GAMENet: Graph Augmented MEmory
Networks for Recommending Medication Combination

Junyuan Shang,[1,2] **Cao Xiao,**[3] **Tengfei Ma,**[3] **Hongyan Li,**[1,2] **Jimeng Sun**[4]

[1] Key Laboratory of Machine Perception, Ministry of Education, Beijing, China
[2] School of EECS, Peking University, Beijing, China   [3] IBM Research
[4] Georgia Institute of Technology

sjy1203@pku.edu.cn, cxiao@us.ibm.com, Tengfei.Ma1@ibm.com, lihy@cis.pku.edu.cn, jsun@cc.gatech.edu

Abstract

Recent progress in deep learning is revolutionizing the healthcare domain including providing solutions to medication recommendations, especially recommending medication combination for patients with complex health conditions. Existing approaches either do not customize based on patient health history, or ignore existing knowledge on drug-drug interactions (DDI) that might lead to adverse outcomes. To fill this gap, we propose the Graph Augmented Memory Networks (GAMENet), which integrates the drug-drug interactions knowledge graph by a memory module implemented as a graph convolutional networks, and models longitudinal patient records as the query. It is trained end-to-end to provide safe and personalized recommendation of medication combination. We demonstrate the effectiveness and safety of GAMENet by comparing with several state-of-the-art methods on real EHR data. GAMENet outperformed all baselines in all effectiveness measures, and also achieved 3.60% DDI rate reduction from existing EHR data.

Introduction

Today abundant health data such as longitudinal electronic health records (EHR) enables researchers and doctors to build better computational models for recommending accurate diagnoses and effective treatments. Medication recommendation algorithms have been developed to assist doctors in making effective and safe medication prescriptions. A series of deep learning methods have been designed for medication recommendation. There are mainly two types of such methods: 1) *Instance-based medication recommendation* models that perform recommendation based only on the current encounter and do not consider the longitudinal patient history, see (Zhang et al. 2017; Wang et al. 2017). As a result, a patient with newly diagnosed hypertension will likely be treated the same as another patient who has suffered chronic uncontrolled hypertension. Such a limitation affects accuracy and utility of the recommendations. 2) *Longitudinal medication recommendation* methods such as (Choi et al. 2016b; Choi et al. 2016a; Lipton et al. 2015; Le, Tran, and Venkatesh 2018) that leverage the temporal dependencies within longitudinal patient history to predict future medication. However, to our best knowledge, none of

them considers drug safety in their modeling, especially ignoring the adverse drug-drug interactions (DDI) which are harder to prevent than single drug adverse reaction. Drugs may interact when they are prescribed and taken together, thus DDIs are common among patients with complex health conditions. Preventing DDIs is important since they could lead to health deterioration or even death.

To fill the gap, we propose Graph Augmented Memory Networks (GAMENet), an end-to-end deep learning model that takes both longitudinal patient EHR data and drug knowledge base on DDIs as inputs and aims to generate effective and safe recommendation of medication combination. In particular, GAMENet consists of 1) patient queries based on representations learned by a dual recurrent neural networks (Dual-RNN), and 2) an integrative and dynamic graph augmented memory module. It builds and fuses across multiple data sources (drug usage information from EHR and DDI knowledge from drug knowledge base (Tatonetti et al. 2012b)) with graph convolutional networks (GCN) (Kipf and Welling 2017) in Memory Bank (**MB**). The knowledge of combined uses of medications and drug-drug interaction relations are thus integrated. It further writes patient history to dynamic memory (**DM**) in key-value form, which mimics case-based retrievals in clinical practice, i.e., considering similar patient representations from the DM. Information from the graph augmented memory module can be retrieved by patient representation as query to generate memory outputs. Then, memory outputs and query will be concatenated to make effective and safe recommendations. GAMENet is optimized to balance between effectiveness and safety by combining multi-label prediction loss from EHR data and DDI loss for DDI knowledge.

To summarize, our work has the following contributions:

- We jointly model the longitudinal patient records as an EHR graph and drug knowledge base as a DDI graph in order to provide effective and safe medication recommendations. This is achieved by optimizing a combined loss that balances between multi-label prediction loss (for effectiveness) and DDI loss (for safety).

- We propose graph augmented memory networks which embed multiple knowledge graphs using a late-fusion mechanism based GCN into the memory component and enable attention-based memory search using query gener-

ated from longitudinal patient records.

- We demonstrate the effectiveness and safety of our model by comparing with several state-of-the-art methods on real EHR data. `GAMENet` outperformed all baselines in effectiveness measures, and achieved 3.60% DDI rate reduction from existing EHR data (i.e., identify and reduce existing DDI cases compared with raw EHR data).

Related Works

Memory Augmented Neural Networks (MANN) have shown initial successes in NLP research areas such as question answering (Weston, Chopra, and Bordes 2015; Sukhbaatar et al. 2015; Miller et al. 2016; Kumar et al. 2016). Memory Networks (Weston, Chopra, and Bordes 2015) and Differentiable Neural Computers (DNC) (Graves et al. 2016) proposed to use external memory components to assist the deep neural networks in remembering and storing things. After that, various MANN based models have been proposed such as (Sukhbaatar et al. 2015; Kumar et al. 2016; Miller et al. 2016). In healthcare, memory networks can be valuable due to their capacities in memorizing medical knowledge and patient history. DMNC (Le, Tran, and Venkatesh 2018) proposed a MANN model for medication combination recommendation task using EHR data alone. In this paper, we use memory component to fuse multi-model graphs as memory bank to facilitate recommendation.

Graph Convolutional Networks (GCN) emerged for inducing informative latent feature representations of nodes from arbitrary graphs (Kipf and Welling 2017; Defferrard, Bresson, and Vandergheynst 2016; Hamilton, Ying, and Leskovec 2017; Chen, Ma, and Xiao 2018). GCN models learn node embeddings in the following manner: Given each graph node initially attached with a feature vector, the embedding vector of each node are the transformed weighted sum of the feature vectors of its neighbors. All nodes are simultaneously updated to perform a layer of forward propagation. The deeper the network, the larger the local neighborhood. Thus global information is disseminated to each graph node for learning better node embeddings. GCNs haven been successfully used to model biomedical n etworks such as drug-drug interaction (DDI) graphs. For example, (Ma et al. 2018) models each drug as a node and DDIs as node labels in the drug association network and extended the GCN to embed multi-view drug features and edges. (Zitnik, Agrawal, and Leskovec 2018) used GCN to model the drug interaction problems by constructing a large two-layer multimodal drug interaction graphs. In this paper, we use GCN to model medication as nodes and DDIs as links.

Medication Combination Recommendation could be categorized into instance-based and longitudinal medication recommendation methods. Instance-based methods focus on current health conditions. Among them, Leap (Zhang et al. 2017) formulates a multi-instance multi-label learning framework and proposes a variant of sequence-to-sequence model based on content-attention mechanism to predict combination of medicines given patient's diagnoses. Longitudinal-based methods leverage the temporal dependencies among clinical events, see (Choi et al. 2016b; Choi et al. 2016a; Lipton et al. 2015; Le, Tran, and Venkatesh 2018; Xiao, Choi, and Sun 2018). Among them, RETAIN (Choi et al. 2016b) is based on a two-level neural attention model which detects influential past visits and significant clinical variables within those visits. DMNC (Le, Tran, and Venkatesh 2018) highlighted the memory component to enhance the memory ability of recurrent neural networks and combined DNC with RNN encoder-decoder to predict medicines based on patient's history records which has shown high accuracy. However, safety issue is often ignored by longitudinal-based methods. In this work, we design a memory component but target at building a structured graph augmented memory, where we not only embed DDI knowledge but also design a DDI loss to reduce DDI rate.

Method

Problem Formulation

Definition 1 (Patient Records) In longitudinal EHR data, each patient can be represented as a sequence of multivariate observations: $\boldsymbol{P}^{(n)} = [\boldsymbol{x}_1^{(n)}, \boldsymbol{x}_2^{(n)}, \cdots, \boldsymbol{x}_{T^{(n)}}^{(n)}]$ where $n \in \{1, 2, \ldots, N\}$, N is the total number of patients; $T^{(n)}$ is the number of visits of the nth patient. To reduce clutter, we will describe the algorithms for a single patient and drop the superscript (n) whenever it is unambiguous. Each visit $\boldsymbol{x}_t = [\boldsymbol{c}_d^t, \boldsymbol{c}_p^t, \boldsymbol{c}_m^t]$ of a patient is concatenation of corresponding diagnoses codes $\boldsymbol{c}_d^t$, procedure codes $\boldsymbol{c}_p^t$ and medications codes $\boldsymbol{c}_m^t$. For simplicity, we use $\boldsymbol{c}_*^t$ to indicate the unified definition for different type of medical codes. $\boldsymbol{c}_*^t \in \{0, 1\}^{|\mathcal{C}_*|}$ is a multi-hot vector, where $\mathcal{C}_*$ denotes the medical code set and $|\mathcal{C}_*|$ the size of the code set.

Definition 2 (EHR&DDI Graph) EHR graph and DDI graph can be denoted as $G_e = \{\mathcal{V}, \mathcal{E}_e\}$ and $G_d = \{\mathcal{V}, \mathcal{E}_d\}$ respectively, where node set $\mathcal{V} = \mathcal{C}_m = \{c_{m_1}, c_{m_2}, \cdots, c_{m_n}\}$ represents the set of medications, $\mathcal{E}_e$ is the edge set of known combination medication in EHR database and $\mathcal{E}_d$ is the edge set of known DDIs between a pair of drugs. Adjacency matrix $\boldsymbol{A}_e, \boldsymbol{A}_d \in \mathbb{R}^{|\mathcal{C}_m| \times |\mathcal{C}_m|}$ are defined to clarify the construction of edge $\mathcal{E}_e, \mathcal{E}_d$. For $\boldsymbol{A}_e$, we firstly create a bipartite graph with drug on one side and drug combination on the other side. Then $\boldsymbol{A}_e = \boldsymbol{A}_b \boldsymbol{A}_b^\mathsf{T}$ where $\boldsymbol{A}_b \in \mathbb{R}^{|\mathcal{C}_m| \times l}$ is the adjacency matrix of the bipartite graph, $\boldsymbol{A}_b[i, j] = 1$ when i^{th} medication exists in j^{th} medications combination and the number of unique medications combination denotes as l. For $\boldsymbol{A}_d$, only pair-wise drug-drug interactions are considered, $\boldsymbol{A}_d[i, j] = 1$ when the ith medication has interaction with the jth one.

Problem 1 (Medication Combination Recommendation) Given medical codes of the current visit at time t (excluding medication codes) $\boldsymbol{c}_d^t, \boldsymbol{c}_p^t$, patient history $\boldsymbol{P} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{t-1}]$ and EHR graph G_e, and DDI graph G_d, we want to recommend multiple medications by generating multi-label output $\hat{\boldsymbol{y}}_t \in \{0, 1\}^{|\mathcal{C}_m|}$.
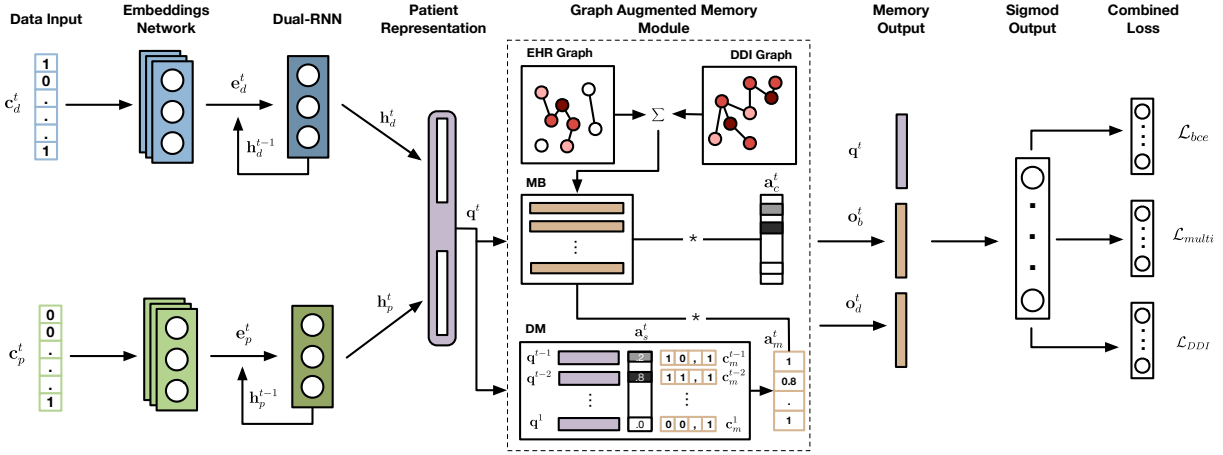
Figure 1: The `GAMENet`: At current t^{th} visit, the multi-hot input c_d^t, c_p^t are input into Embedding Networks to generate embedding e_d^t, e_p^t using Eq. 1. Then Dual-RNN generates current hidden states h_d^t, h_p^t by accepting both embeddings from Embeddings Network and longitudinal hidden state h^{t-1} of RNN denoted by return arrow described in Eq. 2. We use concatenated h_d^t, h_p^t as query q^t (a.k.a. patient representation) in Eq. 3 to output o_b^t by reading from Memory Bank (MB) M_b in Eq. 7 generated from late-fusion based multiple knowledge graph in Eq.4, 5. Meantime, the Dynamic Memory (DM) stores key-value form history information along time by Eq. 6 and can be used to generate o_d^t in Eq. 7. Finally, query and memory outputs are concatenated in Eq. 8 to make recommendation. In training phase, combined loss Eq. 10 is optimized to find optimal model parameters.

The `GAMENet`

As illustrated in Fig. 1, `GAMENet` includes the following components: a medical embedding module, a patient representation module, and a graph augmented memory module. Next we will first introduce these modules and then provide details of training and inference of `GAMENet`.

Medical Embeddings Module   As mentioned before, a visit x_t consists of $[c_d^t, c_p^t, c_m^t]$ where each of c_*^t is a multi-hot vector at the tth visit. The multi-hot vector c_*^t is binary encoded showing the existence of each medical codes recorded at the tth visit. Like (Choi et al. 2016b) used a linear embedding of the input vector, we derive medical embeddings for c_d^t, c_p^t separately at the tth visit as follows:

$$e_*^t = W_{*,e} c_*^t \tag{1}$$

where $W_{*,e} \in \mathbb{R}^{|C_*| \times d}$ is the embedding matrix to learn. Thus a visit x_t is transformed to $\hat{x}_t = [e_d^t, e_p^t, c_m^t]$.

Patient Representation Module   To enable personalized medication recommendation which is tailored using patient EHR data, we design a Dual-RNN to learn patient representations from multimodal EHR data where each RNN encodes only one type of medical codes. The reason is that it is quite possible for a clinical visit to have missing modality (e.g. only diagnosis modality without procedure). Because of that, we model diagnosis and procedure modalities separately using two RNNs. Formally, for each input vector in transformed clinical history $[\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_t]$, we retrieve e_m, e_p and utilize RNN to encode visit-level diagnosis and

procedure embeddings respectively as follows:

$$\begin{aligned} h_d^t &= RNN_d(e_d^1, e_d^2, \cdots, e_d^t) \\ h_p^t &= RNN_p(e_p^1, e_p^2, \cdots, e_p^t) \end{aligned} \tag{2}$$

Thus, the RNNs accept all patient history visit medical embeddings $\{e_*^{t'}\}(t' \leq t)$ to produce hidden states h_*^t for further generating query (a.k.a. patient representation) in Eq. 3.

Graph Augmented Memory Module   To leverage drug knowledge, we construct a graph augmented memory module that not only embeds and stores the EHR graph and the DDI graph as facts in Memory Bank (MB), but also inserts patient history to Dynamic Memory (DM) key-value form to fully capture the information from different views. Inspired by (Weston, Chopra, and Bordes 2015), four memory components **I**, **G**, **O**, **R** are proposed which mimics the architecture of modern computer in some way:

- **I: Input memory representation** converts inputs into query for memory reading. Here we can use hidden states from Dual-RNN to generate query as follows:

$$q^t = f([h_d^t, h_p^t]) \tag{3}$$

  where we concatenate hidden diagnosis state h_d^t and procedure state h_p^t as the input patient health state. $f(\cdot)$ is the transform function which projects hidden states to query and is implemented as single hidden layer fully connected neural network.

- **G: Generalization** is the process of generating and updating the memory representation. We design the memory module by storing graph augmented memory representation as facts in Memory Bank (MB) and insert patient his-

Notation	Description
$\boldsymbol{P} \in \mathbb{R}^{|\mathcal{C}| \times T}$	patient records
$\mathcal{C}_*$	medical codes set of type $*$
$c_{*,i}$	i^{th} medical code in $\mathcal{C}_*$ of type $*$
$\boldsymbol{c}_* \in \mathbb{R}^{|\mathcal{C}_*|}$	multi-hot vector of type $*$
$\boldsymbol{x}_t \in \mathbb{R}^{|\mathcal{C}|}$	concatenation of medical codes $\boldsymbol{c}_*^t$
G_*	EHR or DDI Graph $\{\mathcal{V}, \mathcal{E}_*\}$
$\mathcal{V}$	vertex set same as $\mathcal{C}_m$
$\mathcal{E}_*$	edge set of $*$ dataset
$\boldsymbol{e}_* \in \mathbb{R}^d$	medical embeddings of type $*$
$\boldsymbol{h}_* \in \mathbb{R}^d$	hidden state
$\boldsymbol{q}^t \in \mathbb{R}^d$	query at t^{th} visit
$\boldsymbol{A}_b \in \mathbb{R}^{|\mathcal{C}_m| \times l}$	adjacency matrix of bipartite graph
$\boldsymbol{A}_e \in \mathbb{R}^{|\mathcal{C}_m| \times |\mathcal{C}_m|}$	adjacency matrix of G_e
$\boldsymbol{A}_d \in \mathbb{R}^{|\mathcal{C}_m| \times |\mathcal{C}_m|}$	adjacency matrix of G_d
$\boldsymbol{M}_b \in \mathbb{R}^{|\mathcal{C}_m| \times d}$	Memory Bank (MB)
$\boldsymbol{M}_d^t \in \mathbb{R}^{|t-1| \times (d+|\mathcal{C}_m|)}$	Dynamic Memory (DM)
$\boldsymbol{M}_{d,k}^t \in \mathbb{R}^{|t-1| \times d}$	Keys in DM
$\boldsymbol{M}_{d,v}^t \in \mathbb{R}^{|t-1| \times |\mathcal{C}_m|}$	Values in DM
$\boldsymbol{a}_c^t \in \mathbb{R}^{|\mathcal{C}_m|}$	content-attention weight
$\boldsymbol{a}_s^t \in \mathbb{R}^{|t-1|}$	temporal-attention weight
$\boldsymbol{a}_m^t \in \mathbb{R}^{|\mathcal{C}_m|}$	history medication distribution
$\boldsymbol{o}_*^t \in \mathbb{R}^d$	memory output
$\hat{\boldsymbol{y}}_t \in \mathbb{R}^{|\mathcal{C}_m|}$	multi-label predictions at t^{th} visit
$\hat{Y}$	recommended medication set
Y	ground truth of medication set

Table 1: Notations used in GAMENet

tory to Dynamic Memory (DM) as key-value pairs to fully capture the information from different view.

For Memory Bank (MB) $\boldsymbol{M}_b$, two adjacency matrices $\boldsymbol{A}_e, \boldsymbol{A}_d$ are used. Following the GCN procedure (Kipf and Welling 2017), each $\boldsymbol{A}_*$ is preprocessed as follows:

$$\tilde{\boldsymbol{A}}_* = \tilde{\boldsymbol{D}}^{-\frac{1}{2}}(\boldsymbol{A}_* + \boldsymbol{I})\tilde{\boldsymbol{D}}^{-\frac{1}{2}} \qquad (4)$$

where $\tilde{\boldsymbol{D}}$ is a diagonal matrix such that $\tilde{\boldsymbol{D}}_{ii} = \sum_j \boldsymbol{A}_{ij}$ and $\boldsymbol{I}$ are identity matrices.

Then we applied a two-layer GCN on each graph to learn improved embeddings on drug combination usage and DDIs respectively. The output $\boldsymbol{M}_b$ is generated as a weighted sum of the two graph embeddings.

$$\begin{aligned} \boldsymbol{Z}_1 &= \tilde{\boldsymbol{A}}_e \tanh(\tilde{\boldsymbol{A}}_e \boldsymbol{W}_{e1}) \boldsymbol{W}_1 \\ \boldsymbol{Z}_2 &= \tilde{\boldsymbol{A}}_d \tanh(\tilde{\boldsymbol{A}}_d \boldsymbol{W}_{e2}) \boldsymbol{W}_2 \\ \boldsymbol{M}_b &= \boldsymbol{Z}_1 + \beta \boldsymbol{Z}_2 \end{aligned} \qquad (5)$$

where $\boldsymbol{W}_{e1}, \boldsymbol{W}_{e2} \in \mathbb{R}^{|\mathcal{C}_m| \times d}$ are medication embeddings from EHR graph and DDI graph (each contains $|\mathcal{C}_m|$ number of d-dimensional vectors), $\boldsymbol{W}_1, \boldsymbol{W}_2 \in \mathbb{R}^{d \times d}$ are hidden weight parameter matrices. All $\boldsymbol{W}_*$ are updated during training phase. Then, graph node embeddings $\boldsymbol{Z}_1, \boldsymbol{Z}_2 \in \mathbb{R}^{|\mathcal{C}_m| \times d}$ are generated using GCN. Finally we combine different node embeddings as Memory Bank $\boldsymbol{M}_b \in \mathbb{R}^{|\mathcal{C}_m| \times d}$ where β is a weighting variable to fuse different knowledge graphs.

For Dynamic Memory (DM) $\boldsymbol{M}_d^t$, the combined patient $\{\boldsymbol{q}^{t'}\}(t' < t)$ (the keys) associated with corresponding multi-hot medication vector $\{\boldsymbol{c}_m^{t'}\}$ (the values) are inserted into DM as key-value pairs. This kind of design provides a way to locate most similar patient representation over time and retrieve the proper weighted medications set. Specifically, we can incrementally insert key-value pair after each visit step and treat $\boldsymbol{M}_d^t$ as a vectorized indexable dictionary as follows:

$$\boldsymbol{M}_d^t = \{\boldsymbol{q}^{t'} : \boldsymbol{c}_m^{t'}\}_1^{t-1} \qquad (6)$$

where $\boldsymbol{M}_d^t$ is empty when $t = 1$. For clarity, we use $\boldsymbol{M}_{d,k}^t = [\boldsymbol{q}^1; \boldsymbol{q}^2; \cdots; \boldsymbol{q}^{t-1}] \in \mathbb{R}^{|t-1| \times d}$ to denote the key vectors and $\boldsymbol{M}_{d,v}^t = [\boldsymbol{c}_m^1; \boldsymbol{c}_m^2; \cdots; \boldsymbol{c}_m^{t-1}] \in \mathbb{R}^{|t-1| \times |\mathcal{C}_m|}$ to denote the value vectors at t^{th} visit.

- **O: Output memory representation** produces outputs $\boldsymbol{o}_b^t$ and $\boldsymbol{o}_d^t$ given the patient representation $\boldsymbol{q}^t$ (the query) and the current memory state $\boldsymbol{M}_b, \boldsymbol{M}_d^t$. Here, we apply attention based reading procedure to retrieve most relevant information with respect to query $\boldsymbol{q}^t$ as outputs $\boldsymbol{o}_b^t, \boldsymbol{o}_d^t$ as follows:

$$\begin{aligned} \boldsymbol{o}_b^t &= \boldsymbol{M}_b^\intercal \overbrace{\mathrm{Softmax}(\boldsymbol{M}_b \boldsymbol{q}^t)}^{\boldsymbol{a}_c^t} \\ \boldsymbol{o}_d^t &= \boldsymbol{M}_b^\intercal \overbrace{(\boldsymbol{M}_{d,v}^t)^\intercal \underbrace{\mathrm{Softmax}(\boldsymbol{M}_{d,k}^t \boldsymbol{q}^t)}_{\boldsymbol{a}_s^t}}^{\boldsymbol{a}_m^t} \end{aligned} \qquad (7)$$

where $\boldsymbol{o}_b^t \in \mathbb{R}^d$ is directly retrieved using content-attention $\boldsymbol{a}_c^t$ based on similarity between patient representation (query) and facts in $\boldsymbol{M}_b$.

For $\boldsymbol{o}_d^t \in \mathbb{R}^d$, it firstly considers similar patient representation from patient history records $\boldsymbol{M}_{d,k}^t$ with temporal-attention $\boldsymbol{a}_s^t$. Then $\boldsymbol{a}_s^t$ is utilized to generate history medication distribution $\boldsymbol{a}_m^t$ by weighted sum of history multi-hot medication in $\boldsymbol{M}_{d,v}^t$. Finally, we can get $\boldsymbol{o}_d^t$ by further retrieved information from $\boldsymbol{M}_b$ using $\boldsymbol{a}_m^t$ from temporal aspect.

In addition, the attention based reading procedure makes the model differentiable so that it can be updated end-to-end using back propagation.

- **R: Response** is the final step to utilize patient representation and memory output to predict the multi-label medication as follows:

$$\hat{\boldsymbol{y}}_t = \sigma([\boldsymbol{q}^t, \boldsymbol{o}_b^t, \boldsymbol{o}_d^t]) \qquad (8)$$

where σ is the sigmoid function.

Training and Inference

In the training phase, we need to find the optimal parameters including embedding matrix $\boldsymbol{W}_{e1}, \boldsymbol{W}_{e2}, \boldsymbol{W}_{*,e}$, weight parameter matrix $\boldsymbol{W}_1, \boldsymbol{W}_2$ in GCN, hidden weight in $f(\cdot)$, RNN as **auxiliary model parameter** θ. We introduce

1129

the combined loss in order to find an optimal balance between recommendation accuracy and safety. At the end of the part, training algorithm will be given.

Multi-label Prediction Loss (MLL) Since the medication combination recommendation can be seen as sequential multi-label prediction, we combine two commonly used multi-label loss functions, namely, the binary cross entropy loss $\mathcal{L}_{bce}$ and the multi-label margin loss $\mathcal{L}_{multi}$. We use $\mathcal{L}_{multi}$ since it optimizes to make the predicted probability of ground truth labels has at least 1 margin larger than others. Thus, threshold value in Equation. 11 is easier to be fixed.

$$\mathcal{L}_{bce} = -\sum_t^T \sum_i \boldsymbol{y}_i^t \log \sigma(\hat{\boldsymbol{y}}_i^t) + (1 - \boldsymbol{y}_i^t) \log(1 - \sigma(\hat{\boldsymbol{y}}_i^t))$$
$$\mathcal{L}_{multi} = \sum_t^T \sum_i^{|\mathcal{C}_m|} \sum_j^{|\hat{Y}^t|} \frac{\max(0, 1 - (\hat{\boldsymbol{y}}_t[\hat{Y}_j^t] - \hat{\boldsymbol{y}}_t[i]))}{L}$$
$$\mathcal{L}_p = \pi[0] L_{bce} + \pi[1] L_{multi} \qquad (9)$$

where $\hat{\boldsymbol{y}}[i], \hat{\boldsymbol{y}}_i^t$ means the value at i^{th} coordinate at t^{th} visit, $\hat{\boldsymbol{y}}^t[\hat{Y}_j^t]$ means j^{th} predicted label indexed by predicted label set $\hat{Y}^t$ at t^{th} visit and $\boldsymbol{\pi}[\cdot]$ are the mixture weights ($\pi[0], \pi[1] \geq 0, \pi[0] + \pi[1] = 1$).

DDI Loss (DDI) is designed to control DDIs in the recommendation.

$$\mathcal{L}_{DDI} = \sum_t^T \sum_{i,j} (\boldsymbol{A}_d \odot (\hat{\boldsymbol{y}}_t^\intercal \hat{\boldsymbol{y}}_t))[i, j]$$

where every element in $\hat{\boldsymbol{y}}_t^\intercal \hat{\boldsymbol{y}}_t \in \mathbb{R}^{N \times N}$ gives the pair-wise probability of predicted result. $\odot$ is the element-wise product. Intuitively, for two memory representation i, j, if i j combined to induce a DDI, then $\boldsymbol{A}_d[i, j] = 1$. Thus large pair-wise DDI probability will yield large $\mathcal{L}_{DDI}$.

Combined Loss functions When training, the accuracy and DDI Rate often increase together. The reason is that drug-drug interactions also exist in real EHR data (ground truth medication set Y). Thus both the incorrectly predicted medications and correctly predicted medications may increase the DDI Rate. To achieve the accurate model with low DDI Rate s we need to find the balance between MLL and DDI. Inspired by Simulated Annealing (Kirkpatrick, Gelatt, and Vecchi 1983), we can transform between NRL and MLL with a certain probability as follows:

$$\mathcal{L} = \begin{cases} \mathcal{L}_p & \text{if } s' \leq s \\ \mathcal{L}_{DDI}, \text{with prob. } p = \exp(-\frac{s'-s}{Temp}) & \text{if } s' > s \\ \mathcal{L}_p, \text{with prob. } p = 1 - \exp(-\frac{s'-s}{Temp}) & \text{if } s' > s \end{cases} \qquad (10)$$

on one hand, there will be high probability to use $\mathcal{L}_{DDI}$ when the DDI Rate s' of recommended medication set calculated in this step is larger than the expected DDI Rate s. On the other hand, decay rate ϵ applied on temperature $Temp \leftarrow \epsilon Temp$ makes p low when model becomes stable along training time. Current DDI Rate s' can be calculated using DDI Rate Equation (see Metrics in Experiments section below) without sum across all test samples. The idea to

use combined loss like simulated annealing form helps the model find best combination of parameters to demonstrate effectiveness and safety in the meantime. In inference phase, thank to MLL, if the correctly predicted labels have at least 1 margin larger than others we can fix threshold value as 0.5. Then, the predicted label set corresponds to:

$$\hat{Y}_t = \{\hat{\boldsymbol{y}}_t^j | \hat{\boldsymbol{y}}_t^j > 0.5, 1 \leq j \leq ||\mathcal{C}_m||\}. \qquad (11)$$

The training algorithm is detailed as follows.

Algorithm 1 Training `GAMENet`

Require: Training set $\boldsymbol{R}$, training epoches N, mixture weight $\boldsymbol{\pi}$ in Eq. 9, expected DDI Rate s, initial temperature $Temp$ and weight decay ϵ in Eq. 10;
  Calculate adjacency matrix $\boldsymbol{A}_*$;
  Using uniform distribution to initialize auxiliary model parameters $\theta \sim U(-1, 1)$;
  Obtain Memory Bank $\boldsymbol{M}_b$ using Eq. 4, 5;
  for $i = 1$ to $N * |\boldsymbol{R}|$ **do**
    Sample a patient $\boldsymbol{P} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{T_i}]$ from $\boldsymbol{R}$;
    Reset Dynamic Memory $\boldsymbol{M}_d$;
    for $t = 1$ to T_i **do**
      Obtain medical embeddings $\boldsymbol{e}_d^t, \boldsymbol{e}_p^t$ in Eq. 1;
      Obtain Dual-RNN $\boldsymbol{h}_d^t, \boldsymbol{h}_d^t$ in Eq. 2;
      Generate patient representation $\boldsymbol{q}^t$ in Eq. 3;
      Read from $\boldsymbol{M}_b$ and $\boldsymbol{M}_d^t$ using attention weight $\boldsymbol{a}_c^t, \boldsymbol{a}_s^t, \boldsymbol{a}_m^t$ and generate memory outputs $\boldsymbol{o}_b^t, \boldsymbol{o}_d^t$ in Eq. 7;
      Calculate medication prediction $\hat{\boldsymbol{y}}_t$ using Eq. 8;
      Generate $\boldsymbol{M}_d^{t+1}$ by inserting $(\boldsymbol{q}^t, \boldsymbol{c}_m^t)$ into $\boldsymbol{M}_d^t$ in Eq. 6;
    end for
    Evaluate and obtain DDI Rate s' of current patient;
    Update θ by optimizing loss in Eq. 10 and decay $Temp \leftarrow \epsilon Temp$;
  end for

Experiments

Experimental Setup

We evaluate `GAMENet`[1] model by comparing against other baselines on recommendation accuracy and successful avoidance of DDI. All methods are implemented in PyTorch (Paszke et al. 2017) and trained on an Ubuntu 16.04 with 8GB memory and Nvidia 1080 GPU.

Data Source We used EHR data from MIMIC-III (Johnson et al. 2016). Here we select a cohort where patients have more than one visit. In practice, if we use all the drug codes in an EMR record, the medication set can be very large, each day in hospital, the doctor can prescribe several types of medications for the patient. Hence, we choose the set of medications prescribed by doctors during the first 24-hour as the first 24-hour is often the most critical time for patients to obtain correct treatment quickly. In addition, we used DDI knowledge from TWOSIDES dataset (Tatonetti et al. 2012a). In this work, we keep the Top-40 severity DDI types and transform the drug coding from NDC to ATC Third Level for integrating with MIMIC-III. The statistics of the datasets are summarized in Table 2.

---

[1]https://github.com/sjy1203/GAMENet

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

Baselines   We consider the following baseline algorithms.

- **Nearest** will simply recommend the same combination medications at previous visit for current visit (i.e., $\hat{Y}_t = Y_{t-1}$)

- **Logistic Regression (LR)** is a logistic regression with L2 regularization. Here we represent the input data by sum of one-hot vector. Binary relevance technique (Luaces et al. 2012) is used to handle multi-label output.

- **Leap** (Zhang et al. 2017) is an instance-based medication combination recommendation method.

- **RETAIN** (Choi et al. 2016b) can provide sequential prediction of medication combination based on a two-level neural attention model that detects influential past visits and significant clinical variables within those visits.

- **DMNC** (Le, Tran, and Venkatesh 2018) is a recent work of medication combination prediction via memory augmented neural network based on differentiable neural computers (DNC) (Graves et al. 2016).

Metrics   To measure the prediction accuracy, we used Jaccard Similarity Score (Jaccard), Average F1 (F1) and Precision Recall AUC (PRAUC). Jaccard is defined as the size of the intersection divided by the size of the union of ground truth medications $Y_t^{(k)}$ and predicted medications $\hat{Y}_t^{(k)}$.

$$\text{Jaccard} = \frac{1}{\sum_k^N \sum_t^{T_k} 1} \sum_k^N \sum_t^{T_k} \frac{|Y_t^{(k)} \cap \hat{Y}_t^{(k)}|}{|Y_t^{(k)} \cup \hat{Y}_t^{(k)}|}$$

where N is the number of patients in test set and T_k is the number of visits of the kth patient. Average Precision (Avg-P) and Average Recall (Avg-R), and F1 are defined as:

$$\text{Avg-P}_t^{(k)} = \frac{|Y_t^{(k)} \cap \hat{Y}_t^{(k)}|}{|Y_t^{(k)}|}, \text{Avg-R}_t^{(k)} = \frac{|Y_t^{(k)} \cap \hat{Y}_t^{(k)}|}{|\hat{Y}_t^{(k)}|}$$

$$\text{F1} = \frac{1}{\sum_k^N \sum_t^{T_k} 1} \sum_k^N \sum_t^{T_k} \frac{2 \times \text{Avg-P}_t^{(k)} \times \text{Avg-R}_t^{(k)}}{\text{Avg-P}_t^{(k)} + \text{Avg-R}_t^{(k)}}$$

where t means the tth visit and k means the kth patient in test dataset.

To measure medication safety, we define DDI Rate as percentage of medication recommendation that contain DDIs.

$$\text{DDI Rate} = \frac{\sum_k^N \sum_t^{T_k} \sum_{i,j} |\{(c_i, c_j) \in \hat{Y}_t^{(k)} | (c_i, c_j) \in \mathcal{E}_d\}|}{\sum_k^N \sum_t^{T_k} \sum_{i,j} 1}$$

where the set will count each medication pair (c_i, c_j) in recommendation set $\hat{Y}$ if the pair belongs to edge set $\mathcal{E}_d$ of the DDI graph. Here N is the size of test dataset and T_k is the number of visits of the kth patient.

The relative DDI Rate ($\triangle$ DDI Rate %) is defined as the percentage of DDI rate change compared to DDI rate in EHR test dataset given DDI rate of the algorithm:

$$\triangle \text{DDI Rate \%} = \frac{\text{DDI Rate - DDI Rate (EHR)}}{\text{DDI Rate (EHR)}}$$

Evaluation Strategies   We randomly divide the dataset into training, validation and testing set in a $2/3 : 1/6 : 1/6$ ratio. For LR, we use the grid search technique over typical range of hyper-parameter to search the best hyperparameter values which result in L2 norm penalty with weight as 1.1. For our methods, the hyperparameters are adjusted on evaluation set which result in expected DDI Rate s as 0.05, initial temperature $Temp$ as 0.5, weight decay ϵ as 0.85 and mixture weights $\boldsymbol{\pi} = [0.9, 0.1]$. For all deep learning based methods, we choose a gated recurrent unit (GRU) (Cho et al. 2014) as the implementation of RNN and utilize dropout (Srivastava et al. 2014) with probability of an element to be zeroed as 0.4 on the output of embeddings. The embedding size and hidden layer size for GRU is set as 64 and 64 respectively, word and memory size for DMNC model is 64 and 16 which is the same as (Le, Tran, and Venkatesh 2018). Training is done through Adam (Kingma and Ba 2014) at learning rate 0.0002. We fix the best model on evaluation set within 40 epochs and report the performance in test set.

Results

Table 3 compares the performance on accuracy and safety issue. Results show GAMENet has the highest score among all baselines with respect to Jaccard, PR-AUC, and F1.

As for the baseline models, Nearest and LR achieved about 4% lower score compared to GAMENet in terms of Jaccard and F1. The Nearest method also gives us the clue that the visit is highly important for the medications combination recommendation task. For both methods, the DDI rates are very close to the base DDI rate in the EHR data. This implies without knowledge guidance that it will be hard to remove DDIs that already exist in clinical practice. For deep learning baselines, instance-based method Leap achieved lower performance than those temporal models such as RETAIN and DMNC, which confirmed the important of temporal information in patient past EHRs.

On the other hand, for longitudinal methods such as RETAIN and DMNC, they both achieve higher scores on Jaccard, PRAUC, and F1 compared with others. DMNC however recommends a large bunch of medication combination set which may be one reason that lead to high DDI Rate.

For our methods, we compare the GAMENet and its variant GAMENet (w/o DDI). Without DDI knowledge,

Table 3: Performance Comparison of Different Methods. Note that the base DDI rate in EHR test data is 0.0777.

Methods	DDI Rate	△ DDI Rate %	Jaccard	PR-AUC	F1	Avg # of Med.	# of parameters
Nearest	0.0791	+ 1.80%	0.3911	0.3805	0.5465	14.77	-
LR	0.0786	+ 1.16%	0.4075	0.6716	0.5658	11.42	-
Leap	**0.0532**	**− 31.53%**	0.3844	0.5501	0.5410	14.42	436,884
RETAIN	0.0797	+ 2.57%	0.4168	0.6620	0.5781	16.68	289,490
DMNC	0.0949	+ 22.14%	0.4343	0.6856	0.5934	20.00	527,979
GAMENet (w/o DDI)	0.0853	+ 9.78%	0.4484	0.6878	0.6059	15.13	452,434
GAMENet	**0.0749**	**− 3.60%**	**0.4509**	**0.6904**	**0.6081**	14.02	452,434

Table 4: Example Recommended Medications for a Patient with Two Visits. Here "unseen" indicates the medications predicted but are not in ground truth, while "missed" refers to the medications that are in the ground truth but are not predicted.

Diagnosis	Methods	Recommended Medication Combination
1st Visit: Malignant neoplasm of brain Hyperlipidmia Gout	Ground Truth	N02B, A01A, A02B, A06A, B05C, A12C, C07A, C02D, N02A, B01A, C10A, J01D, N03A, A04A, H04A
	Nearest	0 correct + 15 missed
	LR	3 correct (N02B, A01A, A06A) + 12 missed
	Leap	8 correct (N02B, A02B, A06A, A12C, C07A, B01A, C10A, A04A) + 7 missed
	RETAIN	0 correct + 15 missed
	DMNC	12 correct (N02B, A01A, A02B, A06A, B05C, A12C, C07A, C02D, N02A, B01A, C10A, N03A) + 6 unseen + 3 missed
	GAMENet	11 correct (N02B, A01A, A02B, A06A, B05C, A12C, C07A, C02D, B01A, N03A, A04A) + 4 missed
2nd Visit: Malignant neoplasm of brain Cerebral Edema Hypercholesterolemia Gout	Ground Truth	N02B, A01A, A02B, A06A, B05C, A12C, C07A, C02D, N02A, B01A, J01D, N03A, N05A, A04A
	Nearest	13 correct (N02B, A01A, A02B, A06A, B05C, A12C, C07A, C02D, N02A, B01A, J01D, N03A, A04A) + 2 unseen + 1 missed
	LR	3 correct (N02B, A01A, A06A) + 11 missed
	Leap	7 correct (N02B, A01A, A02B, A06A, B05C, A12C, B01A) + 2 unseen + 7 missed
	RETAIN	10 correct (N02B, A01A, A02B, A06A, B05C, A12C, C07A, N02A, B01A, N03A) + 5 unseen + 4 missed
	DMNC	12 correct (N02B, A01A, A02B, A06A, B05C, A12C, C07A, C02D, N02A, B01A, N03A, A04A) + 7 unseen + 2 missed
	GAMENet	13 correct (N02B, A01A, A02B, A06A, B05C, A12C, C07A, C02D, N02A, B01A, J01D, N03A, A04A) + 1 unseen + 1 missed

GAMENet (w/o DDI) is also better than other methods which shows the overall framework does work. With DDI knowledge, both the performance and DDI rate are improved. The result is statistically significant using two-tailed t-test after ten runs of these two methods.

Case Study

We choose a patient from test dataset based on the consideration of demonstrating the model effect on harder cases: there are diagnoses and medications change among visits. As shown in Table. 4, the patient has 3 diagnoses for the 1st visit and two extra diagnoses Cerabral Edema, Hypercholesterolemia for the 2nd visit. The ground truth medications prescribed by doctors and recommended medications by different methods are listed in the table. Overall, GAMENet performs the best with 11 correct, 13 correct medications for two visit respectively, only missed 4 and 1 medications and wrongly predict 1 (unseen) medication for 2nd visit. For Nearest and RETAIN methods, they lack the ability to recommend medication combination for 1st visit. DMNC tries to recommend more medications which result in more wrongly predicted medications than other methods. To mention that, all methods except LR and GAMENet will recommend the combination of N02B (Analgesics and Antipyretics) and C10A (Lipid-modifying Agents), which can lead to harmful side effect such as Myoma. This harmful combination also existed in ground truth of the patient's at 1st visit. For the 2nd visit, C10A is removed from ground truth medications set, which may indicate doctors also try to correct their decision. Another pair of medications A01A (Stomatological Preparations) and N03A (Antiepileptic Drugs) exists in the ground truth of both visits. Their combined use could cause allergic bronchitis. Most methods including Nearest, RETAIN, DMNC recommend them. GAMENet also recommends them due to the trade off between effectiveness and safety.

Conclusion

In this work, we presented GAMENet, an end-to-end deep learning model that aims to generate effective and safe recommendations of medication combinations via memory networks whose memory bank is augmented by integrated drug usage and DDI graphs as well as dynamic memory based on patient history. Experimental results on real-world EHR showed that GAMENet outperformed all baselines in effectiveness measures, and achieved 3.60% DDI rate reduction from existing EHR data. As we noticed the trade-off between effectiveness and safety measures, a possibly rewarding avenue of future research is to simultaneous recommend medication replacements that share the same indications of the harmful drugs but will not induce adverse DDIs.

Acknowledgment

References

Chen, J.; Ma, T.; and Xiao, C. 2018. FastGCN: Fast learning with graph convolutional networks via importance sampling. In *International Conference on Learning Representations*.

Cho, K.; Van Merriënboer, B.; Bahdanau, D.; and Bengio, Y. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Choi, E.; Bahadori, M. T.; Schuetz, A.; Stewart, W. F.; and Sun, J. 2016a. Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference*, 301–318.

Choi, E.; Bahadori, M. T.; Sun, J.; Kulas, J.; Schuetz, A.; and Stewart, W. 2016b. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, 3504–3512.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *CoRR* abs/1606.09375.

Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Danihelka, I.; Grabska-Barwińska, A.; Colmenarejo, S. G.; Grefenstette, E.; Ramalho, T.; Agapiou, J.; et al. 2016. Hybrid computing using a neural network with dynamic external memory. *Nature* 538(7626):471.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *CoRR* abs/1706.02216.

Johnson, A. E.; Pollard, T. J.; Shen, L.; Li-wei, H. L.; Feng, M.; Ghassemi, M.; Moody, B.; Szolovits, P.; Celi, L. A.; and Mark, R. G. 2016. Mimic-iii, a freely accessible critical care database. *Scientific data* 3:160035.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kirkpatrick, S.; Gelatt, C. D.; and Vecchi, M. P. 1983. Optimization by simulated annealing. *science* 220(4598):671–680.

Kumar, A.; Irsoy, O.; Ondruska, P.; Iyyer, M.; Bradbury, J.; Gulrajani, I.; Zhong, V.; Paulus, R.; and Socher, R. 2016. Ask me anything: Dynamic memory networks for natural language processing. In *International Conference on Machine Learning*, 1378–1387.

Le, H.; Tran, T.; and Venkatesh, S. 2018. Dual memory neural computer for asynchronous two-view sequential learning. In *Proceedings of the 24rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1637–1645. ACM.

Lipton, Z. C.; Kale, D. C.; Elkan, C.; and Wetzel, R. 2015. Learning to diagnose with lstm recurrent neural networks. *arXiv preprint arXiv:1511.03677*.

Luaces, O.; Díez, J.; Barranquero, J.; del Coz, J. J.; and Bahamonde, A. 2012. Binary relevance efficacy for multilabel classification. *Progress in Artificial Intelligence* 1(4):303–313.

Ma, T.; Xiao, C.; Zhou, J.; and Wang, F. 2018. Drug similarity integration through attentive multi-view graph autoencoders. *CoRR* abs/1804.10850.

Miller, A.; Fisch, A.; Dodge, J.; Karimi, A.-H.; Bordes, A.; and Weston, J. 2016. Key-value memory networks for directly reading documents. In *Empirical Methods in Natural Language Processing*, 1400–1409.

Paszke, A.; Gross, S.; Chintala, S.; Chanan, G.; Yang, E.; DeVito, Z.; Lin, Z.; Desmaison, A.; Antiga, L.; and Lerer, A. 2017. Automatic differentiation in pytorch.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research* 15(1):1929–1958.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, 2440–2448.

Tatonetti, N.; Patrick, P.; Daneshjou, R.; and Altman, R. 2012a. Data-driven prediction of drug effects and interactions. *Science translational medicine* 4(125):125ra31–125ra31.

Tatonetti, N.; Ye, P.; Daneshjou, R.; and Altman, R. 2012b. Data-driven prediction of drug effects and interactions. *Science Translational Medicine* 4(125).

Wang, M.; Liu, M.; Liu, J.; Wang, S.; Long, G.; and Qian, B. 2017. Safe medicine recommendation via medical knowledge graph embedding. *arXiv preprint arXiv:1710.05980*.

Weston, J.; Chopra, S.; and Bordes, A. 2015. Memory networks. In *International Conference on Learning Representations*.

Xiao, C.; Choi, E.; and Sun, J. 2018. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*.

Zhang, Y.; Chen, R.; Tang, J.; Stewart, W. F.; and Sun, J. 2017. Leap: Learning to prescribe effective and safe treatment combinations for multimorbidity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1315–1324. ACM.

Zitnik, M.; Agrawal, M.; and Leskovec, J. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34(13):457–466.