Computers in Biology and Medicine 169 (2024) 107865

Contents lists available at ScienceDirect

Computers in Biology and Medicine

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

Graph-guided deep hashing networks for similar patient retrieval

Yifan Gu [a,b,c,1], Xuebing Yang [b,1], Mengxuan Sun [b,c], Chutong Wang [b,c], Hongyu Yang [a,f], Chao Yang [a,f], Jinwei Wang [a,f], Guilan Kong [d,e], Jicheng Lv [a,f,*], Wensheng Zhang [b,g,*]

[a] *Renal Division, Department of Medicine, Peking University First Hospital, Beijing, China*
[b] *State Key Laboratory of Multimodal Artificial Intelligence Systems (MAIS), Institute of Automation, Chinese Academy of Sciences, Beijing, China*
[c] *School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China*
[d] *National Institute of Health Data Science, Peking University, Beijing, China*
[e] *Advanced Institute of Information Technology, Peking University, Hangzhou, China*
[f] *Research Units of Diagnosis and Treatment of Immune-mediated Kidney Diseases, Chinese Academy of Medical Sciences, Beijing, China*
[g] *Guangzhou University, Guangzhou, China*

ARTICLE INFO

Keywords:
Similar patient retrieval
Deep hashing
Graph neural networks
Patient representation learning
Electronic health records

ABSTRACT

With the rapid growth and widespread application of electronic health records (EHRs), similar patient retrieval has become an important task for downstream clinical decision support such as diagnostic reference, treatment planning, etc. However, the high dimensionality, large volume, and heterogeneity of EHRs pose challenges to the efficient and accurate retrieval of patients with similar medical conditions to the current case. Several previous studies have attempted to alleviate these issues by using hash coding techniques, improving retrieval efficiency but merely exploring underlying characteristics among instances to preserve retrieval accuracy. In this paper, drug categories of instances recorded in EHRs are regarded as the ground truth to determine the pairwise similarity, and we consider the abundant semantic information within such multi-labels and propose a novel framework named Graph-guided Deep Hashing Networks (GDHN). To capture correlation dependencies among the multi-labels, we first construct a label graph where each node represents a drug category, then a graph convolution network (GCN) is employed to derive the multi-label embedding of each instance. Thus, we can utilize the learned multi-label embeddings to guide the patient hashing process to obtain more informative and discriminative hash codes. Extensive experiments have been conducted on two datasets, including a real-world dataset concerning IgA nephropathy from Peking University First Hospital, and a publicly available dataset from MIMIC-III, compared with traditional hashing methods and state-of-the-art deep hashing methods using three evaluation metrics. The results demonstrate that GDHN outperforms the competitors at different hash code lengths, validating the superiority of our proposal.

1. Introduction

In past decades, healthcare data stored in electronic health records (EHRs) have experienced a sky-rocketing increase. EHR data are relatively complex because of their massive volume, heterogeneity and high dimensions, calling for efficient processing and exploitation. The solid data foundation has prompted great interest from researchers in personalized healthcare [1–6], which aims to provide patient-specific treatment for better prognosis at a lower cost. Realizing personalized healthcare typically involves two steps, first identifying similar patients and grouping them into cohorts, then analyzing the cohorts for further diagnosis, prescribing, etc [7]. Hence, learning patient similarity plays a basic but crucial role in personalized healthcare. Previous researches have made significative attempts to learn clinically meaningful similarity measures [8], based on statistical techniques, traditional machine learning [9–11] and deep metric learning [7,12,13]. In practice, given the proper similarity measures, more comprehensive applications with respect to patient similarity rely on efficient and accurate similar patient retrieval [14].

The workflow of similar patient retrieval is shown in Fig. 1, referring to the process of retrieving the cases most similar to the current query patient from an existing patient database based on a certain similarity measure. Some existing researches focus on similar patient retrieval based on the medical images [15], while others focus on the conversational-agent-based patient retrieval [16]. Recently, with

[*] Corresponding authors.
E-mail addresses: chenglv@263.net (J. Lv), zhangwenshengia@hotmail.com (W. Zhang).
[1] Equal contribution to this work.

https://doi.org/10.1016/j.compbiomed.2023.107865
Received 12 September 2023; Received in revised form 19 November 2023; Accepted 17 December 2023
Available online 19 December 2023

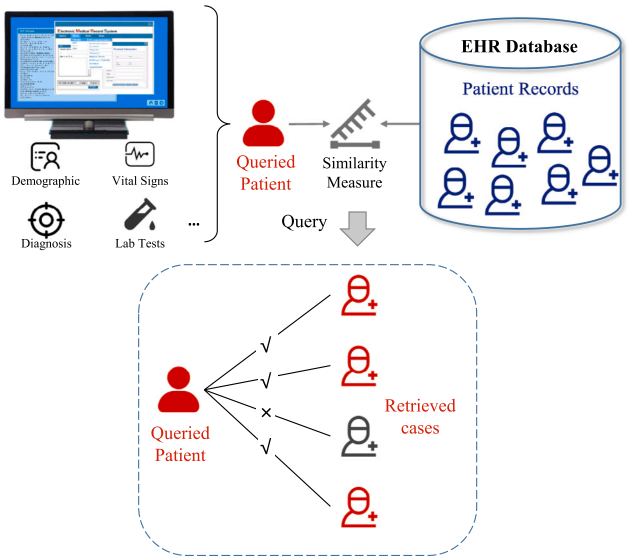


Fig. 1. The workflow of similar patient retrieval.

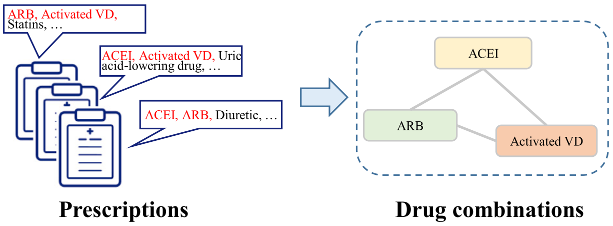

ARB, Activated VD, Statins, ...
ACEI, Activated VD, Uric acid-lowering drug, ...
ACEI, ARB, Diuretic, ...
Prescriptions
ACEI
ARB
Activated VD
Drug combinations

Fig. 2. Different drug combinations reveal the underlying relations among drug categories.

the rise of EHRs, more and more preliminary studies begin to investigate similar patient retrieval on EHR data. Some of them rely on hand-crafted features, e.g., [10] uses a low-rank structure to select features for similar patient retrieval. Some require physicians' manual annotation or feedback, e.g., [17] relies on the pairwise similarity labels annotated by physicians for patient indexing, and [18] performs similar patient retrieval based on a visual system of interactive patient labeling. Other researches are limited to the transductive setting and can only be used for in-database retrieval with a fixed number of patients. For example, [19] refers to the cosine similarity of patient features to retrieve similar patients in-database, and [20] organizes EHRs as a patient graph to learn encounter-level patient representations for in-database retrieval. Considering the complexity and difficulty of EHR data, an end-to-end framework with efficient implementation and scalability to the inductive setting is of urgent need.

To perform similar patient retrieval on EHR data with rapidly expanding volumes, a feasible scheme is to adopt hash coding (abbr. hashing) techniques, which aims to map the high-dimensional raw features to a low-dimensional Hamming space and represent each instance by a compact binary hash code. In this way, the distance between instances can be easily calculated through an XOR operation, thereby achieving efficient information retrieval. The conventional hashing methods are data-independent, and their hash functions are typically designed with randomized projection [21]. Recent researches have concentrated more on data-dependent hashing, which is also known as learning to hash. Along this path, there are generally two categories: unsupervised and supervised. The unsupervised hashing methods derive hash functions directly based on the structure and distribution of the data, while the supervised ones additionally utilize semantic labels for learning hash functions.

Due to the powerful feature extraction capability of deep neural networks, some researchers have introduced deep hashing methods to improve retrieval performance [22–26]. In particular, [14,27] have made attempts to apply deep hashing in the field of similar patient retrieval. In this study, we focus on learning to hash in a supervised manner and regard the drug categories as labels for supervising training processes. Naturally, for a given patient, his/her drug categories in the form of multi-labels [28,29] usually present the expertise of physicians. If two instances share some common drug categories, they are considered to be similar, otherwise, they are considered to be dissimilar. Note that it is the same with most existing deep hashing methods, which treat the multi-labels of instances as the ground truth to determine the pairwise similarity. However, we argue that the abundant semantic information contained in the multi-labels might be overlooked, i.e., except for ground truth, the inherent characteristics within multi-labels can serve to guide the hashing process.

Our inspiration comes from cross-modal hashing [30,31], which adopts deep neural networks for extracting multi-label embeddings to guide the learning procedure of image and text hashing. For EHR data, it can be observed that there are generally underlying relations among the semantic labels, e.g., some drugs have a high probability of co-occurring in prescriptions, while some are rarely prescribed together. As shown in Fig. 2, for some IgA nephropathy patients, physicians may prescribe ACEI and ARB drugs to them simultaneously, while for other patients, these two kinds of drugs may be used with activated vitamin D respectively. As different drug combinations signify different therapies based on the health status of patients [32], our objective is to capture the intrinsic correlation dependencies among multi-labels for better patient hashing. Motivated by graph neural networks (GNNs) [33] with their powerful capability to represent data points that have relations between each other [34–37], we employ GNNs to exploit the underlying relations between semantic labels.

In this paper, we propose a novel deep supervised hashing framework, Graph-guided Deep Hashing Networks (GDHN), for similar patient retrieval on EHR data. GDHN consists of two modules, the patient encoder and the label encoder. The patient encoder uses a multilayer perceptron (MLP) to perform representation learning and hash encoding on the raw features of patients. For the label encoder, we first construct a label graph, where each node represents a label, and the edges are determined according to the pointwise mutual information (PMI) between labels. Then we adopt a graph convolutional network (GCN) to explore the semantic information contained in the labels and the correlation dependencies between them to derive the multi-label embedding of each patient. To ensure the similarity between instances is retained in the embedding space of multi-labels, a pairwise negative log-likelihood function is introduced to measure the multi-label embedding similarity loss. Furthermore, to realize graph guidance, we design another pairwise negative log-likelihood loss between patient embeddings and multi-label embeddings. The above two loss terms are combined as the overall objective to supervise the network optimization in an end-to-end manner, to produce more informative hash codes for efficient and accurate similar patient retrieval.

To summarize, the major contributions of our work lie in three-fold:

- We propose a novel deep supervised hashing framework, namely GDHN, for similar patient retrieval on EHR data. Different from simply regarding the multi-labels as the ground truth to define the pairwise similarity between instances, GDHN makes effective use of the semantic information contained in multi-labels during the process of patient representation learning and binary hash encoding, deriving more informative hash codes.
- To explore the underlying relations among the multiple labels, we construct a label graph, where each node represents a label, and the edges are determined based on the co-occurrence of the labels. We propose to employ a GCN to extract the correlation dependencies between nodes and generate multi-label embeddings, which

are then utilized to guide the patient hashing procedure through well-designed loss functions.

- We conduct extensive similar patient retrieval experiments on two EHR datasets, including the *IgA Nephropathy* dataset and the *MIMIC-III* dataset. The results confirm the effectiveness of the proposed GDHN, and further analysis demonstrates that our method allows the instances with higher similarity to the query to be ranked higher in the retrieved results, which is more valuable for clinical applications.

The rest of this paper is organized as follows. Section 2 provides a brief overview of the related works. In Section 3, we introduce the notations used throughout this article, clarify the problem formulation, and describe the proposed GDHN in detail. The experimental study is presented in Section 4 to verify the effectiveness of our method. The ablation study and parameter analysis are included in Section 5. Finally, we make some concluding remarks and present the future work in Section 6.

2. Related work

Considering the task targeted by our approach, in this section, we first briefly outline the related works about patient similarity learning and similar patient retrieval on EHR data. Then the relevant studies about hashing methods are reviewed.

2.1. Patient similarity learning

The objective of patient similarity learning is to derive a clinically meaningful metric that can measure the similarity between patients according to their EHRs [7]. Earlier related works are mainly based on traditional machine learning methods. Chan et al. [38] proposed a patient similarity measure that could generate similarity scores for 14 clinical indicators, and then fed these scores into a support vector machine (SVM) classifier to determine the similarity between patients suffering from hepatocellular carcinoma. Sun et al. [39] presented locally supervised metric learning (LSML) to obtain a Mahalanobis metric, and adopted the metric in conjunction with physician feedback to assess patient similarity. Girardi et al. [40] considered the International Classification of Diseases (ICD) concept hierarchy and introduced a hierarchical distance measure to calculate the similarity between patient concept sets.

In recent years, deep learning methods have demonstrated remarkable capabilities in feature extraction and have been increasingly adopted in patient similarity learning. Ni et al. [12] proposed a deep metric learning framework with a quadruple objective for fine-grained patient similarity learning. Suo et al. [7] utilized a convolutional neural network (CNN) to learn patient representations, and optimized the model for learning pairwise similarity through the triplet loss or softmax cross entropy loss. Wang et al. [41] presented a triplet architecture based on a dynamic Bayesian neural network (TDBNN). TDBNN can capture the conditional dependencies among medical indicators and integrate them into multivariate time series analysis to learn fine-grained patient similarity. Zhang et al. [42] proposed to learn the representations of local and global patient states through a unified framework, then measured patient similarity according to the local state representations and predicted in-hospital mortality using the global state representations. After determining the proper patient similarity measures, exploiting these measures to retrieve similar patients efficiently and accurately is a prerequisite to personalized healthcare applications, which is also the research focus of our work.

2.2. Similar patient retrieval

Similar patient retrieval refers to the process of retrieving the patients most similar to the given query patient from a database according to a defined similarity measure [14]. Zhan et al. [10] proposed a generalized Mahalanobis similarity function with pairwise constraints to measure patient similarity, and used training samples as queries to retrieve similar patients from the testing pool after deriving the similarity function. Wang [17] presented an adaptive semi-supervised recursive tree partitioning (ART) framework for patient indexing so that similar patients could be retrieved efficiently and correctly with some pairwise similarity labels annotated by physicians as supervision. Liu et al. [18] proposed a method for similar patient retrieval based on interactive patient labeling and automatic model updating, and designed a visual system to assist patient labeling by physicians. Tashkandi et al. [19] implemented patient similarity analysis using the cosine similarity of patient features, and optimized similarity calculation for in-database similar patient retrieval. Gu et al. [20] organized EHRs as a patient graph to learn encounter-level patient representations for in-database similar patient retrieval. The above similar patient retrieval methods either rely on hand-crafted features, require physicians' manual annotation or feedback, or can only be used for in-database retrieval with a fixed number of patients, and are not well adapted to EHR data with rapidly expanding volumes. Different from the above studies, the proposed GDHN extracts patient features automatically, does not rely on physicians' manual annotation, and has the scalability to the inductive setting, achieving efficient similar patient retrieval.

2.3. Hashing methods

To efficiently retrieve similar patients on large-scale EHR data, a feasible solution is to use hashing methods to accelerate the retrieval process. The earlier hashing methods are mainly data-independent, and the most representative is LSH [21]. LSH adopts a series of hash functions with randomized projection to map the original data into some hash buckets. The closer the two instances are to each other in the original high-dimensional space, the higher the probability that they will be mapped into the same bucket. Due to the lack of consideration of data properties, the data-independent methods tend to underperform in many real-world applications [43]. Therefore, later studies focused more on the data-dependent hashing paradigm known as learning-to-hash.

Learning-to-hash methods can be generally divided into two categories: unsupervised methods and supervised methods. The former learns the hashing functions according to the structure and distribution of data, and can be applied to unlabeled training data. One of the most well-known methods is spectral hashing (SH) [44]. SH regards semantic hashing as a certain type of graph partitioning and uses spectral relaxation to calculate the hash codes efficiently. Later, Gong et al. [45] proposed the iterative quantization (ITQ) algorithm. ITQ learns hashing codes by minimizing the quantization error of mapping PCA-projected data to vertices of the binary hypercube. Other representative unsupervised methods include isotropic hashing (IsoHash) [46], scalable graph hashing with feature transformation (SGH) [47], latent semantic minimal hashing (LSMH) [48], etc. Supervised methods leverage supervised information, including pointwise level, pairwise level, tripletwise level, or listwise level, to generate similarity-preserving hash codes [43]. Kernel-based supervised hashing (KSH) [49] takes the pairwise similarity between instances as supervision, and exploits the algebraic equivalence between the inner products and Hamming distances of the hash codes to learn the kernel-based hash function. Supervised discrete hashing (SDH) [50] assumes that good hash codes should be optimal for linear classification, and utilizes discrete cyclic coordinate descent to efficiently optimize the objective function and derive high-quality hash codes. Additional influential supervised methods include binary reconstructive embeddings (BRE) [51], hashing based

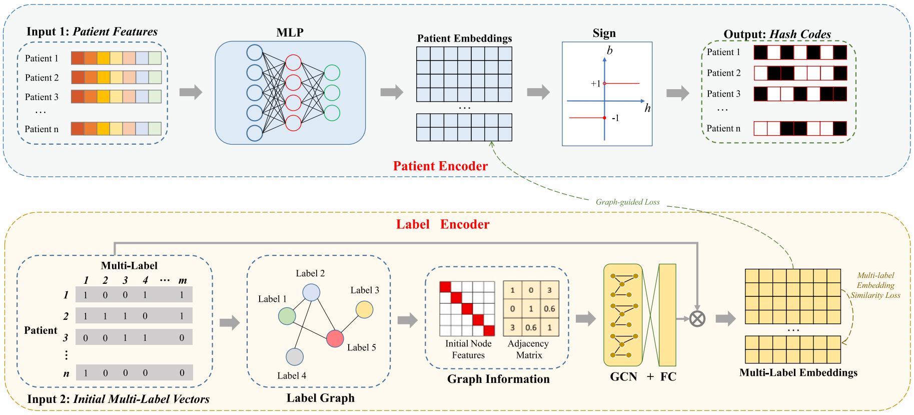


Fig. 3. The overall architecture of the proposed GDHN for similar patient retrieval.

on linear discriminant analysis (LDAHash) [52], and column sampling based discrete supervised hashing (COSDISH) [53], etc.

Deep hashing methods benefit from the powerful feature extraction capability of deep neural networks and have become mainstream of recent learning-to-hash methods. Zhu et al. [22] proposed the deep hashing network (DHN) for supervised hashing. DHN is able to learn representations suitable for hash coding and control quantization errors simultaneously. In order to alleviate the retrieval quality degradation caused by the separate post-step binarization of continuous representations into hash codes, Cao et al. [25] proposed HashNet. HashNet leverages the continuation method with convergence guarantees to learn nearly exactly binary codes from imbalanced similarity data. The above two methods use pairwise similarity as the supervised information, and so do most representative deep hashing methods, e.g., deep supervised hashing (DSH) [23], deep pairwise-supervised hashing (DPSH) [24], and asymmetric deep supervised hashing (ADSH) [54], etc. These methods determine the pairwise similarity coarsely. Two instances are considered similar only if they share at least one label. Such a determination cannot reflect the fine-grained pairwise similarity for multi-label instances. Thus, Zhang et al. [55] proposed an improved deep hashing network (IDHN) that quantifies the pairwise similarity into a percentage based on the normalized semantic labels to improve the multi-label instance retrieval quality.

Recently, some studies have tried to introduce deep hashing into similar patient retrieval. For example, Wang et al. [14]. presented a deep hashing approach adopting multi-task neural networks with attention, and the continuous-valued embedding vectors and binary hash codes of patients obtained from the networks were combined to perform similar patient retrieval in a coarse-to-fine way. Xu et al. [27] proposed the federated patient hashing framework, which derives the binary hash codes of patients in the manner of federated deep learning to alleviate privacy concerns. Compared to traditional hashing methods based on hand-crafted features, deep hashing approaches have shown great progress in retrieval tasks. Nonetheless, most of these deep hashing methods solely use multiple labels of instances as a reference for determining the pairwise similarity, ignoring the rich semantic information inherent in the labels. The proposed GDHN can effectively address this limitation. GDHN organizes the multiple labels as a label graph, and employs a GCN to extract the correlation dependencies between labels and generate multi-label embeddings. The multi-label embeddings are then adopted to guide the patient hashing procedure, ensuring full utilization of the semantic information and correlation dependencies in the multi-labels to generate more informative and discriminative hash codes.

3. Method

In this section, we detail the proposed GDHN. The overview of our framework is demonstrated in Fig. 3. First, we introduce some necessary notations and the problem formulation. Next, the details of the patient encoder and the label encoder are given. Finally, we present the overall objective function and training strategy.

3.1. Notations and problem formulation

We denote the patient data as $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times M}$, where N is the number of patient instances, and $\mathbf{x}_i$ is a M-dimensional feature vector corresponding to the patient i. In this work, we consider the drug categories recorded in the patients' prescriptions as multi-labels. The multi-label matrix is denoted as $\mathbf{L} = \{\mathbf{l}_i\}_{i=1}^{N} \in \{0,1\}^{N\times C}$, where $\mathbf{l}_i = [l_{i1}, l_{i2}, \ldots, l_{iC}]$ is the multi-label annotation of the patient i and C is the total number of distinct drug categories in the dataset. If the patient i is prescribed the jth category drug by a physician, $l_{ij} = 1$, otherwise, $l_{ij} = 0$. Naturally, if two patient instances share some drug categories, they can be considered to have a certain similarity. We denote the similarity set as $\mathcal{S}$, and $s_{ij} \in \mathcal{S}$ represents the similarity between instances i and j. To make better use of semantic information in the multi-labels without losing generality, in this work, we determine the similarity s_{ij} according to the generalized Jaccard index of $\mathbf{l}_i$ and $\mathbf{l}_j$:

$$J(\mathbf{l}_i, \mathbf{l}_j) = \frac{\langle \mathbf{l}_i, \mathbf{l}_j \rangle}{\langle \mathbf{l}_i, \mathbf{l}_i \rangle + \langle \mathbf{l}_j, \mathbf{l}_j \rangle - \langle \mathbf{l}_i, \mathbf{l}_j \rangle}, \tag{1}$$

where $\langle \cdot, \cdot \rangle$ means inner product. Then s_{ij} can be formulated as:

$$s_{ij} = \begin{cases} 1, & J(\mathbf{l}_i, \mathbf{l}_j) > \tau, \\ 0, & J(\mathbf{l}_i, \mathbf{l}_j) \le \tau, \end{cases} \tag{2}$$

where τ is a threshold.

The goal of hashing is to learn a non-linear function to project the original data into compact hash codes while preserving the similarity effectively. The function can be formulated as $f(\cdot) : \mathbf{X} \mapsto \mathbf{B}$, $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^{N} \in \{-1,1\}^{N\times K}$, where $\mathbf{b}_i$ is the K-bit hash code of instance i. Given a pair of hash codes $\mathbf{b}_i$ and $\mathbf{b}_j$, the similarity between them can be measured by the Hamming distance:

$$d_{\mathcal{H}}\left(\mathbf{b}_i, \mathbf{b}_j\right) = \frac{1}{2}\left(K - \langle \mathbf{b}_i, \mathbf{b}_j \rangle\right). \tag{3}$$

Further, we follow the paradigm of learning to hash and define the conditional probability of s_{ij} as:

$$p(s_{ij}|\mathbf{b}_i, \mathbf{b}_j) = \begin{cases} \varphi(\Gamma_{ij}), & s_{ij} = 1, \\ 1 - \varphi(\Gamma_{ij}), & s_{ij} = 0, \end{cases} \\ = \varphi(\Gamma_{ij})^{s_{ij}}(1 - \varphi(\Gamma_{ij}))^{1-sij}, \tag{4}$$

where $\varphi(\Gamma_{ij}) = 1/(1 + e^{-\Gamma_{ij}})$ is the sigmoid function and $\Gamma_{ij} = \langle \mathbf{b}_i, \mathbf{b}_j \rangle$. Apparently, the larger the inner product of $\mathbf{b}_i$ and $\mathbf{b}_j$, the smaller the Hamming distance between them, and the higher the probability of

$s_{ij} = 1$. Conversely, if the inner product between $\mathbf{b}_i$ and $\mathbf{b}_j$ is smaller, the probability of $s_{ij} = 0$ will be higher. The process of learning hash codes $\mathbf{B}$ is equivalent to the maximum likelihood estimation of $\mathbf{B}$ given the pairwise similarity set $\mathcal{S}$.

3.2. Patient encoder

For patient data $\mathbf{X}$, each dimension usually represent a medical indicator, and the ordinal relations among them may not have clear spatial meaning. So we do not necessarily need to use the networks are good at capturing the spatial relations across dimensions, such as CNNs, as the backbone. For the sake of generality, we simply adopt a 3-layer MLP to transform the original instances into K-dimensional hash-like feature vectors. The number of layers is determined following [12]. In order to avoid an excessive number of parameters while ensuring the fitting capability of the patient encoder, the numbers of neurons in the first two layers are both set to $M/2$, which is equal to half the dimensionality of the input patient data. To alleviate the vanishing gradient and make the networks easier to train, we use linear rectification functions (relu) as the activation functions of the first two layers. The third layer is used as the hash layer that outputs the hash-like patient embeddings, and the output dimension is set to be consistent with the hash code lengths. To reduce the quantization errors during the subsequent binarization process, we use a hyperbolic tangent function (tanh) as the activation function in the third layer, which can be viewed as an approximation to the sign function.

This feature extraction process can be expressed as:

$$\mathbf{H} = E_P(\mathbf{X}; \Theta_P), \tag{5}$$

where E_P denotes the patient encoder, Θ_P represents the trainable parameters of E_P, and $\mathbf{H} = \{\mathbf{h}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times K}$ is the output patient embeddings. Note that each element of $\mathbf{h}_i$ is still real-valued at this point, and needs to be binarized to obtain the hash code $\mathbf{b}_i$, which only contains the values of 1 and −1. Referring to [26], the binarization process is usually implemented through the element-wise sign function:.

$$b_{ij} = \mathrm{sgn}(h_{ij}) = \begin{cases} 1, & h_{ij} > 0, \\ -1, & h_{ij} \le 0, \end{cases} \tag{6}$$

where h_{ij} is the jth element of $\mathbf{h}_i$ and b_{ij} is the jth element of $\mathbf{b}_i$.

3.3. Label encoder

Due to the excellent ability of GCNs to deal with the relations between data points [34–37], to better capture and explore the informative label correlations, we adopt a GCN to learn the multi-label embeddings of instances. We first construct a label graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. The drug category label set serves as the node set $\mathcal{V}$, meaning that each node represents a drug category and $|\mathcal{V}| = C$. The initial node feature matrix can be denoted as $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^{C} \in \mathbb{R}^{C\times D}$, where each row $\mathbf{y}_i \in \mathbb{R}^D$ is the D-dimensional feature vector of node i. $\mathcal{E}$ is the edge set, and the edge between node i and node j is denoted as $e_{ij} \in \mathcal{E}$. The topology of $\mathcal{G}$ can be represented as an adjacency matrix $\mathbf{A} \in \mathbb{R}^{C\times C}$, where each element $\mathbf{A}_{ij}$ is the weight of edge e_{ij}. Here we use pointwise mutual information (PMI) to calculate $\mathbf{A}_{ij}$. Given a pair of drug category labels i and j, their PMI value is computed as:

$$\mathrm{PMI}(i,j) = \log\left(\frac{n(i,j)}{n(i)n(j)} \times N\right), \tag{7}$$

where $n(i,j)$ is the number of patients' prescriptions containing both drug categories i and j, and $n(i)$ and $n(j)$ are the total number of patients' prescriptions recording drug category i and j respectively. After obtaining the PMI values, $\mathbf{A}_{ij}$ can be formulated as:

$$\mathbf{A}_{ij} = \begin{cases} \mathrm{PMI}(i,j), & \mathrm{PMI}(i,j) > 0, \\ 0, & \mathrm{PMI}(i,j) \le 0. \end{cases} \tag{8}$$

In order not to lose the information about the nodes themselves, the adjacency matrix usually requires the addition of node self-connections to get $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Accordingly, we can obtain the degree matrix $\tilde{\mathbf{D}}$, where $\tilde{\mathbf{D}}_{ii} = \sum_j \tilde{\mathbf{A}}_{ij}$. Then we feed the graph into a multi-layer GCN for feature extraction. Specifically, the graph convolutional layer takes the following form:

$$\mathbf{Z}^{(q)} = \sigma\left(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{Z}^{(q-1)}\mathbf{W}^{(q-1)}\right), \tag{9}$$

where q is the layer number, $\sigma(\cdot)$ denotes the activation function, $\mathbf{Z}^{(0)} = \mathbf{Y}$ is the input feature matrix, and $\mathbf{Z}^{(q)}$ is the node embedding matrix output by the qth layer. In Eq. (9), $\mathbf{W}^{(q-1)} \in \mathbb{R}^{F\times F'}$ is a layer-specific weight parameter matrix, where F and F' are the input and output dimensions of the layer respectively. Given that we focus more on the drug category co-occurrence information contained in the adjacency matrix $\mathbf{A}$, we simply set $\mathbf{Y} = \mathbf{I}$ as a $C \times C$ identity matrix, which means each drug category is initialized as a one-hot vector. A node can integrate information from its 1st-order neighborhood through a single graph convolutional layer. By stacking multiple graph convolutional layers, a node can further integrate information from its higher-order neighborhoods. However, stacking too many convolutional layers in a GCN may lead to over-smoothing [56], which means the output features of nodes will become indistinguishable and the model performance will decline. Therefore, we follow [57] and adopts just 2 graph convolutional layers in our framework.

To guarantee the fitting capability of the label encoder, the numbers of hidden neurons in each graph convolutional layer are set to the input feature matrix dimensionality C. The activation functions of the graph convolutional layers are set to relu to alleviate the vanishing gradient and make the networks easier to train. The output $\mathbf{Z}^{(2)}$ of the GCN is then fed to a fully connected layer containing K neurons to obtain the hash-like representations $\mathbf{Z}^{(2)\prime} \in \mathbb{R}^{C\times K}$. Note that each row of $\mathbf{Z}^{(2)\prime}$ is the embedding vector of a node. Subsequently, given the multi-label annotation $\mathbf{l}_i$ of patient i, the corresponding multi-label embedding can be derived by integrating the embedding vectors of the drug categories it contains, which is expressed as:

$$\mathbf{z}_i = \sigma\left(\frac{\mathbf{l}_i\mathbf{Z}^{(2)\prime}}{\sum_{j=1}^{C} l_{ij}}\right), \tag{10}$$

where $\mathbf{z}_i \in \mathbb{R}^k$ is the hash-like multi-label embedding of patient i, and $\sigma(\cdot)$ is the tanh activation function. We employ the same activation function as the hash layer of the patient encoder, because the multi-label embeddings output by the label encoder will subsequently used to guide the training process of the patient encoder. The tanh activation function can make sure that the multi-label embeddings have less heterogeneity with the patient embeddings.

The whole process of learning the multi-label embeddings can be formulated as:

$$\mathbf{Z} = E_L(\mathbf{L}, \mathbf{A}, \mathbf{Y}; \Theta_L), \tag{11}$$

where E_L denotes the label encoder, Θ_L is the parameters to be trained, the multi-label matrix $\mathbf{L}$, the adjacency matrix $\mathbf{A}$ of the label graph, and the initial node feature matrix $\mathbf{Y}$ of the label graph are used as inputs, and $\mathbf{Z} = \{\mathbf{z}_i\}_{i=1}^{N} \in \mathbb{R}^{N\times K}$ is the output multi-label embedding matrix.

3.4. Objective function and learning strategy

As mentioned in 3.1, the process of learning hash codes can be viewed as the maximum likelihood estimation of Eq. (4), which is also equivalent to seeking the minimum negative log-likelihood of Eq. (4). Therefore, the loss function of deep hashing can be derived in the following formulation:

$$\begin{aligned} \mathcal{L}_0 &= -\log p(\mathcal{S}|\mathbf{B}) = -\log \sum_{s_{ij}\in\mathcal{S}} p(s_{ij}|\mathbf{b}_i, \mathbf{b}_j) \\ &= \sum_{s_{ij}\in\mathcal{S}} \Big(\log\big(1 + \exp(\langle \mathbf{b}_i, \mathbf{b}_j\rangle)\big) - s_{ij}\langle \mathbf{b}_i, \mathbf{b}_j\rangle\Big). \end{aligned} \tag{12}$$

By minimizing this loss function, we can obtain the exact binary hash codes while preserving the similarity between patient instances. However, this optimization problem is difficult to solve, because directly using Eq. (6) to generate the hash codes will lead to vanishing gradients during training and make the back-propagation infeasible.

Following [25], we adopt continuous relaxation and reformulated the loss function as:

$$\mathcal{L}_1 = \sum_{s_{ij} \in \mathcal{S}} \omega_{ij} \left(\log\left(1 + \exp(\langle \mathbf{h}_i, \mathbf{h}_j \rangle)\right) - s_{ij} \langle \mathbf{h}_i, \mathbf{h}_j \rangle \right), \tag{13}$$

where ω_{ij} is the weight for mitigating data imbalance between the similar and dissimilar pairs, which is defined as $\omega_{ij} = (|\mathcal{S}|/|\mathcal{S}_1|)^{s_{ij}} (|\mathcal{S}|/|\mathcal{S}_0|)^{1-s_{ij}}$, where $\mathcal{S}_1 = \{s_{ij} \in \mathcal{S} : s_{ij} = 1\}$ is the set of similar instance pairs, and $\mathcal{S}_0 = \{s_{ij} \in \mathcal{S} : s_{ij} = 0\}$ is the set of dissimilar instance pairs.

Furthermore, in order to fully exploit the semantic correlations inherent in the label graph, we modify Eq. (13) as:

$$\mathcal{L}_g = \sum_{s_{ij} \in \mathcal{S}} \omega_{ij} \left(\log\left(1 + \exp(\langle \mathbf{h}_i, \mathbf{z}_j \rangle)\right) - s_{ij} \langle \mathbf{h}_i, \mathbf{z}_j \rangle \right), \tag{14}$$

In this way, the semantic correlations contained in the label graph are extracted from the label encoder and transferred to the patient encoder, achieving graph guidance. Additionally, as the guidance information, the multi-label embeddings themselves should also retain the similarity between instances. So we introduce an additional loss function:

$$\mathcal{L}_l = \sum_{s_{ij} \in \mathcal{S}} \omega_{ij} \left(\log\left(1 + \exp(\langle \mathbf{z}_i, \mathbf{z}_j \rangle)\right) - s_{ij} \langle \mathbf{z}_i, \mathbf{z}_j \rangle \right). \tag{15}$$

The final objective function of the overall framework can be formulated as:

$$\min_{\Theta_p, \Theta_l} \mathcal{L} = \mathcal{L}_g + \alpha \mathcal{L}_l, \tag{16}$$

where α is a hyper-parameter that makes a trade-off between $\mathcal{L}_g$ and $\mathcal{L}_l$. Eq. (16) is differentiable, so the model parameters can be optimized by the standard back-propagation algorithm with the mini-batch gradient descent method, and the patient encoder and the label encoder can be jointly trained in an end-to-end manner. To reduce the quantization errors between the real-valued embeddings and the binary hash codes, referring to [25], for the output layers of both encoders, we multiply each value input to the tanh activation function by a scaling factor β. This scaling tanh function has a key property $\lim_{\beta \to \infty} \tanh(\beta x) = \mathrm{sgn}(x)$. By increasing β during training, the tanh function will gradually converge to the sign function, thus decreasing the quantization error. The pseudo-code of training procedure of GDHN is shown in Algorithm 1.

Algorithm 1: Training procedure of GDHN

```
Input: Patient data $\mathbf{X}$; multi-label matrix $\mathbf{L}$; adjacency matrix
    $\mathbf{A}$; node feature matrix $\mathbf{Y}$; similarity matrix $\mathbf{S}$; hash code
    length $K$.
Output: Hash codes $\mathbf{B}$.
Initialization: Encoder parameters $\Theta_P$ and $\Theta_L$;
hyper-parameter $\alpha$; mini-batch size $N_b$; learning rate $\mu$; scaling
parameter $\beta = 1$; update interval of the scaling parameter $T_\beta$;
maximum iteration number $T_{iter} = \lceil \frac{N}{N_b} \rceil$; current iteration
number $iter$; current epoch number $epoch = 1$.
repeat
    if $epoch \bmod T_\beta = 0$ then
        $\beta \leftarrow \sqrt{1 + (1/T_\beta) \times epoch}$;
    end
    for $iter = 1, 2, \cdots, T_{iter}$ do
        Randomly select $N_b$ instances to form a mini-batch;
        Calculate the patient embeddings $\mathbf{H}$ and the multi-label
          embeddings $\mathbf{Z}$ of the current batch through forward
          propagation according to Eqs. (5) and (11);
        Compute the overall loss according to Eq. (16);
        Update $\Theta_P$ and $\Theta_L$ using gradient descent and
          backward propagation;
    end
    $epoch \leftarrow epoch + 1$;
until Converge;
Obtain $\mathbf{B}$ using Eq. (6);
return $\Theta_P$, $\Theta_L$, $\mathbf{B}$.
```

4. Experiments

In this section, we first describe the two datasets used for experimental evaluation. Then we detail the experimental setup, including the evaluation metrics for similar patient retrieval and the competing methods. Finally, we show and analyze the results of the experiments.

4.1. Datasets

IgA Nephropathy is a clinical dataset collected by the Renal Division of Peking University First Hospital [58]. This dataset contains encounters from 843 patients with IgA nephropathy over 13 years, and has undergone an ethical review by the Institutional Review Board (IRB) at Peking University First Hospital. EHRs in this dataset comprise 18 subsets with a total of 141 indicators. The data types covered by these indicators include scalar (e.g., systolic pressure, 24-hour urinary protein quantity, eGFR, etc.), binary (e.g., gender), text (e.g., diagnosis, blood/urine immunoprotein electrophoresis, etc.), and ordinal (e.g., urinary protein level, occult blood level, etc.). We preprocessed these EHR data following the expertise of physicians. Since the text-based indicators are recorded in only a very few encounters, they were converted to binary based on their presence (1 if present, 0 otherwise). The values of ordinal indicators are recorded in the form of ordinal categories, such as negative (−), weak positive (+/−), positive (+), strong positive (++), etc. We assigned ascending integers to encode the values according to the severity. Additionally, we eliminated the indicators that have not been recorded in any encounter, and excluded the encounters with outliers to mitigate the impact of noise. Then the values of scalar and ordinal indicators were normalized. Finally, we selected 1114 encounters with 136 indicators as the experimental instances. Each instance is represented as a 136-dimensional real-valued vector, with each dimension corresponding to an indicator. More details on the description and preprocessing of *IgA Nephropathy* can be found in [20]. As we mentioned before in Section 3.1, the drug categories recorded in each encounter's prescription are regarded as multi-labels. Based on the physicians' expertise, to reflect the patient's conditions relatively precisely, we chose 4th-level Anatomical Therapeutic Chemical (ATC) classification codes[2] as the drug category labels. There are a total of 40 drug categories in *IgA Nephropathy*

MIMIC-III is a publicly available database containing the EHR data of patients admitted to Intensive Care Units (ICUs) at the Beth Israel Deaconess Medical Center [59]. It covers encounters from 38597 distinct adult patients over 11 years. To extract a multi-label dataset suitable for the retrieval task from the original database, we followed [29] and preprocess the raw data using the open-source script.[3] The patient encounters containing prescriptions during the first 24 h were retained because the first 24 h are often the most critical period for patients to be treated promptly and effectively. Furthermore, the duplicate encounters were removed. Finally, 15032 encounters were chosen as experimental instances. The diagnoses and procedures recorded in

[2] https://www.whocc.no/atc_ddd_index/

[3] https://github.com/ycq091044/SafeDrug

Table 1
Statistics of the experimental datasets.

Dataset	Instance	Train	Validation	Query	Retrieve	Feature	Multi-label
IgA Nephropathy	1114	892	111	111	1003	136	40
MIMIC-III	15032	12026	1503	1503	13529	3388	131

these encounters were extracted as the raw features of instances. As the diagnoses and procedures have already been encoded with the ICD-9[4] codes in the EHRs, they can be intuitively converted to multi-hot vectors. The feature vector of each instance was derived by concatenating its corresponding diagnosis and procedure vectors. There are a total of 1958 diagnosis codes and 1430 procedure codes in the extracted dataset, thus each instance is represented as a 3388-dimensional multi-hot vector. Accordingly, the drug categories contained in each encounter's prescription are the corresponding multi-labels. Following [29,60,61], we kept the drug category labels in the form of 3rd-level ATC classification codes. There are a total of 131 drug categories in *MIMIC-III*.

For each of the above datasets, all the instances were randomly divided into training, validation, and test query sets in the ratio of 0.8:0.1:0.1, and the training and validation sets were taken together as the retrieval database. The detailed information on the datasets is summarized in Table 1.

4.2. Experimental setup

4.2.1. Evaluation metrics

We assess the quality of similar patient retrieval with three frequently used metrics: normalized discounted cumulative gains (NDCG) [62], mean average precision (MAP) [63], and precision–recall curves [64].

NDCG is a persuasive evaluation metric for multi-label data retrieval. Given a query q, the DCG score of the top n retrieved instances is formulated as:

$$\mathrm{DCG}@n = \sum_{i=1}^{n} \frac{2^{Rs(q,i)} - 1}{\log(1+i)}, \tag{17}$$

where $Rs(q,i)$ is the relevance score between the query q and the instance i. Note that in this work, we use the generalized Jaccard index $J(\mathbf{l}_q, \mathbf{l}_j)$ as the relevance score $Rs(q,i)$. Then the NDCG score at the position n can be computed through $\mathrm{NDCG}@n = \frac{\mathrm{DCG}@n}{Norm}$, where $Norm$ is a normalization factor equal to the theoretical maximum of DCG@n, also known as the ideal DCG (IDCG) score. A higher NDCG score means better performance.

MAP is a widely used retrieval evaluation metric that is the mean of average precision (AP) scores. For query q, the AP score of the top n retrieved instances can be calculated as:

$$\mathrm{AP}(q)@n = \frac{1}{N_{Rd}(q)@n} \sum_{i=1}^{n} \left(Rd(q,i) \frac{N_{Rd}(q)@i}{i} \right), \tag{18}$$

where $Rd(q,i) \in \{0,1\}$ is a relevance determination indicator function. Without loss of generality, we adopt a relaxed criterion to determine $Rd(q,i)$ following the convention of previous studies. If the instance i and the query q share at least one drug category, or both of them are not prescribed any drug, they are considered to be relevant and $Rd(q,i) = 1$; otherwise, $Rd(q,i) = 0$. $N_{Rd}(q)@i$ is the number of instances relevant to the query q within the top i retrieved results. Then the MAP score is calculated by $\mathrm{MAP} = \frac{1}{Q} \sum_{q=1}^{Q} \mathrm{AP}(q)$, where Q is the cardinal number of the test query set. Similar to NDCG, a higher MAP score indicates superior model performance.

Precision–recall curves plot the precision values at various degrees of recall. When using precision–recall curves to evaluate the model performance, the area under the curve (AUC-PR) is a reasonable measure. The larger the AUC-PR is, the better the performance is.

4.2.2. Competitors and implementation details

We compare our proposed GDHN with several baselines, including well-known traditional hashing methods and state-of-the-art deep hashing methods. A brief description of these competitors and the corresponding parameters is given below.

- **SH** [44] is a classical unsupervised hashing method. It equates semantic hashing with a certain type of graph partitioning and adopts spectral relaxation to obtain hash codes.
- **KSH** [49] is a kernel-based supervised hashing method that exploits the equivalence between the hash codes' inner products and the Hamming distances. The Gaussian radial basis function (RBF) is employed as the kernel to formulate the target hash functions. The number of support samples m is set to 300.
- **SDH** [50] is a supervised discrete hashing method. It treats hashing as a discrete optimization problem, i.e., good hash codes should also be optimal for linear classification. We set the regularization parameter $\lambda = 1$, the penalty parameter $\nu = 1e-5$, and the maximum iteration number $t = 5$. Besides, the number of anchor points m is set to 10 for *IgA Nephropathy* and 800 for *MIMIC-III*, respectively.
- **HashNet** [25] is a famous deep hashing framework that learns nearly exactly binary hash codes from imbalanced similarity data using continuous relaxation with the convergence guarantee. We set the hyper-parameter α, which controls the bandwidth of the adaptive sigmoid function, to $10/K$ for *IgA Nephropathy* and $0.1/K$ for *MIMIC-III*, where K is the hash code length.
- **DHN** [22] is a deep hashing framework capable of simultaneously learning representations suitable for hash coding and controlling quantization errors. We set the quantization penalty parameter λ to 0.001 and 1 for datasets *IgA Nephropathy* and *MIMIC-III*, respectively.
- **DSH** [23] employs a CNN framework, which takes pairs of instances as input and outputs approximate discrete values via an objective function containing contrastive loss terms and a binarization regularizer. The regularization parameter α is set to 0.01, and the margin m of the contrastive loss terms is set to $2K$, where K is the hash code length.
- **DPSH** [24] conducts concurrent learning of feature and hash codes by maximizing the likelihood of the pairwise labels. We set the regularization parameter η to 0.1 for *IgA Nephropathy* and 1 for *MIMIC-III*, respectively.
- **ADSH** [54] is a deep hashing framework applied to the large-scale nearest neighbor search. ADSH only learns the deep hash function for query points, whereas the hash codes for database points are learned directly. We set the iteration numbers $T_{out} = 50$, $T_{in} = 3$, and the constraint coefficient $\gamma = 200$. The numbers of sampled query points $|\Omega|$ are set to 200 for *IgA Nephropathy* and 4000 for *MIMIC-III*, respectively.
- **IDHN** [55] is a deep hashing framework for multi-label instance retrieval. It divides the pairwise instance similarity into "hard similarity" used for the cross-entropy loss, and "soft similarity" used for the mean square error loss. We set the constraint bandwidth controlling parameter $\alpha = 5/K$, the mean square error loss coefficient $\gamma = 0.1/K$, and the quantization loss coefficient $\lambda = 0.1$, where K is the hash code length.

For a fair comparison, all the deep methods use the same backbone as the patient encoder of the proposed GDHN, i.e., the 3-layer MLP mentioned in Section 3.2. The only difference is that the activation

[4] https://www.cdc.gov/nchs/icd/icd9.htm

Table 2
NDCG@10 on *IgA Nephropathy* and *MIMIC-III* datasets.

Method	NDCG@10							
	IgA Nephropathy				*MIMIC III*			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
SH [44]	0.2106	0.2092	0.2029	0.2058	0.4866	0.4885	0.4859	0.4869
SDH [50]	0.1973	0.1971	0.1971	0.1963	0.4009	0.3918	0.3959	0.3923
KSH [49]	0.2283 (0.2331)	0.2235 (0.2269)	0.2226 (0.2263)	0.2192 (0.2249)	0.4776 (*0.5239*)	0.4826 (*0.5282*)	0.4852 (*0.5269*)	0.4891 (*0.5241*)
HashNet [25]	0.2245 (0.2378)	0.2206 (0.2402)	0.2293 (0.2404)	0.2287 (0.2376)	0.4592 (0.4559)	0.4596 (0.4558)	0.4606 (0.4592)	0.4614 (0.4571)
DHN [22]	0.2215 (0.2222)	0.2239 (0.2301)	0.2227 (0.2384)	0.2227 (0.2388)	0.4133 (0.4607)	0.4486 (0.4721)	0.4522 (0.4859)	0.4577 (0.4977)
DSH [23]	0.2187 (0.2161)	0.2266 (0.2165)	0.2194 (0.2182)	0.2215 (0.2202)	0.3882 (0.4646)	0.3915 (0.4691)	0.3882 (0.4674)	0.3882 (0.4712)
DPSH [24]	0.2158 (0.2131)	0.2295 (0.2229)	0.2224 (0.2225)	0.2224 (0.2289)	0.4116 (0.4441)	0.4318 (0.4718)	0.4352 (0.4775)	0.4448 (0.5009)
ADSH [54]	0.2131 (*0.2516*)	0.2122 (*0.2508*)	0.2083 (*0.2518*)	0.2083 (*0.2536*)	0.4545 (0.4558)	0.4525 (0.4608)	0.4518 (0.4618)	0.4514 (0.4601)
IDHN [55]	0.2299	0.2328	0.2337	0.2325	0.4556	0.4580	0.4667	0.4621
GDHN	**0.2746**	**0.2740**	**0.2822**	**0.2753**	**0.5574**	**0.5844**	**0.5957**	**0.5950**

The best results are in boldface, and the second-best results are in italics. For some methods, the results of using the same pairwise similarity determination manner as in their original papers are demonstrated outside parentheses, and those of using the same pairwise similarity determination manner as in our proposed method are shown in parentheses.

functions used in their hash layers are identical to those described in their respective papers. For GDHN, we fix the mini-batch size as 32, and adopt a root mean square propagation (RMSprop) optimizer with a learning rate of $1e-4$ and a weight decay of $5e-4$. The threshold τ in Eq. (2) is chosen to be 0.3, and the hyper-parameter α in Eq. (16) is chosen to be 10 according to the validation set. In addition, since both GDHN and HashNet use the scaled tanh activation function $\tanh(\beta x)$ to reduce quantization errors, the β values of both methods are updated in the same way as in Algorithm 1 for fairness, with the update interval T_β set to 20. To avoid overfitting, all the deep hashing methods, including our GDHN, are trained for no more than 400 epochs and adopt the early-stopping strategy with the patience of 25 epochs.

We implement the traditional hashing methods by Matlab, while the deep hashing methods are implemented using PyTorch [65]. All experiments are conducted on a workstation with an Intel(R) Xeon(R) Gold 5218R @ 2.10 GHz CPU, 251G RAM, and 6 NVIDIA GeForce RTX 3090 GPUs.

4.3. Experimental results

In this section, we compare the performance of the proposed GDHN and the aforementioned competitors on two datasets. For a more comprehensive evaluation, referring to [25], we set four distinct lengths of hash codes generated by each method throughout the experiments, i.e., 16, 32, 48, and 64 bits. The corresponding NDCG and MAP scores of the top 10 retrieved cases (NDCG@10 and MAP@10) are reported as the mean values of 10 random trials, and the precision–recall curves are plotted. Furthermore, the Wilcoxon rank sum test is performed to verify the statistical significance.

4.3.1. Performance comparison in NDCG

Table 2 shows the results of NDCG@10 on the *IgA Nephropathy* and *MIMIC-III* datasets. Note that the pairwise similarity determination of KSH, HashNet, DHN, DSH, DPSH, and ADSH is relatively crude in their original papers, i.e., $s_{ij} = 1$ if instance i and j share no less than one semantic label, otherwise $s_{ij} = 0$. For a fair comparison, we also reproduce these works using the same pairwise similarity determination manner as in our proposed method (Eq. (2)) and present the corresponding results in parentheses. Note that in our experiments, the better results obtained through these two similarity determinations are used to compare performance with GDHN. Although IDHN also requires pairwise similarity as supervised information, we do not make additional changes to it because of its unique approach to determining similarity. By the way, the above setting and clarification of performance evaluation carry over to the MAP metric.

Compared to the second-best method, GDHN achieved improvements of 9.14%, 9.25%, 12.07%, and 8.56% at different code lengths (i.e., 16, 32, 48, and 64 bits) on *IgA Nephropathy*, and 6.82%, 11.44%, 15.00%, and 16.12% at different code lengths on *MIMIC-III*. We can also observe some notable findings for the competitors. Most deep hashing methods do not perform better than the traditional hashing methods on *MIMIC-III*. It may be because the raw features of the instances in the *MIMIC-III* dataset are relatively simple, and the feature extraction capability of deep neural networks does not work well without the help of semantic information in multi-labels. In addition, most of the competitors improve their NDCG scores after adopting the same pairwise similarity determination as in our approach. The possible reason is that our pairwise similarity determination causes the instances with low relevance to the queries to be treated as “dissimilar” and thus ranked lower in the retrieved results, in line with the requirements of the NDCG metric. We also notice that the NDCG scores on *MIMIC-III* are higher than those on *IgA Nephropathy*. It may be due to the fact that, as mentioned in Section 4.1, the raw features of instances in *IgA Nephropathy* contain 4 data types (i.e., scalar, binary, text, and ordinal), whereas the raw features of instances in *MIMIC-III* are solely represented in the code type. The more complex instance features make it more difficult for the retrieval results of each method on *IgA Nephropathy* to be close to the ideal ranking list.

To evaluate the statistical significance of the results in Table 2, we use the Wilcoxon rank sum test [66] to statistically compare the NDCG values over 10 runs of GDHN and each other competitor. Regarding GDHN as the control method, the statistics of the p-value concerning different code lengths are shown in Table 3. In this study, we consider the results of the two methods to be significantly different if the p-value is less than 0.05. It can be found that the statistical significance of GDHN’s superiority over other methods in terms of the NDCG metric is validated. We observe a substantial number of p-values in Table 3 are identical, as our proposal consistently outperforms the competitors and yields the same ranks across most comparisons.

4.3.2. Performance comparison in MAP

The results of MAP@10 on the *IgA Nephropathy* and *MIMIC-III* datasets are presented in Table 4. For the *IgA Nephropathy* dataset, GDHN outperforms the second-best method by 1.24%, 0.27%, 1.11%, and 0.29% at different code lengths, respectively. However, the performance of GDHN on the *MIMIC-III* dataset seems moderate. Compared to the best results, GDHN’s MAP scores exhibit slight decreases of 0.11%, 0.10%, 0.08%, and 0.06% at different code lengths. The quantitative result of GDHN in MAP@10 on the *MIMIC-III* dataset reveals that the relatively lenient criteria employed by the MAP metric in determining the relevance of retrieved results to the query, as mentioned in Section 4.2.1, may cause negative influence to the performance. Due to the same reason, for most of the competitors, using the pairwise similarity determination consistent with ours does not improve the MAP scores.

Though the results on the *MIMIC-III* dataset look impressively favorable in terms of Table 4, the MAP scores of different competitors are intuitively very close. So we also perform the Wilcoxon rank sum test on the MAP scores obtained from 10 trials of the proposed GDHN and

Table 3
p-values of the Wilcoxon rank sum test on NDCG scores where GDHN is the control method.

Method	*IgA Nephropathy*				*MIMIC III*			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
SH [44]	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05
SDH [50]	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05
KSH [49]	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)
HashNet [25]	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (2.17e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)
DHN [22]	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (2.17e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)
DSH [23]	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)
DPSH [24]	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)
ADSH [54]	1.08e−05 (1.08e−05)	1.08e−05 (4.33e−05)	1.08e−05 (2.17e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)
IDHN [55]	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05

Table 4
MAP@10 on *IgA Nephropathy* and *MIMIC-III* datasets.

	MAP@10							
Method	*IgA Nephropathy*				*MIMIC III*			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
SH [44]	0.7409	0.7314	0.7517	0.7483	0.9961	0.9960	0.9965	0.9953
SDH [50]	0.7252	0.7261	0.7249	0.7259	0.9946	0.9944	0.9949	0.9948
KSH [49]	0.7666 (0.7463)	0.7542 (0.7477)	0.7550 (0.7517)	0.7502 (0.7479)	0.9973 (0.9968)	0.9970 (0.9967)	0.9967 (0.9970)	0.9967 (0.9968)
HashNet [25]	0.7921 (0.7745)	0.7846 (0.7730)	0.7779 (0.7669)	0.7781 (0.7747)	0.9977 (0.9978)	0.9973 (0.9976)	0.9974 (0.9976)	0.9974 (0.9977)
DHN [22]	0.7964 (0.7609)	0.8016 (0.7690)	0.7948 (0.7848)	0.7792 (0.7740)	0.9977 (0.9972)	0.9983 (0.9972)	0.9981 (0.9972)	*0.9980* (0.9970)
DSH [23]	*0.8056* (0.8047)	*0.8062* (0.8046)	*0.8108* (0.8104)	*0.8137* (0.8079)	0.9971 (**0.9988**)	0.9969 (**0.9989**)	0.9968 (**0.9986**)	0.9970 (**0.9986**)
DPSH [24]	0.8040 (0.7829)	0.8037 (0.7553)	0.8020 (0.7535)	0.8025 (0.7585)	0.9974 (0.9971)	0.9984 (0.9974)	0.9982 (0.9974)	0.9978 (0.9976)
ADSH [54]	0.8026 (0.7939)	0.8047 (0.7832)	0.7993 (0.7826)	0.8063 (0.7658)	*0.9985* (0.9972)	*0.9985* (0.9969)	*0.9985* (0.9967)	**0.9986** (0.9969)
IDHN [55]	0.7977	0.7916	0.7895	0.7868	0.9977	0.9977	0.9972	0.9978
GDHN	**0.8156**	**0.8134**	**0.8198**	**0.8161**	0.9977	0.9979	0.9978	*0.9980*

Table 5
p-values of the Wilcoxon rank sum test on MAP scores where GDHN is the control method.

Method	*IgA Nephropathy*				*MIMIC III*			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
SH [44]	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.08e−05
SDH [50]	1.08e−05	1.08e−05	1.08e−05	1.08e−05	1.20e−03	7.79e−04	1.94e−02	1.24e−02
KSH [49]	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.08e−05 (1.08e−05)	3.81e−01 (3.03e−04)	3.36e−04 (1.08e−05)	1.08e−05 (1.41e−04)	1.08e−05 (1.08e−05)
HashNet [25]	6.50e−05 (1.08e−05)	2.17e−05 (1.08e−05)	1.08e−05 (1.08e−05)	1.30e−04 (1.08e−05)	9.26e−01 (4.42e−01)	1.95e−01 (1.33e−02)	1.74e−01 (1.35e−01)	8.10e−03 (2.30e−03)
DHN [22]	9.63e−04 (9.63e−04)	2.46e−02 (1.08e−05)	1.90e−03 (3.60e−03)	1.30e−04 (1.08e−05)	6.66e−01 (9.16e−02)	8.66e−04 (1.20e−03)	7.62e−02 (7.10e−03)	6.39e−01 (1.08e−05)
DSH [23]	1.58e−02 (8.10e−03)	1.00e−02 (1.80e−03)	4.92e−02 (3.23e−02)	4.68e−01 (8,19e−02)	2.09e−02 (1.08e−05)	1.20e−03 (2.17e−05)	1.08e−05 (7.58e−05)	1.08e−05 (2.06e−04)
DPSH [24]	9.80e−03 (1.40e−03)	4.13e−02 (1.08e−05)	1.08e−02 (1.08e−05)	2.81e−02 (1.08e−05)	7.00e−02 (1.20e−03)	6.00e−03 (2.87e−01)	3.60e−03 (5.70e−03)	1.41e−01 (1.20e−03)
ADSH [54]	1.10e−03 (2.60e−04)	1.70e−03 (1.84e−04)	6.50e−05 (9.63e−04)	3.39e−02 (1.08e−05)	1.08e−05 (1.67e−02)	1.52e−04 (6.39e−04)	1.52e−04 (1.08e−05)	1.00e−03 (1.08e−05)
IDHN [55]	1.37e−02	1.50e−03	4.33e−05	1.60e−03	5.66e−01	7.08e−02	3.06e−02	2.24e−01

the comparison methods. Table 5 presents the *p*-values at different code lengths, with GDHN as the control method. For the *IgA Nephropathy* dataset, GDHN outperforms other methods significantly, except for DSH when the code length is 64 bits. For the *MIMIC-III* dataset, there are no significant differences between the MAP scores of GDHN and most competing methods. It may be because that, for the patient instances in *MIMIC-III*, the drugs are prescribed within a brief span of the first 24 h [29], and there is a high degree of overlap between the drugs given to different patients in emergencies. While for the *IgA Nephropathy* dataset, the patients are followed up regularly every 3–6 months and prescribed drugs based on their disease conditions by physicians [58]. The drug category labels in *MIMIC-III* may not reflect the medical conditions of patient instances as finely as those in *IgA Nephropathy*, resulting in little difference in the MAP scores of different methods. Moreover, the simplicity of the raw features and the leniency in judging the relevance of retrieved results also lead to a lack of significant disparity in MAP scores on the *MIMIC-III* dataset. Taking the results with respect to *MIMIC-III* in both Tables 4 and 5 into account, we consider it necessary to further verify the superiority of GDHN. Therefore, we have conducted an analytical experiment in the following.

For the instances in the *MIMIC-III* dataset, a total of 131 3rd-level ATC classification codes are used as multi-labels (see Section 4.1). These 3rd-level ATC classification codes can be further grouped into 14 1st-level ATC classification codes. Given each query, the maximum number of the consistent 1st-level ATC classification codes between the query and its top 10 retrieved results is considered, referred to as "the max hit number". That is, the max hit number equal to 14 indicates that among the top 10 retrieved results, there must be at least one instance whose 1st-level ATC classification codes exactly match those of the query. If the most relevant retrieved instance still has one code inconsistent with the query, the max hit number is 13, and so forth. We select several competitors that obtain excellent MAP scores and count the queries with different max hit numbers. During this experiment, we opt for the code length of 32 bits due to the fact that, in terms of the MAP metric, the competitors exhibit more apparent advantages at this particular code length. The mean results obtained from 10 random trials are presented in Fig. 4. It can be observed that in regards to the max hit number, GDHN outperforms other methods. With GDHN, an average of 473.1 queries realize the max hit number of 14, whereas only 291.7 queries reach the max hit number of 14 with the second-best DHN. This finding is promising, proving that GDHN can preferentially return instances with higher similarity to the query, which is more aligned with the requirements of clinical applications.

4.3.3. Precision–recall curves

The precision–recall curves of the various methods on the *IgA Nephropathy* and the *MIMIC-III* datasets are presented in Fig. 5 and Fig. 6, respectively. Considering the results in Table 4, in Fig. 6 we also illustrate the precision–recall curves of the variants of KSH, HashNet, and DSH, which adopt the same pairwise similarity determination as our method. It is clear that GDHN outperforms its competitors in terms

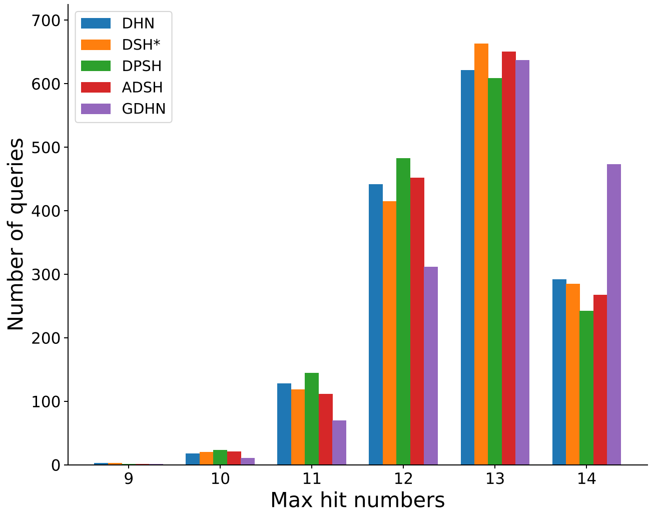
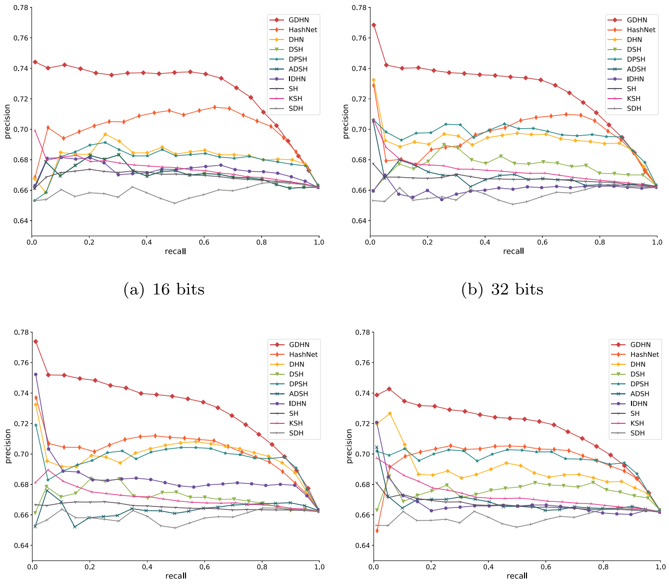


Fig. 4. The number of queries with different max hit numbers on the *MIMIC-III* dataset. All the competitors refer to those with better pairwise similarity determination in MAP scores. DSH* employs the same pairwise similarity determination as GDHN.



(a) 16 bits (b) 32 bits

(c) 48 bits (d) 64 bits

Fig. 5. Precision–recall curves on the *IgA Nephropathy* dataset.

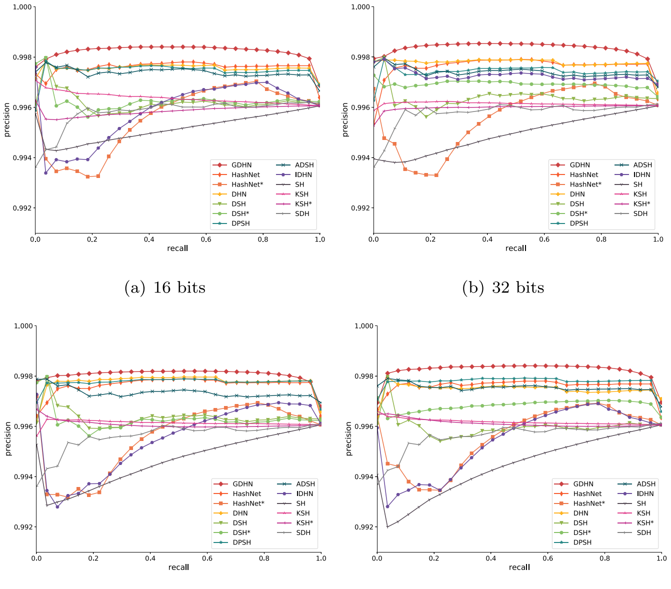
of AUC-PR at different code lengths on both datasets, which further evidences the overall superiority of GDHN.

5. Discussion

In this section, we first conduct an ablation study to validate the efficacy of our design in Section 3. Then, we analyze the influence of the hyper-parameter on GDHN. Finally, we offer some in-depth views of the proposed GDHN, including its adaptability and scalability.

5.1. Ablation study

To gain a better comprehension of how the key ideas contribute to our GDHN for similar patient retrieval, we conduct an ablation study. We mainly focus on the *IgA Nephropathy* dataset and compare GDHN with two variants, GDHN-L and GDHN-G. In GDHN-L, in order not to utilize the semantic information in multi-labels, we remove $\mathcal{L}_l$ in Eq. (16) and change $\mathcal{L}_g$ to $\mathcal{L}_1$ (refer to Eq. (13)), meaning only the patient encoder retained. In GDHN-G, to eliminate the effect of the label graph structure, compared to the full model of GDHN, we substitute the graph convolutional layers in the label encoder with fully connected layers while maintaining the numbers of hidden neurons unchanged. The evaluation results are shown in Table 6.



(a) 16 bits (b) 32 bits

(c) 48 bits (d) 64 bits

Fig. 6. Precision–recall curves on the *MIMIC-III* dataset. The superscript * denotes that the method employs the same pairwise similarity determination as GDHN.

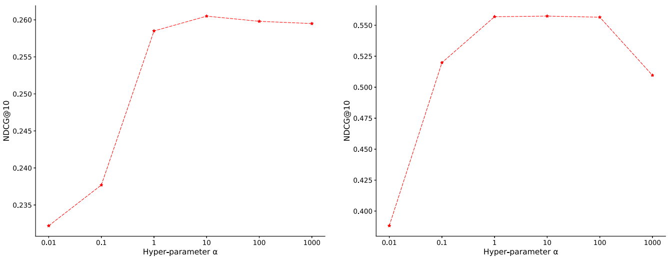
Table 6
Ablation experiments on the *IgA Nephropathy* dataset.

Measure	Method	*IgA Nephropathy*			
		16bit	32bit	48bit	64bit
NDCG@10	GDHN-L	0.2389	0.2428	0.2450	0.2535
	GDHN-G	0.2590	0.2574	0.2593	0.2567
	GDHN	**0.2746**	**0.2740**	**0.2822**	**0.2753**
MAP@10	GDHN-L	0.7821	0.7764	0.7785	0.7753
	GDHN-G	0.8132	0.8040	0.7911	0.8011
	GDHN	**0.8156**	**0.8134**	**0.8198**	**0.8161**

In comparison to the full model, the performance of GDHN-L decreases by 13.00%, 11.39%, 13.18%, and 7.92% at different code lengths in terms of the NDCG metric. For the MAP metric, the performance of GDHN exhibits decreases of 4.11%, 4.55%, 5.04%, and 5.00% at different code lengths. The significant performance degradation strongly supports that the utilization of semantic information contained in multi-labels is critical for generating more informative hash codes. From the evaluation of GDHN-G, we can find that leveraging the semantic information in multi-labels but removing the label graph structure can also lead to a performance decline. The step-by-step performance improvement of the proposed GDHN evidences that appropriately exploring underlying relations among the semantic labels can effectively improve the model's capability in similar patient retrieval.

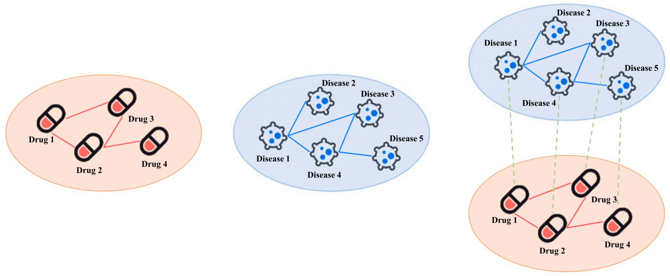
5.2. Parameter analysis

According to Eq. (16), α is a hyper-parameter of GDHN. Fig. 7 shows how the NDCG score changes with the values of α varying on the two



(a) *IgA Nephropathy* (b) *MIMIC-III*

Fig. 7. NDCG@10 of GDHN with different α values on the *IgA Nephropathy* and *MIMIC-III* datasets.



(a) (b) (c)

Fig. 8. Label graphs constructed based on different kinds of multi-labels (taking drug categories and diseases as examples): (a) drug category label graph, (b) disease label graph, (c) drug-disease heterogeneous label graph.

datasets. Note that we only report the results when the code length is 16 bits since the observations display similar trends at the other code lengths. For both of the two datasets, the NDCG scores increase rapidly as α rises from 0.01 to 1. And when the α values exceed 10, the NDCG scores begin to drop. So we set $\alpha = 10$ for both two datasets.

5.3. In-depth views of GDHN

As a deep supervised hashing framework, GDHN in this study chooses drug categories as the multi-labels to supervise its training process. However, in clinical applications, drug categories are not the only option, and diseases, procedures, symptoms and many others recorded in EHRs could also be used as multi-labels. Different kinds of multi-labels can be used to build different label graphs respectively, or be used together to construct heterogeneous label graphs, as indicated in Fig. 8. Once the graph construction is completed, the corresponding adjacency matrix can be input into a GCN to extract semantic information and underlying relations among multi-labels with the proposed GDHN framework. In other words, regardless of the specific multi-labels employed, if they can be organized as graphs, researchers can conveniently adopt our GDHN to process and exploit the EHR data, thus showcasing the adaptability of our proposal. It is worth noting that in order to guarantee the model performance, for different EHR data, the selected multi-labels should reflect the medical conditions of patients as comprehensively as possible.

The retrieval approaches using GCNs usually regard instances as nodes to build graphs, which inevitably limits the scalability of their models. This is because GCNs require a predetermined graph structure. If new instances are added to the database, the graph structure will change, and the entire network parameters will need to be retrained, which is both time and resource consuming. Therefore, using instances as nodes is only suitable for transductive in-database patient retrieval. In contrast, constructing the graph with labels as nodes in GDHN's manner reveals distinct advantages. First, the graph structure is independent of instances and remains unaffected by newly recorded instances. Second, the graph edges are determined according to the co-occurrence of labels in a large number of EHRs, which might reflect the physician's expertise and contain clinical evidence applicable to new patients. Therefore, GDHN has better scalability for rapidly growing EHR data volumes and the inductive setting in clinical applications.

6. Conclusions

This paper introduces a novel deep hashing framework named GDHN for similar patient retrieval on EHR data. GDHN consists of a patient encoder and a label encoder. The patient encoder is used to learn patient representations and generate hash codes. The label encoder organizes the multiple semantic labels (i.e., drug categories) into a graph and acquires the multi-label embeddings of each patient through a GCN. Then, we utilize the multi-label embeddings to improve the patient hashing through a well-designed graph-guided and similarity-preserving objective.

By constructing the label graph and adopting the GCN, GDHN can effectively explore the rich semantic information and underlying relations among the multi-labels recorded in EHRs, thereby obtaining more informative and discriminative hash codes. We conducted extensive experiments on similar patient retrieval using a private dataset *IgA Nephropathy* and a publicly available dataset *MIMIC-III*. The experimental results validate the superiority of GDHN and the respective contributions of various considerations in GDHN.

Our work also has some limitations. First, we focus more on encounter-level similar patient retrieval, wherein each encounter of a patient is treated as an individual instance. This consideration may not be comprehensive enough since some patients need long-term follow-up, and the same patient can have multiple encounters. For the patient data with multiple encounters, models that can capture long-term dependencies are potential to process them. Second, with the increasing diversification of medical testing methods, in addition to the EHRs, patients tend to have medical imaging, genomics, or other medical data. For the different modalities of medical data, using the cross-modal large language models is able to incorporate them. In the future, we will consider whole course medical management and more data sources, and investigate them.

CRediT authorship contribution statement

Yifan Gu: Conceptualization, Investigation, Methodology, Writing – original draft, Writing – review & editing, Software. **Xuebing Yang:** Formal analysis, Funding acquisition, Methodology, Writing – original draft, Writing – review & editing, Investigation. **Mengxuan Sun:** Data curation, Investigation, Methodology, Software. **Chutong Wang:** Investigation, Methodology, Validation, Visualization. **Hongyu Yang:** Data curation, Resources, Validation. **Chao Yang:** Data curation, Resources, Validation. **Jinwei Wang:** Data curation, Resources. **Guilan Kong:** Data curation, Project administration, Resources. **Jicheng Lv:** Data curation, Funding acquisition, Project administration, Resources. **Wensheng Zhang:** Funding acquisition, Investigation, Project administration, Supervision.

Declaration of competing interest

We confirm that this manuscript is entirely original, has not been copyrighted, published, submitted, or accepted for publication elsewhere. We have no conflicts of interest to disclose.

Acknowledgments

The authors are grateful for the fundings of the National Key R&D Program of China (No. 2021ZD0111000), the National Natural Science Foundation of China (Nos. U22B2048, 81925006, 61976212 and 62203437), and National High Level Hospital Clinical Research Funding (Youth clinical research project of Peking University First Hospital, China, No. 2022CR89). The authors would also like to express their thanks to CK-NET and Renal Division of Peking University First Hospital for providing the IgA nephropathy data.

References

[1] J. Lee, D. Maslove, J.A. Dubin, Personalized mortality prediction driven by electronic medical data and a patient similarity metric, PLoS One 10 (5) (2015) e0127428, http://dx.doi.org/10.1371/journal.pone.0127428.

[2] C. Che, C. Xiao, J. Liang, B. Jin, J. Zho, F. Wang, An RNN architecture with dynamic temporal matching for personalized predictions of Parkinson's disease, in: Proceedings of the 2017 SIAM International Conference on Data Mining, SDM, 2017, pp. 198–206, http://dx.doi.org/10.1137/1.9781611974973.23.

[3] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, A. Zhang, J. Gao, Personalized disease prediction using a CNN-based similarity learning method, in: Proceedings of the 2017 IEEE International Conference on Bioinformatics and Biomedicine, BIBM, 2017, pp. 811–816, http://dx.doi.org/10.1109/BIBM.2017.8217759.

[4] Y. Gu, J. Chi, J. Liu, L. Yang, B. Zhang, D. Yu, Y. Zhao, X. Lu, A survey of computer-aided diagnosis of lung nodules from CT scans using deep learning, Comput. Biol. Med. 137 (2021) 104806, http://dx.doi.org/10.1016/j.compbiomed.2021.104806.

[5] A. Heidari, S. Toumaj, N.J. Navimipour, M. Unal, A privacy-aware method for COVID-19 detection in chest CT images using lightweight deep conventional neural network and blockchain, Comput. Biol. Med. 145 (2022) 105461, http://dx.doi.org/10.1016/j.compbiomed.2022.105461.

[6] Q. Yang, Y. Li, B. Li, Y. Gong, A novel multi-class classification model for schizophrenia, bipolar disorder and healthy controls using comprehensive transcriptomic data, Comput. Biol. Med. 148 (2022) 105956, http://dx.doi.org/10.1016/j.compbiomed.2022.105956.

[7] Q. Suo, F. Ma, Y. Yuan, M. Huai, W. Zhong, J. Gao, A. Zhang, Deep patient similarity learning for personalized healthcare, IEEE Trans. NanoBiosci. 17 (3) (2018) 219–227, http://dx.doi.org/10.1109/TNB.2018.2837622.

[8] E. Parimbelli, S. Marini, L. Sacchi, R. Bellazzi, Patient similarity for precision medicine: A systematic review, J. Biomed. Inform. 83 (2018) 87–96, http://dx.doi.org/10.1016/j.jbi.2018.06.001.

[9] A. Gottlieb, G.Y. Stein, E. Ruppin, R.B. Altman, R. Sharan, A method for inferring medical diagnoses from patient similarities, BMC Med. 11 (194) (2013) http://dx.doi.org/10.1186/1741-7015-11-194.

[10] M. Zhan, S. Cao, B. Qian, S. Chang, J. Wei, Low-rank sparse feature selection for patient similarity learning, in: Proceedings of the 2016 IEEE 16th International Conference on Data Mining, 2016, pp. 1335–1340, http://dx.doi.org/10.1109/ICDM.2016.0182.

[11] M.M. Masud, K. Hayawi, S.S. Mathew, A. Dirir, M. Cheratta, Effective patient similarity computation for clinical decision support using time series and static data, in: Proceedings of the Australasian Computer Science Week Multiconference, 2020, pp. 1–8, http://dx.doi.org/10.1145/3373017.3373050.

[12] J. Ni, J. Liu, C. Zhang, D. Ye, Z. Ma, Fine-grained patient similarity measuring using deep metric learning, in: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 1189–1198, http://dx.doi.org/10.1145/3132847.3133022.

[13] Z. Zhu, C. Yin, B. Qian, Y. Cheng, J. Wei, F. Wang, Measuring patient similarities via a deep architecture with medical concept embedding, in: Proceedings of the 2016 IEEE 16th International Conference on Data Mining, 2016, pp. 749–758, http://dx.doi.org/10.1109/ICDM.2016.0086.

[14] K. Wang, E. Xia, S. Zhao, Z. Huang, S. Huang, J. Mei, S. Li, Fast similar patient retrieval from large scale healthcare data: A deep learning-based binary hashing approach, in: Explainable AI in Healthcare and Medicine: Building a Culture of Transparency and Accountability, Springer International Publishing, Cham, 2021, pp. 11–21, http://dx.doi.org/10.1007/978-3-030-53352-6_2.

[15] K. Zhang, S. Qi, J. Cai, D. Zhao, T. Yu, Y. Yue, Y. Yao, W. Qian, Content-based image retrieval with a convolutional siamese neural network: Distinguishing lung cancer and tuberculosis in CT images, Comput. Biol. Med. 140 (2022) 105096, http://dx.doi.org/10.1016/j.compbiomed.2021.105096.

[16] A. Minutolo, E. Damiano, G. De Pietro, H. Fujita, M. Esposito, A conversational agent for querying Italian patient information leaflets and improving health literacy, Comput. Biol. Med. 141 (2022) 105004, http://dx.doi.org/10.1016/j.compbiomed.2021.105004.

[17] F. Wang, Adaptive semi-supervised recursive tree partitioning: The ART towards large scale patient indexing in personalized healthcare, J. Biomed. Inform. 55 (2015) 41–54, http://dx.doi.org/10.1016/j.jbi.2015.01.009.

[18] H. Liu, H. Dai, J. Chen, J. Xu, Y. Tao, H. Lin, Interactive similar patient retrieval for visual summary of patient outcomes, J. Vis. 26 (2023) 577–592.

[19] A. Tashkandi, I. Wiese, L. Wiese, Efficient in-database patient similarity analysis for personalized medical decision support systems, Big Data Res. 13 (2018) 52–64, http://dx.doi.org/10.1016/j.bdr.2018.05.001.

[20] Y. Gu, X. Yang, L. Tian, H. Yang, J. Lv, C. Yang, J. Wang, J. Xi, G. Kong, W. Zhang, Structure-aware siamese graph neural networks for encounter-level patient similarity learning, J. Biomed. Inform. 127 (2022) 104027, http://dx.doi.org/10.1016/j.jbi.2022.104027.

[21] P. Indyk, R. Motwani, Approximate nearest neighbors: Towards removing the curse of dimensionality, in: Proceedings of the 30th Annual ACM Symposium on Theory of Computing, 1998, pp. 604–613.

[22] H. Zhu, M. Long, J. Wang, Y. Cao, Deep hashing network for efficient similarity retrieval, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016, pp. 2415–2421, http://dx.doi.org/10.1609/aaai.v30i1.10235.

[23] H. Liu, R. Wang, S. Shan, X. Chen, Deep supervised hashing for fast image retrieval, in: Proceeding of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2064–2072, http://dx.doi.org/10.1109/CVPR.2016.227.

[24] W.-J. Li, S. Wang, W.-C. Kang, Feature learning based deep supervised hashing with pairwise labels, in: Proceedings of the 25th International Joint Conference on Artificial Intelligence, 2016, pp. 1711–1717.

[25] Z. Cao, M. Long, J. Wang, P.S. Yu, HashNet: Deep learning to hash by continuation, in: Proceedings of the 2017 IEEE International Conference on Computer Vision, 2017, pp. 5609–5618.

[26] A. Singh, S. Gupta, Learning to hash: A comprehensive survey of deep learning-based hashing methods, Knowl. Inf. Syst. 64 (2022) 2565–2597, http://dx.doi.org/10.1007/s10115-022-01734-0.

[27] J. Xu, Z. Xu, P. Walker, F. Wang, Federated patient hashing, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 6486–6493.

[28] E. Choi, M.T. Bahadori, L. Song, W.F. Stewart, J. Sun, GRAM: Graph-based attention model for healthcare representation learning, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 787–795, http://dx.doi.org/10.1145/3097983.3098126.

[29] C. Yang, C. Xiao, F. Ma, L. Glass, J. Sun, SafeDrug: Dual molecular graph encoders for safe drug recommendations, in: Proceedings of the 30th International Joint Conference on Artificial Intelligence, 2021.

[30] X. Zou, S. Wu, N. Zhang, E.M. Bakker, Multi-label modality enhanced attention based self-supervised deep cross-modal hashing, Knowl.-Based Syst. 239 (2022) 107927, http://dx.doi.org/10.1016/j.knosys.2021.107927.

[31] Y. Duan, N. Chen, P. Zhang, N. Kumar, L. Chang, W. Wen, MS^{2}GAH: Multi-label semantic supervised graph attention hashing for robust cross-modal retrieval, Pattern Recognit. 128 (2022) 108676, http://dx.doi.org/10.1016/j.patcog.2022.108676.

[32] Y. Ren, Y. Shi, K. Zhang, X. Wang, Z. Chen, H. Li, A drug recommendation model based on message propagation and DDI gating mechanism, IEEE J. Biomed. Health Inf. 26 (7) (2022) 3478–3485, http://dx.doi.org/10.1109/JBHI.2022.3153342.

[33] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P.S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Netw. Learn. Syst. 32 (1) (2021) 4–24, http://dx.doi.org/10.1109/TNNLS.2020.2978386.

[34] Z.-M. Chen, X.-S. Wei, P. Wang, Y. Guo, Multi-label image recognition with graph convolutional networks, in: Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 5172–5181, http://dx.doi.org/10.1109/CVPR.2019.00532.

[35] Y. Wang, D. He, F. Li, X. Long, Z. Zhou, J. Ma, S. Wen, Multi-label classification with label graph superimposing, in: Proceedings of the 34th AAAI Conference on Artificial Intelligence, 2020, pp. 12265–12272.

[36] Y. Li, H. Zhou, Y. Yin, J. Gao, Multi-label pattern image retrieval via attention mechanism driven graph convolutional network, in: Proceedings of the 29th ACM International Conference on Multimedia, 2021, pp. 300–308, http://dx.doi.org/10.1145/3474085.3475695.

[37] S. Qian, D. Xue, Q. Fang, C. Xu, Integrating multi-label contrastive learning with dual adversarial graph neural networks for cross-modal retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 45 (4) (2023) 4794–4811, http://dx.doi.org/10.1109/TPAMI.2022.3188547.

[38] L.W.C. Chan, T. Chan, L.F. Cheng, W.S. Mak, Machine learning of patient similarity: A case study on predicting survival in cancer patient after locoregional chemotherapy, in: Proceedings of 2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops, 2010, pp. 467–470, http://dx.doi.org/10.1109/BIBMW.2010.5703846.

[39] J. Sun, F. Wang, J. Hu, S. Edabollahi, Supervised patient similarity measure of heterogeneous patient records, ACM SIGKDD Explor. Newsl. 14 (1) (2012) 16–24, http://dx.doi.org/10.1145/2408736.2408740.

[40] D. Girardi, S. Wartner, G. Halmerbauer, M. Ehrenmüller, H. Kosorus, S. Dreiseitl, Using concept hierarchies to improve calculation of patient similarity, J. Biomed. Inform. 63 (2016) 66–73, http://dx.doi.org/10.1016/j.jbi.2016.07.021.

[41] Y. Wang, W. Chen, B. Li, R. Boots, Learning fine-grained patient similarity with dynamic Bayesian network embedded RNNs, in: Database Systems for Advanced Applications, 2019, pp. 587–603.

[42] X. Zhang, B. Qian, Y. Li, Z. Gao, C. Guan, R. Wang, Y. Zheng, H. Zheng, C. Li, Learning representations from local to global for fine-grained patient similarity measuring in intensive care unit, in: Proceedings of the 2022 IEEE 22nd International Conference on Data Mining, 2022, pp. 713–722, http://dx.doi.org/10.1109/ICDM54844.2022.00082.

[43] J. Wang, W. Liu, S. Kumar, S.-F. Chang, Learning to hash for indexing big data–A survey, Proc. IEEE 104 (1) (2016) 34–57, http://dx.doi.org/10.1109/JPROC.2015.2487976.

[44] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Proceedings of Advances in Neural Information Processing Systems, 2009, pp. 1753–1760.

[45] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2013) 2916–2929, http://dx.doi.org/10.1109/TPAMI.2012.193.

[46] W. Kong, W.-J. Li, Isotropic hashing, in: Proceedings of Advances in Neural Information Processing Systems, vol. 25, 2012, pp. 1655–1663.

[47] Q. Jiang, W.-J. Li, Scalable graph hashing with feature transformation, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015, pp. 2248–2254.

[48] X. Lu, X. Zheng, X. Li, Latent semantic minimal hashing for image retrieval, IEEE Trans. Image Process. 26 (1) (2017) 355–368, http://dx.doi.org/10.1109/TIP.2016.2627801.

[49] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, S.-F. Chang, Supervised hashing with kernels, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2074–2081, http://dx.doi.org/10.1109/CVPR.2012.6247912.

[50] F. Shen, C. Shen, W. Liu, H.T. Shen, Supervised discrete hashing, in: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 37–45.

[51] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: Proceedings of Advances in Neural Information Processing Systems, vol. 22, 2009, pp. 1042–1050.

[52] C. Strecha, A. Bronstein, M. Bronstein, P. Fua, LDAHash: Improved matching with smaller descriptors, IEEE Trans. Pattern Anal. Mach. Intell. 34 (1) (2012) 66–78, http://dx.doi.org/10.1109/TPAMI.2011.103.

[53] W.-C. Kang, W.-J. Li, Z.-H. Zhou, Column sampling based discrete supervised hashing, in: Proceedings of the 30th AAAI Conference on Artificial Intelligence, 2016, pp. 1230–1236, http://dx.doi.org/10.1609/aaai.v30i1.10176.

[54] Q.-Y. Jiang, W.-J. Li, Asymmetric deep supervised hashing, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 3342–3349.

[55] Z. Zhang, Q. Zou, Y. Lin, L. Chen, S. Wang, Improved deep hashing with soft pairwise similarity for multi-label image retrieval, IEEE Trans. Multimed. 22 (2) (2020) 540–553, http://dx.doi.org/10.1109/TMM.2019.2929957.

[56] Q. Li, Z. Han, X.-M. Wu, Deeper insights into graph convolutional networks for semi-supervised learning, in: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, 2018, pp. 3538–3545.

[57] T. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of the 5th International Conference on Learning Representations, 2017.

[58] G.-Z. Yu, L. Guo, J.-F. Dong, S.-F. Shi, L.-J. Liu, J.-W. Wang, G.-L. Sui, X.-J. Zhou, Y. Xing, H.-X. Li, J.-C. Lv, H. Zhang, Persistent hematuria and kidney disease progression in IgA nephropathy: A cohort study, Am. J. Kidney Dis. 76 (1) (2020) 90–99, http://dx.doi.org/10.1053/j.ajkd.2019.11.008.

[59] A.E.W. Johnson, T.J. Pollard, L. Shen, L.H. Lehman, M. Feng, M. Ghassemi, B. Moody, P. Szolovits, L.A. Celi, R.G. Mark, MIMIC-III, a freely accessible critical care database, Sci. Data 3 (2016) 160035, http://dx.doi.org/10.1038/sdata.2016.35.

[60] J. Shang, C. Xiao, T. Ma, H. Li, J. Sun, GAMENet: Graph augmented memory networks for recommending medication combination, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 1126–1133, http://dx.doi.org/10.1609/aaai.v33i01.33011126.

[61] M. Sun, J. Niu, X. Yang, Y. Gu, W. Zhang, CEHMR: Curriculum learning enhanced hierarchical multi-label classification for medication recommendation, Artif. Intell. Med. 143 (2023) 102613, http://dx.doi.org/10.1016/j.artmed.2023.102613.

[62] K. Järvelin, J. Kekäläinen, Cumulated gain-based evaluation of IR techniques, ACM Trans. Inf. Syst. 20 (4) (2002) 422–446, http://dx.doi.org/10.1145/582415.582418.

[63] S.M. Beitzel, E.C. Jensen, O. Frieder, MAP, in: Encyclopedia of Database Systems, Springer US, Boston, MA, 2009, pp. 1691–1692, http://dx.doi.org/10.1007/978-0-387-39940-9_492.

[64] N. Japkowicz, Assessment metrics for imbalanced learning, in: Imbalanced Learning, John Wiley & Sons, Ltd, 2013, pp. 187–206, http://dx.doi.org/10.1002/9781118646106.ch8, Ch. 8.

[65] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, PyTorch: An imperative style, high-performance deep learning library, in: Proceedings of the 33rd Conference on Neural Information Processing Systems, 2019, pp. 8024–8035.

[66] M. Hollander, J. Sethuraman, Nonparametric statistics: Rank-based methods, in: International Encyclopedia of the Social & Behavioral Sciences (Second Edition), Elsevier, Oxford, 2015, pp. 891–897, http://dx.doi.org/10.1016/B978-0-08-097086-8.42157-4.