# Multi-Cause Learning for Diagnosis Prediction

Liping Wang[1,2][0000−0002−3824−0806], Qiang Liu[1,2][0000−0002−9233−3827], Huanhuan Ma[1,2][0000−0002−7151−9550], Shu Wu[1,2][0000−0003−2164−3577], and Liang Wang[1,2][0000−0001−5224−8647]

[1] Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences
`wangliping2019@ia.ac.cn`, `qiang.liu@nlpr.ia.ac.cn`, `huanhuan.ma@cripac.ia.ac.cn`, `{shu.wu, wangliang}@nlpr.ia.ac.cn`
[2] School of Artificial Intelligence, University of Chinese Academy of Sciences

**Abstract.** Recently, Electronic Health Records (EHR) have become valuable for enhancing diagnosis prediction. Despite the effectiveness of existing deep learning based methods, one unified embedding fails to capture multiple disease causes of a patient. Even though naive adoption of multi-head attention could produce multiple cause vectors, a strong correlation between these cause representations might mislead the model to learning statistical spurious dependencies between cause vectors and diagnosis predictions. Hence, in this work, we propose a novel **Mul**ti-Cause Learning framework for **Diag**nosis Prediction, named **MulDiag**. Our Multi-Cause Network extracts multiple cause representations for a patient. We introduce HSIC (Hilbert-Schmidt Independence Criterion) to measure the dependencies among each pair of cause representations. Further, sample re-weighting techniques are utilized to conduct cause decorrelation. Experimental results on a publicly available dataset demonstrate the effectiveness of our method.

**Keywords:** Diagnosis prediction · Multi-cause · Decorrelation · Statistical dependency.

## 1 Introduction

Recently, Electronic Health Records (EHR) have become valuable for enhancing medical decision making. EHR data are represented as a temporal sequence of visits, where each visit includes multiple medical codes, representing clinical diagnoses. One critical task is to predict future diagnoses based on historical EHR data of a patient, so as to intervene in advance, i.e., diagnosis prediction.

Meanwhile, deep learning models have achieved great success in various domains [7, 8, 20]. A lot of deep learning based methods have also been proposed to model sequential EHR data. Similar to word embedding [17], each diagnosis is parameterized by a real-valued vector. Recurrent neural networks [8] are adopted to model temporal correlation among EHR sequence data. With a patient's historical EHR data, these deep learning based methods usually generate an overall embedding as patient health status representation.

Despite the effectiveness of these deep learning based approaches, there remain some challenges demanding further exploration. A primary challenge is that it is hard for a unified embedding to reflect different aspects of disease progression. Take an old man as an instance, he may suffer from multiple diseases: diabetes and heart disease. Diagnoses of these two kinds of diseases appear during the historical EHR data. Information of different diseases is fused in the unified patient representation which produces difficulties for accurate predictions. Hence, we propose a multi-cause network to capture multiple disease causes of a patient.

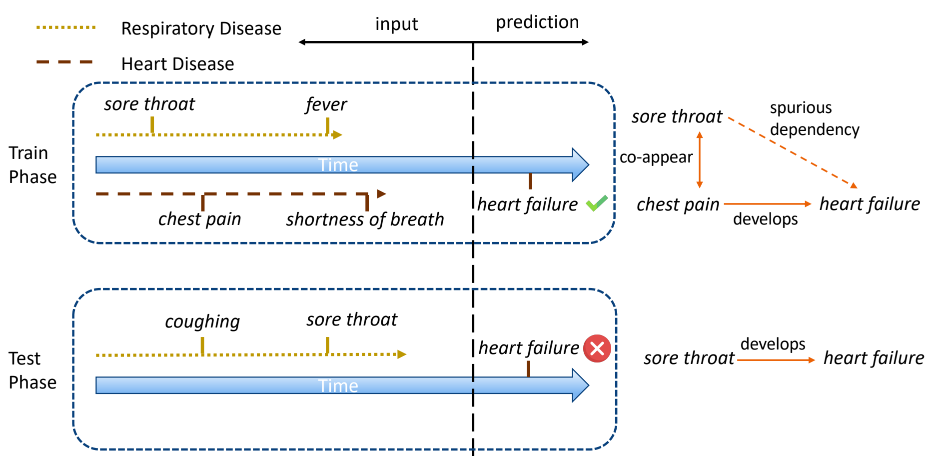


Fig. 1: Since the training dataset is collected in flu season, typical diagnoses of respiratory (*fever, sore throat*) and heart disease (*chest pain*) tend to appear at the same time. Hence, it is possible for the model to learn a spurious dependency between *sore throat* and *heart failure*. Then, in the test phase, the model may make predictions of heart failure according to *sore throat* symptoms.

Some existing methods [2] attempt to adopt multi-head attention mechanisms to capture different aspects of disease progression. However, the performance improvement is limited for two reasons. First of all, without proper regularization, it is hard to obtain a model which can produce diverse cause vectors. Instead, the obtained cause representations will be highly correlated which limits the capability of those methods. Further, the strong statistical correlation may mislead models to learn a statistical spurious dependency between diagnosis prediction and disease cause representation. As a result, when data distribution shifts, the learned statistical spurious dependency may generate false predictions. For instance, as illustrated in Figure 1, during flu season, typical symptoms of respiratory (for example, cold) and heart disease tend to co-appear in some old patients. If a model which attempts to capture multiple disease causes is trained on these data, diagnoses of respiratory (*sore throat*) and heart disease (*heart fail-*

*ure*) would be statistically correlated. This kind of spurious dependency would result in false predictions of *heart failure* if symptom *sore throat* appears in historical visits.

To tackle the above two challenges, we propose a novel **Mul**ti-Cause Learning framework for **Diag**nosis Prediction, named **MulDiag**. With regard to the first challenge, we propose to represent one patient with multiple vectors through a multi-cause network. As for the second challenge, we introduce the Hilbert-Schmidt Independence Criterion (HSIC) to measure the degree of independence among captured disease causes. Inspired by sample re-weighting techniques [10, 25], the cause correlation regularizer aims to estimate a sample weight for each sample such that captured causes are decorrelated on the reweighted training data. These two modules are jointly optimized in our method.

The main contributions of this work are summarized as follows:

- We propose a multi-cause network to capture different causes of a patient.
- We introduce the Hilbert-Schmidt Independence Criterion (HSIC) to measure dependencies among captured causes.
- We adopt re-weighting techniques to conduct cause decorrelation for diagnosis prediction.

## 2 Related Work

### 2.1 Diagnosis Prediction

EHR data contain rich historical health information of patients. Building powerful health risk prediction models based on EHR data paves the way for personalized health care applications. Recently, deep learning techniques, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have achieved great success in various applications among multiple domains, including health risk prediction and diagnosis prediction based on EHR data. In viewing that EHR data exist in temporal sequential form, it is natural to adopt RNNs or LSTMs to model disease progression in the time dimension. In comparison, CNNs are adopted to capture local dependence in EHR data.

In Dipole [14], bidirectional recurrent neural networks are employed to remember all the information of both the past visits and the future visits, and three attention mechanisms are introduced to measure the influence of different visits for the prediction. RETAIN [2] develops a reverse time attention model for EHR data which achieves high accuracy while remaining clinically interpretable. Its two-level neural attention detects influential past visits and significant clinical variables within those visits (e.g. key diagnoses). Some works try to model disease progression by taking time intervals into consideration. For example, StageNet [5] integrates inter-visit time information into LSTM cell states to capture the stage variation of patients' health conditions.

Another line of work proposes to incorporate existing medical knowledge into diagnosis prediction. For example, GRAM [3] infuses information from a medical ontology DAG (Directed acyclic graph) [19] into deep learning models via neural attention. GRAM can learn accurate and interpretable representations for medical concepts and show significant improvement in the prediction performance, especially on low-frequency diseases and small datasets. HAP [23] adopts the same medical ontology DAG with GRAM [3], but hierarchically propagates attention across the entire ontology structure with two rounds of knowledge propagation. Nevertheless, in both GRAM and HAP, medical ontology information is only used when learning code representations. Hence, Ma et al. [15] propose KAME which directly exploits medical knowledge in the whole prediction process, i.e. learning code representations, generating visit embeddings and making predictions. KnowRisk [24] and DG-RNN [22] incorporate a more powerful and larger scale knowledge graph KnowLife [4][3] to enrich the information extracted from insufficient inputs and guide the prediction. And they propose sophisticated knowledge graph attention to obtain the latent information from embeddings of the input events in the knowledge graph.

### 2.2 Stable Learning

In order to tackle the problem of statistical spurious dependency, researchers propose a stable learning framework. The framework usually consists of two steps: learning weights of training samples and training based on weighted data. To be more specific, sample weights are learned to reduce the correlation between features that could be measured by HSIC [6] or similar metrics. Under this framework, a lot of decorrelation methods [18, 10] have been proposed to train linear stable models using re-weighted samples. Then, various deep stable models are also proposed. For instance, StableNet [25] proposes to remove dependencies between features by adopting sample weighting based on RFF (Random Fourier Features). OOD-GNN [12] designs a novel nonlinear graph representation decorrelation method.

For the diagnosis prediction task, Luo et al. [13] propose to use a causal representation learning method called Causal Healthcare Embedding (CHE) which aims at eliminating the spurious statistical relationship by removing the dependencies between diagnoses and procedures. In comparison, we propose MulDiag to eliminate spurious dependencies between different disease causes.

## 3 Preliminary

In this section, we mainly provide some background knowledge about EHR data and formulate the diagnosis prediction task.

### 3.1 Electronic Health Records

Electronic Health Records (EHR) is a special kind of data that consists of the medical history of a patient. For each visit to the hospital of a specific patient,

[3] http://knowlife.mpi-inf.mpg.de/

the diagnoses are recorded as medical codes in a pre-defined system such as ICD[4] (International Classification of Diseases) or CUI [5] (Concept Unique Identifiers).

### 3.2 Basic Notations

In this paper, all the unique medical codes from EHR data are denoted as $\mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_{|C|} \in C$. For a specific patient, the EHR data are denoted as $V = \{v_1, v_2, \ldots, v_t\}$. Visit $v_t$ is a subset of $C$, representing medical codes appearing in the $t$-th visit. For the convenience of calculation, $v_t$ can also be represented as a $|C|$-length multi-hot vector $\mathbf{x}_t \in \{0,1\}^{|C|}$, where each element is zero or one, representing each medical code appears or not respectively. By stacking those multi-hot vectors, we reach a 0-1 valued matrix $\mathbf{X} \in \{0,1\}^{t \times |C|}$ to represent the EHR data.

### 3.3 Diagnosis Prediction Task

Diagnosis prediction is one of the most important tasks in the health care area which aims to predict potential diagnoses according to historical EHR data. Here, we give the formulation based on the notations provided above. For a specific patient, denote his or her EHR data for $t$ consecutive visits as $\mathbf{X} \in \{0,1\}^{t \times |C|}$, the goal is to tell which diagnosis is likely to appear in the next visit, i.e. the value of $\mathbf{x}_{t+1}$.

## 4 Methodology

In Figure 2, we provide an overview of the proposed MulDiag. In the following, we will describe each sub-module and optimization in detail.

### 4.1 Multi-Cause Network

In MulDiag, we employ a parameter embedding matrix $\mathbf{E} \in \mathbb{R}^{|C| \times d}$, where each row encodes a medical code. Given $t$-th visit code $\mathbf{x}_t$, we can obtain the vector representation for $t$-th visit as follows:

$$\mathbf{v}_t = \mathbf{E}\mathbf{x}_t. \tag{1}$$

Inspired by deep multi-interest recommendation models [11, 1], we devise a Multi-Cause Network to generate multiple representations to reflect the disease causes of patients. In previous studies, the attention mechanism has shown strong capability in exploiting temporal EHR visit data. Hence, in this work, we adopt a similar temporal attention mechanism. First, visit embeddings $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t$ are fed into an RNN to encode historical visits information into state vectors:

$$\mathbf{g}_1, \mathbf{g}_2, \ldots, \mathbf{g}_t = \mathrm{RNN}(\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_t). \tag{2}$$

[4] https://www.cdc.gov/nchs/icd/icd9.htm
[5] https://www.nlm.nih.gov/research/umls/new_users/online_learning/Meta_005

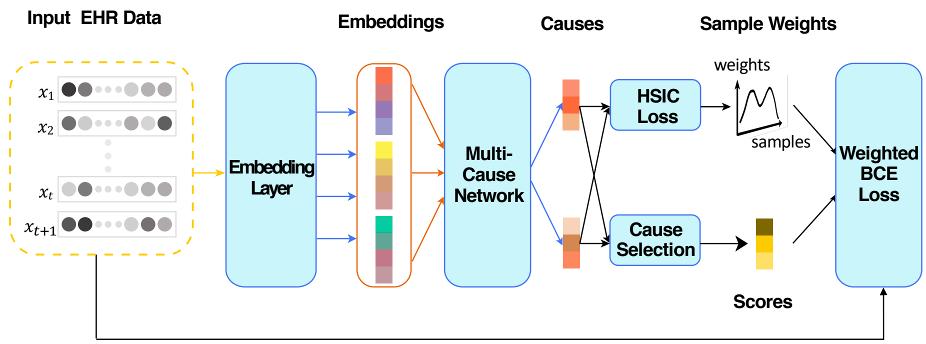


Fig. 2: The overview of the proposed MulDiag. The Embedding Layer first converts visits consisting of medical codes into dense embeddings. Then, Multi-Cause Network extracts multiple cause vectors given visit embeddings. Empirical HSIC statistics is calculated between each pair of cause representations and are optimized through sample weighting. Weighted BCE loss is adopted to optimize model parameters.

Then, based on these state vectors, attention coefficients are given by

$$\alpha_1, \alpha_2, \dots \alpha_t = \mathrm{softmax}(a_1, a_2, \dots, a_t), \tag{3}$$

in which $\alpha_i = \mathbf{w}_a^T \mathbf{g}_i + b$. Finally, we can obtain cause vector representations as follows:

$$\mathbf{c} = \sum_{i=1}^{t} \alpha_i \mathbf{v}_i. \tag{4}$$

We adopt the multi-head attention mechanism (for the sake of brevity, we omit the subscript in the previous text), so there are multiple cause vectors, i.e. $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m$.

### 4.2 Cause Decorrelation

To decorrelate cause representations, we first need to measure the degree of dependence between each pair of cause representation vectors. Cause representations $\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_m$ are samples of a high-dimensional distribution. In this paper, we introduce HSIC to reflect dependence among each pair of cause representations. HSIC is the Hilbert-Schmidt norm of the cross-covariance operator between distributions in Reproducing Kernel Hilbert Space (RKHS). Let $\mathbf{x}, \mathbf{y}$ be random vector variables, and they follow distribution $p_{\mathbf{xy}}$, HSIC is given by

$$\begin{aligned}\mathrm{HSIC}\left(p_{\mathbf{XY}}, \mathbf{x}, \mathbf{y}\right) =& \mathrm{E}_{\mathbf{x}_1,\mathbf{x}_2,\mathbf{y}_1,\mathbf{y}_2}\left[k\left(\mathbf{x}_1, \mathbf{x}_2\right) l\left(\mathbf{y}_1, \mathbf{y}_2\right)\right] + \mathrm{E}_{\mathbf{x}_1,\mathbf{x}_2}\left[k\left(\mathbf{x}_1, \mathbf{x}_2\right)\right] \\ &\cdot \mathrm{E}_{\mathbf{y}_1,\mathbf{y}_2}\left[l\left(\mathbf{y}_1, \mathbf{y}_2\right)\right] - 2\mathrm{E}_{\mathbf{x}_1,\mathbf{y}_1}\left[\mathrm{E}_{\mathbf{x}_2}\left[k\left(\mathbf{x}_1, \mathbf{x}_2\right)\right] \mathrm{E}_{\mathbf{y}_2}\left[l\left(\mathbf{y}_1, \mathbf{y}_2\right)\right]\right],\end{aligned} \tag{5}$$

in which $k(\cdot, \cdot)$ and $l(\cdot, \cdot)$ are kernel functions.

However, the definition of HSIC in Equation 5 is only theoretically valuable. Luckily, given a series of $n$ independent samples $Z := \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\} \subset \mathcal{X} \times \mathcal{Y}$ drawn from $p_{\mathbf{xy}}$, there is an approximately unbiased empirical statistics [6]:

$$\mathrm{HSIC}(Z) = (n-1)^{-2} \operatorname{tr} KHLH, \tag{6}$$

where $H, K, L \in \mathbb{R}^{n \times n}, K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j), L_{ij} = l(\mathbf{y}_i, \mathbf{y}_j)$ and $H_{ij} = \delta_{ij} - n^{-1}$. In this paper, we adopt the Radial Basis Function (RBF) kernel functions, i.e.

$$k(\mathbf{x}_1, \mathbf{x}_2) = l(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|_2^2}{\sigma^2}.\right) \tag{7}$$

**Algorithm 1:** Training of MulDiag

**Input:** Training dataset
**Parameters:** $\mathbf{\Theta}, \mathbf{w}$

- Initialize sample weights $\mathbf{w} \leftarrow \mathbf{1}$
- Randomly initialize model parameters $\mathbf{\Theta}$
- **for** $q \leftarrow 1$ **to** *max_epoch* **do**
  - Keep $\mathbf{w}^{(q-1)}$ fixed and update parameters $\mathbf{\Theta}^{(q)}$ according to Equation 10
  - Keep $\mathbf{\Theta}^{(q)}$ fixed and update sample weights $\mathbf{w}^{(q)}$ according to Equation 12
  - **if** *early stopping condition reaches* **then**
    - return $f_{\mathbf{\Theta}^{(q)}}$
  - **end**
- **end**

Inspired by sample re-weighting techniques, we propose a cause decorrelation framework that aims to estimate a weight for each sample. In this manner, cause representations for re-weighted data are decorrelated. We denote $\mathbf{w} \in \mathbb{R}^n$ as the sample weights, where $n$ is the number of samples. Before training, $\mathbf{w}$ is initialized as $[1, 1, \ldots, 1]$. During training, sample weights $\mathbf{w}$ and model parameters are alternatively optimized as shown in Algorithm 1.

**Model Optimization.** During the optimization of model parameters, sample weights $\mathbf{w}$ is fixed. Given the $k$-th training sample $\mathbf{X}_k = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_t, \mathbf{x}_{t+1})$, for target medical code $i$, cause selection is conducted by choosing a cause representation that is closest to the embedding vector $\mathbf{E}_i$:

$$\hat{s}_i = \max \mathbf{c}_j^T \mathbf{E}_i. \tag{8}$$

The normalized prediction score for $i$-th medical code will be $s_i = \frac{\exp(\hat{s}_i)}{\sum_j \exp(\hat{s}_j)}$. Hence, the BCE (Binary Cross Entropy) loss function for the $k$-th training sample would be

$$\mathcal{L}(\mathbf{X}_k) = -\sum_{i=1}^{|C|} s_i \log(\mathbf{x}_{t+1}[i]) + (1 - s_i) \log(1 - \mathbf{x}_{t+1}[i]). \tag{9}$$

Model parameters $\mathbf{\Theta}$ is updated through the weighted BCE loss:

$$\mathbf{\Theta} \leftarrow \operatorname{argmin} \sum_k \mathbf{w}_k \mathcal{L}(\mathbf{X}_k) \tag{10}$$

**Weight Optimization.** To obtain decorrelated cause representations, MulDiag finds optimal sample weights by minimizing the empirical HSIC statistics between each pair of weighted cause vectors. Formally, given a batch of $B$ samples, let $\mathbf{c}_i^{(b)}$ be the $i$-th cause vector of the $b$-th sample and $\mathbf{w}(b)$ be the sample weight for the $b$-th sample, Then, the HSIC loss would be

$$\text{HSIC loss} = \sum_i \sum_j \text{HSIC}(\{(\mathbf{w}(1)\mathbf{c}_i^{(1)}, \mathbf{w}(1)\mathbf{c}_j^{(1)}), \ldots, (\mathbf{w}(B)\mathbf{c}_i^{(B)}, \mathbf{w}(B)\mathbf{c}_j^{(B)})\}), \tag{11}$$

where HSIC is defined in Equation 6. With model parameters $\mathbf{\Theta}$ fixed, sample weights $\mathbf{w}$ is updated as:

$$\mathbf{w} \leftarrow \operatorname*{argmin}_{\mathbf{w}} \text{HSIC loss}. \tag{12}$$

### 4.3 Complexity Analysis and Model Comparison

In this subsection, we analyze the complexity of MulDiag and compare it with mainstream diagnosis prediction models.

For MulDiag, it takes $O(nmdLK)$ to obtain $m$ cause vectors for $n$ samples, in which $d$ is embedding size, $L$ is the average length of visit data and $K$ is the average number of diagnoses appearing in one visit. Cause decorrelation process takes $O(Bnd)$ to compute the HSIC statistics in which $B$ is the batchsize.

For mainstream diagnosis prediction models such as RETAIN and StageNet, computation complexity is usually $O(ndLK)$. Therefore, MulDiag is as asymptotically efficient as mainstream diagnosis prediction methods.

## 5 Experiments

In this section, we first provide details of experimental settings. Then, we discuss the experimental results of MulDiag and compare them with baseline methods. In addition, we also provide visualization and sensitivity analysis.

### 5.1 Experimental Setup

**Dataset** In this paper, we conduct extensive experiments on a real-world EHR dataset MIMIC-III which includes 7,537 patients' health records from ICU. In the training phase, part of historical diagnoses are employed as an input of our model while future diagnoses serve as supervision signals. Similarly, in the test phase, diagnoses appearing later than those in the training set are adopted to compute the accuracy and precision of our model.

**Baselines** To validate the effectiveness of the proposed MulDiag, we choose four competitive baseline models: LSTM, RETAIN [2], RAIM [21], StageNet[5].

LSTM: We adopt the same embedding method as Dipole [14]. Then, the embeddings of each visit are fed into an LSTM [8] layer. After that, all hidden states are added together to obtain a final feature vector. In the end, a linear classifier is employed to reach final predictions.

RETAIN: RETAIN is a competitive prediction model that adopts a two-level neural attention model that detects influential past visits and significant clinical variables with those visits.

RAIM: RAIM introduces an efficient attention mechanism for continuous monitoring data, which is guided by discrete clinical events. With guided multi-channel attention, high-density multi-channel signals are integrated with discrete events and prove very useful in risk prediction.

StageNet: StageNet is constituted of a stage-aware long-short-term memory (LSTM) module extracting health stage variations with no supervision and a stage-adaptive convolutional module that incorporates stage-related progression patterns.

**Evaluation Metric** Following previous works [3, 15], we adopt two metrics to measure the performance of all methods for the diagnosis prediction task, i.e. visit-level precision@k and code-level accuracy@k. In addition, we sort the medical codes by their frequencies in the training dataset in non-decreasing order, and then divide them into five different groups. We report code-level accuracy in each group to reflect the prediction performance for codes with varying frequencies.

**Implementation Details** In this paper, all the baselines and our models are implemented with PyTorch [6] [16]. The dataset is randomly divided into training, validation and testing sets in a 0.7:0.1:0.2 ratio. Embedding size $d$ is set to 64 for all approaches. The same dropout strategy with a 0.5 drop rate is applied to all the methods. All methods are trained with Adam optimizer [9] with a mini-batch of 128 samples. The learning rate is fixed at 0.001 for all methods.

### 5.2 Performance Comparison

Comparison results at both visit and code levels are reported in Table 1, in which, precision and accuracy for different values of $k$ are included. From the table, we can observe that MulDiag outperforms all the baseline methods. In Table 2, in addition to the overall performance in code-level accuracy, we also report the results for each group which are obtained by dividing the medical codes according to the percentile of their frequencies in the training dataset. For example, 0-20 are the rarest diagnoses while 80-100 represent the most common ones. From the table, we can tell that in addition to the overall performance improvement,

[6] https://pytorch.org/

Table 1: Visit Level Precision@k and Code Level Accuray@k comparison on MIMIC-III. Average results for multiple values of $k$ are also included.

| Model | Visit Level Precision@k | | | | | | Code Level Accuracy@k | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | 15 | 20 | 25 | 30 | Avg | 10 | 15 | 20 | 25 | 30 | Avg |
| LSTM | 34.49 | 34.10 | 36.23 | 38.84 | 41.57 | 37.05 | 22.40 | 27.98 | 32.29 | 35.77 | 38.80 | 31.45 |
| RETAIN | **39.22** | 38.36 | 40.06 | 42.72 | 45.51 | 41.17 | 25.48 | 31.44 | 35.86 | 39.53 | 42.55 | 34.97 |
| RAIM | 23.49 | 23.50 | 25.31 | 28.17 | 30.75 | 26.24 | 15.93 | 20.27 | 24.15 | 27.61 | 30.55 | 23.70 |
| StageNet | 36.69 | 36.57 | 38.95 | 41.89 | 44.85 | 39.79 | 23.82 | 29.85 | 34.69 | 38.48 | 41.77 | 33.72 |
| MulDiag | 39.16 | **38.77** | **40.93** | **43.71** | **46.47** | **41.81** | **25.49** | **31.73** | **36.55** | **40.27** | **43.40** | **35.49** |

Table 2: Code-level accuracy@20. Diagnosis codes are divided into five groups according to their frequencies in the training set. For example, 0-20 are the rarest diagnoses.

| Model | Code-Level Accuracy | | | | | |
|---|---|---|---|---|---|---|
| | 0-20 | 20-40 | 40-60 | 60-80 | 80-100 | Overall |
| LSTM | 2.49 | 12.10 | 19.08 | 47.07 | 81.31 | 32.29 |
| RETAIN | 3.02 | 17.17 | 25.64 | **51.31** | 82.38 | 35.86 |
| RAIM | 0.00 | 0.00 | 0.00 | 26.56 | **96.87** | 24.15 |
| StageNet | 3.51 | 16.39 | 23.85 | 47.75 | 82.43 | 34.69 |
| MulDiag | **5.80** | **20.53** | **28.26** | 49.16 | 79.18 | **36.55** |

MulDiag achieves significant improvement in the prediction of rare diagnoses. In comparison, baseline models perform poorly for those infrequent diagnoses.

### 5.3 Visualization Analysis

For an easier understanding of weight optimization, we visualize the change of HSIC on the test set while MulDiag and MulDiag-NWO are training on the MIMIC-III dataset. Compared with MulDiag, MulDiag-NWO is almost the same except that there is no sample weight optimization (i.e. each sample weight is 1). Since the parameters of models are initialized randomly, HSIC is near 0 at earlier epochs for both MulDiag and MulDiag-NWO. Then, the HSIC begins to decrease. After some epochs, the HSIC of MulDiag-NWO on the test set remains unchanged while the HSIC of MulDiag keeps decreasing. This makes it possible for our MulDiag to update more steps and achieve better performance.

### 5.4 Sensitivity Analysis

We also provide the experimental results for the sensitivity analysis of the number of causes. As Figure 4 illustrates, the number of causes does not impact the performance very much. Cause decorrelation of MulDiag is capable of boosting the performance for various values of the number of causes.

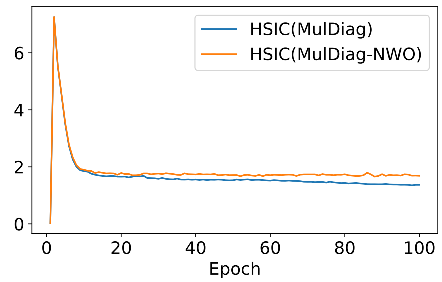


Fig. 3: The change of HSIC on the test set when MulDiag and MulDiag-NWO are trained on MIMIC-III.

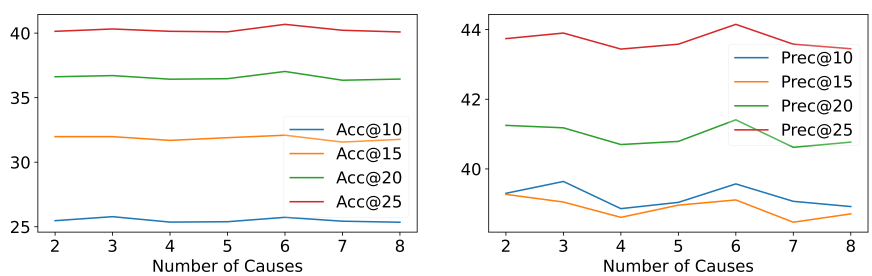


Fig. 4: Accuracy and precision of MulDiag with the number of causes varying from 2 to 8.

## 6 Conclusion

In this paper, we propose MulDiag which aims to capture multiple causes of diseases. To avoid learning statistical spurious dependency between cause representations and diagnosis predictions, we first introduce HSIC to measure the degree of independence among cause vectors. Then, re-weighting techniques are adopted to implement dependency decorrelation. Extensive experiments on the publicly available benchmark dataset demonstrate the effectiveness of our model.

## References

1. Cen, Y., Zhang, J., Zou, X., Zhou, C., Yang, H., Tang, J.: Controllable multi-interest framework for recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020)
2. Choi, E., Bahadori, M.T., Kulas, J.A., Schuetz, A., Stewart, W.F., Sun, J.: Retain: an interpretable predictive model for healthcare using reverse time attention mechanism. In: Proceedings of the 30th International Conference on Neural Information Processing Systems (2016)
3. Choi, E., Bahadori, M.T., Song, L., Stewart, W.F., Sun, J.: Gram: Graph-based attention model for healthcare representation learning. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017)
4. Ernst, P., Meng, C., Siu, A., Weikum, G.: Knowlife: A knowledge graph for health and life sciences. In: 2014 IEEE 30th International Conference on Data Engineering (2014)
5. Gao, J., Xiao, C., Wang, Y., Tang, W., Glass, L.M., Sun, J.: Stagenet: Stage-aware neural networks for health risk prediction. In: Proceedings of The Web Conference 2020 (2020)
6. Gretton, A., Bousquet, O., Smola, A., Schölkopf, B.: Measuring statistical dependence with hilbert-schmidt norms. In: Algorithmic Learning Theory
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016)
8. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Computation (1997)
9. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
10. Kuang, K., Xiong, R., Cui, P., Athey, S., Li, B.: Stable prediction with model misspecification and agnostic distribution shift. In: Proceedings of the AAAI Conference on Artificial Intelligence (2020)
11. Li, C., Liu, Z., Wu, M., Xu, Y., Zhao, H., Huang, P., Kang, G., Chen, Q., Li, W., Lee, D.L.: Multi-interest network with dynamic routing for recommendation at tmall. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management (2019)
12. Li, H., Wang, X., Zhang, Z., Zhu, W.: Ood-gnn: Out-of-distribution generalized graph neural network. arXiv preprint arXiv:2112.03806 (2021)
13. Luo, Y., Liu, Z., Liu, Q.: Deep stable representation learning on electronic health records. arXiv preprint arXiv:2209.01321 (2022)
14. Ma, F., Chitta, R., Zhou, J., You, Q., Sun, T., Gao, J.: Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017)
15. Ma, F., You, Q., Xiao, H., Chitta, R., Zhou, J., Gao, J.: Kame: Knowledge-based attention model for diagnosis prediction in healthcare. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. p. 743–752 (2018)
16. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (2019)

17. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: Empirical Methods in Natural Language Processing (EMNLP) (2014)
18. Shen, Z., Cui, P., Kuang, K., Li, B., Chen, P.: Causally regularized learning with agnostic data selection bias. In: Proceedings of the 26th ACM International Conference on Multimedia (2018)
19. Thulasiraman, K., Swamy, M.N.: Graphs: theory and algorithms. John Wiley & Sons (2011)
20. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems (2017)
21. Xu, Y., Biswal, S., Deshpande, S.R., Maher, K.O., Sun, J.: Raim: Recurrent attentive and intensive model of multimodal patient monitoring data. In: Proceedings of the 24th ACM SIGKDD international conference on Knowledge Discovery and Data Mining (2018)
22. Yin, C., Zhao, R., Qian, B., Lv, X., Zhang, P.: Domain knowledge guided deep learning with electronic health records. In: 2019 IEEE International Conference on Data Mining (2019)
23. Zhang, M., King, C.R., Avidan, M., Chen, Y.: Hierarchical attention propagation for healthcare representation learning. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2020)
24. Zhang, X., Qian, B., Li, Y., Yin, C., Wang, X., Zheng, Q.: Knowrisk: An interpretable knowledge-guided model for disease risk prediction. In: 2019 IEEE International Conference on Data Mining (ICDM) (2019)
25. Zhang, X., Cui, P., Xu, R., Zhou, L., He, Y., Shen, Z.: Deep stable learning for out-of-distribution generalization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (2021)