



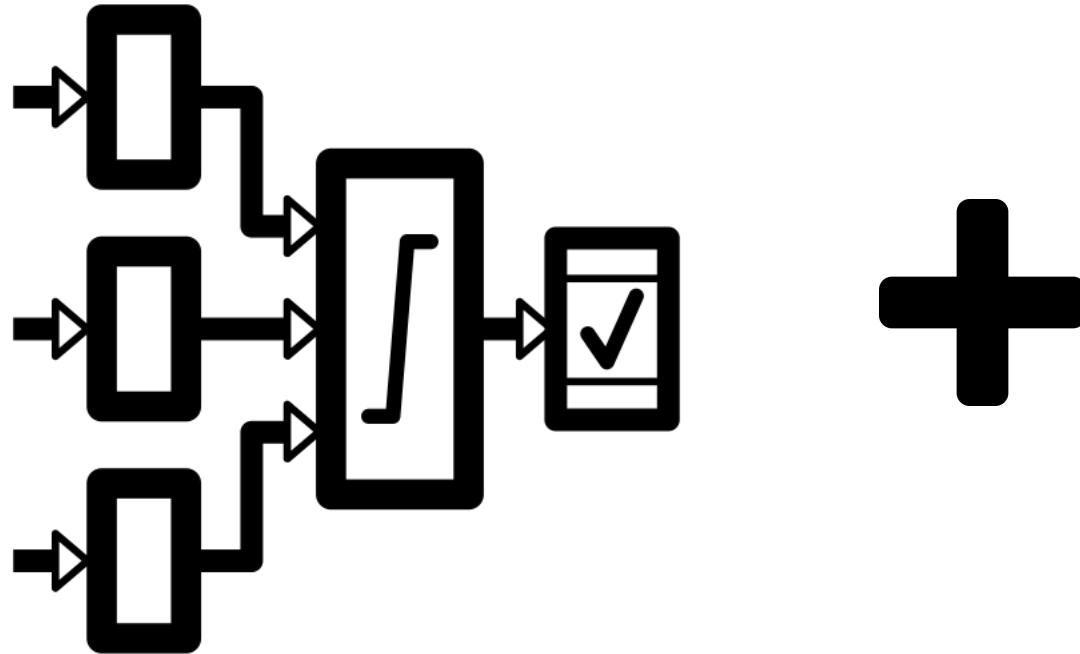
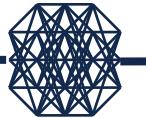
# Deep Learning for Healthcare

Memory  
networks

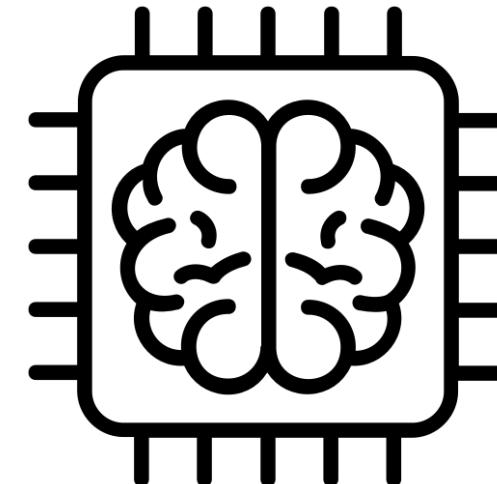
# Outline

- Memory networks
  - Original
  - End to end
- Self attention
  - Transformer
  - BERT
- Case studies
  - Doctor2Vec
  - Medication recommendation
    - GAMENET
    - Pretraining of graph augmented transformer

# Original memory network

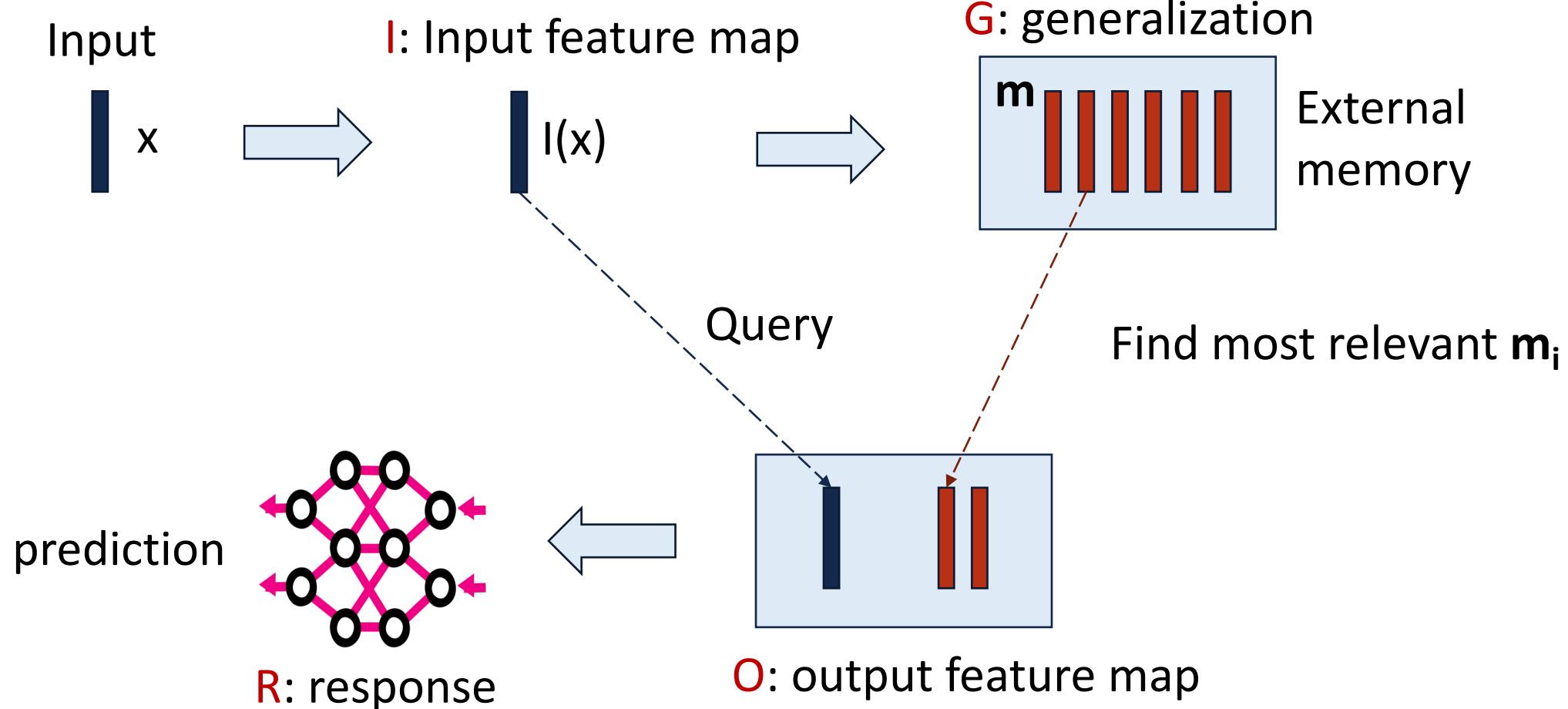


Deep neural network

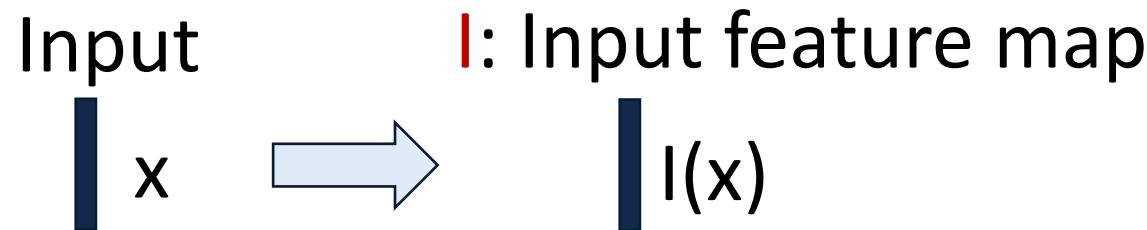


Memory components

# Overview of memory networks

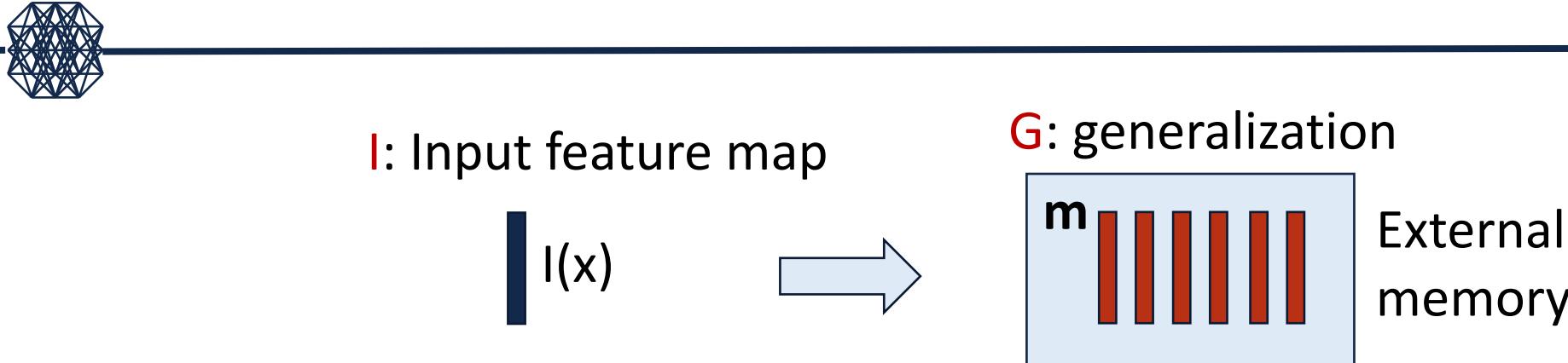


# I: Input feature map



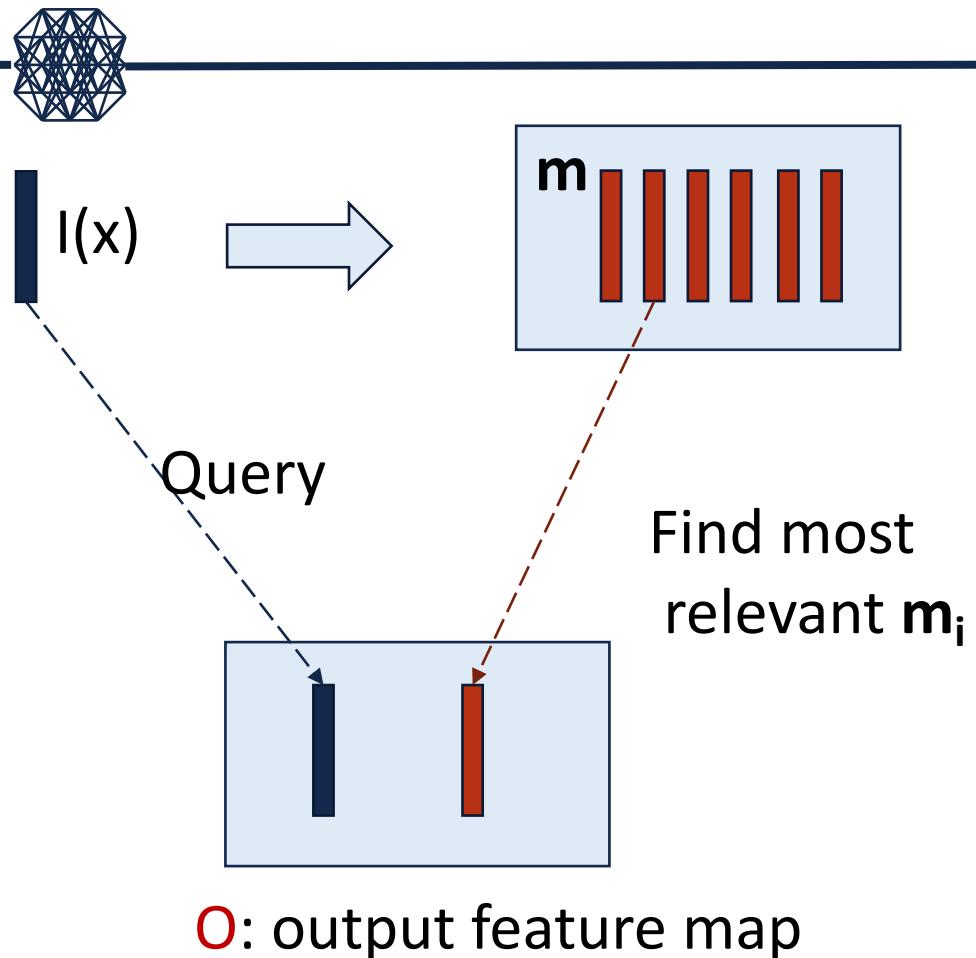
- Convert input  $x$  to an internal feature representation  $I(x)$
- Options:
  - Simple bag of words – one hot encoding
  - RNN if the inputs are sequences

# G: generalization



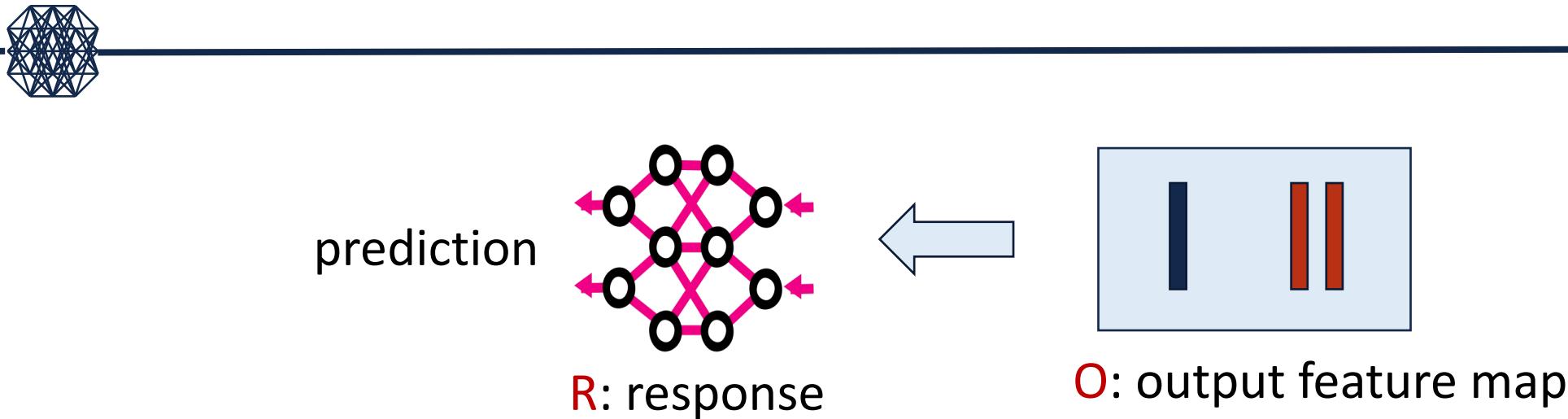
- Updates old memories given new input  $x$
- $m_i = G(m_i, I(x), m)$  for all  $i$ .
- Simplest version is to store  $I(x)$  in a slot of memory
  - $m_{H(x)} = I(x)$  where  $H(x)$  is the hash function for finding the memory location to store.

# O: output feature map



- Compute output feature  $o = O(I(x), m)$
- E.g., find the most relevant memory  $m_i$
- For  $k = 1$ , the most relevant memory is:
- $o_1 = O_1(I(x), m) = \text{argmax } (s_O(I(x), m_i))$  for all  $i$
- Final output  $o = [I(x), m_{o_1}]$

# R: response

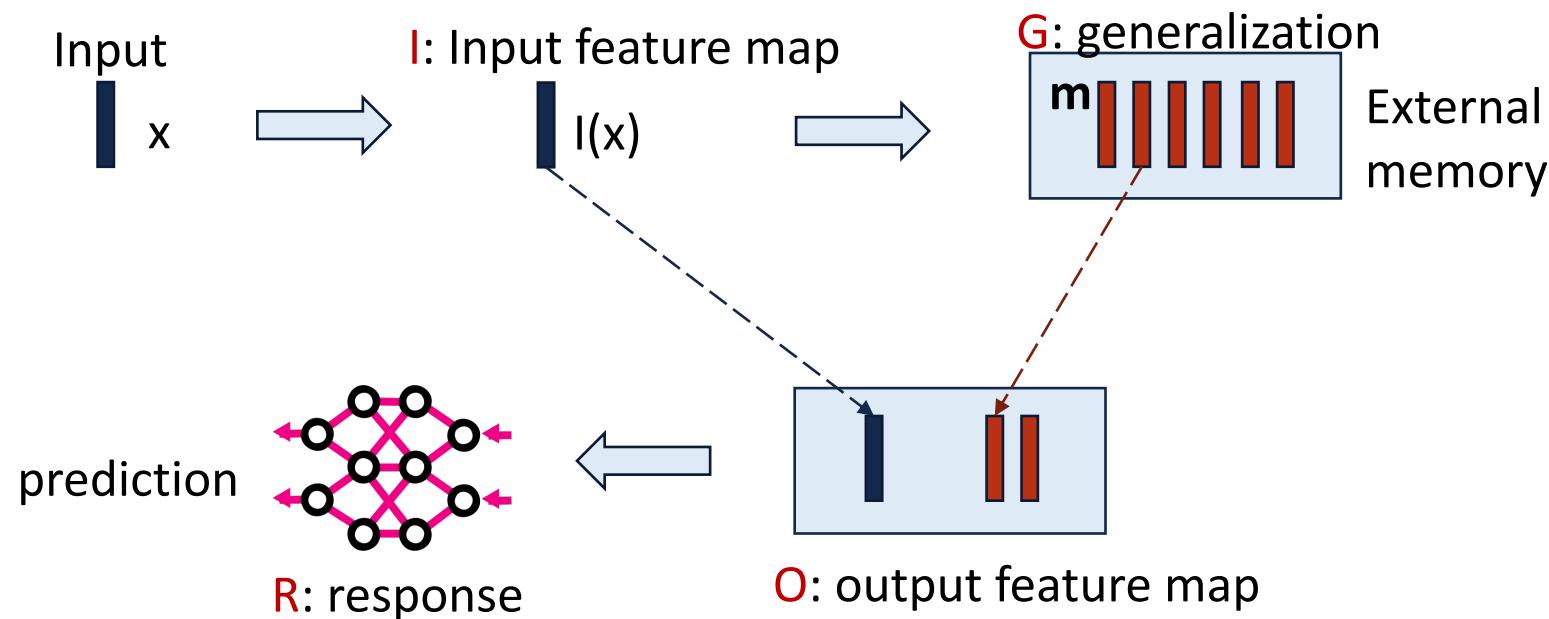


- Map output to the final response  $r = R(o)$
- E.g,
  - Softmax for classification
  - RNN for sequence generation

# Summary of memory network



- It bring a memory component into a deep neural network
- Limitation: not end-to-end trained because of argmax op for finding the optimal memory slot



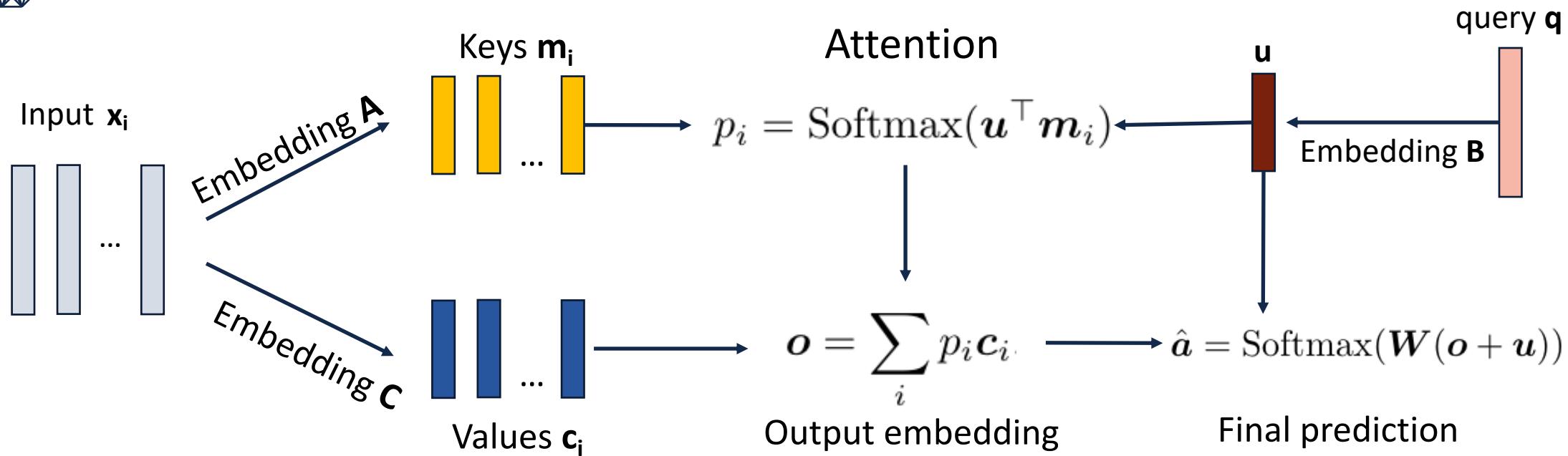
# End-to-end memory network



- **Motivation:** Original memory network cannot be trained end-to-end
- **Insight:** replace the argmax in the original memory network with softmax with attention
- **General idea**
  - Store input  $x_1, x_2, \dots, x_n$  in the memory component
  - Compute attention weights between a query  $q$  and all  $x_i$

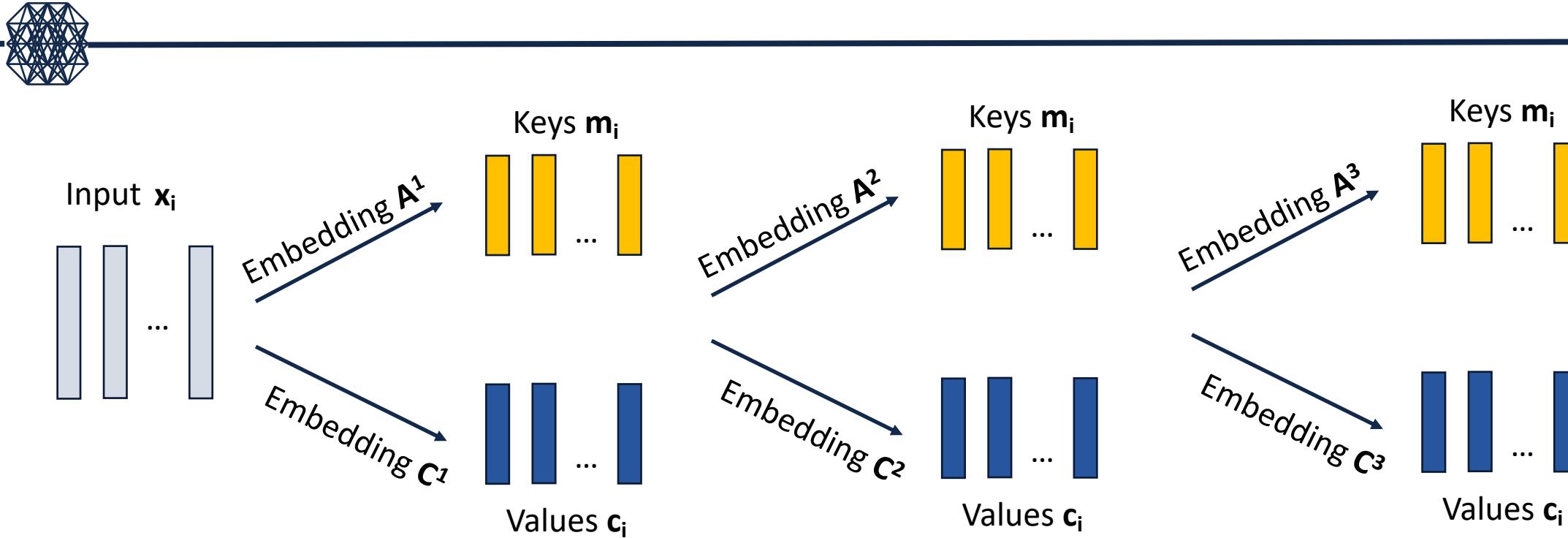
Sukhbaatar, Sainbayar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015.  
“End-To-End Memory Networks.” *arXiv [cs.NE]*. arXiv. <http://arxiv.org/abs/1503.08895>.

# End-to-end memory network



- Model parameters are
  - Embedding matrices A, B, C, W

# Multi-layer end-to-end memory network



- Parameter sharing strategies:
  - Adjacent:  $A^{k+1} = C^k$
  - Layer-wise:  $A^1 = A^2 = \dots = A^k, C^1 = C^2 = \dots = C^k$

# Summary: End-to-end memory network

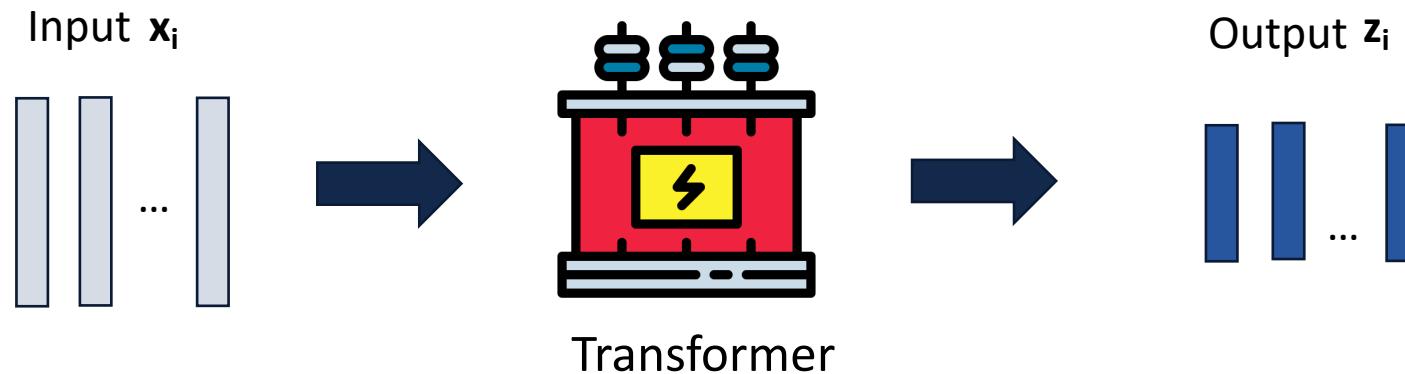


- A variant of memory network that can be trained end-to-end
- Key ideas:
  - Use softmax attention to replace argmax operation
  - Use question answer (QA) template to model memory network
  - Allow end-to-end training

# Transformer

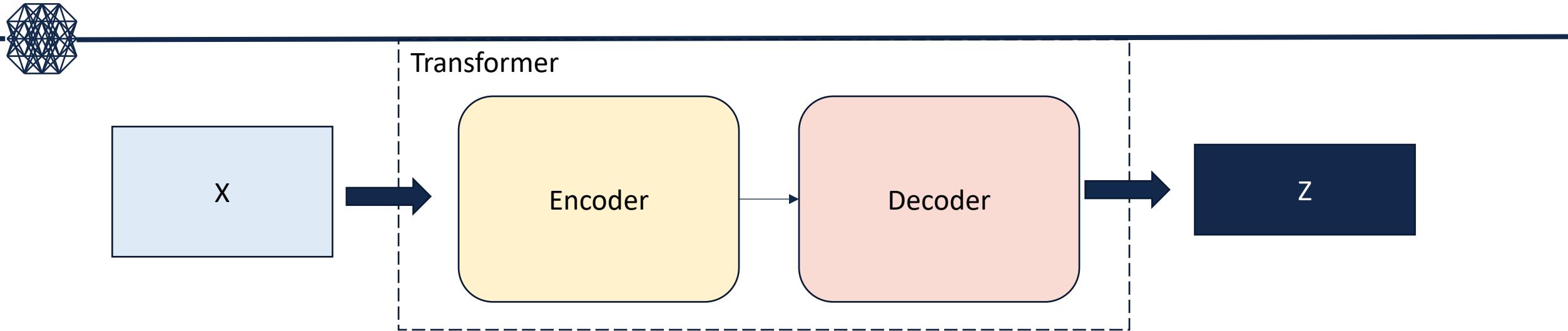


- Transformer is an effective embedding method for sequential data using **self-attention** strategy.
- Question: *Can we make seq2seq model with attention **train faster**?*



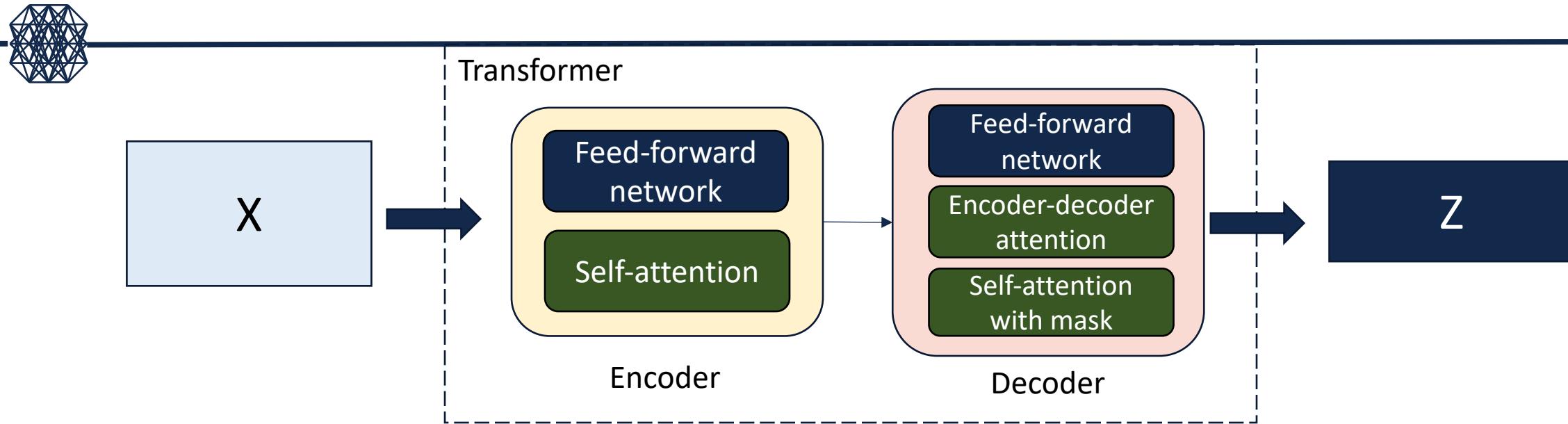
Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. "Attention Is All You Need." *arXiv [cs.CL]*. arXiv. <http://arxiv.org/abs/1706.03762>.

# High-level view of transformer



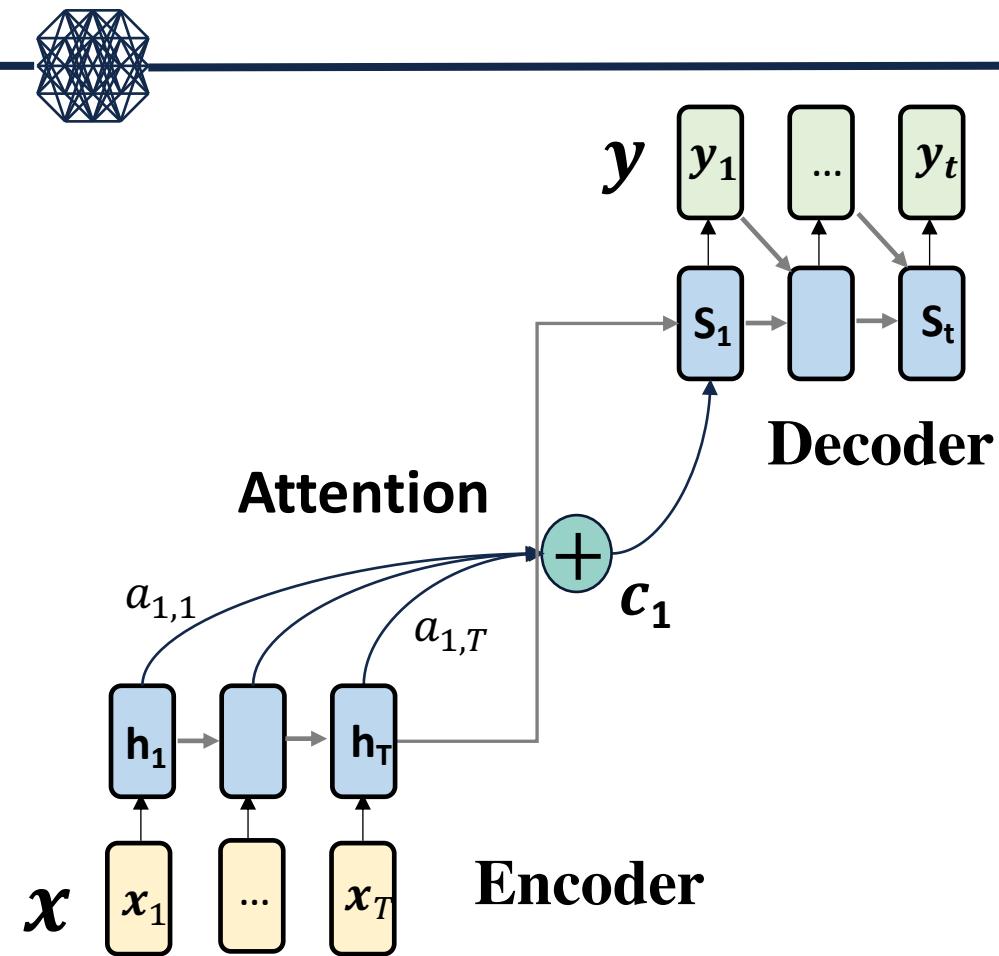
- An encoder-decoder model (sounds familiar, right?)

# High-level view of transformer



- An encoder-decoder model with self-attention modules

# Review: Attention on RNN Model

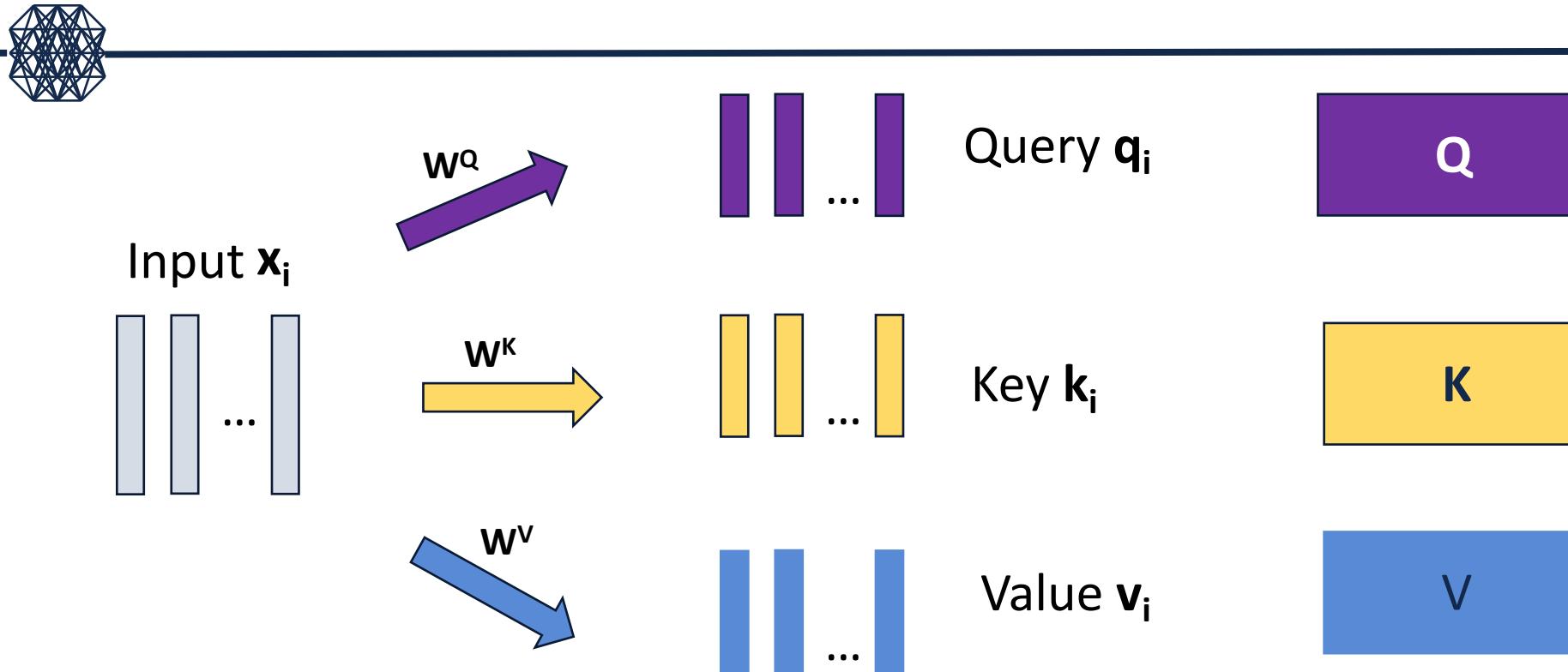


RNN cannot be trained in parallel due to its recurrence dependency



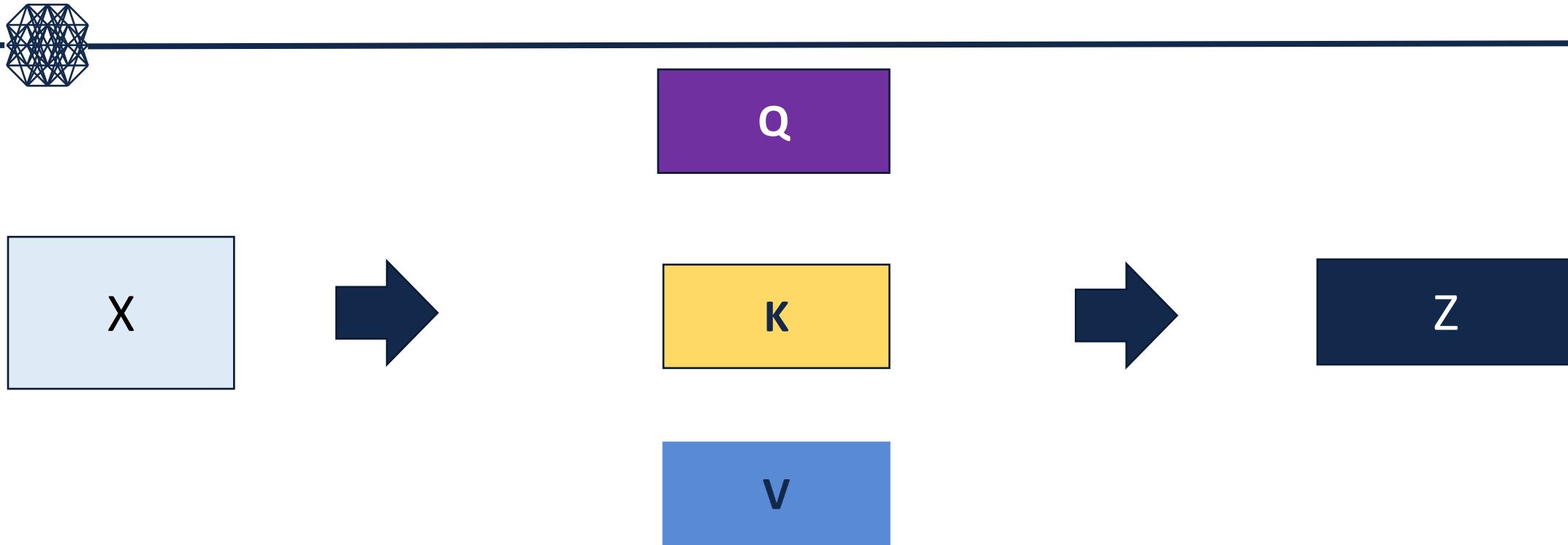
Remove RNN, and put attentions directly on input  $x_i$

# Self-attention



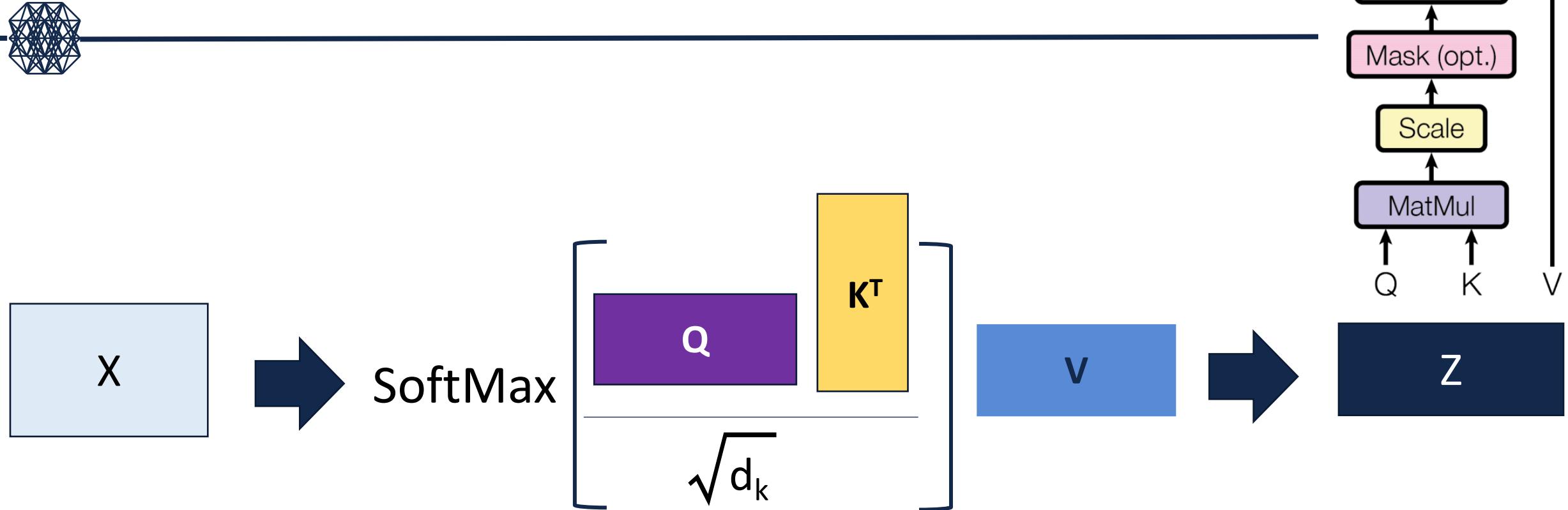
- Attention among input themselves
- Use a query retrieval strategy
- Define 3 embeddings of input  $x$ :  $w^Q$ ,  $w^K$ ,  $w^V$

# Self-attention (cont.)



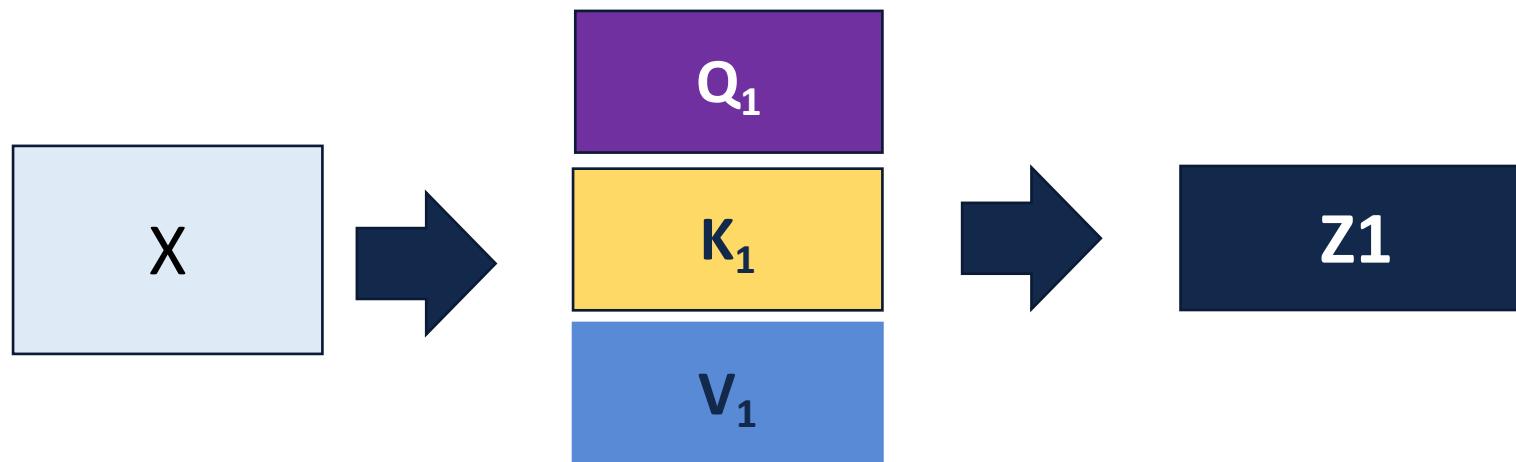
- Three versions of embeddings can be learned in parallel to speed up learning

# Self-attention (cont.)



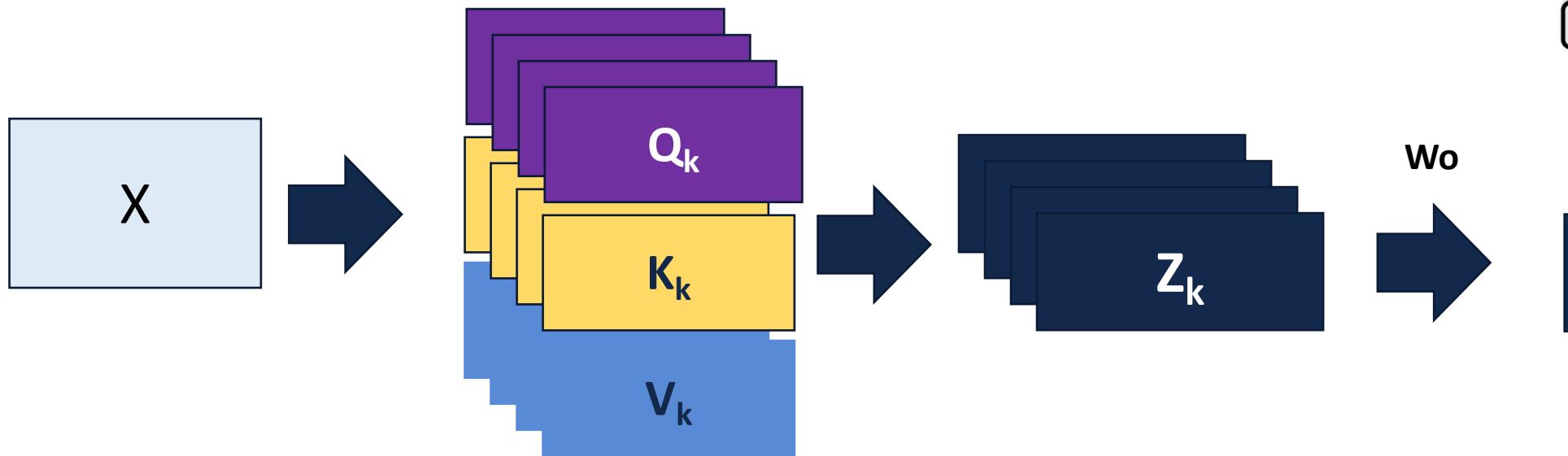
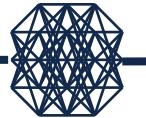
- Scaled dot product is computed as the similarity score vector with  $Q \cdot K$
- Based on similarity score, retrieve/recombine value matrix  $V$

# Multi-head attention

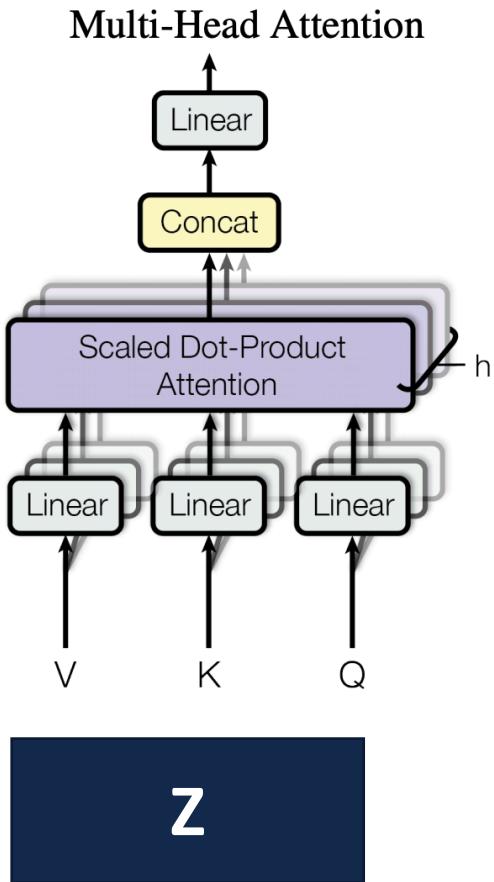


- Repeat the same attention procedures multiple times with different initialization

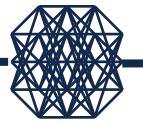
# Multi-head attention



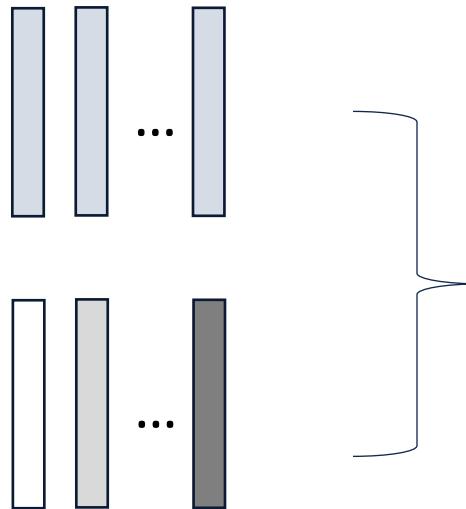
- Repeat the same attention procedures multiple times with different initialization
- Recombine at the end  $Z = [Z_1, Z_2, \dots, Z_k] W_o$



# Other idea: Positional encoding

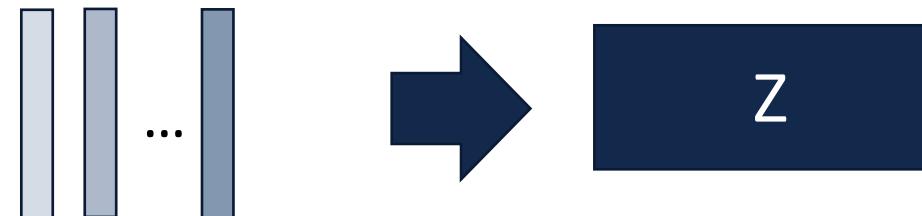


Input  $x_i$



Positional enhanced

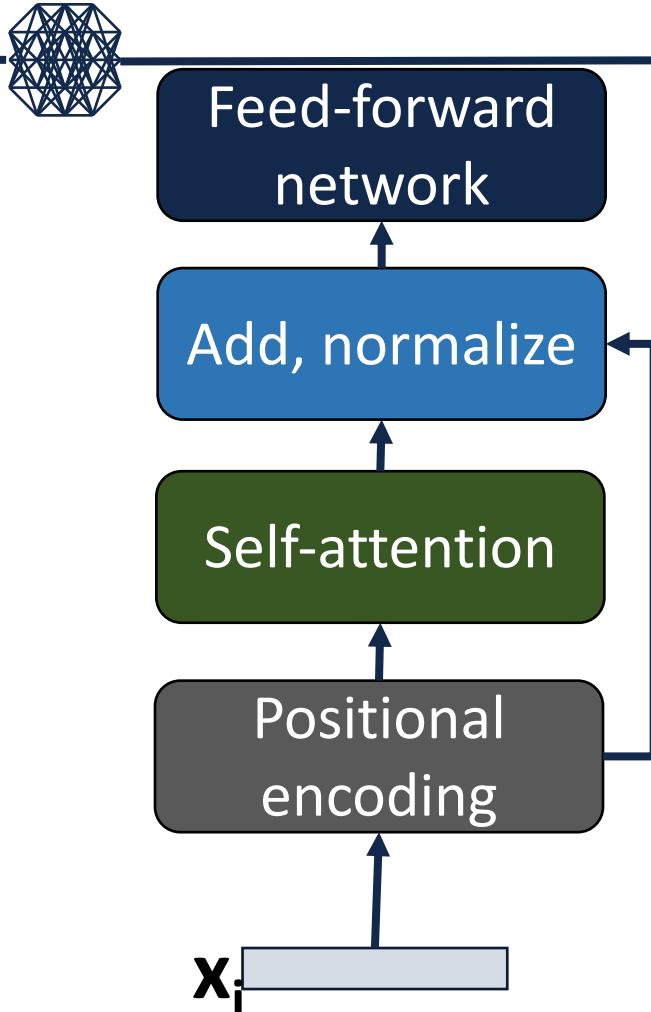
input  $x_i$



Position info  $t_i$

- For sequential data, we can add some position specific information to the input

# Other idea: Residual connection

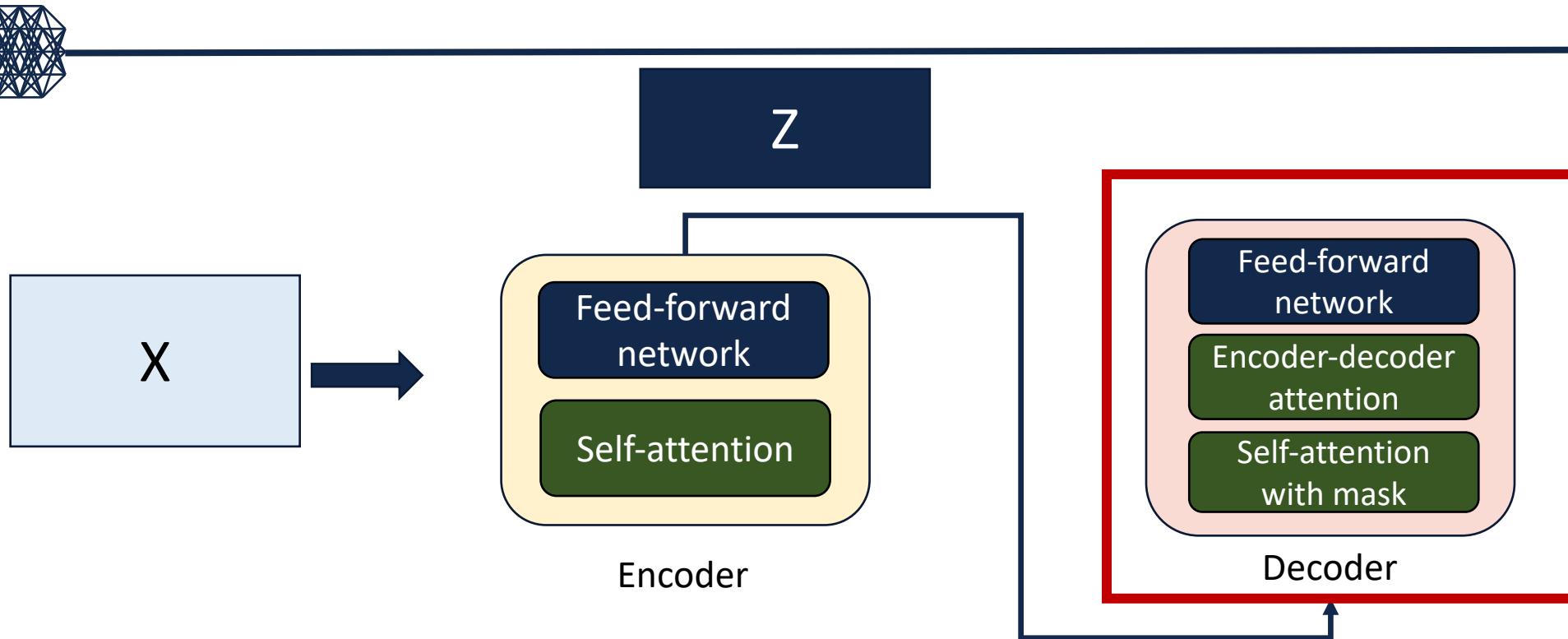


- Add the input to self-attention output
- Use layer normalization
  - subtract mean and rescale by standard deviation across all neurons

$$\mathbf{h}^t = f \left[ \frac{\mathbf{g}}{\sigma^t} \odot (\mathbf{a}^t - \mu^t) + \mathbf{b} \right]$$
$$\mu^t = \frac{1}{H} \sum_{i=1}^H a_i^t \quad \text{Mean}$$
$$\sigma^t = \sqrt{\frac{1}{H} \sum_{i=1}^H (a_i^t - \mu^t)^2} \quad \text{Standard deviation}$$

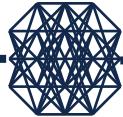
Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. "Layer Normalization." *arXiv [stat.ML]*. arXiv. <http://arxiv.org/abs/1607.06450>.

# Decoder of transformer

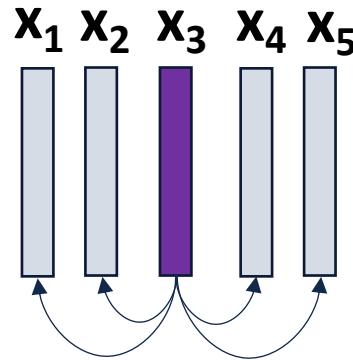


- Output embeddings of encoders become the input to decoder
- Decoder components: variant of self-attention:
  - Self-attention with future mask
  - Encoder-decoder attention

# Self-attention with future mask

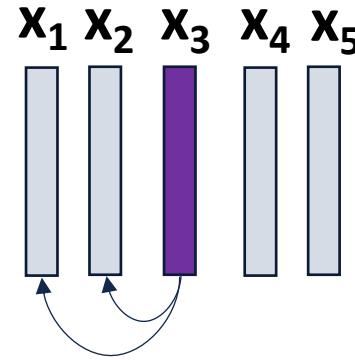


Standard self-attention

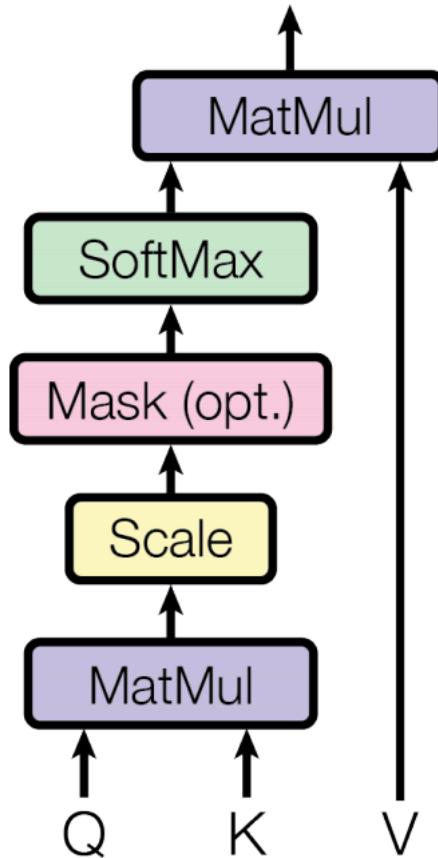


Attention of  $x_3$  to everyone

Attention with future mask

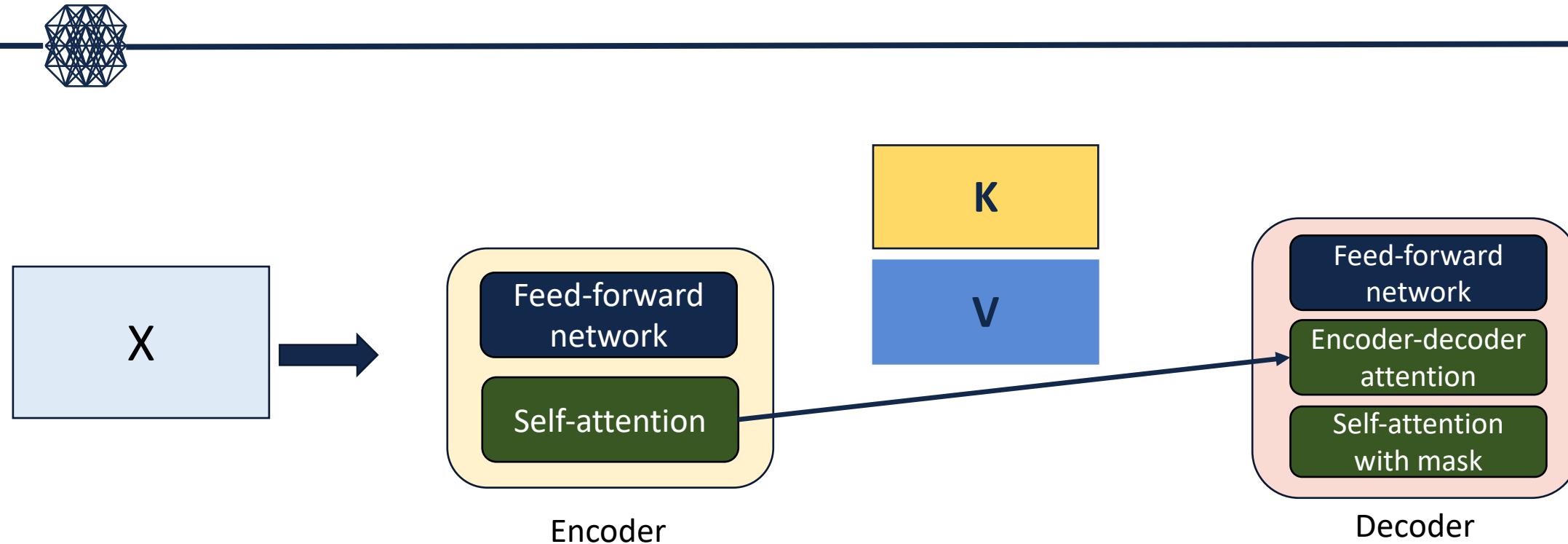


Attention of  $x_3$  only  
on previous timestamps  $x_1, x_2$



- Attention mask will be put  $-\infty$  on the future timestamps

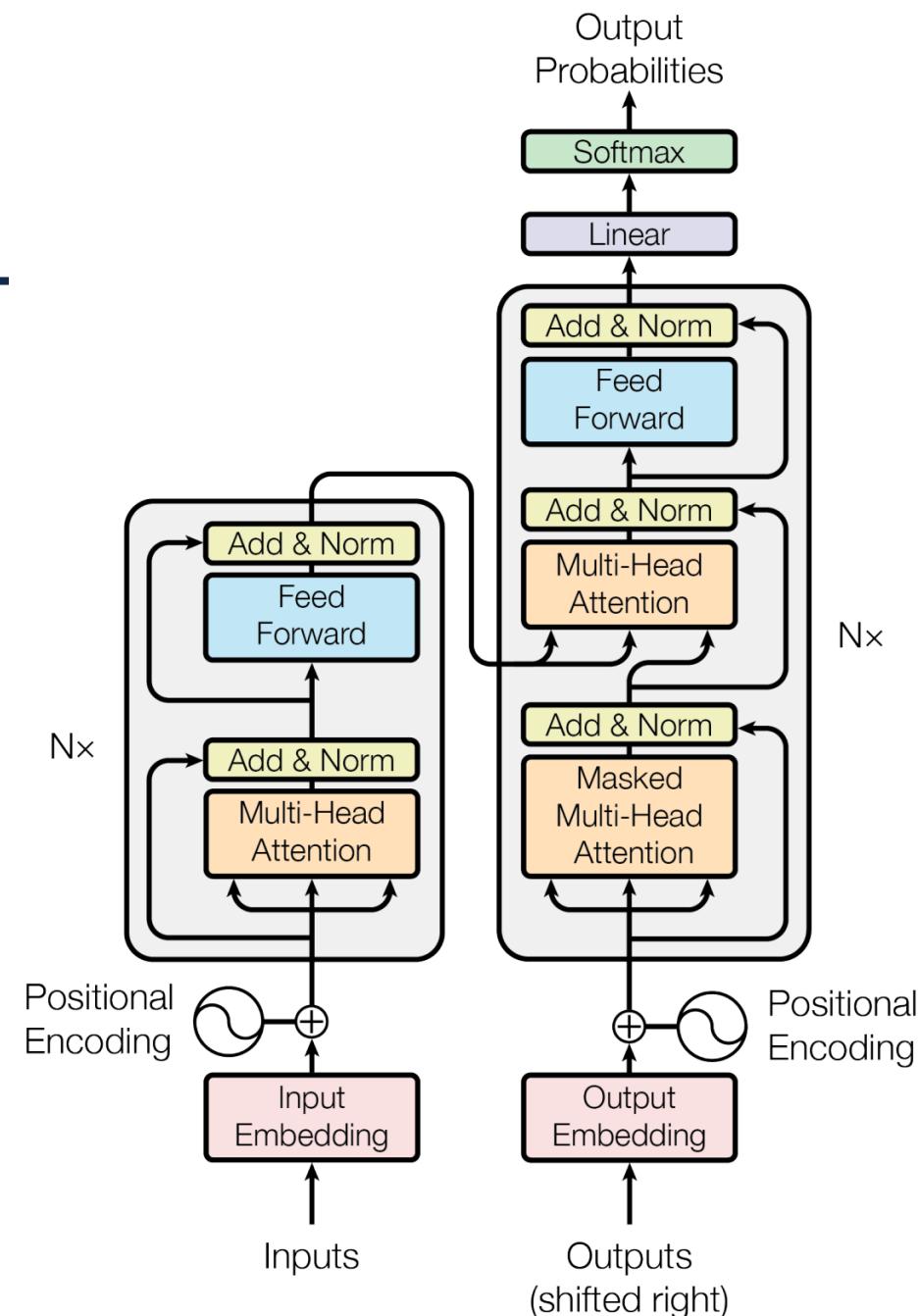
# Encoder-decoder attention



- Keys K and Values V from the encoder are reused in encoder-decoder attention

# Summary of Transformer

- Key idea is self-attention
  - Remove sequential dependency (No RNN module)
  - Maximize parallelism (attention with key-value pairs)
  - Takes 3 embedding matrices  $\mathbf{K}$ ,  $\mathbf{V}$ ,  $\mathbf{Q}$ , which are all transformation from input  $\mathbf{X}$ .
- Other important ideas:
  - Multi-head attention
  - Positional encoding
  - Residual connection



# BERT: Pre-training of Deep Bidirectional Transformers

---

- BERT model is an application of Transformer for NLP
- Key ideas



Context specific embedding



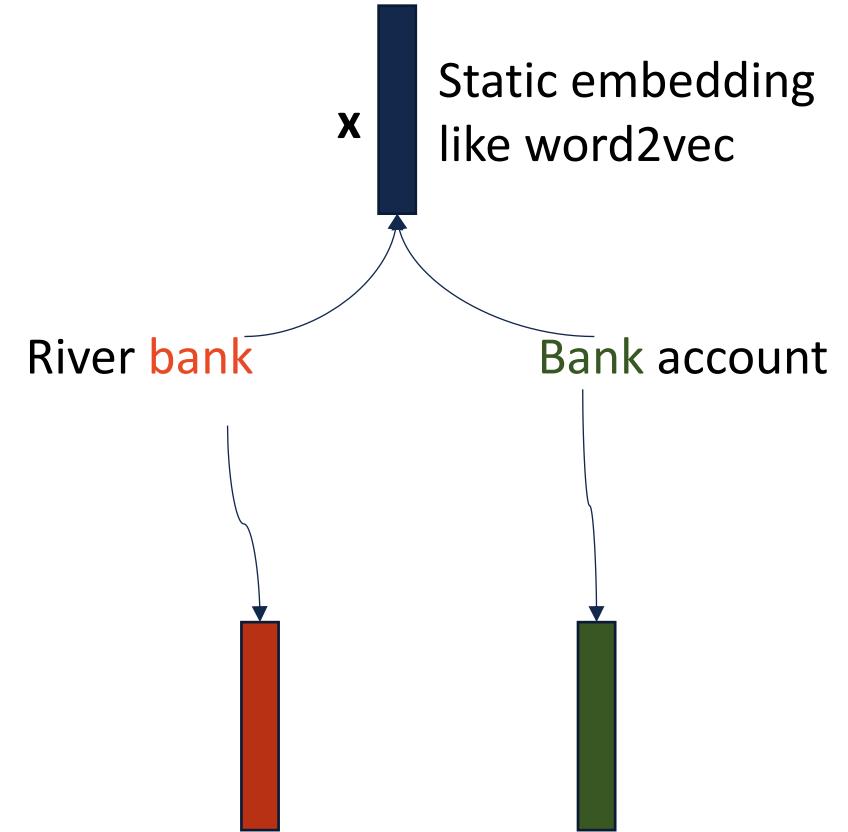
Masked language model

Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. "BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding." *arXiv [cs.CL]*. arXiv. <http://arxiv.org/abs/1810.04805>.

# Context specific embedding



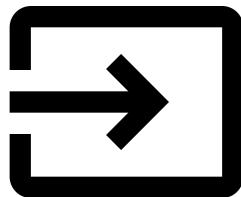
- Previous word embeddings (e.g., Word2Vec):
  - Static embedding for each word
- BERT computes dynamic embedding for each word



# Masked language model



- Existing deep learning language models:
  - RNN: Embeddings are trained from left to right
  - Bidirectional RNN: Concatenation of 2 RNNs (left to right and right to left)
  - BERT masks x% input words, and try to predict what they are



*Nobody ever won a chess game by resigning*



*Nobody ever won a chess ga?me by resigning*

# **Doctor2Vec: Dynamic Doctor Representation Learning for Clinical Trial Recruitment**

Biswal, Siddharth, Cao Xiao, Lucas M. Glass, Elizabeth Milkovits, and Jimeng Sun. 2020.  
“Doctor2Vec: Dynamic Doctor Representation Learning for Clinical Trial Recruitment.” AAAI

# Agenda



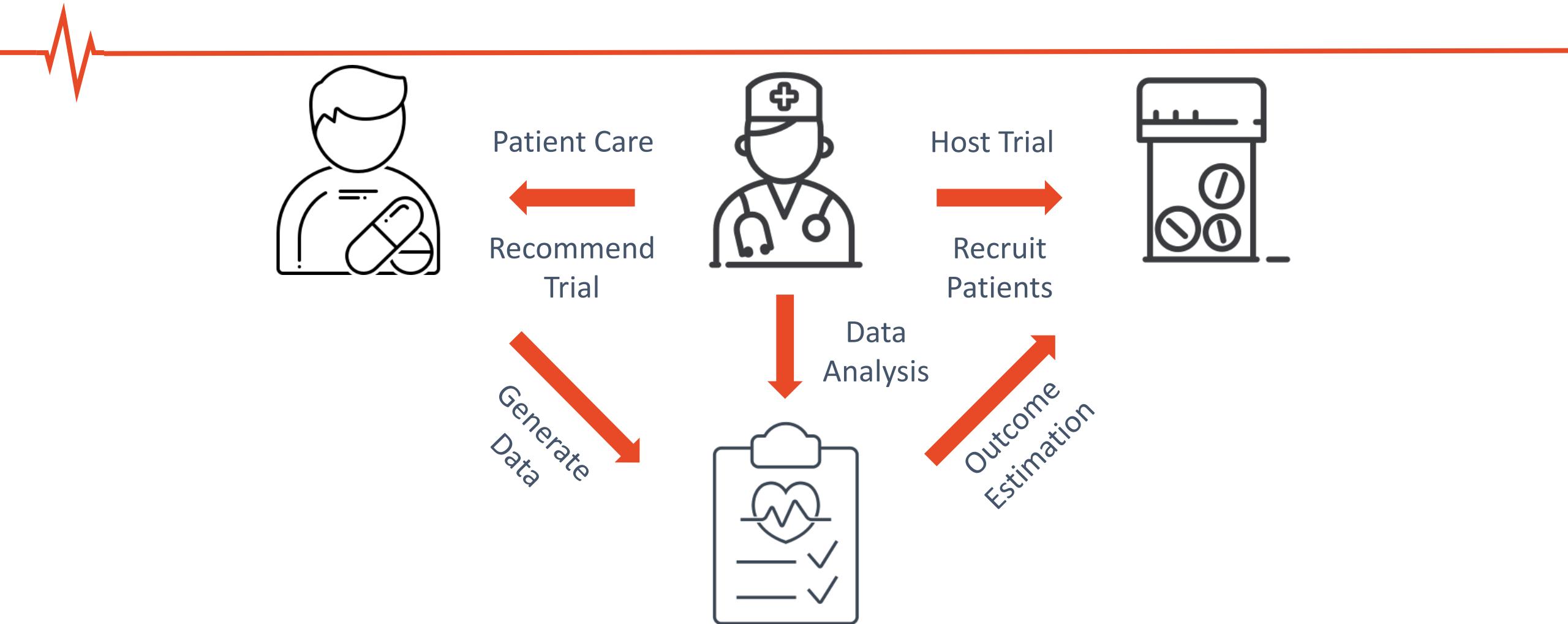
- ▶ Background
- ▶ Doctor2Vec
- ▶ Experiments

# Agenda

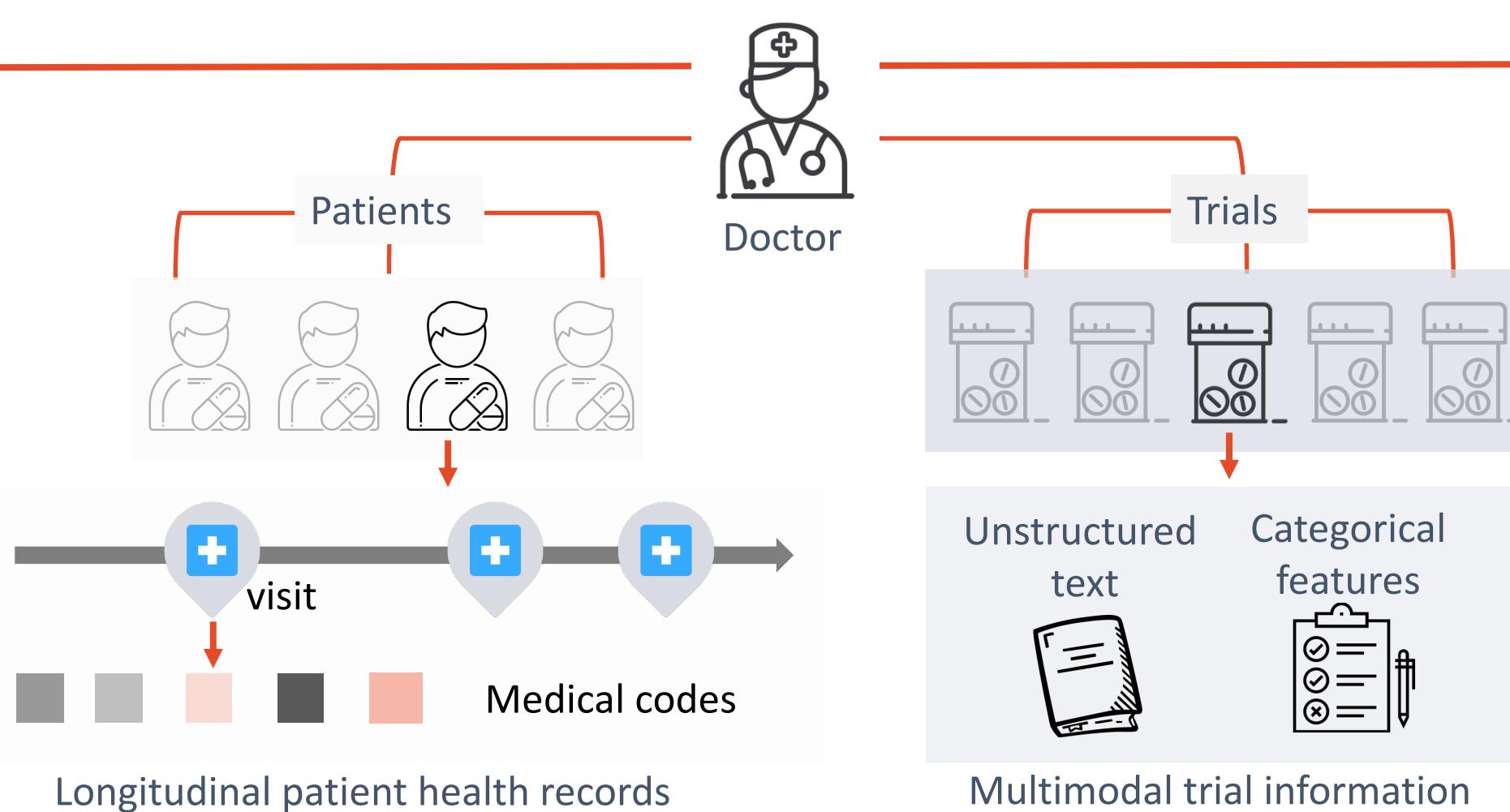


- ▶ Background
- ▶ Doctor2Vec
- ▶ Experiments

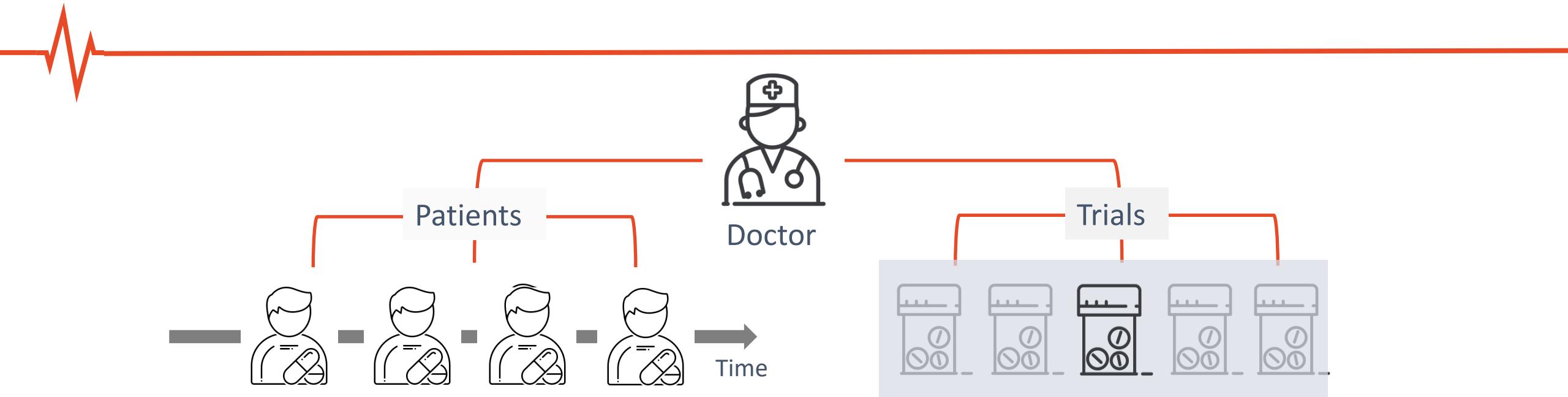
# Doctors Play Pivotal Roles in Healthcare



# Doctor Representation Learning



# Challenges of Doctor Representation Learning



1

Existing works do not capture the time-evolving patterns of doctors experience/expertise.

2

Existing works learn a static doctor representation, rather than a dynamic one based on the corresponding trial.



# Our Contribution

## Challenges

1

Existing works do not capture the time-evolving patterns of doctors experience/expertise.

2

Existing works learn a static doctor representation, rather than a dynamic one based on the corresponding trial.

## Solutions

1

Patient embedding as a memory for dynamic doctor experience encoding.

2

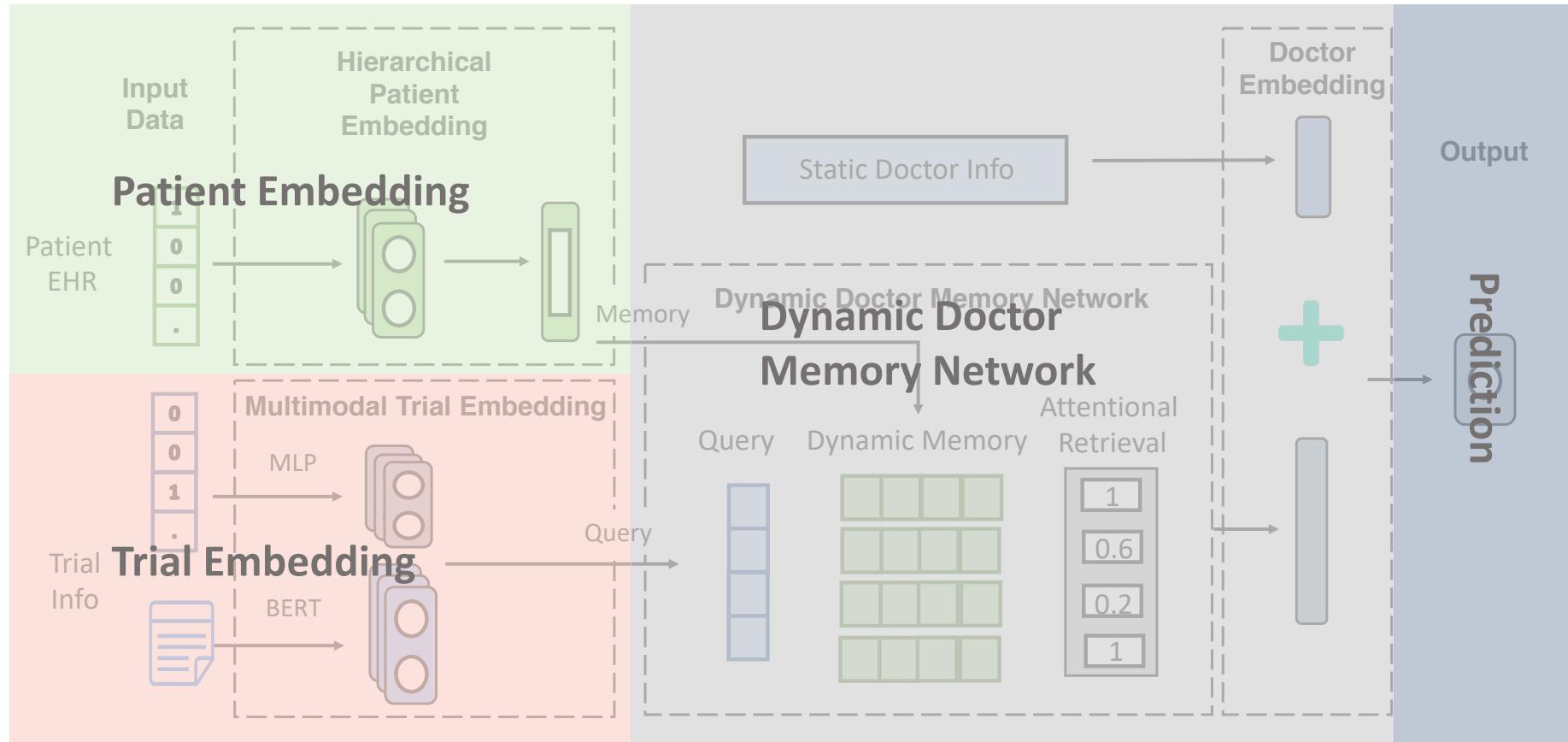
Trial embedding as a query for improved doctor embedding.

# Agenda

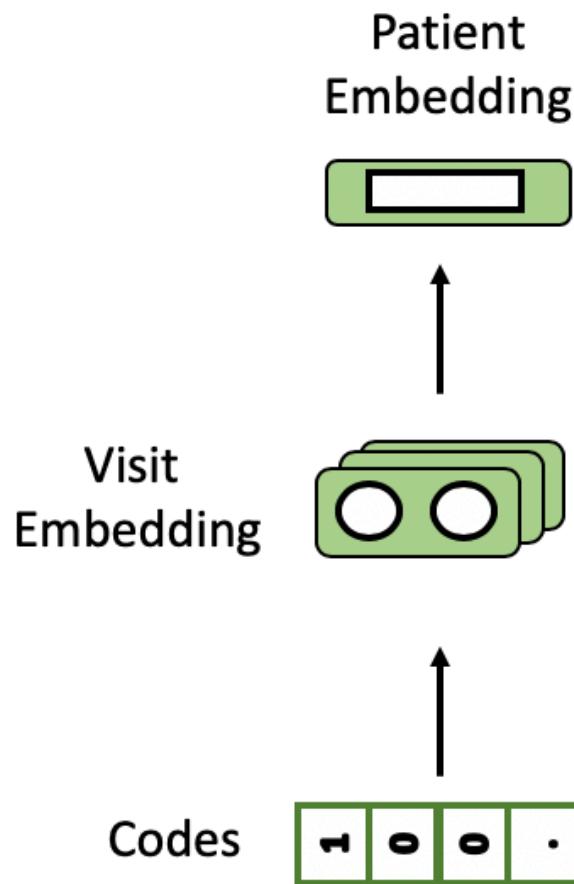


- ▶ Background
- ▶ Doctor2Vec
- ▶ Experiments

# Doctor2Vec: Overview



# Doctor2Vec: Hierarchical Patient Embedding



$$I(k) = \sum \alpha_t \cdot h_t \quad \text{Memory for memory networks}$$

$$g_1, g_2, \dots, g_t = \text{bi-LSTM}(h_1, h_2, \dots, h_t)$$

$$e_t = w_\alpha^T * g_t + b_\alpha$$

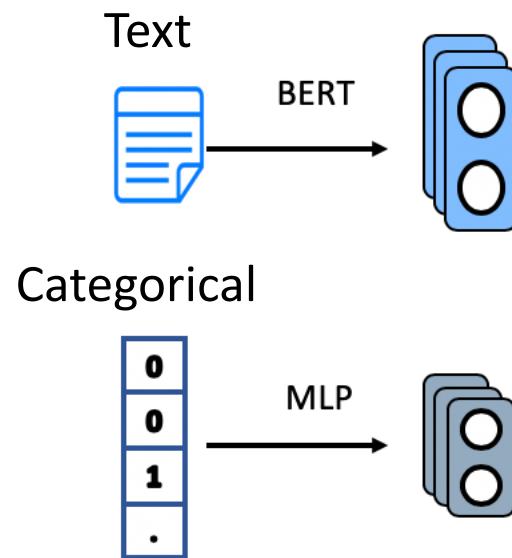
$$\alpha_1, \alpha_2, \dots, \alpha_t = \text{softmax}(e_1, e_2, \dots, e_t)$$

$$h_t(k) = W_{\text{emb}} * v_t(k)$$

# Doctor2Vec: Multimodal Trial Embedding



Pretrained BERT using MIMIC text data to extract embeddings for each word.



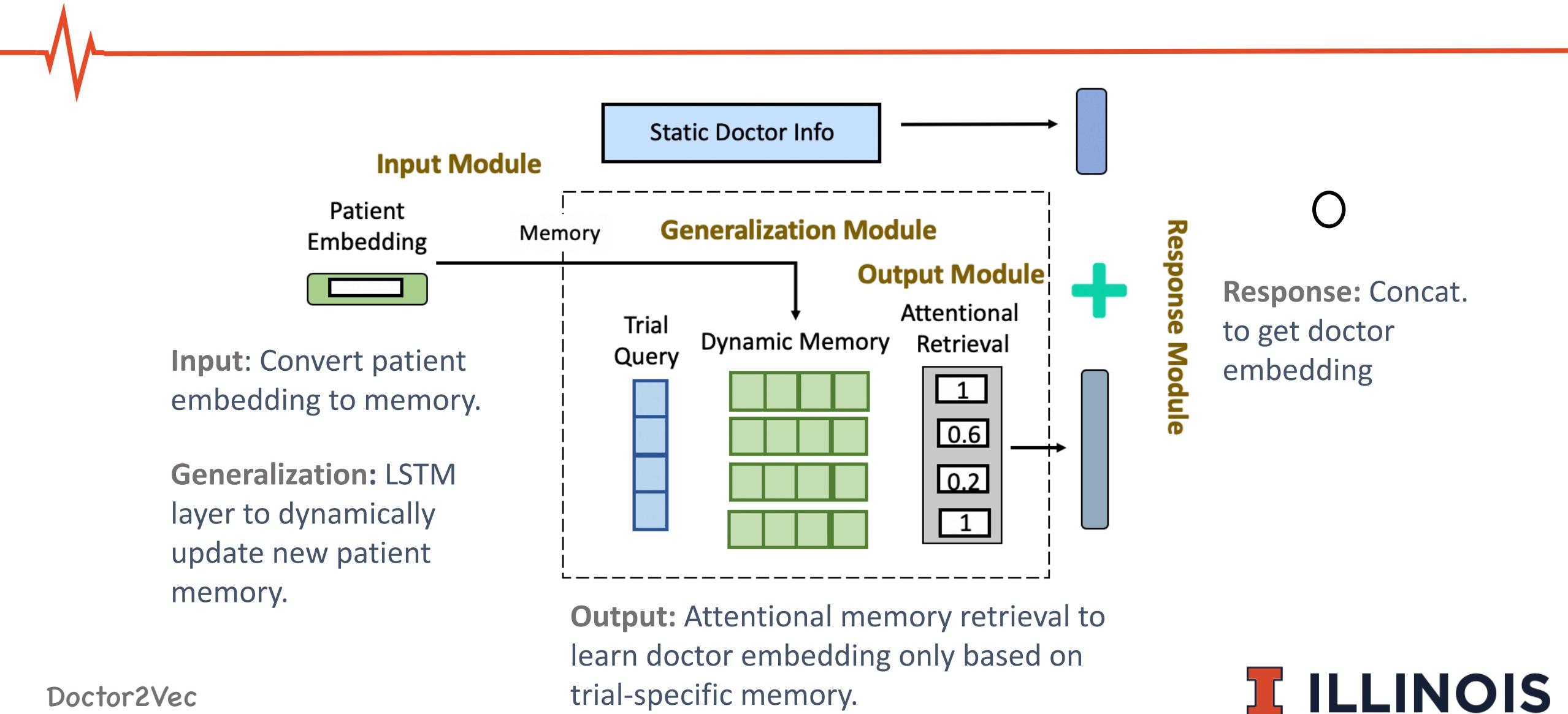
$$Q(\text{text})_{\text{emb}} = \text{BERT}(Q(\text{text})^{(l)})$$
$$Q(\text{cat})_{\text{emb}} = \mathbf{W}_c(Q(\text{cat})^{(l)}) + \mathbf{b}_c$$



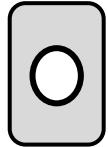
Query for memory networks

$$\mathbf{Q}_{\text{emb}}(l) = (\mathbf{W}_{ci} Q(\text{cat})_{\text{emb}} + \mathbf{b}_{ci}) \odot (\mathbf{W}_{ti} Q(\text{text})_{\text{emb}} + \mathbf{b}_{ti})$$

# Doctor2Vec: Dynamic Doctor Memory Network



# Doctor2Vec: Prediction



Output predicted enrollment rate between 0 and 1

$$\mathbf{Y} = \text{Softmax}([\mathbf{Doc}_{emb}; \mathbf{Q}_{emb}(l); \mathbf{Doc}_{static}])$$

Regression  
Task

Category	[0,0.2]	(0.2,0.4]	(0.4,0.6]	(0.6,0.8)	(0.8,1.0]	Classification
Distribution	12%	33%	37%	12%	6%	Task

# Agenda



- ▶ Background
- ▶ Doctor2Vec
- ▶ Experiments

# Design of Experiments



Q1: Does **Doctor2Vec** have better performance in predicting clinical trial enrollment to support site selection?

Q2: Can **Doctor2Vec** embedding perform in transfer learning setting for trials across countries or across diseases?

# Data

- 
- a) IQVIA trial data about trials formed during 2014 and 2019 across 28 countries.
  - b) clinical trial description from clinicaltrials.gov, matched with IQVIA trial data on NCT ID
  - c) IQVIA claims data

Table 2: Data Statistics

# of clinical trials	2609
# of doctors	25894
# of doctor-trial pair(samples)	102487
# of patients	430,239
Avg # of Dx codes per visit	4.23
Max # of Dx codes per visit	56
Avg # of Procedure codes per visit	1.23
Max # of Procedure codes per visit	18
Avg # of Med codes per visit	9.36



# Metrics

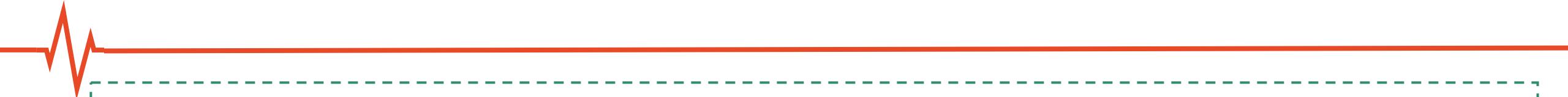
## Classification Task

The precision recall area under curve (PR-AUC) is the area under the PR curve. A good metric for data imbalanced setting. The higher the better.

## Regression Task

The coefficient of determination (R-squared) is the square of the correlation between predicted scores and actual scores. The higher the better.

# Baseline

- 
- **Median Enrollment (Median):** considers the median enrollment rate for each therapeutic area as estimated rate for all trials in that area.
  - **Logistic Regression (LR):** Combine all features and then apply LR.
  - **Random Forest (RF):** Combine all features and then apply RF.
  - **AdaBoost:** Combine all features and then apply AdaBoost.
  - **Multi-layer Perceptron (MLP):** Convert codes to count vectors, convert categorical information of clinical trials to multi-hot vectors and obtain TF-IDF features from text information of clinical trials. Then apply MLP.
  - **Long Short-Term Memory Networks (LSTM):** process all temporal data using LSTM and then concatenate with other features.
  - **DeepMatch:** Features for the doctors are obtained from the top 50 most frequent medical codes and passed through an MLP layer to obtain an embedding vector.

# Results



	PR-AUC	$R^2$ Score
Median	$0.571 \pm 0.014$	$0.54 \pm 0.072$
LR	$0.672 \pm 0.041$	$0.314 \pm 0.082$
RF	$0.731 \pm 0.034$	$0.618 \pm 0.034$
AdaBoost	$0.747 \pm 0.002$	$0.684 \pm 0.146$
MLP	$0.761 \pm 0.019$	$0.762 \pm 0.049$
LSTM	$0.792 \pm 0.034$	$0.780 \pm 0.621$
DeepMatch	$0.735 \pm 0.068$	$0.821 \pm 0.073$
<b>Doctor2Vec</b>	<b><math>0.861 \pm 0.021</math></b>	<b><math>0.841 \pm 0.072</math></b>

**Doctor2Vec** has 8.7% relative improvement in PR-AUC over the best baseline LSTM.

- ✓ LSTM > MLP > Other non temporal models, due to better model the temporal information.
- ✓ DeepMatch models achieved much lower PR-AUC since the model leverages the 50 most frequent codes for embedding, thus miss important but non-frequent information.
- ✓ DeepMatch in the regression settings tends to perform better than MLP and LSTM, due to the skewed data distribution.

# Transfer Learning Results



## Transfer to a less populated or newly explored country

Train model on 1443 clinical trials in the United states during the time 2014-2019 and test on 47 clinical trials in South Africa during the time 2014-2019.

	PR-AUC	R <sup>2</sup> Score
Median	0.524 ± 0.032	0.420 ± 0.039
LR	0.601 ± 0.023	0.279 ± 0.014
RF	0.661 ± 0.038	0.552 ± 0.048
AdaBoost	0.672 ± 0.01	0.581 ± 0.039
LSTM	0.758 ± 0.013	0.721 ± 0.025
DeepMatch	0.703 ± 0.087	0.756 ± 0.031
<b>Doctor2Vec</b>	<b>0.862 ± 0.003</b>	<b>0.817 ± 0.025</b>

**Doctor2Vec** achieved 13.7% better PR-AUC then LSTM and 8.1% R2 then DeepMatch.

Doctor2Vec

## Transfer to rare or low prevalence diseases

Test on 38 clinical trials for drugs about idiopathic pulmonary fibrosis (IPF) and inflammatory bowel disease(IBM). Train on 2569 clinical trials from the rest of the available diseases.

	PR-AUC	R <sup>2</sup> Score
Median	0.413 ± 0.013	0.387 ± 0.001
LR	0.521 ± 0.021	0.225 ± 0.028
RF	0.610 ± 0.019	0.517 ± 0.032
AdaBoost	0.623 ± 0.002	0.548 ± 0.046
LSTM	0.725 ± 0.002	0.623 ± 0.038
DeepMatch	0.638 ± 0.021	0.678 ± 0.049
<b>Doctor2Vec</b>	<b>0.784 ± 0.032</b>	<b>0.716 ± 0.014</b>

**Doctor2Vec** achieved 8.1% better PR-AUC then LSTM and 5.2% R2 then DeepMatch.

**I** ILLINOIS



# Case Study

## Ground Truth

**Trial:** Phase I trial for Gemcitabine plus Cisplatin (a combination of cancer therapy)

**Doctor:** a doctor in USA who has worked in internal medicine during past 3 years. The doctor has a broader coverage of diseases.

**Enrollment Rate:** 0.71

## Prediction

**Best Baseline LSTM:** 0.57

Consider these diseases homogeneously when measuring the match between the doctor and the trial.

**Doctor2Vec:** 0.69

Focus more on the patients who had cancer diagnosis instead of all patients, thus is more accurate.

# Summary: Doctor2Vec

---

- Doctor representation using memory network to encode both dynamic information and static information
- Strong performance in clinical trial recruitment applications

# **GAMENet: Graph Augmented MEmory Networks for Recommending Medication Combination**

Shang, Junyuan, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2018. “GAMENet: Graph Augmented MEmory Networks for Recommending Medication Combination.” AAAI/

# Medication Errors & Adverse Drug-drug Interactions

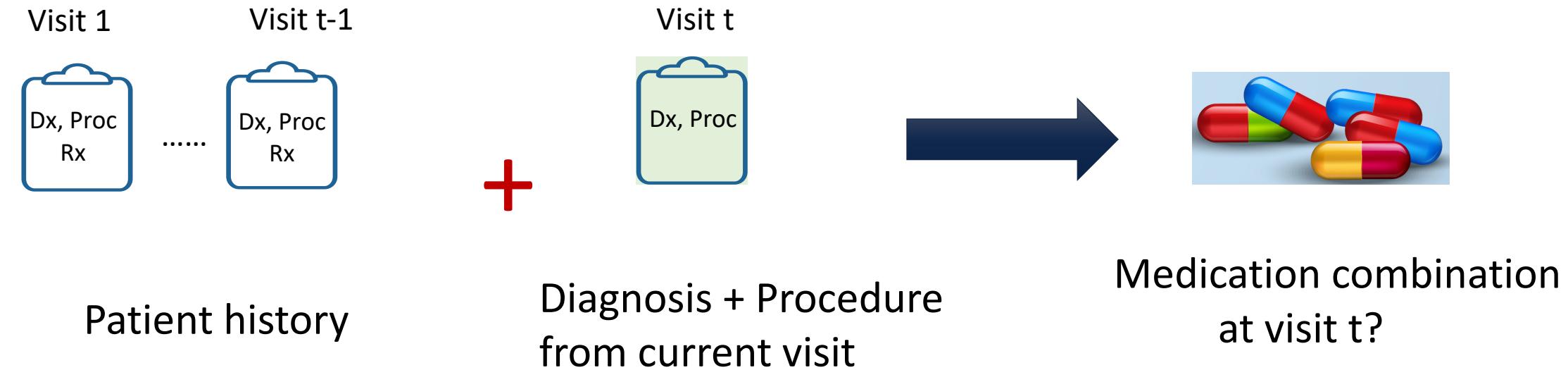


**10 percent** of all U.S. deaths are now due to medical error  
**3rd highest cause** of death in the U.S. is medical error

Adverse drug-drug interactions affects **15 percent** u.s. population.

Cost more than **\$177 billion** per year in disease management

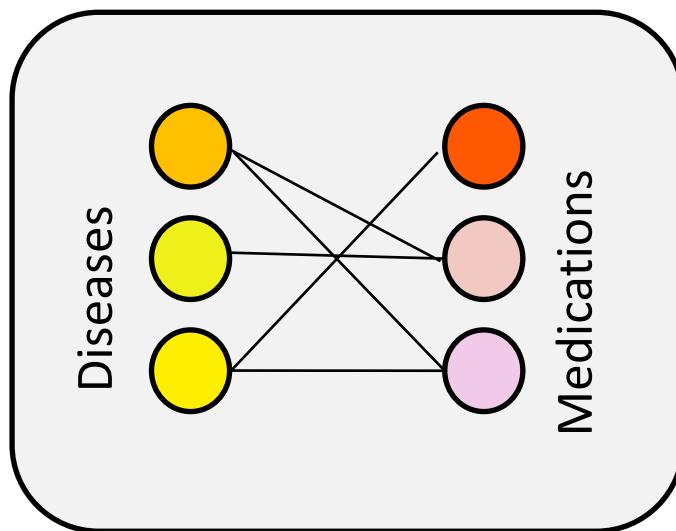
# Task: Recommend Medication Combinations



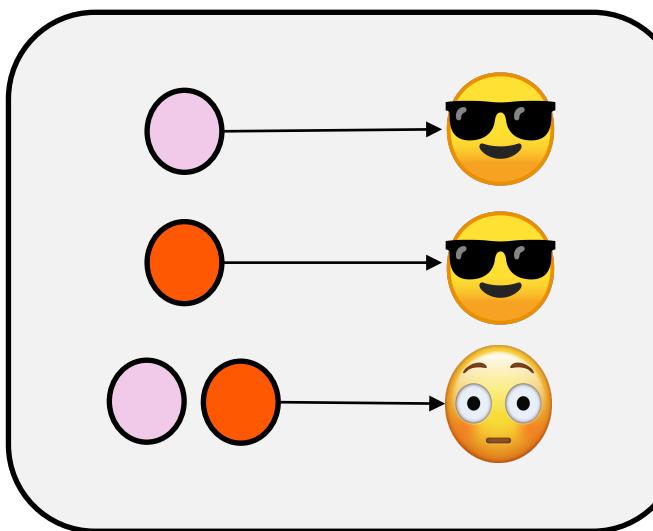
# Challenges for Medication Recommendation



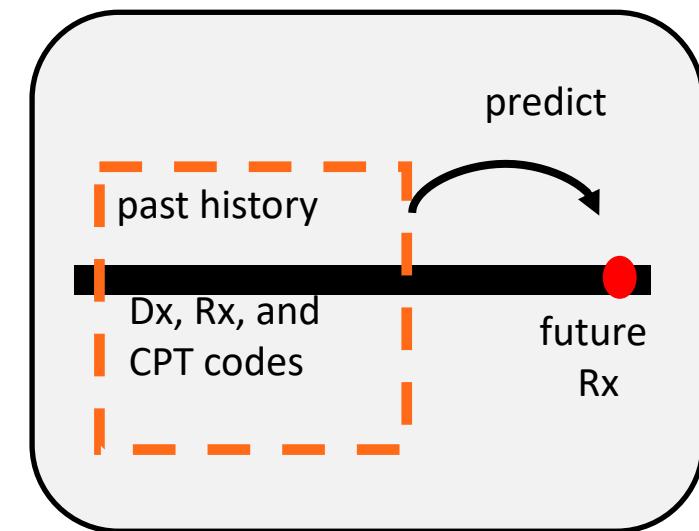
## Complex Dependency



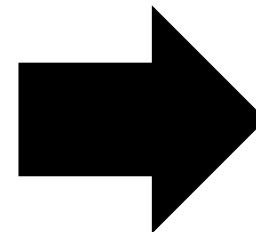
## Drug-drug Interaction



## Patient history



# Graph Augmented Memory Networks (GAMENet)



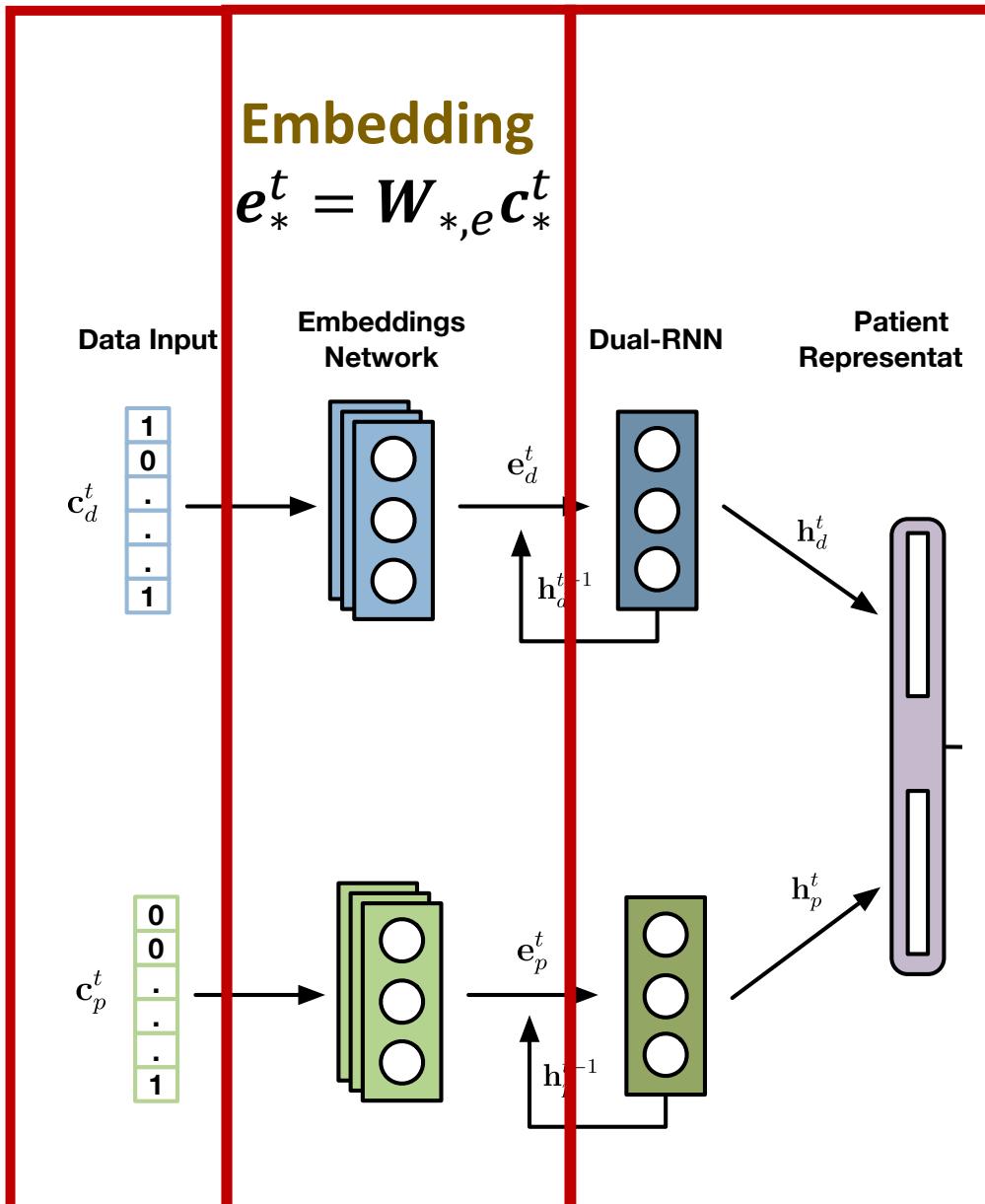
Patient Representation

Graph Augmented  
Memory Network

# Patient Representation Module

INPUT 

Visit codes  $c_*^t$



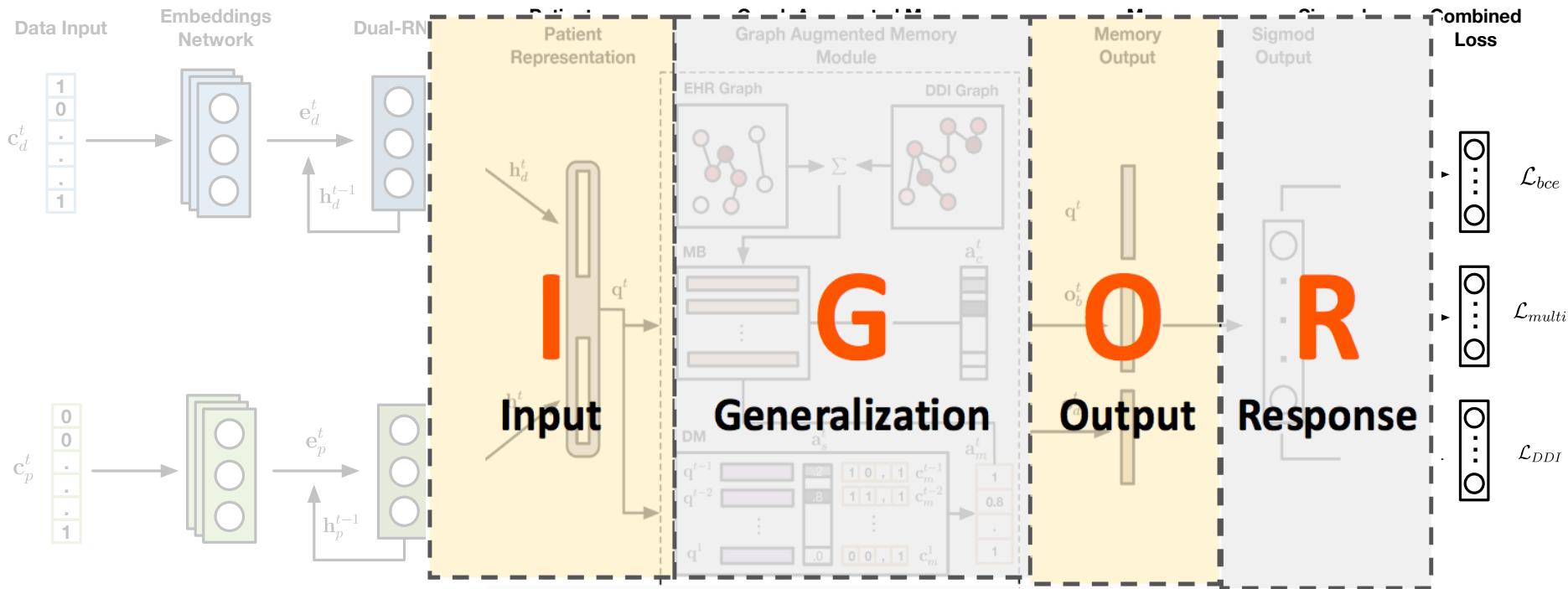
OUTPUT 

Patient Representation  
 $[h_d^t, h_p^t]$

$$h_d^t = RNN_d(e_d^1, \dots, e_d^t) \text{ (diagnosis)}$$

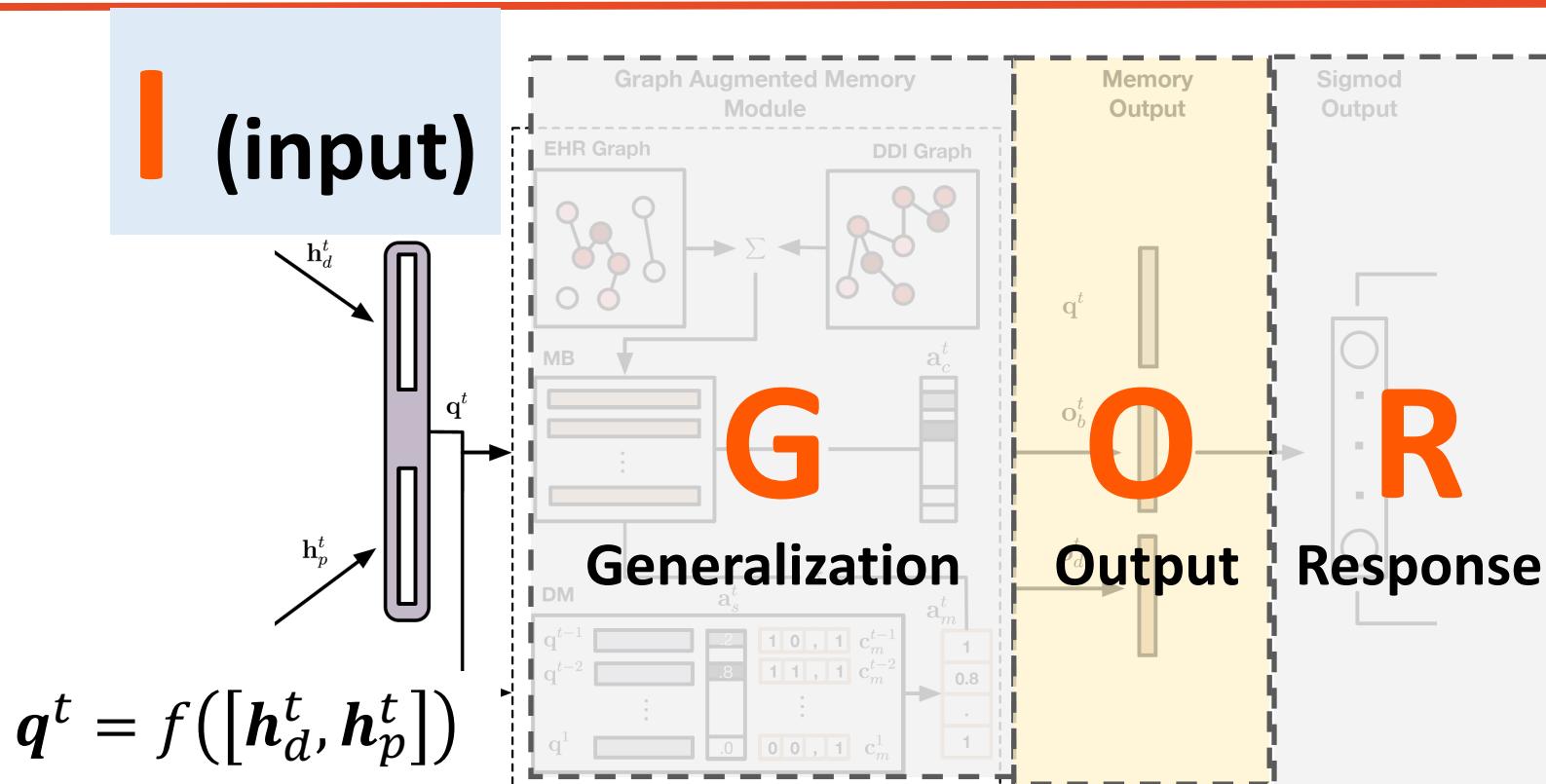
$$h_p^t = RNN_p(e_p^1, \dots, e_p^t) \text{ (procedure)}$$

# Graph Augmented Memory Module (I, G, O, R)



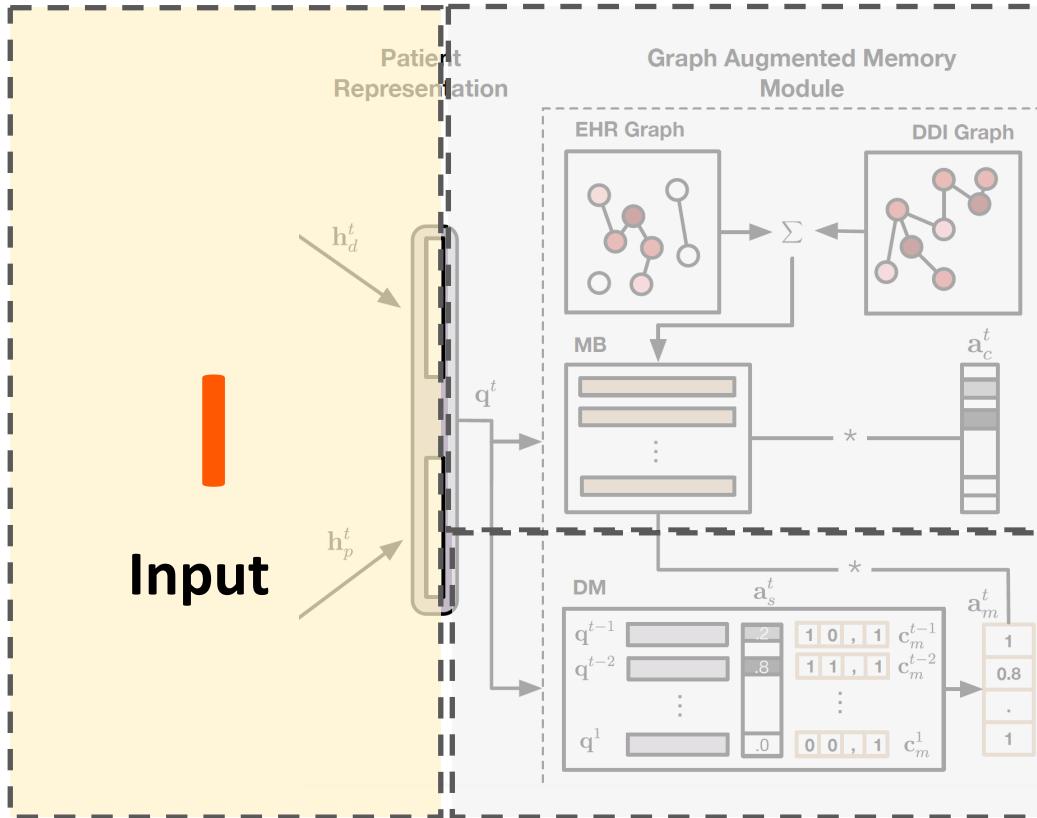
Graph augmented memory network that comprises of memory components **I**, **G**, **O**, **R**.

# Graph Augmented Memory Module (I, G, O, R)



Medical embedding  $h_d^t, h_p^t$  generates patient query  $q^t$ .

# Graph Augmented Memory Module (I, G, O, R)



**G (generalization)**

**Memory Bank (MB)**

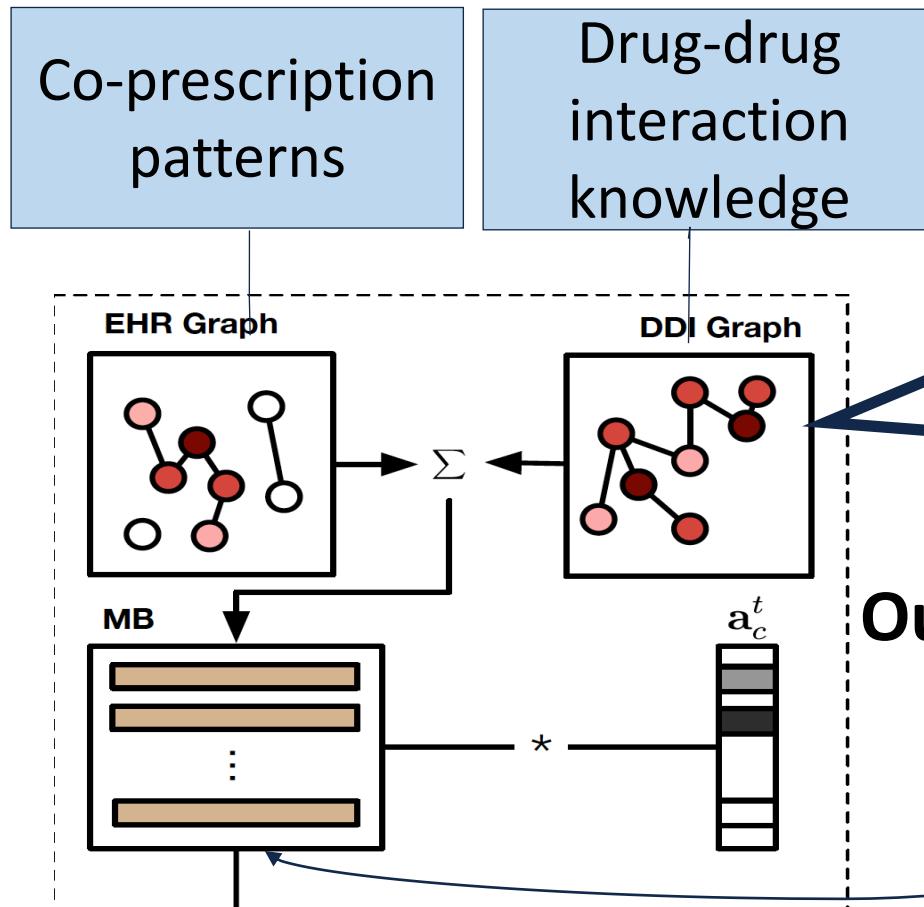
Integrate static knowledge graphs

**Dynamic Memory (DM)**

Incorporate patient medication history

# Static Memory

Memory Bank (MB)



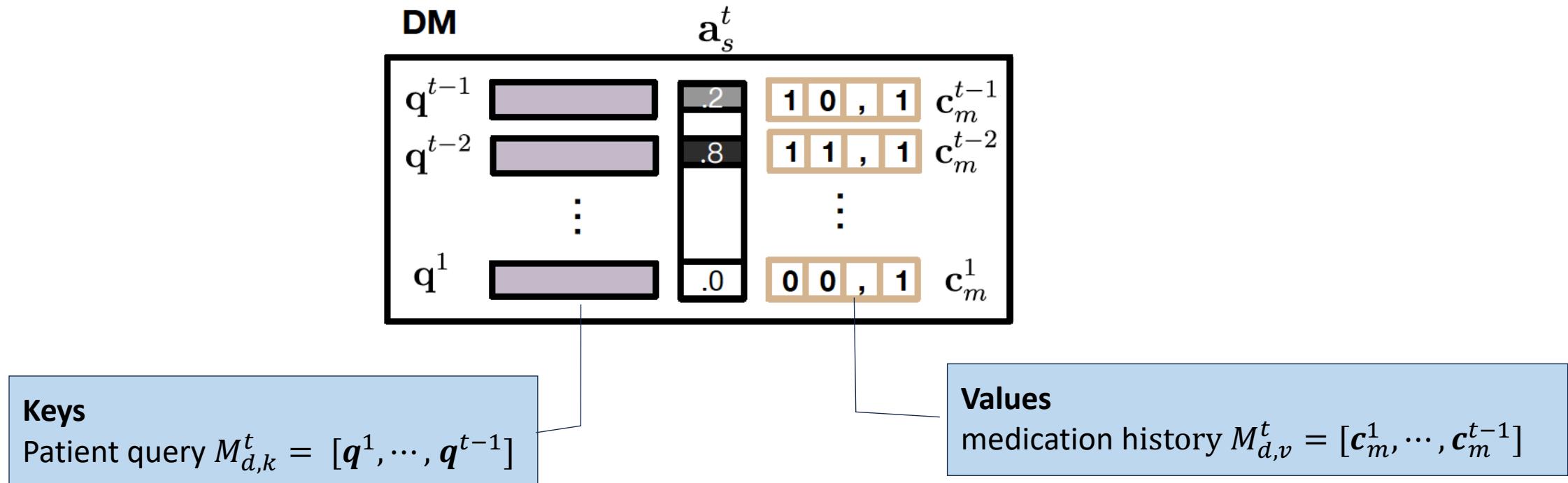
$$\begin{aligned}\tilde{A}_* &= \tilde{D}^{-\frac{1}{2}}(A_* + I)\tilde{D}^{-\frac{1}{2}} \\ Z_1 &= \tilde{A}_* \tanh(\tilde{A}_* W_{e1}) W_1 \\ Z_2 &= \tilde{A}_* \tanh(\tilde{A}_* W_{e2}) W_2 \\ M_b &= Z_1 - \beta Z_2\end{aligned}$$

## Graph Convolutional Networks

Output medication embedding  $M_b$ .

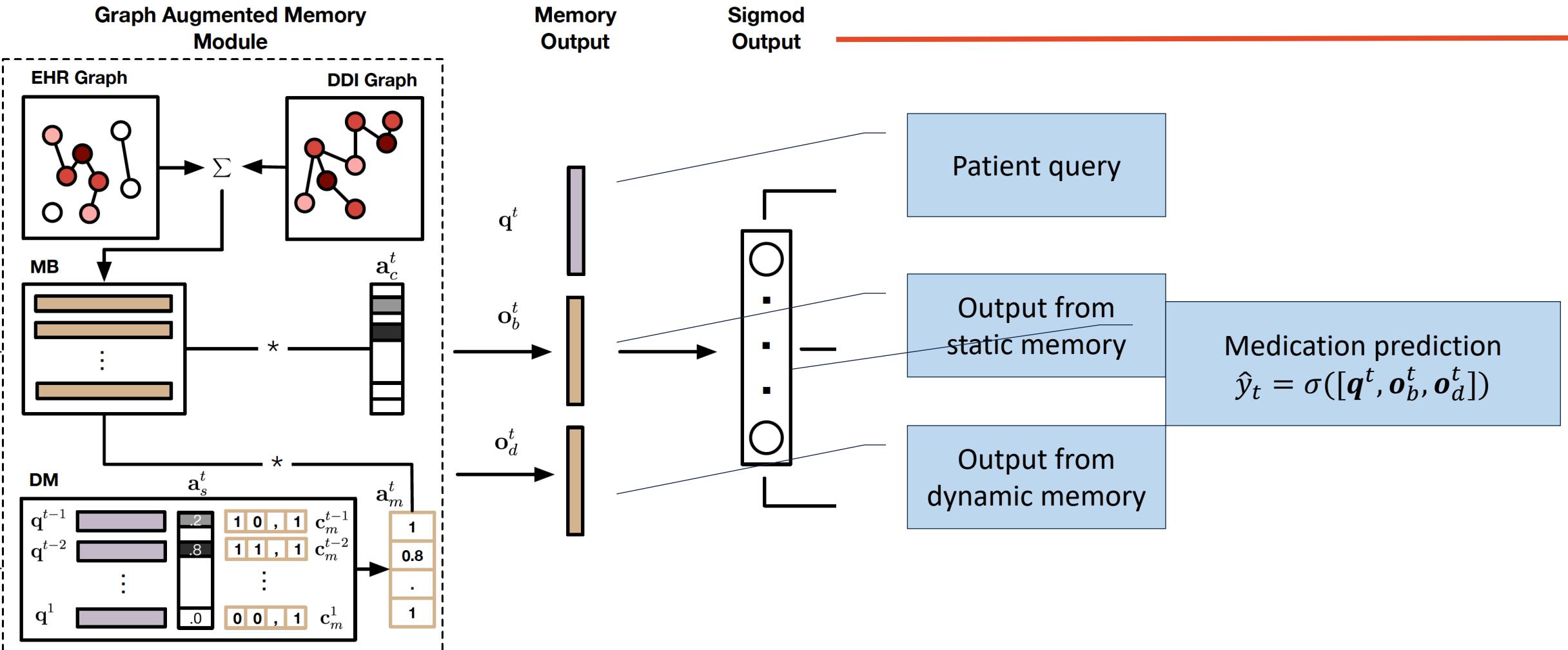
$$\text{Query } o_b^t = M_b^\top \underbrace{\text{Softmax}(M_b q^t)}_{a_c^t}$$

# Dynamic Memory (DM)



$$\text{Query } o_d^t = M_b^\top \underbrace{(M_{d,v}^t)^\top \text{Softmax}(M_{d,k}^t q^t)}_{a_s^t} \overbrace{a_m^t}^{a_m^t}$$

# Output and Response Module (I, G, O, R)



# Experiments

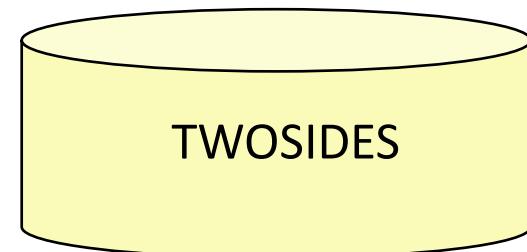


## Patient Record

Table 2: Statistics of the Data

# patients	6,350
# clinical events	15,016
# diagnosis	1,958
# procedure	1,426
# medication	145
avg # of visits	2.36
avg # of diagnosis	10.51
avg # of procedure	3.84
avg # of medication	8.80
# medication in DDI knowledge base	123
# DDI types in knowledge base	40

## Gold-standard DDI Knowledge



Top-40  
severe  
DDIs

*<http://tatonettilab.org/resources>*

- ✓ Patient more than one visit.
- ✓ Medication during the first 24 hours.

**MIMIC-III** *<https://mimic.physionet.org/>*

# Results



Method	DDI rate change	Jaccard	F1
Nearest	+1.80%	0.3911	0.5465
LR	+1.16%	0.4075	0.5658
Leap	-31.53%	0.3844	0.5410
RETAIN	+2.57%	0.4168	0.5781
DMNC	+22.14%	0.4343	0.5934
GAMENet	-3.60%	0.4509	0.6081

Accurate prediction with fewer DDI

# Variants of GAMENet



	Method	DDI rate change	Jaccard	F1	PRAUC
Static memory	DDI only	-4.44%	0.4304	0.5894	0.6736
	EHR only	-1.12%	0.4257	0.5850	0.6665
	Dynamic memory only	4.52%	0.4431	0.6047	0.6891
	GAMENet	-3.60%	0.4509	0.6081	0.6904

# Pre-training of Graph Augmented Transformers for Medication Recommendation

Shang, Junyuan, Tengfei Ma, Cao Xiao, and Jimeng Sun. 2019.  
“Pre-Training of Graph Augmented Transformers for Medication Recommendation.” IJCAI

# Outline

- 
- ✓ Background
  - ✓ Graph Augmented Transformers (G-BERT)
  - ✓ Experiments

# Outline

- 
- ✓ Background
  - ✓ Graph Augmented Transformers (G-BERT)
  - ✓ Experiments

# Background

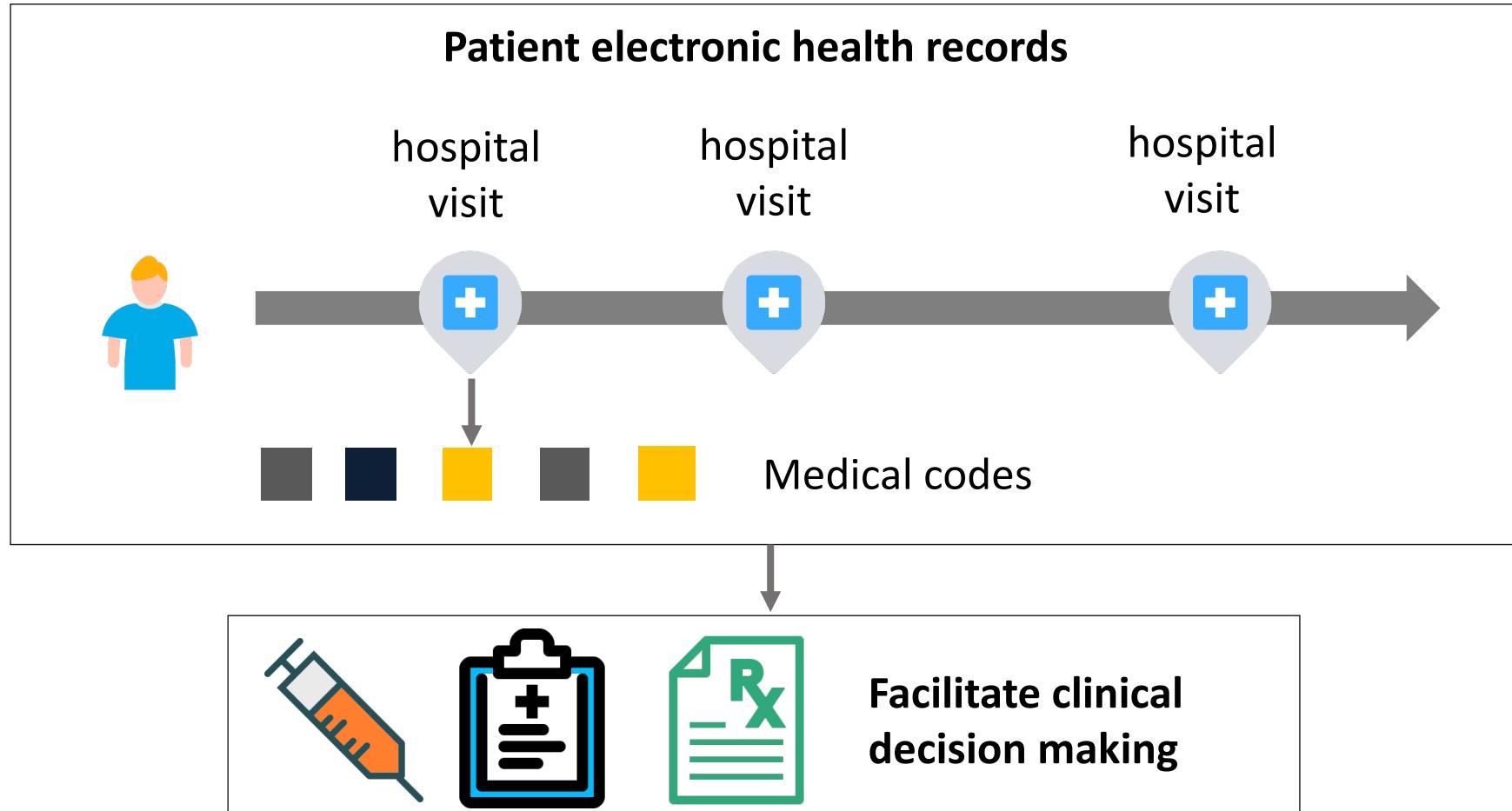


Medical error contributes to 10 percent of all U.S. deaths, and ranks 3<sup>rd</sup> among all causes of death.

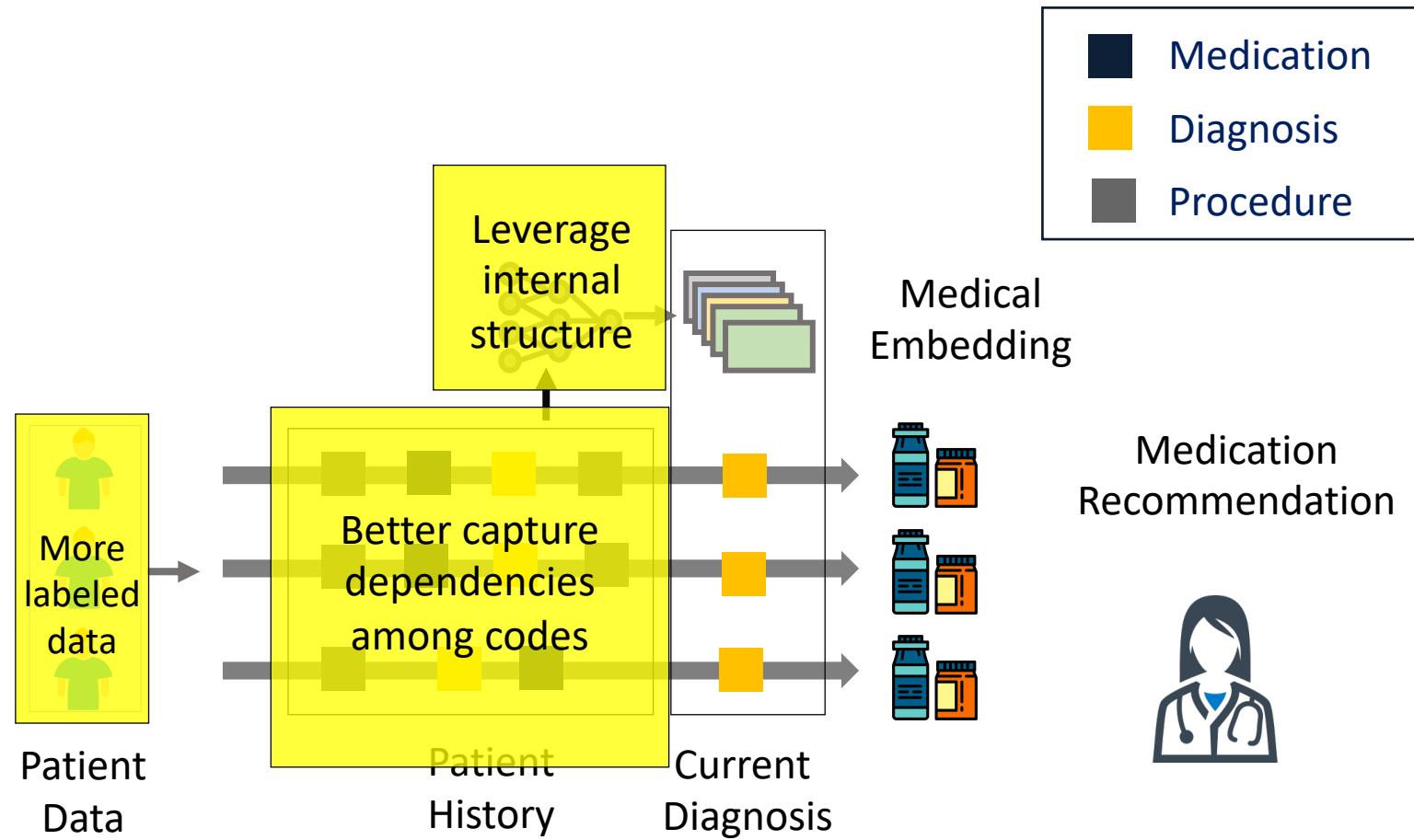


Medical staff shortage problem affect people live in rural areas (~20% of U.S. population).

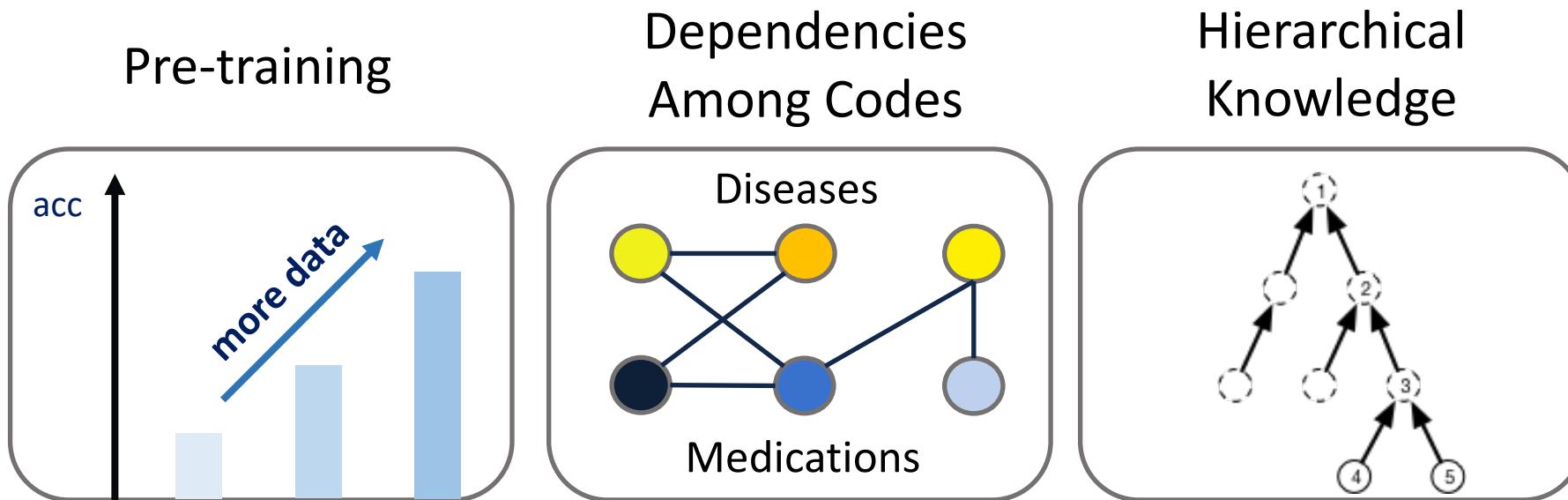
# Background



# Medication Recommendation



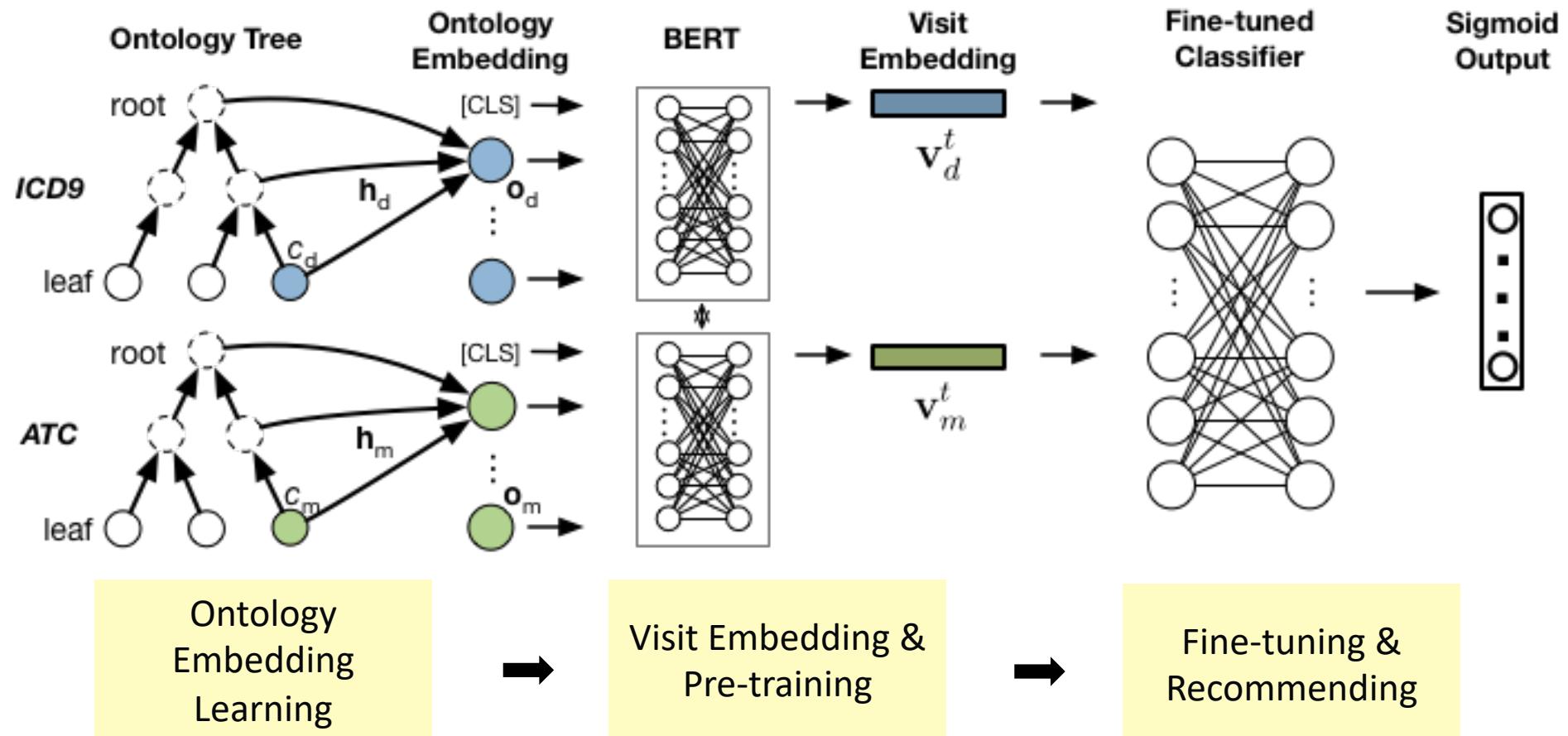
# Our solution - How to improve accuracy



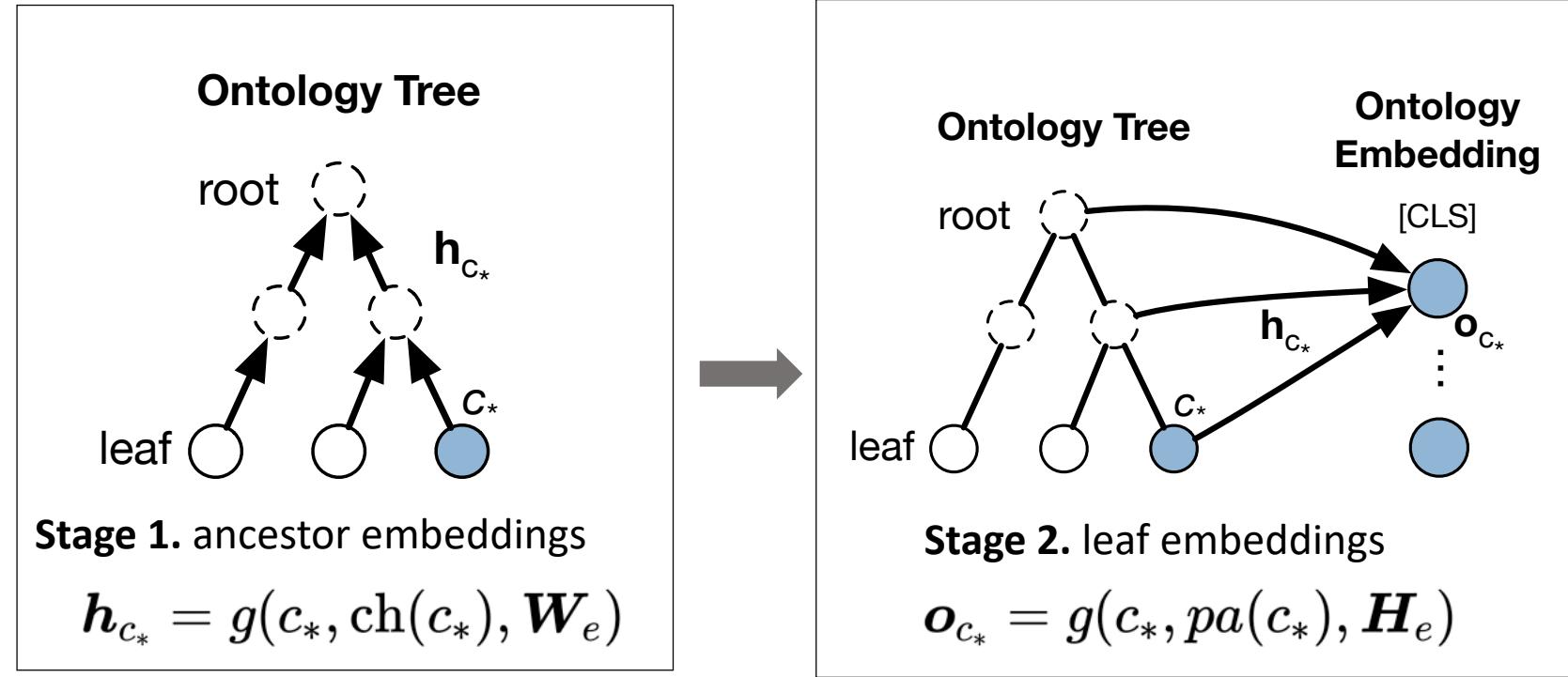
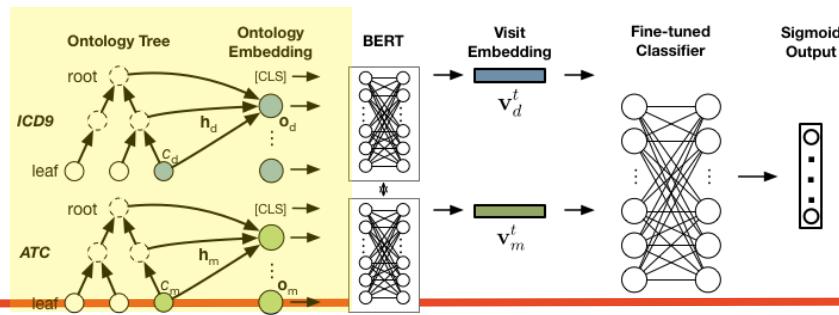
# Outline

- 
- ✓ Background
  - ✓ Graph Augmented Transformers (G-BERT)
  - ✓ Experiments

# Graph Augmented Bidirectional Encoder Representations from Transformers (G-BERT)



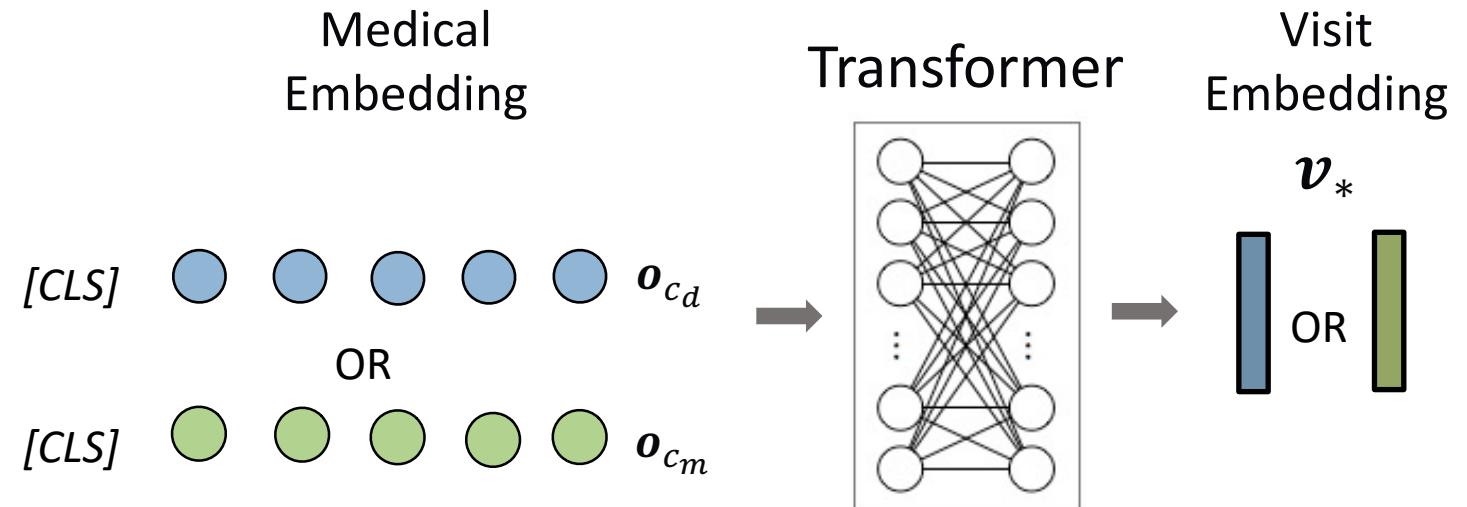
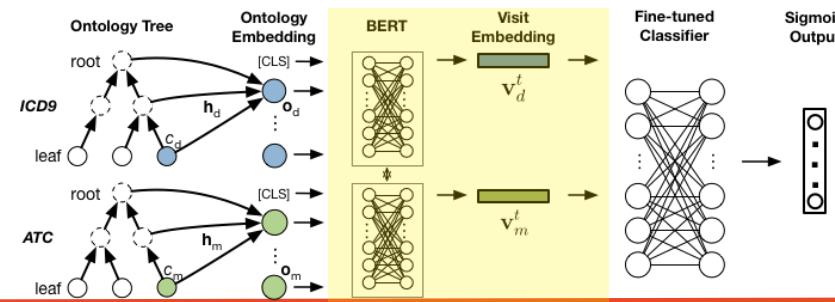
# Ontology Embedding Learning



$$g(c_*, p(c_*), \mathbf{H}_e) = \left\|_{k=1}^K \sigma \left( \sum_{j \in \{c_*\} \cup \text{pa}(c_*)} \alpha_{i,j}^k \mathbf{W}^k \mathbf{h}_j \right) \right\|$$

Implemented as Graph attentional networks (GAT)

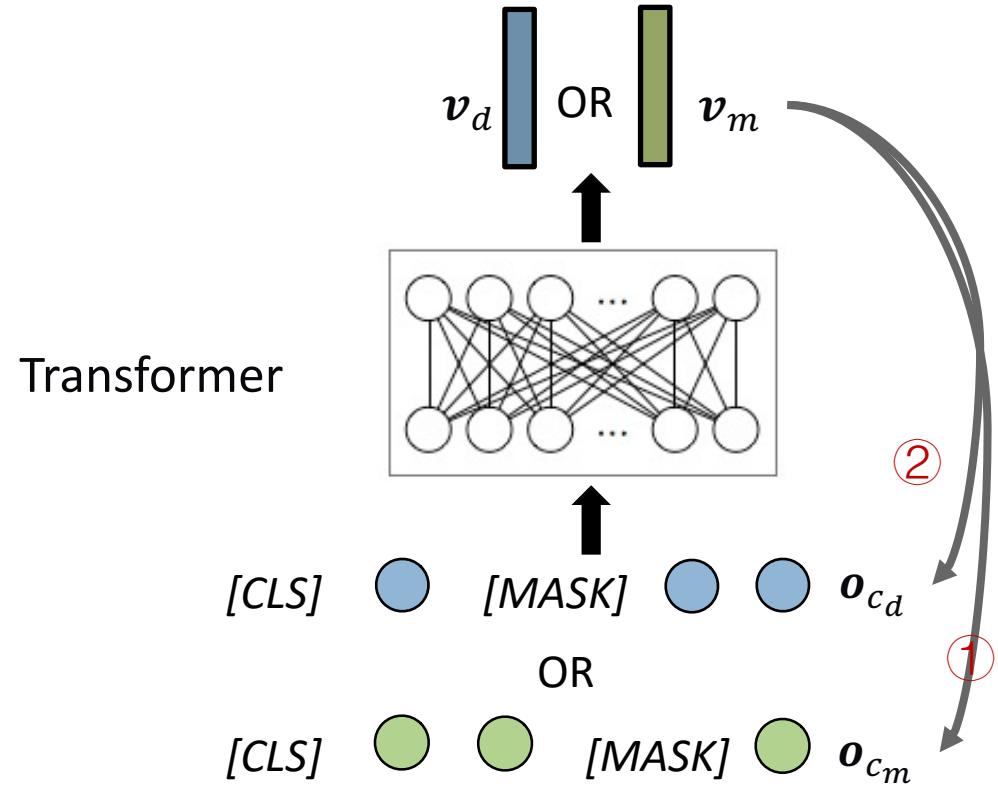
# Visit Embedding



$$\mathbf{v}_*^t = \text{Transformer}(\{[\text{CLS}]\} \cup \{\mathbf{o}_c^t | c_* \in \mathcal{C}_*\})[0]$$

[CLS]: Special token in BERT

# Pre-training



## ① Self-prediction

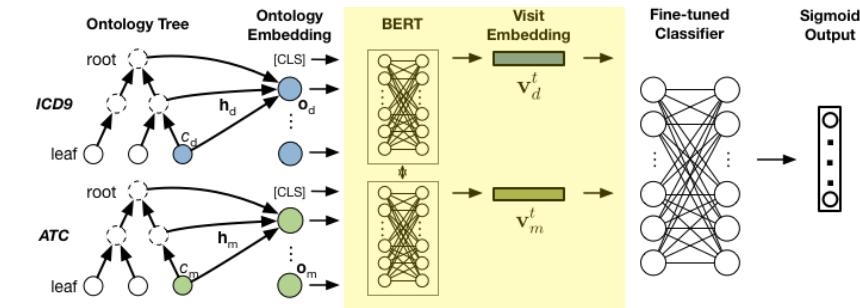
Goal: the visit embedding to recover what it is made of

$$\begin{aligned}\mathcal{L}_{se}(\mathbf{v}_*^1, \mathcal{C}_*^1) &= -\log p(\mathcal{C}_*^1 | \mathbf{v}_*^1) \\ &= -\sum_{c \in \mathcal{C}_*^1} \log p(c_*^1 | \mathbf{v}_*^1) + \sum_{c \in \mathcal{C}_* \setminus \mathcal{C}_*^1} \log p(c_*^1 | \mathbf{v}_*^1)\end{aligned}$$

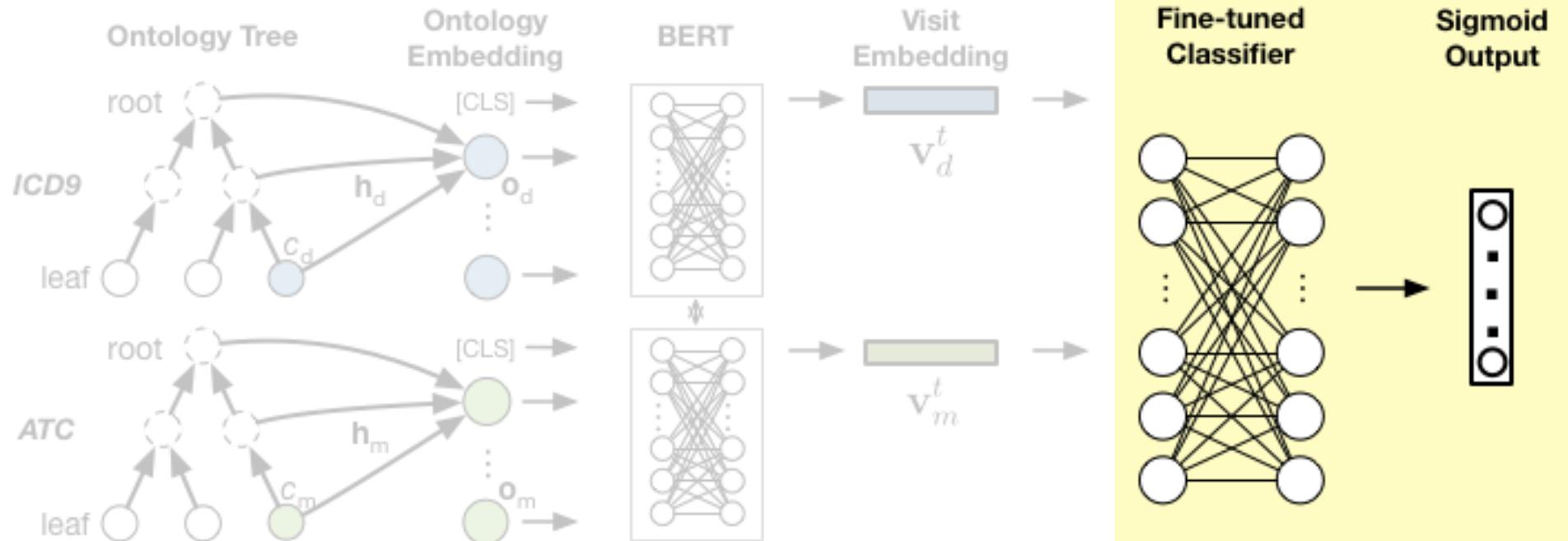
## ② Dual-prediction

Goal: the visit embedding to learn dependencies between codes

$$\mathcal{L}_{du} = -\log p(\mathcal{C}_d^1 | \mathbf{v}_m^1) - \log p(\mathcal{C}_m^1 | \mathbf{v}_d^1)$$



# Fine-tuning & Recommending



Given patient history visit embeddings  $\mathbf{v}_*^\tau (\tau < t)$ , and the latest diagnoses visit embedding  $\mathbf{v}_d^t$ , Output is

$$\hat{y}_t = \sigma(\mathbf{W}_1 \left[ \left( \frac{1}{t-1} \sum \mathbf{v}_d^\tau \right), \left( \frac{1}{t-1} \sum \mathbf{v}_m^\tau \right), \mathbf{v}_d^t \right] + b)$$

# Outline

- 
- ✓ Background
  - ✓ Graph Augmented Transformers (G-BERT)
  - ✓ Experiments

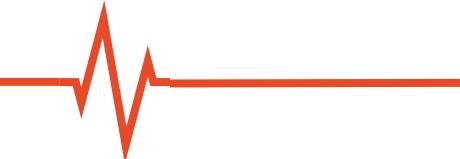
# Datasets



Statistics	1 Visit	>1 Visit
# of patients	30745	6350
Avg # of visits	1	2.36
Avg # of diagnoses	39	10.51
Avg # of medication	52	8.80
# of unique diagnoses	1,997	1,958
# of unique medication	323	145

- ✓ Pre-training: Patient (1 visit) + Patient (>1 visits) in training dataset
- ✓ Diagnoses in ICD-9 form
- ✓ Medication in ATC form

# Results



Methods	Jaccard	PR-AUC	F1	# of parameters
LR	0.4075	0.6716	0.5658	-
GRAM	0.4176	0.6638	0.5788	3,763,668
LEAP	0.3921	0.5855	0.5508	1,488,148
RETAIN	0.4456	0.6838	0.6064	2,054,869
GAMENet	0.4555	0.6854	0.6126	5,518,646
G-BERT	<b>0.4565</b>	<b>0.6960</b>	<b>0.6152</b>	3,034,045

# Summary

---

- An end-to-end deep learning model (G-BERT) that is suitable to learn medical representation and tested on Medication Recommendation task
- Graph Neural Networks can be used to learn enhanced medical ontology embedding
- Pre-training technique has huge potential leverage more data
- Transformer for building dependency between and among instances and labels