CoRelation: Boosting Automatic ICD Coding Through Contextualized Code Relation Learning

Junyu Luo[1], Xiaochen Wang[1], Jiaqi Wang[1], Aofei Chang[1]
Yaqing Wang[2], Fenglong Ma[1]
[1]Pennsylvania State University, [2]Google
[1]{junyu, xcwang, jqwang, aofei, fenglong}@psu.edu, [2]yaqingwang@google.com

Abstract

Automatic International Classification of Diseases (ICD) coding plays a crucial role in the extraction of relevant information from clinical notes for proper recording and billing. One of the most important directions for boosting the performance of automatic ICD coding is modeling ICD code relations. However, current methods insufficiently model the intricate relationships among ICD codes and often overlook the importance of context in clinical notes. In this paper, we propose a novel approach, a contextualized and flexible framework, to enhance the learning of ICD code representations. Our approach, unlike existing methods, employs a dependent learning paradigm that considers the context of clinical notes in modeling all possible code relations. We evaluate our approach on six public ICD coding datasets and the experimental results demonstrate the effectiveness of our approach compared to state-of-the-art baselines.

Keywords: ICD Coding, Medical Text Mining

1. Introduction

The International Classification of Diseases (ICD) is a standard code system devised by the World Health Organization (WHO), which has gained widespread adoption in electronic health records (EHR) and health insurance systems. Currently, medical coders review medical records and other relevant documents to extract information about the patient's conditions, the services provided, and any procedures performed. They then translate this information into standardized ICD codes. This procedure is usually called *ICD coding*. However, manually assigning ICD codes is not only time-consuming but also error-prone.

To alleviate this issue, the concept of **automatic ICD coding** has been proposed recently, which is viewed as a multi-label text classification task (Mullenbach et al., 2018). The model is required to predict the probability distribution of ICD codes based on clinical notes, such as discharge summaries. Although various approaches (Xie et al., 2019; Li and Yu, 2020; Cao et al., 2020; Vu et al., 2020; Yuan et al., 2022; Yang et al., 2022b) have been proposed to enhance the performance of automatic ICD coding, they still have several limitations in *modeling relationships among ICD codes*.

• **Insufficiently Modeling Relations Among ICD Codes**: Existing methods typically utilize the ICD code ontology (Xie et al., 2019) or the co-occurrence graph (Cao et al., 2020) to model the relationships among ICD codes. However, these approaches only partially capture the code relations. The ontology solely encompasses the "child-parent" relation, which aids in enhancing the representation learning of rare ICD codes. On the other hand, the co-occurrence graph can merely indicate whether two codes appear together in the training set's ground truths. These two graphs are insufficiently complex to encompass the intricate relationships among ICD codes.

For instance, "696.0" (*Psoriasis arthropathy*) has been found to have a weak connection with "579.0" (*Celiac disease*) (Sanchez et al., 2018). However, they do not share a common parent code in the ontology and are not connected in the co-occurrence graph, which makes the existing approaches ineffective in modeling this relationship. Furthermore, both the co-occurrence graph and ICD code ontology fail to capture *exclusive code relations* between different code families, e.g., the relationship between code "780.60" (*Fever, unspecified*) and code "659.2" (*Maternal pyrexia during labor unspecified*)[1]. Consequently, there is a need for a novel and effective approach to model the intricate relationships among ICD codes.

• **Ignoring the Importance of Context**: The current approaches to ICD coding involve three main steps: (1) ICD code representation learning, (2) clinical note representation learning, and (3) ICD code extraction based on the outputs from the previous two steps. These approaches typically rely on the ICD code ontology or the co-occurrence graph to enhance the representation learning of ICD codes, which is *independent* of the second step. Consequently, the learned code relations remain **fixed** across all clinical notes.

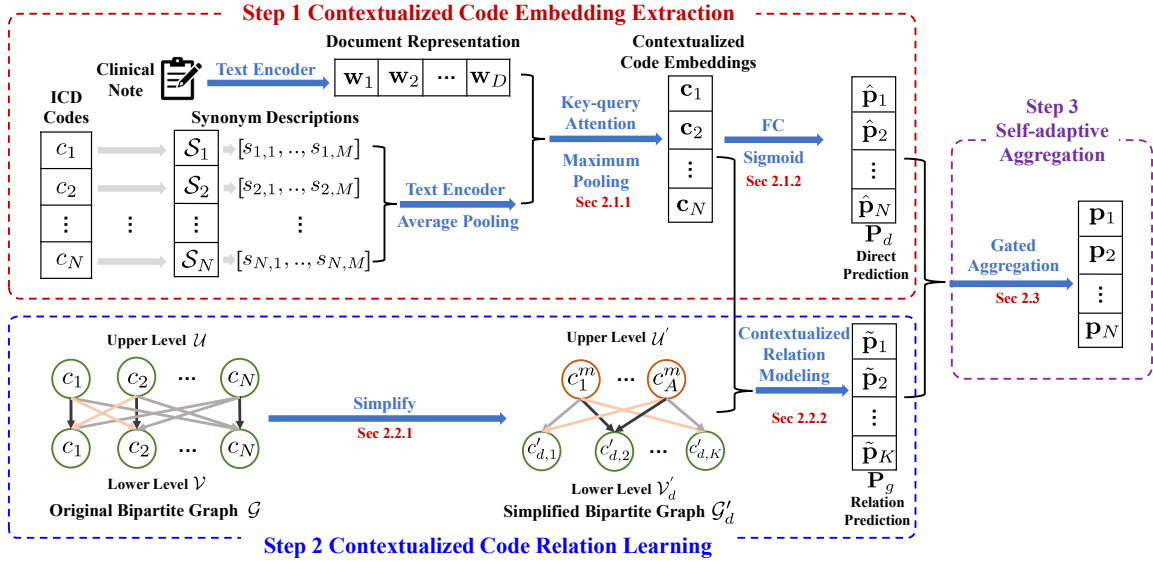However, we contend that the context of clinical notes plays a crucial role in ICD coding. For in-

---

[1]http://www.icd9data.com/2014/Volume1/780-799/780-789/780/780.60.htm

Figure 1: The Proposed CoRelation structure.

stance, codes belonging to the same sub-category, such as "488.81" (*Influenza A with pneumonia*) and "488.01" (*Influenza, avian with viral pneumonia*), are generally mutually exclusive since patients typically contract only one type of influenza. Nevertheless, exceptions may arise if a patient contracts both types of influenza simultaneously (Williams et al., 2011), necessitating the assignment of both codes. Therefore, it is imperative to consider the context in order to learn contextualized and dynamic code relations for this task.

Our Approach: To overcome these limitations, we propose a novel approach called CoRelation[2], which is a contextualized and flexible framework designed to enhance the learning of ICD code representations, as depicted in Figure 1. Unlike existing methods that simultaneously learn ICD code and clinical note representations, our approach employs a dependent learning paradigm.

In Step 1, CoRelation begins by learning contextualized code embeddings based on the input clinical note d and the multi-synonyms of ICD codes, which follows (Yuan et al., 2022). These learned code embeddings are then utilized to directly calculate the prediction probabilities $\mathbf{P}_d$ and are subsequently fed into the graph learning phase.

In Step 2, we construct a flexible and contextualized bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ for each clinical note d, enabling all codes to communicate with each other through an attention-based strategy. To improve computational efficiency, we propose to reduce the size of nodes in $\mathcal{V}$ by retaining only the top K codes with the highest direct code probabilities estimated in Step 1. Besides, we propose to

use coarse-grained ICD code categories to substitute the original codes in $\mathcal{U}$. Most importantly, we use the contextualized code embeddings learned in Step 1 as initializations for the graph update process. Thus, the learned relationship is dependent on both the codes and the processed clinical note. The updated code embeddings are then used to calculate the relation code probabilities $\mathbf{P}_g$.

In Step 3, we introduce a self-adaptive gating mechanism that automatically combines the two sets of probabilities, $\mathbf{P}_d$ and $\mathbf{P}_g$, to obtain the aggregated results for the final prediction. Experimental results on six datasets, including MIMIC-III-50, MIMIC-III-Full, MIMIC-IV-ICD9-50, MIMIC-IV-ICD9-Full, MIMIC-IV-ICD10-50, and MIMIC-IV-ICD10-Full, demonstrate the effectiveness of CoRelation compared to state-of-the-art baselines.

In addition, we also propose a novel selective training strategy to reduce the computation cost brought by relation modeling.

2. Methodology

The aim of automatic ICD coding is to extract a specific set of ICD codes $\mathcal{C} = \{c_1, \cdots, c_N\}$ from the given clinical note $d = [w_1, \cdots, w_D]$. Here, N and D represent the number of ICD codes and the word count of the clinical notes, respectively. Our model, as depicted in Figure 1, comprises three major steps: contextualized code embedding and direct code prediction, contextualized code relation learning, self-adaptive aggregation, and the selective training strategy. In the following subsections, we will provide detailed explanations of each module.

2.1. Contextualized Code Embedding and Direct Code Prediction

2.1.1. Contextualized Code Embedding

Inspired by the recent work MSMN (Yuan et al., 2022), we propose to use a synonym-based code-wise attention framework to extract contextualized code features from clinical notes by referring to the ICD code synonym description at the initial stage. Specifically, we employ the same text encoder to encode both the raw clinical note and the code descriptions. Thus, we have the clinical note embedding as follows:

$$[\mathbf{w}_1, \cdots, \mathbf{w}_D] = \text{TextEncoder}(d), \qquad (1)$$

where $\mathbf{w}_k$ is the representation of the k-th word in d. Each code $c_i \in \mathcal{C}$ contains M synonym descriptions $\mathcal{S}_i = [s_{i,1}, \cdots, s_{i,M}]$, and each $s_{i,j}$ consists of L words. Similarly, for each code synonym, we have $[\mathbf{ws}_1^{i,j}, \cdots, \mathbf{ws}_L^{i,j}] = \text{TextEncoder}(s_{i,j})$. We then use the average pool to obtain the overall synonym embedding as follows:

$$\mathbf{s}_{i,j} = \text{Pool}([\mathbf{ws}_1^{i,j}, \cdots, \mathbf{ws}_L^{i,j}]). \qquad (2)$$

Next, we utilize a standard key-query attention layer (Vaswani et al., 2017) to extract contextualized code embeddings from the text embedding sequence as follows:

$$\mathbf{c}_{i,j} = \text{KeyQueryAttention}(\mathbf{s}_{i,j}, [\mathbf{w}_1, \cdots, \mathbf{w}_D]), \qquad (3)$$

where $\mathbf{c}_{i,j}$ represents the contextualized code embedding. For each code, we use maximum pooling to obtain an overall representation across all the obtained synonym embeddings:

$$\mathbf{c}_i = \text{Pool}([\mathbf{c}_{i,1}, \cdots, \mathbf{c}_{i,M}]). \qquad (4)$$

2.1.2. Direct Code Prediction

Subsequently, we apply a fully connected layer to obtain the prediction weight embedding $\boldsymbol{\alpha}_i$ for code c_i from the synonymy code embeddings $[\mathbf{c}_{i,1}, \cdots, \mathbf{c}_{i,M}]$ learned by Eq. (3):

$$\boldsymbol{\alpha}_i = \text{FC}_\alpha(\text{Pool}([\mathbf{s}_{i,1}, \cdots, \mathbf{s}_{i,M}])). \qquad (5)$$

Finally, the direct code probability is calculated by taking the inner product of the prediction embedding and the extracted contextualized embedding, followed by applying the sigmoid activation function $\sigma()$:

$$\hat{\mathbf{p}}_i = \sigma(\boldsymbol{\alpha}_i \cdot \mathbf{c}_i). \qquad (6)$$

$\mathbf{P}_d = [\hat{\mathbf{p}}_1, \cdots, \hat{\mathbf{p}}_N]$ stands for the initial direct prediction result, and we are going to use contextualized code relation learning to improve it further.
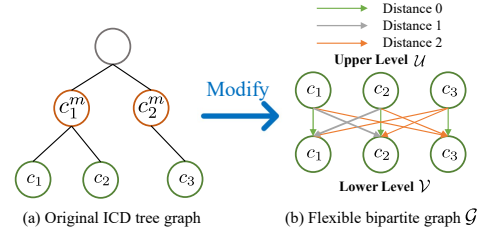

Figure 2: We modify the original ICD ontology into a directed flexible bipartite graph. There is an edge for each code pair, and the edge type depends on the distance between two codes on the original ICD ontology.

2.2. Contextualized Code Relation Learning

To enhance and streamline the ICD coding process, we propose a novel contextualized code relation learning module. This network aims to capture the intricate relationships between ICD codes under the context of the processed case, ultimately improving the accuracy of code assignment.

2.2.1. Code Relation Graph Construction

Directly modeling the potential relations on the original ICD ontology is complicated. As illustrated in Figure 2 (a), c_1 can directly exchange information with c_1^m. Although there is a path from c_1 to c_3, the long path decreases the shared information significantly. As a result, the model cannot effectively learn the relation between c_1 and c_3.

Bipartite Graph Construction. To address this issue, we employ a simplified, flexible bipartite graph $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ to represent the code-to-code connections, as depicted in Figure 2 (b), where $\mathcal{U} = \mathcal{V} = \mathcal{C}$ denote all the ICD codes, and $\mathcal{E}$ denotes the edges. Here, we use the distance (i.e., the number of hops) between any pair of ICD codes on the ontology to represent the edge type. *Such a design allows each code to communicate with all the codes directly, preserves parts of ICD ontology relations, and fully covers the relations used in the co-occurrence graph.* The updated embeddings of the lower-level nodes (i.e., $\mathcal{V}$) will be used to calculate relation-enhanced probabilities in the following subsection.

Bipartite Graph Simplification. Updating all ICD codes in the constructed graph is time-consuming. In the MIMIC-III-Full setting, the number of targeted ICD codes is 8,922, which means there are a total of $8{,}922 \times 8{,}922$ edges. To reduce the computational burden, we propose two tricks to reduce the complexity.

Lower-level: *Top-K Code Selection*. Intuitively, only the codes with higher probabilities calculated by Eq. (6) will be helpful in the final prediction, and the low-probability codes are less likely to affect

the result. Thus, we can decrease the size of $\mathcal{V}$ by selecting the top-K codes with the highest probabilities from $\mathbf{P}_d = [\hat{\mathbf{p}}_1, \cdots, \hat{\mathbf{p}}_N]$, and we use $\mathcal{V}'_d = \{c'_{d,1}, \cdots, c'_{d,K}\}$ denote the reduced lower-level code set, where $c'_{d,k}$ is a selected ICD code.

Upper-level: *Major Code Substitution*. In the ICD code ontology, each leaf code belongs to a coarse-grained category. For example, "250.03 (Diabetes mellitus without mention of complication, type I [juvenile type], uncontrolled)" belongs to the category "250 (Diabetes mellitus)", which is also called the major code. To reduce the size of $\mathcal{U}$, we propose to merge ICD codes belonging to the same coarse-grained category and use the major codes set $\mathcal{U}' = \{c_1^m, \cdots, c_A^m\}$ as the nodes, where A is the number of the major codes.

Simplified Graph. Given the new nodes $\mathcal{V}'_d$ and $\mathcal{U}'$, we then update the edge set $\mathcal{E}_d$ between $\mathcal{V}'_d$ and $\mathcal{U}'$. Let $\mathcal{E}'_d$ denote the new edges, where each edge type $e_{a,k}^d$ is still the distance between a major code c_a^m and a select ICD code $c'_{d,k}$. In such a way, we can have a simplified bipartite graph $\mathcal{G}'_d = (\mathcal{U}', \mathcal{V}'_d, \mathcal{E}'_d)$. Note that the set of $\mathcal{V}'_d$ is dependent on the clinical note d, but for all clinical notes, the major codes $\mathcal{U}'$ are the same. In such a way, the simplified graph can be considered personalized.

2.2.2. Contextualized Relation Modeling

To update the node embeddings on the simplified graph $\mathcal{G}'_d$, we first initialize the node embeddings of both $\mathcal{V}'_d$ and $\mathcal{U}'$. For each selected node $c'_{d,k} \in \mathcal{V}'_d$, we can use the contextualized code embedding learned by Eq. (4) as the initialization. For each major code $c_a^m \in \mathcal{U}'$, we use Eqs. (1-4) to calculate the initialized embeddings. The edge type $e_{a,k}^d$ is randomly initialized as a representation. Therefore, the input of the relation graph is **customized** for each processed note.

Next, we use the graph transformer (Dwivedi and Bresson, 2020) to model the interaction between codes on the simplified graph $\mathcal{G}'_d$ as follows:

$$\{\mathbf{U}^*, \mathbf{V}_d^*, \mathbf{E}_d^*\} = \text{GraphTransformer}(\mathcal{G}'_d). \quad (7)$$

The resulting enhanced embeddings $\mathbf{V}_d^* = [\tilde{\mathbf{c}}_1, \cdots, \tilde{\mathbf{c}}_K]$ from the lower-level are then used to estimate the relation enhanced code probabilities $\mathbf{P}_g = [\tilde{\mathbf{p}}_1, \cdots, \tilde{\mathbf{p}}_K]$ as follows:

$$\tilde{\mathbf{p}}_i = \sigma(\boldsymbol{\beta}_i \cdot \tilde{\mathbf{c}}_i), \quad (8)$$

where $\boldsymbol{\beta}_i$ is the prediction weight vector for relation-enhanced code embedding, which is obtained as follows:

$$\boldsymbol{\beta}_i = \text{FC}_\beta(\text{Pool}([\mathbf{s}_{i,1}, \cdots, \mathbf{s}_{i,M}])). \quad (9)$$

2.3. Self-adaptive Aggregation

To combine the direct prediction probabilities $\mathbf{P}_d = [\hat{\mathbf{p}}_1, \cdots, \hat{\mathbf{p}}_N]$ and the relation enhanced probabili-

ties $\mathbf{P}_g = [\tilde{\mathbf{p}}_1, \cdots, \tilde{\mathbf{p}}_K]$, we propose a novel self-adaptive gating module to aggregate the two results. In detail, we use the raw activation result (element-wise product $\odot$ of $\boldsymbol{\alpha}_i$ and $\mathbf{c}_i$) from the Section 2.1 to calculate the proportion value γ_i of relation enhanced prediction $\tilde{\mathbf{p}}_i$, making the model able to contextually decide the proportion of different inference sources:

$$\gamma_i = \sigma(\text{FC}_\gamma(\boldsymbol{\alpha}_i \odot \mathbf{c}_i)). \quad (10)$$

The final prediction result $\mathbf{p}_i$ is the gated combination of two prediction results as follows:

$$\mathbf{p}_i = (1 - \gamma_i)\hat{\mathbf{p}}_i + \gamma_i\tilde{\mathbf{p}}_i. \quad (11)$$

For the codes that are not selected (i.e., not among top K), we set the γ_i to 0 instead.

The training loss for the prediction results is:

$$\mathcal{L}_{CE} = \frac{1}{N}\sum_{i=1}^{N}\text{CrossEntropy}(\mathbf{p}_i, \mathbf{g}_i), \quad (12)$$

where $\mathbf{g}_i$ is the ground-truth label. In the meanwhile, we use the following loss to encourage the model to use less complex relation inference results by making the average proportion value γ_i as a loss term as follows:

$$\mathcal{L}_{comp} = \frac{1}{N}\sum_{i=1}^{N}\gamma_i. \quad (13)$$

The final training loss is:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda\mathcal{L}_{comp}, \quad (14)$$

where λ is the hyper-parameters for controlling the importance of L_{comp}.

2.4. Selective Training Strategy

In ICD coding, the majority of codes possess a limited number of positive labels. The average positive labels for each case under the MIMIC-III-Full setting is 15. However, there are total of 8,922 labels. Consequently, for full settings, most prediction results consist of negative labels. Despite contributing minimal gradients for model updates, these negative labels still demand equal computational resources during backpropagation. To circumvent this computational inefficiency, we propose a selective training algorithm, detailed in Algorithm 1. Specifically, we initially choose the top K_s training codes based on the output of CoRelation$_{wo}$ (steps 1-5) and then only perform backpropagation on these chosen codes (steps 6-8). Here, CoRelation$_{wo}$ means directly predicting the results without using contextualized code relation learning in Section 2.2. This selective training algorithm enables the model to focus on the most relevant

Algorithm 1: Selective Training

Input: Code set $\mathcal{C}$, dataset $\mathcal{D}$, and model parameter θ.

Output: Updated model parameter θ.

while *Training not finished* **do**

1. Sampling d, label set $\mathbf{G}$ from data set $\mathcal{D}$

$(d, \mathbf{G}) \leftarrow \mathcal{D}$

2. Perform forward propagation without code relation to estimate the scores.

$\hat{\mathbf{P}_{est}} = \text{CoRelation}_{wo}(\mathcal{C}, d)$

3. Choose top K_s codes.

$\mathcal{C}_{K_s} = \{\cdots, c_i, \cdots\}, i \in \text{Top-}K_s(\hat{\mathbf{P}_{est}})$

4. Adding additional K_s random codes, and the ground-truth codes to form $\mathcal{C}_{back}$.

$\mathcal{C}_{back} = \{\mathcal{C}_{K_s}, \mathcal{C}_{ground}, \mathcal{C}_{random}\}$

5. Select the labels of $\mathcal{C}_{back}$ from label set $\mathbf{G}$.

$\mathbf{G}_{back} = \{\cdots, \mathbf{g}_i, \cdots\}, c_i \in \mathcal{C}_{back}$

6. Perform forward calculation for the selected codes only.

$\mathbf{P}_{back} = \text{CoRelation}(\mathcal{C}_{back}, d)$

7. Calculate the loss $\mathcal{L}$.

$\mathcal{L} = LossFunc(\mathbf{P}_{back}, \mathbf{G}_{back})$

8. Update model parameter θ.

$\theta = \theta - \nabla L$

Table 1: Statistics of the six datasets.

Database	MIMIC-III		MIMIC-IV			
Code Version	ICD9		ICD9		ICD10	
Settings	50	Full	50	Full	50	Full
# of codes in $\mathcal{C}$	50	8,922	50	11,331	50	26,096
Training size	8,066	47,723	170,664	180,553	104,077	110,442
Validation size	1,573	1,631	6,406	7,110	3,805	4,017
Testing size	1,729	3,372	12,405	13,709	7,368	7,851
Avg # of tokens	1,478	1,434	1,499	1,459	1,687	1,662

codes during training, thereby reducing computational resources and avoiding the expenditure on negative labels that provide minimal gradient information.

3. Experiments

3.1. Experiment Settings

In this section, we introduce the experimental settings, which include the datasets, baselines, implementation details, and evaluation metrics.

3.1.1. Datasets

To evaluate our method, we utilize the ICD coding datasets derived from the MIMIC-III (Johnson et al., 2016) and MIMIC-IV (Johnson et al., 2020) projects. Specifically, we follow the settings of (Mullenbach et al., 2018; Yuan et al., 2022) to create the MIMIC-III-50 and MIMIC-III-Full datasets, and the settings of (Nguyen et al., 2023b) to create the MIMIC-IV-ICD9-50, MIMIC-IV-ICD9-Full, MIMIC-IV-ICD10-50, and MIMIC-IV-ICD10-Full datasets. The **50** setting focuses on evaluating the top 50 most frequent ICD codes, and the **Full** setting focuses on evaluating all potential ICD codes. The statistics of the six datasets are presented in Table 1.

3.1.2. Baselines

We divide baselines into two classes based on whether they utilize pre-trained language models (PLMs) as encoders: (1) **Non-PLM** meth-

ods include CAML (Mullenbach et al., 2018), MultiResCNN (Li and Yu, 2020), HyperCore (Cao et al., 2020), LAAT and JointLAAT (Vu et al., 2021), MSMN (Yuan et al., 2022), and TwoStage (Nguyen et al., 2023a); and (2) **PLM** methods include KEPT[3] (Yang et al., 2022b), HiLAT (Liu et al., 2022a), and PLM-ICD (Huang et al., 2022a).

We obtain the results of baselines either from the released trained models or the original papers if the codes are unavailable. It is worth noting that these models (Liu et al., 2022b; Ng et al., 2023; Yang et al., 2022a; Niu et al., 2023; Zhang et al., 2022) are not listed in baselines since they require additional information, such as annotation data, information source, and multiple retrieval stages.

3.1.3. Implementation Details

We implement our model in PyTorch, training them on an Ubuntu 20.04 system with 128 GB of memory and four NVIDIA A6000 GPUs.

For our model, we employed a single-layer LSTM with a hidden dimension of 512 as the TextEncoder. Word embeddings are initialized using GLOVE pre-trained embeddings on MIMIC-III notes, as described in the work (Vu et al., 2021). The attention dimension of the KeyQueryAttention and GraphTransformer component is configured to 256.

The top-K parameter K is set to 300 for all the **full** settings. For the **50** settings, $K = 50$ is fixed. For contextualized code relation learning, we use all the major codes as the upper node set $\mathcal{U}'$ of the relation graph for all the settings.

As for the optimizer, we use the Adam optimizer with an initial learning rate of $5e\text{-}4$, accompanied by linear decay, and early stop is applied by checking the Macro AUC score on the validation set. We employed the R-Drop (Wu et al., 2021) regularization technique, as introduced in the previous work MSMN (Yuan et al., 2022). The remaining parameters are summarized in Table 2.

---

[3]For KEPT, we only report the results of the 50 setting since the Full setting results utilize multiple methods to perform a multi-stage retrieval.

Table 2: Hypere parameter settings.

Settings	Train Epoch	R-Drop factor	$\mathcal{L}_{comp}$ weight β	Synonym number M
MIMIC-III-50	40	12.5	0.001	8
MIMIC-III-Full	30	5.0	0.01	8
MIMIC-IV-ICD9-50	15	5.0	0.001	8
MIMIC-IV-ICD9-Full	30	5.0	0.01	4
MIMIC-IV-ICD10-50	15	5.0	0.001	8
MIMIC-IV-ICD10-Full	30	5.0	0.01	4

Table 3: Results on the MIMIC-III-50 test set.

Category	Method	AUC		F1		Pre	
		Macro	Micro	Macro	Micro	P@5	P@8
PLM	HiLAT	92.7	95.0	69.0	<u>73.5</u>	68.1	55.4
	PLM-ICD	90.2	92.7	64.8	69.6	65.0	53.0
	KEPT	92.6	94.8	68.9	72.9	67.3	54.8
Non-PLM	CAML	87.5	90.9	53.2	61.4	60.9	-
	MultiResCNN	89.9	92.8	60.6	67.0	64.1	-
	HyperCore	89.5	92.9	60.9	66.3	63.2	-
	LAAT	92.5	94.6	66.6	71.5	67.5	54.7
	JointLAAT	92.5	94.6	66.1	71.6	67.1	54.6
	TwoStage	92.6	94.5	68.9	71.8	66.7	-
	MSMN	92.8	94.7	68.3	72.5	68.0	54.8
	CoRelation	**93.3**	**95.1**	**69.3**	**73.1**	**68.3**	**55.6**

3.1.4. Evaluation Metrics

Following previous studies (Mullenbach et al., 2018; Vu et al., 2021; Yuan et al., 2022), we report Macro & Micro AUC, Macro & Micro F1, and Precision at K (P@$\mathcal{K}$) metrics, where $\mathcal{K} = 5, 8, 15$ for different settings. The **Bold** notation indicates the best results among non-PLM techniques, while <u>Underline</u> notation signifies the best results when considering the PLM setting.

3.2. Results of the 50 Setting

In this section, we present the experimental outcomes for the three **50** settings, emphasizing the top 50 codes. In this setting, the size of $\mathcal{V}'_d = 50$ for all data. A comprehensive comparison of methodologies on the MIMIC-III and MIMIC-IV settings is provided in Table 3 and Table 4 respectively. Results reveal that our proposed model, CoRelation, outperforms all non-PLM methods across all metrics within the 50 settings. Even when compared to PLM methods, our method still demonstrates state-of-the-art performance on every metric, except for Micro F1 on the MIMIC-III-50 setting. Consequently, our method's advantage in the 50 settings is considerably substantial.

3.3. Results of the Full Setting

Next, we examine the results of three **Full** settings in Table 5 and Table 6. For the Full settings, only one PLM method is included since other PLM methods cannot handle the huge potential label space as illustrated in Table 1. CoRelation once again outperforms existing non-PLM methods across a majority of metrics within the MIMIC-III-Full, MIMIC-

IV-ICD9-Full, and MIMIC-IV-ICD10-Full contexts. Concurrently, it is important to note that the improvement of CoRelation in Precision at $\mathcal{K}$ (P@$\mathcal{K}$) scores is particularly significant, demonstrating that the proposed relation learning technique is effective in refining high-probability code predictions. Compared to PLM methods, although PLM-ICD exhibits superior performance in some P@$\mathcal{K}$ metrics, its underwhelming performance across other metrics - including its poor results in the 50 settings - renders it less competitive. Besides, our model operates with considerably fewer parameters (22 Million vs. 120 Million) compared to PLM methods. As such, the advantage of the CoRelation remains evident.

3.4. Ablation Study

In this section, we present an ablation study to examine the contribution of each proposed module to the overall performance of our model. The results are summarized in Table 7. The following notations represent different configurations of our model:

- **W/O Relation**: This notation signifies our proposed model without the application of the relation learning discussed in Section 2.2.

- **W/O Flexible Relation Graph (W/O FRG)**: This is the configuration where our proposed flexible relation graph, described in Section 2.2.1, is replaced with a fixed ICD ontology + co-occurrence graph, similar to (Cao et al., 2020).

- **W/O Context**: This denotes a version of our model that excludes the use of contextualized code embeddings, as per Section 2.2.2. The non-contextualized code description embeddings $\mathbf{s}_{i,j}$ are employed to initialize the relation graph $\mathcal{G}'_d$, similar to (Xie et al., 2019; Cao et al., 2020).

- **W/O Self-Adaptive Aggregation (W/O SAA)**: This indicates our proposed model without the integration of the self-adaptive aggregation discussed in Section 2.3. The $\mathbf{P}_g$ is directly employed as the final output.

Table 4: Results on the MIMIC-IV-50 test sets.

Category	Method	MIMIC-IV-**ICD9**-50					MIMIC-IV-**ICD10**-50				
		AUC		F1		Pre	AUC		F1		Pre
		Macro	Micro	Macro	Micro	P@5	Macro	Micro	Macro	Micro	P@5
PLM	PLM-ICD	95.0	96.4	71.4	75.5	62.4	93.4	95.6	69.0	73.3	64.6
Non-PLM	CAML	93.1	94.1	65.3	69.2	58.6	91.1	93.2	64.3	67.6	59.6
	LAAT	94.9	96.3	70.0	74.5	62.0	93.2	95.5	68.2	72.6	64.4
	JointLAAT	94.9	96.3	69.9	74.3	62.0	93.4	95.6	68.4	72.9	64.5
	MSMN	95.1	95.5	71.9	75.8	62.6	93.6	95.7	70.3	74.2	65.2
	CoRelation	**95.4**	**96.7**	**72.5**	**76.0**	**62.9**	**93.8**	**96.0**	**70.6**	**74.4**	**65.4**

Table 5: Results on the MIMIC-III-Full test set.

Method	AUC		F1		Pre		
	Macro	Micro	Macro	Micro	P@5	P@8	P@15
PLM-ICD	92.5	98.9	8.4	58.0	83.9	76.7	61.1
CAML	89.5	98.6	8.8	53.9	-	70.9	56.1
MultiResCNN	91.0	98.6	8.5	55.2	-	73.4	58.4
HyperCore	93.0	98.9	9.0	55.1	-	72.2	57.9
LAAT	91.9	98.8	9.9	57.5	81.3	73.8	59.1
JointLAAT	92.1	98.8	**10.7**	57.5	80.6	73.5	59.0
TwoStage	94.6	99.0	10.5	58.4	-	74.4	-
MSMN	95.0	**99.2**	10.3	58.4	82.5	75.2	59.9
CoRelation	**95.2**	**99.2**	10.2	**59.1**	83.4	76.2	60.7

We initiate the comparison with **W/O Relation** and the proposed model. Under both the 50 and Full settings, the omission of the relation results in a decline in most metrics, signifying the efficacy of our proposed relation learning in managing both frequent code settings and full code settings. Concurrently, we individually substitute components of the proposed contextualized relation learning to evaluate the efficiency of each proposed module. **W/O FRG** incurs the smallest decrease, yet it does not suggest that the contribution of flexible relation modeling is trivial. As the flexible relation graph centers on intricate and weak relationships, its enhancements are less apparent in quantitative analysis. The case studies in Section 3.5 will illustrate that the relations inferred within the flexible graph hold significant value. Finally, both **W/O SAA** and **W/O Context** contribute to substantial declines in the final performance. This underlines the significance of contextualized relation modeling and self-adaptive aggregation. In conclusion, all the proposed module contribute to the ultimate performance.

3.5. Case Study on Learned Code Relations

To better comprehend the efficacy of our proposed flexible, context-aware code relationship learning in facilitating code prediction, we conduct case studies to elucidate the relationships inferred from the code. The cases are depicted in Figure 3. Based on our analysis, the inferred relationships can be categorized into two types.

Type 1 relationships concentrate on differentiating similar codes by referencing codes that origi-
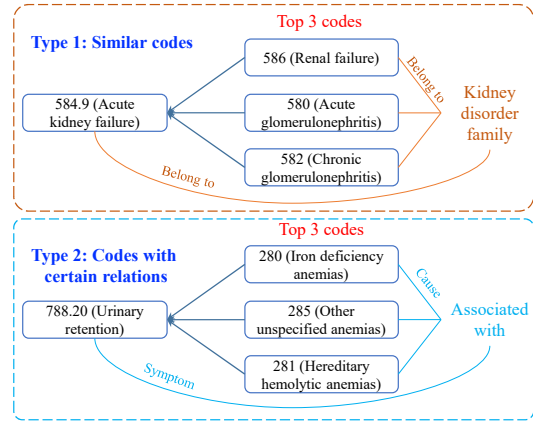

Figure 3: Two typical learned code relation cases.

nate from the same family. For example, the top-3 referenced codes for the code "584.9" (*Acute kidney failure*) are "586" (*Renal failure*), "580" (*Acute glomerulonephritis*), and "582" (*Chronic glomerulonephritis*). All these referenced codes, including "584.9", are part of the *Kidney disorder family*. The model, by taking these analogous codes into consideration, can make more precise predictions by discerning subtle differences.

Type 2 relationships focus on referencing codes that demonstrate specific associations. For instance, the top-3 cited codes for "788.20" (*Urinary retention*) are "280" (*Iron deficiency anemias*), "286" (*Other unspecified anemias*), and "281" (*Hereditary hemolytic anemias*), all falling under the *Anemia* category. The incidence of *Urinary Retention* is frequently associated with the severity of *anemia* (Hung et al., 2015). By utilizing flexible relationship learning, the model is capable of recognizing such associations and employing them to bolster prediction outcomes.

To summarize, the relationships inferred within the flexible graph are highly interpretable and exhibit a robust correlation with real-world medical practices.

3.6. Evaluation of Top-K Code Selection

In this section, we evaluate the efficacy of the top-K code selection strategy, delineated in Sec-

Table 6: Results on the MIMIC-IV-Full test sets.

Category	Method	MIMIC-IV-**ICD9**-Full					MIMIC-IV-**ICD10**-Full				
		AUC		F1		Pre	AUC		F1		Pre
		Macro	Micro	Macro	Micro	P@8	Macro	Micro	Macro	Micro	P@8
PLM	PLM-ICD	96.6	99.5	14.4	<u>62.5</u>	<u>70.3</u>	91.9	99.0	4.9	57.0	69.5
	CAML	93.5	99.3	11.1	57.3	64.9	89.9	98.8	4.1	52.7	64.4
	LAAT	95.2	99.5	13.1	60.3	67.5	93.0	99.1	4.5	55.4	67.0
Non-PLM	JointLAAT	95.6	99.5	14.2	60.4	67.5	93.6	99.3	5.7	55.9	66.9
	MSMN	**96.8**	**99.6**	13.9	61.2	68.9	97.1	**99.6**	5.4	55.9	67.7
	CoRelation	**96.8**	99.5	**15.0**	**62.4**	**70.1**	**97.2**	**99.6**	**6.3**	**57.8**	**70.0**

Table 7: Results of ablation experiments on the MIMIC-III datasets.

Dataset	MIMIC-III-50						MIMIC-III-Full						
Method	AUC		F1		Pre		AUC		F1		Pre		
	Macro	Micro	Macro	Micro	P@5	P@8	Macro	Micro	Macro	Micro	P@5	P@8	P@15
CoRelation	**93.3**	**95.1**	**69.3**	**73.1**	**68.3**	**55.6**	**95.2**	**99.2**	10.2	**59.1**	**83.4**	**76.2**	**60.7**
W/O Relation	93.1	95.0	69.0	72.6	68.1	55.2	95.2	99.1	9.3	58.9	82.8	75.7	60.5
W/O FRG	93.2	95.1	69.0	72.9	68.2	55.5	95.1	99.2	10.0	58.8	83.3	76.0	60.5
W/O Context	92.0	93.7	66.4	70.0	66.2	53.8	95.0	99.1	**10.7**	57.9	81.4	74.3	59.4
W/O SAA	92.5	94.7	68.6	72.2	67.9	55.0	95.0	99.1	9.7	58.8	82.9	75.9	60.1

Table 8: Evaluation of K values on the MIMIC-III-Full.

K	AUC		Pre		Memory
	Macro	Micro	P@5	P@8	
300	95.2	99.2	83.4	76.2	9.15 GB
200	95.2	99.2	83.5	76.1	7.78 GB
100	95.2	99.2	83.1	75.8	6.41 GB
50	95.1	99.2	83.0	75.4	5.05 GB

Table 9: Evaluation of Major Code Substitution on the MIMIC-III-50 dataset.

Method	AUC		P@5	Memory	Time
	Macro	Micro			
Complete	93.3	95.1	68.4	6.0 GB	0.23s
Major	93.3	95.1	68.3	1.5 GB	0.05s

Table 10: GPU memory space comparison results on the MIMIC-III-Full setting.

Method	Memory
CoRelation Selective	9.15 GB
CoRelation W/O Selective	72.76GB

where the upper-level codes, denoted as $\mathcal{U}$, are not replaced with the major codes $\mathcal{U}'$. Conversely, **Major** refers to the implementation of the major code substitution strategy. The reported speed and memory costs are the average per-sample metrics observed during training. As can be deduced from Table 9, employing the major code substitution results in a substantial decrease in memory usage and computational time while sustaining a performance level comparable to the complete approach.

3.8. Evaluation of the Selective Training

To prove the efficiency of the proposed selective training method, we report the per-sample space cost results in Table 10. From Table 10, we can discover that our proposed selective training approach significantly reduces memory cost. By utilizing a more efficient training strategy that selectively samples the code space, our method drastically reduces GPU memory usage from 72.76 GB to just 9.15 GB by nearly 12.5% of the original cost.

4. Related Work

The goal of automatic ICD coding is to infer and assign ICD codes based on the textual clinical note. Currently, a majority of automatic ICD coding techniques, such as CAML (Mullenbach et al., 2018) and MultiResCNN (Li and Yu, 2020), employ a

tion 2.2.1. We manipulate K within a range of 50 to 300, and the corresponding performance metrics along with per-sample GPU memory costs are compiled in Table 8. It is clear that escalating values of K bolster the performance of our model, albeit at the cost of increased GPU memory consumption. Nevertheless, the marginal performance enhancement decelerates as K keeps surging, predominantly due to the dwindling influence of the remaining low-probability codes, elucidated in Section 2.2.1. This indicates that our proposed top-K code selection strategy is potent, without compromising the overall model performance.

3.7. Evaluation of Major Code Substitution

We also assess the efficacy of the proposed major code substitution strategy in Section 2.2.1, and the results are shown in Table 9 on the MIMIC-III-50 dataset. The term **Complete** indicates the strategy

dual framework for ICD code prediction. In particular, clinical notes and codes are independently converted into embeddings. Then, a code-wise attention framework is utilized to extract relevant information from the encoded clinical notes based on code embeddings. Further enhancements to this approach are proposed from multiple perspectives.

Some studies involve leveraging supplementary information or knowledge, for instance, using ICD code descriptions to initialize ICD code embeddings (Dong et al., 2021; Zhou et al., 2021). MSMN (Yuan et al., 2022) expands on this by incorporating synonym descriptions of ICD codes. Additionally, there have been works to improve code representations using ICD relation data. MSATT-KG (Xie et al., 2019) and Teng et al. utilize ICD ontology to enrich the initial code embeddings, while HyperCore (Cao et al., 2020) employs an extra co-occurrence graph to enhance code embeddings. However, as outlined in Section 1, those methods fall short in effectively modeling code relationships. Other research like LAAT (Vu et al., 2021) and TwoStage (Nguyen et al., 2023a) propose to predict codes in a hierarchical manner to optimize the final prediction outcomes. Concurrently, numerous studies (Liu et al., 2022b; Ng et al., 2023; Zhang et al., 2022; Wang et al., 2020) explore more complex methodologies for incorporating external knowledge. Nonetheless, despite these enhancements, these techniques exhibit limited flexibility when dealing with various ICD coding settings due to their dependence on supplementary information sources or resources. Besides the knowledge, there are also studies (Yang et al., 2022a; Niu et al., 2023) that recommend the use of multiple-stage retrieval methods for performance enhancement.

The advent of pre-trained language models (PLMs) has inspired many works to leverage PLMs to enhance ICD coding performance (Huang et al., 2022b; Michalopoulos et al., 2022; Ng et al., 2023; Kang et al., 2023). However, these methods encounter drawbacks due to the substantial computational cost and the over-fitting problem of PLM models. Furthermore, these PLM-based methods frequently under-perform when compared to simpler baseline models, such as LSTM and CNN (Ji et al., 2021; Pascual et al., 2021). Despite these drawbacks, certain approaches, such as KEPT (Yang et al., 2022b) and HiLAT (Liu et al., 2022a), have succeeded in markedly improving the performance of PLM-based methods. They achieve this through the application of prompt-based prediction and hierarchical encoding methods. Nevertheless, these approaches still grapple with the issue of high computational costs.

5. Conclusion

In this paper, we propose a novel contextualized code relation-enhanced ICD coding model. The proposed model, referred to as CoRelation, aims to model the complex yet contextualized relations among ICD codes. CoRelation delivers state-of-the-art performance in comparison to current advanced ICD coding systems on six ICD coding datasets, yet it does so while consuming fewer computational resources without using pre-trained language models. Furthermore, we have undertaken an exploration of the learned code relation within our proposed method. The evidence suggests that our approach is also highly explainable.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Shengping Liu, and Weifeng Chong. 2020. Hypercore: Hyperbolic and co-graph representation for automatic icd coding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3105–3114.

Hang Dong, Víctor Suárez-Paniagua, William Whiteley, and Honghan Wu. 2021. Explainable automated coding of clinical notes using hierarchical label-wise attention networks and label embedding initialisation. *Journal of biomedical informatics*, 116:103728.

Vijay Prakash Dwivedi and Xavier Bresson. 2020. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. 2022a. PLM-ICD: Automatic ICD coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10–20, Seattle, WA. Association for Computational Linguistics.

Chao-Wei Huang, Shang-Chi Tsai, and Yun-Nung Chen. 2022b. Plm-icd: Automatic icd coding with pretrained language models. In *Proceedings of the 4th Clinical Natural Language Processing Workshop*, pages 10–20.

Szu-Chun Hung, Ko-Lin Kuo, Ching-Hsiu Peng, Che-Hsiung Wu, Yi-Chun Wang, and Der-Cherng Tarng. 2015. Association of fluid retention with anemia and clinical outcomes among patients with chronic kidney disease. *Journal of the American Heart Association*, 4(1):e001480.

Shaoxiong Ji, Matti Hölttä, and Pekka Marttinen. 2021. Does the magic of bert apply to medical code assignment? a quantitative study. *Computers in biology and medicine*, 139:104998.

Alistair Johnson, Lucas Bulgarelli, Tom Pollard, Steven Horng, Leo Anthony Celi, and Roger Mark. 2020. Mimic-iv. *PhysioNet. Available online at: https://physionet. org/content/mimiciv/1.0/(accessed August 23, 2021).*

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(1):1–9.

Beichen Kang, Xiaosu Wang, Yun Xiong, Yao Zhang, Chaofan Zhou, Yangyong Zhu, Jiawei Zhang, and Chunlei Tang. 2023. Automatic icd coding based on segmented clinicalbert with hierarchical tree structure learning. In *International Conference on Database Systems for Advanced Applications*, pages 250–265. Springer.

Fei Li and Hong Yu. 2020. Icd coding from clinical text using multi-filter residual convolutional neural network. In *AAAI*, pages 8180–8187.

Leibo Liu, Oscar Perez-Concha, Anthony Nguyen, Vicki Bennett, and Louisa Jorm. 2022a. Hierarchical label-wise attention transformer model for explainable icd coding. *Journal of biomedical informatics*, 133:104161.

Zichen Liu, Xuyuan Liu, Yanlong Wen, Guoqing Zhao, Fen Xia, and Xiaojie Yuan. 2022b. Treeman: Tree-enhanced multimodal attention network for icd coding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 3054–3063.

George Michalopoulos, Michal Malyska, Nicola Sahar, Alexander Wong, and Helen Chen. 2022. Icdbigbird: a contextual embedding model for icd code classification. *arXiv preprint arXiv:2204.10408*.

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

Boon Liang Clarence Ng, Diogo Santos, and Marek Rei. 2023. Modelling temporal document sequences for clinical icd coding. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1632–1641.

Thanh-Tung Nguyen, Viktor Schlegel, Abhinav Kashyap, and Stefan Winkler. 2023a. A two-stage decoder for efficient icd coding. *arXiv preprint arXiv:2306.00005*.

Thanh-Tung Nguyen, Viktor Schlegel, Abhinav Kashyap, Stefan Winkler, Shao-Syuan Huang, Jie-Jyun Liu, and Chih-Jen Lin. 2023b. Mimic-iv-icd: A new benchmark for extreme multilabel classification. *arXiv preprint arXiv:2304.13998*.

Kunying Niu, Yifan Wu, Yaohang Li, and Min Li. 2023. Retrieve and rerank for automated icd coding via contrastive learning. *Journal of Biomedical Informatics*, page 104396.

Damian Pascual, Sandro Luck, and Roger Wattenhofer. 2021. Towards bert-based automatic icd coding: Limitations and opportunities. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 54–63.

Isabelle M Sanchez, Wei Jiang, Eric J Yang, Rasnik K Singh, Kristen Beck, Claire Liu, Ladan Afifi, and Wilson Liao. 2018. Enteropathy in psoriasis: A systematic review of gastrointestinal disease epidemiology and subclinical inflammatory and functional gut alterations. *Current Dermatology Reports*, 7:59–74.

Fei Teng, Wei Yang, Li Chen, LuFei Huang, and Qiang Xu. 2020. Explainable prediction of medical codes with knowledge graphs. *Frontiers in bioengineering and biotechnology*, 8:867.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30:5998–6008.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2020. A label attention model for icd coding from clinical text. *arXiv preprint arXiv:2007.06351*.

Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2021. A label attention model for icd coding from clinical text. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3335–3341.

Qiong Wang, Zongcheng Ji, Jingqi Wang, Stephen Wu, Weiyan Lin, Wenzhen Li, Li Ke, Guohong Xiao, Qing Jiang, Hua Xu, et al. 2020. A study of entity-linking methods for normalizing chinese diagnosis and procedure terms to icd codes. *Journal of Biomedical Informatics*, 105:103418.

Derek J Williams, Matthew Hall, Thomas V Brogan, Reid WD Farris, Angela L Myers, Jason G Newland, and Samir S Shah. 2011. Influenza coinfection and outcomes in children with complicated pneumonia. *Archives of pediatrics & adolescent medicine*, 165(6):506–512.

Lijun Wu, Juntao Li, Yue Wang, Qi Meng, Tao Qin, Wei Chen, Min Zhang, Tie-Yan Liu, et al. 2021. R-drop: Regularized dropout for neural networks. *Advances in Neural Information Processing Systems*, 34:10890–10905.

Xiancheng Xie, Yun Xiong, Philip S Yu, and Yangyong Zhu. 2019. Ehr coding with multi-scale feature attention and structured knowledge graph propagation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 649–658.

Zhichao Yang, Sunjae Kwon, Zonghai Yao, and Hong Yu. 2022a. Multi-label few-shot icd coding as autoregressive generation with prompt. *arXiv preprint arXiv:2211.13813*.

Zhichao Yang, Shufan Wang, Bhanu Pratap Singh Rawat, Avijit Mitra, and Hong Yu. 2022b. Knowledge injected prompt based fine-tuning for multi-label few-shot icd coding. *arXiv preprint arXiv:2210.03304*.

Zheng Yuan, Chuanqi Tan, and Songfang Huang. 2022. Code synonyms do matter: Multiple synonyms matching network for automatic icd coding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 808–814.

Shurui Zhang, Bozheng Zhang, Fuxin Zhang, Bo Sang, and Wanchun Yang. 2022. Automatic icd coding exploiting discourse structure and reconciled code embeddings. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2883–2891.

Tong Zhou, Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, Kun Niu, Weifeng Chong, and Shengping Liu. 2021. Automatic icd coding via interactive shared representation networks with self-distillation mechanism. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5948–5957.