General Artificial Intelligence (1)

SAIR 2-03 Objective Driven Artificial Intelligence and Deep Survival Analysis

Momiao Xiong

Society of Artificial Intelligence Research

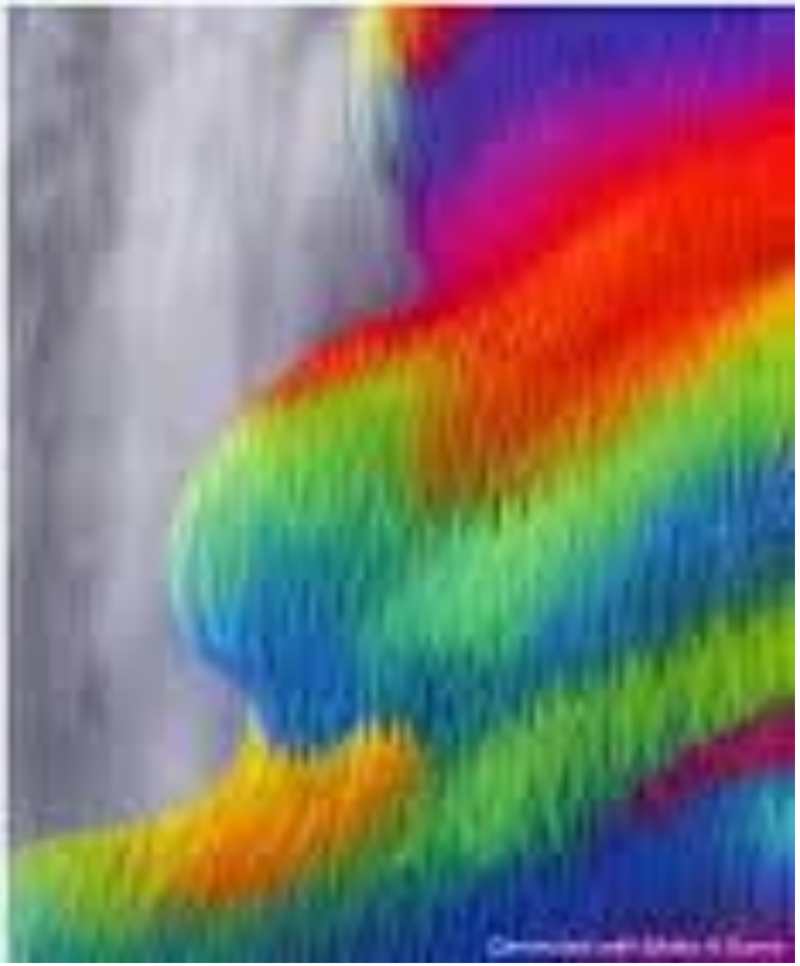
MIT

NEW YORK UNIVERSITY Meta AI

Objective-Driven AI

Towards AI systems that can learn,
remember, reason, plan,
have common sense,
yet are steerable and safe

Yann LeCun
New York University
Meta – Fundamental AI Research



MIT
2023-07-21

Objective-Driven AI

Towards AI systems that can learn,

remember, reason, plan,
have common sense,
yet are steerable and safe

MIT

Humans and animals have common sense
There behavior is driven by objectives

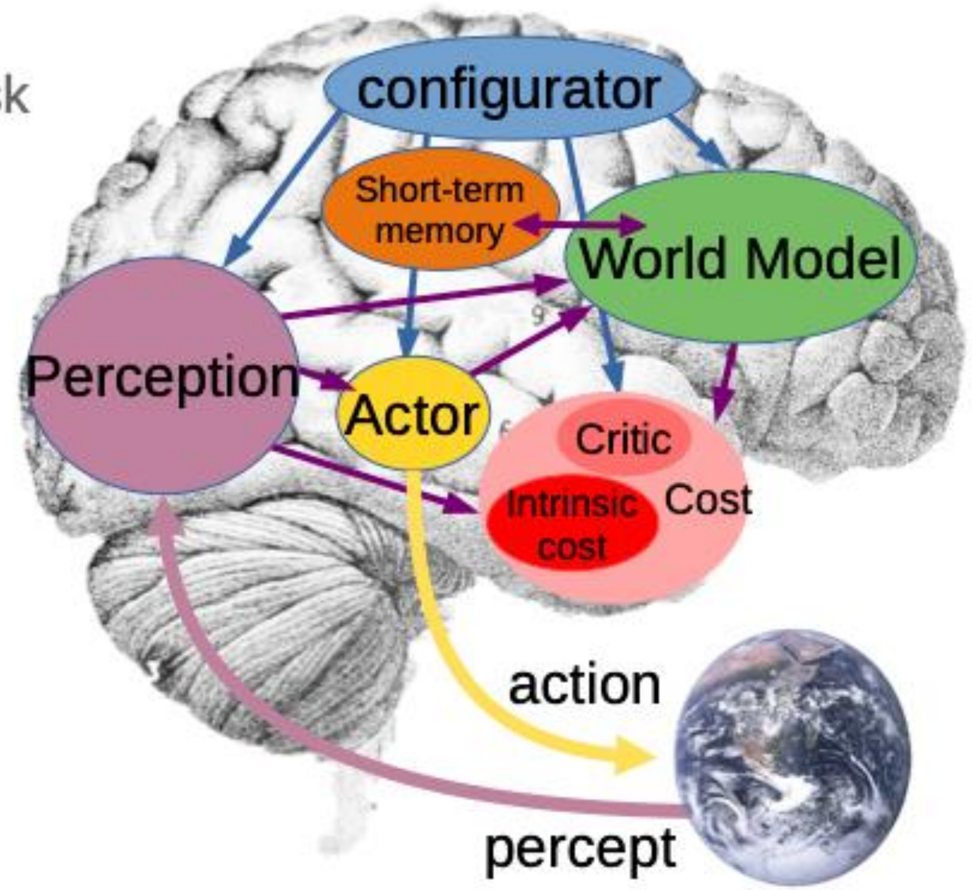
Yann LeCun
New York University
Meta – Fundamental AI Research

https://drive.google.com/file/d/1wzHohvoSgKGZvzOWqZybjm4M4veKR6t3/view

Modular Cognitive Architecture for Objective-Driven AI

Y. LeCun

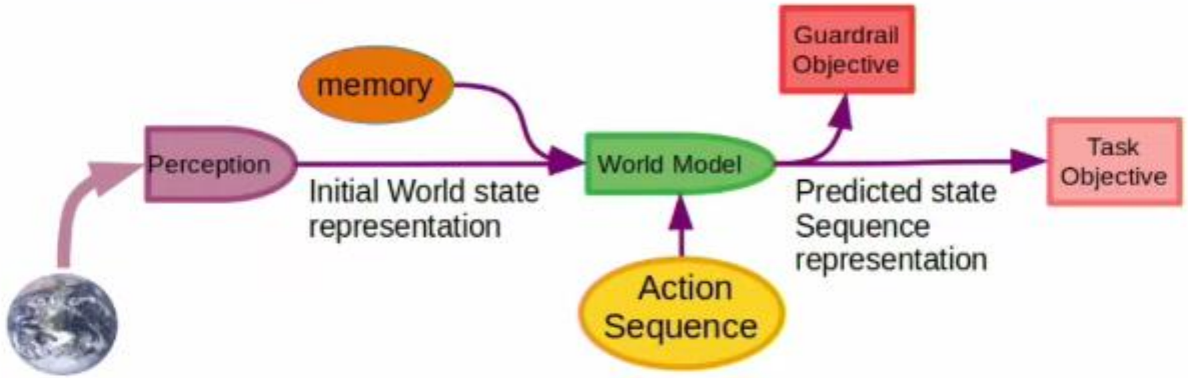
- **Configurator**
  - Configures other modules for task
- **Perception**
  - Estimates state of the world
- **World Model**
  - Predicts future world states
- **Cost**
  - Compute "discomfort"
- **Actor**
  - Find optimal action sequences
- **Short-Term Memory**
  - Stores state-cost episodes



One of Cost in human: Survival Time

Objective-Driven AI

- **Perception:** Computes an abstract representation of the state of the world
  - Possibly combined with previously-acquired information in memory
- **World Model:** Predict the state resulting from an imagined action sequence
- **Task Objective:** Measures divergence to goal
- **Guardrail Objective:** Immutable objective terms that ensure safety
- **Operation:** Finds an action sequence that minimizes the objectives



baiosphere
mcml Munich Center for Machine Learning
bidt
relAI
LMU LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN
BAdW LMU

Deep Survival Analysis

Goal: Make Our Life Longer

Deepsurv: personalized treatment recommender system using a cox proportional hazards deep neural network. BMC medical research methodology, 18(1):24, 2018.

Time to- event prediction with neural networks and cox regression. Journal of machine learning research, 20(129): 1–30, 2019.

1. Basic Concepts

1) Survival Time, Censoring Time and Their Distributions

Initially, assume that survival time T is continuous. Define $f_T(t)$ and $F_T(t) = P(T \leq t)$ be its density and cumulative distribution function, respectively. Then, the survival function of T is defined as

$$S_T(t) = P(T > t) = 1 - F_T(t) \tag{1}$$

The hazard rate is defined as

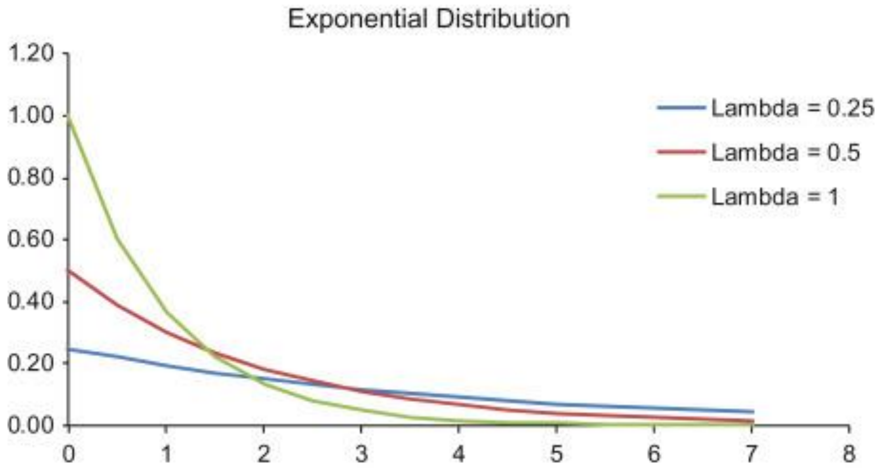
$$h_T(t) = \lim_{\Delta t \to 0} \frac{1}{\Delta t} P(t \leq T < t + \Delta t | T \geq t) = \frac{f_T(t)}{S_T(t)} \tag{2}$$

which is the instantaneous risk of the event occurring given it has not yet occurred at time t.

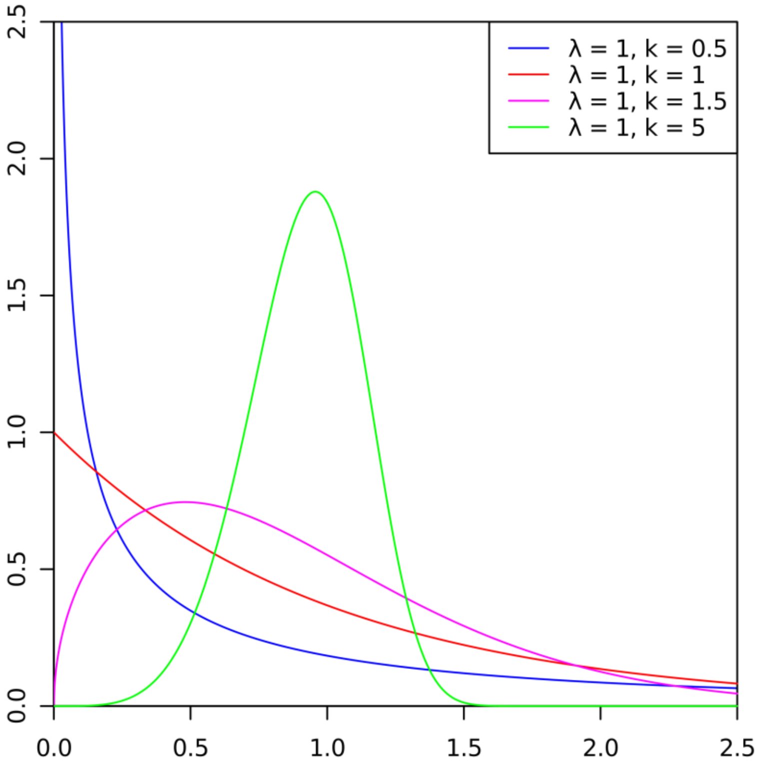
Taking derivative in equation (1) yields

$$f_T(t) = -\frac{dS_T(t)}{dt} \tag{3}$$

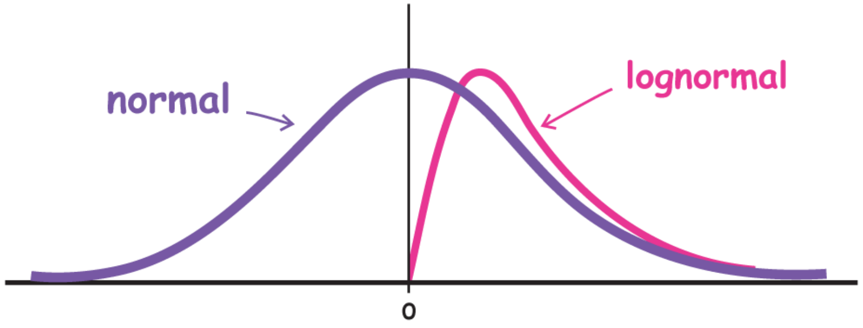
- ## Typical Distribution Examples



$$f(t,\lambda) = \begin{cases} \lambda e^{-\lambda t} & t \geq 0 \\ 0 & t < 0 \end{cases}$$



Weibull distribution



Normal pdf

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

Lognormal pdf

$$f(x) = \frac{1}{\sigma x\sqrt{2\pi}} e^{\frac{-(\ln x-\mu)^2}{2\sigma^2}}$$

$$f(t;\lambda,k) = \begin{cases} \frac{k}{\lambda}\left(\frac{t}{\lambda}\right)^{k-1} e^{-\left(\frac{t}{\lambda}\right)^k} & t \geq 0 \\ 0 & t < 0 \end{cases}$$

Finally, the cumulative hazard, defined as

$$\boldsymbol{H_T(t)} = \int_0^t h_T(u)du = \int_0^t \frac{f_T(u)}{S_T(u)}du$$

Use equation (2)

$$= -\int_0^t \frac{dS_T(u)}{S_T(u)} = -\int_0^t d\log S_T(u) = -\log S_T(u)\Big|_0^t = \boldsymbol{-\log S_T(t)} \qquad \textbf{(4)}$$

Use equation (3)

$$S_T(t) = e^{-H(t)}$$

With discrete event times, **the discrete hazard**

$$h_T(t) = P(T = t | T \geq t) \qquad \textbf{(5)}$$

is the probability of the event occurring in the time interval t conditional upon the individual still being alive at the beginning of t.
This gives rise to **the discrete-time survival probability**

$$S_T(t) = P(T > t) = \prod_{j=1}^{t} (1 - h_T(j)) \qquad \textbf{(6)}$$

2. Framework of Survival Analysis

n: a sample of size

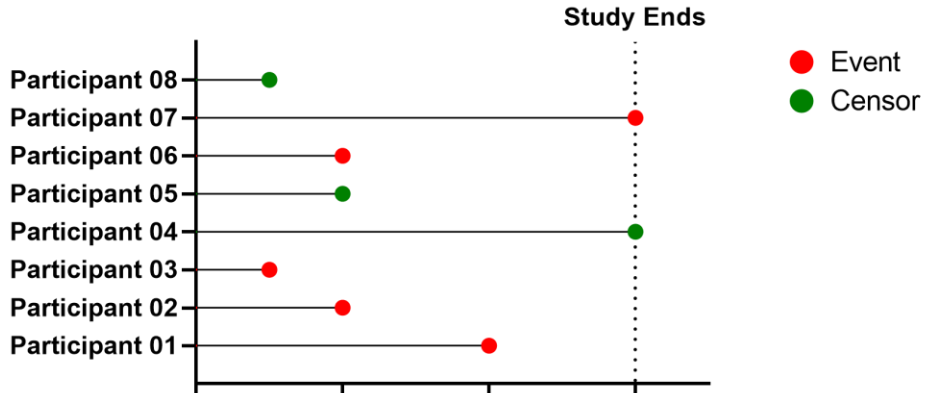
$i \in \{1,2,\dots,n\}$: individual or subject

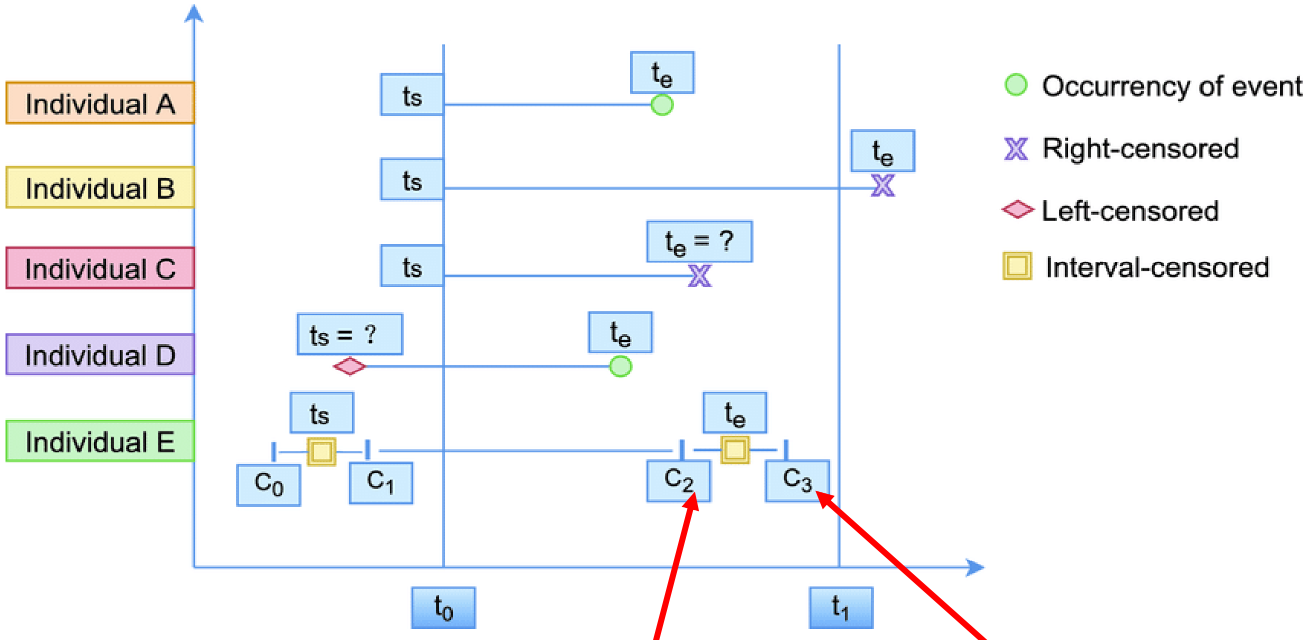
$T_i > 0$: the time until the event of interest for subject i occurs.

X_i: Covariates

δ_i: Indicator, denotes whether it is censored or not.

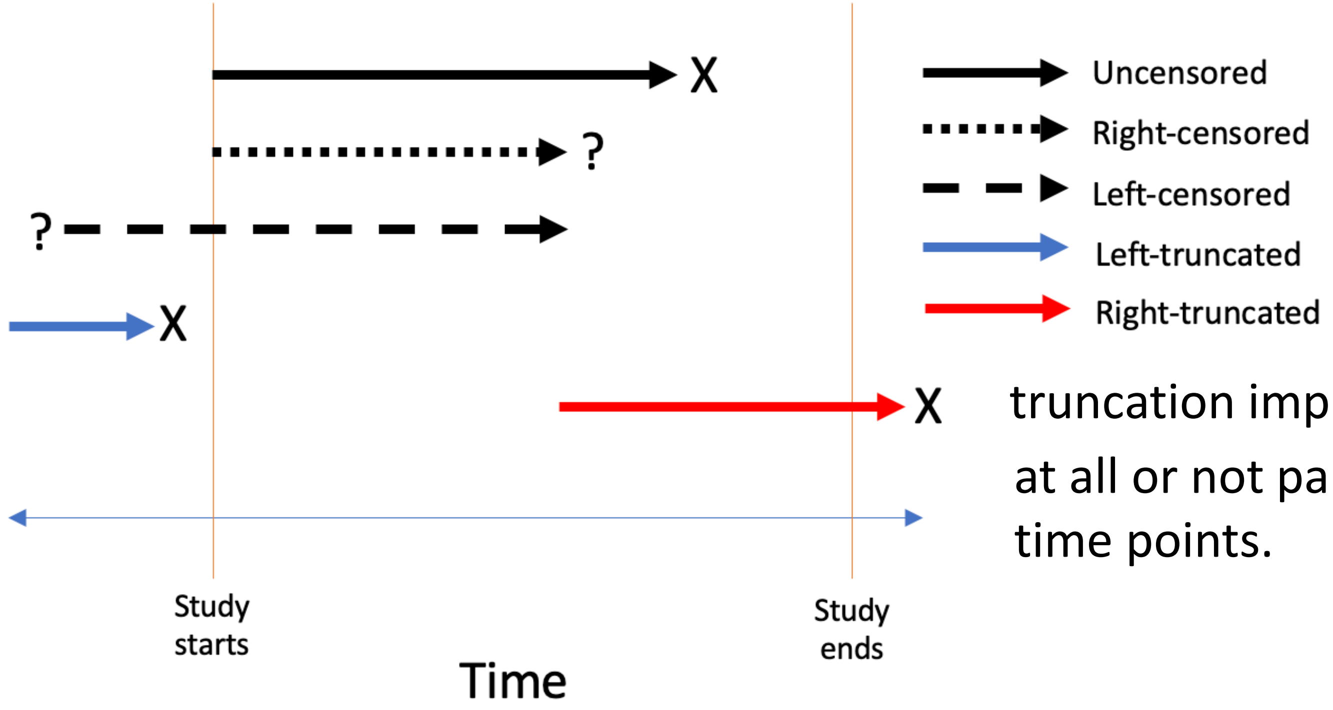
- # Censoring and Truncated





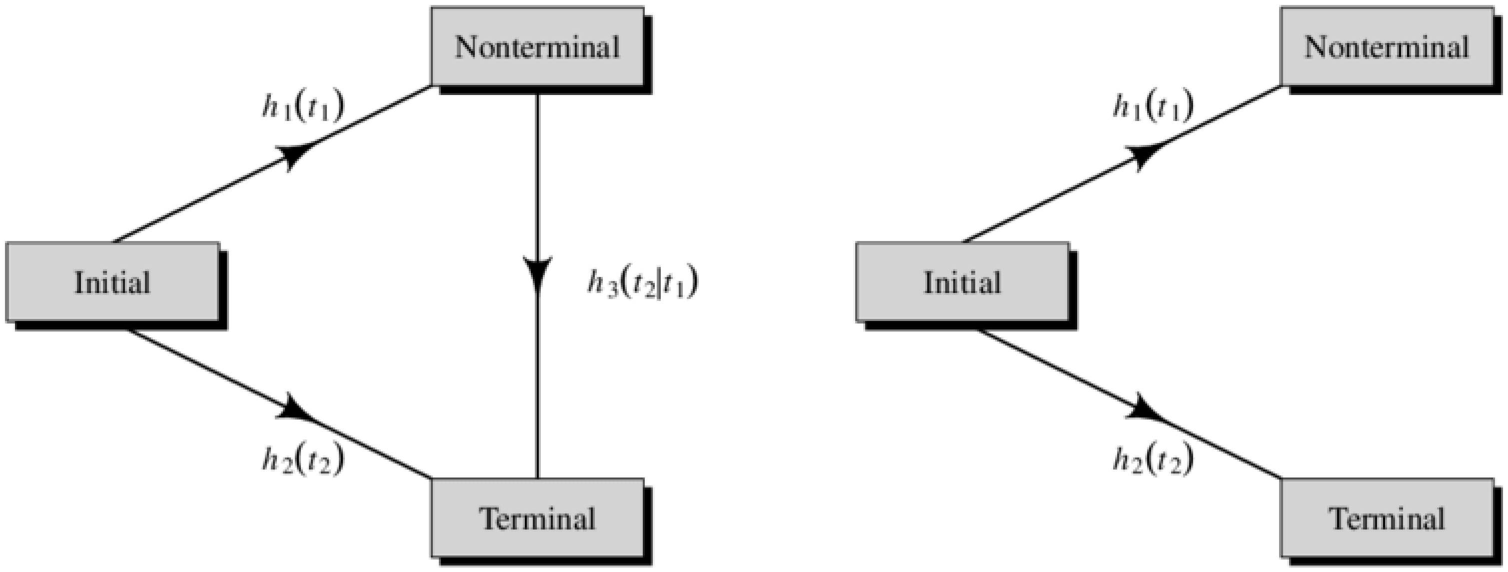
the endpoints of the censoring interval

we only know that the event occurs within the interval, but not the exact time.



truncation implies that subjects are either not part of the dataset at all or not part of the risk set for a specific event at certain time points.

Competing Risk

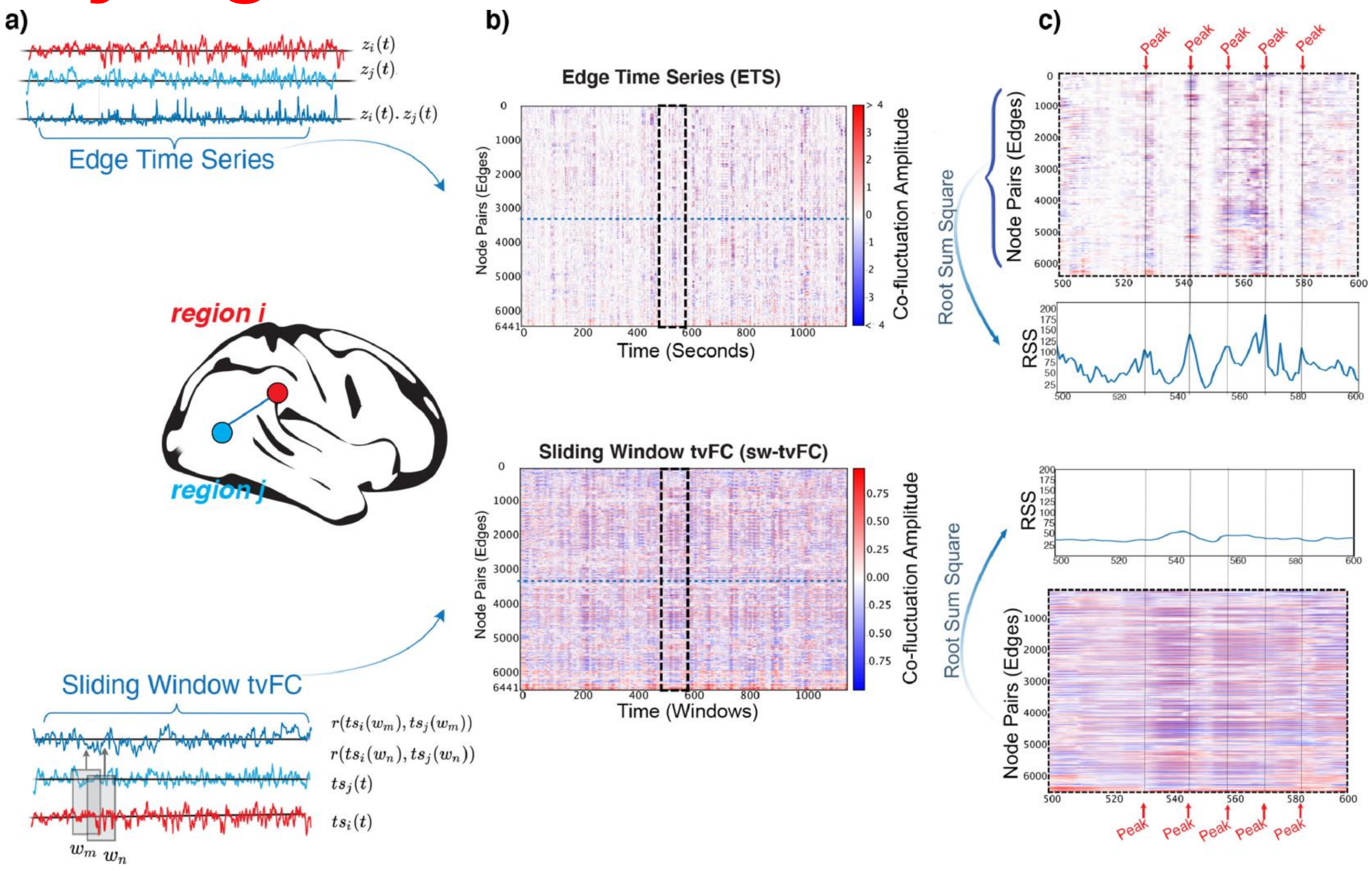


(a) Semi – competing risks.

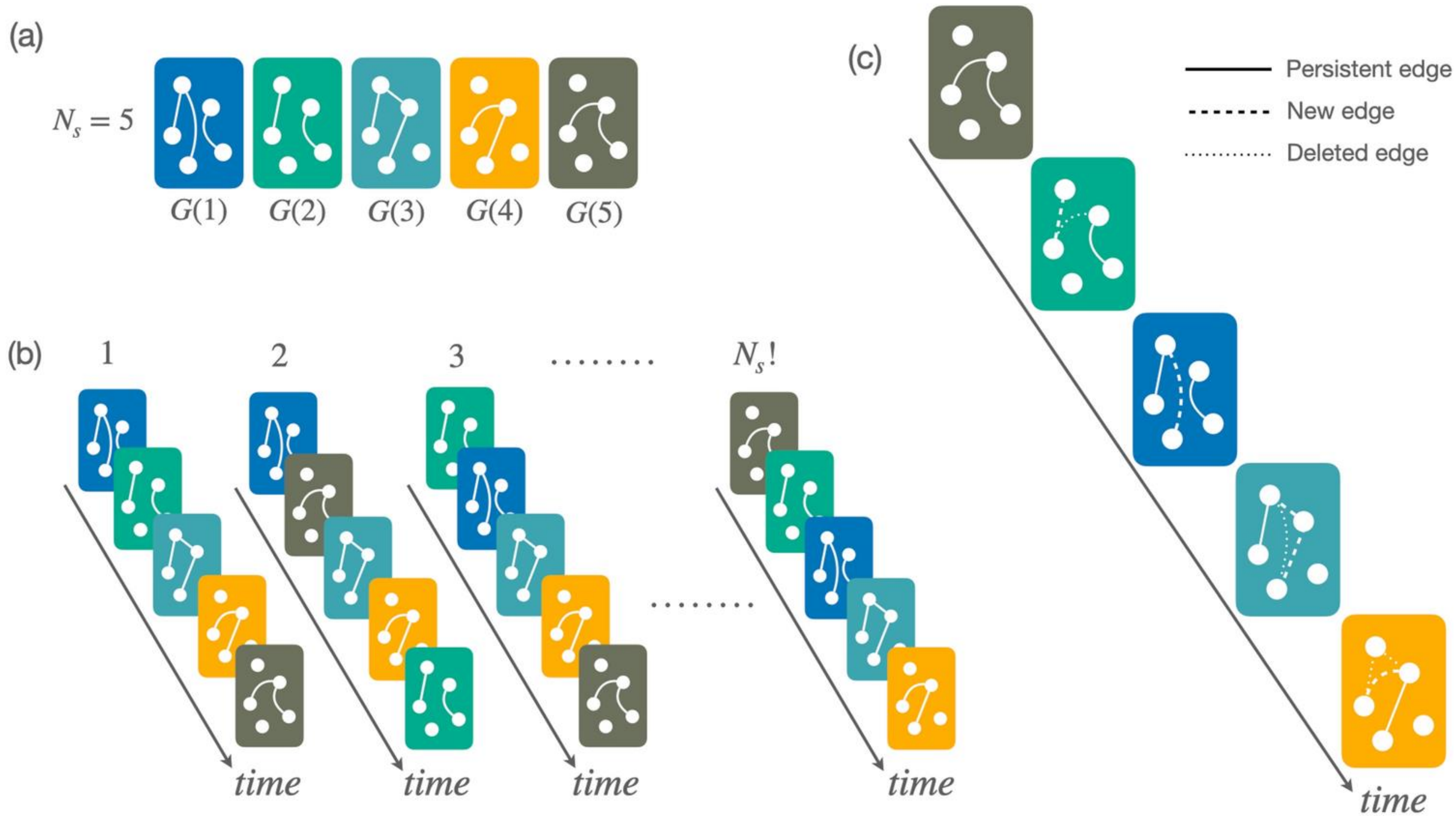
(b) Competing risks.

Alvares, D. Semi CompRisks: An R Package for the Analysis of Independent and Cluster-correlated Semi-competing Risks Data. 2019

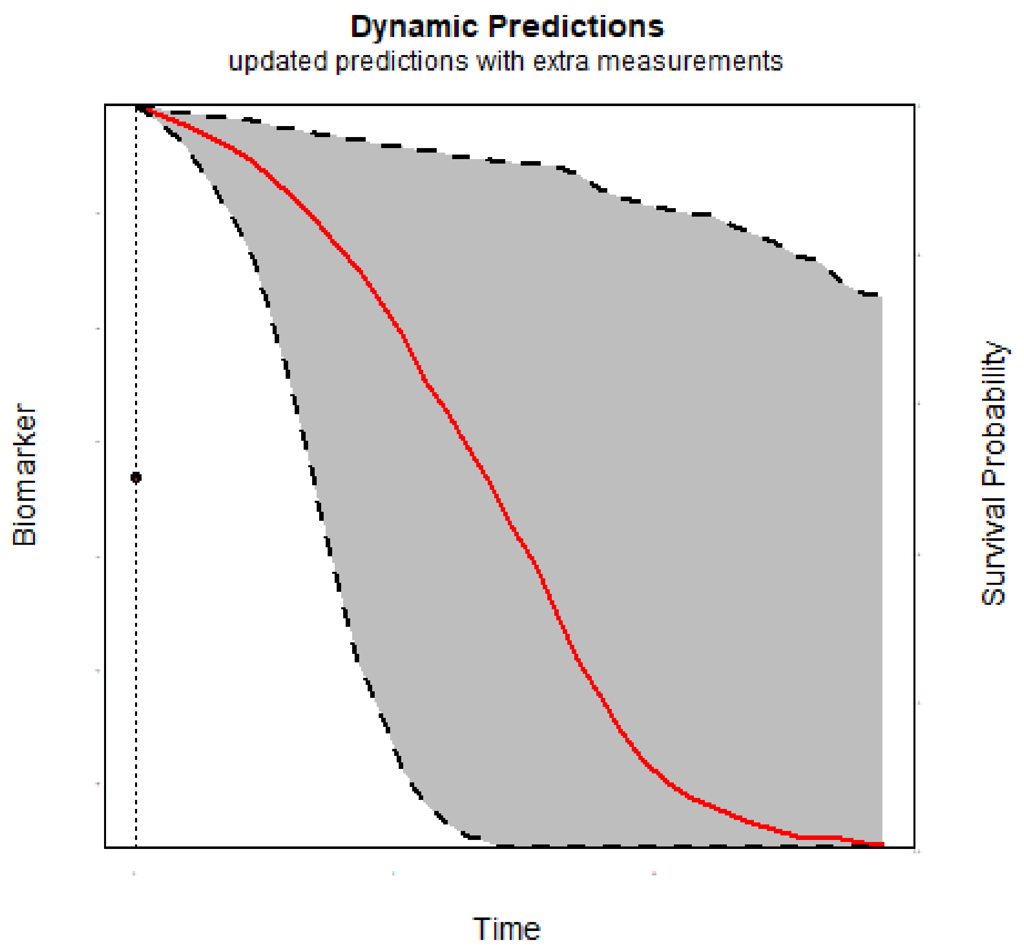
Varying Features and Covariates



Esfahlani FZ et al. 2022 . Edge-centric analysis of time-varying functional brain networks with applications in autism spectrum disorder

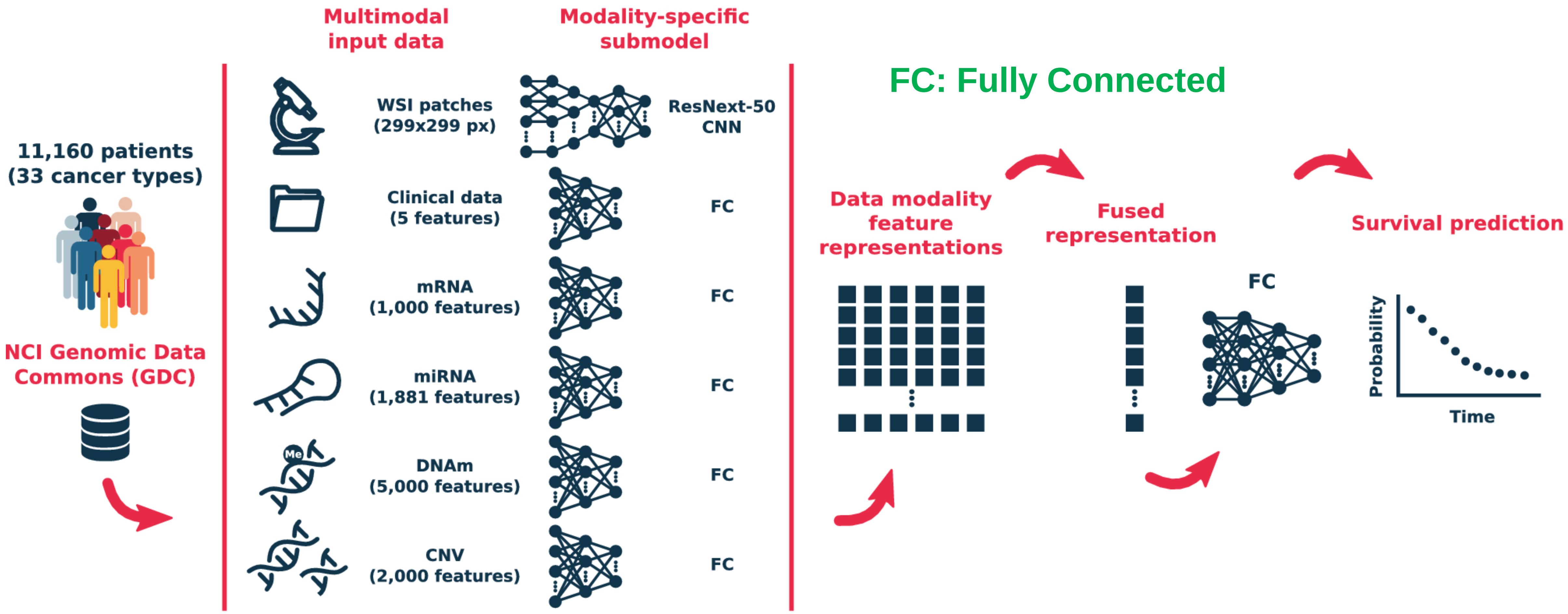


Mingueza FB et al. 2023. Characterization of interactions’ persistence in time-varying networks

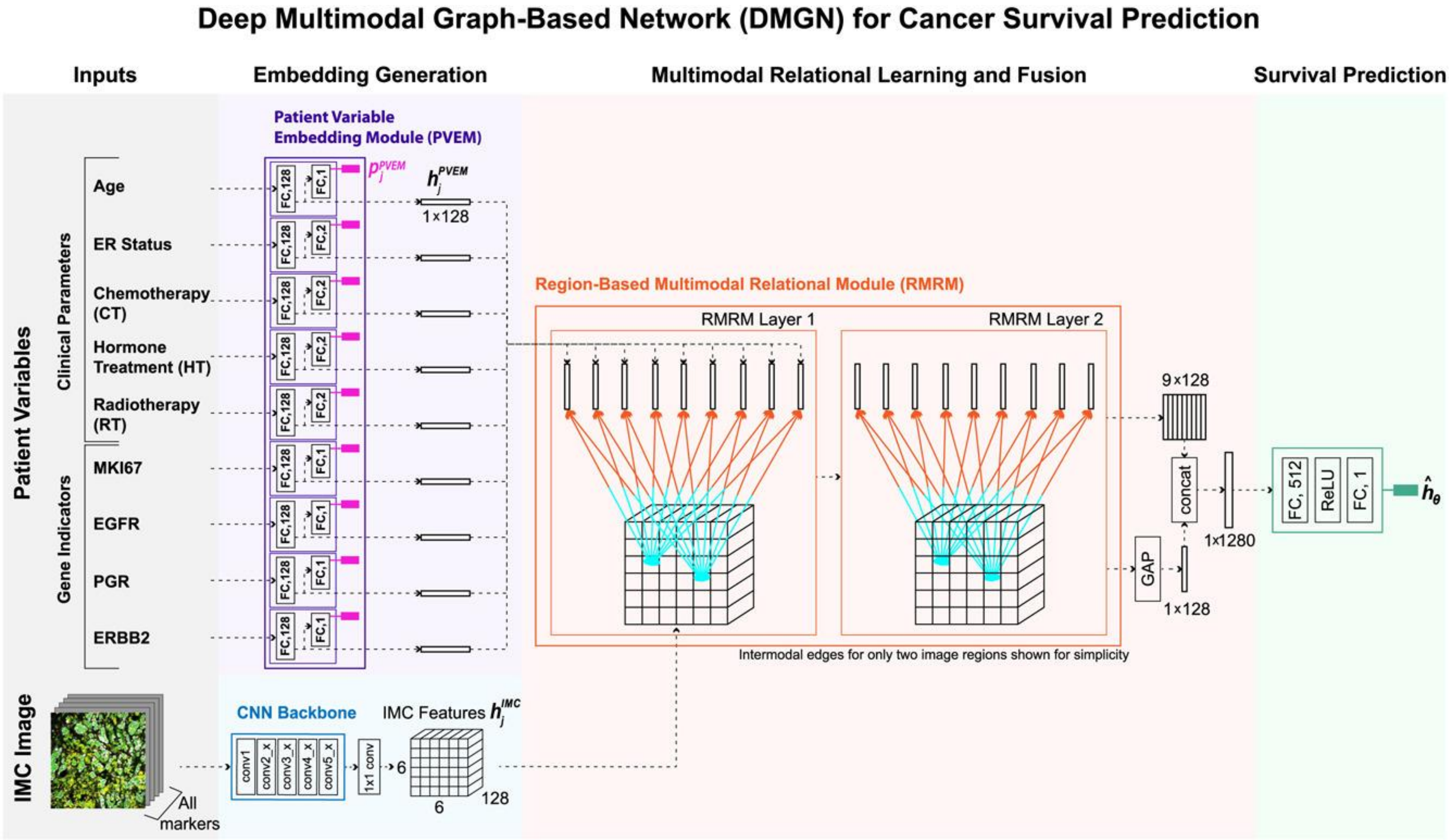


Individualized Predictions, Time-varying Effects and Time-varying Covariates

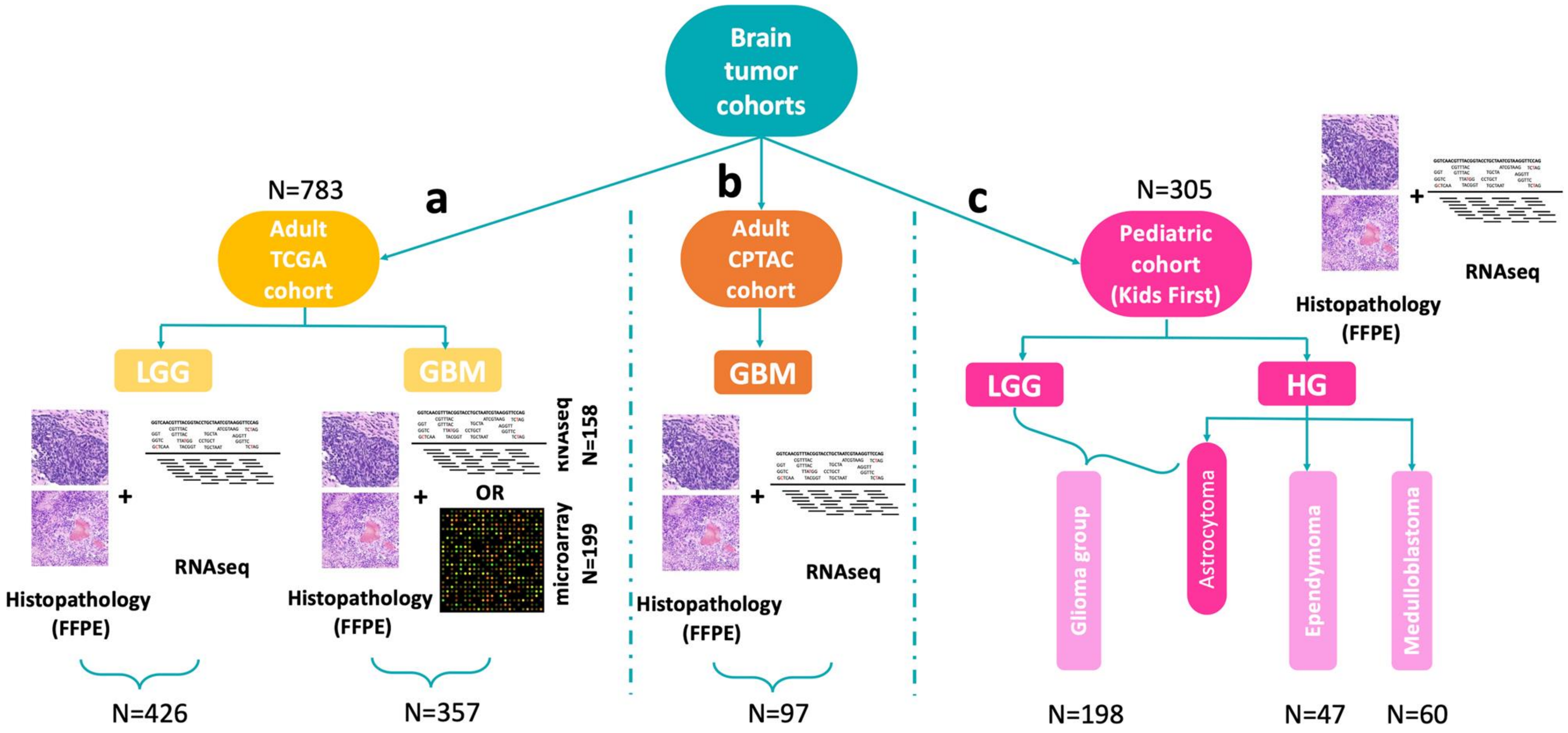
2. Multimodality



Luís A et al. 2021, Long-term cancer survival prediction using multimodal deep learning

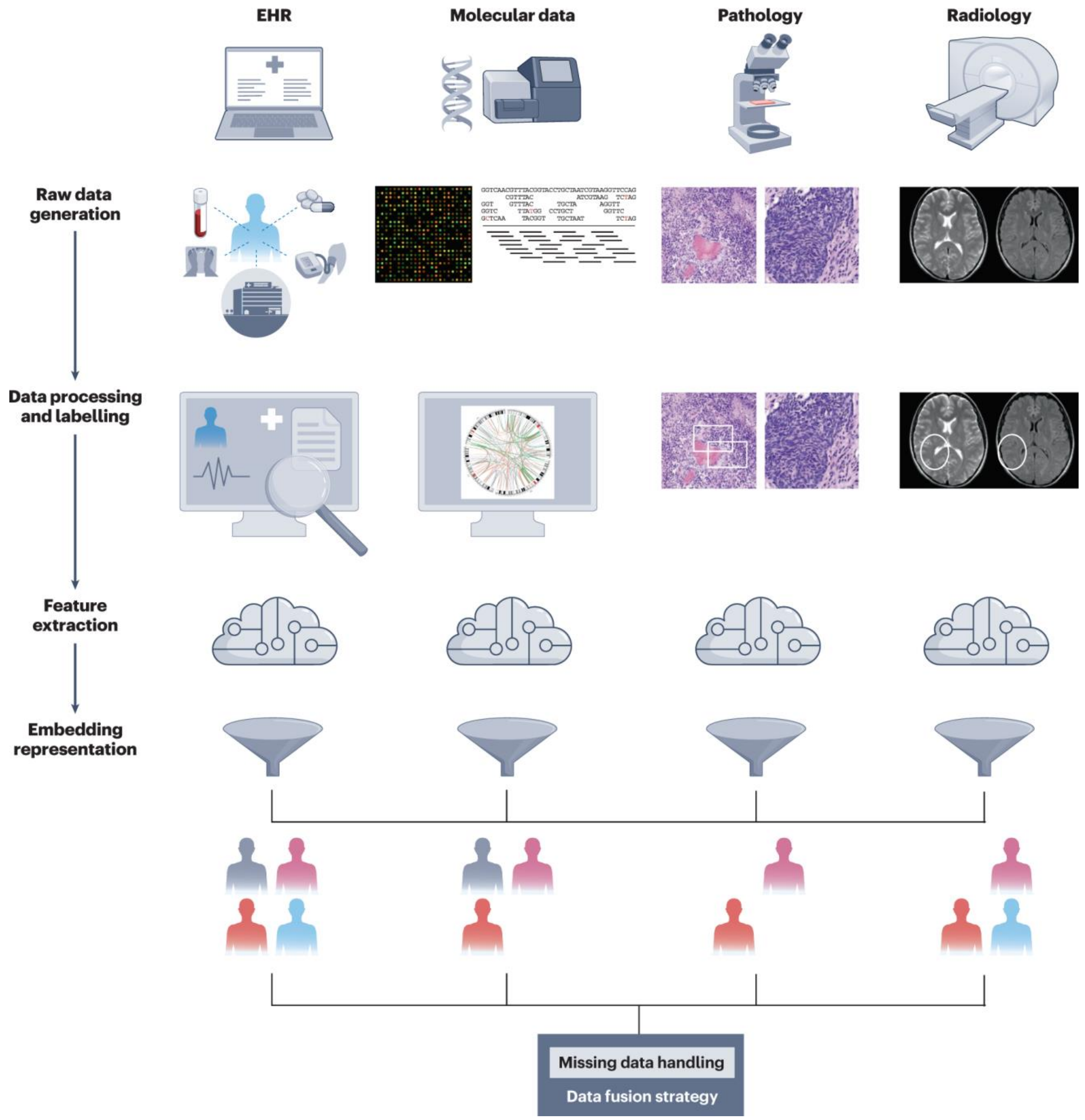


Fu et al. 2023. Deep multimodal graph-based network for survival prediction from highly multiplexed images and patient variables



Steyaert et al. 2023. Multimodal deep learning to predict prognosis in adult and pediatric brain tumors

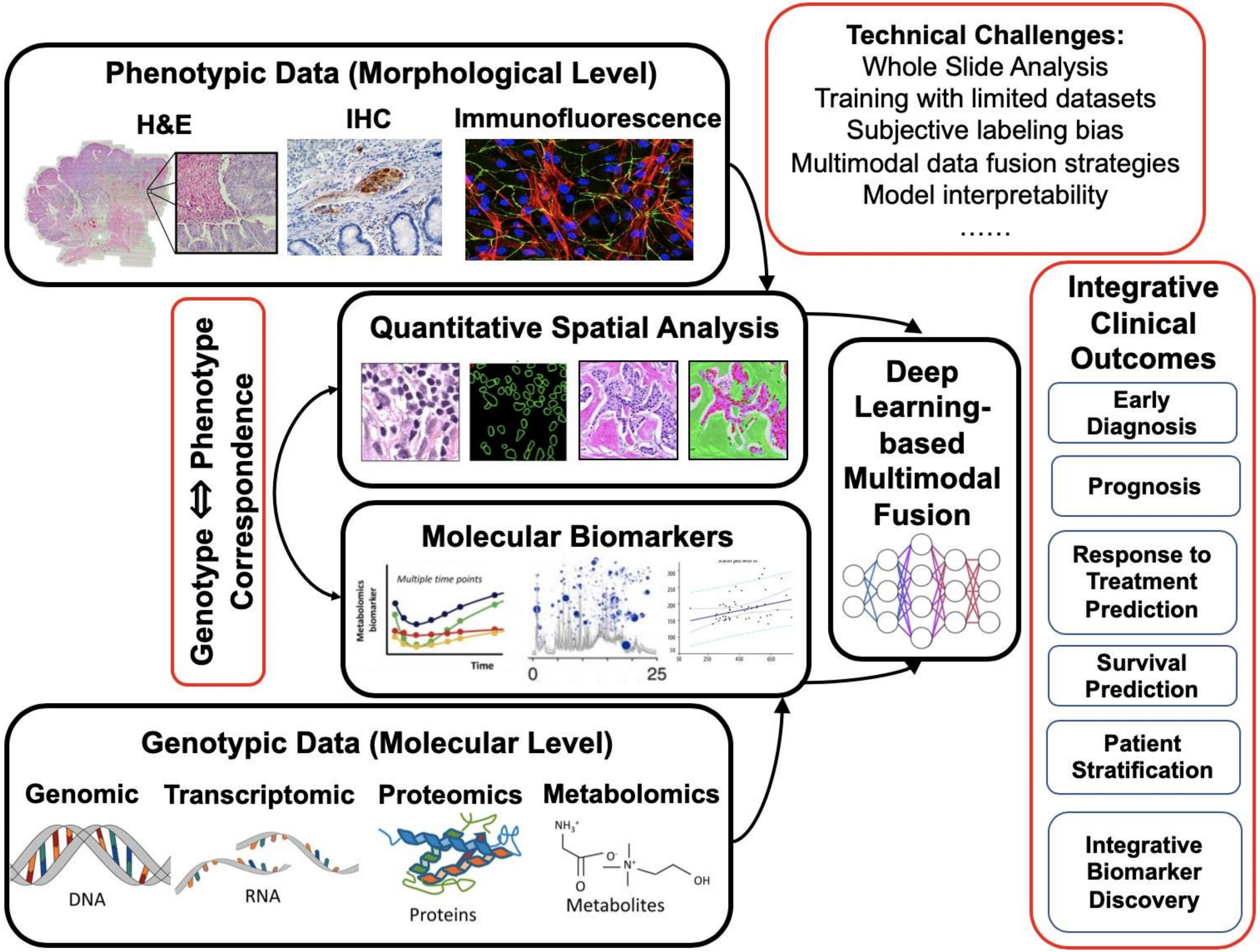
Code are available on GitHub at https://github.com/gevaertlab/MultiModalBrainSurvival and Zenodo at https://doi.org/10.5281/zenodo.7644876.



Multimodal data fusion for cancer biomarker discovery with deep learning.

nature machine intelligence

5, pages351–362 (2023)



Mahmood Lab
AI for Pathology Image Analysis

3. Estimation

- ## Notations

y_i: the observed event time of individual $i = 1, \ldots, n$

x_i: Covariates or features for individual i

δ_i: Indicator variable. $\delta_i = 1,$ indicates that the survival time is observed, $\delta_i = 0$ indicates that the individual i is censored.

- ## Parametric Estimation

Define the density function for an event at time t as

$$f(t|\theta), t \geq 0, \theta = \theta(x) = (g_1(x, \beta_1), g_2(x, \beta_2), \ldots) \quad \textbf{(7)}$$

where $g_1(.,.), g_2(.,.)$ are the real-valued functions of covariates and parameters $\beta_1, \beta_2, \ldots,$

- ## Likelihood Function

Let O, C, L_c be the sets of observed event times, right-censored, and left-censored observations, respectively.

The likelihood function is defined as

$$L(\theta) = \prod_{i \in O} f(y_i) \prod_{j \in C} S(y_j) \prod_{k \in L_c} (1 - S(y_k)) \quad \textbf{(8)}$$

Using equation (4), we obtain

$$S(t) = e^{-\int_0^t h(u)du} \quad \textbf{(9)}$$

Combining equations (2), (8) and (9), we can replace equation (8) by

$$L(\theta) = \prod_{i \in O} h(y_i) e^{-\int_0^{y_i} h(u)du} \prod_{j \in C} e^{-\int_0^{y_j} h(u)du} \prod_{k \in L_c} \left(1 - e^{-\int_0^{y_k} h(u)du}\right) \quad \textbf{(10)}$$

Thus, the likelihood can always be expressed in terms of only the hazard rate.

Full Likelihood Function

- **Right Censoring**

T^*: True event time
T: Observed event time.
C^*: the censoring time
right-censored event time: $T = \min(T^*, C^*)$

- Full Likelihood

$$L = \prod_{i=1}^{n} f(T_i|X_i)^{\delta_i} S(T_i|X_i)^{1-\delta_i}$$

$$= \prod_{i=1}^{n} h(T_i|X_i)^{\delta_i} e^{-H(T_i|X_i)} \qquad \textbf{(11)}$$

- ## the Cox PH regression models

the Cox PH regression models the hazard rate at time t, conditional on features x, as the product of a non-parametrically estimated baseline hazard $h_0(t)$ and the exponentiated log-risk η = g(x, β):

$$h(t|X) = h_0(t)\exp(\eta = g(X, \beta)) \quad \textbf{(12)}$$

Feature effects are multiplicative with respect to the hazard rate independently of time, yielding proportionality of hazards. the relative risk function: $e^{g(X,\beta)}$

- ## Log Partial Likelihood Function

Partial likelihood estimation uses the product of conditional densities as the density of the joint conditional distribution.

$$l(\beta) = \sum_{m=1}^{M}\left(g\left(X_{(m)}, \beta\right) - \log \sum_{j \in R(t_{(m)})} \exp\left(g\left(X_j, \beta\right)\right)\right) \quad \textbf{(13)}$$

where $t_{(m)}$ is the *m*th ordered event (*m* ∈ *{1, . . . , M}*), $R(t_{(m)})$ denotes the risk set at that time point, and $X_{(m)}$ is the feature vector of the individual experiencing the event at $t_{(m)}$.

Or

$$L_{Cox} = \prod_m \left(\frac{\exp(g(X_i, \beta))}{\sum_{j \in R(t_m)} \exp(g(X_j, \beta))} \right)^{\delta_m} \quad \textbf{(14)}$$

and the negative partial log-likelihood can then be used as a loss function

$$loss = \sum_m \delta_m \log\left(\sum_{j \in R(t_m)} \exp\left(g(X_j, \beta) - g(X_m, \beta)\right) \right) \quad \textbf{(15)}$$

- **Linear Functions**

Define linear function

$$g(X, \beta) = X^T \beta$$

The log partial likelihood function is reduced to

$$l(\beta) = \sum_{m=1}^{M} \left(X_{(m)}^T \beta - \log \sum_{j \in R\left(t_{(m)}\right)} \exp(X_j^T \beta) \right) \quad \textbf{(16)}$$

Deep Survival Analysis

- **Proportional and non-proportional extensions of the Cox model.**

  Kvamme et al. 2019. Time-to-Event Prediction with Neural Networks and Cox Regression

A python package for the proposed methods is available at https://github.com/havakv/pycox.

- **Batch as a Risk Set**

  As the loss in (14) sums over risk sets $R(t_m)$, which can be as large as the full data set, it cannot be computed in batches. Nevertheless, it is possible to do batched iterations by subsampling the data set (to a batch) and restrict the set $R(t_m)$ to only contain individuals in the current batch.

  This scales well for proportional methods, but would be very computationally expensive for our non-proportional extension. Hence, propose an approximation of the loss that is easily batched. Weighting likelihood in equation (14) yields

$$L_{Cox} = \prod_{m=1}^{M} \left(\frac{\exp(g(X_m,\beta))}{w_m \sum_{j\in\tilde{R}(t_m)} \exp\left(g\left(X_j,\beta\right)\right)} \right)^{\delta_m}, \tilde{R}(t_m) \text{ is a subset of } R(t_m) \qquad \textbf{(17)}$$

which can be further simplified to

$$loss = \frac{1}{n} \sum_{m:\delta_m=1} \log\left(\sum\nolimits_{j \in \tilde{R}(t_m)} \exp\left(g\left(X_j, \beta\right) - g(X_m, \beta)\right)\right) \quad \textbf{(18)}$$

where n denotes the number of events in the data set. We find that it is often sufficient to sample only one individual j from the risk set, which gives us the loss

$$loss = \frac{1}{n} \sum_{m:\delta_m=1} \log\left(1 + exp\left(g\left(X_j, \beta\right) - g(X_m, \beta)\right)\right), j \in R(t_m) - \{m\} \quad \textbf{(19)}$$

$$1 = \exp((g(X_m, \beta) - g(X_m, \beta))$$

- ## Non-Proportional Cox-Time

The proportionality assumption of the Cox model can be rather restrictive. We now let the relative risk function depend on time.

$$h(t|X) = h_0(t)\exp(\eta = g(t, X, \beta)) \quad \textbf{(20)}$$

loss function:

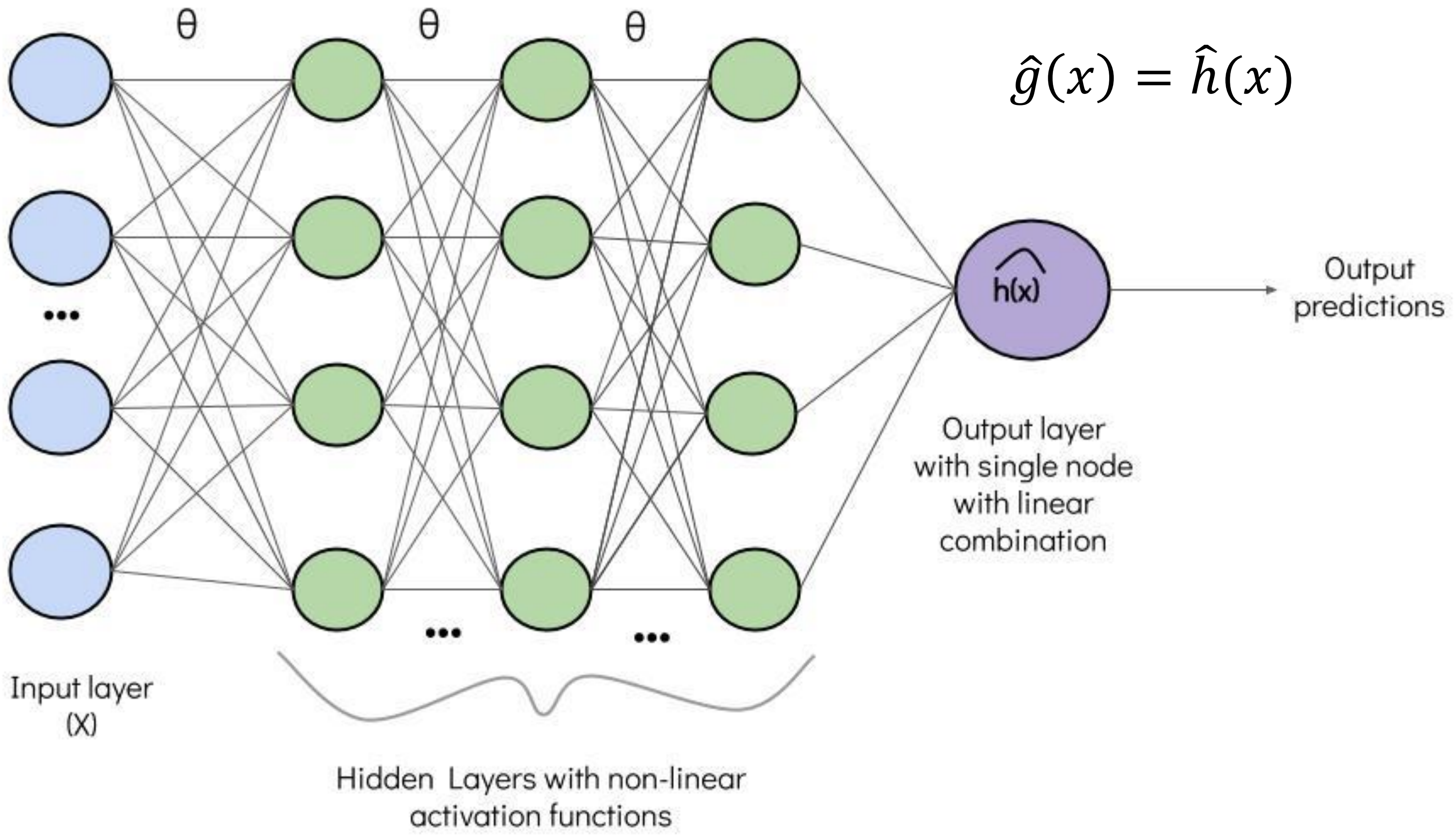
$$loss = \frac{1}{n} \sum_{m:\delta_m=1} \log\left(\sum_{j \in \tilde{R}(t_m)} \exp\left(g\left(T_m, X_j, \beta\right) - g\left(T_m, X_m, \beta\right)\right)\right) \quad \textbf{(21)}$$

Define the penalty:

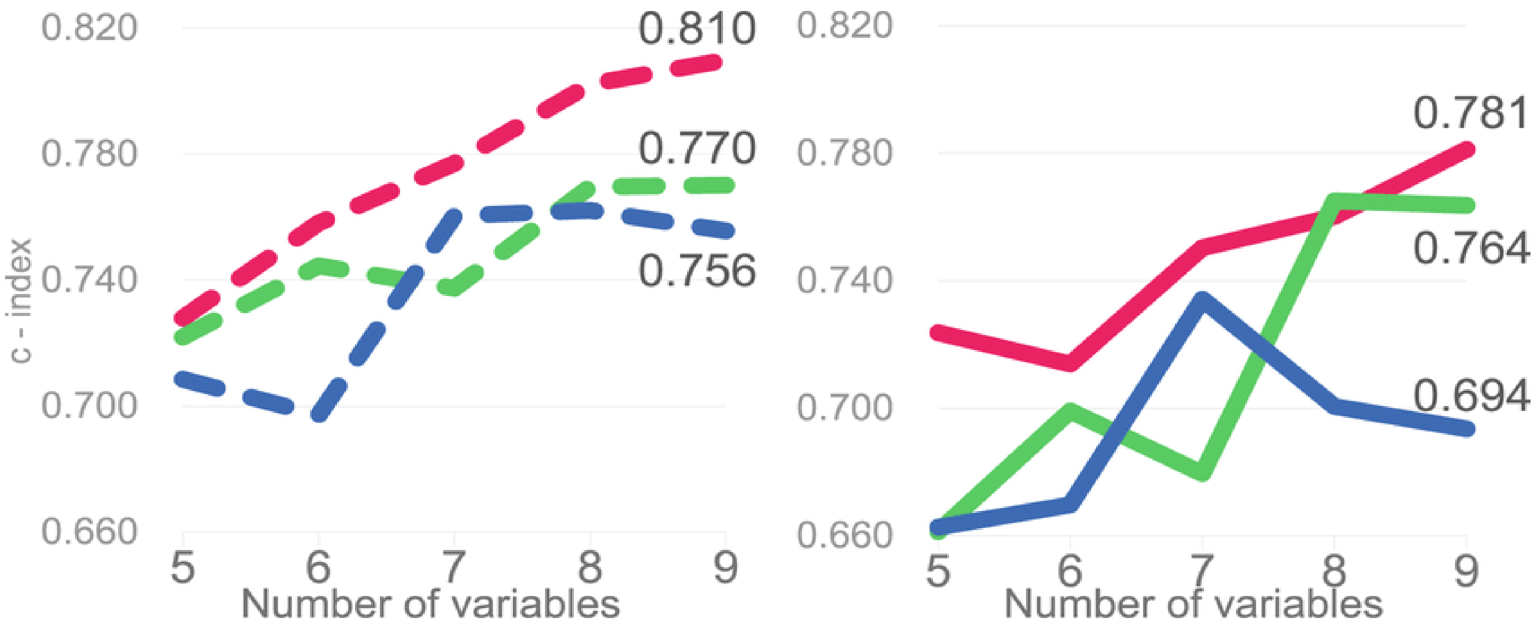
$$\text{Penalty} = \alpha \sum_{m:\delta_m=1} \sum_{j \in \tilde{R}(t_m)} \left|g\left(T_m, X_j, \beta\right)\right| \quad \textbf{(22)}$$

Then, we obtain the final loss function

$$\mathcal{L} = \frac{1}{n} \sum_{m:\delta_m=1} \log\left(\sum_{j \in \tilde{R}(t_m)} \exp\left(g\left(T_m, X_j, \beta\right) - g\left(T_m, X_m, \beta\right)\right)\right) + \alpha \sum_{m:\delta_m=1} \sum_{j \in \tilde{R}(t_m)} \left|g\left(T_m, X_j, \beta\right)\right| \quad \textbf{(23)}$$



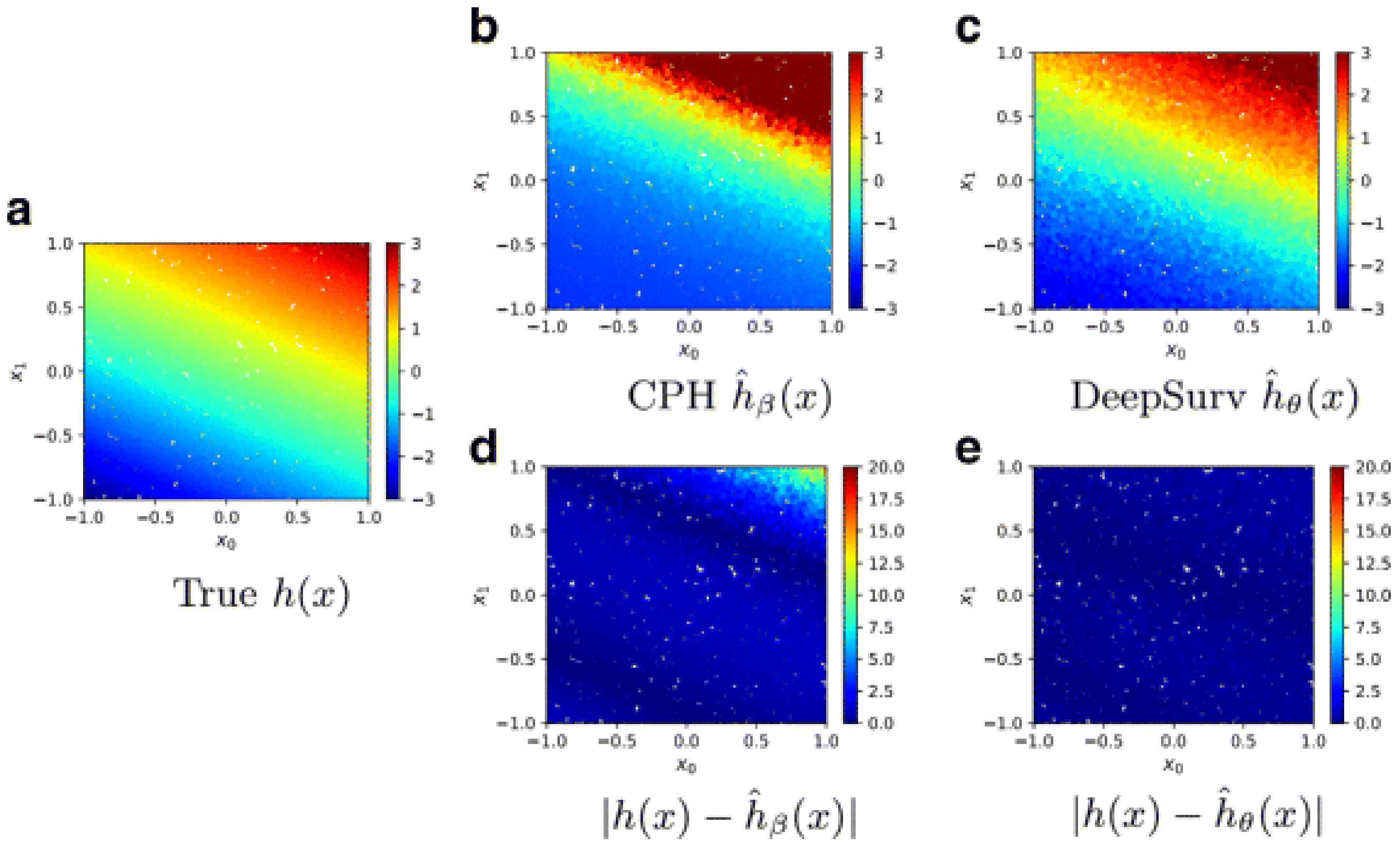
Löschmann and Smorodina, 2020. Deep Learning for Survival Analysis



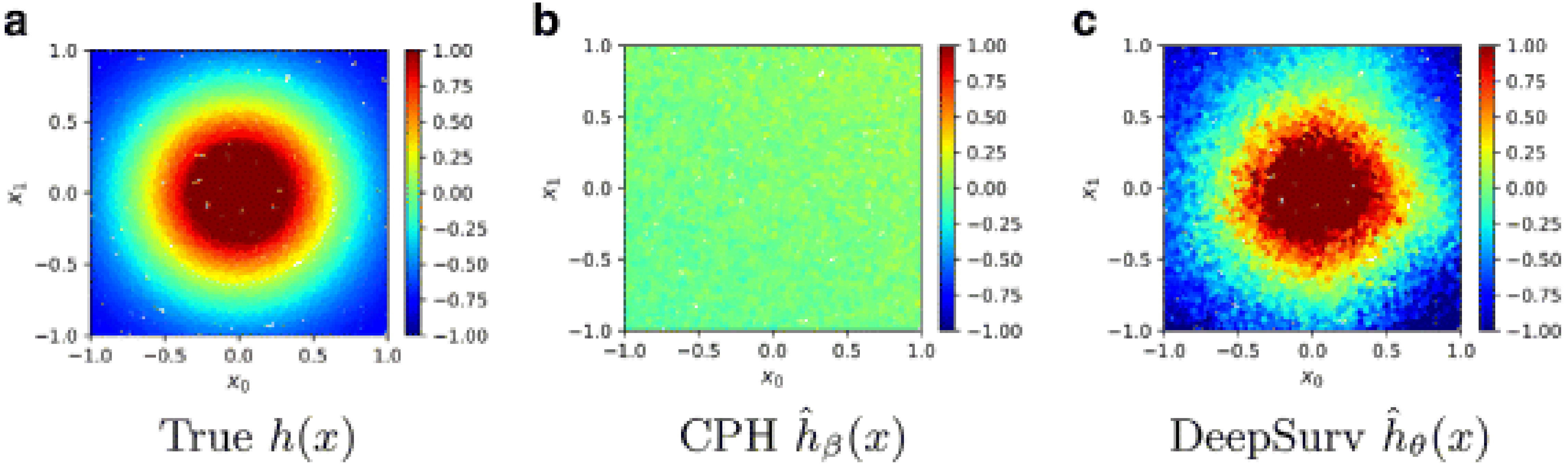
the C-index estimates the probability that, for a random pair of individuals, the predicted survival times of the two individuals have the same ordering as their true survival times

Models & Datasets	Number of variables	5 = T + N + HG + PNI + ENE	6 = T + N + HG + PNI + ENE + LVP	7 = T + N + HG + PNI + ENE + LVP + OR	8 = T + N + HG + PNI + ENE + LVP + OR + BM	9 = T + N + HG + PNI + ENE + LVP + OR + BM + RM
DeepSurv	Train	**0.728*** (0.724-0.732)	**0.758*** (0.754-0.762)	**0.777*** (0.772-0.782)	**0.802*** (0.798-806)	**0.810*** (0.805-815)
	Test	0.724	0.714	0.750	0.760	0.781
RSF	Train	**0.722*** (0.721-0.723)	**0.745*** (0.743-0.746)	0.737 (0.736-738)	**0.770*** (0.768-0.771)	**0.770*** (0.768-0.772)
	Test	0.661	0.699	0.680	0.765	0.764
CPH	Train	0.709 (0.697-0.715)	0.697 (0.692-0.709)	**0.760*** (0.753-0.768)	0.762 (0.752-767)	0.756 (0.753-0.767)
	Test	0.663	0.670	0.734	0.701	0.694

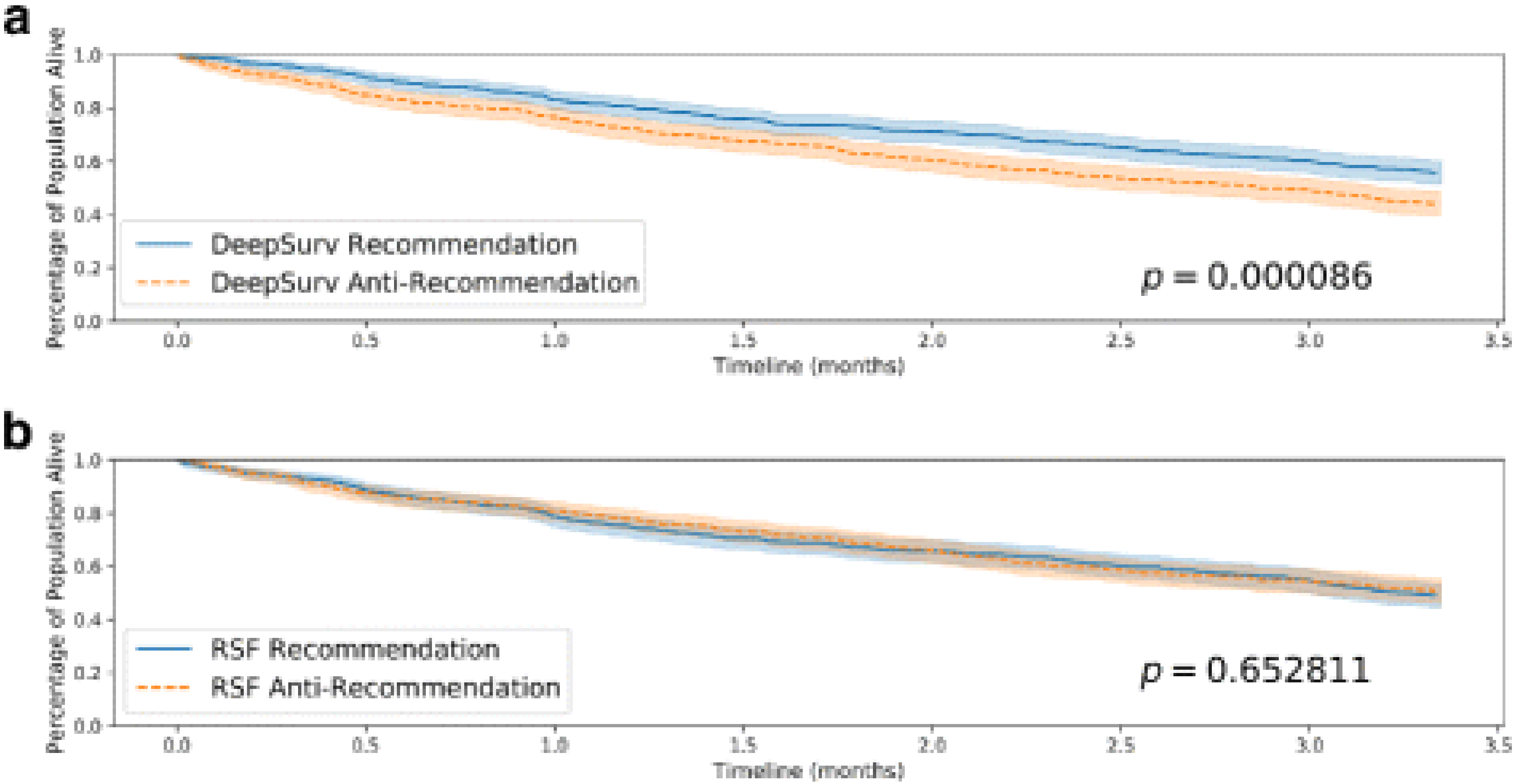
Performance of DeepSurv, RSF, and CPH model in terms of c-index (95%.)



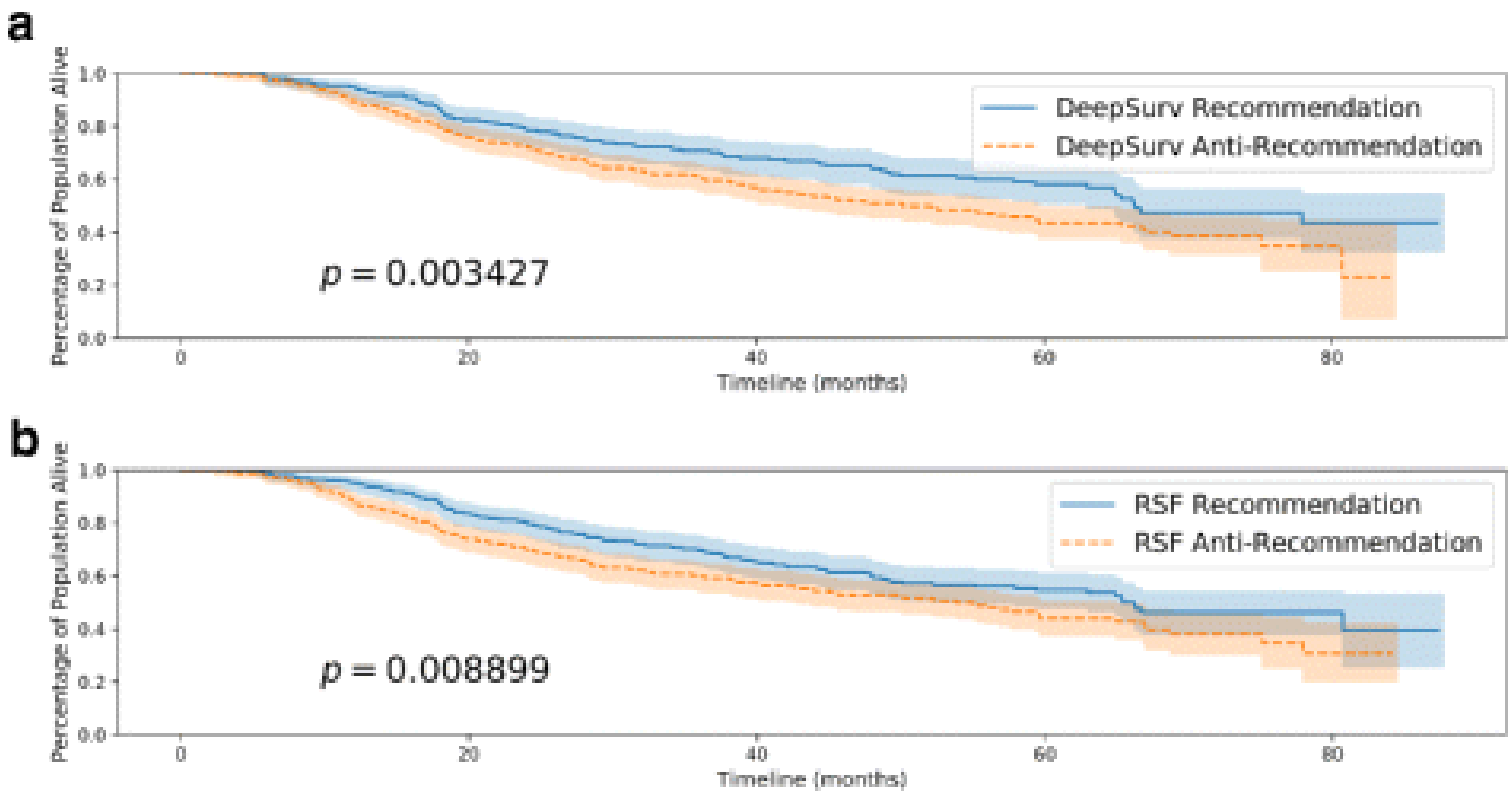
Löschmann and Smorodina, 2020. Deep Learning for Survival Analysis



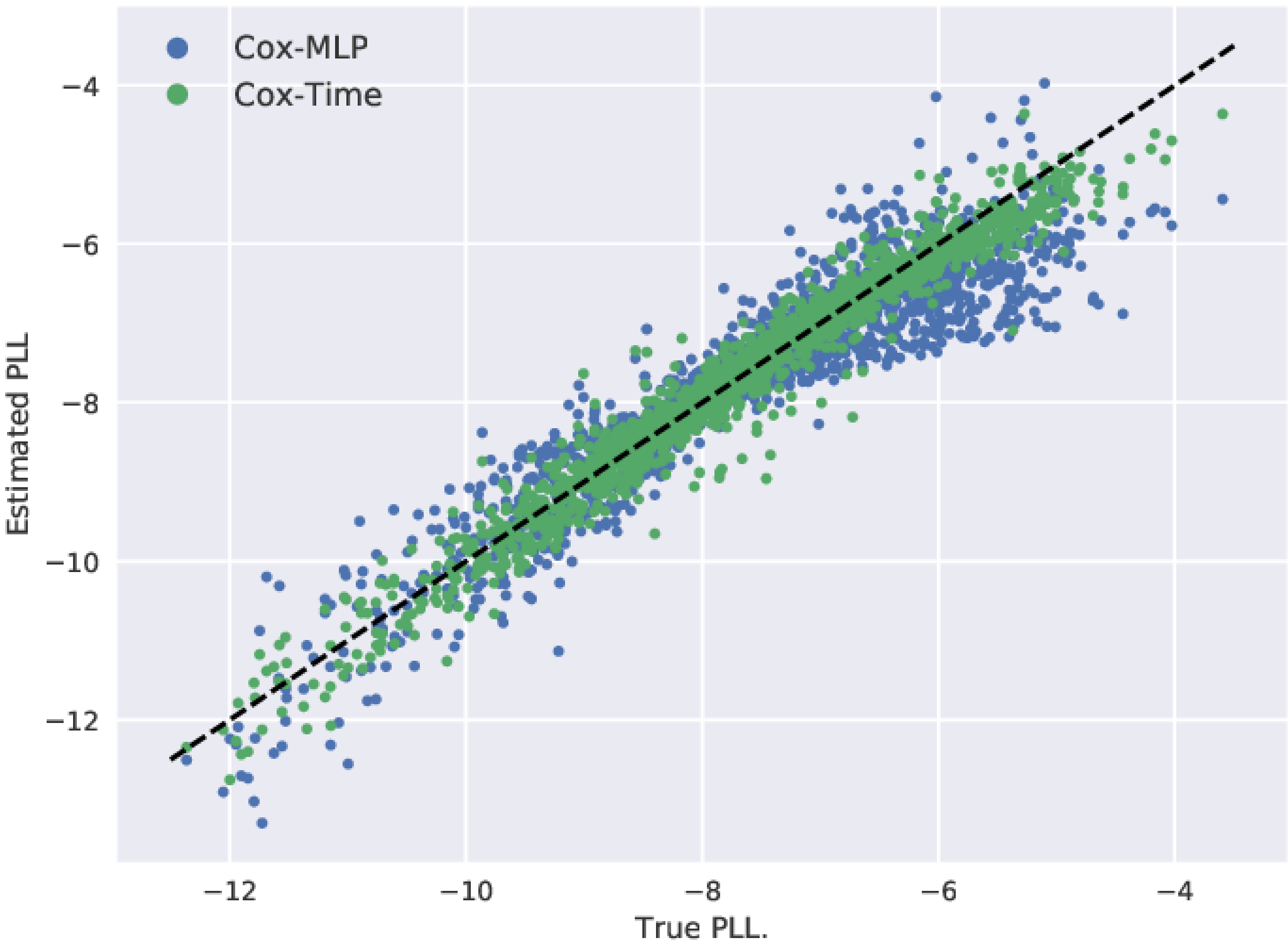
Löschmann and Smorodina, 2020. Deep Learning for Survival Analysis



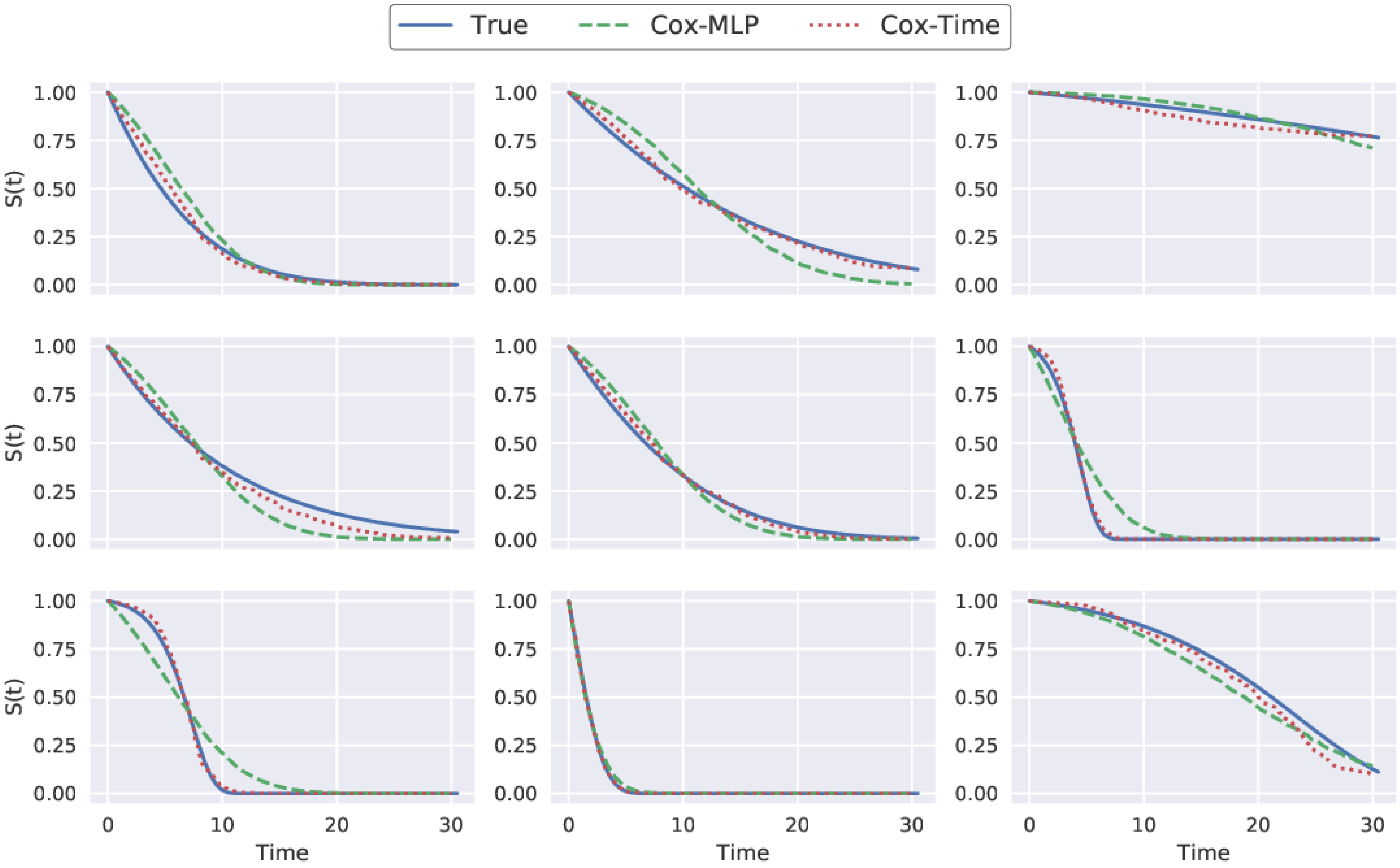
Timeto-event prediction with neural networks and cox regression. Journal of machine learning research, 20(129): 1–30, 2019.



Timeto-event prediction with neural networks and cox regression. Journal of machine learning research, 20(129): 1–30, 2019.

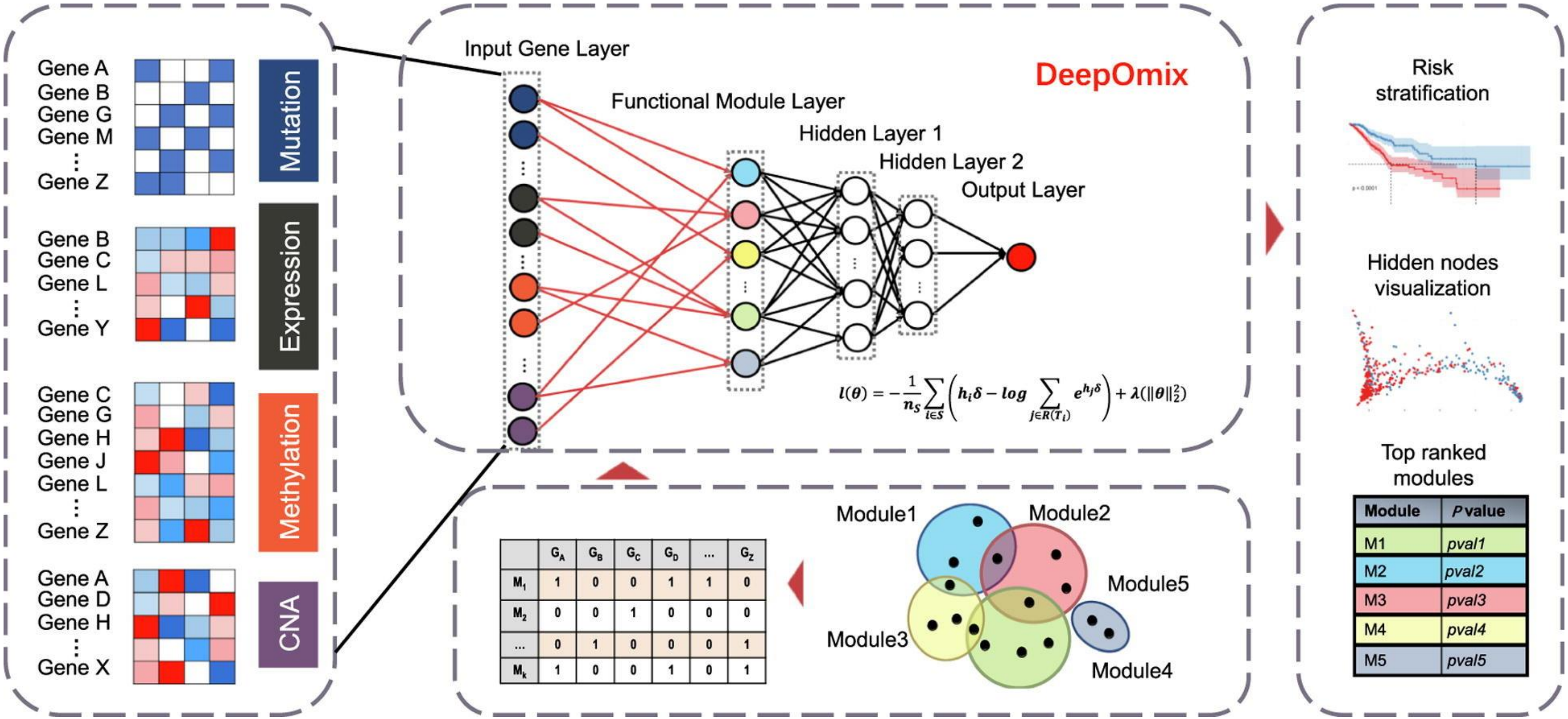


Timeto-event prediction with neural networks and cox regression. Journal of machine learning research, 20(129): 1–30, 2019.



Timeto-event prediction with neural networks and cox regression. Journal of machine learning research, 20(129): 1–30, 2019.

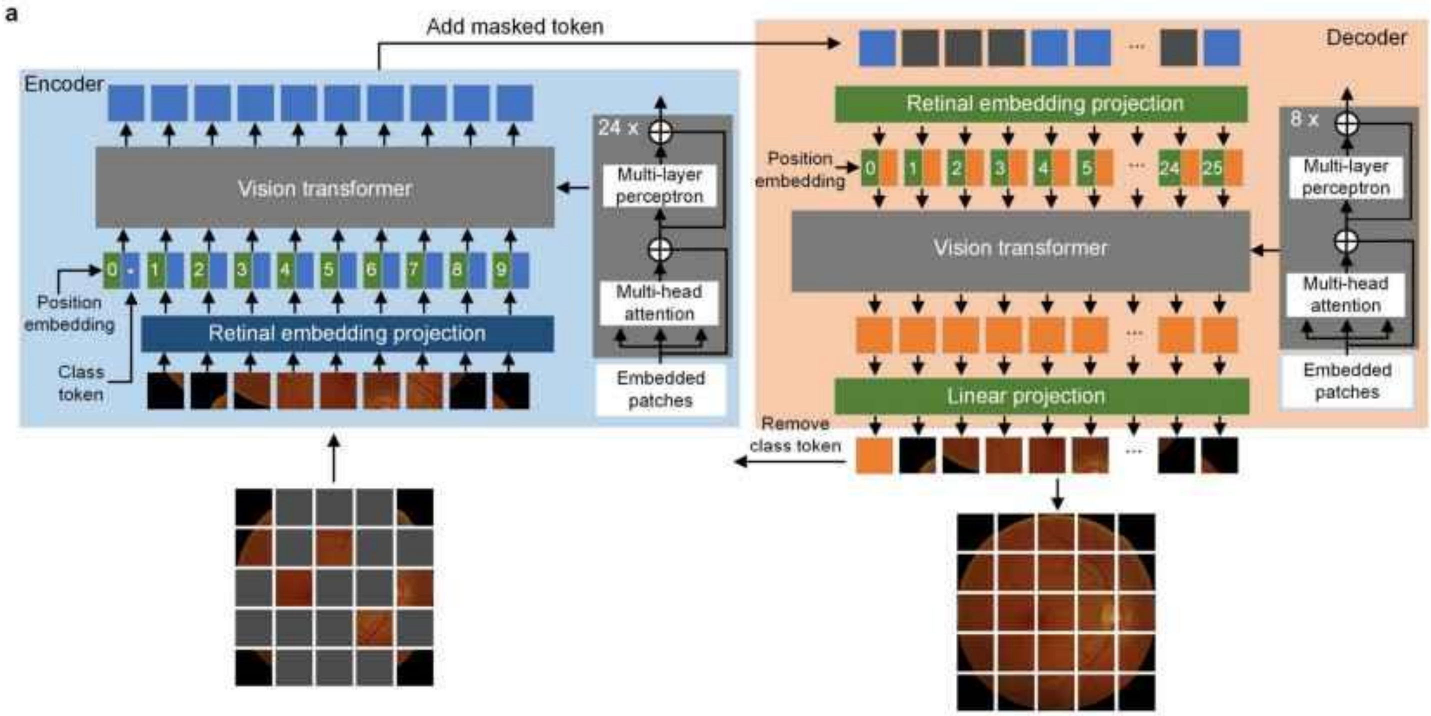
Multimodal Survival Analysis



	G_A	G_B	G_C	G_D	...	G_Z
M_1	1	0	0	1	1	0
M_2	0	0	1	0	0	0
...	0	1	0	0	0	1
M_k	1	0	0	1	0	1

Module	P value
M1	pval1
M2	pval2
M3	pval3
M4	pval4
M5	pval5

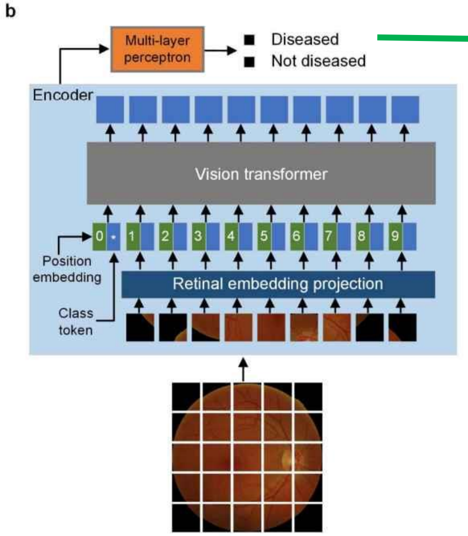
Zhao et al. 2021. DeepOmix: A scalable and interpretable multi-omics deep learning framework and application in cancer survival analysis



A foundation model for generalizable disease detection from retinal images

Deep multimodal graph-based network for survival prediction from highly multiplexed images and patient variables
Fu et al. 2023

All data used in this study are publicly available

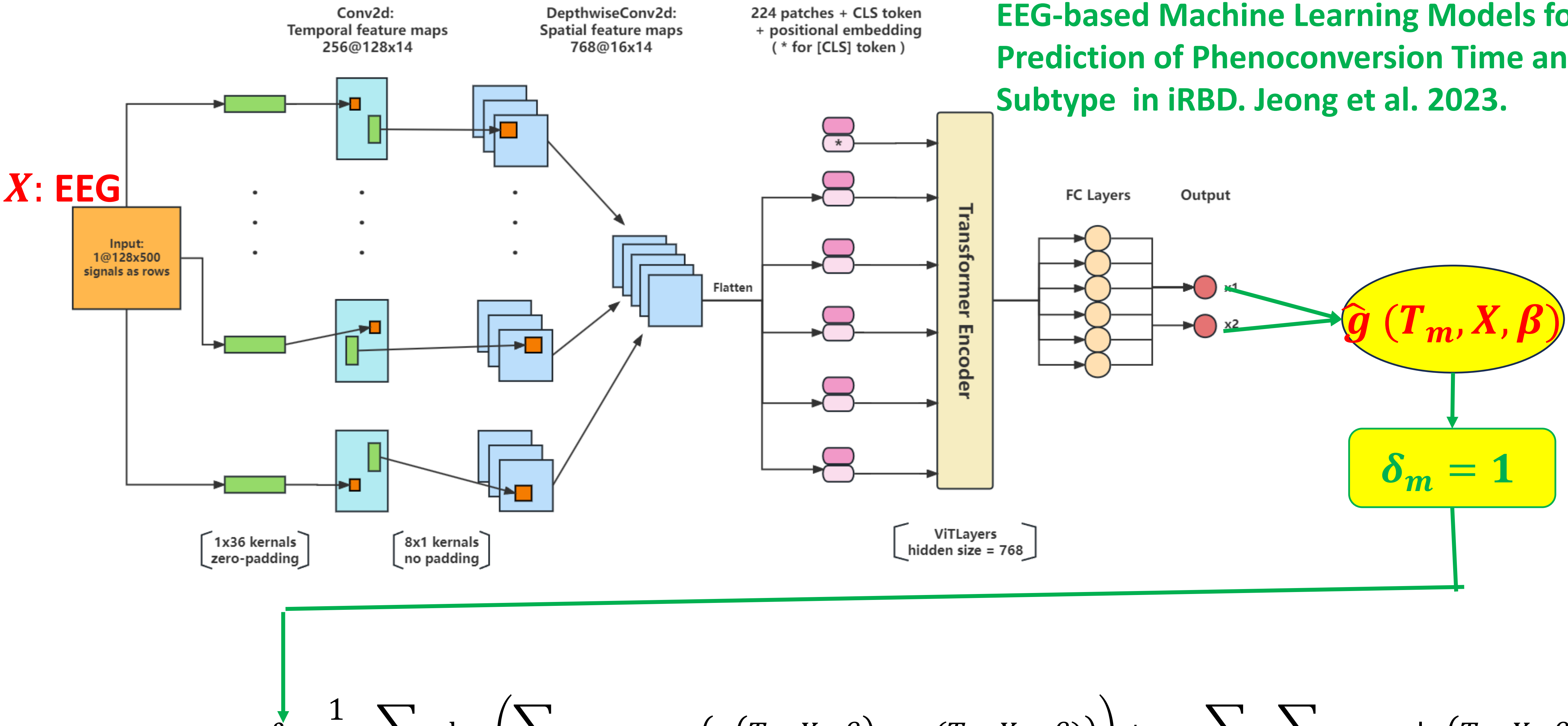


$$\hat{g}\,(T_m, X, \beta)$$

$$\delta_m = 1$$

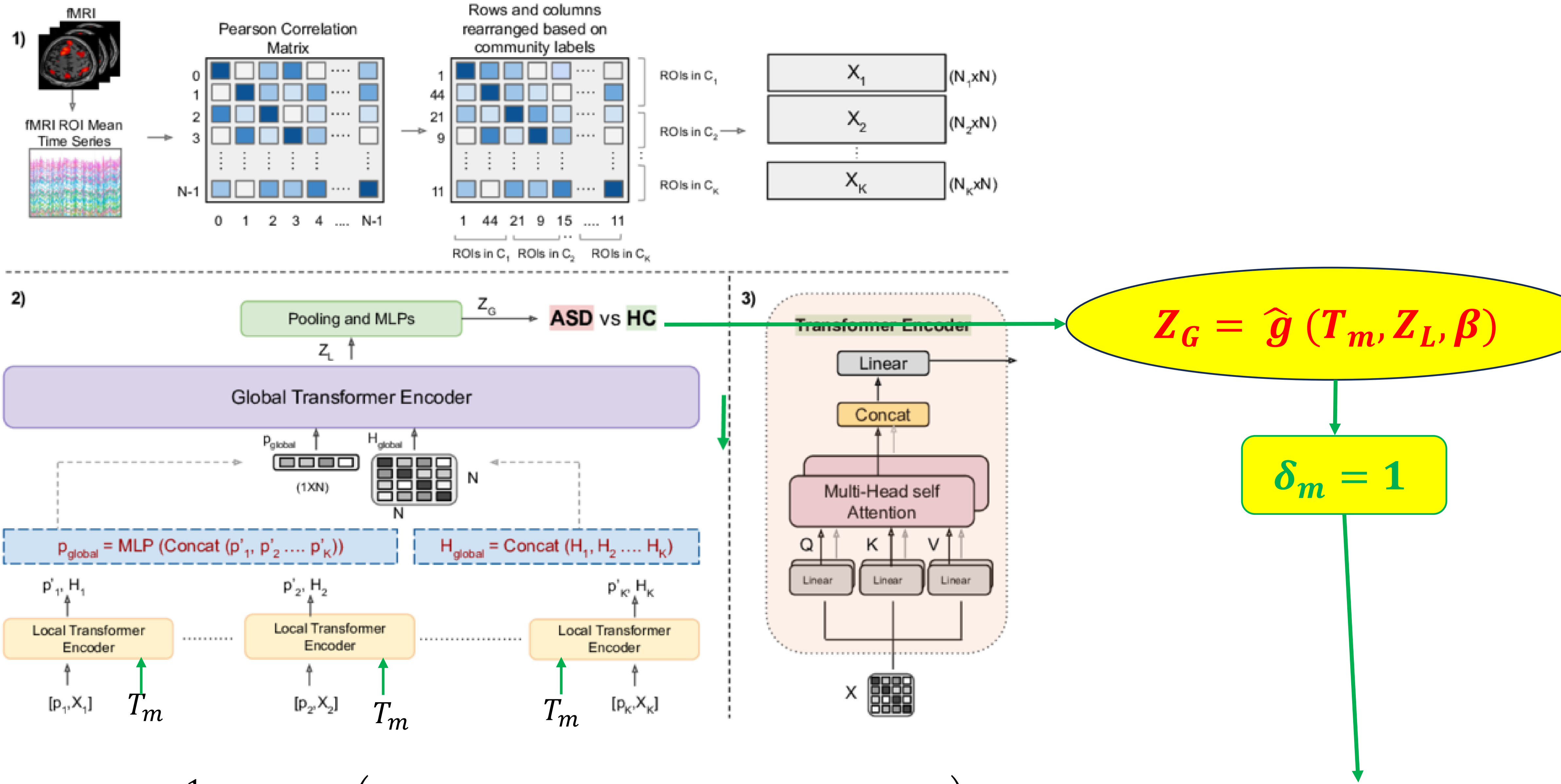
$$\mathcal{L} = \frac{1}{n}\sum_{m:\delta_m=1} \log\left(\sum\nolimits_{j\in\tilde{R}(t_m)} \exp\big(g(T_m, X_j, \beta) - g(T_m, X_m, \beta)\big)\right) + \alpha \sum_{m:\delta_m=1} \sum\nolimits_{j\in\tilde{R}(t_m)} |g(T_m, X_j, \beta)|$$

EEG-based Machine Learning Models for the Prediction of Phenoconversion Time and Subtype in iRBD. Jeong et al. 2023.



$$\mathcal{L} = \frac{1}{n} \sum_{m:\delta_m=1} \log\left(\sum\nolimits_{j\in\tilde{R}(t_m)} \exp\left(g\left(T_m, X_j, \beta\right) - g\left(T_m, X_m, \beta\right)\right)\right) + \alpha \sum_{m:\delta_m=1} \sum\nolimits_{j\in\tilde{R}(t_m)} \left|g\left(T_m, X_j, \beta\right)\right|$$

Data availability statement The data that support the findings of this study are available from the corresponding author upon reasonable request.



$$\mathcal{L} = \frac{1}{n} \sum_{m:\delta_m=1} \log\left(\sum_{j \in \tilde{R}(t_m)} \exp\big(g(T_m, X_j, \beta) - g(T_m, X_m, \beta)\big) \right) + \alpha \sum_{m:\delta_m=1} \sum_{j \in \tilde{R}(t_m)} |g(T_m, X_j, \beta)|$$

Overall Survival Time Prediction of Glioblastoma on Preoperative MRI Using Lesion Network Mapping

Feng Wu
Department of Electronic Engineering and Information Science,
University of Science and Technology of China, Hefei, 230052, China

Data Availability Statement: The dataset generated during and/or analyzed during the current study are not publicly available due to the clinical and confidential nature of the material but can be made available from the corresponding author on reasonable request.

Brain age prediction using fMRI network coupling in youths and associations with 2 psychiatric symptom

Deep multimodal graph-based network for survival prediction from highly multiplexed images and patient variables.

The source code is available at https://github.com/xhelenfu/DMGN_Survival_Prediction.