Change Matters: Medication Change Prediction with Recurrent Residual Networks

Chaoqi Yang[1] , **Cao Xiao**[2] , **Lucas Glass**[2] and **Jimeng Sun**[1*]

[1]Department of Computer Science, University of Illinois Urbana Champaign
[2]Analytics Center of Excellence, IQVIA
chaoqiy2@illinois.edu, cao.xiao@iqvia.com, Lucas.Glass@iqvia.com, jimeng@illinois.edu

Abstract

Deep learning is revolutionizing predictive health-care, including recommending medications to patients with complex health conditions. Existing approaches focus on predicting all medications for the current visit, which often overlaps with medications from previous visits. A more clinically relevant task is to identify *medication changes*.

In this paper, we propose a new recurrent residual networks, named MICRON, for medication change prediction. MICRON takes the changes in patient health records as input and learns to update a hidden medication vector and the medication set recurrently with a reconstruction design. The medication vector is like the memory cell that encodes longitudinal information of medications. Unlike traditional methods that require the entire patient history for prediction, MICRON has a residual-based inference that allows for sequential updating based only on new patient features (e.g., new diagnoses in the recent visit), which is efficient.

We evaluated MICRON on real inpatient and outpatient datasets. MICRON achieves 3.5% and 7.8% relative improvements over the best baseline in F1 score, respectively. MICRON also requires fewer parameters, which significantly reduces the training time to 38.3s per epoch with $1.5\times$ speed-up.

1 Introduction

In recent years, deep learning has demonstrated initial success in potentially assisting clinical decision-making [Almirall *et al.*, 2012; Choi *et al.*, 2017; Xiao *et al.*, 2018; Mao *et al.*, 2019]. Among others, the medication recommendation task has drawn lots of research interest [Wang *et al.*, 2017; Wang *et al.*, 2019; Shang *et al.*, 2019a; Shang *et al.*, 2019b; Zhang *et al.*, 2017; Killian *et al.*, 2019; Wang *et al.*, 2018]. The common strategy of medication recommendation learns representations for medical entities (e.g., diagnoses, medications) from electronic health records, and use the learned representations to predict medications that fit the patient's health condition while avoiding adverse drug interactions.

Many existing works focus on recommending the full set of medications in a visit [Zhang *et al.*, 2017; Shang *et al.*, 2019a;
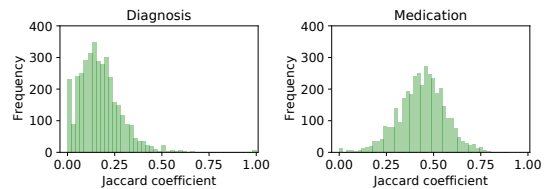

Figure 1: Histogram of Jaccard Coefficients between consecutive visits. We observe relatively weaker overlaps in diagnoses but much stronger overlaps in medications over consecutive visits. It implies that the medication change is potentially more meaningful to predict.

Shang *et al.*, 2019b; Xiao *et al.*, 2018], which can be quite redundant from previous visits. Because the medications often remain stable over time with large overlaps between consecutive visits. For example, we investigate the Jaccard coefficient over consecutive visits on *MIMIC-III* data [Johnson *et al.*, 2016] (Fig. 1). Among patients with multiple visits, although most of them are diagnosed with different conditions during consecutive visits (mean Jaccard below 0.2), the sets of medications remain stable (mean Jaccard around 0.5).

However, such medication patterns were rarely explored and leveraged to augment medication recommendation tasks. Challenges mainly arise from (1) how to accurately characterize the changes of patient health condition for each time step, and (2) how to correctly identify the medication changes based on health changes.

To fill in the gap, we propose a new recurrent residual learning approach, named MedicatIon Change pRedictiON (MICRON) to predict medication changes and simultaneously model longitudinal medical history. MICRON is enabled by the following technical contributions.

- **Efficient representation of changing health conditions**. MICRON uses a residual health representation to sequentially update changes in patient health conditions, which provides more efficient model inference than RNN-based alternatives.

- **Explicit change set prediction:** MICRON decomposes the medication change prediction task into predicting two sets: (1) the *removal set* that removes previous medicines that are no longer needed; and (2) the *addition set* that brings in new medicines for newly developed diseases. Two sets are modeled by a recurrently updated medication vector and addition and removal thresholds selected at the high-confidence

region from validation ROC-curve and thus could provide reliable inclusion-exclusion criterion.

We evaluated MICRON against state-of-the-art models on two real-world patient datasets: one inpatient *MIMIC-III* data and one outpatient dataset. MICRON outperforms the best baseline GAMENet [Shang *et al.*, 2019b] with 3.5% and 7.8% relative improvement in F1 measure, respectively. In addition, MICRON achieves $1.5\times$ speed-up in training and inference compared with GAMENet. Codes and baselines can be found here[1]. A long version of the paper is also available[2].

2 Related Works

Rule-based models [Almirall *et al.*, 2012; Chen *et al.*, 2016] typically rely on human-designed clinical guidelines, which require huge efforts from clinicians. For example, [Lakkaraju and Rudin, 2017] optimizes a sequence of if-then-else rules, which maps the patient status into the prescription decision.

Instance-based methods extract patient features only from current visits. [Zhang *et al.*, 2017] formulated the medication recommendation as a multi-instance multi-label (MIML) task and proposed a content-attention mechanism-based sequence-to-sequence model. [Wang *et al.*, 2017] jointly embedded diseases, medicines, patients, and their corresponding relations into a shared space by the knowledge graph, which requires multiple external data sources.

Longitudinal approach [Wang *et al.*, 2018; Xiao *et al.*, 2018; Bhoi *et al.*, 2020; Yang *et al.*, 2021] is a popular approach that captures the sequential dependency in patient treatment history. [Choi *et al.*, 2016; Bajor and Lasko, 2017] modeled the longitudinal patient history by RNNs for various clinical predictive tasks. [Shang *et al.*, 2019b] and [Le *et al.*, 2018] adopted memory-based networks with RNNs to handle the dependency among longitudinal medical codes. A recent work [Yang *et al.*, 2021] exploits molecule structural information to improve safe recommendation.

Compared with existing works, MICRON is based on a new perspective that focuses on predicting the changes. This is more realistic since clinicians usually update patient prescriptions by only a small proportion for a patient's new visit.

3 Method

3.1 Problem Formulation

Definition 1 (Patient EHR Records) *Patient EHR records are usually represented by an ordered sequence of tuples. For a patient j, we denote his/her clinical documentaries as $\mathbf{X}_j = [\mathbf{x}_j^{(1)}, \mathbf{x}_j^{(2)}, \mathbf{x}_j^{(3)}, \ldots]$, where the t_{th} entry, $\mathbf{x}_j^{(t)}$, records the information of the t_{th} visit, such as diagnoses, procedures and prescription information. In the paper, $\mathbf{x}_j^{(t)} = [\mathbf{d}_j^{(t)}, \mathbf{p}_j^{(t)}, \mathcal{M}_j^{(t)}]$, where $\mathbf{d}_j^{(t)} \in \{0,1\}^{|\mathcal{D}|}$ and $\mathbf{p}_j^{(t)} \in \{0,1\}^{|\mathcal{P}|}$ are multi-hot diagnoses and procedure vectors, while $\mathcal{D}$ and $\mathcal{P}$ are the overall diagnosis and procedure sets. $\mathcal{M}_j^{(t)} \subset \mathcal{M}$ is the t_{th} medication set and $\mathcal{M}$ is a set for all possible medicines. We denote the visit-wise medication addition (new) and removal (old) sets as*

[1]https://github.com/ycq091044/MICRON

[2]https://arxiv.org/abs/2105.01876

$\mathcal{N}_{target}^{(t)}, \mathcal{O}_{target}^{(t)} \subset \mathcal{M}$, *separately, which naturally follows the equality,* $\mathcal{M}^{(t)} = (\mathcal{M}^{(t-1)} \cup \mathcal{N}_{target}^{(t)}) \setminus \mathcal{O}_{target}^{(t)}$.

Problem 1 (Medication Change Prediction) *Medication change prediction aims at determining the medication addition set $\mathcal{N}^{(t)}$ and the removal set $\mathcal{O}^{(t)}$ at visit t, given the last prescription, $\tilde{\mathcal{M}}^{(t-1)}$ and patient health history $[\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(t)}]$ and $[\mathbf{p}^{(1)}, \ldots, \mathbf{p}^{(t)}]$. The model aims to minimize the gap between current estimation $\tilde{\mathcal{M}}^{(t)} = (\tilde{\mathcal{M}}^{(t-1)} \cup \mathcal{N}^{(t)}) \setminus \mathcal{O}^{(t)}$ and real prescriptions $\mathcal{M}^{(t)}$, and also control the incidence of DDIs as denoted by $\mathbf{A} \in \{0,1\}^{|\mathcal{M}|\times|\mathcal{M}|}$, where $\mathbf{A}_{ij} = 1$ implies that medicine i and j could interact.*

3.2 MICRON Method

Overview As shown in Fig. 2, MICRON has three modules: (1) a *patient representation module* that embeds diagnosis and procedure codes into latent health representation; (2) a *prescription reconstruction module (training phase)*, where MICRON trains on consecutive pairs of visits and learns residual medication representations under a new reconstruction design; (3) a *medication updating module (inference phase)* for model inference, where MICRON initializes with previous medication information. For each subsequent visit, MICRON only requires an update of patient health status and then will predict the changes in the existing medications.

The key difference between MICRON and existing medication recommendation models [Shang *et al.*, 2019b; Choi *et al.*, 2016] is that while these models learn global sequential patterns using RNNs, MICRON learns sequential information locally (by every two consecutive visits) and propagates them visit-by-visit to preserve the longitudinal patient information.

3.3 Patient Representation

Patient representation aims to learn a compact and indicative vector to represent a patient's status. In a clinical visit, doctors will recommend medications based on diagnosis and procedure information. Our module also feeds on these two features. Since MICRON is proposed for generic patients, we leave the subscript notation in the following.

Diagnosis and Procedure Encoders. For the t_{th} visit, the input features, $\mathbf{d}^{(t)} \in \mathbb{R}^{|\mathcal{D}|}$ and $\mathbf{p}^{(t)} \in \mathbb{R}^{|\mathcal{P}|}$, can be extracted from clinical documentary. $\mathbf{d}^{(t)}$ is the multi-hot diagnosis vector, while $\mathbf{p}^{(t)}$ is the procedure vector. Following the similar strategy in [Zhang *et al.*, 2017; Shang *et al.*, 2019b], we transform these two vectors into the embedding space using mapping matrices $\mathbf{E}_d \in \mathbb{R}^{s\times|\mathcal{D}|}$ and $\mathbf{E}_p \in \mathbb{R}^{s\times|\mathcal{P}|}$ (s is the size of embedding space),

$$\mathbf{d}_e^{(t)} = \mathbf{E}_d\mathbf{d}^{(t)} \quad \text{and} \quad \mathbf{p}_e^{(t)} = \mathbf{E}_p\mathbf{p}^{(t)}. \tag{1}$$

During training, these two tables are shared among all visits and patients. The results, $\mathbf{d}_e^{(t)}$ and $\mathbf{p}_e^{(t)}$, are of the same dimension $\mathbb{R}^s$.

Patient Hidden Representation. To achieve one compact health representation, $\mathbf{d}_e^{(t)}$ and $\mathbf{p}_e^{(t)}$ are further concatenated and parametrized by a *health representation network*,
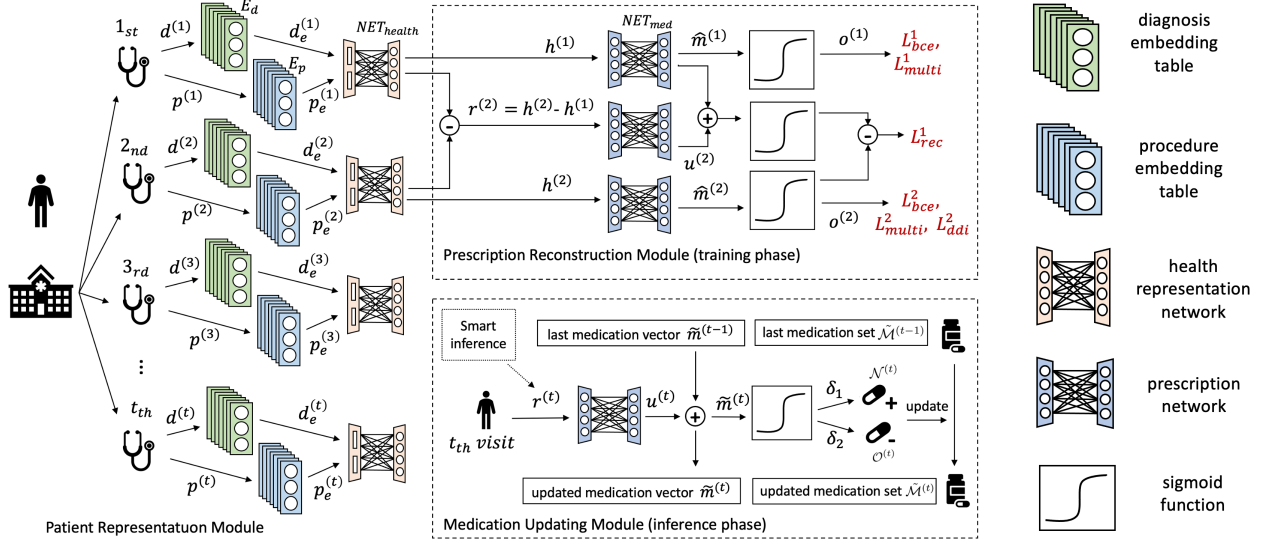
Figure 2: MICRON Framework. To represent a patient health condition, the model first feeds on diagnosis and procedure information and then generates a compact patient health representation by *health representation network*, an affine function. During training, the model uses a feed-forward network for *prescription network* and learns the residual representation under a novel reconstruction loss. In the inference, our model inputs the heath update to the same *prescription network* and then generates addition/removal sets to update the current prescription.

NET_{health},

$$\mathbf{h}^{(t)} = \text{NET}_{health} \left(\left[\mathbf{d}_e^{(t)} \parallel \mathbf{p}_e^{(t)} \right] \right), \qquad (2)$$

which outputs an integrated health representation $\mathbf{h}^{(t)} \in \mathbb{R}^s$. In this paper, we use affine function (one layer neural network without activation) for NET_{health}.

Starting from $\mathbf{h}^{(t)}$, the model architecture differs in training and inference. Next, we elaborate on these two phases.

3.4 Training: Prescription Reconstruction Module

Our MICRON trains on two consecutive visits, e.g., the $(t-1)_{th}$ and the t_{th} visits. Given the health representations, i.e., $\mathbf{h}^{(t-1)}$ and $\mathbf{h}^{(t)}$, a straightforward way for recommending medications is to learn a mapping, i.e., a *prescription network*, $\text{NET}_{med} : \mathbb{R}^s \mapsto \mathbb{R}^{|\mathcal{M}|}$, from hidden embedding space to medication space for two visits, separately.

$$\hat{\mathbf{m}}^{(t-1)} = \text{NET}_{med} \left(\mathbf{h}^{(t-1)} \right), \qquad (3)$$

$$\hat{\mathbf{m}}^{(t)} = \text{NET}_{med} \left(\mathbf{h}^{(t)} \right). \qquad (4)$$

where $\hat{\mathbf{m}}^{(t-1)}, \hat{\mathbf{m}}^{(t)} \in \mathbb{R}^{|\mathcal{M}|}$ are medication representations and each entry quantifies a real value for the corresponding medicine. In the paper, NET_{med} is implemented as a fully connected neural network. To obtain the actual recommendations, a trivial way [Shang *et al.*, 2019a; Shang *et al.*, 2019b] is to apply a *medication output layer*, which consists of a *Sigmoid* function $\sigma(\cdot)$, followed by a predefined threshold δ, picking up medicines with larger activation value. However, in this paper, we hope to utilize and emphasize the dependency usage between $\mathbf{h}^{(t-1)}$ and $\mathbf{h}^{(t)}$ in the model.

Residual Medication Representation. Formally, the difference between $\mathbf{h}^{(t-1)}$ and $\mathbf{h}^{(t)}$, i.e., $\mathbf{r}^{(t)} = \mathbf{h}^{(t)} - \mathbf{h}^{(t-1)}$, is

denoted as *residual health representation*, which encodes the changes in clinical health measurements, indicating an update in patient's health condition. Naturally, the health update $\mathbf{r}^{(t)}$ will cause an update in the resulting medication representation $\mathbf{u}^{(t)}$. Our motivation is that if NET_{med} can map a complete health representation (e.g., $\mathbf{h}^{(t)}$) into a complete medication representation (e.g., $\hat{\mathbf{m}}^{(t)}$), then a residual health representation should also be mapped into an update in the same representation space through NET_{med}. In other words, $\mathbf{r}^{(t)}$ and $\mathbf{u}^{(t)}$ shall also follow the same mapping function, NET_{med},

$$\mathbf{u}^{(t)} = \text{NET}_{med} \left(\mathbf{r}^{(t)} \right). \qquad (5)$$

To learn Eqn. (3) and (4), we could use the medication combinations in the dataset as supervision, however, it is hard to formulate a direct supervision for Eqn. (5). A simple idea is to model the addition and the removal medication sets separately (as we show in the experiment that separate modeling DualNN does not work well). Therefore, we consider reconstructing $\mathbf{u}^{(t)}$ from $\hat{\mathbf{m}}^{(t-1)}$ and $\hat{\mathbf{m}}^{(t)}$ by both unsupervised and supervised regularization.

Unsupervised Residual Reconstruction. To model the medication changes, we design a reconstruction loss. For Eqn. (3), (4) and (5), the inputs follow a residual relation: $\mathbf{h}^{(t-1)} + \mathbf{r}^{(t)} = \mathbf{h}^{(t)}$. Naturally, we also impose a similar relation in the *medication output layer* by introducing an *unsupervised reconstruction loss* ($\sigma(\cdot)$ is a *Sigmoid* function),

$$L_{rec}^{(t)} = \left\| \sigma(\hat{\mathbf{m}}^{(t-1)} + \mathbf{u}^{(t)}) - \sigma(\hat{\mathbf{m}}^{(t)}) \right\|_2, \qquad (6)$$

which is calculated with vector 2-norm. This reconstruction loss enforces the reconstructed recommendations from $\hat{\mathbf{m}}^{(t-1)}$ and the residual $\mathbf{u}^{(t)}$ to be close to the recommendations given by $\hat{\mathbf{m}}^{(t)}$. We show in the experiment that $L_{rec}^{(t)}$ is essential for learning the residual.

Supervised Multi-label Classification. To jointly modeling a low DDI output, we introduce three differentiable loss functions to improve $\hat{\mathbf{m}}^{(t-1)}$ and $\hat{\mathbf{m}}^{(t)}$, so as to achieve a better reconstruction $\mathbf{u}^{(t)}$.

- *Drug-Drug Interaction Loss.* Since adverse drug-drug interaction (DDI) is a leading cause of morbidity and mortality in clinical treatments [Percha and Altman, 2013], we penalize the presence of DDIs in the output medication representation, $\hat{\mathbf{m}}^{(t)}$. First, we transform it by *Sigmoid* function, $\hat{\mathbf{o}}^{(t)} = \sigma(\hat{\mathbf{m}}^{(t)})$, and then design the DDI loss as,

$$L_{ddi}^{(t)} = \sum_{i=1} \sum_{j=1} \mathbf{A}_{ij} \cdot \hat{\mathbf{o}}_i^{(t)} \cdot \hat{\mathbf{o}}_j^{(t)}, \quad (7)$$

where $\mathbf{A}$ is the binary DDI matrix, extracted externally [Tatonetti *et al.*, 2012] and $\mathbf{A}_{ij}$ indicates that medicine i and j have interaction or not. The term $\mathbf{A}_{ij} \cdot \hat{\mathbf{o}}_i^{(t)} \cdot \hat{\mathbf{o}}_j^{(t)}$ is the a scalar product, which is the interaction penalty for medicine i and j, and $\hat{\mathbf{o}}_i^{(t)}$ is the i-th element of the vector. Since we care about the DDI rate in the reconstructed representation, this loss only applies to the current visit t.

- *Binary Cross-entropy Loss.* In addition, we also extract real medication set as supervision. Assume a multi-hot vector $\mathbf{m}^{(t)} \in \{0,1\}^{|\mathcal{M}|}$ is the vectorization of the target medication set $\mathcal{M}^{(t)}$. We adopt binary cross entropy (BCE) loss,

$$L_{bce}^{(t)} = -\sum_{i=1} \mathbf{m}_i^{(t)} log(\hat{\mathbf{o}}_i^{(t)}) + (1-\mathbf{m}_i^{(t)})log(1-\hat{\mathbf{o}}_i^{(t)}), \quad (8)$$

where subscript i indicates each element of the vectors. For this loss function, we compute on both $\hat{\mathbf{m}}^{(t-1)}$ and $\hat{\mathbf{m}}^{(t)}$.

- *Multi-Label Margin Loss.* Then, we employ margin-based loss to enlarge the gap between the recommended medications and the unselected ones. Since $\hat{\mathbf{o}}_i^{(t)} \in (0,1)$, the margin is set 1 in our paper.

$$L_{multi}^{(t)} = \sum_{i,j: \mathbf{m}_i^{(t)}=1, \mathbf{m}_j^{(t)} \neq 1} \frac{\max(0, 1-(\hat{\mathbf{o}}_i^{(t)} - \hat{\mathbf{o}}_j^{(t)}))}{|\mathcal{M}|}. \quad (9)$$

We also consider to penalize both of the visits, i.e., calculating $L_{multi}^{(t-1)}$ and $L_{multi}^{(t)}$ using this loss.

These three losses use external supervision to optimize the *prescription network*, so that during inference, our `MICRON` would predict medication changes more accurately.

Overall Loss Function. In the training process, we hope to find optimal values for embedding tables, $\mathbf{E}_d$ and $\mathbf{E}_p$, parameter matrices in NET_{health} and NET_{med}. The loss functions are combined by weighted sum,

$$L_{total} = \lambda_1 L_{rec}^{(t)} + \lambda_2 L_{ddi}^{(t)} + \lambda_3 \left(\gamma L_{bce}^{(t)} + (1-\gamma)L_{bce}^{(t-1)} \right)$$
$$+ \lambda_4 \left(\gamma L_{multi}^{(t)} + (1-\gamma)L_{multi}^{(t-1)} \right), \quad (10)$$

where $\lambda_i,\ i = 1, 2, 3, 4$, are different weights for four types of loss functions, and γ is introduced to balance two consecutive visits. During the training, one batch contains all visits of one patient, and the loss is back-propagated after each batch. In the paper, we treat the weights as hyperparameters. In Appendix, we also prototype a momentum-based method to select the weights automatically.

3.5 Inference: Medication Updating Module

To predict medication changes, it is essential to maintain a *medication combination*. We hope that for the subsequent visit, it would be enough to derive an update in the combination based on new diagnosis or procedure information. However, like Risperdal for treating schizophrenia and bipolar disorder, some medicines will not be prescribed based on only one or two visits, and it might need long-term clinical observation. We therefore also maintain a *medication vector*, where each element quantifies the cumulative effect of a medicine. After clinical visits, each element in the vector will increase or decrease based on the updates of patient health status. Essentially, the *medication vector* is like the memory cell in RNNs, which is refreshed visit-by-visit. Once it is above or below certain thresholds, the medicine will be added or removed from the current sets. More concretely, the medicine change prediction follows three steps.

Step 1: Medication Vector Update. Specifically, for the t_{th} visit of a patient, the medication changes start from a medication vector, $\tilde{\mathbf{m}}^{(t-1)} \in \mathbb{R}^{|\mathcal{M}|}$, and a medication set, $\tilde{\mathcal{M}}^{(t-1)} \subset \mathcal{M}$. The model first updates the vector based on a residual health representation, $\mathbf{r}^{(t)}$,

$$\tilde{\mathbf{m}}^{(t)} = \tilde{\mathbf{m}}^{(t-1)} + \mathbf{u}^{(t)}$$
$$= \tilde{\mathbf{m}}^{(t-1)} + \text{NET}_{med}\left(\mathbf{r}^{(t)}\right), \quad (11)$$

where $\mathbf{r}^{(t)}$ is calculated by $\mathbf{h}^{(t)} - \mathbf{h}^{(t-1)}$ (defined in Eqn. (2), NET_{health}), which is implemented as an affine function. We use an efficient *smart inference* module to calculate $\mathbf{r}^{(t)}$, in case only the updates in medical codes (e.g., diagnosis and procedure) are accessible. We specify it in Appendix.

Step 2: Addition and Removal. Then, based on the updated medication vector, $\tilde{\mathbf{m}}^{(t)} \in \mathbb{R}^{|\mathcal{M}|}$, we identify which medicines are ready to add or remove. We design two thresholds (δ_1 is for the addition set, while δ_2 is for the removal set, where $1 \geq \delta_1 \geq \delta_2 \geq 0$) to control the size of changes. Specifically, we first apply a *Sigmoid* function $\sigma(\cdot)$, and then the addition and removal sets are generated by applying the thresholds δ_1 and δ_2 element-wise,

$$\mathcal{N}^{(t)} = \{i \mid \sigma(\tilde{\mathbf{m}}_i^{(t)}) \geq \delta_1\}, \quad (12)$$
$$\mathcal{O}^{(t)} = \{i \mid \sigma(\tilde{\mathbf{m}}_i^{(t)}) \leq \delta_2\}, \quad (13)$$

where $\mathcal{N}^{(t)}$ ($\mathcal{O}^{(t)}$) is for addition (removal) set, and subscript i enumerates the index of $\tilde{\mathbf{m}}^{(t)}$. Note that, $\mathcal{N}^{(t)} \cap \mathcal{O}^{(t)} = \varnothing$. For two thresholds, if $\delta_1 = 1$ and $\delta_2 = 0$, then $\mathcal{N}^{(t)} = \mathcal{O}^{(t)} = \varnothing$; in another case, $\delta_1 = \delta_2$, then "medication change prediction" becomes "full medication prediction".

The thresholds δ_1 and δ_2 are selected based on the receiver operating characteristic (ROC) of each medicine. Specifically, we load the pre-trained `MICRON` on the validation set. For each medicine, we collect cut-off thresholds of the ROC curve in descending order. δ_1 is the average of 5-percentile over all medications, while δ_2 is based on 95-percentile. Essentially, δ_1 will provide a low false negative (FN) rate, while δ_2 ensures a low false positive (FP) rate.

Step 3: Medication Set Update. Next, we apply the changes (addition and removal) in the existing medication set. The

generation of a new combination is given by set operations,

$$\tilde{\mathcal{M}}^{(t)} = (\tilde{\mathcal{M}}^{(t-1)} \cup \mathcal{N}^{(t)}) \setminus \mathcal{O}^{(t)}, \qquad (14)$$

where we use set union and subtraction operation. $\mathcal{N}^{(t)}$ and $\tilde{\mathcal{M}}^{(t-1)}$ could have overlaps, while $\mathcal{O}^{(t)}$ could also contain medications that are not in $\tilde{\mathcal{M}}^{(t-1)}$. The overlaps will not affect the final recommendation results due to set operations.

To sum up, the model begins with a medication vector, $\tilde{\mathbf{m}}^{(t-1)}$, and a medication set, $\tilde{\mathcal{M}}^{(t-1)}$, which are provided by the previous visit. During the current visit, `MICRON` uses the update of patient status as input and walks through the above three steps to finish one round of medication change, as well as to update $\tilde{\mathbf{m}}^{(t)}$ and $\tilde{\mathcal{M}}^{(t)}$ for the next visit.

4 Experiments

We evaluate `MICRON` against several baselines in both inpatient and outpatient datasets. We focus on answering the following questions:

- How does `MICRON` perform against the baselines in medication and change prediction?
- How does `MICRON` perform in model efficiency?
- How do different components in `MICRON` contribute to accurate recommendations?

4.1 Experimental Setup

Dataset. We consider a benchmark inpatient dataset: *MIMIC-III* [Johnson *et al.*, 2016], and a private outpatient dataset: *IQVIA PharMetrics Plus* (see processed statistics in Table 1). Details of dataset descriptions, preprocessing, hyperparameter selections can be found in Appendix.

Items	MIMIC-III	IQVIA
# of visits	14,960	30,794
# of patients	6,335	3,023
# of diagnosis codes	1,958	1,744
# of procedure codes	1,430	1,250
# of medication codes	131	155

Table 1: Statistics of Datasets

Baselines. We consider the following baselines (SimNN and DualNN are designed by ourselves).

- **SimNN** use the same patient representation $\mathbf{h}^{(t)}$ as `MICRON` and then learns a simple 3-way classifier for each medicine (add, remove, and remain) with the cross-entropy loss.
- **DualNN** also starts from patient representation $\mathbf{h}^{(t)}$ and then diverges to two different neural networks. The first one is for addition and the second for removal. Each neural network classifier uses the binary cross-entropy loss.
- **LEAP** [Zhang *et al.*, 2017] is an instance-based approach that uses a sequence to sequence model with reinforcement aftermath fine-tuning. This method generates a list of medications based on the diagnoses in the same visit.
- **RETAIN** [Choi *et al.*, 2016] is a longitudinal predictive model, which designs a specialized attention model over RNN. It learns the temporal dependencies between clinical visits and makes medication recommendations.
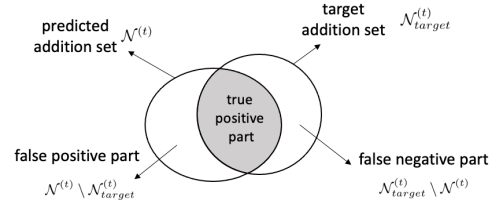

Figure 3: Illustration of Err(add) Metric at the t_{th} Visit

- **GAMENet** [Shang *et al.*, 2019b] is also a longitudinal model, which uses RNN, memory network, and graph neural network. It predicts medications using historical prescriptions as reference.

Evaluation Strategy and Metrics. We use evaluation metrics such as DDI rate, Jaccard Similarity, F1-score similar to evaluate the overall recommended medications, as other related works [Shang *et al.*, 2019b; Zhang *et al.*, 2017]. Also, we design new error metrics to evaluate the accuracy of medication changes: *Err(add)* and *Err(remove)*.

Err(add) computes the sum of false positive part and false negative part (white region in Fig. 3) between the predicted addition set $\mathcal{N}^{(t)}$ and the target addition set $\mathcal{N}^{(t)}_{target}$. The target addition set is calculated by

$$\mathcal{N}^{(t)}_{target} = \mathcal{M}^{(t)} \setminus \tilde{\mathcal{M}}^{(t-1)},$$

where $\mathcal{M}^{(t)}$ is the target medication set in the t_{th} visit, and $\tilde{\mathcal{M}}^{(t-1)}$ is the predicted medication set at the $(t-1)_{th}$ visit. For a particular patient j, this metric is calculated from the second visit, where the medication changes start,

$$\text{Err(add)}_j = \frac{1}{V(j)} \sum_{t=2}^{V(j)} \left(|\mathcal{N}^{(t)}_{target} \setminus \mathcal{N}^{(t)}| + |\mathcal{N}^{(t)} \setminus \mathcal{N}^{(t)}_{target}| \right),$$

where $\mathcal{A} \setminus \mathcal{B}$ is the set subtraction, and $V(j)$ means total number of visits of patient j. Finally, we average over all patients and get *Err(add)*. Similarly, we design *Err(remove)* to denote errors for removal. Also, we report model size and training/inference time as another two measures.

Since the evaluation is on medication change prediction, we assume the earliest visit appeared in the window as the "first" visit of a patient j, where we could extract the initial medication set, $\tilde{\mathcal{M}}^{(1)}_j = \mathcal{M}^{(1)}_j$, and the initial medication vector, $\tilde{\mathbf{m}}^{(1)}_j = \hat{\mathbf{m}}^{(1)}_j$. For a fair comparison, the evaluation of all models starts from the "second" visit. The definition of other metrics can be found in Appendix.

4.2 Experimental Results

We conduct experimental comparison based on five different random seeds and show the mean metric values in Table 2. Due to space limitation, the standard deviation results are reported in the Appendix. The result of *MIMIC-III* and *IQVIA* are reported together and separated by "/". The last two metrics, Model Size and Train(epoch), are based on *MIMIC-III* dataset. For the other five metrics, we select the best results in **bond font** and use *underscore* to select the best baseline. We also report the improvement Δ of our `MICRON` over the

Metrics	DDI	Jaccard	F1	Err(add)	Err(remove)	Model Size	Train(epoch)
SimNN	0.0837 / 0.0152	0.4658 / 0.1920	0.6235 / 0.2383	6.856 / 2.906	7.329 / 1.679	376,009 params	27.53s
DualNN	0.0925 / 0.0156	0.4880 / 0.1120	0.6447 / 0.1783	6.261 / 3.406	7.987 / 2.579	325,702 params	27.90s
RETAIN	0.0932 / 0.0279	0.4796 / 0.3320	0.6389 / 0.4215	8.552 / 2.353	6.338 / 0.927	287,940 params	34.78s
LEAP	0.0880 / **0.0134**	0.4331 / 0.1871	0.5953 / 0.2742	9.105 / 3.663	5.939 / 1.286	433,286 params	199.94s
GAMENet	0.0928 / 0.0166	0.4980 / 0.2025	0.6549 / 0.3016	8.810 / 3.016	5.854 / 2.179	449,092 params	55.31s
MICRON	**0.0695** / 0.0143	**0.5234 / 0.3634**	**0.6778 / 0.4544**	**6.090 / 2.088**	**5.853** / 1.213	275,395 params	38.83s
Δ Improve.	↓ 17.0% / ↑ 6.7%	↑ 5.1% / ↑ 9.5%	↑ 3.5% / ↑ 7.8%	↓ 2.7% / ↓ 11.3%	↓ 0.02% / ↑ 30.9%	—	—

Table 2: Performance Comparison (on *MIMIC-III / IQVIA*)

Metrics	DDI	Jaccard	F1	Err(add)	Err(remove)
MICRON w/o L_{rec}	0.0618 ± 0.0002	0.4449 ± 0.0138	0.6050 ± 0.0014	7.143 ± 0.3753	6.224 ± 0.0598
MICRON w/o $\tilde{\mathbf{m}}^{(t)}$	0.0696 ± 0.0004	0.4509 ± 0.0046	0.6096 ± 0.0368	8.496 ± 0.1401	6.463 ± 0.1468
MICRON w/o L_{multi}	0.0780 ± 0.0015	0.5020 ± 0.0072	0.6590 ± 0.0288	6.544 ± 0.2172	5.509 ± 0.0732
MICRON w/o L_{ddi}	0.0931 ± 0.0005	0.5248 ± 0.0016	0.6793 ± 0.0081	6.402 ± 0.3020	5.897 ± 0.0397
MICRON w/o δ_1, δ_2	0.0628 ± 0.0018	0.5074 ± 0.0016	0.6635 ± 0.0026	7.216 ± 0.1335	5.084 ± 0.2706
MICRON	0.0695 ± 0.0004	0.5234 ± 0.0008	0.6778 ± 0.0007	6.090 ± 0.0189	5.853 ± 0.0219

Table 3: Ablation Study for Different Model Components (on *MIMIC-III*)

best baseline. For *DDI, Err(add), Err(remove)*, the lower the better, while for *Jaccard* and *F1*, the higher the better.

MICRON outperforms most baselines in both inpatient and outpatient settings, especially for Jaccard and F1 metrics. LEAP gives a relatively good DDI measure on two datasets; however, its performance is weaker than other baselines in terms of accuracy. Although SimNN, DualNN, and RETAIN are implemented from very different perspectives, the former two are instance-based while the latter uses sequence modeling. They show neck-to-neck performance on *MIMIC-III*. For outpatient medication change prediction (on *IQVIA*), RETAIN shows strong performance while some recent state of the art baselines failed, such as GAMENet. We hypothesize that time spans between two visits can be much longer for outpatients, and thus the stored memory can be less trustworthy in GAMENet. By learning an effective residual representation, MICRON provides more accurate and safe medication recommendations for inpatient or outpatient settings. Also, MICRON requires much fewer parameters than the state-of-the-art approaches, which is more efficient.

We also test the model's stability and do a T-hypothesis testing of MICRON on each metric. As a summary, most of the p-values are less than 0.001 (mean p-value at 6.2e-5), except in two cases on *IQVIA*: the DDI rate compared to LEAP and the Err(remove) compared to RETAIN.

4.3 Ablation Study on Model Components

In this section, we verify the effectiveness of different components in MICRON. Specifically, we conduct ablation studies on *MIMIC-III* and test on the following variants:

- (i) MICRON *w/o* L_{rec}. We remove the unsupervised loss during training and solely trained on the supervised loss.

- (ii) MICRON *w/o* $\tilde{\mathbf{m}}^{(t)}$. We do not maintain the medication vector, $\tilde{\mathbf{m}}^{(t)}$, and only utilizes the update feature information, $\mathbf{r}^{(t)}$, between two visits;

- (iii) MICRON *w/o* L_{multi}. We remove L_{multi}, and it will be less confident to use thresholds, δ_1 and δ_2;

- (iv) MICRON *w/o* L_{ddi}. We remove DDI loss, and the model probably would provide high-DDI combinations;

- (v) MICRON *w/o* δ_1, δ_2. We set $\delta_1 = \delta_2 = \delta$, which implies medications with score above or equal δ being added, and medications with score less than δ being removed. This is a common strategy used in previous works: $\delta = 0.5$ in [Shang *et al.*, 2019b] and $\delta = 0.3$ in [Shang *et al.*, 2019a] (this model requires ontology information, so it is not included as baseline). We use $\delta = 0.5$ for this model variant.

The comparison results with variances (after $\pm$) are shown in Table 3. Overall, all other variations perform better than variant (i) and (ii), highlighting that the reconstruction design and the medication vector are essential in the model. Without medication vector $\tilde{\mathbf{m}}^{(t)}$, the model cannot retain the longitudinal information, thus variant (ii) provides poor results. We also notice that without DDI loss, variant (iv) outputs a significantly higher DDI rate, and MICRON shows slightly better results than model variant (iii) without L_{multi}. By integrating all components, MICRON achieves a more balanced and stable performance in all metrics.

5 Conclusion

This paper tackles the medication change prediction problem and proposes a recurrent residual learning model, named MICRON, for predicting medication changes. We compare our model with state of the art approaches and show its effectiveness and efficiency on inpatient *MIMIC-III* dataset and a proprietary outpatient *IQVIA* dataset. This paper uses the existing prescriptions as a gold standard. The efficacy of the recommendation is evaluated by comparing it with the prescriptions given by the dataset, which might be a limitation. In the future, we consider performing clinical user studies to evaluate our results.

Acknowledgements

References

[Almirall *et al.*, 2012] Daniel Almirall, Scott N Compton, Meredith Gunlicks-Stoessel, Naihua Duan, and Susan A Murphy. Designing a pilot sequential multiple assignment randomized trial for developing an adaptive treatment strategy. *Statistics in medicine*, 31(17):1887–1902, 2012.

[Bajor and Lasko, 2017] Jacek M Bajor and Thomas A Lasko. Predicting medications from diagnostic codes with recurrent neural networks. In *International Conference on Learning Representation*, 2017.

[Bhoi *et al.*, 2020] Suman Bhoi, Lee Mong Li, and Wynne Hsu. Premier: Personalized recommendation for medical prescriptions from electronic records. *arXiv preprint arXiv:2008.13569*, 2020.

[Chen *et al.*, 2016] Zhuo Chen, Kyle Marple, Elmer Salazar, Gopal Gupta, and Lakshman Tamil. A physician advisory system for chronic heart failure management based on knowledge patterns. *Theory and Practice of Logic Programming*, 16(5-6):604–618, 2016.

[Choi *et al.*, 2016] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua Kulas, Andy Schuetz, and Walter Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *Advances in Neural Information Processing Systems*, pages 3504–3512, 2016.

[Choi *et al.*, 2017] Edward Choi, Andy Schuetz, Walter F Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association*, 24(2):361–370, 2017.

[Johnson *et al.*, 2016] Alistair EW Johnson, Tom J Pollard, Lu Shen, H Lehman Li-Wei, Mengling Feng, Mohammad Ghassemi, Benjamin Moody, Peter Szolovits, Leo Anthony Celi, and Roger G Mark. Mimic-iii, a freely accessible critical care database. *Scientific data*, 3(1):1–9, 2016.

[Killian *et al.*, 2019] Jackson A Killian, Bryan Wilder, Amit Sharma, Vinod Choudhary, Bistra Dilkina, and Milind Tambe. Learning to prescribe interventions for tuberculosis patients using digital adherence data. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2430–2438, 2019.

[Lakkaraju and Rudin, 2017] Himabindu Lakkaraju and Cynthia Rudin. Learning cost-effective and interpretable treatment regimes. In *Artificial Intelligence and Statistics*, pages 166–175, 2017.

[Le *et al.*, 2018] Hung Le, Truyen Tran, and Svetha Venkatesh. Dual memory neural computer for asynchronous two-view sequential learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1637–1645, 2018.

[Mao *et al.*, 2019] Chengsheng Mao, Liang Yao, and Yuan Luo. Medgcn: Graph convolutional networks for multiple medical tasks. *arXiv preprint arXiv:1904.00326*, 2019.

[Percha and Altman, 2013] Bethany Percha and Russ B Altman. Informatics confronts drug–drug interactions. *Trends in pharmacological sciences*, 34(3):178–184, 2013.

[Shang *et al.*, 2019a] Junyuan Shang, Tengfei Ma, Cao Xiao, and Jimeng Sun. Pre-training of graph augmented transformers for medication recommendation. *arXiv preprint arXiv:1906.00346*, 2019.

[Shang *et al.*, 2019b] Junyuan Shang, Cao Xiao, Tengfei Ma, Hongyan Li, and Jimeng Sun. Gamenet: Graph augmented memory networks for recommending medication combination. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1126–1133, 2019.

[Tatonetti *et al.*, 2012] Nicholas P Tatonetti, P Ye Patrick, Roxana Daneshjou, and Russ B Altman. Data-driven prediction of drug effects and interactions. *Science translational medicine*, 4(125):125ra31–125ra31, 2012.

[Wang *et al.*, 2017] Meng Wang, Mengyue Liu, Jun Liu, Sen Wang, Guodong Long, and Buyue Qian. Safe medicine recommendation via medical knowledge graph embedding. *arXiv preprint arXiv:1710.05980*, 2017.

[Wang *et al.*, 2018] Lu Wang, Wei Zhang, Xiaofeng He, and Hongyuan Zha. Supervised reinforcement learning with recurrent neural network for dynamic treatment recommendation. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2447–2456, 2018.

[Wang *et al.*, 2019] Shanshan Wang, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Order-free medicine combination prediction with graph convolutional reinforcement learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 1623–1632, 2019.

[Xiao *et al.*, 2018] Cao Xiao, Edward Choi, and Jimeng Sun. Opportunities and challenges in developing deep learning models using electronic health records data: a systematic review. *Journal of the American Medical Informatics Association*, 25(10):1419–1428, 2018.

[Yang *et al.*, 2021] Chaoqi Yang, Cao Xiao, Fenglong Ma, Lucas Glass, and Jimeng Sun. Safedrug: Dual molecular graph encoders for safe drug recommendations. *arXiv preprint arXiv:2105.02711*, 2021.

[Zhang *et al.*, 2017] Yutao Zhang, Robert Chen, Jie Tang, Walter F Stewart, and Jimeng 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*, pages 1315–1324, 2017.