calibration diagnostics, risk-stratified survival curves, and individual predicted survival trajectories for the synthetic and METABRIC experiments are provided in the Supporting Information (Figures S1–S19)

Discussion

The experiments demonstrate practical trade-offs between computational cost, training duration, and predictive performance when choosing a sampling strategy for deep neural networks (DNN) in the Cox framework. The nested case-control sampler ($m = 5$) consistently demonstrates high predictive fidelity, closely matching the full-risk set baseline in calibration and C-index across both full-batch and mini-batch regimes, while significantly reducing training durations.

Case-cohort sampling, however, exhibits high sensitivity to subcohort size and batching strategy. At severe down-sampling ($k = 5$), it exhibits substantial degradation in predictive performance, with a C-index reduction of approximately 23% relative to the full-risk baseline under full-batch training. Notably, our results show that mini-batching stabilizes case-cohort sampling substantially better than full-batching. While full-batch execution required massive subcohort sizes ($k = 1000$) to fully recover baseline performance—thereby inadvertently increasing training durations beyond that of

normal full-risk models-mini-batch execution recovered near-baseline C-index and IBS, together with improved agreement of the predicted survival curves, at a substantially smaller absolute subcohort size ($k = 100$).

This critical difference is driven by the relative representational density of the subcohort within the current computational graph. In our mini-batch regime (batch size of 200), a subcohort of $k = 100$ preserves a massive amount of the local sample context (100/200) for the model during each gradient update. Conversely, in a full-batch setting encompassing the entire dataset (≈ 6000 patients), $k = 100$ provides an extremely sparse global representation (100/6000). Consequently, the full-batch model must be forced up to $k = 1000$ to achieve a functional density (1000/6000) capable of stabilizing the C-index, IBS, and survival curves. Because processing $k = 100$ per batch is computationally lightweight, mini-batch case-cohort effectively avoids the longer training times observed in the scaled full-batch setup, proving that relative contextual density is more important than absolute subcohort size for balancing computational speed with predictive validity.

Rather than treating sub-cohort selection as a fixed constraint, the static sub-cohort sampling ratio $\alpha = \frac{k}{N}$ operates as a flexible hyperparameter that allows practitioners to explicitly tune the trade-off between computational efficiency and predictive performance. In resource-constrained environments or ultra-large datasets, selecting a lower ratio ($\alpha = 0.05 - 0.10$) drastically reduces the GPU memory footprint and epoch training duration at the cost of a minor reduction in rank discrimination (C-index). A ratio of $\alpha = 0.25$ serves as a balanced Pareto baseline, functioning as a practical default threshold that captures over 98% of full-batch risk-set fidelity while eliminating 75% of matrix operation overhead. Finally, higher sampling ratios ($\alpha \geq 0.50$) establish a high-fidelity regime that closely approximates full-batch partial likelihood evaluation for smaller datasets where compute memory is not a primary bottleneck.

For practitioners, the choice of sampler and batching strategy should be dictated by specific resource constraints and study designs.

Case-Cohort with Mini-Batching: If peak memory and overall training duration are the primary bottlenecks, case-cohort sampling combined strictly with mini-batch execution is highly recommended. Mini-batching stabilizes gradients via higher relative subcohort density per step, allowing practitioners to use smaller absolute subcohorts (e.g., $k = 100$) to retain baseline performance while securing the largest reductions in memory footprint and training time. Full-batch case-cohort should be avoided, as it requires prohibitively large subcohorts to remain stable.

Nested Case-Control: If predictive equivalence to the full-risk model is paramount and stable trajectory predictions are required across individuals, nested case-control sampling is an exceptionally effective compromise. By dynamically matching a small number of controls (e.g., $m = 5$) to failure times, it requires minimal tuning and maintains natural risk-set variance under both full-batch and mini-batch settings, offering reliable computational overhead reduction. Furthermore, the targeted construction of sampled risk sets can provide more informative gradient updates. Case-cohort sampling may be particularly relevant when the underlying data themselves arise from a case-cohort study design, thereby providing a natural alignment between the original sampling design and the analytical framework. All experiments were performed

on a dataset following our described preprocessing pipeline. Real-world datasets may exhibit different censoring mechanisms, event rates, and covariate distributions that could affect these trade offs. The empirical comparisons reported here depend on implementation details, hyperparameter choices (such as controls per event $m = 5$, batch size of 200, or subcohort fractions), and hardware configurations. Therefore, precise numerical thresholds (e.g., the exact subcohort size needed to match baseline in our setup) should not be generalized without further validation. Additionally, while we cited classical and modern references for methodology and evaluation, targeted benchmarking on multiple real-world cohorts would strengthen external validity.
Future work should validate these findings on multiple, diverse, real-world cohorts and investigate variance-reduction strategies (such as gradient accumulation and control variates) alongside advanced event-aware batching strategies to further stabilize mini-batch training. Exploring automated guidelines for dynamically selecting optimal controls per event and subcohort fractions given specific dataset sizes and hardware constraints would be practically useful. Finally, applying these samplers to modern neural survival architectures (e.g., DeepHit) may reveal additional interactions between architecture, sampling strategy, training dynamics, and different types of data such as interval-censored data [3, 4, 8].

Conclusion

Nested case-control and case-cohort samplers present pragmatic options for scaling survival analysis to larger datasets or resource-constrained environments. Nested case-control consistently matches full-risk performance with reduced computation, regardless of batching strategy. Case-cohort sampling yields substantial memory and training duration savings, but its success relies heavily on utilizing a mini-batch regime; the high relative sample density of mini-batching mitigates the need for the massive subcohort sizes that otherwise cripple full-batch performance. Careful sampler configuration, purposefully paired with the correct batching design, substantially mitigates computational trade-offs, enabling deep survival modeling at scale.
This paper has presented a comprehensive framework for integrating classical risk set sampling techniques with modern deep neural networks for survival analysis. We have demonstrated that nested case-control sampling achieves predictive performance nearly identical to full-risk approaches while notably reducing training time. Case-cohort sampling offers even greater memory efficiency and training speed gains, provided it is executed within a mini-batch regime where local contextual density is high enough to maintain stochastic stability without requiring prohibitive subcohort sizes. Our work bridges the methodological gap between traditional epidemiological designs and contemporary deep learning, providing practitioners with practical tools for scaling survival analysis to large-scale datasets. By preserving the predictive capacity and continuous-time formulation of Cox proportional hazards regression while leveraging the representational power of neural networks, our approach offers a principled pathway for efficient survival modeling in the era of big data.

Declarations

Acknowledgements

The first and second authors would like to thank the Emmy Noether Junior Research Group, Beyond Prediction: Statistical Inference with Machine Learning, at the Leibniz Institute for Prevention Research and Epidemiology (BIPS), Bremen, Germany, for their valuable guidance in the design of the model used in this study.

Supporting Information

More information may be available online in the Supporting Information section linked to the article.

Funding

The second author was funded by Georg Forster Research Fellowship (Alexander von Humboldt Foundation).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

This study is based on simulated data and publicly available, de-identified datasets. To ensure reproducibility, all source code and data used in this study have been made publicly available on the project GitHub repository at https://github.com/MichaelFalk1/Cox-Survival-Analysis.

Ethics approval and consent to participate

Ethics approval and informed consent were not required for this study because it was based on simulation experiments and analyses of publicly available secondary data.

Consent for publication

Not applicable. The manuscript does not contain any individual-level, identifiable, or personal data requiring consent for publication.

Authors' contributions

All authors reviewed, revised, and approved the final version of the manuscript.

References

[1] Cox, D.R.: Regression models and life-tables. Journal of the Royal Statistical Society: Series B (Methodological) **34**(2), 187–202 (1972) https://doi.org/10.1111/j.2517-6161.1972.tb00899.x

[2] Turnbull, B.W.: The empirical distribution function with arbitrarily grouped, censored and truncated data. Journal of the Royal Statistical Society: Series B (Methodological) **38**(3), 290–295 (1976) https://doi.org/10.1111/j.2517-6161.1976.tb01597.x

[3] LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015) https://doi.org/10.1038/nature14539

[4] Katzman, J.L., Shaham, A., Cloninger, A., Bates, J., Jiang, T., Kluger, Y.: Deepsurv: Personalized treatment recommender system using a cox proportional hazards deep neural network. BMC Medical Research Methodology **18**(1), 24 (2018) https://doi.org/10.1186/s12874-018-0482-1

[5] Kvamme, H., Borgan, Ø., Scheel, I.: Time-to-event prediction with neural networks and cox regression. Journal of Machine Learning Research **20**(129), 1–30 (2019)

[6] Prentice, R.L.: A case-cohort design for epidemiologic cohort studies and disease prevention trials. Biometrika **73**(1), 1–11 (1986) https://doi.org/10.2307/2336266

[7] Zeng, L., Tang, W., Ren, Z., Ding, Y.: Mini-batch Estimation for Deep Cox Models: Statistical Foundations and Practical Guidance. Taylor & Francis (2026)

[8] Lee, C., Zame, W.R., Yoon, J., Schaar, M.: Deephit: A deep learning approach to survival analysis with competing risks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, pp. 2314–2321 (2018). https://doi.org/10.1609/aaai.v32i1.11842

[9] Breslow, N.E.: Covariance analysis of censored survival data. Biometrics **30**(1), 89–99 (1974)

[10] Falk, M.: Cox Survival Analysis Experiments. GitHub repository. Private repository; available upon request (2025). https://github.com/MichaelFalk1/Cox-Survival-Analysis

[11] Harrell, F.E., Califf, R.M., Pryor, D.B., Lee, K.L., Rosati, R.A.: Evaluating the yield of medical tests. JAMA **247**(18), 2543–2546 (1982) https://doi.org/10.1001/jama.1982.03320430047030

[12] Graf, E., Schmoor, C., Sauerbrei, W., Schumacher, M.: Assessment and comparison of prognostic classification schemes for survival data. Statistics in Medicine **18**(17–18), 2529–2545 (1999) https://doi.org/10.1002/(SICI)1097-0258(19990915/30)18:17/18⟨2529::AID-SIM274⟩3.0.CO;2-5

[13] Pearson, K.: Note on regression and inheritance in the case of two variables. Proceedings of the Royal Society of London **58**, 240–242 (1895)

[14] Spearman, C.: The proof and measurement of association between two things. The American Journal of Psychology **15**(1), 72–101 (1904)

[15] Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York, NY, USA (2006)

[16] Barlow, W.E., Ichikawa, L., Rosner, D., Izumi, S.: Analysis of case-cohort designs. Journal of Clinical Epidemiology **52**(12), 1165–1172 (1999) https://doi.org/10.1016/S0895-4356(99)00102-X

[17] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd edn. Springer, New York, NY, USA (2009). https://doi.org/10.1007/978-0-387-84858-7

[18] Breslow, N.E., Wellner, J.A.: Weighted likelihood for semiparametric models and two-phase stratified samples, with application to cox regression. Scandinavian Journal of Statistics **34**(1), 86–102 (2007) https://doi.org/10.1111/j.1467-9469.2006.00523.x

[19] Oppenheim, A.V., Schafer, R.W.: Discrete-Time Signal Processing, 3rd edn. Pearson, Upper Saddle River, NJ, USA (2010)

[20] Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014) arXiv:1412.6980

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupportingInformation.pdf