Attention and Memory-Augmented Networks for Dual-View Sequential Learning

Yong He, Cheng Wang, Nan Li, Zhenyu Zeng
{sanyuan.hy,youmiao.wc,kido.ln,zhenyu.zzy}@alibaba-inc.com
Alibaba Group
Hangzhou, China

ABSTRACT

In recent years, sequential learning has been of great interest due to the advance of deep learning with applications in time-series forecasting, natural language processing, and speech recognition. Recurrent neural networks (RNNs) have achieved superior performance in single-view and synchronous multi-view sequential learning comparing to traditional machine learning models. However, the method remains less explored in asynchronous multi-view sequential learning, and the unalignment nature of multiple sequences poses a great challenge to learn the inter-view interactions. We develop an AMANet (Attention and Memory-Augmented Networks) architecture by integrating both attention and memory to solve asynchronous multi-view learning problem in general, and we focus on experiments in dual-view sequences in this paper. Self-attention and inter-attention are employed to capture intra-view interaction and inter-view interaction, respectively. History attention memory is designed to store the historical information of a specific object, which serves as local knowledge storage. Dynamic external memory is used to store global knowledge for each view. We evaluate our model in three tasks: medication recommendation from a patient's medical records, diagnosis-related group (DRG) classification from a hospital record, and invoice fraud detection through a company's taxation behaviors. The results demonstrate that our model outperforms all baselines and other state-of-the-art models in all tasks. Moreover, the ablation study of our model indicates that the inter-attention mechanism plays a key role in the model and it can boost the predictive power by effectively capturing the inter-view interactions from asynchronous views.

CCS CONCEPTS

• **Computing methodologies** → **Neural networks**; **Supervised learning by classification**; • **Information systems** → **Clustering and classification**; • **Applied computing** → *Health care information systems.*

KEYWORDS

dual-view sequential learning, inter-attention, intra-attention, dynamic external memory, history attention memory, classification, medication recommendation, DRG classification, invoice fraud detection

ACM Reference Format:
Yong He, Cheng Wang, Nan Li, Zhenyu Zeng. 2020. Attention and Memory-Augmented Networks for Dual-View Sequential Learning. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining USB Stick (KDD '20), August 23–27, 2020, Virtual Event, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394486.3403055

KDD '20, August 23–27, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403055

1 INTRODUCTION

Multi-view data that different views describe distinct perspectives is very common in many real-world applications. Various learning algorithms have been proposed to incorporate the complementary information of the multi-view datasets in order to understand the complex system and make precise data-driven prediction [26]. Due to the advance of the recurrent neural network (RNN) architecture development, multiple sequential events could be integrated into multi-view learning. Such a form of learning is known as multi-view sequential learning. Many models are designed for synchronous sequences that all views share the same time step or could be aligned, while there are few models focused on asynchronous sequences that sequences are in various granularity and with dynamic lengths. Electronic health record (EHR) learning is a notable example of multi-view sequential learning. A patient's medical records consist of multiple views, such as diagnosis, procedure and medication, and are asynchronous in a fine-grained time scale.

The key challenge of multi-view sequential learning is to capture both intra-view interaction within a single view and inter-view interactions across multiple views. RNNs, such as long short-term memory (LSTM) [7] and gated recurrent unit (GRU) [3], have been widely used to learn the intra-view interactions in single view sequential learning problems. However, the sequential nature of RNNs makes parallelization challenging. The transformer, which dispenses recurrence and uses attention mechanism to model the dependencies in sequence, can achieve the state-of-the-art performance in sequential modeling and allow for parallelization [18]. The techniques to learn the inter-view interactions can be classified into early, late, and hybrid fusion approaches [2]. In the early fusion method, features of multiple views are concatenated into a single feature before learning. The late fusion method first learns separated models from individual views and then combines together through averaging or voting to make the final prediction. A hybrid approach attempts to take advantage of both above methods. However, the prerequisite of early or hybrid approach is the alignment of the sequences which cannot be fulfilled in an asynchronous setting.

To address the above challenges in asynchronous multi-view learning, we develop AMANet (Attention and Memory-Augmented

Networks) based on attention and memory mechanisms. Our model consists of three major components for each view: neural controller, history attention memory, and dynamic external memory. The neural controller performs encoding of the input sequence to capture the intra-view interaction relying on the self-attention mechanism. The inter-attention mechanism is introduced to learn the inter-view interaction. The self-attention or intra-attention mechanism could associate different positions within a single sequence. Similarly, the inter-attention relates positions across two sequences. Then the vectors from both self-attention and inter-attention are concatenated together as the encoding vector of the sequence. The history attention memory stores all previous encoding vectors of the same object. The dynamic external memory is shared by all objects in training and stores the common knowledge from the data. Lastly, the encoding vector from the controller, the read vector from dynamic external memory, and the historical attention vector from history attention memory are concatenated together for the prediction task.

We have conducted experiments on three tasks: medication recommendation from a patient's medical records including previous and current diagnoses and medical procedures, DRG classification from a current hospital record with diagnosis and procedure sequences, and invoice fraud detection from a company's monthly tax declaration and invoice sequences. In all tasks, our model is able to outperform baselines and other state-of-the-art models. The rest of the paper is organized as follows. We review related works and techniques in section 2 and introduce the formulation and architecture of our model in section 3. We evaluate our model in three tasks introduced above in section 4. In addition, we also performance ablation studies to investigate the importance of each module in our model. In section 5, we conclude our work and highlight future directions.

2 RELATED WORKS

Multi-view learning is a rapidly growing area of machine learning and recent trends have been reviewed thoroughly [2, 21, 26]. Several deep learning architectures have been developed to expand multi-view learning into sequential datasets such as multi-view LSTM [15], memory fusion network (MFN) [24], and dual memory neural computer [12]. Here we emphasize the works with attention mechanism and memory augmentation method.

The attention mechanism was initially introduced in machine translation to jointly translate and align words [1]. It has become an integral part with RNNs in sequential modeling to abstract the most relevant information of the sequence. Attention mechanism offers the interpretability of deep neural networks, which is one of the most intriguing features of attention. In healthcare studies, attention mechanism is adopted in many neural network architectures, such as Dipole [14], REATIN [4], and RAIM [22], to achieve better predictive power and more effective result interpretation. For example, RAIM integrates continuous monitoring data and discrete clinical event using guided multi-channel attention for two prediction tasks. It has shown excellent performances in prediction and meaningful interpretation in quantitative analysis.

Recently, the transformer solely based on self-attention mechanism has drawn great attention due to the computation efficiency without sacrificing the performance comparing to RNNs [18]. For instances, SAnD (Simple Attend and Diagnose) model adapts the self-attention mechanism as in transformer for multiple diagnosis tasks [17]. In our work, the self-attention layer from the transformer is used for intra-view learning. Moreover, we propose an inter-attention layer by modifying the self-attention layer to capture inter-view interactions. The attention mechanism is also used to retrieve the historical attention vector from history attention memory.

Memory-augmented neural networks (MANN), such as memory networks [20] and differentiable neural computers (DNC) [6], use external memory as dynamic knowledgebase to store the long-term dependencies. The external memory can be read and written to with attentional processes and has been widely used. Graph augmented memory networks (GAMENet) [16] integrates MANN with graph convolution networks (GCN) [11] to embed multiple knowledge graphs. DMNC [12] is a first attempt to build MANN for asynchronous dual-view sequential learning. GAMENet and DMNC have been employed for medication recommendation task and are served as baselines in this work. In our work, the dynamic external memory from DNC is used to store the common knowledge of each view.

3 METHOD

3.1 Problem Formulation

In this work, we use dual-view sequences to introduce our model. The goal of dual-view sequential learning is to predict the target y given two input sequences X^1 and X^2. Let $S_i = (X_i^1, X_i^2, y_i)$ to be the sample with index i, where $X_i^1 = \{x_{i1}^1, x_{i2}^1, ..., x_{iL_i^1}^1\}$, $X_i^2 = \{x_{i1}^2, x_{i2}^2, ..., x_{iL_i^2}^2\}$, and y_i could be a scalar or a vector depending on the prediction task. L_i^1 and L_i^2 are the length of X_i^1 and X_i^2, respectively. The length of the input sequence is not only view type dependent but also sample dependent. The x_{ik}^1 and x_{ik}^2 in the sequences are the initial representation of the input.

In the medication recommendation task, we predict the medication set given previous and current diagnoses and procedures. The current diagnoses and procedures of a patient can be formulated as two sequences for dual-view sequential learning. In MIMIC-III dataset [9], the diagnoses, medical procedures, and medications are represented as standard medical codes with code space size d_{x1}, d_{x2} and d_y, respectively. Therefore, we use one-hot vectors $x_{ik}^1 \in \{0,1\}^{d_{x1}}$, $x_{ik}^2 \in \{0,1\}^{d_{x2}}$, and $y_i \in \{0,1\}^{d_y}$ to represent diagnoses, procedures, and medications, respectively. The previous medical records are incorporated with history attention memory module. This is a multi-label classification task with d_y binary labels.

In the DRG classification task, we predict the DRG according to diagnosis sequence and procedure sequence from a patient's medical record of current visit. This is a multi-class classification task with d_y categories.

In the invoice fraud detection task, we identify the company involving in invoice fraud through its historical invoice and taxation declaration behavior. The daily invoice and monthly taxation declaration values are formulated as two asynchronous sequences with different granularity. This is a binary classification problem.

3.2 Network Architect

Our model is named Attention and Memory-Augmented Networks (AMANet)[1], and Figure 1 shows the AMANet in the dual-view setting. In this section, we describe the modules of our model in details.

Token Embedding. Given the one-hot vector representation of each item in the sequence, two input sequences X_i^1 and X_i^2 are formulated as $M_i^1 \in \{0,1\}^{L_i^1 \times d_{x1}}$ and $M_i^2 \in \{0,1\}^{L_i^2 \times d_{x2}}$, respectively. We derive the embedding of the input matrix as

$$TE_i^1 = M_i^1 W_E^1 \tag{1}$$

$$TE_i^2 = M_i^2 W_E^2 \tag{2}$$

where $W_E^1 \in \mathbb{R}^{d_{x1} \times d}$ and $W_E^2 \in \mathbb{R}^{d_{x2} \times d}$ are the embedding matrices and d is the dimension of embedding.

Positional Encoding. Since the neural controller use solely self-attention instead of RNNs, there is no recurrent and the positional information is lost. In order to incorporate the order of the sequence, positional encoding is introduced to integrate with the input embeddings [18]. We use the same positional encoding as in the transformer [18] as

$$PE(pos, 2j) = \sin(\frac{pos}{1000^{\frac{2j}{d}}}) \tag{3}$$

$$PE(pos, 2j+1) = \cos(\frac{pos}{1000^{\frac{2j}{d}}}) \tag{4}$$

where pos is the position index in the sequence, and $2j$ and $2j+1$ are the even and odd indexes of the embedding vector, respectively.

The sequence embedding vector is calculated by summarizing token embedding and positional encoding together as

$$E_i^1 = TE_i^1 + PE_i^1 \in \mathbb{R}^{L_i^1 \times d} \tag{5}$$

$$E_i^2 = TE_i^2 + PE_i^2 \in \mathbb{R}^{L_i^2 \times d} \tag{6}$$

Multi-Head Self-Attention. The neural controller will perform encoding on the embedding vector with the self-attention mechanism. In general, an attention function is to map a query and a set of key-value pairs to an output, where all of them are vectors. The output is computed as the weighted sum of values, where the weight for each value is the inner product of query and keys with a softmax normalization. In self-attention, queries Q, keys K, and values V are the linear projections from the same input embedding matrix [18]. For each view, the self-attention function is defined as

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{7}$$

where d_k is the dimension of Q and K. The inner product of Q and K is scaled by $\frac{1}{\sqrt{d_k}}$ as in transformer to avoid small gradients due to the large values in the dot product.

Here we use multi-head attention to encode the input sequence. The input embedding vector is projected linearly into h subspaces with a set of matrices $\{Q_m, K_m, V_m\}$ for each subspace, where queries $Q_m = EW_m^Q$, $W_m^Q \in \mathbb{R}^{d \times d_k}$, keys $K_m = EW_m^K$, $W_m^K \in \mathbb{R}^{d \times d_k}$, and values $V_m = EW_m^V$, $W_m^V \in \mathbb{R}^{d \times d_v}$. Then each attention head is computed by apply attention function on the projected matrices in each subspace. Finally, all attention heads are concatenated together and projected to be the intra-encoding vectors as

$$\begin{aligned} E_{intra} &= \text{MultiHead}(E) \\ &= \text{Concat}(head_1, ..., head_h) W_{intra} \end{aligned} \tag{8}$$

where $head_m = \text{Attention}(Q_m, K_m, V_m)$, and $W_{intra} \in \mathbb{R}^{hd_v \times d_{intra}}$, and m is the index of the heads. We set $d_k = d_v = d_{intra}/h$. Given two input sequence embeddings E^1 and E^2, the intra-encoding vectors are calculated as $E_{intra}^1 = \text{MultiHead}(E^1)$ and $E_{intra}^2 = \text{MultiHead}(E^2)$. The intra-encoding vectors will be used for dynamic external memory.

The self-attention module is also known as intra-attention. Its purpose is to obtain the relationship between different elements in the same sequence.

Dynamic External Memory. The intra-encoding vector from the neural controller is mapped to the interface vector ξ [6] followed by global average pooling operator as

$$\xi = \text{pool}(E_{intra} W_\xi) \tag{9}$$

where W_ξ is a trainable matrix and pool is the global average pooling operator. The interface vector ξ is subdivided into several sections for reading from and writing to memory matrix M. We used the same approach as in DNC [6] and more details can be found in the original work.

Each token in the input sequence will read and write memory once in the original paper. But the difference is that we use sample granularity to read and write memory in our work. We think that this method is more in line with the update frequency of global knowledge and faster.

Each view has its own memory matrix as in Figure 1. Two read vectors r^1 and r^2 are retrieved from two memory matrices M^1 and M^2, respectively, and both read vectors are used for the classification layer.

We use dynamic external memory module to store the global knowledge extracted from the dataset, and all samples can read from and write to it. We think that domain knowledge exists in the dataset. Therefore, we adapt this module for domain knowledge learining.

Inter-Attention. In the self-attention, the query, key and value are projected from the same input embedding to capture the intra-view interaction. In the inter-attention, the query is projected from one input embedding, while key and value are projected from the other input embedding. For instances, given two input sequence embeddings E^1 and E^2, the inter-encoding vectors for E^1 from E^2 are calculated as

$$\begin{aligned} E_{inter}^1 &= \text{MultiHead}(E^1, E^2) \\ &= \text{Concat}(head_1, ..., head_h) W_{inter} \end{aligned} \tag{10}$$

where $head_m = \text{Attention}(Q_m^1, K_m^2, V_m^2)$, $Q_m^1 = E^1 W_m^{Q^1}$, $K_m^2 = E^2 W_m^{K^2}$, and $V_m^2 = E^2 W_m^{V^2}$. The inter-encoding vectors for E^2 from E^1 can be calculated as $E_{inter}^2 = \text{MultiHead}(E^2, E^1)$.

The main purpose of this module is to capture the relationship between elements from different sequences, thereby enhancing the representation ability.

[1]The pseudo-code is in appendix, and the code is in https://github.com/non-name-2020/AMANet.

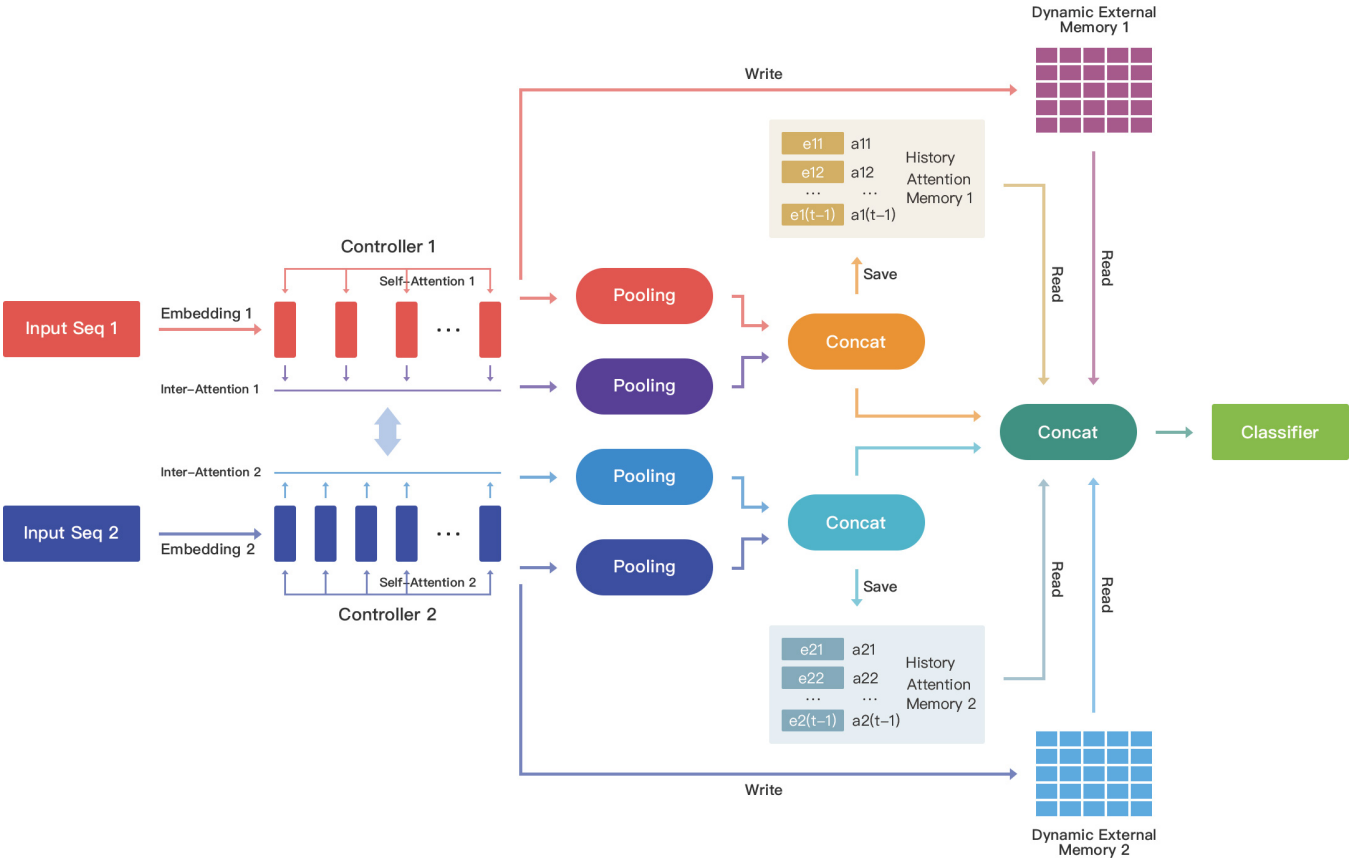


Figure 1: Attention and Memory-Augmented Networks for Dual Sequences. In history attention memory module, t denotes the time index of an object's sample at time t. If an object has only one sample, there is no history attention memory.

Encoding Vector. The final encoding vectors of a view in dual-view sequential data are calculated by applying global average pooling operator the intra-encoding and inter-encoding vectors, respectively, then concatenated them together as

$$e^1 = \text{Concat}(\text{pool}(E^1_{intra}), \text{pool}(E^1_{inter})) \tag{11}$$

$$e^2 = \text{Concat}(\text{pool}(E^2_{intra}), \text{pool}(E^2_{inter})) \tag{12}$$

where the pool is the global average pooling operator. The final encoding vectors e^1 and e^2 flow into two components: 1) directly used for classification layer; 2) saved to history attention memory.

History Attention Memory. The history attention memory is designed to capture an object's historical contribution to current time step t. It stores the historical representations, i.e., the encoding vectors of an object. Given the history attention memory $H_k = \{e_{k1}, e_{k2}, ..., e_{k(t-1)}\}$ for k-th object with all previous time steps in a single view, the attended historical representation h_k is calculated as the weighted sum of historical embedding vectors as

$$h_k = \sum_{i=1}^{t-1} \alpha_{ki} e_{ki} \tag{13}$$

The weights $\alpha_k = \{\alpha_{k1}, \alpha_{k2}, ..., \alpha_{k(t-1)}\}$ is obtained by

$$\alpha_{ki} = \text{softmax}(\mu_{ki}) = \frac{e^{\mu_{ki}^T w_c}}{\sum_j e^{\mu_{kj}^T w_c}} \tag{14}$$

where $\mu_{ki} = \tanh(W_h e_{ki} + b_h)$, $i \in \{1, 2, ..., t-1\}$, and w_c is a trainable history context vector [23]. Each view has its own history attention memory as in Figure 1. Two historical attention vectors h^1 and h^2 are retrieved from two history attention memory, respectively, and they are used for classification layer.

History attention memory module is designed for the task of an object with history samples. Its purpose is to store the history representation information of the object, which is called local knowledge. As in the medication recommendation task, the medication of current visit is related to not only current diagnosis and procedure sequences, but also medical records from previous visits.

Classification Layer. The encoding vectors, read vectors, and historical attention vectors from two views are concatenated together as the input for final classification as

$$\hat{y} = \sigma(\text{Concat}(e^1, e^2, r^1, r^2, h^1, h^2)) \tag{15}$$

where σ is the sigmoid function for binary classification and multi-label classification, and the softmax function for multi-class classification.

3.3 Loss Function

Cross-entropy loss is widely used for classification. However, class imbalance can lead to inefficient in training and affect the performance of the model. To deal with class imbalance, we use focal loss [13], which is a modified version of cross entropy, as our loss function. Given the real class $y_i \in \{0, 1\}$ and predicted probability $\hat{y}_i \in [0, 1]$, the focal loss for binary classification is defined as

$$\begin{aligned} L_i = &- \alpha y_i (1 - \hat{y}_i)^{\beta} \log \hat{y}_i \\ &- (1 - \alpha)(1 - y_i) \hat{y}_i^{\beta} \log(1 - \hat{y}_i) \end{aligned} \tag{16}$$

where α is the weighting factor to balance the importance of positive and negative samples, and β is the focusing parameter to down-weight the well-classified samples. Therefore, local loss emphasizes training on hard samples by reducing the loss contribution from easy samples. The focal loss for multi-label classification is

$$\begin{aligned} L_i = - \sum_{j=1}^{d_y} [&\alpha y_{ij} (1 - \hat{y}_{ij})^{\beta} \log \hat{y}_{ij} + \\ &(1 - \alpha)(1 - y_{ij}) \hat{y}_{ij}^{\beta} \log(1 - \hat{y}_{ij})] \end{aligned} \tag{17}$$

where d_y is the total number of labels, and $y_i = \{y_{i1}, y_{i2}, ..., y_{id_y}\}$, $y_{ij} \in \{0, 1\}$, $\hat{y}_i = \{\hat{y}_{i1}, \hat{y}_{i2}, ..., \hat{y}_{id_y}\}$, $\hat{y}_{ij} \in [0, 1]$. We set $\alpha = 0.6$ and $\beta = 2$ in this study as recommended by [13].

Since each category c_i needs a specific weight α_i, it is complicate to apply the focal loss for multi-class classification. Therefore, the original cross-entropy loss is used for multi-class classification.

4 EXPERIMENTS

In this section, we first compare the AMANet model to baselines and other state-of-the-art models in three tasks: medication recommendation, diagnosis-related group (DRG) classification, and invoice fraud detection. Then, we report the evaluation of each module in our model in the ablation study.

4.1 Medication Recommendation

The increasing volume of EHRs provides a great opportunity to improve the efficiency and quality of healthcare by developing predictive models. One critical task is to predict medications based on the medical records [8]. In this paper, we formulate medication recommendation as a multi-label classification problem and it recommends medicine combination based on diagnoses and medical procedures. The diagnoses and medical procedures are ordered in a hospital visit. Therefore, we treat the diagnoses and procedures in current visit as two sequential views and apply AMANet to predict the medication combinations.

Dataset. We use the medical data from the MIMIC-III database[2], which comprises rich medical information relating to patients within the intensive care units (ICUs) [9]. We follow the procedure similar to GAMENet to process the medical codes in this experiment [16]. The NDC drug code in MIMIC-III is mapped to third level ATC code as prediction label. The statistics of the dataset are summarized in Table 1.

Table 1: Statistics of MIMIC-III Dataset

	Count		
patients	6350		
visits	15032		
diagnosis categories	1958		
procedure categories	1430		
medication categories	151		
	Mean	Max	Min
count of visit	2.37	29	1
length of diagnosis sequence	13.63	39	1
length of procedure sequence	4.54	32	1
count of medication combinations	19.86	55	1

Baselines. We compare our model with the following baseline and state-of-the-art algorithms.

- Nearest: This algorithm uses the medication combinations from the previous visit as current visit's prediction.
- Logistic regression (LR): Logistic regression with L2 regularization is evaluated as a baseline model. The input is the multi-hot vector of diagnoses and procedures from the current visit.
- LEAP: LEAP (Learn to Prescribe) [25] uses a recurrent decoder to model label dependency and content-based attention to model the label-to-instance mapping for medicine recommendation.
- RETAIN: RETAIN (Reverse Time Attention) [4] is an interpretable model with reverse time attention mechanism by mimicking physician's practice. Therefore, recent visits are likely to receive high attention.
- DMNC: DMNC [12] uses dual memory neural computer for medication recommendation task.
- GAMENet: GAMENet [16] recommends medication through graph augmented memory module whose memory bank based on drug usage and drug-drug interaction (DDI) and dynamic memory based on patient's history. We have done hyperparameter searching for GAMENet such as learning rate and embedding size. And our result for GAMENet is better than the reported performances in [16] for medication recommendation task.

Evaluation Metrics. In order to measure the model performance, we use Jaccard similarity, F1-score, and the area under the Precision-Recall curve (PRAUC) as the evaluation metrics, where all of them are macro-averaged. We randomly divide the dataset into training, validation, and test with ratio 4:1:1 and report the performance from the test set.

Implementation Details. For our method, both token embedding size and position embedding size are 100. Both self-attention size and inter-attention size are 64 with 4 heads. The dynamic external memory is 256×64 matrix and has 4 read heads. The α and β of focal loss are 0.6 and 2, respectively. We train the model using the

[2] https://mimic.physionet.org/

Adam [10] optimizer at learning rate 0.0001 and stop at 15 epochs. The other models are evaluated the same as in GAMENet[16]. All models are trained on macOS Mojave system with 2.2 GHz Intel Core i7 and 16 GB memory.

Results. Table 2 summarizes our results and we can see that the AMANet model outperforms other models in all metrics. For Jaccard similarity and F1 score, our model achieves about 0.04-0.05 improvement comparing to the next-best model (GAMENet without DDI). AMANet also scores about 2% better than the next-best model LR in the PRAUC metric. When IA, DEM and HAM components are removed respectively, there have been different levels of performance decline. The removing of IA leads to the most performance drop among three components. We can conclude from these comparative experiments that each component is useful, especially the IA component.

Table 2: Experiments on Medication Recommendation

Method	Jaccard	F1	PRAUC
Nearest	0.2548	0.3398	0.3187
LR	0.4579	0.6177	0.7602
DMNC	0.4231	0.5849	0.6418
LEAP	0.4365	0.5997	0.6367
RETAIN	0.4647	0.6269	0.7336
GAMENet	0.4551	0.6172	0.7275
GAMENet(w/o DDI)	0.4796	0.6390	0.7484
AMANet(w/o IA)	0.5067	0.6641	0.7629
AMANet(w/o DEM HAM)	0.5136	0.6706	0.7710
AMANet(w/o DEM)	0.5201	0.6757	0.7716
AMANet(w/o HAM)	0.5200	0.6761	0.7746
AMANet	**0.5259**	**0.6809**	**0.7772**

4.2 DRG Classification

Diagnosis-related groups (DRGs) has been one of the most popular prospective payment systems in many countries. The basic idea of the DRG system is that patients admitted to a hospital are classified into a limited number of DRGs. Patients in each DRG share similar clinical and demographical patterns and are likely to consume similar amounts of hospital resources. The hospitals are reimbursed a set fee for the treatment in a single DRG category, regardless of the actual cost for an individual patient. Therefore, DRGs can put pressure on hospitals to control cost, to reduce the length of stay, and to increase the number of admitted inpatients. The DRG for each patient is assigned at the time of discharge[19]. Previous study has shown that accurately classifying an patient's DRG at the early stage of hospital visit can allow hospital effectively allocate limited hospital resources[5]. In this study, we extract diagnosis sequence and procedure sequence from EHRs as input and apply AMANet to predict DRG. Since DRGs are determined solely on current visit, we use AMANet without history attention memory module in this task.

Dataset. The dataset of this task is from a *Grade Three Class B* hospital in China. It contains 21356 hospital records of 2019. Each hospital record consists of two sequences: diagnosis sequence andprocedure sequence. The total number of DRG categories is 270. The statistics of the dataset are summarized in Table 3.

Table 3: Statistics of DRG Dataset

	Count		
records	21356		
diagnosis categories	3052		
procedure categories	1177		
DRG categories	270		
	Mean	Max	Min
length of diagnosis sequence	3.85	11	1
length of procedure sequence	1.3	5	1

Baselines. LEAP and GAMENet are not suitable for this task because LEAP outputs sequence and GAMENet is developed for medication recommendation purpose. Here we list the baseline models in this task as follows:

- LR, RETAIN, and DMNC are used as the baseline algorithms similar to medication recommendation task.
- Dual-LSTM: We add Dual-LSTM model, which encodes two input sequences using LSTM respectively and then combines two encoding vectors together for final classification layer.

Evaluation Metrics. We use accuracy, F1-score and PRAUC as the evaluation metric in this task. The dataset is randomly divided into training, validation, and test set with 4:1:1 ratio while the ratio of each category samples is kept the same for each set. We report the performance from the test set.

Implementation Details. The hyperparameters of our model are the same as in medication recommendation task except for the learning rate and epoch. We train the model at learning rate 0.0005 and stop at 10 epochs. All models are trained on macOS Mojave system with 2.2 GHz Intel Core i7 and 16 GB memory.

Results. Table 4 indicates that AMANet performs the best compared to the other four models in all three metrics. Compared to DMNC and Dual-LSTM, which are solely based on late fusion, our model achieves about 0.02 better performance for Accuracy and F1 score, and 0.04 for PRAUC. We can observe significant performance drop by removing IA module, which indicates the importance of IA in AMANet model to capture the inter-view interactions and improve the ability of representation.

Table 4: Experiments on DRG Classification

Method	Accuracy	F1	PRAUC
LR	0.7743	0.7458	0.6271
RETAIN	0.7631	0.7483	0.6048
DMNC	0.8473	0.8407	0.7324
Dual-LSTM	0.8525	0.8433	0.7305
AMANet(w/o IA)	0.7497	0.7416	0.6116
AMANet(w/o DEM)	0.8623	0.8612	0.7832
AMANet	**0.8712**	**0.8690**	**0.7975**

4.3 Invoice Fraud Detection

The value-added tax (VAT) is a general tax based on the value added to goods and services in the majority of countries. It is charged as a percentage of current sale price deducting the tax paid at the preceding stage. In the VAT collection process, the company provides the invoice from its selling for current tax calculation and the invoice of its buying from suppliers for the deduction. Some companies are specifically established for invoice fraud by illegally providing selling invoices to other companies without the business in between, and the companies buying invoices can get better VAT deduction. To identify the company involving in invoice fraud is a common task for the taxation agency. Here we formulate the invoice fraud detection task as a binary classification problem to identify the company involving in invoice fraud through historical invoice and tax declaration information. We construct the daily invoice and monthly tax declaration information as two sequential views and apply AMANet to identify the company with abnormal behaviors. Since there is only one sample for a company, we use AMANet without history attention memory module in this task.

Dataset. The dataset is from a texation agency in China. In the dataset, it contains 8700 companies in total and 1/3 of them is involved in the invoice fraud. For each company, two timely order sequences, i.e., daily invoice values and monthly tax declaration values, are used as input. The statistics of the dataset are summarized in Table 5. The invoice value and tax declaration value are continuous values and we discretize the values of each view into 2000 categories using quantile-based discretization function. Then, we use the one-hot encoding to represent the invoice and tax declaration values.

Table 5: Statistics of Invoice Fraud Dataset

	Count		
companies	8700		
invoice fraud companies	2900		
invoice categories	2000		
declaration categories	2000		
	Mean	Max	Min
length of invoice sequence	56.05	1256	1
length of declaration sequence	27.08	166	1

Baselines. The baseline models for this task are the same as in the DRG classification task, except for the changing from multi-class classification into binary classification.

Evaluation Metrics. It is the same as the DRG classification task.

Implementation Details. It is the same as the DRG classification task.

Results. Table 6 indicates that AMANet performs the best compared to the other four models in all three metrics. Compared to Dual-LSTM and DMNC, our model achieves about 3% better performance consistently. When IA component is removed, the effect is not as good as that of DMNC and Dual-LSTM. This demonstrates that self-attention and inter-attention mechanism coupling with memory can capture the intra-view and inter-view interactions effectively.

Table 6: Experiments on Invoice Fraud Detection

Method	Accuracy	F1	PRAUC
LR	0.7724	0.7222	0.8839
RETAIN	0.8628	0.8152	0.8910
DMNC	0.8669	0.8323	0.9005
Dual-LSTM	0.8655	0.8352	0.9220
AMANet(w/o IA)	0.8324	0.7805	0.9030
AMANet(w/o DEM)	0.8924	0.8556	0.9467
AMANet	**0.8938**	**0.8623**	**0.9503**

4.4 Hyperparameters

In this subsection, we study the effect of the hyperparameters to our model. Due to limit space, we use medication recommendation task as an example. We focus on four hyperparameter sets including embedding size d, learning rate LR, number of head NH and attention size AS, and α and β in the focal loss. We vary the values in each hyperparameter set with other sets fixed. The results for different hyperparameter settings are shown in Table 7. Training with focal loss yields more than 0.01 improvement over cross-entropy loss (i.e., $\alpha = 0.5$ and $\beta = 0$) in Jaccard similarity and F1 score. The differences among other hyperparameter sets are smaller than 0.006 in all three metrics, which is not significant. Therefore, our proposed model is not sensitive to most of the hyperparameters and adapting focal loss in training can boost the result significantly.

4.5 Ablation Study

We perform ablation studies on our model by removing specific module one at a time to explore their relative importance. The three modules evaluated in this section are inter-attention (IA), dynamic external memory (DEM), and history attention memory (HAM). As we can see from table 2, table 4, and table 6, the performance degrading in all ablation experiments indicates that all three modules in our model are necessary.

Inter-Attention. We design the inter-attention to capture the inter-view interaction across views by relating one input embedding to another input embedding and align different sequences. The inter-view encoding for sequence A from sequence B could be viewed as softly aligning sequence A to sequence B. Firstly, the relevance matrix is calculated between A and B, where each value in the matrix represents the relevance between one position in A and another position in B. Then encoding vector of each position in A is computed by relevance weighted summation of all positions in B. The removing of inter-attention can lead to the biggest performance dropping in all tasks and all three metrics comparing to memory modules, which indicates the crucial role of inter-attention in our model.

In medication recommendation task, all three metrics have dropped around 0.02 when excluding IA component. The removal of IA leads to dramatic performance drop over 0.12 in DRG classification. In invoice fraud detection task using AMANet model without IA,

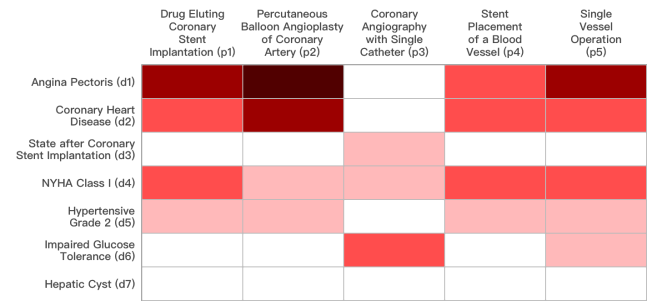
Table 7: Experiments on Varying Hyperparameters

Hyper Parameters	Jaccard	F1	PRAUC
Varying $d(LR = 0.0001, AS = 64, NH = 4, \alpha = 0.6, \beta = 2)$			
$d = 64$	0.5249	0.6802	0.7780
$d = 100$	0.5259	0.6809	0.7772
$d = 128$	0.5242	0.6797	0.7759
Varying $LR(d = 100, AS = 64, NH = 4, \alpha = 0.6, \beta = 2)$			
$LR = 0.0001$	0.5259	0.6809	0.7772
$LR = 0.0002$	0.5220	0.6774	0.7760
$LR = 0.0005$	0.5255	0.6804	0.7761
$LR = 0.001$	0.5235	0.6788	0.7747
Varying $AS, NH(LR = 0.0001, d = 100, \alpha = 0.6, \beta = 2)$			
$AS = 32, NH = 8$	0.5205	0.6763	0.7755
$AS = 32, NH = 4$	0.5207	0.6764	0.7749
$AS = 64, NH = 4$	0.5259	0.6809	0.7772
$AS = 64, NH = 2$	0.5239	0.6793	0.7779
Varying $\alpha, \beta(LR = 0.0001, d = 100, AS = 64, NH = 4)$			
$\alpha = 0.5, \beta = 0$	0.5056	0.6626	0.7753
$\alpha = 0.6, \beta = 0$	0.5183	0.6741	0.7756
$\alpha = 0.6, \beta = 1.5$	0.5248	0.6798	0.7791
$\alpha = 0.6, \beta = 2.0$	0.5259	0.6809	0.7772
$\alpha = 0.6, \beta = 2.5$	0.5198	0.6759	0.7722
$\alpha = 0.6, \beta = 3.0$	0.5208	0.6766	0.7721
$\alpha = 0.7, \beta = 2.5$	0.5218	0.6777	0.7763
$\alpha = 0.7, \beta = 3.0$	0.5241	0.6797	0.7763

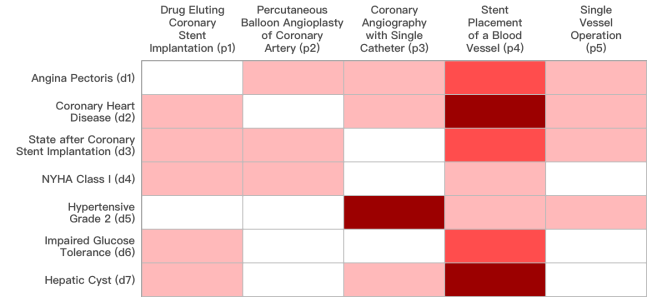
the accuracy, F1 score, and PRAUC drop 0.061, 0.082, and 0.047, respectively. The empirical results from three tasks suggest that integrating information from two views is beneficial for outcome prediction in certain tasks and our designed inter-attention mechanism can capture the inter-view interaction effectively in asynchronous setting.

Moreover, we observe the improved medical code representation with inter-attention module. We demonstrate this finding with an example from DRG classification task as in Figure 2. In this example, a patient with heart disease is treated with a sequence of medical procedures. All procedure codes and the diagnosis code d1, d2, d4 are heart disease related. Procedure code p3 is a diagnostic operation, which is different from other therapeutic operations here. In addition, hepatic cyst (d7) is not treated and it should not be related with any procedure codes. Figure 2(a) is well aligned with the medical knowledge above. However, the medical code representation from AMANet without IA module is not consistent with the medical knowledge as in Figure 2(b). Therefore, we can conclude that the inter-attention module can contribute to the representation learning across code sets.

Memory. We incorporate two memory modules in our model to store the global and local information by using dynamic external memory and history attention memory, respectively. In medication recommendation task, the dynamic external memory and history attention memory contribute similarly to the overall prediction in all three metrics. In the above three tasks, the effect is reduced when the memory component is removed. Both modules are necessary based on the empirical experimental result.



(a) AMANet with IA module



(b) AMANet without IA module

Figure 2: The cosine similarity between a diagnosis sequence (vertical axis) and a procedure sequence (horizontal axis). Each block is colored by the similarity value. The positive cosine similarity value is colored in red, while the negative similarity value is colored in white.

5 CONCLUSIONS

In this work, we propose a novel architecture AMANet based on attention and memory for dual asynchronous sequential learning. The new model learns the intra-view interaction and inter-view interaction through self-attention and inter-attention mechanism, respectively. We use dynamic external memory to represent the common knowledge from the data. We also introduce a history attention memory module to store the historical representation of the same object. We evaluate our model on medication recommendation task, DRG classification task, and invoice fraud detection with superior performance comparing to other baseline and state-of-the-art models. The ablation study emphasizes the importance of modeling inter-view interactions and the inter-attention module could model the inter-view interaction effectively. The AMANet framework is formulated using dual-view sequences in this study, but it can be applied for multi-view sequences directly. In the future, we will employ this framework to study multi-view sequential learning problems.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This work is supported in the Department of Data Intelligence Products of Alibaba Cloud in Alibaba Group.

REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations (ICLR)*.

[2] T. Baltrušaitis, C. Ahuja, and L. P. Morency. 2019. Multimodal machine learning: A survey and taxonomy. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, Vol. 41(2). 423–443.

[3] K. Cho, B. Merrienboer, C. Gulcehre, F. Bougares, H. Schwenk, and Y. Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. 1724–1734.

[4] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart. 2016. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*. 3504–3512.

[5] Daniel Gartner, Rainer Kolisch, Daniel B Neill, and Rema Padman. 2015. Machine learning approaches for early DRG classification and resource allocation. *INFORMS Journal on Computing* 27, 4 (2015), 718–734.

[6] A. Graves, G. Wayne, M. Reynolds, T. Harley, I. Danihelka, A. Grabska-Barwińska, and et al. 2016. Hybrid computing using a neural network with dynamic external memory. In *Nature*, Vol. 538(7626). 471.

[7] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. In *Neural Computation*, Vol. 9(8). 1735–1780.

[8] B. Jin, H. Yang, L. Sun, C. Liu, Y. Qu, and J. Tong. 2018. A Treatment Engine by Predicting Next-Period Prescriptions. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1608–1616.

[9] A. E. Johnson, T. J. Pollard, L. Shen, H. L. Li-wei, M. Feng, M. Ghassemi, and et al. 2016. MIMIC-III, a freely accessible critical care database. In *Scientific Data*, Vol. (3). 160035.

[10] D. P. Kingma and J. Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

[11] T. N. Kipf and M. Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICLR)*.

[12] H. Le, T. Tran, and S. Venkatesh. 2018. Memory fusion network for multi-view sequential learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 1637–1645.

[13] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. 2980–2988.

[14] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, and J. Gao. 2017. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining (KDD)*. 1903–1911.

[15] S. S. Rajagopalan, L. P. Morency, T. Baltrusaitis, and R. Goecke. 2016. Extending long short-term memory for multi-view structured learning. In *European Conference on Computer Vision (ECCV)*. 338–353.

[16] J. Shang, C. Xiao, T. Ma, H. Li, and J. Sun. 2019. Graph augmented memory networks for recommending medication combination. In *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI)*. 1126–1133.

[17] H. Song, D. Rajan, J. J. Thiagarajan, and A. Spanias. 2018. Attend and diagnose: Clinical time series analysis using attention models. In *Thirty-Second AAAI Conference on Artificial Intelligence*. 4091–4098.

[18] A. Vaswani, N. Shazeer, N. Shazeer, J. Uszkoreit, L. Jones, A. N. Gomez, and et al. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*. 5998–6008.

[19] Zhaoxin Wang, Rui Liu, Ping Li, and Chenghua Jiang. 2014. Exploring the transition to DRGs in developing countries: a case study in Shanghai, China. *Pakistan journal of medical sciences* 30, 2 (2014), 250.

[20] J. Weston, S. Chopra, and A. Bordes. 2015. Memory networks. In *3rd International Conference on Learning Representations (ICLR)*.

[21] C. Xu, D. Tao, and C. Xu. 2013. A survey on multi-view learning. In *ArXiv*. arXiv:1304.5634.

[22] Y. Xu, S. Biswal, S. R. Biswal, K. O. Maher, and J. Sun. 2018. RAIM: Recurrent Attentive and Intensive Model of Multimodal Patient Monitoring Data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD)*. 2565–2573.

[23] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. 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 (NAACL-HLT)*. 1480–1489.

[24] A. Zadeh, P. P. Liang, N. Mazumder, S. Poria, E. Cambria, and L. P. Morency. 2018. Memory fusion network for multi-view sequential learning. In *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*. 5634–5641.

[25] Y. Zhang, R. Chen, J. Tang, W. F. Stewart, and J. Sun. 2017. LEAP: learning to prescribe effective and safe treatment combinations for multimorbidity. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1315–1324.

[26] J. Zhao, X. Xie, X. Xu, and S. Sun. 2017. Multi-view learning overview: Recent progress and new challenges. In *Information Fusion*, Vol. 38. 43–54.

Algorithm 1 Training AMANet

Input: LR, d, AS, NH, α, β, $epoch$, Sequences X^1, Sequences X^2 and labels y
Output: Jaccard, F1, PRAUC

procedure TRAINING
  Initialize $W_E^{\{1,2\}}$ (Two Token Embedding Matrices), $W_\xi^{\{1,2\}}$ (Two DEM Interface Matrices), $W_{DEM}^{\{1,2\}}$ (Two DEM Content Matrices), $W_h^{\{1,2\}}$, $b_h^{\{1,2\}}$ (Two HAM Matrices and Two HAM Vectors), $w_c^{\{1,2\}}$ (Two History Context Vectors)
  for $i = 1 \rightarrow epoch$ **do**
    for $object_i$ in training set **do**
      for $visit_j$ in $object_i'$ visits **do**
        $X_{ij}^1, X_{ij}^2, y_{ij} \leftarrow visit_j$
        $M_{ij}^{\{1,2\}} \leftarrow$ onehot encoding of $X_{ij}^{\{1,2\}}$
        $TE_{ij}^{\{1,2\}} \leftarrow M_{ij}^{\{1,2\}} \times W_E^{\{1,2\}}$
        $PE_{ij}^{\{1,2\}} \leftarrow$ positional encoding of $X_{ij}^{\{1,2\}}$
        $E_{ij}^{\{1,2\}} \leftarrow TE_{ij}^{\{1,2\}} + PE_{ij}^{\{1,2\}} \in \mathbb{R}^{L_{ij}^{\{1,2\}} \times d}$
        $E_{intra_{ij}}^{\{1,2\}} \leftarrow \text{MultiHead}(E_{ij}^{\{1,2\}})$
        $\xi_{ij}^{\{1,2\}} \leftarrow \text{pool}(E_{intra_{ij}}^{\{1,2\}} \times W_\xi^{\{1,2\}})$
        update $W_{DEM}^{\{1,2\}}$ by $\xi_{ij}^{\{1,2\}}$
        $r_{ij}^{\{1,2\}}$ read from $W_{DEM}^{\{1,2\}}$
        $E_{inter_{ij}}^1 \leftarrow \text{MultiHead}(E_{ij}^1,\ E_{ij}^2)$
        $E_{inter_{ij}}^2 \leftarrow \text{MultiHead}(E_{ij}^2,\ E_{ij}^1)$
        $e_{ij}^{\{1,2\}} \leftarrow \text{concat}(\text{pool}(E_{intra_{ij}}^{\{1,2\}}),\ \text{pool}(E_{inter_{ij}}^{\{1,2\}}))$
        save $e_{ij}^{\{1,2\}}$ to HAM
        $h_{ij}^{\{1,2\}} \leftarrow \sum_{k=1}^{j-1} \alpha_{ik}^{\{1,2\}} e_{ik}^{\{1,2\}}$
          where $\alpha_{ik}^{\{1,2\}} = \frac{e^{\mu_{ik}^{\{1,2\}^T} w_c^{\{1,2\}}}}{\sum_l e^{\mu_{il}^{\{1,2\}^T} w_c^{\{1,2\}}}}$
            $\mu_{ik}^{\{1,2\}} = \tanh(W_h^{\{1,2\}} e_{ik}^{\{1,2\}} + b_h^{\{1,2\}})$
            $k \in 1 \rightarrow j-1$
        $\hat{y}_{ij} \leftarrow classify(\text{concat}(e_{ij}^1, e_{ij}^2, r_{ij}^1, r_{ij}^2, h_{ij}^1, h_{ij}^2))$
        compute loss
        update the network parameters basis on loss using Back Propagation
      end for
    end for
    compute Jaccard, F1, PRAUC on validation set
  end for
  best_model $\leftarrow \arg\max_{model}\{\text{F1}\}$
  return best_model
end procedure
procedure EVALUATION
  compute Jaccard, F1, PRAUC on test set
  return Jaccard, F1, PRAUC
end procedure

Algorithm 2 Training AMANet without HAM

Input: LR, d, AS, NH, α, β, $epoch$, Sequences X^1, Sequences X^2 and labels y
Output: Accuracy, F1, PRAUC

procedure TRAINING
  Initialize $W_E^{\{1,2\}}$ (Two Token Embedding Matrices), $W_\xi^{\{1,2\}}$ (Two DEM Interface Matrices), $W_{DEM}^{\{1,2\}}$ (Two DEM Content Matrices)
  for $i = 1 \rightarrow epoch$ **do**
    for $object_i$ in training set **do**
      $X_i^1, X_i^2, y_i \leftarrow object_i$
      $M_i^{\{1,2\}} \leftarrow$ onehot encoding of $X_i^{\{1,2\}}$
      $TE_i^{\{1,2\}} \leftarrow M_i^{\{1,2\}} \times W_E^{\{1,2\}}$
      $PE_i^{\{1,2\}} \leftarrow$ positional encoding of $X_i^{\{1,2\}}$
      $E_i^{\{1,2\}} \leftarrow TE_i^{\{1,2\}} + PE_i^{\{1,2\}} \in \mathbb{R}^{L_i^{\{1,2\}} \times d}$
      $E_{intra_i}^{\{1,2\}} \leftarrow \text{MultiHead}(E_i^{\{1,2\}})$
      $\xi_i^{\{1,2\}} \leftarrow \text{pool}(E_{intra_i}^{\{1,2\}} \times W_\xi^{\{1,2\}})$
      update $W_{DEM}^{\{1,2\}}$ by $\xi_i^{\{1,2\}}$
      $r_i^{\{1,2\}}$ read from $W_{DEM}^{\{1,2\}}$
      $E_{inter_i}^1 \leftarrow \text{MultiHead}(E_i^1,\ E_i^2)$
      $E_{inter_i}^2 \leftarrow \text{MultiHead}(E_i^2,\ E_i^1)$
      $e_i^{\{1,2\}} \leftarrow \text{concat}(\text{pool}(E_{intra_i}^{\{1,2\}}),\ \text{pool}(E_{inter_i}^{\{1,2\}}))$
      $\hat{y}_i \leftarrow classify(\text{concat}(e_i^1,\ e_i^2,\ r_i^1,\ r_i^2)$
      compute loss
      update the network parameters basis on loss using Back Propagation
    end for
    compute Accuracy, F1, PRAUC on validation set
  end for
  best_model $\leftarrow \arg\max_{model}\{\text{F1}\}$
  return best_model
end procedure
procedure EVALUATION
  compute Accuracy, F1, PRAUC on test set
  return Accuracy, F1, PRAUC
end procedure

A PSEUDO-CODE

The pseudo-code of our model is as **Algorithm 1** and **Algorithm 2**. **Algorithm 1** is applicable to medication recommendation task, **Algorithm 2** is applicable to DRG classification and invoice fraud detection task. The code is opened in https://github.com/non-name-2020/AMANet. The dataset of medication recommendation task is available in https://mimic.physionet.org. And we will open the datasets of DRG classification and invoice fraud detection task.