Dual-Granularity Clustering for Medication Recommendation with Autoregressive Contextualization

Qiyu Ge, Haixu Ma, Shi Mu, Shunpan Liang†

School of Artificial Intelligence(School of Software), Yanshan University, Qinhuangdao 066004, China

School of Information Science and Engineering, Xinjiang College of Science & Technology, Korla 841000, China

Abstract

With the advancement of modern artificial intelligence technologies, the integration of Artificial Intelligence (AI) and medicine is deepening, and medication recommendation systems that utilize patients' health records to recommend medication combinations are experiencing rapid development. Existing medication recommendation systems generally focus on the connections between multiple patient visits, while neglecting the distinction of the importance of the data points within a single visit. In addition, Intra-visit temporal information is often ignored, resulting in suboptimal recommendation accuracy. To address these limitations, we propose a medication recommendation method (DualArMed) that combines Autoregressive Model and Clustering Algorithms. Our method first employs a soft clustering algorithm to process fine-grained historical visit information, followed by an attention mechanism to assign different weights to the data points within each visit. Subsequently, we input both the soft clustering results and the original data into a hierarchical clustering module, and then leverage a graph neural network to learn the integrated coarse-grained and fine-grained information. In addition, we utilize a pretrained autoregressive model to capture the contextual relationships of each data point within a single patient visit. By combining these aspects, we obtain enhanced patient representations and improve recommendation accuracy. Extensive experiments demonstrate that our model outperforms the current state-of-the-art models across multiple metrics, with an improvement in accuracy of 4.11%.

Keywords: Intelligent healthcare management; Medication recommendation; Recommender System

1. Introduction

With the advancement of modern productivity and population growth, healthcare systems are under increasing pressure. The rise in medical workload, unequal distribution of resources, and disease treatment complexity heighten the risk of physician errors. In response, AI-based medication recommendation systems [8, 24, 14] have emerged. These systems analyze patient visit data to recommend precise medication combinations, assisting physicians in decision-making. They help reduce resource disparities, ease medical workload, and improve the accuracy and efficiency of disease and treatment.

†Corresponding author: Shunpan Liang (Email: liangshunpan@ysu.edu.cn)

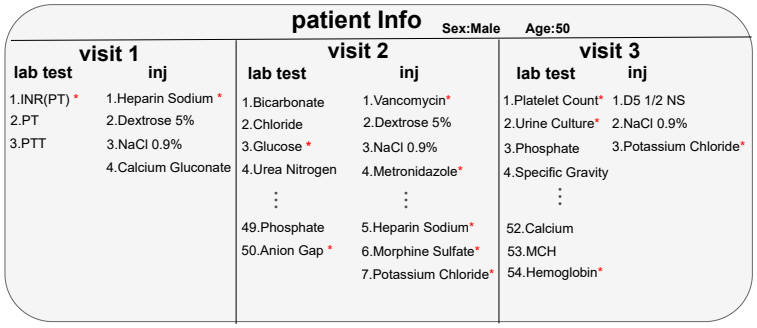


Figure 1: Detailed laboratory tests and injections of a patient in the dataset, where items marked with an asterisk (*) represent relatively important laboratory tests or injections.(The above data comes from the MIMIC-III dataset.)

Traditional recommendation systems typically focus solely on recommendation accuracy. However, medication recommendation systems face unique requirements and challenges, and thus the algorithms of other recommendation systems [27, 25, 26] cannot be directly transplanted into this domain. In the beginning, the patient's current physical condition is what the medication recommendation system's focus on [28, 6, 4], later researchers recognize the importance of patients' long-term medical histories and begin to integrate techniques from sequential recommendation systems to model patients' historical records based on temporal order [7, 30, 29], achieving significant improvements in the accuracy of medication recommendations. Although the medication recommendation system has made some achievements, existing medication recommendation systems still exhibit the following shortcomings and challenges:

Traditional medication recommendation systems mainly focus on the information reflected in patients' Electronic Health Records(EHRs) [31, 5], such as diseases, procedures, and medications, while neglecting laboratory test results and injection medication information. These supplementary data, together with their interrelationships, encompass a wealth of information indicative of the patient's physiological condition, which doctors need to consider during diagnosis and treatment. The volume of laboratory test results and injection medication information is also significantly larger compared to the data handled by traditional medication recommendation systems. Effectively mining and utilizing these two types of information has therefore become a challenge.

In recent years, researchers have focused on the relationships among visit events in patients' EHRs, while often neglecting the intrinsic connections between finer-grained data points within each visit, such as multiple laboratory results within a laboratory test[32]. In other words, Traditional medication recommendation systems frequently overlook such fine-grained information and fail to leverage its large volume and heterogeneous importance to more accurately characterize patient profiles. However, in reality, as Figure 1 shows, different data points within a visit have varying degrees of influence on the final medication recommendations.Due to the large and complex nature of fine-grained data, efficiently processing it and distinguishing data points based on their importance has become a significant challenge. Introducing clustering algorithms in this context can effectively reduce the data scale, mitigate these issues, and thereby improve model training efficiency and recommendation accuracy[33, 13].Introducing clustering algorithms could impact the final training quality of the deep learning model, thereby increasing

the precision of medication recommendations.

Furthermore, besides the inadequate consideration of the importance of different data points in each visit, the temporal relationships between the previous and later data points within each visit are also not given due consideration. In fact, the data points in each medical visit contain temporal associations. As time progresses, the health of a patient continually evolves; the condition of each moment is deeply influenced by previous states, insights that serve as the basis for predicting their future health trajectory. While this temporal relationship could be captured and utilized easily by an Autoregressive model [34, 9], previous research has often overlooked it, which in turn affects the quanity of patient representations learned and further reduces the accuracy of medication recommendations. To address these two limitations, we propose a method that combines a clustering algorithm and an Autoregressive model, named DualArMed. First, The main contributions of our method are summarized as follows:

- We apply soft clustering algorithm to separately process the fine-grained laboratory and injection data points within each patient visit, effectively distinguishing the importance of data points within each visit. This enables the model to better utilize the available data, thereby enhancing the accuracy of recommendations.

- We combine a global perspective and an internal perspective by employing a hierarchical clustering mechanism approach in conjunction with graph neural networks. This mechanism clusters each weighted visit record along with the original, unweighted data and propagates the information through a graph neural network, effectively strengthening the patient features.

- We introduce an Autoregressive model, which effectively captures the temporal relationships among data points within patient visit records and integrates these relationships into the patient representation, thereby significantly improving the accuracy of recommendations.

- We conduct extensive experiments on the real-world datasets MIMIC-III and MIMIC-IV, which demonstrate that, compared to state-of-the-art models, our method achieves a 4.11% improvement in accuracy.

We outline the different sections of the paper as follows: (1) Introduction: introduces the primary innovations and motivations behind this work; (2) Related Work: summarizes typical studies and current trends in RS-based medication recommendation; (3) Problem Definition: clearly explains the inputs and outputs as well as the specific terminologies mentioned in this paper; (4) Method: presents the core ideas of the model along with detailed technical implementations; (5) Experiments: introduces the experimental setup and results, and provides a series of supporting experiments to offer an in-depth analysis of the outcomes; and (6) Conclusion: summarizes the research findings and offers prospects for future research directions.

2. Related Work

In this section, we review past work from three perspectives: medication recommendation, clustering methods, and auto-regression.

2.1. Medication Recommendation

In recent years, with the rapid development of artificial intelligence, medication recommendation systems have garnered increasing attention. Currently, mainstream medication recommendation systems primarily focus on two aspects: enhancing medication recommendation techniques and incorporating external knowledge.

Following the tremendous success and widespread attention of deep learning in recent years, the field of medication recommendation has increasingly adopted this technology. For example, Gated Recurrent Unit (GRU)-based models employs recurrent neural networks to learn patient representations from patient records and billing codes in EHRs for medication prediction [36]; LEAP utilizes a multi-instance multi-label sequence-to-sequence model along with a recurrent decoder to simulate medication–disease relationships and train the model using Drug-Drug Interactions(DDIs) found in EHRs, ultimately making predictions based on the patient's current visit [37]; the RETAIN model leverages a dual-layer neural network, modeling longitudinal information based on recurrent neural networks (RNN) while emphasizing significant visits to extract key information [1]; GAMENet introduces a graph-enhanced memory network by integrating a DDI knowledge graph into a memory module implemented as a graph convolutional network, and it models longitudinal patient records to provide safe and personalized medication combination recommendations [19]; GRASP constructs a graph by combining current patient information with auxiliary data from similar patients, assigning different weights to each and adaptively fusing them to achieve a more comprehensive patient representation, thereby enhancing the performance of healthcare models[18]; MICRON incorporates a recurrent residual network into medication recommendation by sequentially updating patient representations through capturing differences between consecutive patient visits, thus improving recommendation accuracy and reducing training duration[3]; COGNet employs a Transformer architecture to encode both visit and medical information and designs a medication replication mechanism that determines whether to replicate past prescriptions or prescribe new medications to deliver optimal recommendations[20]; and TRANS models EHRs as a novel temporal heterogeneous graph, where the introduced temporal convolution integrates time embeddings with heterogeneous message passing, enabling the extraction of graphical structural features from individual visits while capturing complex higher-order temporal dependencies across historical visits[22]. This integration of local structure encoding with global positional encoding ultimately enhances graphical feature learning and leads to the recommendation of optimal medication combinations.HIDR does not recommend medications solely on the basis of a single patient's medical history; rather, it leverages all previous cases with analogous health profiles and precisely models inter-medication relationships through their concurrence patterns, thus improving the safety and precision of its recommendations.

Incorporating external knowledge into medication recommendation systems helps improve the accuracy and practicality of these models. Medical data are often limited and unevenly distributed, so relying solely on patient records may be insufficient to capture complex disease–medication relationships. For instance, SafeDrug incorporates molecular information of medications to represent inter-medication relationships, thereby reducing the DDI rate and enhancing both the safety and accuracy of medication recommendations[8]. MoleRec further introduces information on molecular substructures, simulating these substructures to better understand medication interactions and to provide more precise and safe medication recommendations for patients [21].

Compared with existing work, our research leverages new data and fully exploits its fine-grained features by applying an auto-regressive model to learn temporal information and using

clustering methods to differentiate the importance of data. This approach enables the mining and learning of more detailed and precise patient representations, significantly improving the accuracy of medication recommendations and achieving a more precise response to the individualized needs of patients.

2.2. Clustering algorithms

Clustering algorithms are a type of unsupervised learning algorithm used to partition a dataset into several groups. Common clustering methods can be categorized into two types: soft clustering and hard clustering [15]. Hard clustering can further be divided into partition-based clustering, hierarchical clustering, density-based clustering, grid-based clustering, and model-based clustering [43, 15]. This paper mainly introduces partition-based clustering and hierarchical clustering.

The basic idea of partition-based clustering is to treat the centers of the data as the cluster centroids of the corresponding clusters (represented by K-Means) and partition the data into non-overlapping clusters. K-Means (and its classical algorithmic formulation) is widely used due to its simplicity and relatively low time complexity, which yields high computational efficiency on many datasets [38, 44]. Hierarchical clustering (represented by agglomerative hierarchical clustering), on the other hand, analyzes data at different levels based on the similarity between clusters, forming a tree-like clustering structure; classical treatments include Johnson's hierarchical schemes and Ward's minimum-variance method [46, 47]. The theoretical advantage of hierarchical clustering is that it does not require specifying the number of clusters a priori to discover the intrinsic structure of the data. In our architecture, we leverage this flexible, bottom-up tree-building process. Rather than forcing the data into a rigid number of groups from the start (as in K-Means), we employ a dynamic agglomeration strategy and introduce a soft stopping threshold simply to guarantee structural stability for the downstream Graph Neural Network (GNN). Therefore, we use hierarchical clustering as the core method for our hard clustering module. The most classic algorithm for soft clustering is the Fuzzy C-Means (FCM) algorithm. FCM allows a data point to belong to multiple clusters simultaneously by using membership grades, unlike hard methods such as K-Means where a point belongs to only one cluster [45]. The advantage of FCM is its ability to handle overlapping clusters better, making it suitable for soft classification tasks; however, it is prone to local optima and sensitive to noise [15]. The Possibilistic Fuzzy C-Means (PFCM) algorithm is a hybrid method that combines ideas from possibilistic and fuzzy clustering to reduce sensitivity to outliers and to produce both membership and typicality (possibility) measures, improving robustness at the cost of additional computation [48, 49]. This model draws on the ideas of both FCM and PFCM, calculating the likelihood that a point belongs to different clusters to distinguish their importance.

2.3. Autoregressive Model

Sequential recommendation is often formulated as an autoregressive next-item prediction problem. Classical Markov-chain methods, such as Factorizing Personalized Markov Chains (FPMC), model the next action conditioned on recent states [35]. Recurrent neural approaches like GRU4Rec encode session or user histories in a left-to-right manner using gated recurrent units to predict the subsequent item [17]. Convolutional sequence models such as Caser capture local sequential patterns via causal convolutional filters for next-item recommendation [39], and generative convolutional networks like NextItNet use causal dilated convolutions to efficiently

model long-range dependencies for next-item generation [40]. Transformer-style sequential recommenders, for example SASRec, employ unidirectional (causal) self-attention to autoregressively predict the next interaction [41].

Considering that the Laboratory tests and Inj data used in this paper exhibit clear short-term temporal dependencies that change over time, we first pretrain a GRU model. Then, we use the pretrained GRU model to predict the future physical condition of patients, and ultimately use the predicted patient condition as part of the patient.

3. Problem Definition

In this section, we will introduce medical entities and define the problem of medication combination recommendation.

3.1. Preliminaries

Medical entities refer to the specific medical concepts that constitute medical data. Here, we use $\mathcal{D} = \{d_1, d_2, \ldots\}$, $\mathcal{P} = \{p_1, p_2, \ldots\}$, $\mathcal{M} = \{m_1, m_2, \ldots\}$, $\mathcal{L} = \{l_1, l_2, \ldots\}$, and $\mathcal{J} = \{j_1, j_2, \ldots\}$ to represent the medical entities corresponding to diseases, procedures, medications, laboratory tests, and injectable medications, respectively.

3.2. Input and Output

Our model aims to recommend medications for the patient's current condition based on the historical data of the patient in EHRs. We use $\mathcal{V}$ to represent the patient's historical records, $\mathcal{V} = \{\mathcal{V}_1, \mathcal{V}_2, \ldots, \mathcal{V}_n\}$, where n is the total number of visits, and $v_t = \{d_t, p_t, m_t, l_t, j_t\}$ represents the patient's t-th visit record, where d_i, p_i, m_i, l_i, j_i represent the i-th diseases, procedures, medications, laboratory tests, and injection the patient had during the t-th visit, respectively. By inputting $\mathcal{V}$ into the model, the output $\hat{M}_t$ represents the medications recommended to the patient in the next visit.

4. The Proposed Approach

In the entity representation part, we map the patient's diseases, procedures, and medications into embedding matrices, and then process them using a GRU to obtain the initial patient representations. In the relation mining phase, we mine finer-grained weight relationships within medical entities through soft clustering and re-weight the medical entities. In the clustering learning phase, we perform clustering based on the similarity between the re-weighted and original medical entities, and put the clustering results into the GNN to generate new patient representations. Meanwhile, we use a pretrained N-order autoregressive model to provide preliminary medication recommendations based on the patient's historical medical records. Finally, we concatenate the three types of patient representations obtained from clustering, entity representations, and the autoregressive model, thereby producing the optimal recommendation results.Figure 2 illustrates our model.For ease of reading, we list important notations and their explanations in Table 1 of our paper.

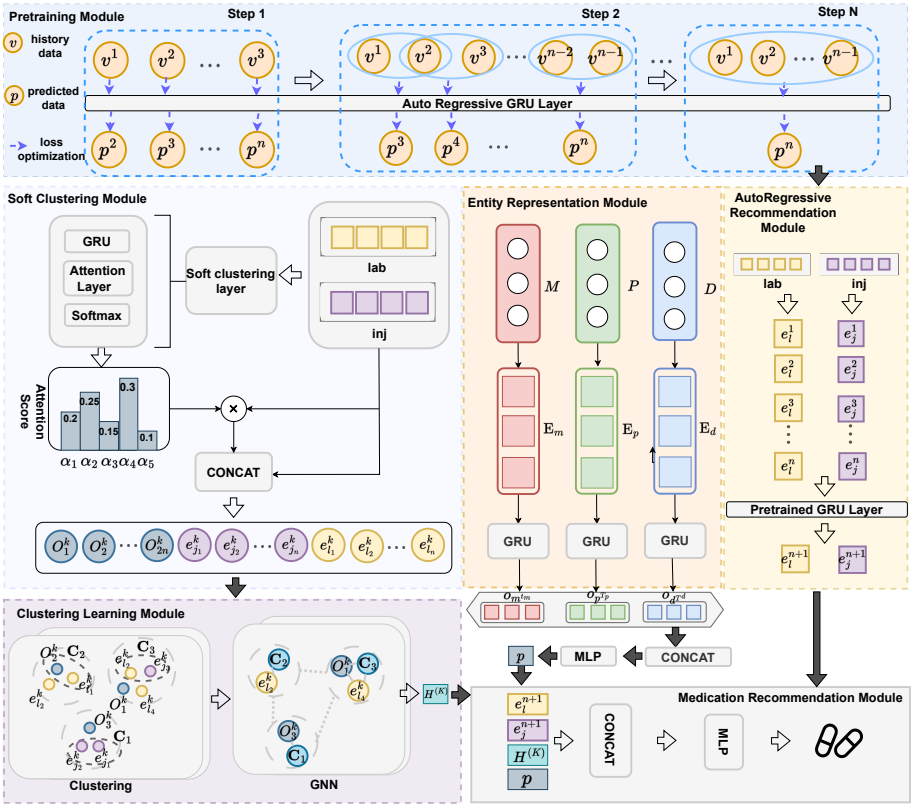


Figure 2: The proposed framework consists of five components: (1)the Entity Representation module transforms medical entities into embeddings and employs a GRU to capture their sequential features; (2)the Relationship Mining module distinguishes the importance between laboratory tests and inpatient visit data and assigns appropriate weights; (3)the Clustering Learning module processes both raw and weighted data through hierarchical clustering and leverages a graph neural network to propagate information based on similarity for patient representation; (4)the Autoregressive Recommendation module predicts future laboratory tests and injection events with a pretrained autoregressive model; (5)and finally, the Medication Prediction module integrates the learned patient representations to generate personalized medication recommendations.

Table 1: Key Mathematical Notations and Definitions in the DualArMed Model

Notation	Description
	Time and Index Notations
t	Time step or index for longitudinal macro-visits
τ	Time step or index for intra-visit micro-event sequences
k	Index for soft clustering centers
s	Index for merging steps in agglomerative hierarchical clustering
L	Layer index for Graph Neural Networks (GNN)
i, i'	Index for medical entities (e.g., the i-th positive and i'-th negative medication)
	Sets and Input Features
v_t	The t-th historical visit record of a patient
$\mathcal{M}$	The set of medication entities
e_d, e_p, e_m	Embedding vectors for diseases, procedures, and historical medications
e_l, e_j	Embedding vectors for laboratory tests and injections
	Intermediate Hidden States and Feature Representations
o_t	Concatenated basic hidden state vector for the t-th visit after GRU processing
p_t	Initial (coarse-grained) patient representation for the t-th visit
c_t^k	The k-th soft cluster center of the t-th visit
$\overline{w}_\tau^k$	Adaptive weight for soft clustering of fine-grained data points
$\alpha_l^\tau, \alpha_j^\tau$	Attention scores for the τ-th data point in laboratory test and injection sequences
$output_l^t, output_j^t$	Final weighted representation output of fine-grained data for the t-th visit
H^K	Fine-grained feature representation matrix generated after GNN propagation
$\hat{e}_i^t, \hat{e}_j^t$	Intra-visit temporal prediction feature vector output by the autoregressive model
h_t	Final comprehensive representation of the t-th visit after feature fusion
	Model Prediction and Training
$y_i, \hat{y}_i$	Ground-truth prescription label (0 or 1) and predicted probability for the i-th medication
δ	Probability threshold for determining the final medication recommendation
$\mathcal{L}_{\text{bce}}$	Binary Cross-Entropy loss for optimizing single-label prediction accuracy
$\mathcal{L}_{\text{multi}}$	Margin-based Ranking Loss for optimizing the medication recommendation ranking
β	Hyperparameter balancing the two aforementioned loss functions

4.1. Entity Representation

In this stage, we encode each visit record. Given the t-th patient's health record

$$\mathcal{V}_t = \{\mathcal{D}_t, \mathcal{P}_t, \mathcal{M}_t, \mathcal{L}_t, \mathcal{J}_t\}$$

, We design five embedding matrices E_d, E_p, E_m, E_l, E_j to obtain embeddings for each type of medical entity.

$$e_{d^t} = d^t \mathrm{E}_d, \quad e_{p^t} = p^t \mathrm{E}_p, \quad e_{m^t} = m^t \mathrm{E}_m, \tag{1}$$

$$e_{l^t} = l^t \mathrm{E}_l, \quad e_{j^t} = j^t \mathrm{E}_j \tag{2}$$

Where, $d^t \subset D_t$, $p^t \subset P_t$, $m^t \subset M_{t-1}$, $l^t \subset \mathcal{L}_t$, and $j^t \subset \mathcal{J}_t$ each is one-hot vector, consisting of 0 or 1, representing the t-th disease, procedure, medication, laboratory test and injection. Subsequently, given the robust capability of GRU in managing complex sequential challenges, this study employs GRU to analyze historical records, aiming to utilize the sequential dependencies inherent in patients' historical diseases, procedures, and medication to find the developing trends of patient's health. Specifically, we use three independent GRU to obtain hidden states and output vectors.The output states for each medical entity are computed using GRU as follows:

$$o_{d^t} = \mathrm{GRU}(o_{d^{t-1}}, e_{d^t}) \tag{3}$$

$$o_{p^t} = \mathrm{GRU}(o_{p^{t-1}}, e_{p^t}) \tag{4}$$

$$o_{m^t} = \mathrm{GRU}(o_{m^{t-1}}, e_{m^t}) \tag{5}$$

where $o_{d^{t-1}}$ is the intermediate output vector produced by the GRU for the disease entity. We initialize o_{d^0} as a zero vector. Similarly, $o_{p^{t-1}}$ and $o_{m^{t-1}}$ follow the same logic for procedures and medications, respectively.

Subsequently, we concatenate the final output vectors and then we use a Multi-Layer Perceptron (MLP) to convert the dimension of the vector to the number of medications. This concatenation serves as part of the final patient representation.

$$o_t = \mathrm{CONCAT}(o_{d^{(T_d)}}, o_{p^{(T_p)}}, o_{m^{(T_m)}})$$

$$p = \mathrm{MLP(CONCAT)}(o_\tau)$$

4.2. Relationship Mining

Considering the complexity of laboratory tests and injections in each visit compared to disease, procedures, and historical medication information, we employ a soft clustering approach at this stage to process the finer-grained laboratory tests and injections. Additionally, we utilize an attention mechanism to better differentiate the importance of the processed data, thereby enhancing patient representation for improved learning.

First, we use MLP layers to transform the raw laboratory tests and injection information as follows:

$$e^*_{l^\tau} = MLP(e_{l^\tau}), \quad e^*_{j^\tau} = MLP(e_{j^\tau}) \tag{6}$$

We then cluster the laboratory tests and injections in each visit into K clusters based on their feature similarities, where K means the total number of clusters and the center of the k-th cluster

in the t-th visit is c_t^k. e_τ^* means the raw embedding vector of the τ-th data point in the visit, $\bar{w}_t^k$ is a learned parameter, representing the weight of τ-th cluster center.

$$\min \quad \sum_{\tau=1}^{N} \left\| e_\tau^* - \sum_{k=1}^{K} \bar{w}_t^k c_t^k \right\|_2^2 \tag{7}$$

$$\text{s.t.} \quad \sum_{k=1}^{K} \bar{w}_t^k = 1, \quad \forall \tau \tag{8}$$

$$\bar{e} = \text{MLP}(\bar{w}_t^k c_t^k) \tag{9}$$

The sum of the weights of the k-th clusters in the t-th visit should be equal to one, as shown above.

We introduce a soft-clustering framework to reconcile the dual objectives of feature amplification and noise suppression. Initially, an MLP projects each scalar feature into a K-dimensional embedding, which facilitates subsequent soft-clustering operations. A learnable weight vector assigns importance to each cluster center, and the weighted clusters' sum strives to approximate the original input with minimal reconstruction error. The resulting cluster-weighted representations are concatenated and passed through a secondary MLP layer, yielding a denoised, feature-expanded output $\bar{e}$. This two-stage process enables the model to highlight subtle, informative signals in laboratory tests and injections while mitigating the amplification of spurious noise.

After obtaining the processed information through training, we proceed to use GRU to process the data:

$$h_l^\tau = \text{GRU}(\bar{e}_l^\tau,\ h_l^{\tau-1}) \tag{10}$$

$$h_j^\tau = \text{GRU}(\bar{e}_j^\tau,\ h_j^{\tau-1}) \tag{11}$$

We then element-wise multiply the representation $h^{\tau-1}$ from the last time step with each time step's output to obtain similarity scores:

$$s_l^\tau = \langle h_l^\tau,\ h_l^T \rangle, \tag{12}$$

$$s_j^\tau = \langle h_j^\tau,\ h_j^T \rangle. \tag{13}$$

These scores reflect the similarity and relative relevance of different laboratory tests and injections to the final patient state in each visit. After normalizing these similarity scores across all time steps with the softmax function, we obtain the final attention weights:

$$\alpha_l^\tau = \text{softmax}(s_l^\tau), \tag{14}$$

$$\alpha_j^\tau = \text{softmax}(s_j^\tau) \tag{15}$$

we multiply these normalized attention weights with the clustered laboratory tests and injection to obtain the final weighted data:

$$\text{output}_l = \sum_{\tau=1}^{T} \alpha_l^\tau \cdot \bar{e}_l^\tau \tag{16}$$

$$\text{output}_j = \sum_{\tau=1}^{T} \alpha_j^\tau \cdot \bar{e}_j^\tau \tag{17}$$

4.3. Clustering Learning

In this part, we input the weighted laboratory tests and injections obtained from relation mining stage, along with the unweighted original data which has not been processed through the above stage, into a hierarchical clustering module. By combining features from complementary perspectives—namely, the fine-grained, detail-amplified embeddings obtained via soft-clustering and the original, coarse-grained representations—we comprehensively capture both internal and inter-class relationships in the laboratory tests and injections, thereby generating richer, more informative features. Next, the cluster centers obtained from hierarchical clustering are input into a Graph Neural Network (GNN) [42, 16], which integrates the features of different cluster centers through its propagation mechanism, enabling the model to capture global relationships between the cluster centers at a higher level. Ultimately, this method generates a more precise patient representation, providing a more effective feature foundation for subsequent medication recommendation tasks.

First, we concatenate the weighted laboratory tests and injections with the original embedding data as follows:

$$m = \text{CONCAT}(\text{output}_l, \text{output}_j, e_l, e_j) \tag{18}$$

Afterwards, we perform agglomerative hierarchical clustering on the concatenated data[15]. Initially, each data point is treated as an independent cluster, N is the number of data point.

$$C^0 = \{\{\mathbf{m}_1\}, \{\mathbf{m}_2\}, \ldots, \{\mathbf{m}_N\}\} \tag{19}$$

Subsequently, at step s, the two data points with the smallest Euclidean distance are merged into a single cluster. While agglomerative hierarchical clustering theoretically proceeds until a single root is formed, we halt the merging process when a preset safety threshold of clusters (denoted as k) is reached:

$$(a^*, b^*) = \arg\min_{a \neq b} D(C_a^s, C_b^s) \tag{20}$$

$$C^{s+1} = (C^s \setminus \{C_{a^*}, C_{b^*}\}) \cup \{C_{a^*} \cup C_{b^*}\} \tag{21}$$

As the above matrix shows, we denote by C^t the current set of clusters and by C_{a^*} and C_{b^*} the two clusters selected for merging (those minimizing the distance between clusters). This early-stopping mechanism acts as a lower-bound threshold rather than a rigid initial cluster assignment. It effectively prevents the algorithm from over-merging the data into too few groups, ensuring that a sufficient number of distinct, informative cluster nodes are retained to construct a robust graph for the subsequent GNN propagation. Here, the notation $\setminus$ indicates set difference (i.e. deletion of the two chosen clusters) and $\cup$ denotes the union operation that forms the new merged cluster.

This mechanism ensures that exactly one cluster is eliminated at each step while preserving all remaining data points. The merging iterations continue until the algorithm terminates, ultimately yielding several disjoint clusters.

Following clustering, we use GNNs to further process the data to better merge the information of each cluster node. GNNs iteratively update node features using a neighborhood aggregation strategy, i.e., updating a node's representation based on the representations of its neighbors. The update process is defined as follows:

$$a^L(u) = \text{AGG}^L\left(\{h_n^{L-1}(v) \mid v \in N(u)\}\right) \tag{22}$$

$$h_n^L(u) = \text{COMBINE}^L\left(a^L(u), h_n^{L-1}(u)\right) \tag{23}$$

When constructing the graph, for each cluster center, we select the n closest cluster centers and connect them to facilitate propagation in the GNN. Here, $h_n^{(k)}(v)$ represents the node feature at the k-th layer, with $h_n^{(0)}(v)$ being the initial node feature, i.e., the feature of the cluster center. $N(u)$ denotes the set of neighbor nodes for each node; in this case, we select the two closest nodes to each cluster center as its neighbors.Where $\text{AGG}^{(k)}(\cdot)$ aggregates information from the neighboring cluster centers of node u to obtain the neighborhood embedding $a^{(k)}(u)$, while $\text{COMBINE}^{(k)}(\cdot)$ integrates the aggregated neighborhood information with the previous representation of node u to generate the updated node embedding $h_n^{(k)}(u)$. Through this iterative message-passing process, each cluster center can capture both its own characteristics and the contextual information from related clusters, resulting in more informative representations for subsequent medication recommendation.

After propagation through the GNN, the output node features are:

$$H^K = [h_1^L, h_2^L, \ldots, h_n^L] \tag{24}$$

Finally, we concatenate all the output node features to form part of the final patient representation.

4.4. Auto-Regressive Recommendation

Evidently, considering only the relative importance of features within a single medical visit is inadequate; it is equally crucial to account for the temporal evolution of a patient's health indicators. To address this limitation, we adopt an autoregressive framework. Leveraging the high-frequency, strictly time-sequenced continuous laboratory tests and injections typical of ICU patients, this framework captures the dynamic trends of patient indicators exclusively within a visit.

Unlike traditional linear autoregressive models, we employ GRU to intrinsically accumulate historical information. We formulate the pre-training phase as a sequence prediction task: the model learns to predict the 2nd time step from the 1st, the 3rd from the 1st and 2nd, generalizing this mechanism to predict the n-th step based on the preceding $n-1$ steps. Specifically, for a sequence of laboratory test representations $\{e_l^{(1)}, e_l^{(2)}, \ldots, e_l^{(n)}\}$ and injection representations $\{e_j^{(1)}, e_j^{(2)}, \ldots, e_j^{(n)}\}$, the GRU updates its hidden state at each time step τ ($1 \le \tau \le n-1$):

$$h_l^\tau = \text{GRU}_l(e_l^\tau, h_l^{\tau-1}) \tag{25}$$

$$h_j^\tau = \text{GRU}_j(e_j^\tau, h_j^{\tau-1}) \tag{26}$$

where h_l^τ and h_j^τ represent the encoded historical states that compress the medical sequence information up to time τ. Subsequently, these hidden states are utilized to predict the indicator representations of the next time step $\tau + 1$:

$$\hat{e}_l^{\tau+1} = W_l h_l^\tau + b_l \tag{27}$$

$$\hat{e}_j^{\tau+1} = W_j h_j^\tau + b_j \tag{28}$$

in which $W_{\{l,j\}}$ and $b_{\{l,j\}}$ denote the learnable weight matrices and biases of the prediction layers, respectively. To optimize the autoregressive model during this pre-training stage, we minimize the Mean Squared Error (MSE) between the predicted representations and the actual observed representations across all time steps. The pre-training loss function $\mathcal{L}_{pre}$ is formulated as:

$$\mathcal{L}_{pre} = \frac{1}{n-1} \sum_{\tau=1}^{n-1} \left(\|\hat{e}_l^{\tau+1} - e_l^{\tau+1}\|_2^2 + \|\hat{e}_j^{\tau+1} - e_j^{\tau+1}\|_2^2 \right) \tag{29}$$

Through this autoregressive pre-training over all available historical data, the model effectively captures the high-order temporal dependencies of patient health trajectories. Once the pre-training is complete, the parameters of this autoregressive model are strictly frozen and are not updated during the subsequent downstream training. The motivation for this freezing strategy is that the pre-training phase has already learned continuous temporal representations with distinct physiological evolution significance. If we were to open these parameters to end-to-end joint training, the gradient updates driven by discrete downstream label classification would likely cause 'catastrophic forgetting' (representation collapse) of these temporal features. Freezing these parameters effectively decouples the optimization objectives and significantly reduces the risk of overfitting.

4.5. Medication Recommendation

In this section, we concatenate the coarse-grained patient representation obtained from entity representation with the fine-grained patient representation derived from clustering to form a unified patient representation. This representation is passed through an MLP layer with an activation function to convert it into recommendation scores. These scores are combined with the recommendation scores from the autoregressive model using a trainable parameter. Medications m_i with recommendation scores exceeding the threshold δ are recommended to the patient.

$$h_t = \text{CONCAT}(p, H^K, \hat{e}_l^{n+1}, \hat{e}_j^{n+1}) \tag{30}$$

$$score = \sigma(\text{MLP}(h_t)) \tag{31}$$

$$\hat{m}_i = \begin{cases} 1, & \text{if } score_{m_i} \geqslant \delta \\ 0, & \text{if } score_{m_i} < \delta \end{cases} \tag{32}$$

4.6. Model Training

We treat our medication recommendation as a multi-label classification task and use two loss functions to jointly optimize the trainable parameters. It is crucial to note that while the GRUs in the Entity Representation module and the downstream MLPs are updated during this phase,

the pre-trained Autoregressive GRU parameters remain strictly frozen. The first loss function is Binary Cross-Entropy (BCE), which is used to measure the independent prediction accuracy of each medication label. The second is Margin-based Ranking Loss, which is used to optimize the predicted ranking of medication labels. The final loss function is a weighted sum of these two losses.

$$\begin{cases} \mathcal{L}_{\text{bce}} = -\sum_{i=1}^{|\mathcal{M}|} \left[m_i \log(\hat{m}_i) + (1 - m_i) \log(1 - \hat{m}_i) \right], \\ \mathcal{L}_{\text{multi}} = \frac{1}{|\mathcal{M}|} \sum_{i,i';m_i=1,m_{i'}=0} \max(0, 1 - (\hat{m}_i - \hat{m}_{i'})) \end{cases}$$

where m_i is the ground-truth value for medication i in the disease, and $\hat{m_i}$ is our model's predicted value for the medication. In the loss combination phase, we balance the importance of each loss function through a weighted sum:

$$L = \beta L_{\text{bce}} + (1 - \beta) L_{\text{multi}},$$

where β is a hyperparameter.

5. Experiments

In this section, we present experiments designed to address the following research questions:

- **RQ1:** Does DualArMed outperform SOTA models in medication recommendation task?
- **RQ2:** Do the key components of DualArMed enhance performance of medication recommendation?
- **RQ3:** How does DualArMed perform in terms of computational efficiency and scalability?
- **RQ4:** How do hyperparameters influence the performance of DualArMed?

5.1. Datasets

As shown in the Table 2, this study employs the MIMIC-III [10] and MIMIC-IV [11] datasets for model training and testing. MIMIC-III and MIMIC-IV are widely used medical datasets for researching and analyzing clinical data from intensive care units, including clinical records, physiological monitoring, laboratory results, medication records, and medical images of tens of thousands of ICU patients. We split our dataset into training, validation, and test sets in the proportions of 3/4, 1/10, and 3/20, respectively, based on the extracted data. We adopt a simple and effective approach for data preprocessing, where we sequentially assign numerical values to patients' diagnoses, procedures, specific medication names, laboratory tests, and injections. Each patient may have multiple visits, with each visit containing sets of multiple diagnoses, procedures, medications, laboratory data, and injection data.To ensure a fair and leakage-free evaluation, we perform the dataset split at the patient level rather than the visit level, such that all visits of a single patient are assigned exclusively to one of the training, validation, or test sets. This prevents information leakage across different data splits.

During model development, the validation set is exclusively used for hyperparameter tuning, including learning rate, dropout rate, and acceptance threshold selection. The test set is strictly

Table 2: Statistics of the datasets.

Item	MIMIC-III	MIMIC-IV
# patients	15,407	43170
# visits events	18,557	54166
# diseases	273	277
# procedures	204	222
# medications	195	200
# lab items	669	841
# inj items	279	325
avg. # of visit	1.2045	1.2547
avg. # of diseases	12.7193	14.7097
avg. # of procedures	3.3714	3.2842
avg. # of medications	34.2526	35.8490

held out and is only used for final performance evaluation after all model configurations have been determined.

This protocol ensures that the reported results reflect the true generalization ability of the proposed model and avoids potential data leakage.

5.2. Baselines

To evaluate our model, we select the following high-performance models for comparison with our proposed approach:

- **RETAIN** [1]: A sequence data analysis model based on attention mechanisms, RETAIN integrates attention mechanisms with time-series information. It trains an RNN in reverse temporal order and employs a two-level attention mechanism to identify critical past clinical events, enabling more accurate medication combination recommendations for patients.
- **Transformer** [2]: This model introduces a positional encoder based on self-attention mechanisms, addressing the long-range dependency and sequential computation issues of traditional RNNs. It is well-suited for capturing long-sequence medical records of patients in EHRs.
- **GRASP**[18]: A general framework for healthcare models, GRASP enhances patient representation learning by leveraging information from patients with similar conditions, thereby improving the accuracy of recommendations.
- **GAMENet**[19]: A medication recommendation model incorporating graph neural networks, GAMENet uses a memory module with graph convolutional networks to integrate DDI relationships. It models longitudinal patient records as queries, providing safer and more personalized medication recommendations.
- **SafeDrug**[8]: This model comprehensively encodes the connectivity and functionality of medication molecules, utilizing knowledge of medication-related molecular structures to simulate medication interactions. It effectively reduces the DDI rate, enhancing the safety of recommendations.

Medication Recommendation								
Model	**MIMIC-III**				**MIMIC-IV**			
	F1-score	**Jaccard**	**PR-AUC**	**ROC-AUC**	**F1-score**	**Jaccard**	**PR-AUC**	**ROC-AUC**
RETAIN (2016)	60.26 (3.3)	43.12 (3.4)	71.61 (3.8)	90.87 (1.2)	62.18 (1.8)	45.12 (1.9)	73.25 (2.0)	91.81 (0.5)
Transformer (2017)	59.75 (1.3)	42.52 (1.5)	73.95 (3.0)	92.08 (0.9)	58.11 (3.5)	40.96 (3.5)	70.22 (3.7)	90.86 (1.2)
GRASP (2018)	62.02 (2.3)	47.49 (2.0)	76.16 (2.2)	92.89 (1.5)	63.01 (2.4)	47.53 (2.3)	75.98 (2.7)	91.75 (1.5)
GAMENet (2020)	63.43 (2.4)	48.94 (1.9)	77.63 (2.1)	93.40 (1.6)	63.78 (2.6)	49.33 (2.2)	78.31 (2.5)	93.65 (1.6)
SafeDrug (2021)	59.60 (2.2)	45.04 (2.0)	75.46 (2.4)	92.34 (1.5)	59.59 (3.6)	44.90 (2.2)	75.34 (2.7)	92.20 (1.6)
Micron (2021)	61.71 (2.3)	46.98 (2.0)	75.05 (2.3)	92.61 (1.7)	62.79 (2.7)	48.33 (2.4)	77.12 (2.6)	93.18 (1.8)
MoleRec (2023)	64.44 (2.7)	49.62 (2.8)	76.77 (2.4)	92.63 (1.7)	64.85 (1.8)	50.18 (2.6)	76.88 (2.7)	92.11 (1.8)
Trans (2024)	63.49 (3.0)	46.44 (3.2)	75.73 (3.0)	91.95 (0.9)	64.13 (2.7)	47.13 (2.6)	76.14 (3.0)	92.37 (0.9)
CausalMed (2024)	65.87 (2.7)	49.93 (2.1)	76.74 (1.8)	91.74 (0.6)	64.15 (2.4)	49.11 (2.5)	77.29 (2.6)	92.69 (1.6)
REFINE (2024)	66.73 (2.6)	50.07 (2.9)	78.25 (2.6)	92.54 (0.7)	66.05 (1.0)	49.66 (1.1)	78.29 (1.5)	92.11 (0.3)
Ours	**68.53** (3.1)*	**52.13** (3.6)*	**78.56** (2.8)	**93.69** (0.7)	**70.78** (1.9)*	**54.77** (2.2)*	**80.67** (2.1)*	**94.27** (0.6)*

Table 3: Performance comparison on medication recommendation task over MIMIC-III and MIMIC-IV datasets.The top model is in **bold**, and the second-best is underlined.Models marked with an asterisk (*) exhibit significant improvements over the second-best baseline (REFINE), as determined by a paired t-test across 10 runs. Statistical significance was defined as p <0.05.

- **MICRON**[3]: Employing a residual recurrent network, MICRON does not rely on a patient's complete medical history. It requires only changes in the patient's health records as input, periodically updating hidden medication vectors and medication sets to refine recommended medication combinations. This approach improves recommendation accuracy with fewer parameters, effectively reducing training time.
- **COGNet**[20]: Utilizing a Transformer architecture for medication recommendation, COGNet retrieves information from historical diagnoses and past medications to uncover relationships between historical records and current conditions. It then determines whether to prescribe new medications or continue existing ones.
- **MoleRec**[21]: This model incorporates knowledge of molecular substructures into medication recommendation. By modeling interactions between substructures and the impact of individual substructures on a patient's health condition, MoleRec learns more precise molecular representation knowledge, thereby enhancing recommendation accuracy.
- **TRANS**[22]: This model proposes a novel patient EHR representation method based on temporal heterogeneous graphs, integrating temporal and structural information. It captures health changes through time-aware nodes and introduces a temporal graph transformer to combine temporal features with graph structure encoding, producing interpretable predictions.
- **CausalMed** [23]:employs causal discovery and causal estimation to capture point-to-point relationships between diseases, procedures, and medications, while introducing a Dynamic Self-Adaptive Attention mechanism to model patient-specific health states, in order to provide personalized and safe medication recommendations.
- **REFINE** [12] employs GNN to capture trends in laboratory test results and medication dosage adjustments, while also incorporating medication interaction severity, in order to provide personalized and safe prescription recommendations.

Model	MIMIC-III						MIMIC-IV					
	Code-Level Accuracy@k			Visit-level Precision@k			Code-Level Accuracy@k			Visit-level Precision@k		
	30	40	50	30	40	50	30	40	50	30	40	50
RETAIN	46.75 (2.3)	51.17 (1.7)	56.87 (3.1)	64.38 (2.0)	64.14 (2.2)	64.20 (2.5)	44.71 (1.8)	49.45 (2.1)	52.41 (3.0)	61.34 (2.0)	64.28 (2.3)	63.36 (2.8)
Transformer	47.43 (1.4)	52.91 (1.2)	56.78 (0.8)	64.61 (1.1)	64.19 (0.7)	64.92 (0.7)	45.91 (1.3)	50.43 (1.1)	53.70 (1.1)	63.66 (1.6)	62.06 (1.3)	62.53 (1.0)
GRASP	46.71 (1.9)	53.28 (2.4)	57.66 (2.5)	65.48 (2.0)	63.54 (2.3)	64.76 (2.7)	46.16 (2.1)	52.03 (2.3)	55.04 (2.8)	64.07 (2.1)	66.09 (2.5)	65.05 (2.9)
GAMENet	47.72 (2.2)	54.32 (2.5)	58.94 (2.1)	66.27 (2.3)	64.92 (2.1)	65.33 (2.6)	47.16 (2.0)	54.74 (2.4)	56.89 (2.9)	65.65 (2.2)	66.43 (2.1)	66.40 (2.7)
SafeDrug	46.17 (2.2)	52.62 (2.5)	59.83 (2.9)	63.39 (2.3)	62.54 (2.0)	63.81 (2.7)	46.26 (1.9)	52.65 (2.2)	54.75 (2.5)	62.32 (2.6)	63.19 (2.4)	63.55 (2.3)
Micron	47.67 (2.1)	51.87 (2.0)	54.75 (2.3)	65.55 (1.8)	64.69 (2.2)	64.70 (2.1)	45.58 (1.8)	52.11 (2.4)	55.31 (2.6)	64.54 (2.5)	63.24 (2.3)	64.25 (2.6)
MoleRec	49.94 (2.7)	58.89 (2.4)	61.28 (2.2)	65.78 (2.5)	65.53 (2.3)	65.86 (2.9)	47.81 (2.4)	52.13 (2.5)	57.87 (2.6)	64.22 (2.1)	64.50 (2.8)	65.01 (2.3)
Trans	47.15 (2.9)	52.33 (2.9)	56.35 (3.4)	64.98 (1.5)	64.12 (1.8)	65.00 (2.2)	47.05 (2.1)	51.61 (2.1)	55.16 (2.4)	65.66 (2.5)	63.91 (1.9)	64.37 (2.0)
CausalMed	50.96 (2.3)	56.17 (2.6)	60.69 (2.1)	65.87 (2.6)	63.73 (2.4)	66.00 (2.8)	46.05 (2.4)	52.96 (2.4)	59.03 (3.0)	64.90 (2.5)	66.85 (2.6)	65.86 (2.8)
REFINE	47.95 (1.6)	53.64 (1.2)	58.04 (0.9)	64.11 (1.3)	63.42 (0.7)	65.85 (0.4)	48.39 (1.4)	53.93 (1.3)	58.02 (0.8)	66.32 (0.9)	65.07 (0.8)	65.68 (0.8)
Ours	**59.45** (1.5)	**69.36** (1.1)	**77.49** (0.8)	**82.38** (1.3)	**85.06** (1.0)	**88.06** (1.1)	**57.92** (1.2)	**68.10** (1.1)	**76.43** (0.9)	**83.50** (1.2)	**85.66** (1.0)	**88.38** (0.9)

Table 4: The average performance (%) and standard deviation (in brackets) of each model for MIMIC-III and MIMIC-IV on the medication recommendation task, evaluated using recommendation system metrics. The top model is in **bold**, and the second best is underlined.

5.3. Evaluation Metrics

We utilize the following six metrics to evaluate the performance of our model: Jaccard coefficient, F1 score, ROC-AUC, PRAUC, code-level precision@K, and visit-level precision@K. These metrics provide insights into the effectiveness of our model. Below, we provide detailed explanations of each metric.

- **Jaccard (Jaccard Similarity Score)**: A simple and commonly used method to measure the similarity between two sets, calculated as the size of the intersection divided by the size of the union of the two sets. In the context of medication recommendation, a higher Jaccard similarity indicates that the predicted set of medications closely aligns with the actual prescriptions, signifying greater accuracy.

$$\text{Jaccard}(t) = \frac{|\{i : \hat{d}_i = 1\}| \cap |\{i : d_i = 1\}|}{|\{i : \hat{d}_i = 1\}| \cup |\{i : d_i = 1\}|}, \tag{33}$$

$$\text{Jaccard} = \frac{1}{N_h} \sum_{t=1}^{N_h} \text{Jaccard}(t), \tag{34}$$

  whereN_h represents the total number of visits for patient h.

- **F1 (F1-Score)**: A metric that combines precision and recall, serving as a standard measure of a model's accuracy. In medication recommendation, a higher F1 score indicates that the model effectively balances precision and recall, capturing the medications actually used by patients as comprehensively as possible.

$$\text{Precision}(t) = \frac{|\{i : \hat{d}_i = 1\} \cap \{i : d_i = 1\}|}{|\{i : \hat{d}_i = 1\}|}, \tag{35}$$

$$\text{Recall}(t) = \frac{|\{i : \hat{d}_i = 1\} \cap \{i : d_i = 1\}|}{|\{i : d_i = 1\}|}, \tag{36}$$

$$\text{F1}(t) = \frac{2}{\frac{1}{\text{Precision}(t)} + \frac{1}{\text{Recall}(t)}}, \tag{37}$$

$$\text{F1} = \frac{1}{N_h} \sum_{i=1}^{N_h} \text{F1}(i), \tag{38}$$

where $\hat{d}_i$ represents the predicted outcome, d_i represents the real label.

- **PRAUC (Precision-Recall Area Under Curve)**: The area under the precision-recall curve, used to assess machine learning model performance. It evaluates the relationship between precision and recall, reflecting the model's ability to predict minority classes. A higher PRAUC score indicates that the model maintains high precision while achieving high recall, enabling more accurate medication recommendations with a low error rate.

$$\text{PR-AUC}_t = \sum_{k=1}^{|D|} \text{Precision}_{k_t} \triangle \text{Recall}_{k_t}, \tag{39}$$

$$\triangle Recall_{k_t} = \text{Recall}_{k_t} - \text{Recall}_{k-1_t}, \tag{40}$$

where $|D|$ denotes the number of medications, k is the rank in the sequence of the retrieved medication, and $Precision_k(t)$ represents the precision at cut-off k in the ordered retrieval list and $\triangle Recall_{k_t}$ denotes the change in recall of a medication's ranking from $k-1$ to k. We averaged the PR-AUC across all of the patient's visits.

- **ROC-AUC (Receiver Operating Characteristic - Area Under Curve)**: A widely used metric that combines the false positive rate (FPR) and true positive rate (TPR) to measure a classification model's discriminative ability. In medication recommendation, a higher ROC-AUC value indicates that the model more accurately distinguishes between medications that patients need and those they do not, optimizing recommendation performance by minimizing incorrect recommendations while enhancing the identification of required medications.

$$\begin{aligned} \text{ROC-AUC} = \sum_{i=1}^{n-1} \frac{1}{2} \times (FPR_{i+1} - FPR_i) \\ \times (TPR_{i+1} + TPR_i), \end{aligned} \tag{41}$$

where FPR_i and TPR_i are the false positive rate and the true positive rate at the i threshold, respectively, and n is the number of thresholds.

- **Visit-level Precision@k** measures the per-visit accuracy of top-k predictions. For a visit v_t with ground-truth label set D_t, let the model's top-k predictions be $\{\hat{d}_1, \ldots, \hat{d}_k\}$. The visit-level precision@k for this visit is

$$\text{VisitPrecision@}k(v_t) = \frac{\sum_{i=1}^{k} \mathbf{1}\left(\hat{d}_i \in D_t\right)}{\min(k, |D_t|)}. \tag{42}$$

We report the averaged visit-level precision@k over all visits.

- **Event-level Recall@k** (also referred to as event-level coverage) measures the fraction of true events that are recovered by top-k predictions across all visits:

$$\text{EventRecall@}k = \frac{\sum_{t=1}^{|V|} \sum_{i=1}^{k} \mathbf{1}\left(\hat{d}_i \in D_t\right)}{\sum_{t=1}^{|V|} |D_t|}, \tag{43}$$

where visits with $|D_t| = 0$ are typically excluded from the denominator (or handled according to dataset convention).

5.4. Setup Protocol

In this subsection, we offer a comprehensive overview of the experimental environment setup, parameter selection, model configuration, and sampling methods for the testing phase.

5.4.1. Experimental Environment

The experiments were conducted on an Ubuntu 22.04 system equipped with 30GB of memory, 12 CPUs, and a 24GB NVIDIA RTX3090 GPU, using PyTorch 2.0.0 and CUDA 11.7.

5.4.2. Configuration and Parameter

For all entity embeddings, we set the embedding dimension to 128. We employ ReLU as the activation function for the MLP in this study and set the dropout rate to 0.7. The early-stopping safety threshold k for hierarchical clustering is set to 4 to ensure sufficient GNN node features. For the loss function, we assign β a value of 0.95 and set the acceptance threshold δ to 0.6. The training process consists of 100 epochs, utilizing the AdamW optimizer with a learning rate of 0.0005 and a regularization rate of 0.01.

5.4.3. Sampling Approach

Due to the scarcity of publicly available electronic health record (EHR) data, we adopt a bootstrap sampling approach at this stage, which is particularly suitable for scenarios with limited samples.

5.5. Performance Comparison

In this section, we compare our model with baseline models, assessing model accuracy using key metrics. Among the six metrics, F1 score, Jaccard index, PR-AUC, and ROC-AUC are used to measure the model's accuracy, while Event-level accuracy and Visit-level precision are used to evaluate the model's performance at the fine-grained and coarse-grained levels, respectively.For baseline methods with a provided test model, we utilize the baseline model directly. For methods without a provided test model, we retrain the models and perform evaluations. We use the optimal parameter settings described in the respective papers to derive results from each baseline method. Table 3 and Table 4 presents the comparative results.

We begin by comparing several earlier baseline models, which generally exhibit relatively lower performance.

It can be observed that the PR-AUC and ROC-AUC of RETAIN are lower compared to other models, indicating that although RETAIN incorporates a linear model into medication recommendation, it has relatively low ability in distinguishing between "medications that should be recommended" and "medications that should not be recommended." Compared to RETAIN, Transformer exhibits lower F1 and Jaccard scores, suggesting that while it introduces a self-attention mechanism, the model lacks sufficient balance between the positive and negative classes.

Next, we conduct a comprehensive comparison and discussion of several recent high-performance baselines.

Compared to previous models, Grasp enhances the current patient representation by utilizing relevant information from similar patients, significantly improving the accuracy of medication recommendations. However, its comprehensiveness is limited, as there is no improvement in accuracy at the top-K event/visit levels. GAMENet improves the model's accuracy by incorporating the patient's visit history into the model framework, showing a noticeable improvement

Table 5: The performance of each ablation model on the test set regarding accuracy and representation quality. The best and the runner-up results are highlighted in bold and underlined respectively.

Model	MIMIC-III				MIMIC-IV			
	Jaccard	F1	PRAUC	ROCAUC	Jaccard	F1	PRAUC	ROCAUC
DualArMed *w/o* C	0.5044	0.6706	<u>0.7839</u>	0.9340	<u>0.5303</u>	<u>0.6931</u>	<u>0.8011</u>	<u>0.9410</u>
DualArMed *w/o* Ar	<u>0.5203</u>	<u>0.6845</u>	0.7809	<u>0.9355</u>	0.5111	0.6765	0.7783	0.9317
DualArMed *w/o* C+Ar	0.5005	0.6671	0.7832	0.9335	0.5113	0.6766	0.7970	0.9411
DualArMed	**0.5213**	**0.6853**	**0.7856**	**0.9369**	**0.5477**	**0.7078**	**0.8067**	**0.9427**

at the top-K event/visit levels. SafeDrug explicitly models the relationships between medication molecules, but its accuracy has decreased compared to previous high-performance baselines. MICRON improves accuracy by adjusting the recommended medication combinations based on changes in the patient's physical condition between two visits. MoleRec integrates external molecular knowledge by establishing relationships between medication molecules, their substructures, and the patient's diseases and procedures, leading to a significant improvement in recommendation accuracy. Trans models patient information in EHRs using a spatiotemporal heterogeneous graph, capturing health changes through time-aware nodes; while its overall accuracy does not show improvement, it significantly enhances the accuracy at the top-K event/visit levels. The Refine model incorporates new laboratory test and injection data, which effectively improves the F1 and Jaccard scores, but shows no significant improvement at the top-K event/visit level.

Crucially, while many of these baselines rely heavily on cross-visit historical sequences—which are often sparse in real-world clinical datasets—they overlook the rich temporal dynamics and heterogeneous relationships strictly within a single visit. To overcome the limitations associated with sparse longitudinal records, our approach exclusively exploits this fine-grained, intra-visit data. Specifically, our model addresses the heterogeneous importance of internal records—such as laboratory tests and injections—by applying an attention mechanism to assign weights among these data points, subsequently combining them with the original data through hard clustering. Concurrently, an autoregressive mechanism is utilized to capture the high-frequency temporal dynamics strictly within a single medical visit. This comprehensive intra-visit modeling ultimately yields an enhanced patient representation that significantly improves medication recommendation accuracy. It is worth highlighting that our model performed particularly well on the more complex MIMIC-IV dataset, a result that strongly confirms the efficacy of our fine-grained data mining approach.

6. Discussions

In this section, we conduct an in-depth analysis of the experimental outcomes previously discussed. Additionally, we carry out a series of supplementary experiments to confirm the comprehensiveness and rationale of our approach, to ensure its robustness and validate its advancements in the field.

6.1. *Effectiveness Analysis*

The comparative experiments above showcase the superior performance of our model in terms of accuracy and representation quality.

Previous medication recommendation models have predominantly focused on static clinical features—namely diagnosis, procedures, and medications,while largely overlooking crucial dynamic patient information, such as historical laboratory tests and historical injection records. This omission has constrained their ability to discern both intra-visit relationships relationships (the relative importance of heterogeneous data at a single visit) and longitudinal dependencies (the evolution and correlation of data across multiple visits). Consequently, prior approaches have failed to fully represent the multifaceted nature of patient trajectories, resulting in suboptimal predictive accuracy and reduced adaptability to complex clinical scenarios.

In contrast, our framework addresses these shortcomings through a two-stage augmentation of patient representation. First, we apply soft clustering to capture the importance differences among data points within a single visit. We then perform hard clustering to effectively combine the weighted data with the original (unweighted) data, thereby integrating features from both perspectives. Subsequently, we employ an autoregressive model to integrate information across time, thereby modeling the temporal dependencies among laboratory tests and injections. By fusing these enriched cross-sectional and longitudinal embeddings, our method generates a strengthened, temporally aware patient representation that demonstrably enhances recommendation performance.

6.2. *Efficiency Analysis*

To more accurately present the model's efficiency, we perform a step-by-step analysis of the model's time complexity. In the entity representation and soft clustering modules, we use an embedding layer, an MLP, and a GRU to obtain patient representations and differentiate the importance of laboratory tests and injections within each visit. The time complexity of the embedding layer is $O(L \cdot \dim)$, where L is the length of the input sequence and dim is the embedding dimension size. The time complexity of the MLP is $O(n_{\text{in}} \cdot n_{\text{out}})$, where n_{in} and n_{out} are the input and output dimensions, respectively. The time complexity of the GRU is $O(g \cdot h^2)$, where g is the length of the input sequence and h is the hidden state dimension.In the hard clustering module, we use hierarchical clustering and a GNN to combine the weighted fine-grained information obtained from the soft clustering module with the unprocessed coarse-grained information to obtain the enhanced patient representation. Specifically, in the hierarchical clustering step, we use single-linkage agglomerative clustering, yielding a time complexity of $O(N_c^2 \log N_c)$, where N_c is the number of input data points. The time complexity of the 1-layer GNN is $O(N + E)$, where N and E are the number of nodes and edges input to the GNN, respectively (assuming constant feature dimensions).In the autoregressive prediction module, we utilize K pretrained autoregressive models to predict the recommended medications over m steps. Assuming an autoregressive model of order p, the prediction time complexity is $O(K \cdot m \cdot p)$.Therefore, combining all components, the overall time complexity of our proposed model is:

$$O(L \cdot \dim + n_{\text{in}} \cdot n_{\text{out}} + g \cdot h^2 + N_c^2 \log N_c + (N + E) + K \cdot m \cdot p)$$

6.3. *Ablation Study*

To evaluate the contribution of our innovations to the overall performance of the model and minimize interference from external factors, we remove key modules and make logical modifications to set up several variant experiments.

To evaluate the contributions of different components of the proposed model DualArMed, as Table 5 shows, we tested their importance by removing key components.

DualArMed w/o AR: We removed the autoregressive prediction module from DualArMed. The model no longer captures the internal temporal information of each visit but instead recommends potential medications to the patient based on the importance of different internal data.

DualArMed w/o C: We removed both the soft clustering and hard clustering modules from the model. The model no longer captures fine-grained information from each visit and global coarse-grained information, but instead directly uses the relationships between medications, diagnoses, and procedures processed by the GRU, which are then integrated with the autoregressive model's predictions for recommendation.

DualArMed w/o C+AR: We removed both the clustering and autoregressive prediction modules. The model no longer leverages soft and hard clustering to capture local and global patterns, nor relies on the autoregressive module to extract temporal information from laboratory tests and injections. Instead, predictions are made solely based on the entity representation module.

The results of the ablation experiments are shown in the table above, and they align with our expectations.

DualArMed w/o Ar: The removal of autoregressive module indicates that our autoregressive module strengthens the patient representation by capturing the temporal information within fine-grained data from each visit, effectively improving the accuracy of medication recommendation. The ablation results show a performance drop, underscoring the module's critical role in modeling sequential dependencies.

DualArMed w/o C: The removal of Clustering module results show a significant drop across all evaluation metrics, indicating that compared to the full model, the removal of the clustering module leads to a loss in the model's ability to capture the internal details and global features of each visit. This results in reduced performance scores, underscoring the essential role of the integrated clustering module in maintaining overall accuracy.

DualArMed w/o C+Ar: DualArMed w/o C+AR demonstrates the combined necessity of both modules, with performance metrics dropped a substantial decline from the full model. This confirms that the absence of both the autoregressive and clustering modules severely limits the model's ability to capture temporal dynamics and refine feature representations, emphasizing their indispensable synergy in achieving optimal medication recommendation performance.

6.4. Parameter Sensitivity

We conducted parameter sensitivity experiments on the MIMIC-III dataset to investigate the effects of GNN layer depth, learning rate, batch size, and embedding dimension; the experimental results are shown in the figure.

First, we examined the effect of GNN layer depth on model performance. The results indicate that a one-layer GNN achieves the best performance: it captures the complex relationships among medications better. When the depth increases to more than one layer, performance degrades, suggesting that overly deep architectures introduce noise and exacerbate overfitting, thereby harming generalization.

Next, we analyzed the trend for learning rate. When the learning rate is set to 0.0005 the model attains the highest Jaccard score, whereas an excessively high learning rate induces instability and performance oscillation, manifested as declines in F1 and Jaccard. Overall, an appropriate learning rate is crucial for balancing convergence speed and final performance.

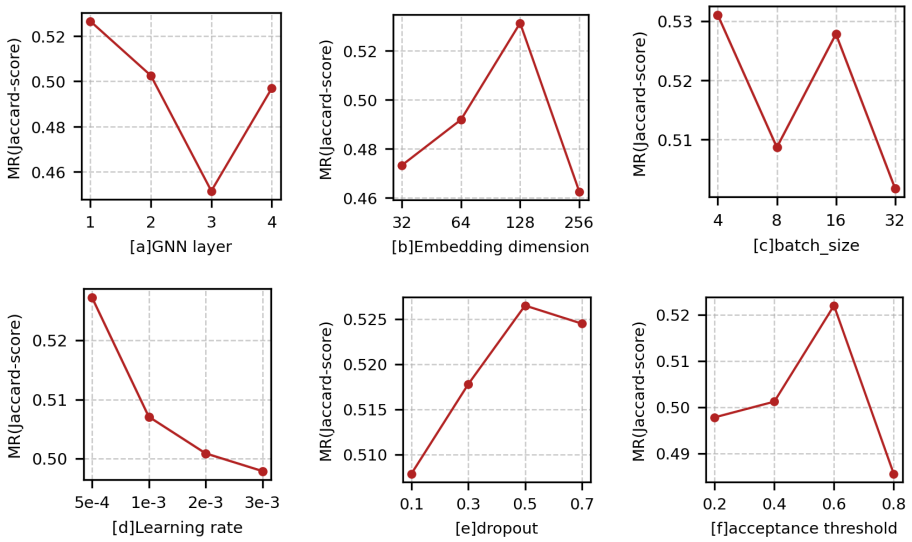


Figure 3: Experiments on parameter sensitivity on MIMIC-III dataset.

Next, we investigate the impact of batch size on the performance of the proposed model. The results indicate that while a smaller batch size of 4 achieves the best Jaccard score, the gain over a batch size of 16 is relatively marginal. In contrast, a batch size of 16 yields superior performance on PR-AUC and ROC-AUC and provides more stable gradient updates during training. Therefore, considering the small difference in Jaccard score and the overall improvement in the remaining evaluation metrics, we ultimately select a batch size of 16 as the default setting for our model.

Finally, we investigated the influence of embedding dimension on model performance. A moderate dimensionality (128) attains relatively good performance; too small a dimension (32 or 64) constrains the expressive capacity of medication features, whereas an excessively large dimension (256) introduces redundant information and increases the risk of overfitting, leading to degraded performance. Therefore, selecting an appropriate embedding dimension is essential to balance representational power and model complexity.

Furthermore, we evaluated the impact of the dropout rate on model performance. As illustrated in Figure 3(e), the Jaccard score initially improves as the dropout rate increases, reaching its peak at 0.5. This indicates that an appropriate level of dropout effectively mitigates overfitting by preventing complex co-adaptations among neurons during training. However, despite achieving the highest Jaccard score at a dropout rate of 0.5, we observed that its performance on ranking-based metrics (PR-AUC and ROC-AUC) was relatively inferior. Recognizing that medication recommendation requires a strict balance between set-level matching and precise ranking, we further evaluated higher dropout rates. Ultimately, we found that setting the dropout rate to 0.7 provided the best overall trade-off, securing competitive Jaccard scores while maintaining high stability and accuracy across PR-AUC and ROC-AUC.

Lastly, we explored the effect of the acceptance threshold, which dictates the final multi-label medication output. Figure 3(f) demonstrates that the model achieves its optimal Jaccard score at a threshold of 0.6. A lower threshold (e.g., 0.2 or 0.4) tends to introduce excessive

noise by incorporating irrelevant or less confident medications into the recommendation set, thus degrading overall accuracy. Conversely, an overly strict threshold (0.8) conservatively filters out valid medication candidates, leading to a substantial performance drop due to missed positive labels. Therefore, a threshold of 0.6 optimally balances the model's prediction sensitivity and specificity.

In summary, the best hyperparameter configuration we observed is: one GNN layer, learning rate = 0.0005, batch size = 16, embedding dimension = 128, dropout rate = 0.7 and acceptance threshold = 0.6. This configuration offers a favorable trade-off among convergence speed, predictive performance, and generalization.

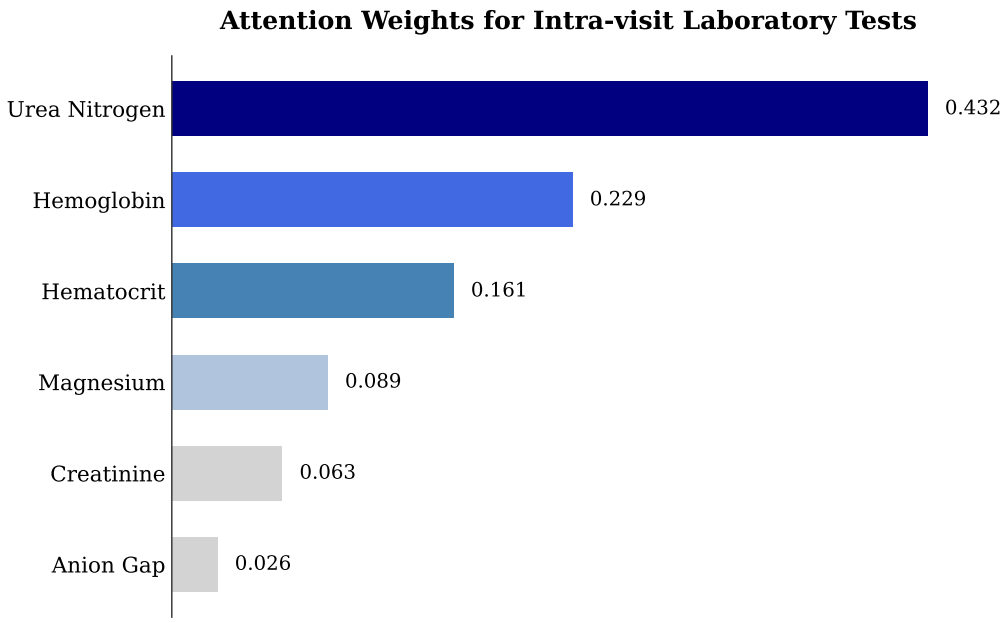


Figure 4: Attention weights assigned to various laboratory tests during a specific ICU visit.

6.5. Case Study

To intuitively demonstrate the interpretability and clinical validity of the DualArMed model, particularly the efficacy of the Relationship Mining module in differentiating the importance of fine-grained intra-visit data, we conduct a case study on a specific patient visit from the MIMIC-III dataset. This case involves a critical care patient experiencing acute renal impairment complicated by severe anemia, a common clinical presentation in ICU settings that often indicates severe fluid depletion or gastrointestinal bleeding.During this specific visit, the patient underwent a multitude of laboratory tests. Table 6 illustrates the normalized attention weights (α) assigned by the DualArMed model to these fine-grained laboratory features. As shown in Figure 4 and Table 6, the model does not treat all laboratory tests equally. Instead, it dynamically assigns the highest attention weights to Urea Nitrogen (0.432), Hemoglobin (0.229), and Hematocrit (0.161).From a clinical perspective, this weight distribution is highly logical and medically sound. Urea Nitrogen is a primary indicator of renal perfusion and intravascular volume status, while Hemoglobin

and Hematocrit are critical measures of the patient's oxygen-carrying capacity and acute blood loss. Conversely, routine monitoring metrics that are less relevant to the acute deterioration, such as the Anion Gap, receive significantly lower attention (0.026).By effectively filtering out clinical noise and focusing on the most critical physiological indicators, the soft clustering and attention mechanisms empower the model to accurately capture the patient's acute health status. When combined with the temporal dependencies extracted by the autoregressive module, DualArMed successfully leverages these prioritized features to recommend targeted interventions (e.g., specific intravenous fluid resuscitation like NaCl 0.9% and relevant medications for gastrointestinal protection or anemia management), while avoiding contraindicated drugs. This case strongly validates that the proposed dual-granularity clustering framework not only improves recommendation accuracy but also provides profound clinical interpretability.

Table 6: Attention weights assigned by DualArMed to various laboratory tests during a specific ICU visit.

Laboratory Test	**Attention Weight (α_l)**	**Clinical Relevance**
Urea Nitrogen	**0.432**	Key indicator of renal function, fluid status, and potential GI bleeding.
Hemoglobin	**0.229**	Direct measure of anemic severity and blood loss.
Hematocrit	**0.161**	Assesses the proportion of red blood cells, confirming anemia/bleeding.
Magnesium	0.089	Routine electrolyte monitoring.
Creatinine	0.063	Indicator of kidney function, evaluated alongside Urea Nitrogen.
Anion Gap	0.026	General acid-base balance indicator (clinical noise in this context).

By effectively filtering out clinical noise and focusing on the most critical physiological indicators, the soft clustering and attention mechanisms empower the model to accurately capture the patient's acute health status. When combined with the temporal dependencies extracted by the autoregressive module, DualArMed successfully leverages these prioritized features to recommend targeted interventions—such as specific intravenous fluid resuscitation (e.g., NaCl 0.9%) and relevant medications for gastrointestinal protection or anemia management—while avoiding contraindicated drugs. This case strongly validates that the proposed dual-granularity clustering framework not only improves recommendation accuracy but also provides profound clinical interpretability.

7. Conclusion and Future Work

This paper presents a medication recommendation framework that integrates fine-grained relation modeling, hierarchical clustering, graph neural propagation, and autoregressive modeling to generate clinically meaningful patient representations. First, we apply soft clustering to fine-grained intra-visit data such as laboratory tests and injections, enabling the model to identify the relative importance of different measurements and better utilize available information. Second, we design a hierarchical clustering strategy where each re-weighted visit record is clustered together with its original unweighted version, and the resulting cluster information is incorporated into a graph neural network to propagate and refine patient features across the graph. Third, we

introduce an autoregressive model to explicitly capture temporal dependencies among intra-visit data points and embed these temporal dynamics into patient representations. Experiments on public clinical datasets show that this multi-stage design achieves significant improvements in recommendation accuracy over baseline methods.

Beyond empirical gains, this work provides a practical pipeline for integrating soft clustering, hierarchical clustering, and graph neural networks, together with an autoregressive model for temporal dynamics. The proposed strategies—per-visit re-weighting, joint clustering of weighted and unweighted records, and sequential fusion—are broadly applicable to other clinical prediction and recommendation tasks.

However, it is important to note that the current framework does not explicitly model medication safety constraints, such as drug–drug interactions or adverse effects. Instead, this study focuses primarily on improving the quality of patient representation and recommendation accuracy through fine-grained intra-visit modeling and temporal dependency learning. We acknowledge this as an important limitation of the present work.

Future work will explore incorporating safety-aware mechanisms, such as integrating external medical knowledge (e.g., drug–drug interaction knowledge graphs) or designing constraint-based optimization strategies, to explicitly reduce the risk of unsafe medication combinations. In addition, we plan to scale the framework with larger foundation models, incorporate additional modalities (e.g., laboratory tests, imaging, and clinical notes), and enhance causal and counterfactual evaluation to improve robustness and real-world applicability. These directions will help advance medication recommendation systems toward more reliable and generalizable deployment in diverse clinical settings.

Author Contributions

Qiyu Ge: Conceptualization, Methodology, Software, Validation, Formal analysis, Data Curation, Investigation, Writing - Original Draft, Writing - Review & Editing, Visualization. **Haixu Ma**: Writing - Review & Editing. **Shi Mu**: Supervision, Writing - Review & Editing. **Shunpan Liang**: Supervision, Funding acquisition.

Acknowledgements

This work was supported by the funded project of the Natural Science Foundation of Hebei Province (No. F2025203076), the S&T Program of Hebei, China (No. 236Z0302G), and the Natural Science Foundation of Xinjiang Uygur Autonomous Region Funding Project, China (No. 2024D01C61).

References

[1] E. Choi, M. T. Bahadori, J. Sun, J. Kulas, A. Schuetz, and W. Stewart, "RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism," in *Adv. Neural Inf. Process. Syst.*, vol. 29, 2016.

[2] A. Vaswani *et al.*, "Attention Is All You Need," in *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[3] C. Yang, C. Xiao, L. Glass, and J. Sun, "Change Matters: Medication Change Prediction with Recurrent Residual Networks," in *Proc. 30th Int. Joint Conf. Artif. Intell. (IJCAI)*, pp. 3728–3734, 2021.

[4] M. Wang *et al.*, "Safe Medicine Recommendation via Medical Knowledge Graph Embedding," 2017, arXiv:1710.05980.

[5] J. X. Wang, D. K. Sullivan, A. C. Wells, and J. H. Chen, "ClinicNet: Machine Learning for Personalized Clinical Order Set Recommendations," *JAMIA Open*, vol. 3, no. 2, pp. 216–224, 2020.

[6] Y. An, L. Zhang, M. You, X. Tian, B. Jin, and X. Wei, "MeSIN: Multilevel Selective and Interactive Network for Medication Recommendation," *Knowl.-Based Syst.*, Art. no. 107534, 2021.
[7] Z. Zheng *et al.*, "Drug Package Recommendation via Interaction-Aware Graph Induction," 2021, arXiv:2101.00000.
[8] C. Yang, C. Xiao, F. Ma, L. Glass, and J. Sun, "SafeDrug: Dual Molecular Graph Encoders for Recommending Effective and Safe Drug Combinations," 2021, arXiv:2105.02711.
[9] H. Wu, J. Xu, J. Wang, and M. Long, "Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting," in *Adv. Neural Inf. Process. Syst.*, vol. 34, pp. 22419–22430, 2021.
[10] A. E. W. Johnson *et al.*, "MIMIC-III, a Freely Accessible Critical Care Database," *Sci. Data*, vol. 3, Art. no. 160035, 2016.
[11] A. E. W. Johnson *et al.*, "MIMIC-IV, a Freely Accessible Electronic Health Record Dataset," *Sci. Data*, vol. 10, Art. no. 1, 2023.
[12] S. Bhoi, M. L. Lee, W. Hsu, and N. C. Tan, "REFINE: A Fine-Grained Medication Recommendation System Using Deep Learning and Personalized Drug Interaction Modeling," in *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 24013–24024, 2023.
[13] Y. Liu *et al.*, "End-to-End Learnable Clustering for Intent Learning in Recommendation," in *Adv. Neural Inf. Process. Syst.*, vol. 37, pp. 5913–5949, 2024.
[14] X. Li, S. Liang, Y. Hou, and T. Ma, "StratMed: Relevance Stratification Between Biomedical Entities for Sparsity on Medication Recommendation," *Knowl.-Based Syst.*, vol. 284, Art. no. 111239, 2024.
[15] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Comput. Surv.*, vol. 31, no. 3, pp. 264–323, 1999.
[16] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The Graph Neural Network Model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, 2009.
[17] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, "Session-Based Recommendations with Recurrent Neural Networks," 2015, arXiv:1511.06939.
[18] C. Zhang *et al.*, "GRASP: Generic Framework for Health Status Representation Learning Based on Incorporating Knowledge From Similar Patients," in *Proc. AAAI Conf. Artif. Intell.*, pp. 715–723, 2021.
[19] J. Shang, C. Xiao, T. Ma, H. Li, and J. Sun, "GAMENet: Graph Augmented Memory Networks for Recommending Medication Combination," in *Proc. AAAI Conf. Artif. Intell.*, pp. 1126–1133, 2019.
[20] R. Wu, Z. Qiu, J. Jiang, G. Qi, and X. Wu, "Conditional Generation Net for Medication Recommendation," in *Proc. ACM Web Conf.*, pp. 935–945, 2022.
[21] N. Yang, K. Zeng, Q. Wu, and J. Yan, "MoleRec: Combinatorial Drug Recommendation With Substructure-Aware Molecular Representation Learning," in *Proc. ACM Web Conf.*, pp. 4075–4085, 2023.
[22] J. Chen, C. Yin, Y. Wang, and P. Zhang, "Predictive Modeling With Temporal Graphical Representation on Electronic Health Records," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, p. 5763, 2024.
[23] X. Li *et al.*, "CausalMed: Causality-Based Personalized Medication Recommendation Centered on Patient Health State," in *Proc. 33rd ACM Int. Conf. Inf. Knowl. Manag. (CIKM)*, pp. 1276–1285, 2024.
[24] Q. Chen, X. Li, K. Geng, and M. Wang, "Context-Aware Safe Medication Recommendations With Molecular Graph and DDI Graph Embedding," in *Proc. AAAI Conf. Artif. Intell.*, 2024.
[25] X. Wang, X. He, M. Wang, F. Feng, and T. S. Chua, "Neural Graph Collaborative Filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, pp. 165–174, 2019.
[26] F. Yu *et al.*, "TAGNN: Target Attentive Graph Neural Networks for Session-Based Recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, pp. 1921–1924, 2020.
[27] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An Open Architecture for Collaborative Filtering of Netnews," in *Proc. ACM Conf. Comput. Supported Cooperative Work (CSCW)*, pp. 175–186, 1994.
[28] Z. Wang, Y. Liang, and Z. Liu, "FFBDNet: Feature Fusion and Bipartite Decision Networks for Recommending Medication Combination," in *Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases (ECML PKDD)*, pp. 419–436, 2022.
[29] H. Le, T. Tran, and S. Venkatesh, "Dual Memory Neural Computer for Asynchronous Two-View Sequential Learning," 2018, arXiv:1800.00000.
[30] B. Jin *et al.*, "A Treatment Engine by Predicting Next-Period Prescriptions," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, pp. 1608–1616, 2018.
[31] Q. Chen, X. Li, K. Geng, and M. Wang, "Context-Aware Safe Medication Recommendations With Molecular Graph and DDI Graph Embedding," in *Proc. AAAI Conf. Artif. Intell.*, pp. 7053–7060, 2023.
[32] Y. Xu, S. Biswal, S. R. Deshpande, K. O. Maher, and J. Sun, "RAIM: Recurrent Attentive and Intensive Model of Multimodal Patient Monitoring Data," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, pp. 2565–2573, 2018.
[33] J. Das, S. Majumder, and K. Mali, "Clustering Techniques to Improve Scalability and Accuracy of Recommender Systems," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 29, no. 4, pp. 621–651, 2021.

[34] W. C. Kang and J. McAuley, "Self-Attentive Sequential Recommendation," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pp. 197–206, 2018.
[35] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Factorizing Personalized Markov Chains for Next-Basket Recommendation," in *Proc. 19th Int. Conf. World Wide Web (WWW)*, pp. 811–820, 2010.
[36] E. Choi, M. T. Bahadori, A. Schuetz, W. F. Stewart, and J. Sun, "Doctor AI: Predicting Clinical Events via Recurrent Neural Networks," in *Proc. Mach. Learn. Healthcare Conf.*, pp. 301–318, 2016.
[37] Y. Zhang, R. Chen, J. Tang, W. F. Stewart, and J. Sun, "LEAP: Learning to Prescribe Effective and Safe Treatment Combinations for Multimorbidity," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, pp. 1315–1324, 2017.
[38] J. B. MacQueen, "Some Methods for Classification and Analysis of Multivariate Observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, pp. 281–297, 1967.
[39] J. Tang and K. Wang, "Personalized Top-N Sequential Recommendation via Convolutional Sequence Embedding," in *Proc. 11th ACM Int. Conf. Web Search Data Mining (WSDM)*, pp. 565–573, 2018.
[40] F. Yuan *et al.*, "A Simple Convolutional Generative Network for Next-Item Recommendation (NextItNet)," in *Proc. 12th ACM Int. Conf. Web Search Data Mining (WSDM)*, pp. 582–590, 2019.
[41] W. C. Kang and J. McAuley, "Self-Attentive Sequential Recommendation (SASRec)," in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, pp. 197–206, 2018.
[42] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural Message Passing for Quantum Chemistry," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, pp. 1263–1272, 2017.
[43] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York, NY, USA: Wiley, 1990.
[44] S. Lloyd, "Least Squares Quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, pp. 129–137, 1982.
[45] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*. New York, NY, USA: Plenum Press, 1981.
[46] S. C. Johnson, "Hierarchical Clustering Schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.
[47] J. H. Ward Jr., "Hierarchical Grouping to Optimize an Objective Function," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 236–244, 1963.
[48] R. Krishnapuram and J. M. Keller, "A Possibilistic Approach to Clustering," *IEEE Trans. Fuzzy Syst.*, vol. 1, no. 2, pp. 98–110, 1993.
[49] N. R. Pal, K. Pal, J. M. Keller, and J. C. Bezdek, "A Possibilistic Fuzzy C-Means Clustering Algorithm," *IEEE Trans. Fuzzy Syst.*, vol. 13, no. 4, pp. 517–530, 2005.