The Annals of Statistics
2022, Vol. 50, No. 3, 1348–1375
https://doi.org/10.1214/21-AOS2153

DEEP LEARNING FOR THE PARTIALLY LINEAR COX MODEL

BY QIXIAN ZHONG[1,a], JONAS MUELLER[2,b] AND JANE-LING WANG[3,c]

[1]*Department of Statistics and Data Science, School of Economics, Xiamen University,* [a]*qxzhong@xmu.edu.cn*
[2]*Amazon Web Services,* [b]*jonasmue@amazon.com*
[3]*Department of Statistics, University of California, Davis,* [c]*janelwang@ucdavis.edu*

While deep learning approaches to survival data have demonstrated empirical success in applications, most of these methods are difficult to interpret and mathematical understanding of them is lacking. This paper studies the partially linear Cox model, where the nonlinear component of the model is implemented using a deep neural network. The proposed approach is flexible and able to circumvent the curse of dimensionality, yet it facilitates interpretability of the effects of treatment covariates on survival. We establish asymptotic theories of maximum partial likelihood estimators and show that our nonparametric deep neural network estimator achieves the minimax optimal rate of convergence (up to a polylogarithmic factor). Moreover, we prove that the corresponding finite-dimensional estimator for treatment covariate effects is $\sqrt{n}$-consistent, asymptotically normal and attains semiparametric efficiency. Extensive simulation studies and analyses of two real survival data sets show the proposed estimator produces confidence intervals with superior coverage as well as survival time predictions with superior concordance to actual survival times.

1. Introduction. Over the past decade, deep learning has begun substantially outperforming other statistical learning methods in many domains such as image analysis (Krizhevsky, Sutskever and Hinton (2012), Farabet et al. (2012), Szegedy et al. (2015)), speech recognition (Hinton et al. (2012), Graves, Mohamed and Hinton (2013)) and natural language processing (Collobert et al. (2011), Sarikaya, Hinton and Deoras (2014)). More recently, a new class of deep learning models have been introduced for survival analysis (Faraggi and Simon (1995), Chapfuwa et al. (2018), Katzman et al. (2018), Ren et al. (2019)), leading to sizeable performance improvements in this area. To shed light on the success of these methods, this paper provides theoretical analysis of deep neural networks (DNNs) applied to right censored data. For an in-depth overview on DNNs and their applications, we refer the reader to the review paper by LeCun, Bengio and Hinton (2015) and the recent monograph by Goodfellow, Bengio and Courville (2016).

A neural network is a parameterized composition of simple functions that can accurately model complex relationships when stacked in multiple layers. Typically, the operation at each layer is simply a linear transformation followed by a simple elementwise nonlinear transformation (which is called the *activation function* and might, e.g., be the rectifier function: $\max\{x, 0\}$). In the last decades, the neural network has been theoretically established as a powerful tool for function approximation. For instance, Cybenko (1989), Hornik, Stinchcombe and White (1989), Leshno et al. (1993) established that *shallow* neural networks with single *hidden layer* can approximate any continuous function to any degree of accuracy. This is often referred to as *universal approximation* in the machine learning literature.

Received May 2021; revised August 2021.
MSC2020 subject classifications. Primary 62N02, 62G20; secondary 62C20, 62G08.
Key words and phrases. Censored data, survival analysis, neural network, minimax estimation, partial likelihood, semiparametric efficiency.

Given a known degree of accuracy, Barron (1993, 1994), Mhaskar (1996), Pinkus (1999) further investigated the number of parameters required in a shallow neural network to approximate a certain smoothness class of functions. Compared to shallow networks, deep neural networks with many layers can achieve similar approximation error using exponentially fewer number of parameters (Telgarsky (2015), Mhaskar, Liao and Poggio (2017)). In order to approximate certain r-dimensional functions at a specified error level ϵ, DNNs only need $O(1/\epsilon)$ parameters while similar shallow networks require at least $O(1/\epsilon^r)$ parameters. This underscores one major advantage of deep over shallow neural networks. More extensive approximation theory for DNNs has been developed to study how certain architectures can model some specific classes of underlying functions, such as nonsmooth functions (Imaizumi and Fukumizu (2019)), Sobolev spaces (Yarotsky (2017), Gühring, Kutyniok and Petersen (2020)) and composite functions (Schmidt-Hieber (2017, 2020), Bauer and Kohler (2019)). The latter group of papers establish that DNNs are able to learn the parsimony structure of a composite function.

Extending this function approximation theory (which only quantifies the underlying bias), Schmidt-Hieber (2017, 2020) and Bauer and Kohler (2019) investigated the asymptotic properties of deep-learned statistical estimators for nonparametric regression. For n i.i.d. observations $(X_i, Y_i) \in [0,1]^r \times \mathbb{R}$, they considered the regression model

$$Y_i = g_0(X_i) + \epsilon_i, \tag{1}$$

where g_0 is an unknown function and ϵ_i are i.i.d. noise variables. It is well known that attempts to estimate g_0 generally suffer from a *curse of dimensionality* if the dimension r of X_i is large. Schmidt-Hieber (2017, 2020) and Bauer and Kohler (2019) showed that under mild smoothness/structure assumptions for g_0, DNN estimators cannot only circumvent the curse of dimensionality, but also achieve *optimal minimax rate of convergence* (up to some logarithmic factors). In contrast, estimators based on wavelet series are merely able to obtain suboptimal convergence rates for some classes of g_0, which illuminates clear theoretical advantages of DNN models (Schmidt-Hieber (2017, 2020)).

Although the nonparametric regression model (1) is highly flexible (thanks to the function approximation property of DNN), it does not facilitate intepretability of the underlying relationship between X and Y, particularly if we wish to understand the effect of particular treatment covariates $Z \in \mathbb{R}^p$ that comprise a subset of the variables in X. Furthermore, it is nontrivial to fit this model (1) when data have been *right censored*, which often occurs in survival analysis when a patient drops out of the study before the event of interest occurs. Here, right censored data refer to situation when the actual event time U is subject to right censoring by a censoring variable C, so the actual observations are (T, Δ), where $T = \min\{U, C\}$ is the observed event time and $\Delta = 1(U \le C)$ is an indicator variable with $\Delta = 1$ if T equals to an actual survival time U and $\Delta = 0$ otherwise.

For example, it is of principal interest to determine whether chemotherapy or hormonal treatment has an effect on breast cancer survival in the Rotterdam Breast Cancer Study (Foekens et al. (2000)), where the event time for 57.35% of patients were right censored. Nonparametric regression models like (1) are not applicable for right censored data, and also cannot provide interpretation or statistical inference for the effects of particular covariates. This paper addresses these issues by introducing an interpretable model that can fit right censored survival data, while still leveraging the powerful representation-learning capabilities of deep learning.

We consider the use of DNNs to augment the partially linear Cox model (PLCM) first introduced by Sasieni (1992a). In PLCM, the hazard function of the survival time U conditional on a vector covariate $(Z, X) \in \mathbb{R}^p \times \mathbb{R}^r$ is represented as

$$\lambda(u|X, Z) = \lambda_0(u) \exp\{\theta_0^\top Z + g_0(X)\}. \tag{2}$$

Here, λ_0 is an unknown baseline hazard function, $\theta_0 \in \mathbb{R}^p$ is an unspecified parameter and $g_0 : \mathbb{R}^r \to \mathbb{R}$ is an unknown function. Here, we leverage DNN to represent the function g_0 and distinguish our method as the *deep partially linear Cox model* (DPLCM). This model is quite flexible: it includes both the popular Cox proportional hazards model [CPH, Cox (1972, 1975)] in the absence of nontreatment covariates X, as well as the nonparametric Cox models (Hastie and Tibshirani (1990), Sleeper and Harrington (1990), O'Sullivan (1993), Kooperberg, Stone and Truong (1995), Chen and Zhou (2007), Chen et al. (2010)) in the absence of treatments Z. Thus, our model not only inherits the simple interpretation of the finite-dimensional parameter θ_0 in the Cox proportional hazards model, but also models more complex nonlinear effects of the covariate X, and thus can more accurately capture properties of real survival data.

Previous work has studied the general DPLCM model in (2) but has difficulties to handle multivariate X. For example, Therneau, Grambsch and Fleming (1990) and Fleming and Harrington (1991) used martingale residuals to investigate the case of univariate X. Sasieni (1992a, 1992b) focused on calculating information bounds and asymptotic efficiency of θ_0 estimates, and suggested a spline estimate of function g_0 without details about its asymptotics. Dabrowska (1997) established asymptotic properties of PLCM estimators obtained by maximizing a smoothed profile likelihood, but the required multivariate numerical integration will be intractable in practice when there are many auxiliary covariates. With an additional additive assumption of g_0, Huang (1999) showed that the partial likelihood estimates of θ_0 achieves the information bound and the estimates of g_0 converges to g_0 at the standard one-dimensional nonparametric convergence rate. Later on, Du, Ma and Liang (2010) studied variable selection of PLCM with high-dimensional covariates Z.

Although deep learning has received increasing attention in survival analysis, to date there is little theoretical understanding of model (2) when g_0 is approximated using a DNN. Merely considering nontreatment covariates X, Faraggi and Simon (1995) employed an one hidden-layer neural network to estimate $g_0(X)$. However, their model did not produce significant improvements (Xiang et al. (2000)) in terms of concordance index (Harrell et al. (1982)), a version of the receiver operating characteristic curve that measures the predictive power of survival models on censored survival data. Later, with the same replacement, Katzman et al. (2018) considered multilayer neural network to approach the same model and obtained remarkable results in applications. Similar variants of the CPH model using more complex neural network architectures have been developed for particular applications, such as genomic data (Yousefi et al. (2017), Ching, Zhu and Garmire (2018)), clinical research (Matsuo et al. (2019)) and medical imaging data (Haarburger et al. (2019), Li et al. (2019)). Additional deep learning methods to study survival data has also emerged recently including the hierarchical generative approach (Ranganath et al. (2016)), the generative adversarial network approach (Chapfuwa et al. (2018)) and the recurrent neural network approach (Giunchiglia, Nemchenko and Van der Schaar (2018), Ren et al. (2019)). Despite their success in prediction, the aforementioned neural network approaches are black-box models that lack interpretability for treatment effects, since the treatment covariate is lumped with all other covariates in a nonlinear/nonconvex neural network. It is thus difficult to provide trustworthy estimates of the treatment effects and uncertainty quantification for estimated effects.

In addition to being ill suited for treatment effect estimation, current approaches to deep learning for survival analysis elude our current mathematical understanding (Katzman et al. (2018), Ranganath et al. (2016), Lee et al. (2018), Hao et al. (2019)). In this work, we maximize a partial likelihood function to estimate the nonparametric function g_0 in (2) using a deep neural network, and establish theoretical properties of the resulting estimator. Analysis of neural networks for nonlinear function approximation is the subject of a vast literature (Anthony and Bartlett (1999), Bauer and Kohler (2019), Unser (2019), Dou and Liang (2021), Farrell, Liang and Misra (2021)).

One advantage of the deep neural network approach is that it can accommodate a rich class of g_0 to avoid the curse of dimensionality and yields faster convergence rates than nonparametric smoothing methods. Here, we postulate an intrinsic dimension assumption on g_0 and show that its DNN-estimator is consistent and converges at a rate that depends only on the intrinsic dimension and smoothness of this function. This intrinsic dimension, introduced in Section 2, essentially represents the complexity of function g_0 and contains a large number of previously-studied function classes, such as, single (multiple) index functions (Ichimura (1993), Härdle, Hall and Ichimura (1993)), (generalized) additive functions (Stone (1985), Horowitz and Mammen (2007)) and (generalized) hierarchical interaction functions (Bauer and Kohler (2019)).

Under mild regularity conditions, and using a DNN whose complexity grows with the data (as would be used in practical applications), we establish consistency and convergence rate of the nonparametric function estimator and asymptotic normality for the parametric component of the DPLCM model. The convergence rate of the function estimator is minimax optimal (up to a polylogarithmic factor) and the parametric estimator is semiparametric efficient (Bickel et al. (1993), van der Vaart (2000), Kosorok (2008)).

To summarize, this paper provides a flexible-yet-interpretable partially linear Cox model that performs well for multivariate covariates by leveraging a powerful DNN model to represent nonlinear effects. Optimal asymptotic theory is established for both the parametric (linear) and nonparametric (DNN) components of our model. To the best of our knowledge, this is the first paper that contains theoretical support for proportional hazards models that utilize deep learning. Our work is inspired by the recent theoretical developments in deep learning for nonparametric regression (Schmidt-Hieber (2017, 2020), Bauer and Kohler (2019)). However, our analysis contains major differences, in part due to complications from random censoring in survival data. First, DPLCM is comprised of two nonparametric components as well as a parametric component, while nonparametric regression models only involve a single nonparametric component. DPLCM's parametric component facilitates interpretability of treatment effects but its analysis requires different theoretical tools than nonparametric regression. For instance, to establish semiparametric efficiency, we require martingale theory to derive the efficient score and information bound for θ_0. Second, the theory of DNN nonparametric regression is built on the framework of least squares loss, while our approach maximizes a log partial likelihood. Third, establishing a minimax lower bound for the nonparametric target requires restrictions on the error distributions in the nonparametric regression setting, and Schmidt-Hieber (2017, 2020) restrictively assume errors are normally distributed. In contrast, our proportional hazards assumption itself allows a minimax bound to be established without any such additional assumptions.

We present our proposed estimation procedure in Section 2 and the main theoretical properties of the estimators in Section 3. In Section 4, we apply the proposed method to simulated and real data and compare it with CPH (Cox (1972)) and the partially linear additive Cox model (Huang (1999)), two popular methods for estimating treatment effects on survival. Section 5 concludes the discussion and the proofs are relegated to Section 6. Many auxiliary results are provided in the Supplementary Material (Zhong, Mueller and Wang (2022)).

2. Methodology.

2.1. *Deep neural networks*. We briefly review the use of DNNs as function approximators; for further details; see Goodfellow, Bengio and Courville (2016). Let K be a positive integer and $\boldsymbol{p} = (p_0, \ldots, p_K, p_{K+1})$ be some positive integer sequence. A $(K+1)$-layer DNN with layer-width $\boldsymbol{p}$ is a composite function $g : \mathbb{R}^{p_0} \to \mathbb{R}^{p_{K+1}}$ recursively defined as

$$
\begin{aligned}
&g(x) = W_K g_K(x) + v_K, \\
&g_K(x) = \sigma\big(W_{K-1} g_{K-1}(x) + v_{K-1}\big), \ldots, g_1(x) = \sigma(W_0 x + v_0).
\end{aligned} \tag{3}
$$

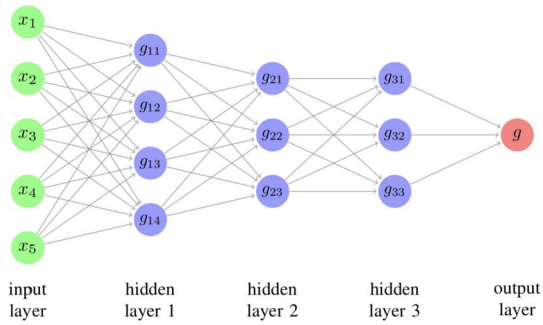


FIG. 1. *A 4-layer deep neural network with* $\boldsymbol{p} = (5, 4, 3, 3, 1)$.

Here, the matrices $W_k \in \mathbb{R}^{p_{k+1} \times p_k}$ and vectors $v_k \in \mathbb{R}^{p_{k+1}}$ (for $k = 0, \ldots, K$) are the parameters of this DNN g. Chosen a priori, the activation functions σ are simple nonlinear transformations that operate componentwise, that is, $\sigma((x_1, \ldots, x_{p_k})^\top) = (\sigma(x_1), \ldots, \sigma(x_{p_k}))^\top$, which thus gives $g_k = (g_{k1}, \ldots, g_{kp_k})^\top : \mathbb{R}^{p_{k-1}} \to \mathbb{R}^{p_k}$ for $k = 1, \ldots, K$. While many choices of activation function are considered in deep learning, the most popular one is the rectified linear unit (ReLU) in Nair and Hinton (2010):

$$\sigma(x) = \max\{x, 0\}.$$

Throughout this paper, we focus on ReLU activation for its widespread use (LeCun, Bengio and Hinton (2015), Ramachandran, Zoph and Le (2017)), empirical success (Krizhevsky, Sutskever and Hinton (2012)) and theoretical support (Liang and Srikant (2016), Yarotsky (2017), Schmidt-Hieber (2020)). For the DNN in (3), K denotes the depth of the network and vector $\boldsymbol{p}$ lists the width of each layer (p_0 is the dimension of the input variable, $p_1, \ldots, p_K$ are the dimensions of the K hidden layers, and p_{K+1} is the dimension of the output layer). The matrix entries $(W_k)_{i,j}$ are the weight linking the jth neuron in layer k to the ith neuron in layer $k+1$, and the vector entries $(v_k)_i$ represent a shift term associated with the ith neuron in layer $k+1$. For example, Figure 1 depicts a 4-layer DNN with $\boldsymbol{p} = (5, 4, 3, 3, 1)$.

Let $\mathbb{N}_+$ be the set of all positive natural numbers. Given $K \in \mathbb{N}_+$ and $\boldsymbol{p} \in \mathbb{N}_+^{K+2}$, we consider a class of DNN:

$$\begin{aligned}\mathcal{G}(K, \boldsymbol{p}) = \big\{ & g : g \text{ is a DNN with } (K+1) \text{ layers and width vector } \boldsymbol{p} \text{ such that} \\ & \max\{\|W_k\|_\infty, \|v_k\|_\infty\} \le 1, \text{ for all } k = 0, \ldots, K\big\},\end{aligned} \tag{4}$$

where $\|\cdot\|_\infty$ denotes the sup-norm of matrix or vector. Empirically the size of the learned matrices W_k and vectors v_k are rarely large when the size of initial matrices and vectors used to initialize stochastic gradient training are relatively small (as is typically the case). Thus we just consider the DNNs whose matrices and vectors are bounded by one. In practice, a deep feedforward network with fully-connected layers contains a huge number of parameters, which can lead to overfitting. This issue can be mitigated by pruning weights, which reduces the total number of nonzero parameters such that the network's layers are only sparsely connected (Han et al. (2015), Srinivas, Subramanya and Venkatesh Babu (2017), Schmidt-Hieber (2017, 2020)). Following similar methodology, we consider, for $s \in \mathbb{N}_+$ and $D > 0$, a class of sparse neural networks

$$\mathcal{G}(K, s, \boldsymbol{p}, D) := \left\{ g \in \mathcal{G}(K, \boldsymbol{p}) : \sum_{k=1}^{K} \|W_k\|_0 + \|v_k\|_0 \le s, \|g\|_\infty \le D \right\}, \tag{5}$$

where $\|\cdot\|_0$ is the number of nonzero entries of matrix or vector, and $\|g\|_\infty$ is the sup-norm of function g.

2.2. *Estimation.* Consider a survival study with right-censored data, where U and C denote survival and censored time, respectively, and $(Z, X) \in \mathbb{R}^p \times \mathbb{R}^r$ form a $(p+r)$-dimensional vector of covariates (Z contains the treatment indicator). Due to censoring, we only observe n i.i.d. copies $(T_1, \Delta_1, Z_1, X_1), \ldots, (T_n, \Delta_n, Z_n, X_n)$ from (T, Δ, Z, X), where $T = \min\{U, C\}$ is the observed event time and $\Delta = 1(U \le C)$ is an indicator variable with $\Delta = 1$ if T equals to an actual survival time U and $\Delta = 0$ otherwise. As is standard in survival analysis (Cox and Oakes (1984), Fleming and Harrington (1991)), we assume that the survival time U and censored time C are independent conditional on the covariates (Z, X).

In the DPLCM (2), the parameter θ_0, nonparametric function g_0, and baseline function λ_0 are all unknown.

We approximate g_0 using a DNN $g \in \mathcal{G}$, whose input is the r-dimensional vector X and output is a scalar-value. Here, we employ the shorthand $\mathcal{G} = \mathcal{G}(K, s, \boldsymbol{p}, \infty)$. More precisely, we first estimate (θ_0, g_0) by maximizing the log partial likelihood (Cox (1972, 1975)):

$$(\hat{\theta}, \hat{g}) = \underset{(\theta, g) \in \mathbb{R}^p \times \mathcal{G}}{\arg\max}\, L_n(\theta, g), \tag{6}$$

where $L_n(\theta, g) = \frac{1}{n}\sum_{i=1}^n \Delta_i [\theta^\top Z_i + g(X_i) - \log \sum_{j: T_j \ge T_i} \exp\{\theta^\top Z_j + g(X_j)\}]$.

3. Theoretical results. In this section, we study the asymptotic properties of the log partial likelihood estimators in (6). Some restrictions on the nonparametric function g_0 are needed and we assume that it belongs to a Hölder class of smooth functions, which is fairly broad and also adopted by Schmidt-Hieber (2017, 2020). Specifically, a Hölder class of smooth functions with parameters $\alpha, M > 0$ and domain $\mathbb{D} \subset \mathbb{R}^r$ is

$$\mathcal{H}_r^\alpha(\mathbb{D}, M) = \Bigg\{ g : \mathbb{D} \to \mathbb{R} : \sum_{\beta: |\beta| < \alpha} \left\| \partial^\beta g \right\|_\infty + \sum_{\beta: |\beta| = \lfloor \alpha \rfloor} \sup_{x, y \in \mathbb{D}, x \ne y} \frac{|\partial^\beta g(x) - \partial^\beta g(y)|}{\|x - y\|_\infty^{\alpha - \lfloor \alpha \rfloor}} \le M \Bigg\},$$

where $\lfloor \alpha \rfloor$ is the largest integer strictly smaller than α, $\partial^\beta := \partial^{\beta_1} \ldots \partial^{\beta_r}$ with $\beta = (\beta_1, \ldots, \beta_r)$, and $|\beta| = \sum_{k=1}^r \beta_k$. Let $q \in \mathbb{N}$, $M > 0$, $\alpha = (\alpha_0, \ldots, \alpha_q) \in \mathbb{R}_+^{q+1}$ and $\boldsymbol{d} = (d_0, \ldots, d_{q+1}) \in \mathbb{N}_+^{q+2}$, $\tilde{\boldsymbol{d}} = (\tilde{d}_0, \ldots, \tilde{d}_q) \in \mathbb{N}_+^{q+1}$ with $\tilde{d}_j \le d_j$, $j = 0, \ldots, q$, where $\mathbb{R}_+$ is the set of all positive real numbers. We further assume that g_0 belongs to a *composite smoothness* function class:

$$\begin{aligned} \mathcal{H}(q, \alpha, \boldsymbol{d}, \tilde{\boldsymbol{d}}, M) := \big\{ & g = g_q \circ \cdots \circ g_0 : g_i = (g_{i1}, \ldots, g_{id_{i+1}})^\top \text{ and} \\ & g_{ij} \in \mathcal{H}_{\tilde{d}_i}^{\alpha_i}([a_i, b_i]^{\tilde{d}_i}, M]), \text{for some } |a_i|, |b_i| \le M \big\}. \end{aligned} \tag{7}$$

Functions in this class are characterized by two kind of dimensions, $\boldsymbol{d}$ and $\tilde{\boldsymbol{d}}$, where the latter represents the *intrinsic dimension* of the function. For example, if

$$g(x) = g_{21}\big(g_{11}\big(g_{01}(x_1, x_2), g_{02}(x_3, x_4)\big), g_{12}\big(g_{03}(x_5, x_6), g_{04}(x_7, x_8)\big)\big), \quad x \in [0, 1]^8, \tag{8}$$

and g_{ij} are twice continuously differentiable, then smoothness $\alpha = (2, 2, 2)$, dimensions $\boldsymbol{d} = (8, 4, 2, 1)$ and $\tilde{\boldsymbol{d}} = (2, 2, 2)$.

The composite smoothness class $\mathcal{H}(q, \alpha, \boldsymbol{d}, \tilde{\boldsymbol{d}}, M)$ subsumes a rich set of classical smoothness classes. For instance, Stone (1985) considered nonparametric regression with generalized additive functions

$$g(x) = \sum_{i=1}^{I_1} g_i \left(\sum_{j=1}^r a_{ij} x_j \right),$$

where the $g_i, i = 1, \ldots, I_1$, are univariate Hölder smoothness functions and some $I_1 \in \mathbb{N}_+$. Horowitz and Mammen (2007) analyzed a more complex nonparametric regression model

$$g(x) = g_{i_0}\Bigg(\sum_{i_1=1}^{I_1} g_{i_1}\Bigg(\sum_{i_2=1}^{I_2} g_{i_1,i_2}\Bigg(\cdots \sum_{i_k=1}^{I_k} g_{i_1,\ldots,i_k}(x_{i_1,\ldots,i_k})\Bigg)\Bigg)\Bigg),$$

where $g_{i_0}, g_{i_1}, \ldots, g_{i_1,\ldots,i_k}$ are univariate Hölder smoothness functions, $x_{i_1,\ldots,i_k}$ are one-dimensional elements of a vector $x \in \mathbb{R}^r$ and $I_1, \ldots, I_k \in \mathbb{N}_+$.

Furthermore, we denote $\tilde{\alpha}_i = \alpha_i \prod_{k=i+1}^{q}(\alpha_k \wedge 1)$ and $\gamma_n = \max_{i=0,\ldots,q} n^{-\tilde{\alpha}_i/(2\tilde{\alpha}_i+\tilde{d}_i)}$ with notation $a \wedge b := \min\{a, b\}$. Recalling the DNN definition in (3), we first assume the following about the structure of the DNN model and the covariate:

(A1) $K = O(\log n)$, $s = O(n\gamma_n^2 \log n)$ and $n\gamma_n^2 \lesssim \min(p_k)_{k=1,\ldots,K} \le \max(p_k)_{k=1,\ldots,K} \lesssim n$.

(A2) The covariate (Z, X) takes value in a bounded subset of $\mathbb{R}^{p+r}$ with joint probability density function bounded away from zero, and there exists a norm bound for the parameter $\theta_0 \in \mathbb{R}^p_M := \{\theta \in \mathbb{R}^p : \|\theta\| \le M\}$. Without loss of generality, we assume that the domain of X is taken to be $[0, 1]^r$.

Assumption (A1) determines the structure of the neural network family $\mathcal{G}(K, s, \boldsymbol{p}, D)$ in (5). According to Anthony and Bartlett (1999) and Schmidt-Hieber (2017, 2020), more flexible neural networks (with more parameters) can achieve smaller *approximation error*. The latter is defined as the distance between true function g_0 and $\tilde{g}$, the projection of g_0 onto the space of functions that can be implemented by a neural network from $\mathcal{G}(K, s, \boldsymbol{p}, \infty)$. However, a larger neural network often leads to larger *estimation error*, the distance between $\hat{g}$ and $\tilde{g}$. Assumption (A1) thus provides a trade-off between the approximation error and estimation error, while (A2) is a standard assumption for semi/nonparametric regression (Horowitz (2009)).

Because of the presence of two nonparametric components, the baseline hazard function λ_0 and g_0, model (2) is not identifiable. But this can be corrected by a constraint $\mathbb{E}\{g_0(X)\} = 0$ along with additional identifiability assumptions below on the covariate (Z, X).

(B1) The nonparametric function g_0 is an element of $\mathcal{H}_0 = \{g \in \mathcal{H}(q, \alpha, \boldsymbol{d}, \tilde{\boldsymbol{d}}, M) : \mathbb{E}\{g(X)\} = 0\}$ and the matrix $\mathbb{E}\{Z - \mathbb{E}(Z|X)\}^{\otimes 2}$ is nonsingular, where $v^{\otimes 2} = vv^\top$ for a column vector v.

Our theoretical analysis also adopts the following standard assumptions on Cox model:

(B2) The study ends at time τ and there exits a small constant $\delta > 0$ such that $\mathbb{P}(\Delta = 1|X, Z) \ge \delta$ and $\mathbb{P}(U \ge \tau|X, Z) \ge \delta$ almost surely with respect to the probability measure of (X, Z).

(B3) There exist constants $0 < c_1 < c_2 < \infty$ such that the subdensity $p(t, x, \Delta = 1)$ of $(T, X, \Delta = 1)$ satisfies $c_1 < p(t, x, \Delta = 1) < c_2$ for all $(t, x) \in [0, \tau] \times [0, 1]^r$.

(B4) For some $k > 1$, the kth partial derivative of the subdensity $p(t, x, z, \Delta = 1)$ of $(T, X, Z, \Delta = 1)$ with respect to $(t, x) \in (0, \tau) \times (0, 1)^r$ exists and is bounded.

The space $\mathcal{H}_0$ in (B1) is a rich class, since for any $g \in \mathcal{H}(q, \alpha, \boldsymbol{d}, \tilde{\boldsymbol{d}}, M)$, $g/2 - \mathbb{E}\{g(X)/2\} \in \mathcal{H}_0$. The first part of Assumption (B2), $\mathbb{P}(\Delta = 1|Z, X) \ge \delta$, is a minimal assumption that guarantees a nonnull probability of observing noncensored data. The second part, $\mathbb{P}(U \ge \tau|Z, X) \ge \delta$, is needed to ensures that the baseline cumulative hazard function $\Lambda_0(t) = \int_0^t \lambda_0(s)\,ds$ is bounded at the end of the study time τ, and this is often satisfied in practice when some subjects are still alive at the end of the study. In Assumption (B3), the

subdensity $p(t, x, \Delta = 1)$ is defined as

$$p(t, x, \Delta = 1) = \frac{\partial^2 \mathbb{P}(T \le t, X \le x, \Delta = 1)}{\partial t \partial x}.$$

Likewise, the subdensity $p(t, x, z, \Delta = 1)$ in Assumption (B4) has a similar definition. Assumption (B3) ensures that the information bound for θ exists, and Assumption (B4) is used to establish the asymptotic normality of $\hat{\theta}$. These assumptions are not particularly stringent, especially compared with existing survival analysis theory (Huang (1999), Jiang and Jiang (2011)). We also provide one simple example that simultaneously satisfies all of these assumptions in the Supplementary Material (Zhong, Mueller and Wang (2022)), noting there are many more such examples.

THEOREM 3.1. *Under assumptions* (A1), (A2), (B1) *and* (B2), *there exists an estimator* $\hat{g}$ *in* (6) *satisfying* $\mathbb{E}\{\hat{g}(X)\} = 0$, *such that*

$$\|\hat{g} - g_0\|_{L^2([0,1]^r)} = O_p(\gamma_n \log^2 n).$$

It is well known that DPLCM (2) suffers from a severe "curse-of-dimensionality" when nonparametric smoothing methods (e.g., kernels or splines) are employed to estimate g_0. Under these classical models, accurate estimation is thus challenging even for a covariate X of moderately high dimensionality. Under our DNN-based DPLCM model, the curse-of-dimensionality is alleviated by projecting the data onto a much lower-dimensional representational space (Yarotsky (2017), Schmidt-Hieber (2017, 2020), Bauer and Kohler (2019)), where the DNN is able to accurately approximate this representational space. This is a key advantage of utilizing a DNN estimator rather than traditional smoothing methods.

Under the assumption of the representational space $\mathcal{H}(q, \alpha, \boldsymbol{d}, \tilde{\boldsymbol{d}}, M)$ in (7), Theorem 3.1 reveals that the convergence rates are jointly determined by the smoothness α and intrinsic dimension $\tilde{\boldsymbol{d}}$ of the function g_0, rather than the dimension $\boldsymbol{d}$. Thus, the proposed DPLCM can circumvent the curse of dimensionality and enjoys faster convergence rate when the intrinsic dimension $\tilde{\boldsymbol{d}}$ is relative low. For example, if the true function g_0 has a form in (8), then the traditional smoothing methods lead to slow convergence rate of order $n^{-1/6}$, in contract, the proposed method yields convergence rate of order $n^{-1/3} \log^2 n$. Moreover, the convergence rate of the DPLCM estimator for g_0 enjoys the one-dimensional nonparametric convergence rate, up to a polylogarithmic factor (as the estimator in Huang (1999) when the true model is PLACM).

Below, we show the minimax lower bound for estimating g_0.

THEOREM 3.2. *Let* $\Omega_0 = \{\lambda : \int_0^\tau \lambda(s)\, ds < \infty$ *and* $\lambda \ge 0\}$. *Under assumptions* (A2), (B1) *and* (B2), *there exists a constant* $0 < c < \infty$, *such that*

$$\inf_{\hat{g}} \sup_{(\theta_0, \lambda_0, g_0) \in \mathbb{R}_M^p \times \Omega_0 \times \mathcal{H}_0} \mathbb{E}\{\hat{g}_0(X) - g_0(X)\}^2 \ge c\gamma_n^2, \tag{9}$$

where the infimum is taken over all possible estimators $\hat{g}$ *based on the observed data.*

Therefore, the partial likelihood estimate in Theorem 3.1 is rate optimal because it attains the minimax lower bound (*up to a polylogarithm factor*).

Next, we establish the efficient score and information bound for estimating θ_0 (Bickel et al. (1993), van der Vaart (2000), Kosorok (2008)). Let Ω_{λ_0} be the collection of all subfamilies $\{\log \lambda_a : a \in (-1, 1)\} \subset \{\log \lambda : \lambda \in \Omega_0\}$ such that $\lim_{a \to 0} \|a^{-1}(\log \lambda_a - \log \lambda_0) -$

$h\|_{L^2([0,\tau])} = 0$ where $h \in L^2([0,\tau])$, and let

$$\mathbb{T}_{\lambda_0} = \Big\{ h \in L^2([0,\tau]) : \lim_{a\to 0} \|a^{-1}(\log\lambda_a - \log\lambda_0) - h\|_{L^2([0,\tau])} = 0 \text{ for some subfamily } \{\log\lambda_a : a \in (-1,1)\} \in \Omega_{\lambda_0} \Big\}.$$

Likewise, let $\mathcal{H}_{g_0}$ denote the collection of all subfamilies $\{g_b \in L^2([0,1]^r) : b \in (-1,1)\} \subset \mathcal{H}_0$ such that $\lim_{b\to 0} \|b^{-1}(g_b - g_0) - g\|_{L^2([0,1]^r)} \to 0$, and let

$$\mathbb{T}_{g_0} = \Big\{ g \in L^2([0,1]^r) : \lim_{b\to 0} \|b^{-1}(g_b - g_0) - g\|_{L^2([0,1]^r)} = 0 \text{ for some subfamily } \{g_b : b \in (-1,1)\} \in \mathcal{H}_{g_0} \Big\}.$$

Set $\overline{\mathbb{T}}_{\lambda_0}$ and $\overline{\mathbb{T}}_{g_0}$ be the closed linear span (the closure under linear combinations) of $\mathbb{T}_{\lambda_0}$ and $\mathbb{T}_{g_0}$, respectively.

Let $M(t) = \Delta 1(T \le t) - \int_0^t 1(T \ge s)\exp\{\theta_0^\top V + g_0(X)\}\lambda_0(s)\,ds$ be the counting process martingale associated with the model (Andersen and Gill (1982)).

THEOREM 3.3. *Under assumptions* (A2) *and* (B1)–(B3), *the efficient score for* θ_0 *is*

$$\ell^*_{\theta_0}(V,\Delta,T) = \int_0^\tau \{Z - \boldsymbol{h}_*(t) - \boldsymbol{g}_*(X)\}\,dM(t).$$

Here, the vector function $(\boldsymbol{h}_*^\top, \boldsymbol{g}_*^\top)^\top \in (\overline{\mathbb{T}}_{\lambda_0})^p \times (\overline{\mathbb{T}}_{g_0})^p$ *is the minimizer of* $\mathbb{E}\{\Delta\|Z - \boldsymbol{h}(T) - \boldsymbol{g}(X)\|_c^2\}$, *where the notation* $\|v\|_c^2 = (v_1^2,\dots,v_p^2)^\top$ *for vector* $v = (v_1,\dots,v_p)^\top$ *and the minimization operates componentwise on the vector.*

Moreover, the information bound for θ_0 *is*

$$I(\theta_0) = \mathbb{E}\{\ell^*_{\theta_0}(V,\Delta,T)\}^{\otimes 2} = \mathbb{E}[\Delta\{Z - \boldsymbol{h}_*(T) - \boldsymbol{g}_*(X)\}^{\otimes 2}].$$

The next theorem establishes the asymptotic normality of $\hat\theta$ with $\sqrt{n}$-consistency.

THEOREM 3.4. *Under assumptions* (A1), (A2) *and* (B1)–(B4). *If the information matrix* $I(\theta_0)$ *is nonsingular and* $n\gamma_n^4 \to 0$ *as* $n \to \infty$. *Then we have*

$$\sqrt{n}(\hat\theta - \theta_0) = n^{-1/2} I(\theta_0)^{-1} \sum_{i=1}^n \ell^*_{\theta_0}(V_i,\Delta_i,T_i) + o_p(1) \xrightarrow{d} N(0,\Sigma),$$

where $\Sigma = I(\theta_0)^{-1}$.

Although the nonparametric estimator $\hat g$ in Theorem 3.1 converges slower than $\sqrt{n}$, the maximum partial likelihood estimator for parameter θ_0 can still attain $\sqrt{n}$-consistency and asymptotic normality. Most impressive is that the information bound in Theorem 3.3 is attained by our estimator $\hat\theta$, so it is semiparametrically efficient. We provide an estimate of the asymptotic variance Σ in Section 4.1 and additional discussion about it in the Supplementary Material (Zhong, Mueller and Wang (2022)).

4. Numerical implementation and results.

4.1. *Computational details.* The maximization of the log partial likelihood function in (6), was implemented using *Pytorch* (Paszke et al. (2019)). Since the log partial likelihood

function is nonconvex with respect to the parameters θ, W_k and v_k, it is challenging to compute the estimator $(\hat{\theta}, \hat{g})$ via classical optimization techniques. In this work, we employ the *Adam* optimizer, which has become extremely popular in deep learning as it is practically performant and reliable across models/data sets (Kingma and Ba (2014)).

Initialization. We initialize the stochastic optimization algorithm by choosing an initial value for θ through the solution of the conventional CPH model (Cox (1972, 1975)) when treating $g(X)$ as a linear predictor. This solution is obtained from the *Python* package *lifelines* (Davidson-Pilon (2019)). As it is less clear how to select good initial values for matrices W_k and vectors v_k of the function g (Glorot and Bengio (2010), Martens (2010), Saxe, McClelland and Ganguli (2013)), we simply use *Pytorch*'s default random initialization.

Choice of hyperparameters. Implementing g_0 and θ_0 in practice requires the specification of the number of hidden layers K, number of neurons p_k in all K hidden layers, dropout rate (Srivastava et al. (2014)) and the learning rate (Goodfellow, Bengio and Courville (2016)). These tuning parameters are referred to as the "hyperparameters" in the deep learning community. For simplicity, we use the same number of neurons in every hidden layer (i.e., $p_k = p_j$ for $1 \le k,\ j \le K$). The dropout is the rate of randomly ignored neurons during training, so it only involves g and not θ.

The learning rate is defined as the step size for the gradient descent in the *Adam* algorithm (Kingma and Ba (2014)). We treat these decisions as hyperparameters and tune them via a grid search where the log partial likelihood is evaluated over a held-out validation set after each training trial. Detailed configurations of these hyperparameters are displayed in the Supplementary Material (Zhong, Mueller and Wang (2022)) for the simulations and data analysis. To mitigate overfitting, in each run we hold out 20% of the training set as the validation set, where training of the neural network is early stopped once the log partial likelihoods on the validation set stop reliably improving (Goodfellow, Bengio and Courville (2016)).

Calculation of the information bound. To perform inference for the parametric θ_0, we need to estimate the asymptotic covariance matrix $\Sigma = I(\theta_0)^{-1}$ in Theorem 3.4. Recall that the information bound $I(\theta_0) = \mathbb{E}[\Delta\{Z - \boldsymbol{h}_*(T) - \boldsymbol{g}_*(X)\}^{\otimes 2}]$ with minimizer $\boldsymbol{h}_*$ and $\boldsymbol{g}_*$ defined in Theorem 3.3. In practice, we estimate $(\boldsymbol{h}_*, \boldsymbol{g}_*)$ via minimizing empirical objective function

$$(\hat{\boldsymbol{h}}_*, \hat{\boldsymbol{g}}_*) = \mathop{\arg\min}_{(\boldsymbol{h}_*, \boldsymbol{g}_*)} \frac{1}{n} \sum_{i=1}^{n} \Delta_i \| Z_i - \boldsymbol{h}_*(T_i) - \boldsymbol{g}_*(X_i) \|_c^2.$$

However, it is difficult to get the convincing solution by using the classical nonparametric methods (e.g., kernel regression and spline regression) due to the high dimensionality of function $\boldsymbol{g}_*$. Hence, we employ a DNN to approach $(\hat{\boldsymbol{h}}_*, \hat{\boldsymbol{g}}_*)$. The inputs and outputs of this DNN are (T, X) and $\boldsymbol{h}_*(T) + \boldsymbol{g}_*(X)$, respectively. The implementation details of these networks are similar to the network we use to maximize log partial likelihoods (6) described above. Subsequently, with the resulting estimate $(\hat{\boldsymbol{h}}_*, \hat{\boldsymbol{g}}_*)$, we can estimate the information bound via

$$\hat{I}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \Delta_i \{ Z_i - \hat{\boldsymbol{h}}_*(T_i) - \hat{\boldsymbol{g}}_*(X_i) \}^{\otimes 2}.$$

4.2. *Simulation study.* We carry out simulation studies to illustrate the finite sample performance of the proposed DPLCM method. Here, we also provide numerical comparisons with the Cox proportional hazards model [CPH, Cox (1972, 1975)] and partially linear additive Cox model [PLACM, Huang (1999)].

For all simulations, covariates X with dimension $d = 5$ are generated from a Gaussian copula on $[0, 2]$ with correlation parameter 0.5. Each coordinate of X is marginally distributed according to the continuous uniform distribution on $[0, 2]$. The covariate Z is set to be Bernoulli ($p = 0.5$) distributed or normally distributed as $N(0.5, 0.5)$. The treatment effect is fixed at $\theta = 1.0$. Given the covariates (Z, X), the survival time U is generated based on the hazard function:

$$\lambda(t) = \lambda_0(t) \exp\{\theta_0 Z + g_0(X)\},$$

where the baseline λ_0 is chosen to be a linear function $0.1t$, under which the event times follow a Weibull distribution. We consider four different cases for the underlying function $g_0(x)$, where $x \in [0, 2]^5$ for each case:

$$\begin{aligned}
&\textit{Case 1 (Linear)}: && g_0(x) = x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 - 15.5,\\
&\textit{Case 2 (Additive)}: && g_0(x) = x_1^2 + 2x_2^2 + x_3^3 + \sqrt{x_4 + 1} + \log(x_5 + 1) - 8.6,\\
&\textit{Case 3 (Deep 1)}: && g_0(x) = x_1^2 x_2^3 + \log(x_3 + 1) + \sqrt{x_4 x_5 + 1} + \exp(x_5/2) - 8.2,\\
&\textit{Case 4 (Deep 2)}: && g_0(x) = \{x_1^2 x_2^3 + \log(x_3 + 1) + \sqrt{x_4 x_5 + 1} + \exp(x_5/2)\}^2/20 - 6.0.
\end{aligned}$$

We add various intercept terms 15.5, 8.6, 8.2 and 6.0 to g_0 to ensure all four cases satisfy $\mathbb{E}\{g_0(X)\} = 0$. The first two cases, where g_0 is linear or additive, are designed to evaluate the performance in simple settings. Note that Case 1 and Case 2 satisfy the settings of the CPH model and PLACM model, respectively. To study the performance in more complex general DPLCM settings, we consider: Case 3, which represents a compositional highly-nonlinear underlying function, and Case 4, which further increases the underlying function's compositionality by composing the g_0 from Case 3 inside of an extra squared function. Specifically, g_0 in Case 3 can be expressed as

$$g_0(x) = h_{11}\big(h_{01}(x_1, x_2), h_{02}(x_3), h_{03}(x_4, x_5), h_{04}(x_5)\big), \tag{10}$$

where $h_{01}(x, y) = x^2 y^3$, $h_{02}(x) = \log(x + 1)$, $h_{03}(x, y) = \sqrt{xy + 1}$, $h_{04}(x) = \exp(x/2)$, and $h_{11}(x, y, z, w) = x + y + z + w - 8.2$. And g_0 in Case 4 is

$$g_0(x) = h_{21}\big(h_{11}\big(h_{01}(x_1, x_2), h_{02}(x_3), h_{03}(x_4, x_5), h_{04}(x_5)\big)\big),$$

where h_{01}, h_{02}, h_{03}, h_{04}, h_{11} are the same as in (10) but $h_{21}(x) = (x^2 + 16.4x + 8.2^2)/20 - 6.0$. To simulate right censoring, censoring times C are independently generated from an exponential distribution with parameter μ. We choose μ to control the overall censoring rate, using two different values in each simulation case, which roughly produce 40% or 60% censoring (for Case 1: $\mu = 18$ or 2.5, for Case 2: $\mu = 18$ or 5, for Case 3: $\mu = 28$ or 10, for Case 4: $\mu = 45$ or 18).

In each simulation case, we perform $Q = 200$ simulation runs with sample sizes $n = 500, 1000$ or 2000. With $T_i = \min\{U_i, C_i\}$ and $\Delta_i = 1(U_i \le C_i)$, our models are fit to observations of the form: $\{(Z_i, X_i, T_i, \Delta_i) : i = 1, \ldots, n\}$. We use 64% of the simulated data in each simulation run to compute our estimates under a particular hyperparameter configuration (e.g., number of layers/neurons, dropout and learning rates) and 16% of the simulated data is reserved as validation data used to tune these hyperparameters. The remaining 20% of the simulated data are held out as test data to evaluate the resulting estimates.

We evaluate the performance of estimates $\hat{g}$ using the relative error

$$\mathrm{RE}(\hat{g}) = \left[\frac{\frac{1}{n_1}\sum_i^{n_1}\{(\hat{g}(X_i) - \bar{\hat{g}}) - g_0(X_i)\}^2}{\frac{1}{n_1}\sum_{i=1}^{n_1}\{g_0(X_i)\}^2}\right]^{1/2}, \tag{11}$$

where $\hat{g}$ and g_0 are evaluated on the covariates of the test set $\{X_i : i = 1, \ldots, n_1\}$ and $\bar{\hat{g}} = \sum_{i=1}^{n_1} \hat{g}(X_i)/n_1$. We subtract the mean of $\hat{g}$ on the test set, because the solution of maximizing the log partial likelihood is only unique up to a constant.

Table 1 reports the biases and standard deviations of the estimated $\hat{\theta}$ over 200 simulation runs from each case. Results of the proposed DPLCM method under each simulation setting suggest that θ_0 can be estimated almost unbiasedly and the mean square errors decrease steadily as the sample size increases from 500 to 2000. As expected, low censoring rates lead to more precise estimates of θ_0 for both binary and continuous covariates. DPLCM greatly outperforms the CPH and PLACM methods in complex settings of Case 3 and Case 4, where the overly restrictive CPH and PLACM methods result in large biases. Under the simpler Case 1 (which meets CPH assumptions) or Case 2 (which meets PLACM assumptions), the DPLCM method remains strongly competitive with little loss of efficiency.

For each simulation run, we also estimated the information bound $I(\theta_0)$, whose inverse can be used to calculate the asymptotic variance of $\hat{\theta}$ (Theorem 3.4). We use the estimated asymptotic variance to build a 95% confidence interval for θ_0. Table 2 displays the observed coverage of these confidence intervals. Generally, the coverage rate is near 95% for our proposed DPLCM method, especially when the sample size n is large. Under Case 3 or Case 4, the poor coverage of CPH and PLACM confidence intervals indicates the uncertainty quantification produced by these methods may be unreliable in practice.

Table 3 compares the performance of all three methods in estimating the nonparametric function $\hat{g}$. Here, we report the relative error (RE) on the test data from each simulation run. When the underlying function stems from Case 1 or Case 2, the DPLCM method fares only slightly worse than the perfectly specified CPH or PLACM model. However, when the underlying function belongs to Case 3 or Case 4, DPLCM is substantially more accurate than the other methods. Note that perfect specification of CPH or PLACM is unlikely for real-world data. As the sample size n increases, the RE of the DPLCM estimator decreases substantially as guaranteed by Theorem 3.1. As expected, each method produces better estimates of θ_0 and g_0 with larger n in correctly specified settings. Between Case 3 and Case 4, all methods perform worse under the more complicated Case 4 setting, yet the performance gaps between the two cases become narrower for DPLCM as n increases while this gap remains large for CPH and PLACM.

In the Supplementary Material (Zhong, Mueller and Wang (2022)), we also evaluate the predictive performance of all three methods using the *concordance index* (Harrell et al. (1982)). We find that DPLCM produces superior concordance between the ranks of the predicted survival times and the ranks of actual survival times when compared to CPH and PLACM models. In conclusion, these simulations demonstrate the appealing performance of our DPLCM method for estimation and prediction with and without model misspecification. The empirical results qualitatively agree with our theoretical analysis from the previous section.

4.3. *Rotterdam Breast Cancer Data*. To demonstrate our proposed DPLCM method on real data, we consider the Rotterdam Breast Cancer Data (publicly available at the R package *survival*). Foekens et al. (2000) used these data to study potential factors that affect the survival time of cancer patients, defined as the days from primary surgery to the earlier of disease recurrence or death. The data consist of 2982 subjects with 57.35% censoring rate and nine baseline covariates (i.e., age, progesterone receptors, estrogen receptors, number of positive lymph nodes, menopausal status, tumor size, tumor grade, chemotherapy and hormonal treatment), among which the first four variables are continuous and the last five are discrete. Using this data, we investigate the efficacy of chemotherapy or hormonal treatments.

As in Section 4.2, we model the data with three approaches: the proposed deep partially linear Cox model (DPLCM), Cox proportional hazards (CPH) and the partially linear additive Cox model (PLACM). We report results from five-fold cross-validation, where in each fold: we hold out 20% of training data as a validation set to choose hyperparameters and subsequently compute concordance index on a separate set of test data. For this data, we entered

TABLE 1

The bias and standard deviation (in parentheses) of $\hat{\theta}$ for the DPLCM, CPH and PLACM methods. We report the results over 200 *simulated data sets for each of the four cases. In each simulation run, only* 64% *of the data were used to obtain the estimates, so the actual sample sizes are* 320, 640, 1280 *for* $n = 500, 1000, 2000$

		$Z \sim \text{Bernoulli}(p = 0.5)$						$Z \sim N(0.5, 0.5)$					
		40% censoring rate			60% censoring rate			40% censoring rate			60% censoring rate		
	n	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM
Case 1	500	−0.0489	0.0046	0.0226	0.0579	0.0374	−0.0551	0.0621	0.0444	−0.0220	0.0538	0.0378	−0.0494
		(0.1626)	(0.1592)	(0.1617)	(0.2182)	(0.2036)	(0.2137)	(0.1747)	(0.1691)	(0.1716)	(0.2210)	(0.2080)	(0.2081)
	1000	0.0238	0.0114	−0.0100	0.0295	0.0132	−0.0122	0.0261	0.0098	−0.0076	0.04528	0.0309	−0.0405
		(0.1206)	(0.1150)	(0.1166)	(0.1479)	(0.1398)	(0.1421)	(0.1232)	(0.1177)	(0.1216)	(0.1343)	(0.1289)	(0.1307)
	2000	0.0070	0.0003	−0.0051	0.0172	0.01025	−0.0017	0.0095	−0.0041	−0.0090	0.0197	0.0121	0.0131
		(0.0776)	(0.0754)	(0.0764)	(0.1021)	(0.0965)	(0.0987)	(0.0698)	(0.0680)	(0.0684)	(0.0989)	(0.0983)	(0.0972)
Case 2	500	−0.0595	−0.1938	−0.0575	−0.0158	−0.1616	−0.0726	−0.0602	−0.1862	−0.0581	−0.0703	−0.1760	−0.0639
		(0.1931)	(0.1724)	(0.1863)	(0.2218)	(0.2074)	(0.2141)	(0.1807)	(0.1483)	(0.1602)	(0.2032)	(0.1850)	(0.1953)
	1000	0.0229	−0.1927	−0.0215	0.0260	−0.1784	−0.0255	0.0233	−0.2017	−0.0259	0.0228	−0.1768	−0.0345
		(0.1150)	(0.1089)	(0.1020)	(0.1372)	(0.1347)	(0.1278)	(0.1188)	(0.1132)	(0.1161)	(0.1288)	(0.1223)	(0.1219)
	2000	0.0134	−0.1968	−0.0127	0.0143	−0.1838	−0.0047	0.0191	−0.1915	−0.0037	0.0174	−0.1753	−0.0137
		(0.0749)	(0.0759)	(0.0753)	(0.0954)	(0.0935)	(0.0953)	(0.0816)	(0.0777)	(0.0816)	(0.1023)	(0.0935)	(0.0933)
Case 3	500	−0.1046	−0.3208	−0.3285	−0.2136	−0.3798	−0.4083	−0.1146	−0.3289	−0.3408	−0.2214	−0.3941	−0.4157
		(0.1642)	(0.1591)	(0.1768)	(0.2125)	(0.2056)	(0.2275)	(0.1587)	(0.1622)	(0.1824)	(0.2018)	(0.1799)	(0.1965)
	1000	−0.0402	−0.3293	−0.3340	−0.0756	−0.3645	−0.3677	−0.0517	−0.3411	−0.3476	−0.0624	−0.3796	−0.3952
		(0.1194)	(0.1188)	(0.1397)	(0.1441)	(0.1394)	(0.1432)	(0.1253)	(0.1150)	(0.1240)	(0.1382)	(0.1300)	(0.1368)
	2000	−0.0476	−0.3435	−0.3549	−0.0490	−0.3805	−0.3802	−0.0442	−0.3490	−0.3560	−0.0596	−0.3820	−0.3824
		(0.0735)	(0.0741)	(0.0855)	(0.0977)	(0.0933)	(0.1047)	(0.0756)	(0.0755)	(0.0862)	(0.0978)	(0.0863)	(0.0942)
Case 4	500	−0.1397	−0.3121	−0.2995	−0.2831	−0.438596	−0.4404	−0.1420	−0.3244	−0.2965	−0.2707	−0.4332	−0.4168
		(0.1638)	(0.1737)	(0.2018)	(0.2059)	(0.1932)	(0.2335)	(0.1788)	(0.1828)	(0.2051)	(0.1915)	(0.1803)	(0.2136)
	1000	−0.0773	−0.3140	−0.2940	−0.1671	−0.4443	−0.4381	−0.0949	−0.3401	−0.3209	−0.1589	−0.4379	−0.4226
		(0.1182)	(0.1228)	(0.1426)	(0.1432)	(0.1360)	(0.1559)	(0.1262)	(0.1175)	(0.1348)	(0.1476)	(0.1376)	(0.1666)
	2000	−0.0732	−0.3257	−0.3247	−0.1228	−0.4294	−0.4237	−0.0677	−0.3310	−0.3320	−0.1213	−0.4250	−0.4256
		(0.0911)	(0.0771)	(0.0920)	(0.1008)	(0.0880)	(0.1024)	(0.0968)	(0.0900)	(0.1070)	(0.0999)	(0.0884)	(0.1038)

TABLE 2

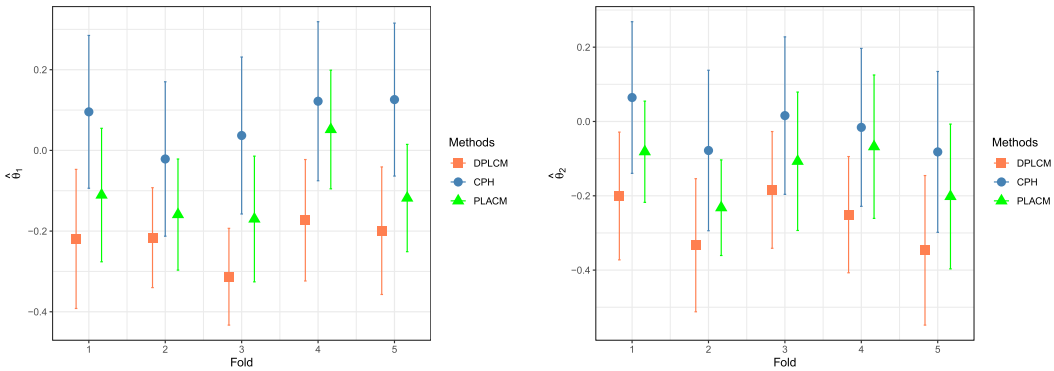
Empirical coverage probability of 95% *confidence intervals for* θ_0 *for the DPLCM, CPH and PLACM methods. We report coverage observed over* 200 *simulated data sets for each of the four cases. In each simulation run, only* 64% *of the data were used to fit models, so the actual sample sizes are* 320, 640, 1280 *for* $n = 500, 1000, 2000$

		$Z \sim \text{Bernoulli}(p = 0.5)$						$Z \sim N(0.5, 0.5)$					
		40% censoring rate			60% censoring rate			40% censoring rate			60% censoring rate		
	n	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM
Case 1	500	0.965	0.955	0.955	0.945	0.935	0.940	0.920	0.930	0.930	0.915	0.945	0.920
	1000	0.960	0.950	0.945	0.965	0.960	0.950	0.935	0.940	0.935	0.930	0.950	0.925
	2000	0.950	0.950	0.955	0.955	0.945	0.940	0.955	0.950	0.945	0.960	0.950	0.945
Case 2	500	0.935	0.725	0.955	0.940	0.830	0.935	0.930	0.790	0.935	0.945	0.855	0.950
	1000	0.935	0.555	0.950	0.955	0.710	0.945	0.935	0.520	0.945	0.955	0.725	0.945
	2000	0.950	0.220	0.950	0.960	0.450	0.950	0.955	0.295	0.955	0.965	0.500	0.950
Case 3	500	0.965	0.500	0.905	0.930	0.480	0.885	0.915	0.465	0.885	0.915	0.450	0.890
	1000	0.960	0.165	0.815	0.950	0.210	0.835	0.945	0.125	0.770	0.935	0.195	0.735
	2000	0.955	0.005	0.545	0.950	0.015	0.630	0.955	0.005	0.425	0.940	0.005	0.505
Case 4	500	0.925	0.465	0.905	0.890	0.385	0.860	0.915	0.445	0.890	0.895	0.395	0.875
	1000	0.920	0.225	0.890	0.900	0.085	0.790	0.930	0.150	0.840	0.915	0.090	0.750
	2000	0.930	0.010	0.700	0.925	0.000	0.650	0.935	0.035	0.565	0.925	0.010	0.435

TABLE 3

The relative error and standard deviation (in parentheses) of $\hat{g}$ for the DPLCM, CPH and PLACM methods. We report results over 200 *simulated data sets for each of the four cases. In each simulation run, only* 64% *of the data were used to obtain the estimates, so the actual sample sizes are* 320, 640, 1280 *for* $n = 500, 1000, 2000$

		$Z \sim \text{Bernoulli}(p = 0.5)$						$Z \sim N(0.5, 0.5)$					
		40% censoring rate			60% censoring rate			40% censoring rate			60% censoring rate		
	n	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM	DPLCM	CPH	PLACM
Case 1	500	0.1684	0.0482	0.1063	0.2432	0.0713	0.1561	0.1735	0.0521	0.1082	0.2377	0.0668	0.1552
		(0.0737)	(0.0277)	(0.0431)	(0.1072)	(0.0463)	(0.0610)	(0.0794)	(0.0338)	(0.0477)	(0.1140)	(0.0422)	(0.0626)
	1000	0.0995	0.0364	0.0632	0.1402	0.0412	0.0954	0.1009	0.0340	0.0622	0.1403	0.0428	0.0982
		(0.0391)	(0.0223)	(0.0256)	(0.0716)	(0.0272)	(0.0384)	(0.0517)	(0.0215)	(0.0250)	(0.0650)	(0.0275)	(0.0432)
	2000	0.0610	0.0247	0.0406	0.0863	0.0303	0.0642	0.0630	0.0249	0.0386	0.0841	0.0295	0.0613
		(0.0285)	(0.0141)	(0.0160)	(0.0403)	(0.0181)	(0.0251)	(0.0264)	(0.0151)	(0.0147)	(0.0409)	(0.0163)	(0.0254)
Case 2	500	0.2085	0.2314	0.1022	0.3302	0.2537	0.1344	0.2092	0.2327	0.1042	0.3222	0.2470	0.1331
		(0.0612)	(0.0160)	(0.0380)	(0.1159)	(0.0340)	(0.0476)	(0.0573)	(0.0144)	(0.0365)	(0.1024)	(0.0298)	(0.0499)
	1000	0.1610	0.2292	0.0671	0.3065	0.2426	0.0870	0.1623	0.2298	0.0671	0.2979	0.2404	0.0864
		(0.0397)	(0.0105)	(0.0258)	(0.0880)	(0.01648)	(0.0367)	(0.0373)	(0.0104)	(0.0243)	(0.0886)	(0.0180)	(0.0314)
	2000	0.1045	0.2283	0.0436	0.2109	0.2377	0.0568	0.1049	0.2278	0.0436	0.2159	0.2397	0.0577
		(0.0229)	(0.0070)	(0.0144)	(0.0609)	(0.0114)	(0.0235)	(0.0255)	(0.0072)	(0.0161)	(0.0667)	(0.0128)	(0.0223)
Case 3	500	0.2191	0.4773	0.4580	0.2874	0.4503	0.4036	0.2245	0.4795	0.4615	0.2800	0.4434	0.4055
		(0.0528)	(0.0527)	(0.0574)	(0.0952)	(0.0351)	(0.0451)	(0.0616)	(0.0448)	(0.0480)	(0.0913)	(0.0383)	(0.0477)
	1000	0.1338	0.4744	0.4458	0.1784	0.4323	0.3798	0.1363	0.4744	0.4448	0.1840	0.4319	0.3890
		(0.0250)	(0.0345)	(0.0386)	(0.0438)	(0.0234)	(0.0329)	(0.0265)	(0.0377)	(0.0402)	(0.0467)	(0.0192)	(0.0311)
	2000	0.0945	0.4729	0.4408	0.1169	0.4293	0.3761	0.0947	0.4752	0.4449	0.1203	0.4301	0.3768
		(0.0143)	(0.0261)	(0.0280)	(0.0211)	(0.0130)	(0.0226)	(0.0155)	(0.0245)	(0.0245)	(0.0224)	(0.0129)	(0.0254)
Case 4	500	0.3230	0.5108	0.4815	0.3075	0.4569	0.4322	0.31610	0.5030	0.4738	0.3134	0.4603	0.4369
		(0.0764)	(0.0773)	(0.0760)	(0.0769)	(0.0709)	(0.0752)	(0.0747)	(0.0804)	(0.0792)	(0.0748)	(0.0724)	(0.0757)
	1000	0.1705	0.4963	0.4626	0.1604	0.4565	0.4248	0.1683	0.4944	0.4610	0.1649	0.4607	0.4311
		(0.0478)	0.0560)	(0.0550)	(0.0451)	(0.0547)	(0.0592)	(0.0508)	(0.0584)	(0.0561)	(0.0434)	(0.0493)	(0.0530)
	2000	0.1188	0.4937	0.4607	0.1125	0.4544	0.4206	0.1197	0.4927	0.4589	0.1121	0.4567	0.4222
		(0.0299)	(0.0395)	(0.0386)	(0.0240)	(0.0378)	(0.0390)	(0.0308)	(0.0424)	(0.0412)	(0.0217)	(0.0377)	(0.0386)



(a) $\hat{\theta}_1$ (b) $\hat{\theta}_2$

FIG. 2. *Estimates and* 95% *confidence intervals of* θ_1 (*chemotherapy*) *and* θ_2 (*hormonal treatment*) *for five folds in Rotterdam data*.

the chemotherapy, the hormonal treatment along with the other three discrete covariates as linear predictor Z and model the remaining four continuous covariates nonparametrically (as X) for DPLCM and PLACM.

Figures 2(a) and 2(b) display the estimated θ_1 and θ_2 of the chemotherapy and hormonal treatment, respectively, as well as corresponding 95% confidence intervals across all five folds. Note that the intervals derived from DPLCM do not cover zero while all intervals derived from CPH cover zero. This shows that the proposed method is able to identify the effect of the two treatments consistent with the medical consensus that both chemotherapy and hormone treatments are effective for breast cancer (https://www.cancer.org/cancer/breast-cancer/treatment). In contrast, CPH did not detect any treatment effects and PLACM produced inconsistent results on the five-fold cross-validated data sets with three folds resulting in no treatment effects but the other two folds resulting in significant treatment effects for both treatments. Such incongruity suggests PLACM estimates are unstable, which does not help practitioners decide how to guide patient treatment.

In the Supplementary Material (Zhong, Mueller and Wang (2022)), we also evaluate how well each model is able to predict patients' survival in terms of concordance index, and we repeat this comparison on another real-world data set from the Worcester Heart Attack Study (Hosmer, Lemeshow and May (2008)). On both data sets (and many train/test folds), DPLCM produces substantially better survival predictions than CPH and PLACM, showing the proposed model better fits real-world data.

5. Discussion. With its ability to flexibly model complex characteristics of real-world data, deep learning for survival analysis has garnered considerable attention. This paper studied a DNN approach to DPLCM, which not only provides a powerful tool to remedy the curse of dimensionality with many covariates, but also allows us to easily interpret treatment effects while still providing a flexible/accurate model. Estimators of treatment effect coefficients obtained by maximizing the log partial likelihood are shown to achieve asymptotic efficiency, and our estimator of the unknown nonlinear function g_0 is rate optimal.

This paper has only investigated the common setting where covariates X are Euclidean vectors. Yet there are many other complex data types of interest for future work. For example, deep learning has shown great promise for image and text classification (Krizhevsky, Sutskever and Hinton (2012), Szegedy et al. (2015), Lee and Dernoncourt (2016)), yet a comprehensive study for survival data with treatment and image/text covariates still remains to be conducted. Image (or text) data are often represented by a convolutional (or transformer) neural networks, which compose convolutional layers with fully connected layers. These neural

architectures are different than simpler fully-connected architectures we studied in (5) and their theoretical analysis remains an important challenge. Medical images and clinician notes may provide valuable information for better modeling patient survival.

Another idea for future work is to extend the methodology to high-dimensional Z if one is interested in identifying a small number of key treatments for clinical decision making. Our DPLCM methodology could be applied to high-dimensional partially linear Cox models by appropriately regularizing θ via LASSO-type penalties or smoothly clipped absolute deviation (Du, Ma and Liang (2010), Liu et al. (2016), Wu et al. (2020)).

6. Proofs of theorems.

6.1. *Notation.* For any vector $v = (v_1, \ldots, v_p)^\top \in \mathcal{R}^p$, $\|v\| = (\sum_{i=1}^p v_i^2)^{1/2}$ and $\|v\|_\infty = \max_i |v_i|$, and for any matrix $W = (w_{ij}) \in \mathbb{R}^{m\times n}$, $\|W\|_\infty = \max_{i,j} |w_{ij}|$. For any function h, $\|h\|_\infty$ and $\|h\|_{L^2}$ are the sup-norm and L^2-norm of h, respectively, and for any vector function $\boldsymbol{h} = (h_1, \ldots, h_p)^\top$, $\|h\|_\infty = \max_i \|h_i\|_\infty$. Denote $a_n \lesssim b_n$ as $a_n \le cb_n$ for some $c > 0$ and any n. And $a_n \asymp b_n$ means $a_n \lesssim b_n$ and $b_n \lesssim a_n$.

With $\eta = (\theta, g)$ and $V = (Z, X)$, write $\xi_\eta(V) = \theta^\top Z + g(X)$. We denote the true parameter by $\eta_0 = (\theta_0, g_0)$. For any $\eta_1 = (\theta_1, g_1)$ and $\eta_2 = (\theta_2, g_2)$, define $d(\eta_1, \eta_2) = [\mathbb{E}\{\xi_{\eta_1}(V) - \xi_{\eta_2}(V)\}^2]^{1/2}$. With $Y(t) = 1(T \ge t)$, define

$$R_{0n}(t, \eta) = \frac{1}{n}\sum_{i=1}^n Y_i(t)\exp\{\xi_\eta(V_i)\}, \qquad R_0(t, \eta) = \mathbb{E}\big[Y(t)\exp\{\xi_\eta(V)\}\big],$$

and for any vector function $\boldsymbol{h}$ of $V = (Z, X)$,

$$R_{1n}(t, \eta, \boldsymbol{h}) = \frac{1}{n}\sum_{i=1}^n Y_i(t)\boldsymbol{h}(V_i)\exp\{\xi_\eta(V_i)\}, \qquad R_1(t, \eta, \boldsymbol{h}) = \mathbb{E}\big[Y(t)\boldsymbol{h}(V)\exp\{\xi_\eta(V)\}\big].$$

Then define

$$l_n(t, V, \eta) = \{\xi_\eta(V) - \log R_{0n}(t, \eta)\}1(0 \le t \le \tau),$$
$$l_0(t, V, \eta) = \{\xi_\eta(V) - \log R_0(t, \eta)\}1(0 \le t \le \tau).$$

Since the log partial likelihood $L_n(\eta)$ in (6) is $L_n(\eta) = \frac{1}{n}\sum_{i=1}^n\{\Delta_i l_n(T_i, V_i, \eta) - \Delta_i \log n\}$, and

$$\arg\max_{\eta\in\mathbb{R}^p\times\mathcal{G}} L_n(\eta) = \arg\max_{\eta\in\mathbb{R}^p\times\mathcal{G}} \frac{1}{n}\sum_{i=1}^n \Delta_i l_n(T_i, V_i, \eta),$$

with an abuse of notation, we replace below the log partial likelihood $L_n(\eta)$ by $\frac{1}{n}\sum_{i=1}^n\{\Delta_i \times l_n(T_i, V_i, \eta)\}$.

Furthermore, we denote $\mathbb{P}_n$ and $\mathbb{P}$ as the empirical and probability measure of (V_i, Δ_i, T_i) and (V, Δ, T), that is, for any function h of (V, Δ, T),

$$\mathbb{P}_n h(V, \Delta, T) = \frac{1}{n}\sum_{i=1}^n h(V_i, \Delta_i, T_i) \quad \text{and} \quad \mathbb{P}h(V, \Delta, T) = \mathbb{E}h(V, \Delta, T).$$

Therefore, we have $L_n(\eta) = \mathbb{P}_n\{\Delta l_n(T, V, \eta)\}$ and further define $L_0(\eta) = \mathbb{P}\{\Delta l_0(T, V, \eta)\}$.

6.2. *Proof of Theorem 3.1.* We first only consider the estimator $\hat{\eta}^* = (\hat{\theta}^*, \hat{g}^*)$ with $\mathbb{E}\{\xi_{\hat{\eta}^*}(V)\} = \mathbb{E}\{\xi_{\eta_0}(V)\}$ in (6). In fact, for any estimator $\hat{\eta} = (\hat{\theta}, \hat{g})$ in (6), its transformation $\hat{\eta}^* = (\hat{\theta}, \hat{g} - \mathbb{E}\{\xi_{\hat{\eta}}(V) - \xi_{\eta_0}(V)\}$ is also an estimator in (6), because $L_n(\hat{\eta}) = L_n(\bar{\eta})$ and one can check that this transformation satisfies $\mathbb{E}\{\xi_{\hat{\eta}^*}(V)\} = \mathbb{E}\{\xi_{\eta_0}(V)\}$.

We now show that $d(\hat{\eta}^*, \eta_0) \xrightarrow{p} 0$ as $n \to \infty$, and

$$d(\hat{\eta}^*, \eta_0) = O_p(\gamma_n \log^2 n).$$

For some $D > 0$, let $\mathbb{R}_D^p = \{\theta \in \mathbb{R}^p : \|\theta\|_\infty < D\}$ and $\mathcal{G}_D := \mathcal{G}(K, s, \boldsymbol{p}, D)$ set in (5). Define

$$\hat{\eta}_D^* = (\hat{\theta}_D^*, \hat{g}_D^*) = \mathop{\arg\max}_{\substack{\eta \in \mathbb{R}_D^p \times \mathcal{G}_D, \\ \mathbb{E}\{\xi_\eta(V)\} = \mathbb{E}\{\xi_{\eta_0}(V)\}}} \{L_n(\theta, g)\}. \tag{12}$$

Note that $\mathbb{P}(d(\hat{\eta}^*, \eta_0) < \infty) = 1$. If it is not true, there exist a constant $\epsilon_1 > 0$, such that $\mathbb{P}(d(\hat{\eta}^*, \eta_0) \geq c) \geq \epsilon_1$ for any $c > 0$. However, this contradicts the fact that the $\hat{\eta}^*$ is the maximizer of $L_n(\eta)$ and $\mathbb{P}(L_n(\eta) \to -\infty$ as $d(\eta, \eta_0) \to \infty$ with $\mathbb{E}\{\xi_\eta(V)\} = \mathbb{E}\{\xi_{\eta_0}(V)\}) = 1$. Thus, it suffices to show that $d(\hat{\eta}_D^*, \eta_0) \xrightarrow{p} 0$ as $n \to \infty$ for some large enough D.

Observe that

$$\begin{aligned}
&|L_n(\eta) - L_0(\eta)| \\
&\quad \leq |\mathbb{P}_n\{\Delta l_n(T, V, \eta)\} - \mathbb{P}_n\{\Delta l_0(T, V, \eta)\}| + |\mathbb{P}_n\{\Delta l_0(T, V, \eta)\} - \mathbb{P}\{\Delta l_0(T, V, \eta)\}| \\
&\quad \leq \mathbb{P}_n\{\Delta |\log R_{0n}(T, \eta) - \log R_0(T, \eta)|\} + |(\mathbb{P}_n - \mathbb{P})\{\Delta l_0(T, V, \eta)\}| \\
&\quad \lesssim \sup_{0 \leq t \leq \tau} |R_{0n}(t, \eta) - R_0(t, \eta)| + |(\mathbb{P}_n - \mathbb{P})\{\Delta l_0(T, V, \eta)\}| \\
&\quad = \sup_{0 \leq t \leq \tau} |(\mathbb{P}_n - \mathbb{P})\{Y(t) e^{\xi_\eta(V)}\}| + |(\mathbb{P}_n - \mathbb{P})\{\Delta l_0(T, V, \eta)\}|.
\end{aligned}$$

By Lemma 1, we know that $\mathcal{F}_1 = \{Y(t)e^{\xi_\eta(V)} : 0 \leq t \leq \tau, \eta \in \mathbb{R}_D^p \times \mathcal{G}_D\}$ and $\mathcal{F}_2 = \{\Delta l_0(T, V, \eta) : \eta \in \mathbb{R}_D^p \times \mathcal{G}_D\}$ are *P-Glivenko–Cantelli*, then it follows:

$$\sup_{\eta \in \mathbb{R}_D^p \times \mathcal{G}_D} |L_n(\eta) - L_0(\eta)| \xrightarrow{p} 0. \tag{13}$$

Define

$$\tilde{g}_1 = \mathop{\arg\min}_{g \in \mathcal{G}(K, s, \boldsymbol{p}, D/2)} \|g - g_0\|_{L^2}.$$

By the proof of Theorem 1 in Schmidt-Hieber (2020), we know $\|\tilde{g}_1 - g_0\|_{L^2} = O(\gamma_n \log^2 n)$. Let $\tilde{g} = \tilde{g}_1 - \mathbb{E}\{\tilde{g}_1(X)\}$. Then $\tilde{g} \in \mathcal{G}_D$ and

$$\begin{aligned}
\|\tilde{g} - g_0\|_{L^2} &= \|\tilde{g}_1 - g_0 - \mathbb{E}\{\tilde{g}_1(X) - g_0(X)\}\|_{L^2} \\
&\lesssim \|\tilde{g}_1 - g_0\|_{L^2} = O(\gamma_n \log^2 n).
\end{aligned} \tag{14}$$

Then, by (13), Lemma 2 and the law of large numbers, we have

$$\begin{aligned}
|L_n(\theta_0, \tilde{g}) - L_n(\theta_0, g_0)| &\leq |L_n(\theta_0, \tilde{g}) - L_0(\theta_0, \tilde{g})| + |L_0(\theta_0, \tilde{g}) - L_0(\theta_0, g_0)| \\
&\quad + |L_0(\theta_0, g_0) - L_n(\theta_0, g_0)| \\
&= o_p(1).
\end{aligned}$$

Since $\hat{\eta}_D^*$ is the maximizer of (12), we have $L_n(\hat{\theta}_D^*, \hat{g}_D^*) \geq L_n(\theta_0, \tilde{g}) = L_n(\theta_0, g_0) - o_p(1)$, which gives

$$L_n(\hat{\eta}_D^*) \geq L_n(\eta_0) - o_p(1). \tag{15}$$

Moreover, Lemma 2 implies that, for any small $\epsilon > 0$,

$$\sup_{\substack{d(\eta,\eta_0)\geq\epsilon,\\ \mathbb{E}\{\xi_\eta(V)\}=\mathbb{E}\{\xi_\eta(V)\}}} L_0(\eta) < L_0(\eta_0). \tag{16}$$

Therefore, the conditions of Theorem 5.7 in van der Vaart (2000) follows from (13), (15) and (16), and this implies that $d(\hat{\eta}_D^*, \eta_0) \to 0$ as $n \to \infty$.

Next, we show the convergence rates $d(\hat{\eta}_D^*, \eta_0) = O_p(\gamma_n \log^2 n)$. Let $\mathcal{A}_\delta = \{\eta = (\theta, g) \in \mathbb{R}_D^p \times \mathcal{G}_D : \delta/2 \leq d(\eta, \eta_0) \leq \delta\}$. We first need to show that

$$\mathbb{E}^* \sup_{\eta\in\mathcal{A}_\delta} \sqrt{n}\big|(L_n - L_0)(\eta) - (L_n - L_0)(\eta_0)\big| \lesssim \phi_n(\delta), \tag{17}$$

where $\mathbb{E}^*$ is the outer measure and $\phi_n(\delta) = \delta\sqrt{s \log \frac{U}{\delta}} + \frac{s}{\sqrt{n}} \log \frac{U}{\delta}$ with $U = K \prod_{k=0}^K (p_k + 1) \sum_{k=0}^K p_k p_{k+1}$. By calculation,

$$\begin{aligned}(L_n - L_0)(\eta) - (L_n - L_0)(\eta_0) &= (\mathbb{P}_n - \mathbb{P})\big\{\Delta l_0(T, V, \eta) - \Delta l_0(T, V, \eta_0)\big\} \\ &\quad + \mathbb{P}_n\left\{\Delta \log \frac{R_0(T, \eta)}{R_0(T, \eta_0)} - \Delta \log \frac{R_{0n}(T, \eta)}{R_{0n}(T, \eta_0)}\right\} \\ &\triangleq \mathrm{I} + \mathrm{II}.\end{aligned}$$

By Lemma 3, we obtain

$$\sup_{\eta\in\mathcal{A}_\delta} |\mathrm{I}| = O\big(n^{-1/2}\phi_n(\delta)\big). \tag{18}$$

For the second term II, we have

$$\begin{aligned}\sup_{\eta\in\mathcal{A}_\delta} |\mathrm{II}| &\leq \sup_{\eta\in\mathcal{A}_\delta, t\in[0,\tau]} \left|\log \frac{R_0(t, \eta)}{R_0(t, \eta_0)} - \log \frac{R_{0n}(t, \eta)}{R_{0n}(t, \eta_0)}\right| \\ &\lesssim \sup_{\eta\in\mathcal{A}_\delta, t\in[0,\tau]} \left|\frac{R_0(t, \eta)}{R_0(t, \eta_0)} - \frac{R_{0n}(t, \eta)}{R_{0n}(t, \eta_0)}\right| \\ &= \sup_{\eta\in\mathcal{A}_\delta, t\in[0,\tau]} \left|\frac{R_0(t, \eta)R_{0n}(t, \eta_0) - R_0(t, \eta_0)R_{0n}(t, \eta)}{R_0(t, \eta_0)R_{0n}(t, \eta_0)}\right|.\end{aligned}$$

The denominator $R_0(t, \eta_0)R_{0n}(t, \eta_0)$ is bounded away from zero with probability tending to one. And the numerator has

$$\begin{aligned}&R_0(t, \eta)R_{0n}(t, \eta_0) - R_0(t, \eta_0)R_{0n}(t, \eta) \\ &\quad = \big\{R_{0n}(t, \eta_0) - R_0(t, \eta_0)\big\}\big\{R_0(t, \eta) - R_0(t, \eta_0)\big\} \\ &\qquad - R_0(t, \eta_0)\big\{R_{0n}(t, \eta) - R_{0n}(t, \eta_0) - R_0(t, \eta) + R_0(t, \eta_0)\big\}.\end{aligned} \tag{19}$$

For the first term of the right-hand side in (19), it follows from the central limit theorem that $R_{0n}(t, \eta_0) - R_0(t, \eta_0) = O_p(n^{-1/2})$, and

$$\begin{aligned}\big|R_0(t, \eta) - R_0(t, \eta_0)\big| &\leq \mathbb{E}\big|e^{\xi_\eta(V)} - e^{\xi_{\eta_0}(V)}\big| \\ &\lesssim \big[\mathbb{E}\big\{\xi_\eta(V) - \xi_{\eta_0}(V)\big\}^2\big]^{1/2} \\ &= d(\eta, \eta_0).\end{aligned}$$

For the second term, $R_0(t, \eta_0) = O(1)$ and

$$\begin{aligned}&R_{0n}(t, \eta) - R_{0n}(t, \eta_0) - R_0(t, \eta) + R_0(t, \eta_0) \\ &\quad = (\mathbb{P}_n - \mathbb{P})\big[Y(t)\big\{e^{\xi_\eta(V)} - e^{\xi_{\eta_0}(V)}\big\}\big] \\ &\quad \triangleq \mathrm{III}.\end{aligned}$$

Lemma 3 implies that

$$\sup_{\eta\in\mathcal{A}_\delta}|\mathrm{III}| = O\big(n^{-1/2}\phi_n(\delta)\big).$$

Then

$$\sup_{\eta\in\mathcal{A}_\delta}|\mathrm{II}| \le O\big(n^{-1/2}\delta\big) + O\big(n^{-1/2}\phi_n(\delta)\big) = O\big(n^{-1/2}\phi_n(\delta)\big). \tag{20}$$

Thus, the result (17) follows from (18) and (20).

Furthermore, Lemma 2 shows that

$$\sup_{\eta\in\mathcal{A}_\delta}\mathbb{P}\big\{\Delta l_0(T,V,\eta) - \Delta l_0(T,V,\eta_0)\big\} \lesssim -\delta^2. \tag{21}$$

Denote $\tau_n = \gamma_n \log^2 n$. By assumption (A1), it is clear that

$$\tau_n^{-2}\varphi_n(\tau_n) \le \sqrt{n}.$$

On the other hand, by analogy to (17) and $\tilde{g}$ in (14), we have

$$\begin{aligned}\big|L_n(\theta_0,\tilde{g}) - L_n(\theta_0,g_0)\big| &\lesssim O_p\big(n^{-1/2}\phi_n(\tau_n)\big) + \big|L_0(\theta_0,\tilde{g}) - L_0(\theta_0,g_0)\big| \\ &\lesssim O_p\big(n^{-1/2}\phi_n(\tau_n)\big) + \|\tilde{g}-g_0\|_{L^2}^2 \\ &\le O_p(\tau_n^2).\end{aligned}$$

Thus, by the definition of $\hat{\eta}_D^* = (\hat{\theta}_D^*, \hat{g}_D^*)$ in (12),

$$L_n\big(\hat{\theta}_D^*, \hat{g}_D^*\big) \ge L_n(\theta_0,\tilde{g}) \ge L_n(\theta_0,g_0) - O_p\big(\tau_n^2\big).$$

By Theorem 3.4.1 in van der Vaart and Wellner (1996), we have

$$d\big(\hat{\eta}_D^*, \eta_0\big) = O_p(\tau_n).$$

This gives $d(\hat{\eta}^*,\eta_0) = O_p(\tau_n)$.

Furthermore, we have

$$\begin{aligned}d^2\big(\hat{\eta}^*,\eta_0\big) &= \mathbb{E}\big[\big(\hat{\theta}^*-\theta_0\big)^\top\{Z-\mathbb{E}(Z|X)\} + \big(\hat{\theta}^*-\theta_0\big)^\top\mathbb{E}(Z|X) + \{\hat{g}^*(X)-g_0(X)\}\big]^2 \\ &= \mathbb{E}\big[\big(\hat{\theta}^*-\theta_0\big)^\top\{Z-\mathbb{E}(Z|X)\}\big]^2 + \mathbb{E}\big[\{\hat{g}^*(X)-g_0(X)\} + \big(\hat{\theta}^*-\theta_0\big)^\top\mathbb{E}(Z|X)\big]^2.\end{aligned}$$

Thus, by assumptions (A2), (B1) and (B2), it follows $\|\hat{\theta}^*-\theta_0\| = O_p(\tau_n)$ and

$$\big\|\hat{g}^*-g_0\big\|_{L^2} = O_p(\tau_n).$$

Let $\hat{g} = \hat{g}^* - \mathbb{E}\{\hat{g}^*(X)\}$, then $\mathbb{E}\{\hat{g}(X)\} = 0$ and

$$O\big(\tau_n^2\big) = \mathbb{E}\big\{\hat{g}^*(X)-g_0(X)\big\}^2 = \mathbb{E}\big\{\hat{g}(X)-g_0(X)\big\}^2 + \big[\mathbb{E}\big\{\hat{g}^*(X)\big\}\big]^2 \ge \mathbb{E}\big\{\hat{g}(X)-g_0(X)\big\}^2.$$

This implies the result.

6.3. *Proof of Theorem 3.2.* Let $P_{(\theta_0,\lambda_0,g_0)}$ be the probability distribution with respect to the parameter θ_0, baseline hazard function λ_0 and nonparametric function g_0. Denote $\mathcal{P}_0 = \{P_{(\theta_0,\lambda_0,g_0)} : \theta_0 \in \mathbb{R}_M^p, \lambda_0 \in \Omega_0 \text{ and } g_0 \in \mathcal{H}_0\}$ and $\mathcal{P}_1 = \{P_{(\theta_0,\lambda_0,g_0)} : \theta_0 \in \mathbb{R}_M^p, \lambda_0 \in \Omega_1 \text{ and } g_0 \in \mathcal{H}_1\}$, where $\Omega_1 = \{\lambda : \int_0^\tau \lambda(s)\,ds = 1 \text{ and } \lambda \ge 0\}$ and $\mathcal{H}_1 = \mathcal{H}(q,\alpha,\boldsymbol{d},\tilde{\boldsymbol{d}},M/2)$. For any $(\theta,\lambda_1,g_1) \in \mathbb{R}_M^p \times \Omega_1 \times \mathcal{H}_1$, we know that $P_{(\theta,\lambda_1,g_1)} \overset{d}{=} P_{(\theta,\lambda_1\exp(c),g_1-c)}$ and $P_{(\theta,\lambda_1\exp(c),g_1-c)} \in \mathcal{P}_0$, where $c = \mathbb{E}\{g_1(X)\}$ and $P \overset{d}{=} Q$ means P and Q have the same probability measure. That is, $\mathcal{P}_1$ is a subset of $\mathcal{P}_0$. Furthermore, if $\hat{g}_1$ is an estimator of $g_1 \in \mathcal{H}_1$ based on the observed data $\{(T_i,\Delta_i,V_i), i = 1,\dots,n\}$ under some model $P_{(\theta,\lambda_1,g_1)} \in \mathcal{P}_1$,

then $\hat{g}_0 := \hat{g}_1 - c$ with $c = \mathbb{E}\{g_1(X)\}$ is also an estimator of $g_0 := g_1 - c$ based on same copies of the observed data $\{(T_i, \Delta_i, V_i), i = 1, \ldots, n\}$ under $P_{(\theta,\lambda_1 \exp(c),g_0)}$ ($\stackrel{d}{=} P_{(\theta,\lambda_1,g_1)}$) $\in \mathcal{P}_0$. It is easy to see that $\hat{g}_1 - g_1 = \hat{g}_0 - g_0$, hence

$$\begin{aligned}(22)\qquad &\inf_{\hat{g}} \sup_{(\theta_0,\lambda_0,g_0)\in\mathbb{R}_M^p\times\Omega_0\times\mathcal{H}_0} \mathbb{E}_{P_{(\theta_0,\lambda_0,g_0)}}\{\hat{g}_0(X) - g_0(X)\}^2 \\ &\quad\geq \inf_{\hat{g}_1} \sup_{(\theta_1,\lambda_1,g_1)\in\mathbb{R}_M^p\times\Omega_1\times\mathcal{H}_1} \mathbb{E}_{P_{(\theta_1,\lambda_1,g_1)}}\{\hat{g}_1(X) - g_1(X)\}^2,\end{aligned}$$

where $\mathbb{E}_P$ is the expectation under the distribution P and the infimum is taken over all possible estimators $\hat{g}$ and $\hat{g}_1$ based on the observed data under the probabilities in $\mathcal{P}_0$ and $\mathcal{P}_1$, respectively.

Next, we find a lower bound for the right-hand side of (22), which is also a lower bound for the left-hand side of (22).

For $(\theta_0, \lambda_0) \in \mathbb{R}_M^p \times \Omega_1$ and $g^{(0)}, g^{(1)} \in \mathcal{H}_1$, let P_0 and P_1 be the joint probability distribution of the observed data $\{(T_i, \Delta_i, V_i), i = 1, \ldots, n\}$ under $P_{(\theta_0,\lambda_0,g^{(0)})}$ and $P_{(\theta_0,\lambda_0,g^{(1)})}$, respectively. By Lemma 4, there exist a constant $c > 0$, such that

$$(23)\qquad \mathrm{KL}(P_1, P_0) \leq cn\|g^{(1)} - g^{(0)}\|_{L^2}^2,$$

where $\mathrm{KL}(\cdot,\cdot)$ is the Kullback–Leibler distance between P_1 and P_0.

By the proof of Theorem 3 in Schmidt-Hieber (2017), there exist $g^{(0)}, \ldots, g^{(N)} \in \mathcal{H}_1$ and constant $c_1, c_2 > 0$, such that

$$(24)\qquad \|g^{(j)} - g^{(k)}\|_{L^2} \geq 2c_1\gamma_n > 0 \quad \text{and} \quad \frac{cn}{N}\sum_{j=1}^{N}\|g^{(j)} - g^{(0)}\|_{L^2}^2 \leq c_2 \log N.$$

Then with (23) and (24), Theorem 2.5 in Tsybakov (2009) implies that

$$\inf_{\hat{g}_1}\sup_{g_1\in\mathcal{H}_1} \mathbb{P}\big(\|\hat{g}_1 - g_1\|_{L^2} \geq c_1\gamma_n\big) \geq \frac{\sqrt{N}}{1+\sqrt{N}}\left(1 - 2c_2 - \sqrt{\frac{2c_2}{\log N}}\right).$$

This shows that

$$\inf_{\hat{g}_1} \sup_{(\theta_1,\lambda_1,g_1)\in\mathbb{R}_M^p\times\Omega_1\times\mathcal{H}_1} \mathbb{E}_{P_{(\theta_1,\lambda_1,g_1)}}\{\hat{g}_1(X) - g_1(X)\}^2 \geq c_3\gamma_n^2,$$

for some constant $0 < c_3 < \infty$.

Therefore, the proof is completed.

6.4. *Proof of Theorem 3.3.* For any $\theta \in \mathbb{R}^p$, the subfamily $\{\log\lambda_s : s \in (-1, 1)\} \in \Omega_{\lambda_0}$ and the subfamily $\{g_s : s \in (-1, 1)\} \in \mathcal{H}_{g_0}$, we consider a one-dimensional submodel $\{P_{(\theta_0+s\theta,\lambda_s,g_s)} : s \in (-1, 1)\}$. By definition of the subfamilies $\{\log\lambda_s : s \in (-1, 1)\}$ and $\{g_s : s \in (-1, 1)\}$, there exist $h \in \overline{\mathbb{T}}_{\lambda_0}$ and $g \in \overline{\mathbb{T}}_{g_0}$, such that

$$\left.\frac{\partial \log\lambda_s}{\partial s}\right|_{s=0} = h \quad \text{and} \quad \left.\frac{\partial g_s}{\partial s}\right|_{s=0} = g.$$

Note that the log likelihood for a single observation (Z, X, Δ, T) is

$$\ell(\theta,\lambda,g) = \Delta\{\log\lambda(T) + Z^\top\theta + g(X)\} - \Lambda(T)\exp\{Z^\top\theta + g(X)\},$$

where $\Lambda(t) = \int_0^t \lambda(u)\,du$. Then taking derivative of the likelihood $\ell(\theta_0 + s\theta, \lambda_s, g_s)$ with respect to s at $s = 0$, we have

$$\left.\frac{d\ell(\theta_0 + s\theta, \lambda_s, g_s)}{ds}\right|_{s=0} = \theta^\top\dot{\ell}_{\theta_0} + \dot{\ell}_{\lambda_0,h} + \dot{\ell}_{g_0,g},$$

where

$$\dot{\ell}_{\theta_0} = \Delta Z - \int_0^\infty ZY(t)\lambda_0(t)e^{\xi_{\eta_0}(V)}\,dt = \int_0^\infty Z\,dM(t),$$

$$\dot{\ell}_{\lambda_0,h} = \Delta h(T) - \int_0^\infty h(t)Y(t)\lambda_0(t)e^{\xi_{\eta_0}(V)}\,dt = \int_0^\infty h(t)\,dM(t),$$

$$\dot{\ell}_{g_0,g} = \Delta g(X) - \int_0^\infty g(X)Y(t)\lambda_0(t)e^{\xi_{\eta_0}(V)}\,dt = \int_0^\infty g(X)\,dM(t)$$

are the score vector and functions corresponding to θ_0, λ_0 and g_0, respectively. The efficient score function [see Chapter 3 of Kosorok (2008)] for θ_0 is

$$\ell^*_{\theta_0}(V,\Delta,T) := \dot{\ell}_{\theta_0} - \Pi_{\lambda_0,g_0}(\dot{\ell}_{\theta_0}|\dot{\mathbf{P}}_1 + \dot{\mathbf{P}}_2),$$

where $\Pi_{\lambda_0,g_0}(\dot{\ell}_{\theta_0}|\dot{\mathbf{P}}_1 + \dot{\mathbf{P}}_2)$ is the projection of $\dot{\ell}_{\theta_0}$ onto the sumspace $\dot{\mathbf{P}}_1 + \dot{\mathbf{P}}_2$ with $\dot{\mathbf{P}}_1 := \{\dot{\ell}_{\lambda_0,h} : h \in \overline{\mathbb{T}}_{\lambda_0}\}$ and $\dot{\mathbf{P}}_2 := \{\dot{\ell}_{g_0,g} : g \in \overline{\mathbb{T}}_{g_0}\}$. Finding $\Pi_{\lambda_0,g_0}(\dot{\ell}_{\theta_0}|\dot{\mathbf{P}}_1 + \dot{\mathbf{P}}_2)$ is equivalent to finding the vector function $(\boldsymbol{h}_*^\top, \boldsymbol{g}_*^\top)^\top \in (\overline{\mathbb{T}}_{\lambda_0})^p \times (\overline{\mathbb{T}}_{g_0})^r$ such that

$$\begin{aligned}
&\mathbb{E}\{(\dot{\ell}_{\theta_0} - \dot{\ell}_{\lambda_0,\boldsymbol{h}_*} - \dot{\ell}_{g_0,\boldsymbol{g}_*})\dot{\ell}_{\lambda_0,h}\} = 0 \quad \text{for all } h \in \overline{\mathbb{T}}_{\lambda_0},\\
&\mathbb{E}\{(\dot{\ell}_{\theta_0} - \dot{\ell}_{\lambda_0,\boldsymbol{h}_*} - \dot{\ell}_{g_0,\boldsymbol{g}_*})\dot{\ell}_{g_0,g}\} = 0 \quad \text{for all } g \in \overline{\mathbb{T}}_{g_0}.
\end{aligned} \tag{25}$$

Then $\Pi_{\lambda_0,g_0}(\dot{\ell}_{\theta_0}|\dot{\mathbf{P}}_1 + \dot{\mathbf{P}}_2) = \dot{\ell}_{\lambda_0,\boldsymbol{h}_*} + \dot{\ell}_{g_0,\boldsymbol{g}_*}$. By Lemma 1 in Sasieni (1992a), (25) is equivalent to

$$\begin{aligned}
&\mathbb{E}\{\Delta(Z - \boldsymbol{h}_* - \boldsymbol{g}_*)h\} = 0 \quad \text{for all } h \in \overline{\mathbb{T}}_{\lambda_0},\\
&\mathbb{E}\{\Delta(Z - \boldsymbol{h}_* - \boldsymbol{g}_*)g\} = 0 \quad \text{for all } g \in \overline{\mathbb{T}}_{g_0}.
\end{aligned}$$

This implies that $(\boldsymbol{h}_*^\top, \boldsymbol{g}_*^\top)^\top \in (\overline{\mathbb{T}}_{\lambda_0})^p \times (\overline{\mathbb{T}}_{g_0})^r$ minimizes

$$\mathbb{E}\{\Delta\|Z - \boldsymbol{h}(T) - \boldsymbol{g}(X)\|_c^2\}.$$

By assumptions (A2), (B2) and (B3), Lemma 1 in Stone (1985), and Appendix A.4 in Bickel et al. (1993), we know that the minimizer $(\boldsymbol{h}_*^\top, \boldsymbol{g}_*^\top)^\top$ is well defined. Therefore, the efficient score is

$$\ell^*_{\theta_0}(V,\Delta,T) := \dot{\ell}_{\theta_0} - \dot{\ell}_{\lambda_0,\boldsymbol{h}_*} - \dot{\ell}_{g_0,\boldsymbol{g}_*} = \int_0^\tau \{Z - \boldsymbol{h}_*(t) - \boldsymbol{g}_*(X)\}\,dM(t),$$

and the information matrix is

$$I(\theta_0) = \mathbb{E}\{\ell^*_{\theta_0}(V,\Delta,T)\}^{\otimes 2} = \mathbb{E}\{\Delta(Z - \boldsymbol{h}_*(T) - \boldsymbol{g}_*(X))^{\otimes 2}\}.$$

6.5. *Proof of Theorem 3.4.* For vector function $\boldsymbol{h}$ of $v = (z,x)$, define

$$r_n(t,\eta,\boldsymbol{h}) = \boldsymbol{h} - \frac{R_{1n}(t,\eta,\boldsymbol{h})}{R_{0n}(t,\eta)}, \qquad r(t,\eta,\boldsymbol{h}) = \boldsymbol{h} - \frac{R_1(t,\eta,\boldsymbol{h})}{R_0(t,\eta)}.$$

Taking the derivative of the partial likelihood (i.e., the partial score functions) with respect to the parameters, the partial score functions for θ and for g in some direction $\boldsymbol{g}_1$ are

$$\dot{\ell}_{n,\theta}(\theta,g) = \mathbb{P}_n\{\Delta r_n(T,\eta,\boldsymbol{I})\} \quad \text{and} \quad \dot{\ell}_{n,g}(\theta,g,\boldsymbol{g}_1) = \mathbb{P}_n\{\Delta r_n(T,\eta,\boldsymbol{g}_1)\},$$

respectively, where $\boldsymbol{I}$ is the identity map of z, that is, $\boldsymbol{I}(z) = z$. By the definition of $(\hat{\theta},\hat{g})$, we have $\dot{\ell}_{n,\theta}(\hat{\theta},\hat{g}) = 0$ and $\dot{\ell}_{n,g}(\hat{\theta},\hat{g},\boldsymbol{g}) = 0$, for all $\boldsymbol{g} \in (\overline{\mathbb{T}}_{g_0})^p$. Combining this with Lemma 5 and the definition of $\boldsymbol{g}_*$ in Theorem 3.3, we have

$$\sqrt{n}\mathbb{P}[\Delta\{r(T,\eta_0,\boldsymbol{I} - \boldsymbol{g}_*)\}^{\otimes 2}](\hat{\theta} - \theta_0) = \sqrt{n}\mathbb{P}_n\{\Delta r_n(T,\eta_0,\boldsymbol{I} - \boldsymbol{g}_*)\} + o_p(1). \tag{26}$$

Denote

$$M_i(t) = \Delta_i 1(T_i \le t) - \int_0^t Y_i(s)\exp\{\xi_{\eta_0}(V_i)\}\lambda_0(s)\,ds,$$

then

$$\sqrt{n}\mathbb{P}_n\{\Delta r_n(T,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)\} = \frac{1}{\sqrt{n}}\sum_{i=1}^n\int_0^\tau\left\{Z_i - \boldsymbol{g}_*(X_i) - \frac{R_{1n}(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_{0n}(t,\xi_{\eta_0})}\right\}dM_i(t).$$

It follows that

$$\begin{aligned}&\sqrt{n}\mathbb{P}_n\{\Delta r_n(T,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)\} - \frac{1}{\sqrt{n}}\sum_{i=1}^n\int_0^\tau\left\{Z_i - \boldsymbol{g}_*(X_i) - \frac{R_1(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_0(t,\eta_0)}\right\}dM_i(t)\\ &\quad= -\frac{1}{\sqrt{n}}\sum_{i=1}^n\int_0^\tau\left\{\frac{R_{1n}(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_{0n}(t,\eta_0)} - \frac{R_1(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_0(t,\eta_0)}\right\}dM_i(t).\end{aligned}$$

Lenglart's inequality (Lenglart (1977)) and

$$\frac{1}{n}\sum_{i=1}^n\int_0^\tau\left\{\frac{R_{1n}(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_{0n}(t,\eta_0)} - \frac{R_1(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_0(t,\eta_0)}\right\}^2 Y_i(t)\exp\{\xi_{\eta_0}(V_i)\}\lambda_0(t)\,dt \xrightarrow{p} 0$$

together imply

$$\begin{aligned}\sqrt{n}\mathbb{P}_n\{\Delta r_n(T,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)\} &= \frac{1}{\sqrt{n}}\sum_{i=1}^n\int_0^\tau\left\{Z_i - \boldsymbol{g}_*(X_i) - \frac{R_1(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_0(t,\eta_0)}\right\}dM_i(t)\\ &\quad+ o_p(1).\end{aligned}$$

By Lemma 2 in Sasieni (1992b) and the definition of $\boldsymbol{g}_*$ and $\boldsymbol{h}_*$ in Theorem 3.3, we obtain that

$$\frac{R_1(t,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)}{R_0(t,\eta_0)} = \mathbb{E}\{Z - \boldsymbol{g}_*(X)|T=t,\Delta=1\} = \boldsymbol{h}_*(t).$$

Hence, by the definition of the efficient score function $\ell^*_{\theta_0}$, we have

$$\sqrt{n}\mathbb{P}_n\{\Delta r(T,\eta_0,\boldsymbol{I}-\boldsymbol{g}_*)\} = \frac{1}{\sqrt{n}}\sum_{i=1}^n\ell^*_{\theta_0}(V_i,\Delta_i,T_i) + o_p(1) \xrightarrow{d} \mathcal{N}\big(0, I(\theta_0)\big).$$

This and (26) complete the proof.

APPENDIX: KEY LEMMAS

The following lemmas are used to establish Theorem 3.1–3.4 and their proofs are given in the Supplementary Material (Zhong, Mueller and Wang (2022)).

LEMMA 1. *Define* $\mathcal{F}_1 = \{Y(t)e^{\xi_\eta(V)} : \eta \in \mathbb{R}^p_D \times \mathcal{G}(K,s,\boldsymbol{p},D), t \in [0,\tau]\}$ *and* $\mathcal{F}_2 = \{\Delta l_0(T,V,\eta) : \eta \in \mathbb{R}^p_D \times \mathcal{G}(K,s,\boldsymbol{p},D)\}$. *Then for any* $D > 0$, $\mathcal{F}_1$ *and* $\mathcal{F}_2$ *are P-Glivenko–Cantelli.*

LEMMA 2. *Under assumptions* (A2) *and* (B1)–(B4), *we have*

$$L_0(\eta) - L_0(\eta_0) \asymp -d^2(\eta,\eta_0),$$

for all $\eta \in \{\eta : d(\eta,\eta_0) \le c, \mathbb{E}\{\xi_\eta(V)\} = \mathbb{E}\{\xi_{\eta_0}(V)\}\}$ *with some small* $c > 0$.

LEMMA 3. *Let* $\mathcal{B}_\delta = \{\eta = (\theta, g) \in \mathbb{R}^p \times \mathcal{G}(K, s, \boldsymbol{p}, D) : \|\theta - \theta_0\| \le \delta, \|g - g_0\|_{L^2} \le \delta\}$. *Define* $\mathbb{G}_n = \sqrt{n}(\mathbb{P}_n - \mathbb{P})$ *and*

$$\mathrm{I} = \mathbb{G}_n\{\Delta l_0(T, V, \eta) - \Delta l_0(T, V, \eta_0)\},$$
$$\mathrm{II} = \mathbb{G}_n[Y(t)\exp\{\xi_\eta(V)\} - Y(t)\exp\{\xi_{\eta_0}(V)\}],$$

then

$$\mathbb{E}^* \sup_{\eta\in\mathcal{B}_\delta} |\mathrm{I}| = O\left(\delta\sqrt{s\log\frac{U}{\delta}} + \frac{s}{\sqrt{n}}\log\frac{U}{\delta}\right),$$
$$\mathbb{E}^* \sup_{\eta\in\mathcal{B}_\delta} |\mathrm{II}| = O\left(\delta\sqrt{s\log\frac{U}{\delta}} + \frac{s}{\sqrt{n}}\log\frac{U}{\delta}\right) \quad \textit{for any } t \in [0, \tau],$$

where $\mathbb{E}^*$ *is the outer measure and* $U = K\prod_{k=0}^{K}(p_k + 1)\sum_{k=0}^{K} p_k p_{k+1}$.

LEMMA 4. *For fixed baseline hazard function* λ_0 *and parameter* θ, *let* P_1 *and* P_2 *be the joint probability distribution of the observed data* $\{(T_i, \Delta_i, V_i), i = 1, \ldots, n\}$ *from nonparametric function* $g^{(0)}$ *and* $g^{(1)}$, *respectively. Then under assumptions* (A2), (B1) *and* (B2), *we have*

$$\mathrm{KL}(P_1, P_0) \le cn\|g^{(1)} - g^{(0)}\|_{L^2}^2,$$

where $\mathrm{KL}(P_1, P_0) := \mathbb{E}_{P_1}\log\frac{P_1}{P_0}$ *is the Kullback–Leibler distance between* P_1 *and* P_0 *and* c *is a constant independent to* n.

LEMMA 5.

$$\mathbb{P}_n[\Delta\{r_n(T, \hat{\eta}, \boldsymbol{I} - \boldsymbol{g}_*) - r_n(T, \eta_0, \boldsymbol{I} - \boldsymbol{g}_*)\}]$$
$$= -\mathbb{P}[\Delta\{r(T, \eta_0, \boldsymbol{I} - \boldsymbol{g}_*)\}^{\otimes 2}](\hat{\theta} - \theta_0) + o_p(n^{-1/2}).$$

Acknowledgments. The authors are grateful to the Editor, Associate Editor and referees for their helpful comments that led to numerous improvements of the paper. The first author also thanks Professor Ying Yang at Department of Mathematical Sciences, Tsinghua University for kind support.

Funding. The first author was supported by National Science Foundation of China Grant NSFC-11931001 and Key Laboratory of Econometrics (Xiamen University), Ministry of Education.

The third author was supported by NSF Grant DMS-1914917 and NIH Grant UG3-0D023313 (ECHO Program).

SUPPLEMENTARY MATERIAL

Supplement to "Deep learning for the partially linear Cox model" (DOI: 10.1214/21-AOS2153SUPP; .pdf). This supplement contains mathematical proofs and examples to complete the main text, additional simulation results and empirical analysis of another real data set.

REFERENCES

ANDERSEN, P. K. and GILL, R. D. (1982). Cox's regression model for counting processes: A large sample study. *Ann. Statist.* **10** 1100–1120. MR0673646

ANTHONY, M. and BARTLETT, P. L. (1999). *Neural Network Learning*: *Theoretical Foundations*. Cambridge Univ. Press, Cambridge. MR1741038 https://doi.org/10.1017/CBO9780511624216

BARRON, A. R. (1993). Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Trans. Inf. Theory* **39** 930–945. MR1237720 https://doi.org/10.1109/18.256500

BARRON, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Mach. Learn.* **14** 115–133.

BAUER, B. and KOHLER, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Ann. Statist.* **47** 2261–2285. MR3953451 https://doi.org/10.1214/18-AOS1747

BICKEL, P. J., KLAASSEN, C. A. J., RITOV, Y. and WELLNER, J. A. (1993). *Efficient and Adaptive Estimation for Semiparametric Models*. *Johns Hopkins Series in the Mathematical Sciences*. Johns Hopkins Univ. Press, Baltimore, MD. MR1245941

CHAPFUWA, P., TAO, C., LI, C., PAGE, C., GOLDSTEIN, B., CARIN, L. and HENAO, R. (2018). Adversarial time-to-event modeling. In *Proceedings of the* 35*th International Conference on Machine Learning*.

CHEN, S. and ZHOU, L. (2007). Local partial likelihood estimation in proportional hazards regression. *Ann. Statist.* **35** 888–916. MR2336873 https://doi.org/10.1214/009053606000001299

CHEN, K., GUO, S., SUN, L. and WANG, J.-L. (2010). Global partial likelihood for nonparametric proportional hazards models. *J. Amer. Statist. Assoc.* **105** 750–760. MR2724858 https://doi.org/10.1198/jasa.2010.tm08636

CHING, T., ZHU, X. and GARMIRE, L. X. (2018). Cox-nnet: An artificial neural network method for prognosis prediction of high-throughput omics data. *PLoS Comput. Biol.* **14** e1006076.

COLLOBERT, R., WESTON, J., BOTTOU, L., KARLEN, M., KAVUKCUOGLU, K. and KUKSA, P. (2011). Natural language processing (almost) from scratch. *J. Mach. Learn. Res.* **12** 2493–2537.

COX, D. R. (1972). Regression models and life-tables. *J. Roy. Statist. Soc. Ser. B* **34** 187–220. MR0341758

COX, D. R. (1975). Partial likelihood. *Biometrika* **62** 269–276. MR0400509 https://doi.org/10.1093/biomet/62.2.269

COX, D. R. and OAKES, D. (1984). *Analysis of Survival Data*. *Monographs on Statistics and Applied Probability*. CRC Press, London. MR0751780 https://doi.org/10.1201/9781315137438

CYBENKO, G. (1989). Approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems* **2** 303–314. MR1015670 https://doi.org/10.1007/BF02551274

DABROWSKA, D. M. (1997). Smoothed Cox regression. *Ann. Statist.* **25** 1510–1540. MR1463563 https://doi.org/10.1214/aos/1031594730

DAVIDSON-PILON, C. (2019). lifelines: Survival analysis in Python. *J. Open Sour. Softw.* **4** 1317. https://doi.org/10.21105/joss.01317

DOU, X. and LIANG, T. (2021). Training neural networks as learning data-adaptive kernels: Provable representation and approximation benefits. *J. Amer. Statist. Assoc.* **116** 1507–1520. MR4309289 https://doi.org/10.1080/01621459.2020.1745812

DU, P., MA, S. and LIANG, H. (2010). Penalized variable selection procedure for Cox models with semiparametric relative risk. *Ann. Statist.* **38** 2092–2117. MR2676884 https://doi.org/10.1214/09-AOS780

FARABET, C., COUPRIE, C., NAJMAN, L. and LECUN, Y. (2012). Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1915–1929.

FARAGGI, D. and SIMON, R. (1995). A neural network model for survival data. *Stat. Med.* **14** 73–82. https://doi.org/10.1002/sim.4780140108

FARRELL, M. H., LIANG, T. and MISRA, S. (2021). Deep neural networks for estimation and inference. *Econometrica* **89** 181–213. MR4220387 https://doi.org/10.3982/ecta16901

FLEMING, T. R. and HARRINGTON, D. P. (1991). *Counting Processes and Survival Analysis*. *Wiley Series in Probability and Mathematical Statistics*: *Applied Probability and Statistics*. Wiley, New York. MR1100924

FOEKENS, J. A., PETERS, H. A., LOOK, M. P., PORTENGEN, H., SCHMITT, M., KRAMER, M. D., BRÜNNER, N., JÄNICKE, F., MEIJER-VAN GELDER, M. E. et al. (2000). The urokinase system of plasminogen activation and prognosis in 2780 breast cancer patients. *Cancer Res.* **60** 636–643.

GIUNCHIGLIA, E., NEMCHENKO, A. and VAN DER SCHAAR, M. (2018). RNN-SURV: A deep recurrent model for survival analysis. In *International Conference on Artificial Neural Networks* 23–32.

GLOROT, X. and BENGIO, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the* 30*th International Conference on Artificial Intelligence and Statistics* 249–256.

GOODFELLOW, I., BENGIO, Y. and COURVILLE, A. (2016). *Deep Learning*. *Adaptive Computation and Machine Learning*. MIT Press, Cambridge, MA. MR3617773

GRAVES, A., MOHAMED, A.-R. and HINTON, G. (2013). Speech recognition with deep recurrent neural networks. In 2013 *IEEE International Conference on Acoustics*, *Speech and Signal Processing* 6645–6649.

GÜHRING, I., KUTYNIOK, G. and PETERSEN, P. (2020). Error bounds for approximations with deep ReLU neural networks in $W^{s,p}$ norms. *Anal. Appl. (Singap.)* **18** 803–859. MR4131039 https://doi.org/10.1142/S0219530519410021

HAARBURGER, C., WEITZ, P., RIPPEL, O. and MERHOF, D. (2019). Image-based survival prediction for lung cancer patients using CNNS. In 2019 *IEEE* 16*th International Symposium on Biomedical Imaging* 1197–1201.

HAN, S., POOL, J., TRAN, J. and DALLY, W. (2015). Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems* 1135–1143.

HAO, J., KIM, Y., MALLAVARAPU, T., OH, J. H. and KANG, M. (2019). Interpretable deep neural network for cancer survival analysis by integrating genomic and clinical data. *BMC Med. Genom.* **12** 1–13.

HÄRDLE, W., HALL, P. and ICHIMURA, H. (1993). Optimal smoothing in single-index models. *Ann. Statist.* **21** 157–178. MR1212171 https://doi.org/10.1214/aos/1176349020

HARRELL, F. E., CALIFF, R. M., PRYOR, D. B., LEE, K. L. and ROSATI, R. A. (1982). Evaluating the yield of medical tests. *JAMA* **247** 2543–2546.

HASTIE, T. J. and TIBSHIRANI, R. J. (1990). *Generalized Additive Models. Monographs on Statistics and Applied Probability* **43**. CRC Press, London. MR1082147

HINTON, G., DENG, L., YU, D., DAHL, G. E., MOHAMED, A.-R., JAITLY, N., SENIOR, A., VANHOUCKE, V., NGUYEN, P. et al. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Process. Mag.* **29** 82–97.

HORNIK, K., STINCHCOMBE, M. and WHITE, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Netw.* **2** 359–366.

HOROWITZ, J. L. (2009). *Semiparametric and Nonparametric Methods in Econometrics. Springer Series in Statistics*. Springer, New York. MR2535631 https://doi.org/10.1007/978-0-387-92870-8

HOROWITZ, J. L. and MAMMEN, E. (2007). Rate-optimal estimation for a general class of nonparametric regression models with unknown link functions. *Ann. Statist.* **35** 2589–2619. MR2382659 https://doi.org/10.1214/009053607000000415

HOSMER, D. W., LEMESHOW, S. and MAY, S. (2008). *Applied Survival Analysis*: *Regression Modeling of Time-to-Event Data*, 2nd ed. *Wiley Series in Probability and Statistics*. Wiley Interscience, Hoboken, NJ. MR2383788 https://doi.org/10.1002/9780470258019

HUANG, J. (1999). Efficient estimation of the partly linear additive Cox model. *Ann. Statist.* **27** 1536–1563. MR1742499 https://doi.org/10.1214/aos/1017939141

ICHIMURA, H. (1993). Semiparametric least squares (SLS) and weighted SLS estimation of single-index models. *J. Econometrics* **58** 71–120. MR1230981 https://doi.org/10.1016/0304-4076(93)90114-K

IMAIZUMI, M. and FUKUMIZU, K. (2019). Deep neural networks learn non-smooth functions effectively. In *The* 22*nd International Conference on Artificial Intelligence and Statistics* 869–878.

JIANG, J. and JIANG, X. (2011). Inference for partly linear additive Cox models. *Statist. Sinica* **21** 901–921. MR2829860 https://doi.org/10.5705/ss.2011.039a

KATZMAN, J. L., SHAHAM, U., CLONINGER, A., BATES, J., JIANG, T. and KLUGER, Y. (2018). DeepSurv: Personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC Med. Res. Methodol.* **18** 24.

KINGMA, D. P. and BA, J. (2014). Adam: A method for stochastic optimization. Preprint. Available at arXiv:1412.6980.

KOOPERBERG, C., STONE, C. J. and TRUONG, Y. K. (1995). The L_2 rate of convergence for hazard regression. *Scand. J. Stat.* **22** 143–157. MR1339748

KOSOROK, M. R. (2008). *Introduction to Empirical Processes and Semiparametric Inference. Springer Series in Statistics*. Springer, New York. MR2724368 https://doi.org/10.1007/978-0-387-74978-5

KRIZHEVSKY, A., SUTSKEVER, I. and HINTON, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems* 1097–1105.

LECUN, Y., BENGIO, Y. and HINTON, G. (2015). Deep learning. *Nature* **521** 436–444.

LEE, J. Y. and DERNONCOURT, F. (2016). Sequential short-text classification with recurrent and convolutional neural networks. Preprint. Available at arXiv:1603.03827.

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*.

LENGLART, E. (1977). Relation de domination entre deux processus. *Ann. Inst. Henri Poincaré B, Calc. Probab. Stat.* **13** 171–179. MR0471069

LESHNO, M., LIN, V. Y., PINKUS, A. and SCHOCKEN, S. (1993). Multilayer feedforward networks with a nonpolynomial activation function can approximate any function. *Neural Netw.* **6** 861–867.

LI, H., BOIMEL, P., JANOPAUL-NAYLOR, J., ZHONG, H., XIAO, Y., BEN-JOSEF, E. and FAN, Y. (2019). Deep convolutional neural networks for imaging data based survival analysis of rectal cancer. In 2019 *IEEE* 16*th International Symposium on Biomedical Imaging* 846–849. IEEE, New York.

LIANG, S. and SRIKANT, R. (2016). Why deep neural networks for function approximation? Preprint. Available at arXiv:1610.04161.

LIU, J., ZHANG, R., ZHAO, W. and LV, Y. (2016). Variable selection in partially linear hazard regression for multivariate failure time data. *J. Nonparametr. Stat.* **28** 375–394. MR3488604 https://doi.org/10.1080/10485252.2016.1163355

MARTENS, J. (2010). Deep learning via Hessian-free optimization. In *International Conference on Machine Learning* **27** 735–742.

MATSUO, K., PURUSHOTHAM, S., JIANG, B., MANDELBAUM, R. S., TAKIUCHI, T., LIU, Y. and ROMAN, L. D. (2019). Survival outcome prediction in cervical cancer: Cox models vs deep-learning model. *Am. J. Obstet. Gynecol.* **220** 381.e1–381.e14.

MHASKAR, H. N. (1996). Neural networks for optimal approximation of smooth and analytic functions. *Neural Comput.* **8** 164–177.

MHASKAR, H., LIAO, Q. and POGGIO, T. (2017). When and why are deep networks better than shallow ones? In *Proceedings of the AAAI Conference on Artificial Intelligence* **31**.

NAIR, V. and HINTON, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *International Conference on Machine Learning* 807–814.

O'SULLIVAN, F. (1993). Nonparametric estimation in the Cox model. *Ann. Statist.* **21** 124–145. MR1212169 https://doi.org/10.1214/aos/1176349018

PASZKE, A., GROSS, S., MASSA, F., LERER, A., BRADBURY, J., CHANAN, G., KILLEEN, T., LIN, Z., GIMELSHEIN, N. et al. (2019). PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems* 8024–8035.

PINKUS, A. (1999). Approximation theory of the MLP model in neural networks. In *Acta Numerica*, 1999. *Acta Numer.* **8** 143–195. Cambridge Univ. Press, Cambridge. MR1819645 https://doi.org/10.1017/S0962492900002919

RAMACHANDRAN, P., ZOPH, B. and LE, Q. V. (2017). Searching for activation functions. Preprint. Available at arXiv:1710.05941.

RANGANATH, R., PEROTTE, A., ELHADAD, N. and BLEI, D. (2016). Deep survival analysis. In *Proceedings of Machine Learning Research* **56** 101–114.

REN, K., QIN, J., ZHENG, L., YANG, Z., ZHANG, W., QIU, L. and YU, Y. (2019). Deep recurrent survival analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence* **33** 4798–4805.

SARIKAYA, R., HINTON, G. E. and DEORAS, A. (2014). Application of deep belief networks for natural language understanding. *IEEE/ACM Trans. Audio Speech Lang. Process.* **22** 778–784.

SASIENI, P. (1992a). Information bounds for the conditional hazard ratio in a nested family of regression models. *J. Roy. Statist. Soc. Ser. B* **54** 617–635. MR1160487

SASIENI, P. (1992b). Nonorthogonal projections and their application to calculating the information in a partly linear Cox model. *Scand. J. Stat.* **19** 215–233. MR1183198

SAXE, A. M., MCCLELLAND, J. L. and GANGULI, S. (2013). Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. Preprint. Available at arXiv:1312.6120.

SCHMIDT-HIEBER, J. (2017). Nonparametric regression using deep neural networks with ReLU activation function. Preprint. Available at arXiv:1708.06633.

SCHMIDT-HIEBER, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *Ann. Statist.* **48** 1875–1897. MR4134774 https://doi.org/10.1214/19-AOS1875

SLEEPER, L. A. and HARRINGTON, D. P. (1990). Regression splines in the Cox model with application to covariate effects in liver disease. *J. Amer. Statist. Assoc.* **85** 941–949. https://doi.org/10.1080/01621459.1990.10474965

SRINIVAS, S., SUBRAMANYA, A. and VENKATESH BABU, R. (2017). Training sparse neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* 138–145.

SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I. and SALAKHUTDINOV, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15** 1929–1958. MR3231592

STONE, C. J. (1985). Additive regression and other nonparametric models. *Ann. Statist.* **13** 689–705. MR0790566 https://doi.org/10.1214/aos/1176349548

SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V. and RABINOVICH, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 1–9.

TELGARSKY, M. (2015). Representation benefits of deep feedforward networks. Preprint. Available at arXiv:1509.08101.

THERNEAU, T. M., GRAMBSCH, P. M. and FLEMING, T. R. (1990). Martingale-based residuals for survival models. *Biometrika* **77** 147–160. MR1049416 https://doi.org/10.1093/biomet/77.1.147

TSYBAKOV, A. B. (2009). *Introduction to Nonparametric Estimation. Springer Series in Statistics*. Springer, New York. MR2724359 https://doi.org/10.1007/b13794

UNSER, M. (2019). A representer theorem for deep neural networks. *J. Mach. Learn. Res.* **20** Paper No. 110, 30 pp. MR3990464 https://doi.org/10.1093/biostatistics/kxx066

VAN DER VAART, A. W. (2000). *Asymptotic Statistics. Cambridge Series in Statistical and Probabilistic Mathematics* **3**. Cambridge Univ. Press, Cambridge. MR1652247 https://doi.org/10.1017/CBO9780511802256

VAN DER VAART, A. W. and WELLNER, J. A. (1996). *Weak Convergence and Empirical Processes*: *With Applications to Statistics. Springer Series in Statistics*. Springer, New York. MR1385671 https://doi.org/10.1007/978-1-4757-2545-2

WU, Q., ZHAO, H., ZHU, L. and SUN, J. (2020). Variable selection for high-dimensional partly linear additive Cox model with application to Alzheimer's disease. *Stat. Med.* **39** 3120–3134. MR4151923 https://doi.org/10.1002/sim.8594

XIANG, A., LAPUERTA, P., RYUTOV, A., BUCKLEY, J. and AZEN, S. (2000). Comparison of the performance of neural network methods and Cox regression for censored survival data. *Comput. Statist. Data Anal.* **34** 243–257.

YAROTSKY, D. (2017). Error bounds for approximations with deep ReLU networks. *Neural Netw.* **94** 103–114. https://doi.org/10.1016/j.neunet.2017.07.002

YOUSEFI, S., AMROLLAHI, F., AMGAD, M., DONG, C., LEWIS, J. E., SONG, C., GUTMAN, D. A., HALANI, S. H., VEGA, J. E. V. et al. (2017). Predicting clinical outcomes from large scale cancer genomic profiles with deep survival models. *Sci. Rep.* **7** 11707. https://doi.org/10.1038/s41598-017-11817-6

ZHONG, Q., MUELLER, J. and WANG, J.-L. (2022). Supplement to "Deep learning for the partially linear Cox model." https://doi.org/10.1214/21-AOS2153SUPP