Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full length article

MEGACare: Knowledge-guided multi-view hypergraph predictive framework for healthcare

Jialun Wu [a,b], Kai He [e], Rui Mao [c], Chen Li [b,d], Erik Cambria [c,*]

[a] *School of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China*
[b] *Shaanxi Provincial Key Laboratory of Big Data Knowledge Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China*
[c] *School of Computer Science and Engineering, Nanyang Technological University, 50 Nanyang Ave, Singapore, 639798, Singapore*
[d] *National Engineering Lab for Big Data Analytics, Xi'an Jiaotong University, Xi'an, Shaanxi, 710049, China*
[e] *Saw Swee Hock School of Public Health, University of Singapore, 117549, Singapore*

ARTICLE INFO

Keywords:
Electronic health record
Healthcare
Hypergraph
Multi-view learning
Information bottleneck

ABSTRACT

Predicting a patient's future health condition by analyzing their Electronic Health Records (EHRs) is a trending subject in the intelligent medical field, which can help clinicians prescribe safely and effectively, and also make more accurate diagnoses. Benefiting from powerful feature extraction capabilities, graph representation learning can capture complex relationships and achieve promising performance in many clinical prediction tasks. However, existing works either exclusively consider single domain knowledge with an independent task or do not fully capitalize on domain knowledge that can provide more predictive signals in the code encoding stage. Moreover, the heterogeneous and high-dimensional nature of EHR data leads to a deficiency of hardly encoding implicit high-order correlations. To address these limitations, we proposed a knowledge-guided Multi-viEw hyperGrAph predictive framework (MEGACare) for diagnosis prediction and medication recommendation. Our MEGACare leveraged multi-faceted medical knowledge, including ontology structure, code description, and molecular information to enhance medical code presentations. Furthermore, we constructed an EHR hypergraph and a multi-view learning framework to capture the high-order correlation between patient visits and medical codes. Specifically, we propose three perspectives around the pairwise relationship between patient visits and medical codes to comprehensively learn patient representation and enhance the robustness of our framework. We evaluated our MEGACare framework against a set of state-of-the-art methods for two clinical outcome prediction tasks in the public MIMIC-III dataset, and the results showed that our proposed framework was superior to the baseline methods.[1]

1. Introduction

Precision medicine refers to individualized diagnosis and treatment strategies that match patient characteristics [1,2]. The widespread adoption and rapid accumulation of Electronic Health Records (EHRs) of patient histories in Intensive Care Units (ICUs) have enabled researchers to investigate data-driven models to assist and facilitate clinical decision-making. Among them, diagnosis prediction and medication recommendation are two of the most concerning issues [3–6], which can directly facilitate early intervention to prevent disease progression, help clinicians formulate safe and effective prescriptions, and ultimately improve the quality of personal healthcare. The mining of extensive information concealed in EHR data holds promise as a feasible approach for disease prediction and medication recommendation.

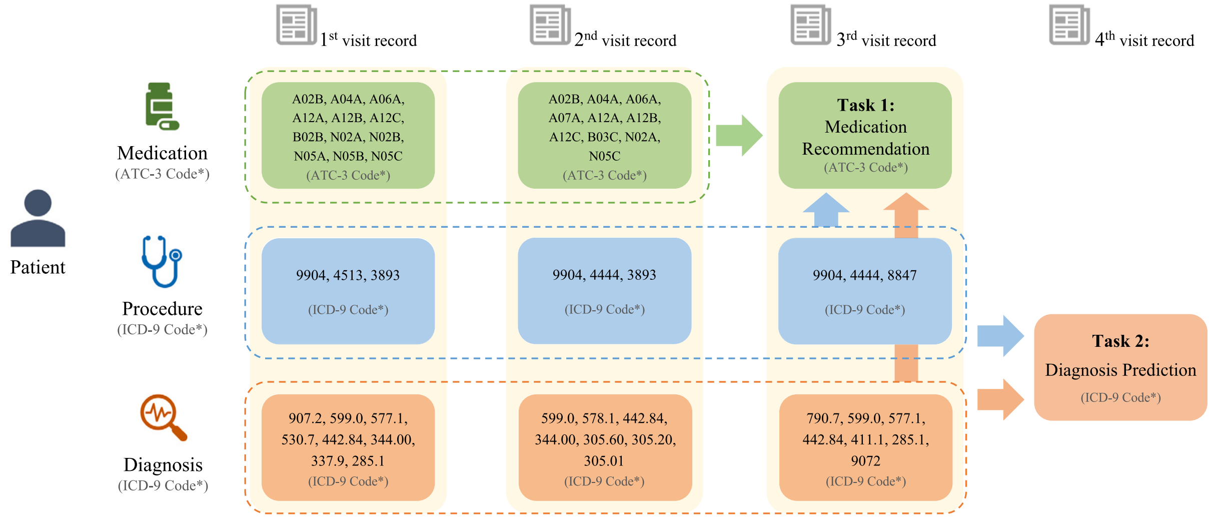
As shown in Fig. 1, crucial data such as diagnosis histories, procedures, and medication in EHR can provide sufficient information for dealing with the two tasks mentioned. Nonetheless, such EHR data that amalgamate information from various origins may exhibit a variety of formats, standards, and terminologies, alongside the presence of errors, missing values, or incomplete data.

These inherent issues in EHR data, such as heterogeneity and data deviation, present formidable obstacles to traditional machine learning approaches that rely on manual feature engineering. In recent years, deep learning methods that can automatically and efficiently extract task-relevant features from EHR data have been widely applied to various clinical prediction tasks and achieved promising results [7–10].

[*] Corresponding author.
E-mail address: cambria@ntu.edu.sg (E. Cambria).

[1] Our code and data are released at https://github.com/senticnet/MEGACare.

https://doi.org/10.1016/j.inffus.2023.101939
Received 1 April 2023; Received in revised form 24 June 2023; Accepted 18 July 2023
Available online 22 July 2023



* The Anatomical Therapeutic Chemical (ATC) Classification System is a drug classification system that classifies the active ingredients of medications. Each code represents a specific medication information in ATC-3 (*e.g.,* A02B represents "Drugs for peptic ulcer and gastro-oesophageal reflux disease").

* The International Classification of Diseases, Ninth Revision, Clinical Modification (ICD-9-CM) is the official system of assigning codes to diagnoses and procedures. Each code represents a specific diagnosis or procedure information in ICD-9 (*e.g.,* 599.0 represents "Urinary tract infection", 9904 represents "Packed cell transfusion").

Fig. 1. Sample EHR sequence with three visits and simple visualizations of two clinical prediction tasks.

Deep learning-based research with EHR data can be categorized into three groups, namely recurrent neural network (RNN)-based methods [1,11], graph neural network (GNN)-based methods [12,13], and knowledge-guided methods [14–16]. To learn patient representations, the RNN-based methods focus on capturing the time-order features; The GNN-based methods focus on spatial features among medical codes of EHR data. However, separated time-order and spatial features are straightforward and low-order, which cannot fully reflect robust and significant features for diagnosis prediction and medication recommendation. Noticeably, EHR data are typically structured based on individual patient visits, thereby disregarding significant high-level attributes such as potential correlations among various or identical medical codes over multiple visits. Considering some medical codes of rare diseases infrequently appear in the EHR data, the knowledge-guided methods introduce external medical knowledge to provide supplementary information. The efficacy of these models is highly contingent upon the selection of external knowledge because the utilization of incorrect or limited domain knowledge will lead to performance degradation. Summarily, previous works have limits in (1) encoding the implicit higher-order correlations in EHR data, as well as (2) fully capitalizing on the abundant information in multi-source domain knowledge.

Specifically,

(1) Insufficient higher-order correlations. Current approaches typically employ RNNs and GNNs to construct EHR data in order to analyze the temporal and pairwise relationships between medical codes [1,11], while disregarding the implicit high-level correlations between patient visits and medical codes or between historical and current visit (*e.g.*, the same medical code may have different interpretations when applied to different patient visits, and different medical codes may have similar interpretations when applied to similar patient visits). These correlations are a fundamental component of the data modeling process, which offer a more comprehensive approach to encode EHR data.

(2) Domain knowledge underutilization. Existing methods for encoding medical codes either solely contemplate single domain knowledge with independent tasks (*e.g.*, encoding diagnosis information based on code descriptions and medical ontology graphs) [14,15,17], or do not capitalize on domain knowledge that can furnish more predictive signals *e.g.*, learning medication representations from atom graphs is not as efficacious as exploiting substructure-level correlations [18,19]. None of the existing works has considered both issues, simultaneously.

To address the above limits, we propose a knowledge-guided Multi-viEw hyperGrAph predictive framework (MEGACare) for clinical outcome predictions (*i.e.*, diagnosis prediction, and medication recommendation). Our proposed MEGACare utilized a variety of information fusion techniques, including the integration of multiple domain knowledge for more supportive data, the amalgamation of multi-view features for better representations, and the combination of multiple losses for stable training.

Particularly, to encode higher-order correlations, MEGACare obtains patient representations containing high-order correlations from EHR data through a multi-view hypergraph network in the patient representation module. We first construct an EHR hypergraph with medical codes as nodes and each visit record as a hyperedge. Additionally, three different views are designed, based on the hypergraph to fuse multi-view representations and enhance hypergraph modeling: *the medical code graph* learns the strength of pairwise connections between codes; *the enhanced hypergraph* optimizes the hypergraph structure by strengthening/reducing the connections between a visit and a code that are relevant/irrelevant to tasks; and *the sub-hypergraph* aggregates the hyperedges to form a sub-hypergraph to learn the relation between historical and current visits.

Furthermore, the multi-view hypergraph network is optimized by a multiple-loss combination. For example, following the information bottleneck (IB) principle [20], a hypergraph IB loss and a multi-view IB loss are fused into an information-constrained loss to alleviate the noise and learn accurate patient representations.

To address the problem of domain knowledge under-utilization, MEGACare incorporates multiple domain knowledge through the message-passing process with information flow in the code initialization module. MEGACare amalgamates the semantic information of code descriptions with the hierarchical information of ontology structures and uses an ontology loss to pre-train the diagnosis and procedure embedding. Additionally, MEGACare constructs a molecule substructure graph and combines it with a triplet learning loss to pre-train the medication embedding.

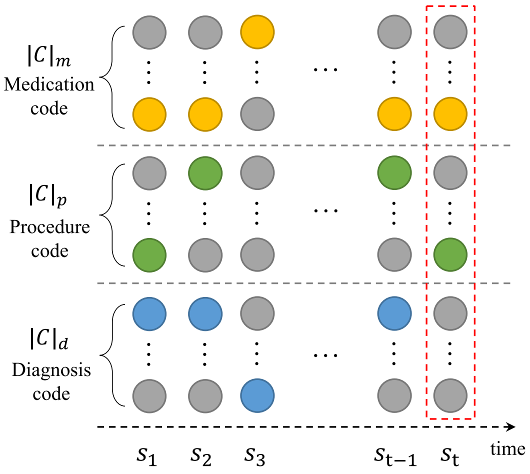


Fig. 2. The EHR data of a patient consists of a sequence of visits $s_1, s_2, \dots, s_T$. The red box means visit s_t. Each visit contains a subset of medical codes (*i.e.*, the diagnosis code d, the procedure code p, and the medication code m.), which could be represented by a binary vector $s_t \in \{0,1\}^{|C|_d,|C|_p,|C|_m}$, where the ith element is set to 1 (the color point) if the tth visit contains the medical code c_i, otherwise 0 (the gray point).

In summary, our contributions are as follows:

- We construct an EHR hypergraph to obtain the higher-order and longitudinal correlations among EHR data. MEGACare employs a multi-view architecture based on the hypergraph and further integrates information from different views to cooperatively learn comprehensive patient representations.
- We propose a novel medical code embedding method that fully leverages domain knowledge through pre-training with an information flow that incorporates the semantic information of code description, ontology hierarchical information, and molecular substructural information.
- We conduct extensive experiments on the MIMIC-III database [21]. MEGACare outperforms the strongest state-of-the-art method with 0.70%, 0.97%, 1.16% improvements in Code-Level Precision@10, 20, 30 respectively, and 0.51%, 0.76%, and 0.57% improvements in Visit-Level Precision@10, 20, 30 respectively (diagnosis prediction task), and 0.47% reduction in antagonistic drug–drug interaction (ADDI) rate, 0.85% improvements in synergistic drug–drug interaction (SDDI) rate, 0.95% in Jaccard similarity, 1.12% in F1-score and 1.29% in PRAUC score (medication recommendation task).

2. Related works

2.1. Deep learning for mining EHRs

In the medical informatics field, there has been a growing trend of researchers utilizing deep learning methods to extract intricate EHR data, construct data-driven models, and provide assistance for personalized and precision medicine in recent years. As illustrated in Fig. 2, the EHR data contain rich and diverse patient medical history and health information.

The early warning system is a significant application in EHR data mining, which leverages latent patterns in patient medical records to predict future health information, such as diagnostic information and medication information [22]. A critical challenge for such tasks is how to accurately and effectively match complex patient health status with the relevant medical codes. Such tasks rely on informative patient representations, efficient medical code encoding, and multiple domain knowledge (*e.g.*, ontology structure and code description of diagnosis codes, molecular graphs, and varied DDI relationships).

Most of the existing healthcare application models are reliant upon RNN and attention mechanisms [23–25] due to their superior capacity to learn time-series EHR data. These methods can be further classified into instance-based and longitudinal-based approaches. *Instance-based methods* learn the patient representations from the current visit. For example, LEAP [26] was proposed as a multi-instance and multi-label learning framework with an attention mechanism to predict medications, based on the clinical events within the current visit. *Longitudinal-based methods* learn the patient representations by capturing dependencies among clinical events from longitudinal patient history. RETAIN [11] was proposed with a two-level reverse time attention mechanism (*i.e.*, visit-level attention and code-level attention) to model the patient's longitudinal history record. Dipole [1] applied Bi-LSTM to resist the performance degradation brought by long sequence data, thereby enhancing the temporal data modeling ability of the predictive model. GAMENet [27] was proposed with a memory network to store patient history representations as references during medication recommendations.

Although these methods have achieved good performance, they fail to consider the graph structure of EHR data, thus precluding the possibility of capturing the intricate relationships between medical codes and patients. Therefore, EHR data can be treated as a medical code graph. Existing works utilize the graph neural networks (GNNs) [28] to encode the EHR graph and to learn the connections between different nodes. The improved Graph Convolutional Transformer (GCT) [29] was proposed to learn potential correlations between medical codes. SafeDrug [18] was proposed, using a graph-based molecule encoder with MPNN and learnable fingerprints, aiming at aggregating and convolving atom information across single molecule graphs into vector embeddings. In addition to the domain knowledge of molecular graphs, the knowledge-guided methods, such as GRAM [14], KAME [15] and G-BERT [30], which incorporate the domain knowledge of medical ontology information to solve the insufficient data problem through an additional embedding process.

However, the existing medical code encoding methods either only consider single-domain knowledge or neglect domain knowledge that could offer more predictive signals. Meanwhile, current methods only analyze the spatio-temporal and pairwise relationships between medical codes, disregarding the implicit high-order correlations in the data. These correlations are a fundamental component of the data modeling process, which offer a more comprehensive approach to interpreting EHR data.

2.2. Hypergraph neural networks

There are increasing works that focused on efficient graph representation learning, which has wide-ranging applications in molecular structure deduction as well as in healthcare. However, graphs have limitations in representing high-order relationships. In hypergraphs, intricate associations are represented by hyperedges that can link an indefinite amount of nodes [31]. The utilization of hyperedges and hypergraphs has enabled the expansion of binary relationships in graph structures to multivariate relationships, thereby allowing to represent the intricate correlations between data. Compared with graph modeling [32,33], hypergraph modeling [34] has been gaining traction in recent years due to its increased flexibility in depicting complex data associations. It is possible to construct a hypergraph from the patient visit record — medical code relationship in the EHR system, where medical codes are represented as nodes and each patient visit is represented as a hyperedge.

In order to facilitate the acquisition of an initial hypergraph structure to improve performance in subsequent predictive tasks, deep hypergraph representation learning techniques can be divided into spectral-based and spatial-based approaches.

(1) The spectral-based methods define the hypergraph convolution from the hypergraph spectral theory [35], which mainly design different message-passing strategies, and show advantages in learning

high-order representations. Hypergraph neural networks [36] first formulate the spectral convolution on hypergraph structure as a neural network layer, which uses the hypergraph Laplacian eigenbasis to approximate the Fourier transformation. In HyperGCN [37], each hyperedge is estimated as a simple graph by selecting two central nodes and connecting the rest nodes in the hyperedge with these two centers, and thus the original problem is approximated as a graph learning problem. Gao et al. [38] assigned different weights to hyperedges, where the weights are adaptively adjusted during the learning process.

(2) The spatial-based methods formulate the hypergraph convolution as aggregating information from immediate neighbors, where the added layer allows messages to be propagated to more distant neighbors. HyperSAGE [39] characterized the spacial-based message-passing process on hypergraph by a two-stage procedure, i.e., from nodes to hyperedges and from hyperedges to nodes. To solve the high computational cost and over-smoothing issue in HyperSAGE, UniGNN [40] further proposed a general framework to describe the spacial-based message propagation method in both graph and hypergraph neural networks. UniGNN uses clique expansion to convert hypergraphs into graphs and applies graph embedding techniques. The HGNN method [36] uses clique expansion to convert hypergraphs into graphs and applies graph embedding techniques. A more general HGNN framework [41] that implements HGNN layers as a composition of two multiset functions and covers most existing HGNN propagation methods. However, since noisy data often exist in the real world, it becomes particularly important to remove irrelevant information in the initial graph and learn an improved hypergraph structure.

In our work, we propose a multi-view representation learning framework based on EHR hypergraphs for clinical prediction tasks. First, our proposed EHR hypergraph can model high-order dependencies between codes and patient visits. Second, we construct three different views, based on this hypergraph to jointly learn patient representations. Thirdly, we optimize the hypergraph structure, which helps to reduce the noisy data existing in medical scenarios, thereby enhancing the prediction accuracy of the model.

3. Preliminaries and task formulation

3.1. Preliminaries

Longitudinal Patient Records. The longitudinal EHRs contain a variety of sequential medical events of patients, *e.g.* diagnosis, procedures, and medications. The sequence medical codes in EHR are denoted as: $c_1, c_2, \dots, c_{|C|}$, where $|C|$ is the total number of unique medical codes. Each patient can be represented as a series of medical codes, taking patient i as an example, $\mathrm{S}_i = [s_i{}^{(1)}, s_i{}^{(2)}, \dots, s_i{}^{(\mathrm{T}_i)}]$, where $i \in \{1, 2, \cdots, \mathrm{N}\}$, N is the total number of all patients, and T_i denotes the total visit times of patient i. We utilize the medical codes set $[d_i^{(t)}, p_i^{(t)}, m_i^{(t)}]$ to represent the clinical visit $e_i^{(t)}$ of the patient i, where $d_i{}^{(t)} \in \{0,1\}^{|D|}$, $p_i{}^{(t)} \in \{0,1\}^{|P|}$ and $m_i{}^{(t)} \in \{0,1\}^{|M|}$ are multi-hot diagnoses, procedure, and medication vectors, respectively, $|\cdot|$ denotes the cardinality, while D, P, M are the diagnosis, procedure, and medication sets, respectively. Meanwhile, the disease state from $s_i{}^{(1)}$ to $s_i{}^{(t)}$ of patient i is denoted as $S_i{}^{1:t}$. In the rest of this paper, we drop the subscript i whenever it is unambiguous.

Medical Ontology. Medical codes are usually categorized according to the classification system with a tree structure possessing the parent–child relations such as ICD-9 diagnosis and procedure code ontology graph and the ATC medication code ontology graph. We use O_d, O_p, O_m to denote the ontology for diagnosis, procedure, and medication, respectively.

3.2. Task formulation

Based on the notations above, we introduce the problems of diagnosis prediction and medication recommendation as follows:

- **Diagnosis Prediction** is one of the core research tasks in EHR data mining, which aims to learn a function $f_{DP}(\cdot)$ to predict the future visit information according to the historical visit records, given a sequence of visits $s_i{}^{(1)}, s_i{}^{(2)}, \dots, s_i{}^{(\mathrm{T}_i)}$. Assuming that at the tth visit of a patient, given the current and historical diagnoses and clinical procedures $[d^{(1)}, d^{(2)}, \dots, d^{(t)}]$ and $[p^{(1)}, p^{(2)}, \dots, p^{(t)}]$, the function $f_{DP}(\cdot)$ can predict the next visit diagnosis codes.
- **Medication Recommendation** task aims to learn a function $f_{DR}(\cdot)$ to recommend medications at each time-step t of different patients. Based on the same assumption, the function $f(\cdot)$ can predict medications $\hat{m}^{(t)} \in \{0,1\}^{|M|}$, given the current and historical visit records. Our goal is to make the prediction $\hat{m}^{(t)}$ as close as possible to the real prescription $m^{(t)} \in \{0,1\}^{|M|}$.

In this sense, these two tasks can be regarded as multi-label classification problems. The main notations used in this paper are listed in Table 1.

4. Method

In this section, we describe the architecture of MEGACare for clinical outcome predictions. As illustrated in Fig. 3, our proposed framework consists of the following three modules:

(I) The **Code Initialization module** that leverages multiple domain knowledge to derive embeddings of different medical codes. This module consists of two code encoding components that operate independently. The initialization embeddings of diagnosis and procedure codes are established based on the semantic and hierarchical information of the code description and the knowledge graph, and are further constrained by an ontology loss. The initialization embedding of medication code, on the other hand, encodes the molecular graph using a message-passing network and is constrained by a triplet learning loss.

(II) The **Patient Representation module** that learns patient representations from longitudinal history data using higher-order features. This module is designed to construct an EHR hypergraph and employ multi-view learning from various perspectives, including the code graph view, enhanced hypergraph view, and sub-hypergraph view. The learning process is constrained by the information bottleneck principle to enhance the comprehensiveness and reliability of the patient representation.

(III) The **Outcome Prediction module** predicts the healthcare outcomes of the target patient based on a feed-forward neural network.

Each component of the MEGACare is detailed below in turn.

4.1. Code initialization module

This module is used for robust initialization representations of codes of diagnosis, procedure, and medication information in EHR data. As shown in Fig. 3.I, the proposed Code Initialization Module is composed of the diagnosis and procedure encoding part in Fig. 3.I.(a) and (b), and the medication code encoding part in Fig. 3.I.(c). Specifically,

- In the diagnosis and procedure code encoding part, each medical code has its formal description and is mapped to the corresponding ontology structure. The semantic embeddings of the medical code and the hierarchical features of the ontology structure are fused through a message-passing process with information flow, resulting in the generation of diagnosis and procedure embeddings.

Table 1
Notations used in MEGACare.

Notation	Description
${S_i}^{1:\mathrm{t}} = {s_i}^{(1)}, \dots, {s_i}^{(\mathrm{t})}$	The health status of patient i
${s_i}^{(\mathrm{t})} = \{d_i^{(t)}, p_i^{(t)}, m_i^{(t)}\}$	The clinical visit of the patient i at visit t
D, P, M	The diagnosis, procedure, and medication set
$c_e, c_e^{inh}, c_e^{uni}$	The medical code, inherent, and unique embeddings
$F_{\mathrm{Agg}}(\cdot, \cdot, \cdot)$	The node embedding aggregation function
$ch(\cdot), pa(\cdot)$	The child node set, the parent node set
X_d, X_p, X_m	The diagnosis, procedure, and medication embeddings
$\mathcal{HG} = (\mathcal{V}, \mathcal{E})$	The EHR hypergraph
$V_{\mathcal{CG}}, V_{\mathcal{EH}}, V_{\mathcal{SH}}$	The code-graph, enhanced hypergraph, and sub-hypergraph view
$\mathcal{HG}_{\mathrm{HSO}}$	The hypergraph structure optimization
$\mathcal{HG}_{\mathrm{HSO\_HD}}$	The Hyperedge Dropping strategy
$\mathcal{HG}_{\mathrm{HSO\_NA}}$	The Node Adding strategy
$\mathcal{HG}_{\mathrm{HSO\_ND}}$	The Node Deleting strategy
$\mathcal{HG}_{\mathrm{HSO\_SH}}$	The Structure Holding strategy
$\rho_{e_i}, \rho_{v_i}, \omega_{\mathrm{HD}}, \omega_{\mathrm{ND}}$	The Bernoulli distribution parameters
q_{CG}, q_{EH}, q_{SH}	The patient representations from three views
$\hat{o}_{dp}^{(t)}, \hat{o}_{mr}^{(t)}$	The clinical predictions probability of the patient i at visit t
$o_{dp}^{(t)}, o_{mr}^{(t)}$	The clinical predictions combination of the patient i at visit t

- In the medication code encoding part, each medication code has its chemical molecular graph. We initially construct a substructure-level graph based on the molecular structure, followed by a graph-based encoder with a learnable substructure embedding to convolve and aggregate the information of substructures. The connections between the substructures and medications are further encoded into the medication embeddings.

4.1.1. Diagnosis and procedure code encoding part

In this part, we learn diagnosis and procedure code embeddings d_e and p_e via medical code descriptions and ontology structure. In this section, we briefly use c_e for the medical code embeddings. The EHR of the patient is structured according to the International Classification of Diseases (ICD) coding system. The used ICD coding system is a hierarchical ontology, with leaf nodes representing the ICD codes for diagnosis or procedure information, and ancestor nodes representing medical hierarchies with medical-specific taxonomic significance. Each code in the ICD coding system has a formal medical description that can be obtained from the ICD Data website.[2] In the ICD coding system, different codes may describe medical code information with similar meanings. So we learn the semantic information and hierarchical information according to the formal description and ontology structure of different codes. By utilizing the semantic information of these codes, as well as the hierarchical information in the ontology structure, a comprehensive medical code embedding can be learned.

We first learn semantic embeddings of medical codes from related descriptions via a BERT-based encoder. Then, according to the structure of the ICD coding system, we design a message-passing mechanism with information flow (aggregate the information of child nodes to the parent node, and update the child nodes through the information of the parent node) to fuse the semantic features of the code and the ontology graph hierarchical features and obtain the diagnosis and procedure embedding.

(1) BERT-based Semantic Encoder

In order to obtain the semantic embeddings from medical codes descriptions, we employ a fine-tuned Bio-Clinical BERT [42] encoder, and apply pooling to obtain low-dimensional semantic feature for each medical code. The Bio-Clinical BERT model was initialized from BioBERT [43] and trained on all MIMIC notes [21]. Fig. 3.I.(a) shows the whole semantic encoding process:

- **Preprocessing**: For each medical code, its description text consists of a sequence of words. The input to each sequence starts with a special token denoted [CLS], and the corresponding vector of the token in the last layer contains the semantic representation of the entire description information.
- **Word embedding**: A sequence of tokens from medical codes is used as input through an MLP layer and further input to a fine-tuned Bio-Clinical BERT model which contains multi-head attention layers, fully connected layers, and the output layer. The word embedding process can be formulated as:

$$\begin{aligned} h_i^0 &= W_{emb} w_i + b_{emb}, \\ h_i^{\ell} &= \mathrm{Transformer}(h_i^{\ell-1}), \end{aligned} \tag{1}$$

  where h_i^0 is the token embedding through the MLP layer, h_i^{ℓ} is the token hidden state embedding in ℓ_{th} layer. W_{emb}, w_i, and the Transformer block are pretrained on the large EHR corpora using several pretraining tasks (*e.g.* natural language inference task and named entity recognition task). We extract all the medical code and description as a corpus to fine-tune the Bio-Clinical BERT model.
- **Description embedding**: The description embedding can be obtained by utilizing the attention polling mechanism [44] for the vectors of all tokens from the last layer.

Through the preprocessing, word embedding, and description embedding steps, we utilize the Bio-Clinical BERT model to convert each medical code description into fixed-dimensional vectors, which serve as node features.

(2) Hierarchical Ontology Encoding

The ontology structure of diagnosis and procedure codes, which is constructed from the ICD coding system and the corresponding domain knowledge, reflects the hierarchy and dependencies between the various medical codes. When constructing the ontology structure in accordance with the ICD coding principle, the parent node can be considered as a summary of its child nodes, the child nodes can acquire the attributes of their parent nodes, thus enabling the nodes in the ontology structure to provide more precise information through these two steps [30]. In order to obtain an effective embedding of medical codes, we design a message-passing method to learn the representations of medical codes at each layer following the ontology structure. As shown in Fig. 3.I.(b), we split the embedding of each medical code c_e into two parts, the unique part, and the inherited part. The unique part c_e^{uni} contains local information of the code, which is used to distinguish it from its parent node. We adopt the semantic embedding of each medical code description to represent the unique part embedding.

[2] http://www.icd9data.com/

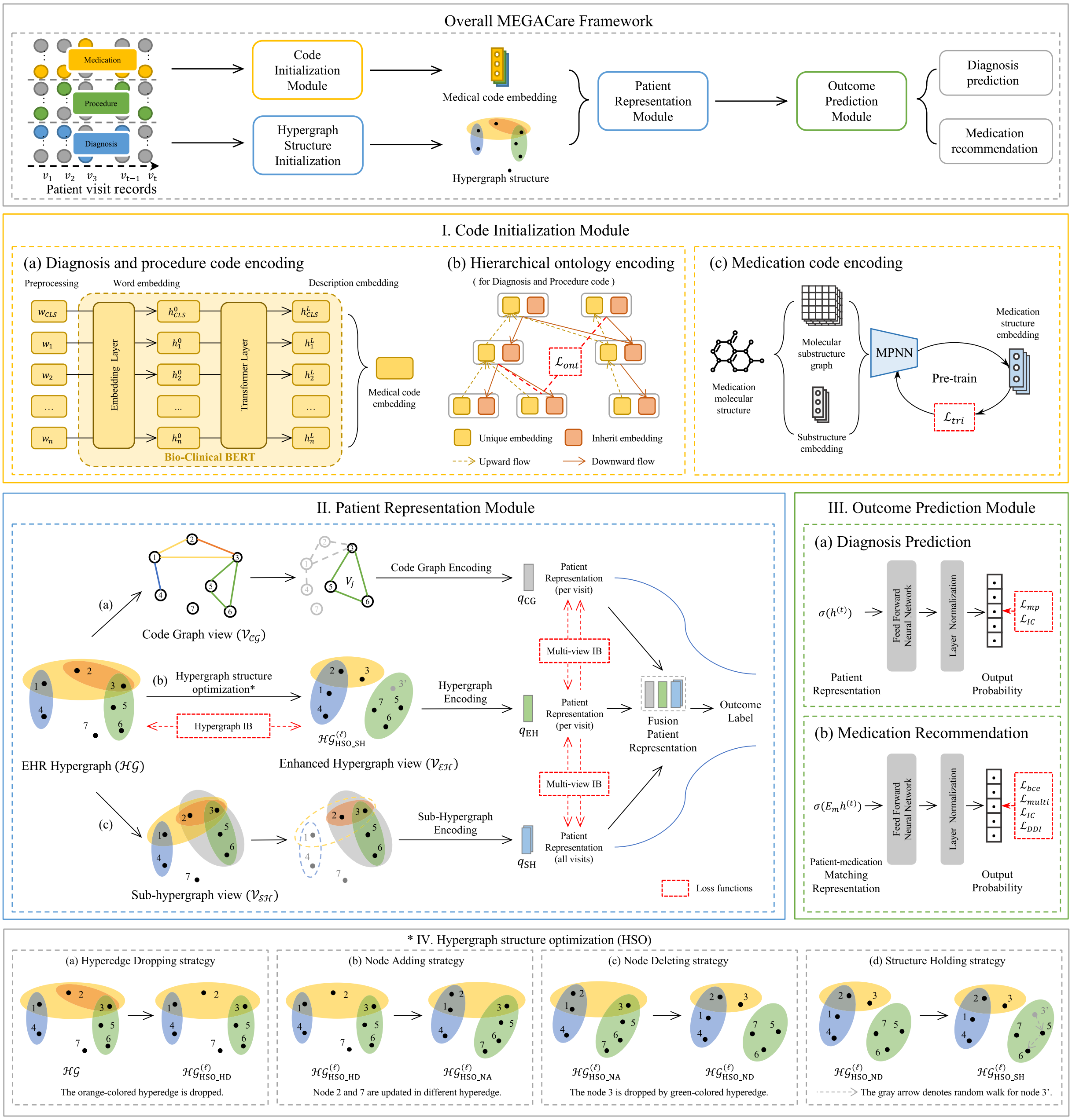

Overall MEGACare Framework
Medication
Procedure
Diagnosis
v_1 v_2 v_3 v_{t-1} v_t
Patient visit records
Code Initialization Module
Hypergraph Structure Initialization
Medical code embedding
Hypergraph structure
Patient Representation Module
Outcome Prediction Module
Diagnosis prediction
Medication recommendation
I. Code Initialization Module
(a) Diagnosis and procedure code encoding
Preprocessing
Word embedding
Description embedding
w_{CLS}
w_1
w_2
...
w_n
Embedding Layer
h^0_{CLS}
h^0_1
h^0_2
h^0_n
Transformer Layer
h^L_{CLS}
h^L_1
h^L_2
h^L_n
Bio-Clinical BERT
Medical code embedding
(b) Hierarchical ontology encoding
(for Diagnosis and Procedure code)
$\mathcal{L}_{ont}$
Unique embedding
Inherit embedding
Upward flow
Downward flow
(c) Medication code encoding
Medication molecular structure
Molecular substructure graph
Substructure embedding
MPNN
Pre-train
$\mathcal{L}_{tri}$
Medication structure embedding
II. Patient Representation Module
(a)
Code Graph view ($\mathcal{V}_{\mathcal{CG}}$)
V_j
Code Graph Encoding
q_{CG}
Patient Representation (per visit)
Multi-view IB
(b)
Hypergraph structure optimization*
Hypergraph IB
EHR Hypergraph ($\mathcal{HG}$)
$\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}$
Enhanced Hypergraph view ($\mathcal{V}_{\mathcal{EH}}$)
Hypergraph Encoding
q_{EH}
Patient Representation (per visit)
Fusion Patient Representation
Outcome Label
Multi-view IB
(c)
Sub-hypergraph view ($\mathcal{V}_{\mathcal{SH}}$)
Sub-Hypergraph Encoding
q_{SH}
Patient Representation (all visits)
Loss functions
III. Outcome Prediction Module
(a) Diagnosis Prediction
$\sigma(h^{(t)})$
Feed Forward Neural Network
Layer Normalization
$\mathcal{L}_{mp}$
$\mathcal{L}_{IC}$
Patient Representation
Output Probability
(b) Medication Recommendation
$\sigma(E_m h^{(t)})$
$\mathcal{L}_{bce}$
$\mathcal{L}_{multi}$
$\mathcal{L}_{IC}$
$\mathcal{L}_{DDI}$
Patient-medication Matching Representation
Output Probability
* IV. Hypergraph structure optimization (HSO)
(a) Hyperedge Dropping strategy
$\mathcal{HG}$
$\mathcal{HG}^{(\ell)}_{\text{HSO\_HD}}$
The orange-colored hyperedge is dropped.
(b) Node Adding strategy
$\mathcal{HG}^{(\ell)}_{\text{HSO\_HD}}$
$\mathcal{HG}^{(\ell)}_{\text{HSO\_NA}}$
Node 2 and 7 are updated in different hyperedge.
(c) Node Deleting strategy
$\mathcal{HG}^{(\ell)}_{\text{HSO\_NA}}$
$\mathcal{HG}^{(\ell)}_{\text{HSO\_ND}}$
The node 3 is dropped by green-colored hyperedge.
(d) Structure Holding strategy
$\mathcal{HG}^{(\ell)}_{\text{HSO\_ND}}$
$\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}$
The gray arrow denotes random walk for node 3'.

Fig. 3. The architecture of overall MEGACare framework. The input to the MEGACare is patient EHR data, and the output is task-related predictions. MEGACare consists of (I) a Code Initialization module, (II) a Patient Representation module, and (III) an Outcome Prediction module. The red dashed box represents the loss functions. The grey box at the bottom of the figure visualizes the four strategies used in the hypergraph structure optimization (HSO) stage.

The inherited part c_e^{inh}, an initialized trainable embedding vector, is intended to represent information inherited from its parent. The initialized code embedding c_e is formulated by combining the inherited embedding and the unique embedding with a trainable coefficient λ_c:

$$c_e = \lambda_c c_e^{inh} + (1 - \lambda_c) c_e^{uni}. \tag{2}$$

In order to learn the information transfer process in the hierarchical structure of medical codes, we utilize a two-direction hierarchical embedding method that uses the inherited and unique embedding vector of each medical code for layer-by-layer information transferring to learn the complete code representation. Fig. 3.I.(b) shows the information transferring process in two directions, which is:

- **Upward flow** (The process of the parent nodes aggregating from the child nodes): The unique part of a parent node c_i consists of the unique parts of all its children, which can be formulated as:

$$c_{e_i}^{uni} = F_{\text{Agg}}(c_i, ch(c_i), W_{uni}), \tag{3}$$

where $F_{\text{Agg}}(\cdot,\cdot,\cdot)$ is a node embedding aggregation function, which accepts $c_i, ch(c_i), W_{uni}$ as input. c_i is the target medical code. $c_{e_i}^{uni}$ is the unique part of code c_i. $ch(c_i)$ represents for the child node set to code c_i. W_{uni} is the corresponding embedding matrix.

- **Downward flow** (The process of the child nodes inheriting from the parent node): The inherited part of the child node c_j consists of the complete code embeddings of all its parents, which can be formulated as:

$$c_{e_j}^{inh} = F_{\text{Agg}}(c_j, pa(c_j), W_{inh}), \tag{4}$$

where $F_{\text{Agg}}(\cdot,\cdot,\cdot)$ is the same as above. $c_{e_j}^{inh}$ is the inherit part of target code c_j. $pa(c_j)$ represents for the parent node set to code c_j. W_{inh} is the corresponding embedding matrix.

The node embedding aggregation function $F_{\text{Agg}}(\cdot,\cdot,\cdot)$ is capable of transmitting and merging information from both the target node and its immediate children nodes (or parent node) that are directly connected to the target node. This process results in a more correlated embedding representation of the target node with the embedding representation of the child (or parent) nodes. Here, we choose Graph Attention Network (GAT) [45] as our aggregation function, which has been demonstrated to be capable of effectively learning medical embeddings on graph-structured tasks.

Through the above steps, we obtain the final diagnosis embedding $X_d \in \mathbb{R}^{|D|\times dim}$ and procedure embedding $X_p \in \mathbb{R}^{|P|\times dim}$ by concatenating all single diagnosis and procedure embedding together respectively, which can be formulated as:

$$X_d = \text{concat}([d^{(1)}, d^{(2)}, \dots, d^{(|D|)}]), \tag{5}$$

$$X_p = \text{concat}([p^{(1)}, p^{(2)}, \dots, p^{(|P|)}]). \tag{6}$$

4.1.2. Medication code encoding part

Considering the functionalities of medications are mainly reflected by molecule substructures (i.e., functional groups), learning medication representations from medical descriptions or atom–atom graphs may lose or underutilize substructure-level correlations with more predictive signals for accurately matching the medication and patients' health states. Therefore, MEGACare learns medication representations with hierarchical information via substructure-level correlations, rather than by encoding the atom–atom graph or the ATC classification ontology as studies [18,46].

In the medication code encoding part, each medication code has an independent molecular structure, which is available from the DrugBank website.[3] Firstly, We construct the substructure–substructure graph for each medication, based on its molecular structure. Then, we design a graph-based encoder (*i.e.*, MPNN) with a learnable substructure embedding to convolve and aggregate the information of the substructures across the corresponding molecular graph. Further, the correlations between the substructures and medications can be encoded into the output medication embeddings.

(1) Substructure Graph Construction for Medication:

Initially, the SMILES protocol [47] is employed to convert the molecular structures of medications into their corresponding string representations. Subsequently, the BRICS molecule segmentation method [48] is utilized to segment each SMILES string into a collection of functional substructures. Given $|M|$ medications ($|M|$ is the medication set; $|\cdot|$ denotes the length of a set), we can obtain a total of $|Sub|$ unique substructures, where $Sub = \{sub_1, sub_2, \dots, sub_{|Sub|}\}$, $Sub^{(k)} \subseteq Sub$, and $k \in \{1,\dots,|M|\}$. Drawing upon this approach, every medication can be depicted as a substructure graph, where the nodes consist of a subset of Sub, while the edges correspond to the bonds linking the substructures. We construct an adjacency matrix $A^{(m)} \in \{0,1\}^{|Sub|\times|Sub|}$ for each medication: $A^{(k)}_{i,j|i\neq j} = 1$ if substructures sub_i and sub_j are the nodes of the kth medication's substructure graph and there exists a connection bond, otherwise, $A^{(k)}_{i,j|i\neq j} = 0$.

(2) Message-Passing Neural Network (MPNN):

Given the initial learnable embedding table for substructures $\mathrm{E}_s \in \mathbb{R}^{|Sub|\times dim}$ and the adjacency matrix $A^{(m)}$ of a medication's substructure graph, we perform two stage message passing over the graph to encode the aforementioned components.

$$\varphi_i^{(\ell+1)} = \sum_{j:A_{ij}\neq 0} \mathrm{M}_\ell(g_i^{(\ell)}, g_j^{(\ell)}, W_{\text{MPNN}}^{(\ell)}), \tag{7}$$

$$g_i^{(\ell+1)} = \mathrm{U}_\ell(g_i^{(\ell)}, \varphi_i^{(\ell+1)}), i = 0, \dots, n, \tag{8}$$

where in ℓ_{th} layer, $\varphi_i^{(\ell+1)}$ is the encoded message from the neighbors of $g_i^{(\ell)}$ using the message-passing function M_ℓ, $g_i^{(\ell)}|_{\ell=0}$ is the ith row in E_s that is randomly initialized, $g_i^{(\ell+1)}$ is the updated state using the node update function U_ℓ, and $W_{\text{MPNN}}^{(\ell)} \in \mathbb{R}^{dim\times dim}$ is the layer-wise parameter matrix.

Upon the completion of L rounds of message passing, the state of each substructure $g_i^{(L)}$ is updated by its neighboring substructure nodes within the same medication. In order to derive the medication embeddings, a straightforward readout function $\text{avg}(\cdot)$ is employed to compute the average of all substructure states belonging to a particular medication:

$$m^{(k)} = \text{avg}(\sum_{i\in S^{(k)}} g_i^{(L)}), \tag{9}$$

where $S^{(k)} \subseteq S$ is the substructure set of the kth medication and $m^{(k)} \in \mathbb{R}^{dim}$ is the medication embedding.

The medication embedding matrix $X_m \in \mathbb{R}^{|M|\times dim}$ can be generated by concatenating all individual medication embeddings together as:

$$X_m = \text{concat}([\mathrm{m}^{(1)}, \mathrm{m}^{(2)}, \dots, \mathrm{m}^{(|\mathrm{M}|)}]). \tag{10}$$

4.2. Patient Representation Module

The intricacy of EHRs is derived from the multiple visit data of different patients. Generally, each patient visit is regarded as an edge and each medical code is regarded as a node. There are numerous medical codes in each visit and multiple visits form a complete graph for a patient. Existing GNN-based methods are restricted in their capacity to represent complex relationships that involve many nodes connected by one edge in a graph. To solve this problem, the proposed MEGACare conceptualizes the EHR data as a hypergraph, where each patient visit is regarded as a hyperedge, and each medical code is regarded as a node.

In the Patient Representation Module, a hypergraph is initially constructed between patient visits and medical codes. MEGACare employs codes of diagnosis, and procedure to form an EHR Hypergraph. Subsequently, three distinct views of the constructed Hypergraph are designed to improve hypergraph modeling, namely Code Graph View, Enhanced Hypergraph View, and Sub-hypergraph View. These three views are employed in a synergistic manner to generate the learned patient representation, followed by a multi-view representation fusion process.

4.2.1. Definition of EHR hypergraph

MEGACare initially constructs an EHR Hypergraph $\mathcal{HG} = (\mathcal{V}, \mathcal{E})$ with all inputs. As depicted in Fig. 3.II, $\mathcal{V} = \{v_i\}_{i=1}^{|C|}$ is the set of nodes in the hypergraph which represent the utilized diagnosis and procedure codes in EHRs, where $|C|$ is the total number of utilized unique medical codes (medication codes are not employed in the construction of EHR Hypergraph). The representation of each node is the embeddings of diagnosis code X_d and procedure code X_p learned by the code initialization module. In the following sections, the medical code embedding of each node in the hypergraph is represented by $\mathcal{V}$. Specifically, $\mathcal{V}$ equals X_d in the diagnosis prediction task and the combination of X_d and X_p in the medication recommendation task. $\mathcal{E} = \{e_i^{(t)}\}_{i=1,t=1}^{i=\mathrm{N},t=\mathrm{T}_i}$ is the hyperedge set which represents the patient visits, where $i \in \{1, 2, \cdots, \mathrm{N}\}$, N is the total number of all patients, and T_i denotes the total visit times of patient i. Hyperedges are a general and flexible way of encoding high-order interactions, as they can represent an edge with an unlimited number of nodes.

In the proposed hypergraph structure, we assign different weights to nodes, hyperedges, and hyperedge-node correlations. The assigned node weights are represented by an incidence matrix HG, which signify the importance of different nodes in the same hyperedges. Fig. 3.II illustrates a schematic graph of the constructed EHR hypergraph, where nodes are represented by medical codes and hyperedges are represented

[3] https://go.drugbank.com/

by patient visits. While each visit can contain multiple medical codes, medical codes can exist in multiple visits. Thus, we can have multiple hyperedges on the same node. As shown in Fig. 3.II, nodes 1, 2, and 3 exist in multiple visits.

4.2.2. Code graph view

The code graph view aims to encode the fundamental graph information. As shown in Fig. 3.II.(a), we simplify hypergraph $\mathcal{HG}$ into code graph view V_{CG} with general graph structure by transforming each hyperedge into a sub-graph. Each sub-graph corresponds to a visit e_j. For example, the hyperedge denoted as the green ellipse is converted into the sub-graph with the green-colored edge. The converted sub-graph still contains nodes 3, 5, and 6. Since an edge in a general graph can only be connected to two nodes, the view V_{CG} only considers pairwise information. Then, we use rule-based search to convert each sub-graph in V_{CG} into a medical code sequence, which can be denoted as $e_j = \{c_1, c_2, \dots, c_i\}$, where $j \in \{1, 2, \dots, T_i\}$. Next, e_j is fed into Transformers [24] with L_{CG} layers to be encoded as $e'_j = \{c'_1, c'_2, \dots, c'_i\}$.

$$e'_j = \text{Transformers}(\{c_1, c_2, \dots, c_i\}), \tag{11}$$

The node features in the last layer of used Transformers are aggregated by an attention-based function to form patient-level embedding q_{CG} as:

$$q_{\text{CG}} = \sum_{c'_i \in e'_j} \frac{\exp(w_{CG} c'_i)}{\sum_{c'_k \in e'_j} \exp(w_{CG} c'_k)} c'_i, \tag{12}$$

where w_{CG} is the trainable parameters in the attention-based function.

4.2.3. Enhanced hypergraph view

Graph-based neural networks typically rely on graph adjacency matrices to acquire information. However, real-world graph data may contain a substantial amount of irrelevant information [49], which can lead to an accumulation of noise as the number of network layers increases. The enhanced hypergraph view $V_{\mathcal{EH}}$ is designed for alleviating the influences of noise in MEGACare.

MEGACare utilizes L_{EH} layers in constructing V_{EH}. As shown in Fig. 3.II.(b), each layer consists of a hypergraph structure optimization (HSO) step and a hypergraph encoding (HE) step. HSO aims to update used hypergraph structures through the proposed hyperedge dropping, node adding, node deleting, and structure holding strategies. HE is employed to obtain the hyperedge and node representations of enhanced hypergraph view. HSO consists of hyperedge optimization and node optimization, while HE involves a two-stage embedding updating. HSO and HE in the ℓ_{th} layer ($\ell \in \{1, \dots, L_{\text{EH}}\}$) are formulated as:

$$\mathcal{HG}^{(\ell)}_{\text{HSO}} = F_{\text{HSO}}(v^{(\ell-1)}, \mathcal{HG}^{(\ell-1)}_{\text{HSO}}, \mathcal{HG}^{(0)}), \tag{13}$$

$$e^{(\ell)} = F^{\text{hyperedge}}_{\text{HE}}(\mathcal{HG}^{(\ell)}_{\text{HSO}}, \mathcal{V}), \tag{14}$$

$$v^{(\ell)} = F^{\text{node}}_{\text{HE}}(\mathcal{HG}^{(\ell)}_{\text{HSO}}, \mathcal{V}), \tag{15}$$

where $v^{(\ell)}, e^{(\ell)}$ are the node embedding and hyperedge embedding of the ℓ_{th} layer, respectively. HSO and HE in each layer share the same parameters.

(1) Hypergraph Structure Optimization (HSO)

Considering hypergraph structures have distinct influences for learned representations, MEGACare designs the hyperedge dropping, node adding, node deleting, and structure holding strategies for optimizing the structure of the entire EHR hypergraph.

Hyperedge Dropping Strategy is responsible for filtering out noisy hyperedges. It should notice that nodes are retained if the related noisy hyperedge is filtered. MEGACare first implements a learnable hyperedge dropping component parameterized with $\omega^{(\ell)}_{\text{HD}}$ at each layer ℓ. Then, Gumbel softmax [50] is used to generate a mask to determine whether the hyperedge should be kept. This process is formulated as:

$$\rho^{(\ell)}_{e_i} = \text{Gumbel}(\omega^{(\ell)}_{\text{HD}} \cdot e^{(\ell-1)}_i), \tag{16}$$

where $e^{(\ell-1)}_i$ is the hyperedge embedding in $(\ell-1)_{th}$ layer. $\omega^{(\ell)}_{\text{HD}}$ is the parameters of hyperedge dropping component. $\rho^{(\ell)}_{e_i}$ is the mask for hyperedge e_i in ℓ_{th} layer. Then, we can obtain the updated EHR hypergraph in ℓ_{th} layer, which is formulated as:

$$\mathcal{HG}^{(\ell)}_{\text{HSO\_HD}} = (\mathcal{V}^{(\ell-1)}, \{e_i \odot \rho^{(\ell)}_{e_i}\}), \tag{17}$$

where $\rho^{(\ell)}_{ei} \sim \text{Bern}\,(0, 1)$ and Bern denotes the Bernoulli distribution.

Node Adding Strategy is used to mine and add potential nodes into hyperedges. MEGACare attempts to establish the implicit connection between nodes and hyperedges by calculating similarity. Especially, for the disconnected node v_i and hyperedge e_j, MEGACare calculates a cosine similarity between them after multi-head attention weighted [51], which is formulated as follows:

$$S_{ij} = \frac{1}{n_c} \sum_{k=1}^{n_c} \cos(w_k \odot v_i, w_k \odot e_j), \tag{18}$$

where n_c is the numbers of heads of multi-head attention, w_k is a weight generated in k_{th} attention head. The pair of node v_i and hyperedge e_j are updated into the EHR Hypergraph if their similarity surpasses a manually specified threshold. This step converts $\mathcal{HG}^{(\ell)}_{\text{HSO\_HD}}$ into $\mathcal{HG}^{(\ell)}_{\text{HSO\_NA}}$.

Node Deleting Strategy tries to delete improper nodes from existing hyperedges after mining potential connections between nodes and hyperedges. The hyperedge dropping strategy serves to eliminate superfluous hyperedges, which can be regarded as coarse-grained optimization. The node adding and deleting strategy carries out the update process of node-hyperedges at a fine-grained level, by adding potential nodes to hyperedges and removing irrelevant nodes from hyperedges. Similarly to the hyperedge dropping strategy, MEGACare utilizes a learnable neural component with Gumbel softmax to decide which node should be removed from a hyperedge, which is formulated as follows:

$$\rho^{(\ell)}_{v_i} = \text{Gumbel}(\omega^{(\ell)}_{\text{NU}} \cdot v^{(\ell-1)}_i), \tag{19}$$

$$\mathcal{HG}^{(\ell)}_{\text{HSO\_ND}} = (\{v_i \odot \rho^{(\ell)}_{v_i}\}, \mathcal{E}^{(\ell)-1}), \tag{20}$$

where $v^{(\ell-1)}_i$ is the node embedding in $(\ell-1)_{th}$ layer, $\omega^{(\ell)}_{\text{NU}}$ is the parameters of learnable neural component, and $\rho^{(\ell)}_{v_i}$ is the mask for $v^{(\ell-1)}_i$, $\rho^{(\ell)}_{v_i} \sim \text{Bern}\,(0, 1)$.

Structure Holding Strategy aims to preserve the integrity of the EHR Hypergraph structure after deleting nodes from hyperedges. Considering deleting nodes may lead to a drastic alteration in the EHR hypergraph structure and further affecting information aggregation, MEGACare is designed to insert a virtual node in the same position when a node is deleted. In particular, for the deleted node v_i, we perform random walk on its neighbor nodes with k steps, and the sampled nodes are pooled as the representation of the added virtual node for v_i. This step converts $\mathcal{HG}^{(\ell)}_{\text{HSO\_ND}}$ into $\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}$.

(2) Hypergraph Encoding (HE)

After HSO, MEGACare follows a spatial hypergraph convolutional layer [52] to encode the optimized EHR hypergraph. For each layer $\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}$ in the optimized EHR hypergraph, $F^{\text{hyperedge}}_{\text{HE}}$ and $F^{\text{node}}_{\text{HE}}$ are utilized to encode hyperedge embeddings $e^{(\ell)}$ and node embeddings $v^{(\ell)}$, respectively.

$$\begin{aligned} e^{(\ell)} &= F^{\text{hyperedge}}_{\text{HE}}(\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}, \mathcal{V}) \\ &= D_e(\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}})^{-1} \mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}{}^{\top} \mathcal{V} \Theta_e, \end{aligned} \tag{21}$$

$$\begin{aligned} v^{(\ell)} &= F^{\text{node}}_{\text{HE}}(\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}, \mathcal{V}) \\ &= D_v(\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}})^{-1} \mathcal{HG}^{(\ell)}_{\text{HSO\_SH}} D_e(\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}})^{-1} \mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}{}^{\top} \mathcal{V} \Theta_v, \end{aligned} \tag{22}$$

where Θ_e and Θ_v are the trainable parameters, $\mathcal{V}$ keeps the same with Eq. (15), which is learned by the code initialization module, D_v and D_e denote the diagonal matrices whose diagonal items are the node degree and hyperedge degree, respectively. In the first stage of HE, the node embeddings are transformed into the hyperedge embeddings by the incidence matrix $\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}{}^{\top}$. Secondly, the hyperedge embeddings are further transformed into updated node embeddings in accordance with the incidence matrix $\mathcal{HG}^{(\ell)}_{\text{HSO\_SH}}$.

By employing the aforementioned steps, MEGACare obtains the updated hyperedge embeddings and node embeddings in each layer of the EHR hypergraph. Finally, we use the updated representations of nodes in the last layer $v^{(L_{\text{EH}})}$ to acquire the enhanced patient embedding. Each hyperedge corresponds to a visit, and each visit corresponds to a patient embedding q_{EH}:

$$\begin{aligned} q_{\text{EH}} &= F_{\text{HE}}^{\text{hyperedge}}(\mathcal{HG}^{(L_{\text{EH}})}_{\text{HSO}}, v^{(L_{\text{EH}})}) \\ &= D_e(\mathcal{HG}^{(L_{\text{EH}})}_{\text{HSO}})^{-1}\mathcal{HG}^{(L_{\text{EH}})\top}_{\text{HSO}} v^{(L_{\text{EH}})}\Theta_e, \end{aligned} \tag{23}$$

4.2.4. Sub-hypergraph view

The above code graph view and enhanced hypergraph view focus on encoding patient information limited in the current visit record. However, insights gleaned from prior visits may also be of vital importance in assessing a patient's overall health condition. To address this concern, MEGACare introduces the concept of a sub-hypergraph view that integrates information from both current and prior visits. This approach offers a more comprehensive patient representation, encompassing data from multiple visits. For constructing sub-hypergraph view V_{SH}, MEGACare first updates the HG, and then encodes each sub-hypergraph into a patient representation (we divide $\mathcal{HG}$ into sub-hypergraphs by patients).

(1) Sub-Hypergraph Updating

In the sub-hypergraph view, MEGACare first updates hyperedge $e_j \in \mathcal{E}$ and nodes $v_i \in \mathcal{V}$ with a L_{SH} layers attention. In the first layer, we can obtain $v_i^{(0)}$ from the Code Initialization Module. Each $e_j^{(0)}$ is initialized by averaging all $v_i^{(0)}$ which belong to $e_j^{(0)}$. Then, in each $\ell \in L_{SH}$, we calculates attention score $a_E(e_j^{(\ell)}, v_i^{(\ell)})$ as:

$$s(e_j^{(\ell-1)}, v_i^{(\ell)}) = (W_N v_i^{(\ell)} + b_N) \odot (W_E e_j^{(\ell-1)} + b_E), \tag{24}$$

$$a_E(e_j^{(\ell-1)}, v_i^{(\ell)}) = \frac{\exp(s(e_j^{(\ell-1)}, v_i^{(\ell)}))}{\sum_{v_{i'} \in e_j} \exp(s(e_j^{(\ell-1)}, v_{i'}^{(\ell)}))}, \tag{25}$$

where W_N, b_N, W_E, and b_E are trainable parameters. Next, we calculate the hyperedge's representation in layer ℓ,

$$e_j^{(\ell)} = \sum_{v_i \in e_j} a_E(e_j^{(\ell-1)}, v_i^{(\ell)}) v_i^{(\ell)}. \tag{26}$$

After updating a hyperedge from $e_j^{(\ell-1)}$ to $e_j^{(\ell)}$, MEGACare updates $v_i^{(\ell)}$ to $v_i^{(\ell+1)}$ follows:

$$s(v_i^{(\ell)}, e_j^{(\ell)}) = (W_E e_j^{(\ell)} + b_E) \odot (W_N v_i^{(\ell)} + b_N), \tag{27}$$

$$a_V(v_i^{(\ell)}, e_j^{(\ell)}) = \frac{\exp(s(v_i^{(\ell)}, e_j^{(\ell)}))}{\sum_{e_{j'} \ni v_i} \exp(s(v_i^{(\ell)}, e_{i'}^{(\ell)}))}, \tag{28}$$

where W_N, b_N, W_E, and b_E are trainable parameters. Next, we calculate the hyperedge's representation in layer $\ell + 1$,

$$v_i^{(\ell+1)} = \sum_{e_j \ni v_i} a_V(v_i^{(\ell)}, e_j^{(\ell)}) e_j^{(\ell)}. \tag{29}$$

Due to the superiority of hypergraph, with an updated hyperedge $e_j^{(\ell)}$, MEGACare can update $v_i^{(\ell)}$ to $v_i^{(\ell+1)}$ with more related nodes to obtain more robust representations.

(2) Sub-hypergraph Encoding

After updating sub-hypergraphs in L_{SH} layers, MEGACare encodes each sub-hypergraph with the weighted sub-graph attention mechanism [53]. We compute the sub-hypergraph attention over node and hyperedge as follows:

$$a(\mathcal{G}_j, v_i^{(L_{\text{SH}})}) = \frac{\exp({w_v}^T v_i^{(L_{\text{SH}})})}{\sum_{v_{i'} \in \mathcal{G}_j} \exp({w_v}^T v_{i'}^{(L_{\text{SH}})})}, \tag{30}$$

$$a(\mathcal{G}_j, e_i^{(L_{\text{SH}})}) = \frac{\exp({w_e}^T e_i^{(L_{\text{SH}})})}{\sum_{e_{i'} \in \mathcal{G}_j} \exp({w_e}^T e_{i'}^{(L_{\text{SH}})})}, \tag{31}$$

where w_v and w_e are the learnable vectors, respectively. With these two sub-hypergraph level attentions, we compute patient representation q_{SH} by nodes and hyperedges of each sub-hypergraph in the last layer L_{SH}:

$$\mathcal{G}_{j_v} = \sum_{v_i \in \mathcal{G}_j} a(\mathcal{G}_j, v_i^{(L_{\text{SH}})}) v_i^{(L_{\text{SH}})}, \tag{32}$$

$$\mathcal{G}_{j_e} = \sum_{e_i \in \mathcal{G}_j} a(\mathcal{G}_j, e_i^{(L_{\text{SH}})}) e_i^{(L_{\text{SH}})}, \tag{33}$$

$$q_{\text{SH}} = \text{concat}[\mathcal{G}_{j_v}, \mathcal{G}_{j_e}]. \tag{34}$$

4.3. Outcome Prediction Module

Through Patient Representation Module, we have learned patient representations in three views, which are the code graph patient representation q_{CG}, the enhanced hypergraph patient representation q_{EH}, and the sub-hypergraph patient representation q_{SH}. We concatenate these three vectors together to get the final patient representation.

$$h^{(t)} = \text{concat}[q_{CG}, q_{EH}, q_{SH}]. \tag{35}$$

4.3.1. Diagnosis prediction task

MEGACare formulates the diagnosis prediction task as a multi-label classification task. After obtaining the patient representation $h^{(t)}$, MEGACare provides diagnosis prediction follows:

$$o_{dp}^{(t)} = \text{Softmax}(W_{dp} h^{(t)} + b_{dp}), \tag{36}$$

where W_{dp} and b_{dp} are learnable parameters. $o_{dp}^{(t)}$ denotes the final predicting scores for a patient. By comparing $o_{dp}^{(t)}$ with a pre-defined threshold parameter δ_{dp}, we can obtain the final diagnosis prediction $\hat{o}_{dp}^{(t)} \in \mathbb{R}^{|D|}$ for the diagnosis prediction task.

4.3.2. Medication recommendation task

Different from the diagnosis prediction task limited in patient information, the medication recommendation task requires correlations between patient and medication to further predict the proper medications for given patients.

After obtaining the patient representation $h^{(t)}$ and medication embedding matrix $X_m \in \mathcal{R}^{|M| \times dim}$, we need to find out the most relevant medications to the patient historical health states. Following previous work [18], we use a patient-medication matching function:

$$\Omega^{(t)} = \text{sigmoid}(X_m h^{(t)}), \tag{37}$$

where $\Omega^{(t)} \in \mathbb{R}^{|M|}$ consists of the matching scores of $|M|$ medications to the patient representation. Then, an MLP layer and a skip connection are used to generate the final medications:

$$o_{mr}^{(t)} = \text{sigmoid}(W_{mr} \Omega^{(t)} + b_{mr}), \tag{38}$$

where W_{mr} and b_{mr} denotes the learnable parameters, $o_{mr}^{(t)}$ denotes the final matching scores for a patient. By comparing the matching scores $o_{mr}^{(t)}$ to a pre-defined threshold parameter δ_{mr}, we can obtain the final medication combinations $\hat{o}_{mr}^{(t)} \in \mathbb{R}^{|M|}$ recommended by MEGACare.

4.4. Loss functions

Our MEGACare Framework is trained with the combination of multiple losses: (1) the *Pretraining Loss* for constraining medical code embeddings during the pre-training stage in the code initialization module, (2) the *Multi-label Prediction Loss* for accurately predicting the medical code combinations, and (3) the *Information Constraint Loss* to enhance the robustness of the model by utilizing the hypergraph information bottleneck principle and multi-view information bottleneck principle.

4.4.1. Pretraining loss

(1) Pretrain loss for initialization of diagnosis and procedure code

In order to restrict the divergence of the obtained medical code embedding in Section 4.1.1, we hope that the distance between two linked nodes is smaller than the distance between non-connected nodes. We use the adjacency matrix A to represent the set of connected nodes in the ontology structure. The distance $d(c_{e_i}, c_{e_j})$ between the embedding vectors of two medical codes c_{e_i} and c_{e_j} of the utilized ontology loss function $\mathcal{L}_{ont}$ can be formulated as:

$$d(c_{e_i}, c_{e_j}) = \cosh^{-1}\left(1 + 2 \times \frac{\|c_{e_i}\| - \|c_{e_j}\|^2}{(1 - \|c_{e_i}\|^2)(1 - \|c_{e_j}\|^2)}\right), \tag{39}$$

$$\mathcal{L}_{ont} = -\sum_{(i,j)\in A} \log \frac{e^{-d(c_{e_i}, c_{e_j})}}{\sum_{l\in N(i)} e^{-d(c_{e_i}, c_{e_l})}}, \tag{40}$$

where $N(i)$ denotes the set of non-adjacent nodes for c_i, $d(\cdot,\cdot)$ is the hyperbolic distance [54] between two embeddings. The $\mathcal{L}_{ont}$ loss aims to minimize the distance between the representations of connected code nodes and maximize those of non-connected nodes. Through the pretraining process, we obtain the final diagnosis embedding X_d and procedure embedding X_p that was used in the EHR hypergraph formulation.

(2) Pretrain loss for initialization of medication code

In order to maximize the similarities between representations of synergistic medication pairs while minimizing those of antagonistic medication pairs in Section 4.1.2. We proposed the TL loss that aims to minimize the distance between representations of synergistic medication pairs (*e.g.*, $(m^{(i)}, m_s^{(j)})$) and maximize those of antagonistic medication pairs (*e.g.*, $(m^{(i)}, m_a^{(k)})$). Such a constraint would increase the probability of recommending synergistic medication combinations while decreasing antagonistic medications, which may improve the effectiveness and safety of the recommended medications.

To begin, we represent the knowledge of SDDI and ADDI through the use of two matrices, $\mathcal{M}^s$ and $\mathcal{M}^a$, both of which have dimensions $\mathbb{R}^{|M|\times|M|}$. Specifically, $\mathcal{M}^s_{i,j} = 1$ if the ith and jth medications form a synergistic pair, and $\mathcal{M}^s_{i,j} = 0$ otherwise. Similarly, $\mathcal{M}^a_{i,j} = 1$ if the two medications form an antagonistic pair, and $\mathcal{M}^a_{i,j} = 0$ otherwise. For each medication embedding $m^{(i)}$, we construct the triplet as $\langle m^{(i)}, m^{(j)}|_{\mathcal{M}^{(s)}_{i,j}=1}, m^{(k)}|_{\mathcal{M}^{(a)}_{i,k}=1}\rangle$, which can be represented in shorthand as $\langle m^{(i)}, m_s^{(j)}, m_a^{(k)}\rangle$. Then, we formulate the TL loss as:

$$\mathcal{L}_{tri} = \sum \max(0, d(m^{(i)}, m_s^{(j)}) - d(m^{(i)}, m_a^{(k)}) + \theta), \tag{41}$$

where θ is a margin. In order to obtain the final medication embedding X_m for use in the medication recommendation task, we employ backpropagation to pretrain the medication representations using the $\mathcal{L}_{tri}$ loss function.

4.4.2. Multi-label prediction loss

We consider both the disease prediction and medication recommendation tasks as multi-label binary classification tasks, and we use two common multi-label loss functions. We use c_i to denote the predict medical code label in diagnosis prediction task $c_i = d_i$ and medication recommendation task $c_i = m_i$. The first one is Multi-Label Margin (MLM) loss [55], which ensures the predicted probability of ground truth labels has at least 1 margin larger than others, which can be mathematically described as:

$$\mathcal{L}_{multi} = \sum_{i,j: c_i^{(t)}=1, c_j^{(t)}=0} \frac{\max(0, 1 - (o_i^{(t)} - o_j^{(t)}))}{|C|}. \tag{42}$$

The second one is the Binary Cross-Entropy (BCE) loss, which can be formulated as:

$$\mathcal{L}_{bce} = -\sum_{i=1}^{|C|} [c_i{}^{(t)} \log(o_i{}^{(t)}) + (1 - c_i{}^{(t)}) \log(1 - o_i{}^{(t)})]. \tag{43}$$

The multi-label prediction loss is formulated by combining the MLM loss and BCE loss with a balance hyper-parameter μ_{multi}:

$$\mathcal{L}_{mp} = \mu_{multi}\mathcal{L}_{multi} + (1 - \mu_{multi})\mathcal{L}_{bce}. \tag{44}$$

4.4.3. Information constraint loss

In order to obtain a concise and comprehensive patient representation, we extend the information bottleneck principle to the medical outcome prediction task. The initial input of our framework is a hypergraph structure composed of real-world medical codes, which may contain erroneous and absent connections and is vulnerable to task-irrelevant connections. To address this, we devise multiple views based on the original hypergraph structure, which eliminates extraneous and noisy information and acquires precise joint representations.

In order to achieve a balance between model expressiveness and robustness, two loss functions are designed based on the information bottleneck (IB) principle. The first loss function optimizes the hypergraph structure, reduces superfluous information from the original hypergraph structure, and captures the most essential information for downstream prediction tasks. The second loss function maximizes the mutual information between the labels and the learned joint representation, while simultaneously minimizing the mutual information between the learned latent representation and the original data representation for each view, in order to fuse knowledge from multiple views and improve predictive performance.

(1) Hypergraph Structure IB

In Enhanced Hypergraph View, the original hypergraph structure is optimized in each iteration and forms the hypergraph structure optimization flow, the overall objective of hypergraph structure IB ($\mathrm{HS_{IB}}$) can be formulated as:

$$\mathcal{L}^{(\ell)}_{\mathrm{HS_{IB}}}(\mathcal{HG}^{(0)}; Y; q_{EH}) = -\mathcal{I}(Y; q_{EH}) + \beta\mathcal{I}(\mathcal{HG}^{(0)}; q_{EH}), \tag{45}$$

where β is the trade-off hyperparameter to balance the weights of two items. Since the patient representation depends only on $\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}}$, according to data-processing inequality, we have:

$$\mathcal{I}(\mathcal{HG}^{(0)}; q_{EH}) \le \mathcal{I}(\mathcal{HG}^{(0)}; \mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}}). \tag{46}$$

Thus, combining the Eq. (46) with Eq. (45), we obtain an upper bound as the objective function to minimize:

$$\mathcal{L}^{(\ell)}_{\mathrm{HS_{IB}}} = -\mathcal{I}(Y; q_{EH}) + \beta\mathcal{I}(\mathcal{HG}^{(0)}; \mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}}). \tag{47}$$

For the first term $\min -\mathcal{I}(Y; q_{EH})$, it can be approximated as the cross-entropy loss $\mathcal{L}_{bce}$. The second term $\mathcal{I}(\mathcal{HG}^{(0)}; \mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}})$ measures the mutual information between the initial hypergraph structure and optimized hypergraph structure. We follow the VIB inference process [56] to specify the upper bound as:

$$\mathcal{I}(\mathcal{HG}^{(0)}; \mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}}) \le D_{KL}(\mathbb{P}(\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}} \mid \mathcal{HG}^{(0)}) || \mathbb{Q}(\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}})), \tag{48}$$

where D_{KL} is the Kullback Leibler divergence, $\mathbb{Q}(\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}})$ is a non-informative prior and the elements in $\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}}$ are $i.i.d$ Bernoulli distributions: $\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}} = \cup_{i,j}\{h_{ij} \in \{0,1\} \mid h_{ij} \overset{\mathrm{iid}}{\sim} .Bernoulli.(0.5)\}$. We assume that elements in $\mathbb{Q}(\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}})$ have a probability of 0.5 to be 1 or 0 because there is no prior information about whether node belongs to hyperedge or not. Thus, the estimation of $\mathcal{I}(\mathcal{HG}^{(0)}; \mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}})$ can be formulate as:

$$\mathcal{I}(\mathcal{HG}^{(0)}; \mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}}) \longrightarrow \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} D_{KL}(\mathrm{Bernoulli}(\mathcal{HG}^{(L_{\mathrm{EH}})}_{\mathrm{HSO}_{ij}})||\mathrm{Bernoulli}(0.5)). \tag{49}$$

To simplify the formulation of $\mathrm{HS_{IB}}$, we define the Eq. (49) as the $\mathrm{HS_{IB}}$ loss.

(2) Multi-view IB

In the patient representation fusion step, the patient representations from three views q_{CG}, q_{EH}, q_{SH} are aggregated into the final patient representation h. The overall objective of our multi-view IB ($\mathrm{MV}_{\mathrm{IB}}$) can be expressed as:

$$\mathcal{L}_{\mathrm{MV}_{\mathrm{IB}}}(X_i, Y; h) = -\mathcal{I}(Y; h) + \sum_{i=1}^{k} \beta_i \mathcal{I}(X_i; q_i), \tag{50}$$

where β_i refers to regularization parameter for the i_{th} view.

For the first term $\min -\mathcal{I}(Y; h)$, it can be replaced with the risk associated with h to the prediction performance on Y according to the cross-entropy loss $\mathcal{L}_{bce}$. The second term $\sum_{i=1}^{k} \beta_i \mathcal{I}(X_i; q_i)$ measures the mutual information between different view and patient representations. Here we use the matrix-based Rényi's α-entropy to estimate $\mathcal{I}_\alpha(X_i; q_i)$ in each view [57], which is given by:

$$\mathcal{I}_\alpha(X_i; q_i) = H_\alpha(\mathcal{A}_{x_i}) + H_\alpha(\mathcal{A}_{q_i}) - H_\alpha(\mathcal{A}_{x_i}, \mathcal{A}_{q_i}), \tag{51}$$

where $\mathcal{A}_*$ is the normalized version of K_*, i.e., $\mathcal{A}_* = \frac{\mathcal{K}_*}{\mathrm{tr}(\mathcal{K}_*)}$. The multi-view loss is obtained by adding up the losses for three views: $\mathcal{L}_{\mathrm{MV}_{\mathrm{IB}}} = \sum_{i=1}^{3} \mathcal{I}_\alpha(X_i; q_i)$.

The information constraint loss $\mathcal{L}_{IC}$ is formulated by combining the hypergraph structure loss and multi-view loss as:

$$\mathcal{L}_{\mathrm{IC}} = \mu_{\mathrm{hs}} \mathcal{L}_{\mathrm{HS}_{\mathrm{IB}}} + \mu_{\mathrm{mv}} \mathcal{L}_{\mathrm{MV}_{\mathrm{IB}}}, \tag{52}$$

where μ_{hg} and μ_{mv} are the weights for different loss functions.

4.4.4. Total loss function for diagnosis prediction

During the training process of the diagnosis prediction task, the total loss function $\mathcal{L}$ is obtained by combining the multi-label prediction loss and the information constraint loss through the weighted sum to optimize the neural network, it can be formulated as:

$$\mathcal{L}_{dp} = \mu_{\mathrm{mp}} \mathcal{L}_{mp} + (1 - \mu_{\mathrm{mp}}) \mathcal{L}_{\mathrm{IC}}, \tag{53}$$

where μ_{mp} balance the weights for different loss functions.

4.4.5. Comprehensive DDI controllable loss

For the medication prediction task, we introduce the Comprehensive DDI Controllable (CDC) loss, which is designed to explicitly constrain the rates of both SDDI and ADDI in the predicted medications. More precisely, our goal is to minimize the ADDI rate in order to reduce the occurrence of potential side effects resulting from ADDIs. Additionally, we strive to maximize the SDDI rate to enhance the efficacy of the medication. One approach to achieving our objective is to minimize the following loss function:

$$\begin{aligned} \mathcal{L}_s &= 1 - \sum_{i=1}^{|M|} \sum_{j=1}^{|M|} \mathcal{M}^s_{i,j} \cdot \hat{o}^{(t)}_i \cdot \hat{o}^{(t)}_j, \\ \mathcal{L}_a &= \sum_{i=1}^{|M|} \sum_{j=1}^{|M|} \mathcal{M}^a_{i,j} \cdot \hat{o}^{(t)}_i \cdot \hat{o}^{(t)}_j. \end{aligned} \tag{54}$$

Nevertheless, the ground true labels (*i.e.*, the real prescriptions within the MIMIC-III dataset) may contain a certain quantity of ADDIs. Recklessly reducing ADDIs could cause adverse effects and potentially harm prediction accuracy. Furthermore, blindly increasing SDDIs may result in the model recommending unnecessary synergistic medications, which could also impact prediction accuracy. To fulfill the aforementioned criteria, we introduce a novel CDC loss that enables the control of both SDDI and ADDI rates in medications. The SDDI and ADDI rates in a medication for a given visit can be calculated as follows:

$$\begin{aligned} \mathrm{DDI}^{(t)}_s &= \frac{\sum_{k,l \in \{i : \hat{\mathbf{o}}^{(t)}_i = 1\}} \mathbf{1}\{\mathcal{M}^s_{kl} = 1\}}{\sum_{k,l \in \{i : \hat{\mathbf{o}}^{(t)}_i = 1\}} 1}, \\ \mathrm{DDI}^{(t)}_a &= \frac{\sum_{k,l \in \{i : \hat{\mathbf{o}}^{(t)}_i = 1\}} \mathbf{1}\{\mathcal{M}^a_{kl} = 1\}}{\sum_{k,l \in \{i : \hat{\mathbf{o}}^{(t)}_i = 1\}} 1}. \end{aligned} \tag{55}$$

Table 2
Statistics of the MIMIC-III datasets for diagnosis prediction. # indicates the number of.

	MIMIC-III
# patients	7499
# visits	19911
avg. / max # of visits per patient	2.67/37
# of unique ICD-9 codes	4880
avg. / max # of ICD-9 codes per visit	13.06/39
# of category codes	171
avg. / max # of category codes per visit	10.16/30

Subsequently, we pre-define the target SDDI and ADDI rates, denoted by γ_s and γ_a, respectively. The differences between the predicted DDIs and the target DDIs can then be calculated as DDI margins $\psi_a and \psi_s$ using the following expressions:

$$\begin{aligned} \psi^{(t)}_s &= \max(0, \gamma_s - \mathrm{DDI}^{(t)}_s), \\ \psi^{(t)}_a &= \max(0, \mathrm{DDI}^{(t)}_a - \gamma_a). \end{aligned} \tag{56}$$

Now, we can formulate our CDC loss according to the margins $\psi^{(t)}_s$, $\psi^{(t)}_a$:

$$\mathcal{L}_c = \frac{\psi^{(t)}_s}{\psi^{(t)}_s + \psi^{(t)}_a} \mathcal{L}_s + \frac{\psi^{(t)}_a}{\psi^{(t)}_s + \psi^{(t)}_a} \mathcal{L}_a, \tag{57}$$

where the coefficients before $\mathcal{L}_s$ and $\mathcal{L}_a$ are adaptively adjusted in the training process, balancing the importance of $\mathcal{L}_s$ and $\mathcal{L}_a$ dynamically.

4.4.6. Total loss function for medication recommendation

In the medication recommendation task, we train the parameters of the patient representation module and outcome prediction module by minimizing a combined loss as:

$$\mathcal{L}_{mr} = \mu_c \mathcal{L}_{dp} + (1 - \mu_c) \mathcal{L}_c, \tag{58}$$

where μ_c is a balance hyper-parameter. For further balancing the prediction loss $\mathcal{L}_{dp}$ and the CDC loss $\mathcal{L}_c$ dynamically, we propose to adjust μ_c during the training process by the Proportional–Integral-Derivative (PID) controller [58]:

$$\lambda = \begin{cases} 1, & \mathrm{DDI}^{(t)}_s \geq \gamma_s, \mathrm{DDI}^{(t)}_a \leq \gamma_a \\ \max\{0, 1 - \frac{\psi^{(t)}_s + \psi^{(t)}_s}{K_p}\}, & others \end{cases}, \tag{59}$$

where K_p is the correcting factor for the proportional signal.

5. Experiment setup

5.1. Dataset

We conduct experiments on the publicly accessible MIMIC-III database [21] following the process protocol from the study [59]. To ensure the availability of historical health information, we only keep the patients with more than one visit in our dataset. Following the previous works [15,27], we divide MIMIC-III into two different datasets for predicting diseases and recommending medications.

Diagnosis Prediction Task: The used data consists of 7499 patients' medical records from the Intensive Care Unit (ICU). The goal of the disease prediction task is to predict the disease codes in the next visit. We use the nodes in the second hierarchy of ICD-9 codes[4] as the category labels. Table 2 lists the details about our Diagnosis Prediction Dataset.

Medication Recommendation Task: In this task, We employ the medication records for each patient within the first 24 h in the MIMIC-III database. The used diagnosis and procedure data are based on the

[4] http://www.icd9data.com/

Table 3
Statistics of the MIMIC-III datasets for medication recommendation. # indicates the number of.

	MIMIC-III
# patients	6,350
# clinical events	15,031
# diagnoses	1,958
# medications (ATC 3rd)	132
# procedures	1,430
avg. / max # of visits per patient	2.37/29
avg. / max # of diagnoses per visit	10.51/128
avg. / max # of procedures per visit	3.84/50
avg. / max # of medications per visit	11.18/64
# DDI types in the knowledge base	40
# medications in the DDI knowledge base	266

ICD-9 codes and the medication data is coded by the ATC 3rd level. ATC 3rd level contains 132 medications.[5] We extract DDI information of the top-40 most common types from DrugBank [60], where the medications are presented by DrugBank IDs and PubChem IDs.[6] To integrate the DDI data, we transform the two types of IDs to the ATC 3rd level. Then, we further group the DDIs into three categories (SDDI, ADDI, and no interaction) with the help of domain experts. Table 3 lists the details of our Medication Recommendation Dataset.

5.2. Baselines and evaluation metrics

(1) We compare MEGACare with the following eight baselines in **Diagnosis Prediction Task**.

- **CNN** [61] consists of three convolutional layers and an output layer to predict the probability of each class.
- **RNN** [23] generate the embedding with the Long Short-Term Memory (LSTM) layer and further are directly used to predict results by a linear classifier.
- **GCN** [28] is developed by Kiptf and Welling is considered to be one of the strongest baselines for graph convolutional networks.
- **RETAIN** [11] employs a two-level attention mechanism, which could enhance both the performance and interpretability of the model.
- **GRAM** [14] employs a medical knowledge graph associated with EHR data to learn the medical code representations via attention mechanisms and RNNs.
- **Dipole** [1] is an attention-based bidirectional recurrent neural network, and it takes the same raw input as GRAM. We select the local-based attention to obtain the final context vector.
- **KAME** [15] shares the framework with GRAM, we employ a supplementary branch that generates knowledge vectors, and then concatenate the output with the hidden vector which is generated by the GRU from GRAM before the last classification layer.
- **GNDP** [13] learns the spatial and temporal patterns from patients' sequential graphs, in which the domain knowledge is naturally infused.

For the diagnosis prediction task, MEGACare uses visit-level precision@k and code-level precision@k as evaluation measures.

- **Visit-level precision@k** measures the prediction precision of individual visits within patient sequences. The visit-level precision@k is defined as the number of correct medical codes among the ranked top k predictions divided by $\min(k, |\mathrm{o}_t|)$, where $|\mathrm{o}_t|$ is the number of category labels. We report the average visit-level precision@k of all visits.

  For a single visit, the final output of our framework is $\hat{o} = [\hat{o}_1, \hat{o}_2, \ldots, \hat{o}_l]$, and the ground truth label is $d = [d_1, d_2, \ldots, d_l]$, where $d_n \in \{0, 1\}$. The visit-level precision@k is defined as:

$$\text{visit-level precision@}k = \frac{|\hat{o}_{\text{correct}}|_k}{\min(k, |d_t|)}, \tag{60}$$

  where $|\hat{o}_{\text{correct}}|$ is denoted by the number of correct predictions among the top-k outputs of $\hat{o}$ which are ranked by their probability, and $|d_t|$ is the total number of appeared category labels in the target visit.
- **Code-level precision@k** measures the overall accuracy of the model predictions, which is defined as the number of correctly predicted codes divided by the total number of predicted codes among the ranked top k predictions.

  For multiple patient sequences, the code level-precision@k is defined as:

$$\text{code-level precision@}k = \frac{\sum_{i=1}^{P} |\hat{o}_{\text{correct}}|_k}{\sum_{i=1}^{P} |d_t|} \tag{61}$$

  where P indicates the total number of patients. We tune k from 5 to 30 to evaluate the coarse-grained and fine-grained performance of each model, and the greater values indicate better performance.

(2) We compare MEGACare with the following seven baselines in **Medication Recommendation Task**.

- **Logistic Regression (LR)** [62] is a linear classifier with $L2$ regularization.
- **RETAIN** [11] can provide sequence prediction using a two-level RNN neural attention model.
- **LEAP** [26] is an instance-based method that treats medication recommendation as a sentence generation task.
- **GAMENet** [27] adopts memory-augmented neural networks and stores historical medication records for prediction. Both DMNC and GAMENet contain extra ontology data.
- **SafeDrug** [18] captures the molecule structure information with the global and local encoders.
- **MICRON** [63] proposes a recurrent residual learning model to predict medication change.
- **COGNet** [64] utilizes a copy-or-predict mechanism to generate the medication combinations.

To measure the accuracy, effectiveness, and safety of a medication prediction, MEGACare uses Synergistic DDI (SDDI) rate, Antagonistic DDI (ADDI) rate, Jaccard similarity score (Jaccard) [65], Precision Recall Area Under Curve (PRAUC) [66], and Average F1 (F1). All of the metrics are macro-averaged.

- **SDDI** rate for patient i is calculated as :

$$\mathrm{SDDI}_i = \frac{\sum_{t=1}^{T_i} \sum_{k,l \in \{\hat{o}_{i,j}^{(t)}=1\}} 1\{\mathcal{M}_{kl}^{s} = 1\}}{\sum_{t=1}^{T_i} \sum_{k,l \in \{\hat{o}_{i,j}^{(t)}=1\}} 1}, \tag{62}$$

  where T_i represents the total visits of patient i, $\hat{o}_{i,j}^{(t)}$ denotes the jth elements in the predicted medication vector of the ith patient.
- **ADDI** is similar to the SDDI rate, which is calculated as:

$$\mathrm{ADDI}_i = \frac{\sum_{t=1}^{T_i} \sum_{k,l \in \{\hat{o}_{i,j}^{(t)}=1\}} 1\{\mathcal{M}_{kl}^{a} = 1\}}{\sum_{t=1}^{T_i} \sum_{k,l \in \{\hat{o}_{i,j}^{(t)}=1\}} 1}. \tag{63}$$

- **Jaccard** [65] is defined as the length ratio score of the intersection set of ground truth medications ${m_i}^{(t)}$ and predicted medications $\hat{o}_i^{(t)}$ divides the union set between ${m_i}^{(t)}$ and $\hat{o}_i^{(t)}$.

$$\mathrm{Jaccard}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} \frac{|\{o_{i,j}^{(t)} = 1\} \cap \{\hat{o}_{i,j}^{(t)} = 1\}|}{|\{o_{i,j}^{(t)} = 1\} \cup \{\hat{o}_{i,j}^{(t)} = 1\}|}. \tag{64}$$

[5] https://www.whocc.no/atc/structure_and_principles/

[6] https://pubchem.ncbi.nlm.nih.gov/

- **F1-score** is defined as:

$$\mathrm{F1}_i = \frac{1}{T_i}\sum_{t=1}^{T_i}\frac{2\mathrm{P}_i^{(t)}\mathrm{R}_i^{(t)}}{\mathrm{P}_i^{(t)}+\mathrm{R}_i^{(t)}}. \tag{65}$$

where

$$\mathrm{P}_i^{(t)} = \frac{|\{m_{i,j}^{(t)}=1\}\cap\{\hat{o}_{i,j}^{(t)}=1\}|}{|\{\hat{o}_{i,j}^{(t)}=1t\}|}, \tag{66}$$

$$\mathrm{R}_i^{(t)} = \frac{|\{m_{i,j}^{(t)}=1\}\cap\{\hat{o}_{i,j}^{(t)}=1\}|}{\{m_{i,j}^{(t)}=1\}|}, \tag{67}$$

- **PRAUC** [66] is defined as

$$\mathrm{PRAUC}_i = \frac{1}{T_i}\sum_{t=1}^{T_i}\sum_{k=1}^{|M|}\mathrm{P}(k)_i^{(t)}(\mathrm{R}(k)_i^{(t)}-\mathrm{R}(k-1)_i^{(t)}), \tag{68}$$

where k is the rank in the sequence of the retrieved medications.

5.3. Implementation details

For the diagnosis prediction task, we randomly split each dataset into training, validation, and testing sets in a 15 : 2 : 3 ratio with the same setup of previous work [13]. The threshold δ_{mr} for prediction diagnosis $o_{dp}^{(t)}$ is set to 0.5. For the medication recommendation task, we split the dataset into training, validation, and testing as 4 : 1 : 1 with the same setup of previous work [67]. The threshold δ_{mr} for prediction medications $o_{mr}^{(t)}$ is set to 0.75. The validation sets for both tasks are used to determine the best values of parameters in the training iterations. In the evaluation process, a bootstrapping sampling technique is employed instead of the conventional cross-validation approach according to the previous work [18]. The initial step involves training all models on a predetermined fixed training set and selecting hyperparameters based on a fixed validation set. Subsequently, a round of evaluation is carried out by repeatedly sampling 80% of the data points from the test set with replacement. This sampling-evaluation procedure is repeated for 10 rounds, and the resulting mean and standard deviation values are reported as the final outcomes.

In our framework, the dimensions of E_d, E_p, E_m, and the size of hidden layers in the transformer are all set to 256. In the graph-based medication encoding module, the number of hidden layers L in MPNN is 2. For the message-passing function M_ℓ, we implement it as one linear layer plus basic ReLU activation, and a mean operator is used for the update function U_ℓ. We apply the same graph-based medication encoding module with shared parameters for each single drug molecule. The weight μ_{multi} in MP loss $\mathcal{L}_{multi}$ is 0.75, and the weight μ_{mp} in dp loss $\mathcal{L}_{dp}$ is 0.9. The controlling parameters γ_a and γ_s in CDC loss $\mathcal{L}_c$ are set to 0.06 and 0.17, respectively. All the models in our experiments are optimized by the Adam optimizer [68]. We use a 2×10^{-4} learning rate to train our framework within 100 epochs. We implement our framework and all the baselines with Python 3.7.5 and PyTorch 1.6.0[7]. All the models are trained on 8 NVIDIA 2080Ti GPUs with 48 Intel Xeon CPUs. All the baselines are trained and implemented with the optimized parameters from the references.

6. Result and discussion

In this section, we first compare MEGACare with baselines in diagnosis prediction and medication recommendation tasks. Secondly, we present ablation studies for each module of MEGACare. Thirdly, we analyze the influences of different visit times for the two tasks. Additionally, we conduct a controllable DDI analysis and case studies given two patients with multiple visits for the medication prediction task. Finally, we explore multi-task learning in the framework and analyzed the experimental results.

6.1. Main results

In the diagnosis prediction task, Table 4 reports the results measured by code-level precision@k and visit-level precision@k, Table 5 reports the results measured by Jaccard, F1-score, and PRAUC score, respectively. It is noteworthy that the Avg. column in the Table 4 is the average of the metrics in Code-Levelprecision and Visit-Level precision, the Avg. in the Table 5 is the average of the metrics in Jaccard, F1-score and PRAUC, which is used to evaluate the overall performance of the various models. It is evident that the knowledge-guided models (MEGACare, GNDP, KAME, and GRAM) are generally superior to non-knowledge-guided models (CNN, RNN, GCN, and Dipole). In comparison with the best non-knowledge-guided model Dipole, MEGACare significantly improves the code-level precision by 6.95%, 9.06%, 8.93%, 9.81%, 7.98%, 8.43% when k = 5, 10, 15, 20, 25, 30, respectively, and increases the visit-level precision by 12.72%, 9.48%, 9.13%, 9.55%, 8.50%, 7.89% when k = 5, 10, 15, 20, 25, 30, respectively. Table 5 illustrates consistent outcomes, indicating that MEGACare exhibited a 2.62%, 3.42%, and 3.15% improvement in Jaccard, F1, and PRAUC indicators, respectively. The results of these experiments demonstrate that the introduction of medical knowledge is beneficial for diagnosis prediction.

For knowledge-guided models, We can observe that the proposed MEGACare outperforms all baselines on both evaluation metrics. Compared with the best knowledge-guided models GNDP, MEGACare improves the code-level precision by 0.37%, 0.70%, 0.83%, 0.97%, 0.78%, 1.24% when k = 5, 10, 15, 20, 25, 30, respectively, and improves the visit-level precision in 0.59%, 0.51%, 0.46%, 0.76%, 0.54%, 0.57% when k = 5, 10, 15, 20, 25, 30, respectively. As illustrated in Table 5, our MEGACare exhibited a 0.67% and 0.73% improvement in Jaccard and F1 score indicators, respectively. Different from existing knowledge-guided models, MEGACare combines the semantic embedding of the diagnosis code description with the hierarchical information of the ontology structure to provide a more comprehensive medical code embedding. Additionally, MEGACare utilizes multi-view learning to form patient representations from multiple perspectives, demonstrating the effectiveness and robustness of the knowledge introduction and multi-view information fusion methods of MEGACare. The above two points are the main reasons for MEGACare's improvements, and we carried out detailed ablation studies to demonstrate their effectiveness.

Table 6 reports the results measured by ADDI rate, SDDI rate, Jaccard score, F1 score, and PRAUC score for the medication recommendation task. It is noteworthy that the Avg. in the table is the average of the metrics in Jaccard, F1-score and PRAUC, which is used to evaluate the overall performance of the various models. We evaluate the performance of MEGACare in comparison to other baseline models that do not incorporate DDI knowledge (*i.e.*, LR, CNN, RNN, Transformer, and RETAIN), and those that underutilize the DDI knowledge (*i.e.*, LEAP, GAMENet, SafeDrug, and COGNet). On the one hand, the results indicate that our framework has excellent predictive accuracy, which outperforms all baselines in Jaccard, PRAUC, and F1 scores, which are at least 0.95%, 1.12%, and 1.29% higher, respectively. On the other hand, the results demonstrate that the medications predicted by our framework are safe and effective. MEGACare achieves the lowest ADDI rate (at least 0.47% lower) and the highest SDDI rate (at least 0.85% higher), demonstrating its ability to more comprehensively leverage the DDI knowledge compared to the baselines.

The results presented in the table indicate that the medication sequence generation models, namely LEAP, performed poorly in comparison to the other baseline models using multi-label prediction methods. Specifically, the baselines that did not incorporate drug–drug interaction (DDI) information (*i.e.*, LR, RETAIN, CNN, RNN, MLP) exhibited unfavorable DDI rates. While the recommendation results generated by GAMENet and COGNet were based on historical medication combinations, it is worth noting that real medication records often

[7] https://pytorch.org/

Table 4
Comparative experiments results of diagnosis prediction task on the MIMIC-III dataset.

	Code-Level Precision@k							Visit-Level Precision@k						
Model	5	10	15	20	25	30	Avg.	5	10	15	20	25	30	Avg.
CNN	0.3026	0.4824	0.6025	0.6921	0.7590	0.8140	0.6087	0.6399	0.5840	0.6267	0.6984	0.7626	0.8160	0.6873
RNN	0.2941	0.4836	0.6106	0.6961	0.7629	0.8119	0.6098	0.6580	0.6186	0.6637	0.7254	0.7836	0.8272	0.7127
GCN	0.2465	0.3902	0.4909	0.5941	0.6790	0.7317	0.5222	0.5526	0.5328	0.5751	0.6249	0.7010	0.7324	0.6198
Dipole	0.2774	0.4565	0.5891	0.6687	0.7456	0.7902	0.5879	0.6220	0.5869	0.6315	0.6932	0.7542	0.8017	0.6815
RETAIN	0.2959	0.4758	0.5974	0.6855	0.7584	0.8137	0.6044	0.6284	0.5760	0.6318	0.7018	0.7687	0.8212	0.6879
GRAM	0.3123	0.5028	0.6296	0.7142	0.7786	0.8260	0.6272	0.6686	0.6448	0.6847	0.7369	0.7997	0.8414	0.7293
KAME	0.3167	0.5100	0.6379	0.7210	0.7862	0.8303	0.6336	0.6703	0.6568	0.6967	0.7562	0.8090	0.8470	0.7393
GNDP	0.3432	0.5401	0.6701	0.7571	0.8176	0.8629	0.6651	0.7433	0.6766	0.7182	0.7811	0.8338	0.8749	0.7711
MEGACare	**0.3469**	**0.5471**	**0.6784**	**0.7668**	**0.8254**	**0.8745**	0.6738	**0.7492**	**0.6817**	**0.7228**	**0.7887**	**0.8392**	**0.8806**	0.7773

Table 5
Another comparative experiments results of diagnosis prediction task.

Model	Jaccard	F1 score	PRAUC	Avg.	Avg. # of Diag.
CNN	0.4581 ± 0.0016	0.6178 ± 0.0023	0.7281 ± 0.0024	0.6013	17.0412 ± 0.1025
RNN	0.4608 ± 0.0011	0.6254 ± 0.0026	0.7373 ± 0.0032	0.6045	15.1922 ± 0.1334
Dipole	0.4601 ± 0.0021	0.6212 ± 0.0014	0.7277 ± 0.0029	0.6031	15.5530 ± 0.0803
RETAIN	0.4623 ± 0.0028	0.6273 ± 0.0017	0.7391 ± 0.0025	0.6095	15.9061 ± 0.0762
GRAM	0.4679 ± 0.0017	0.6364 ± 0.0025	0.7465 ± 0.0023	0.6169	16.8994 ± 0.1573
KAME	0.4711 ± 0.0013	0.6397 ± 0.0024	0.7477 ± 0.0031	0.6195	16.6465 ± 0.1737
GNDP	0.4796 ± 0.0019	0.6481 ± 0.0021	**0.7562** ± 0.0028	0.6270	15.2277 ± 0.0831
MEGACare	**0.4863** ± 0.0018	**0.6554** ± 0.0018	0.7553 ± 0.0021	0.6336	15.4369 ± 0.1114

Table 6
Comparative experiments results of medication recommendation task on the MIMIC-III dataset.

Model	ADDI ↓	SDDI ↑	Jaccard ↑	F1-score ↑	PRAUC ↑	Avg.	Avg. # of Med.
LR	0.0881 ± 0.0008	0.2133 ± 0.0015	0.4604 ± 0.0026	0.6218 ± 0.0026	0.6641 ± 0.0042	0.5821	18.7424 ± 0.0738
MLP	0.0821 ± 0.0006	0.2077 ± 0.0003	0.4587 ± 0.0011	0.6171 ± 0.0013	0.6938 ± 0.0025	0.5898	18.0731 ± 0.0835
CNN	0.0849 ± 0.0005	0.2125 ± 0.0011	0.4508 ± 0.0023	0.6154 ± 0.0024	0.7062 ± 0.0013	0.5908	18.7962 ± 0.0854
RNN	0.0851 ± 0.0007	0.2059 ± 0.0007	0.4648 ± 0.0021	0.6245 ± 0.0018	0.7293 ± 0.0017	0.6062	19.3136 ± 0.0725
Transformer	0.0875 ± 0.0005	0.2149 ± 0.0011	0.4706 ± 0.0013	0.6379 ± 0.0021	0.7357 ± 0.0016	0.6143	21.0318 ± 0.1511
RETAIN	0.0933 ± 0.0015	0.2238 ± 0.0031	0.4752 ± 0.0027	0.6373 ± 0.0027	0.7359 ± 0.0038	0.6161	16.5626 ± 0.1008
LEAP	0.0879 ± 0.0007	0.2082 ± 0.0010	0.4353 ± 0.0021	0.5982 ± 0.0021	0.7028 ± 0.0025	0.5787	15.8053 ± 0.0721
GAMENet	0.0881 ± 0.0004	0.2179 ± 0.0007	0.4978 ± 0.0028	0.6372 ± 0.0025	0.7438 ± 0.0028	0.6262	20.7781 ± 0.2208
MICRON	0.0716 ± 0.0006	0.2122 ± 0.0028	0.4996 ± 0.0028	0.6435 ± 0.0024	0.7463 ± 0.0022	0.6298	17.7604 ± 0.0990
SafeDrug	0.0664 ± 0.0004	0.2172 ± 0.0013	0.5003 ± 0.0017	0.6478 ± 0.0017	0.7483 ± 0.0026	0.6321	20.7939 ± 0.0628
COGNet	0.0876 ± 0.0004	0.2169 ± 0.0013	0.5165 ± 0.0023	0.6684 ± 0.0025	0.7558 ± 0.0030	0.6469	28.3909 ± 0.2008
MEGACare	**0.0617** ± 0.0004	**0.2264** ± 0.0011	**0.5231** ± 0.0017	**0.6796** ± 0.0013	**0.7687** ± 0.0020	0.6571	20.1396 ± 0.1144
MEGACare$_{med}$	0.0653 ± 0.0004	0.2215 ± 0.0013	0.5087 ± 0.0021	0.6537 ± 0.0025	0.7527 ± 0.0027	0.6383	18.7535 ± 0.0706

↓ means the corresponding metric is the lower the better and ↑ means the opposite.

Table 7
The model complexity comparison.

Model	# of Param.	Training time	Inference time
RETAIN	305,355	71.54 s/Epoch	6.89 s
LEAP	379,975	606.98 s/Epoch	36.27 s
GAMENet	419,397	184.88 s/Epoch	11.39 s
SafeDrug	388,798	155.69 s/Epoch	13.56 s
MICRON	307,535	117.09 s/Epoch	12.82 s
COGNet	1,357,560	253.95 s/Epoch	193.38 s
MEGACare	570,823	192.78 s/Epoch	15.59 s

involve high DDI rates, leading to an high DDI rate in their respective recommendations.

In the results of comparative experiments, we found that SafeDrug's ADDI rate also remains at a low level, because SafeDrug contains a medication encoding module based on the atom–atom graph. For a fair comparison of the medication coding modules between MEGACare and SafeDrug, we propose MEGACare$_{med}$, which utilizes the same patient representation settings as the SafeDrug model. The comparison between MEGACare$_{med}$ and SafeDrug highlights the effectiveness of our medication encoding step in the code initialization module, which leverages the substructure of the molecular graph and employs a triplet learning loss incorporating two different DDI knowledge to constrain the learned medication embeddings. Moreover, MEGACare provides constraints on both the ADDI and SDDI rates with the controllable DDI loss function, making the recommended medications not only secure but also with more synergistic effects.

The model complexity of our proposed MEGACare framework and several other deep learning models on the medication recommendation task are assessed and compared in Table 7. To ensure a fair comparison, identical experimental conditions were employed, including the utilization of a batch size of 64 and the same used devices. Specifically, our model outperforms SafeDrug and MICRON in terms of accuracy, but at the expense of longer training time. In terms of efficiency, our model demonstrates relatively lower space and time complexity compared to COGNet, which is currently considered as the state-of-the-art model in this field. These findings suggest that MEGACare can accurately recommend medications that are both safe and effective for patient treatment.

Table 8
Ablation experiments results of diagnosis prediction task on the MIMIC-III dataset.

	Code-Level Precision@k							Visit-Level Precision@k						
Model	5	10	15	20	25	30	Avg.	5	10	15	20	25	30	Avg.
MEGACare	**0.3469**	**0.5471**	**0.6784**	**0.7668**	**0.8254**	**0.8745**	0.6738	**0.7492**	**0.6817**	**0.7228**	**0.7887**	**0.8392**	**0.8806**	0.7773
MEGACare w/o CodeInit	0.3357	0.5374	0.6531	0.7458	0.8101	0.8484	0.6550	0.7362	0.6673	0.6998	0.7751	0.8193	0.8654	0.7605
MEGACare w/o $V_{\mathcal{CG}}$	0.3401	0.5397	0.6683	0.7569	0.8026	0.8585	0.6610	0.7342	0.6698	0.7125	0.7733	0.8290	0.8678	0.7644
MEGACare w/o $V_{\mathcal{EH}}$	0.3325	0.5177	0.6466	0.7374	0.7960	0.8397	0.6449	0.7276	0.6534	0.6902	0.7581	0.8049	0.8484	0.7471
MEGACare w/o $V_{\mathcal{SH}}$	0.3342	0.5247	0.6512	0.7419	0.8019	0.8408	0.6497	0.7295	0.6568	0.6964	0.7607	0.8075	0.8510	0.7503
MEGACare w/o $\mathcal{L}_{\mathrm{IC}}$	0.3307	0.5231	0.6513	0.7397	0.7941	0.8374	0.6460	0.7279	0.6527	0.6957	0.7620	0.8218	0.8604	0.7534
MEGACare w/o $\mathcal{L}_{\mathrm{HS_{IB}}}$	0.3339	0.5271	0.6563	0.7451	0.8003	0.8624	0.6541	0.7321	0.6591	0.7018	0.7712	0.8259	0.8711	0.7602
MEGACare w/o $\mathcal{L}_{\mathrm{MV_{IB}}}$	0.3341	0.5262	0.6559	0.7480	0.8036	0.8624	0.6550	0.7424	0.6615	0.6998	0.7694	0.8237	0.8683	0.7609

Table 9
Ablation experiments results of medication recommendation task on the MIMIC-III dataset.

Model	ADDI ↓	SDDI ↑	Jaccard ↑	F1-score ↑	PRAUC ↑	Avg.	Avg. # of Med.
MEGACare	**0.0617** ± 0.0004	**0.2264** ± 0.0011	**0.5231** ± 0.0017	**0.6796** ± 0.0013	**0.7687** ± 0.0020	0.6571	20.1396 ± 0.1144
MEGACare w/o CodeInit	0.0653 ± 0.0005	0.2236 ± 0.0013	0.5176 ± 0.0011	0.6718 ± 0.0015	0.7583 ± 0.0019	0.6492	19.5021 ± 0.1698
MEGACare w/o $V_{\mathcal{CG}}$	0.0648 ± 0.0008	0.2233 ± 0.0015	0.5174 ± 0.0016	0.6738 ± 0.0019	0.7624 ± 0.0021	0.6512	21.6457 ± 0.1435
MEGACare w/o $V_{\mathcal{EH}}$	0.0659 ± 0.0007	0.2198 ± 0.0013	0.5132 ± 0.0019	0.6687 ± 0.0021	0.7574 ± 0.0023	0.6464	20.3739 ± 0.0738
MEGACare w/o $V_{\mathcal{SH}}$	0.0651 ± 0.0008	0.2203 ± 0.0011	0.5157 ± 0.0020	0.6701 ± 0.0023	0.7595 ± 0.0019	0.6485	20.1247 ± 0.0738
MEGACare w/o $\mathcal{L}_{\mathrm{IC}}$	0.0645 ± 0.0011	0.2243 ± 0.0017	0.5054 ± 0.0026	0.6668 ± 0.0031	0.7574 ± 0.0042	0.6432	21.4274 ± 0.1528
MEGACare w/o $\mathcal{L}_{\mathrm{HS_{IB}}}$	0.0636 ± 0.0009	0.2198 ± 0.0015	0.5096 ± 0.0023	0.6703 ± 0.0021	0.7578 ± 0.0034	0.6459	20.3780 ± 0.1406
MEGACare w/o $\mathcal{L}_{\mathrm{MV_{IB}}}$	0.0644 ± 0.0010	0.2203 ± 0.0016	0.5087 ± 0.0020	0.6698 ± 0.0023	0.7601 ± 0.0031	0.6462	20.9799 ± 0.1670
MEGACare w/o $\mathcal{L}_{\mathrm{DDI}}$	0.0868 ± 0.0006	0.2179 ± 0.0013	0.5135 ± 0.0016	0.6748 ± 0.0012	0.7641 ± 0.0022	0.6508	21.0715 ± 0.2104
GAMENet	0.0881 ± 0.0004	0.2179 ± 0.0007	0.4978 ± 0.0028	0.6372 ± 0.0025	0.7438 ± 0.0028	0.6262	20.7781 ± 0.2208
GAMENet + PatRepr	0.0871 ± 0.0007	0.2163 ± 0.0017	0.5101 ± 0.0028	0.6512 ± 0.0027	0.7598 ± 0.0029	0.6404	23.2187 ± 0.1604
SafeDrug	0.0664 ± 0.0004	0.2172 ± 0.0013	0.5003 ± 0.0017	0.6478 ± 0.0017	0.7483 ± 0.0026	0.6321	20.7939 ± 0.0628
SafeDrug + PatRepr	0.0652 ± 0.0004	0.2169 ± 0.0015	0.5134 ± 0.0021	0.6592 ± 0.0021	0.7613 ± 0.0025	0.6446	21.5862 ± 0.1353
COGNet	0.0876 ± 0.0004	0.2169 ± 0.0013	0.5165 ± 0.0023	0.6684 ± 0.0025	0.7558 ± 0.0030	0.6469	28.3909 ± 0.2008
COGNet + PatRepr	0.0862 ± 0.0005	0.2157 ± 0.0013	0.5222 ± 0.0024	0.6746 ± 0.0023	0.7599 ± 0.0021	0.6517	29.5331 ± 0.1418

6.2. Results of ablation experiments

Considering MEGACare is a relatively complicated framework, we perform detailed ablation studies to examine the effectiveness and necessity of the proposed components for the Diagnosis Prediction Task in Table 8 and for Medication Recommendation Task in Table 9.

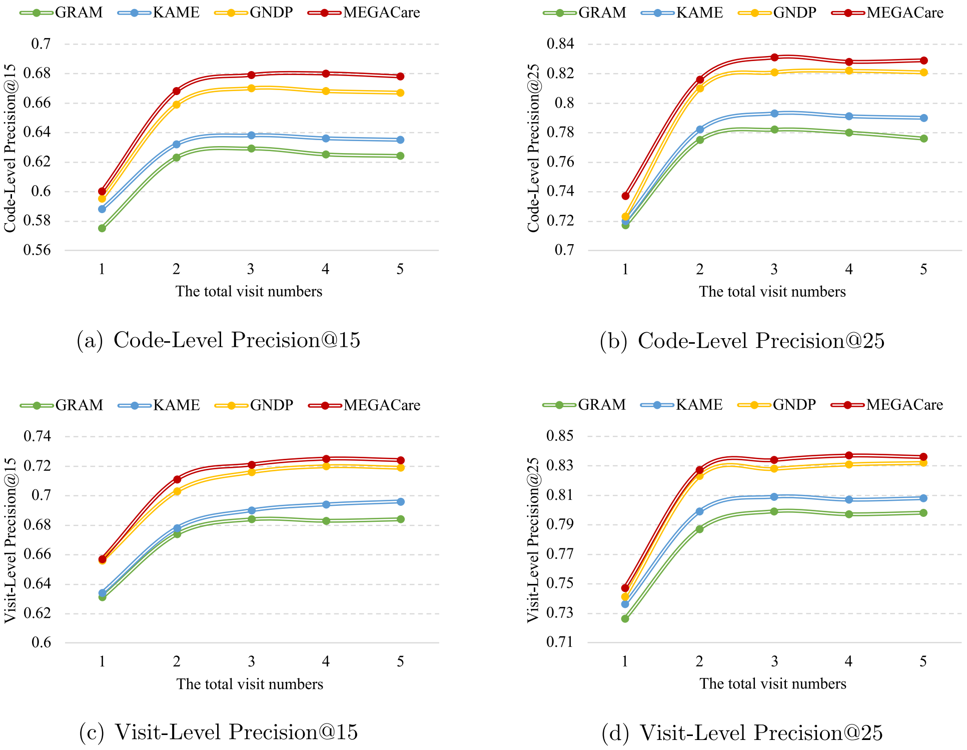
- MEGACare w/o CodeInit means without the medical code pretraining in the code initialization module.
- MEGACare w/o $V_{\mathcal{CG}}$ means without Code Graph View in the patient representation module.
- MEGACare w/o $V_{\mathcal{EH}}$ means without Enhanced Hypergraph View in the patient representation module.
- MEGACare w/o $V_{\mathcal{SH}}$ means without Sub-hypergraph View in the patient representation module.
- MEGACare w/o $\mathcal{L}_{\mathrm{IC}}$ means without the information constraint loss function.
- MEGACare w/o $\mathcal{L}_{\mathrm{HS_{IB}}}$ means without the hypergraph structure IB loss function.
- MEGACare w/o $\mathcal{L}_{\mathrm{MV_{IB}}}$ means without the multi-view IB loss function.
- MEGACare w/o $\mathcal{L}_{\mathrm{DDI}}$ means without the controllable DDI loss function.
- GAMENet + PatRepr means that we replaced the patient representation module of GAMENet with our patient representation module.
- SafeDrug + PatRepr means that we replaced the patient representation module of SafeDrug with our patient representation module.
- COGNet + PatRepr means that we replaced the patient representation module of COGNet with our patient representation module.

In Table 8, the results of ablation studies are reported by code-level precision@k and visit-level precision@k scores with different k values and average performance. In Table 9, the results of ablation studies are reported by ADDI rate, SDDI rate, Jaccard, F1 score, PRAUC, and average performance score. We conducted four sets of experiments.

(1) In the first experiment set, we sought to investigate the influence of the Code Initialization Module on our framework. Without the initial diagnosis and procedure code pretraining, the hypergraph neural network performs subsequent learning based on randomly initialized node embeddings, which detrimentally affected the model performance.

(2) In the second experiment set, we aim to explore the impact of different views in the patient representation module on model performance . By comparing the results of MEGACare w/o $V_{\mathcal{CG}}$, MEGACare w/o $V_{\mathcal{EH}}$, and MEGACare w/o $V_{\mathcal{SH}}$ in both prediction tasks, it can be observed that the Enhanced Hypergraph View ($V_{\mathcal{EH}}$) exhibited significant improvement in both tasks, the Code Graph View ($V_{\mathcal{CG}}$) and Sub-hypergraph View ($V_{\mathcal{SH}}$) are proposed to enhance the capacity of hypergraph modeling. In our proposed MEGACare framework, three different relationships (relationships among medical codes, patient-visit-code relationships, and patient-visit relationships) are simultaneously considered to describe high-level associations among complex EHR data. Neglecting information from any of the views impairs the patient representation learning process, thus preventing the model from detecting valid correlations between medical codes, patient visits, and patient medical trajectories, thus resulting in suboptimal results.

(3) In the third experiment set, we sought to analyze the effect of different loss functions on the framework's performance. By comparing the results of MEGACare w/o $\mathcal{L}_{\mathrm{IC}}$, MEGACare w/o $\mathcal{L}_{\mathrm{HS_{IB}}}$, and MEGACare w/o $\mathcal{L}_{\mathrm{MV_{IB}}}$ in both prediction tasks. it can be observed that both $\mathcal{L}_{\mathrm{HS_{IB}}}$ and $\mathcal{L}_{\mathrm{MV_{IB}}}$ obtain complete patient representations from multiple perspectives by constraining the hypergraph structure and multi-view learning respectively. We also consider the role of $\mathcal{L}_{\mathrm{DDI}}$ in the drug recommendation task. Without incorporating the controllable



(a) Code-Level Precision@15

(b) Code-Level Precision@25

(c) Visit-Level Precision@15

(d) Visit-Level Precision@25

Fig. 4. The effect of the number of visits for various models.

DDI Loss Function, the variant of MEGACare was observed to have similar DDI rates to the MIMIC-III dataset, while elevating the average number of recommended medications. In comparison to models without the DDI loss function(*i.e.*, GAMENet, and COGNet), it was observed that the proposed framework effectively simulated the performance of doctors and exhibited satisfactory accuracy in prescribing medications.

(4) In the final experiment set, the patient representation module of the MEGACare framework was incorporated into GAMENet, SafeDrug, and COGNet respectively, and the variants of these models were compared with the original models in the medication recommendation task. The results revealed that the variants of GAMENet, SafeDrug, and COGNet, which incorporated the patient representation module of MEGACare, exhibited improved prediction performance and a lower ADDI rate than the models themselves, thereby demonstrating the comprehensiveness and effectiveness of the patient representation learning module in our proposed framework.

In conclusion, it is evident that each component of the proposed MEGACare framework contributes to the enhancement of precision and accuracy.

6.3. Influences of the number of visits

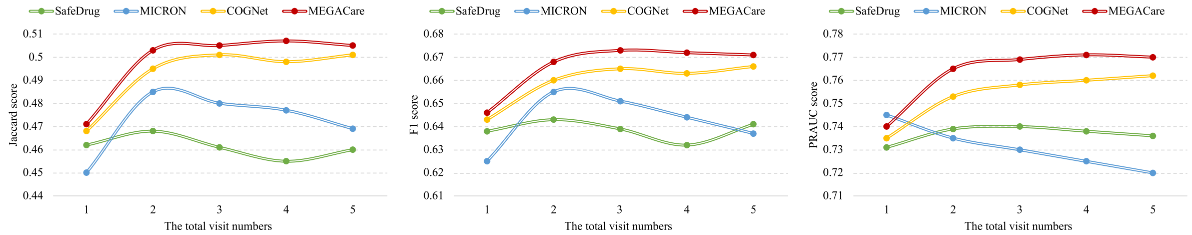
To further explore whether our MEGACare can better capture historical medication information, we conduct two sets of analysis experiments to explore the influences of total patient visits and utilized historical visits on diagnosis prediction and drug recommendation tasks, respectively.

6.3.1. The influences of the total patient visits

First, we analyze the impact of the different total number of visits in the two tasks on the performance of the framework. The statistical analysis of Tables 2 and 3 shows that the average number of visits for different patients in our dataset is less than three, with the proportion of patients with more than five visits not exceeding 10%. Therefore, we stratified the datasets based on the total visit numbers to analyze the effect of varying total visit numbers on the performance of the framework. For the diagnosis prediction task, we choose the GRAM, KAME, and GDNP as stronger baselines for comparison, and select Code-Level Precision@15, Code-Level Precision@25, Visit-Level Precision@15, Visit-Level Precision@25 as metrics to measure the effectiveness of each model. For the medication recommendation task, we choose the SafeDrug, MICRON, and COGNet as stronger baselines, which also incorporate historical information, to conduct the comparison analysis, and select Jaccard, F1 score, and PRAUC as metrics to measure the effectiveness of each model. The comparison results of various methods on the different number of visits are in Figs. 4 and 5. The horizontal axis represents the patient visit times and the vertical axis represents the values of the different evaluation metrics.

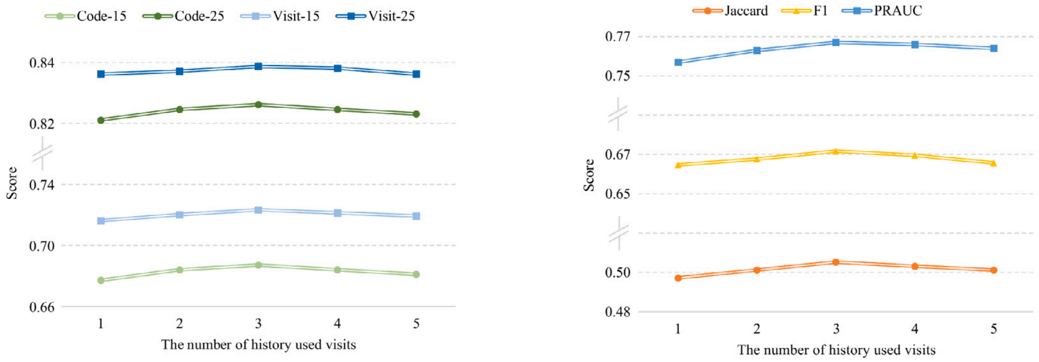
The results show that MEGACare almost achieves the best performance with different visit times on two prediction tasks. Specifically, (1) In the diagnosis prediction task, an increase in visits was observed to improve the performance of each model, indicating that these models effectively incorporated patient historical information. The representational power of hypergraphs enabled our framework to learn more comprehensive representations and make more accurate predictions. (2) In the medication recommendation task, MEGCare and COGNet achieve relatively better performance with more visits, showing that both models effectively use patient historical information. On the contrary, the performance of SafeDrug remains flat and MICRON shows a decreasing trend. SafeDrug utilizes RNN to model the patient history, which cannot model long-range relationships. MICRON iteratively updates the past medication combination to form the new medication set, which will lead to an error accumulation problem.

In conclusion, It is evident that our MEGACare can stably provide accurate diagnosis prediction and secure and effective medication



(a) Jaccard score (b) F1 score (c) PRAUC score

Fig. 5. The effect of the number of visits for various models.



(a) Diagnosis Prediction task (b) Medication Recommendation task

Fig. 6. Study of different k for MEGACare in multi-visit clinical outcome prediction tasks.

recommendations with increasing visits while comparing with other models.

6.3.2. The influences of utilized historical visits

In Section 6.3.1, we analyze the influence of the total number of visits on the performance of the framework. As indicated by Figs. 4 and 5, the highest accuracy was achieved with four visits. To further explore the effectiveness of different historical visit numbers k, a comparison experiment involving varying numbers of used history visit times of our MEGACare was conducted in two tasks, as depicted in Fig. 6.

We considered values of k ranging from 1 to 5, where the value of k corresponds to the scenario which only considers the information from the last k visits. We use the same evaluation metrics as Section 6.3.1. Fig. 6 shows the results of effectiveness metrics for different k. When k = 3, the model performance of MEGACare is the best. All effectiveness metrics increase as k increases from 1 to 3 and decrease slightly from k = 4 to 5. The results above suggest that multiple historical visit records have a negligible influence on current predictions. Due to our framework's adherence to the traditional prediction paradigm based on multiple visits without considering the time interval between them, the long-range historical records may generate noise information that could mislead the current prediction. Therefore, the value of k should not be too high.

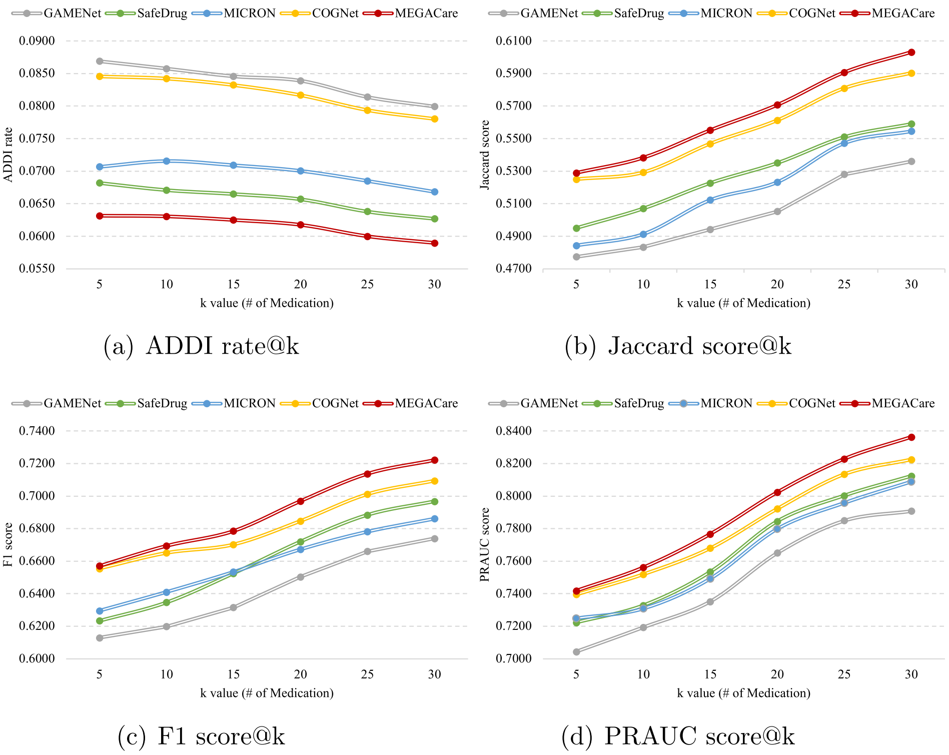
6.4. Recommended medication number analysis

In this section, our analysis focuses on investigating the influences of the number of recommended medications. We employ the recommended number of medications as a medication quantity threshold (ranges from 0 to 30) to evaluate the model's performance. We calculate six evaluation metrics (*i.e.*, Code-Level Precision@k, Visit-Level Precision@k, ADDI rate, Jaccard score, F1 score, and PRAUC score) and present the results in Table 10 and Fig. 7. Our findings indicate that the model's overall performance improves as the medication number threshold increases. This can be attributed to the fact that defining the task as a multi-label prediction problem with a smaller medication quantity threshold leads to a reduction in available data and an increment in noise, which ultimately results in a loss of model accuracy.

6.5. Controllable DDI analysis

We evaluate the efficacy of our proposed CDC loss in controlling the ADDI rate and SDDI rate in the recommended medications. To this end, we train our MEGACare model using various combinations of acceptance thresholds. In particular, we select the ADDI acceptance threshold γ_a from the range of 0.00 to 0.09, and the SDDI acceptance threshold γ_s from 0.16 to 0.25. Given that the DDI ground truth in the MIMIC III dataset comprises inherent ADDI and SDDI rates of 0.813 and 0.1583, respectively, our objective is to ensure that the ADDI rate of the medications recommended by our framework is lower than the ground truth, while the SDDI rate is higher. It is worth noting that a higher SDDI rate may result in redundant medications being prescribed to patients. As such, we impose an upper limit of 0.25 on the SDDI rate. Subsequently, we perform 10 rounds of testing for each combination of acceptance thresholds, and we record the trend of changes for each indicator in a coordinate system. The average value of each indicator is plotted in Fig. 8.

Each sub-figure in Figure shows the trend of each indicator as the ADDI acceptance thresholds γ_a and SDDI acceptance thresholds γ_s are varied. For example, when the SDDI acceptance thresholds γ_s are increased while holding the ADDI acceptance threshold constant, it results in a disturbance in the ADDI rate (as shown in Fig. 8.(d)) and a decline in the overall recommendation performance (as shown in Fig. 8.(a)). This observation confirms that a recklessly increase in the SDDI acceptance threshold γ_s may compel the model to recommend redundant synergistic medications, leading to a deterioration in



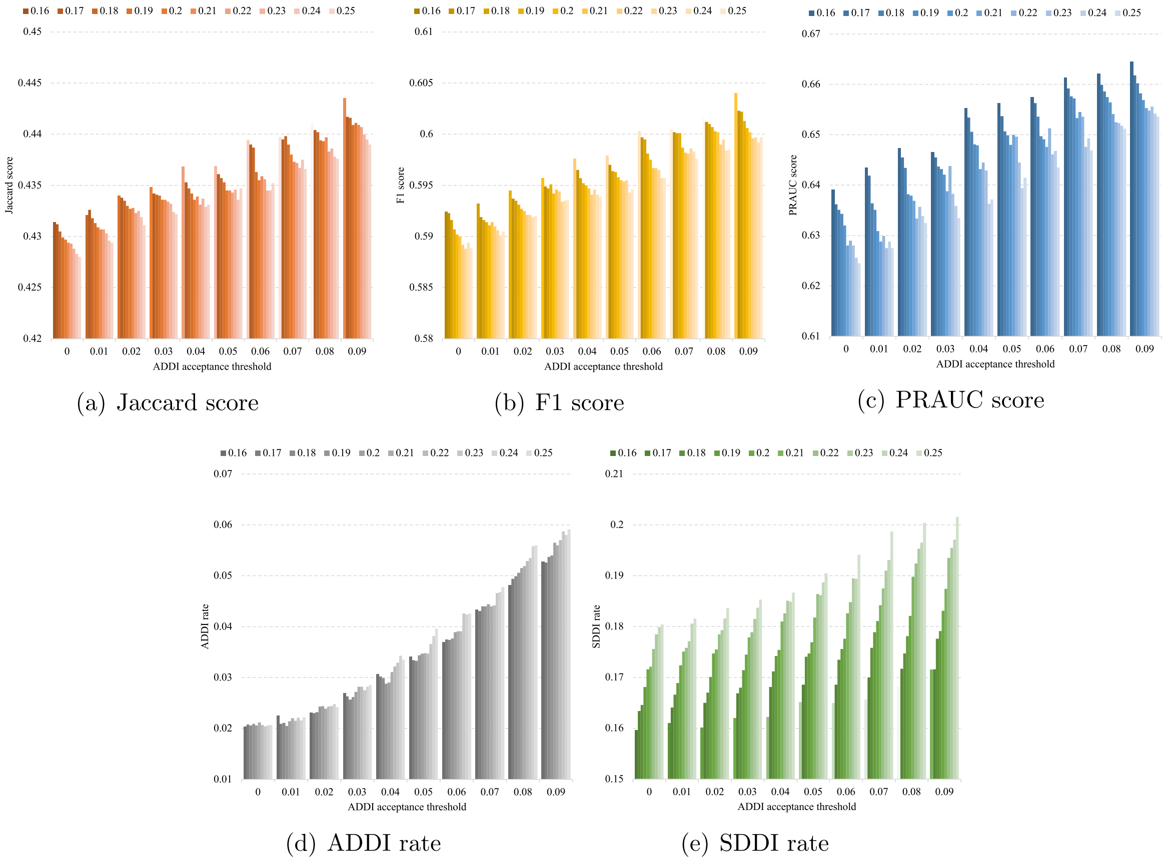
(a) ADDI rate@k

(b) Jaccard score@k

(c) F1 score@k

(d) PRAUC score@k

Fig. 7. The effect of the number of medication for medication recommendation task.



(a) Jaccard score

(b) F1 score

(c) PRAUC score

(d) ADDI rate

(e) SDDI rate

Fig. 8. Sensitivity analysis of controllable DDI w.r.t. the SDDI acceptance threshold γ_s and the ADDI acceptance threshold γ_a.

Table 10
Comparative experiments results of medication recommendation task on the MIMIC-III dataset.

	Code-Level Precision@k							Visit-Level Precision@k						
Model	5	10	15	20	25	30	Avg.	5	10	15	20	25	30	Avg.
LEAP	0.1955	0.3705	0.5210	0.6440	0.7402	0.8065	0.5463	0.4158	0.4527	0.5586	0.6610	0.7443	0.8216	0.6091
RETAIN	0.2434	0.4312	0.5617	0.6614	0.7417	0.8190	0.5765	0.5271	0.5434	0.5941	0.6798	0.7507	0.8317	0.6558
GAMENet	0.2491	0.4436	0.5816	0.6866	0.7667	0.8305	0.5930	0.5306	0.5478	0.6298	0.7113	0.7871	0.8501	0.6761
MICRON	0.2554	0.4557	0.5914	0.7057	0.7849	0.8387	0.6053	0.5467	0.5731	0.6328	0.7287	0.7994	0.8495	0.6883
SafeDrug	0.2539	0.4539	0.5978	0.7012	0.7784	0.8340	0.6032	0.5432	0.5568	0.6341	0.7242	0.7892	0.8546	0.6837
COGNet	0.2534	0.4492	0.5982	0.7087	0.7898	0.8466	0.6077	0.5440	0.5628	0.6404	0.7311	0.8058	0.8546	0.6898
MEGACare	**0.2589**	**0.4670**	**0.6190**	**0.7279**	**0.8046**	**0.8562**	0.6223	**0.5577**	**0.5847**	**0.6632**	**0.7506**	**0.8208**	**0.8681**	0.7075

Table 11
Two examples of recommending medications.

Patient	Visit	Method	ADDI ↓	Jaccard ↑	Correct	Missed	Unseen
A	1st	GAMENet	0.1083	0.3902	16	10	15: A01A$^{(A\#2)}$, A02B$^{(A\#2)}$, A07A, A12B$^{(A\#1)}$, N06A$^{(A\#6)}$, A03B, C10A$^{(A\#5)}$, C01B, C09A$^{(A\#6)}$, N05A, C01D$^{(A\#1)}$, R05C, R01A$^{(A\#5)}$, C01A, C03B
		SafeDrug	0.0588	0.3750	12	14	6: A01A$^{(A\#1)}$, A02B$^{(A\#1)}$, A12B$^{(A\#1)}$, A03B, R05C, R01A$^{(A\#4)}$
		COGNet	0.1023	0.3333	14	12	16: A01A$^{(A\#1)}$, A02B$^{(A\#2)}$, A07A, A12B$^{(A\#1)}$, J01M, A03B, C01B, B02B, N05A, C01D$^{(A\#1)}$, R05C, R01A$^{(A\#5)}$, A07D, D04A, R05D$^{(A\#1)}$, C03B
		MEGACare	0.0708	0.5294	18	8	8: A01A$^{(A\#1)}$, A02B$^{(A\#1)}$, A12A, A12B$^{(A\#1)}$, J01M, N05A, R05C, R01A$^{(A\#4)}$
	2nd	GAMENet	0.0808	0.4693	23	8	18: N01A$^{(A\#1)}$, N06A$^{(S\#1,A\#6)}$, J01M, B01A$^{(A\#3)}$, C10A$^{(A\#5)}$, C01B, D01A, J01X$^{(A\#3)}$, B02B, N03A$^{(A\#3)}$, N05A, D11A, A07E$^{(A\#5)}$, R05C$^{(A\#2)}$, A03F, R01A, V03A, L04A
		SafeDrug	0.0620	0.5128	20	11	8: N01A$^{(A\#1)}$, N06A$^{(S\#1,A\#6)}$, J01M, B01A$^{(A\#3)}$, C01B, A07E$^{(A\#5)}$, R05C, R01A$^{(A\#4)}$
		COGNet	0.0896	0.5455	24	7	13: N01A$^{(A\#1)}$, B01A$^{(A\#3)}$, C01B, C09A$^{(A\#5)}$, J01X$^{(A\#3)}$, N05A, A07E$^{(A\#6)}$, R01A$^{(A\#5)}$, P01A, V03A, M04A$^{(A\#2)}$, C09C$^{(A\#5)}$, J01G
		MEGACare	0.0602	0.6410	25	6	8: J01M, B01A$^{(A\#2)}$, C01B, N05A, A07E, R05C, R01A$^{(A\#5)}$, V03A, R05D
	3rd	GAMENet	0.0910	0.4883	21	5	17: A01A$^{(S\#2,A\#1)}$, A12A, N01A, C03C, N02A$^{(A\#2)}$, N06A$^{(A\#8)}$, A02A, C10A$^{(A\#5)}$, C01B, N03A$^{(A\#2)}$, N05A, C08C, N05B$^{(A\#2)}$, R05C, J05A$^{(A\#2)}$, J01F, L04A$^{(A\#7)}$
		SafeDrug	0.0791	0.4848	16	10	7: A01A$^{(S\#1,A\#2)}$, C03C, N02A$^{(A\#2)}$, C01B, N03A, N05B$^{(A\#2)}$, R05C
		COGNet	0.0867	0.5428	19	7	9: A01A$^{(S\#2,A\#2)}$, C03C, A02A, C01B, C09A$^{(A\#5)}$, R05C, C01A, J01F, M04A$^{(A\#1)}$
		MEGACare	0.0763	0.6875	22	4	6: A01A$^{(S\#2,A\#2)}$, C03C, N02A$^{(A\#2)}$, C01B, N05B$^{(A\#2)}$, R05C
B	1st	GAMENet	0.0910	0.2917	7	9	8: A01A$^{(S\#1,A\#1)}$, A02B$^{(S\#1)}$, N02A, B01A$^{(A\#2)}$, C02D, A02B$^{(S\#1)}$, N03A$^{(A\#1)}$
		SafeDrug	0.0791	0.3478	8	8	7: A01A$^{(S\#1,A\#1)}$, A02B$^{(S\#1)}$, N02A, B01A$^{(A\#2)}$, C02D, R06A$^{(S\#1,A\#1)}$, J01X
		COGNet	0.0867	0.3043	8	8	7: A01A$^{(S\#1,A\#1)}$, A02B$^{(S\#1)}$, N02A, B01A$^{(A\#2)}$, C02D, N05B$^{(A\#1)}$, N03A$^{(A\#1)}$
		MEGACare	0.0763	0.3636	8	8	6: A02B$^{(S\#4)}$, N02A, N06A$^{(S\#2)}$, B01A$^{(A\#2)}$, N05B$^{(A\#1)}$, N03A$^{(A\#1)}$
	2nd	GAMENet	0.0767	0.3940	13	8	12: A01A$^{(A\#1)}$, A07A, A12B$^{(A\#1)}$, N06A$^{(A\#6)}$, A02A, J01M, B01A$^{(A\#3)}$, C01B, N03A, A07E$^{(A\#5)}$, R03A$^{(A\#2)}$, R05C
		SafeDrug	0.0715	0.3571	10	11	7: A01A$^{(A\#1)}$, A12B$^{(A\#1)}$, A02A, B01A$^{(A\#2)}$, A11C$^{(A\#2)}$, R03A$^{(A\#2)}$, M04A$^{(A\#1)}$
		COGNet	0.0832	0.4516	14	7	10: A01A$^{(A\#1)}$, A07A, A12B$^{(A\#1)}$, A02A, J01M, B01A$^{(A\#3)}$, C01B, N05C$^{(A\#2)}$, A07E$^{(A\#5)}$, R03A$^{(A\#2)}$
		MEGACare	0.0671	0.5714	16	5	7: A01A$^{(A\#1)}$, A07A, A12B, J01M, B01A$^{(A\#2)}$, C01B, R03A$^{(A\#1)}$

The column of Unseen means the drug occurred in our recommendations while not in the goal EHR data. More details are in the Section 6.6.

the model's performance. Similarly, decreasing the ADDI acceptance threshold γ_a while holding the SDDI acceptance threshold constant results in a significant reduction in the ADDI rate, along with a decrease in all other evaluation metrics (as shown in Figs. 8.(a)–(c)). The rationale behind this lies in the fact that MIMIC-III dataset inherently contains ADDIs, as stated earlier. Thus, when the ADDI acceptance threshold γ_a is lowered, medications with specific ADDI rates from the ground truth will be excluded from our recommendations, leading to a deviation from the ground truth. These observations validate the effectiveness of our proposed CDC loss, which can effectively and accurately control the ADDI and SDDI rates in the recommended medications.

6.6. Case studies

Table 11 illustrates our case studies. The "Correct" column indicates the element number of the interaction of goal EHR data and our recommendations, while the "Missed" column indicates medications that were present in the goal EHR data but not identified by our recommendations. The "Unseen" column lists medications that were recommended by MEGACare but not present in the gold EHR data. For example, in the first row, there are 15 wrong predicted medications. A01 A$^{(A\#1)}$ indicate there exists s antagonistic interaction between the medication A01 A with 16 correct medication predictions. N06 A$^{(S\#1,A\#6)}$ in the fifth row indicates there exists 1 synergistic interaction and 6 antagonistic interactions between the medication N06 A with 23 correct medication predictions.

In the case of patient A, MEGACare demonstrates a significant improvement over other baseline methods in each visit, as evidenced by the Jaccard measure. This suggests that MEGACare is highly effective in providing medication recommendations that meet the patient's needs. Furthermore, as the number of visits increases, MEGACare consistently shows incremental improvements in performance. Specifically, there is an 11.16% increase in Jaccard from visit 1 to visit 2, and a 4.64% increase from visit 2 to visit 3. Compared to the initial visit 1, the final visit 3 shows an impressive 15.81% increment. In contrast, SafeDrug and COGNet demonstrate a decrease in performance of 2.80% and 0.27% from visit 2 to visit 3, highlighting the superiority of MEGACare.

Table 12
Multi-task learning framework.

Model	Code-level Precision@K						Visit-level Precision@K					
	5	10	15	20	25	30	5	10	15	20	25	30
MEGACare	0.3469	0.5471	0.6784	0.7668	0.8254	0.8745	0.7492	0.6817	0.7228	0.7887	0.8392	0.8806
MEGACare$_{\text{multi}}$	0.3172	0.5013	0.6450	0.7376	0.7903	0.8363	0.6927	0.6626	0.6871	0.7580	0.8132	0.8528
Model	ADDI ↓		SDDI ↑		Jaccard ↑		F1 score ↑		PRAUC ↑		Avg. # of Med.	
MEGACare	0.0617 ± 0.0004		0.2264 ± 0.0011		0.5231 ± 0.0017		0.6796 ± 0.0013		0.7687 ± 0.0020		20.1396 ± 0.1144	
MEGACare$_{\text{multi}}$	0.0642 ± 0.0006		0.2235 ± 0.0013		0.5048 ± 0.0017		0.6534 ± 0.0021		0.7517 ± 0.0025		19.5893 ± 0.0744	

Furthermore, MEGACare can keep a low ADDI rate when providing high Jaccard.

For patient B, which consists of two visits, all models failed to make accurate medication recommendations on the first visit. Upon analyzing the DDIs between the wrong and correct medications predicted by each model, it was found that the wrong medications predicted by MEGACare had more synergistic and fewer antagonistic interactions with its correctly predicted medications than other models, providing a good constraint on the ADDI rates. In the second visit, the accuracy of the predictions of all the models increased after incorporating the historical information, which is consistent with the results of the analysis in Section 6.3. This proves again that considering the historical visit information of the patients with limited times is beneficial to medication recommendation. Moreover, this case shows how to improve the ability of medication recommendation under insufficient visits is a problem to be solved.

6.7. Multi-task learning

Through the experiments and analysis presented herein, we can extrapolate that our framework has the potential to be utilized in automated diagnostic systems for clinical applications, such as diagnosis prediction and medication recommendation tasks. The increased accuracy of our model's predictions will lead to a decrease in the amount of time clinicians spend on diagnosing and prescribing medication for common cases, allowing them to allocate more attention to the diagnosis and treatment of rare cases.

It is worth mentioning that our framework is task-agnostic and can be applied to a wide array of prediction tasks, enabling the development of a comprehensive system capable of performing clinical output prediction. For such reason, we try to develop MEGACare as a multi-task framework to achieve multiple health outcomes tasks simultaneously. To prevent one task from dominating the learning process and resulting in poor performance on the other task [69], we adjust the weights of the corresponding loss functions throughout the training process. As shown in Table 12, MEGACare$_{\text{multi}}$ denotes the performance of the MEGACare framework with multi-task learning. Unfortunately, we can see that the performance of both tasks is reduced to varying degrees compared to the performance of MEGACare for the single task, which still exhibits similar or even superior performance when compared to the baseline models.

We have investigated the cause of the suboptimal performance of our framework on multi-task learning. It can be mainly attributed to the incorporation of information that is unrelated to the current task. MEGACare fuses much information from diffident perspectives. When objecting to a single object, we can filter out useless information by a clear object loss function. On the contrary, when MEGACare is required to achieve multiple tasks with different object-based common encoded patient representations, multi-task learning could lead to the introduction of certain information that is beneficial for one task but detrimental for other tasks [70]. The hypergraph-based MEGACare lacks related information filter components.

7. Conclusion

In this paper, we proposed a novel clinical outcome prediction framework MEGACare for diagnosis prediction and medication recommendation. MEGACare utilized multi-view EHR hypergraph learning in the patient representation module and introduced external knowledge to acquire medical code embeddings with a code initialization module. The used hypergraphs structure can model high-order relationships between patients and medical codes, inherently fitting for EHR data. Additionally, MEGACare utilizes various loss functions to control and combine information from different components with distinct perspectives, thereby enabling the generation of robust patient representations. Our research is conducted on the real-world MIMIC-III dataset, and our experiments demonstrate the effectiveness of each of the proposed components, resulting in satisfactory outcomes.

Funding

This work has been supported by the Innovative Research Group of the National Natural Science Foundation of China (61721002); The National Natural Science Foundation of China under Grant (62106191); The Key Research and Development Program of Ningxia Hui Nationality Autonomous Region (2022BEG02025); The Key Research and Development Program of Shaanxi Province (2021GXLH-Z-095); The National Research Foundation Singapore under AI Singapore Programme (Award Number: AISG-GC-2019-001-2A and AISG2-TC-2022-004); The RIE2025 Industry Alignment Fund (I2101E0002 – Cisco-NUS Accelerated Digital Economy Corporate Laboratory).

CRediT authorship contribution statement

Jialun Wu: Conceptualization, Methodology, Investigation, Software, Writing – review & editing. **Kai He:** Conceptualization, Methodology, Investigation, Writing – review & editing. **Rui Mao:** Formal analysis, Writing – review & editing. **Chen Li:** Supervision, Writing – review & editing. **Erik Cambria:** Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Kai He reports financial support was provided by the National Research Foundation Singapore under AI Singapore Programme. Kai He reports financial support was provided by the RIE2025 Industry Alignment Fund.

Data availability

The authors do not have permission to share data

References

[1] F. Ma, R. Chitta, J. Zhou, Q. You, T. Sun, J. Gao, 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, 2017, pp. 1903–1911.

[2] L. Ma, J. Gao, Y. Wang, C. Zhang, J. Wang, W. Ruan, W. Tang, X. Gao, X. Ma, AdaCare: Explainable clinical health status representation learning via scale-adaptive feature extraction and recalibration, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 01, 2020, pp. 825–832.

[3] E. Choi, M.T. Bahadori, A. Schuetz, W.F. Stewart, J. Sun, Doctor AI: Predicting clinical events via recurrent neural networks, in: Machine Learning for Healthcare Conference, PMLR, 2016, pp. 301–318.

[4] X. Li, Y. Xu, H. Cui, T. Huang, D. Wang, B. Lian, W. Li, G. Qin, L. Chen, L. Xie, Prediction of synergistic anti-cancer drug combinations based on drug target network and drug induced gene expression profiles, Artif. Intell. Med. 83 (2017) 35–43.

[5] F. Cheng, I.A. Kovács, A.L. Barabási, Network-based prediction of drug combinations, Nature Commun. 10 (1) (2019) 1–11.

[6] K. He, N. Hong, S. Lapalme-Remis, Y. Lan, M. Huang, C. Li, L. Yao, Understanding the patient perspective of epilepsy treatment through text mining of online patient support groups, Epilepsy Behav. 94 (2019) 65–71.

[7] K. He, L. Yao, J. Zhang, Y. Li, C. Li, Construction of genealogical knowledge graphs from obituaries: Multitask neural network extraction system, J. Med. Internet Res. 23 (8) (2021) e25670.

[8] B. Mao, C. Jia, Y. Huang, K. He, J. Wu, T. Gong, C. Li, Uncertainty-guided mutual consistency training for semi-supervised biomedical relation extraction, in: 2022 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, IEEE, 2022, pp. 2318–2325.

[9] K. He, R. Mao, T. Gong, E. Cambria, C. Li, JCBIE: a joint continual learning neural network for biomedical information extraction, BMC Bioinformatics 23 (1) (2022) 1–20.

[10] S. Ji, T. Zhang, L. Ansari, J. Fu, P. Tiwari, E. Cambria, MentalBERT: Publicly available pretrained language models for mental healthcare, in: Proceedings of the Thirteenth Language Resources and Evaluation Conference, 2022, pp. 7184–7190.

[11] E. Choi, M.T. Bahadori, J. Sun, J. Kulas, A. Schuetz, W.F. Stewart, RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism, in: Advances in Neural Information Processing Systems, 2016, pp. 3512–3520.

[12] Y. Li, B. Qian, X. Zhang, H. Liu, Graph neural network-based diagnosis prediction, Big Data 8 (5) (2020) 379–390.

[13] Y. Li, B. Qian, X. Zhang, H. Liu, Knowledge guided diagnosis prediction via graph spatial-temporal network, in: Proceedings of the 2020 SIAM International Conference on Data Mining, SIAM, 2020, pp. 19–27.

[14] E. Choi, M.T. Bahadori, L. Song, W.F. Stewart, J. Sun, GRAM: graph-based attention model for healthcare representation learning, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 787–795.

[15] F. Ma, Q. You, H. Xiao, R. Chitta, J. Zhou, J. Gao, Kame: Knowledge-based attention model for diagnosis prediction in healthcare, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018, pp. 743–752.

[16] K. He, B. Mao, X. Zhou, Y. Li, T. Gong, C. Li, J. Wu, Knowledge enhanced coreference resolution via gated attention, in: 2022 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, IEEE, 2022, pp. 2287–2293.

[17] F. Ma, Y. Wang, H. Xiao, Y. Yuan, R. Chitta, J. Zhou, J. Gao, Incorporating medical code descriptions for diagnosis prediction in healthcare, BMC Med. Inform. Decis. Mak. 19 (6) (2019) 1–13.

[18] C. Yang, C. Xiao, F. Ma, L. Glass, J. Sun, SafeDrug: Dual molecular graph encoders for safe drug recommendations, 2021, arXiv preprint arXiv:2105.02711.

[19] A.K. Nyamabo, H. Yu, J.Y. Shi, SSI–DDI: Substructure–substructure interactions for drug–drug interaction prediction, Brief. Bioinform. (2021).

[20] N. Tishby, N. Zaslavsky, Deep learning and the information bottleneck principle, in: 2015 IEEE Information Theory Workshop, Itw, IEEE, 2015, pp. 1–5.

[21] A.E. Johnson, T.J. Pollard, L. Shen, H.L. Li-Wei, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (1) (2016) 1–9.

[22] Y. Li, X. Ma, X. Zhou, P. Cheng, K. He, C. Li, Knowledge enhanced lstm for coreference resolution on biomedical texts, Bioinformatics 37 (17) (2021) 2699–2705.

[23] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).

[25] S. Han, R. Mao, E. Cambria, Hierarchical attention network for explainable depression detection on Twitter aided by metaphor concept mappings, in: Proceedings of the 29th International Conference on Computational Linguistics, COLING, International Committee on Computational Linguistics, Gyeongju, Republic of Korea, 2022, pp. 94–104.

[26] Y. Zhang, R. Chen, J. Tang, W.F. Stewart, J. Sun, 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, 2017, pp. 1315–1324.

[27] J. Shang, C. Xiao, T. Ma, H. Li, J. Sun, GAMENet: Graph augmented memory networks for recommending medication combination, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, 2019, pp. 1126–1133.

[28] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, 2016, arXiv preprint arXiv:1609.02907.

[29] E. Choi, Z. Xu, Y. Li, M. Dusenberry, G. Flores, E. Xue, A. Dai, Learning the graphical structure of electronic health records with graph convolutional transformer, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, No. 01, 2020, pp. 606–613.

[30] J. Shang, T. Ma, C. Xiao, J. Sun, Pre-training of graph augmented transformers for medication recommendation, 2019, arXiv preprint arXiv:1906.00346.

[31] A. Bretto, Hypergraph theory, in: An Introduction. Mathematical Engineering, Springer, Cham, 2013.

[32] Y. Ma, R. Mao, Q. Lin, P. Wu, E. Cambria, Multi-source aggregated classification for stock price movement prediction, Inf. Fusion 91 (2023) 515–528.

[33] Q. Lin, R. Mao, J. Liu, F. Xu, E. Cambria, Fusing topology contexts and logical rules in language models for knowledge graph completion, Inf. Fusion 90 (2023) 253–264.

[34] J. Jiang, Y. Wei, Y. Feng, J. Cao, Y. Gao, Dynamic hypergraph neural networks, in: IJCAI, 2019, pp. 2635–2641.

[35] R. Mulas, D. Zhang, Spectral theory of Laplace operators on oriented hypergraphs, Discrete Math. 344 (6) (2021) 112372.

[36] Y. Feng, H. You, Z. Zhang, R. Ji, Y. Gao, Hypergraph neural networks, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 33, No. 01, 2019, pp. 3558–3565.

[37] N. Yadati, M. Nimishakavi, P. Yadav, V. Nitin, A. Louis, P. Talukdar, Hypergcn: A new method for training graph convolutional networks on hypergraphs, Adv. Neural Inf. Process. Syst. 32 (2019).

[38] Y. Gao, Z. Zhang, H. Lin, X. Zhao, S. Du, C. Zou, Hypergraph learning: Methods and practices, IEEE Trans. Pattern Anal. Mach. Intell. 44 (5) (2020) 2548–2566.

[39] D. Arya, D.K. Gupta, S. Rudinac, M. Worring, Hypersage: Generalizing inductive representation learning on hypergraphs, 2020, arXiv preprint arXiv:2010.04558.

[40] J. Huang, J. Yang, Unignn: a unified framework for graph and hypergraph neural networks, 2021, arXiv preprint arXiv:2105.00956.

[41] E. Chien, C. Pan, J. Peng, O. Milenkovic, You are allset: A multiset function framework for hypergraph neural networks, 2021, arXiv preprint arXiv:2106.13264.

[42] E. Alsentzer, J.R. Murphy, W. Boag, W.H. Weng, D. Jin, T. Naumann, M. McDermott, Publicly available clinical BERT embeddings, 2019, arXiv preprint arXiv:1904.03323.

[43] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C.H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics 36 (4) (2020) 1234–1240.

[44] M.J. Er, Y. Zhang, N. Wang, M. Pratama, Attention pooling-based convolutional neural network for sentence modelling, Inform. Sci. 373 (2016) 388–403.

[45] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, 2017, arXiv preprint arXiv:1710.10903.

[46] Y. An, B. Jin, X. Wei, KnowAugNet: Multi-source medical knowledge augmented medication prediction network with multi-level graph contrastive learning, 2022, arXiv preprint arXiv:2204.11736.

[47] D. Weininger, SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules, J. Chem. Inf. Comput. Sci. 28 (1) (1988) 31–36.

[48] J. Degen, C. Wegscheid-Gerlach, A. Zaliani, M. Rarey, On the art of compiling and using 'drug-like' chemical fragment spaces, ChemMedChem 3 (10) (2008) 1503.

[49] W. Li, L. Zhu, R. Mao, E. Cambria, SKIER: A symbolic knowledge integrated model for conversational emotion recognition, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2023, pp. 13121–13129.

[50] E. Jang, S. Gu, B. Poole, Categorical reparameterization with gumbel-softmax, 2016, arXiv preprint arXiv:1611.01144.

[51] Y. Chen, L. Wu, M.J. Zaki, Deep iterative and adaptive learning for graph neural networks, 2019, arXiv preprint arXiv:1912.07832.

[52] Y. Gao, Y. Feng, S. Ji, R. Ji, HGNN+: General hypergraph neural networks, IEEE Trans. Pattern Anal. Mach. Intell. 45 (3) (2023) 3181–3199.

[53] Y. Li, D. Tarlow, M. Brockschmidt, R. Zemel, Gated graph sequence neural networks, 2015, arXiv preprint arXiv:1511.05493.

[54] D. Krioukov, F. Papadopoulos, M. Kitsak, A. Vahdat, M. Boguná, Hyperbolic geometry of complex networks, Phys. Rev. E 82 (3) (2010) 036106.

[55] S. Ji, J. Ye, Linear dimensionality reduction for multi-label classification, in: Twenty-First International Joint Conference on Artificial Intelligence, 2009, pp. 1077–1082.

[56] A.A. Alemi, I. Fischer, J.V. Dillon, K. Murphy, Deep variational information bottleneck, 2016, arXiv preprint arXiv:1612.00410.

[57] Q. Zhang, S. Yu, J. Xin, B. Chen, Multi-view information bottleneck without variational approximation, in: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2022, pp. 4318–4322.

[58] W. An, H. Wang, Q. Sun, J. Xu, Q. Dai, L. Zhang, A PID controller approach for stochastic optimization of deep networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8522–8531.

[59] A.L. Goldberger, L.A. Amaral, L. Glass, J.M. Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.K. Peng, H.E. Stanley, PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals, Circulation 101 (23) (2000) e215–e220.

[60] D.S. Wishart, Y.D. Feunang, A.C. Guo, E.J. Lo, A. Marcu, J.R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, et al., DrugBank 5.0: a major update to the DrugBank database for 2018, Nucleic Acids Res. 46 (D1) (2018) D1074–D1082.

[61] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Commun. ACM 60 (6) (2017) 84–90.

[62] S. Indra, L. Wikarsa, R. Turang, Using logistic regression method to classify tweets into the selected topics, in: 2016 International Conference on Advanced Computer Science and Information Systems, Icacsis, IEEE, 2016, pp. 385–390.

[63] C. Yang, C. Xiao, L. Glass, J. Sun, Change matters: Medication change prediction with recurrent residual networks, 2021, arXiv preprint arXiv:2105.01876.

[64] R. Wu, Z. Qiu, J. Jiang, G. Qi, X. Wu, Conditional generation net for medication recommendation, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 935–945.

[65] S. Niwattanakul, J. Singthongchai, E. Naenudorn, S. Wanapu, Using of jaccard coefficient for keywords similarity, in: Proceedings of the International Multiconference of Engineers and Computer Scientists, Vol. 1, No. 6, 2013, pp. 380–384.

[66] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proceedings of the 23rd International Conference on Machine Learning, 2006, pp. 233–240.

[67] F. Gong, M. Wang, H. Wang, S. Wang, M. Liu, SMR: Medical knowledge graph embedding for safe medicine recommendation, Big Data Res. 23 (2021) 100174.

[68] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.

[69] R. Liu, G. Chen, R. Mao, E. Cambria, A multi-task learning model for gold-two-mention co-reference resolution, in: 2023 International Joint Conference on Neural Networks, IJCNN, 2023, pp. 1–9.

[70] R. Mao, X. Li, Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 35, No. 15, 2021, pp. 13534–13542.