Deep Hash-based Relevance-aware Data Quality Assessment for Image Dark Data

YU LIU and YANGTAO WANG, Huazhong University of Science and Technology, China
LIANLI GAO, University of Electronic Science and Technology of China, China
CHAN GUO and YANZHAO XIE, Huazhong University of Science and Technology, China
ZHILI XIAO, Tencent Inc., China

Data mining can hardly solve but always faces a problem that there is little meaningful information within the dataset serving a given requirement. Faced with multiple unknown datasets, to allocate data mining resources to acquire more desired data, it is necessary to establish a data quality assessment framework based on the relevance between the dataset and requirements. This framework can help the user to judge the potential benefits in advance, so as to optimize the resource allocation to those candidates. However, the unstructured data (e.g., image data) often presents dark data states, which makes it tricky for the user to understand the relevance based on content of the dataset in real time. Even if all data have label descriptions, how to measure the relevance between data efficiently under semantic propagation remains an urgent problem. Based on this, we propose a Deep Hash-based Relevance-aware Data Quality Assessment framework, which contains off-line learning and relevance mining parts as well as an on-line assessing part. In the off-line part, we first design a Graph Convolution Network (GCN)-AutoEncoder hash (GAH) algorithm to recognize the data (i.e., lighten the dark data), then construct a graph with restricted Hamming distance, and finally design a Cluster PageRank (CPR) algorithm to calculate the importance score for each node (image) so as to obtain the relevance representation based on semantic propagation. In the on-line part, we first retrieve the importance score by hash codes and then quickly get the assessment conclusion in the importance list. On the one hand, the introduction of GCN and co-occurrence probability in the GAH promotes the perception ability for dark data. On the other hand, the design of CPR utilizes hash collision to reduce the scale of graph and iteration matrix, which greatly decreases the consumption of space and computing resources. We conduct extensive experiments on both single-label and multi-label datasets to assess the relevance between data and requirements as well as test the resources allocation. Experimental results show our framework can gain the most desired data with the same mining resources. Besides, the test results on Tencent1M dataset demonstrate the framework can complete the assessment with a stability for given different requirements.

CCS Concepts: • **Computing methodologies → Image representations**;

Additional Key Words and Phrases: Resource allocation, data mining, data quality assessment, relevance, GAH, CPR

This work is supported by the Innovation Group Project of the National Natural Science Foundation of China No. 61821003 and the National Key Research and Development Program of China under grant No. 2016YFB0800402 and the National Natural Science Foundation of China No. 61902135.
Authors' addresses: Y. Liu, Y. Wang (corresponding author), C. Guo, and Y. Xie, Huazhong University of Science and Technology, 1037 Luoyu Road, Wuhan, China; emails: {liu_yu, ytwbruce, guochan0554, yzxie}@hust.edu.cn; L. Gao, University of Electronic Science and Technology of China, 2006 Xiyuan Avenue, Chengdu, China; email: lianli.gao@uestc.edu.cn; Z. Xiao, Tencent Inc., Shenzhen, China; email: tomxiao@tencent.com.

2577-3224/2021/03-ART11 $15.00
https://doi.org/10.1145/3420038

ACM Reference format:
Yu Liu, Yangtao Wang, Lianli Gao, Chan Guo, Yanzhao Xie, and Zhili Xiao. 2021. Deep Hash-based Relevance-aware Data Quality Assessment for Image Dark Data. *ACM/IMS Trans. Data Sci.* 2, 2, Article 11 (March 2021), 26 pages.
https://doi.org/10.1145/3420038

1 INTRODUCTION

Data mining has always been a hot topic in the field of information science and has continuously made great progress with data inflation. However, the enrichment of data content and the explosive growth of data indirectly dilutes meaningful information, and thus simply improving the ability of data mining algorithms becomes insufficient to extract valuable information from mass data. In practice, there is no guarantee that the data being mined contain enough information that is valuable to the mining target. Blindly conducting data mining will likely bring the results shown in Figure 1. It is necessary to optimize data mining resource allocation to make limited resources applied to those more appropriate datasets and thus acquire more desired data.

This allocation is determined by possible benefits from the dataset. In fact, this is an issue of data quality (DQ) assessment. DQ is defined as "fitness for use" and includes five evaluation dimensions, i.e., availability, reliability, relevance, usability, and presentation quality, where the relevance is defined as the degree of relevance between the content of data and the user's expectations or demands [32], which helps pre-judge the result of data mining. With limited resources, it is necessary to evaluate and ameliorate the data relevance quality according to the given requirement before mining, which reasonably allocates resources to capture the most desired data.

However, as the basis of evaluating relevance, understanding the content distribution of dataset and the desired content of requirement become highly tricky. Most of the datasets in the open scenarios present dark data states. They are unstructured, untagged and untapped [28], which brings great challenges for the user to real-time understand the relevance between data and requirements. Besides, this phenomenon is popular for unstructured data such as images and videos. According to a study by IBM, over 80% of all data are dark and unstructured, and this will rise to 93% by 2020.[1] There are two reasons resulting in this phenomenon. First, the analysis efficiency of such data lags far behind its storage efficiency, resulting in a large amount of data without labels, which hinders the exposure of relevance between data and requirements. Second, most of such data will not be reused after generated, leading to the lack of associated information, which degrades a large amount of data into the dark data, thereby exacerbating the difficulty of real-time understanding the relevance between data and requirements [23, 49].

However, there is a lack of assessment work on DQ in terms of relevance. The data cleaning method is regarded as a solution to ameliorate DQ, but the existing methods cannot real-time strip invalid data according to the content relevance when faced with dark data [1]. Data retrieval methods can form datasets based on the similar data, which sounds like a good choice for ameliorating relevance quality, especially considering that this method has achieved good results based on content semantics for unstructured data [29]. However, related data are not equivalent to similar data. As shown in Figure 2, related data include not only the same and similar data but also semantically propagated data. For example, although *dog* and *horse* are not similar, they are related through a chain. This relevance cannot be captured by simply using the retrieval technology.

In this article, we propose a deep hash-based relevance-aware data quality assessment framework (DHR-DQA) for image dark data, which evaluates the relevance of the image dark dataset

[1] https://www.answerminer.com/blog/dark-data.

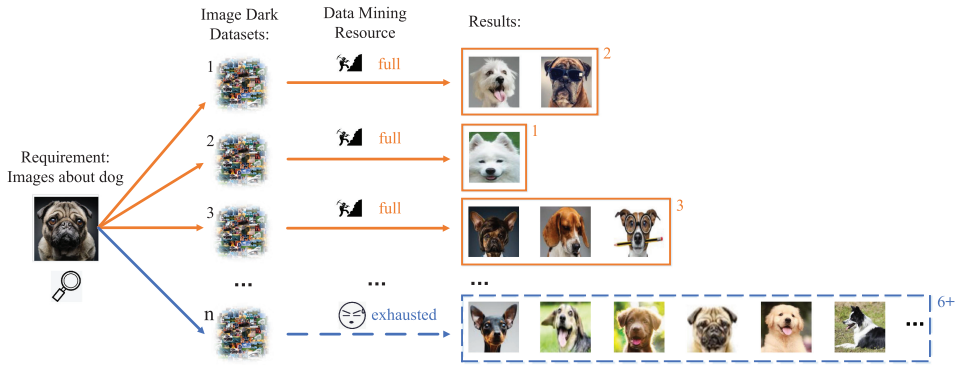


Fig. 1. The user's requirement is to mine images about dogs from n datasets whose content are untapped. With limited resources, the user conducted data mining in order and obtained 2, 1 and 3 relevant images form the first 3 datasets respectively. Although the nth dataset contains more than 6 matched images, it has not been explored, because the data mining resources are exhausted at this moment. The user has not efficiently utilized data mining resources and thus missed much desired data.

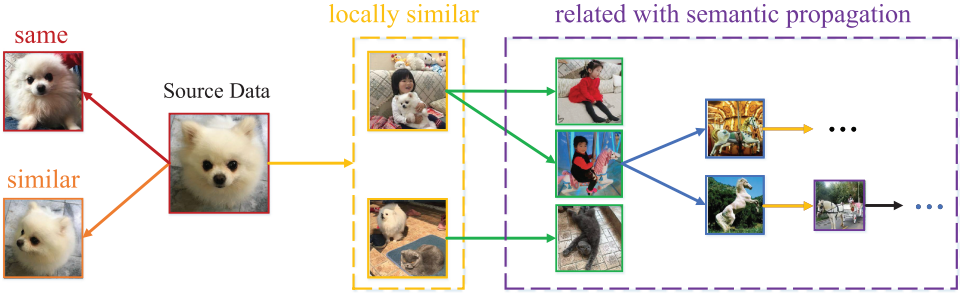


Fig. 2. Dog, as the source data, is related to the same, similar, and locally similar data. Besides, in locally similar data, child, as new content differening from the source data, is semantically propagated and then passed to the carousel while the carousel is passed to the carriage, thereby forming an endless chain. These semantically propagated data are also related to the source data, but the relevance degree of these data is inferior to that of those same, similar, and locally similar data.

quality according to the given requirement. Note that hash codes greatly speed up the graphing process, which promotes the evaluation efficiency by Hamming space retrieval. Facing multiple dark datasets, the users can invest limited mining resources in datasets that can produce the highest value. Based on deep image hashing and retrieval technology, our framework introduces the semantic importance ranking and completes the DQ assessment in terms of relevance. For the relevance evaluation, DHR-DQA (1) utilizes a semi-supervised deep hash method with generalization ability to perceive the data distribution in the dark dataset and (2) gives comprehensive assessment values according to the relevance degree of all data. In the implement, DHR-DQA offline completes the hash and semantic ranking process and on-line achieves assessment with the input of agent data (AD) to ensure the efficiency in large-scale scenarios. To evaluate our work, we implement our DHR-DQA on self-designed sub-datasets based on public datasets. Extensive experiments show that our DHR-DQA not only helps the user obtain the most desired data with limited resources but also has higher efficiency than other schemes. At last, we show the relevance assessment result on real-world datasets for requirements, which verifies our framework can be widely and robustly used for different requirements.

The major contributions of this article are summarized as follows:

- We improve the calculation-query-assessment framework consisting of off-line calculation and on-line assessment for image dark data on the precision of evaluation and graphing efficiency, especially for multi-label images.
- We propose a semi-supervised deep hash method called GCN-AutoEncoder hashing (GAH), which utilizes Graph Convolution Network (GCN) to improve the recognition ability of GAH for multi-label images to enable a more practical hash model.
- We propose a Cluster PageRank (CPR) algorithm to make full use of hash collision, which improves graphing efficiency by merging the same hash codes into one node.

The remainder of this article is organized as follows. Section 2 introduces the motivation of our work. We illustrate the detailed design of DHR-DQA in Section 3, followed by the description of our proposed GCN-AutoEncoder Hashing algorithm in Section 4 and Cluster PageRank algorithm in Section 5. Section 6 compares the experimental results of DHR-DQA with the state-of-the-art methods, and Section 7 talks about the related works. At last, we conclude this article in Section 8.

2 MOTIVATION

Measuring the relevance plays a vital role for the user to effectively allocate the computing resources in data mining. There are two-level evaluations for the relevance: (1) the amount of data that meets the user's requirements and (2) the matching degree of these relevant data [3]. However, subject to the incomplete structured information of dark data and the diversity of requirements, it is not easy to quantify the evaluation, especially for unstructured data. Moreover, the existing work lacks research on measuring DQ in terms of relevance. Based on this, we take the representative unstructured data, i.e., images, as our research object. Assuming that all the images that remain to be mined present dark data states, we study the DQ assessment method of measuring the relevance between dataset and requirements.

Considering the two-level evaluations for the relevance, we must first make clear what data are related to the requirement. Due to the semantic propagation shown in Figure 2, all the data in the dataset must be taken into consideration, but only the degree of relevance needs to be measured. Therefore, the measurement process is implemented as follows. First, the content semantics of requirements (represented by those images that contain one or more objects) are materialized to form AD. Then, AD is added to the dataset, and we conduct the content semantic feature generation and relevance computation for all data. Finally, the relevance between data and a given requirement is determined according to the relevance degree of all data. There are three key challenges in the process.

(1) Data lightening and semantic representation generating for the dark data. Understanding the content semantics of data is the basis of measuring the relevance, so the first step is to lighten dark data. According to the definition in Reference [10], lightening dark data means getting the data content representations and relationships. Classification models based on deep learning have great advantages in generating image representations. However, conventional deep models subject to the distribution of training samples and thus fail to achieve a good result when applied to the dataset with unknown data distribution. However, the relationships between different data have to be established according to those representations, but the numerical label (e.g., 1, 2) or text annotation (e.g., cat, dog) cannot accomplish this task intuitively, so it becomes necessary to find other type of representations. In our previous dark data assessment work [22], we adopt a deep self-taught hashing (DSTH) [21] algorithm to lighten dark data, which generates a deep model with generalization ability by learning the pseudo labels to overcome the problem of inconsistent distribution. Besides, DSTH utilizes the similarity hashing technique to embed the content

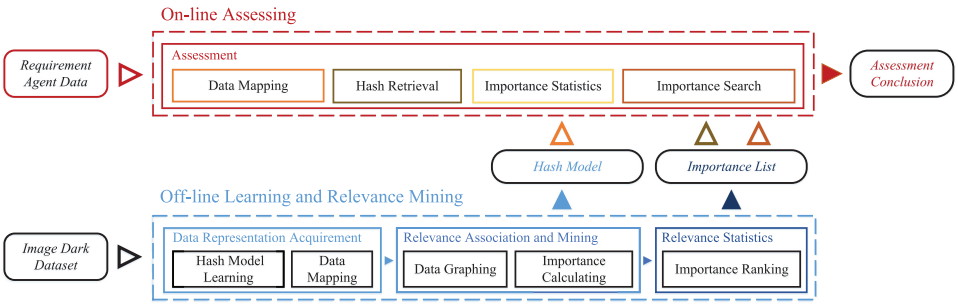


Fig. 3. DHR-DQA framework.

semantics of data and thus can fast measure the similarity by Hamming distance. However, in practice, we find DSTH [48] has a defect in the learning process, which spends much extra space in constructing the graph with all data. In addition, it owns a good generalization ability for single-label images but fails to obtain a great result on multi-label images, while the images in the real world tend to contain multiple objects, and the candidate algorithm must have the ability to recognize locally similar images. Therefore, it is necessary to design a deep hashing algorithm to promote the semantic expression ability on multi-label images with less resource overhead.

(2) Relevance evaluation under semantic propagation. Using similarity hash codes as representations can efficiently construct relationships for all data, which calculates the degree of similarity according to the Hamming distance between data. For the requirement, we can measure the relevance by calculating the distance between the hash codes that respectively correspond to this requirement and each data in the dataset, but this approach may lead to low efficiency in the large-scale scenarios. For a fixed dataset, our previous DQ assessment work [22] adopted the PageRank- [33] Semantic Hash Ranking (SHR) algorithm to calculate the semantic importance score for each data and then determined the metric value according to the ranking list corresponding to the hash codes. This method not only follows the principle of semantic propagation, but also converts large-scale on-line comparison into off-line matrix calculations. Although hash codes have already been the most concise feature representations, the construction and calculation of iteration matrix for PageRank still cause huge memory and time overhead. Therefore, the scale of iteration matrix needs to be further reduced.

(3) Efficient on-line assessment scheme to adapt to the diversity of requirements. The dataset is fixed while the requirements are flexible. If AD is placed in the dataset, then the time-consuming association calculation cannot be conducted off-line, which severely reduces the efficiency of on-line assessment. Therefore, the best assessment strategy should be independent of the association calculation, which can dynamically associate the content semantics of AD. At the same time, both the semantic representations of AD and the dataset must be generated from a unified model to ensure the reliability of the relevance measurement.

3 DHR-DQA DESIGN

Based on the analysis and challenges mentioned above, we design the DHR-DQA framework.

3.1 Framework Overview

As shown in Figure 3, DHR-DQA is a framework to assess the relevance of a dataset based on the given requirements. Our framework integrates Convolution Neural Network– (CNN) based image feature extraction technique, GCN-based hashing technique, hash-based graph structure, improved PageRank algorithm with restricted Hamming distance and image retrieval technique.

Considering that the dataset is fixed and the requirements are diverse, DHR-DQA first off-line learns and analyzes the dataset and then on-line completes the assessment for different requirements. The workflow of the framework is described as follows.

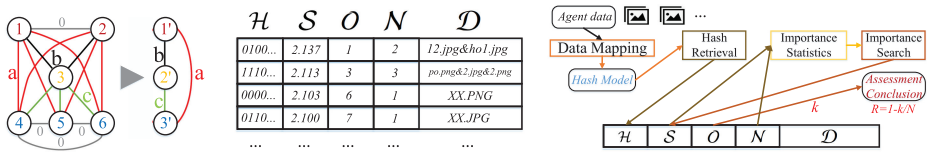
Off-line learning and relevance mining. The off-line part takes image dark data as input, which first generates the hash model in the data acquirement module, and then uses the hash model to map all data. Then in the relevance association and mining module, all data are organized into a graph structure where each node denotes a hash code and each edge denotes the Hamming distance between two nodes, and next, PageRank algorithm is adopted to calculate the importance score for each node. Finally, all data are sorted according to the importance scores to obtain the importance list. Both of the hash model and importance list (i.e., the output of the off-line part) are input to the on-line assessing part. It is worth noting that this part can also be regarded to lighten dark data, which not only annotates the unknown data by hash codes but also constructs the association between data. In addition, as long as the hash model can deal with other types of data, this model can lighten the corresponding type of dark data.

On-line assessing. The on-line assessing module takes the AD as input, which first maps AD to hash codes via the hash model, then uses these hash codes to retrieve data from the dataset, next counts the importance score for AD according to the retrieved data, and finally search the rank of this score in the importance list to obtain the final assessment conclusion. Specifically, assuming that h_i and h_j respectively represent the hash codes of two data, we use $\mathrm{HD}(h_i, h_j)$ to denote the Hamming distance between h_i and h_j, and τ to denote the Hamming radius. If $\mathrm{HD}(h_i, h_j) \leq \tau$, then the two images are matched. Assume that the importance score of AD is denoted as S(AD) and the ranked scores in importance list are listed as $\{S_1, S_2, \ldots, S_N\}$. If $\exists k \in Z$ that satisfies $S_k \leq \mathrm{S(AD)} < S_{k-1}$, then we can obtain the assessment conclusion that $\mathrm{R(AD)} = 1 - \frac{k}{N}$. When $\mathrm{R(AD)} \in [0, 1]$ becomes larger, it indicates that the degree of relevance between the requirement and the dataset is higher, so it is more worthwhile for the user to invest data mining resources on this dataset.

It is worth emphasizing that the design of off-line learning and on-line assessing separates the time-consuming learning and mining from real-time processing, which ensures the efficiency by just adopting model-based data mapping and hash-based retrieval. Meanwhile, for acquiring S(AD), we obtain the weighted importance scores of those retrieved similar data instead of real-time importance score calculation, thereby ensuring a highly efficient on-line assessing scheme that adapts to the diversity of requirements (mentioned in Section 2 (3)). In addition, we give a new hash algorithm to improve DSTH to address the problem of space overhead and precision with multi-label data for dark data learning (mentioned in Section 2 (1)). And then, we improve SHR to address the challenge of high computation overhead under semantic propagation (mentioned in Section 2 (2)). We now introduce above modules in detail.

3.2 Data Representation Acquirement

This module aims to obtain data representations from dark data. This representation based on content semantics requires the ability to not only understand the objects in an image but also reasonably allocate distances between different objects according to the data distribution of dataset. Dark data assessment work [23] adopts DSTH to get the representations, but it takes a large amount of storage space to construct the spectral graph. Besides, the vertices of the graph are generated by the empirical model (i.e., GoogLeNet on ImageNet), leading to poor performance on multi-label (e.g., locally similar) images. To address these deficiencies and ensure our model can perceive the semantic distribution of dark data, we introduce GCN [5], Cauchy loss [4] and AutoEncoder (AE) [44] to design GAH algorithm to complete the model learning, where (1) GCN enables better weight learning for multiple objects in an image and reduces storage space by getting rid of constructing graph; (2) Cauchy loss promotes the precision of hash codes within small Hamming



(a) change for graphing (b) importance list (c) assessing process

Fig. 4. (a) The change of graph constructing when merging the same hash codes into one node. (b) The structure of Importance list. (c) The calculation process of on-line assessing.

sphere, which makes AD match more similar data in given distance; and (3) AE effectively perceives the data distribution of the dark dataset. In the following process, we use the learned hash model to map all data into hash codes. Note that all images are first converted into 256×256 RGB and then input to the model. The detailed processing is introduced in Section 4.

3.3 Relevance Associating and Mining

This module aims to mine the correlation between data by constructing the association of the dataset. Similarly to our previous DQ assessment work [22], we first construct the graph with restricted Hamming distance, then adopt SHR [33] algorithm to calculate the importance score (used as a look-up table for measuring the relevance) for each node (image). However, the construction of graph and matrix used for importance score will cause huge storage overhead by the original method. In practice, we find there exists hash collision phenomenon and those same hash codes will get the same score. Therefore, we propose an improved construction scheme by using each non-repeating hash code as a node to construct the graph. As shown in Figure 4(a), node 1 and node 2 own the same hash code while node 4, 5, and 6 own the same hash code. We merge node 1 and 2 into a new node $1'$, while node 3 becomes a new node $2'$ and node 3, 4, and 5 are merged into a new node $3'$. Compared with the original graph, the structure of the new graph looks concise, which not only retains the edges information of a, b, and c, but also reduces the 6×6 matrix by half into a 3×3 matrix. On this new graph structure, we design a CPR algorithm according to the number of data that each node contains, which degrades the calculation scale as well as ensures the same results as the original scale. The detailed processing is introduced in Section 5.

3.4 Importance List

The purpose of importance list is to provide look-up table for on-line assessing. We rank all the hash codes in a descending order by their importance scores. As shown in Figure 4(b), each unit in the list contains five properties: $\mathcal{H}$ denotes hash code, $\mathcal{S}$ denotes importance score, $\mathcal{O}$ denotes importance order, $\mathcal{N}$ denotes number of the data presented by $\mathcal{H}$ and $\mathcal{D}$ denotes filenames of those data. Each property forms a vertical column to serve the hash retrieval and importance score search.

3.5 Assessment

The on-line assessing process is shown in Figure 4(c). AD is represented by single or multiple images and then each image is mapped into a hash code by learned hash model. Next, the system pointer first moves on the column of $\mathcal{H}$ to search hash codes in the given Hamming radius and count the importance score of AD according to the matched set of $\mathcal{S}$ and $\mathcal{N}$, then climbs on the column of $\mathcal{S}$ to find the importance ranking of this score. Finally, we quantify the assessment conclusion. Mathematically, we let q denote AD consisting of n images and I_i denote the ith image where $i \in [1, n]$, m_i denotes the number of matched images for the ith image. Meanwhile, we let

$S_j(I_i)$ denote the score of the jth image where $j \in [1, m_i]$. Therefore, the score of q is defined as follows:

$$S(q) = \sum_{i=1}^{n} \frac{1}{m_i} \sum_{j=1}^{m_i} W_i S_j(I_i)$$
$$s.t. \sum_{i=1}^{n} W_i = 1, \tag{1}$$

where $W_i \in [0, 1]$ represents the importance weight of I_i. Compared with the ranked scores in importance list $\{S_1, S_2, \ldots, S_N\}$, if $\exists k \in Z$ that satisfies $S_k \leq \mathrm{S(AD)} < S_{k-1}$, we can obtain the assessment conclusion that $\mathrm{R(AD)} = 1 - \frac{k}{N}$.

4 GCN-AUTOENCODER HASHING

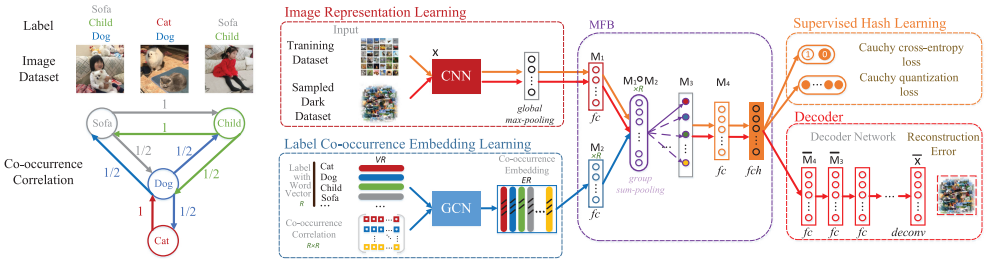
In this section, we describe the design of GAH, which combines the co-occurrence probability [43], GCN [37], Multi-modal Factorized Bilinear (MFB) [17], Cauchy loss function [4] and AE [44]. The overall architecture of GAH is shown in Figure 4, which is divided into 5 modules: Image Representation Learning, Label Co-occurrence Embedding Learning, MFB, Supervised Hash Learning and Decoder. Specifically, Image Representation Learning completes the feature extraction of images, Label Co-occurrence Embedding Learning maps label word vectors to the co-occurrence embeddings that contain co-occurrence relationship, MFB fuses the image representations and co-occurrence embeddings into hash vectors, Supervised Hash Learning trains and updates the hash vectors with the input of training dataset, and Decoder trains and feedbacks the hash vectors with the input of sampled dark dataset. Note that the Decoder is an unsupervised learning.

By and large, GAH is essentially a semi-supervised hash algorithm. Although we need a supervised process to fill parameters of ResNet and GCN, the last parameters are determined by the learning of AutoEncoder. Note that GAH does need labels. However, this requirement just occurs at supervised learning stage on the public training dataset. At the AutoEncoder stage, GAH just applies label concepts, i.e., inputs of GCN to learning the dark dataset. Of course, these inputs are noisy but empiric parameters are fit for these concepts. As an upstream operation, these empiric parameters help complete downstream task, i.e., hash mapping with representative features.

A complete training process (TP) consists of stage TP-*A* and TP-*B*, where the orange arrow flow (Training Dataset as input) and blue arrow flow constitute stage TP-*A* while the red arrow flow (Sampled Dark Dataset as input) and blue arrow flow constitute stage TP-*B*. During a TP, stage TP-*A* aims to get better parameters of GCN and CNN according to the labels while stage TP-*B* focuses on fine-tuning the parameters of CNN according to the dark data distribution and trained co-occurrence relationship. The network will be iteratively updated after multiple TPs until the number of convergence epoch within a TP is lower than the preset threshold.

4.1 Co-occurrence Possibility

As the input of GCN, co-occurrence correlation is constructed by co-occurrence probability that means the probability that one object occurs in the conditional of another object appearing. As shown in Figure 5(a), *dog* appears twice, *cat* appears once, while *dog* and *cat* co-occur once. The blue arrow where *dog* points to *cat* indicates the probability (i.e., $\mathrm{P}(dog|cat) = 1$) that *dog* occurs in the conditional of *cat* appearing while the red arrow where *cat* points to *dog* indicates the probability (i.e., $\mathrm{P}(cat|dog) = 1/2$) that *cat* occurs in the conditional of *dog* appearing. Co-occurrence probability reflects the correlation between objects in the real world, and GCN can express this correlation via the attention mechanism, which helps guide the multi-label image representation learning. The



(a) Co-occurrence Correlation (b) Architecture of GAH

Fig. 5. (a) Example of constructing the co-occurrence correlation. (b) Architecture of GCN-AutoEncoder Hashing that includes Image Representation Learning, Label Co-occurrence Embedding Learning, MFB, Supervised Hash Learning, and Decoder.

introduction of co-occurrence probability cannot only meet the feature extraction's requirement under semantic propagation but also promote the recognition precision on multi-label images.

4.2 Image Representation Learning

Following ML-GCN [5], we select ResNet-101 as the backbone of CNN model. For an image x, we acquire a 2048×14×14-dimensional feature vector from the "$conv5\_x$" layer. And then we get the image-level feature $\tilde{x}$ by global max-pooling $\mathcal{F}_{gmp}$:

$$\tilde{x} = \mathcal{F}_{gmp}(\mathcal{F}_{cnn}(x;\theta)),$$

where θ denotes the CNN parameters and $D(\tilde{x}) = 2048$.

Note that, as input for the training dataset, we assume that there are N samples $\{x_i|i = 1, 2, \ldots, N\}$ in the training set, where $L(x_i)$ denotes the label set of the ith sample and s_{ij} denotes the similarity between x_i and x_j ($s_{ij} = 1$ if x_i is similar to x_j while $s_{ij} = 0$ if they are dissimilar). Specifically, for the multi-label images, we stipulate $s_{ij} = 1$ if $L(x_i) \cap L(x_j) \neq \emptyset$; otherwise, $s_{ij} = 0$. When training model in the TP-A, we input pairs of $\{(x_i, x_j, s_{ij})\}$. In addition, as the input for the sampled dark dataset, we assume that there are N^* samples $\{x_i^*|i = 1, 2, \ldots, N^*\}$ in the training set. When training model in the TP-B, we will input x_i^*.

4.3 Label Co-occurrence Embedding Learning

We assume that there are R objects $\{r_g|g = 1, 2, \ldots, R\}$ in the whole label set and $ER = \{E(r_g)|g = 1, 2, \ldots, R\}$, where $E(r_g)$ denotes the co-occurrence embedding of the gth object. We use the GloVe [26] model to get word vector $V(r)$, where $V^c \in \mathbb{R}^{R \times D(V(r))}$ represents the input in cth layer and $D(V(r))$ represents the dimension of $V(r)$. The input of relationship is the correlation matrix $A \in \mathbb{R}^{R \times R}$, and the updated node features are denoted as $V^{c+1} \in \mathbb{R}^{R \times D(V(r))'}$. Each GCN layer propagation function is described as:

$$V^{c+1} = \mathcal{F}_{gcn}(\hat{A}V^cW^c),$$

where $\hat{A} = \tilde{D}^{-\frac{1}{2}}(A + I_R)\tilde{D}^{-\frac{1}{2}}$ with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$ and $\tilde{A} = A + I_R$. At last, we send ER to MFB, where $D(E(r)) \times R$ denotes the dimension of ER.

Referring to ML-GCN, we adopt the data-driven method to construct matrix A, which is the key of co-occurrence learning. Specifically, to determine each element of matrix A, we have to first collect the occurrence times as well as co-occurrence times of each object according to the label set. Let T_i, T_j respectively denote the occurrence times of r_i, r_j in the label set and T_{ij} (which equals T_{ji}) denote the co-occurrence times of these two objects. Then, we model the label correlation

dependency in the form of conditional probability, i.e.,

$$P_{ij} = P(r_i|r_j) = \frac{T_{ij}}{T_j},$$

which denotes the probability of occurrence of r_i when r_j appears. Based on this, $A_{ij} = P_{ji}$, where A_{ij} denotes the ith row and jth column element of A.

4.4 MFB

Given two feature vectors in different modalities, i.e., image representation $\tilde{x} \in \mathbb{R}^{D(\tilde{x})}$ and co-occurrence embedding $E(r) \in \mathbb{R}^{D(E(r))}$, the multi-modal bilinear model with two low-rank matrices for the ith object is defined as follows:

$$\begin{aligned} z_i &= \tilde{x}^T U_i V_i^T E(r) = \sum_{d=1}^{k} \tilde{x}^T u_d v_d^T E(r) \\ &= \mathbb{1}^T \left(U_i^T \tilde{x} \circ V_i^T E(r) \right), \end{aligned} \tag{2}$$

where k is the latent dimensionality of the factorized matrices $U = [u_1, \ldots, u_k] \in \mathbb{R}^{D(\tilde{x}) \times k}$ and $V = [v_1, \ldots, v_k] \in \mathbb{R}^{D(E(r)) \times k}$, $\circ$ is the Hadmard product, i.e., the element-wise multiplication of two vectors, $\mathbb{1} \in \mathbb{R}^k$ is an all-one vector. In practice, as shown in Figure 5(b), we adopt two parallel k-dimensional fully connected (fc) layers (red and blue fc layers shown in MFB framework) to complete this transform respectively, where $M_1 = U_i^T \tilde{x}$ and $M_2 = V_i^T E(r)$.

To obtain the output $Z = [z_1, \ldots, z_R]$, the weights to be learned are two three-order tensors $\tilde{U} = [U_1, \ldots, U_R] \in \mathbb{R}^{D(\tilde{x}) \times k \times R}$ and $\tilde{V} = [V_1, \ldots, V_R] \in \mathbb{R}^{D(E(r)) \times k \times R}$. Without loss of generality, we can reformulate $\tilde{U}$ and $\tilde{V}$ as two-dimensional matrices $\tilde{U} \in \mathbb{R}^{D(\tilde{x}) \times (k \times R)}$ and $\tilde{V} \in \mathbb{R}^{D(E(r)) \times (k \times R)}$, respectively. At last, we obtain the result of group sum-pooling as follows:

$$Z = \mathcal{F}_{sum-pooling}(\tilde{U}^T \tilde{x} \circ \tilde{V}^T E(r), k),$$

where the function $\mathcal{F}_{sum-pooling}(*, k)$ means using a one-dimensional non-overlapped window with the size k to perform sum-pooling over $*$. Different from Reference [43], we improve this work where all the R identical vectors in $\tilde{U}$ are generated from $\tilde{x}$ while the vectors in $\tilde{V}$ correspond to the elements of ER transformed by fc layer.

At last, GAH transforms Z into K-dimensional continuous code $\tilde{Z} \in \mathbb{R}^K$ for each image x, and then transforms $\tilde{Z}$ into K-dimensional hash code by $h = sgn(tanh(\tilde{Z})) \in \{-1, 1\}^K$ in the fch layer.

4.5 Supervised Hash Learning

According to DCH [4], the hash function based on Cauchy distribution can achieve better results when Hamming radius ≤ 2 in terms of Mean Average Precision, Precision and Recall. Therefore, we adopt Cauchy distribution-based function as our hash function to promote precision in small Hamming radius by jointly preserving similarity of pairwise images and controlling the quantization error.

Assume that $h_i, h_j \in \{-1, 1\}^K$ respectively denote the output of x_i and x_j after the fch layer while $\{(x_i, x_j, s_{ij}) : s_{ij} \in \mathcal{S}^*\}$ denotes the input. Based on the Cauchy distribution, we design the probability function:

$$\sigma(\delta(h_i, h_j)) = \frac{\gamma}{\gamma + \delta(h_i, h_j)}, \tag{3}$$

where $\sigma(*)$ is a well-defined probability function, $\delta(h_i, h_j)$ denotes the Hamming distance between h_i, and h_j, γ is the scale parameter of the Cauchy distribution. Note that the smaller γ, the higher

aggregation degree within short Hamming radius. We repeatedly conduct extensive experiments and choose $\gamma = 0.15$.

To learn high-quality hash codes and control the quantization error $||h_i - sgn(h_i)||$ resulting from continuous relaxation, we combine the parameter γ (Equation (3)) and Cauchy distribution to design prior for each hash code as follows:

$$\mathcal{P}_{h_i} = \frac{\gamma}{\gamma + \delta(|h_i|, \mathbb{1})}, \tag{4}$$

where $\mathbb{1} \in \mathbb{R}^K$. To cooperate with the continuous relaxation based on Cauchy distribution, we adopt $\delta(h_i, h_j) = \frac{K}{2}(1 - cos(h_i, h_j))$ to normalize Euclidean distance as the approximation of Hamming distance to ease the optimization. The normalized Euclidean distance compares each pair of continuous codes on a unit sphere, while the Hamming distance compares each pair of hash codes on a unit hyper-cube.

According to the deduction of Bayesian learning in DCH and Equation (3), the Cauchy cross-entropy loss $\mathcal{L}_{cce}$ is derived as

$$\mathcal{L}_{cce} = \sum_{s_{ij}} \omega_{ij} \left(s_{ij} log \frac{\delta(h_i, h_j)}{\gamma} + log(1 + \frac{\gamma}{\delta(h_i, h_j)}) \right), \tag{5}$$

where

$$\omega_{ij} = \begin{cases} |\mathcal{S}^*|/|\mathcal{S}^*_s|, & s_{ij} = 1 \\ |\mathcal{S}^*|/|\mathcal{S}^*_d|, & s_{ij} = 0 \end{cases},$$

where $\mathcal{S}^*_s = \{s_{ij} \in \mathcal{S}^* : s_{ij} = 1\}$ is the set of similar pairs and $\mathcal{S}^*_d = \{s_{ij} \in \mathcal{S}^* : s_{ij} = 0\}$ is the set of dissimilar pairs.

In addition, according to Equation (4), the Cauchy quantization loss $\mathcal{L}_{cq}$ is

$$\mathcal{L}_{cq} = \sum_{i=1}^{N} log \left(1 + \frac{\delta(|h_i, \mathbb{1}|)}{\gamma} \right). \tag{6}$$

Based on the above equations, the completed hash function is

$$\mathcal{L} = \lambda \mathcal{L}_{cce} + (1 - \lambda)\mathcal{L}_{cq},$$

where λ is a hyper-parameter to tradeoff two loss functions. Note that our sign function is designed as

$$sgn(h_i) = \begin{cases} 1, & h_i > 0, \\ -1, & \text{otherwise.} \end{cases}$$

where h_i is a element in vector $h \in \mathbb{R}^K$. This binarization will not severely affect the retrieval result, because it have been adjusted by the quantization loss.

4.6 Decoder

We set up 4 fc layers and a set of deconvolution modules in Decoder frame of Figure 5(b), where $\overline{M_4}$ and $\overline{M_3}$ respectively correspond to the dimensions of M_4 and M_3 in the MFB frame, and the dimensions of other two fc layers correspond to the dimensions of M_1 and global max-pooling layer. The deconvolution layers follow the result of global max-pooling layer and generate $\overline{x}$, which is the same dimension of the input of Sampled Dark Data, i.e., 448×448×3. Based on this, we construct

the reconstruction loss function $\mathcal{L}_f$:

$$\begin{aligned} \mathcal{L}_f &= \sum_{i=1}^{N} \kappa_1 \|M_3^{(i)} - \overline{M_3^{(i)}}\|_2^2 + \kappa_2 \|M_4^{(i)} - \overline{M_4^{(i)}}\|_2^2 + \kappa_3 \|x^{(i)} - \overline{x^{(i)}}\|_2^2, \\ s.t. &\sum \kappa_t = 1, \end{aligned} \tag{7}$$

where the set of κ_t are hyper-parameters. In addition, to prevent over-fitting, we design the regularizer term $\mathcal{L}_r$ as follows:

$$\mathcal{L}_r = \frac{1}{4} \sum_{i=1}^{N} \left\| W_f^{(i)} \right\|_F^2, \tag{8}$$

where W_f are the weights of *fc* layers in Decoder. Base on above design, the terminal loss function $\mathcal{L}$ is:

$$\mathcal{L} = \alpha \mathcal{L}_r + \beta \mathcal{L}_f, \tag{9}$$

where α and β are hyper-parameters that satisfy $\alpha + \beta = 1$. If the difference between two consecutive iterations is less than a preset threshold, then we stipulate $\mathcal{L}$ has converged.

5 CLUSTER PAGERANK

In this section, we first describe the principle of constructing the hash-based graph, then review our previous hash-based rank algorithm [33], and finally propose our designed CPR algorithm.

5.1 Graph Construction

Given a graph G consisting of n nodes, each node is denoted as a l-bits hash code. N_* denotes the $*$th node of Graph G and $H(N_*)$ denote the hash code of N_*. We define XOR operation as $\oplus$ and threshold $\Omega \in [1, l] \cap Z^+$. We stipulate that two nodes are connected only if the Hamming distance between them does not exceed threshold Ω. The reason why all nodes are not fully connected with each other is that there no longer exists semantic similarity between two nodes once their Hamming distance exceeds half of the length of hash code [4]. Therefore, the Hamming distance weight on undirected edge between N_i and N_j is defined as:

$$d_{ij} = \begin{cases} H(N_i) \oplus H(N_j) & i \neq j, H(N_i) \oplus H(N_j) \leq \Omega, \\ NULL & \text{otherwise.} \end{cases} \tag{10}$$

According to the characteristic of similarity hash, the less d_{ij} is, the more similar N_i and N_j are. Also, we advisedly cut off those edges where Hamming distance exceeds given Ω. Generally, Ω is set to be a integer less than $\frac{l}{2}$. Each node is connected with other qualified nodes. As a result, the hash-based graph is established.

5.2 Hash-based Rank Algorithm

Our goal is to calculate the importance score of each node. For $\forall\, i \in [1, n]$, T_i is defined as the set including orders of all nodes connected with N_i, where $T_i \subset [1, n]$.

As defined above, l denotes the length of hash code and d_{ij} denotes weight of the edge between N_i and N_j. We denote $R(N_*)$ as the importance score of N_*. Referring to PageRank, we also intend to calculate the ultimate $R(N_*)$ by means of iteration. Draw impact factor $I(N_{ij})$ for N_j to N_i, which measures how N_j contributes to N_i, where $I(N_{ij})$ is defined as:

$$I(N_{ij}) = \begin{cases} \dfrac{l - d_{ij}}{\sum_{t \in T_j} l - d_{tj}} R(N_j) & \exists d_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

Theoretically, we design Equation (11) according to two principals. First, the less d_{ij} is, the greater influence N_j contributes to N_i is. Meanwhile, the longer hash code (l) is, the more compact the similarity presented by d_{ij} is. Second, PageRank considers all (unweighted) edges as the same, but we extend it to be applied to different weights on edges. Specially, when all weights on edges are the same, our hash-based rank algorithm will turn into undirected PageRank. Consequently, $R(N_i)$ should be equal to the sum of the impact factors of all nodes connected to N_i, where N_i is expressed as $R(N_i) = \sum_{j=1, j\neq i}^{n} I(N_{ij})$.

Let f_{ij} represent the coefficient of $R(N_j)$ in $I(N_{ij})$, where

$$f_{ij} = \begin{cases} \dfrac{l - d_{ij}}{\sum_{t \in T_j} l - d_{tj}} & \exists d_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{12}$$

We define coefficient matrix D as $\begin{bmatrix} 0 & f_{12} & \cdots & f_{1n} \\ f_{21} & 0 & \cdots & f_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ f_{n1} & f_{n2} & \cdots & 0 \end{bmatrix}$ and calculate each column sum of D according to Equation (12), we take the jth column as

$$\begin{aligned} & f_{1j} + f_{2j} + \ldots + f_{nj} \\ =& \frac{l - d_{1j}}{\sum_{t \in T_j} l - d_{tj}} + \frac{l - d_{2j}}{\sum_{t \in T_j} l - d_{tj}} + \ldots + \frac{l - d_{nj}}{\sum_{t \in T_j} l - d_{tj}} \\ =& \sum_{t \in T_j} \frac{l - d_{tj}}{\sum_{t \in T_j} l - d_{tj}} = 1. \end{aligned} \tag{13}$$

Usually, the initial value is set as $R^0 = [R^0(N_1)\, R^0(N_2) \ldots R^0(N_n)]^T = [1\, 1 \ldots 1]^T$. We draw iteration formula as

$$R^{k+1} = DR^k, \tag{14}$$

where $R^k = [R^k(N_1)\, R^k(N_2) \ldots R^k(N_n)\,]^T$, and k is the number of iteration rounds. We define the termination condition as $R^{k+1}(N_m) - R^k(N_m) \leq \varepsilon$, where $m \in [1, n]$. Meanwhile, ε is set to a small constant (say 0.0001).

5.3 CPR Algorithm

Based on our previous work, in this part, we propose the CPR algorithm, which merges the same hash codes into one node.

5.3.1 Problem Redefinition. For convenience, we redefine this problem. Given a graph consisting of n nodes, there exist num_1 nodes N_1, num_2 nodes N_2, ..., num_m nodes N_m, where $num_1 + num_2 + \ldots + num_m = n$. In other words, n nodes form m clusters, and each node of the same cluster has the same hash code. Specially, for $\forall t \in [1, m]$, we denote the num_t node N_t as $N_{t1}, N_{t2}, \ldots, N_{t(num_t)}$. If we adopt our previous hash rank algorithm (Equation (12)) to calculate

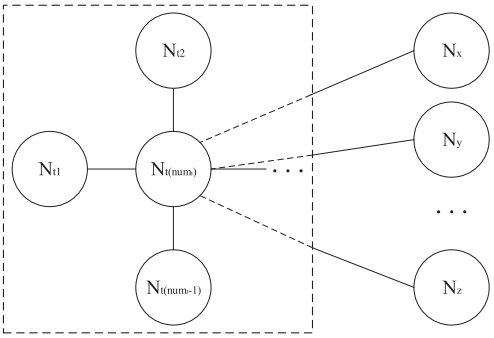


Fig. 6. Example graph.

scores for these n nodes, then the iteration formula is

$$\begin{bmatrix} R^{c+1}(N_{11}) \\ \cdots \\ R^{c+1}(N_{1(num_1)}) \\ R^{c+1}(N_{21}) \\ \cdots \\ R^{c+1}(N_{2(num_2)}) \\ \cdots \\ R^{c+1}(N_{m1}) \\ \cdots \\ R^{c+1}(N_{m(num_m)}) \end{bmatrix} = \begin{bmatrix} 0 & \cdots & f_{11} & f_{12} & \cdots & f_{12} & \cdots & f_{1m} & \cdots & f_{1m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\ f_{11} & \cdots & 0 & f_{12} & \cdots & f_{12} & \cdots & f_{1m} & \cdots & f_{1m} \\ f_{21} & \cdots & f_{21} & 0 & \cdots & f_{22} & \cdots & f_{2m} & \cdots & f_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\ f_{21} & \cdots & f_{21} & f_{22} & \cdots & 0 & \cdots & f_{2m} & \cdots & f_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\ f_{m1} & \cdots & f_{m1} & f_{m2} & \cdots & f_{m2} & \cdots & 0 & \cdots & f_{mm} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdot \\ f_{m1} & \cdots & f_{m1} & f_{m2} & \cdots & f_{m2} & \cdots & f_{mm} & \cdots & 0 \end{bmatrix} \begin{bmatrix} R^{c}(N_{11}) \\ \cdots \\ R^{c}(N_{1(num_1)}) \\ R^{c}(N_{21}) \\ \cdots \\ R^{c}(N_{2(num_2)}) \\ \cdots \\ R^{c}(N_{m1}) \\ \cdots \\ R^{c}(N_{m(num_m)}) \end{bmatrix}, \tag{15}$$

where f_{ij} (defined in Equation (12)) represents the impact factor coefficient that N_j contributes to N_i. When above Equation (15) converges, the final n scores satisfy that for $\forall t \in [1, m]$, $R(N_{t1}) = R(N_{t2}) = \ldots = R(N_{t(num_t)})$.

5.3.2 Algorithm. As shown in Figure 6, we organize the same nodes into groups. For $\forall t \in [1, m]$, take node N_t for example, inside the group, $N_{t(num_t)}$ receives contributions from other $num_t - 1$ nodes (N_t). Outsides the group, $N_{t(num_t)}$ also receives contributions from nodes that belong to $\{N_x, N_y, \ldots, N_z\} \backslash \{N_t\}$.

As we know, $\forall t \in [1, m]$, $R(N_{t1}) = R(N_{t2}) = \ldots = R(N_{t(num_t)})$. For convenience, we use R_t to denote all $\{R_{t1}, \ldots, R_{t(num_t)}\}$ during the iteration process. According to the cluster relationship, we redesign the iteration matrix, and introduce our cluster rank algorithm (Equation (16) and Equation (17)) consisting of two steps.

Step (1): We first use new iteration matrix to calculate scores for all nodes $\{N_1, \ldots, N_m\}$ (m clusters, n nodes). The iteration formula is designed as follows:

$$\begin{bmatrix} R^{c+1}(N_1) \\ R^{c+1}(N_2) \\ \cdots \\ R^{c+1}(N_m) \end{bmatrix} = \begin{bmatrix} (num_1 - 1)f_{11} & num_1 f_{12} & \cdots & num_1 f_{1m} \\ num_2 f_{21} & (num_2 - 1)f_{22} & \cdots & num_2 f_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ num_m f_{m1} & num_m f_{m2} & \cdots & (num_m - 1)f_{mm} \end{bmatrix} \begin{bmatrix} R^{c}(N_1) \\ R^{c}(N_2) \\ \cdots \\ R^{c}(N_m) \end{bmatrix}. \tag{16}$$

Step (2): To ensure the calculated scores are the same as that of hash-based rank algorithm (Equation (15)). We must adjust above scores using the following assignment statement after step

(1):

$$R(N_t)\,(i.e., R(N_{t1}), \ldots, R(N_{t(num_t)})) = \frac{R(N_t)}{num_t}\frac{n}{m} \quad \forall t \in [1, m]. \tag{17}$$

We mark cluster coefficient matrix as

$$\hat{D} = \begin{bmatrix} (num_1 - 1)f_{11} & num_1 f_{12} & \cdots & num_1 f_{1m} \\ num_2 f_{21} & (num_2 - 1)f_{22} & \cdots & num_2 f_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ num_m f_{m1} & num_m f_{m2} & \cdots & (num_m - 1)f_{mm} \end{bmatrix}. \tag{18}$$

We must emphasize the the principle of our design. In step (1), the diagonal elements of matrix $\hat{D}$ are no longer all zeros, because any one node N_t of the cluster receives contributions from other $num_t - 1$ identical nodes. Besides, other elements represent the interrelationships between different clusters. In step (2), we expect to ensure the column sum of final result identical between previous hash-based algorithm and our designed cluster algorithm.

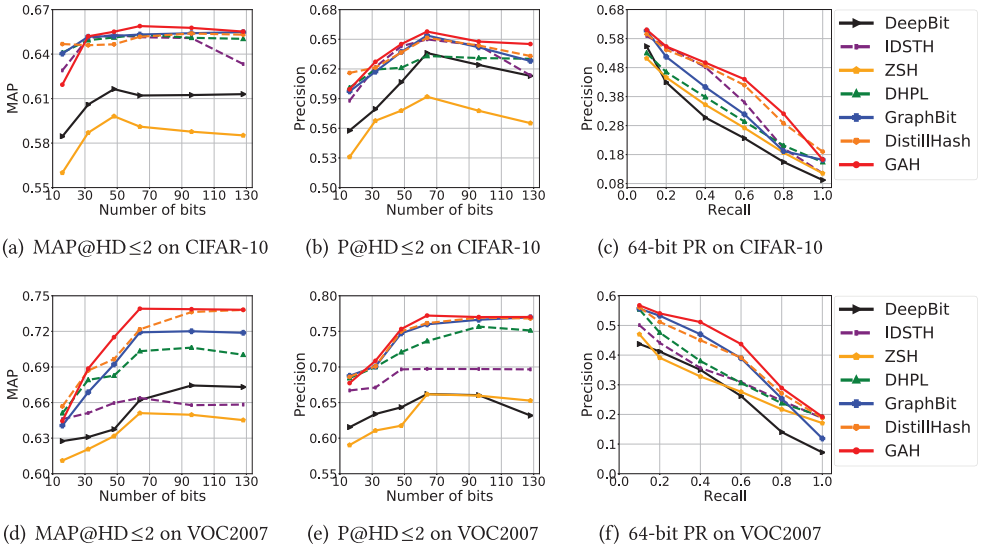
6 EVALUATION

In this section, we evaluate our framework and conduct extensive experiments as follows.

(1) For selecting retrieval parameters in the on-line assessing part, we test the performance of GAH with different hash length and retrieval radius on single-label and multi-label datasets.
(2) To illustrate the superiority of usage of hash codes in DHR-DQA, we manifest the graphing efficiency compared with other metrics.
(3) We test the acquirement efficiency of valuable data and resource consumption of DHR-DQA on designed datasets that are sampled from a public single-label dataset.
(4) We test the acquirement efficiency of valuable data and resource consumption of DHR-DQA on designed datasets that are sampled from a public multi-label dataset.
(5) We give the assessment results on a real-world dataset.

In the experiment, we adopt MS-COCO as Training Dataset. Meanwhile, we implement the experiment (1) on CIFAR-10 and VOC2007, the experiment (3) on CIFAR-10, the experiment (4) on VOC2007, and the experiment (5) on Tencent1M. We summarize these datasets as follows:
MS-COCO [19] is a popular multiple object dataset for image recognition, segmentation and captioning, which contains 118,287 training images, 40,504 validation images and 40,775 test images, where each image is labeled by some of the 80 semantic concepts.
CIFAR-10 [13] consists of 60,000 single object images that are labeled with 10 classes. There are 5,000 training images and 1,000 test images in each class.
VOC2007 [7] consists of 9,963 multi-label images and 20 object classes. On average, each image is annotated with 1.5 labels. Note that we think 0 is 1 in label dataset.
Tencent1M [22] is a subset of QQ album data from 2017 to 2018 of Tencent. The size of the data is around 5 TB consisting of 1M images.

Note that, except for MS-COCO datasets, other datasets are used as dark dataset without labels. We reorganize multiple sub-datasets as Sampled Dark Dataset according to the selected target so that a certain type of data occupies a larger proportion in each dataset. Our framework sorts these sub-datasets by calculating the R(AD) and take the total number of targets from the top sub-datasets as the evaluation results. We compare two baseline methods: One randomly selects (RS) the sub-datasets and the other sorts these sub-datasets referring to SDQA.



(a) MAP@HD≤2 on CIFAR-10 (b) P@HD≤2 on CIFAR-10 (c) 64-bit PR on CIFAR-10

(d) MAP@HD≤2 on VOC2007 (e) P@HD≤2 on VOC2007 (f) 64-bit PR on VOC2007

Fig. 7. MAP@HD≤2 and P@HD≤2 with different code lengths and 64-bit PR curves on CIFAR-10 and VOC2007.

Our framework is executed using PyTorch and Scikit-Learn library, which includes hash model learning, graph structuring and retrieval. Our experiments are run on two 10-core Intel Xeon E5-2640 machines with 128 GB of DDR4 memory.

6.1 Retrieval Parameters

In terms of precision, we compare GAH with IDSTH [22], the state-of-the-art unsupervised hashing (i.e., DeepBit [18], GraphBit [6], DistillHash [39]) methods and zero-shot hashing (i.e., ZSH [41], DHPL [47]) methods on CIFAR-10 and VOC2007. At different hash length, we report three standard evaluation metrics to measure the quality of hash codes within Hamming radius 2: Mean Average Precision within Hamming Radius 2 (MAP@HD≤2), Precision curves within Hamming Radius 2 (P@HD≤2), and Precision-Recall (PR) curves under the hash length that achieves the highest precision. We will choose 10 different images from each category of the test set for retrieval and calculate the average value of the retrieval results. According this experiment result, we determine the optimal retrieval parameters for subsequent experiments.

Figure 7(a), (b), and (c) shows the performance of GAH on CIFAR-10. From Figure 7(a) and (b), GAH achieves the better results than others in all cases except for 16-bit, and respectively obtains the highest MAP@HD≤2 (0.6589) and P@HD≤2 (0.6578) at 64-bit. From Figure 7(c), at 64-bit, except when the recall ration is equal to 100%, GAH outperforms others in all case, which verifies GAH can obtain enough data under the premise of ensuring high precision.

Figure 7(d), (e), and (f) shows the performance of GAH on VOC2007. We stipulate a retrieved image is successful as long as one of the labels is matched. From Figure 7(d) and (e), GAH achieves the better results than others in all cases except for 16-bit and respectively obtains the highest MAP@HD≤2 (0.7391) and P@HD≤2 (0.7722) at 64-bit. From Figure 7(f), at 64-bit, GAH outperforms others in all case.

To further determine the retrieval radius, we list the retrieval precision and average number of matched images with different radii under 64-bit in Table 1. With the increase of HD, the precision will decrease while the number of matched images will increase. However, there is a risk that no

Table 1. For 64-bit Hash Code Generated by GAH, the Precision and Return Number at Different Hamming Distance Radii (HD) on CIFAR-10 and VOC2007

	CIFAR-10@HD						VOC2007@HD					
	=0	≤1	≤2	≤3	≤4	≤5	=0	≤1	≤2	≤3	≤4	≤5
Precision	0.6611	0.6593	0.6578	0.6318	0.6162	0.5507	0.7801	0.7734	0.7722	0.7196	0.6601	0.6147
Return Number	0.57	0.89	2.38	5.77	11.74	27.13	0.68	1.01	1.69	3.31	5.22	8.91

Fig. 8. Top-10 retrieval results on CIFAR-10 with queries of *plane* and *deer*. The red frame denotes the mismatched image.

one image can be retrieved when HD = 0 or HD = 1. When HD ≤ 2, the precision will not decrease significantly compared with setting HD to 0 or 1, but there is a higher chance that at least one image can be successfully retrieved. Based on this observation, in the subsequent experiments, we choose hash length to 64 and use the retrieval radius no more than 2.

For the sake of fairness, we compare GAH's retrieval results with supervised hash methods such as WeGAH [34] (state-of-the-art hash method with GAN), DCH [4] (state-of-the-art hash method with Cauchy loss) and OLAH [11] (state-of-the-art multi-label image hash method). In the implement, we also use MS-COCO to train the supervised hash methods and then test hash models on test set of CIFAR-10 and dataset of VOC2007. The top-10 retrieval results are shown in Figure 8 and Figure 9.

As shown in Figure 8, we first choose *plane* as the query data that have been trained on MS-COCO. According to the top-10 retrieval images, WeGAN acquires the optimal results without any mismatched images. while GAH produces one mismatched image that is same to DCH. Then, we try to query *deer* to verify performance for the unseen sample where *deer* has not been trained on

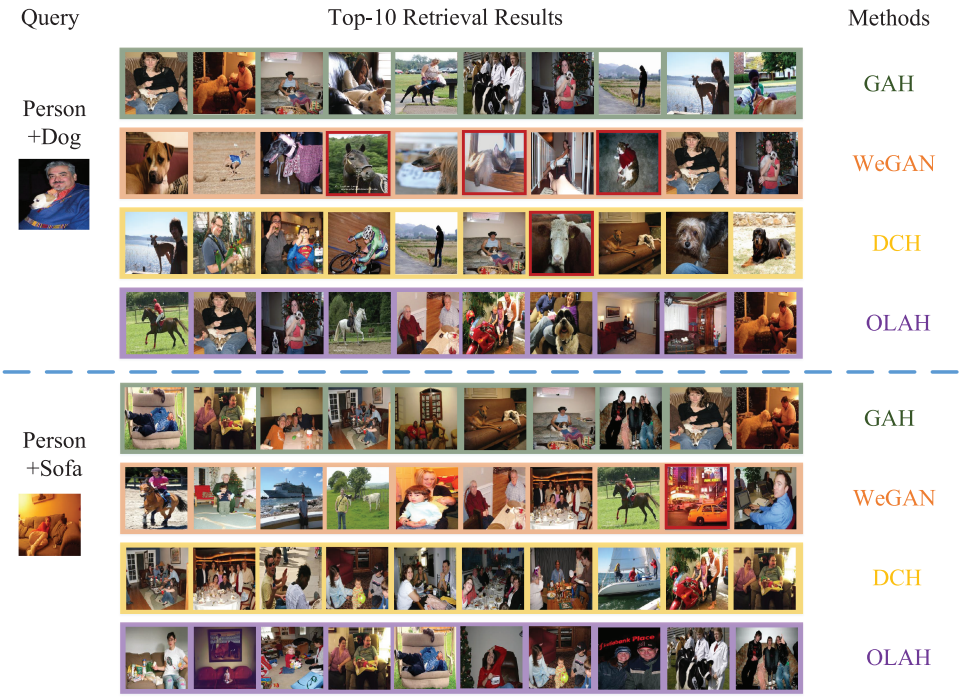


Fig. 9. Top-10 retrieval results on VOC2007 with queries of *person+dog* and *person+sofa*. The red frame denotes the mismatched image.

MS-COCO. We find GAH is as accurate as WeGAN (with four mismatched images), and outperform DCH. Therefore, we believe the performance of GAH has been promoted than DCH in retrieving unseen samples resulting from integrating AE into our framework. However, compared with the state-of-the-art hash method, GAH has no arresting advantage on single-label images, because the co-occurrence relationships cannot be sparked. Therefore, we further test top-10 retrieval results on VOC2007.

As shown in Figure 9, we first choose *person+dog* as the query data, both of which have been trained on MS-COCO. It is easy to see that GAH gets 10 related images. Although OLAH also returns 10 similar images, our retrieval results contain more images including both the dog and the person at the same time. Then, we query *person+sofa* where the sofa is the unseen sample. Obviously, GAH obtains the best results and returns more images containing the sofa. Therefore, we believe the co-occurrence correlations promote the precision of hash codes, because the objects are clustered in different degrees depending on the semantic related degree between them. Moreover, the correlations also affect the unseen samples under AE by pixel comparison. This is critical for the practicality in open scenarios.

6.2 Graphing Efficiency

In terms of graphing efficiency, we compare hash code-based Hamming distances with floating vector-based Cosine and Euclidean distance. In the implementation, we manifest the time cost of graphing with 48-bit hash codes and floating vectors. We randomly sample data from the overall data to form different data scales. Under each scale, we construct the graph for 30 times, and count the graphing time cost. As shown in Table 2, with the same scale of nodes, the graphing time

Table 2. Time Cost of Graphing with Different Scale of Nodes Using Hamming, Cosine, and Euclidean Distance (Unit: s)

Metrics	Scale of Nodes				
	10K	20K	50K	80K	100K
Euclidean	9527±612	81091±1291	734316±2002	997239± 2669	1244818±3178
Cosine	9607±439	92991±711	757933±1101	1093825±1471	1268299±1790
Hamming	688±43	1134±51	9317±127	13116±301	15347±355

Table 3. Reorganization Sub-datasets on CIFAR-10 Where [] Denotes the Category and () Denotes the Proportion of the Selected Category

Sub-dataset	Reconstruction Scheme	Amount				G&IC	
		dog	cat	airplane	animal	NN	CT(s)
CDS1	[airplane] (10%) + [automobile] (10%)+ [bird] (10%)+ [cat] (10%)+[deer] (10%)+[dog] (10%)+[frog] (10%)+[horse] (10%)+[ship] (10%)+[truck] (10%)	600	600	600	3600	**5531** 6000	**89.73** 93.67
CDS2	[airplane] (20%)+[automobile] (20%)+[ship] (20%)+[truck] (20%)+[dog] (20%)	1200	0	1200	1200	**5337** 6000	**86.01** 93.63
CDS3	[bird] (20%)+[cat] (20%)+[deer] (20%)+[dog] (20%)+[ship] (20%)	1200	1200	0	4800	**5333** 6000	**86.00** 93.65
CDS4	[dog] (45%)+[cat] (45%)+[airplane] (10%)	2700	2700	600	5400	**5266** 6000	**85.57** 93.61
CDS5	[dog] (60%)+[cat] (20%)+[airplane] (20%)	3600	1200	1200	4800	**5279** 6000	**85.79** 93.63
CDS6	[cat] (60%)+[dog] (20%)+[ship] (20%)	1200	3600	0	4800	**5287** 6000	**85.86** 93.62
CDS7	[dog] (80%)+[deer] (10%)+[bird] (10%)	4800	0	0	6000	**5280** 6000	**85.68** 93.63
CDS8	[airplane] (50%)+[ship] (30%)+[deer] (10%)+[horse] (10%)	0	0	3000	1200	**5394** 6000	**86.28** 93.65

In graphing and importance computation (G&IC), node number (NN) denotes the number of nodes that construct the graph, and computational time (CT) denotes the time cost (unit: s) of calculating the importance score. In the cell named G&IC, the data above represents the result of CPR while the data below represents the result of SHR. Bold denotes less NN and CT.

cost of using Hamming distance is stable and nearly 100 times less than those of using Cosine and Euclidean, which demonstrates that Hamming distance has overwhelming predominance over other metrics.

6.3 Assessment on Single-label Dataset

CIFAR-10 contains 10 categories: *airplane*, *automobile*, *bird*, *cat*, *deer*, *dog*, *frog*, *horse*, *ship*, and *truck*, where each category consists 6,000 images and the categories are completely mutually exclusive. In this part, we reorganize CIFAR-10 to form 8 sub-datasets. To be specific, we randomly select a certain proportion of data from each category to design the sub-datasets in Table 3.

We respectively search *dog*, *cat*, *airplane*, and *animals* images from BAIDU as AD to obtain the R(AD). Note that we will choose at least 10 images as AD for each requirement and then calculate their average value as the final assessment result. In addition, after obtaining the hash model, Table 3 lists the number of nodes used to construct graphs on 8 sub-datasets as well as the average time cost of calculating the importance score using DHR-DQA and SDQA, respectively. Table 4

Table 4. Assessment Results with Different AD on Designed Sub-datasets Sampled from CIFAR-10, Where RS Is the Random Selection, SDQA Is the Semantic-aware Data Quality Assessment and DHR-DQA Is Our Framework

AD	Method	Order(R(AD))								Collected Number		
		CDS1	CDS2	CDS3	CDS4	CDS5	CDS6	CDS7	CDS8	top1	top2	top3
dog	RS	2	7	5	4	3	6	8	1	0	600	4200
	SDQA	7(0.035)	5(0.284)	4(0.291)	3(0.618)	2(0.747)	6(0.278)	1(0.923)	8(0.011)	**4800**	**8400**	**11100**
	DHR-DQA	7(0.098)	6(0.273)	5(0.311)	3(0.592)	2(0.662)	4(0.333)	1(0.915)	8(0.021)	**4800**	**8400**	**11100**
cat	RS	4	1	3	6	2	8	7	5	0	1200	2400
	SDQA	5(0.080)	8(0.005)	3(0.456)	2(0.616)	4(0.278)	1(0.706)	6(0.017)	7(0.008)	**3600**	**6300**	**7500**
	DHR-DQA	5(0.093)	7(0.022)	4(0.300)	2(0.553)	3(0.318)	1(0.617)	6(0.043)	8(0.002)	**3600**	**6300**	**7500**
airplane	RS	6	2	4	8	1	7	5	3	1200	2400	5400
	SDQA	4(0.110)	2(0.137)	7(0.059)	5(0.083)	3(0.128)	8(0.051)	6(0.067)	1(0.643)	**3000**	**4200**	**5400**
	DHR-DQA	4(0.136)	2(0.191)	7(0.046)	5(0.111)	3(0.163)	8(0.022)	6(0.052)	1(0.577)	**3000**	**4200**	**5400**
zoo	RS	3	1	7	8	4	5	2	6	1200	7200	10800
	SDQA	6(0.433)	8(0.086)	5(0.719)	3(0.849)	2(0.852)	4(0.730)	1(0.909)	7(0.088)	**6000**	10800	**16200**
	DHR-DQA	6(0.577)	7(0.168)	5(0.789)	2(0.924)	3(0.835)	4(0.806)	1(0.976)	8(0.133)	**6000**	**11400**	**16200**

Order is the sequence number generated by R(AD). Top is the collected number statistics according to the order. Bold denotes more number of relevant data.

gives the R(AD) and selection results by RS, SDQA, and DHR-DQA, where RS is the average result after random sorting for 10 times.

From Table 3, DHR-DQA spends less resource consumption than SDQA, which not only reduces the number of nodes used to construct the graph but also speeds up calculating the importance score. This benefits from that CPR merges the same hash codes and thus greatly reduces the scale of iteration matrix.

From Table 4, both DHR-DQA and SDQA produce an overwhelming advantage over RS. DHR-DQA obtains the same selection result as SDQA when using single-label images as AD, but the R(AD) of DHR-DQA is closer to the proportion of the target in the dataset. At the same time, DHR-DQA shows better consistency than SDQA when assessing those datasets where the targets occupy the same proportion. For example, for *dog* or *cat*, if the data depicting animals have a higher proportion in the dataset, then it will obtain a higher R(AD). On the contrary, for *airplane*, if the data depicting machines have a higher proportion, it will obtain a higher score. The reason for this phenomenon lies in that DHR-DQA better perceives the semantics of local data and give a global score based on the semantic propagation. When using *animals* as AD, DHR-DQA produces a better result than SDQA, which reflects GAH has a stronger ability to under multi-label images compared with IDSTH (conclusion on Section 6.1).

6.4 Assessment on Multi-label Dataset

In this part, we implement the experiment on VOC2007. Different from single-label images, it calls for stronger semantic expression ability of hash codes and put forwards high requirements for the judgment ability of semantic importance owing that each image contains multiple objects.

The key of images in VOC2007 is *person*, and the specific object distribution as well as the co-occurrence relationships are listed in Table 5. With this condition, we further explore the

Table 5. Co-occurrence Relationships Statistics for VOC2007, Where MD Denotes Data of the Most Co-occurrence Category as Well as SMD Denotes Data of the Second Most Co-occurrence Category

	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	diningtable	dog	horse	motorbike	person	pottedplant	sheep	sofa	train	tvmonito
amount	442	482	612	353	456	360	1434	659	862	268	390	839	561	467	4015	469	193	452	520	485
MD	person	person	person	person	person	person	person	person	person	person	chair	person	person	person	car	person	person	chair	person	person
co-occurrence amount	55	311	43	105	281	157	502	63	333	64	289	231	439	325	498	131	27	153	103	154
SMD	car	car	boat	car	diningtable	bicycle	bus	chair	diningtable	horse	bottle	sofa	car	car	chair	chair	dog	person	car	chair
co-occurrence amount	16	31	8	14	114	17	119	44	226	10	87	85	23	65	473	128	8	150	25	153

Table 6. Reorganization Sub-datasets on VOC2007 Where [] Denotes the Category, () Denotes Selected Quantity, <> Denotes the Multiple of Replication, {} Denotes the Number of Co-occurrence Images

Sub-dataset	Reconstruction Scheme	Amount				G&IC	
		car	dog	PC	PD	NN	CT(s)
VDS1	[car] (1434)<1>+[person] (4015)<1>	1932	242	5449{1004}	4520{239}	**4899**	**81.52**
						5449	88.18
VDS2	[dog] (839)<7>	77	5873	1638{56}	5873{1617}	**809**	**3.35**
						5873	91.25
VDS3	[car] (1434)<4>	5736	77	5736{2008}	2020{32}	**1411**	**5.92**
						5736	90.37
VDS4	[dog] (839)<2>+[person] (4015)<1>	524	1955	4483{518}	5693{693}	**4571**	**80.02**
						5693	89.58
VDS5	[car] (1434)<2>+[aeroplane] (442)<6>	2964	22	3282{1016}	1340{16}	**1831**	**7.13**
						5520	89.08
VDS6	[dog] (839)<5>+[sheep] (193)<5>	70	4235	1315{45}	4340{1185}	**1002**	**4.05**
						5160	85.11
VDS7	[car] (1434)<2>+[dog] (839)<3>	2901	2539	2380{1028}	3596{709}	**2240**	**10.66**
						5385	86.02

PC is the image depicting *person* and *car*, and PD is the image depicting *person* and *dog*. In graphing and importance computation (G&IC), node number (NN) denotes the number of nodes that construct the graph, and computational time (CT) denotes the time cost (unit: s) of calculating the importance score. In the cell named G&IC, the data above represents the result of CPR while the data below represents the result of SHR. The bold font denotes less NN and CT.

perception ability of our framework on sub-important objects. As shown in Table 6, we reorganize VOC2007 to form 7 sub-datasets according to a certain class. Note that, there is overlap in a new sub-dataset.

We randomly select 10 images that contain both *person* and *car*, and 10 images that contain both *person* and *dog* from VOC2007. At the same time, we randomly select 10 images that contain *car* (i.e., *automobile*) and 10 images that contain *dog* from CIFAR-10. After obtaining the hash model, Table 6 lists the number of nodes used to construct graphs on 7 sub-datasets as well as the average time cost of calculating the importance score using DHR-DQA and SDQA, respectively.

From Table 6, DHR-DQA uses smaller nodes and owns faster computing speed than SDQA. Especially on VDS2, DHR-DQA reduces the scale of nodes by 6 times and time cost by 29 times. This is because that there are many overlapping data in the VDS construction. and these data participate in constructing the graph as well as calculating the importance score, which shows our superiority of resource consumption.

Table 7. Assessment Results with Different AD on Designed Sub-datasets Designed from VOC2007, Where RS Is the Random Selection, SDQA Is the Semantic-aware Data Quality Assessment, and DHR-DQA Is Our Framework

AD	Method	Order(R(AD))							Collected Number		
		VDS1	VDS2	VDS3	VDS4	VDS5	VDS6	VDS7	top1	top2	top3
car	RS	3	6	7	5	4	1	2	70	2971	4903
	SDQA	4(0.235)	6(0.001)	1(0.999)	5(0.037)	3(0.657)	6(0.001)	2(0.663)	**5736**	8637	**11601**
	DHR-DQA	4(0.359)	7(0.001)	1(0.992)	5(0.073)	2(0.516)	6(0.002)	3(0.507)	**5736**	**8700**	**11601**
dog	RS	6	5	4	7	1	3	2	22	2561	6796
	SDQA	5(0.009)	1(0.999)	6(0.001)	4(0.147)	7(0.000)	2(0.895)	3(0.366)	**5873**	**10108**	**12647**
	DHR-DQA	5(0.037)	1(0.993)	6(0.016)	4(0.321)	7(0.003)	2(0.837)	3(0.484)	**5873**	**10108**	**12647**
PC	RS	3	1	2	4	7	5	6	1638{56}	7374{2064}	12823{3068}
	SDQA	2(0.977)	6(0.102)	1(0.999)	3(0.886)	4(0.824)	7(0.0891)	5(0.774)	**5736{2008}**	**11185{3012}**	**15668**{3530}
	DHR-DQA	2(0.918)	6(0.177)	1(0.981)	4(0.715)	3(0.738)	7(0.116)	5(0.684)	**5736{2008}**	**11185{3012}**	14467{**4018**}
PD	RS	3	6	2	7	4	1	5	4340{1185}	6360{1217}	10880{1456}
	SDQA	4(0.938)	2(0.997)	6(0.067)	1(0.998)	7(0.052)	3(0.951)	5(0.924)	5693{693}	**11566**{2310}	**15906{3495}**
	DHR-DQA	5(0.788)	1(0.963)	6(0.114)	3(0.863)	7(0.086)	2(0.922)	4(0.797)	**5873{1617}**	10213{**2802**}	**15906{3495}**

Order is the sequence number generated by R(AD). Top denotes the collected number statistics according to the order, where {} is the number of co-occurrence images. Bold denotes more number of relevant data.

From Table 7, both DHR-DQA and SDQA achieve much better results than RS. Compared with SDQA, DHQ-DQA obtains a similar effect on *car* and a slightly better effect on *dog*. Although the effect is similar, the score obtained by DHR-DQA has better stability. According to the R(AD) for VDS1, VDS2, VDS4, and VDS6 using *car* as AD as well as the R(AD) for VDS1, VDS3, VDS4, and VDS7 using *dog* as AD, we find the result of DHR-DQA is closer to the real proportion for those data with a less than 50% proportion in the dataset. With PC and PD as AD, there are different assessment standards between DHR-DQA and SDQA. According to the R(AD) for VSD4 and VSD5 using PC as AD as well as the R(AD) for VSD2, VSD4, and VSD6 using PD as AD, DHR-DQA can measure not only the number of relevant target but also the number of co-occurrence data corresponding to AD. This illustrates that GAH owns better perception ability on multi-label images and DHR-DQA is more suitable for assessing multi-label datasets.

Based on the above experiments, our framework can assess the value for both single-label and multi-label dataset according to given AD, which helps the user choose more priority mining objects and obtain more valuable data.

6.5 Assessment on Real-world Dataset

In this part, we verify our framework can produce relevance assessment for different requirements by setting a variety of AD on Tencent1M dataset. We set the semantic requirement of task-*A* is that a man is playing on the lake and then use an image containing the tourist and lake as AD. We set the semantic requirement of task-*B* is also that a man is playing on the lake, but we use two equally weighted images that respectively contain the tourist and lake as AD. We set the semantic requirement of task-*C* is that a man is driving the car, and use two different weights to evaluate for the AD with separate contents when we use two kinds of AD to evaluate.

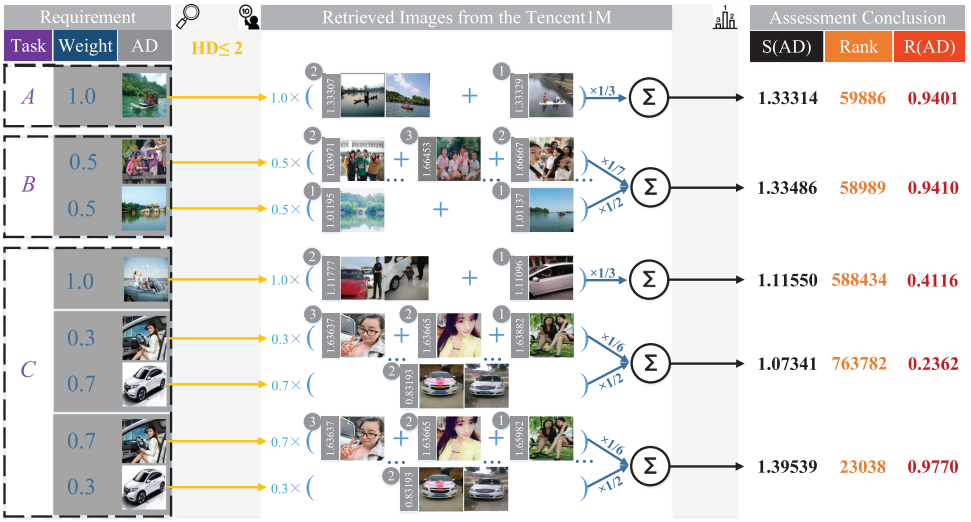


Fig. 10. Assessment with one AD and multiple AD on Tencent1M dataset.

Based on our DHR-DQA framework, we construct the graph by nodes mapped from Tencent1M, then find the number of nodes is about 0.56M (there exist a large number of three or five consecutive duplicate data in this dataset), the time cost of constructing iteration matrix is 4.3 hours and the iteration time is 1.6 minutes, which only spends the 44.14% and 32% time cost of SDQA. Figure 10 illustrates the assessment conclusions under 3 tasks. Task-*A* retrieves 3 images that contain the similar content as AD, where two images share the same hash code with $S = 1.33307$. According to Equation (1), S(AD) = 1.33314 and R(AD) = 0.9401 based on the rank of Importance list. Task-*B* has the same requirement as Task-*A* except that there exists no *bamboo raft* in its AD. Our framework perceives this, but still gets the similar assessment conclusion as task-*A*, which shows the hash model not only correctly understands the targets in images but also owns stability to calculate the importance score. As for Task-*C*, we give different weights to same targets and assess the same tasks. The results show the user's bias to a certain target will have an importance impact on judgment of the relevance. When choosing *person* as a more important semantic target, the R(AD) = 0.9770 while R(AD) = 0.2362 when using *car* as a more important target. At the same time, the AD that depicts a man is driving the car obtains a medium R(AD) = 0.4116. This reflects *person* is the main semantic content of Tencent1M, and *cat* occupies the rare content in the dataset. Therefore, the combination of *person* and *car* neutralizes the importance scores of these two targets, so the medium R(AD) is deemed reasonable.

7 RELATED WORKS

7.1 Data Quality Assessment

The definition of DQ was proposed in the 1990s. Professor Richard Y. Wang first divided four categories including 15 dimensions for DQ and enlightened about assessment [32]. However, as an important dimension of DQ, the relevance lacks enough research on assessment because of blur concept of relevance and evaluation standard. Shirlee-ann Knight introduce this concept to manage relevance quality of algorithm for crawling search engine [12]. Cai Li et al. proposed a series of challenges of DQ in big data and mentioned the importance for relevance [3]. SDQA [22]

is the first framework for DQ assessment at relevance dimension, which uses IDSTH, SHR and retrieval technology to achieve evaluation. Our framework draws on the backbone of this work.

7.2 Dark Data Solution

Dark data were proposed by Heidorn in 2008 and demonstrated owing value by long tail theory [10]. However, the concept and processing of dark data are controversial. A mainstream solution is to establish an active query mechanism to change the state of dark data. Cafarella mentioned that the value of dark data depends on both the requirements of the task and the ability of value extraction [2]. Ce Zhang proposed GeoDeepDive [45] and DeepDive [46] to mine information in dark data. But these works all implement on local data, and lack of overall cognition on the data in the known space. For improving these jobs, our framework builds a ranked list with similarity hash codes that helps any data match and understand the dark data.

7.3 Image Similarity Hashing

Similarity hash is a conventional method for unstructured data search, especially for image. Locality sensitive hashing (LSH) [51], as a famous data-independent hashing method, explores the random projections to embed high-dimensional features into a low-dimensional Hamming space. Data-aware hashing methods [31, 35, 38–40, 42] supersede LSH because of recognition ability for local features, especially the counterpart based on CNN. With development of machine learning, some hash methods [9, 34, 36] introduce GAN to extract salient features serving for hashing. Meanwhile, several studies [8, 20, 29, 30] apply quantization learning to controlling distances between hash codes. For exploiting coding policy, some hash methods [6, 25] complete hashing by reinforcement learning. These hash algorithms make full use of the advantages of deep learning and achieve better results, but lack of research on multi-objective hash learning. Some researchers pay attentions to this problem and proposed multi-label image hash methods [11, 14, 16]. However, they neglect relationship between objects resulting in falling into the bottleneck. GCN-based hashing methods can further improve the accuracy, because it not only learns representation by CNN but also guides this representation learning by an input of relationship. GCH [37] constructs the relationship between texts according to the appearance of labels, which conducts the feature extraction and hash mapping of image data. GCNH [50] constructs different graphs (relationships according to training set features, anchor points and query features) as GCN input, which achieves the state-of-the-art performance. However, they cannot implement on the unlabeled data.

7.4 PageRank Algorithm

PageRank [24] considers out-degree of related nodes as impact factor for data ranking. Fabian et.al. [27] applies random walking to ranking community images for searching, which has achieved good results. Personalized Rank [15] introduced the concept of probability to improve the RegEx in PageRank. SHR [33] is the first job that applies PageRank on graph constructed by similarity hash codes. However, it ignores the benefits resulting from hashing collisions.

8 CONCLUSION

For allocating data mining resource and thus acquiring more desired data, we propose a DHR-DQA framework to assess the relevance between image dark data and requirements. Our framework contains off-line and on-line parts. The off-line part first designs a GAH algorithm to recognize each data, then constructs a graph with restricted Hamming distance, and finally designs a CPR algorithm to calculate the importance score for each node (image) so as to obtain the relevance representation based on semantic propagation. The on-line part first retrieves the importance score by hash codes, then quickly gets the assessment conclusion by looking up the table (i.e., importance

list). Experimental results show GAH achieves better effects than existing unsupervised hash algorithms, especially on the multi-label datasets, which greatly promotes the perception ability of our framework. In addition, compared with SHR, CPR merges the same hash codes and thus reduces the resources consumption as well as time cost without affecting the sorting results. At the same time, our framework can gain the most desired data with the same mining resources and complete the assessment with a stability for given different requirements on a real-world dataset.

REFERENCES

[1] Danilo Ardagna, Cinzia Cappiello, Walter Samá, and Monica Vitali. 2018. Context-aware data quality assessment for big data. *Fut. Gen. Comput. Syst.* 89 (2018), 548–562.

[2] Michael J. Cafarella, Ihab F. Ilyas, Marcel Kornacker, Tim Kraska, and Christopher Ré. 2016. Dark data: Are we solving the right problems? In *ICDE*. 1444–1445.

[3] Li Cai and Yangyong Zhu. 2015. The challenges of data quality and data quality assessment in the big data era. *Data Sci. J.* 14 (2015), 2.

[4] Yue Cao, Mingsheng Long, Bin Liu, and Jianmin Wang. 2018. Deep cauchy hashing for hamming space retrieval. In *CVPR*. 1229–1237.

[5] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. 2019. Multi-label image recognition with graph convolutional networks. In *CVPR*. 5177–5186.

[6] Yueqi Duan, Ziwei Wang, Jiwen Lu, Xudong Lin, and Jie Zhou. 2018. GraphBit: Bitwise interaction mining via deep reinforcement learning. In *CVPR*. 8270–8279.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. 2010. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis.* 88, 2 (Jun. 2010), 303–338.

[8] Lianli Gao, Xiaosu Zhu, Jingkuan Song, Zhou Zhao, and Heng Tao Shen. 2019. Beyond product quantization: Deep progressive quantization for image retrieval. In *IJCAI*. 723–729.

[9] Tao He, Yuan-Fang Li, Lianli Gao, Dongxiang Zhang, and Jingkuan Song. 2019. One network for multi-domains: Domain adaptive hashing with intersectant generative adversarial networks. In *IJCAI*. 2477–2483.

[10] P. Bryan Heidorn. 2008. Shedding light on the dark data in the long tail of science. *Libr. Trends* 57, 2 (2008), 280–299.

[11] Chang-Qin Huang, Shang-Ming Yang, Yan Pan, and Hanjiang Lai. 2018. Object-location-aware hashing for multi-label image retrieval via automatic mask learning. *IEEE Trans. Image Process.* 27, 9 (2018), 4490–4502.

[12] Shirlee-ann Knight and Janice Burn. 2005. Developing a framework for assessing information quality on the world wide web. *Inf. Sci.* 8 (2005), 159–172.

[13] Alex Krizhevsky, Geoffrey Hinton, et al. 2009. *Learning Multiple Layers of Features from Tiny Images*. Technical Report. Citeseer.

[14] Hanjiang Lai, Pan Yan, Xiangbo Shu, Yunchao Wei, and Shuicheng Yan. 2016. Instance-aware hashing for multi-label image retrieval. *IEEE Trans. Image Process.* 25, 6 (2016), 2469–2479.

[15] Yu Lei, Wenjie Li, Ziyu Lu, and Miao Zhao. 2017. Alternating pointwise-pairwise learning for personalized item ranking. In *CIKM*. 2155–2158.

[16] Jun Li, Xianglong Liu, Wenxuan Zhang, Mingyuan Zhang, Jingkuan Song, and Nicu Sebe. 2020. Spatio-temporal attention networks for action recognition and detection. *IEEE Trans. Multimedia* 22, 11 (2020), 2990–3001.

[17] Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. 2017. Factorized bilinear models for image recognition. In *ICCV*. 2098–2106.

[18] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. 2016. Learning compact binary descriptors with unsupervised deep neural networks. In *CVPR*. 1183–1192.

[19] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common objects in context. In *ECCV*. 740–755.

[20] Xianglong Liu, Qiang Fu, Deqing Wang, Xiao Bai, Xinyu Wu, and Dacheng Tao. 2020. Distributed complementary binary quantization for joint hash table learning. *IEEE Trans. Neur. Netw. Learn. Syst.* 31, 12 (2020), 5312–5323.

[21] Yu Liu, Jingkuan Song, Ke Zhou, Lingyu Yan, Li Liu, Fuhao Zou, and Ling Shao. 2019. Deep self-taught hashing for image retrieval. *IEEE Trans. Cybernet.* 49, 6 (2019), 2229–2241.

[22] Yu Liu, Yangtao Wang, Ke Zhou, Yujuan Yang, and Yifei Liu. 2020. Semantic-aware data quality assessment for image big data. *Fut. Gen. Comput. Sci.* 102 (2020), 53–65.

[23] Yu Liu, Yangtao Wang, Ke Zhou, Yujuan Yang, Yifei Liu, Jingkuan Song, and Zhili Xiao. 2019. A framework for image dark data assessment. In *APWeb-WAIM*. 3–18.

[24] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. *The PageRank Citation Ranking: Bringing Order to the Web*. Technical Report. Stanford InfoLab.

[25] Yuxin Peng, Jian Zhang, and Zhaoda Ye. 2020. Deep reinforcement learning for image hashing. *IEEE Trans. Multimedia* 22, 8 (2020), 2061–2073.
[26] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*. 1532–1543.
[27] Fabian Richter, Stefan Romberg, Eva Hörster, and Rainer Lienhart. 2010. Multimodal ranking for image search on community databases. In *MIR*. 63–72.
[28] Manish Shukla, Sumesh Manjunath, Rohit Saxena, Sutapa Mondal, and Sachin Lodha. 2015. POSTER: WinOver enterprise dark data. In *SIGSAC*. 1674–1676.
[29] Jingkuan Song, Lianli Gao, Li Liu, Xiaofeng Zhu, and Nicu Sebe. 2018. Quantization-based hashing: A general framework for scalable image and video retrieval. *Pattern Recogn.* 75 (2018), 175–187.
[30] Jingkuan Song, Xiaosu Zhu, Lianli Gao, Xin-Shun Xu, Wu Liu, and Heng Tao Shen. 2019. Deep recurrent quantization for generating sequential binary codes. In *IJCAI*. 912–918.
[31] Jingdong Wang, Ting Zhang, Jingkuan Song, Nicu Sebe, and Heng Tao Shen. 2018. A survey on learning to hash. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 4 (2018), 769–790.
[32] Richard Y. Wang and Diane M. Strong. 1996. Beyond accuracy: What data quality means to data consumers. *J. Manage. Inf. Syst.* 12, 4 (1996), 5–33.
[33] Yangtao Wang, Yu Liu, Yifei Liu, Ke Zhou, Yujuan Yang, Jiangfeng Zeng, Xiaodong Xu, and Zhili Xiao. 2019. Analysis and management to hash-based graph and rank. In *APWeb-WAIM*. 289–296.
[34] Yuebin Wang, Liqiang Zhang, Feiping Nie, Xingang Li, Zhijun Chen, and Faqiang Wang. 2020. WeGAN: Deep image hashing with weighted generative adversarial networks. *IEEE Trans. Multimedia* 22, 6 (2020), 1458–1469.
[35] Yan Wu, Xianglong Liu, Haotong Qin, Ke Xia, Sheng Hu, Yuqing Ma, and Meng Wang. 2021. Boosting temporal binary coding for large-scale video search. *IEEE Trans. Multimedia* 23 (2021), 353–364.
[36] De Xie, Cheng Deng, Chao Li, Xianglong Liu, and Dacheng Tao. 2020. Multi-task consistency-preserving adversarial hashing for cross-modal retrieval. *IEEE Trans. Image Process.* 29 (2020), 3626–3637.
[37] Ruiqing Xu, Chao Li, Junchi Yan, Cheng Deng, and Xianglong Liu. 2019. Graph convolutional network hashing for cross-modal retrieval. In *IJCAI*. 982–988.
[38] Yi Xu, Xianglong Liu, Binshuai Wang, Renshuai Tao, Ke Xia, and Xianbin Cao. 2021. Fast nearest subspace search via random angular hashing. *IEEE Trans. Multimedia* 23 (2021), 342–352.
[39] Erkun Yang, Tongliang Liu, Cheng Deng, Wei Liu, and Dacheng Tao. 2019. DistillHash: Unsupervised deep hashing by distilling data pairs. In *CVPR*. 2946–2955.
[40] Huei-Fang Yang, Kevin Lin, and Chu-Song Chen. 2018. Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.* 40, 2 (2018), 437–451.
[41] Yang Yang, Yadan Luo, Weilun Chen, Fumin Shen, Jie Shao, and Heng Tao Shen. 2016. Zero-shot hashing via transferring supervised knowledge. In *MM*. 1286–1295.
[42] Zhaoda Ye and Yuxin Peng. 2020. Sequential cross-modal hashing learning via multi-scale correlation mining. *ACM Trans. Multim. Comput. Commun. Appl.* 15, 4 (2020), 105:1–105:20.
[43] Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *ICCV*. 1839–1848.
[44] Xiaofeng Yuan, Biao Huang, Yalin Wang, Chunhua Yang, and Weihua Gui. 2018. Deep learning-based feature representation and its application for soft sensor modeling with variable-wise weighted SAE. *IEEE Trans. Industr. Inf.* 14, 7 (2018), 3235–3243.
[45] Ce Zhang, Vidhya Govindaraju, Jackson Borchardt, Tim Foltz, Christopher Ré, and Shanan Peters. 2013. GeoDeepDive: Statistical inference using familiar data-processing languages. In *SIGMOD*. 993–996.
[46] Ce Zhang, Jaeho Shin, Christopher Ré, Michael J. Cafarella, and Feng Niu. 2016. Extracting databases from dark data with deepdive. In *SIGMOD*. 847–859.
[47] Haofeng Zhang, Li Liu, Yang Long, and Ling Shao. 2018. Unsupervised deep hashing with pseudo labels for scalable image retrieval. *IEEE Trans. Image Process.* 27, 4 (2018), 1626–1638.
[48] Ke Zhou, Yu Liu, Jingkuan Song, Linyu Yan, Fuhao Zou, and Fumin Shen. 2015. Deep self-taught hashing for image retrieval. In *MM*. 1215–1218.
[49] Ke Zhou, Yangtao Wang, Yu Liu, Yujuan Yang, Yifei Liu, Guoliang Li, Lianli Gao, and Zhili Xiao. 2020. A framework for image dark data assessment. *World Wide Web* 23, 3 (2020), 2079–2105.
[50] Xiang Zhou, Fumin Shen, Li Liu, Wei Liu, Liqiang Nie, Yang Yang, and Heng Tao Shen. 2020. Graph convolutional network hashing. *IEEE Trans. Cybern.* 50, 4 (2020), 1460–1472.
[51] Erkang Zhu, Fatemeh Nargesian, Ken Q. Pu, and Renée J. Miller. 2016. LSH ensemble: Internet-scale domain search. *Proc. VLDB* 9, 12 (2016), 1185–1196.

Received March 2020; revised August 2020; accepted August 2020