Agenda

- Background
  - healthcare data
  - analytical tasks
  - why deep learning models?
  - deep learning architectures
- Success of Deep Learning in Computational Healthcare
  - Medical Classification
  - Sequential Prediction
  - Concept Embedding
  - Data Augmentation
- Open Challenges (10 min)
- Q&A

Medical Classification

- CAML-NAACL’ 18 (free text)
- LEAP (discrete clinical codes)
- Image based Classification (Nature skin cancer)

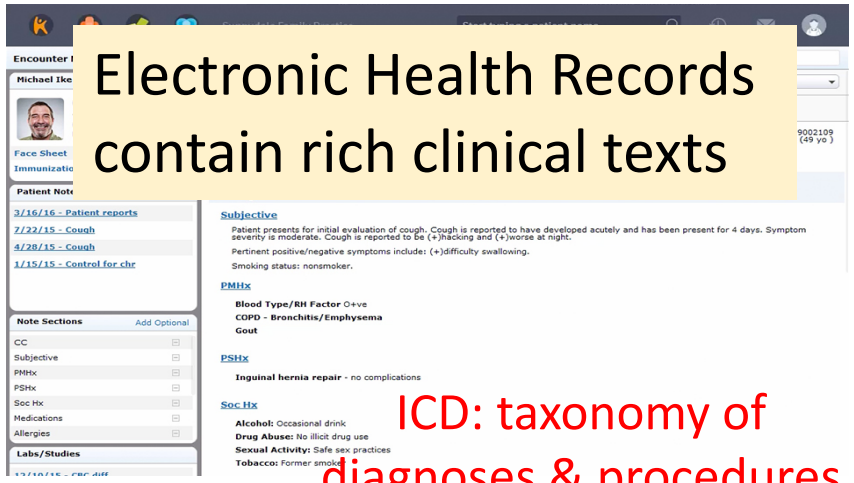
Explainable Prediction of Medical Codes from Clinical Text

James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, Jacob Eisenstein

NAACL-HLT' 18

CAML: Clinical Coding Problem as Multi-label Classification

Clinical Coding



Features	ICD-9
Possible Codes	14,000
Characters	3–5

Laborious and error-prone

Multi-label Classification

- Highly multi-label classification
  - 14K ICD9, 68K ICD10 labels
- Testbed for document representations
- Documents are long and loosely structured

CAML: The MIMIC-III Dataset

- Open-access, de-identified
- 47k admissions -> 47k documents for training
- **Loosely structured:**

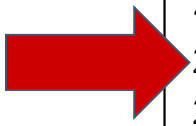
Many labels:

```
Admission Date:  [**2118-6-2**]       Discharge Date:  [**2118-6-14**]

Date of Birth:                    Sex:  F

Service:  MICU and then to [**Doctor Last Name **] Medicine

HISTORY OF PRESENT ILLNESS:  This is an 81-year-old female
with a history of emphysema (not on home O2), who presents...
```



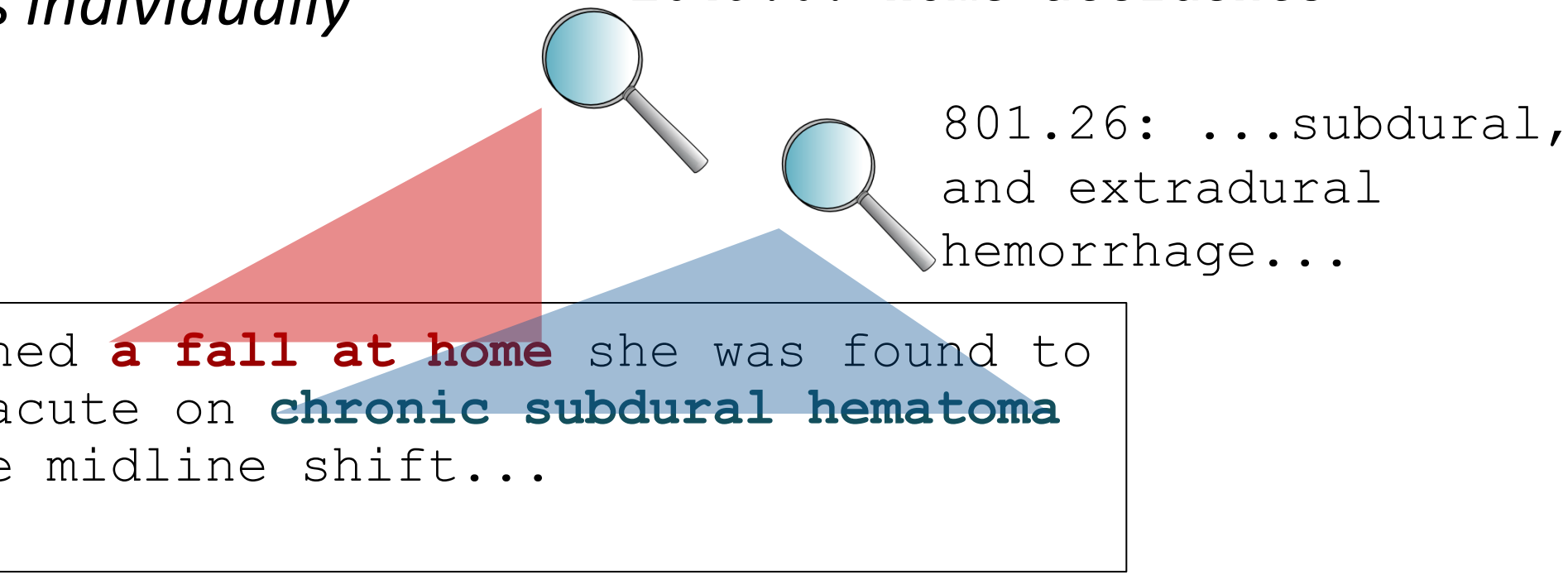
```
519.1: 'Other disease…'
491.21: 'Obstructive …'
518.81: 'Acute respir…'
486: 'Pneumonia, orga…'
276.1: 'Hyposmolality…'
244.9: 'Unspecified h…'
31.99: 'Other operati…'
.
.
.
```

Long: Median post-processed document length: 1,341

Median # labels: 14

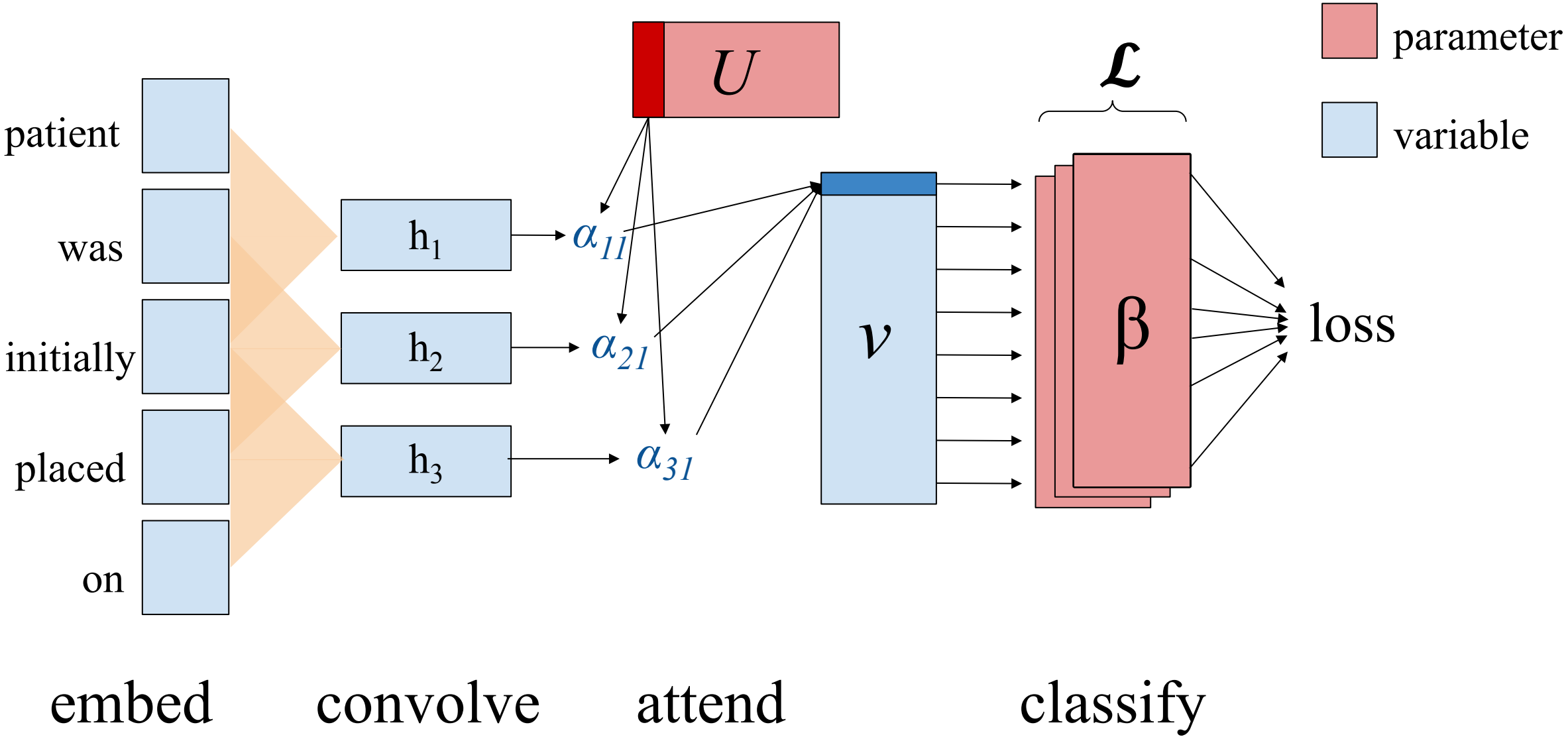
CAML: Modeling Consideration

- Focus on the parts that matter
- Treat labels *individually*
- Be fast!



...who sustained **a fall at home** she was found to have a large acute on **chronic subdural hematoma** with extensive midline shift...

The CAML model



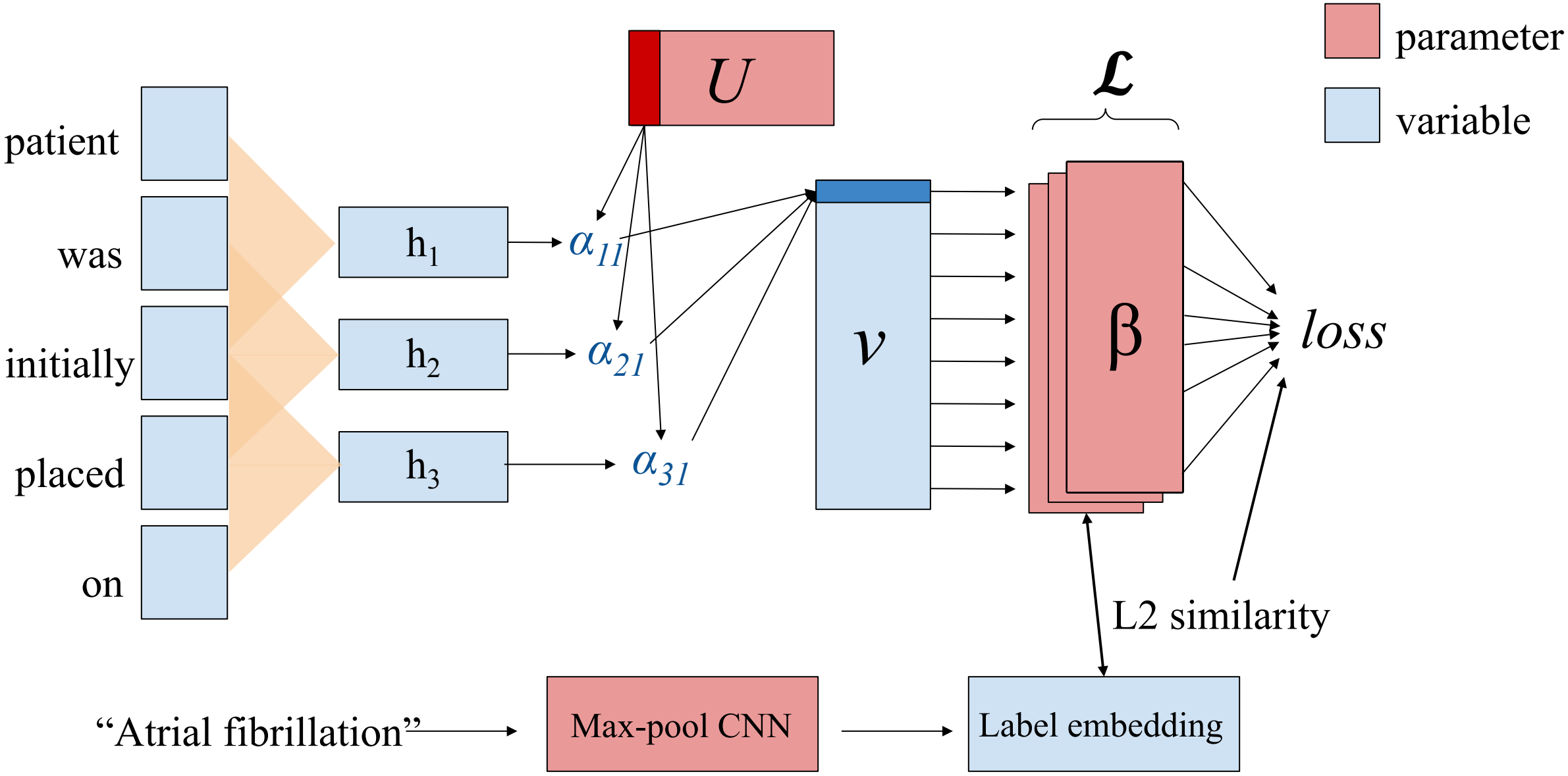
CAML: Dealing with the Long Tail

- Huge label space (nearly 9,000 total)
- Many labels are similar

250.00: “Diabetes mellitus without mention of complication, type II or unspecified type, not stated as uncontrolled”

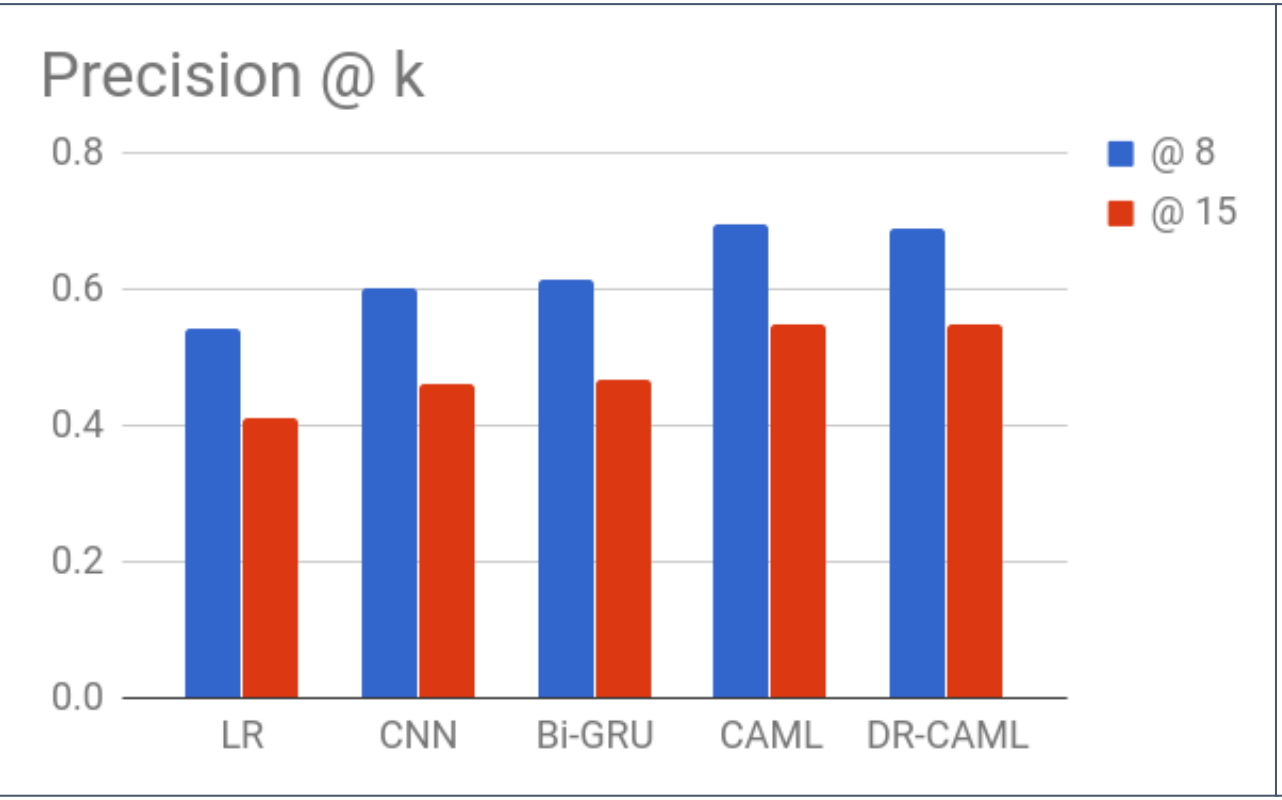
250.02: “Diabetes mellitus without mention of complication, type II or unspecified type, uncontrolled”

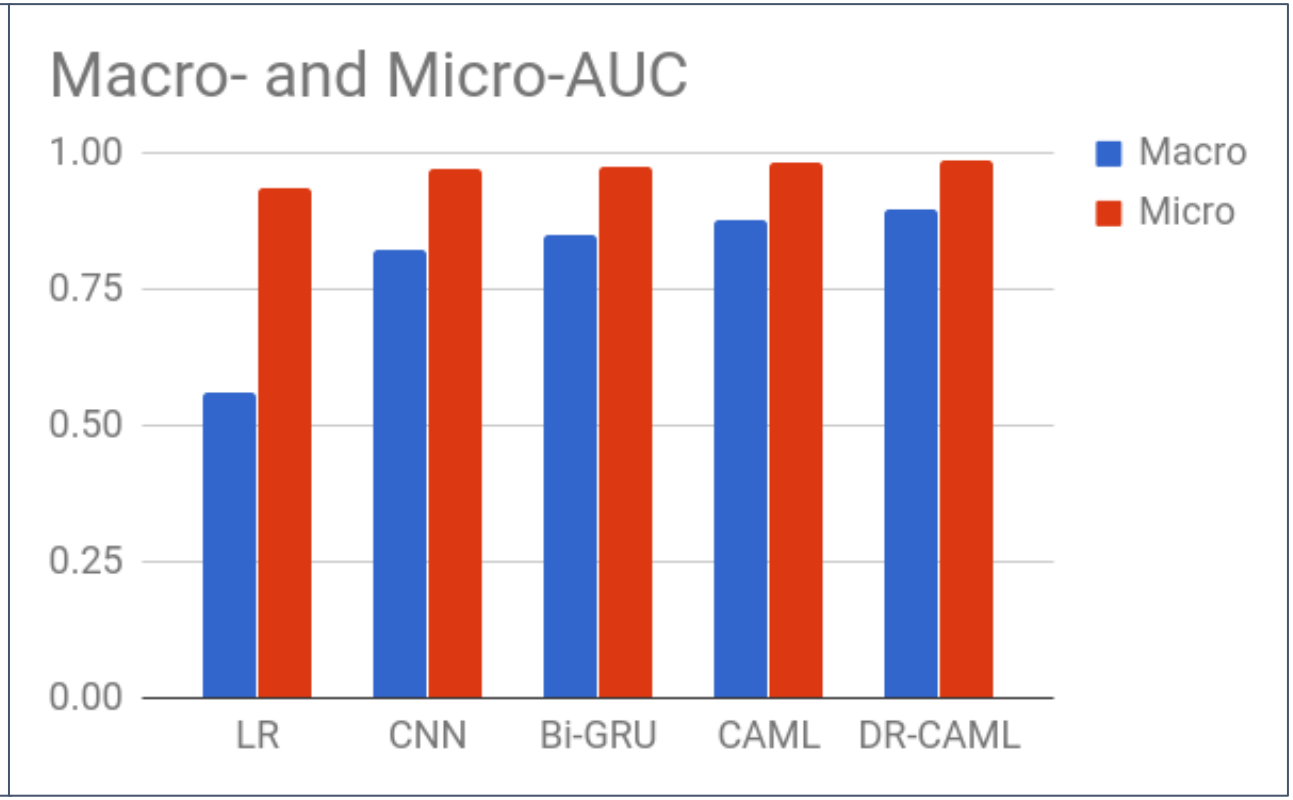
DR-CAML


parameter
variable
U
$\mathcal{L}$
patient
was
initially
placed
on
h_1
h_2
h_3
α_{11}
α_{21}
α_{31}
v
β
loss
L2 similarity
"Atrial fibrillation"
Max-pool CNN
Label embedding

CAML: Experiment Results

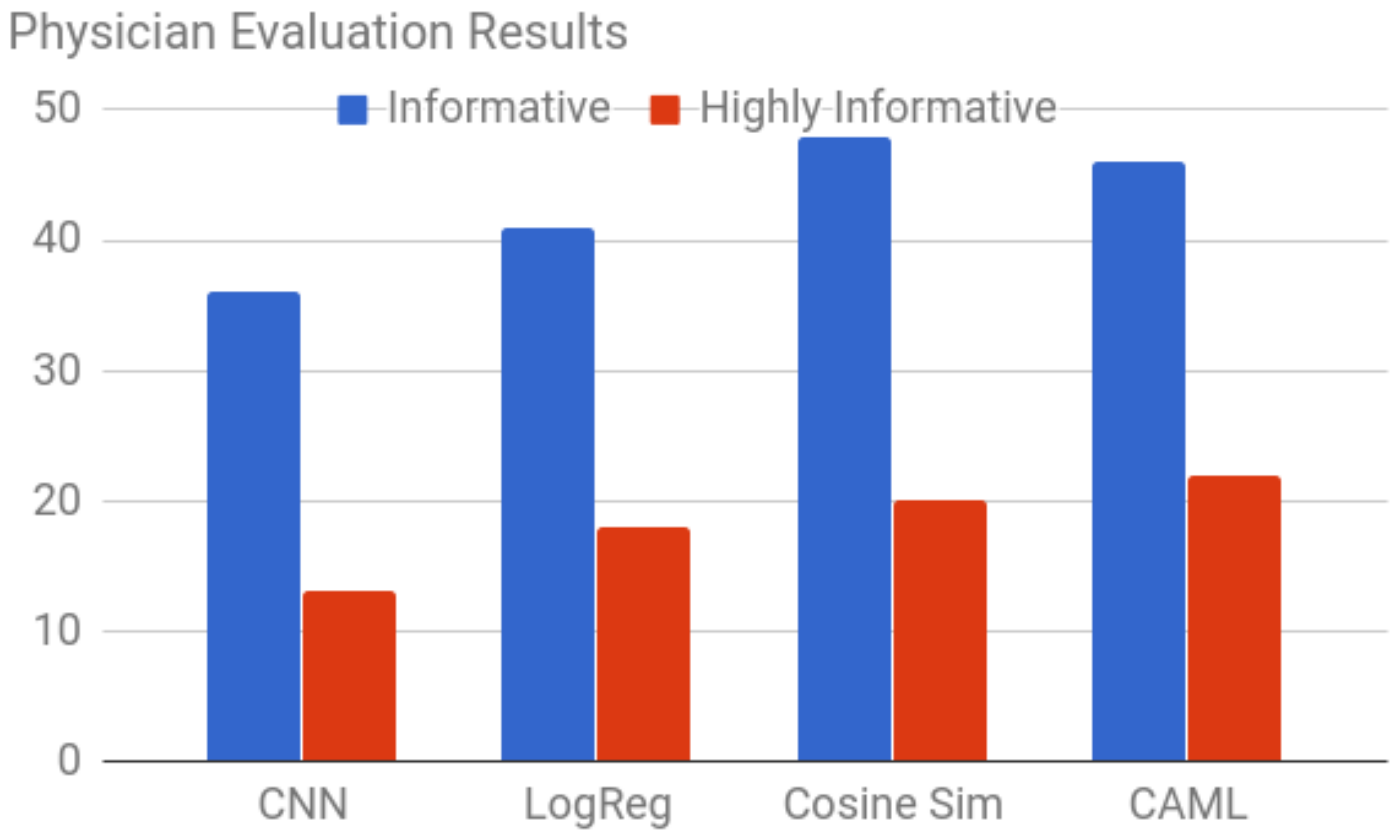
- Enable future comparison
- Precision @ k: decision support use-case



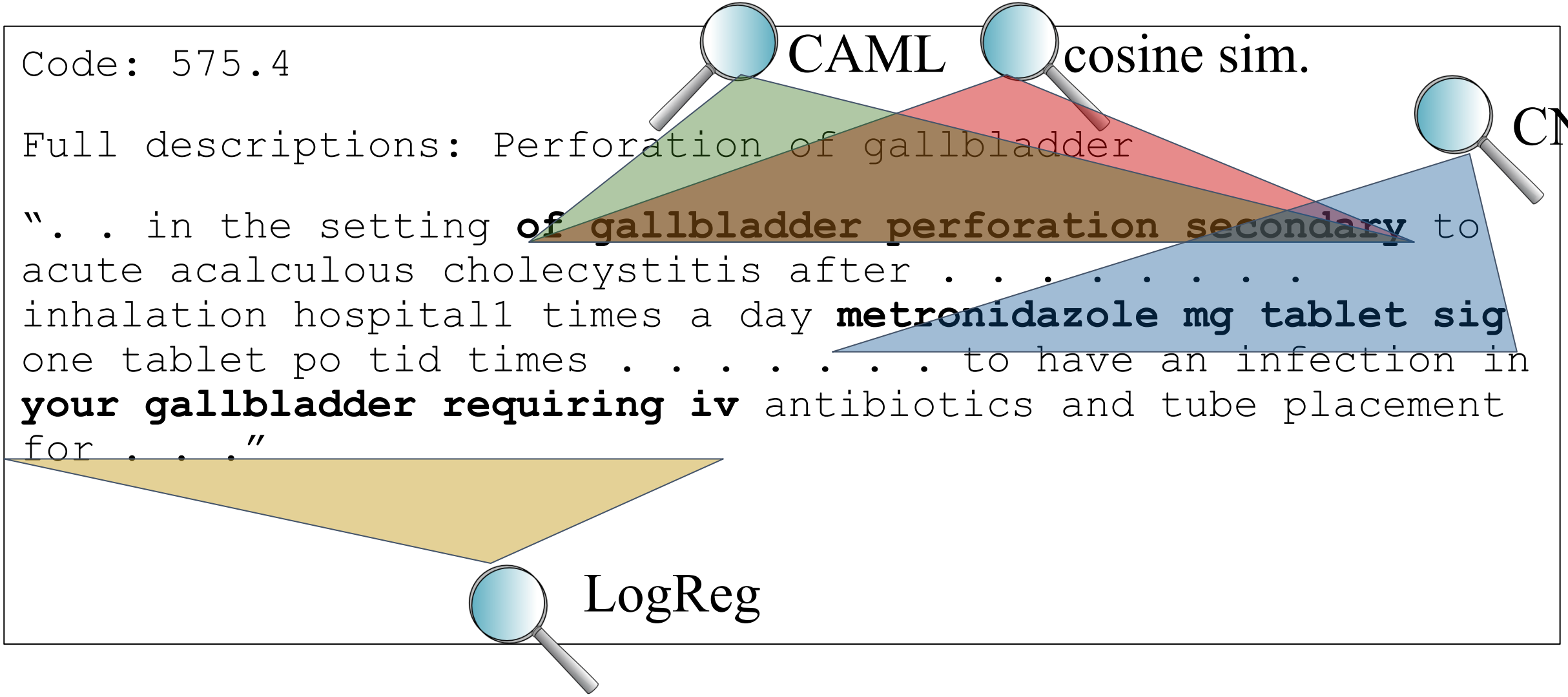


CAML: Physician Evaluation Results

- Improves upon CNN, LogReg
- More experts needed!

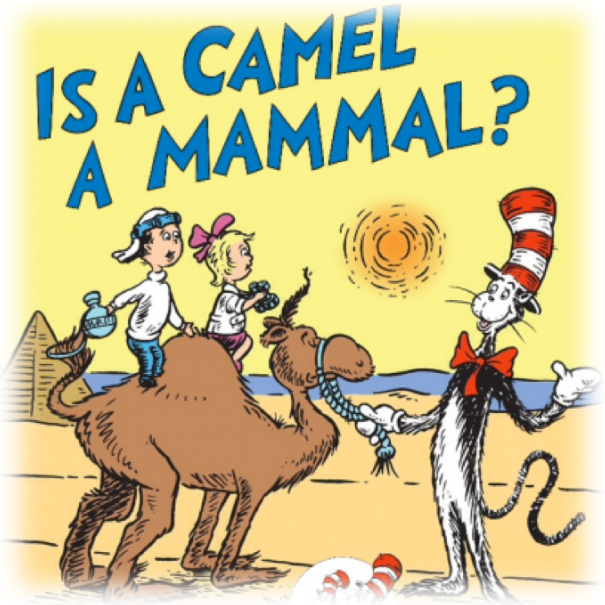


CAML: Physician Evaluation Example



CAML: Summary

- ICD coding is valuable and challenging
- Convolution + attention works well
- Attention can explain the predictions



LEAP: Learning to Prescribe Effective and Safe Treatment Combinations for Multimorbidity

Yutao Zhang, Robert Chen, Jie Tang, Walter F. Stewart, Jimeng Sun

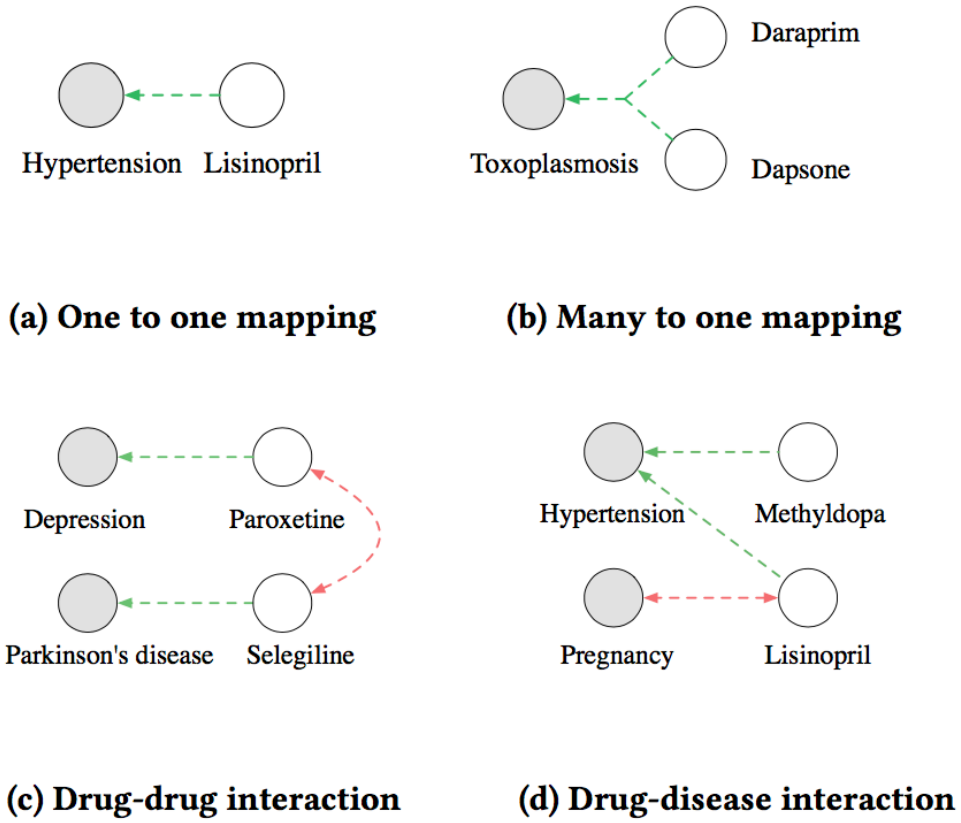
KDD' 17

LEAP: Multi-morbidity

- Co-occurrence of multiple medical conditions
- Traditional way of prescribing is based on doctors' intuition.
- Clinical decisions can be sub-optimal due to knowledge gaps.

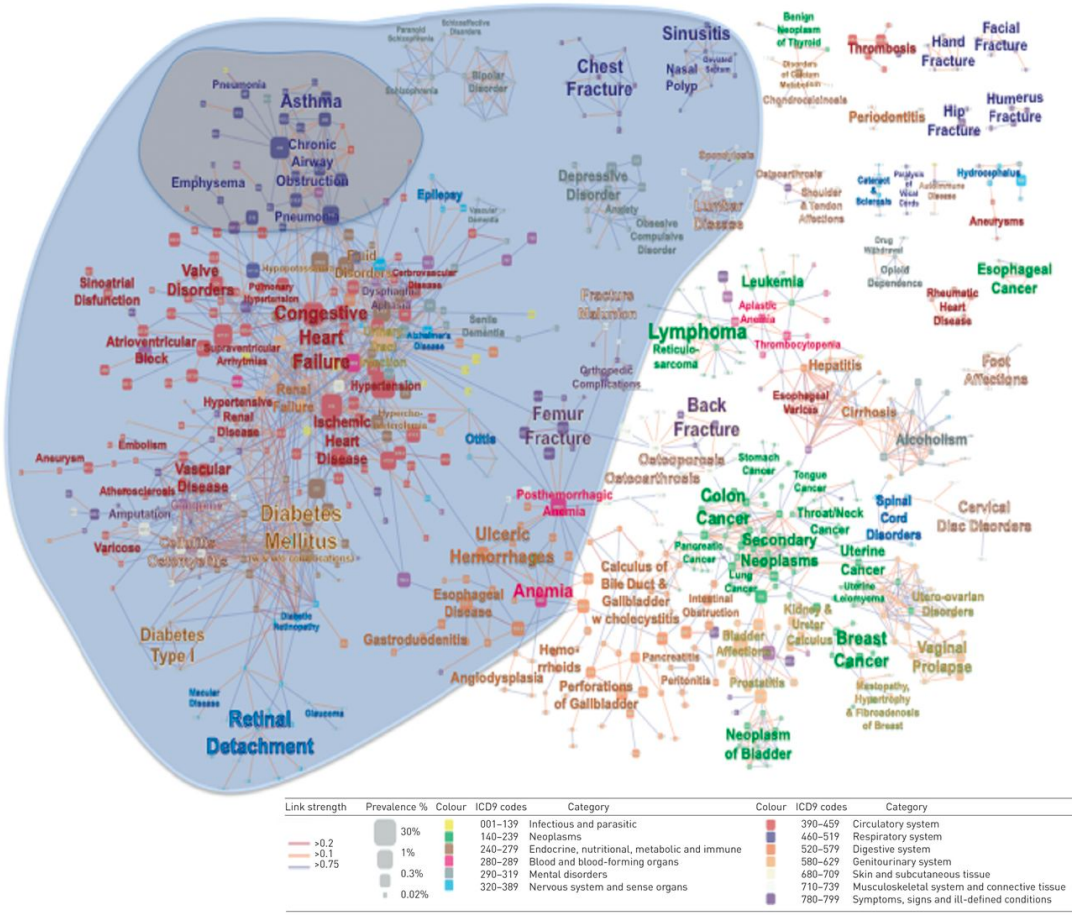
Complex Dependency between Drugs and Diseases

- One drug can treat multiple diseases.
- One disease requires a combination of multiple drugs.
- Potential adverse interactions among drugs.
- Potential conflicts between drugs and diseases.



LEAP: Treatment Recommendation for Multi-morbidity Patients

- Many-to-many mapping:
  - Patents are diagnosed with multiple diseases.
  - Requires a combination of drugs ad a treatment.
- Learning from EHR data:
  - No explicit mapping between drugs and diseases.
  - Complex dependencies.



LEAP: Multi-Instance Multi-Label Learning

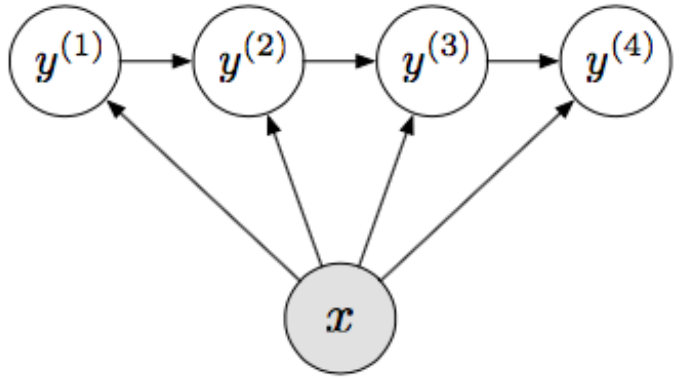
- Multi-Instance Learning:
  - A discrete set is given as input.
  - Assign a label to each input set.
- Multi-Label Learning:
  - Predict a variable length label set for each instance.

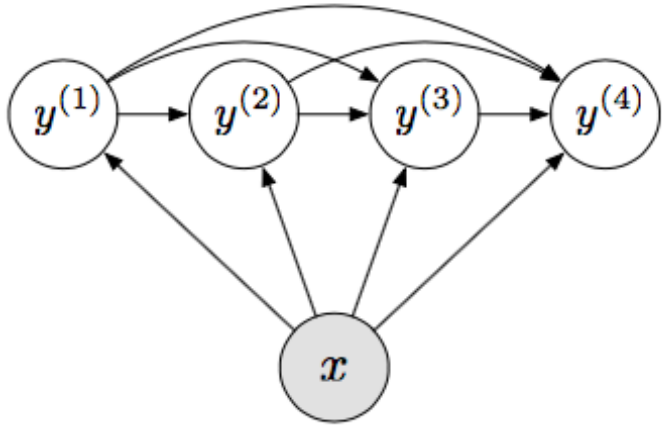
LEAP: Multi-Instance Learning Problem

- Independency assumption
  - The instances within an input set are independent.
- Equal contribution assumption
  - Each instances within an input set contribute equally to the generation of the corresponding label.
- Not true in the setting of treatment recommendation due to the complex high-order dependencies.

LEAP: Multi-Label Learning Problem

- Binary Relevance
  - The simplest solution.
  - Assume independency among labels.
  - Translate the Multi-label problem into a set of binary problems.
- Classifier Chains
  - Use the previous predictions as input of the next classifier.





LEAP Model

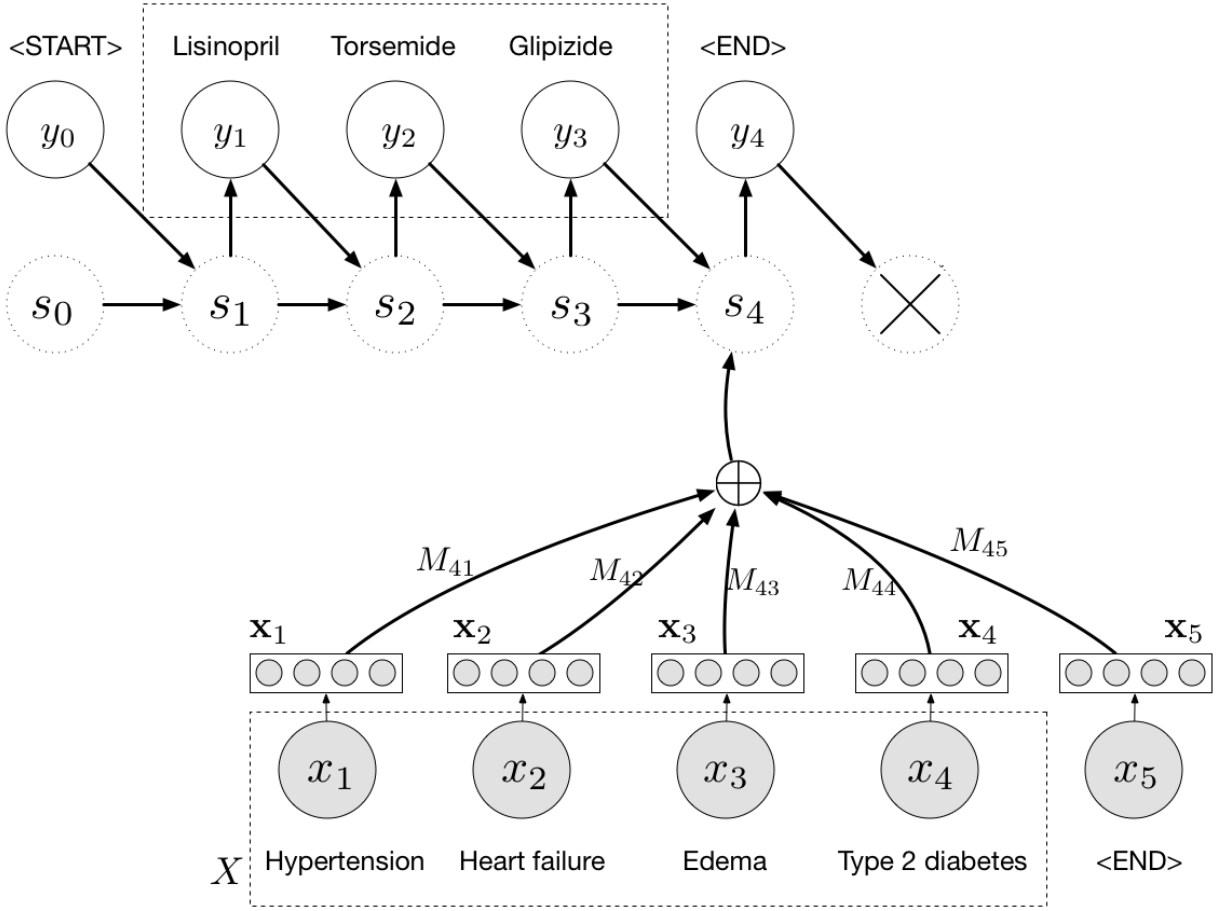
- Objective: Conditional probability p(Y|X) of drug set X given disease set Y
- Inspired by Classifier Chains:
  - Decompose the **Combinatorial Optimization** problem as a **Sequential Decision Making** problem:

$$p(Y|X) = \prod_{t=1}^{|Y|} p(y_t|X, y_1, y_2, \ldots, y_{t-1})$$

$$= \prod_{t=1}^{|Y|} p(y_t|\{x_1, x_2, \ldots, x_{|X|}\}, y_1, y_2, \ldots, y_{t-1}),$$

LEAP Model

- At step t, predict the next label based on input set X and the previous predictions.
- Different instances in the input set contribute differently to the currently prediction.
  - Eg. The weight of an input disease should be decreased if it has already been covered by some predicted drug.

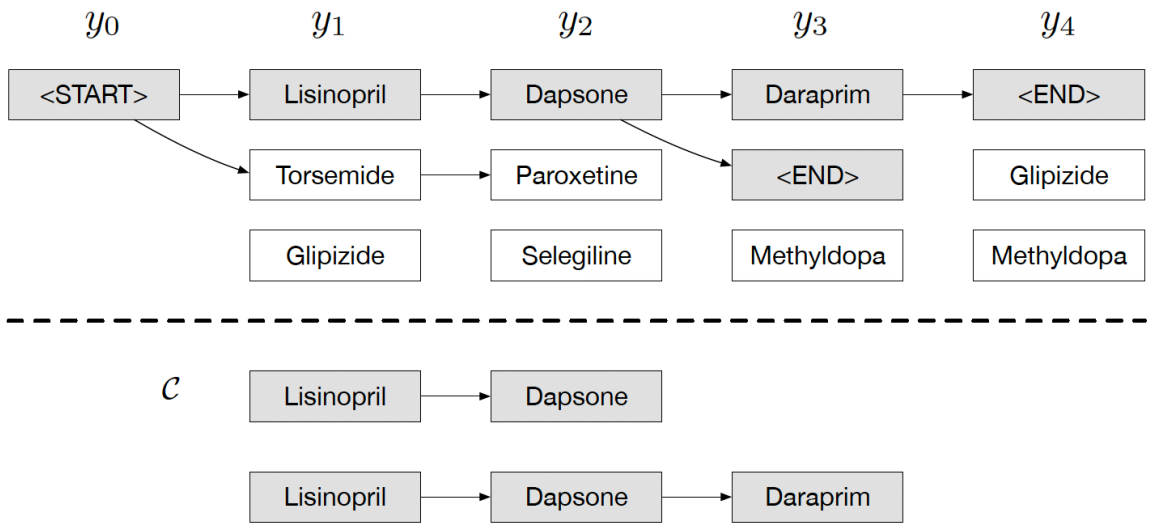


LEAP Model

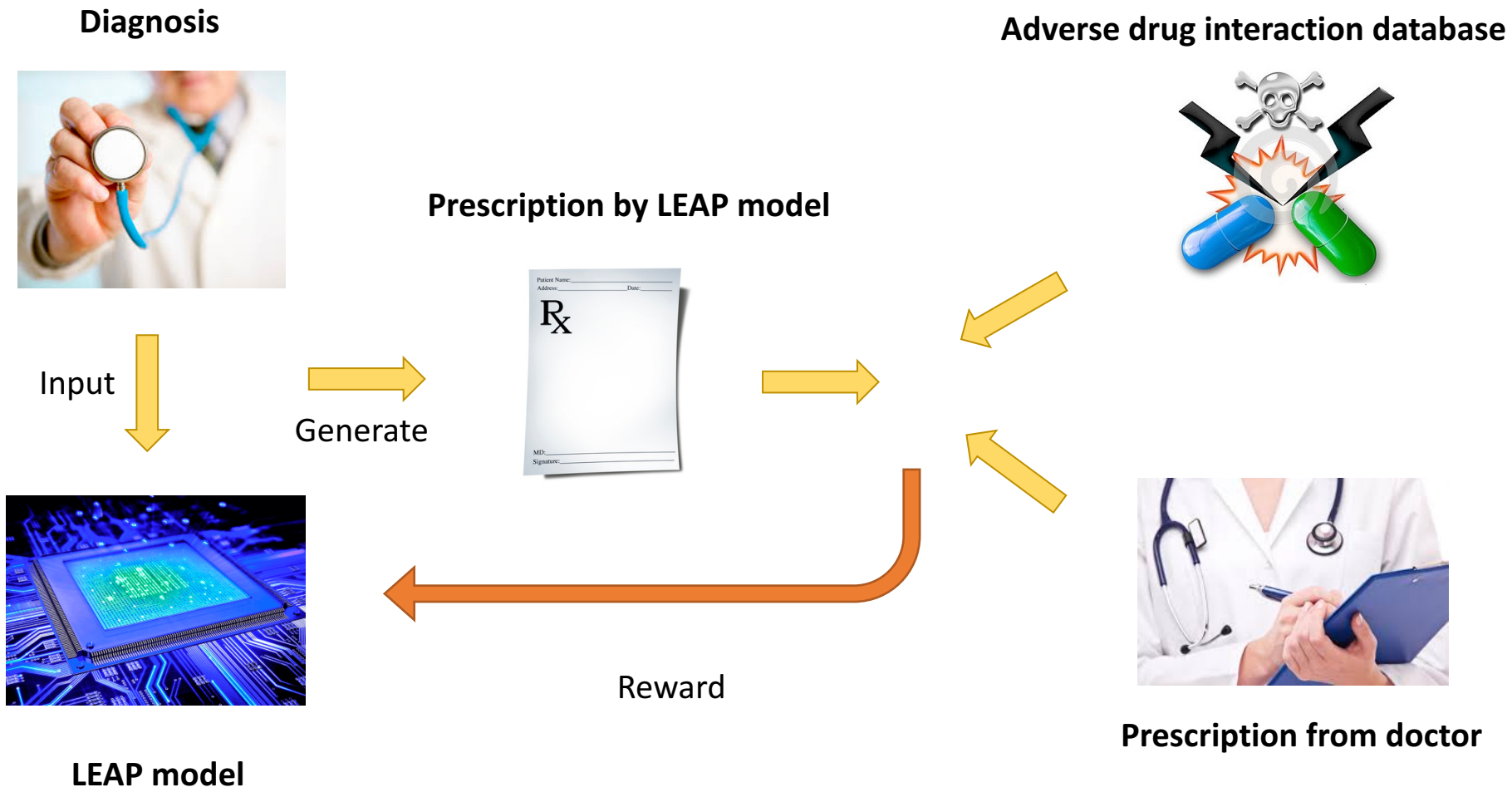
- $\mathbf{s}_t = g(\mathbf{s}_{t-1}, y_{t-1}, \Psi_t(X))$ the state variable at step t.
  - state : variable at step t-1 $\mathbf{s}_{t-1}$
  - $\mathbf{y}_{t-1}$ prediction at step t-1
  - $\Psi_t(X)$ an attention function over the input set, capturing the compatibility between input instances and state variable
- Use RNN to model the Sequential Decision Making process.

LEAP: Beam Search Decoding

- Starting from a <START> label.
- Keep the top-K prediction results.
- Move the prediction path into the candidate set when a <END> label is predicted.
- Terminates if there is no better prediction path.



LEAP: Reinforcement Fine-Tuning



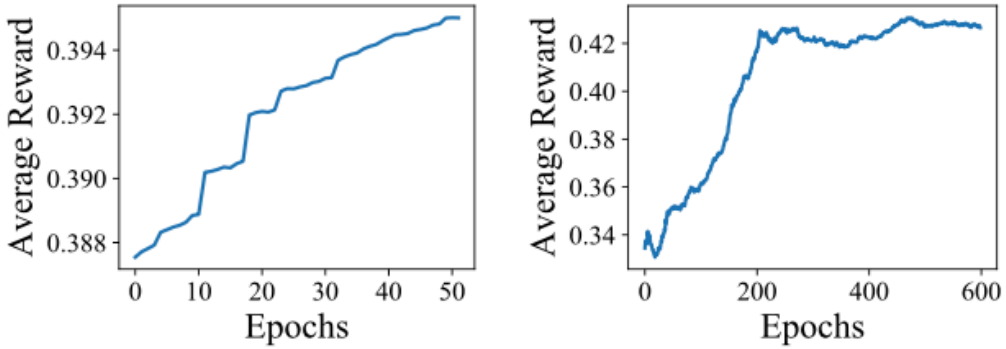
LEAP: Experimental Results

- Evaluated on real world EHR data from Sutter(heart disease and epilepsy patients) and MIMIC-3(ICU patients).
- Use the first and third level of GPI codes and the candidate drug set.

	Sutter		MIMIC-3	
Granularity	1	3	1	3
Rule-based	0.3207	0.2770	0.2753	0.2354
K-Most frequent	0.4283	0.3181	0.2609	0.2616
Softmax MLP	0.4908	0.3739	0.4897	0.3342
Classifier Chains	0.4839	0.3620	0.4621	0.3204
Basic LEAP	0.5270	0.3936	0.5107	0.3865
LEAP	**0.5341**	**0.4073**	**0.5582**	**0.4342**

LEAP: Reinforcement Fine-tuning

- Average reward w.r.t training epochs



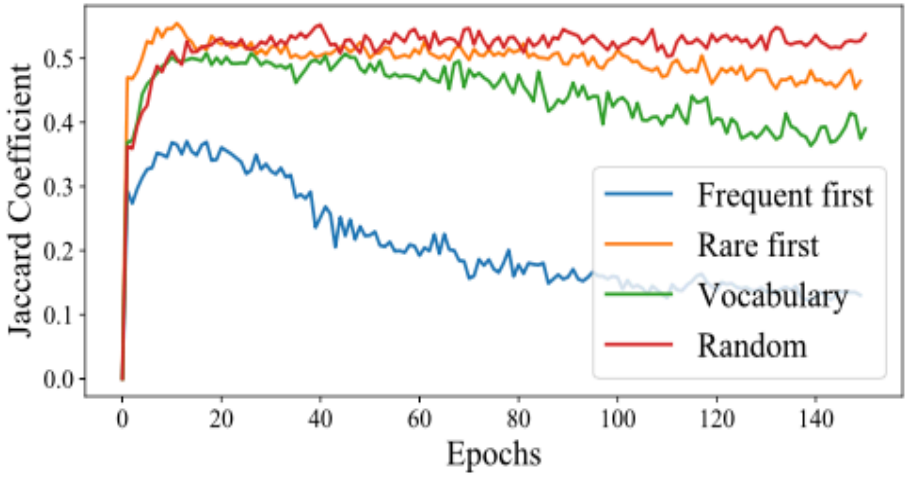
(a) Sutter Dataset **(b) MIMIC-3 Dataset**

- Performance of avoiding adverse drug interactions.

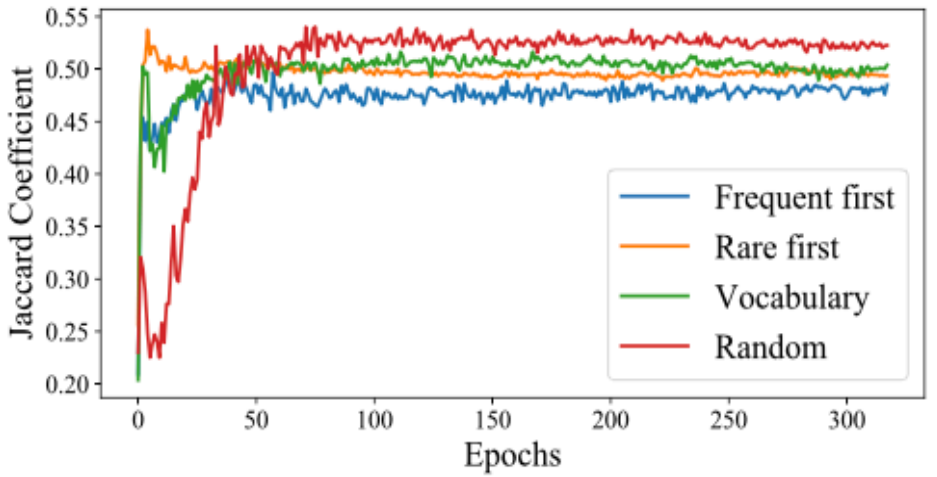
Method	Drug Interaction Rate
K-Most frequent	12.06%
Softmax MLP	3.51%
Basic LEAP	2.41%
LEAP	0.23%

LEAP: Different Order of Labels

- Frequent-first performs the worst.
- Rare-first converges fast.
- Random-shuffle is the most robust one.



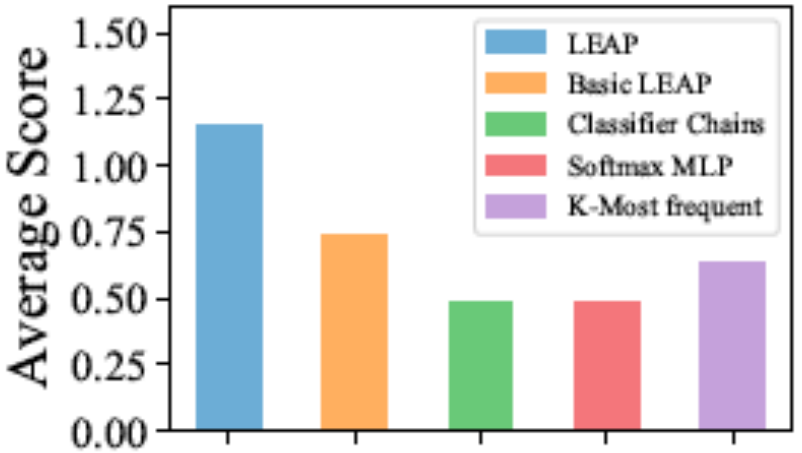
(a) Sutter Dataset



(b) MIMIC-3 Dataset

LEAP: Qualitative Evaluation

- The results are manually scored by a clinical expert
  - 2: Completely covers the diseases with no conflict.
  - 1: Partially covers the diseases (>50%) with no conflict.
  - 0: Conflict or incomplete (<50%)



- Case study:

Diagnosis	Methods	Recommended Treatments
Type 2 diabetes Hyperlipidemia Depressive disorder Hypertension	*K*-Most frequent	PEG KCl Bicarb, Quinapril, Pravastatin, Metformin, Paroxetine
	Softmax MLP	Metformin
	Basic LEAP	Metformin, Quinapril, Pravastatin, Paroxetine
	LEAP	Metformin, Amiloride/HCTZ, Fenofibrate, Paroxetine
Depressive Disorder Acute bronchitis Imbalance (gait)	*K*-Most frequent	Azithromycine, Privastatin, Paroxetine
	Softmax MLP	Paroxetine, Azithromycin
	Basic LEAP	Azithromycin
	LEAP	Azithromycin, Dextromethorphan-K Guaiacolsulfonate, Paroxetine

Table 3: Example Recommended Treatments by Different Methods

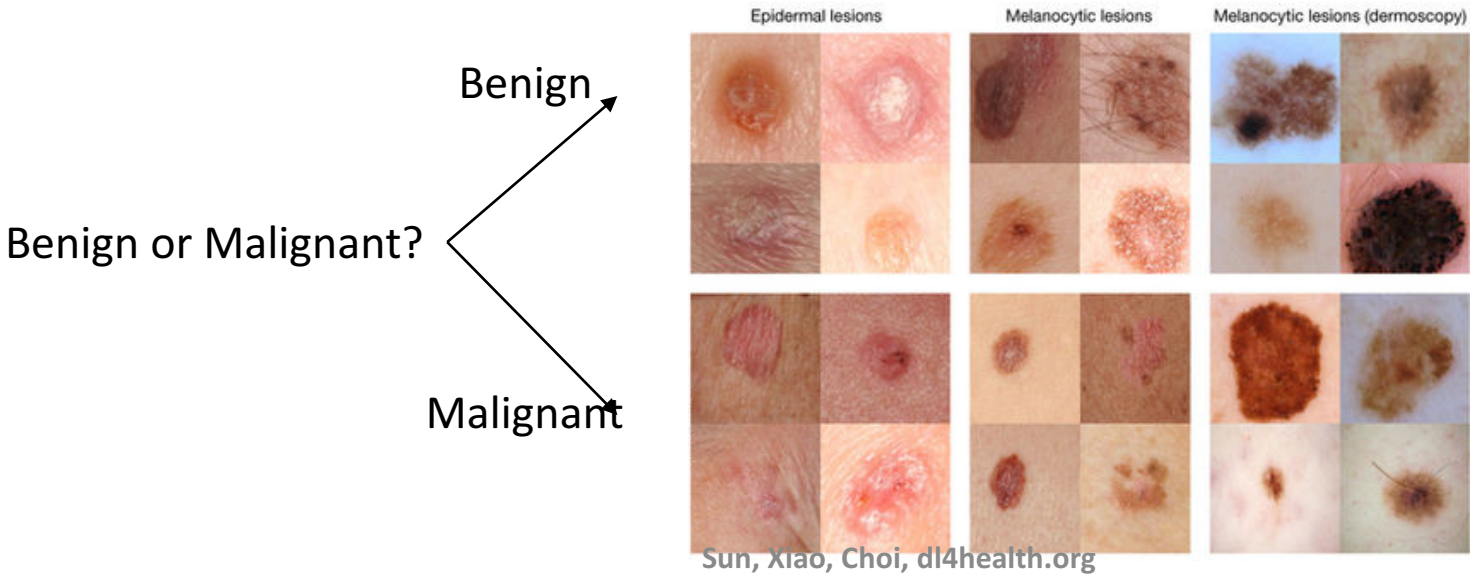
Dermatologist-level classification of skin cancer with deep neural networks

A. Esteva, B. Kuprel, R.A. Novoa, J. Ko, S.M. Swetter, H.M. Blau, S. Thrun

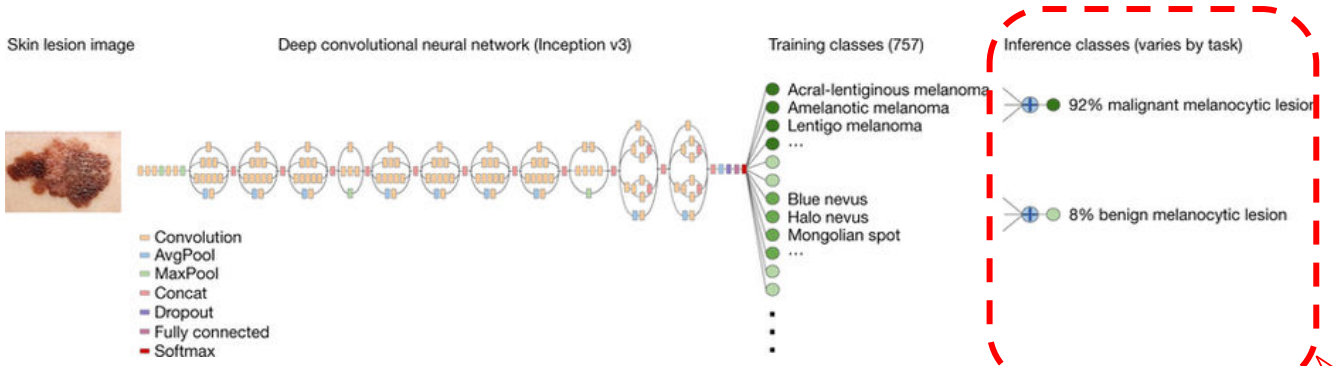
Nature 542, p115-118 (2017)

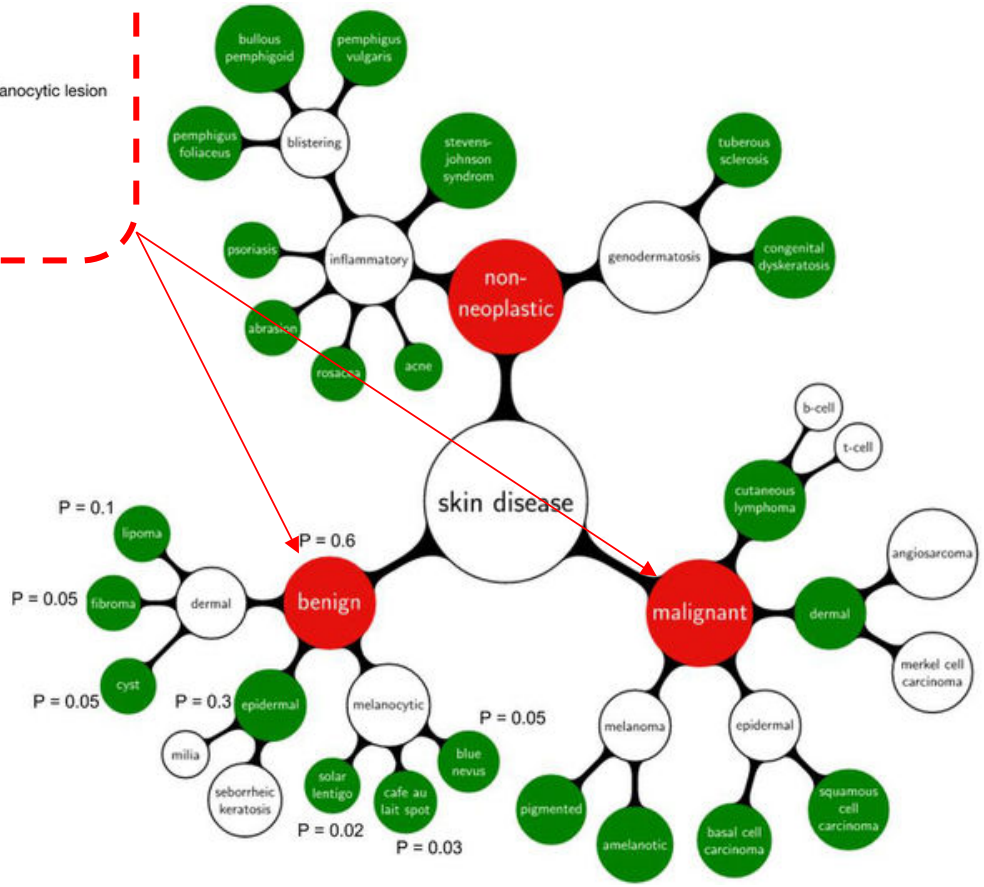
Detecting Skin Cancer

- Dermatologist-level classification of skin cancer with deep neural networks
  - Esteva, Andre, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau, and Sebastian Thrun, Nature 2017
- Given clinical images, classify
  - Keratinocyte carcinomas VS benign seborrheic keratoses
  - malignant melanomas VS benign



Model architecture: inception v3



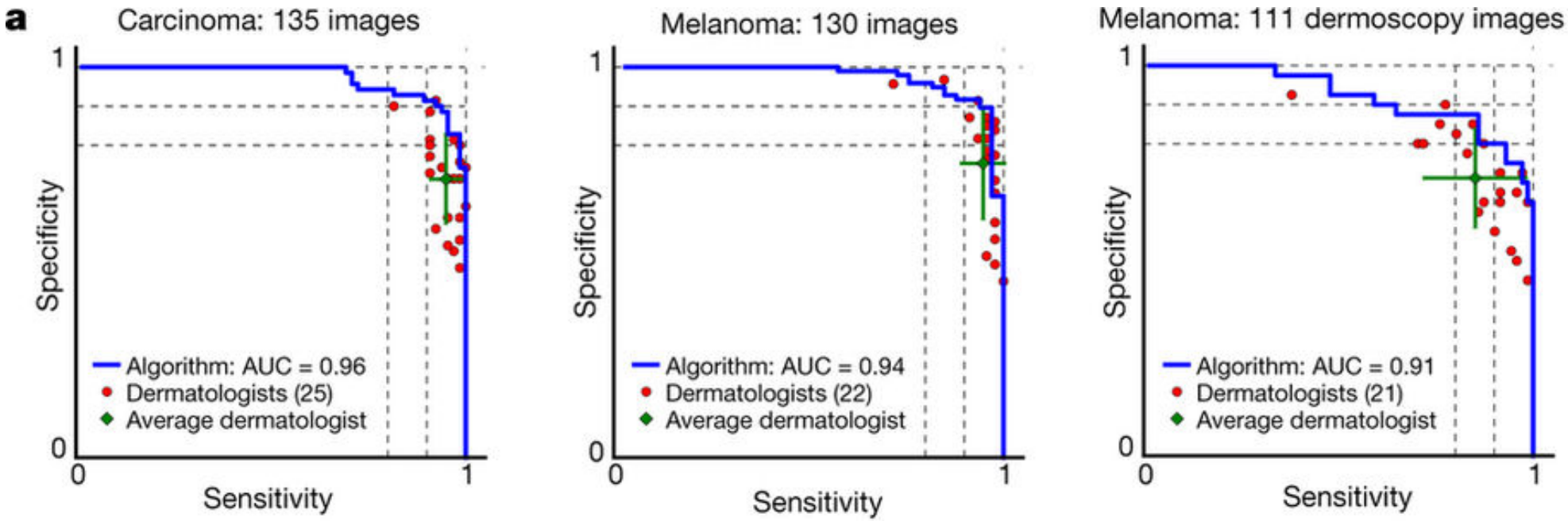


Sun, Xiao, Choi, dl4health.org

Disease hierarchy curated by experts

Model performance

- Better than average board-certified dermatologists



Sequential Prediction

- Dr. AI/JAMIA-HF-RNN (EHR- multilabel-binary)
- RETAIN (EHR-interpretability)
- RAIM (multimodal)
- CONTENT (readmission prediction, PLOS One)

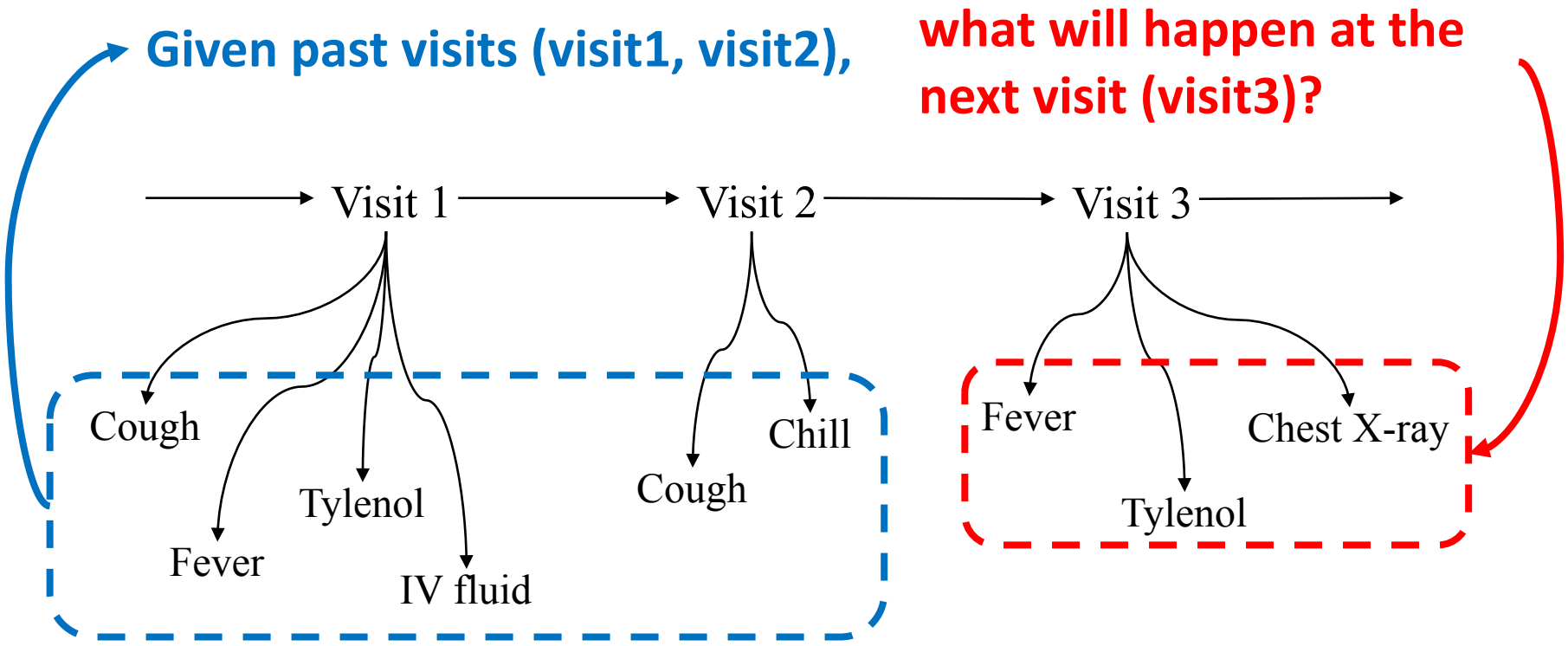
Doctor AI: Predicting Clinical Events via Recurrent Neural Networks

Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, Jimeng Sun

Machine Learning for Healthcare Conference, 2016

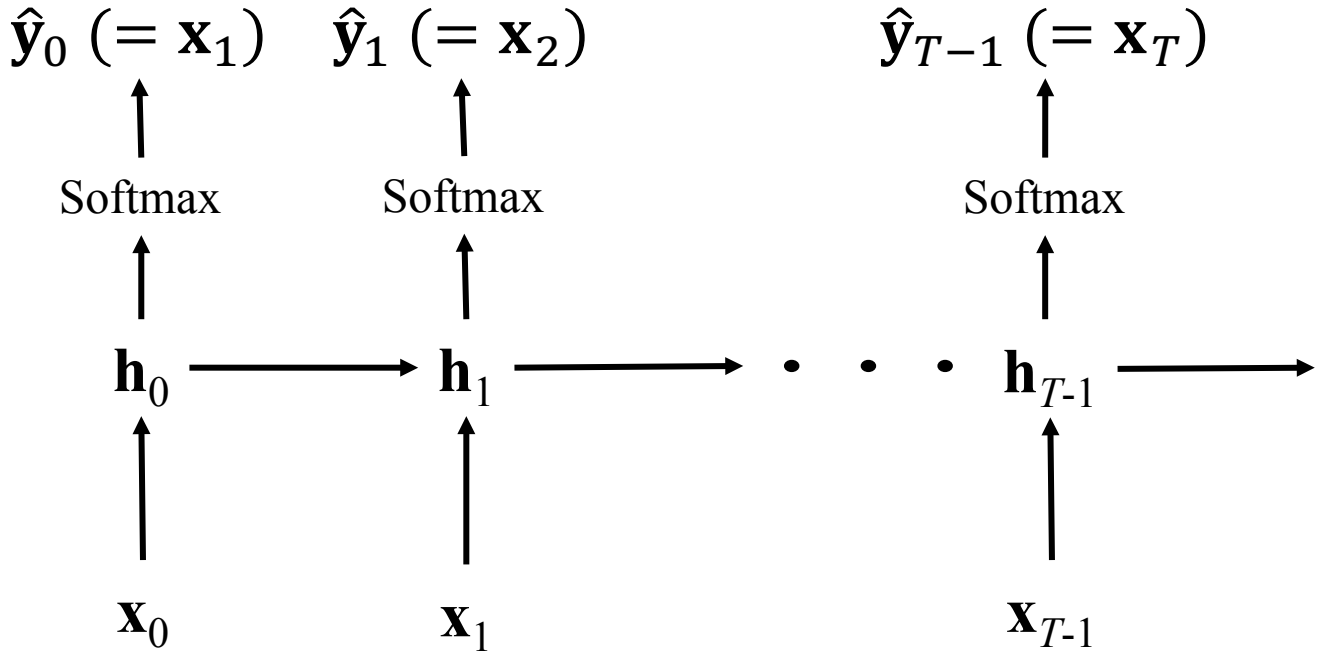
Doctor AI: Background

- Disease progression modeling



Doctor AI: Model

- Feed visits into the RNN
  - One visit at each timestep.
  - Predict next events at each timestep.

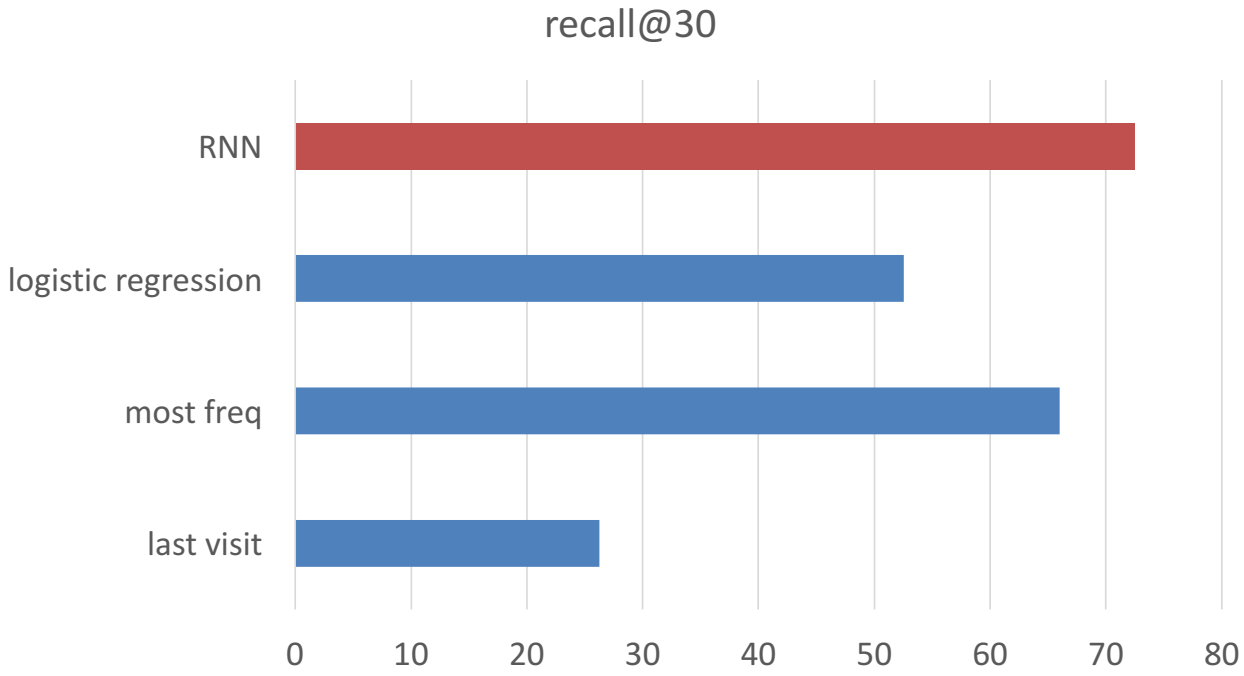


Doctor AI: Data

- Data
  - 260K patients from Sutter Health
  - Patient records over 10 years
  - Input codes
    - Diagnosis codes, medication codes, procedure codes (38,000 codes)
  - Output labels
    - 1,183 diagnosis codes

Doctor AI: Sequential Prediction

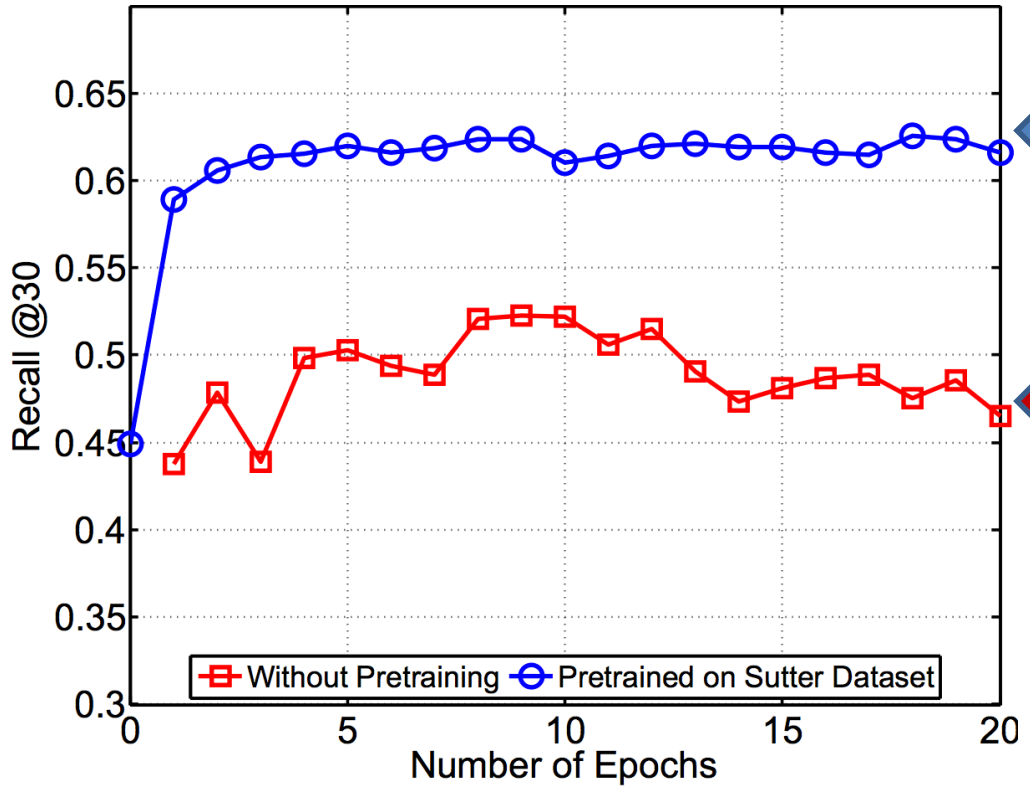
- Predicting diagnoses in the next visit



$$\text{top-}k \text{ recall} = \frac{\#\text{ of true positives in the top } k \text{ predictions}}{\#\text{ of true positives}}$$

Doctor AI: Knowledge Transfer

- Generalize RNN model from one hospital to another



Perform better when warm start the existing model

Perform worse when cold-start from random initialization

RETAIN: An Interpretable Predictive Model for Healthcare using Reverse Time Attention Mechanism

Edward Choi, Mohammad Taha Bahadori, Joshua A. Kulas, Andy Schuetz, Walter F. Stewart, Jimeng Sun

NIPS' 16

RETAIN: Interpretable Models

What do we mean by interpretability of a model?

Three categories of models:

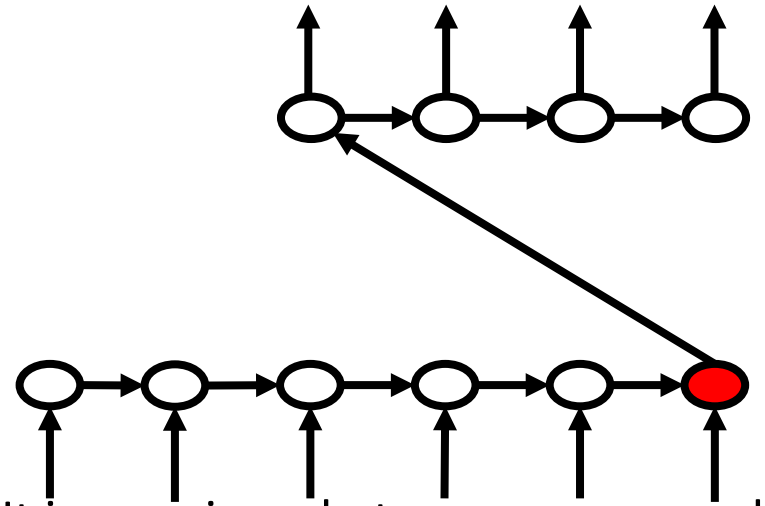
- Rule based: *e.g.* decision trees
- Case based: *e.g.* nearest neighbor methods
- Risk factor based: *e.g.* sparse linear regression

- Temporal models? Latent variable models

RETAIN: Neural Attention Mechanism

Regular Machine Translation

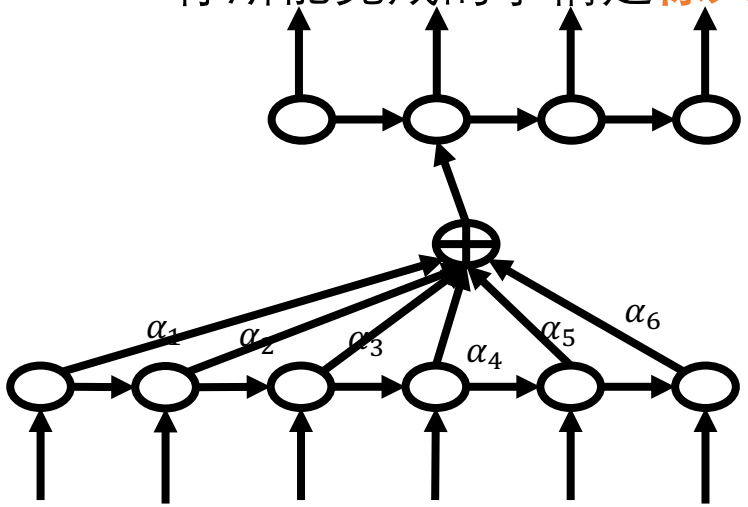
如果你不在乎谁获得了荣誉，
你所能完成的事情是**惊人的**。



It is amazing what you can accomplish
if you do not care who gets the credit

Neural Attention Mechanism

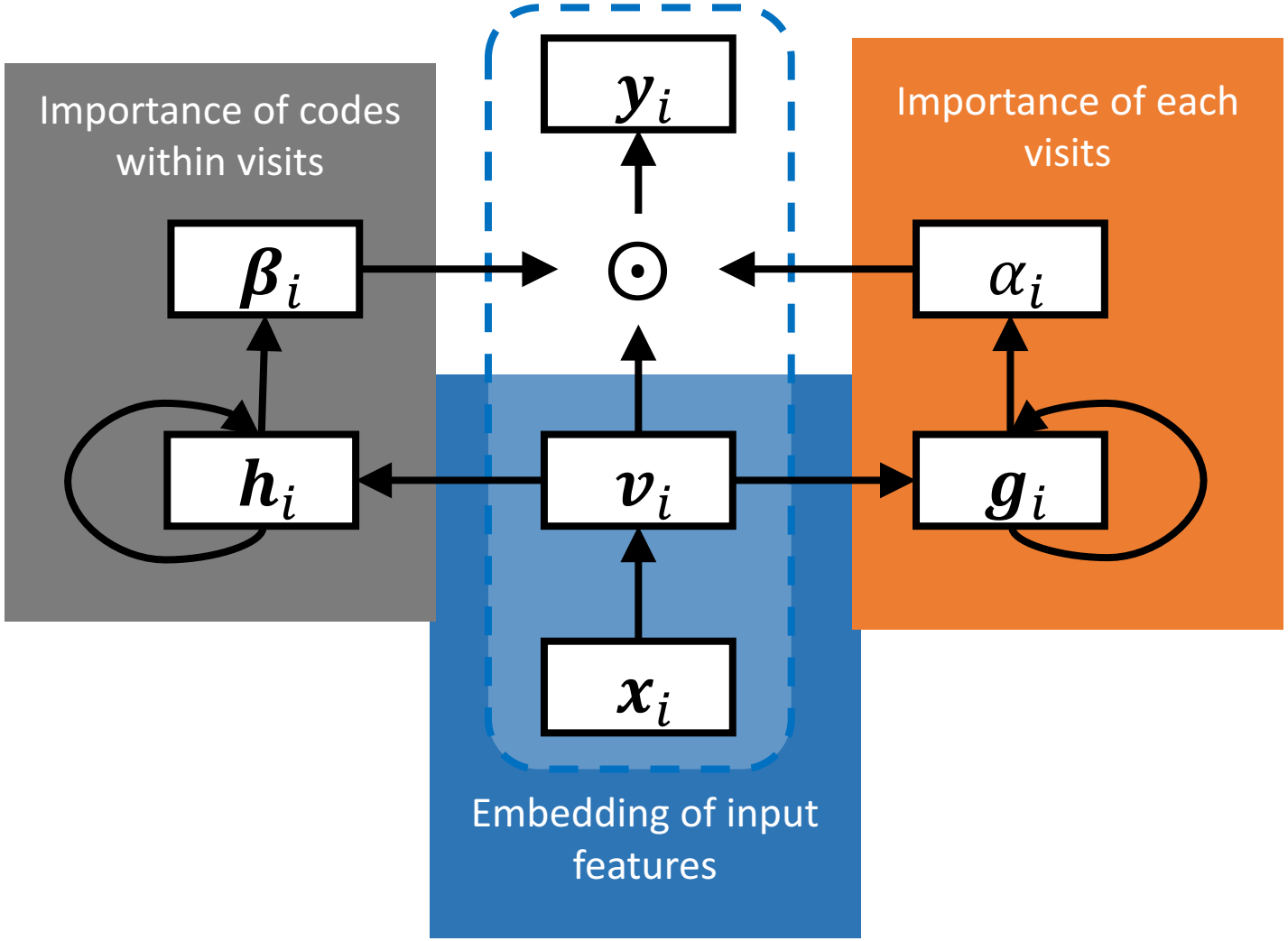
如果你不在乎谁获得了荣誉，
你所能完成的事情是惊人的。



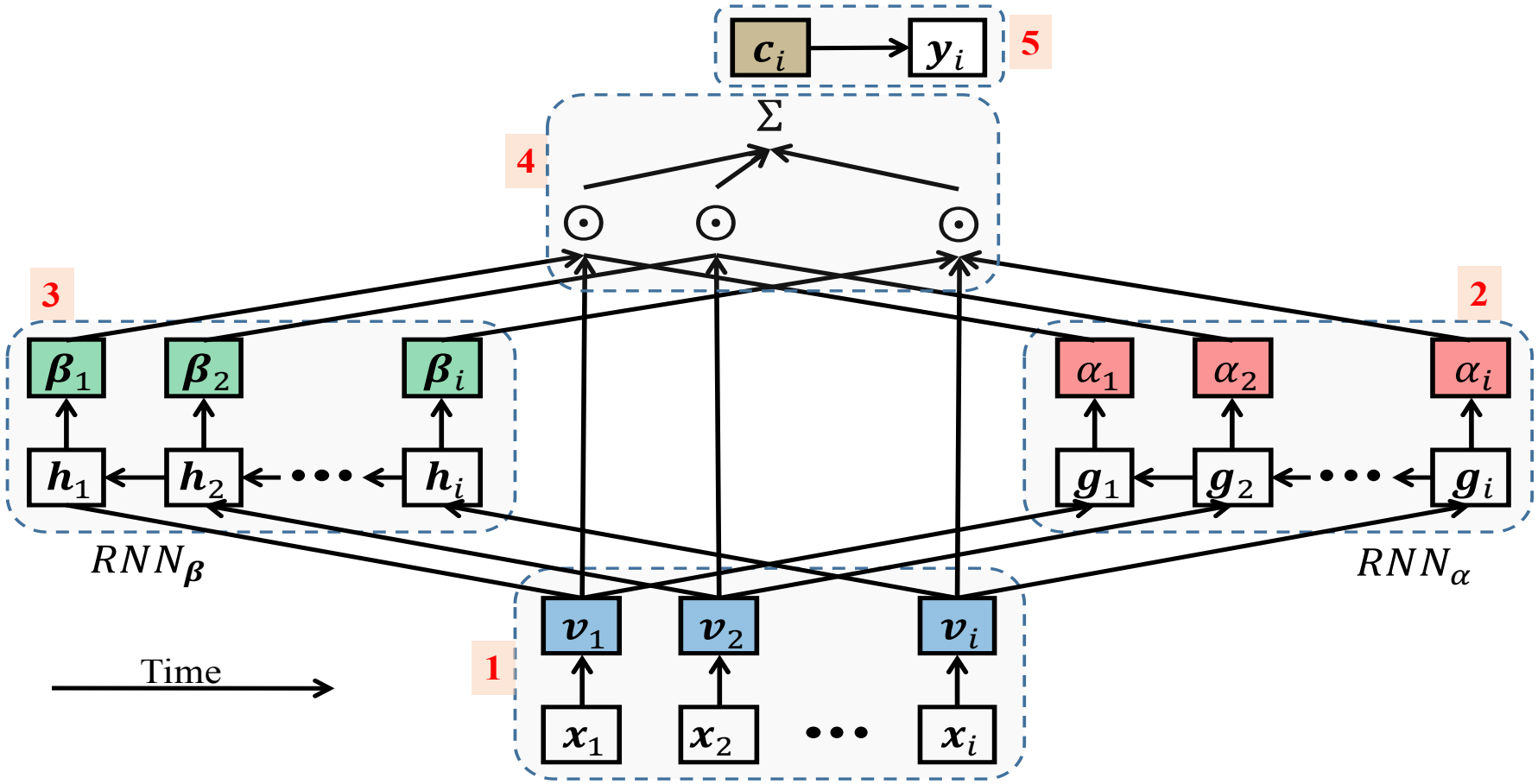
It is amazing what you can accomplish
if you do not care who gets the credit

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2014. "Neural Machine Translation by Jointly Learning to Align and Translate."

RETAIN: REverse Time AttentIoN model

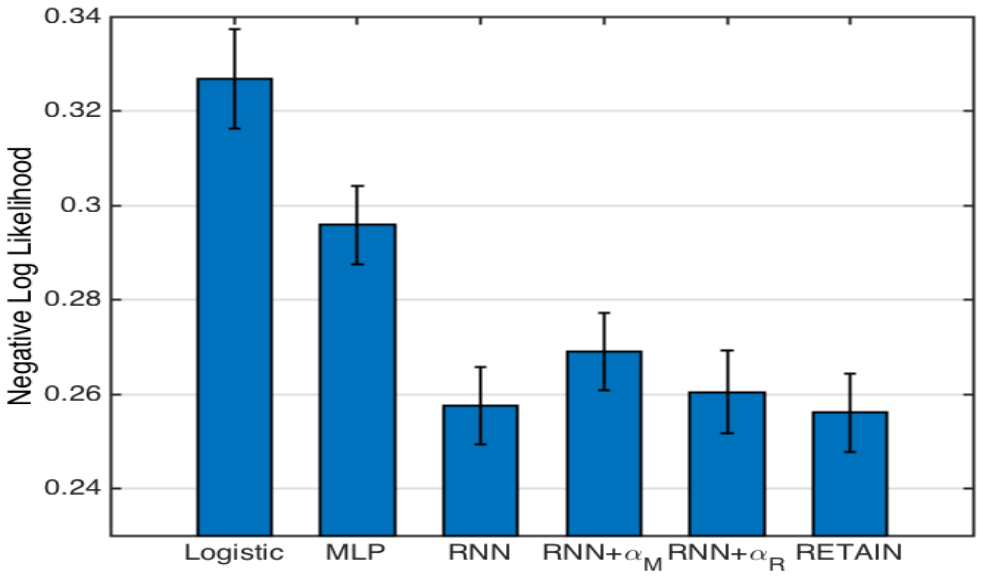


Details of RETAIN

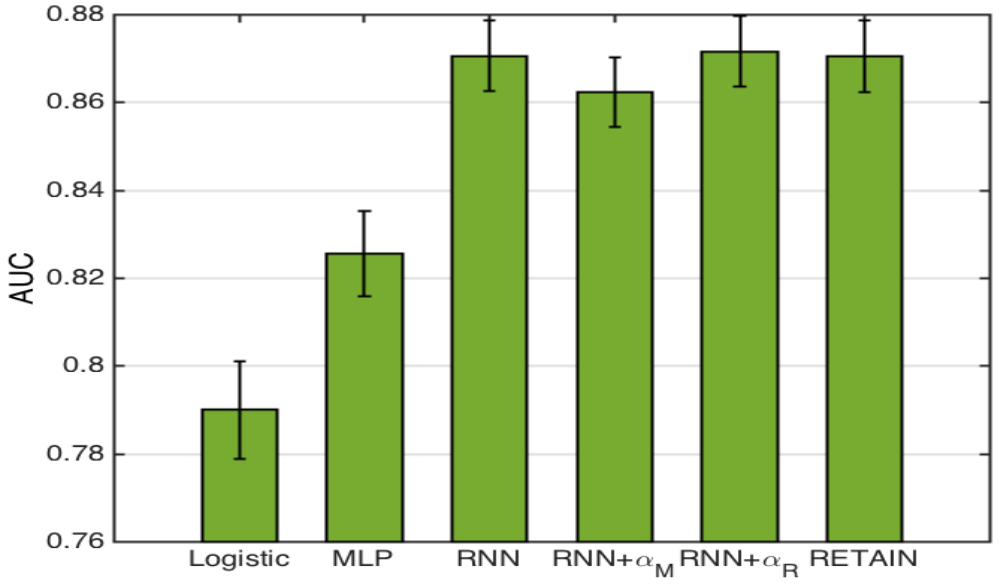


RETAIN: Heart Failure Prediction Results

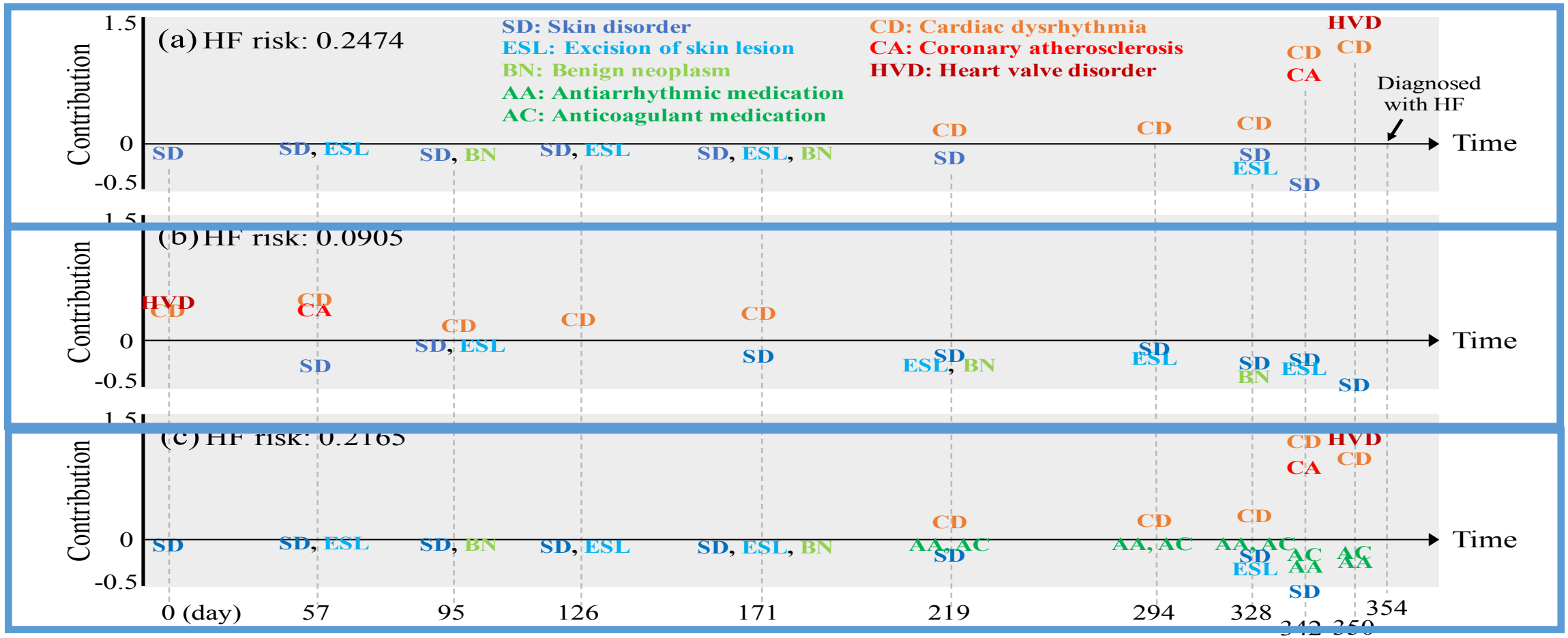
Negative Log Likelihood on Test Set



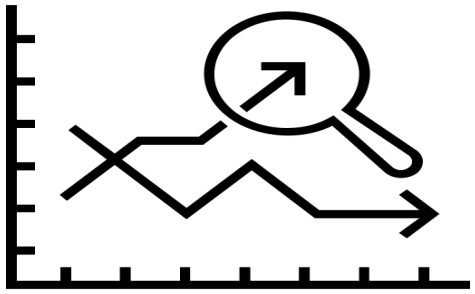
Classification AUC



RETAIN: Model Interpretability



RETAIN: Summary



- Challenge: Deep learning models are often difficult to interpret
- RETAIN is a temporal attention model on electronic health records
  - Great predictive power
  - Good interpretation

RAIM: Recurrent Attentive and Intensive Model of Multimodal Patient Monitoring Data

Yanbo Xu, Siddharth Biswal, Shriprasad Deshpande, Kevin Maher, and Jimeng Sun

KDD' 18

RAIM: Motivation

Recurrent

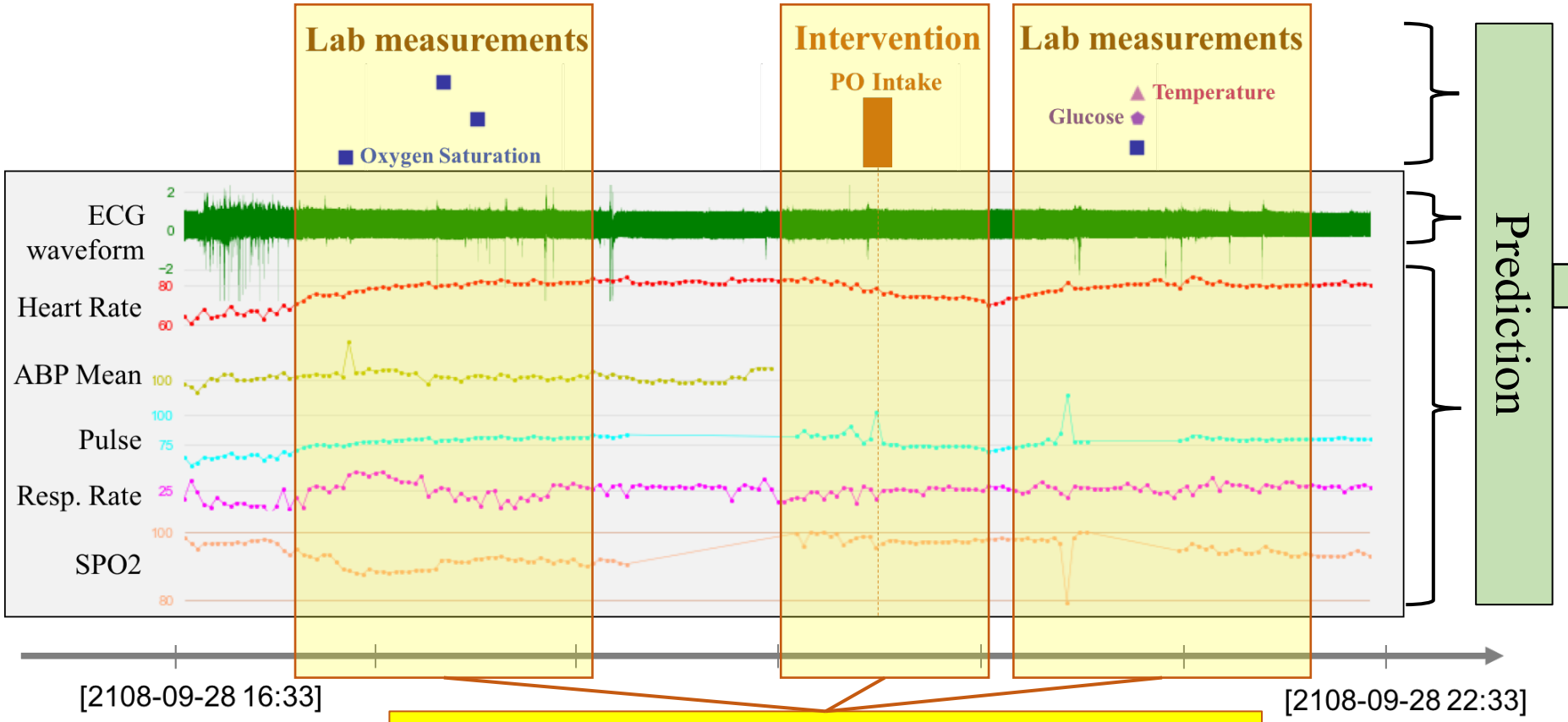
Attentive

Intensive

Model

1. Multi-channel high-density signal processing
2. Multiple data modalities
3. Interpretability

Challenges: interpretable predictions

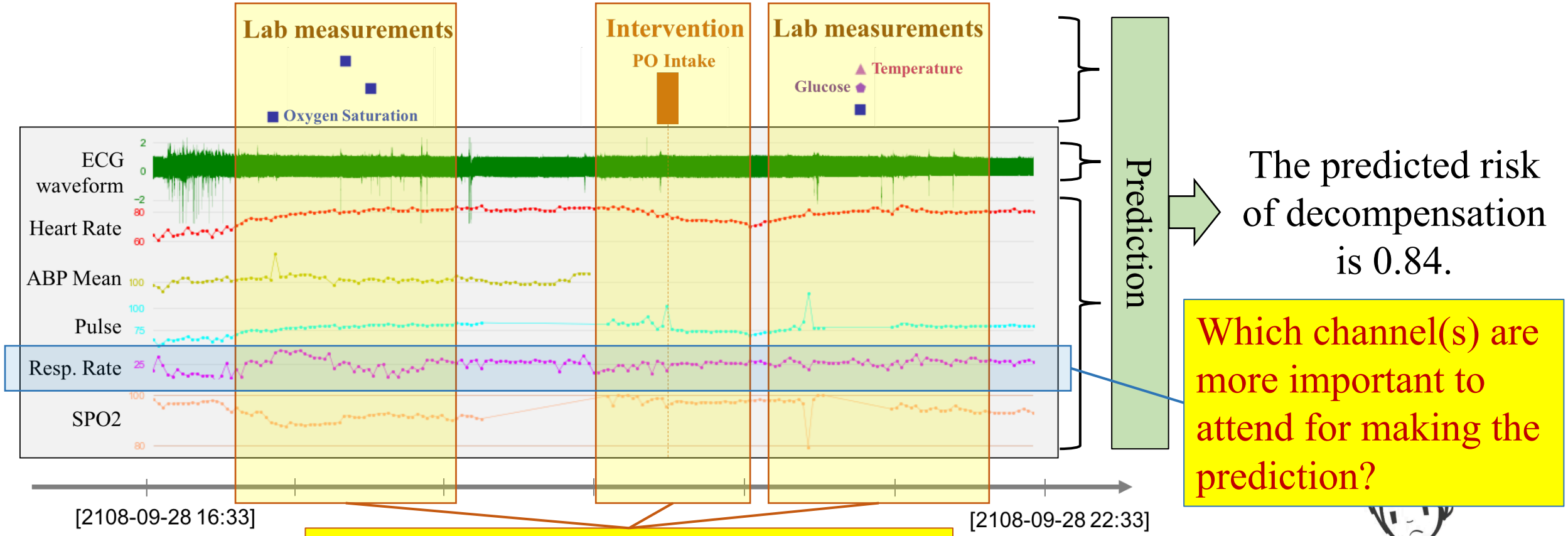


The predicted risk of decompensation is 0.84.

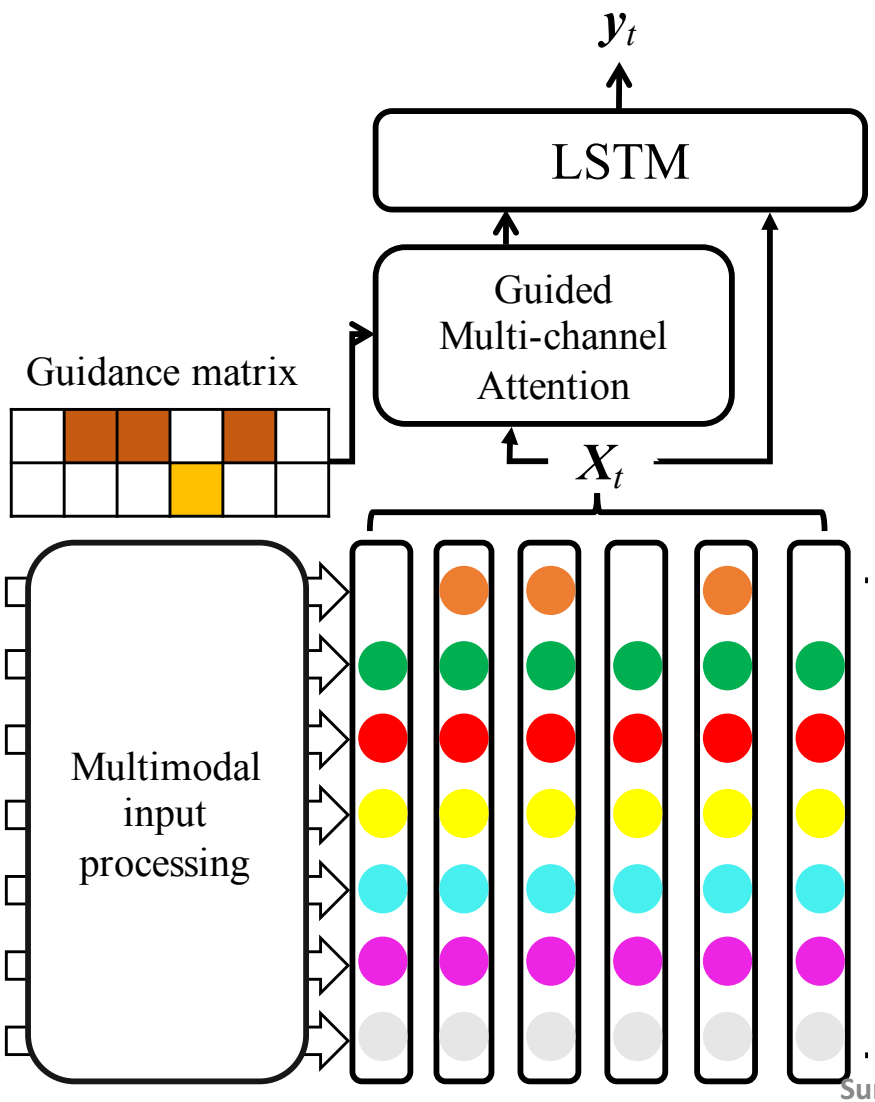
An example CU

More informative and probably influence most on the final prediction?

Challenges: interpretable predictions

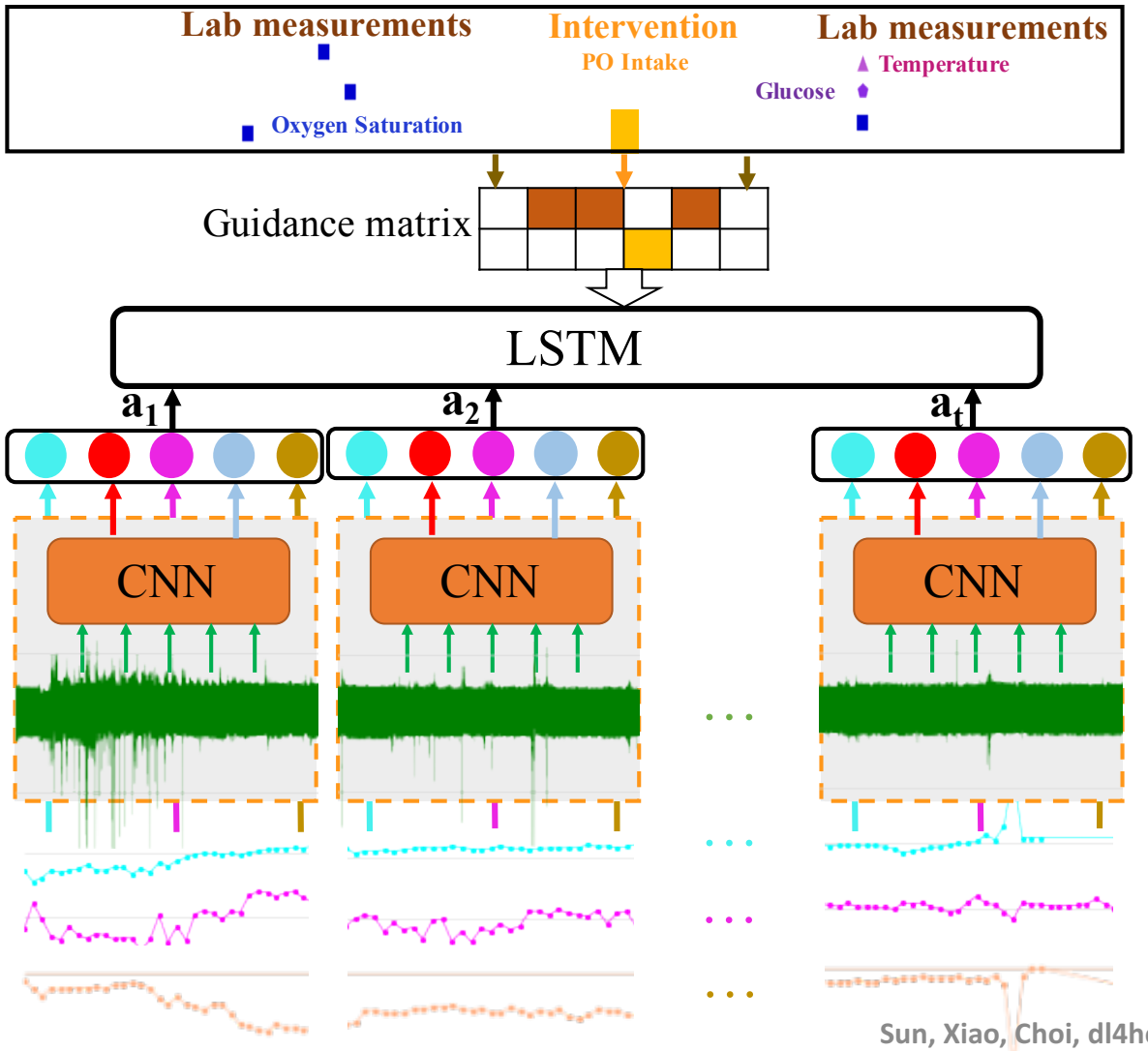


RAIM: A recurrent attentive and intensive model



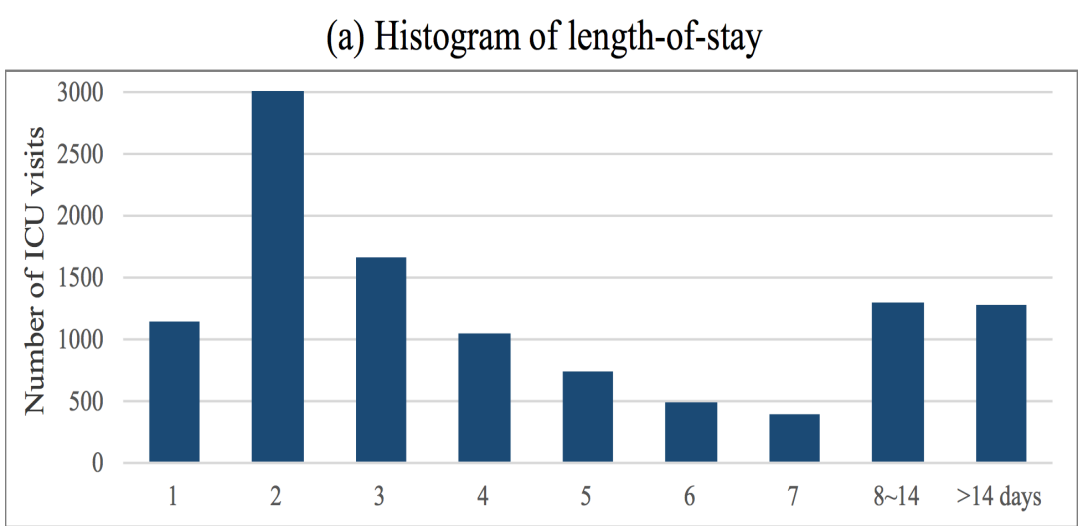
- ✓ Multimodal input integrating
  - Irregular and regular discrete data
  - Continuous streaming data
- ✓ High-density signal processing
  - Recurrent deep neural networks
- ✓ Attention-based interpretable modeling
  - Multi-channel attention
  - Guided attention

RAIM: Multimodal input processing

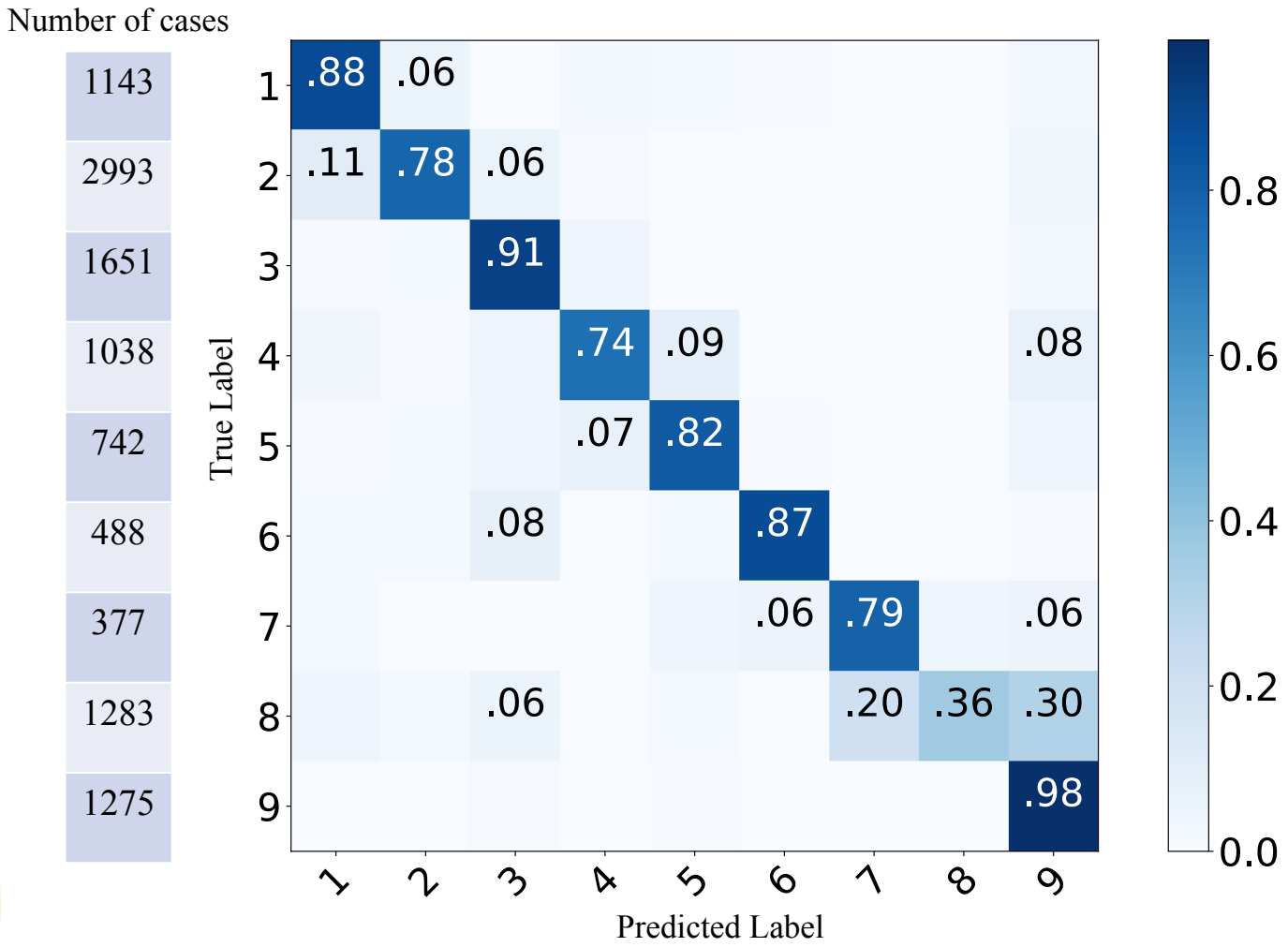


- Labs and medications are prioritized as a guidance matrix telling which time steps are more important to attend.
- RNN is applied for encoding long-term sequential behaviors.
- CNNs are applied for encoding short-term dense signals.
- Regularly recorded discrete values are unified into input vectors.

RAIM: Predicting Length of Hospital Stay



	Decompensation			Length of Stay	
	AUC-ROC	AUC-PR	Accuracy	Kappa	Accurac
CNN (ECG)	87.84%	21.56%	88.38%	0.7681	82.16%
CNN-RNN	87.45%	23.19%	88.25%	0.8027	85.34%
CNN-AttRNN	88.19%	25.81%	89.28%	0.8186	84.89%
RAIM-0	87.81%	25.56%	88.96%	0.8125	85.84%
RAIM-1	88.25%	25.61%	88.91%	0.8215	86.74%
RAIM-2	88.77%	26.85%	90.27%	0.8217	85.21%
RAIM-3	90.18%	27.93%	90.89%	0.8291	86.82%

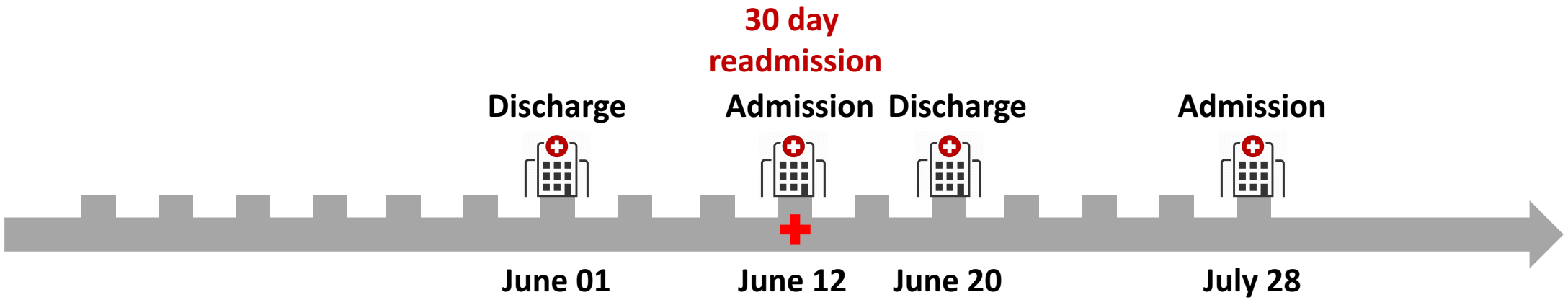


Readmission prediction via deep contextual embedding of clinical concepts

Cao Xiao , Tengfei Ma , Adji B. Dieng, David M. Blei, Fei Wang

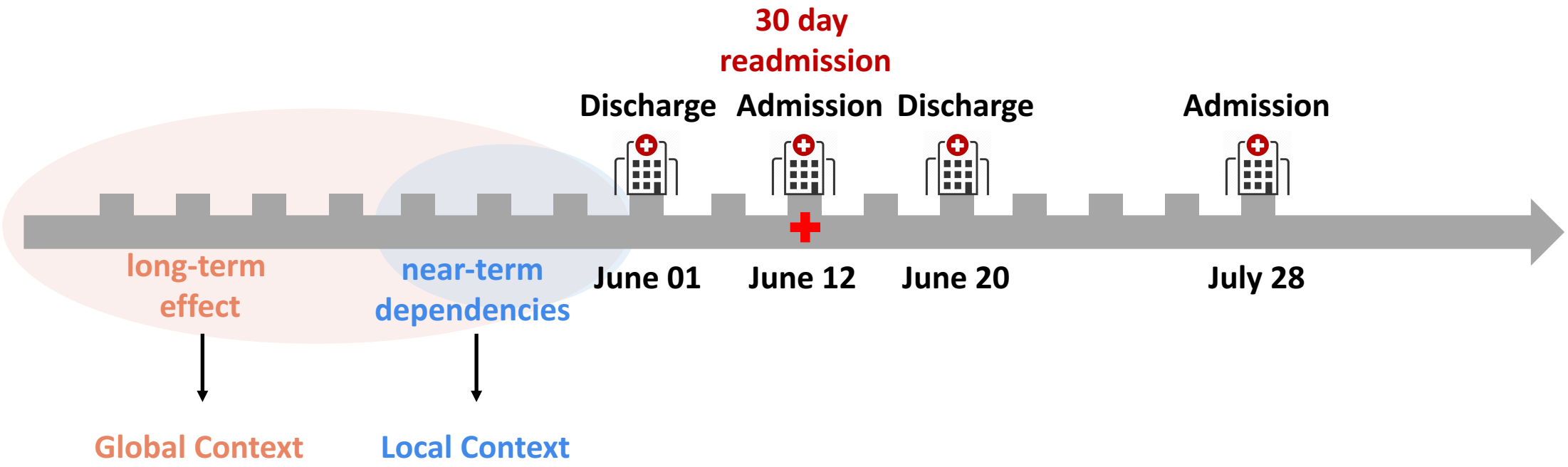
PLOS One, 2018

Hospital Readmission



- In the U.S., involve 17.6% of hospital-admitted patients, accounted for $17:9 billion Medicare spending per year, while 76% of them are potentially avoidable.
- Targeted follow-ups that focus on patients with high risks of readmissions could help reduce readmission rate.
- Readmission risks are hard to predict due to its complex entanglements with the patients' health conditions.

CONTENT: Deep Contextual Embedding of Clinical Concepts



CONTENT is an end-to-end hybrid deep learning model structure that combines **topic modeling** and **Recurrent Neural Network** (RNN) to distill the complex knowledge hidden in those contexts.

The CONTENT Model

Input patient health record

Generate global context $\boldsymbol{\theta}$

For tth visit, compute $\boldsymbol{h}_t = GRU(\boldsymbol{V}_{t-1}, \boldsymbol{h}_{t-1}; \boldsymbol{W}_v, \boldsymbol{W}_h)$

Compute readmission indicator $y_t \sim \sigma(\boldsymbol{Q}^t \boldsymbol{h}_t + \boldsymbol{B}_t^T \boldsymbol{\theta})$ where $\boldsymbol{B}_t = \frac{1}{M_t} \sum \boldsymbol{b}_m$, $\boldsymbol{b}_m$ is the latent topic vector for medical code m in this visit, M_t is the number of codes in this visit.

CONTENT: Model Inference

Original Objective

$$\log p(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{\Theta}) = \log \int p(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{\Theta}, \boldsymbol{\theta}) p(\boldsymbol{\theta}) d\boldsymbol{\theta}.$$

↓

Approximate Objective

$$\log p(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{\Theta}) = D_{KL}(q(\boldsymbol{\theta})||p(\boldsymbol{\theta}|\boldsymbol{y})) + ELBO.$$

↓

New Objective

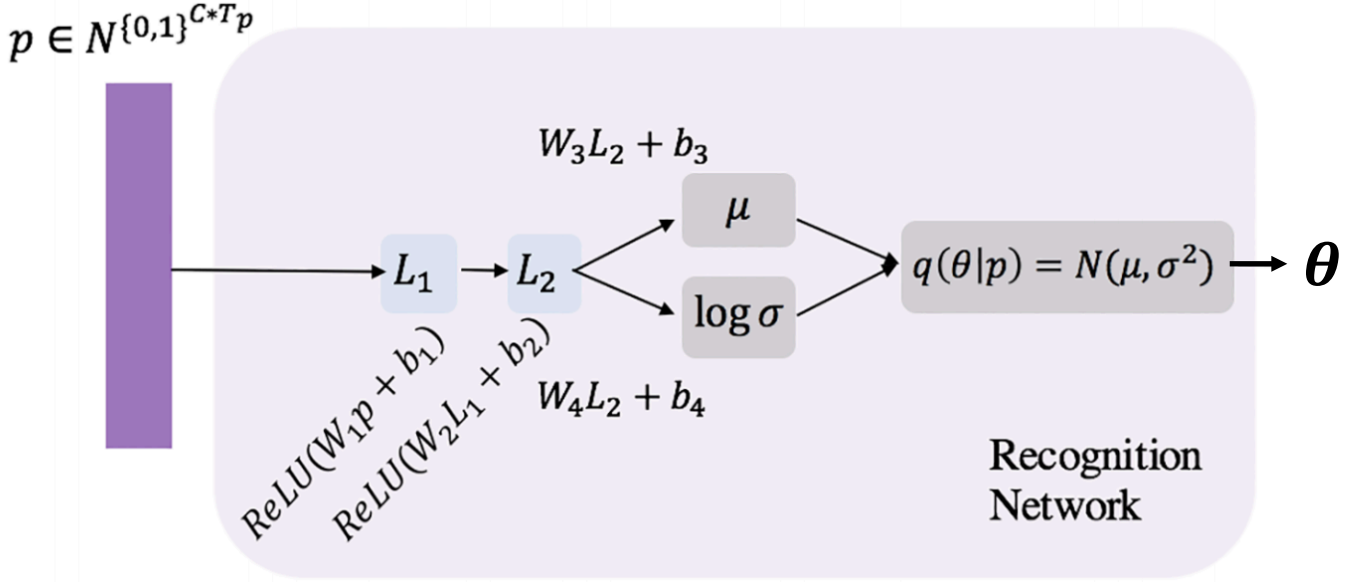
$$ELBO = E_{q(\boldsymbol{\theta})}[\log p(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{\theta}, \boldsymbol{\Theta}) + \log p(\boldsymbol{\theta}) - \log q(\boldsymbol{\theta})] \leq \log p(\boldsymbol{y}|\boldsymbol{h}, \boldsymbol{\Theta})$$

↓

Inference Network

CONTENT: Inference Network $q(\boldsymbol{\theta})$

Input: Patient Representation Matrix



Generate Global Context Vector θ

$$\boldsymbol{r}_1 = ReLU(\boldsymbol{W}_{r_1}\boldsymbol{p} + \boldsymbol{b}_{r_1})$$

$$\boldsymbol{r}_2 = ReLU(\boldsymbol{W}_{r_2}\boldsymbol{r}_1 + \boldsymbol{b}_{r_2})$$

$$\boldsymbol{\mu}(\boldsymbol{p}) = \boldsymbol{W}_{\mu}\boldsymbol{r}_2 + \boldsymbol{b}_{\mu}$$

$$\log\boldsymbol{\sigma}(\boldsymbol{p}) = \boldsymbol{W}_{\sigma}\boldsymbol{r}_2 + \boldsymbol{b}_{\sigma}$$

$$q(\boldsymbol{\theta}|\boldsymbol{p}) = N(\boldsymbol{\mu}(\boldsymbol{p}), diag(\boldsymbol{\sigma}^2(\boldsymbol{p})))$$

$$\boldsymbol{\theta} \sim q(\boldsymbol{\theta}|\boldsymbol{p}).$$

CONTENT: Readmission Prediction

Dataset	Congestive Heart Failure
# patients	5, 393
# visits	455, 106
# events	1, 306, 685
Avg. # of visits per patient	84.4
Avg. # of events per patient	242.3
# of unique event codes	618

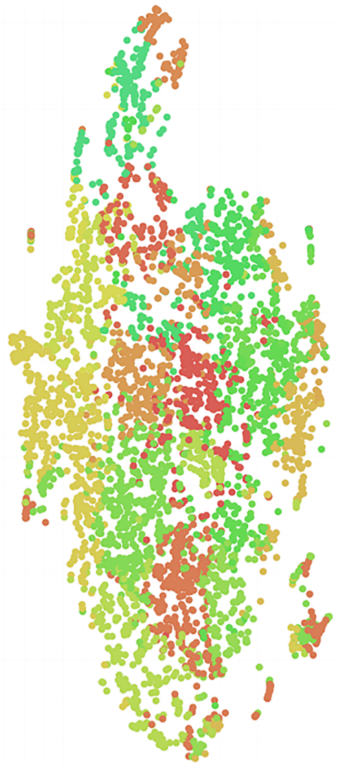
Data split

- training: 4000 patients
- validation: 700 patients
- testing: 693 patients

Method	PR-AUC	ROC-AUC	ACC
Word2vec+LR	0.3445±0.0204	0.5360±0.0246	0.6828±0.0120
Med2vec+LR	0.3836±0.0149	0.5937±0.0120	0.6915±0.0095
GRU	0.3862±0.0136	0.5998±0.0124	0.6856±0.0082
GRU+Word2Vec	0.3430±0.0157	0.5616±0.0157	0.6731±0.0091
RETAIN	0.3720±0.0148	0.5707±0.0140	0.6814±0.0111
CONTENT	0.3894±0.0153	0.6103±0.0130	0.6934±0.0090

CONTENT: Interpretability

Cluster 1 Cluster 2 Cluster 3 Cluster 4



Cluster 1: non-severe non-cardiac comorbidity group
(average # readmission = 13.20)

Count	Name of Clinical Event
51	anemia
30	disorder of joint
28	disorder of back
26	osteoarthrosis and allied disorders
19	symptoms involving respiratory system

Cluster 2: transplant surgery patient comorbidity group
(average # readmission = 33.00)

Count	Name of Clinical Event
78	organ or tissue replaced by transplant
64	after-surgery care
63	acute renal failure
48	pneumonia
24	disorders of urethra and urinary tract

Cluster 3: cardiac-cancer comorbidity group
(average # readmission = 21.09)

Count	Name of Clinical Event
965	diabetes mellitus
622	chronic airways obstruction
489	disorders of lipid metabolism
441	chronic ischemic heart disease
407	malignant neoplasm of female breast

Cluster 4: traumatic brain injury comorbidity group
(average # readmission = 16.45)

Count	Name of Clinical Event
209	hypertensive heart disease
196	anemia
126	symptoms involving nervous and musculoskeletal system
125	intracranial injury
114	symptoms involving head and neck

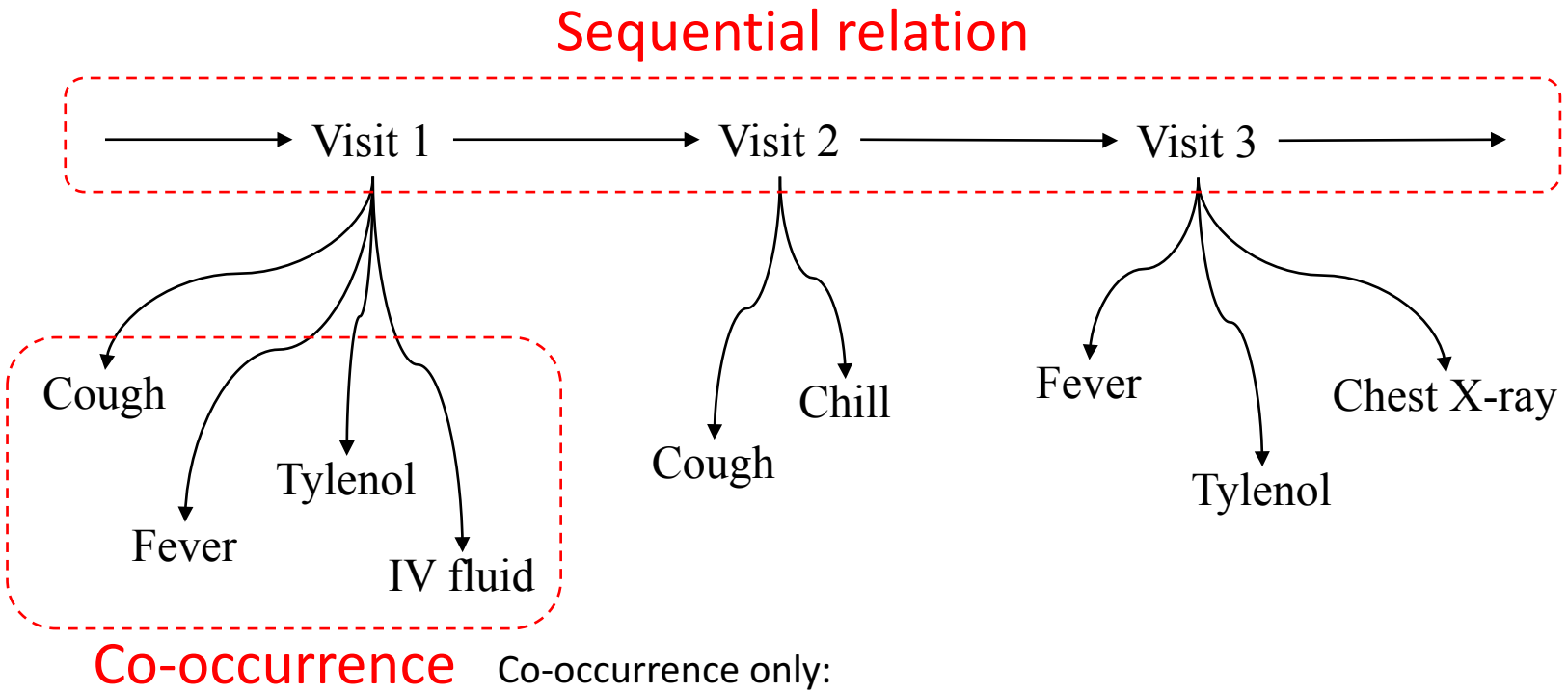
Med2Vec: Multi-layer Representation Learning for Medical Concepts

Edward Choi, Mohammad T. Bahadori, Elizabeth Searles, Catherine Coffey, Jimeng Sun

KDD'16

Med2Vec: Background

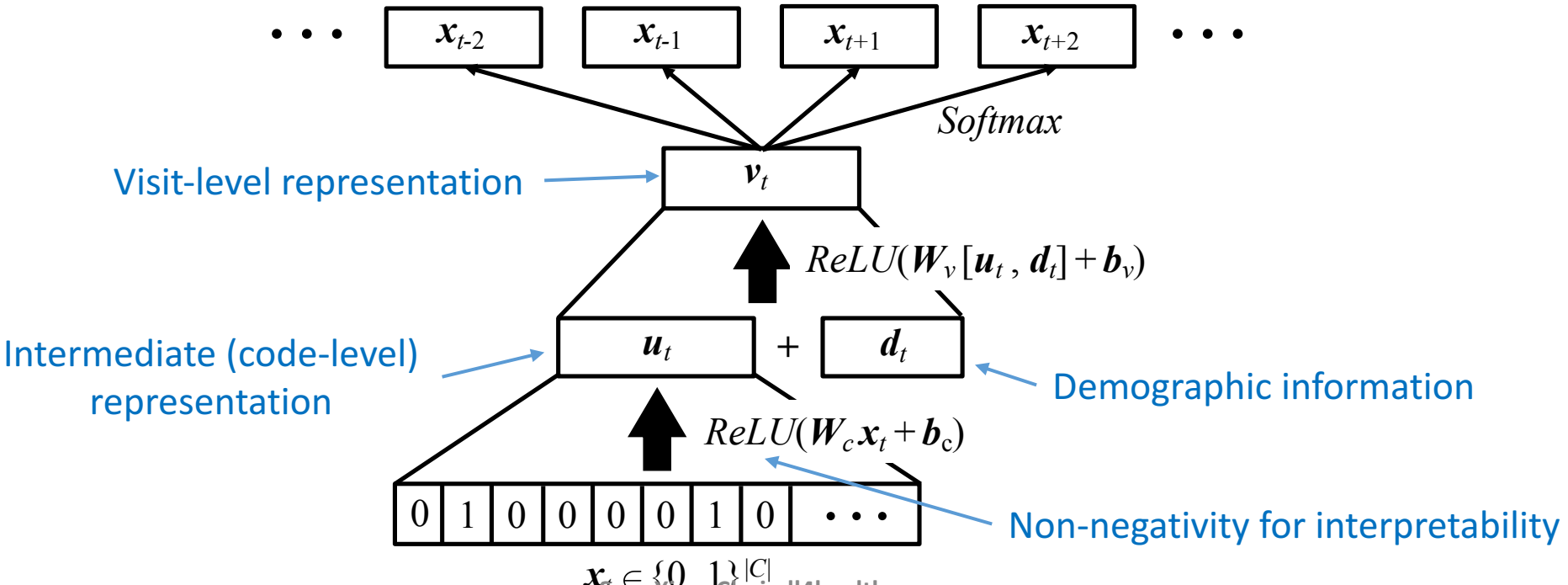
- Learn good representations of medical concepts
  - Diagnosis/medication/procedure codes
- Utilize 2-level structure of EHR



Co-occurrence only:
Choi, Youngduck, Chill Yi-I. Chiu, and David Sontag.
"Learning low-dimensional representations of medical concepts.", AMIA 2016

Med2Vec: Model

- Model architecture
  - Exploit two-layer structure of longitudinal EHR
    - Intra-visit codes provide co-occurrence information
    - Visit sequence provides sequential information



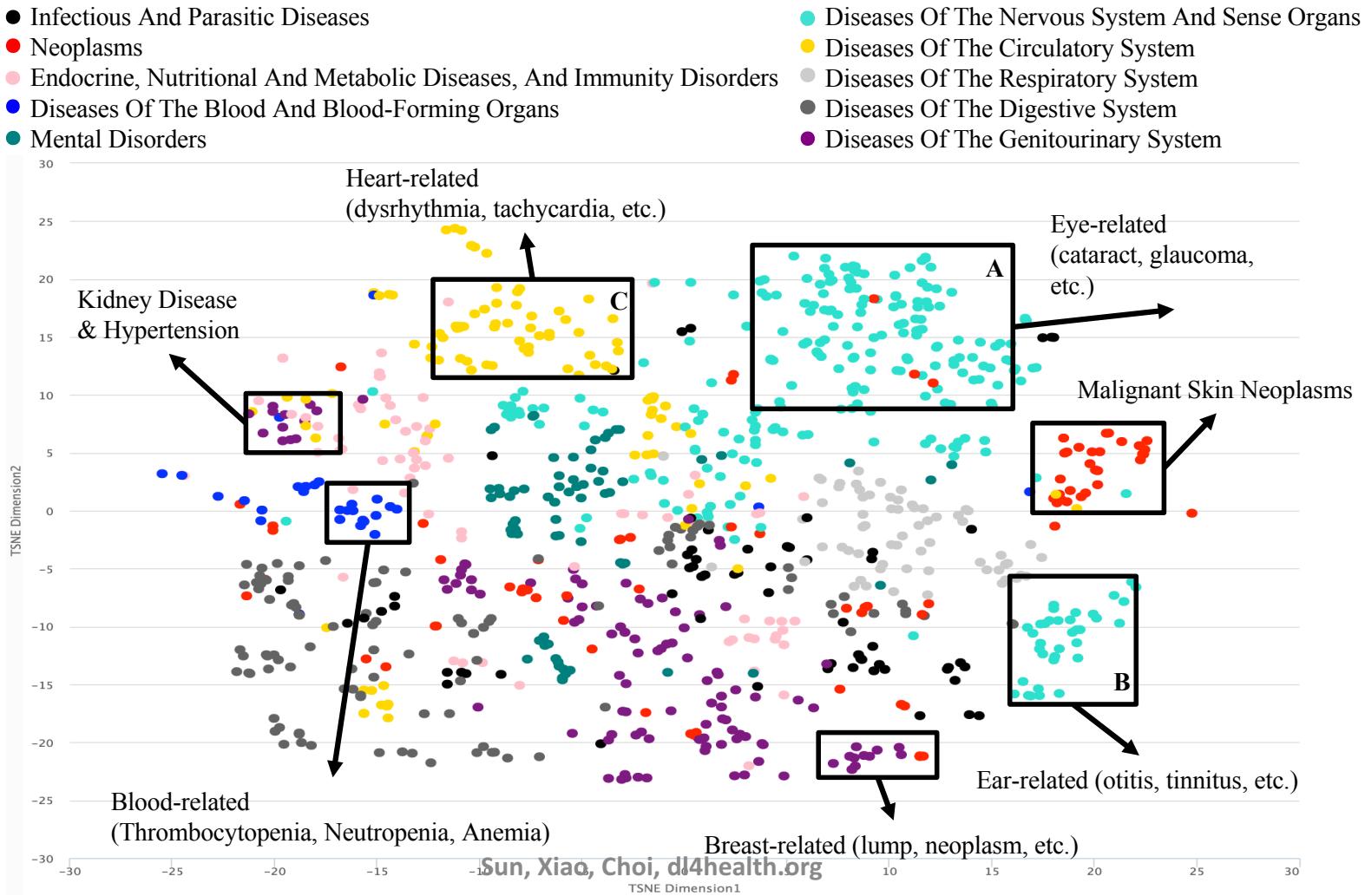
Med2Vec: Data

- Data
  - Children's Healthcare of Atlanta

Dataset	**CHOA**
# of patients	550,339
# of visits	3,359,240
Avg. # of visits per patient	6.1
# of unique medical codes	28,840
- # of unique diagnosis codes	10,414
- # of unique medication codes	12,892
- # of unique procedure codes	5,534
Avg. # of codes per visit	7.88
Max # of codes per visit	440
(95%, 99%) percentile # of codes per visit	(22, 53)

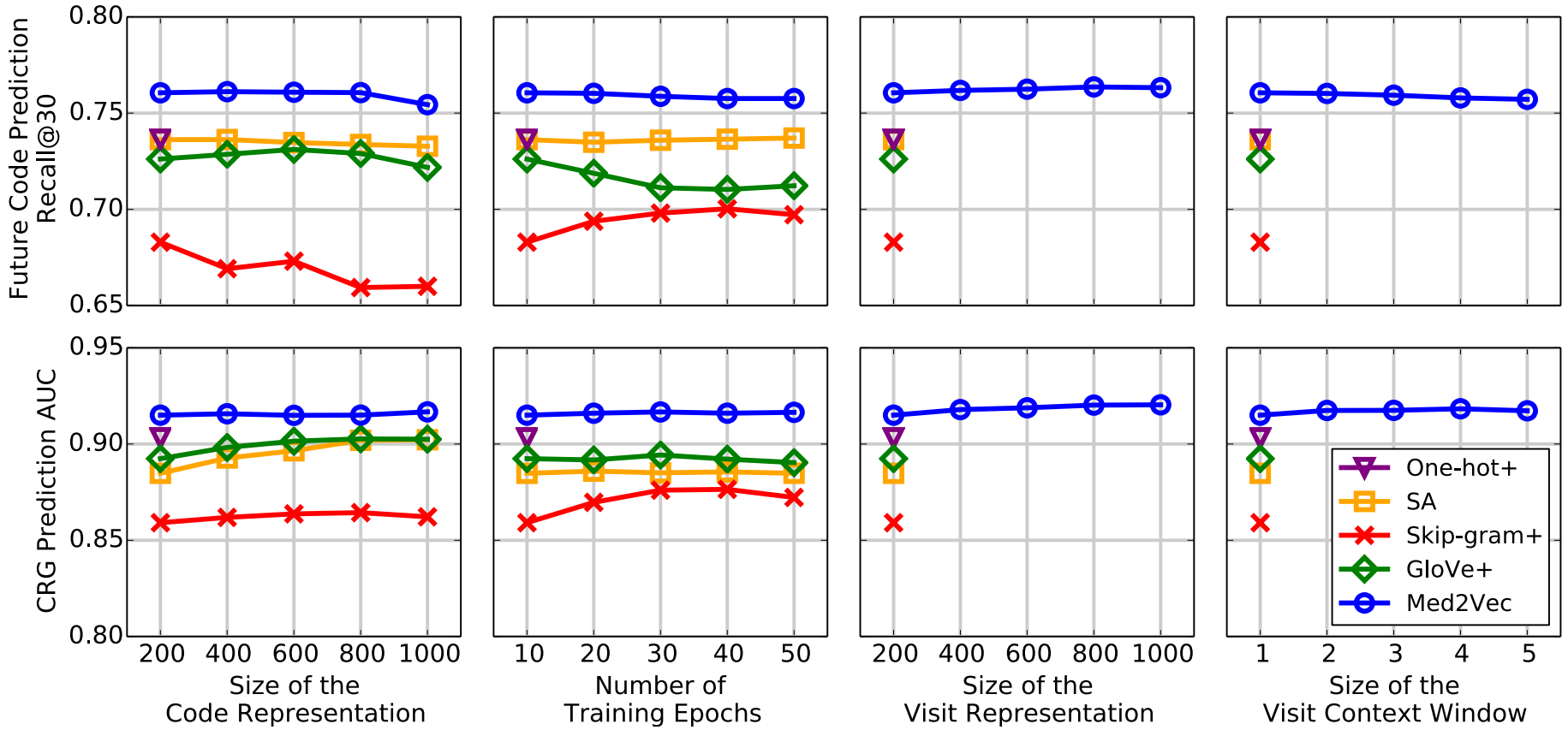
Med2Vec: Result

- Visualizing the learned representations



Med2Vec: Prediction

- Using the learned representations for prediction



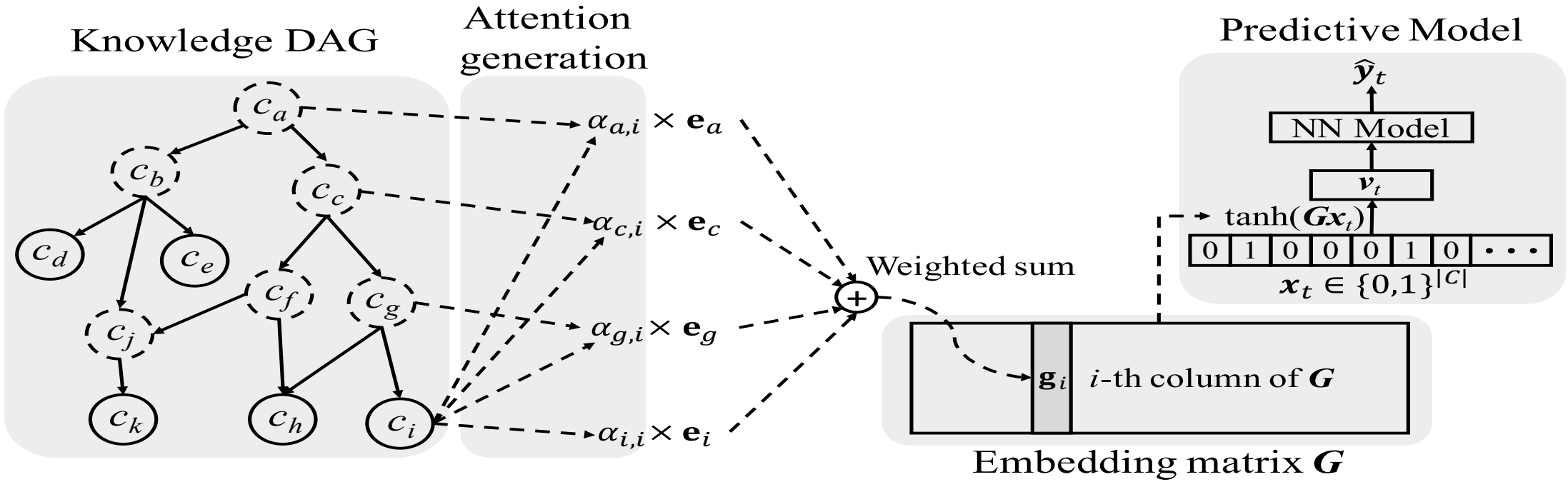
GRAM: Graph-based Attention Model for Healthcare Representation Learning

Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, Jimeng Sun

KDD' 17

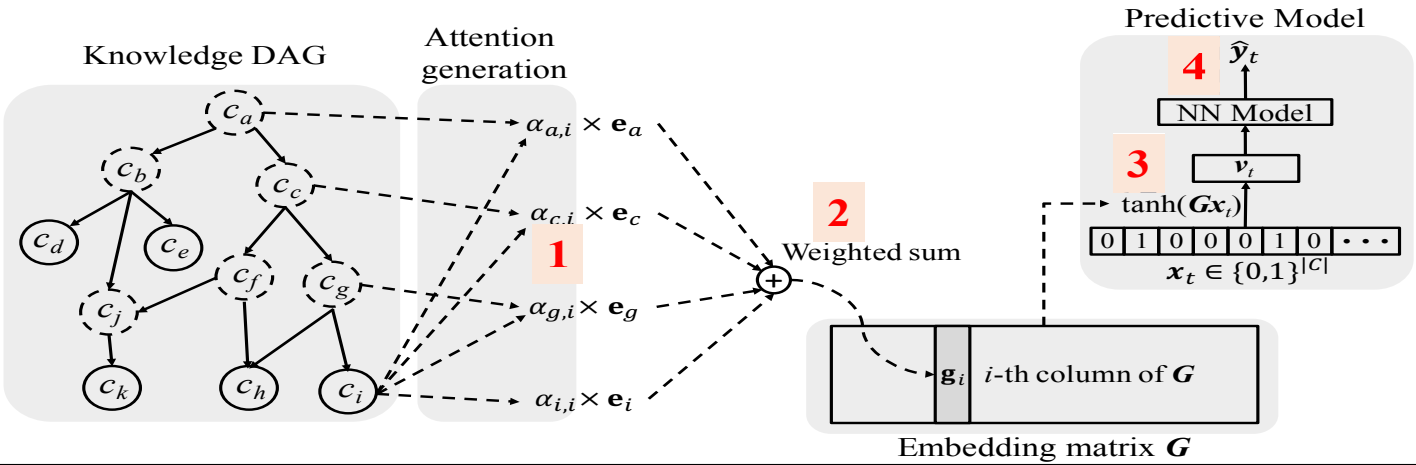
GRAM: Learn representations of medical codes leveraging medical ontologies

- Method: Generate a medical code representation vector by combining the representation vectors of its ancestors using the attention mechanism



Model structure of GRAM

GRAM: Algorithm

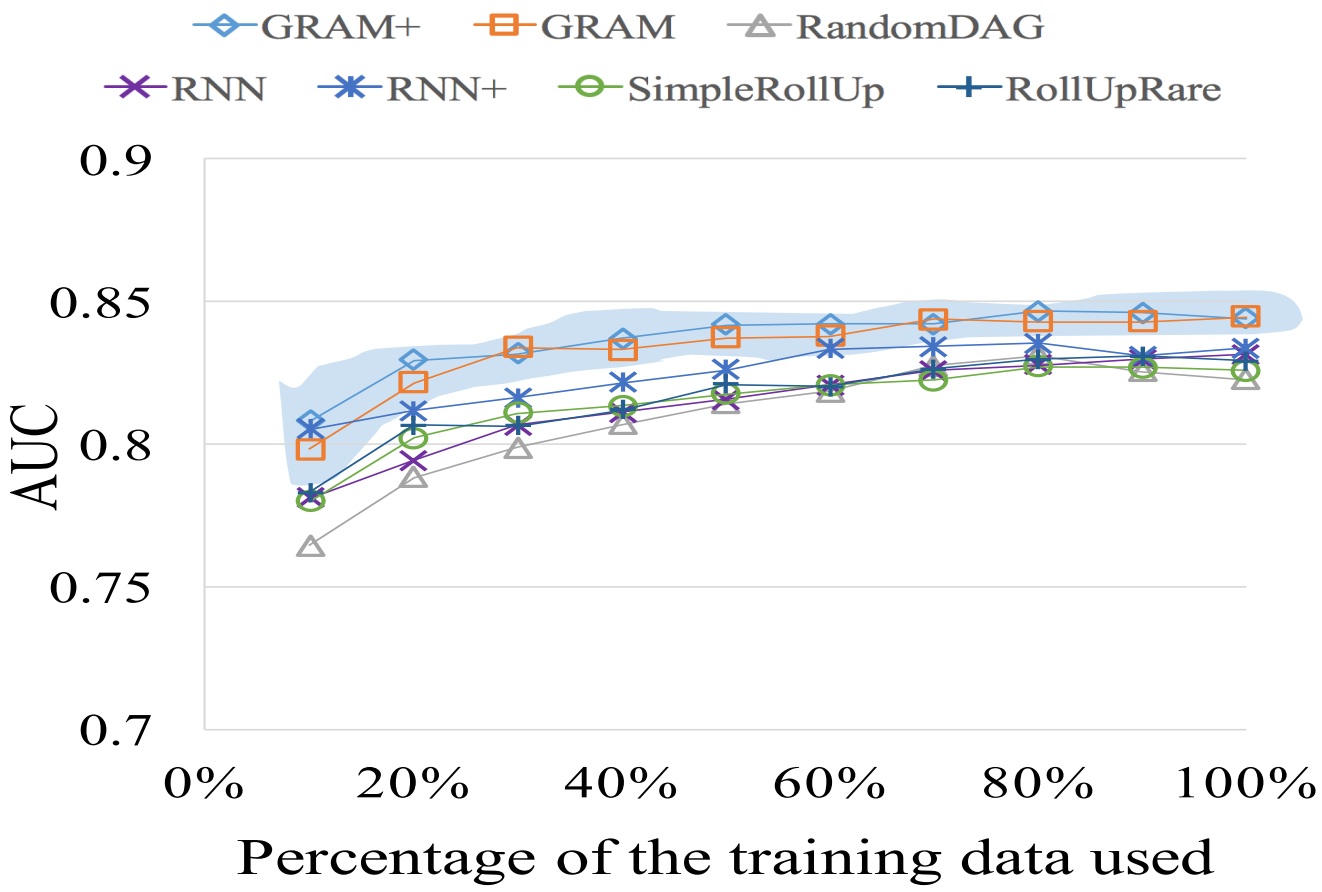


1	$\alpha_{ij} = \frac{\exp(f(\mathbf{e}_i, \mathbf{e}_j))}{\sum_{k \in \mathcal{A}(i)} \exp(f(\mathbf{e}_i, \mathbf{e}_k))}$ where $f(\mathbf{e}_i, \mathbf{e}_j) = \mathbf{u}_a^\top \tanh(\mathbf{W}_a \begin{bmatrix} \mathbf{e}_i \\ \mathbf{e}_j \end{bmatrix} + \mathbf{b}_a)$
	Attention weights are generated for all pairs of basic embeddings and its ancestors .
2	$\mathbf{g}_i = \sum_{j \in \mathcal{A}(i)} \alpha_{ij} \mathbf{e}_j,$
	Final representation is the weighted sum of attention weights and basic embeddings.
3	$\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t = \tanh(\mathbf{G}[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t])$
	Sequence of visit representations are obtained using the Embedding matrix $\boldsymbol{G}$.
4	$\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_t = \text{RNN}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t, \theta_r),$ $\widehat{\mathbf{y}}_t = \widehat{\mathbf{x}}_{t+1} = \text{Softmax}(\mathbf{W}\mathbf{h}_t + \mathbf{b}),$
	Performing sequential diagnoses prediction, outcomes are generated by RNN and Softmax.

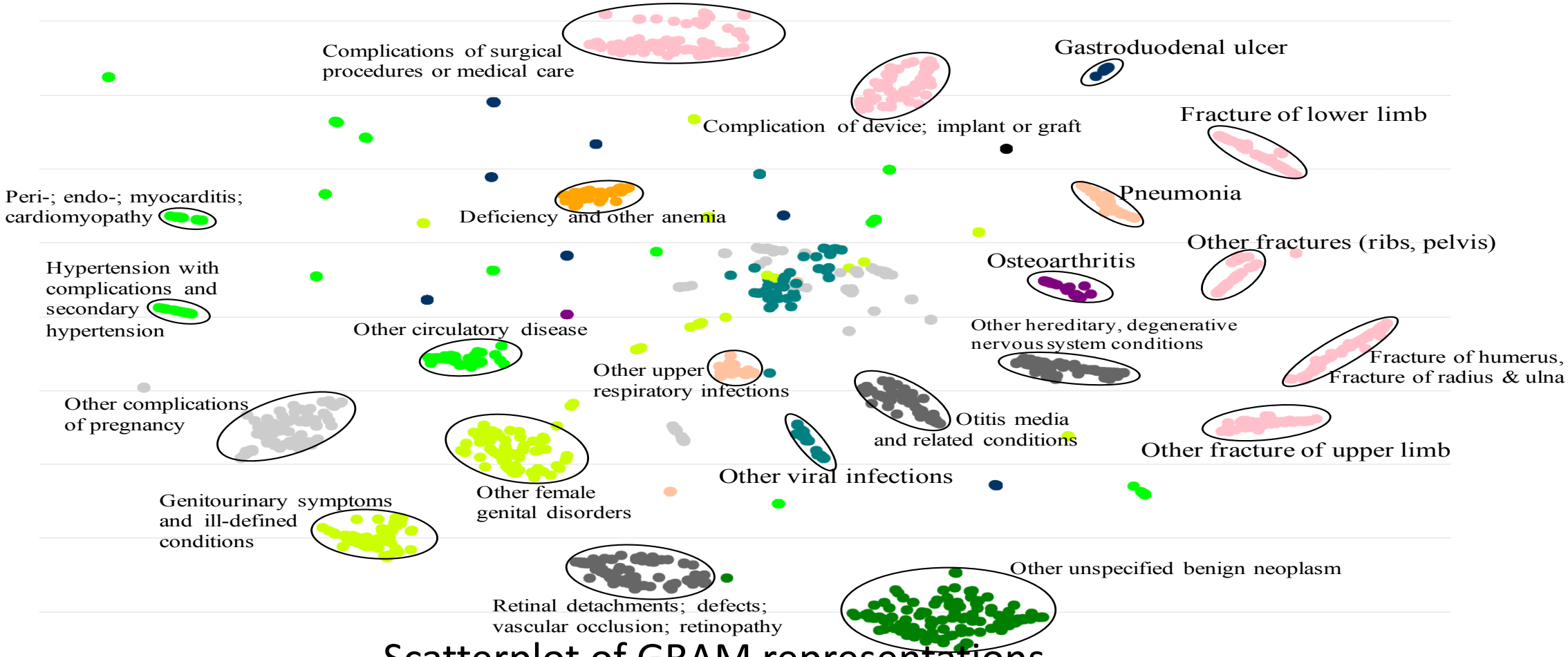
GRAM provide accurate prediction

GRAM shows better predictive performance under data constraints

HF prediction using varying sizes of training data

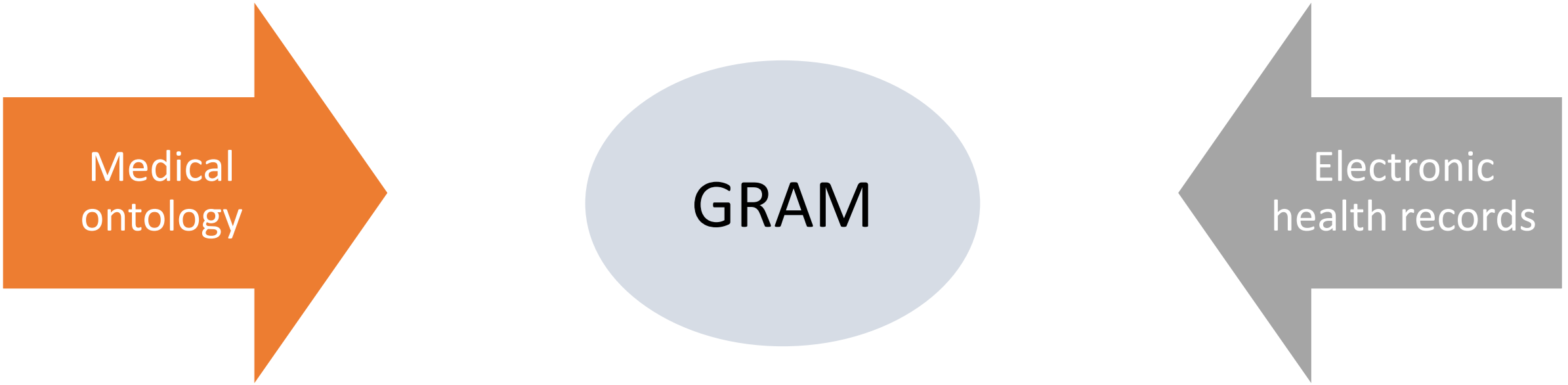


GRAM learns representations well aligned with knowledge ontology



Scatterplot of GRAM representations

GRAM: Summary



- Robust representation against data insufficiency
- Interpretable: Well aligned with medical knowledge

Choi, Edward, et al."GRAM: Graph-based Attention Model for Healthcare Representation Learning." Proceedings of KDD, 2017.

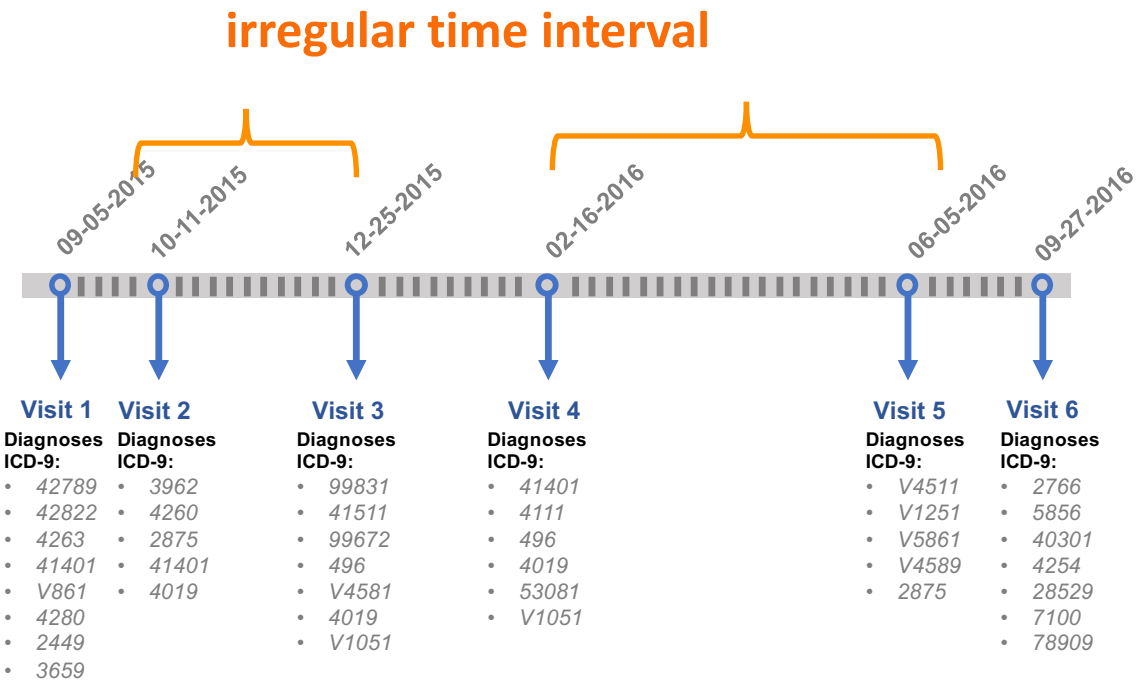
T-LSTM:Patient Subtyping via Time-Aware LSTM Networks

Inci M. Baytas, Cao Xiao, Xi Zhang, Fei Wang, Anil K. Jain, Jiayu Zhou

KDD' 17

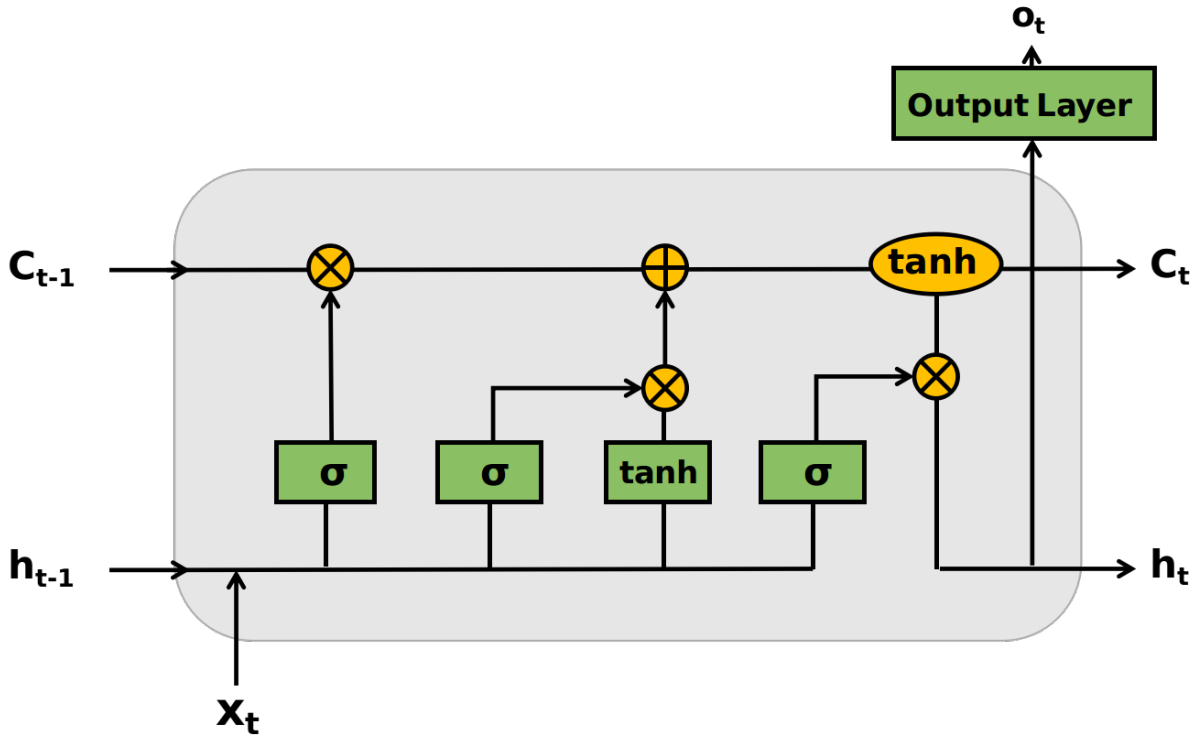
T-LSTM: Background

- **Patient subtyping** seeks patient groups with similar disease progression pathways based on longitudinal EHR.
- **Elapsed time** has a significance in clinical decision making.
- It is crucial to capture the relationships and the dependencies between clinical events under time interval irregularities.

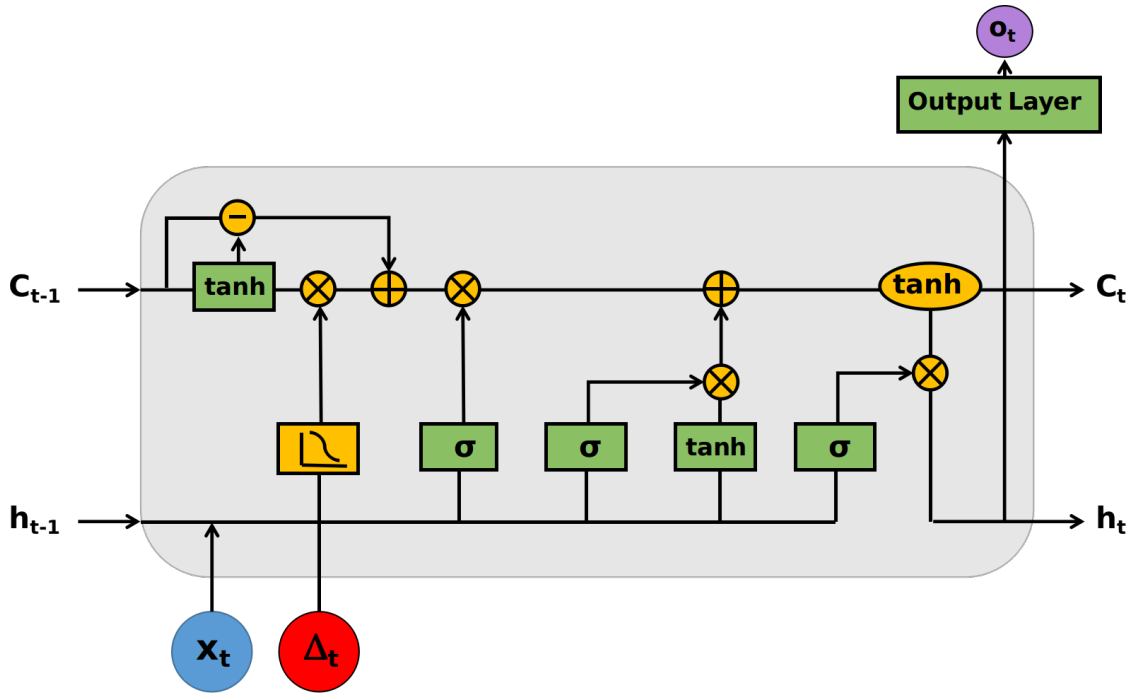


T-LSTM: Limitation of LSTM

- The Long-Short Term Memory (LSTM) is an appealing model for capturing long-range dependency in EHR data.
- However, LSTM assumes the elapsed time is uniform throughout the sequence.



The Time-Aware LSTM (T-LSTM) Unit



Input **Elapsed**
Records **Time**

Subspace Decomposition

$$C^S_{t-1} = \tanh\left(W_d C_{t-1} + b_d\right) \quad \text{(Short-term memory)}$$

$$\hat{C}^S_{t-1} = C^S_{t-1} * g\left(\Delta_t\right) \quad \text{(Discounted short-term memory)}$$

$$C^T_{t-1} = C_{t-1} - C^S_{t-1} \quad \text{(Long-term memory)}$$

$$C^*_{t-1} = C^T_{t-1} + \hat{C}^S_{t-1} \quad \text{(Adjusted previous memory)}$$

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad \text{(Forget gate)}$$

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1} + b_i\right) \quad \text{(Input gate)}$$

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1} + b_o\right) \quad \text{(Output gate)}$$

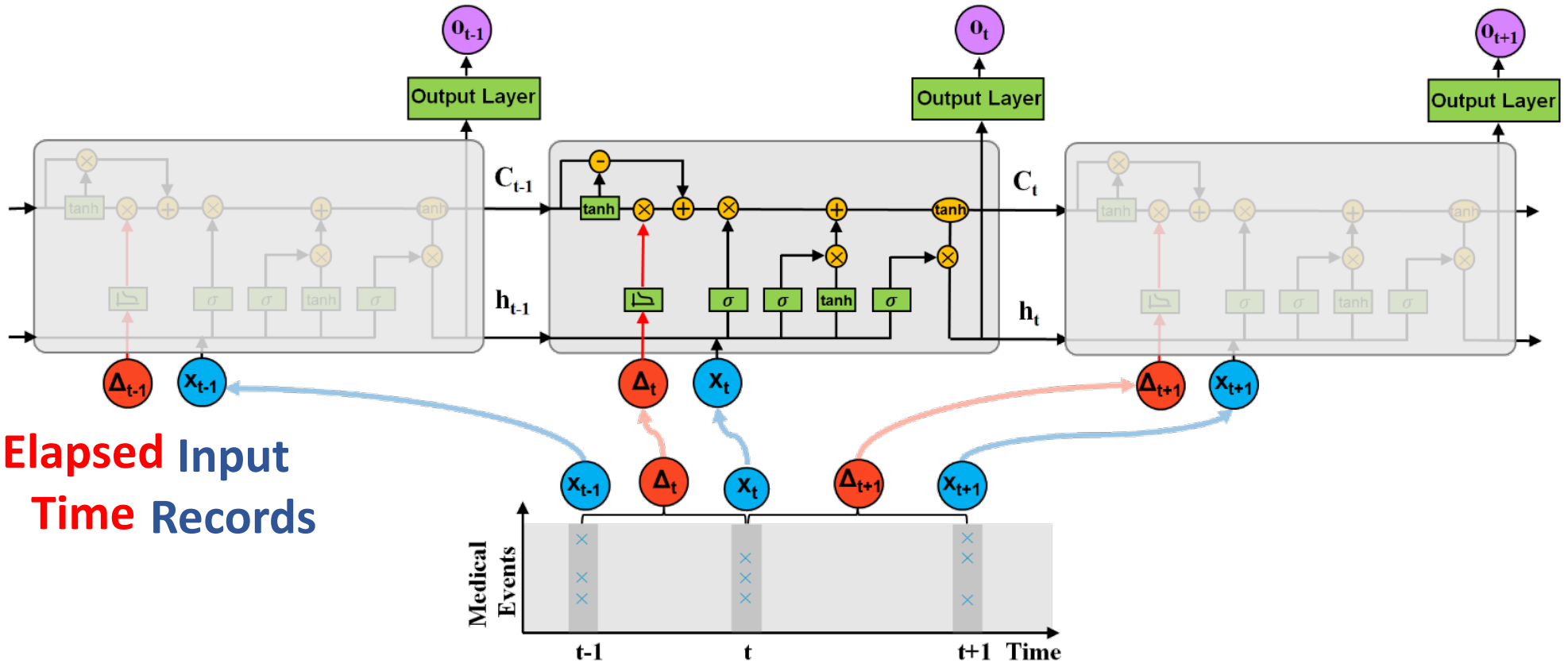
$$\tilde{C} = \tanh\left(W_c x_t + U_c h_{t-1} + b_c\right) \quad \text{(Canditate memory)}$$

$$C_t = f_t * C^*_{t-1} + i_t * \tilde{C} \quad \text{(Current memory)}$$

$$h_t = o * \tanh\left(C_t\right), \quad \text{(Current hidden state)}$$

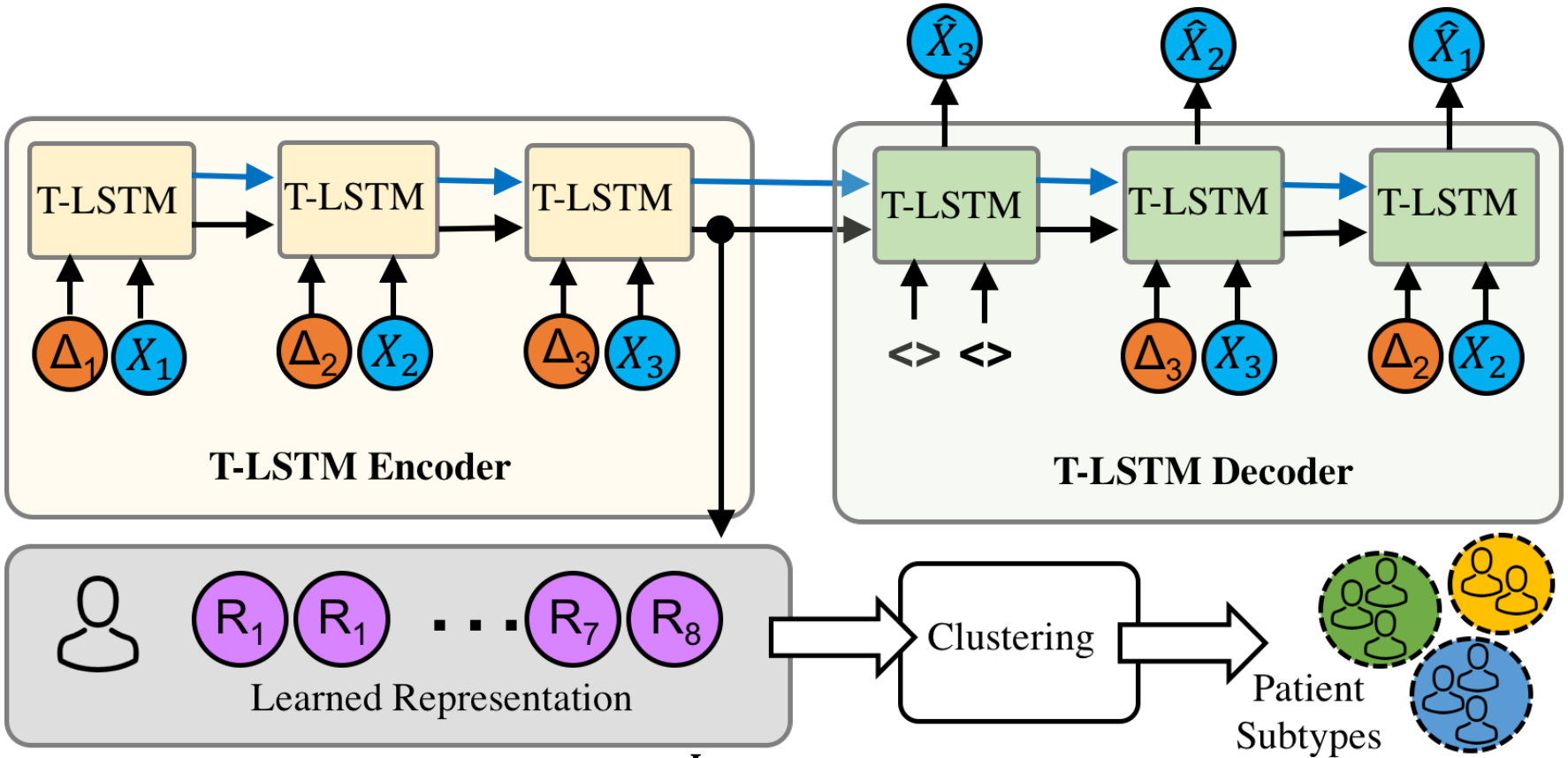
Standard LSTM

Time-Aware LSTM (T-LSTM)



T-LSTM decomposes the previous memory into long and short term components and utilizes the elapsed time (Δt) to discount the short term effects.

T-LSTM Autoencoder



$$E_r = \sum_{i=1}^{L} ||x_i - \hat{x}_i| \Big|_2^2$$

Time-Aware LSTM: Supervised Task

- Binary Classification of diabetes mellitus
- Clinical codes from 6730 patients
- Data are EMRbot
- http://www.emrbots.org/

Methods	Avg. Test AUC	Stdev.
T-LSTM	**0.91**	0.01
MF1-LSTM	0.87	0.02
MF2-LSTM	0.82	0.09
LSTM	0.85	0.02
LR	0.56	0.01

Time-Aware LSTM: Parkinson's Disease Patient Subtyping

- Features include:
  - Demographics
  - Motor severity measures such as Unified Parkinson's Disease Rating Scale, Hoehn and Yahr staging,
  - Non-motor manifestations: depression, anxiety, cognitive status, sleep disorders,
  - Imaging assessment such as DaTScan.

Parkinson's Progression Markers Initiative (PPMI) data, 654 patients
Elapsed time interval: [1,26] months
Average sequence length: 25
Input feature dimension: 319
Target dimension: 82

Time-Aware LSTM: Parkinson's Disease Patient Subtyping

Feature	P-Value	Cluster1 Mean	Cluster2 Mean
T-LSTM			
BJLO	9.51×10^{-8}	16.5	24.7
MoCA	0.001	40.0	41.2
DaTScan1	0.042	2.29	2.07
DaTScan2	0.027	2.31	2.08
DaTScan4	0.001	1.4	1.1
MF1-LSTM			
CSF-Total tau	0.007	87.9	46.72
MoCA	2.16×10^{-17}	47.5	41.05
SDM	0.005	58.5	41.5
MF2-LSTM			
HVLT-Retention	0.03	0.84	0.83
SDM	0.007	36.61	41.68

- Chi-square test for the categorical features, F-test for the normal continuous features, Kruskal-Wallis test for non-normal continuous features, and Fisher's exact test for sparse features.
- If the p-value is less than 0.05, a significant group effect is considered for the associated feature.
- Method producing more features with $p < 0.05$, is considered as providing a more sensible patient subtyping result.

Time-Aware LSTM: Parkinson's Disease Patient Subtyping

- Orange color represents the cluster mean which is higher than the total mean of the patients and the shades of blue show lower mean values for the corresponding feature with $p < 0.05$.
- PD patients are known to have lower DaTScan SBR values than healthy subjects.

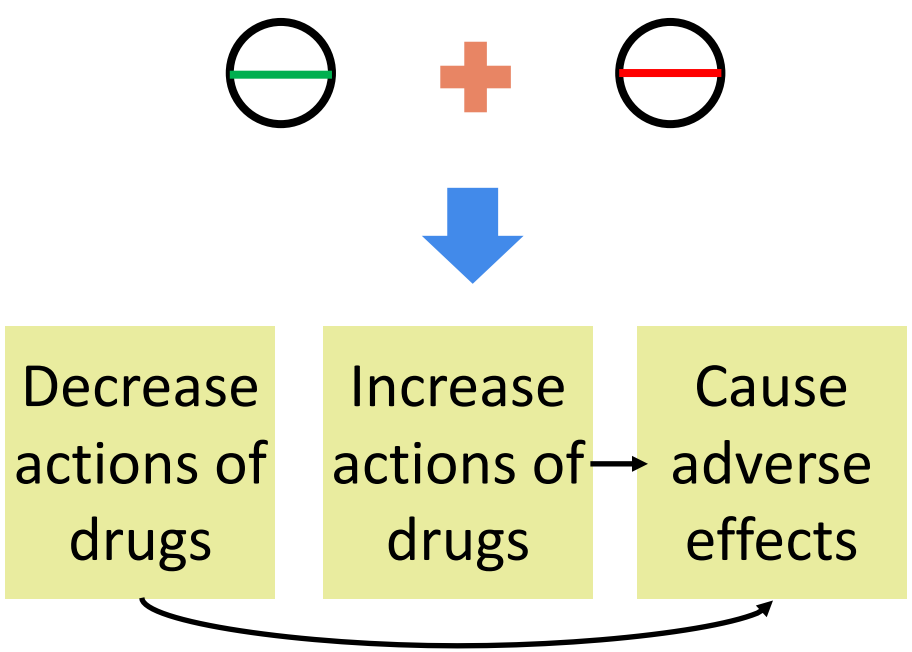
Drug Similarity Integration Through Attentive Multi-view Graph Auto-Encoders

Tengfei Ma, Cao (Danica) Xiao, Jiayu Zhou, Fei Wang

IJCAI' 18

Drug-Drug Interaction (DDI)

Combined Use of Drugs



Facts of DDIs

- Common among patients with complex diseases or comorbidities.
- Hard to observe in clinical testing.
- Affects 15% U.S. population. Cost more than $177 billion per year in disease management.

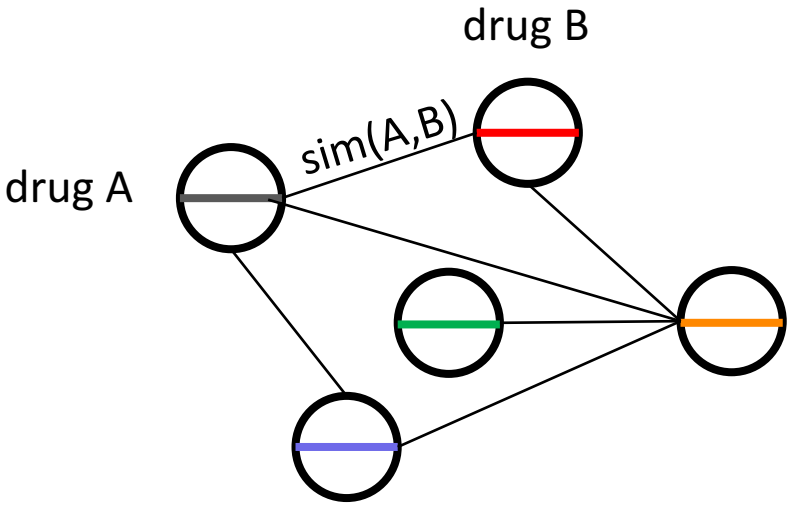
DDI Prediction based on Multiview Data

- **Drug Features** (database)
  - Label Side Effect (SIDER)
  - Off-Label Side Effect (OFFSIDES)
  - Molecular substructure
  - Drug Indication (MedDRA)
  - ……
- **Assumption**: similar drugs may interact with the same drug.
- **Approach**: We consider each type of feature as a view that has partial correlation with drug similarity, so we aim at integrating similarity metrics across multiple views for more accurate similarity learning.

State-of-the-art and Challenges

- State-of-the-art
  - Nearest neighbor methods (Zhang *et al*, 2015, Zhang *et al*, 2017)
  - Random walk based methods (Wang *et al*, 2010)
  - Unsupervised iterative methods (Angione *et al*, 2014, Xu *et al*, 2016)
  - Multiple kernel learning (Zhuang *et al*, 2011, McFee *et al*, 2011)
- Challenges
  - the underlying relations of biomedical events are often **nonlinear** and **complex** over all types of features
  - features have **different importance** toward different target outcomes

Model Overview



- Construct drug similarity graph, where DDIs are node labels
- Graph convolutional network (GCN) based model for node embedding and prediction
- Attention mechanism to integrate similarities from different views

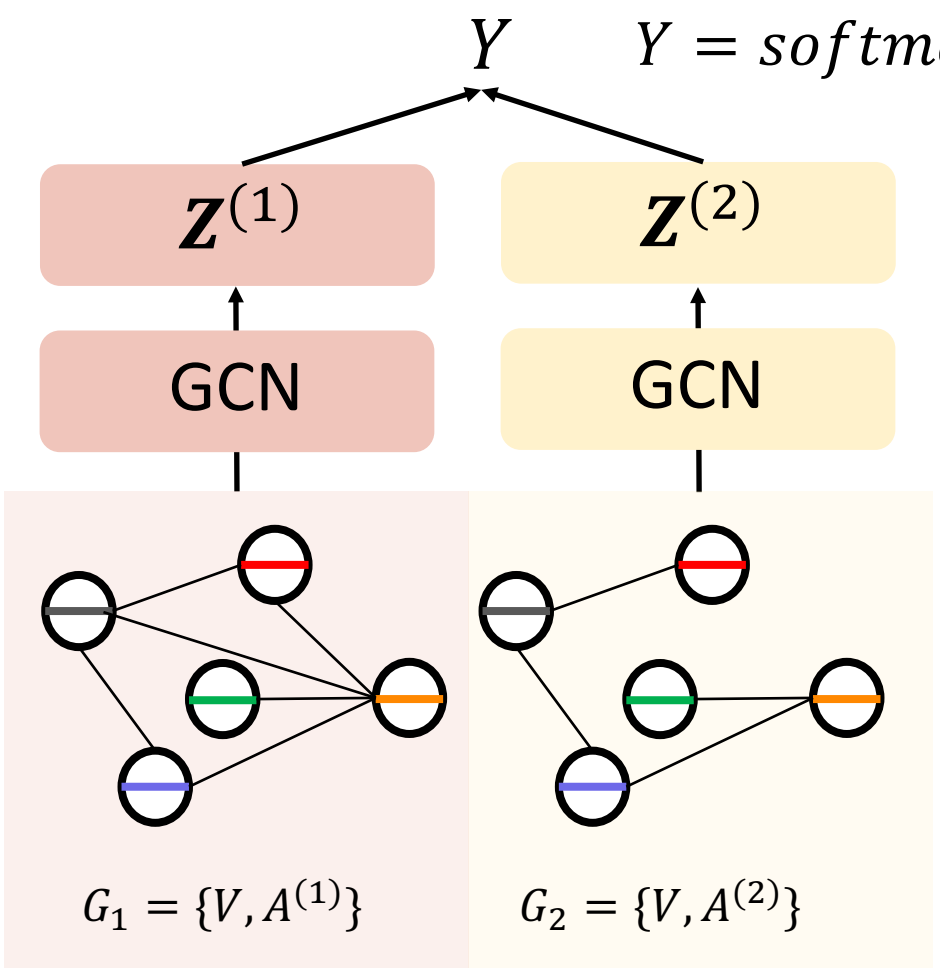
Recall: GCN (Kipf and Welling, 2016)

- Objective: graph node embedding for arbitrary graphs, distance as (dis-)similarity of local graph structures
- Input
  - Node features $\boldsymbol{x}$
  - Adjacency matrix $\boldsymbol{A}$ that represents graph structure
- Output: Node embedding $\boldsymbol{Z}$
- Method

$$\boldsymbol{H}^{(l+1)} = f(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\boldsymbol{H}^{(l)}\boldsymbol{W}^{(l)})$$

where $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}_N$, $\boldsymbol{D}$ is a diagonal matrix such that $D_{ii} = \sum_j \tilde{A}_{ij}$, $\boldsymbol{W}^{(l)}$ is layer-specific parameter matrix, $\boldsymbol{H}^{(l)}$ is node representation of lth layer.

Multi-View GCN



Embedding

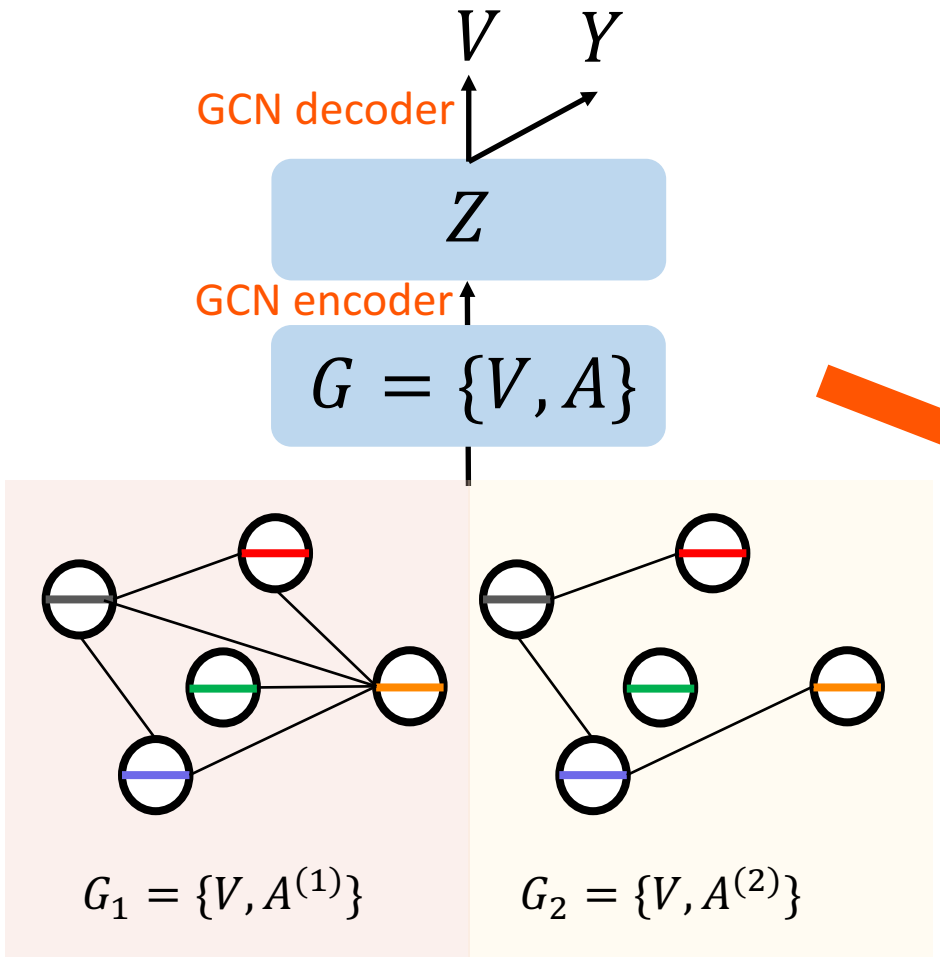
For view $u \in \{1,\ldots,T\}$, we encode the nodes as

$$\boldsymbol{Z}^{(u)} = f\left(\boldsymbol{X}^{(u)}, \boldsymbol{A}^{(u)}; \boldsymbol{W}_1^{(u)}\right)$$

$$= softmax(\widehat{\boldsymbol{A}}^{u} ReLU(\widehat{\boldsymbol{A}}^{u} \boldsymbol{X}^{(u)} \boldsymbol{W}_0^{(u)}) \boldsymbol{W}_1^{(u)})$$

where $\widehat{\boldsymbol{A}}^{u} = \widetilde{\boldsymbol{D}}^{(u)-\frac{1}{2}} \widetilde{\boldsymbol{A}}^{(u)} \widetilde{\boldsymbol{D}}^{(\boldsymbol{u})-\frac{1}{2}}$, and $\boldsymbol{W}_0^{(u)}$ and $\boldsymbol{W}_1^{(u)}$ are weight matrices.

Attentive Multiview Similarity Fusion with Graph Auto-Encoders (GAE)



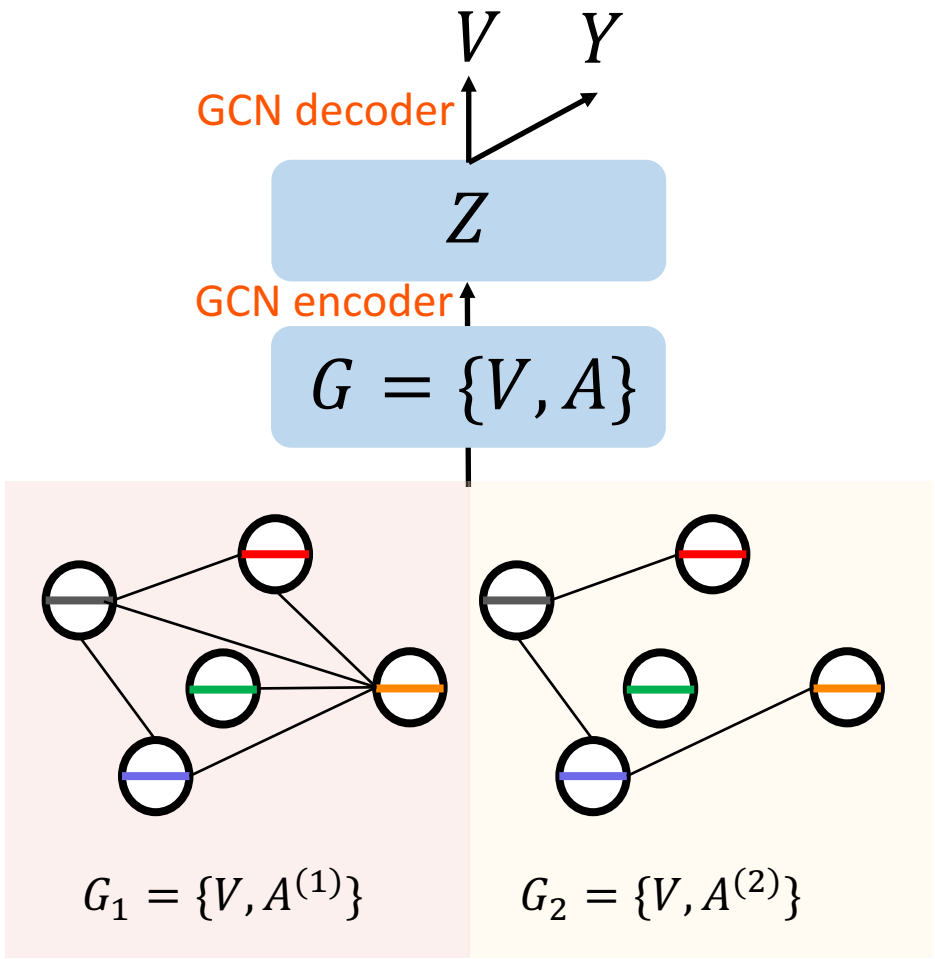
When we do not have labels, we reconstruct $\boldsymbol{V}$ from $\boldsymbol{Z}$ to minimize autoencoder loss, $\mathrm{L}_{ed} = \sum|\boldsymbol{X} - \boldsymbol{X}'|^2$.

$$A = \sum_u diag(g^u) * A^u$$

$$g'^u = W^u A^u + b^u \quad \text{Normalize} \rightarrow \quad g^u$$

attention weights decided by data and target

Semi-Supervised Extensions (SemiGAE)



SemiGAE (for partial labels)

Objective: $\min L = L_{train} + L_{ed}$

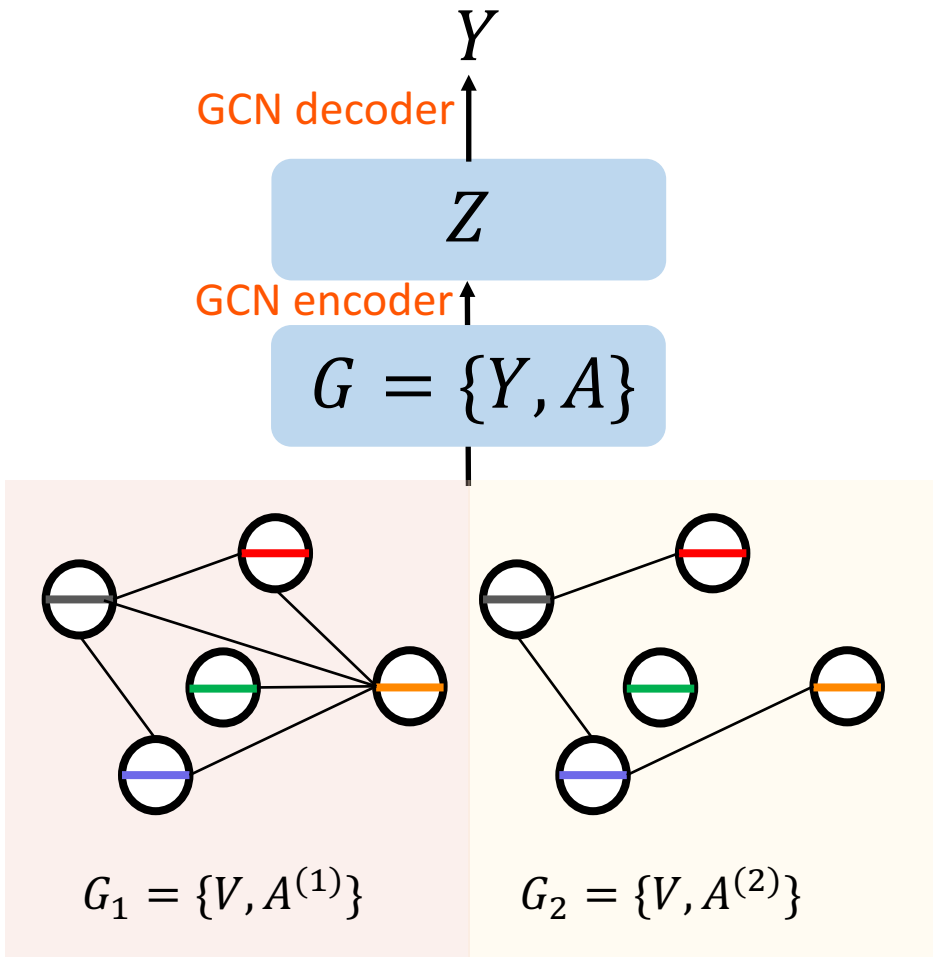
Training loss for labeled data

$$L_{train} = \sum_{y \in Y_{train}} \sum_{y' \in h(Z_{train})} -y \ln y'$$

Auto-encoder loss for all data

$$L_{ed} = \sum |X - X'|^2$$

Transductive Extensions (TransGAE)



TransGAE (for lack of node features)

For the case when we do not have node feature

$$\min L = |Y'_{train} - Y_{train}|^2 + |Y'_{test} - Y_{test}|^2 + \mu |Y_{test}|^2$$

where we also treat DDI label as input variable

$$\{Y'_{train}, Y'_{test}\} = f'(f(\{Y_{train}, Y_{test}\}, \hat{A}), \hat{A})$$

Experiments

Data (Binary)	Dimension
drugs (pairs)	645 (63473)
DDI	1318
label ADR	4192
off-label ADR	10093
Substructure	645 x 1024
Data (Multi)	**Dimension**
drugs (pairs)	222 (63473)
DDI	1301
indication	1702
CPI	611
TTD	207
Substructure	645 x 582

Baselines

- Nearest neighbor [Vilar et al. 2012]
- Label Propagation [Zhang et al. 2015]
- Multiple Kernel Learning [Strazar and Curk 2016]
- Basic Multi-view GraphCNN

Evaluation

- Same strategy as [Zhang et al. 2015]
- Selecting a fixed percentage of drugs randomly and all DDIs associated with these drugs are used for testing
- For the remaining training data, 90%/10% split for training/validation
- Evaluation metrics: ROC-AUC, PR-AUC

Results

Table 2: Predicting Specific DDI Types (Multiple Outcomes) on Dataset 2.

	Methods	Test Split (25%)		Test Split (50%)	
		ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
Using Single View					
Baselines	NN	0.627 ± 0.043	0.594 ± 0.078	0.594 ± 0.033	0.554 ± 0.061
	LP	0.773 ± 0.025	$\mathbf{0.670 \pm 0.052}$	0.747 ± 0.028	$\mathbf{0.650 \pm 0.053}$
	GraphCNN	0.738 ± 0.047	0.594 ± 0.080	0.698 ± 0.090	0.583 ± 0.102
Proposed	SemiGAE	$\mathbf{0.798 \pm 0.029}$	0.661 ± 0.059	$\mathbf{0.784 \pm 0.028}$	0.649 ± 0.059
	TransGAE	0.790 ± 0.028	0.661 ± 0.068	0.770 ± 0.031	0.633 ± 0.080
Using Multiple Views					
Baselines	LP	0.774 ± 0.025	0.672 ± 0.052	0.748 ± 0.028	0.653 ± 0.055
	GraphCNN	0.601 ± 0.067	0.526 ± 0.120	0.578 ± 0.067	0.526 ± 0.108
	MKL	0.766 ± 0.030	0.650 ± 0.061	0.724 ± 0.026	0.586 ± 0.066
Proposed	AttSemiGAE	$\mathbf{0.802 \pm 0.029}$	$\mathbf{0.678 \pm 0.060}$	$\mathbf{0.786 \pm 0.030}$	$\mathbf{0.662 \pm 0.064}$
	AttTransGAE	0.782 ± 0.026	0.670 ± 0.058	0.764 ± 0.025	0.652 ± 0.061

Analysis of Interpretability

		Attention Weights			
DDI Type	AUC	Chem.	indi.	TTDS	CPI
Chest Pain	0.772	0.151	0.303	0.144	0.402
Insomnia	0.755	0.380	0.261	0.078	0.291
indication	0.774	0.117	0.301	0.283	0.299

Results

Views "indication" and "CPI" receive high weights for the ADR "Chest pain".

Clinical Evidence

Many DDI cases of chest pain are due to particular drug overdose [Nachimuthu et al., 2012]. For example, the co-use of many medications that treat depression can cause overdose and prolong the QT interval via CPI, and eventually cause chest pain.

Data Augmentation

- MedGAN (generate EHR statistics)
- Real-valued (medical) Time Series Generation with Recurrent Conditional GANs (NIPS ML4H 17)
- Recent Trend in Molecular Graph Generation for Drug Discovery

Generating Multi-label Discrete Electronic Health Records using Generative Adversarial Networks

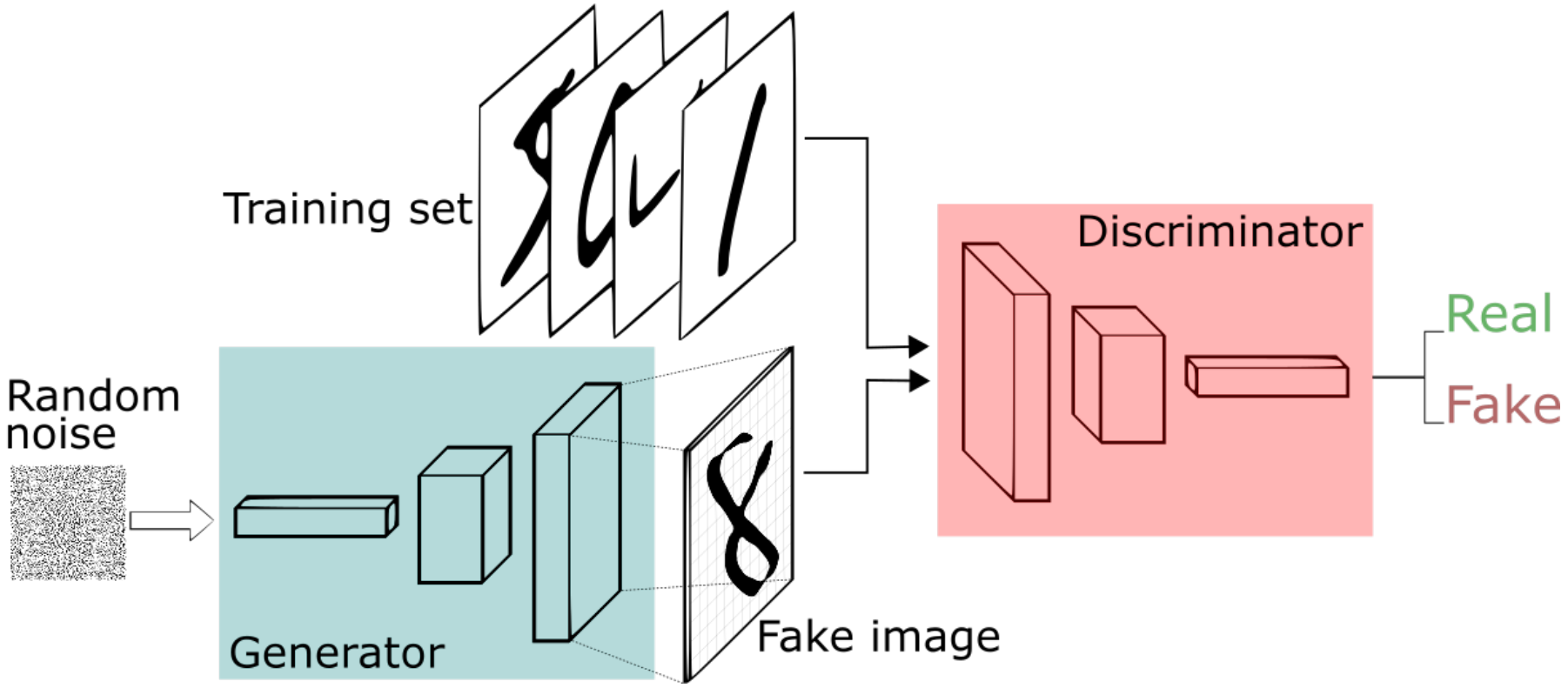
Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, Jimeng Sun

MLHC 2017

medGAN: Background

- Generating Multi-label Discrete Electronic Health Records using Generative Adversarial Networks
  - Choi, Edward, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun, MLHC 2017
- Generate synthetic patient records that are
  - Similar (statistically) to the real records
  - Does now divulge individual patient information

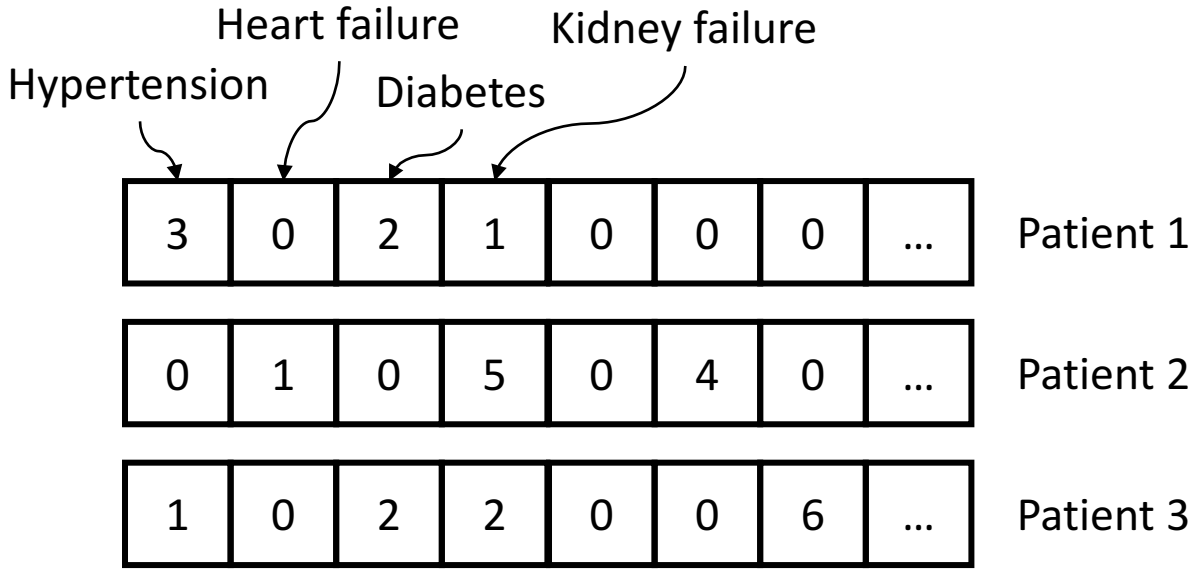
medGAN: Generative Adversarial Networks



https://sthalles.github.io/intro-to-gans/

medGAN: Patient Record

- Real patient dataset
  - Patients' records are aggregated over time
  - Discrete values (count, binary)

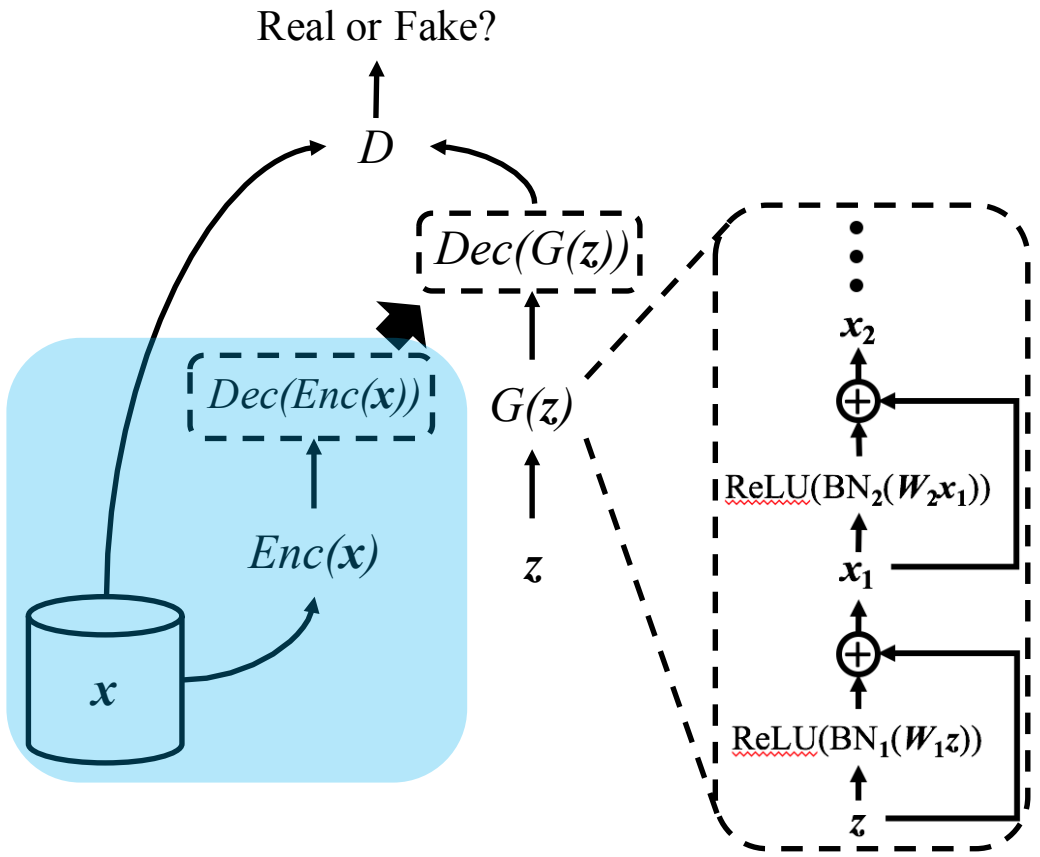


medGAN Architecture

Pre-train Autoencoder (ex: binary variable)

$$\frac{1}{m}\sum_{i=0}^{m} \mathbf{x}_i \log \mathbf{x}'_i + (1-\mathbf{x}_i)\log(1-\mathbf{x}'_i)$$

where $\mathbf{x}'_i = Dec(Enc(\mathbf{x}_i))$

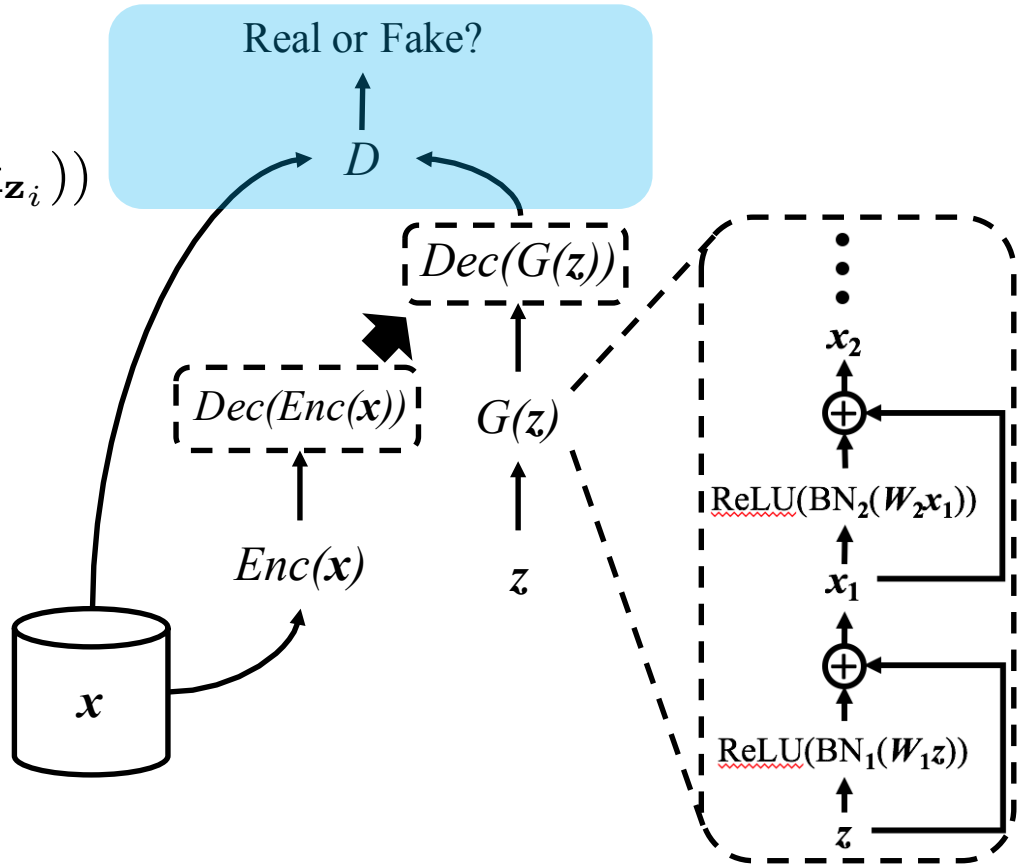


medGAN Architecture

Update Discriminator parameters

$$\theta_d \leftarrow \theta_d + \alpha \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \log D(\mathbf{x}_i) + \log(1 - D(\mathbf{x}_{\mathbf{z}_i}))$$

where $\mathbf{x}_{\mathbf{z}_i} = Dec(G(\mathbf{z}_i))$

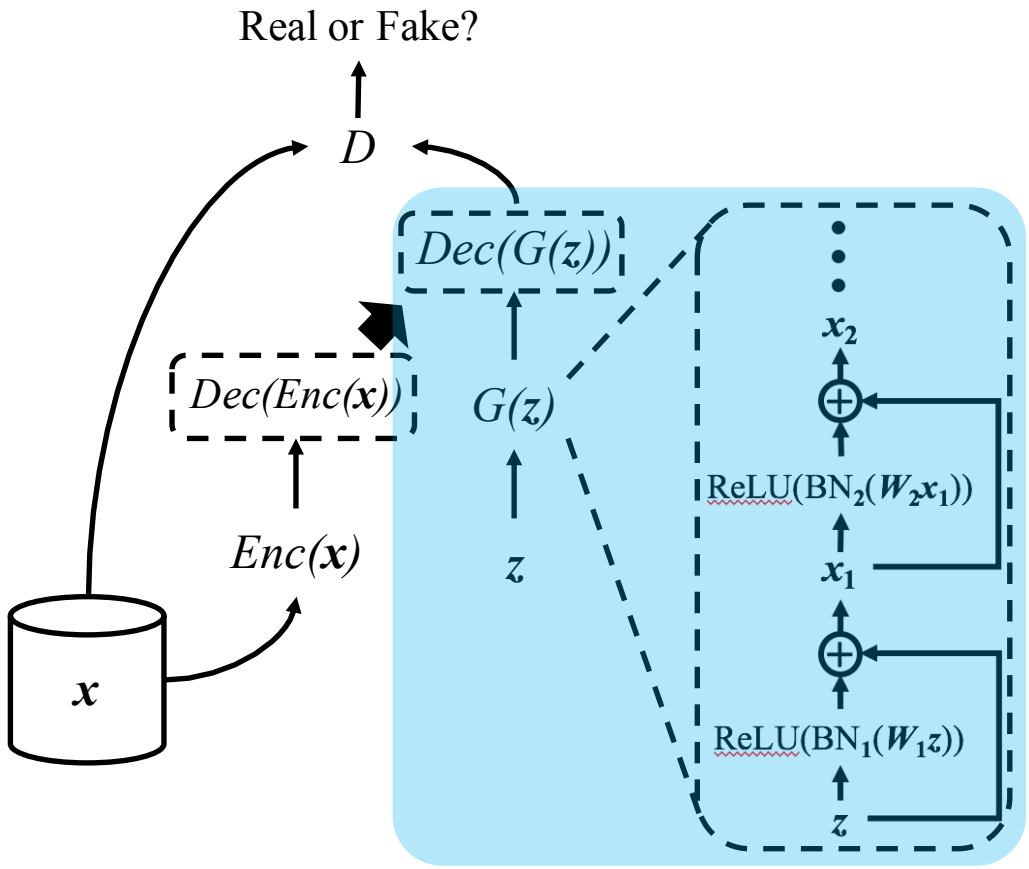


medGAN Architecture

Update Generator parameters

$$\theta_{g,dec} \leftarrow \theta_{g,dec} + \alpha \nabla_{\theta_{g,dec}} \frac{1}{m} \sum_{i=1}^{m} \log D(\mathbf{x}_{\mathbf{z}_i})$$

where $\mathbf{x}_{\mathbf{z}_i} = Dec(G(\mathbf{z}_i))$



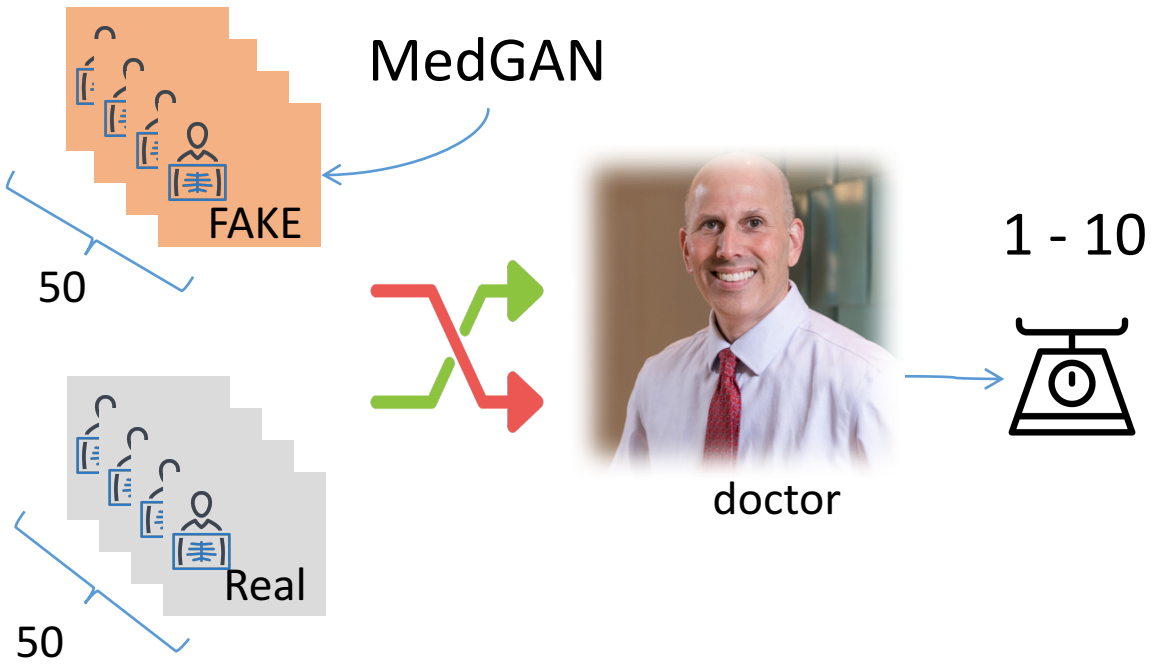
medGAN: Data

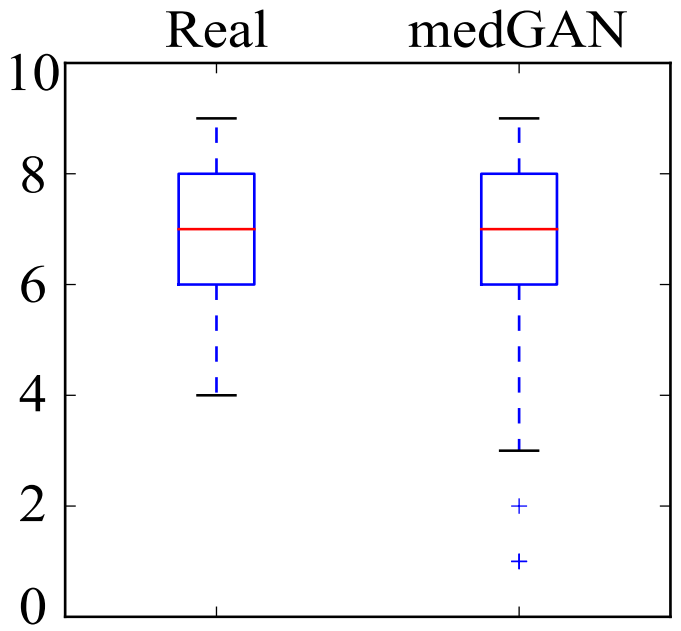
- Data
  - 260K patients from Sutter Health
  - Patient records over 10 years
  - 615 variables
    - Diagnosis/medication/procedure codes

- Binarized the count values
  - Either the code occurred, or did not occur

medGAN: Result

- Qualitative evaluation by a medical expert





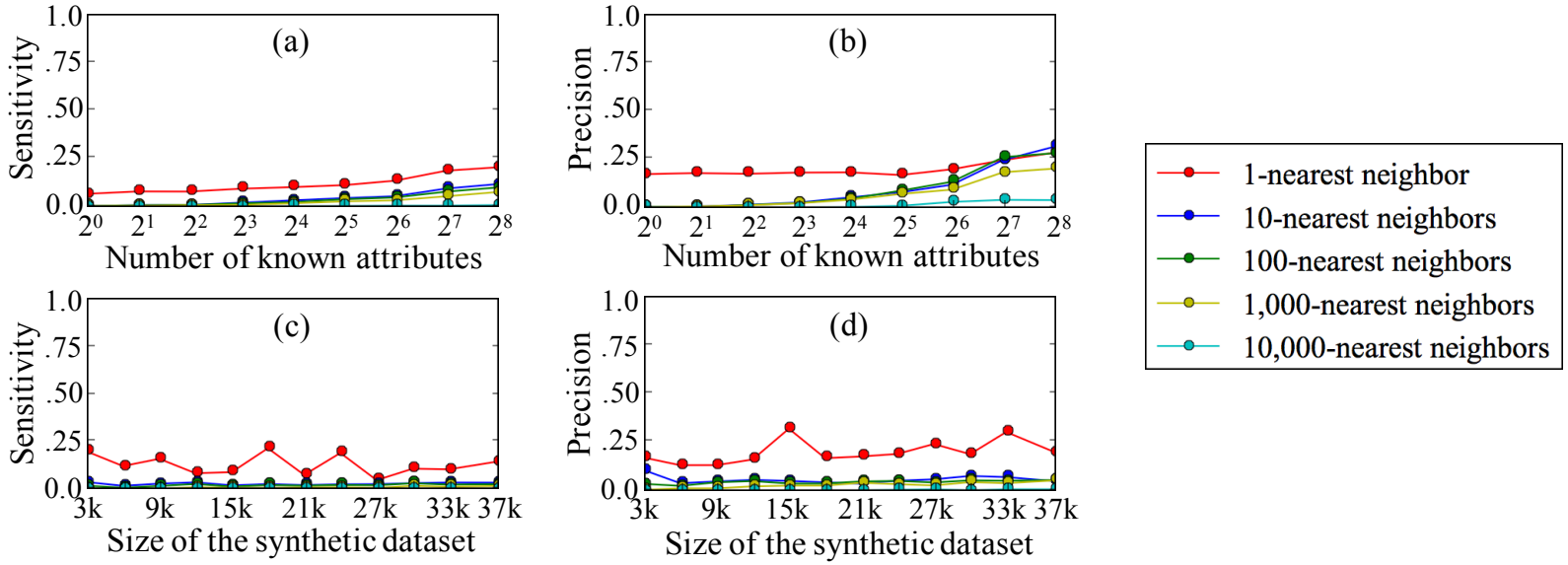
Indistinguishable to a human doctor except a few outliers!

medGAN: Privacy

- Attribute disclosure attack
  - When an attacker knows subset of patients' info, can they know more by looking at synthetic records?

medGAN: Privacy

- Attribute disclosure attack



- The attacker who knows 1% of the target patient's attributes (8-16 attributes)
  - Can correctly estimate at most 10% of positive unknown attributes
- Size of the synthetic data has little influence on the effectiveness of the attack

Attacker cannot gain useful knowledge on unknown attributes!

Real-valued (medical) Time Series Generation with Recurrent Conditional GANs

Esteban, Cristóbal, Stephanie L. Hyland, and Gunnar Rätsch

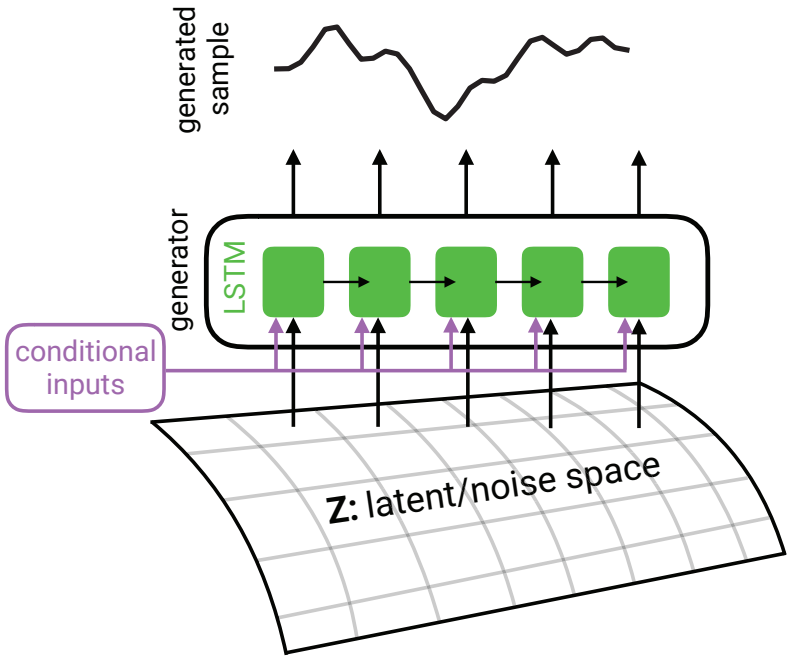
NIPS ML4H 17

RCGAN

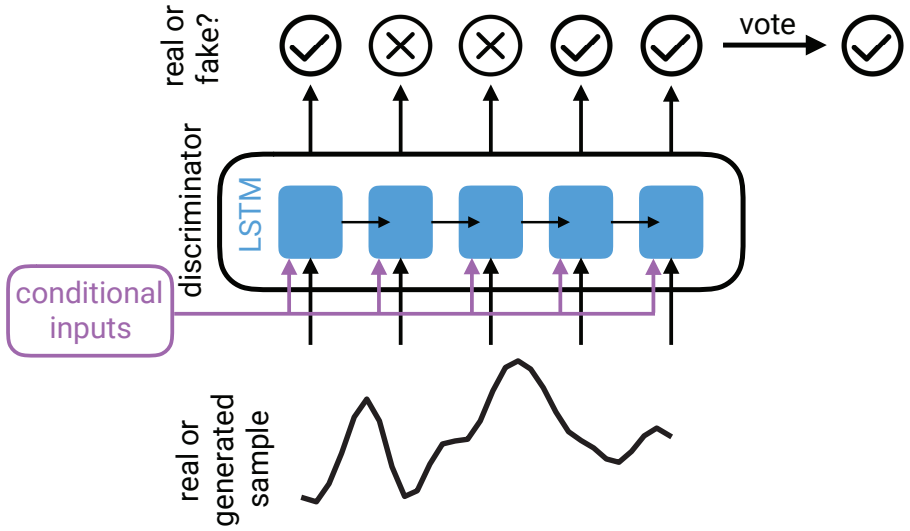
- Real-valued (medical) time series generation with recurrent conditional GANs.
  - Esteban, Cristóbal, Stephanie L. Hyland, and Gunnar Rätsch, NIPS ML4H Workshop 2017
- Generate a continuous-valued multi-variate time-series
  - Values such as measurements from ICU

RCGAN: Model

- Generating time-series using GAN



Generator



Discriminator

$$\mathrm{G}_{\mathrm{loss}}(Z_n) = -\mathrm{CE}(\mathrm{RNN}_{\mathrm{D}}(\mathrm{RNN}_{\mathrm{G}}(Z_n)), \mathbf{1})$$

$$\mathrm{D}_{\mathrm{loss}}(X_n, \mathbf{y}_n) = -\mathrm{CE}(\mathrm{RNN}_{\mathrm{D}}(X_n), \mathbf{y}_n)$$

RCGAN: Data

- Dataset
  - 17,693 patients from Philips eICU dataset
- Features
  - Four most frequently recorded variables by bed-side monitors
  - Oxigen saturation (SpO2), heart rate (HR), respiratory rate (RR), mean arterial pressure (MAP)
  - 16 measurement per each variable (16-step time-series)
    - Measurements for a 4-hour period
  - Exclude patients with missing data

RCGAN: Result

- Evaluation of the synthetic data
  - Train Random Forest on synthetic training data
  - Test the trained RF on real test data
- Prediction tasks
  - Based on 4-hours observation (16-timesteps), predict future binary label
  - Will {SpO2, HR, RR, MAP} be high/low in the next hour?

		SpO2 < 95	HR < 70	*HR > 100*
AUROC	real	0.9587 ± 0.0004	0.9908 ± 0.0005	0.9919 ± 0.0002
	TSTR	0.88 ± 0.01	0.96 ± 0.01	0.95 ± 0.01
AUPRC	real	0.9059 ± 0.0005	0.9855 ± 0.0002	0.9778 ± 0.0002
	TSTR	0.66 ± 0.02	0.90 ± 0.02	0.84 ± 0.03
	random	0.16	0.26	0.18

		RR < 13	RR > 20	MAP < 70	MAP > 110
AUROC	real	0.9735 ± 0.0001	0.963 ± 0.001	0.9717 ± 0.0001	0.960 ± 0.001
	TSTR	0.86 ± 0.01	0.84 ± 0.02	0.875 ± 0.007	0.87 ± 0.04
AUPRC	real	0.9557 ± 0.0002	0.891 ± 0.001	0.9653 ± 0.0001	0.8629 ± 0.0007
	TSTR	0.73 ± 0.02	0.50 ± 0.06	0.82 ± 0.02	0.42 ± 0.07
	random	0.26	0.1	0.39	0.05

RCGAN: Conclusion

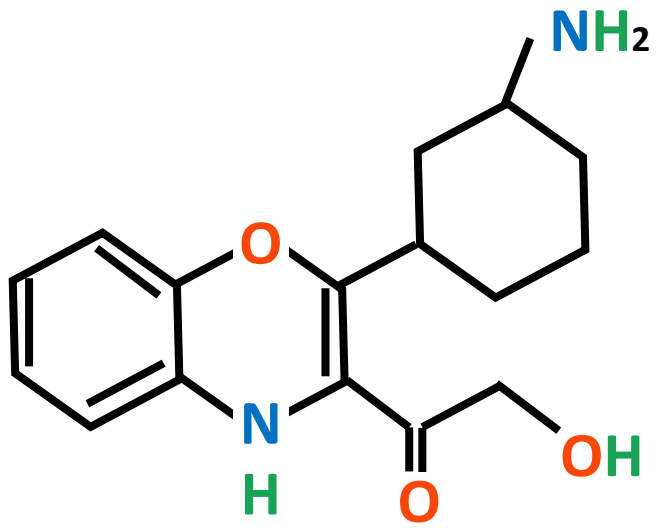
- Is RCGAN just memorizing?
  - Are there privacy risks?
- Use Maximum Mean Discrepancy (MMD) for evaluation
  - MMD: measure of the distance between two distributions

- Null hypothesis: RCGAN has **not** memorized the training data
  - MMD(synthetic data, test data) < MMD (synthetic data, train data)
- P-value: 0.40+-0.05
  - Cannot reject the null hypothesis

- Synthetic samples do not look more similar to the training set than they do to the test set

Recent Trend in Variational Autoencoders based Molecular Graph Generation for Drug Discovery

Molecular Graph Generation: Molecule Encoding



The simplified molecular-input line-entry system (SMILES) is a format of encoding molecules in text strings.

OCC(=O)C=1Nc3ccccc3OC=1C2CCC(N)C2

Consider a molecule as a sequential data and apply NLP techniques.

Molecular Graph Generation: Benchmark Data

	# Graphs	# Nodes	# Node Types	# Edge Types
QM9 (molecule)	134k	9	4	3
ZINC (molecule)	250k	38	9	3

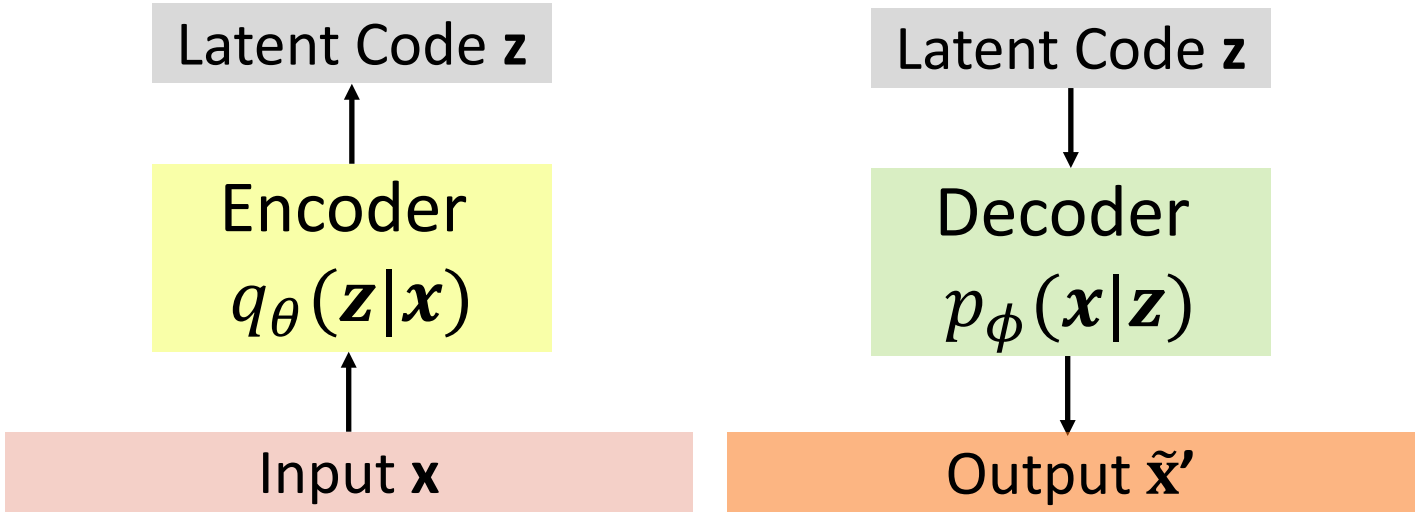
QM9: L. Ruddigkeit, R. van Deursen, L. C. Blum, J.-L. Reymond, Enumeration of 166 billion organic small molecules in the chemical universe database GDB-17, J. Chem. Inf. Model. 52, 2864–2875, 2012.

ZINC: Irwin, Sterling, Mysinger, Bolstad and Coleman, J. Chem. Inf. Model. 2012 DOI: 10.1021/ci3001277.

Molecular Graph Generation: Challenges & Opportunities

- **Goal**: Train generative models to construct more complex, discrete data types.
- **Challenge**: 1. Formal Languages is very strict 2. Small changes in output leads to syntax error
- **Opportunities**: 1. Syntax is context free 2. Grammar is known and fixed 3. Parses are unique

Tools: Variational AutoEncoder (Kingma et al., 13; Rezende et al., 14)

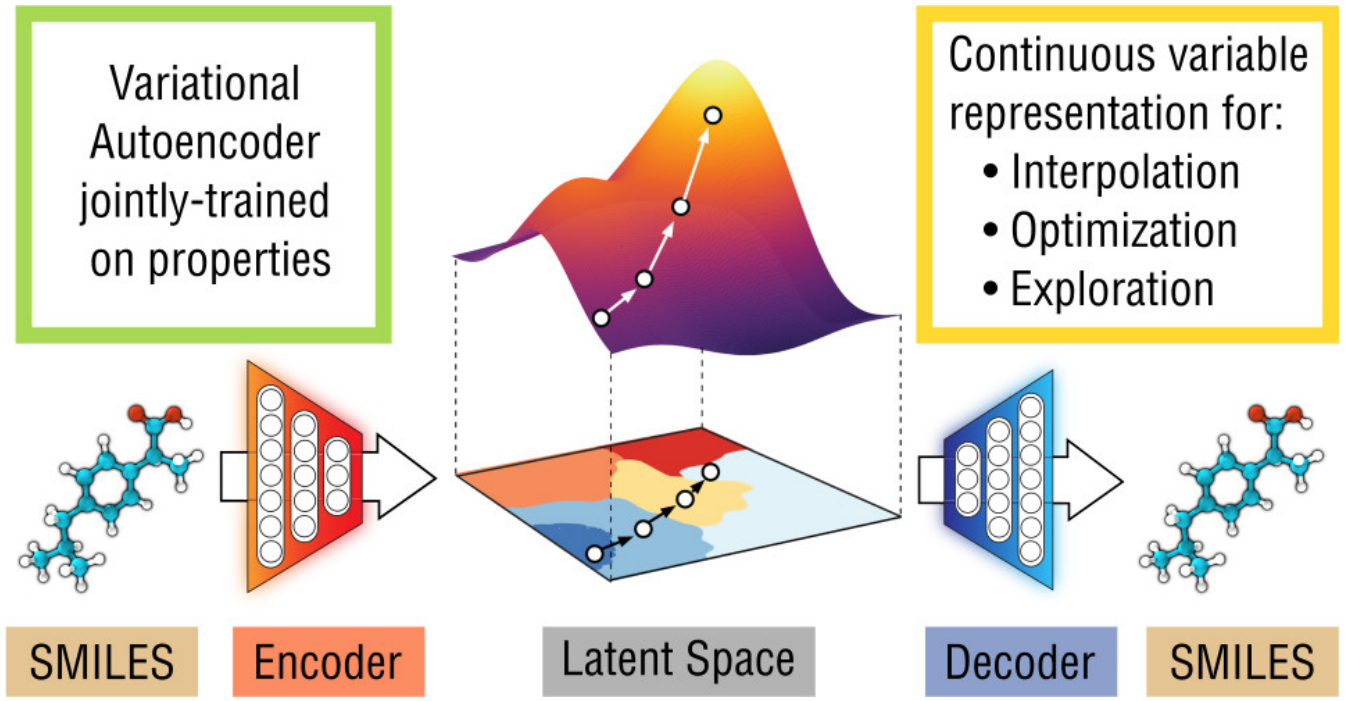


$$L(\theta,\phi) = -E_{q_\theta(\boldsymbol{z}|\boldsymbol{x})}\left[\log(p_\phi(\boldsymbol{x}|\boldsymbol{z}))\right] + KL(q_\theta(\boldsymbol{z}|\boldsymbol{x}), p_\phi(\boldsymbol{z}))$$

reconstruction loss

penalty for information loss

Molecular Graph Generation: CVAE (Rafael Gómez-Bombarelli et al., 2016)

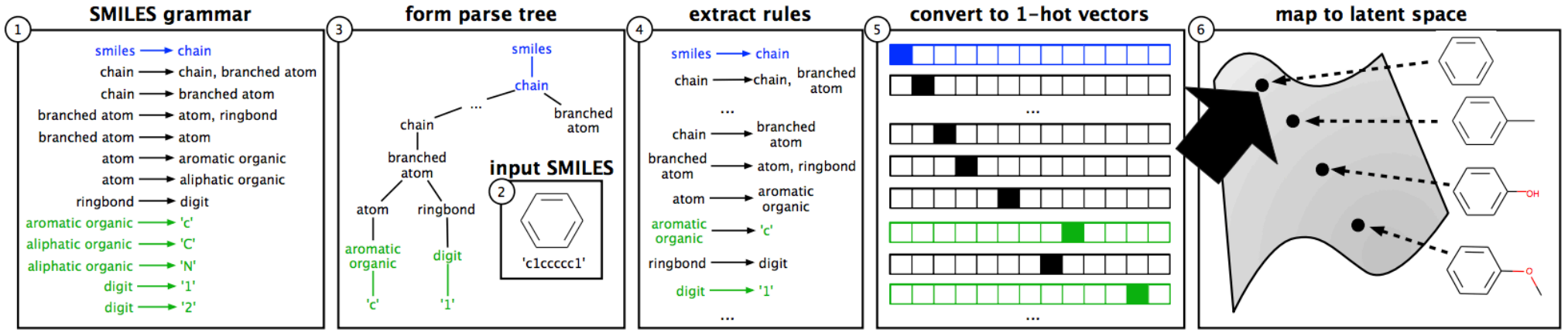


- Encode and decode molecules represented as SMILE with VAE.
- Latent representations could be used to learn in semi-supervised setting.
- The learned model can be used to find molecules with desired property by optimizing representation in latent space and decoding it.

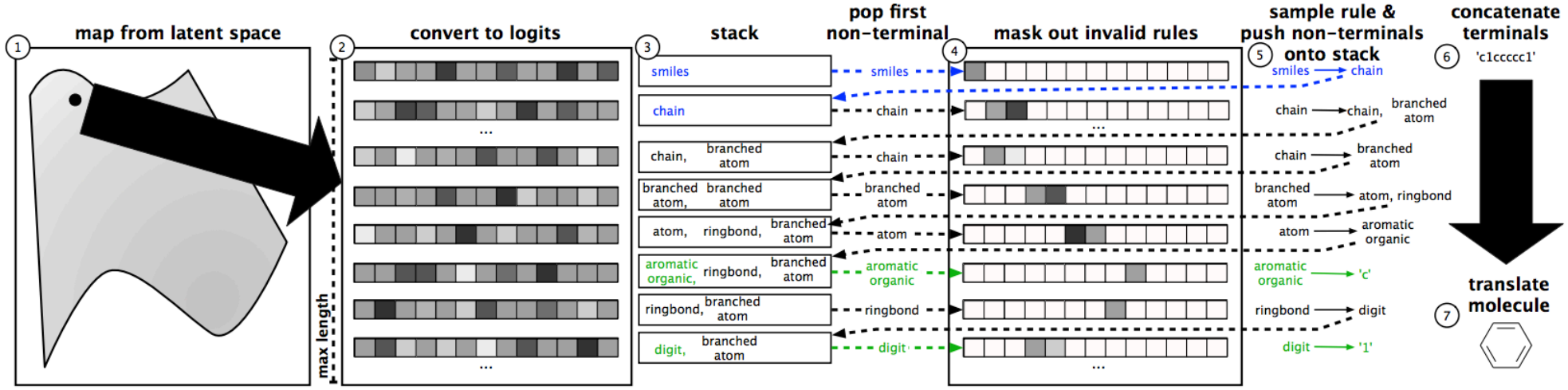
Rafael Gómez-Bombarelli *et al.,* Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules., ACS Cent. Sci., 2018, 4 (2), pp 268–276

Molecular Graph Generation: Grammar VAE (GVAE, Kusnar et al., 2017)

Encode

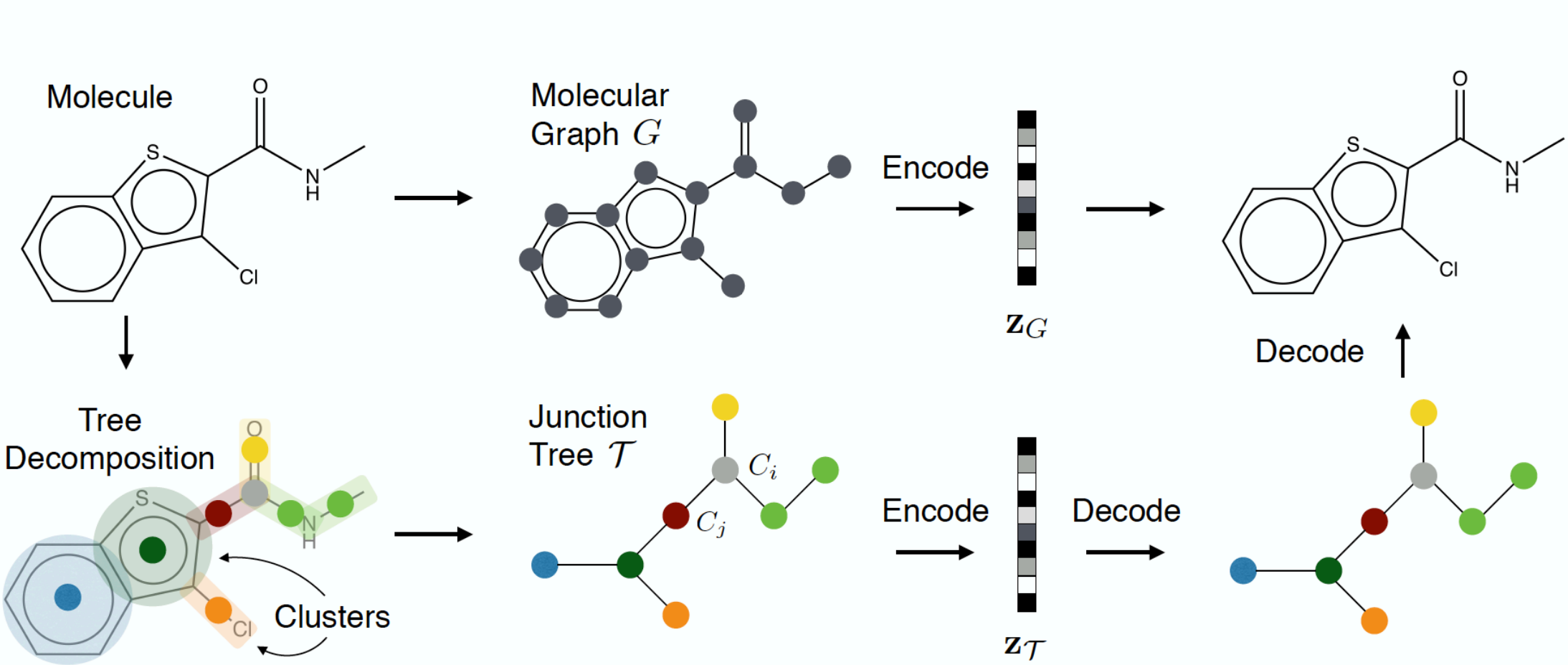


Decode



Kusnar et al., Grammar Variational Autoencoder, ICML' 17

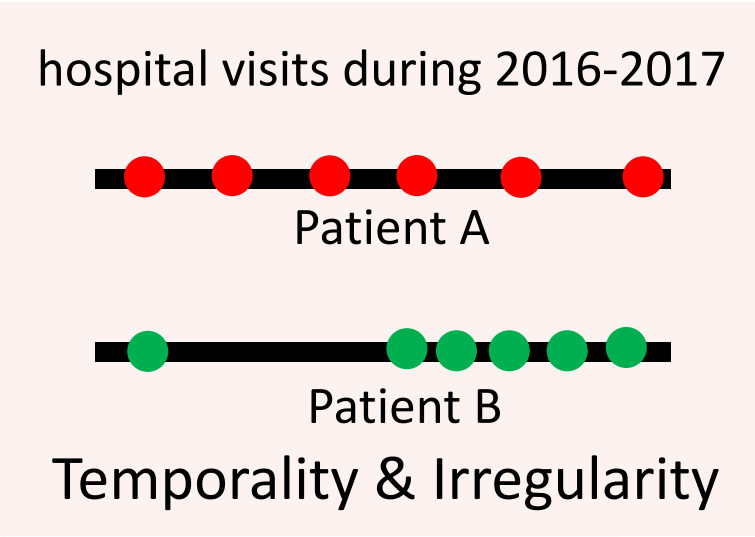
Molecular Graph Generation: Junction Tree VAE (Jin et al., 2018)

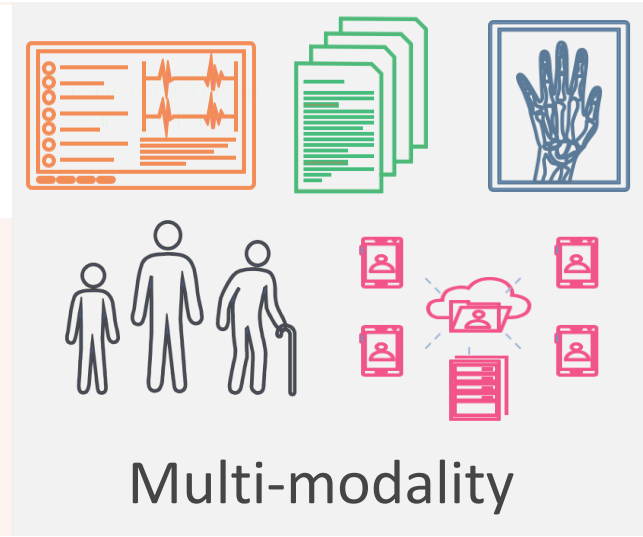


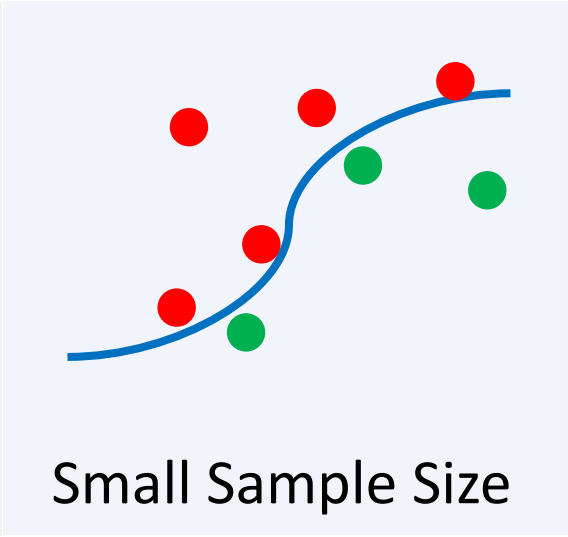
Agenda

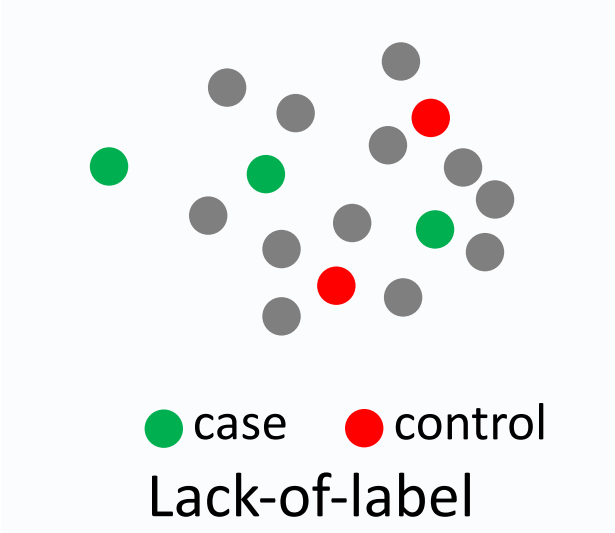
- Background (30 min)
  - healthcare data
  - analytical tasks
  - why deep learning models?
  - deep learning architectures
- Success of Deep Learning in Computational Healthcare (2 hr)
  - Medical Classification
  - Sequential Prediction
  - Concept Embedding
  - Data Augmentation
- Open Challenges
- Q&A

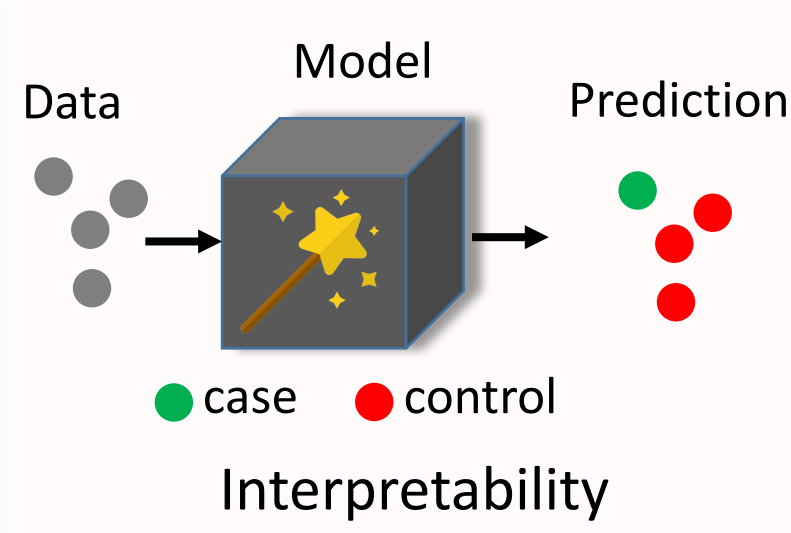
Challenges of Deep Learning for Healthcare Applications

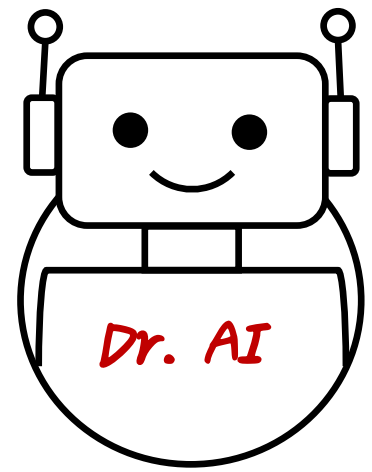












What's next?

Modeling heterogeneous data sources

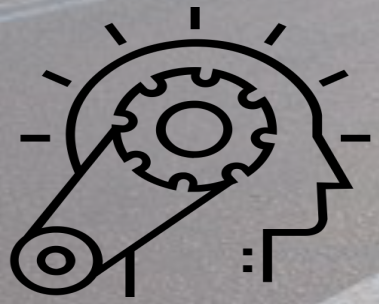
Clinical notes

-Omic data

sensor

Medical imaging

Model interpretation



More complex output

Clinical question & answer

Deep Learning for Healthcare

Jimeng Sun, Cao (Danica) Xiao, Edward Choi

August 7, 2018