Order-free Medicine Combination Prediction with Graph Convolutional Reinforcement Learning

Shanshan Wang
Shandong University
wangshanshan5678@gmail.com

Pengjie Ren
University of Amsterdam
p.ren@uva.nl

Zhumin Chen
Shandong University
chenzhumin@sdu.edu.cn

Zhaochun Ren
Shandong University
zhaochun.ren@sdu.edu.cn

Jun Ma
Shandong University
majun@sdu.edu.cn

Maarten de Rijke
University of Amsterdam
derijke@uva.nl

ABSTRACT

Medicine Combination Prediction (MCP) based on Electronic Health Record (EHR) can assist doctors to prescribe medicines for complex patients. Previous studies on MCP either ignore the correlations between medicines (i.e., MCP is formulated as a binary classification task), or assume that there is a sequential correlation between medicines (i.e., MCP is formulated as a sequence prediction task). The latter is unreasonable because the correlations between medicines should be considered in an order-free way. Importantly, MCP must take additional medical knowledge (e.g., Drug-Drug Interaction (DDI)) into consideration to ensure the safety of medicine combinations. However, most previous methods for MCP incorporate DDI knowledge with a post-processing scheme, which might undermine the integrity of proposed medicine combinations.

In this paper, we propose a graph convolutional reinforcement learning model for MCP, named Combined Order-free Medicine Prediction Network (CompNet), that addresses the issues listed above. CompNet casts the MCP task as an order-free Markov Decision Process (MDP) problem and designs a Deep Q Learning (DQL) mechanism to learn correlative and adverse interactions between medicines. Specifically, we first use a Dual Convolutional Neural Network (Dual-CNN) to obtain patient representations based on EHRs. Then, we introduce the medicine knowledge associated with predicted medicines to create a dynamic medicine knowledge graph, and use a Relational Graph Convolutional Network (R-GCN) to encode it. Finally, CompNet selects medicines by fusing the combination of patient information and the medicine knowledge graph. Experiments on a benchmark dataset, i.e., MIMIC-III, demonstrate that CompNet significantly outperforms state-of-the-art methods and improves a recently proposed model by 3.74%pt, 6.64%pt in terms of Jaccard and F1 metrics.

CCS CONCEPTS

• **Applied computing → Health care information systems**;

KEYWORDS

Medicine combination prediction, Medicine knowledge graph, Reinforcement learning, Relational graph convolutional network

ACM Reference Format:
Shanshan Wang, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. Order-free Medicine Combination Prediction with Graph Convolutional Reinforcement Learning. In *The 28th ACM International Conference on Information and Knowledge Management (CIKM '19), November 3–7, 2019, Beijing, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3357384.3357965

1 INTRODUCTION

Recently, deep learning techniques have impacted various Clinical Decision Support (CDS) applications, including, e.g., disease diagnosis [26, 44, 46], and mortality prediction [13, 49]. While medical professionals will not be replaced by AI systems in the foreseeable future, they have already benefited a great deal from the assistance of AI systems [8]. *Medicine Combination Prediction* (MCP) plays an important role in treating multiple complex diseases [5]. Given a patient's health condition, the MCP task is to predict a combination of medicines that can assist doctors to offer increased therapeutic efficacy and reduced toxicity [42].

Some previous work casts MCP as a multi-label binary classification problem, e.g., [2]. However, these approaches assume that the medicines involved are independent and ignore correlations between medicines [23, 40]. This is unreasonable because medicine correlations are common in reality, e.g., in the treatment of "Helicobacter pylori infection," if a patient takes "proton pump inhibitors," then there is a high probability that "antibiotics" and "pectin" will be taken together. Li et al. [23] model MCP as a sequence prediction problem, where they predict a medicine at each timestamp. The medicine predicted at the current step is affected by previously predicted medicines. In this way, these methods can capture sequential correlations between medicines to some extent. However, these methods assume that there are orders among medicines. During training, they need to pre-set the orders for medicines, and different medicine orders may produce different results [48].

Unlike conventional decision making tasks, there is an additional challenge for MCP, i.e., how to effectively introduce medical knowledge to leverage correlative medicine interactions while avoiding adverse medicine combinations. In reality, when prescribing medicines for patients, doctors must fully consider patients' conditions and the interactions between different medicines, especially adverse interactions [42]. For example, the combination of

CIKM '19, November 3–7, 2019, Beijing, China

ACM ISBN 978-1-4503-6976-3/19/11…$15.00
https://doi.org/10.1145/3357384.3357965

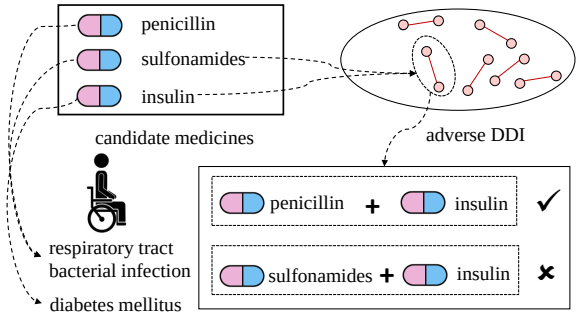

Figure 1: Complex medical relationships in Medicine Combination Prediction (MCP).

"sulfonamides"and "insulin" can cause severe hypoglycemia. Adverse medicine combinations can be avoided with a simple and intuitive post-processing approach, i.e., removing adverse medicine combinations from the prediction results based on a *Drug-Drug Interaction* (DDI) knowledge base. For example, Zhang et al. [48] first use Multi-Instance Multi-Label Learning (MIML) to train a MCP model, and then they employ reinforcement learning to fine-tune the model's parameters based on a DDI knowledge base. However, this post-processing approach undermines the integrity of the medicine combination [48]. In other words, post-prediction fine-tuning will influence the optimal parameters learned in the prediction procedure. For example, as shown in Fig. 1, adverse DDI exists between "sulfonamides" and "insulin". If "sulfonamides" is removed, the disease "respiratory tract bacterial infection" will not be treated. Similarly, if "insulin" is removed, the disease "diabetes mellitus" will not be treated.

Recently, Shang et al. [34] have proposed to aggregate the Electronic Health Record (EHR) medicine graph, the DDI medicine graph, and longitudinal patient records to predict medicines. However, they use all of the knowledge in the EHR and DDI graphs for medicine prediction. This is problematic because most of the knowledge in the EHR and DDI is irrelevant when making medicine predictions for an individual patient, and only a small portion of knowledge should be considered. When the EHR and DDI graphs are large, this introduces unnecessary computing costs and noise.

To address the issues listed above, we propose a graph convolutional reinforcement learning model, namely *Combined Order-free Medicine Prediction Network* (CompNet), for the MCP task. To effectively leverage medicine correlations while alleviating unreasonable assumption on the order of medicines, we cast MCP as a Markov Decision Process (MDP) problem and design a Deep Q Learning (DQL) mechanism to learn CompNet. To take correlative and adverse medicine interactions into consideration, we employ Relational Graph Convolutional Network (R-GCN) in a recurrent way to encode the correlations between predicted medicines and relevant adverse knowledge from a knowledge base. Specifically, we first use a Dual Convolutional Neural Network (Dual-CNN) network to obtain patient representations based on EHRs. Then, we introduce knowledge associated with predicted medicines to create a dynamic medicine knowledge graph, which is encoded using a R-GCN. Finally, a DQL network is used to select medicines step by step according to the current state consisting of the patient representation, the medicine graph representation, and the hidden state representation of CompNet from the previous step.

Our key technical contributions can be summarized as follows:

- To the best of our knowledge, we are the first to propose a framework, CompNet, that combines reinforcement learning with a relational graph convolutional network to perform Medicine Combination Prediction (MCP) and that considers correlations between medicines while eliminating previously made unreasonable assumptions about the order of different medicines.
- We incorporate dynamic medical knowledge in a medicine knowledge graph to capture the correlative and adverse relations between medicines, which can adaptively adjust medical knowledge according to the current predicted medicines.
- We demonstrate the effectiveness and safety of CompNet by comparing it with several state-of-the-art methods on a real EHR dataset.

2 METHOD

2.1 Preliminaries

Given diagnoses C_i^d (represented by a sequence of ICD-9 codes,[1] e.g., { "1125", "v433", "v4581" }) and procedures C_i^p (represented by a sequence of ICD-9 codes, e.g., {"0066", "3761", "3950" }) of a patient p_i, the task of *Medicine Combination Prediction* (MCP) is to select an optimal medicine set $Y_i \subseteq M$, where M are all candidate medicines and Y_i is a subset of M prescribed for patient p_i. Without loss of generality, we omit the notation i in the following sections.

We formulate MCP as a *Markov Decision Process* (MDP) ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{T}$, $\mathcal{R}$, γ), where $\mathcal{S}$ is an infinite set of states, $\mathcal{A}$ is a finite set of actions, $\mathcal{T}$ is the transition probability function of state, $\mathcal{R}$ is a reward function, and $\gamma \in (0, 1]$ is the discount factor. Our goal is to learn a policy $\pi(\theta)$ parameterized by θ to maximize the accumulated discounted rewards:

$$\pi(\theta)^* = \max \sum_{t=0}^{\infty} \gamma^t r_t, \tag{1}$$

where r_t is the immediate reward at timestamp t. With DQL, the goal of Eq. 1 is transformed to minimize the following loss function:

$$L(\theta) = \mathbb{E}_{(s_t, a_t, r_t, s_{t+1})}[(r_t + \gamma \max_a Q(s_{t+1}, a; \theta) - Q(s_t, a_t; \theta))^2], \tag{2}$$

where $Q(s_t, a_t; \theta)$ is the Q-function, which estimates the expected future reward of action a at state s. The update formula of parameter θ can be described as follow:

$$\theta \leftarrow \theta + \alpha(r_t + \gamma \max_a Q(s_{t+1}, a_t; \theta) - Q(s_t, a_t; \theta))\nabla_\theta Q(s_t, a_t; \theta), \tag{3}$$

where α is the learning rate.

2.2 CompNet

We propose CompNet to implement the above MDP process. As shown in Fig. 2, CompNet employs a Deep Neural Network (DNN) to approximate the Q-function, which outputs a Q-value for each given state-action pair (s_t, a_t) at timestamp t. Here, s_t is defined as a combination of patient representation $\hat{z}_t$ and medicine knowledge graph representation g_t associated with the current predicted medicines. At each timestamp t, CompNet greedily selects a medicine

[1] https://en.wikipedia.org/wiki/List_of_ICD-9_codes

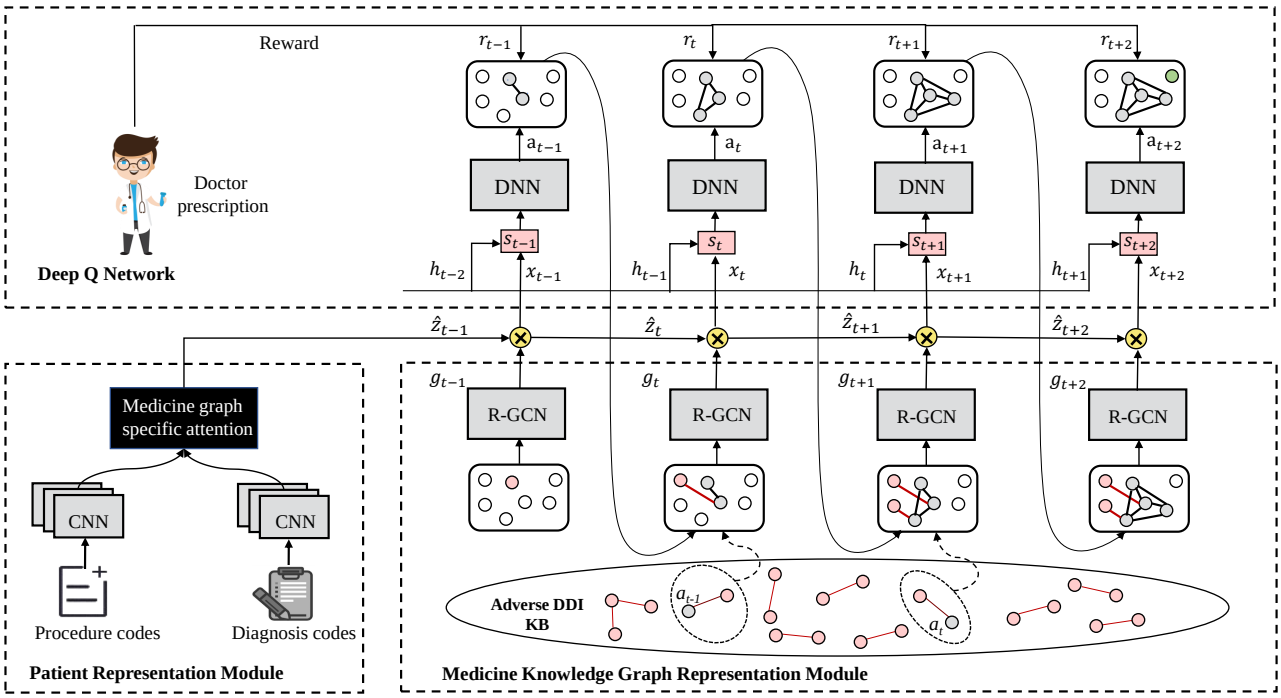

Figure 2: The architecture of CompNet. White dots and grey dots represent candidate medicines and selected medicines, respectively. Pink dots indicate medicines that appear in the adverse DDI knowledge base (KB), which holds adverse DDI pairs. The black edges connecting two grey dots indicate correlative relation while the red edges connecting grey and pink nodes or two pink dots indicate adverse relation. The green dot is "END" marker which indicates the end of one episode.

a_t according to the Q-value. The selected medicine a_t receives a reward r_t from the doctor, based on which CompNet updates its policy with Eq. 3. Meanwhile, the selected medicine will introduce new knowledge (nodes and edges) to the medicine knowledge graph, which requires CompNet to update its state to s_{t+1}. This process is iterated until the current episode ends. Next, we introduce how we model $Q(s_t, a_t; \theta)$, $\mathcal{S}$, $\mathcal{T}$, and $\mathcal{R}$ in detail.

Q-function $Q(s_t, a_t; \theta)$. We use a DNN with l layers to approximate the Q-function, which can be formulated as:

$$\begin{aligned} Q(s_t, a_t; \theta) &= F_l(\dots F_2(F_1(\boldsymbol{s_t}, a_t; \boldsymbol{\theta_1}); \boldsymbol{\theta_2}) \dots), \\ F_i(s_t, a_t; \theta_i) &= f(\boldsymbol{W}_{\theta_i} \boldsymbol{s}_t + \boldsymbol{b}_i), \end{aligned} \tag{4}$$

where F_i is the output of the i-th layer with ReLU activation function; $\boldsymbol{W}_{\theta_i}$ is the parameter matrix, and $\boldsymbol{b}_i$ is bias. $Q(s_t, a_t; \theta)$ corresponds to the output vector of the last layer.

State $\mathcal{S}$. The state $\boldsymbol{s}_t \in \mathcal{S}$ is calculated as follows:

$$\boldsymbol{s}_t = \sigma(\boldsymbol{W}_s \boldsymbol{h}_t), \tag{5}$$

where $\boldsymbol{W}_s$ is the learnable parameter matrix; σ is the sigmoid activation function; $\boldsymbol{h}_t$ is the hidden state, which consists of two parts as in Eq. 6:

$$\boldsymbol{h}_t = \sigma(\boldsymbol{W}_h \boldsymbol{x}_t + \boldsymbol{U}_h \boldsymbol{h}_{t-1}), \tag{6}$$

where $\boldsymbol{W}_h$ and $\boldsymbol{U}_h$ are parameter matrices, and $\boldsymbol{h}_{t-1}$ is the hidden state representation of CompNet at the previous step $t-1$; $\boldsymbol{h}_0$ is initialized with a zero vector; and $\boldsymbol{x}_t$ is the interaction representation between patient and medicine knowledge graph at step t, which is further calculated as follows:

$$\boldsymbol{x}_t = \boldsymbol{g}_t \odot \hat{\boldsymbol{z}}_t, \tag{7}$$

where $\boldsymbol{g}_t$ is the medicine knowledge graph representation at step t (see §2.3 for details); $\hat{\boldsymbol{z}}_t$ is the patient representation at step t (see §2.4 for details); and $\odot$ is the element-wise multiplication operation. We also tried other choices for the interactions, including addition $\boldsymbol{x}_t = \boldsymbol{g}_t + \hat{\boldsymbol{z}}_t$ and concatenation $\boldsymbol{x}_t = \boldsymbol{g}_t \oplus \hat{\boldsymbol{z}}_t$. In preliminary experiments we have found that different interaction choices have little influence on the results.

Transition $\mathcal{T}$. $\mathcal{T}$ represents the function of transition to next state, where each state is considered to be a possible result of selecting an action in a particular state. In our case, the transition function $\mathcal{T}$ is deterministic, which means the next state s_{t+1} is not stochastic and only depends on the current state and action pair (s_t, a_t). Specifically, the transition function transforms the medicine knowledge graph representation $\boldsymbol{g}_t$ to $\boldsymbol{g}_{t+1}$ and further changes the interaction representation $\boldsymbol{x}_t$ and hidden state $\boldsymbol{h}_{t-1}$, and finally obtains the new state $\boldsymbol{s}_{t+1}$. So the transition can be expressed as:

$$\mathcal{T}(\boldsymbol{s}_t, a_t) = \mathcal{T}((\boldsymbol{x}_t, \boldsymbol{h}_{t-1}), a_t) = \mathcal{T}(\boldsymbol{x}_{t+1}, \boldsymbol{h}_t) = \boldsymbol{s}_{t+1}. \tag{8}$$

Reward $\mathcal{R}$. The reward $r_t \in \mathcal{R}$ indicates how well the selected medicine is at step t. During online learning, we can define the reward based on feedback from real doctors who interact with our model. In this paper, we define the reward with an offline simulation. Specifically, we know the prescribed medicines for each patient, so if the medicine predicted by CompNet is in the prescribed

medicine set by a doctor, we give CompNet a positive reward $r_t = 1$, otherwise we give it a negative reward $r_t = -1$.

2.3 Medicine knowledge graph representation module

This module encodes the medicine knowledge graph to obtain representation $\boldsymbol{g}_t$, which is used to update the state $\boldsymbol{s}_t$ in Eq. 5. We propose an adaptive medicine knowledge graph mechanism to create an undirected multi-relational medicine knowledge graph $G_t = (\mathcal{V}_t, \mathcal{E}_t, \mathfrak{V})$ at t. Here, $\mathcal{V}_t = \mathcal{V}_{<t} \cup \mathcal{V'}_{V_{<t}}$ is the set of all nodes, where $\mathcal{V}_{<t}$ is the set of currently predicted medicines and $\mathcal{V'}_{V_{<t}}$ is the set of one-hop neighbors of $V_{<t}$; $\mathcal{E}_t$ is the set of all edges between two nodes in $\mathcal{V}_t$; and $\mathfrak{V}$ is the set of relation types. Here, we consider two types of medicine relation for $\mathcal{E}_t$, i.e., adverse medicine relation and correlative medicine relation. The adverse medicine relation is obtained from a DDI knowledge base, as indicated by red edges between nodes in Fig. 2. The correlative medicine relation is obtained by the CompNet model. The medicine selected at the current step has a correlative relationship with each of the selected medicines at previous steps. In Fig. 2, the correlative medicine relations are indicated by black edges between nodes.

We use a Relational Graph Convolutional Network (R-GCN) [32] to encode G_t, which is an extension of the Graph Convolutional Network (GCN) [19] on multi-relational graphs. Given a specific node v_i in the medicine knowledge graph, we use the following convolution operation to calculate the representation of node v_i at the l-th layer:

$$\boldsymbol{h}_i^l = \sum_{r \in \mathfrak{V}} \sum_{j \in N_i^r} \frac{1}{c_{i,j}} \boldsymbol{W}_r^{l-1} \boldsymbol{h}_j^{l-1} + \boldsymbol{W}_o^{l-1} \boldsymbol{h}_i^{l-1}, \tag{9}$$

where $\boldsymbol{h}_i^l \in \mathbb{R}^{c_l}$ is the representation of node v_i at the l-th layer; c_l is the number of output channels of the l-th graph convolution layer; $\mathfrak{V}$ is the set of all relations considered in the medicine knowledge graph G_t; N_i^r represents neighbors of node v_i according to the relation r; $c_{i,j}$ is a problem-specific normalization constant that can either be learned or chosen in advance, here we set $c_{i,j} = |N_i^r|$; and $\boldsymbol{W}_r^{l-1}$ and $\boldsymbol{W}_o^{l-1}$ are parameter matrices.

In our model, we need to get the representation of entire medicine knowledge graph G_t. Inspired by [47], we use the following formula to obtain the representation $\boldsymbol{g}_t$:

$$\boldsymbol{g}_t = \sum_{j=1}^{|\mathfrak{V}|*|M|} \boldsymbol{H}_j^{1:L}, \tag{10}$$

where $\boldsymbol{g}_t \in \mathbb{R}^{\sum_1^L c_l}$ is obtained by adding each row of $\boldsymbol{H}^{1:L} \in \mathbb{R}^{|\mathfrak{V}|*|M| \times \sum_1^L c_l}$; $|\mathfrak{V}|$ is the number of relations; $|M|$ is number of medicine nodes; and L is the number of graph convolution layers. The matrix $\boldsymbol{H}^{1:L}$ is calculated by concatenating the output of the R-GCN in each layer along the last axis:

$$\boldsymbol{H}^{1:L} = \boldsymbol{H}^1 \oplus \boldsymbol{H}^2 \oplus \cdots \oplus \boldsymbol{H}^l \oplus \cdots \oplus \boldsymbol{H}^L, \tag{11}$$

where $\oplus$ is the concatenation operation. $\boldsymbol{H}^l \in R^{|\mathfrak{V}|*|M| \times c_l}$ is obtained by concatenating all the node representation $\boldsymbol{h}_i^l$ at the l-th layer along the first axis, which can be formalized as:

$$\begin{aligned} \boldsymbol{H}^l &= \boldsymbol{h}_1^l \oplus \boldsymbol{h}_2^l \oplus \cdots \oplus \boldsymbol{h}_i^l \oplus \cdots \oplus \boldsymbol{h}_{|M|}^l, \\ \boldsymbol{H}^1 &= \boldsymbol{A}_a \oplus \boldsymbol{A}_c, \end{aligned} \tag{12}$$

where $\boldsymbol{h}_i^l$ is the representation of medicine node v_i at the l-th layer (Eq. 9). The representation of the first layer $\boldsymbol{H}^1$ is initialized by concatenating the adverse adjacency matrix $\boldsymbol{A}_a$ and correlative adjacency matrix $\boldsymbol{A}_c$ of the medicine knowledge graph G_t.

2.4 Patient representation module

This module encodes the patient representation $\hat{\boldsymbol{z}}_t$, which is used to update the state $\boldsymbol{s}_t$ in Eq. 5. We calculate $\hat{\boldsymbol{z}}_t$ as follows:

$$\hat{\boldsymbol{z}}_t = \boldsymbol{Z} \boldsymbol{\alpha}_t, \tag{13}$$

where $\boldsymbol{Z} = \boldsymbol{z}^d \oplus \boldsymbol{z}^p$, which is the concatenation of the diagnoses representation $\boldsymbol{z}^d$ and procedures representation $\boldsymbol{z}^p$ along the first axis; Due to the effectiveness of the attention mechanism [? ?], attention weights $\boldsymbol{\alpha}_t$ are used to balance $\boldsymbol{z}^d$ and $\boldsymbol{z}^p$; $\boldsymbol{\alpha}_t$ is assigned by a softmax function based on the medicine graph representation $\boldsymbol{g}_t$:

$$\boldsymbol{\alpha}_t = \text{softmax}(\boldsymbol{Z}^T \boldsymbol{g}_t). \tag{14}$$

The diagnoses representation $\boldsymbol{z}^d$ and procedures representation $\boldsymbol{z}^p$ can be calculated by the following formulas:

$$\begin{aligned} \boldsymbol{z}^d &= \sigma(\boldsymbol{W}_d \boldsymbol{v}^d + \boldsymbol{b}_d) \\ \boldsymbol{z}^p &= \sigma(\boldsymbol{W}_p \boldsymbol{v}^p + \boldsymbol{b}_p), \end{aligned} \tag{15}$$

where $\boldsymbol{W}_d$ and $\boldsymbol{W}_p$ are the weight matrices, and $\boldsymbol{b}_d$ and $\boldsymbol{b}_p$ are the biases. The vectors $\boldsymbol{v}^d$ and $\boldsymbol{v}^p$ are obtained by flattening matrices $\boldsymbol{V}^d$ and $\boldsymbol{V}^p$. $\boldsymbol{V}^d$ and $\boldsymbol{V}^p$ are obtained by a linear convolution operation followed by a non-linear transformation function as in Eq. 16:

$$\begin{aligned} \boldsymbol{V}^d &= \tanh(\boldsymbol{W}_{conv}^d * \boldsymbol{E}^d + \boldsymbol{B}^d) \\ \boldsymbol{V}^p &= \tanh(\boldsymbol{W}_{conv}^p * \boldsymbol{E}^p + \boldsymbol{B}^p), \end{aligned} \tag{16}$$

where $*$ denotes the convolution operator and tanh represents the tanh activation function; $\boldsymbol{W}_{conv}^d$ and $\boldsymbol{W}_{conv}^p$ are the parameter matrices; $\boldsymbol{B}^d$ and $\boldsymbol{B}^p$ are biases; and $\boldsymbol{E}^d$ and $\boldsymbol{E}^p$ are diagnose embedding and procedures embedding, respectively, which are obtained by linear projections as follows:

$$\begin{aligned} \boldsymbol{E}^d &= \boldsymbol{W}_e^d \boldsymbol{m}^d \\ \boldsymbol{E}^p &= \boldsymbol{W}_e^p \boldsymbol{m}^p, \end{aligned} \tag{17}$$

where $\boldsymbol{W}_e^d$ and $\boldsymbol{W}_e^p$ are the embedding matrices, which are jointly learned with model parameters; $\boldsymbol{m}^d$ and $\boldsymbol{m}^p$ are bag-of-word vectors that are converted from diagnosis codes C^d and procedure codes C^p, respectively.

2.5 Learning process

The learning process of CompNet is shown in Algorithm 1. In order to avoid predicting duplicate medicines for the same patient, we use the set C to record candidate medicines and set B to record selected medicines. For each episode, we first empty B and reset $C = \mathcal{A}$. Then, we run ε-greedy policy [39] using CompNet on the candidate medicine set C. Specifically, we select a medicine

Algorithm 1: Learning CompNet for MCP.

1 Initialize replay memory D, the whole medicine set $\mathcal{A}$;
2 **for** *epoch = 1: EPOCHS* **do**
3 **for** *each patient* **do**
4 Initialize candidate medicine set $C = \mathcal{A}$;
5 Initialize selected medicine set B as empty set;
6 Initialize state $\boldsymbol{s}_0$ with Eq. 5 and 6;
7 **for** *t=1:T* **do**
8 Select medicine a_t from candidate medicine set C using ε-greedy policy;
9 Get reward r_t for medicine a_t;
10 Update medicine knowledge graph and compute next state $\boldsymbol{s}_{t+1}$ according to Eq. 5;
11 Store transition $(\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}_{t+1})$ to D;
12 Store medicine a_t into selected medicine set B;
13 Update candidate medicine set $C = \mathcal{A} - B$;
14 Update $\boldsymbol{s}_t$ with $\boldsymbol{s}_{t+1}$;
15 Select $|D_S|$ samples from D randomly;
16 Compute loss on D_S with Eq. 2 and update the parameters of CompNet with Eq. 3;
17 **end**
18 **end**
19 **end**

a_t by either following the greedy policy $\arg\max_a Q(\boldsymbol{s}_t, a)$ with probability $1 - \varepsilon$ or following a random policy with probability ε from C (line 8). With the selected a_t, we can get the reward r_t from doctors (simulated doctors in our experiments). After that, we update the medicine knowledge graph based on the selected a_t (see §2.3 for details) and derive the next state $\boldsymbol{s}_{t+1}$ (line 10). We store the tuple $(\boldsymbol{s}_t, a_t, r_t, \boldsymbol{s}_{t+1})$ into the replay memory D and update the selected medicine set B and candidate medicine set C accordingly (line 11–13). Finally, we sample a set of tuples D_S from D and compute the loss to update CompNet based on D_S (line 16).

3 EXPERIMENTAL SETUP

We set up a series of experiments to evaluate the performance of CompNet. Details of our experimental setting are given below.

3.1 Research questions

Our experiments are meant to answer the following research questions.

(RQ1) What is the performance of CompNet on the MCP task? Does it outperform state-of-the-art methods? (See §4.)
(RQ2) Where do the improvements of CompNet come from? What are the effects of different components? (See §5.1.)
(RQ3) Which one is more effective, incorporating all DDI knowledge or adaptively incorporating DDI knowledge associated with the predicted medicines? (See §5.2.)
(RQ4) Is the training process of CompNet stable? How do different evaluation metrics change with respect to training iterations? (See §5.3.)

3.2 Datasets

We perform experiments on a publicly available dataset, namely MIMIC-III [17].[2] Descriptive statistics of MIMIC-III dataset are given in Table 1. As with previous studies [e.g., 34], we choose medicines that are prescribed by doctors for each patient within the first 24 hours as medicine set since it is usually a critical period for each patient to get accurate and rapid treatment in the first 24 hours [9, 10]. All methods in our experiments use DDI knowledge from the TWOSIDES dataset [38] to avoid adverse medicine combinations. We use the top-40 severe DDI types and transform medicine codes from NDC to ATC Level 3 for integration with the MIMIC-III dataset. Besides, we filter out samples whose prescriptions contain adverse DDI.

Table 1: Statistics of the MIMIC-III datasets.

MIMIC-III	Quantity
# patients	5,847
# clinical events	13,727
# diagnoses	1,954
# procedures	1,352
# medicines	138
avg # of visits	10.824
avg # of diagnoses	6.441
avg # of procedures	3.883
avg # of medicines	2.348
# related DDI pairs	460
# medicine in DDI knowledge base	123

3.3 Implementation details

We randomly divide the MIMIC-III dataset into training, validation and test sets with 2/3 : 1/6 : 1/6 ratios. We use a DNN with 3 hidden layers to implement the deep Q network, where the hidden size of each layer is set to 512 (Eq. 4). In the medicine knowledge graph representation module, the output channel for each graph convolutional layer is set to 50, and we use two layers (Eq. 9). In the patient representation module, the hidden size of diagnoses representation z^d and procedures representation z^p are set to 100 (Eq. 15). We use 3 convolutional layers, and the filter sizes are all set to 128. The kernel size of each convolutional layer is set to 3. After the last convolutional layer, we apply dropout [35] and the drop ratio is set to 0.5 (Eq. 16). The embedding sizes of both $\boldsymbol{W}_e^d$ and $\boldsymbol{W}_e^p$ are set to 100 (Eq .17). During CompNet training phase, the size of the replay memory D is set to 2,000 (Algorithm 1). The initial exploration rate ε is 0.995, and minimum exploration rate is 0.05. We initialize model parameters randomly with the Xavier method [12] and set the training batch size to 64. We choose Adam [43] to optimize all parameters in CompNet. The learning rate $\alpha = 0.00001$ and the momentum parameters are set to default $\beta 1 = 0.9$ and $\beta 2 = 0.999$. CompNet has been implemented in PyTorch and trained on a GeForce GTX TitanX GPU.[3]

[2]The dataset used in this paper is available at https://mimic.physionet.org
[3]The source codes are available at https://github.com/WOW5678/CompNet

3.4 Methods used for comparison

We compare CompNet to the following baselines:

K-frequent. K-frequent predicts medicines by counting the co-occurring frequency of each diagnosis with each medicine. For each diagnosis, it simply selects the top K most frequently occurring medicines. We tried K from 1 to 8 and finally set K to 5 according to its performance on validation set.

K-nearest. To prescribe medicines for a patient p_i, K-nearest selects the medicines prescribed for patient p_j who has the most similar diagnoses with p_i. Similarity between two patients is measured by Jaccard distance; Here, we set K to 1.

Multi-layer Perceptron (MLP). MLPs are conventional methods to solve multi-label classification problem. We learn a multi-label classification model with a three-layer perceptron. The last layer uses sigmoid as activation function to predict the probability of each medicine [11].

Classifier Chain (CC). CC [31] is another commonly used multi-label learning method. We use 3 base classifiers to form a classifier chain to predict medicines and each base classifier is a decision tree [29].

SGM. SGM [45] views multi-label classification as a sequence generation task and uses a seq2seq model to predict labels. Here, we use SGM for predicting medicine combinations.

LEAP. LEAP [48] formulates the medicine prediction problem as a *Multi-Instance Multi-Label Learning* (MIML) problem. Similar to SGM, it uses a Recurrent Neural Network (RNN) to predict medicines. Reinforcement learning is used to tune parameters.

GAMENet. GAMENet [34] uses a memory network to embed the DDI knowledge graph and EHR graph based on a GCN. Then it concatenates the patient representation and the memory output to predict medicines.

3.5 Evaluation metrics

We use the Mean Jaccard Coefficient [20], Average Recall, Average Precision and Average F1 to measure the performance of all methods. For a particular patient p_i, assume that the predicted medicines are Y_i, and $\hat{Y}_i$ is the ground truth medicines that doctors prescribe for the patient. The Mean Jaccard Coefficient is defined as the size of the intersection divided by the size of the union of predicted medicines and ground truth medicines. Recall can measure the completeness of predicted medicines and Precision can measure the correctness of predicted medicines. F1 is the harmonic mean of Precision and Recall, and is often used as a comprehensive evaluation metric of prediction model:

$$Jaccard = \frac{1}{m}\sum_{i}^{m}\frac{|Y_i \cap \hat{Y}_i|}{|Y_i \cup \hat{Y}_i|}$$

$$Recall = \frac{1}{m}\sum_{i}^{m}\frac{|Y_i \cap \hat{Y}_i|}{|\hat{Y}_i|}$$

$$Precision = \frac{1}{m}\sum_{i}^{m}\frac{|Y_i \cap \hat{Y}_i|}{|Y_i|}$$

$$F1 = \frac{1}{m}\sum_{i}^{m}\frac{2 * Precision_i * Recall_i}{Precision_i + Recall_i},$$

where i is a sample index in the test set and m is the size of the test set. $Precision_i$ and $Recall_i$ denote the precision value and recall value of patient p_i, separately.

In addition, we use the DDI Rate to measure the safety of MCP models. The DDI Rate is defined as the number of DDI pairs occurring in predicted medicines divided by the pair number of all possible medicine combinations:

$$\text{DDI Rate} = \frac{1}{m}\sum_{i}^{m}\frac{|(c_j, c_k) \in Y_i \ \ \& \ \ (c_j, c_k) \in \mathcal{E}_{ddi}|}{\sum_{j,k} 1},$$

where $\mathcal{E}_{ddi}$ represents all DDI pairs in the DDI knowledge base.

4 RESULTS (RQ1)

The results of comparing all methods are shown in Table 2. CompNet outperforms all baselines in terms of multiple metrics. From the results, we can gain several insights.

Table 2: Performance comparison (%) of different methods.

Method	Jaccard	Recall	Precision	F1	DDI Rate
K-frequent	29.64	48.01	48.57	43.65	5.32
K-nearest	22.39	39.74	40.41	34.12	7.02
MLP	23.03	34.36	44.30	34.38	2.29
CC	16.16	31.69	30.07	25.93	**2.01**
SGM	27.90	52.08	40.32	41.66	5.45
LEAP	25.76	42.74	45.53	42.74	7.08
GAMENet	28.77	43.00	**50.02**	41.04	6.94
CompNet	**32.51***	**57.05***	45.53	**47.68***	2.78

Bold face indicates the best result in terms of the corresponding metric. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.01$).

First, CompNet significantly outperforms all baselines in terms of most evaluation metrics (i.e., Jaccard, Recall and F1). The improvements over CC and K-frequent are 16.35% (at most) and 2.87% (at least) in terms of Jaccard. The improvements over CC and SGM on Recall are 25.36% (at most) and 4.97% (at least) respectively. And on F1, the improvements reach to 21.75% (at most) and 4.03% (at least) over CC and K-frequent. The reasons for the improvements are three-fold. (1) CompNet uses a recurrent DQL structure to predict medicines. Medicine selection in a given step depends on selected medicines in previous steps, so the correlation between medicines can be captured. (2) CompNet is not sensitive to the order of medicine combinations and does not have to make unreasonable order assumptions during training. (3) CompNet dynamically introduces knowledge related with selected medicines and uses R-GCN to distinguish between different types of knowledge. The introduction of medical knowledge makes the model perform well (see §5.1 for an in-depth analysis).

Second, CompNet performs better than all baselines except for GAMENet on Precision. Although the Precision of GAMENet is 4.49% higher than CompNet, the Recall value of GAMENet is significantly lower (14.05%) than CompNet. This phenomenon shows that GAMENet is more inclined to the correctness of predicted medicines, but misses most of the ground truth medicines. However,

in medical cases, the Recall metric is significantly more meaningful than Precision metric. The reason is that current MCP models cannot totally replace doctors and only help doctors prescribe medicines for patients as an assistant [21]. As a result, the more important function of MCP models is to help doctors screen possible medicines more comprehensively.

Third, CompNet performs better than all baselines except for MLP and CC on DDI Rate. The DDI Rate of CompNet is 2.78% which is lower than the DDI Rates of K-frequent, K-nearest, SGM, LEAP and GAMENet. But it is higher than CC and MLP, whose DDI Rates are 2.01% and 2.29% respectively. Since the number of correct medicines for each patient is much smaller (the average number of medicines is 2.348 for each patient, see Table 1) than the number of candidate medicines (138 medicines, see Table 1), MLP and CC tend to select fewer medicines [14]. Because of this, their DDI Rates are lower than CompNet. In fact, MLP predicts each medicine separately, regardless of correlations between different medicines, which means that MLP does not have the ability to avoid adverse medicine combinations like CompNet. As to CC, it uses multiple concatenation classifiers and utilizes the classification results of the previous classifiers, the correlations between medicines preserve. But the results can vary for different orders of classifier chains [15].

Fourth, the K-frequent method performs better than many other baselines, including recently proposed deep learning models, such as LEAP and GAMENet. K-frequent is a rule-based approach that performs medicine prediction through the co-occurrence of medicines and diagnoses. This implies that many doctors choose frequently used medicines for the same or similar diseases (diagnoses). This does not imply that K-frequent is the most effective and useful method in practice: doctors are already familiar with frequently used medicines. As an assistant, K-frequent cannot recommend novel and less frequently used medicines to doctors. Besides, because frequently used medicines are seldom updated, K-frequent cannot learn from historic data to adaptively adjust its strategy.

5 ANALYSIS

5.1 Ablation study (RQ2)

The effectiveness and safety of CompNet have been demonstrated in the previous section. To show where the improvements of CompNet come from, we report the results of CompNet with 4 different settings, as shown in Table 3 and described next:

No Graph. To test the effect of the medicine knowledge graph and the proposed R-GCN, we replace R-GCN with an MLP to project selected medicines into the vector $\boldsymbol{g}_t$ at timestamp t, and then together with the patient representation $\hat{\boldsymbol{z}}_t$ to constitute the current state $\boldsymbol{s}_t$.

No CMR. To verify the effectiveness of the correlative medicine relation, we remove this relation from the medicine knowledge graph.

No AMR. To test the effect of adverse medicine relations, we remove the DDI knowledge from the medicine knowledge graph.

With COR. GAMENet shows that the medicine co-occurrence relation helps to improve performance. We also tried this relation by adding it to the medicine knowledge graph. The co-occurrence relation is extracted from the training set. If two medicines are co-occurring in an EHR, we consider there to be a co-occurrence relation between them.

Table 3: Analysis of different components in CompNet (%).

Method	Jaccard	Recall	Precision	F1	DDI Rate
No Graph	12.56	28.32	19.27	21.68	**1.90**
No CMR	31.01	51.30	48.03	46.15	2.22
No AMR	32.13	49.78	**52.08**	47.31	8.33
With COR	27.64	53.64	39.16	42.23	3.85
CompNet	**32.51***	**57.05***	45.53	**47.68***	2.78

The superscript * indicates that CompNet significantly outperforms other models with different settings, using a t-test with $p < 0.01$.

From the results in Table 3, we obtain the following insights. First, the performance of CompNet decreases dramatically after removing medicine knowledge graph (i.e., *No Graph*). Specifically, the results of all evaluation metrics drop more than 20%. Although the DDI Rate of *No Graph* is slightly better than CompNet, this improvement comes at the expense of its effectiveness. The result shows that medicine knowledge graph plays a crucial role in CompNet.

Second, the performance of CompNet decreases after removing the correlative medicine relation or adverse medicine relation. *No CMR* only uses adverse medicine knowledge while *No AMR* only uses correlative medicine knowledge. The Jaccard, Recall and F1 results of *No CMR* drop a lot compared with CompNet, which means that correlative medicine knowledge helps. The DDI Rate of *No AMR* is the worst, which means adverse medicine knowledge plays a key role in the safety of CompNet. Besides, the DDI Rate of *No CMR* is 0.56% lower than CompNet. This is because *No CMR* only needs to focus on adverse knowledge, while CompNet needs to take two kinds of knowledge (i.e., correlative and adverse medicine knowledge) into consideration.

Third, the performance of CompNet drops after adding the co-occurrence medicine relation (i.e., *With COR*). This is because the co-occurrence relation actually brings a lot of noise and interferes with the model's learning. Some patients have multiple diseases which are not related. The prescribed medicines for these patients do not have any co-effects or correlations [7]. However, this will mislead the learning of the model.

5.2 Effect of adaptive medical knowledge (RQ3)

To evaluate the rationality of the adaptive medicine knowledge graph mechanism in CompNet (see §2.3.), we compare the performance of CompNet with a modified version, named *With all KG*, that loads *all* adverse medicine knowledge at each step for medicine selection, rather than just introducing related medicine knowledge. The results are shown in Table 4. The performance of CompNet

Table 4: Analysis of adaptive medical knowledge in CompNet (%).

Method	Jaccard	Recall	Precision	F1	DDI Rate
With all KG	31.02	**57.81**	42.60	45.81	3.84
CompNet	**32.51**	57.05	**45.53**	**47.68**	**2.78**

with all knowledge added (i.e., *With all KG*) drops in terms of Jaccard, Precision and F1, while the DDI rate increases 1.06%. This

suggests that introducing all medical knowledge in each step will result in a higher risk of predicting adverse medicine combinations. This is because in each step, most knowledge in the DDI knowledge base is irrelevant, which makes it harder for the model to recognize useful knowledge. However, the Recall score of *With all KG* is a little better, as is to be expected, which means that our adaptive medicine knowledge graph mechanism also loses some useful knowledge.

5.3 Prediction results on different training batches (RQ4)

In Fig. 3 we plot the results of different evaluation metrics with respect to training batches. We change the number of training batches from 0 to 18,000, and evaluate the model's performance on validation set every 100 training batches. The Jaccard, Recall, and F1

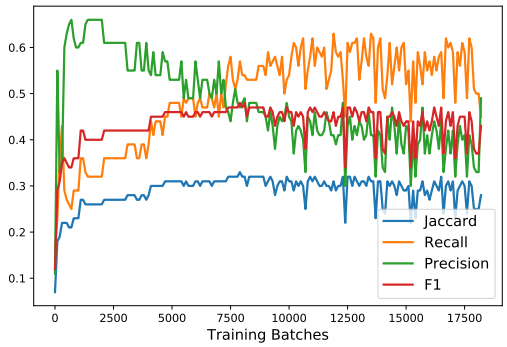

Figure 3: Change curves of different metrics under different batches on validation set.

scores show an increasing tendency generally, but are also in fluctuation due to the instability of DQL [1]. During the training process, the Precision values show a decreasing tendency. This is because Precision is sensitive to the number of predicted medicines. For example, in an extreme case, the model only predicts one medicine for a patient. Even if the predicted medicine is in the ground truth medicine set and then Precision reaches 100%, this does not mean we obtain a good model. Therefore, we should pay more attention to Jaccard, Recall, and F1 values.

5.4 Case study

We select a patient with complicated diagnoses from the test set to analyze the results from different methods. Table 5 shows ground truth medicines prescribed by doctors and medicines predicted by different methods. In summary, CompNet has the best performance with 6 correct medicines, 1 wrong medicine and 0 missing medicine. K-frequent and K-nearest can correctly predict 5 and 4 medicines respectively. K-frequent only focuses on the co-occurrence of diagnoses and medicines. As a result, it only correctly predicts 5 frequently used medicines. K-nearest predicts 17 medicines for this patient, of which only 4 are correct. The MLP and CC methods predict 0 and 2 medicines, respectively. This suggests that these two methods tend to predict fewer medicines, resulting in a large number of missing medicines. SGM, LEAP, and GAMENet predict 3, 3, 6 correct medicines and 4, 5, 5 wrong medicines, respectively. CompNet predicts a total of 7 medicines for this patient and the number of correct medicines is 6. We can see that CompNet is more effective not only in predicting medicine combinations but also the number of medicines.

6 RELATED WORK

We consider three types of related work: medicine combination prediction, reinforcement learning for healthcare, and graph convolutional networks for healthcare.

6.1 Medicine combination prediction

We survey *Binary Relevance* (BR) based methods and *Sequence Prediction* (SP) based methods for MCP.

BR based methods. Binary Relevance (BR)-based methods formulate MCP as multiple binary classification problems. Bajor and Lasko [2] use an RNN to encode patients by taking patients' historical diagnoses into account; they then use an MLP to predict medicines for patients. Choi et al. [6] propose a two-layer attention neural network, named RETAIN, to detect past influential EHRs for the current visit. Wang et al. [42] establish a patient-medicine bipartite graph and a patient-disease bipartite graph, and then use knowledge representation to model patients for MCP. Le et al. [22] design a memory augmented neural network to perform MCP based on historical EHRs. Wang et al. [41] propose a trilinear model to integrate multi-source patient information from EHRs, such as demographic information and laboratory indicators, to predict medicines for patients.

Furthermore, some methods use genomic information to predict appropriate medicines for patients, such as [4]. But genomic information is difficult to acquire. The methods mentioned above all ignore complex correlations between different medicines. Medicines are considered to be independent of each other, and each medicine is predicted separately. Because of this, the methods listed above (except for [42]) all ignore safety in the MCP task. That is, when multiple medicines are used together, there is a high probability of producing adverse DDI. Unlike the methods listed above, CompNet captures medicine relationships by employing an adaptive medicine knowledge graph to incorporate the correlative relations between the predicted medicines and the adverse relations between the related medicines.

SP based methods. Sequence Prediction (SP)-based methods formulate MCP as a sequence prediction problem, where they predict one medicine at a time during both training and testing. For example, Zhang et al. [48] propose LEAP, which uses an encoder-decoder framework for MCP, and reinforcement learning to fine-tune the model based on the DDI knowledge base so as to avoid adverse medicine combinations. Shang et al. [33] utilize an encoder-decoder framework to learn representations of diagnoses and medicines, and then establish a mapping between them for MCP. Although the methods above consider correlations between different medicines, they need to pre-set the orders for medicine combinations during training. Different pre-setting orders will affect the model results, which is unreasonable. Medicine combination should be essentially an order-free set. In this paper, we propose CompNet, where medicine selection at a given step is dependent on selected medicines at previous steps, so the correlations between medicines

Table 5: Example of predicted medicines by different methods.

Diagnoses	Method	Predicted medicine combination
1125, 99592, 78552, 2760, 42731, 4254, 496, 4821, 78003, 70703, 70707, 70705, 51883, 27651, 5859, v433, v4581, 25000, v5867, 2720, 40390, v441, v440, 2449, 1122	Ground Truth	7 (B05C, B01A, A12C, H03A, J01D, A02B, A07A)
	K-frequent	5 correct +0 wrong + 2 missed (A12C, B05C, A02B, A07A, B01A)
	K-nearest	4 correct+ 13 wrong + 0 missed (A07A, N02B, A01A, A02B, A06A, A12A, A12C, B05C, N07A, A10A, C01C, N01A, C01D, A03F, N02A, D07A, B05A)
	MLP	0 correct + 0 wrong + 7 missed (None)
	CC	2 correct + 0 wrong+ 5 missed (A02B,A12C)
	SGM	3 correct + 4 wrong + 0 missed (A06A, A02B, N02B, B05C, C02D, A12C)
	LEAP	3 correct + 5 wrong +0 missed (A06A, A01A, A02B, A12C, B05C, A12A, C01C, N01A)
	GAMENet	6 correct + 5 wrong + 0 missed (A07A, A06A, N02B, A01A,A02B, A12C, B05C, B01A, J01D, R01A, R03A)
	CompNet	6 correct + 1 wrong + 0 missed (A12C, A02B, B05C, J01D, A07A, N02A, B01A)

can be captured. Meanwhile, we do not make any sequential assumptions during training. Instead, we make the model explore order-free correlations in a reinforcement learning manner.

6.2 Reinforcement learning for healthcare

In healthcare, Reinforcement Learning (RL) can be used to perform a variety of medical tasks, such as disease diagnosis [3, 18, 37]. Tang et al. [37] use RL to create an effective and efficient symptom checker to predict disease by asking patient questions. This method not only improves the accuracy of symptom checker, but also minimizes interaction number of symptom checker and patients. Besson et al. [3] focus on disease diagnosis based on RL by minimizing the average number of medical tests. Kao et al. [18] use hierarchical reinforcement learning for selecting symptoms to inquire and diagnose within the expertise of different anatomical part to improve diagnosis accuracy. Nemati et al. [27] leverage a combination of Hidden Markov Models and deep Q-networks to predict optimal heparin dosing for ICU patients. Similar to Nemati et al. [27], Raghu et al. [30] use RL to find optimal dosages of intravenous fluids and vasopressors during the treatment of sepsis patients which can lead to improved treatment. However, RL has rarely been applied to the MCP task. Although LEAP [48] introduces RL in MCP problem, the core of its MCP model depends on encoder-decoder framework and RL is just for fine-tuning model. We formulate the MCP task as a MDP and propose an RL model to solve it.

6.3 Graph convolutional networks for healthcare

The Graph Convolutional Network (GCN) is used to extract nodes and associated information from a graph so as to represent node or graph structures [16, 19]. GCN has been successfully used in healthcare. For example, Ma et al. [24] create drug association graph with drugs as nodes and DDI as edges, and further extend a GCN to encode multi-view drug features and edges to measure drug similarity. Zitnik et al. [50] create a multimodal graph including protein-protein interactions, drug-protein target interactions, and DDI. And then they use a GCN to encode them for a multirelational link prediction task. Mao et al. [25] propose MedGCN, which is a heterogeneous medical graph including multiple types of medical entities and relations. Then they employ a GCN to learn representations based on MedGCN, which are used for lab test imputation and medication recommendation. Most recently, [34] propose GAMENet which uses GCN to enhance performance on the MCP task. They first create the EHR medicine graph and the DDI medicine graph. Then they use GCN to encode two graphs and integrate them for MCP. In this paper, we propose CompNet which further combines R-GCN with RL. The former takes care of the multiple relations between medicines while the later forces our model to learn order-free dependencies out of the various relations. Additionally, CompNet creates a dynamic multi-relational medicine graph that adaptively introduces related medicine knowledge w.r.t. current predicted medicines, which we have shown in the experiments to be more effective than using the whole knowledge graph for all predictions.

7 CONCLUSION AND FUTURE WORK

We have presented a novel model named CompNet for MCP, which is meant to capture useful correlations between medicines while eliminating the unreasonable assumption on medicine orders made in previous work. We have verified the effectiveness of CompNet through extensive experiments on a benchmark dataset for MCP. The results demonstrate that the proposed modules in CompNet bring improvements and CompNet achieves the best performance compared with state-of-the-art methods. Especially, CompNet outperforms GAMENet (a recently proposed model) by a large margin (3.74%pt, 6.64%pt in terms of Jaccard and F1 metrics, respectively). Meanwhile, CompNet achieves a much lower DDI ratio than GAMENet in terms of safety.

A limitation of CompNet is that the learning process will be extremely hard and unstable when extended to a large medicine space, which is a common issue for the current RL technologies. As to future work, CompNet can be advanced in two aspects. Firstly, we hope to make CompNet work on datasets with a larger number of medicine candidates by proposing better learning algorithms. Secondly, we also hope to learn a model that can simulate doctors to give feedback to guide the learning of CompNet.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This work is supported by the Natural Science Foundation of China (61672324, 61672322, 61972234, 61902219), the Natural Science Foundation of Shandong province (2016ZRE27468), the Tencent AI Lab Rhino-Bird Focused Research Program (JR201932), the Fundamental Research Funds of Shandong University, Ahold Delhaize, the Association of Universities in the Netherlands (VSNU), and the Innovation Center for Artificial Intelligence (ICAI). All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

REFERENCES

[1] Oron Anschel, Nir Baram, and Nahum Shimkin. 2017. Averaged-DQN: variance reduction and stabilization for deep reinforcement learning. In *ICML 2017*. 176–185.

[2] Jacek M. Bajor and Thomas A. Lasko. 2017. Predicting medications from diagnostic codes with recurrent neural networks. In *ICLR 2017*.

[3] Remi Besson, Erwan Le Pennec, Stephanie Allassonniere, J Stirnemann, Emmanuel Spaggiari, and Antoine Neuraz. 2018. A model-based reinforcement learning approach for a rare disease diagnostic task. *CoRR* abs/1811.10112 (2018).

[4] Nikhil Cheerla and Olivier Gevaert. 2017. MicroRNA based Pan-Cancer diagnosis and treatment recommendation. *BMC bioinformatics* 18, 1 (2017), 32.

[5] Feixiong Cheng, István A Kovács, and Albert-László Barabási. 2019. Network-based prediction of drug combinations. *Nature communications* 10, 1 (2019), 1197.

[6] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. 2016. Retain: an interpretable predictive model for healthcare using reverse time attention mechanism. In *NIPS 2016*. 3504–3512.

[7] Leslie Citrome. 2009. Quantifying risk: the role of absolute and relative measures in interpreting risk of adverse reactions from product labels of antipsychotic medications. *Current drug safety* 4, 463 (2009), 229–237.

[8] Victor J Dzau and Celynne A Balatbat. 2018. Health and societal implications of medical and technological advances. *Science translational medicine* 10, 463 (2018), 463.

[9] David J Eveson, Thompson G Robinson, and John F Potter. 2007. Lisinopril for the treatment of hypertension within the first 24 hours of acute ischemic stroke and follow-up. *American J. Hypertension* 20, 3 (2007), 270–277.

[10] Gregg C Fonarow, R Scott Wright, Frederick A Spencer, Paul D Fredrick, Wei Dong, Nathan Every, William J French, National Registry of Myocardial Infarction 4 Investigators, et al. 2005. Effect of statin use within the first 24 hours of admission for acute myocardial infarction on early morbidity and mortality. *American J. Cardiology* 96, 5 (2005), 611–616.

[11] Matthew W. Gardner and Stephen R. Dorling. 1998. Artificial neural networks (the multilayer perceptron): a review of applications in the atmospheric sciences. *Atmospheric Environment* 32, 14-15 (1998), 2627–2636.

[12] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS 2010*. 249–256.

[13] David N Hager, Varshitha Tanykonda, Zeba Noorain, Sarina K Sahetya, Catherine E Simpson, Juan Felipe Lucena, and Dale M Needham. 2018. Hospital mortality prediction for intermediate care patients: assessing the generalizability of the Intermediate Care Unit Severity Score (IMCUSS). *J. Critical Care* 46 (2018), 94–98.

[14] Haibo He and Edwardo A. Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 9 (2009), 1263–1284.

[15] Shiyi He, Chang Xu, Tianyu Guo, Chao Xu, and Dacheng Tao. 2018. Reinforced multi-label image classification by exploring curriculum. In *AAAI 2018*. 3183–3190.

[16] Chen Jie, Tengfei Ma, and Xiao Cao. 2018. FastGCN: fast learning with graph convolutional networks via importance sampling. In *ICLR 2018*.

[17] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. 2016. MIMIC-III, a freely accessible critical care database. *Scientific data* 3 (2016), 160035.

[18] Haocheng Kao, Kaifu Tang, and Edward Y Chang. 2018. Context-aware symptom checking for disease diagnosis using hierarchical reinforcement learning. In *AAAI 2018*. 2305–2313.

[19] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR 2017*.

[20] Sven Kosub. 2019. A note on the triangle inequality for the Jaccard distance. *Pattern Recognition Letters* 120 (2019), 36–38.

[21] C Krittanawong. 2018. The rise of artificial intelligence and the uncertain future for physicians. *European J. Internal Medicine* 48 (2018), e13–e14.

[22] Hung Le, Truyen Tran, and Svetha Venkatesh. 2018. Dual memory neural computer for asynchronous two-view sequential learning. In *ACM SIGKDD 2018*. 1637–1645.

[23] Cheng Li, Bingyu Wang, Virgil Pavlu, and Javed A Aslam. 2016. Conditional bernoulli mixtures for multi-label classification. In *ICML 2016*. 2482–2491.

[24] Tengfei Ma, Cao Xiao, Jiayu Zhou, and Fei Wang. 2018. Drug similarity integration through attentive multi-view graph auto-encoders. In *IJCAI 2018*. 3477–3483.

[25] Chengsheng Mao, Liang Yao, and Yuan Luo. 2019. MedGCN: graph convolutional networks for multiple medical tasks. *CoRR* abs/1904.00326 (2019).

[26] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. In *NAACL-HLT 2018*. 1101–1111.

[27] Shamim Nemati, Mohammad M Ghassemi, and Gari D Clifford. 2016. Optimal medication dosing from suboptimal clinical examples: a deep reinforcement learning approach. In *EMBC 2016*. 2978–2981.

[29] John Ross Quinlan. 1986. Induction of decision trees. *Machine Learning* 1, 1 (1986), 81–106.

[30] Aniruddh Raghu, Matthieu Komorowski, and Sumeetpal Singh. 2018. Model-based reinforcement learning for sepsis treatment. *CoRR* abs/1811.09602 (2018).

[31] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier chains for multi-label classification. *Machine Learning* 85, 3 (2011), 333–359.

[] Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. 2019. RepeatNet: A Repeat Aware Neural Recommendation Machine for Session-Based Recommendation. In *AAAI 2019*. 4806–4813.

[] Pengjie Ren, Zhumin Chen, Zhaochun Ren, Furu Wei, Liqiang Nie, Jun Ma, and Maarten Rijke. 2018. Sentence Relations for Extractive Summarization with Deep Neural Networks. *ACM Trans. Inf. Syst.* 36, 4 (2018), 39:1–39:32.

[32] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *ESWC 2018*. 593–607.

[33] Junyuan Shang, Shenda Hong, Yuxi Zhou, Meng Wu, and Hongyan Li. 2018. Knowledge guided multi-instance multi-label learning via neural networks in medicines prediction. In *ACML 2018*. 831–846.

[34] Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. 2019. GAMENet: graph augmented memory networks for recommending medication combination. In *AAAI 2019*.

[35] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *J. Machine Learning Research* 15, 1 (2014), 1929–1958.

[37] Kai-Fu Tang, Hao-Cheng Kao, Chun-Nan Chou, and Edward Y Chang. 2016. Inquire and diagnose: neural symptom checking ensemble using deep reinforcement learning. In *NIPS 2016*.

[38] Nicholas P Tatonetti, Patrick Ye, Roxana Daneshjou, and Russ B Altman. 2012. Data-driven prediction of drug effects and interactions. *Science Translational Medicine* 4, 125 (2012), 125ra31–125ra31.

[39] Michel Tokic. 2010. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In *GCAAI 2010*.

[40] Grigorios Tsoumakas and Ioannis Katakis. 2007. Multi-label classification: an overview. *International J. Data Warehousing and Mining* 3, 3 (2007), 1–13.

[41] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. 2018. Personalized prescription for comorbidity. In *DASFAA 2018*. 3–19.

[42] Meng Wang, Mengyue Liu, Jun Liu, Sen Wang, Guodong Long, and Buyue Qian. 2017. Safe medicine recommendation via medical knowledge graph embedding. *CoRR* abs/1710.05980 (2017).

[43] T G Wolfsberg, P. Primakoff, D G Myles, and J M White. 1995. ADAM, a novel family of membrane proteins containing a disintegrin and metalloprotease domain: multipotential functions in cell-cell and cell-matrix interactions. *J. Cell Biology* 131, 2 (1995), 275–8.

[44] Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Pengtao Xie, and Eric Xing. 2018. Multimodal machine learning for automated ICD coding. *CoRR* abs/1810.13348 (2018).

[45] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. 2018. SGM: sequence generation model for multi-label classification. In *COLING 2018*. 3915–3926.

[46] Zhongliang Yang, Yongfeng Huang, Yiran Jiang, Yuxi Sun, Yu-Jin Zhang, and Pengcheng Luo. 2018. Clinical assistant diagnosis for electronic medical record based on convolutional neural network. *Scientific reports* 8, 1 (2018), 6329.

[47] Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. 2018. An end-to-end deep learning architecture for graph classification. In *AAAI 2018*. 4438–4445.

[48] Yutao Zhang, Robert Chen, Tang Jie, Walter F. Stewart, and Jimeng Sun. 2017. LEAP: learning to prescribe effective and safe treatment combinations for multimorbidity. In *ACM SIGKDD 2017*. 1315–1324.

[49] Hanzhong Zheng and Dejia Shi. 2018. Using a LSTM-RNN based deep learning framework for ICU mortality prediction. In *WISA 2018*. 60–67.

[50] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. 2018. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics* 34, 13 (2018), i457–i466.