ML for Electronic Health Records (EHR)

+ ML Products

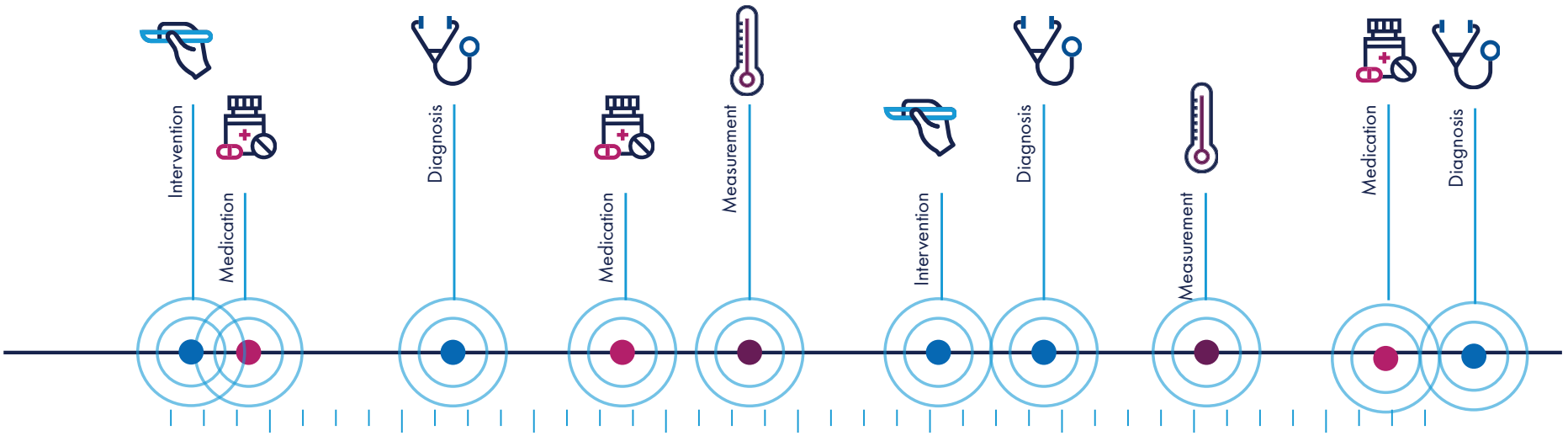
Reza Khorshidi, DPhil (Oxon)

Deep Medicine Program, University of Oxford

@RezaKhorshidi

What is EHR?

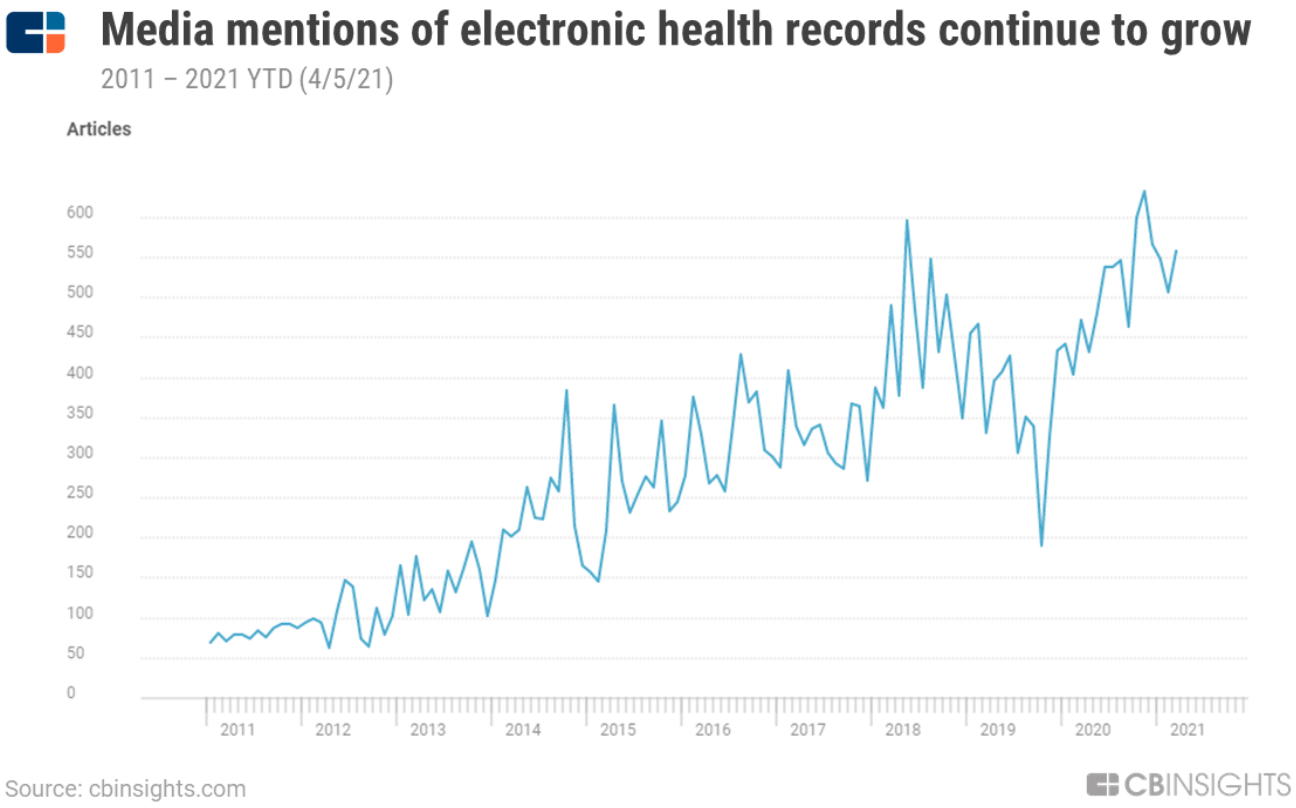
Electronic health records (EHR) contain sequences of mixed-type multimodal concepts (or tokens) that occur in irregular intervals, and show long-range dependencies.



EHR offers a unique opportunity for pre-clinical ML

During the last few years, we saw a dramatic growth in EHR adoption. Despite all of its shortcomings (e.g., missing data, biases, inferior UX, and more), EHR can provide a unique window into policy and practice of care — both personal and population-wide.

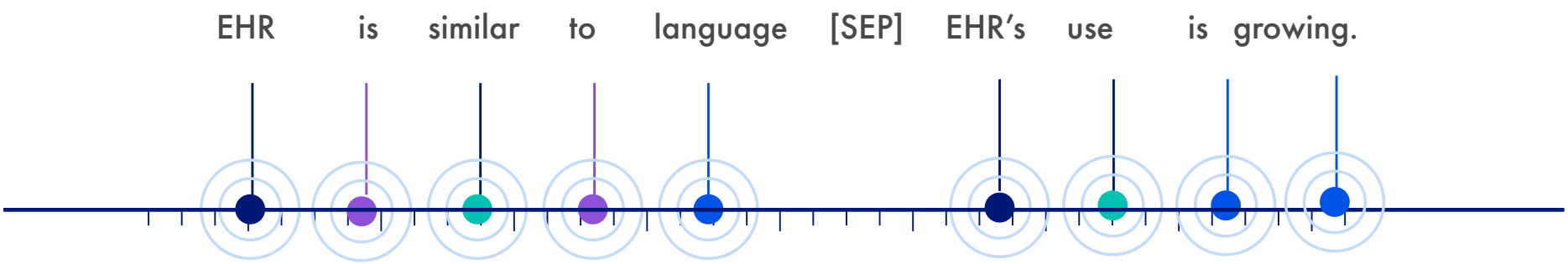
If the goal is to plug an ML model into a medical system, EHR provides a unique window into how the system operates.



EHR is similar to many other forms of real-world data

Similarities between EHR and NLP, for instance, inspired many developments in ML+EHR.

ML has been employed in NLP due to its ability to deal with a sequence of tokens; thanks to the latest developments in neural sequence models. While there are many differences between EHR and language, one can assume that EHR is also a sequence of tokens and has take inspirations from other domains.



Section I

Problem Formulation

AIG

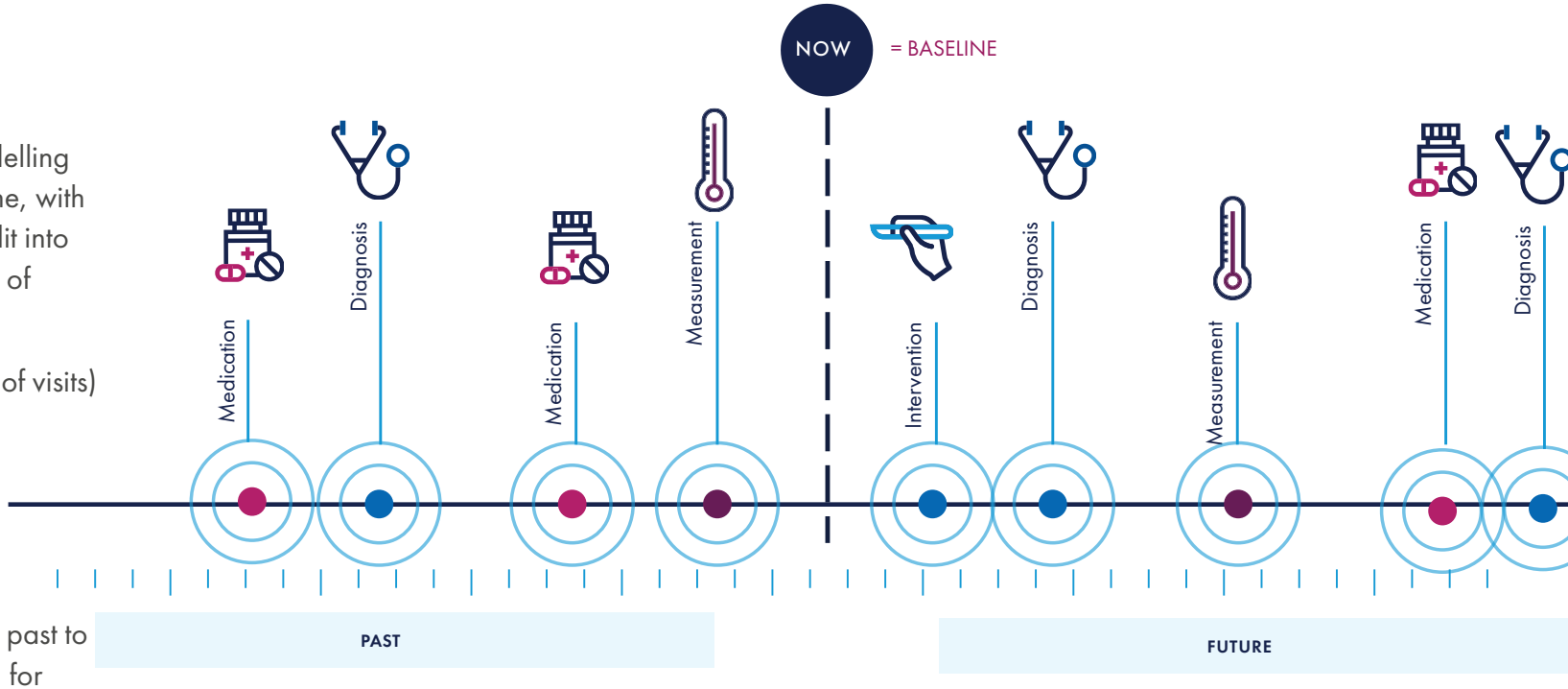
Machine learning and EHR — baseline

$$\mathscr{D} = \left\{\mathbf{x_i}, y_i\right\}_{i=1}^{N}$$

Early/common approaches for modelling EHR data rely on choosing a baseline, with respect to which, the data will be split into past and future. Common definitions of baseline can be:

- A data quality threshold (number of visits)
- Age
- An event (eg, a diagnosis)
- A date (01/01/2007)

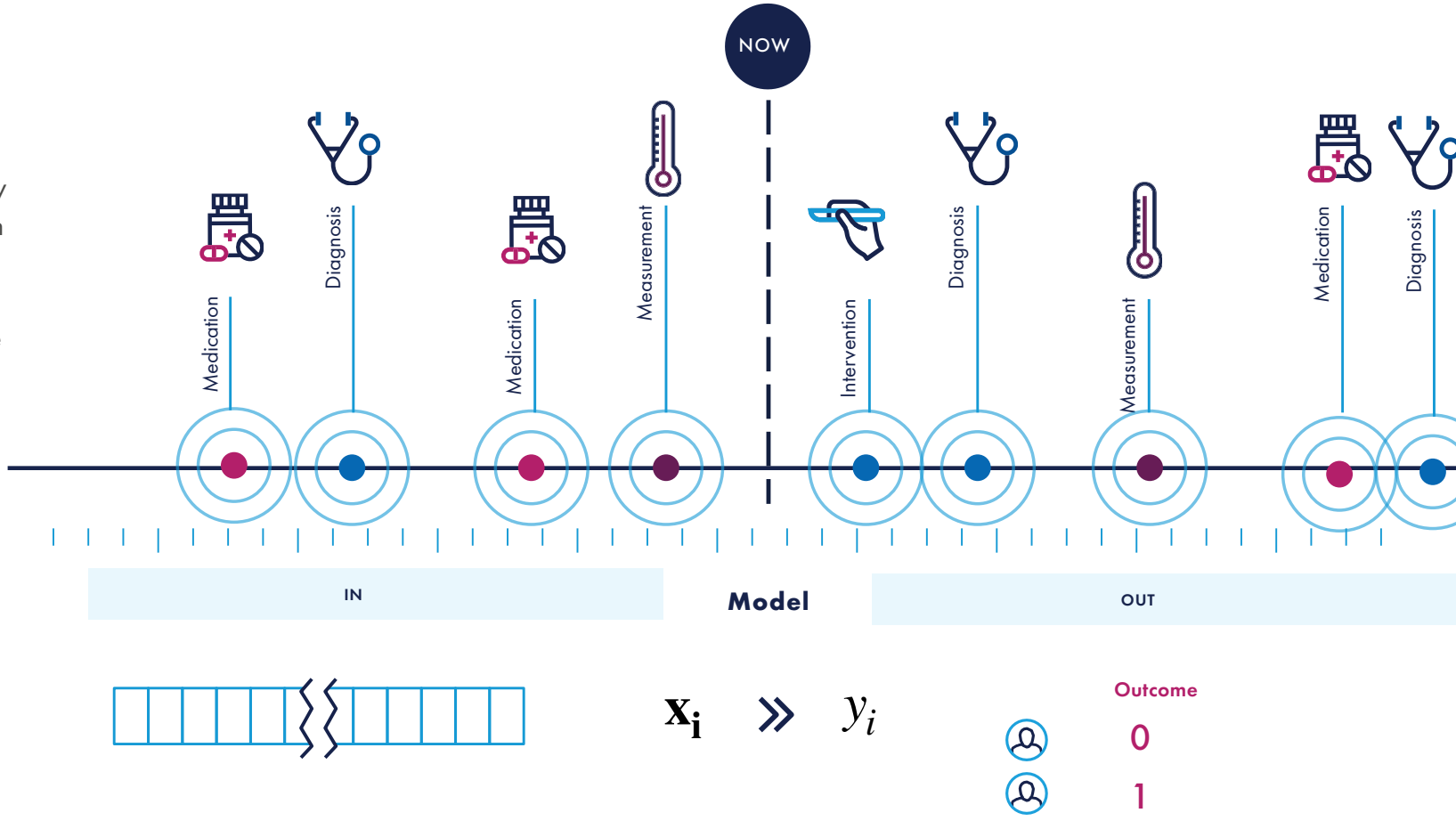
While we commonly aim to map the past to future, we also use "the past" alone, for various phenotyping and data mining works.



Machine learning and EHR — classification

$$\mathscr{D} = \left\{\mathbf{x_i}, y_i\right\}_{i=1}^{N}$$

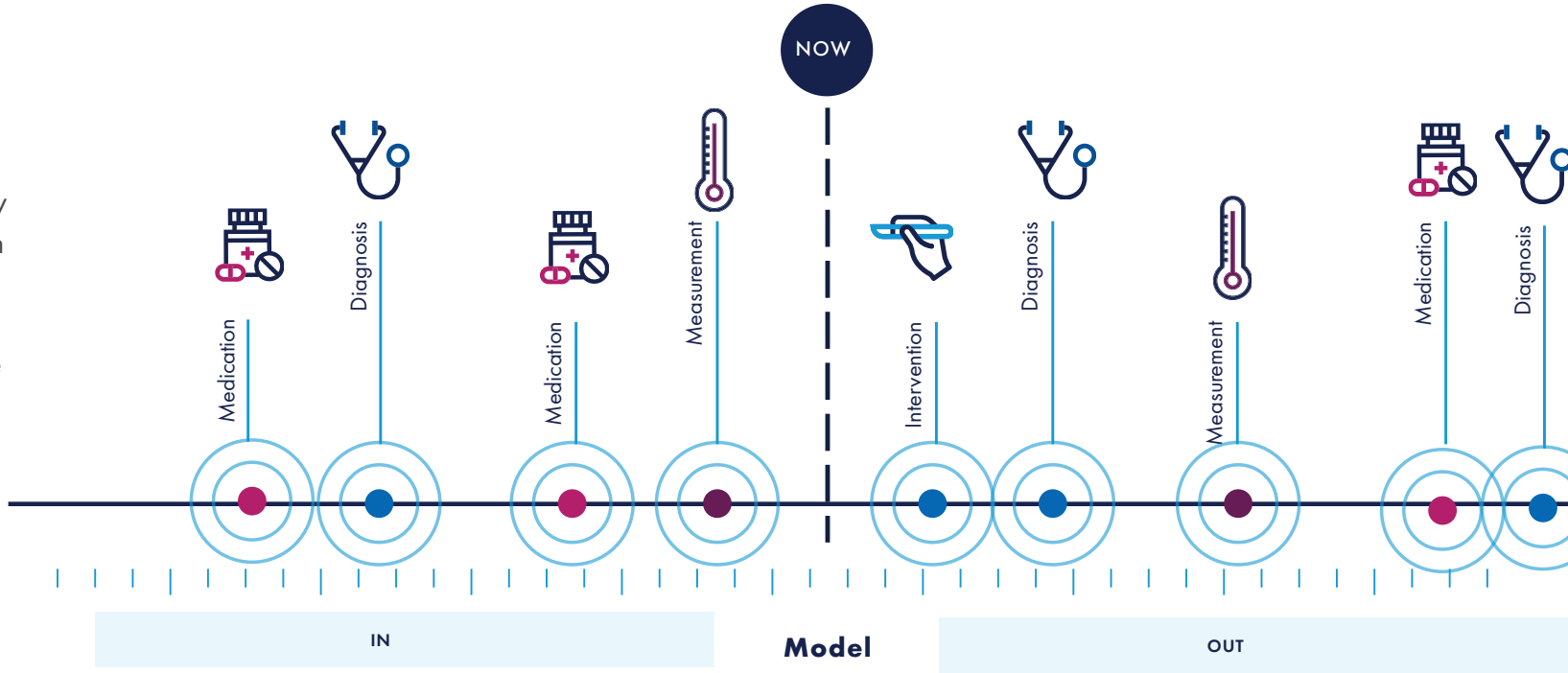
Modelling medical data in medicine goes way back — many well-established methods for both observational and interventional studies. Therefore, in a first attempt, such approaches can be applied to EHR data.

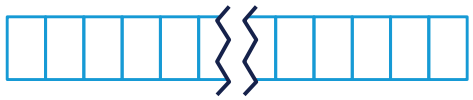


Machine learning and EHR — regression

$$\mathscr{D} = \left\{\mathbf{x_i}, y_i\right\}_{i=1}^{N}$$

Modelling medical data in medicine goes way back — many well-established methods for both observational and interventional studies. Therefore, in a first attempt, such approaches can be applied to EHR data.





$$\mathbf{x_i} \gg y_i$$

Outcome

0.1

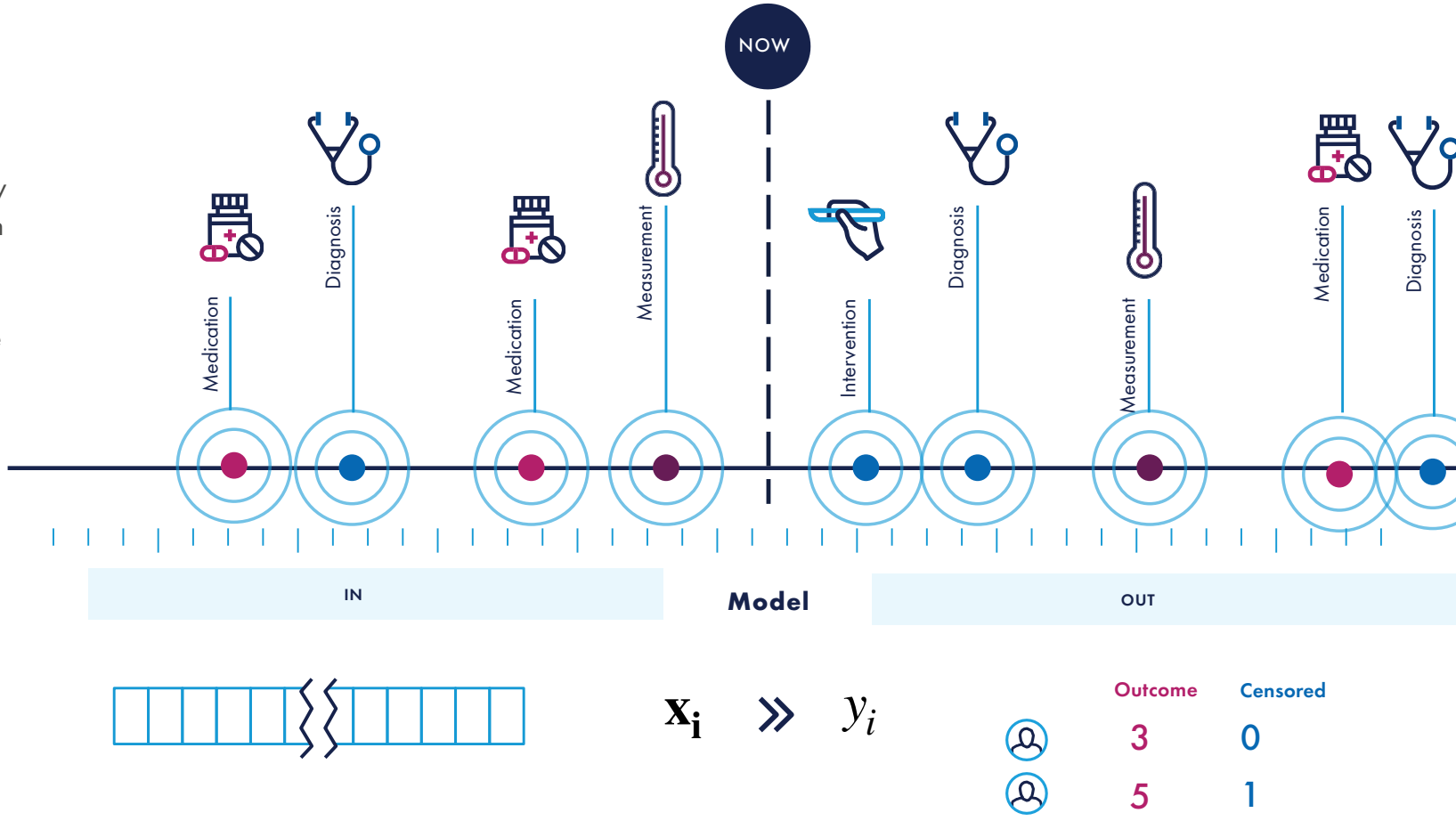
-2

1.3

Machine learning and EHR — survival

$$\mathscr{D} = \left\{\mathbf{x_i}, (y_i, c_i)\right\}_{i=1}^{N}$$

Modelling medical data in medicine goes way back — many well-established methods for both observational and interventional studies. Therefore, in a first attempt, such approaches can be applied to EHR data.



Example I: Atrial fibrillation and vascular risk

Using the EHR to study this, we observed that baseline AF was associated with a 31% higher risk of any vascular event (HR 1.31, CI 1.28, 1.34) and a 89% higher risk of a fatal vascular event (HR 1.89, CI 1.81, 1.96).

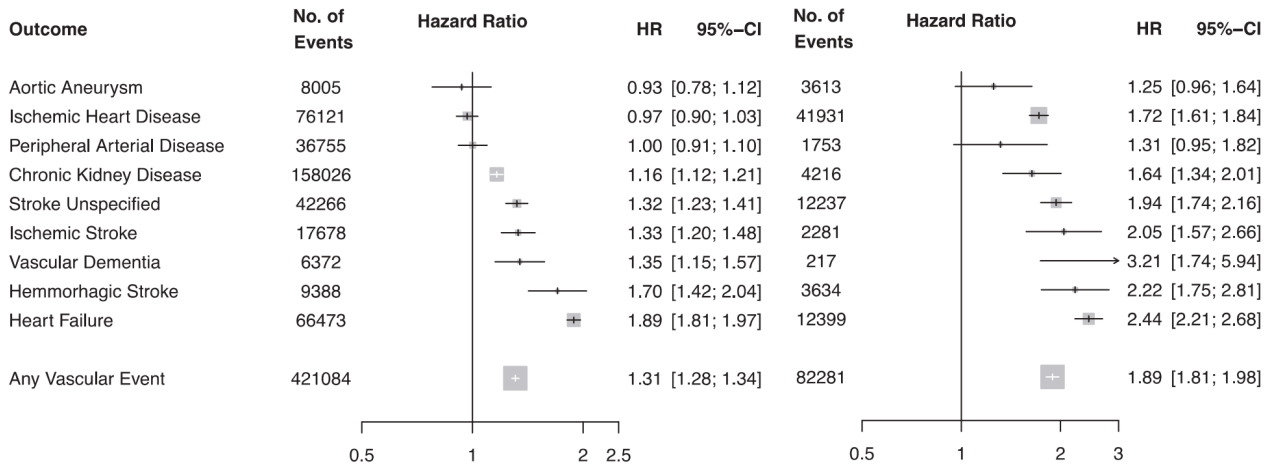


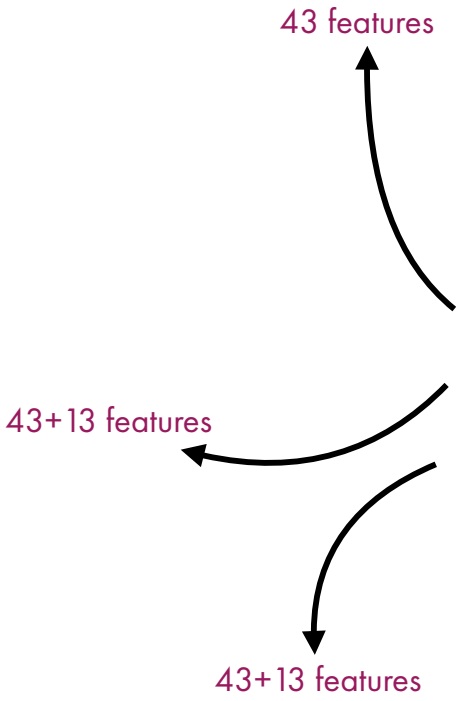
Figure 3 Adjusted hazard ratios of baseline atrial fibrillation for nine different vascular events. Adjustments were for age, BMI, smoking status, sex, baseline diabetes, baseline antihypertensive use, baseline lipid-lowering drug (statin) use, baseline anticoagulant usage, baseline antiplatelet usage and baseline atrial fibrillation (plotted). Restricted to (A) fatal and non-fatal vascular events; and (B) only fatal vascular events. Area of each square is proportional to the inverse variance of the estimate.

Emdin et al 2017

Example II: Risk of emergency readmission

In this study, we see how improvement along the "better model" and "better representation" axes, can lead to 10% improvement (in AUC terms) in model results.

This shows the importance of selecting important features.

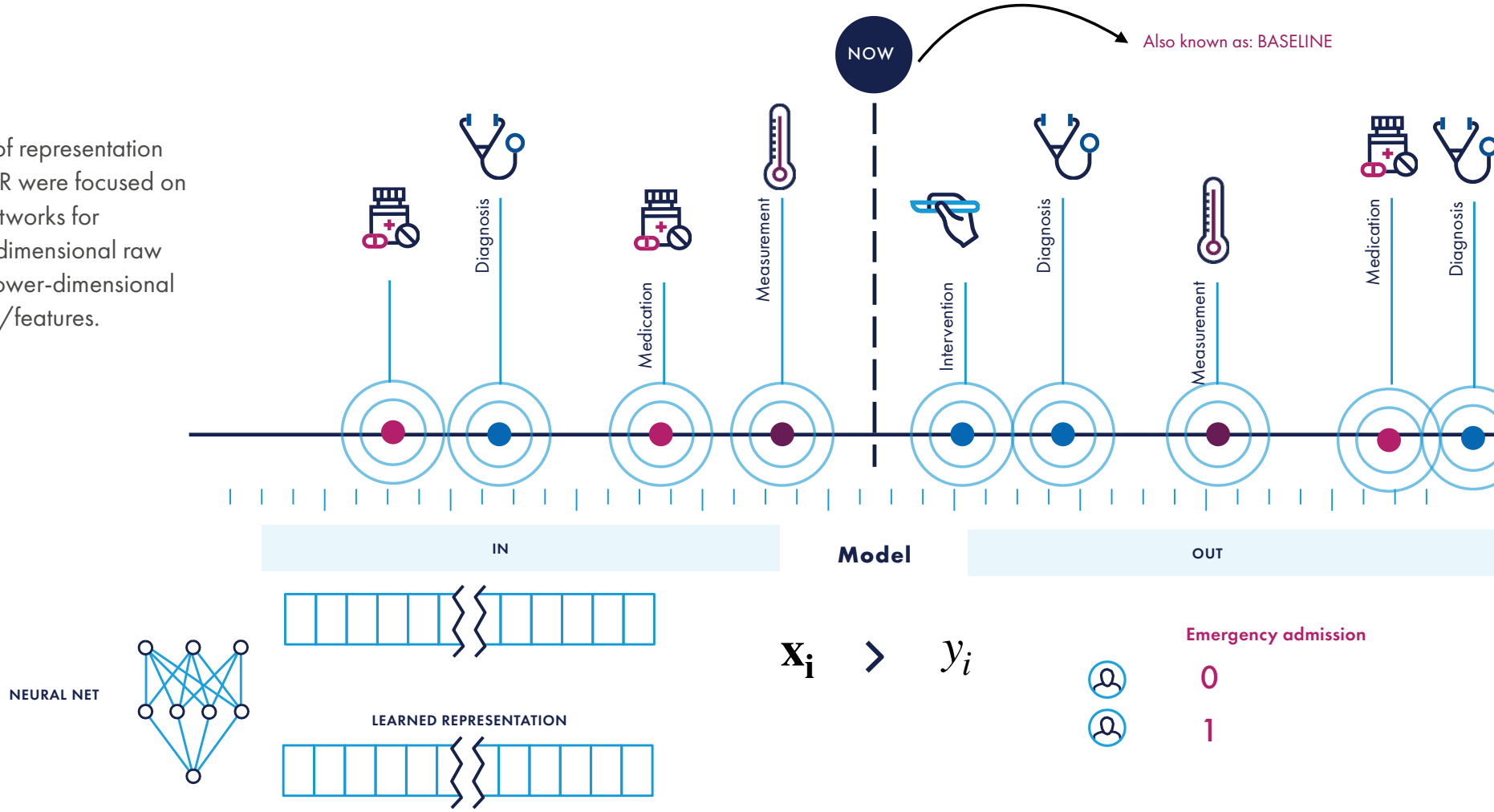


Predictors	Model (AUC)		
	CPH	RF	GBC
QA	0.736	0.736	0.796
QA+	0.743	0.799	0.810
T	0.788	0.810	0.826

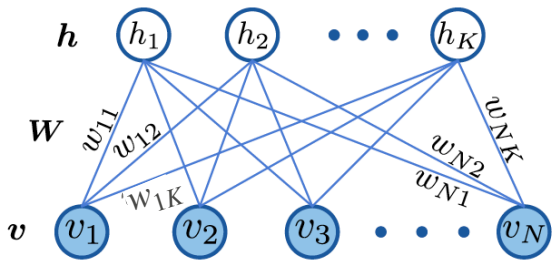
Rahimian et al 2018

Given the complexities of EHR, representation learning is expected to play an important role.

The early uses of representation learning for EHR were focused on using neural networks for mapping high-dimensional raw input data, to lower-dimensional representations/features.



Representation learning using DNNs — RBM



$$E(v,h) = -\sum_i a_i v_i - \sum_j b_j h_j - \sum_i \sum_j v_i w_{i,j} h_j$$

$$P(v,h) = \frac{1}{Z} e^{-E(v,h)} \qquad P(v) = \frac{1}{Z} \sum_h e^{-P(v,h)}$$

$$\arg\max_W \prod_{v \in V} P(v)$$

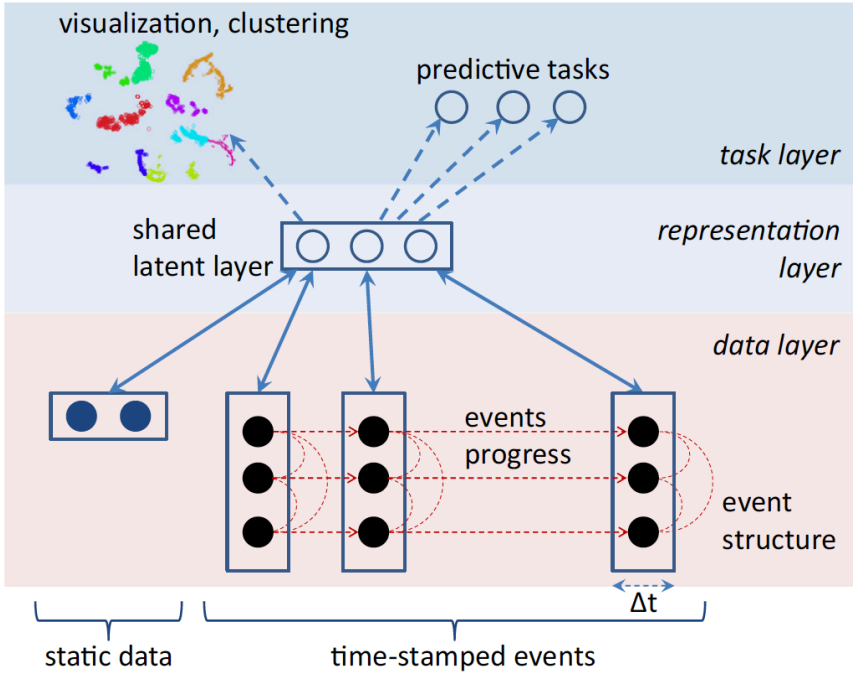
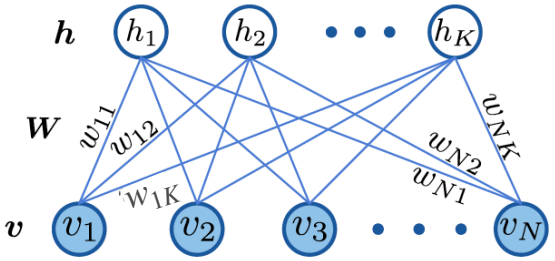


Fig. 1. *e*NRBM for EMR modeling, visualization and prognosis. The data layer represents raw information extracted from EMR; the representation layer exhibits higher-level semantics; and the task layer makes use of the derived representation for tasks of interest. The connections between the data and representation layers are undirected, letting patterns emerge through information passing in both directions. Filled nodes represent observed variables, empty nodes the hidden. Boxes represent groups of variables that share the same property (e.g., time interval). Event structures and progression (represented as thin dashed lines and curves) are implicitly captured through regularization in the learning process

Tran et al 2015

Representation learning using DNNs — RBM



$$v_i \rightarrow [w_{i1}, w_{i2}, \ldots, w_{iK}]$$

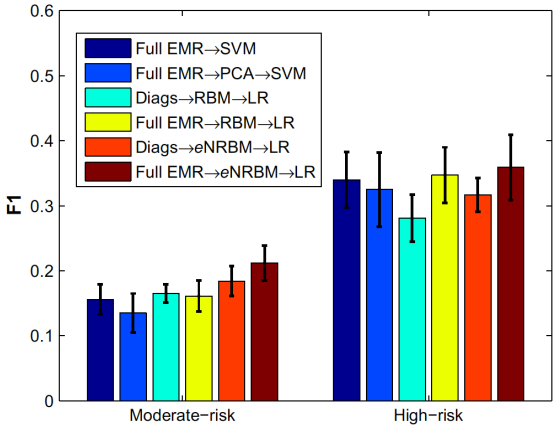
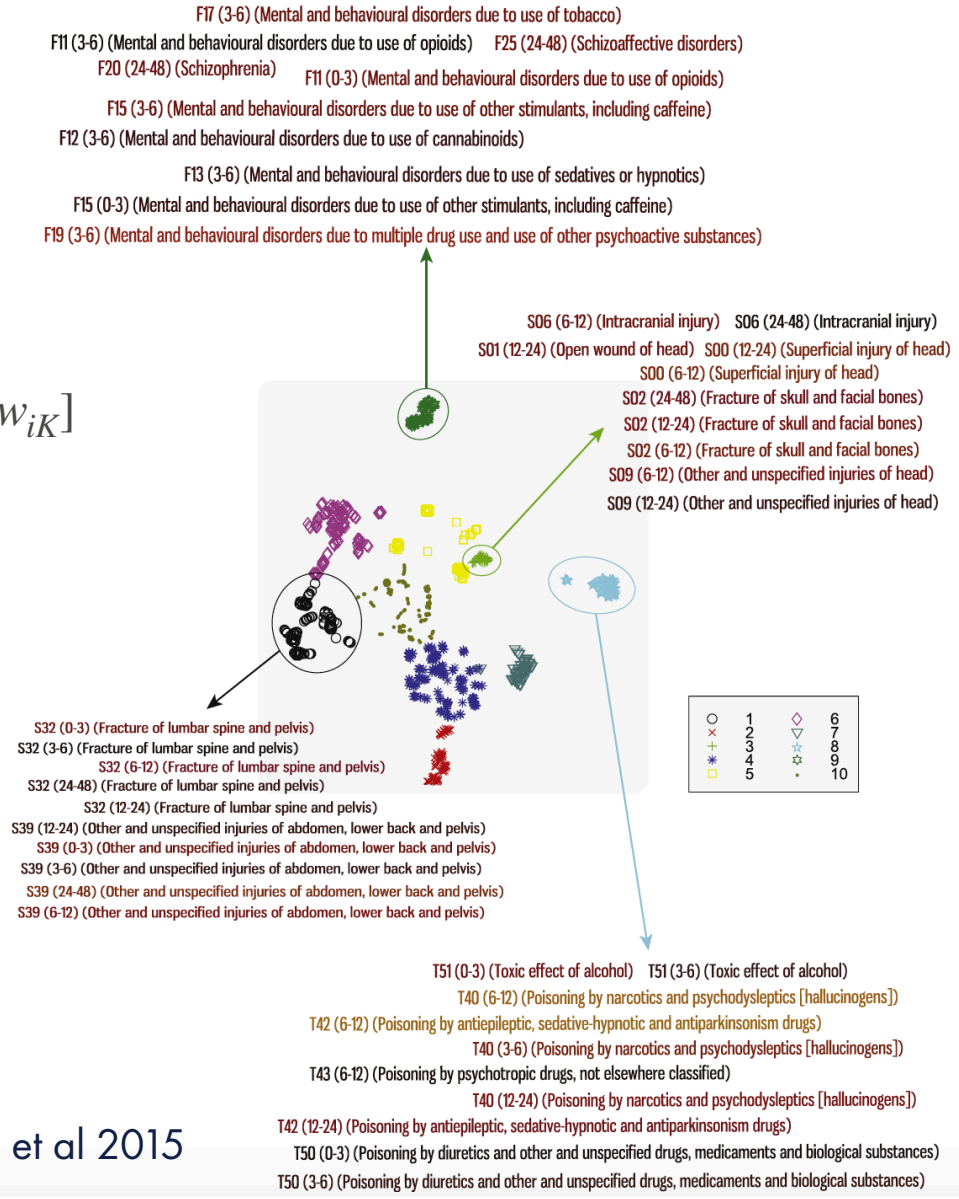


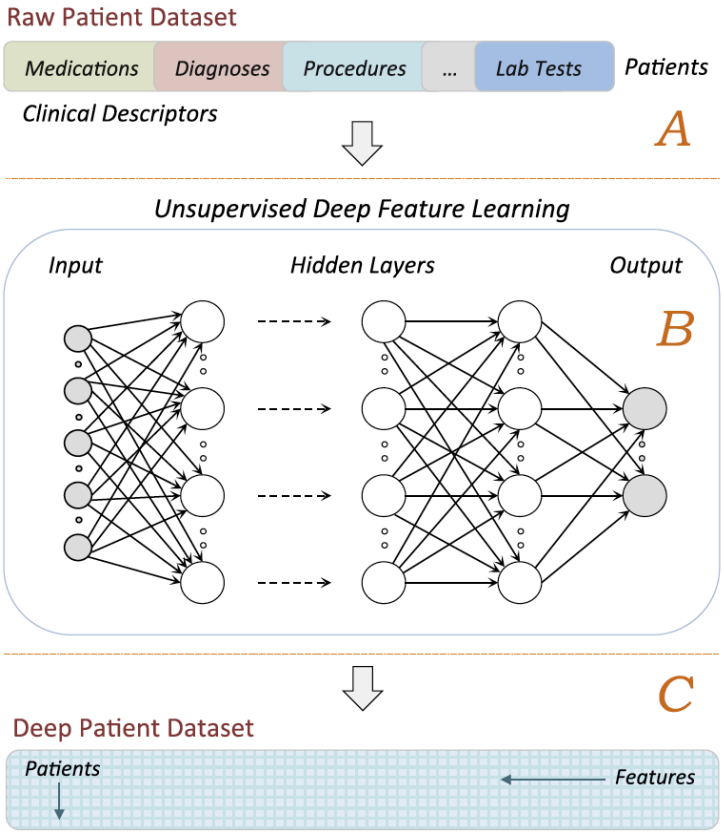
Fig. 7. *F*-scores (F_1) for moderate and high-risk within 3 months. Arrows indicate the flow. *Diags* means using only diagnoses as input. Full EMR contains demographics, diagnoses, procedures, diagnosis related groups (DRG) and Elixhauser comorbidities [2].



Tran et al 2015

Representation learning using DNNs — SDA

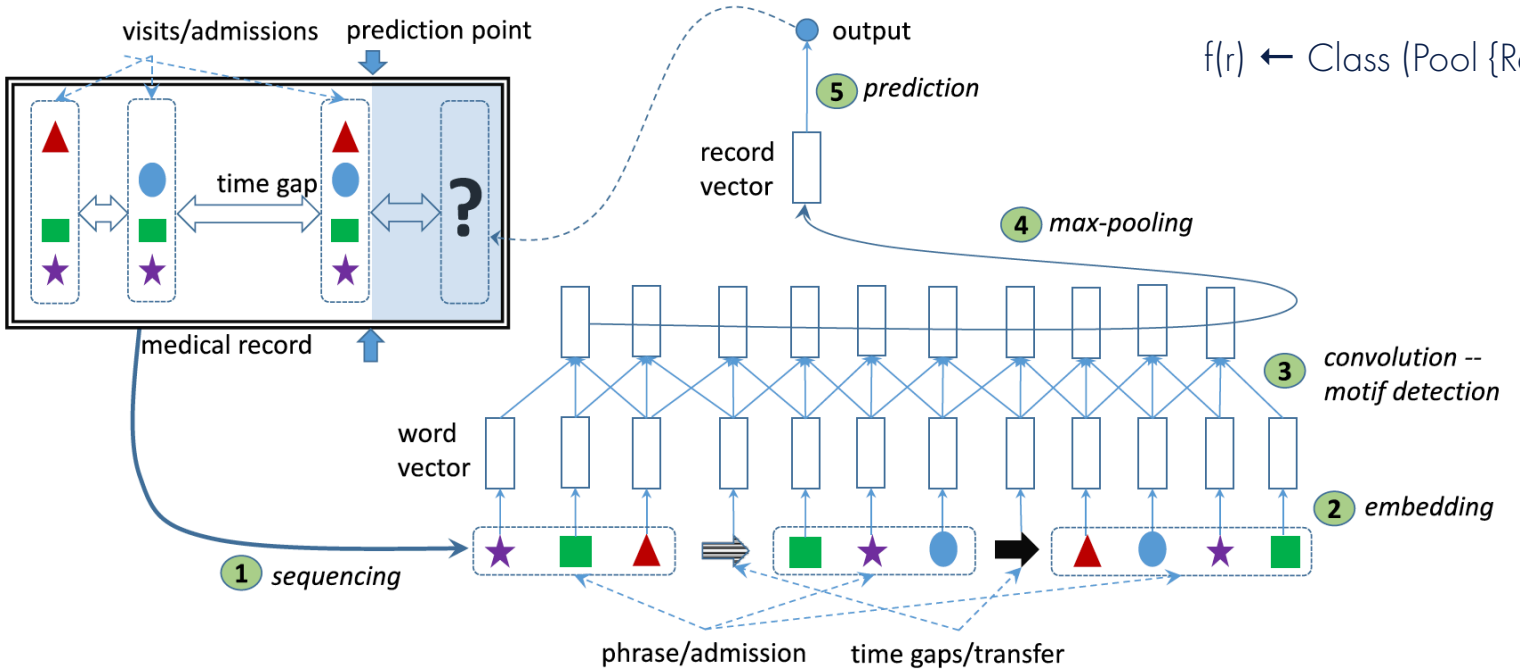
The early uses of representation learning for EHR were focused on using neural networks for mapping high-dimensional raw input data, to lower-dimensional representations/features.



Time Interval = 1 year (76,214 patients)			
	Area under the ROC curve		
Disease	RawFeat	PCA	DeepPatient
Diabetes mellitus with complications	0.794	0.861	**0.907**
Cancer of rectum and anus	0.863	0.821	**0.887**
Cancer of liver and intrahepatic bile duct	0.830	0.867	**0.886**
Regional enteritis and ulcerative colitis	0.814	0.843	**0.870**
Congestive heart failure (non-hypertensive)	0.808	0.808	**0.865**
Attention-deficit and disruptive behavior disorders	0.730	0.797	**0.863**
Cancer of prostate	0.692	0.820	**0.859**
Schizophrenia	0.791	0.788	**0.853**
Multiple myeloma	0.783	0.739	**0.849**
Acute myocardial infarction	0.771	0.775	**0.847**

Miotto et al 2016

Representation learning using DNNs — CNN plus time tokens



f(r) ← Class (Pool {ReLU (Conv [Embed {Seq (r)}]]}}

Nguyen et al 2017

Fig. 1. Overview of Deepr for predicting future risk from medical record. Top-left box depicts an example of medical record with multiple visits, each of which has multiple coded objects (diagnosis and procedure). The future risk is unknown [question mark (?)]. *Steps from-left-to-right*: (1) Medical record is sequenced into phrases separated by coded time-gaps/transfers; then *from-bottom-to-top*: (2) Words are embedded into continuous vectors, (3) local word vectors are convoluted to detect local motifs, (4) max-pooling to derive record-level vector, (5) classifier is applied to predict an output, which is a future event. Best viewed in color.

Representation learning using DNNs — CNN plus time tokens

```
  1910 Z83 911 1008 D12 K31 1-3m R94
RAREWORD H53 Y83 M62 Y92 E87 T81
RAREWORD RAREWORD 1893 D12 S14 738
1910 1916 Z83 0-1m T91 RAREWORD Y83
Y92 K91 M10 E86 6-12m K31 1008 1910
Z13 Z83.
```

Nguyen et al 2017

TABLE I
AUC MEASURES ON THRE-MONTH AND SIX-MONTH UNPLANNED READMISSION PREDICTION FOLLOWING A RANDOM INDEX DISCHARGE WITH AND WITHOUT TIME-GAPS

	3 months		6 months	
Method	W/o time	With time	W/o time	With time
BoW + LR	0.786	0.797	0.797	0.811
`Deepr` (rand init)	**0.791**	**0.797**	**0.806**	**0.814**
`Deepr` (*word2vec*)	**0.795**	**0.800**	**0.809**	**0.819**

Rand init refers to random initialization of the embedding matrix. *Word2vec* init refers to pretraining the embedding matrix using the *word2vec* algorithm

From DNN to SEQ2SEQ models - *a la* NLP

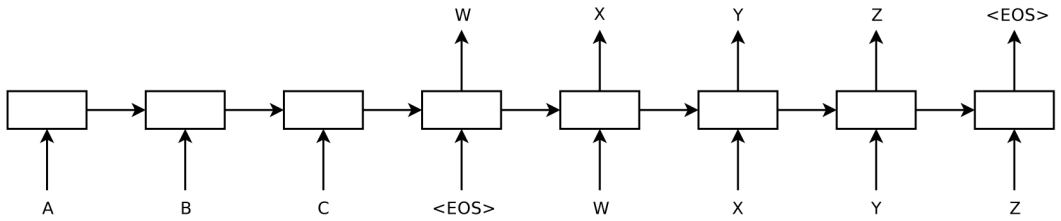


Figure 1: Our model reads an input sentence "ABC" and produces "WXYZ" as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

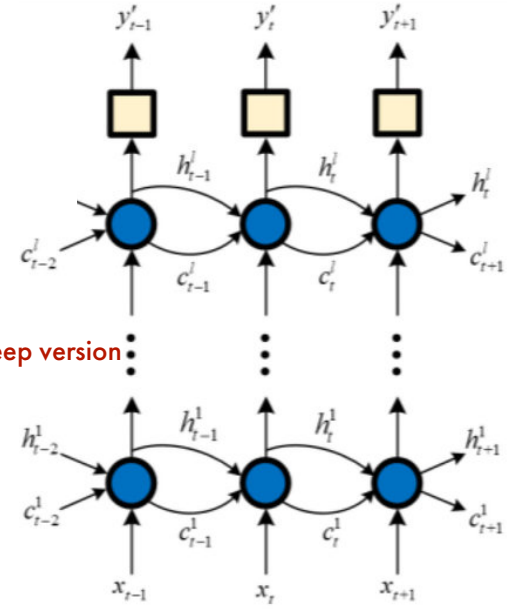
Unlike DNN, SEQ2SEQ models can deal with variable-length inputs and outputs.

They "encode a variable-length sequence into a fixed-length vector representation; decode a fixed-length vector representation into a variable-length sequence"

$$p(y_1, \ldots, y_{T'} | x_1, \ldots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \ldots, y_{t-1})$$

training objective: $1/|\mathcal{S}| \sum_{(T,S)\in\mathcal{S}} \log p(T|S)$

translation/inference: $\hat{T} = \arg\max_T p(T|S)$



A deep version

Sutskever et al 2016

Bidirectional RNN with attention mechanism — RETAIN

RETAIN consists of two RNNs — one looking forward and the other backward — generating alpha and beta outputs. The attention mechanism learns how to pool these value to generate the context vector for prediction.

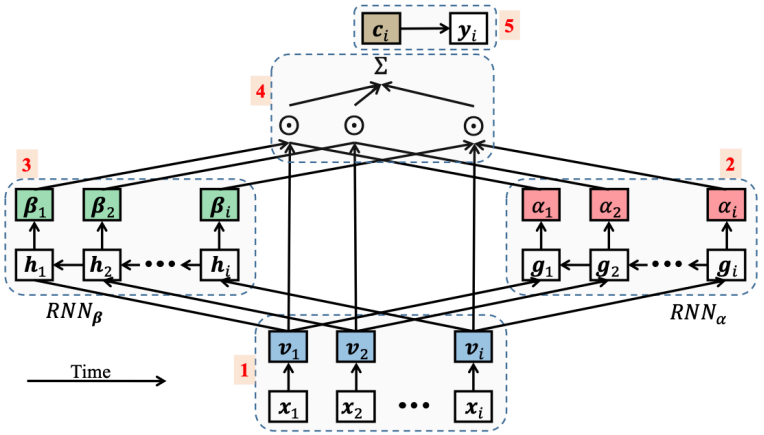


Figure 2: Unfolded view of RETAIN's architecture: Given input sequence $\mathbf{x}_1, \ldots, \mathbf{x}_i$, we predict the label $\mathbf{y}_i$. **Step 1**: Embedding, **Step 2**: generating α values using RNN_α, **Step 3**: generating β values using RNN_β, **Step 4**: Generating the context vector using attention and representation vectors, and **Step 5**: Making prediction. Note that in Steps 2 and 3 we use RNN in the reversed time.

Table 2: Heart failure prediction performance of RETAIN and the baselines

Model	Test Neg Log Likelihood	AUC	Train Time / epoch	Test Time
LR	0.3269 ± 0.0105	0.7900 ± 0.0111	0.15s	0.11s
MLP	0.2959 ± 0.0083	0.8256 ± 0.0096	0.25s	0.11s
RNN	0.2577 ± 0.0082	0.8706 ± 0.0080	10.3s	0.57s
RNN+α_M	0.2691 ± 0.0082	0.8624 ± 0.0079	6.7s	0.48s
RNN+α_R	0.2605 ± 0.0088	$\mathbf{0.8717} \pm 0.0080$	10.4s	0.62s
RETAIN	$\mathbf{0.2562} \pm 0.0083$	0.8705 ± 0.0081	10.8s	0.63s

Choi et al 2016

Section II

EHR + ML for Risk Prediction

AIG

Clinical practice research datalink (CPRD)

1,100 GP practices

50M patients

14M currently registered;

75% have 20+yr follow up.

30+ years (since 1985)

2,500+ peer-reviewed publications

Representation learning in EHR — a review

Many ML+EHR works use different datasets and models, to predict different outcomes. Therefore, an apple vs apple comparison is needed.

Table 7
Comparison for the Demographics + Diagnoses + Medications scenario (Emergency Admission).

Model	AUROC	AUPRC	F1-Score
eNRBM	0.831 (0.831–0.832)	0.071 (0.071–0.071)	0.063 (0.062–0.063)
Deep Patient	0.813 (0.813–0.813)	0.060 (0.060–0.061)	0.059 (0.059–0.059)
Deepr	0.829 (0.828–0.831)	0.069 (0.067–0.071)	0.131 (0.118–0.144)
RETAIN	**0.847 (0.845–0.849)**	**0.083 (0.082–0.083)**	**0.153 (0.151–0.154)**
BOW + LR	0.646 (0.576–0.717)	0.019 (0.015–0.023)	0.054 (0.046–0.063)
RBM	0.840 (0.840–0.840)	0.072 (0.072–0.073)	0.066 (0.066–0.066)

Table 8
Comparison for the Demographics + Diagnoses + Medications scenario (Heart Failure).

Model	AUROC	AUPRC	F1-Score
eNRBM	0.920 (0.920–0.921)	0.020 (0.019–0.021)	0.014 (0.014–0.014)
Deep Patient	0.947 (0.947–0.948)	0.040 (0.039–0.041)	0.023 (0.022–0.023)
Deepr	0.949 (0.947–0.952)	0.039 (0.032–0.046)	0.085 (0.049–0.120)
RETAIN	**0.950 (0.946–0.954)**	**0.054 (0.053–0.056)**	**0.117 (0.098–0.136)**
BOW + LR	0.682 (0.613–0.752)	0.006 (0.002–0.009)	0.019 (0.011–0.027)
RBM	0.917 (0.917–0.917)	0.023 (0.022–0.023)	0.014 (0.014–0.014)

Ayala Solares et al 2020

Transformer, BERT, ...

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

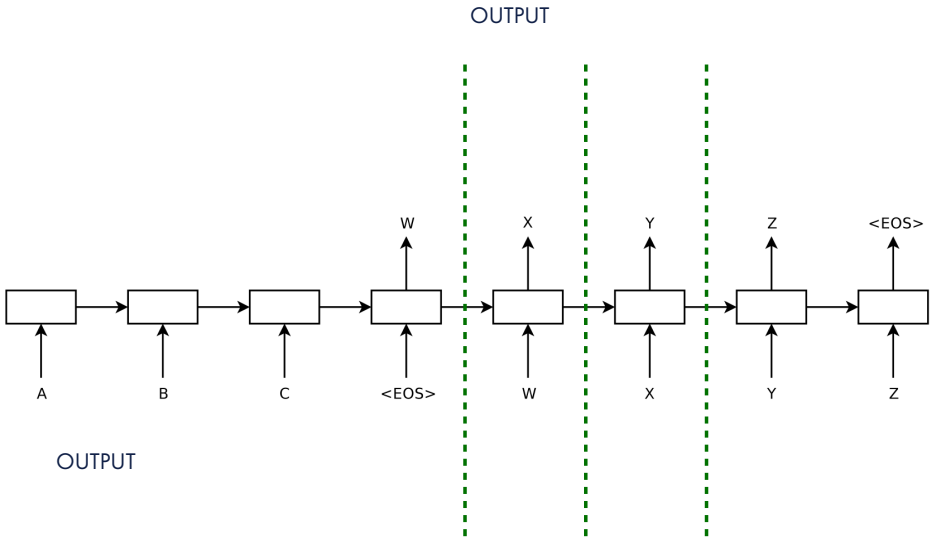
2017

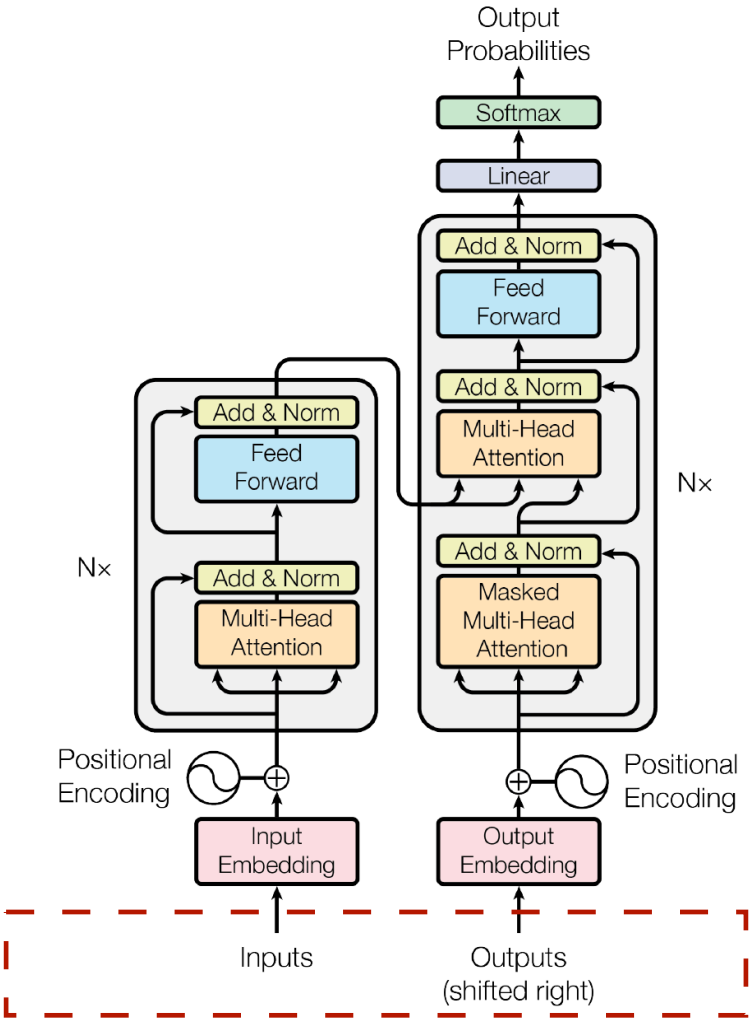
BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

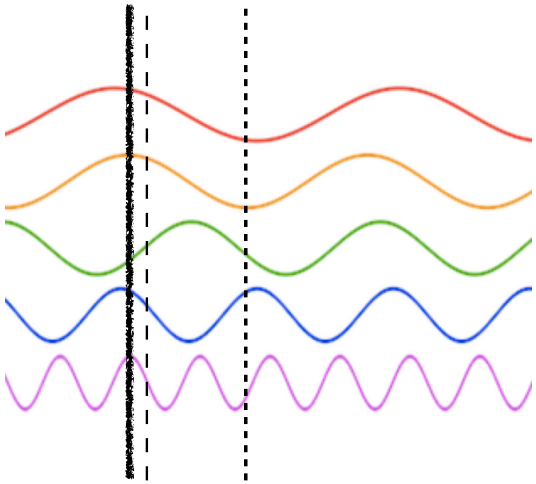
2019

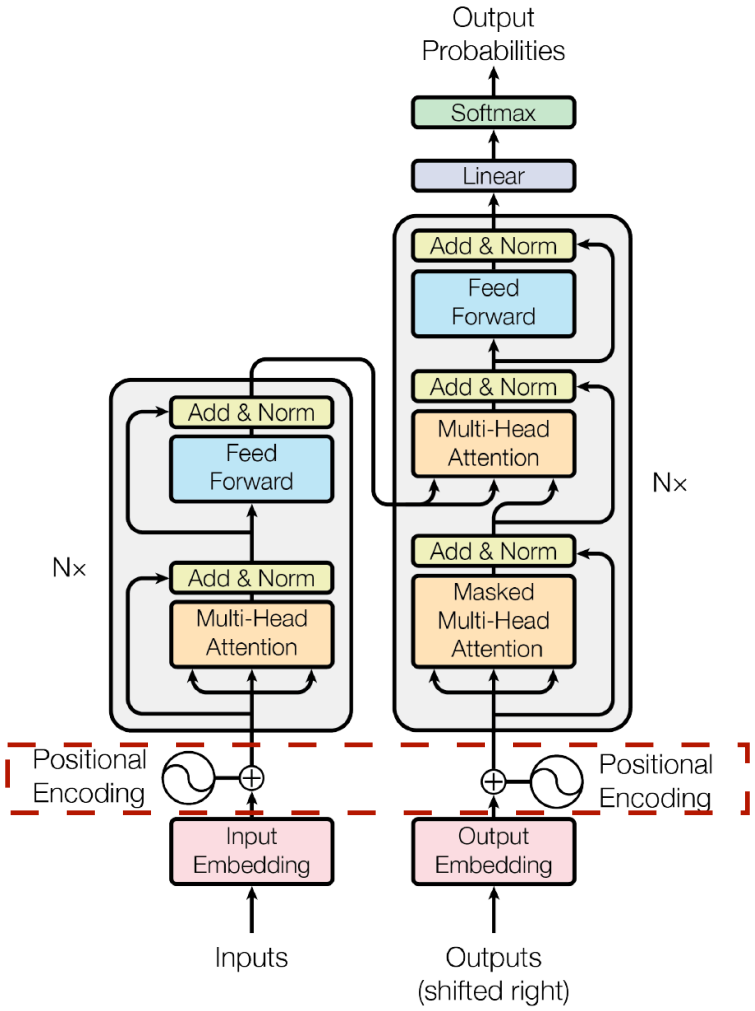
Transformer, BERT, ...



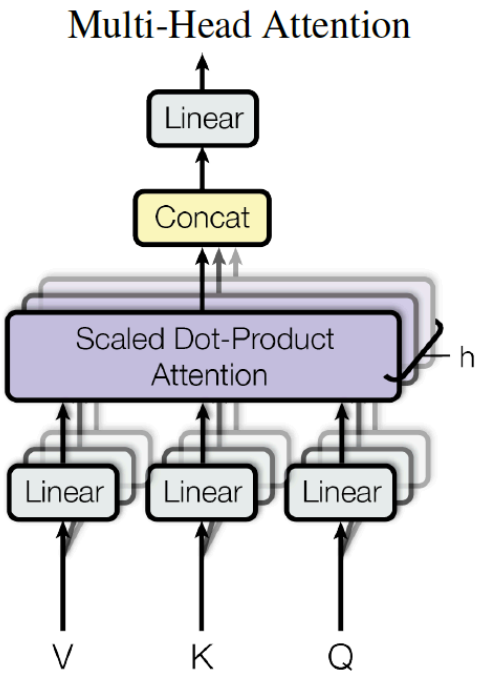


Transformer, BERT, ...



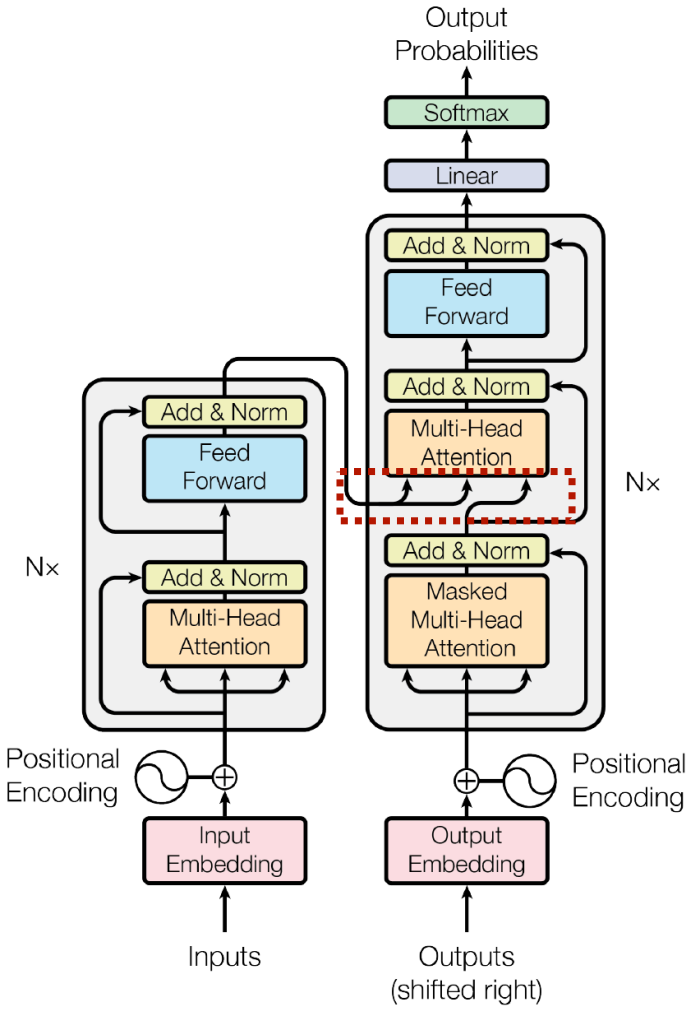


Transformer, BERT, ...

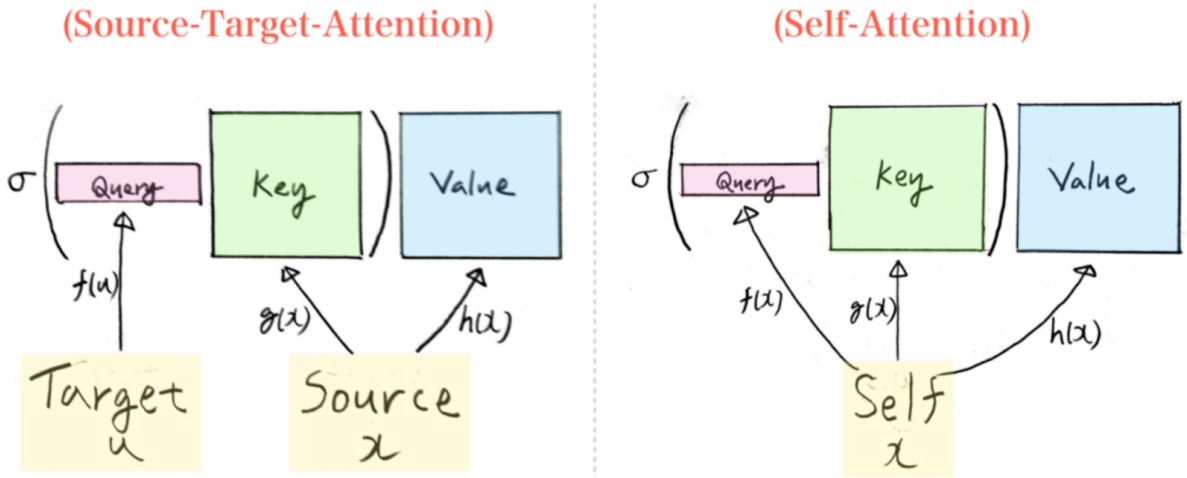


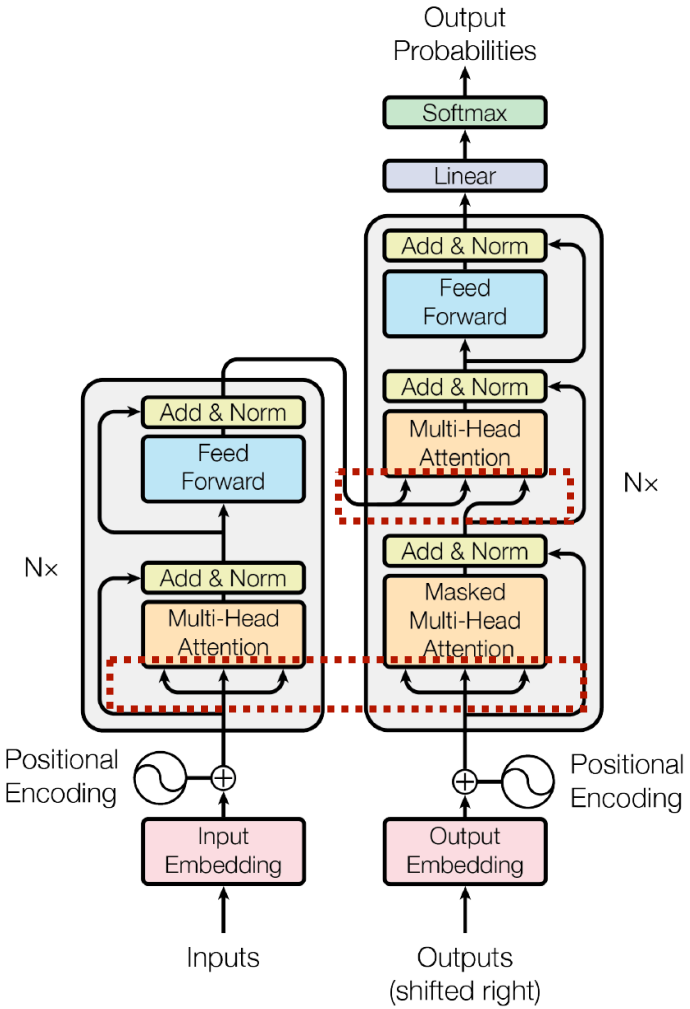
$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$



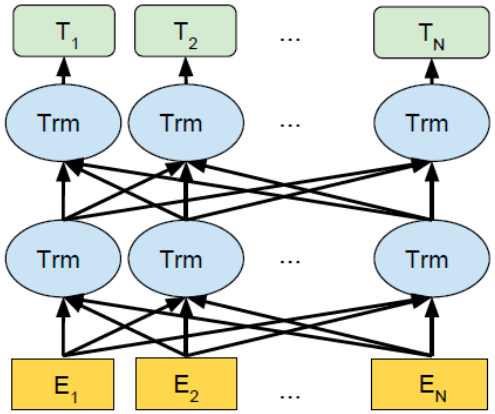
Transformer, BERT, ...

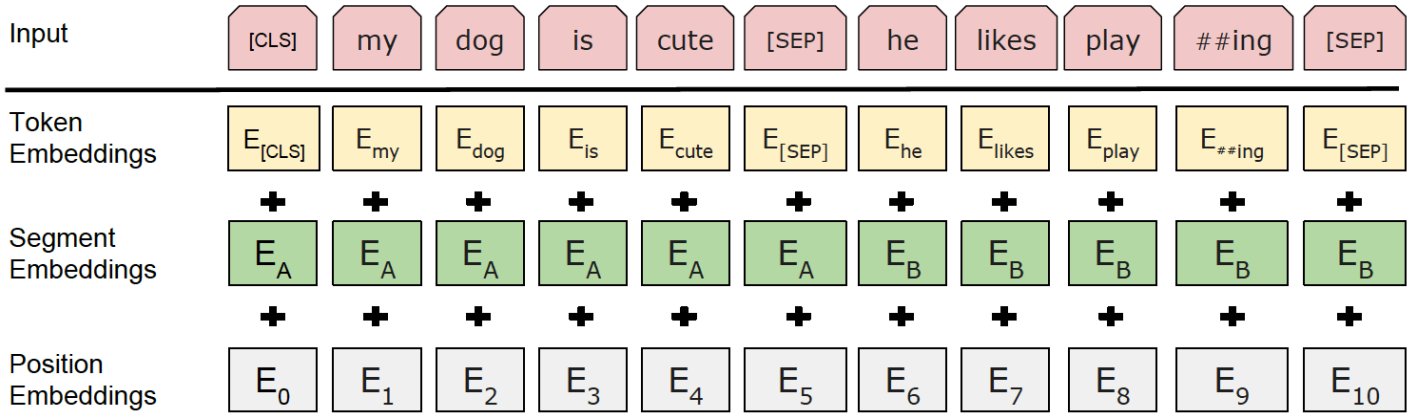




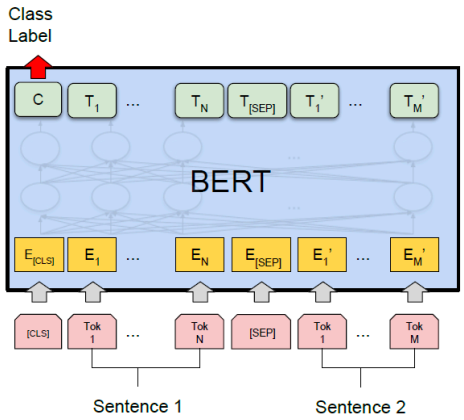
http://jalammar.github.io/illustrated-transformer

Transformer, BERT, ...

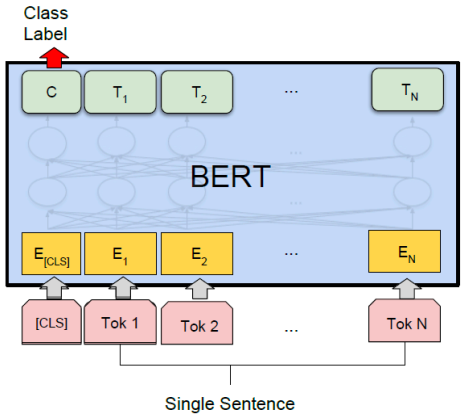




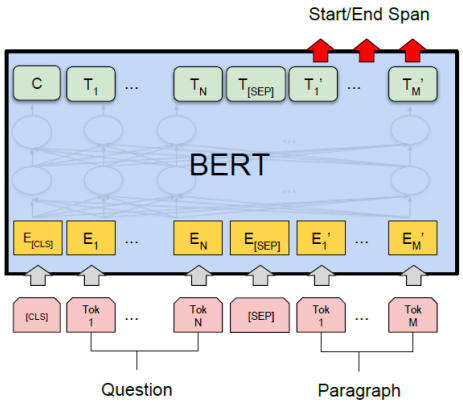
Transformer, BERT, ...



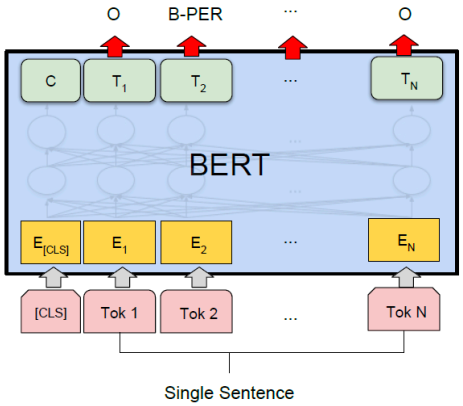
(a) Sentence Pair Classification Tasks: MNLI, QQP, QNLI, STS-B, MRPC, RTE, SWAG



(b) Single Sentence Classification Tasks: SST-2, CoLA

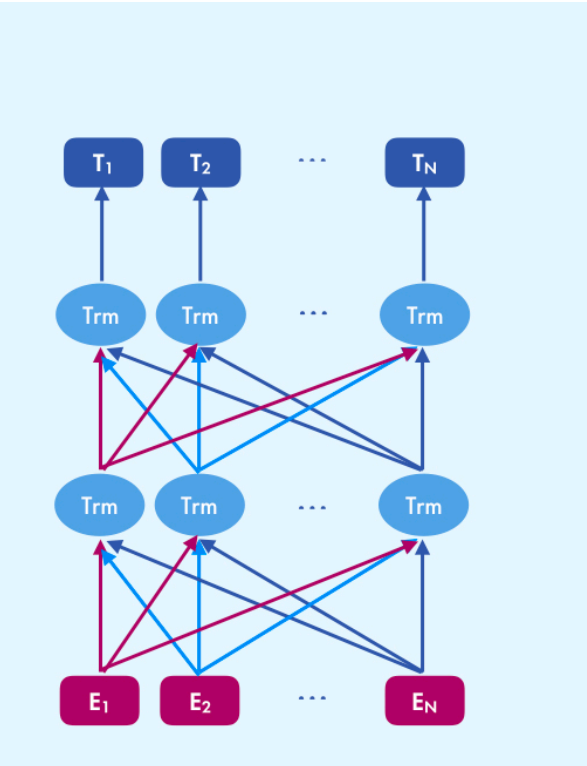


(c) Question Answering Tasks: SQuAD v1.1



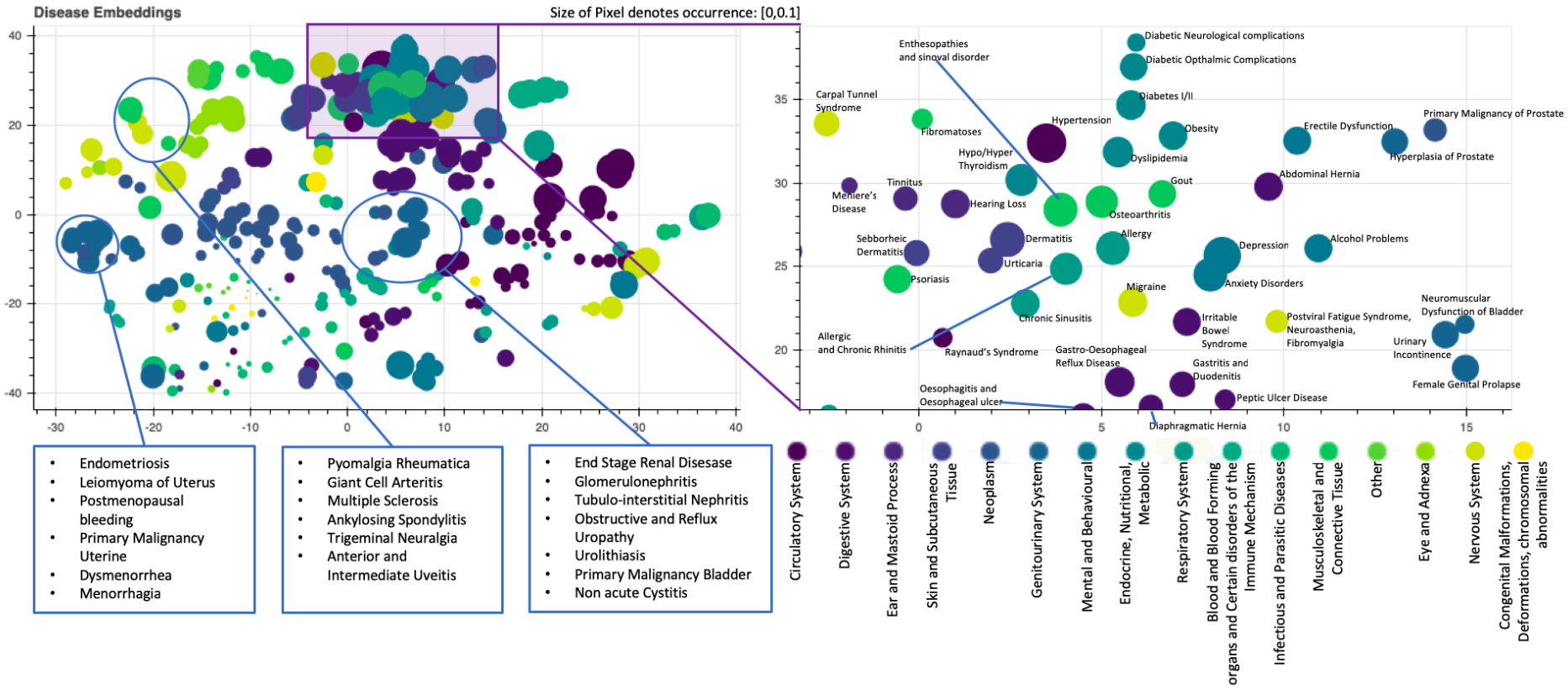
(d) Single Sentence Tagging Tasks: CoNLL-2003 NER

BEHRT — A Transformer model (*a la* BERT) for EHR



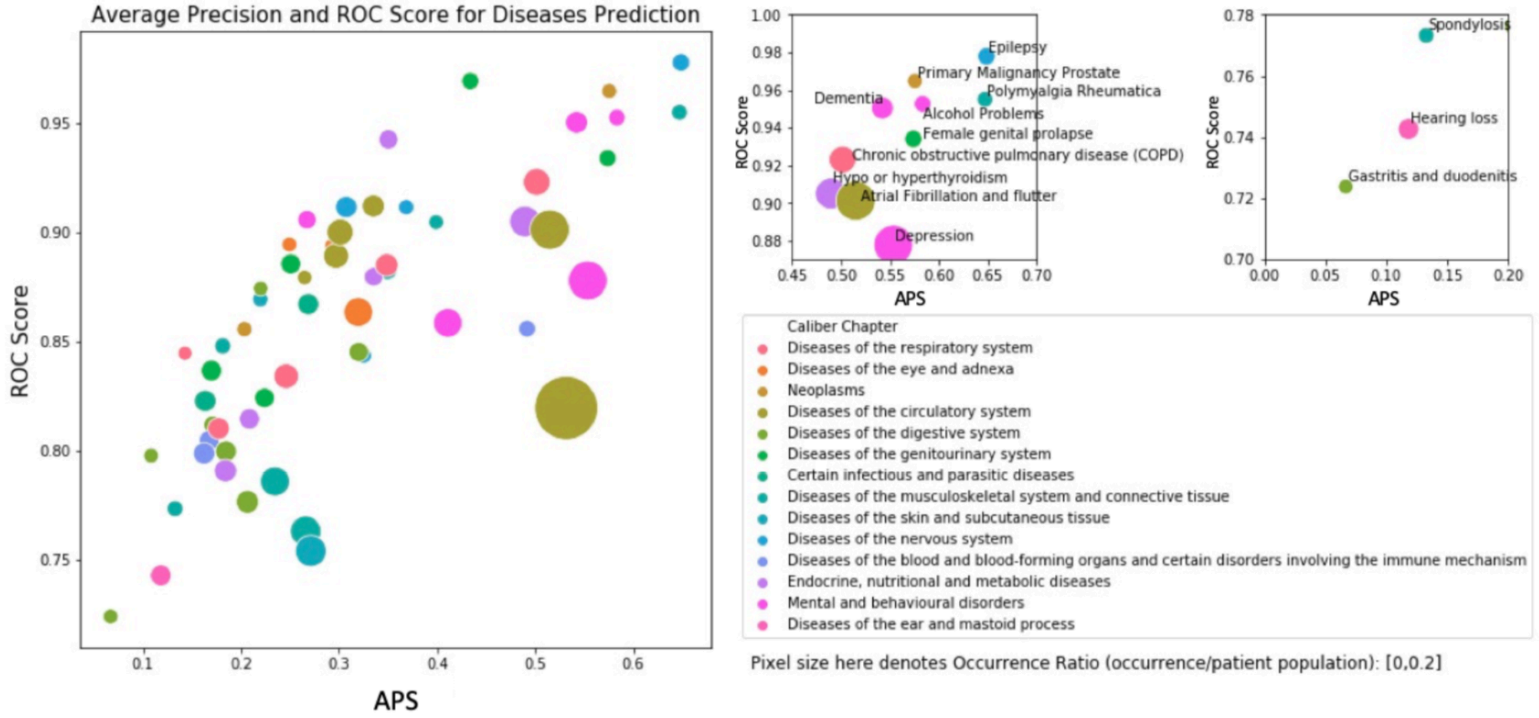
Li and Rao et al 2019

Semantic evaluation of the BEHRT's learned representations



Li and Rao et al 2019

BEHRT provides a universal representation for a range of diseases



Li and Rao et al 2019

Model Name	Next Visit (APS | AUROC)	Next 6M (APS | AUROC)	Next 12M (APS | AUROC)
BEHRT	**0.462 | 0.954**	**0.525 | 0.958**	**0.506 | 0.955**
DeepR	0.360 | 0.942	0.393 | 0.943	0.393 | 0.943
RETAIN	0.382 | 0.921	0.417 | 0.927	0.413 | 0.928

Assigning confidence to predictions — BNN

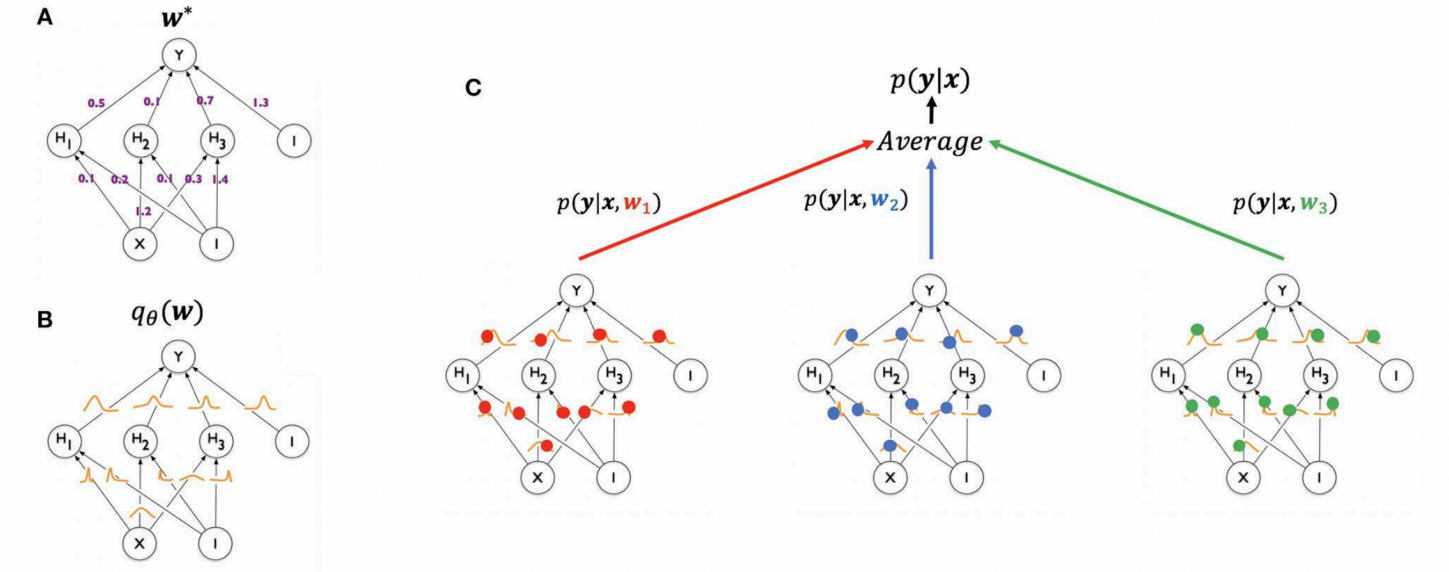
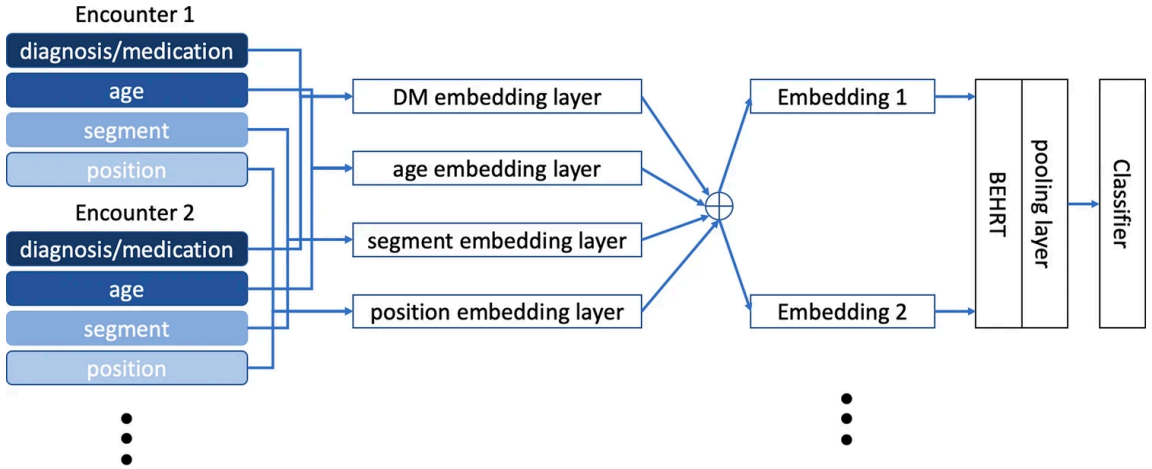


FIGURE 1 | Illustration of generating a prediction from a Bayesian neural network using Monte Carlo sampling (modified from Blundell et al., 2015). A standard neural network (**A**, top left) has one weight for each of its connections ($\mathbf{w}^*$), learned from the training set and used in generating a prediction for a test example. A Bayesian neural network (**B**, bottom left) has, instead, a posterior distribution for each weight, parameterized by theta ($q_\theta(\mathbf{w})$). The process of training starts with an assigned prior distribution for each weight, and returns an approximate posterior distribution. At test time (**C**, right), a weight sample $\mathbf{w}_1$ (red) is drawn from the posterior distribution of the weights, and the resulting network is used to generate a prediction $p(\mathbf{y}|\mathbf{x}, \mathbf{w}_1)$ for an example x. The same can be done for samples $\mathbf{w}_2$ (blue) and $\mathbf{w}_3$ (green), yielding predictions $p(\mathbf{y}|\mathbf{x}, \mathbf{w}_2)$ and $p(\mathbf{y}|\mathbf{x}, \mathbf{w}_3)$, respectively. The three networks are treated as an ensemble and their predictions averaged.

Bayesian BEHRT, for estimating prediction uncertainty



Model	HF		Diabetes		Depression	
	AUROC	AP	AUROC	AP	AUROC	AP
DBGP	0.941 (± 1e−3)	0.625 (± 5e−3)	0.834 (± 2e−3)	0.533 (± 4e−3)	0.776 (± 2e−3)	0.416 (± 3e−3)
Sparse-GP	0.945 (± 1e−3)	0.645 (± 2e−3)	0.834 (± 1e−3)	0.538 (± 3e−3)	0.782 (± 1e−3)	0.433 (± 2e−3)
KISS-GP	0.945 (± 1e−3)	0.649 (± 5e−3)	0.837 (± 2e−3)	0.538 (± 4e−3)	0.782 (± 1e−3)	0.433 (± 2e−3)
BE	0.942 (± 1e−3)	0.631 (± 7e−3)	0.830 (± 1e−3)	0.529 (± 2e−3)	0.774 (± 1e−3)	0.409 (± 2e−3)
BO	0.933 (± 1e−3)	0.645 (± 6e−3)	0.825 (± 1e−3)	0.525 (± 5e−3)	0.765 (± 1e−3)	0.425 (± 4e−3)
BE+BO	0.941 (± 1e−3)	0.628 (± 5e−3)	0.835 (± 2e−3)	0.538 (± 2e−3)	0.778 (± 1e−3)	0.419 (± 1e−3)

Table 1. Metrics for marginalized predictions on heart failure, diabetes, and depression. 95% confidence intervals are computed via a validation set bootstrapping with 50 bootstrap sets. *AP* average precision.

Li et al 2021

Bayesian BEHRT, for estimating prediction uncertainty

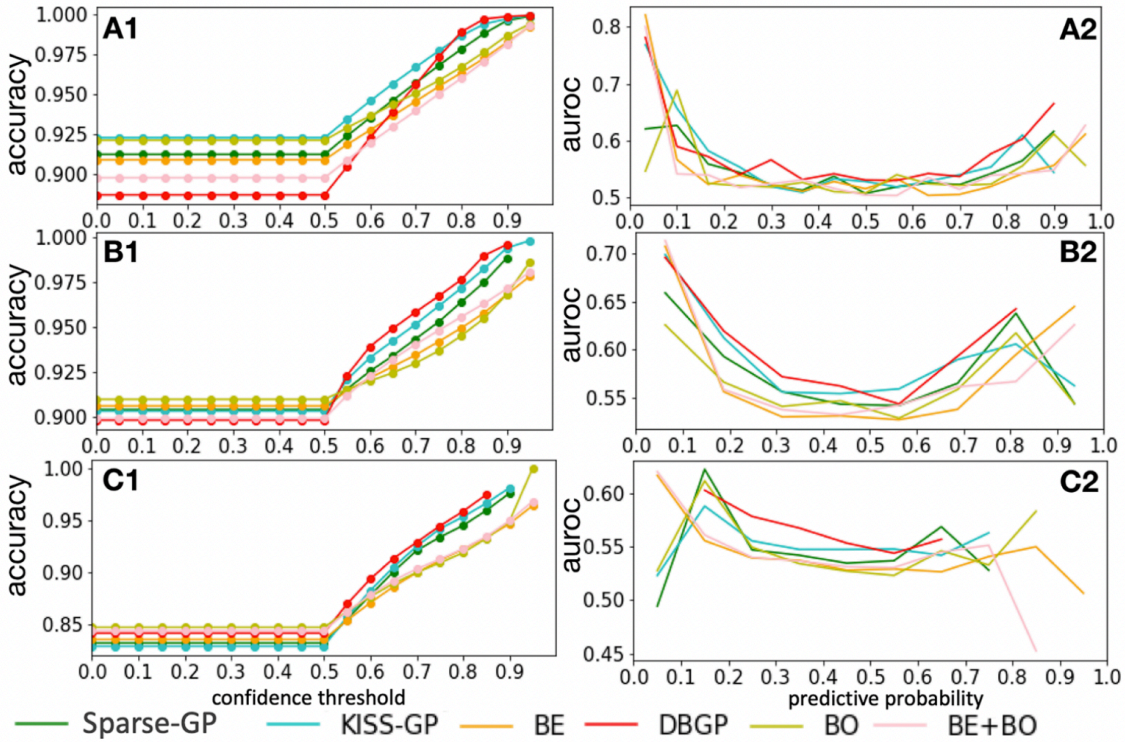


Figure 6. Accuracy and AUROC vs confidence curves. A: Heart failure, B: Diabetes, C: Depression. DBGP has higher accuracy in general and especially for predictions with high confidence, and it means DBGP is better at avoiding making overconfident predictions.

Li et al 2021

Causal ML — the data challenge

$$X^{FULL} = (Y(1), Y(0), W) \sim P_0 \qquad n \text{ i.i.d. copi}$$

subject	$Y(1)$	$Y(0)$	W_1	W_2
1	2.1	1.8	23	0
2	3.0	2.9	12	0
⋮				
n	2.5	2.5	30	1

$Y_i(1)$ = outcome when subject i treated

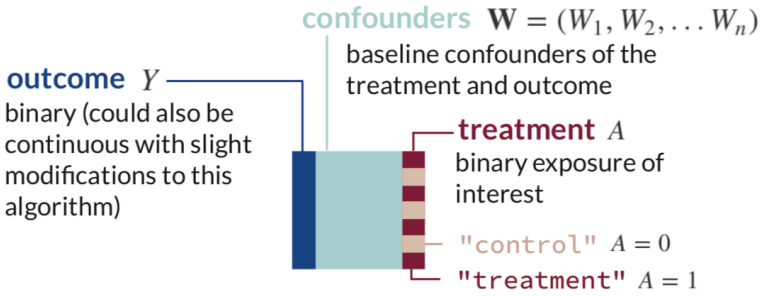
$Y_i(0)$ = outcome when subject i comparator

W_i = vector of baseline covariates

Average outcome if everyone treated: $\mu_1 = EY(1)$

Average outcome if everyone comparator: $\mu_0 = EY(0)$

$$\text{ATE} = \mu_1 - \mu_0, \qquad \text{RR} = \frac{\mu_1}{\mu_0}, \qquad \text{OR} = \frac{\mu_1/(1-\mu_1)}{\mu_0/(1-\mu_0)}$$



Targeted-BEHRT — A framework for observational causal inference on EHR

In standard epidemiology, RR is used to estimate the causal effect of an intervention. Therefore, one can think about the causal DL as a way to estimate RR more accurately.

$$RR = \frac{\mathbb{E}\left[Y^{T=1}\right]}{\mathbb{E}\left[Y^{T=0}\right]}$$

$$\hat{O}(X_i;\theta) = \text{CrossEntropy}(H(X_i, T_i;\theta), Y_i) + \text{CrossEntropy}(g(X_i;\theta), T_i).$$

$$\hat{\psi} = \mathbb{E}\left[\frac{\mathbb{E}[Y|X, T=1]}{\mathbb{E}[Y|X, T=0]}\right]$$

Rao et al 2022

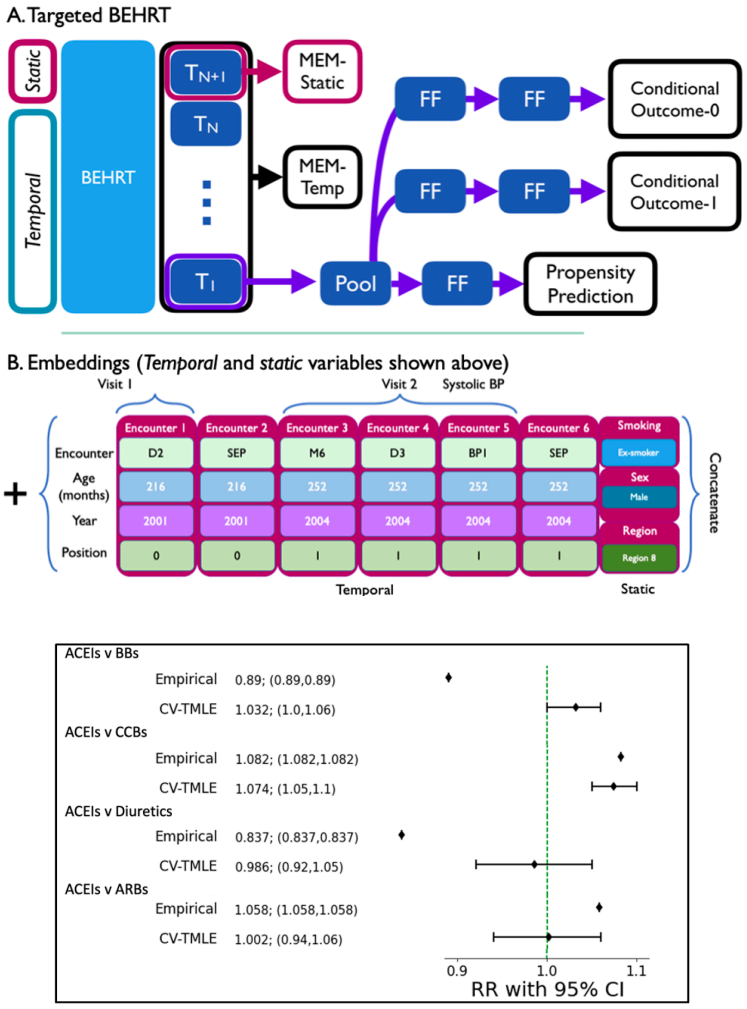
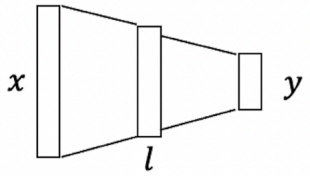


Fig. 5. Application of T-BEHRT on routine clinical data. Effect of ACEI on incident cancer with respect to BBs, CCBs, diuretics, and ARBs. This forest plot has four parts; one for each comparison to other antihypertensive drug classes. We demonstrate CV-TMLE risk ratio (RR) estimates with 95% confidence intervals (CIs) on our T-BEHRT model. In addition, we show empirical RR in the observational cohort selected for these experiments. The ground truth is assumed to be 1.0 (null) for all four associations validated by meta-analysis of RCTs. BBs: beta blockers; CCBs: calcium channel blockers; ACEIs: angiotensin-converting-enzyme inhibitors; ARBs: angiotensin receptor blockers; RR: risk ratio.

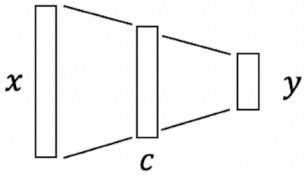
Outcome prediction under hypothetical interventions

Straight through representation learning (STRL)



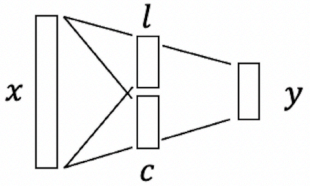
You can expect a y (and l) for any x

Concept bottleneck model (CBM)



Just like STRL, except that l is constrained to be human-defined concepts (c)

Partial concept bottleneck (PCB) model



A mix of reward in STRL and CBM; unconstrained representation (for prediction) from STRL (l), and interpretable concepts (c) from CBM

Li et al (under review)

Outcome prediction under hypothetical interventions

Table 1: Task performance with 95% confidence interval over 5 random seeds.

	STANDARD	PCB
AUROC	0.96(± 0.01)	0.95(± 0.00)
AUPRC	0.77(± 0.01)	0.75(± 0.01)

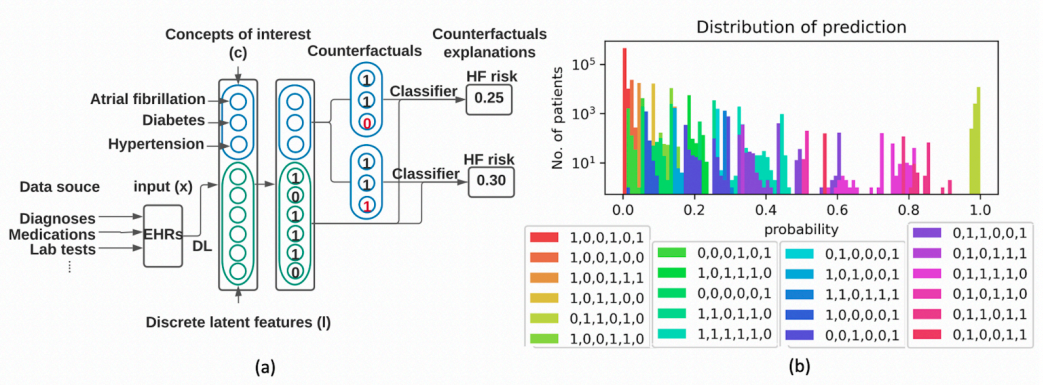


Figure 3: (a) Example of counterfactual explainations, where we intervene on the concetps of interest to investigate the "what if" question. For a group of patients with latent feature "1,0,1,1,1,0", the model thinks having hypertension as a comorbidity of AF and diabetes would increase the HF risk by 20%. (b) Risk distribution of predicted HF risk on the validation set. The color represents different clusters.

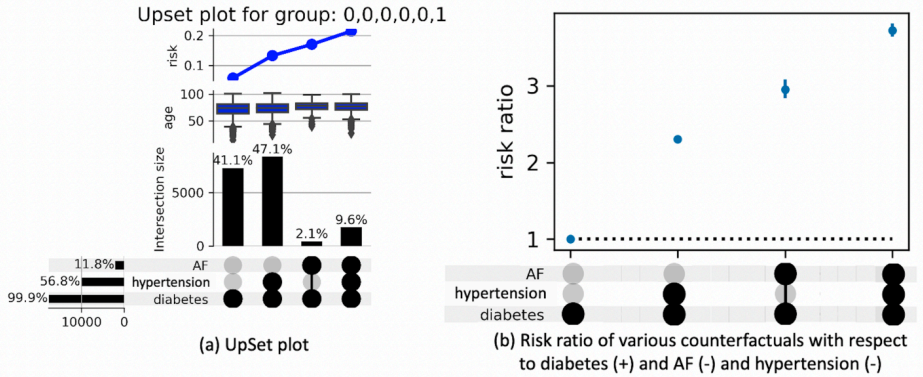


Figure 4: Counterfactual analysis for patients within group "0,0,0,0,0,1". (a) UpSet plot with age distribution and estimated HF risk from PCB model for counterfactuals. (b) RR of counterfactuals with respect to AF(-) and hypertension (-) and diabetes (+)

Li et al (under review)

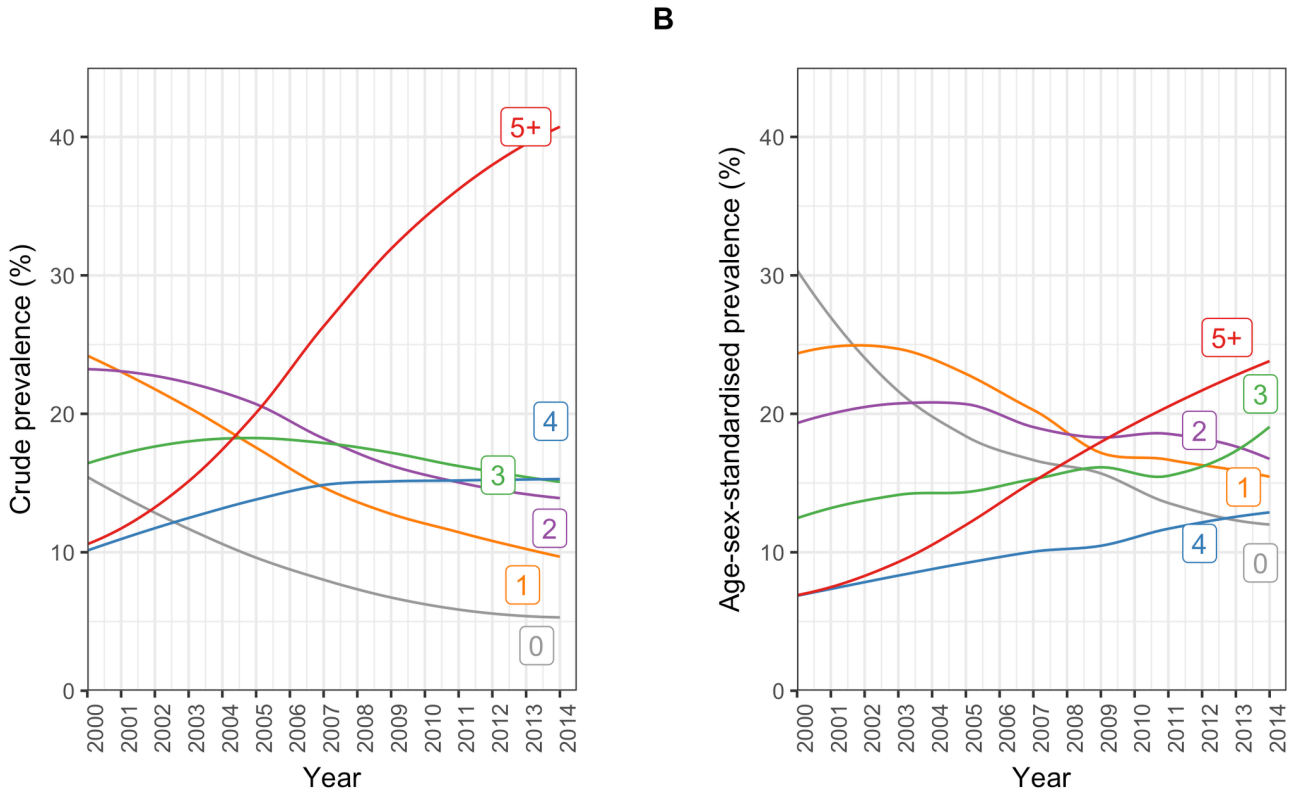
Section III

EHR + ML for Phenotyping

AIG

The importance of precision phenotyping, for understanding multimorbidity

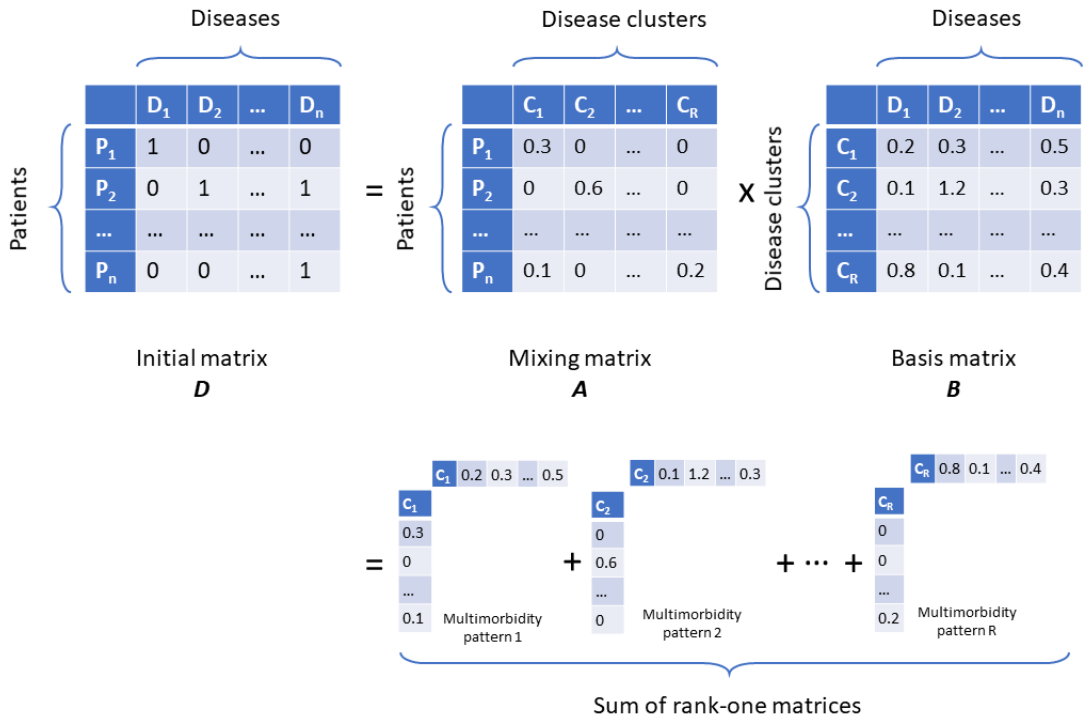
Due to changing demographics, advances in healthcare, and more, we are currently seeing a growth in multimorbidity — the simultaneous presence of more than one chronic disease in the same individual.



Tran et al 2018

Matrix and tensor factorisation has been the most common approach for mining latent EHR patterns.

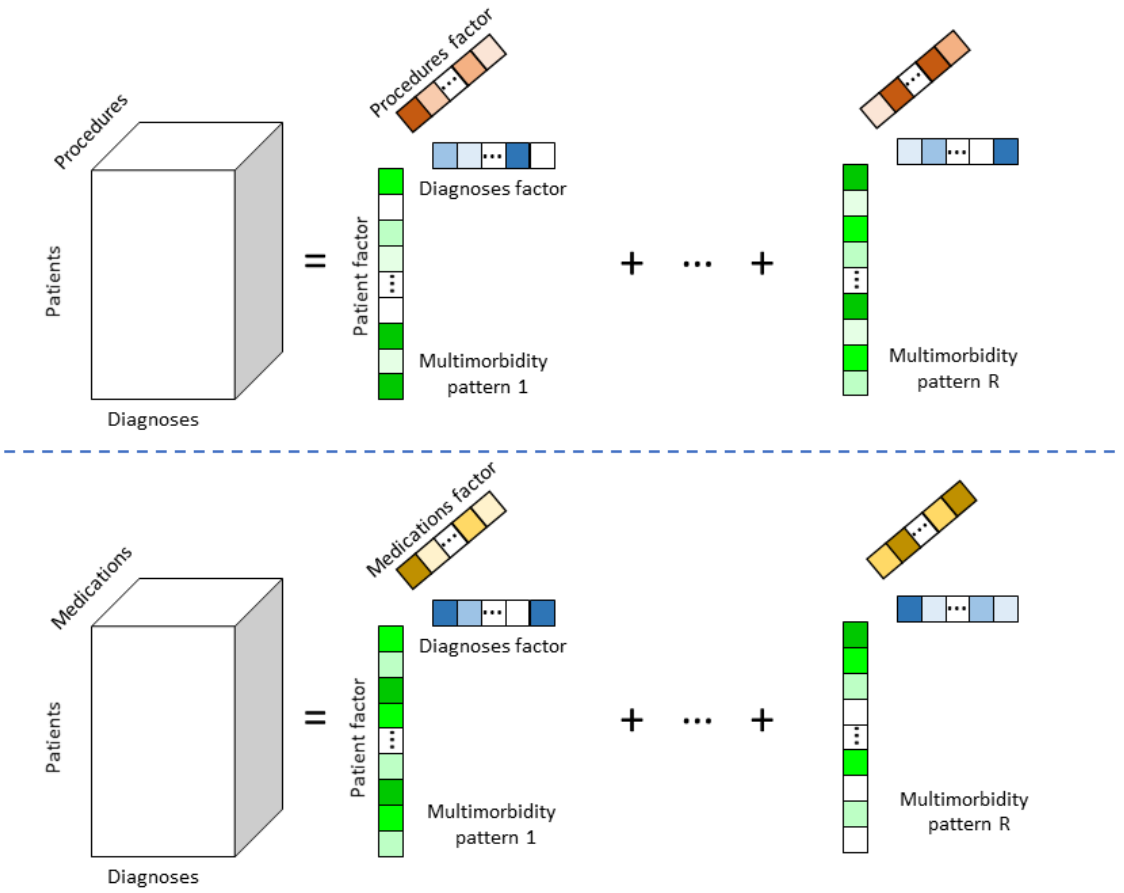
In such approaches, the data is summarised into matrix or tensor forms, along the key dimensions of interest (e.g., diseases, medications, and other variables). When factorised, the results will show the common cooccurrence patterns and more.



Matrix and tensor factorisation has been the most common approach for mining latent EHR patterns.

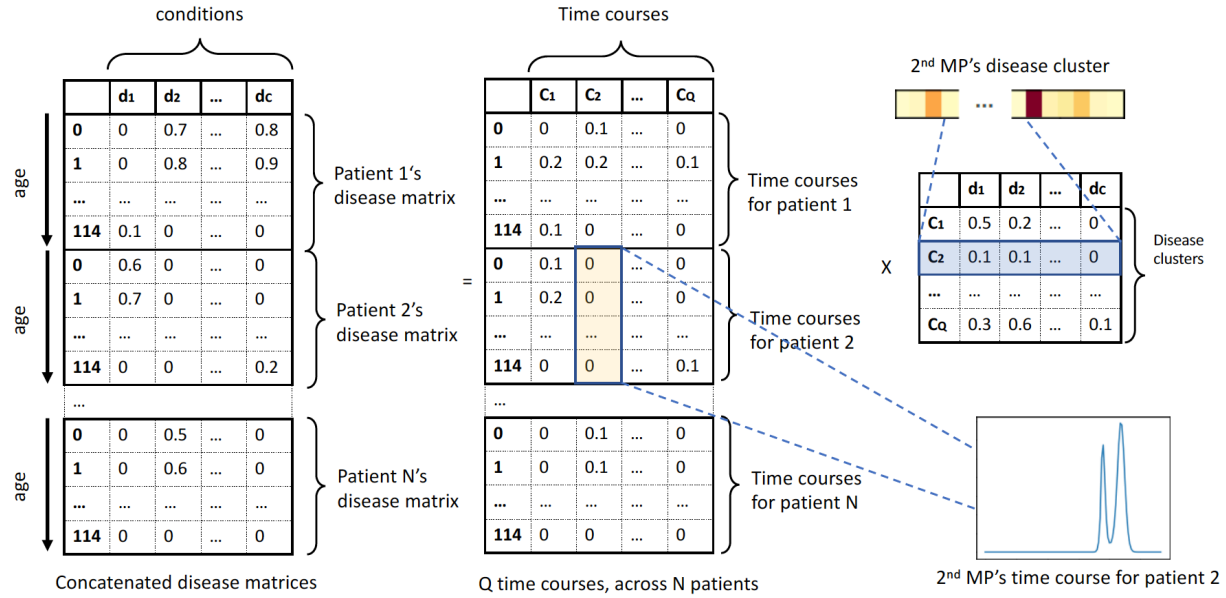
By adding more dimensions — as going from matrix to tensor — one can study additional properties of disease / multimorbidity clusters. For instance, having disease and medications (in addition to patients) can result in:

1. Disease clusters
2. Medication clusters
3. And they are expressed in each patient.

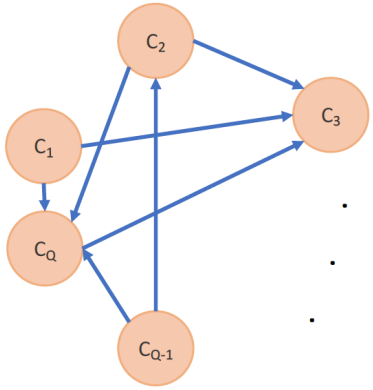


Studying the temporal patterns of multimorbidity over time, using NMF

Step 1: Matrix factorisation



Step 2: Multimorbidity network



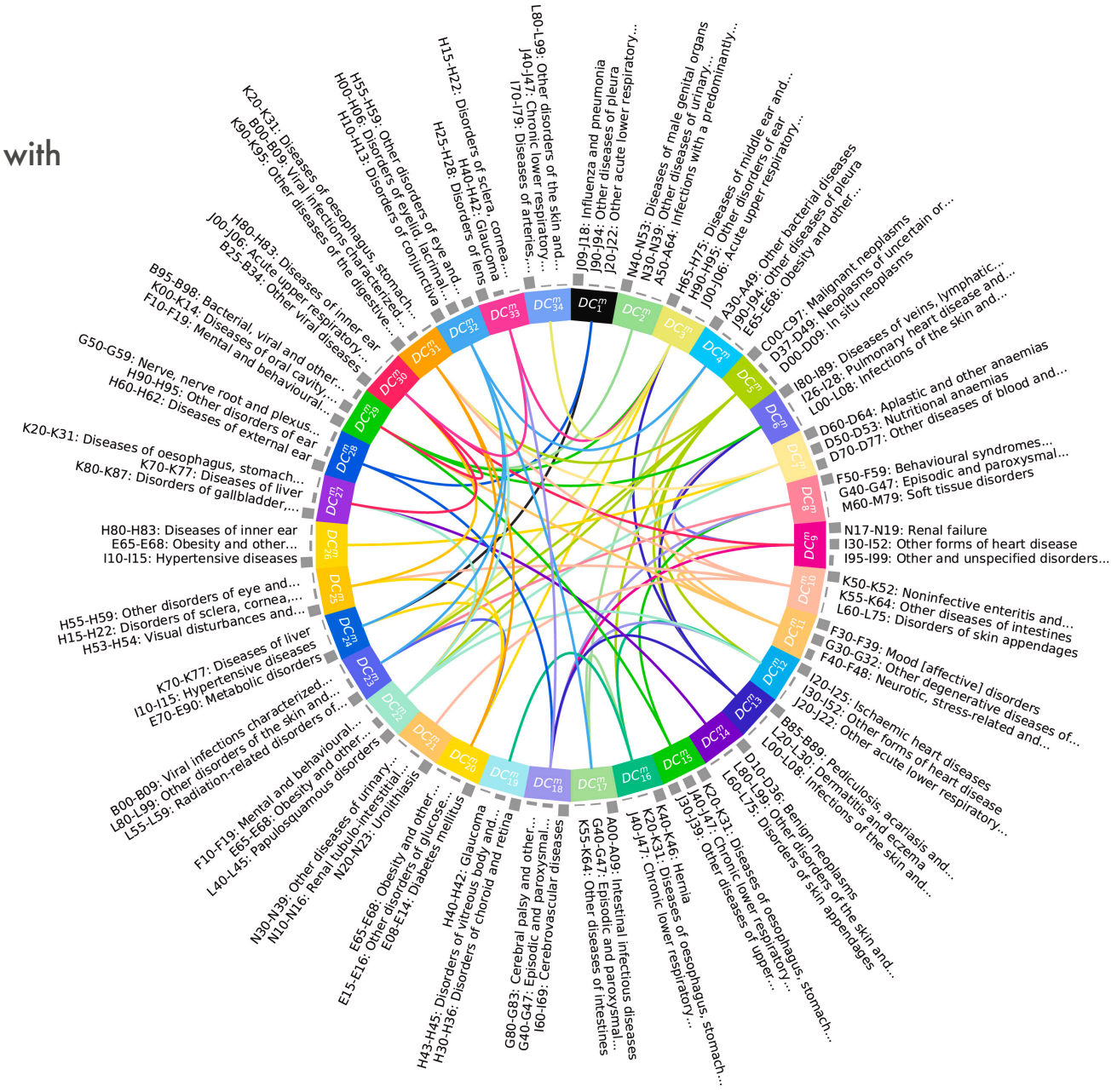
The multimorbidity network inferred from Q time courses (covering patient 1 to N's lifetime).

Hassaine et al 2020

Hassaine & Salimi-Khorshidi et al 2020

OXML2022 — ML FOR ELECTRONIC HEALTH RECORDS (EHR)

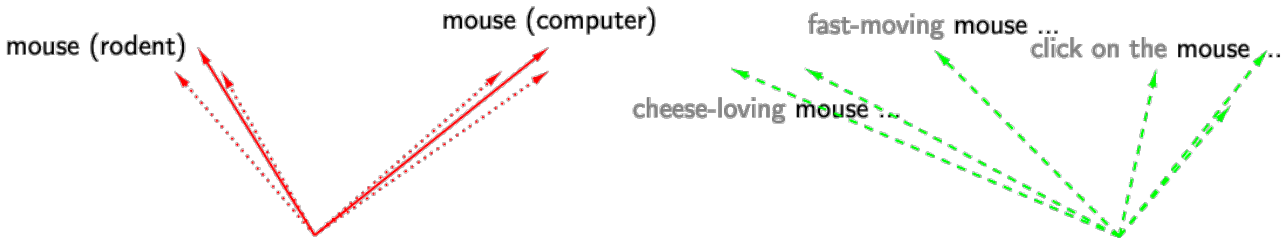
Disease clusters and their causal relationship with each other



Hassaine et al 2020

Hassaine & Salimi-Khorshidi et al 2020

Using contextualised embeddings to study multimorbidity and phenotyping (I)



Following scores:

1. Average cosine similarity of a word with itself across all the contexts in which it appears.
2. The average cosine similarity between a word and its context
3. The proportion of variance in a word's representations that can be explained by their first principal component.

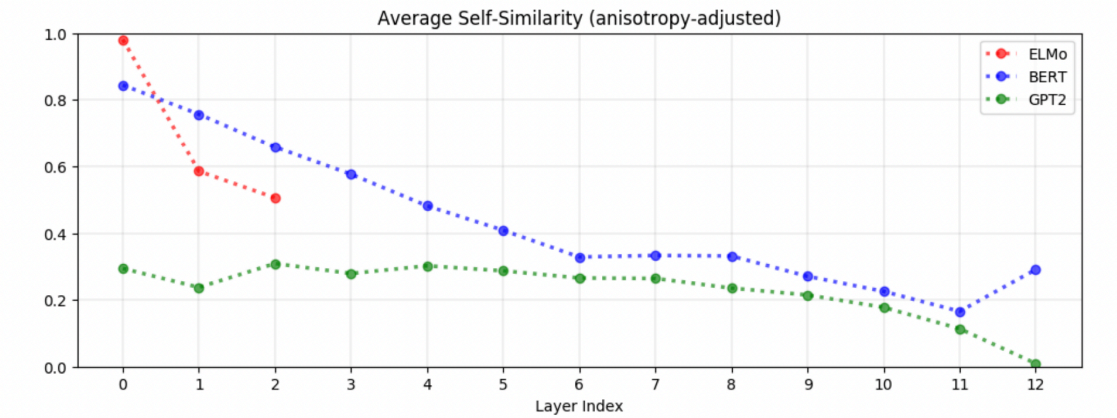
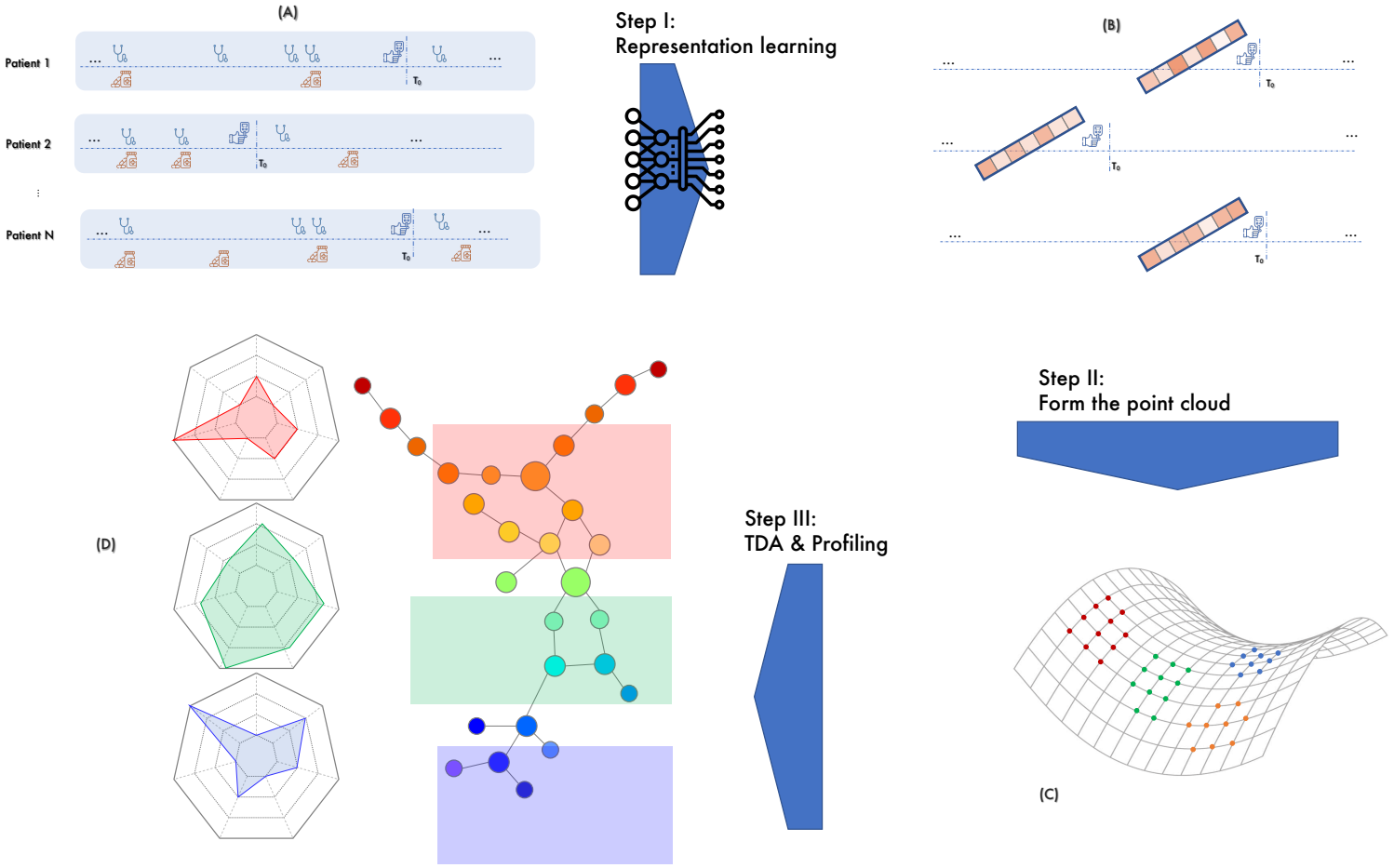


Figure 2: The average cosine similarity between representations of the same word in different contexts is called the word's *self-similarity* (see Definition 1). Above, we plot the average self-similarity of uniformly randomly sampled words after adjusting for anisotropy (see section 3.4). In all three models, the higher the layer, the lower the self-similarity, suggesting that contextualized word representations are more context-specific in higher layers.

Ethayarajh 2019

Using contextualised embeddings to study multimorbidity and phenotyping (II)

Assuming that contextualised representations (eg, from BEHRT) can capture the context well, enables one to assume such a representation of a disease is sufficient for phenotyping. That is, any difference in a disease like diabetes in patient A and B at a point in time, can be captured by the difference in their corresponding diabetes vectors at that point in time.

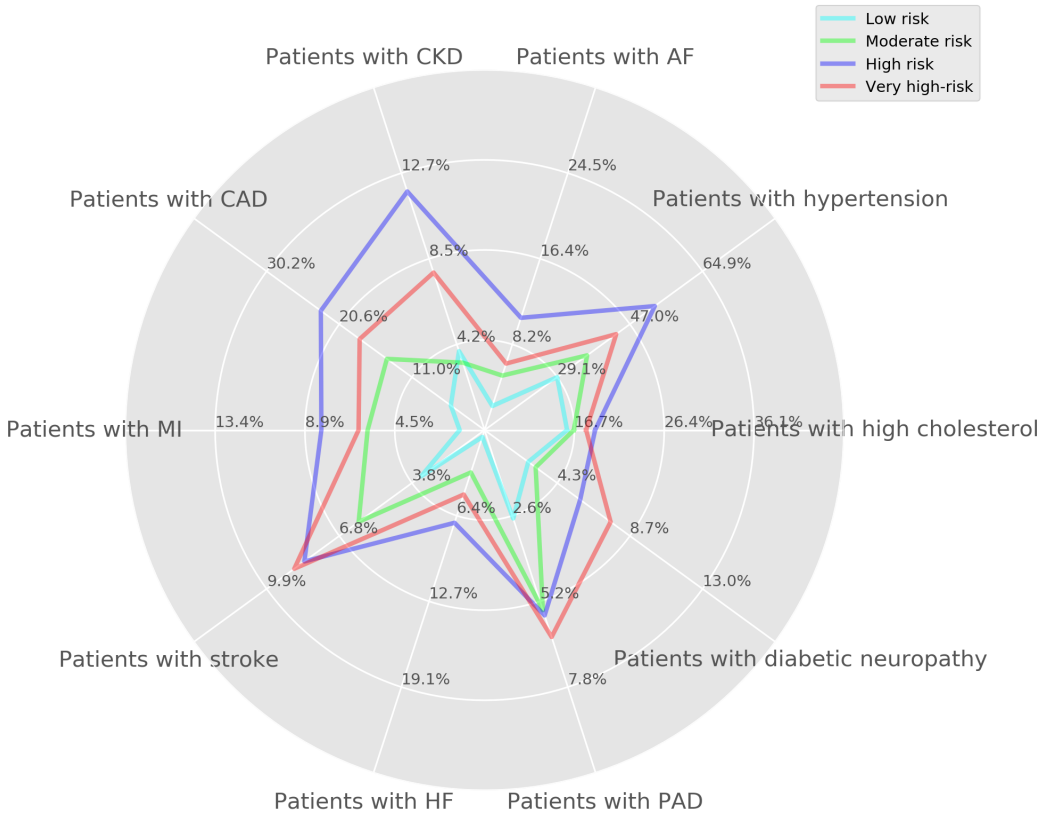


Wamil et al (under review)

OXML2022 — ML FOR ELECTRONIC HEALTH RECORDS (EHR)

Using contextualised embeddings to study multimorbidity and phenotyping (III)

Our analyses led to 4 phenotypes, each showing different properties, in terms of diseases and risk profiles.



Wamil et al (under review)

Emerging frameworks for ML+EHR for real-world clinical use

Use of deep learning to develop continuous-risk models for adverse event prediction from electronic health records, by DeepMind.

The protocol involves five broad stages: (a) formal problem definition, (b) data pre-processing, (c) architecture selection, (d) calibration and uncertainty estimation and (e) model generalisability evaluation.

It results in a 33-step procedure.

Tomasev et al 2021

	Development	Execution
Problem definition (Steps 1–6)	Variable	–
Data pre-processing (Steps 7–16)	4–12 weeks	6–48 h
Model architecture (Steps 17–25)	2–8 weeks	1–4 d for each hyperparameter sweep
Calibration and uncertainty (Steps 26–29)	2–4 weeks	2–6 d
Generalizability evaluation (Steps 30–33)	2–4 weeks	4–8 d

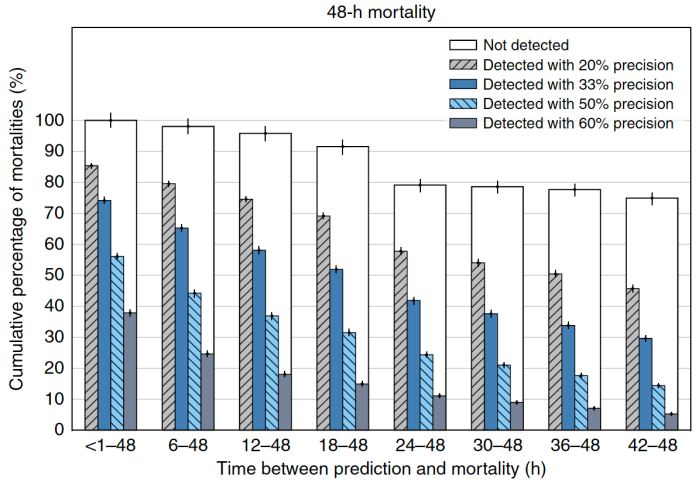


Fig. 8 | Early prediction histogram for mortality in 48 h. Model performance at timesteps before mortality. Error bars show bootstrap pivotal 95% confidence intervals; $n = 200$). The boxed area shows the upper limit on possible predictions for each time window.

Using ML+EHR for TREWS for sepsis

Taking ML beyond back-testing, by testing it for RR in real hospital settings. The result was that:

- 18.2%+ relative reduction in sepsis mortality from timely response to the system
- 5.7 hour lead time relative to when providers were ordering antibiotics for sepsis cases who previously died in the hospital
- 89% adoption rate amongst providers - legacy tools garner 10-15% adoption. Showing that providers are willing to trust and adopt AI/ML in their clinical practice?

Prospective, multi-site study of patient outcomes after implementation of the TREWS machine learning-based early warning system for sepsis

Roy Adams[1,2], Katharine E. Henry[2,3], Anirudh Sridharan[4], Hossein Soleimani[5], Andong Zhan[2,3], Nishi Rawat[6], Lauren Johnson[7], David N. Hager[8], Sara E. Cosgrove[8], Andrew Markowski[9], Eili Y. Klein[10], Edward S. Chen[8], Mustapha O. Saheed[10], Maureen Henley[7], Sheila Miranda[11], Katrina Houston[7], Robert C. Linton[4], Anushree R. Ahluwalia[7], Albert W. Wu[6,8,12,13,14] ✉ and Suchi Saria[1,3,8,12,15] ✉

Early recognition and treatment of sepsis are linked to improved patient outcomes. Machine learning-based early warning systems may reduce the time to recognition, but few systems have undergone clinical evaluation. In this prospective, multi-site cohort study, we examined the association between patient outcomes and provider interaction with a deployed sepsis alert system called the Targeted Real-time Early Warning System (TREWS). During the study, 590,736 patients were monitored by TREWS across five hospitals. We focused our analysis on 6,877 patients with sepsis who were identified by the alert before initiation of antibiotic therapy. Adjusting for patient presentation and severity, patients in this group whose alert was confirmed by a provider within 3 h of the alert had a reduced in-hospital mortality rate (3.3%, confidence interval (CI) 1.7, 5.1%, adjusted absolute reduction, and 18.7%, CI 9.4, 27.0%, adjusted relative reduction), organ failure and length of stay compared with patients whose alert was not confirmed by a provider within 3 h. Improvements in mortality rate (4.5%, CI 0.8, 8.3%, adjusted absolute reduction) and organ failure were larger among those patients who were additionally flagged as high risk. Our findings indicate that early warning systems have the potential to identify sepsis patients early and improve patient outcomes and that sepsis patients who would benefit the most from early treatment can be identified and prioritized at the time of the alert