Joint Modeling of Document and Label with Clause Interaction Hypergraph for ICD Medical Code Assignment

Haoran Wu[1,2], Linghui Meng[1,2], Shuang Xu[1], Bo Xu[1,2]
[1] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100190, China
{wuhaoran2018, menglinghui2019, shuang.xu, xubo}@ia.ac.cn

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892499

***Abstract*—Automatic medical code assignment for clinical records is the fundamental problem of medical statistical research and informatization. Due to the high dimension and sparse distribution of label space, it is necessary to make full use of the description information of the labels. However, most of the current work is based on similarity matching at the level of token or n-gram, and ignores information fusion with richer semantic structure and representation. In this paper, we propose a Clause Interaction HyperGraph (CIHG) to jointly model documents and label descriptions, which construct a richer semantic interaction at the level of clause. The CIHG models the high-order co-occurrence relationship between document and labels based on hypergraph, and uses the semantic structure of the document to constrain the encoding of labels. Experiments on widely used medical code assignment datasets show that our method successfully constrains the embedding of labels and significantly improves predictive precision[1].**

***Index Terms*—ICD Code Assignment, Hypergraph, Graph Convolution, Joint Modeling of Document and Label, Deep Learning**

I. Introduction

The International Classification of Disease (ICD) system maintained by the World Health Organization is widely used in clinical data analyzing and monitoring health issues, and automatic medical code assignment is the foundation of its use [1]–[5]. As shown in Fig 1, medical code assignment can be regarded as a large-scale multi-label text classification problem [6], which is a classic challenge in natural language processing. The characteristics of this task are large label space and sparse supervision signal. For example, ICD-9 has more than 20000 labels and ICD10 has more than 60000 labels. In addition, according to our statistics, in 50000 samples, more than 40% of the labels appear less than three times. Therefore, traditional sequence modeling methods are difficult to deal with this kind of problem, and the challenge to solve this problem is how to make better use of the text descriptions and the hierarchical organization structure of the label.

In recent years, there have been a series of researches on the assignment of ICD codes. Previous works treat the clinical record text as a single sequence to interact with the label information at the token level with GRU [7] or the n-gram level with CNN [8]. [9] maps label-wise document representation and label descriptions to hyperbolic space for matching and [10] builds interactive connections between each word and label. However, the interaction based on the entire document will lose too many details, and based on the token or n-gram cannot contain enough contextual information. These methods lack information fusion with richer semantic representation. In addition, the interaction of these methods stays at the level of similarity matching, rather than exploiting the semantic and structural information contained in the documents and labels.

To solve this problem, we propose the *Clause Interaction HyperGraph* (**CIHG**) to model the documents and labels jointly. *Clause* is a finer segmentation of sentence. Since clauses (Avg. tokens 7.8) and label descriptions (Avg. tokens 6.6) have similar lengths, modeling at the clause granularity enables more precise interactions. Therefore, we build the hierarchical document tree and label tree based on the clause segmentation of documents and the hierarchical structure of labels respectively. *Hypergraph* have the hyperedge, which can connect more than two nodes to capture high-dimensional

Ms. [Known lastname**] was admitted to [**Hospital1 18**] on [**11-3**]. She had an EVD placed in the ED by Dr. [**Last Name (STitle) 739]. …She had a HCT drop of 10 points and received one unit of PRBCS. …CTA head [**11-3**]: 1.Massive subarachnoid hemorrhage. …Extensive intraventricular hemorrhage with severe hydrocephalus. …They made her DNR. …She expired on [**2189-11-5**]…**

ICD-9 Codes	Disease Name
790.01	Precipitous drop in hematocrit
430	Subarachnoid hemorrhage
331.4	Obstructive hydrocephalus
V66.7	Encounter for palliative care
…	…

Fig. 1. An example of automatically predicting ICD codes for clinical notes. The upper part of the figure is the clinical note fragment, and the bottom part is the ICD codes with text descriptions. The text fragments and ICD codes with corresponding relationships are marked with the same color.

This work was supported by the Key Research Program of the Chinese Academy of Sciences under Grant No. ZDBS-SSW-JSC006-2 and Strategic Priority Research Program of the Chinese Academy of Sciences under Grant No. XDA27030300.

[1]The code is available at https://github.com/CKRE/CIHG

978-1-7281-8671-9/22/$31.00 ©2022 IEEE

copolymerization relationships. We build a clause hypergraph based on the semantic structure of the document. The clause hypergraph can capture the copolymerization of the information of clauses in the document, and realize the long-distance interaction in the document. It ensures the separation of different semantic information and the aggregation of similar semantic information in the document. Specifically, We obtain the category of clauses as structural information through clustering, and the clause category is taken as the hyperedge to construct the clause hypergraph on the basis of retaining the original sequence relationship of the document. *Interaction* represents interactive connection. To build the communication and semantic information fusion between the two tree structures of the document and labels, we use fuzzy matching [11] to assign a corresponding label for each clause and connect. The interactive connection and hypergraph constrain the encoding of labels with the semantic structure of the document. In the embedding space, the distance between labels with similar medical semantics is shortened, and with confused semantics is lengthened. It brings better discrimination to label prediction. This is equivalent to converting the clause semantic structure into the copolymerization structure of the label. Our contributions are summarized as follows:

- We propose a Clause Interaction Hypergraph (**CIHG**) to jointly model document and labels, which realizes the rich semantic interaction at the level of clause between documents and labels.
- The high-order co-occurrence relationship between document and label tree is modeled by hypergraph simultaneously, which introduces effective constraints for the representation of documents and labels.
- Experiments on widely used ICD assignment datasets show that our method successfully constrains the embedding of labels and significantly improves the predictive precision over the baseline.

II. Related Work

a) Automatic Medical Code Assignment: Automatic medical code assignment is an ongoing challenge in the field of medical informatization [10], [12], [13]. There are already many traditional machine learning approaches that provide solutions for this task such as Bayesian ridge regression [14] and hierarchical SVM [15]. In recent years, a series of deep learning methods have achieved significant improvements in this task [16]–[18]. [7], [8] introduced two label-wise attention methods to map document features into the label space. In order to strengthen the interaction between the document and label, [9] performs similarity matching in hyperbolic space and [10] constructs a complete bipartite graph. Different from existing work, we use clause hypergraph to model documents and hierarchical labels simultaneously, and achieve efficient information interaction.

b) Hypergraph: Hypergraph is used to model the high-order multivariate relationship between nodes [19] and have been used in many fields such as question answering [20], anchor link prediction [21], text classification [22] and so on. In order to realize the representation learning of the hypergraph structure, [23] proposed a hypergraph neural network suitable for the hypergraph, and [24] added an attention mechanism on the basis of it. In addition, [22] proposed the HyperGAT based on two-stage attention to solve the problem of text hypergraph representation. Since there is a magnitude difference between the number of bidirectional edges and hyperedges in our clause hypergraph, we use a two-stage convolution to fuse the calculation of hypergraph and simple graph.

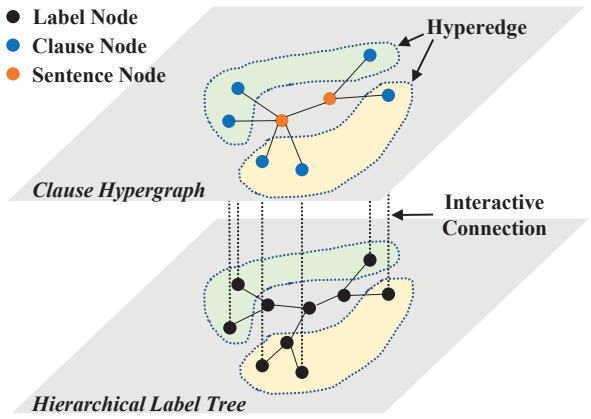


Fig. 2. Overview of Clause Interaction Hypergraph. The circles of the same color in the upper part and lower part of the figure represent the same hyperedge. The clause node and label node with interactive connection are connected by the dotted line.

III. Proposed Method

A. Problem Definition

We treat ICD code assignment as a large-scale multi-label text classification problem. Given a clinical document $T = \{\omega_1, \omega_2, \ldots, \omega_n\}$ with n words and a set of labels $L = \{L_1, L_2, \ldots, L_m\}$ with m labels having an inherent hierarchical tree structure, our goal is to select all labels from L relevant to the clinical document T.

B. Clause Interaction Hypergraph Construction

a) Clause Document Graph: Fine-grained hierarchical segmentation is conducive to document modeling [25], [26]. The clause is a further segmentation of the sentence, with semantic information more concentrated than the sentence and more complete than n-gram. Due to the similar length, better semantic matching can be achieved between clause and code description. Therefore, we choose the clause for document modeling and label interaction. The clauses are obtained based on punctuation. Specifically, we first divide the document into sentences based on periods and paragraphs, and then divide the sentence into clauses based on commas, semicolons, etc. To preserve the original sequential relationship of the document, we retain the granularity of the sentence and concatenate them in order. The clause nodes are connected to the sentence they belong to, and all the edges are bidirectional. The document graph structure is the upper part of Fig. 2.

b) Hierarchical Label Tree: ICD codes are organized into a hierarchical tree structure. Starting from the root node, all codes are allocated layer by layer according to the continuous subdivision of the disease, such as "001-999.9" → "001-139.99" → "001-009.99" → "001" → "001.1". As shown in the lower part of Fig. 2, we retained this hierarchical structure when building the graph, and all the edges point from the root node to the leaf node.

c) Interactive Connection: A tree-like graph network without additional constraints cannot encode higher-dimensional relationships between labels. In order to constrain the label encoding and strengthen the information fusion between the document and labels, we need to construct an interaction between the two tree structures. Specifically, we calculate the edit distance [11] between each clause and all the label descriptions. As shown in Fig. 2, the clause node is matched to the label node with the highest matching score. This is a virtual matching relationship without real edges.

d) Clause Hypergraph: Hypergraph is a special graph structure used to describe high-dimensional copolymerization relationships. In a bipartite graph, an edge can only connect two nodes, while an edge can connect multiple nodes in a hypergraph. This special edge is called hyperedge. In general, there are two ways to construct a hypergraph: structure-based and similarity-based. Structure-based hypergraph is usually constructed based on the structure of the data itself. In the document, the sentence is a natural structure. Taking each sentence as a hyperedge, including the corresponding clause nodes and the matching label nodes. The graph constructed in this way is called the sentence hypergraph. Similarity-based hypergraph is usually based on clustering relationships. In the two hierarchical graphs we constructed above, the similarity structure of each side can be used to constrain the representation of the other side. Fig. 2 shows a scheme based on clause clustering. Specifically, we use the combination of TF-IDF [27] and K-Means to cluster clauses. Based on this clustering relationship, we obtain the semantic structure of the document. Then, each category of the clause can be regarded as a hyperedge, including the corresponding clause nodes and the matching label nodes. This connection spans different sentences, covers the entire document and label tree, which realizes the long-distance interaction of document information and transfers the structural constraints of the document to the labels encoding. This scheme is called the clause hypergraph. Symmetrically, label hypergraph can also be constructed by the clustering relationship of labels. The main experiment in our paper is based on the clause hypergraph.

C. Input Layer

We utilize word2vec [28] to obtain word embedding matrix $\tilde{\boldsymbol{E}} \in \mathbb{R}^{v \times d_e}$, where v, d_e are the vocab size and embedding size. Each node in the clause interaction hypergraph contains a text segment $T_{node} = [\omega_1, \omega_2, \ldots, \omega_{n_g}]$ with n_g words, so the feature $\boldsymbol{e}_{node}$ of one node in the graph can be obtained by looking up in $\tilde{\boldsymbol{E}}$ and performing average pooling. Similarly, we can obtain the embedding $\boldsymbol{E}_d = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_n]$ of the whole document from $\tilde{\boldsymbol{E}}$. Then, we use a bidirectional GRU [29] layer to obtain the contextual representation $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{n \times d_d}$ of the document, where d_d is the output dimension of document encoder:

$$\boldsymbol{x}_t = \boldsymbol{BiGRU}(\boldsymbol{x}_{t-1}, \boldsymbol{e}_t) \tag{1}$$

where $\boldsymbol{x}_t$, $\boldsymbol{e}_t$ is the hidden state and the word embedding of the t-th token.

D. Mixed Hypergraph Convolution

Although the bipartite edge can be regarded as a special case of the hyperedge, due to the large gap between the number of simple edges and hyperedges in clause interaction hypergraph, we adopt the combination of graph convolution [30] and hypergraph convolution [24] to obtain higher-level feature of the graph.

A mixed hypergraph is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{H}, \mathcal{W})$, where $\mathcal{E}$ denotes the bipartite edge set with b edges and $\mathcal{H}$ denotes the hyperedge set with c hyperedges. $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ denotes the node set with a nodes, including a_c clause nodes, a_s sentence nodes and a_l label nodes. $\mathcal{W}$ denotes the importance weight between hyperedge. In order to ensure the consistency of the encoding space, we use the text embedding result $\boldsymbol{e}_{node}$ for all sentence, clause and label nodes as the graph features:

$$\boldsymbol{G} = [\boldsymbol{E}_l, \boldsymbol{E}_s, \boldsymbol{E}_c] \tag{2}$$

where $\boldsymbol{G} \in \mathbb{R}^{a \times d_e}$, $\boldsymbol{E}_l = \{\boldsymbol{e}_{l_i}\}_{i=1}^{a_l}$, $\boldsymbol{E}_s = \{\boldsymbol{e}_{s_i}\}_{i=1}^{a_s}$ and $\boldsymbol{E}_c = \{\boldsymbol{e}_{c_i}\}_{i=1}^{a_c}$. We iteratively update the features with bipartite edges and hyperedges.

a) Bipartite Edge Convolution: For the bipartite edges, we use the adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{a \times a}$ to represent the pairwise connection between nodes. Then, we update the graph features with graph convolution as

$$\boldsymbol{G}_b = \sigma(\tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{G} \boldsymbol{P}_b) \tag{3}$$

where $\tilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I}$ and $\boldsymbol{I} \in \mathbb{R}^{a \times a}$ is the unit matrix. $\tilde{\boldsymbol{D}} \in \mathbb{R}^{a \times a}$ is the degree matrix of $\tilde{\boldsymbol{A}}$, $\boldsymbol{P}_b \in \mathbb{R}^{d_e \times d_e}$ is a learnable weight matrix for the dimension projection and σ is the activation function $Relu$. At this point we have completed the status update of all nodes connected by the bipartite edges.

b) Hypergraph Convolution: Since not all nodes are included in the hypergraph, we only operate the feature of nodes contained in the hyperedge. The hypergraph can be represented by an incidence matrix $\boldsymbol{H} \in \mathbb{R}^{k \times c}$, where k represents the number of nodes contained in the hyperedge. In the incidence matrix $\boldsymbol{H}$, row i represents the node and column h represents the hyperedge. If the node i is contained in the hyperedge h, $\boldsymbol{H}_{ih} = 1$, otherwise 0. In addition, the importance of clauses is different. We counted the proportion of the clauses contained in each hyperedge to the total clauses and store them in the diagonal matrix $\boldsymbol{W} \in \mathbb{R}^{c \times c}$ as the importance weight. Then the node degree and hyperedge degree can be calculated as

$$\boldsymbol{D}_{ii} = \sum_{h=1}^{c} \boldsymbol{W}_{hh} \boldsymbol{H}_{ih}, \boldsymbol{B}_{hh} = \sum_{i=1}^{k} \boldsymbol{H}_{ih} \tag{4}$$

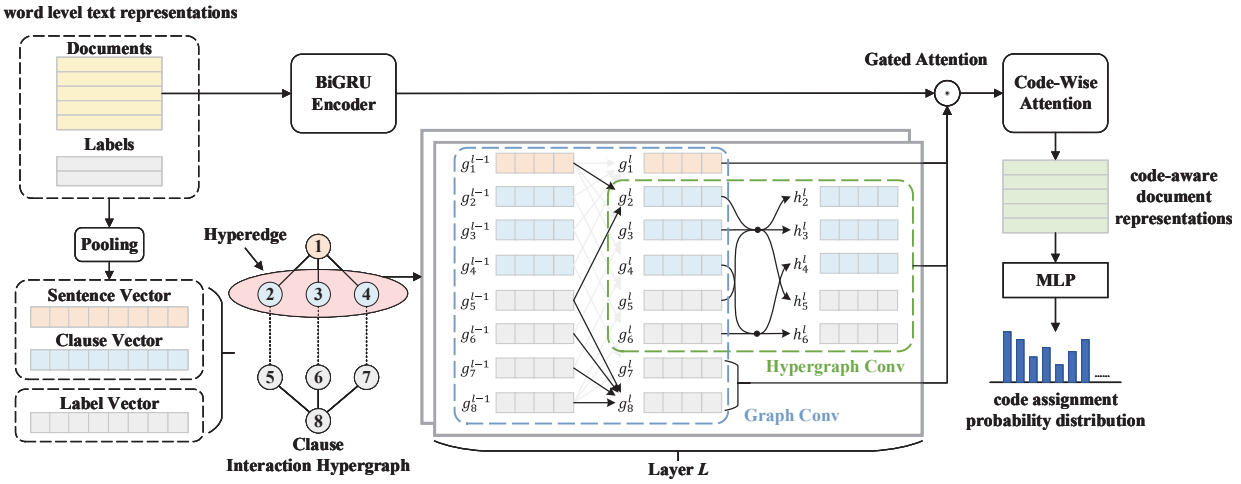


Fig. 3. The overview of our model. The word-level text representations are the word embedding of clinical notes and label texts obtained based on word2vec. The document representation is partitioned into sentences and clauses according to the pre-processing. And together with label description representations, the corresponding vector is obtained by average pooling as the feature of the clause hypergraph nodes.

Similar to graph convolution, the feature update of hypergraph convolution can be calculated as

$$\boldsymbol{G}_h = \sigma(\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{H}\boldsymbol{W}\boldsymbol{B}^{-1}\boldsymbol{H}^T\boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{G}_s\boldsymbol{P}_h) \tag{5}$$

where $\boldsymbol{P}_h \in \mathbb{R}^{d_e \times d_e}$ is the learnable weight matrix.

Finally, the mixed graph convolution of each layer is calculated as follows and each layer is connected by residuals.

$$\boldsymbol{G}_l = \boldsymbol{G}_{l-1} + \boldsymbol{HGCN}(\boldsymbol{GCN}(\boldsymbol{G}_{l-1}, \boldsymbol{A}), \boldsymbol{H}, \boldsymbol{W}) \tag{6}$$

c) Gated Attention: Through the clause interaction hypergraph, we obtained the feature $\boldsymbol{G}_l$ after the entire interaction and aggregation of clinical documents and label text. Since the structure of the graph is not necessarily optimal, we select the node information in the graph with gated attention as follows:

$$\boldsymbol{C} = \boldsymbol{Relu}(\boldsymbol{X}\boldsymbol{P}_x) \cdot \boldsymbol{Relu}(\boldsymbol{G}_l\boldsymbol{P}_g)^T \tag{7}$$

$$\boldsymbol{Z} = \boldsymbol{Softmax}(\boldsymbol{C}) \cdot \boldsymbol{G}_l \tag{8}$$

$$\boldsymbol{R} = [\boldsymbol{Sigmoid}([\boldsymbol{X}; \boldsymbol{Z}]\boldsymbol{P}_s); \boldsymbol{Tanh}([\boldsymbol{X}; \boldsymbol{Z}]\boldsymbol{P}_t)] \tag{9}$$

where $\boldsymbol{P}_x \in \mathbb{R}^{d_d \times d_e}$, $\boldsymbol{P}_g \in \mathbb{R}^{d_e \times d_e}$, $\boldsymbol{P}_s \in \mathbb{R}^{(d_d+d_e) \times d_d}$, $\boldsymbol{P}_t \in \mathbb{R}^{(d_d+d_e) \times d_d}$ are learnable parameter matrices and ";" represents the concatenation of vectors.

E. Output layer

After the mixed convolution of the graph, we can obtain the fusion feature $\boldsymbol{R} \in \mathbb{R}^{n \times d_d}$. In order to implement multi-label classification, we introduce a label-wise attention matrix $\boldsymbol{M} \in \mathbb{R}^{m \times d_d}$ to map document features to the label feature space.

$$\boldsymbol{U} = \boldsymbol{Softmax}(\boldsymbol{M} \cdot \boldsymbol{R}^T) \cdot \boldsymbol{R} \tag{10}$$

Each row $\boldsymbol{U}_j$ included in $\boldsymbol{U} \in \mathbb{R}^{m \times d_d}$ represents the information related to the label j in the document. Then for each label j, we use a fully connected layer to predict the probability of belonging to the document.

$$p_j = \boldsymbol{Sigmoid}(\boldsymbol{MLP}(\boldsymbol{U}_j)) \tag{11}$$

TABLE I
THE STATISTICAL SUMMARY OF DATASETS.

	MIMIC-III-full	MIMIC-III-50
# Train.	47724	8067
# Dev.	1631	1574
# Test.	3372	1730
Avg. tokens	1485	1530
Avg. labels	15.9	5.7
Number of labels	8921	50

This probability is used to judge whether the label j belongs to the document according to a predefined threshold. We minimize the following objective based on the BCE loss function as

$$\mathcal{L} = -\sum_{j=1}^{m} l_j log(p_j) + (1 - l_j) log(1 - p_j) \tag{12}$$

where $l_j = 1$ means that the label j belongs to the document, otherwise $l_j = 0$.

IV. Experiment

In this section, we describe the experimental details of the clause interaction hypergraph and provide further analysis of the model and results.

A. Dataset

We verify the effect of clause interaction hypergraph on a widely used dataset MIMIC-III [31], which includes clinical notes and their corresponding ICD-9 codes labeled by human coders. Following the previous works [8], [16], we use the discharge summaries as the summary of all clinical notes and extract the corresponding ICD codes according to the patient number. MIMIC-III-Full and MIMIC-III-50 are two common settings. MIMIC-III-Full contains the complete dataset and MIMIC-III-50 only contains the instance including the top 50

TABLE II
MAIN RESULT ON TWO DATASETS. THE MODEL WE RAN 5 SEEDS AND REPORT THE MEAN ± STANDARD DEVIATION. THE NUMBERS IN BOLD IN THE TABLE REPRESENT THE BEST PERFORMANCE.

Model	MIMIC-III-50					MIMIC-III-Full				
	AUC-ROC		F1		P@5	AUC-ROC		F1		P@8
	Macro	Micro	Macro	Micro		Macro	Micro	Macro	Micro	
C-MemNN	83.3	-	-	-	42.0	-	-	-	-	-
CNN	87.6	90.7	57.6	62.5	62.0	80.6	96.9	4.2	41.9	58.1
Attentive LSTM	-	90.0	-	53.2	-	-	-	-	-	-
CAML	87.5	90.9	53.2	61.4	60.9	82.0	96.6	4.8	44.2	52.3
DR-CAML	88.4	91.6	57.6	63.3	61.8	82.6	96.6	4.9	45.7	51.5
LEAM	88.1	91.2	54.0	61.9	61.2	-	-	-	-	-
BERT-LWAN	81.4	85.3	42.5	49.8	52.1	84.0	97.4	2.3	32.5	53.6
HyperCore	89.5±0.3	92.9±0.2	60.9±0.1	66.3±0.1	63.2±0.2	**93.0**±0.1	**98.9**±0.5	9.0±0.3	55.1±0.1	72.2±0.2
GatedCNN-NCI	91.5±0.3	93.8±0.1	62.9±0.5	68.6±0.1	65.3±0.1	92.2±0.2	**98.9**±0.3	9.2±0.2	56.3±0.1	73.6±0.3
BiGRU	82.8	86.8	48.4	54.9	59.1	82.2	97.1	3.8	41.7	58.5
MultiResCNN	89.9±0.4	92.8±0.2	60.6±1.1	67.0±0.3	64.1±0.1	91.0±0.2	98.6±0.1	8.5±0.7	55.2±0.5	73.4±0.2
BiGRU+CIHG	**92.0**±0.2	**94.1**±0.1	**65.7**±0.3	**70.8**±0.1	**66.8**±0.1	88.1 ±0.2	98.4±0.3	8.6 ±0.2	56.6 ±0.1	**75.1**±0.1
MultiResCNN+CIHG	91.8±0.1	93.8±0.1	64.9±0.5	69.6±0.1	65.8±0.1	90.6±0.3	98.7±0.1	**9.7**±0.4	**57.9**±0.1	74.5±0.2

most frequent codes. The statistics of the dataset are shown in Table I.

B. Experiment Settings

For MIMIC-III-Full and MIMIC-III-50, we use the same word2vec embedding and K-means clause clustering results, both of which are pre-trained on all discharge summaries. The embedding size d_e is 100 and the number of clause categories is 20. In the process of establishing the interactive connection, the similarity score between the clause text and label description text is calculated by FuzzyWuzzy. Then we select the label with the highest similarity score for each clause to match. Our model is implemented based on PyTorch [32], and we train and test the model on one Titan XP GPU and Xeon E5-2643 v4 CPU. In the experiments, the batch size is 4, the dropout rate is 0.3, the number of layers of mixed hypergraph convolution is 2. We use Adam optimizer [33] for training, and the learning rate is 1e-4. For the experiments based on MultiResCNN, following [34], the convolution filter size is the combination of $\{3, 5, 9, 15, 17, 25\}$, and each filter output size is 50. For the BiGRU-based experiment, the hidden size is 300. The threshold for all label predictions is set to 0.5.

C. Baselines

We select some representative work for comparison.

CAML [8] uses CNN to extract the n-gram features of the clinical note and trains the label representation based on label-wise attention for classification.

MultiResCNN [34] combines multi-filter convolutional network and residual blocks.

HyperCore [9] maps documents and ICD codes into hyperbolic space, combining the hierarchical information and the co-occurrence relationship of ICD codes.

BERT-LWAN [6] combines BERT encoder and label-wise attention. It achieved SOTA on multiple large-scale multi-label classification datasets. Here we use the Clinical-BERT [35], which has been fine-tuned on biomedical documents, including discharge summaries.

GatedCNN-NCI [10] uses dilation convolution with a forgetting mechanism to extract features of clinical notes, and constructs a complete bipartite graph between documents and labels at word level for interaction.

Some typical models that are also compared including C-MemNN [36], Attentive LSTM [37] and LEAM [38]. The results of BiGRU and CNN are reported by [8].

D. Metrics

Following the previous work [8], we use AUC-ROC, F1 and P@k to evaluate the performance of our model. AUC-ROC is defined as the area enclosed by the coordinate axis under the ROC curve. F1 is the most common metric in the classification problem, and P@k represents the precision of the model to predict the top k labels. We set $k = 5$ for MIMIC-III-50 and $k = 8$ for MIMIC-III-Full. Macro and Micro are two different averaging algorithms [39]. Macro calculates the metrics for each category separately and then takes the average, while Micro does not distinguish between categories and directly uses the population samples for evaluation.

E. Result

In this section, we add the Clause Interaction Hypergraph to the two representative and high performance ICD assignment baselines (MultiResCNN and GRU) for experimentation. The results in Table II show that our method has advantages in most metrics, and the performance has been significantly improved after adding the CIHG.

(1) The MIMIC-III-50 can be used to verify the classification performance on a medium-scale high-frequency label set. From the result, we can see that our method based on GRU achieved the best performance on all metrics. Compared with strong baselines, our method achieves an improvement of 4.5% on F1-Macro and 3.4% on F1-Micro, and also achieves an improvement of 2.3% in the precision of top-5 labels. This

TABLE III
THE ABLATION EXPERIMENTAL RESULTS ON MIMIC-III-50 ABOUT CLAUSE DOCUMENT GRAPH (GD.), HIERARCHICAL LABEL TREE (GL.) AND HYPERGRAPH (HG.).

Model	F1		P@5
	Macro	Micro	
MultiResCNN	59.6	67.5	64.2
MultiResCNN + GD.	64.4	69.5	65.6
MultiResCNN + GL.	63.8	68.6	65.3
MultiResCNN + GL. + GD.	63.8	68.9	65.4
MultiResCNN + GL. + GD. + HG.	64.9	69.6	65.8
BiGRU	60.1	66.5	64.3
BiGRU + GD.	62.4	68.6	66.4
BiGRU + GL.	64.7	70.2	66.5
BiGRU + GL. + GD.	64.1	70.4	66.6
BiGRU + GL. + GD. + HG.	65.7	70.8	66.8

TABLE IV
THE EXPERIMENTAL RESULTS ON MIMIC-III-50 ABOUT DIFFERENT TYPES OF HYPERGRAPH, WHERE CH. REPRESENTS CLAUSE HYPERGRAPH, SH. REPRESENTS SENTENCE HYPERGRAPH AND LH. REPRESENTS LABEL HYPERGRAPH.

Model	F1		P@5
	Macro	Micro	
MultiResCNN	59.6	67.5	64.2
MultiResCNN + SH.	64.5	69.2	65.3
MultiResCNN + LH.	64.6	69.4	65.7
MultiResCNN + CH.	64.9	69.6	65.8
BiGRU	60.1	66.5	64.3
BiGRU + SH.	63.8	69.6	66.5
BiGRU + LH.	65.2	70.0	66.6
BiGRU + CH.	65.7	70.8	66.8

result shows the superiority of our method in high-frequency label data.

(2) Since MIMIC-III-Full has a huge label assignment space, accurate prediction of the corresponding label requires full use of the information carried by the labels. It can be used to measure the ability of the model to utilize external knowledge. In this dataset, our method still achieves the improvement of 2.8% on F1-Micro and 2% on the precision of top-8 labels. It indicates that our method can still maintain the precision of code assignment even after the number of labels has increased to a large scale.

F. Ablation Analysis

a) The effectiveness of the Clause Interaction Hypergraph: In this section, we conduct ablation experiments on several main components (clause document graph, hierarchical label tree and hypergraph) of the clause interaction hypergraph to examine the effectiveness. Our experiment is still carried out on the two representative baselines, with and without the specific components. The experimental results are shown in Table III. It can be seen that the addition of clause document graph and hierarchical label tree both can bring significant performance improvement to the baseline. This proves that the incorporation of document structure information and label description information can improve the ability of document modeling. However, the joint addition of clause document graph and hierarchical label tree does not lead to better improvement. This is due to the lack of effective interaction between the two. Once we use hypergraph to bridge the gap between the clause document graph and label tree, the performance of the model can be further improved. This proves the effectiveness of the hypergraph.

b) Comparison between different hypergraph structures: In order to explore the quality of different hypergraph construction schemes, we compare the performance of clause hypergraph, sentence hypergraph and label hypergraph based on two representative models. The results are shown in Table IV. Compared with the baseline, all the three structures of the hypergraph can provide performance improvements. Compared with the other two structures, the performance of the sentence hypergraph is poor. We deem this is because sentence hypergraph can only use the local structure information of the document at the sentence level, and lacks the global copolymerization structure. Both the clause hypergraph and the label hypergraph utilize the global copolymerization structure of the document and show advantages in performance. However, the label hypergraph composed of label clustering will cause confusion on similar labels, so that the precision is not as good as the clause hypergraph.

G. Discussion

a) Computational Cost Analysis: In this section, we analyze the computational cost of the model conducted from parameter amount, training time, training epoch and inference speed, with and without the addition of clause interaction hypergraph. The results are shown in Table V. It can be seen from the table that the clause interaction hypergraph does not bring too much parameter growth, but the training time has increased by about 250s every epoch. And the addition of CIHG accelerates the convergence of the model and reduces the number of training epochs. In the terms of inference speed, the model after adding CIHG still maintains the speed of more than 15d/s, which is enough to handle the daily new medical documents. Therefore, the addition of CIHG will increase computational complexity, but the increased cost is acceptable.

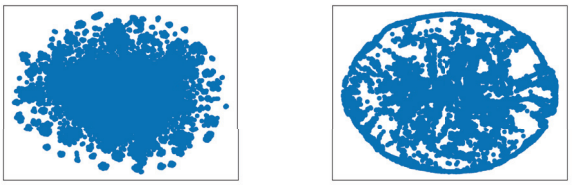
b) Limitations of AUC-ROC Scores: From Table II, we observe that there is an inconsistency in the AUC-ROC scores under the two settings of MIMIC-III. The CIHG brings a comprehensive improvement under the setting of 50, but it only improves precision and F1 under the setting of full and has little effect on AUC-ROC. We deem the reason is that most of the current methods did not consider the zero-shot problem in the MIMIC-III dataset. Table VI show the AUC-ROC and F1 scores of labels with different frequencies with and without CIHG. The comparison shows that CIHG can significantly increase model's performance with training data, even for few-shot labels. It shows the effectiveness of our method for joint

TABLE V
ANALYSIS OF COMPUTATIONAL COST WITH AND WITHOUT CLAUSE INTERACTION HYPERGRAPH BASED ON TWO REPRESENTATIVE MODELS. "M", "S", "EP" AND "D" DENOTE MILLION, SECOND, EPOCH AND DOCUMENT RESPECTIVELY.

Model	Parameter Amount	Training Time	Training Epoch	Inference Speed
MultiResCNN	6.58m	64s/ep	37	222d/s
MultiResCNN+CIHG	6.84m	323s/ep	30	31.9d/s
BiGRU	5.98m	198s/ep	72	23.4d/s
BiGRU+CIHG	6.51m	435s/ep	63	17.3d/s

TABLE VI
THE MACRO AUC-ROC AND F1 SCORES OF LABELS WITH DIFFERENT FREQUENCIES ON MIMIC-III-FULL BASED ON MULTIRESCNN WITH AND WITHOUT CIHG. "N" REPRESENTS THE NUMBER OF TIMES THE LABEL APPEARS IN THE TRAINING SET.

Type	No CIHG		With CIHG	
	AUC	F1	AUC	F1
Frequent (n>20)	93.9	23.2	95.0	31.1
Few (n≤20)	84.8	0.4	86.8	1.5
Zero (n=0)	66.8	0.0	66.8	0.0
All	90.4	9.0	90.6	9.7



(a) Without CIHG (b) With CIHG

Fig. 4. The visualization of label embedding with and without hypergraph.

TABLE VII
THE EUCLIDEAN DISTANCE OF LABEL EMBEDDING WITH AND WITHOUT HYPERGRAPH BETWEEN ICD CODES, WHERE THE CODE "96.72" AND "518.81" ARE GOLDEN LABELS.

ICD-9 code	
96.71	Continuous mechanical ventilation for less than 96 consecutive hours
96.72	Continuous mechanical ventilation for 96 consecutive hours or more
518.81	Acute respiratory failure
518.84	Acute and chronic respiratory failure

ICD-9 Code	No Hypergraph	With Hypergraph
96.71-96.72	2.4×10^{-5}	0.52
518.81-518.84	6.6×10^{-4}	4.17
96.71-(all_code)	2.54	13.32
96.72-(all_code)	2.54	12.67

modeling of documents and label descriptions. However, since there is no dedicated component to deal with the zero-shot problem, the model cannot assign zero-shot labels perfectly. These labels rarely appear in the test data (0.2% for zero-short labels and 2.4% for few-short labels), so they have little effect on the precision calculation but only affect the macro AUC-ROC. This explains why the improvement in AUC-ROC is not as significant in precision. Based on this explanation, the comparison between GRU-based method and MultiResCNN-based method scores in Table II shows that the GRU-based method has higher requirements for label frequency. With sufficient training data, the GRU-based method is stronger than CNN-based method. It also indicates that the key to further improve the performance of the medical code assignment task is to solve the zero-shot problem.

c) Embedding distance between ICD codes.: We conduct further analysis on one sample (HADM_ID 149498) to verify the effectiveness of the CIHG. In order to show the effectiveness of our method for the embedding of ICD codes, we calculated the Euclidean distance between some codes. The $(0, 1)$ normalization is used on ICD code embedding to ensure the comparability of Euclidean distances of different representations. The results are shown in Table VII. It can be seen that **(1) Distinction of similar codes:** The ICD code "96.71" and "96.72" are almost identical in text description but have completely opposite semantics. Our method achieves a distinction that traditional word2vec cannot achieve. **(2) Effectiveness of Interactive Connection.** The code "518.81" is connected to a clause node by the interaction connection. This is reflected in its high-strength representation that provides sufficient support for the correct prediction of the label. **(3) Aggregation of related labels.** The sum of Euclidean distances between all gold labels and "96.71" or "96.72" show that our method shortens the distance of the associated labels in one sample, which can be helpful for the prediction of easily confusing labels.

d) Embedding Visualization of Labels: In order to verify the constraint effect of clause interaction hypergraph on the label encoding, we use t-SNE [40] to visualize all the 22334 label representations of one instance encoded with and without hypergraph. The results are shown in Fig. 4. It can be seen that when there is no clause interaction hypergraph constraint, the labels show a divergent distribution. The label representations have learned a divergent structure, and do not obtain the co-occurrence constraint. With the constraint of clause hypergraph, the label representations show hierarchical and directional division. This demonstrates the effectiveness of

using document semantic structure to constrain label encoding.

V. Conclusion

In this paper, we propose a Clause Interaction Hypergraph (**CIHG**) to jointly model documents and label descriptions for the large-scale medical code assignment problem. Our method uses the interactive connection to construct the semantic fusion between the document and label, and uses the hypergraph to capture the co-occurrence relationship in document and labels. The combination of the two realizes the constraint of the document structure on the label encoding. Several experimental results show the effectiveness of our method. In the future, we will focus on solving the zero-short problem and extend this method to more fields.

References

[1] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: predicting clinical events via recurrent neural networks," in *MLHC*, ser. JMLR Workshop and Conference Proceedings, vol. 56. JMLR.org, 2016, pp. 301–318.

[2] Y. Du, P. Luo, X. Hong, T. Xu, Z. Zhang, C. Ren, Y. Zheng, and E. Chen, "Inheritance-guided hierarchical assignment for clinical automatic diagnosis," in *DASFAA (3)*, ser. Lecture Notes in Computer Science, vol. 12683. Springer, 2021, pp. 461–477.

[3] C. Sen, B. Ye, J. Aslam, and A. Tahmasebi, "From extreme multi-label to multi-class: A hierarchical approach for automated ICD-10 coding using phrase-level attention," *CoRR*, vol. abs/2102.09136, 2021.

[4] P. Rajendran, A. Zenonos, J. Spear, and R. Pope, "A meta-embedding-based ensemble approach for ICD coding prediction," *CoRR*, vol. abs/2102.13622, 2021.

[5] T. Zhou, P. Cao, Y. Chen, K. Liu, J. Zhao, K. Niu, W. Chong, and S. Liu, "Automatic ICD coding via interactive shared representation networks with self-distillation mechanism," in *ACL/IJCNLP (1)*. Association for Computational Linguistics, 2021, pp. 5948–5957.

[6] I. Chalkidis, M. Fergadiotis, S. Kotitsas, P. Malakasiotis, N. Aletras, and I. Androutsopoulos, "An empirical study on large-scale multi-label text classification including few and zero-shot labels," in *EMNLP (1)*. Association for Computational Linguistics, 2020, pp. 7503–7515.

[7] T. Vu, D. Q. Nguyen, and A. Nguyen, "A label attention model for ICD coding from clinical text," in *IJCAI*. ijcai.org, 2020, pp. 3335–3341.

[8] J. Mullenbach, S. Wiegreffe, J. Duke, J. Sun, and J. Eisenstein, "Explainable prediction of medical codes from clinical text," in *NAACL-HLT*. Association for Computational Linguistics, 2018, pp. 1101–1111.

[9] P. Cao, Y. Chen, K. Liu, J. Zhao, S. Liu, and W. Chong, "Hypercore: Hyperbolic and co-graph representation for automatic ICD coding," in *ACL*. Association for Computational Linguistics, 2020, pp. 3105–3114.

[10] S. Ji, S. Pan, and P. Marttinen, "Medical code assignment with gated convolution and note-code interaction," *CoRR*, vol. abs/2010.06975, 2020.

[11] G. Navarro, "A guided tour to approximate string matching," *ACM Comput. Surv.*, vol. 33, no. 1, pp. 31–88, 2001.

[12] L. S. Larkey and W. B. Croft, "Combining classifiers in text categorization," in *SIGIR*. ACM, 1996, pp. 289–297.

[13] J. Medori and C. Fairon, "Machine learning and features selection for semi-automatic ICD-9-CM encoding," in *Louhi@NAACL-HLT*. Association for Computational Linguistics, 2010, pp. 84–89.

[14] L. V. Lita, S. Yu, R. S. Niculescu, and J. Bi, "Large scale diagnostic code classification for medical patient records," in *IJCNLP*. The Association for Computer Linguistics, 2008, pp. 877–882.

[15] A. J. Perotte, R. Pivovarov, K. Natarajan, N. G. Weiskopf, F. D. Wood, and N. Elhadad, "Diagnosis code assignment: models and evaluation metrics," *J. Am. Medical Informatics Assoc.*, vol. 21, no. 2, pp. 231–237, 2014.

[16] H. Shi, P. Xie, Z. Hu, M. Zhang, and E. P. Xing, "Towards automated icd coding using deep learning," *arXiv preprint arXiv:1711.04075*, 2017.

[17] T. Baumel, J. Nassour-Kassis, R. Cohen, M. Elhadad, and N. Elhadad, "Multi-label classification of patient notes: Case study on ICD code assignment," in *AAAI Workshops*, ser. AAAI Workshops, vol. WS-18. AAAI Press, 2018, pp. 409–416.

[18] T. Bai and S. Vucetic, "Improving medical code prediction from clinical text via incorporating online knowledge sources," in *WWW*. ACM, 2019, pp. 72–82.

[19] P. Li and O. Milenkovic, "Inhomogeneous hypergraph clustering with applications," in *NIPS*, 2017, pp. 2308–2318.

[20] J. Han, B. Cheng, and X. Wang, "Open domain question answering based on text enhanced knowledge graph with hyperedge infusion," in *EMNLP (Findings)*. Association for Computational Linguistics, 2020, pp. 1475–1481.

[21] H. Chen, H. Yin, X. Sun, T. Chen, B. Gabrys, and K. Musial, "Multi-level graph convolutional networks for cross-platform anchor link prediction," in *KDD*. ACM, 2020, pp. 1503–1511.

[22] K. Ding, J. Wang, J. Li, D. Li, and H. Liu, "Be more with less: Hypergraph attention networks for inductive text classification," in *EMNLP (1)*. Association for Computational Linguistics, 2020, pp. 4927–4936.

[23] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, "Hypergraph neural networks," in *AAAI*. AAAI Press, 2019, pp. 3558–3565.

[24] S. Bai, F. Zhang, and P. H. S. Torr, "Hypergraph convolution and hypergraph attention," *Pattern Recognit.*, vol. 110, p. 107637, 2021.

[25] Y. Fang, S. Sun, Z. Gan, R. Pillai, S. Wang, and J. Liu, "Hierarchical graph network for multi-hop question answering," in *EMNLP (1)*. Association for Computational Linguistics, 2020, pp. 8823–8838.

[26] H. Wu, W. Chen, S. Xu, and B. Xu, "Counterfactual supporting facts extraction for explainable medical record based diagnosis with graph network," in *NAACL-HLT*. Association for Computational Linguistics, 2021, pp. 1942–1955.

[27] K. S. Jones, "A statistical interpretation of term specificity and its application in retrieval," *J. Documentation*, vol. 60, no. 5, pp. 493–502, 2004.

[28] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *NIPS*, 2013, pp. 3111–3119.

[29] K. Cho, B. van Merrienboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *EMNLP*. ACL, 2014, pp. 1724–1734.

[30] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *ICLR (Poster)*. OpenReview.net, 2017.

[31] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L. A. Celi, and R. G. Mark, "Mimic-iii, a freely accessible critical care database," *Scientific data*, vol. 3, no. 1, pp. 1–9, 2016.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Köpf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019, pp. 8024–8035.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR (Poster)*, 2015.

[34] F. Li and H. Yu, "ICD coding from clinical text using multi-filter residual convolutional neural network," in *AAAI*. AAAI Press, 2020, pp. 8180–8187.

[35] E. Alsentzer, J. R. Murphy, W. Boag, W. Weng, D. Jin, T. Naumann, and M. B. A. McDermott, "Publicly available clinical BERT embeddings," *CoRR*, vol. abs/1904.03323, 2019.

[36] A. Prakash, S. Zhao, S. A. Hasan, V. V. Datla, K. Lee, A. Qadir, J. Liu, and O. Farri, "Condensed memory networks for clinical diagnostic inferencing," in *AAAI*. AAAI Press, 2017, pp. 3274–3280.

[37] H. Shi, P. Xie, Z. Hu, M. Zhang, and E. P. Xing, "Towards automated ICD coding using deep learning," *CoRR*, vol. abs/1711.04075, 2017.

[38] M. Li, Z. Fei, M. Zeng, F. Wu, Y. Li, Y. Pan, and J. Wang, "Automated ICD-9 coding via A deep learning approach," *IEEE ACM Trans. Comput. Biol. Bioinform.*, vol. 16, no. 4, pp. 1193–1202, 2019.

[39] V. Van Asch, "Macro-and micro-averaged evaluation measures," *Belgium: CLiPS*, vol. 49, 2013.

[40] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.