UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

Coding Electronic Health Records with Adversarial Reinforcement Path Generation

Wang, S.; Ren, P.; Chen, Z.; Ren, Z.; Nie, J.-Y.; Ma, J.; de Rijke, M.

DOI
10.1145/3397271.3401135

Publication date
2020

Document Version
Author accepted manuscript

Published in
SIGIR '20

Link to publication

Citation for published version (APA):
Wang, S., Ren, P., Chen, Z., Ren, Z., Nie, J.-Y., Ma, J., & de Rijke, M. (2020). Coding Electronic Health Records with Adversarial Reinforcement Path Generation. In *SIGIR '20: proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval : July 25-30, 2020, virtual event, China* (pp. 801–810). Association for Computing Machinery. https://doi.org/10.1145/3397271.3401135

General rights
It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations
If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: https://uba.uva.nl/en/contact, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

UvA-DARE is a service provided by the library of the University of Amsterdam (https://dare.uva.nl)

Download date:10 Nov 2026

Coding Electronic Health Records with Adversarial Reinforcement Path Generation

Shanshan Wang[1] Pengjie Ren[2*] Zhumin Chen[1*] Zhaochun Ren[1]
Jian-Yun Nie[3] Jun Ma[1] Maarten de Rijke[2,4]
[1]Shandong University, Qingdao, China
[2]University of Amsterdam, Amsterdam, The Netherlands
[3]Université de Montréal, Montreal, Canada
[4]Ahold Delhaize, Zaandam, The Netherlands
wangshanshan5678@gmail.com,{p.ren,m.derijke}@uva.nl
{chenzhumin,zhaochun.re,majun}@sdu.edu.cn,nie@iro.umontreal.ca

ABSTRACT

Electronic Health Record (EHR) coding is the task of assigning one or more International Classification of Diseases (ICD) codes to every EHR. Most previous work either ignores the hierarchical nature of the ICD codes or only focuses on parent-child relations. Moreover, existing EHR coding methods predict ICD codes from the leaf level with the greatest ICD number and the most fine-grained categories, which makes it difficult for models to make correct decisions. In order to address these problems, we model EHR coding as a path generation task. For this approach, we need to address two main challenges: (1) How to model relations between EHR and ICD codes, and relations between ICD codes? (2) How to evaluate the quality of generated ICD paths in order to obtain a signal that can be used to supervise the learning? We propose a coarse-to-fine ICD path generation framework, named *Reinforcement Path Generation Network (RPGNet)*, that implements EHR coding with a Path Generator (PG) and a Path Discriminator (PD). We address challenge (1) by introducing a Path Message Passing (PMP) module in the PG to encode three types of relation: between EHRs and ICD codes, between parent-child ICD codes, and between sibling ICD codes. To address challenge (2), we propose a PD component that estimates the reward for each ICD code in a generated path. RPGNet is trained with Reinforcement Learning (RL) in an adversarial manner. Experiments on the MIMIC-III benchmark dataset show that RPGNet significantly outperforms state-of-the-art methods in terms of micro-averaged F1 and micro-averaged AUC.

CCS CONCEPTS

• **Information systems** → **Content analysis and feature selection**; • **Applied computing** → **Health care information systems**; • **Computing methodologies** → *Adversarial learning*; *Reinforcement learning*.

*Co-corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SIGIR '20, July 25–30, 2020, Virtual Event, China
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401135

KEYWORDS

EHR coding, Path generation, Adversarial reinforcement learning

ACM Reference Format:
Shanshan Wang, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jian-Yun Nie, Jun Ma, and Maarten de Rijke. 2020. Coding Electronic Health Records with Adversarial Reinforcement Path Generation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401135

1 INTRODUCTION

An Electronic Health Record (EHR) contains a variety of clinical patient information, such as medical history, vital signs, lab test results and clinical notes [23]. EHR coding aims to assign multiple International Classification of Diseases (ICD) codes to EHRs. ICD codes can be used in search, data mining, billing, epidemiology assessment, and quality control of health care providers [1, 5]. Fig. 1 shows an example of ICD codes organized in a hierarchical tree structure; as the level increases, the granularity of the ICD codes becomes finer; the ICD codes at level-4 (the leaf level) are the most fine-grained. Automatic EHR coding is increasingly attracting attention because manual coding is time-consuming and error-prone [15].

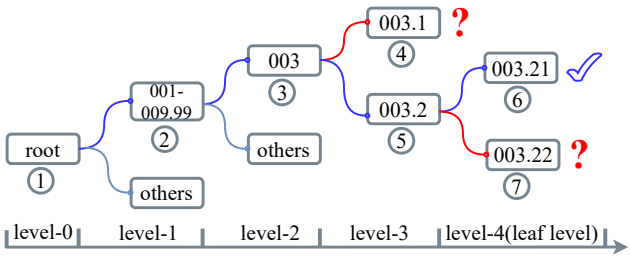

Figure 1: A schematic diagram of the ICD-9 taxonomy.

Most existing methods for EHR coding treat the task as a multi-label classification task on the leaf codes of the ICD taxonomy [19, 31, 40]. There are some downsides to this approach. First, the code space is high-dimensional, with over 18,000 codes in ICD-9-CM,[1] and the distribution of ICD codes is extremely unbalanced; most of the codes are seldom used in EHRs. Second, most classification

[1]https://www.cdc.gov/nchs/icd/icd9cm.htm

methods neglect relations between ICD codes, which is not conducive to ICD prediction. For example, in Fig. 1 "003.21" (ICD code of salmonella meningitis) and "003.22" (ICD code of salmonella pneumonia) share common characteristics (i.e., they are all caused by salmonella infections) and may be strongly related to each other. Third, few publications [24, 39] consider such relations and only focus on the relations between parent-child ICD codes.

To address the above issues, we propose to reformulate automatic EHR coding as a path generation task along the ICD tree. This new formulation generates ICD paths level by level. Intuitively, the number of ICD codes at lower levels is generally smaller than the number of leaf ICD codes; in other words, the candidate space of ICD codes is reduced. The generated ICD codes at coarse-grained levels (i.e., lower levels) can be used to facilitate the ICD generation at more fine-grained levels (i.e., higher levels). Moreover, each ICD code in a path can be seen as a prediction result for the EHR at a particular level, so the path generation method can yield multi-grained EHR coding.

However, there are two challenges to be addressed: (1) How to model the relations between EHR and ICD codes, and between ICD codes? There are at least three types of relation, i.e., relations between EHRs and the ICD hierarchy, relations between parent and child ICD codes, and relations between sibling ICD codes. (2) How to evaluate the quality of the generated ICD paths in order to define a supervision signal for learning? In many cases, the generated path is partially correct (i.e., the path ① → ② → ③ → ④ and the path ① → ② → ③ → ⑤ → ⑦ in Fig. 1). For example, the agent gets the correct path at level-1 and level-2, but makes wrong decisions at level-3 because of finer ICD granularity at level-3, which corresponds to the path ① → ② → ③ → ④ in Fig. 1. In addition, the classification granularity at each level is different, and the rewards obtained at each level should probably also be different. Defining rewards as a signal to learn from remains a challenge [10].

In this paper, we address the issues and challenges listed above by proposing a coarse-to-fine ICD path generation framework, namely Reinforcement Path Generation Network (RPGNet), for automatic EHR coding. RPGNet is a *coarse-to-fine* path generation framework because it generates ICD codes from lower levels to higher levels in the ICD hierarchy. RPGNet solves the EHR coding task by taking an EHR as input and returning a set of paths whose final nodes are codes to be assigned to the record. RPGNet contains two main components: a Path Generator (PG) and a Path Discriminator (PD). The PG consists of an EHR Encoder and a Path Message Passing (PMP) module. Given an EHR, the EHR Encoder is used to obtain latent EHR representations. We implement the EHR Encoder with a multi-channel Convolutional Neural Network (CNN). Then, the PMP module is introduced to model relations between EHRs and ICD codes, and between ICD codes to obtain representations of the relations. Specifically, PMP addresses challenge (1) by propagating information between an EHR and related ICD codes. Finally, a hybrid policy network is exploited to generate ICD paths for a given EHR based on the relation representations. To address challenge (2) and train RPGNet, we design a PD to evaluate the intermediate reward for each ICD code in a path. RPGNet is trained with Reinforcement Learning (RL) in an adversarial learning fashion [13]. Specifically, the PG is meant to generate paths that are indistinguishable from positive paths, while the PD is meant to distinguish positive paths and generated paths.

We conduct extensive experiments on the MIMIC-III benchmark dataset [17] to obtain empirical evidence for the effectiveness of RPGNet. Our experimental results demonstrate that RPGNet significantly outperforms state-of-the-art methods by a large margin. We also conduct a number of analyses to show how RPGNet performs w.r.t. the two challenges listed above.

To sum up, the contributions of this work are as follows:

- We formulate automatic EHR coding as a path generation task and propose a reinforcement path generation framework, RPGNet, for coarse-to-fine EHR coding.
- We propose a Path Message Passing module that models the relations between EHRs and ICD codes, and between ICD codes by allowing information to spread between them.
- We devise a Path Discriminator with an adversarial reward learning mechanism to evaluate the intermediate rewards as supervision signals for learning RPGNet.
- We carry out extensive experiments on a benchmark dataset to verify and analyze the effectiveness of RPGNet.

2 RELATED WORK

2.1 Automatic EHR coding

In recent years, the automatic EHR coding task has been studied in a large number of publications, in information retrieval, machine learning, and healthcare. Kavuluru et al. [19] treat the task as a multi-label classification problem and develop a label ranking method based on features selected from the EHR text. Shi et al. [31] use a neural architecture with LSTM and attention mechanism, which takes diagnosis descriptions as input to predict ICD codes. Wang et al. [35] propose a model that can jointly capture word and label embeddings and exploit the cosine similarity between them to predict ICD codes. Sadoughi et al. [30] solve the task as a multi-task classifier problem and exploit the description of ICD codes for regularizing the attention for each individual classifier. Xu et al. [40] investigate several separate machine learning models to handle EHR data from different modalities, and then employ an ensemble method to integrate all modality-specific models to predict ICD codes. None of the above methods takes the hierarchical relations between ICD codes into account, which is problematic because ICD codes are organized in a hierarchical structure.

There have been some methods that attempt to encode the hierarchical structure of ICD codes. Perotte et al. [27] utilize "flat" and "hierarchical" SVMs based on tf-idf document features for EHR coding; the former treats each code as an individual prediction, while the latter exploits the ICD code ontology for hierarchical classification. Mullenbach et al. [24] adopt a CNN-based model with a per-label attention mechanism to assign ICD codes based on free-text descriptions. In addition, they use a Graph Neural Network (GNN) [42] to capture the parent-child relations between ICD codes and obtain a representation of each ICD code. Similarly, Xie et al. [39] leverage GNNs to encode the ICD hierarchy and incorporate multi-scale feature attention for EHR coding.

The differences between the work listed above and our work are at least two-fold. First, no previous work formulates automatic EHR coding as a path generation task, which is a more appropriate

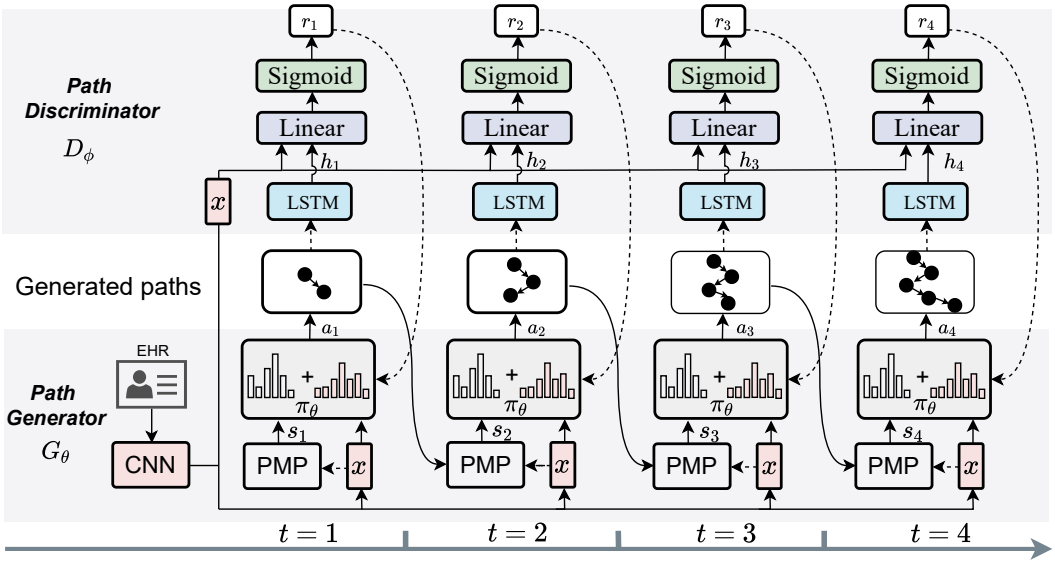

Figure 2: Overview of RPGNet. RPGNet contains two major components: a path generator G_θ and a path discriminator D_ϕ. The PMP in G_θ represents path message passing module. The input of RPGNet is an EHR and the output consists of coarse-to-fine ICD paths.

modeling choice considering how ICD codes are organized. Second, although some previous work has proposed to model relations between ICD codes, only parent-child relations have so far been considered while relations between sibling codes have been ignored: we model different kinds of relation in a unified framework.

2.2 Reinforcement learning for healthcare

To the best of our knowledge, there is no prior work that uses RL work for automatic EHR coding, so we survey related work from a broader healthcare perspective. Tang et al. [33] use RL to create an effective and efficient symptom checker to predict diseases by asking patient questions. Nemati et al. [25] leverage a combination of Hidden Markov Models and deep Q-networks to predict optimal heparin dosing for ICU patients. Besson et al. [4] 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 so as to improve diagnosis accuracy. Wang et al. [36] propose a graph convolutional RL model for medicine combination prediction by learning correlative and adverse interactions between medicines.

In this work, we study how to apply RL to automatic EHR coding where we need to generate coarse-to-fine paths of ICD codes. And unlike the work listed above, rather than setting rewards manually, we exploit adversarial learning to estimate rewards automatically by determining the authenticity of the paths.

2.3 Adversarial networks for healthcare

The idea of Generative Adversarial Networks (GANs) has achieved great success in many domains, especially in computer vision. The training process is formalized as a game in which the generative model is trained to generate outputs to fool the discriminator; the discriminator judges the output of the generator and directs the generator to produce indistinguishable results. In healthcare, several GAN-based models have been proposed to handle tabular data in EHRs. RGAN and RCGAN [8] can generate real-valued time-series data. Choi et al. [6] and Guan et al. [12] have proposed medGAN and mtGAN, respectively, to generate realistic synthetic patient records. Yu et al. [41] use GANs in a semi-supervised learning setting to improve the detection performance on rare diseases.

Unlike previous work, we make use of the idea of adversarial learning in GAN and combine it with RL to explore ICD code paths and estimate intermediate rewards for each code in a path. The discriminator is used to determine the authenticity of a path explored by our model; its results are considered as intermediate rewards for each code in a path to guide the learning of our model.

3 METHOD

Given an EHR x in free text, i.e., $x = [w_1, w_2, \ldots, w_i, \ldots]$, where w_i denotes a token, the task of EHR coding with ICD paths is to generate several optimal ICD paths $Y = [y^1, y^2, \ldots, y^j, \ldots]$, where $y^j = [c_1, c_2, \ldots, c_t, \ldots]$ is the j-th ICD path and $c_t \in C$ is the t-th ICD code in the path, where C represents all candidate ICD codes. Note that all paths start with the root code c_1 in the ICD tree taxonomy. Without loss of generality, we omit the notation j in the following sections. Next, we describe the proposed Reinforcement Path Generation Network (RPGNet) in detail.

3.1 Overview

RPGNet consists of two components: a path generator G_θ and a path discriminator D_ϕ, as shown in Figure 2.

The path generator G_θ, parameterized by θ, generates ICD paths given an EHR. G_θ contains two modules: an EHR Encoder and a *Path Message Passing (PMP)* module. The EHR Encoder is used to obtain the representation of each EHR; the PMP module is used to merge information from the EHR, the current ICD code, and the candidate ICD codes. The information obtained from the PMP

module determines the initial state $\boldsymbol{s}_t$. A hybrid policy network is designed to generate ICD path y_t till timestamp t along the ICD tree taxonomy. The path y_t is then integrated with the EHR information to get a new state $\boldsymbol{s}_{t+1}$ through the PMP module. This process is repeated several times until the path reaches leaf ICD codes.

The path discriminator D_ϕ, parameterized by ϕ, is a binary classifier that takes an EHR x and an ICD path y_t till timestamp t as inputs and outputs a probability r_t indicating whether the current path is a ground truth path or generated by G_θ. Specifically, the generated path y_t till timestamp t is first encoded into a vector representation $\boldsymbol{h}_t$ using an Long Short Term Memory (LSTM), and concatenates with an EHR representation $\boldsymbol{x}$. The concatenated representation is then fed into a fully-connected layer with sigmoid activation function to get r_t, which indicates the probability of y_t being a part of a ground truth path. r_t is considered as a reward to guide the learning of the path generator G_θ.

3.2 Markov Decision Process formulation

We model the above process as a Markov Decision Process (MDP) [3] $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \gamma \rangle$, where $\mathcal{S}$ is a continuous state space, $\mathcal{A}$ is the set of all available actions, $\mathcal{T}$ is the state transition function, $\mathcal{R}$ is the reward function of each (state, action) pair, and γ is the discount factor. Next, we introduce how we model $\mathcal{S}, \mathcal{A}, \mathcal{T}$, and $\mathcal{R}$ in detail.

- **State $\mathcal{S}$.** The state $\mathcal{S}$ contains EHR information, current ICD information, and children ICD information.
- **Action $\mathcal{A}$.** $\mathcal{A}$ is organized in a hierarchical structure consistent with the ICD taxonomy, where the action space is different for different steps when generating an ICD path. For example, if the agent reaches ICD code a_{t-1} whose level is $t-1$, the candidate actions at timestamp t are all ICD codes at level t.
- **Transition $\mathcal{T}$.** $\mathcal{T}$ represents the transition function from current state to the next state. In our case, the transition function is deterministic, which means that the next state $\boldsymbol{s}_{t+1}$ is not stochastic and only depends on the current state $\boldsymbol{s}_t$ and action a_t.
- **Reward $\mathcal{R}$.** The reward $r_t \in \mathcal{R}$ indicates how good the selected ICD code is at timestamp t which is evaluated automatically by the path discriminator D_ϕ.

In order to encourage the path generator to generate correct ICD paths that are indistinguishable from ground truth paths, we propose to maximize the expected rewards of the path generator using the reinforce algorithm [37]:

$$\begin{aligned} J(\theta) &= \mathbb{E}_{a\sim\pi(a|\boldsymbol{s},\boldsymbol{x};\theta)}\left(\sum_t R_{(\boldsymbol{s}_t,\boldsymbol{x}),a_t}\right) \\ &= \sum_t \sum_{a_t\in\mathcal{A}} \pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta) R_{(\boldsymbol{s}_t,\boldsymbol{x}),a_t}, \end{aligned} \tag{1}$$

where $J(\theta)$ is the expected total rewards for one episode; $\pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta)$ is the hybrid policy network of the path generator G_θ, which maps the state vector $\boldsymbol{s}_t$ and EHR representation $\boldsymbol{x}$ to a stochastic policy a_t. a_t is the generated ICD code based on the current state $\boldsymbol{s}_t$ and EHR representation $\boldsymbol{x}$. $R_{(\boldsymbol{s}_t,\boldsymbol{x}),a_t}$ indicates the reward value the agent receives for executing a_t, which is implemented with the path discriminator D_ϕ. We can update θ with the policy gradient as follows:

$$\nabla J(\theta) = \sum_t \sum_{a_t\in\mathcal{A}} \pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta) \nabla_\theta \log \pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta). \tag{2}$$

Next, we will show how to model $\pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta)$ and $R_{(\boldsymbol{s}_t,\boldsymbol{x}),a_t}$ in the path generator G_θ and path discriminator D_ϕ, respectively.

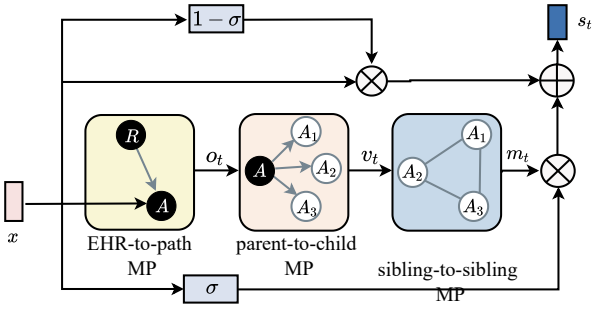

Figure 3: Path Message Passing (PMP) module. First, the message is propagated between EHR $\boldsymbol{x}$ and ICD path $R \longrightarrow A$. Second, the relation representation $\boldsymbol{o}_t$ is spread to children (i.e., A_1, A_2 and A_3) to get the relation representation $\boldsymbol{v}_t$ between parent code A and each child A_i. Third, the $\boldsymbol{v}_t$ passes between siblings (i.e., A_1, A_2 and A_3) to get the final relation representation $\boldsymbol{m}_t$. Lastly, after passing through the two gates σ and $1-\sigma$, the relation representation $\boldsymbol{m}_t$ and EHR representation $\boldsymbol{x}$ are added together as state s_t.

3.3 Path generator G_θ

We model $\pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta)$ with a hybrid policy network that consists of two policies, a local policy π_l and a global policy π_g:

$$\pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta) = \lambda \pi_l(a_t \mid \boldsymbol{s}_t; \theta_l) + (1-\lambda)\pi_g(a_t \mid \boldsymbol{x}; \theta_g). \tag{3}$$

Here, λ is a trade-off factor to balance the two policies. The local policy π_l decides the next ICD code from child ICD codes of the last action a_{t-1} based on state $\boldsymbol{s}_t$ to extend the path, while the global policy π_g decides the ICD code from all the ICD codes at the t-th level of the ICD taxonomy (not only the child codes of a_{t-1}). The intuition behind the global policy is that the local policy only selects the next ICD code from child ICD codes of the previous ICD code, which easily causes error accumulation, i.e., if the agent makes a wrong decision at lower levels, all subsequent decisions are wrong. The global policy alleviates the error accumulation by allowing the agent to decide the next ICD code from all t-th level child codes so as to correct its decisions.

The global policy uses a fully-connected layer to directly map the EHR representation $\boldsymbol{x}$ to the probability distribution over ICD codes at the corresponding level. Another fully-connected layer is used by the local policy to map the state $\boldsymbol{s}_t$ to the probability distribution over child ICD codes of the last action. The two probability distributions are weighted together to obtain the final probability distribution over all candidate ICD codes:

$$\pi(a_t \mid \boldsymbol{s}_t, \boldsymbol{x}; \theta) = \lambda \sigma(\boldsymbol{W}_l(\boldsymbol{s}_t) + \boldsymbol{b}_l) + (1-\lambda)\sigma(\boldsymbol{W}_g(\boldsymbol{x}) + \boldsymbol{b}_g), \tag{4}$$

where $\boldsymbol{W}_l$ and $\boldsymbol{W}_g$ are weight matrices; $\boldsymbol{b}_l$ and $\boldsymbol{b}_g$ are bias terms; and σ is the sigmoid activation function.

3.3.1 Path Message Passing. Inspired by [21, 32], we propose a PMP module to encode the state $\boldsymbol{s}_t$ by taking into account the relation between an EHR and the hierarchical ICD structure, parent-child relations, and sibling relations of ICD codes, as shown in Figure 3. Formally, $\boldsymbol{s}_t$ is defined as:

$$\boldsymbol{s}_t = \boldsymbol{m}_t \odot T(\boldsymbol{x}, \boldsymbol{W}_T) + \boldsymbol{x} \odot (1 - T(\boldsymbol{x}, \boldsymbol{W}_T)). \tag{5}$$

where $\boldsymbol{W}_T$ is the weight matrix; T is a gate that is used to control the information transforming from the EHR representation $\boldsymbol{x}$ and $\boldsymbol{m}_t$, respectively; $\boldsymbol{m}_t$ represents the relation representation obtained by three steps of message passing: EHR-to-path message passing, parent-to-child message passing, and sibling-to-sibling message passing, as shown in Figure 3. Next, we introduce the modeling of these three steps one by one.

Step 1: EHR-to-path message passing. This step is used to integrate the EHR representation and ICD path representation. The relation representation between an EHR and an ICD path is denoted as $\boldsymbol{o}_t$, which can be obtained as follows:

$$\boldsymbol{o}_t = \tanh(\boldsymbol{W}_o(\boldsymbol{u}_t)), \tag{6}$$

where $\boldsymbol{W}_o$ is the weight matrix and $\boldsymbol{u}_t$ is obtained through various transformations of EHR representation $\boldsymbol{x}$ and path representation $\boldsymbol{c}_t$. Here, $\boldsymbol{u}_t$ is computed as follows:

$$\boldsymbol{u}_t = \boldsymbol{x} \oplus \boldsymbol{c}_t \oplus (\boldsymbol{x} \odot \boldsymbol{c}_t) \oplus (\boldsymbol{x} + \boldsymbol{c}_t) \oplus (\boldsymbol{x} - \boldsymbol{c}_t) \oplus (\boldsymbol{c}_t - \boldsymbol{x}), \tag{7}$$

where $\oplus$ indicates vector concatenation, $\odot$ indicates the element-wise product, and $\boldsymbol{c}_t$ is the representation of selected ICD code at timestamp t (i.e., the representation of a_t).

Step 2: Parent-to-child message passing. This step is used to capture the relation between parent and child ICD codes of ICD code a_t. After obtaining relation representation $\boldsymbol{o}_t$ between an EHR and an ICD path, we propagate this relation representation from parent code to all its child codes to generate relation representation $\boldsymbol{v}_t$:

$$\boldsymbol{v}_t = \boldsymbol{o}_t \odot \boldsymbol{c}_t^c, \tag{8}$$

where $\odot$ indicates the element-wise product operation, and $\boldsymbol{c}_t^c$ is the vector representation of each child ICD.

Step 3: Sibling-to-sibling message passing. This step is used to capture the relation between sibling ICD codes by spreading information between sibling ICD codes. Specifically, we achieve this by passing information between sibling ICD codes using intra-attention (also called self-attention [34]) neural message passing, which enables ICD codes to attend over their sibling ICD codes differently. This allows for the network to learn different degrees of importance for different sibling ICD codes. Sibling-to-sibling message passing is formulated as follows:

$$\boldsymbol{m}_t = \boldsymbol{v}_t + \sum_{j \in N_{v_t}} M_{attn}(\boldsymbol{v}_t, \boldsymbol{v}_t^j), \tag{9}$$

where M_{attn} is the attention function; N_{v_t} indicates all the sibling ICD codes of v_t, and v_t^j represents the j-th sibling ICD of code v_t. M_{attn} is used to pass the message from the j-th sibling ICD code to the current ICD code v_t using the learned attention weights $\boldsymbol{\alpha}_t^j$:

$$M_{attn}(\boldsymbol{v}_t, \boldsymbol{v}_t^j) = \boldsymbol{\alpha}_t^j \boldsymbol{v}_t^j. \tag{10}$$

The attention weights $\boldsymbol{\alpha}_t^j$ for a pair of sibling codes $(\boldsymbol{v}_t, \boldsymbol{v}_t^j)$ can be calculated using the attention function $a(\cdot)$:

$$\boldsymbol{\alpha}_t^j = \text{softmax}_j(\boldsymbol{e}_t^j) = \frac{\boldsymbol{e}_t^j}{\sum_{k \in N(v_t)} \exp(\boldsymbol{e}_t^k)}, \tag{11}$$

where $\boldsymbol{e}_t^j$ represents the importance of the j-th sibling code for the current code before normalization; the $\boldsymbol{e}_t^j$ are normalized across all sibling nodes of the current ICD code using a softmax function (Eq. 11) to get $\boldsymbol{\alpha}_t^j$. For the attention function $a(\cdot)$, we use a dot product with linear transformations $\boldsymbol{W}_q$ on code $\boldsymbol{v}_t$ and $\boldsymbol{W}_u$ on $\boldsymbol{v}_t^j$:

$$e_t^j = a(\boldsymbol{v}_t, \boldsymbol{v}_t^j) \tag{12}$$

$$a(v_t, v_t^j) = (\boldsymbol{W}_q \boldsymbol{v}_t)^\top (\boldsymbol{W}_u \boldsymbol{v}_t^j). \tag{13}$$

3.3.2 EHR encoder. This module applies a multi-channel CNN [20] to encode each EHR to a vector representation $\boldsymbol{x}$. Let $\boldsymbol{w}_i$ be the d-dimensional token vector corresponding to the i-th token in the EHR. An EHR of length n is represented as:

$$\boldsymbol{w}_{1:n} = \boldsymbol{w}_1 \oplus \boldsymbol{w}_2 \oplus \cdots \oplus \boldsymbol{w}_n, \tag{14}$$

where $\oplus$ is the concatenation operator. A convolution operation involves a filter $\boldsymbol{W}_f \in \mathbb{R}^{lk}$, which is applied to a window of l words to produce a new feature. For example, a feature z_i is generated from a window of words $\boldsymbol{w}_{i:i+l-1}$ by

$$z_i = relu(\boldsymbol{W}_f \circledast \boldsymbol{w}_{i:i+l-1} + b), \tag{15}$$

where $\circledast$ indicates the convolution operation, $b \in \mathbb{R}^k$ is a bias term and $relu$ is a non-linear function. This filter is applied to each possible window in the EHR to produce a feature map $\boldsymbol{z}$:

$$\boldsymbol{z} = z_1 \oplus z_2 \oplus \cdots \oplus z_{n-h+1}. \tag{16}$$

Then we apply a max-over-time pooling operation [7] over the feature map and take the maximum value as the feature $\boldsymbol{z}$ corresponding to this particular filter:

$$\hat{z} = \max(\boldsymbol{z}). \tag{17}$$

This module uses multiple filters (with varying window sizes) to obtain multiple features. These feature are concatenated together as a vector representation $\boldsymbol{x}$ of an EHR as follows:

$$\boldsymbol{x} = \hat{z_1} \oplus \hat{z_2} \oplus \cdots \oplus \hat{z_k}, \tag{18}$$

where $\hat{z_k}$ represents the k-th feature map obtained by the k-th filter.

3.4 Path discriminator D_ϕ

We design the path discriminator module D_ϕ to get the reward r_t for each code in the generated path y_t until timestamp t. Specifically, we model r_t as the discrimination probability as follows:

$$r_t = R_{(s_t, x), a_t} = p_D(y_t, \boldsymbol{x}) = \sigma(\boldsymbol{W}_r(\boldsymbol{h}_t \oplus \boldsymbol{x})), \tag{19}$$

where $\oplus$ denotes the concatenation operation, and $\boldsymbol{W}_r$ is the weight matrix; $\boldsymbol{h}_t$ is the representation of the current generated path y_t, which is obtained by recurrently applying an LSTM to the ICD code path $y_t = (c_1, c_2, \ldots, c_k, \ldots, c_t)$:

$$\boldsymbol{h}_k = LSTM(\boldsymbol{h}_{k-1}, \boldsymbol{c}_k), \tag{20}$$

where $\boldsymbol{h}_{k-1}$ is the hidden vector at timestamp $k-1$; $\boldsymbol{c}_k$ is the k-th ICD code representation.

To learn the path discriminator D_ϕ, we adapt an adversarial training schema, where we consider the paths generated from the path generator G_θ as negative samples and the ground truth paths as positive samples. The objective of path discriminator D_ϕ can be formulated as minimizing the following cross-entropy function:

$$L_D = -\sum_{(y_t,\boldsymbol{x})\in Q^+} \log p_D(y_t,\boldsymbol{x}) - \sum_{(y_t,\boldsymbol{x})\in Q^-} \log(1-p_D(y_t,\boldsymbol{x})), \quad (21)$$

where Q^+ and Q^- denote positive and negative samples, respectively; $p_D(y_t,\boldsymbol{x})$ represents the probability of sample $(y_t,\boldsymbol{x})$ belonging to a positive sample.

The update of path discriminator D_ϕ is identical to the common binary classification problem, which can be optimized by any gradient-based algorithm.

3.5 Adversarial reinforcement learning

The alternating training process of the path generator G_θ and the path discriminator D_ϕ is shown in Algorithm 1. After randomly initializing G_θ and D_ϕ, we use ground truth paths to pre-train G_θ (line 1–2). The training times for G_θ and D_ϕ in each epoch are M and N respectively (line 4 and line 12). For training G_θ, we generate a single path y for each EHR x in the batch EHRs (line 6). Then we calculate the reward for each ICD code in the generated path y (line 7). We use batch EHRs and generated paths to update the generator G_θ with policy gradient (line 9). To make the exploration of G_θ more effective, we train G_θ on batch EHRs and ground truth paths (line 10). For training D_ϕ, we generate a single path y for each EHR x in the batch EHRs (line 14) and also sample a ground truth path for this EHR (line 15). Then, we train the path discriminator D_θ with generated paths and ground truth paths (line 17).

Algorithm 1: Adversarial reinforcement learning algorithm for training RPGNet.

1 Initialize G_θ, D_ϕ with random weights θ, ϕ;
2 Pre-train G_θ on dataset D with teacher forcing using the ground truth paths;
3 **for** *epoch = 1: EPOCHS* **do**
4 **for** *each* $j = 1, 2, \ldots, M$ **do**
5 **for** *each EHR* x *in batch EHRs* **do**
6 Sample a path for EHR x from G_θ, $y \sim G_\theta$;
7 Compute rewards for each code in y with Eq. 19;
8 **end**
9 Update generator G_θ via policy gradient with Eq. 2;
10 Train G_θ on batch EHRs and ground truth paths with teacher forcing;
11 **end**
12 **for** *each* $j = 1, 2, \ldots, N$ **do**
13 **for** *each EHR* x *in batch EHRs* **do**
14 Sample a path for EHR x from G_θ;
15 Sample a ground truth path for EHR x from D;
16 **end**
17 Train path discriminator D_ϕ with Eq. 21;
18 **end**
19 **end**

4 EXPERIMENTAL SETUP

We seek to answer the following research questions: (RQ1) What is the performance of RPGNet on the EHR coding task? Does it outperform state-of-the-art methods? (RQ2) How does RPGNet perform at different ICD levels of the ICD taxonomy? (RQ3) Where do the improvements of RPGNet come from? What are the effects of different components? (RQ4) How does the trade-off coefficient (λ) between the local and global policy influence the performance?

4.1 Dataset

We conduct experiments on a real-world dataset: the MIMIC-III dataset,[2] which is a large, freely-available database comprising de-identified health-related data associated with over forty thousand patients between 2001 and 2012. This is the only benchmark dataset that is commonly used on this task and publicly available [17, 39].

As with previous studies [e.g., 24, 28, 29], we focus on discharge summaries in EHR, which summarize the information about a stay into a single document. We clean discharge summaries by removing noisy information, such as doctors' information and hospital information. There are two experimental settings for comparison:

(1) Top-50 label setting: We only predict the 50 most frequent ICD codes, and filter each dataset down to the instances that have at least one of the top 50 most frequent codes; and
(2) Full-label setting: We keep all diagnosis ICD codes appearing in discharge summaries.

We randomly divide the MIMIC-III dataset into training, validation and test sets with 4:1:1 ratios for the above two different settings.

Table 1: Statistics of the MIMIC-III dataset.

Statistics	Top-50 labels	Full-label
# discharge summaries	49,354	52,722
# total ICD codes	122	9,219
# ICD codes in level-1	23	155
# ICD codes in level-2	38	1,098
# ICD codes in level-3	47	4,475
# ICD codes in level-4	50	6,918
# avg ICD codes per EHR in level-1	4.05	8.56
# avg ICD codes per EHR in level-2	4.55	10.78
# avg ICD codes per EHR in level-3	4.30	11.13
# avg ICD codes per EHR in level-4	1.69	5.36

Table 1 shows the statistics of two dataset settings, respectively. From the statistics we can see that: (1) In the MIMIC-III dataset, the number of ICD codes at different levels varies greatly. Generally, the number of ICD codes at coarse-grained levels is much smaller than that at fine-grained levels. This is why we believe that using the ICD hierarchy structure to explore ICD paths from lower levels to higher levels can effectively reduce the candidate ICD space, and the difficulty of model learning. (2) The average number of ICD codes per EHR is quite small compared to the total number of ICD codes in the hierarchical structure. This is why we think it is critical to integrate the relations of ICD codes between different levels and same levels to reduce the influence of non-relevant ICD information when predicting multiple ICD codes for each EHR.

[2]The dataset used in this paper is available at https://mimic.physionet.org/

4.2 Baselines

In order to demonstrate the effectiveness of RPGNet, we compare it with several methods, including state-of-the-art models for EHR coding and a hierarchical text classification method:

- **Flat SVMs** [27]. This method uses 10,000 tf-idf unigram features to train multiple binary SVMs for EHR coding.
- **Multi-layer Percepetion (MLP)**. We learn a multi-label classification model with a three-layer perceptron to predict the probability of each ICD code.
- **BI-GRU** [43]. This method uses a bidirectional gated recurrent unit to encode EHR and then performs binary classification on each ICD code based on the EHR representation.
- **HA-GRU** [2]. HA-GRU uses a hierarchical attention Bidirectional Gate Recurrent Unit (BI-GRU) to encode EHR by identifying the most relevant sentences in EHR w.r.t. each ICD code.
- **CAML** and **DR-CAML** [24]. CAML exploits Text-CNN [20] to obtain a representation of each EHR and then uses label-dependent attention to learn the most informative representation for each ICD code, based on which it does binary classification. DR-CAML enhances CAML by adding an ICD description regularization term to the final classification weights.
- **MSATT-KG** [39]. MSATT-KG leverages a densely connected convolutional neural network to produce variable n-gram features for clinical notes and incorporates multi-scale feature attention to adaptively select multi-scale features. The multi-scale features are used to perform multi-label classification over all the ICD codes. It also leverages graph convolutional neural networks to capture the hierarchical structure of the ICD taxonomy.
- **HARNN** [16]. This method was initially proposed for multi-label text classification. It also takes the hierarchical structure of the taxonomy into account by integrating text with the hierarchical category structure through an hierarchical attention-based recurrent layer. We apply HARNN to this task because the ICD taxonomy has a similar hierarchical structure.

4.3 Evaluation metrics

We report a variety of metrics that are commonly used by previous studies on this task [24, 28, 39], including micro-averaged and macro-averaged metrics. Micro-averaged metrics are calculated by treating each sample as a separate prediction, while macro-averaged metrics are calculated by averaging metrics computed per-label. The macro-averaged metrics pay more attention to rare label prediction. Specifically, we use Precision, Recall, F1 and Area Under Curve (AUC) as evaluation metrics [11, 14]. F1 score is the harmonic mean of Precision and Recall, and AUC summarizes performances under different thresholds. We also use the Jaccard Similarity Coefficient [26] to measure the overlap between two sets, which are prediction results by EHR coding methods and ground truth ICD codes. It is defined as $\mathit{Jaccard} = \frac{1}{m}\sum_i^m |Y_i \cap \hat{Y}_i|/|Y_i \cup \hat{Y}_i|$, where m indicates the number of instances of the dataset, Y_i is the prediction result by different EHR coding methods, and $\hat{Y}_i$ is the ground truth ICD set.

4.4 Implementation details

The local policy and global policy both use 300 hidden units for the fully connected layers (Eq. 3). In the EHR encoder module, the $\boldsymbol{w}_i$ (Eq. 14) is a 100-dimensional vector that is randomly initialized. We use 3 convolutional layers, and the filter sizes are all set to 100. The kernel size of each layer is set to 3, 4, 5 respectively. After the last convolutional layer, we apply dropout with a drop ratio of 0.5 (Eq. 18). In the PMP module, the hidden size of the transform gate T is set to 300 (Eq. 5). The embedding size of each ICD code is set to 100 (Eq. 7 and Eq. 8). The hidden sizes in the transformations W_q and W_u are set to 300 and 500, respectively (Eq. 13). During training, the number of epochs is limited to 200. The number of training iterations M and N for path generator G_θ and path discriminator D_ϕ at each epoch are both set to 1 (Algorithm 1). We initialize model parameters randomly and use a batch size of 32. The Adam optimizer [38] is used to optimize all parameters. The learning rate $\alpha = 0.0001$ and the momentum parameters are set to default $\beta 1 = 0.9$ and $\beta 2 = 0.999$. We implement RPGNet in PyTorch and train it on a GeForce GTX TitanX GPU.

5 RESULT AND ANALYSIS

5.1 Overall performance (RQ1)

To answer RQ1, we report the evaluation metrics for both the top-50 label setting and the full-label setting in Table 2 and 3, respectively. We first analyze the results on the top-50 label setting, which is mostly adopted in the literature [24, 28, 31]. From Table 2, we have the following observations:

First, RPGNet achieves the best performance on most of the evaluation metrics. This indicates that RPGNet is able to effectively perform EHR coding by exploiting the hierarchical ICD structure. The micro Recall value gets the most significant improvement, i.e., 14.5% over the best baseline HA-GRU. The reason is that RPGNet explores ICD paths through adversarial reinforcement learning, which helps to discover rare ICD paths, which in turn increases coverage of ICD codes.

Second, CAML, DR-CAML, and MSATT-KG all have higher Precision values and lower Recall values compared to RPGNet. CAML achieves the best micro Precision with even 18.45% higher than RPGNet. However, its micro and macro Recall values are much lower (20.63% and 14.76% lower) compared to RPGNet. Lower Recall values of CAML, DR-CAML and MSATT-KG indicate that they miss a lot of correct ICD codes, which is a more severe problem because a higher coverage of correct ICD codes, especially the rare ones, is more important in practice. In addition, the F1 values and AUC values of RPGNet are better than those of CAML, DR-CAML, and MSATT-KG.

Third, generally RNN-based models (i.e., BI-GRU, HA-GRU, and HARNN) have worse performance compared to the other ones. This is because an EHR is a long sequence, which easily gives rise to vanishing gradient problems with RNN-based models [22]. This also suggests that sequence information for words is not as important as it is in natural language models [39]. For EHR coding, the keywords and phrases are more important and this type of information can already be better captured by CNN-based models. Especially, HARNN performs the worst with a macro F1 value of 2.13% and a micro F1 value of 15.84%. Although HARNN considers

Table 2: Results (%) on the MIMIC-III dataset for the top-50 label setting. Bold face indicates the best result in terms of the corresponding metric. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.01$). The results for some baselines do not match the results published by MSATT-KG[39] as (1) we use different data preprocessing and different data scale, and (2) we use a different split of training, validation and test sets.

		Precision		Recall		F1		AUC	
Method	**Jaccard**	micro	macro	micro	macro	micro	macro	micro	macro
Flat SVMs	9.21	36.77	13.66	11.64	4.75	17.68	5.93	54.79	51.18
MLP	17.01	36.86	16.95	21.20	12.09	26.86	12.70	58.75	53.96
BI-GRU	16.05	28.93	2.39	25.03	8.02	26.76	3.54	59.38	50.01
HA-GRU	6.55	7.29	3.94	42.31	**54.00**	12.44	7.20	43.56	50.00
CAML	30.43	**59.65**	51.03	36.18	29.75	45.04	34.71	66.84	63.47
DR-CAML	31.82	59.09	48.94	37.93	28.90	46.21	32.49	67.63	62.90
MSATT-KG	31.35	57.68	**51.19**	37.92	30.10	45.76	35.01	67.54	63.41
HARNN	11.35	39.70	4.31	9.89	2.03	15.84	2.13	54.72	52.40
RPGNet	**33.34***	41.20	33.59	**56.81**	44.51	**48.35**	**35.40***	**75.14***	**70.50***

Table 3: Results (%) on the MIMIC-III dataset for the full-label setting. Bold face indicates the best result in terms of the corresponding metric. Significant improvements over the best baseline results are marked with * (t-test, $p < 0.01$).

		F1		AUC	
Method	**Jaccard**	micro	macro	micro	macro
Flat SVMs	2.71	4.43	0.08	51.16	50.04
MLP	10.81	18.19	0.11	56.58	51.19
BI-GRU	8.51	14.02	0.02	54.70	50.00
HA-GRU	0.18	0.35	0.17	50.86	50.00
CAML	13.69	21.87	**1.63**	56.67	50.97
DR-CAML	13.34	21.55	1.11	56.56	50.69
MSATT-KG	5.73	9.28	1.55	57.72	51.02
HARNN	0.10	1.07	0.09	50.00	50.01
RPGNet	**14.68***	**22.83**	1.20	**59.08***	**54.53***

the hierarchical structure of the ICD taxonomy, it only considers on the relations of all codes in the same level. The parent-child and sibling relations between ICD codes are not considered.

We also show the coding results for the full-label setting, in Table 3. RPGNet performs better than other baselines on most of metrics in this setting, which shows the merits of RPGNet on a large ICD space. As the ICD space gets larger, it is getting more difficult to predict the correct ICD codes. RPGNet alleviates this by narrowing down the space along the ICD taxonomy, i.e., predicting ICD codes from the corresponding levels one at a time. As a result, it gets better results than other models. Besides, we find that for all models, the metrics that drop the most are the macro-averaged metrics, such as the macro F1 of CAML drops from 34.71% to 1.63% and macro F1 of RPGNet drops from 35.40% to 1.20%. This is because the imbalanced distribution of ICD codes becomes severe in the full-label setting. Some ICD codes only occur 1 or 2 times, which is not enough for learning, even for RPGNet.

5.2 Performance at different ICD levels (RQ2)

It is important to evaluate performance at different levels because in some cases, a different granularity of EHR coding may be required. Since RPGNet can generate ICD paths for each EHR and each ICD code in a path can be seen as a prediction result for the EHR at a particular level, we conduct an experiment under the top-50 label setting to compare the performance at different levels of the ICD taxonomy. The results are shown in Fig. 4.

For the results, we can see that the performance of RPGNet decreases as the level goes deeper generally, e.g., the micro-average F1 gradually decreases from 61.55% to 48.35% from level 1 to level 4. This is to be expected because the number of ICD categories increases as the level goes deeper, as shown in Table 1, which, in turn, increases the difficulty of prediction. But note that the decrease in performance is acceptable for RPGNet. For example, when the level increases from 1 to 2, the micro Precision of RPGNet decreases from 63.42% to 50.49%, while when hierarchical level increases from 3 to 4, the micro Precision drops by just 3.31%. Once again, this is because RPGNet considers the relations between EHRs and the hierarchical structure of the ICD taxonomy. There are also some exceptions. For example, the micro Recall, micro AUC of RPGNet at level-4 are higher than those at level-3. We think the reason is that the ICD codes at lower levels facilitate the generation at higher levels in RPGNet. At the same time, the global policy in the hybrid policy network can correct the decisions at higher levels.

These results show that RPGNet is effective for EHR coding by considering the relations between ICD codes and using a hybrid policy network to make decisions.

5.3 Ablation study (RQ3)

We conduct an ablation study to analyze the effects of different modules in RPGNet, as shown in Table 4. These are the variants of RPGNet that we consider: (1) *No ARL* denotes RPGNet without adversarial reinforcement learning. We remove the adversarial reinforcement learning process and just use teacher forcing with ground truth paths to train our model. (2) *No PMP* denotes RPGNet without the PMP module. (3) *No ICD-MP* denotes RPGNet without

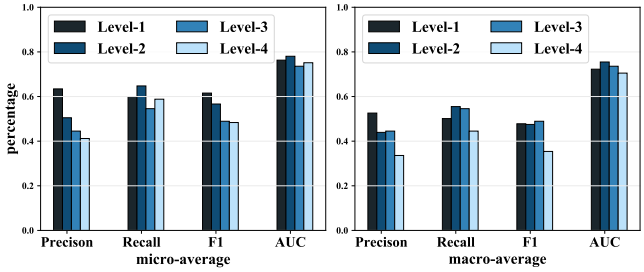

Figure 4: Performance at different levels of hierarchical ICD.

Table 4: Ablation study for the top-50 label setting. Bold face indicates the best result in terms of corresponding metric.

		F1		AUC	
Method	**Jaccard**	micro	macro	micro	macro
RPGNet	**33.34**	**48.35**	**35.40**	**75.14**	**70.50**
No ARL	30.25	44.70	31.59	72.05	68.38
No PMP	32.16	46.98	33.98	73.30	69.50
No ICD-MP	32.79	47.53	34.89	73.67	68.72

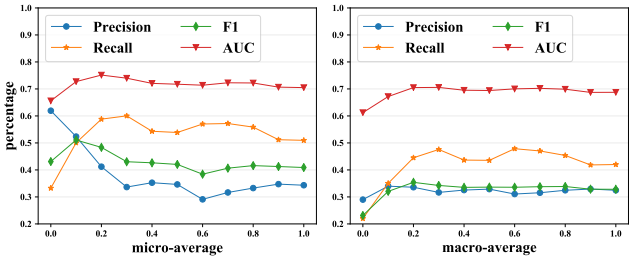

Figure 5: Performance with different values of λ.

the parent-to-child message passing and sibling-to-sibling message passing in PMP. From the results, we obtain the following insights:

First, the performance of RPGNet decreases dramatically after removing adversarial reinforcement learning (i.e., *No ARL*). Specifically, the micro F1 and macro F1 drop by 3.65% and 3.81% respectively. The result shows that adversarial reinforcement learning for RPGNet plays a crucial role. This is because adversarial reinforcement learning helps RPGNet to generate ICD paths closer to the ground truth paths through feedback from the path discriminator. In addition, RPGNet samples ICD paths based on a stochastic strategy, which helps to explore new rare paths [9].

Second, the performance of RPGNet decreases significantly after removing all three stages of message passing (i.e., *No PMP*) or parent-to-child message passing and sibling-to-sibling message passing (i.e., *No ICD-MP*). Specifically, the micro F1 and macro F1 drop by 1.37% and 1.42% respectively with *No PMP*. This indicates that relations between EHRs and ICD paths, and between ICD codes play an important role in the path generation process. The micro F1 and macro F1 drops by 0.82% and 0.51% respectively with *No ICD-MP*. This verifies the importance of relations between ICD codes.

5.4 The impact of λ (RQ4)

The trade-off parameter λ is used to balance the local and global policies in Eq. 3. When λ is smaller, RPGNet tends to prioritize the global policy during learning. Conversely, when λ is larger, RPGNet relies more on its local policy. We conduct an experiment by setting different λ values from $\{0, 0.1, 0.2, \ldots, 1.0\}$ to see the effects of λ. The results are shown in Fig. 5.

First, as λ increases, most metrics except micro Precision increase gradually at the beginning, but they decrease afterward. The best performance is not achieved when $\lambda = 0$ or $\lambda = 1$. This indicates that combining local and global policies properly is necessary.

Second, the overall trend of micro Precision is downward and it reaches the maximum when $\lambda = 0$ (i.e., 61.94%). However, the micro Recall, macro Precision, and macro Recall at that time are very low, with 33.26%, 29%, and 22% respectively. Since when $\lambda = 0$, the prediction result is completely dependent on the global policy, the result indicates that the global policy pays more attention to the frequent ICD codes, but ignores rare ICD codes.

Third, with increases in λ, the macro Recall, macro F1 show an overall growth trend. Specifically, the macro Recall reaches maximum (i.e., 47.86%) when $\lambda = 0.6$. This result indicates that unlike the global policy, the local policy pays more attention to rare ICD codes. Eventually, taking both global and local policies into account, RPGNet performs best at $\lambda = 0.2$.

6 CONCLUSION AND FUTURE WORK

In this work, we reformulate Electronic Health Record (EHR) coding as a path generation task, and propose Reinforcement Path Generation Network (RPGNet), which incorporates a Path Message Passing (PMP) module to encode the relations between EHR and International Classification of Diseases (ICD) codes. We also devise an adversarial reinforcement learning schema for training RPGNet. Experiments on a benchmark dataset show that RPGNet significantly outperforms recent state-of-the-art methods. Further analysis demonstrates the effectiveness of the proposed PMP and of the adversarial reinforcement learning mechanisms.

A limitation of RPGNet is that it only uses unstructured free-text of EHRs; it does not take semi-structured and/or structured information into account. As to future work, we plan to first extract semi-structured and/or structured information from EHRs and ground it to a knowledge base. Then, we hope to further improve RPGNet by exploring how to model the unstructured, semi-structured and structured information in EHRs with grounded knowledge.

DATA AND CODE

To facilitate reproducibility of our work, we are sharing all resources used in this paper at https://github.com/WOW5678/RPGNet.

ACKNOWLEDGMENTS

We want to thank our anonymous reviewers for their feedback.

This work was partially supported by the Natural Science Foundation of China (61972234, 61902219, 61672324, 61672322), the Key Scientific and Technological Innovation Program of Shandong Province (2019JZZY010129), the Tencent AI Lab Rhino-Bird Focused Research Program (JR201932), the Fundamental Research Funds

of Shandong University, 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] Anand Avati, Kenneth Jung, Stephanie Harman, Lance Downing, Andrew Y. Ng, and Nigam H. Shah. 2018. Improving Palliative Care with Deep Learning. *BMC Med. Inf. & Decision Making* 18, S-4 (2018), 55–64.

[2] Tal Baumel, Jumana Nassour-Kassis, Raphael Cohen, Michael Elhadad, and Noémie Elhadad. 2018. Multi-Label Classification of Patient Notes: Case Study on ICD Code Assignment. In *The Workshops of the Thirty-Second AAAI Conference on Artificial Intelligence*. 409–416.

[3] Casey C Bennett and Kris Hauser. 2013. Artificial Intelligence Framework for Simulating Clinical Decision-making: A Markov Decision Process Approach. *Artificial intelligence in medicine* 57, 1 (2013), 9–19.

[4] 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).

[5] Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. 2016. Doctor AI: Predicting Clinical Events via Recurrent Neural Networks. In *Proceedings of the 1st Machine Learning in Health Care, MLHC 2016*. 301–318.

[6] Edward Choi, Siddharth Biswal, Bradley A. Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. 2017. Generating Multi-label Discrete Patient Records using Generative Adversarial Networks. In *MLHC*. 286–305.

[7] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.* 12 (2011), 2493–2537.

[8] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. 2017. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. *CoRR* abs/1706.02633 (2017).

[9] Vincent François-Lavet, Peter Henderson, Riashat Islam, Marc G. Bellemare, and Joelle Pineau. 2018. An Introduction to Deep Reinforcement Learning. *Foundations and Trends in Machine Learning* 11, 3-4 (2018), 219–354.

[10] Justin Fu, Katie Luo, and Sergey Levine. 2017. Learning Robust Rewards with Adversarial Inverse Reinforcement Learning. *arXiv preprint arXiv:1710.11248* (2017).

[11] Cyril Goutte and Eric Gaussier. 2005. A Probabilistic Interpretation of Precision, Recall and F-score, with Implication for Evaluation. In *ECIR*. Springer, 345–359.

[12] Jiaqi Guan, Runzhe Li, Sheng Yu, and Xuegong Zhang. 2018. Generation of Synthetic Electronic Medical Record Text. In *BIBM*. 374–380.

[13] Jonathan Ho and Stefano Ermon. 2016. Generative Adversarial Imitation Learning. In *NIPS*. 4565–4573.

[14] Jin Huang and Charles X Ling. 2005. Using AUC and Accuracy in Evaluating Learning Algorithms. *IEEE Transactions on knowledge and Data Engineering* 17, 3 (2005), 299–310.

[15] Jinmiao Huang, Cesar Osorio, and Luke Wicent Sy. 2019. An Empirical Evaluation of Deep Learning for ICD-9 Code Assignment Using MIMIC-III Clinical Notes. *Computer Methods and Programs in Biomedicine* 177 (2019), 141–153.

[16] Wei Huang, Enhong Chen, Qi Liu, Yuying Chen, Zai Huang, Yang Liu, Zhou Zhao, Dan Zhang, and Shijin Wang. 2019. Hierarchical Multi-label Text Classification: An Attention-based Recurrent Network Approach. In *CIKM*. 1051–1060.

[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] Ramakanth Kavuluru, Anthony Rios, and Yuan Lu. 2015. An Empirical Evaluation of Supervised Learning Approaches in Assigning Diagnosis Codes to Electronic Medical Records. *Artificial Intelligence in Medicine* 65, 2 (2015), 155–166.

[20] Yoon Kim. 2014. Convolutional Neural Networks for Sentence Classification. In *EMNLP*. 1746–1751.

[21] Jack Lanchantin, Arshdeep Sekhon, and Yanjun Qi. 2019. Neural Message Passing for Multi-Label Classification. *CoRR* abs/1904.08049 (2019).

[22] Zhen Li, Tao Tang, and Chunhai Gao. 2019. Long Short-Term Memory Neural Network Applied to Train Dynamic Model and Speed Prediction. *Algorithms* 12, 8 (2019), 173.

[23] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley. 2017. Deep Learning for Healthcare: Review, Opportunities and Challenges. *Briefings in bioinformatics* 19, 6 (2017), 1236–1246.

[24] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable Prediction of Medical Codes from Clinical Text. In *NAACL-HLT*. 1101–1111.

[25] Shamim Nemati, Mohammad M Ghassemi, and Gari D Clifford. 2016. Optimal Medication Dosing from Suboptimal Clinical Examples: a Deep Reinforcement Learning Approach. In *EMBC*. 2978–2981.

[26] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. 2013. Using of Jaccard Coefficient for Keywords Similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, Vol. 1. 380–384.

[27] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2013. Diagnosis Code Assignment: Models and Evaluation Metrics. *Journal of the American Medical Informatics Association* 21, 2 (2013), 231–237.

[28] Aaditya Prakash, Siyuan Zhao, Sadid A. Hasan, Vivek V. Datla, Kathy Lee, Ashequl Qadir, Joey Liu, and Oladimeji Farri. 2017. Condensed Memory Networks for Clinical Diagnostic Inferencing. In *AAAI*. 3274–3280.

[29] Anthony Rios and Ramakanth Kavuluru. 2018. EMR Coding with Semi-Parametric Multi-Head Matching Networks. In *NAACL-HLT*. 2081–2091.

[30] Najmeh Sadoughi, Greg P. Finley, James Fone, Vignesh Murali, Maxim Korenevsky, Slava Baryshnikov, Nico Axtmann, Mark Miller, and David Suendermann-Oeft. 2018. Medical Code Prediction with Multi-view Convolution and Description-regularized Label-dependent Attention. *CoRR* abs/1811.01468 (2018).

[31] Haoran Shi, Pengtao Xie, Zhiting Hu, Ming Zhang, and Eric P. Xing. 2017. Towards Automated ICD Coding Using Deep Learning. *CoRR* abs/1711.04075 (2017).

[32] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway Networks. *arXiv preprint arXiv:1505.00387* (2015).

[33] 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*.

[34] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *NIPS*. 5998–6008.

[35] Guoyin Wang, Chunyuan Li, Wenlin Wang, Yizhe Zhang, Dinghan Shen, Xinyuan Zhang, Ricardo Henao, and Lawrence Carin. 2018. Joint Embedding of Words and Labels for Text Classification. In *ACL*. 2321–2331.

[36] 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 *CIKM*. 1623–1632.

[37] Ronald J Williams. 1992. Simple Statistical Gradient-following Algorithms for Connectionist Reinforcement Learning. *Machine learning* 8, 3-4 (1992), 229–256.

[38] Tyra G. Wolfsberg, Paul Primakoff, Diana G. Myles, and Judith 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.

[39] Xiancheng Xie, Yun Xiong, Philip S. Yu, and Yangyong Zhu. 2019. EHR Coding with Multi-scale Feature Attention and Structured Knowledge Graph Propagation. In *CIKM*. 649–658.

[40] Keyang Xu, Mike Lam, Jingzhi Pang, Xin Gao, Charlotte Band, Piyush Mathur, Frank Papay, Ashish K. Khanna, Jacek B. Cywinski, Kamal Maheshwari, Pengtao Xie, and Eric P. Xing. 2019. Multimodal Machine Learning for Automated ICD Coding. In *MLHC*. 197–215.

[41] Kezi Yu, Yunlong Wang, Yong Cai, Cao Xiao, Emily Zhao, Lucas Glass, and Jimeng Sun. 2019. Rare Disease Detection by Sequence Modeling with Generative Adversarial Networks. *CoRR* abs/1907.01022 (2019).

[42] Jie Zhou, Ganqu Cui, Zhengyan Zhang, Cheng Yang, Zhiyuan Liu, and Maosong Sun. 2018. Graph Neural Networks: A Review of Methods and Applications. *CoRR* abs/1812.08434 (2018).

[43] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *ACL*. 207–212.