Multitask Balanced and Recalibrated Network for Medical Code Prediction

WEI SUN and SHAOXIONG JI, Aalto University, Finland
ERIK CAMBRIA, Nanyang Technological University, Singapore
PEKKA MARTTINEN, Aalto University, Finland

Human coders assign standardized medical codes to clinical documents generated during patients' hospitalization, which is error prone and labor intensive. Automated medical coding approaches have been developed using machine learning methods, such as deep neural networks. Nevertheless, automated medical coding is still challenging because of complex code association, noise in lengthy documents, and the imbalanced class problem. We propose a novel neural network, called the Multitask Balanced and Recalibrated Neural Network, to solve these issues. Significantly, the multitask learning scheme shares the relationship knowledge between different coding branches to capture code association. A recalibrated aggregation module is developed by cascading convolutional blocks to extract high-level semantic features that mitigate the impact of noise in documents. Also, the cascaded structure of the recalibrated module can benefit learning from lengthy notes. To solve the imbalanced class problem, we deploy focal loss to redistribute the attention on low- and high-frequency medical codes. Experimental results show that our proposed model outperforms competitive baselines on a real-world clinical dataset called the Medical Information Mart for Intensive Care (MIMIC-III).

CCS Concepts: • **Applied computing** → **Health care information systems**; Document management and text processing; • **Computing methodologies** → *Natural language processing;*

Additional Key Words and Phrases: Medical code prediction, multitask learning, imbalanced class problem, balanced and recalibrated network

ACM Reference format:
Wei Sun, Shaoxiong Ji, Erik Cambria, and Pekka Marttinen. 2022. Multitask Balanced and Recalibrated Network for Medical Code Prediction. *ACM Trans. Intell. Syst. Technol.* 14, 1, Article 17 (November 2022), 20 pages.
https://doi.org/10.1145/3563041

1 INTRODUCTION

Clinicians write discharge summaries based on different kinds of clinical notes, such as diagnosis reports, prescriptions, and treatment procedure documents. Healthcare institutions annotate

This work was supported by the Academy of Finland (grant no. 336033) and EU H2020 (grant no. 101016775). We acknowledge the computational resources provided by the Aalto Science-IT project. The authors wish to acknowledge CSC - IT Center for Science, Finland, for computational resources.
Authors' addresses: W. Sun, S. Ji (corresponding author), and P. Marttinen, Konemiehentie 2, 02150 Espoo, Finland; emails: {wei.sun, shaoxiong.ji, pekka.marttinen}@aalto.fi; E. Cambria, 50 Nanyang Ave, Singapore 639798, Singapore; email: cambria@ntu.edu.sg.

2157-6904/2022/11-ART17
https://doi.org/10.1145/3563041

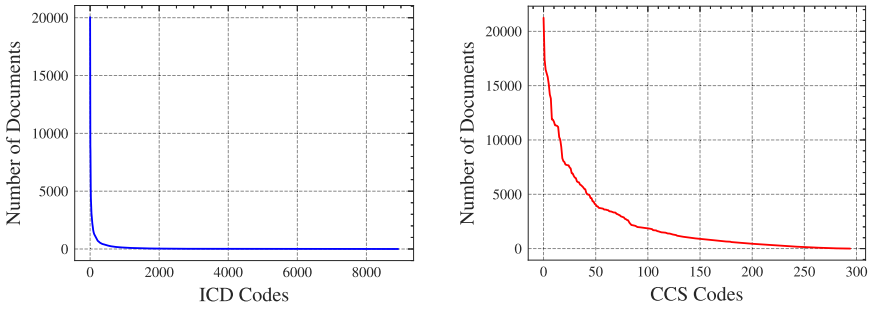


Fig. 1. Distributions of ICD and CCS codes in the MIMIC-III dataset. We omit a multitude of x-axis tick labels for better demonstration. The ICD codes are sorted by code frequency, which decreases from left to right.

these notes with standardized medical codes to facilitate information acquisition and management. The International Classification of Disease (ICD),[1] one of the most widely used medical coding systems, is maintained by the World Health Organization (WHO). The ICD coding system converts disease, pathology, symptoms, and signs into standard ICD codes, which is helpful in various medically related services, including insurance reimbursement [33], statistical data analysis, and clinical decision support [10]. Since medical code annotation by humans is error prone [32] and labor intensive [38], a surge of feature engineering-based machine learning [20, 35] and deep learning [15, 24, 38, 40] approaches have been proposed for automating the medical coding task.

However, the automatic medical coding task is still challenging, mainly due to the following three aspects.

Imbalanced Class Problem: Many medical coding datasets, such as the third version of the Medical Information Mart for Intensive Care (MIMIC-III), suffer from a severe imbalanced class problem. We take the MIMIC-III dataset as an example, the frequency distribution of ICD codes and Clinical Classifications Software (CCS) codes demonstrated in Figures 1(a) and 1(b), respectively. The main reason is that people encounter diseases such as hyperlipidemia and type II diabetes that are more frequent than other diseases, such as angioneurotic edema and quadriplegia. Model learning will be biased toward frequent labels if we train models on an imbalanced dataset without any rebalance strategies. Therefore, rebalancing the learning of low- and high-frequency code can improve the performance of medical coding approaches.

Some advanced ICD coding approaches mitigate the imbalanced class problem based on the nature of ICD codes. For example, the **La**bel **at**tention model (LAAT) [40] proposed the hierarchical joint learning mechanism to capture the structure of ICD codes to help predict low-frequency ICD codes.

Inspired by the imbalanced problem between easy and hard samples in the object detection task, we regard low- and high-frequency medical codes as hard-classified and easy-classified samples. Our framework leverages focal loss [26] to rebalance the attention on low- and high-frequency code by adjusting loss weights dynamically.

Code Association: There are some connections between medical codes. For example, in the ICD taxonomy system, "427.31" and "427.89" represent "Atrial fibrillation" and "Other specified cardiac dysrhythmias," respectively, which can be classified into "Dysrhythmia." It is challenging to capture associations between medical codes to facilitate the automatic medical coding task. Existing ICD coding models, such as **Multi**-Filter **Res**idual **C**onvolutional **N**eural **N**etwork (MultiResCNN) [24]

[1] https://www.who.int/standards/classifications/classification-of-diseases.

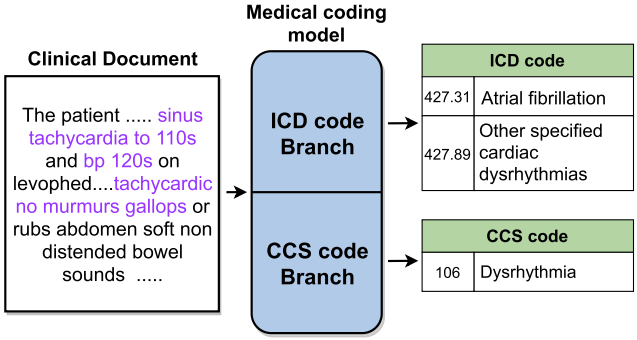


Fig. 2. An example of multitask coding with ICD and CCS codes.

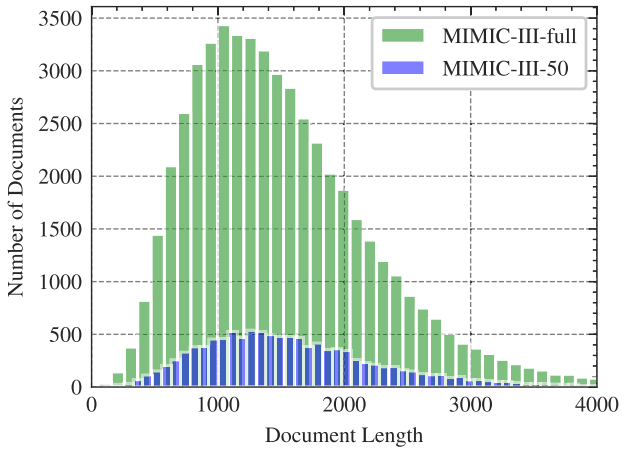


Fig. 3. Distributions of clinical document lengths in the MIMIC-III datasets.

and **C**onvolutional **A**ttention for **M**ulti-**L**abel classification (CAML) [31], did not take medical code association into consideration.

We design a multitask learning scheme to effectively transfer ICD code association knowledge from the auxiliary task branch. The CCS system maps high-dimensional ICD codes into low-dimensional CCS codes. The code projection system is based on the predefined medical knowledge of code association information provided by the Healthcare Cost and Utilization Project (HCUP). The object is to jointly train two medical coding branches: ICD and CCS coding. The CCS coding branch is an auxiliary task to transfer ICD code association information. Figure 2 shows an example of two-branch multitask medical coding, in which two ICD codes —"427.31" (Atrial fibrillation) and "427.89" (Other specified cardiac dysrhythmias) map to the same CCS code "106" (Dysrhythmia).

Noisy and Lengthy Document: Clinical documents contain noisy information, including error spelling and incoherent clinical notes, affecting representation learning from text. Learning rich and robust document features is required to provide reliable medical coding results. Moreover, discharge summaries in the MIMIC-III dataset are extremely long; 96.3% and 97.2% documents' lengths exceed 512 tokens in the MIMIC-III-full and MIMIC-III-50 datasets, respectively. Figure 3 shows that most documents' lengths are on the interval of [1000, 2000], which are quite long compared with texts in other domains, such as the Internet Movie Database (IMDB) movie review dataset [29]. To improve feature learning from clinical documents, we design a module called the

Recalibrated **A**ggregation **M**odule (RAM). The RAM suppresses the noise in the clinical notes by injecting contextually enhanced document features. The cascaded convolution structure of RAM provides the model with the capability to better deal with lengthy clinical documents.

To address these three problems, we propose a novel framework called the **M**ultitask b**A**lanced and **R**ecalibrated **N**etwork (MARN) for medical code prediction. This article is an extension of our previous work, which proposed a model called MT-RAM [38] to jointly train two different medical coding branches and achieve competitive improvement on overall evaluation metrics. Our additional work includes the efforts to balance learning from frequent and infrequent codes, the extension of experiments on a full-code dataset with improved performance, and further analysis of the problem of imbalanced class and code association. Our proposed MARN model combines multitask learning, the bidirectional gated recurrent unit (BiGRU), RAM, the label-aware attention mechanism, and focal loss. We summarize our main contributions as follows:

- This article deals with the imbalanced class problem and leverages focal loss to dynamically redistribute the weight between low- and high-frequency codes.
- We utilize a multitask learning scheme to jointly train two medical coding systems with different granularities for capturing code associations.
- The **R**ecalibrated **A**ggregation **M**odule (RAM) is designed to refine textual features extracted from lengthy and noisy clinical notes.
- Experimental results show strong performance of our model across different evaluation metrics on a widely used dataset, MIMIC-III, in a comparison against several strong baseline models.

Our article is organized as follows: Section 2 introduces related work; Section 3 describes our proposed MARN model; Section 4 describes a series of experiments and explores components of the MARN; Section 5 discusses future directions; and Section 6 presents our conclusions.

2 RELATED WORK

Automatic Medical Coding. Automatic medical coding is a challenging but essential task in medical text mining [35]. Early automatic medical coding models mainly depended on complicated handcrafted document features. Larkey and Croft [22] designed an ICD code classifier by conflating a potpourri of machine learning components, including K-nearest neighbor, relevance feedback, and Bayesian independence classifiers. Perotte et al. [35] proposed two ICD coding models: a flat classifier and a hierarchy-based support vector machine (SVM) classifier. Comparison experiments demonstrated the superiority of the hierarchy-based model because it can capture the hierarchical structure of ICD codes, which can benefit the prediction of ICD codes.

Recent years have witnessed the advances in deep learning approaches. Mullenbach et al. [31] proposed the **C**onvolutional **A**ttention network for **M**ulti-**L**abel classification (CAML) for automatic ICD coding. Li and Yu [24] designed a **Mul**ti-Filter **Res**idual **C**onvolutional **N**eural **N**etwork (MultiResCNN). Ji et al. [13] developed a dilated convolutional network. Dong et al. [9] devised a hierarchical label-wise attention network. Vu et al. [40] developed a label attention model (LAAT) to predict ICD codes. Xie et al. [42] designed a multi-scale feature attention and structured knowledge graph propagation (MSATT-KG), which is the combination of a densely connected convolutional neural network (CNN), multi-scale feature attention, and graph convolutional neural networks. The densely connected CNN generates the n-gram features, and the multi-scale feature attention captures the most informative n-gram document features. Also, the MSAAT-KG uses a graph CNN to obtain the hierarchical structure of ICD codes and the semantics of each ICD code.

Imbalanced Class Problem. The imbalanced class problem is founded on the different distributions of class labels in the dataset [6]. The distribution of the label space demonstrates a

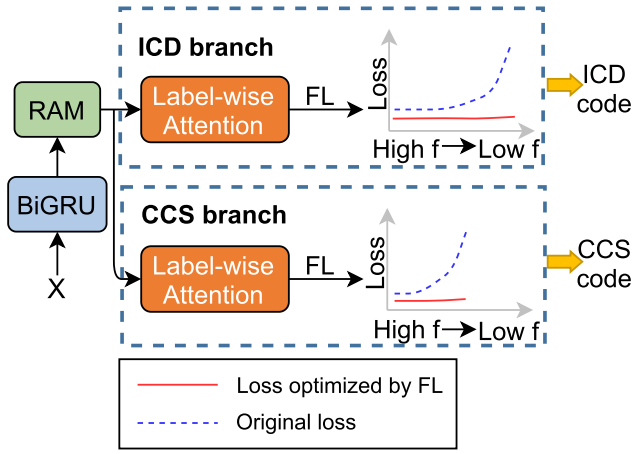


Fig. 4. Overall architecture of MARN. f represents the code frequency. After utilizing focal loss, the losses of low- and high-frequency medical codes can be balanced.

head-tail shape, where head and tail labels represent high- and low-frequency labels, respectively. Most multi-label datasets suffer from a serious imbalanced class problem [39]. The conventional way to alleviate the imbalanced class problem divides instances into majority (classes with notable instances) and minority groups (classes with few instances) [4]. Then, sampling algorithms are applied to reduce the number of majority samples or increase the number of minority samples. In past decades, many resampling approaches have been proposed, such as Random Over-Sampling (ROS), SMOTE [5] and Random Under-Sampling (RUS) [21]. However, reconstructing the data from a medical dataset is expensive and inefficient. Therefore, most automatic coding works tend to deal with the imbalanced class problem by leveraging the hierarchical structure of ICD codes. For example, LAAT [40] proposed the joint hierarchical mechanism to deal with the imbalanced class problem. In contrast, we determine to handle the imbalanced class problem by dynamically redistributing the loss weights between low- and high-frequency medical codes.

Multitask Learning. Multitask learning (MTL) is inspired by human activities in which people can utilize experience from other tasks to prompt the learning process of the new task [46]. MTL contributes to information communication between related tasks by sharing parameters and increases training efficiency [3, 44]. Additionally, MTL can alleviate the over-fitting problem by regularizing the learned model parameters to improve the model's generalization ability for each task branch [27]. Many MTL-based approaches have been proposed to verify the feasibility and effectiveness of applying the MTL scheme to medical natural language processing (NLP) tasks, such as medical named entity recognition [7, 47], clinical information extraction [2, 37], and mortality prediction [36]. However, only a few narratives deploy the MTL scheme on the automatic medical coding task. Interian et al. [12] studied two different healthcare tasks—medical code prediction and mortality prediction—to perform multitask training.

3 METHOD

This section introduces our proposed model, **M**ultitask b**A**alanced and **R**ecalibrated **N**etwork (MARN), which consists of an MTL scheme, the **R**ecalibrated **A**ggregation **M**odule (RAM), and optimization with focal loss (FL). The overall architecture of the MARN is presented in Figure 4. First, we input word embeddings of clinical notes pretrained with the word2vec algorithm [30]. Second, we adopt the BiGRU as the feature extractor to extract textual representations from medical documents. Next, the RAM is plugged in to improve the quality of document

features learned by the BiGRU and to better handle noisy and lengthy clinical documents. Then, we jointly train ICD and CCS coding branches by utilizing the MTL scheme to capture the associations among different medical codes. Finally, we utilize the FL to alleviate the imbalanced class problem by redistributing the loss weight on high-frequency and low-frequency labels.

3.1 Input Layer and Base Encoder

Let D be a clinical document consisting of n tokens, $\{w_1, w_2, \ldots, w_n\}$. The word embedding matrix is obtained by pretrained word2vec embeddings from each clinical document, denoted as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n]^{\mathrm{T}}$, assembling to a word a vector $\mathbf{x}_n$ whose word embedding size is d_e. We choose the BiGRU as the backbone neural network to extract features from clinical documents. Hidden states of the BiGRU on token x_i (where $i \in 1, 2, \ldots, n$) are denoted as

$$\overrightarrow{\mathbf{h}_i} = \overrightarrow{\mathrm{GRU}}(\mathbf{x}_i, \overrightarrow{\mathbf{h}_{i-1}}) \tag{1}$$

$$\overleftarrow{\mathbf{h}_i} = \overleftarrow{\mathrm{GRU}}(\mathbf{x}_i, \overleftarrow{\mathbf{h}_{i+1}}), \tag{2}$$

where $\overrightarrow{\mathrm{GRU}}$ and $\overleftarrow{\mathrm{GRU}}$ denote forward and backward GRUs, respectively. Bidirectional hidden states are obtained by horizontal concatenation of $\overrightarrow{\mathbf{h}_i}$ and $\overleftarrow{\mathbf{h}_i}$, which is presented as

$$\mathbf{h}_i = \mathrm{Concat}(\overrightarrow{\mathbf{h}_i}, \overleftarrow{\mathbf{h}_i}). \tag{3}$$

The dimension of each directional GRU is set as d_r so that the bidirectional hidden state $\mathbf{h}_i$ has dimension $\mathbb{R}^{2d_r}$. The final hidden representation matrix is represented as $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n]^{\mathrm{T}} \in \mathbb{R}^{n \times 2d_r}$.

3.2 Recalibrated Aggregation Module

We designed the **R**ecalibrated **A**ggregation **M**odule (RAM) to provide the model with better representation learning from lengthy and noisy document features, which is inspired by the squeeze-and-excitation networks in the computer vision domain [11]. Three observations motivate us to enhance the ability to handle the lengthy and noisy features. (1) When the document is very long, the BiGRU also encounters a vanishing gradient problem, which may affect the model's performance when stacking additional layers upon the recurrent layers. (2) Lower-level semantic features extracted by the BiGRU contain rich textural information with more document details, while higher-level document features capture abstract features with global receptive fields. The combination of multi-level features can benefit representation learning from clinical documents. (3) Higher-level semantic features provide contextual information for recalibrating noisy input document features. In the following, we will give the details about the calculation flow of the RAM and how the RAM addresses these three aspects.

The structure and feature flow of the RAM is illustrated in Figure 5. The RAM consists of three kinds of blocks: an Up-sample Block, Lateral-sample Block, and Down-sample Block, whose structure and dimensionalities are demonstrated in Figures 6(a), 6(b), and 6(c), respectively.

We construct our basic blocks with stacked small convolution filters (*kernel size* = 3) to have an appropriate receptive field [34]. To simplify the description, we regard the operations of Down-sample Block, Lateral-sample Block, and Up-sample Block as $\mathcal{F}_{\mathcal{DB}}(\cdot)$, $\mathcal{F}_{\mathcal{LB}}(\cdot)$, and $\mathcal{F}_{\mathcal{UB}}(\cdot)$, respectively. The calculations of three blocks could be represented as follows:

$$\mathcal{F}_{\mathcal{DB}}(\cdot) = \mathcal{F}^{l}_{3\times3}(\tanh(\mathcal{F}^{d}_{3\times3}(\cdot))) \tag{4}$$

$$\mathcal{F}_{\mathcal{LB}}(\cdot) = \mathcal{F}^{l}_{3\times3}(\tanh(\mathcal{F}^{l}_{3\times3}(\cdot))) \tag{5}$$

$$\mathcal{F}_{\mathcal{UB}}(\cdot) = \mathcal{F}^{l}_{3\times3}(\tanh(\mathcal{F}^{u}_{3\times3}(\cdot))), \tag{6}$$

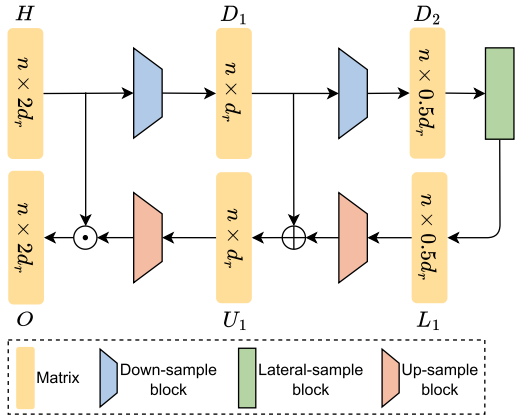


Fig. 5. The structure and feature flow of the RAM. ⊙ denotes element-wise multiplication. ⊕ represents the element-wise addition operation.

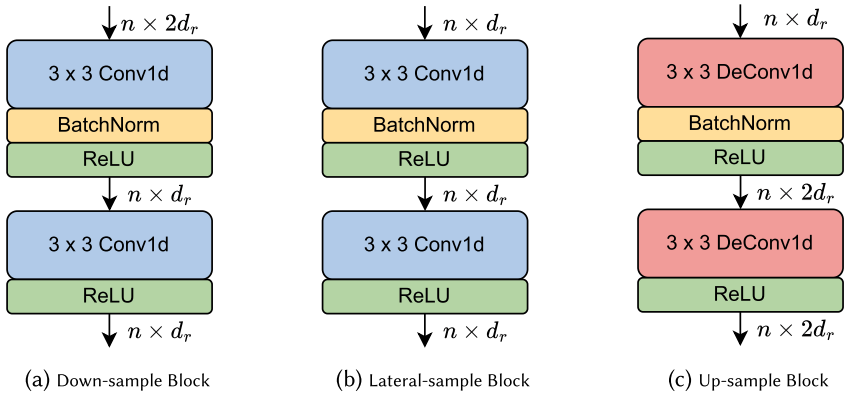


Fig. 6. Illustrations for the structure and dimension transformation of three basic blocks in the RAM.

where $\mathcal{F}^{d}_{3\times3}$ is a 3×3 convolution layer followed by the BatchNorm operation, where the number of output channels is reduced to half of the input channels. The operation of $\mathcal{F}^{l}_{3\times3}$ is quite similar to $\mathcal{F}^{d}_{3\times3}$ except that $\mathcal{F}^{l}_{3\times3}$ retains the input channels' number. $\mathcal{F}^{u}_{3\times3}$ performs the deconvolution operation that doubles the input channels, and the output feature is passed through the batch normalization layer.

The RAM is divided into three stages: feature abstraction, feature smoothing, and feature aggregation. First, the RAM utilizes cascaded convolutional blocks to get abstract features that have larger receptive fields compared with the input features of the BiGRU. Each convolutional layer incorporates k-grams (k is the convolutional kernel size) into 1-gram, so that large-scale n-grams' information can be embedded into 1-gram if input features are passed through cascaded convolutional blocks. In this way, the RAM has better ability to handle lengthy clinical documents. Second, the abstract features are processed to smooth the input features by convolutional blocks whose input dimension sizes are the same as the output ones. We denote this stage as feature smoothing because of no variation on feature dimensions. Third, we use deconvolutional blocks to restore feature dimensions (consistent receptive fields) and to avoid feature inconsistency[2] in order to

[2]Experimental results show that the deconvolutional up-sample block outperforms the simple linear projection across all scores by 0.05% to 1.73%.

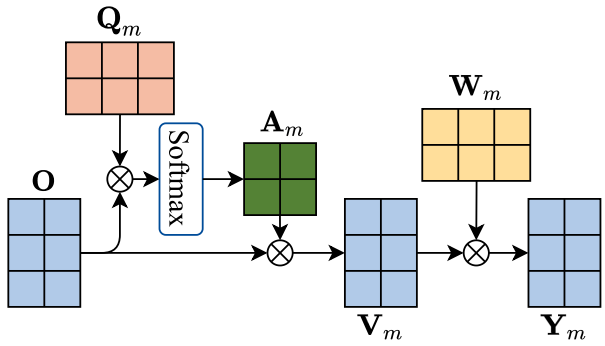


Fig. 7. An illustration of the label attention layer. ⊗ represents the matrix multiplication operation.

facilitate the aggregation of abstract features (containing context-aware information) and lower-level feature (containing detailed document information) to generate richer document representations for recalibrating noisy input features. When dealing with long clinical documents, the RAM does not worsen the vanishing gradient problem caused by the BiGRU backbone because of identity paths in the convolutional blocks and the activation function.

Feature Abstraction: We leverage two down-sample blocks to abstract the input document features $\mathbf{H}$. The computation process is denoted as

$$\mathbf{D}_1 = \mathcal{F}_{\mathcal{DB}}(\mathbf{H}) \tag{7}$$

$$\mathbf{D}_2 = \mathcal{F}_{\mathcal{DB}}(\mathbf{D}_1), \tag{8}$$

where the higher-level document features $\mathbf{D}_1 \in \mathbb{R}^{n \times d_r}$ and $\mathbf{D}_2 \in \mathbb{R}^{n \times 0.5 d_r}$ are extracted as the input features for next two stages.

Feature Smoothing: We use the lateral-sample block to further extract high-level features and take the feature $\mathbf{D}_2$ as input:

$$\mathbf{L}_1 = \mathcal{F}_{\mathcal{LB}}(\mathbf{D}_2), \tag{9}$$

where the output feature of this stage is represented as $\mathbf{L}_1 \in \mathbb{R}^{n \times 0.5 d_r}$.

Feature Aggregation: Two up-sample blocks are utilized to recover the dimension of the feature $\mathbf{L}_1$ for keeping the receptive field consistent when fusing the output features with original input features $\mathbf{H}$. At the same time, we use two different fusion operations, element-wise addition and multiplication, to inject lower-level features $\mathbf{H}$ and $\mathbf{D}_1$ into features containing rich contextual information. The output features $\mathbf{O}$ are followed by a tanh function to activate output features and a dropout layer is added after activated features to prevent overfitting. This conflation process is demonstrated as

$$\mathbf{U}_1 = \mathcal{F}_{\mathcal{UB}}(\mathbf{L}_1) \oplus \mathbf{D}_1, \tag{10}$$

$$\mathbf{O} = \tanh(\mathcal{F}_{\mathcal{UB}}(\mathbf{U}_1) \odot \mathbf{H}), \tag{11}$$

where $\mathbf{U}_1 \in \mathbb{R}^{n \times d_r}$ is the fused feature and $\mathbf{O} \in \mathbb{R}^{n \times 2 d_r}$ is the final output of RAM.

3.3 Attention Classification Layers

The feature $\mathbf{O}$ output by the RAM is label agnostic. We deploy the label-aware attention mechanism to connect the label information of each medical code with different positions of the clinical document feature vector $\mathbf{O}$. We set two medical coding branches, the ICD and CCS coding branches, with separate label-aware attention mechanisms, shown in Figure 7.

To simplify the description of the two coding branches, we combine two subscripts of ICD and CCS codes into one subscript m because the calculations are the same in two branches, where m denotes either ICD or CCS codes, and omit the bias item in the equations. The attention score of each medical code is computed as

$$\mathbf{A}_m = \text{Softmax}(\mathbf{O}\mathbf{Q}_m), \tag{12}$$

where $\mathbf{O}$ is the output feature from the RAM, $\mathbf{Q}_m \in \mathbb{R}^{2d_r \times d_m}$ denotes the trainable parameter matrix of the queries in the label-aware attention layer, and d_m refers to the number of target medical codes. The label-attentive document features $\mathbf{V}_m \in \mathbb{R}^{d_m \times 2d_r}$ are generated by multiplying attention score matrix $\mathbf{A}_m$ with the calibrated feature $\mathbf{O}$ from RAM:

$$\mathbf{V}_m = \mathbf{A}_m^{\mathrm{T}}\mathbf{O}. \tag{13}$$

The label-aware attention mechanism can capture code-related information and encode that information into documents. Then, the label-attentive features $\mathbf{V}_m$ are transformed into score vectors $\mathbf{Y}_m \in \mathbb{R}^{d_m \times 2d_r}$ of medical codes by using the fully connected layer. We pass the score vectors through the sum pooling operation followed by a sigmoid activation function to generate probabilities $\hat{\mathbf{y}}_m$ for final medical code prediction, which are denoted as:

$$\mathbf{Y}_m = \mathbf{W}_m\mathbf{V}_m^{\mathrm{T}} \tag{14}$$

$$\hat{\mathbf{y}}_m = \sigma(\text{Pooling}(\mathbf{Y}_m)), \tag{15}$$

where $\mathbf{W}_m \in \mathbb{R}^{d_m \times 2d_r}$ are the learnable parameters of the fully connected layer and σ is the sigmoid activation function.

3.4 Multitask Learning with Focal Loss

We perform multitask training to incorporate the two medical coding branches for the ICD and CCS codes, respectively. The probabilities of medical codes, $\hat{\mathbf{y}}_i$, produced by the label-aware classification layer, are fed into the loss function separately. The sparsity of codes poses a severe imbalanced class problem, which we alleviate using focal loss when training the model [26]. The focal loss of each medical coding branch m can be written as

$$\mathcal{FL}_m = \sum_{i=1}^{d_m}[-y_i\alpha(1-\hat{y}_i)^\gamma \log(\hat{y}_i) - (1-y_i)(1-\alpha)\hat{y}_i^\gamma \log(1-\hat{y}_i)], \tag{16}$$

which is the sum of losses over all d_m medical codes present in the m-th coding system. The parameter α represents the weighting factor that balances the loss for different classes ($y = 1$ or $y = 0$ depending on whether the code was present or not). We set $\alpha = 0.999$, which places a strong emphasis on instances with a certain code present (i.e., $y = 1$), to reflect the sparsity of the codes. In addition, the focal loss includes a modulating factor, $(1-\hat{y}_i)^\gamma$. If $\gamma > 0$, the modulating factor places less emphasis on confident predictions and more emphasis on uncertain predictions, further making the model focus on the learning of codes that are difficult to classify, which often are the low-frequency codes. The loss weights of low- and high-frequency codes are adjusted by the confidences of prediction dynamically. Note that for simplicity in Equation (16), we omitted the subscript m that identifies the code branch in the ground truth labels $\mathbf{y}_m$ and prediction probabilities $\hat{\mathbf{y}}_m$.

We treat the medical code prediction as a multitask problem and, consequently, formulate the joint loss as

$$\mathcal{FL}_M = \lambda_d\mathcal{FL}_d + \lambda_s\mathcal{FL}_s, \tag{17}$$

where $\mathcal{FL}_d$ and $\mathcal{FL}_s$ denote the focal losses for the ICD and CCS coding branches, respectively, and λ_d and λ_s are the loss weights of the ICD and CCS coding branches, respectively.

4 EXPERIMENTS

In this section, we evaluate the effectiveness of our proposed model MARN on public real-world datasets. The source code is available at https://github.com/VRCMF/MARN.

4.1 Datasets

MIMIC-III (ICD codes): The third version of the Medical Information Mart for Intensive Care (MIMIC-III)[3] is a large, open-access dataset consisting of clinical data associated with greater than 40,000 inpatients in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012 [16]. Following Mullenbach et al. [31] and Li and Yu [24], we use the discharge summaries as the input clinical documents. Human experts annotate each summary document with corresponding diagnosis and procedure codes. The first dataset is the full dataset, with 8,921 unique ICD-9-CM codes in total. The MIMIC-III full-code dataset has 52,722 discharge summaries, with 47,719, 1,631, and 3,372 documents for training, validation, and testing, respectively. The second dataset is the MIMIC-III top-50 codes dataset, for which we divide all discharge summary documents based on the patient IDs and generate the top 50 most frequent ICD codes. The top-50 dataset has 8,067 discharge summaries for training, and 1,574 and 1,730 documents for validation and testing, respectively. We refer to the two datasets of ICD codes as **MIMIC-III-full (ICD codes)** and **MIMIC-III-50 (ICD codes)**.

MIMIC-III (CCS codes): We leverage the ICD-CCS mapping scheme maintained by the HCUP[4] to project the ICD codes into lower-dimensional CCS codes. The top-50 ICD codes and full ICD codes (8,921) are converted into top-38 CCS codes and full CCS codes (295). The MIMIC-III datasets of CCS codes share the same discharge summaries with the MIMIC-III ICD code dataset for training, validation, and testing. We denote the full and top-38 CCS code datasets as **MIMIC-III-full (CCS codes)** and **MIMIC-III-38 (CCS codes)**, respectively.

4.2 Settings

Data Preprocessing: Following the previous works, we remove the non-alphabetic tokens, such as punctuation and numbers, from clinical documents. We transform all tokens into the lowercase format and change all tokens appearing in fewer than three notes into the UNK token. The medical word embeddings are established by the word2vec technique from all discharge summaries. The dimension of word embedding d_e is set as 100, consistent with previous works. We set the maximum length of each document as 4,000, with the exceeded part truncated.

Evaluation Metrics: We use the same evaluation metrics as previous works to validate the effectiveness of our proposed model on the datasets of two kinds of medical codes. The evaluation metrics include macro-averaged and micro-averaged AUC-ROC (area under the receiver operating characteristic curve), macro-averaged and micro-averaged F1, precision at k (termed P@k, where $k \in \{8, 15\}$). P@k is the precision score indicating the top-k scored predictions in the ground truth labels.

Hyper-parameter Tuning: Our implementation details are as follows. We train our model with the optimizer Adam [18] and set the learning rate to 0.001. The batch sizes of the top-n ($n \in \{38, 50\}$) and full-code data are 16 and 64, respectively. We apply the early stopping trick to exit the model training by monitoring the P@k score, which avoids model over-fitting. The training will stop

[3] https://mimic.physionet.org/gettingstarted/access/.
[4] www.hcup-us.ahrq.gov/toolssoftware/ccs/ccs.jsp.

if the P@k score on the validation set does not improve in 10 rounds. We set the kernel size of each block in the RAM to 3. The dropout rate of the RAM is 0.2. For multitask learning, we set ICD scaling factor λ_d and CCS scaling factors λ_s as 0.7 and 0.3, respectively. In the focal loss, the weighting factor α is 0.999, and the focusing parameter γ is 2, tuned from 1 to 5. Section 4.6 studies the detailed experimental settings about hyperparameters of multitask learning and focal loss.

4.3 Baselines

CNN [31]: The vanilla CNN model utilizes a max-pooling **C**onvolutional **N**eural **N**etwork [17] to predict ICD codes.

BiGRU [31]: This model uses a bidirectional recurrent architecture with gated recurrent units as the feature extractor for ICD coding.

CAML [31]: The **C**onvolutional **A**ttention network for **M**ulti-**L**abel classification (CAML) uses a convolutional neural network to extract the document features and the label-wise attention mechanism to enhance feature learning.

DR-CAML [31]: **D**escription **R**egularized-**CAML** (DR-CAML) is an extension model of the CAML, which incorporates textual descriptions of ICD codes to regularize the CAML model.

MultiResCNN [24]: **Mul**ti-Filter **Res**idual **C**onvolutional **N**eural **N**etwork (MultiResCNN) leverages a convolutional layer with multiple filters to capture various text patterns and adopts a residual block to increase the receptive field on the model.

LAAT [40]: Vu et al. designed the new **la**bel **at**tention model (LAAT) by choosing bidirectional Long-Short Term Memory (BiLSTM) as the feature extractor and deploying a label self-attention mechanism to learn label-specific vectors for ICD code predictions.

JointLAAT [40]: JointLAAT extends the LAAT by applying a hierarchical joint learning model to capture the hierarchical structure of ICD codes.

Fusion [28]: Fusion utilizes a compressed convolutional layer to encode clinical notes into informative local features, which are fused into output representations for ICD code predictions.

MDBERT [45]: Medical Document BERT (MDBERT) is a bottom-up hierarchical framework that combines features at the word level, sentence level, and document level to efficiently encode long documents. The hierarchical encoding model is first proposed by Yang et al.[43] and applied to ICD coding by Dong et al. [9].

4.4 Results

MIMIC-III-50 (ICD code): Table 1 shows the experimental results of baseline models and our proposed model on the MIMIC-III-50 (ICD code) dataset. We observe that the MARN outperforms the other models (CNN, BiGRU, CAML, DR-CAML, MultiResCNN, and MT-RAM) clearly across all evaluation metrics. The JointLAAT model uses a hierarchical joint learning mechanism to deal with the imbalanced class issue. Compared with JointLAAT, our proposed model (MARN) improves macro-AUC, micro-AUC, macro-F1, micro-F1, and P@5 scores by 0.2%, 0.1%, 2.1%, 0.2%, and 0.2%, respectively. The MARN makes significant improvements, especially on the macro-F1 score, by 7.6%, 10.6%, 19.8%, 7.6%, and 3.0% compared with MultiResCNN, DR-CAML, CAML, BiGRU, CNN, and MT-RAM. Fusion improved all scores about 0.1% $\sim$ 0.7% compared with the MARN.

In recent years, the pretrained language models with the transformer architecture, such as BERT [8], have dominated many NLP tasks [23]. However, applying the BERT to medical coding tasks suffers from the limited document sequence (512 tokens) [14] on the medical coding task. MDBERT, a BERT-based ICD coding framework, achieved competitive performance on the MIMIC-III-50 (ICD code) dataset, while the MARN outperformed the MDBERT by 0.9%, 1.1%, 2.3%, 2.6%, and 1.9% in macro-AUC, micro-AUC, macro-F1, micro-F1, and P@5 scores, respectively.

Table 1. MIMIC-III-50 (ICD code) Dataset Results (in %)

Models	AUC-ROC		F1		P@5
	Macro	Micro	Macro	Micro	
CNN	87.6	90.7	57.6	62.5	62.0
BiGRU	82.8	86.8	48.4	54.9	59.1
CAML	87.5	90.9	53.2	61.4	60.9
DR-CAML	88.4	91.6	57.6	63.3	61.8
MultiResCNN	89.9	92.8	60.6	67.0	64.1
MT-RAM	92.1	94.3	65.2	70.7	66.4
LAAT	92.5	94.6	66.6	71.5	67.5
JointLAAT	92.5	94.6	66.1	71.6	67.1
Fusion	**93.1**	**95.0**	**68.3**	**72.5**	**67.9**
MDBERT	91.8	93.6	65.9	69.2	65.4
MARN (ours)	92.7	94.7	68.2	71.8	67.3

Table 2. MIMIC-III-full (ICD Code) Dataset Results (in %)

Models	AUC-ROC		F1		P@k	
	Macro	Micro	Macro	Micro	8	15
CNN	80.6	96.9	4.2	41.9	58.1	44.3
BiGRU	82.2	97.1	3.8	41.7	58.5	44.5
CAML	89.5	98.6	8.8	53.9	70.9	56.1
DR-CAML	89.7	98.5	8.6	52.9	69.0	54.8
MultiResCNN	91.0	98.6	8.5	55.2	73.4	58.4
LAAT	91.9	98.8	9.9	57.5	73.8	59.1
JointLAAT	92.1	98.8	10.7	57.5	73.5	59.0
Fusion	91.5	98.7	8.3	55.4	73.6	-
MDBERT	**92.5**	**98.9**	10.1	55.5	72.7	57.7
MARN (ours)	91.3	98.8	**11.6**	**58.4**	**75.4**	**60.2**

MIMIC-III-full (ICD code): Table 2 shows the results of the MARN and other strong baseline models. The MARN performs better on macro-F1, micro-F1, P@8, and P@15 scores than other baseline models. When compared with the state-of-the-art model (JointLAAT), our proposed model has improved the scores of macro-F1, micro-F1, P@8, and P@15 by 0.9%, 0.9% 1.9%, and 1.2%, respectively. Compared with convolution-based models—including CNN, CAML, DR-CAML, MultiResCNN, and Fusion—the MARN v astly increases the micro-F1 score by 16.5%, 4.5%, 5.5%, 3.2%, and 3.0%, respectively. The MARN improved the macro-F1, micro-F1, p@8, and p@15 scores by 1.5%, 2.9%, 2.7%, and 2.5%, respectively, compared with the BERT-based MDBERT model.

MIMIC-III-50 (CCS code): We validate the BiGRU, CAML, DR-CAML, and the MultiResCNN on the MIMIC-III-50 (CCS code) dataset and show the evaluation results in Table 3. The MARN outperforms all baseline models by large margins across all evaluation metrics. Significantly, MARN improves the macro-F1 and micro-F1 scores by 8.3% and 6.2% compared with the MultiResCNN. Our model also outperforms the BiGRU, CAML, and DR-CAML on macro-F1 and micro-F1 scores with 10% $\sim$ 14% and 8% $\sim$ 13%, respectively.

MIMIC-III-full (CCS code): We also evaluate the same baseline models on the MIMIC-III full (CCS code) and compare our model to verify the effectiveness of the MARN. Table 4 shows that our model improves the macro-F1 score by 7.8% compared with the MultiResCNN model. Our

Table 3. MIMIC-III-50 Results (CCS Code) Dataset Results (in %)

Models	AUC-ROC		F1		P@5
	Macro	Micro	Macro	Micro	
BiGRU	87.6	90.7	57.6	62.5	62.0
CAML	89.2	92.2	60.9	67.5	64.5
DR-CAML	87.5	90.5	59.3	65.6	62.6
MultiResCNN	89.2	92.4	62.9	68.8	64.6
MARN (ours)	**92.8**	**95.0**	**71.2**	**75.0**	**69.0**

Table 4. MIMIC-III-full (CCS Code) Dataset Results (in %)

Models	AUC-ROC		F1		P@k	
	Macro	Micro	Macro	Micro	8	15
BiGRU	91.2	96.4	50.1	68.4	81.1	64.0
CAML	88.8	96.1	44.4	66.5	80.5	63.6
DR-CAML	85.7	95.5	41.3	66.0	78.9	62.5
MultiResCNN	90.6	96.5	50.8	69.0	81.8	64.8
MARN(ours)	**93.9**	**97.4**	**58.6**	**72.2**	**84.3**	**67.2**

Table 5. Ablation Results (in %) of MIMIC-III-50 (ICD Code)

Models	AUC-ROC		F1		P@5
	Macro	Micro	Macro	Micro	
MARN	**92.7**	**94.7**	**68.2**	**71.8**	**67.3**
w/o MTL	91.9	94.0	64.4	69.4	66.1
w/o MTL + FL	91.7	93.4	62.4	68.1	64.7
w/o RAM	92.3	94.3	64.8	69.9	66.5
w/o RAM + FL	91.8	94.1	64.6	69.9	66.2
w/o MTL + RAM + FL	91.2	93.4	59.2	67.2	65.5

model also promotes other evaluation scores, with macro-AUC, micro-AUC, micro-F1, P@8, and P@15 increased by 3.1%, 0.9%, 3.2%, 2.5%, and 2.4%, respectively.

4.5 Detailed Analysis of MARN

This section studies the properties of the proposed MARN model through several research questions.

How does each component of MARN affect the prediction? We conduct experiments to validate the effectiveness of each component of the MARN on the MIMIC-III-50 (ICD code) and MIMIC-III-full (ICD code) datasets, with the following specific building components considered:

- **M**ulti-**t**ask **L**earning scheme (MTL)
- **R**ecalibrated **A**ggregation **M**odule (RAM)
- **F**ocal **L**oss (FL)

From Tables 5 and 6, we can observe that all components contribute to performance improvement and are complementary to each other. The MTL scheme has a higher performance gain on the MIMIC-III-50 (ICD code) dataset than the RAM, while the MIMIC-III-full (ICD code) dataset shows the opposite situation. The model optimized with focal loss outperforms the one with BCE loss.

Table 6. Ablation Results (in %) of MIMIC-III-full (ICD Code)

Models	AUC-ROC		F1		P@k	
	Macro	Micro	Macro	Micro	8	15
MARN	**91.3**	**98.8**	**11.6**	**58.4**	**75.4**	**60.2**
w/o MTL	89.9	98.6	10.5	57.1	73.5	58.5
w/o MTL + FL	89.1	98.4	9.1	55.6	72.9	58.0
w/o RAM	90.4	98.6	10.3	56.0	72.6	57.9
w/o RAM + FL	88.8	98.2	7.4	50.7	69.5	54.8
w/o MTL + RAM + FL	88.9	98.3	6.8	51.5	69.9	54.7

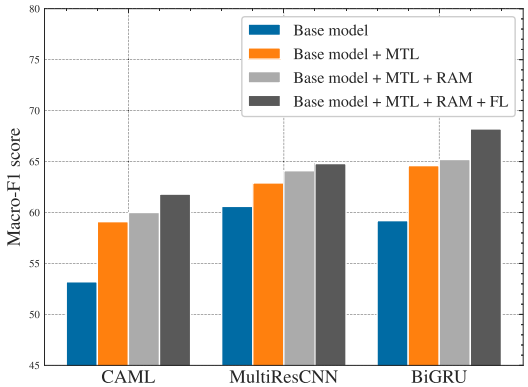


Fig. 8. Macro-F1 scores (in %) of different base models, including the different building blocks of MARN.

How Compatible are the building blocks with different base models? We choose two convolution-based models (i.e., CAML, MultiResCNN) and an RNN model (BiGRU) to explore the compatibility of different building modules. We call them *base models*. Figure 8 shows that MTL, RAM, and FL can improve other base models' performance. Significantly, the BiGRU model optimized with FL yields better performance than the CAML and MultiResCNN.

Can multitask learning connect different medical coding systems? We leverage Principal Component Analysis (PCA) to reduce the dimension of document features and plot the embeddings of ICD codes and CCS codes in the resulting two-dimensional space. Figure 9 demonstrates that the cloud of ICD codes surrounds the cluster of CCS codes, reflecting the mapping from ICD to CCS codes. We also notice that ICD codes associated with a particular CCS code are often clustered around the respective CCS code, as shown for several representative CCS codes and their corresponding ICD codes in Figure 10. To study this phenomenon in detail, we define a circular region around each CCS code using 10% of the longest distance between two ICD codes as the radius. We then calculate the number of significant CCS codes, defined as CCS codes, such that their respective regions contain significantly more *relevant* ICD codes than expected by chance, where relevant ICD codes are those that are known to map to the CCS code. We approximate the distribution of the number of relevant ICD codes within a region using binomial distribution:

$$P(k) = Binomial(k|n, p), \tag{18}$$

where n is the total number of ICD codes in the region and p is the overall proportion of relevant ICD codes for the respective CCS code. For each CCS code, we define a threshold T (the number of relevant ICD codes in the CCS code region), such that $P(k > T) < 0.1$. If the observed number of

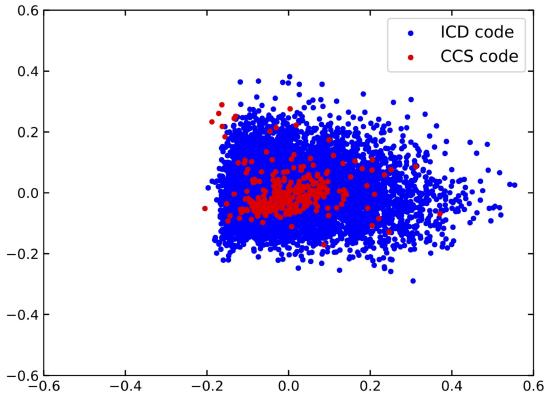
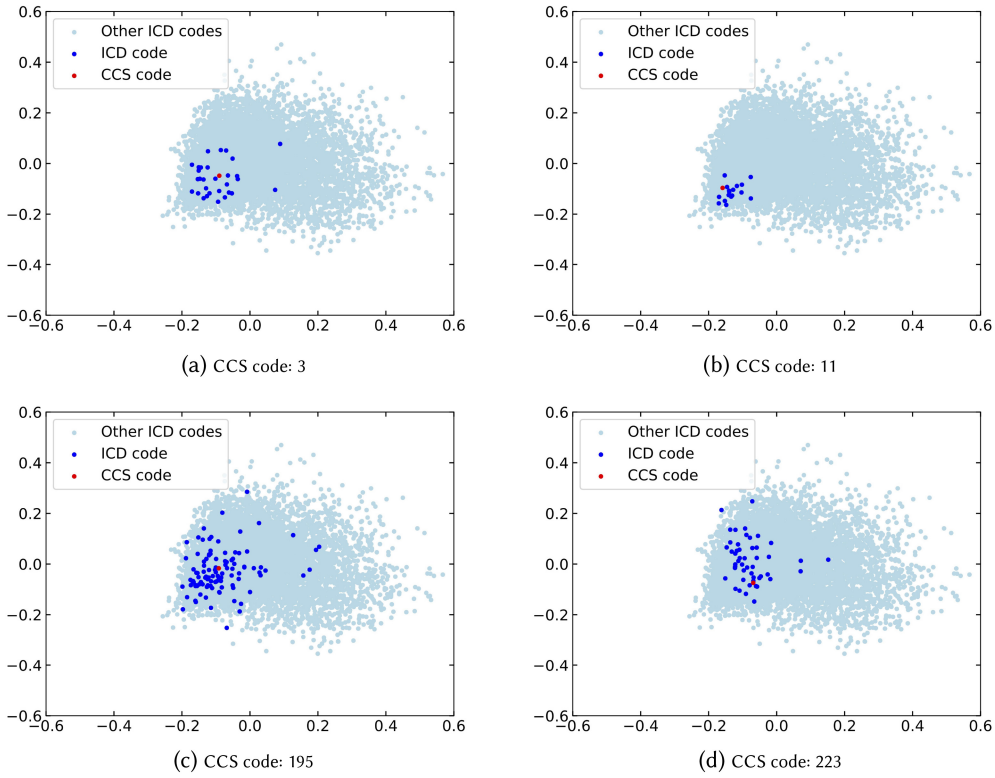


Fig. 9. The first two principal components of ICD and CCS code embeddings.



(a) CCS code: 3 (b) CCS code: 11

(c) CCS code: 195 (d) CCS code: 223

Fig. 10. The embeddings of representative significant CCS codes and their corresponding ICD codes. We see that the relevant ICD codes are clustered around the respective significant CCS code, reflecting the ability of MARN to learn representations that capture informative relationships between the codes.

relevant codes within a region exceeds the threshold, we consider the CCS code to be significant. As a result, there are 157 significant CCS codes among the total of 295 CCS codes. Hence, we conclude that MTL can establish informative connections between related ICD codes to benefit medical code prediction.

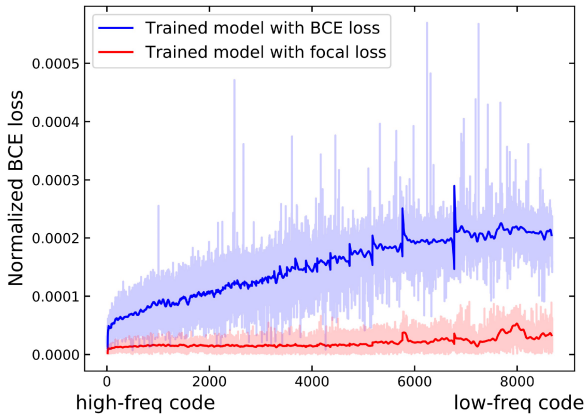


Fig. 11. Normalized binary cross entropy loss of each ICD code, with x-axis sorted by code frequency. The high-frequency codes are on the left; the low-frequency codes are on the right.

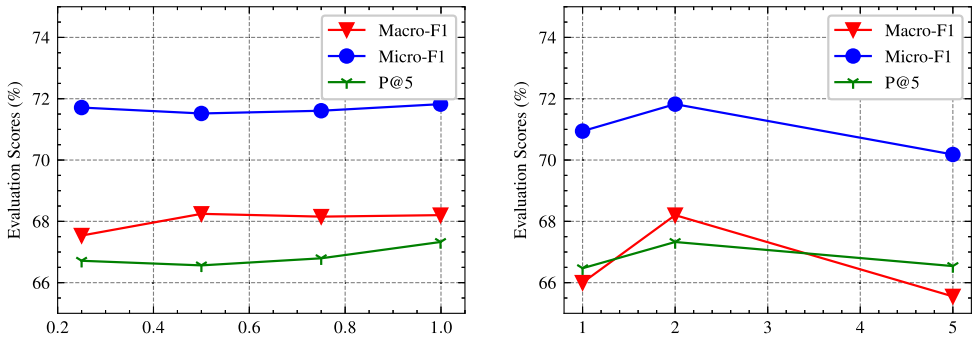
Does the model optimized with focal loss balance the learning between low- and high-frequency codes? We plot the normalized loss value of each code to explore whether the focal loss can balance the loss of high- and low-frequency ICD codes. First, we train the model with different loss functions, take a forward pass of the trained model, and calculate a unified loss function, that is, binary cross entropy (BCE) loss, for a fair comparison. Then, we normalize the loss of each ICD code by dividing the BCE loss by the frequency of the code and the total number of documents. Figure 11 shows the normalized loss curve of models trained by the BCE loss and focal loss. We see that optimizing the models with focal loss improves results overall especially for low-frequency codes. The normalized loss of the model optimized by focal loss is balanced compared with the model optimized by BCE loss. Thus, we can conclude that the model optimized with BCE loss cannot balance the learning of high- and low-frequency codes. In contrast, the model optimized with focal loss can effectively handle the imbalanced class problem in this study.

4.6 Hyperparameter Studies

We study two hyperparameters of the MTL scheme and focal loss in this section.

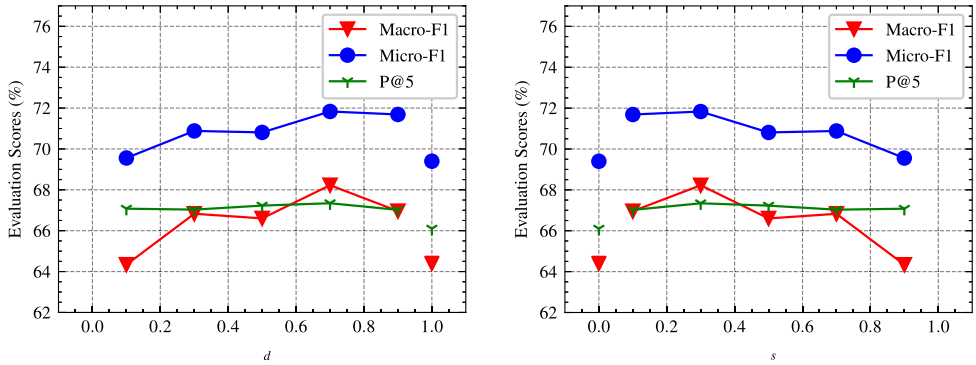
Figure 12(a) shows the predictive performance of the MARN on the MIMIC-III dataset by applying different α. The variations of the MARN's performance is small, which means the adjustment of the hyperparameter α does not largely influence the evaluation results. The oscillations of three scores are slight. We find that the optimal α is 0.999. Figure 12(b) shows that γ largely affects the results of our proposed model, and the optimal γ is 2. Based on these two figures, we found that the γ contributes more to balancing tlearning of the high- and low-frequency codes and the performance increment of the parameter *alpha* is slight.

We evaluate our proposed MARN on the MIMIC-III dataset by setting different loss weights of medical coding branches: λ_d for the ICD coding and λ_s for the CCS coding. Results are shown in Figures 13(a) and 13(b), respectively. Two figures are symmetrical because the summation of these two coefficients is 1. We take a common loss weighting strategy for our MTL scheme, which is to assign the same weights for each coding task [25], and we fine-tune these loss weights to get the best performance. Intuitively, the CCS coding branch as an auxiliary task should be assigned with smaller loss weight to enable the association knowledge transferred from the CCS coding branch to the ICD branch. The variations of three scores are small in Figure 13(a) when λ_d is from



(a) Evaluation results with $\gamma = 2$. (b) Evaluation results with $\lambda = 0.999$.

Fig. 12. Predictive performance on MIMIC-III dataset with different γ and λ.



(a) Evaluation results with different loss weight of ICD codes (λ_d). (b) Evaluation results with different loss weight of CCS codes (λ_s).

Fig. 13. Predictive performance on MIMIC-III dataset with different λ_d and λ_s ($\lambda_d + \lambda_s = 1$).

0.5 to 0.9. However, if we further increase the λ_d to 1 (i.e., removing the MTL scheme from the MARN), the Macro-F1, Micro-F1, and P@5 dramatically drop to 64.4, 69.4, and 66.1. In summary, the evaluation results are not sensitive to the value of the λ_d and λ_s and the model's performance largely decreases if the MARN does not include the MTL scheme.

5 FUTURE WORK

In recent years, the transformer-based language model has become a new paradigm for NLP tasks. With the support of self-attention mechanisms, the transformer-based models can capture token-dependent patterns for boosting contextualized text learning. The transformer and its variants can suffer from the quadratic memory and time complexity problem caused by the self-attention mechanism. Although efficient transformer-based models have been proposed—such as Reformer [19], Linformer [41], and Longformer [1]—they also need a substantial amount of computational resources for neural network training. By contrast, the GRU-based model is positioned nicely on the Pareto frontier of the computation-performance curve. In the next stage of our research, we plan to study how to effectively utilize contextual embeddings to obtain semantically enriched document features for medical code prediction.

6 CONCLUSION

This paper proposes a novel model, **M**ultitask b**A**alanced and **R**ecalibrated **N**etwork (MARN), to tackle three challenges of automated medical coding: the imbalanced class problem, capturing code association, and dealing with lengthy and noisy documents. We leverage focal loss to alleviate the imbalanced class issue by redistributing the loss weights between low- and high-frequency medical codes. We design the **R**ecalibrated **A**ttention **M**odule (RAM) to inject high-level semantic features into the original feature for noise suppressing. The cascaded convolutional structure of the RAM can improve representation learning from long and noisy documents. The multitask learning scheme that enables code relationship knowledge transfer between two different coding systems (ICD and CCS) is developed to capture code association and improve coding performance. The experimental results show that our proposed model outperforms competitive baseline models in the real-world MIMIC-III database.

REFERENCES

[1] Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

[2] Jinbo Bi, Tao Xiong, Shipeng Yu, Murat Dundar, and R. Bharat Rao. 2008. An improved multi-task learning approach with applications in medical diagnosis. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, Berlin, 117–132.

[3] Rohitash Chandra, Abhishek Gupta, Yew-Soon Ong, and Chi-Keong Goh. 2016. Evolutionary multi-task learning for modular training of feedforward neural networks. In *International Conference on Neural Information Processing*. Springer, Cham, 37–46.

[4] Francisco Charte, Antonio J. Rivera, María J. del Jesus, and Francisco Herrera. 2015. Addressing imbalance in multilabel classification: Measures and random resampling algorithms. *Neurocomputing* 163 (2015), 3–16.

[5] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[6] Nitesh V. Chawla, Nathalie Japkowicz, and Aleksander Kotcz. 2004. Special issue on learning from imbalanced data sets. *ACM Special Interest Group on Knowledge Discovery and Data Mining Explorations Newsletter* 6, 1 (2004), 1–6.

[7] Shanta Chowdhury, Xishuang Dong, Lijun Qian, Xiangfang Li, Yi Guan, Jinfeng Yang, and Qiubin Yu. 2018. A multitask bi-directional RNN model for named entity recognition on Chinese electronic medical records. *BMC Bioinformatics* 19, 17 (2018), 75–84.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018).

[9] 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 (2021), 103728.

[10] Steven Horng, David A. Sontag, Yoni Halpern, Yacine Jernite, Nathan I. Shapiro, and Larry A. Nathanson. 2017. Creating an automated trigger for sepsis clinical decision support at emergency department triage using machine learning. *PloS One* 12, 4 (2017), e0174708.

[11] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Salt Lake City, Utah, 7132–7141.

[12] Yannet Interian, Lara Reichmann, and Gilmer Valdes. 2020. Multitask learning from clinical text and acute physiological conditions differentially improve the prediction of mortality and diagnosis at the ICU. *medRxiv*.

[13] Shaoxiong Ji, Erik Cambria, and Pekka Marttinen. 2020. Dilated convolutional attention network for medical code assignment from clinical text. arXiv preprint arXiv:2009.14578 (2020).

[14] 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 (2021), 104998.

[15] Shaoxiong Ji, Shirui Pan, and Pekka Marttinen. 2021. Medical code assignment with gated convolution and note-code interaction. arXiv preprint arXiv:2010.06975 (2020).

[16] Alistair E. W. 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 (2016), 1–9.

[17] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[18] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
[19] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. *arXiv preprint arXiv:2001.04451*.
[20] Bevan Koopman, Guido Zuccon, Anthony Nguyen, Anton Bergheim, and Narelle Grayson. 2015. Automatic ICD-10 classification of cancers from free-text death certificates. *International Journal of Medical Informatics* 84, 11 (2015), 956–965.
[21] S. B. Kotsiantis and P. E. Pintelas. 2003. Mixture of expert agents for handling imbalanced data sets. *Annals of Mathematics, Computing & Teleinformatics* 1, 1 (2003), 46–55.
[22] Leah S. Larkey and W. Bruce Croft. 1996. Combining classifiers in text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. Association for Computing Machinery, Zurich, 289–297.
[23] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2021. FNet: Mixing tokens with Fourier transforms. *arXiv preprint arXiv:2105.03824*.
[24] Fei Li and Hong Yu. 2020. ICD coding from clinical text using multi-filter residual convolutional neural network. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 5 (2020), 8180–8187.
[25] Baijiong Lin, Feiyang Ye, and Yu Zhang. 2021. A closer look at loss weighting in multi-task learning. *arXiv preprint arXiv:2111.10603*.
[26] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*. IEEE, Hawaii, 2980–2988.
[27] Xiaodong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding. *arXiv preprint arXiv:1901.11504*.
[28] Junyu Luo, Cao Xiao, Lucas Glass, Jimeng Sun, and Fenglong Ma. 2021. Fusion: Towards automated ICD coding via feature compression. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. ACL, Thailand, 2096–2101.
[29] Andrew Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. ACL, Portland, 142–150.
[30] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*.
[31] James Mullenbach, Sarah Wiegreffe, Jon Duke, Jimeng Sun, and Jacob Eisenstein. 2018. Explainable prediction of medical codes from clinical text. arXiv preprint arXiv:1802.05695 (2018).
[32] Kimberly J. O'Malley, Karon F. Cook, Matt D. Price, Kimberly Raiford Wildes, John F. Hurdle, and Carol M. Ashton. 2005. Measuring diagnoses: ICD code accuracy. *Health Services Research* 40, 5p2 (2005), 1620–1639.
[33] Jong-Ku Park, Ki-Soon Kim, Tae-Yong Lee, Kang-Sook Lee, Duk-Hee Lee, Sun-Hee Lee, Sun-Ha Jee, Il Suh, Kwang-Wook Koh, So-Yeon Ryu, et al. 2000. The accuracy of ICD codes for cerebrovascular diseases in medical insurance claims. *Journal of Preventive Medicine and Public Health* 33, 1 (2000), 76–82.
[34] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. 2017. Large kernel matters–improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, Hawaii, 4353–4361.
[35] Adler Perotte, Rimma Pivovarov, Karthik Natarajan, Nicole Weiskopf, Frank Wood, and Noémie Elhadad. 2014. Diagnosis code assignment: Models and evaluation metrics. *Journal of the American Medical Informatics Association* 21, 2 (2014), 231–237.
[36] Yuqi Si and Kirk Roberts. 2019. Deep patient representation of clinical notes via multi-task learning for mortality prediction. *AMIA Summits on Translational Science Proceedings* 2019 (2019), 779.
[37] Heung-Il Suk, Seong-Whan Lee, and Dinggang Shen. 2016. Deep sparse multi-task learning for feature selection in Alzheimer's disease diagnosis. *Brain Structure and Function* 221, 5 (2016), 2569–2587.
[38] Wei Sun, Shaoxiong Ji, Erik Cambria, and Pekka Marttinen. 2021. Multitask recalibrated aggregation network for medical code prediction. In *Proceedings of the Joint European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. Springer, Cham.
[39] Muhammad Atif Tahir, Josef Kittler, and Ahmed Bouridane. 2012. Multilabel classification using heterogeneous ensemble of multi-label classifiers. *Pattern Recognition Letters* 33, 5 (2012), 513–523.
[40] Thanh Vu, Dat Quoc Nguyen, and Anthony Nguyen. 2021. A label attention model for ICD coding from clinical text. arXiv preprint arXiv:2007.06351 (2020).
[41] Sinong Wang, Belinda Z. Li, Madian Khabsa, Han Fang, and Hao Ma. 2020. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*.
[42] 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*. ACM, New York, NY, 649–658.

[43] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. ACL, San Diego, 1480–1489.

[44] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*.

[45] Ning Zhang and Maciej Jankowski. 2022. Hierarchical BERT for medical document understanding. *arXiv preprint arXiv:2204.09600*.

[46] Yu Zhang and Qiang Yang. 2017. A survey on multi-task learning. *arXiv preprint arXiv:1707.08114* (2017), 1–1.

[47] Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. 2019. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. AAAI Press, Honolulu, Hawaii, 817–824.

Received 10 April 2022; revised 12 August 2022; accepted 30 August 2022