# Heterogeneous Translated Hashing: A Scalable Solution Towards Multi-Modal Similarity Search

YING WEI, Hong Kong University of Science and Technology
YANGQIU SONG, University of Illinois at Urbana-Champaign
YI ZHEN, Georgia Institute of Technology
BO LIU and QIANG YANG, Hong Kong University of Science and Technology

Multi-modal similarity search has attracted considerable attention to meet the need of information retrieval across different types of media. To enable efficient multi-modal similarity search in large-scale databases recently, researchers start to study multi-modal hashing. Most of the existing methods are applied to search across multi-views among which explicit correspondence is provided. Given a multi-modal similarity search task, we observe that abundant multi-view data can be found on the Web which can serve as an auxiliary bridge. In this paper, we propose a *Heterogeneous Translated Hashing* (HTH) method with such auxiliary bridge incorporated not only to improve current multi-view search but also to enable similarity search across heterogeneous media which have no direct correspondence. HTH provides more flexible and discriminative ability by embedding heterogeneous media into different Hamming spaces, compared to almost all existing methods that map heterogeneous data in a common Hamming space. We formulate a joint optimization model to learn hash functions embedding heterogeneous media into different Hamming spaces, and a translator aligning different Hamming spaces. The extensive experiments on two real-world datasets, one publicly available dataset of Flickr, and the other MIRFLICKR-Yahoo Answers dataset, highlight the effectiveness and efficiency of our algorithm.

Categories and Subject Descriptors: H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.4 [**Information Systems Applications**]: Miscellaneous

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Hash function learning, heterogeneous translated hashing, similarity search, scalability

**ACM Reference Format:**
Ying Wei, Yangqiu Song, Yi Zhen, Bo Liu, and Qiang Yang. 2016. Heterogeneous translated hashing: A scalable solution towards multi-modal similarity search. ACM Trans. Knowl. Discov. Data 10, 4, Article 36 (July 2016), 28 pages.
DOI: http://dx.doi.org/10.1145/2744204



## 1. INTRODUCTION

With the explosive growth of data on and off the Web, heterogeneity arising from different data sources has become ubiquitous. There exist numerous interactions among a diverse range of heterogeneous media: automatically musicalizing a prepared

This work is supported by State Key Development Program for Basic Research of China 2014CB340304 and Hong Kong RGC Projects 621013, 620812, and 621211.
Authors' addresses: Y. Wei, B. Liu, and Q. Yang, Computer Science Department, Hong Kong University of Science and Technology, Hong Kong; Y. Song, University of Illinois at Urbana-Champaign, Urbana, IL; Y. Zhen, Georgia Institute of Technology, Atlanta, GA; emails: yweiad@cse.ust.hk, yqsong@gmail.com, zhenyisx@gmail.com, bliuab@cse.ust.hk, qyang@cse.ust.hk.

DOI: http://dx.doi.org/10.1145/2744204

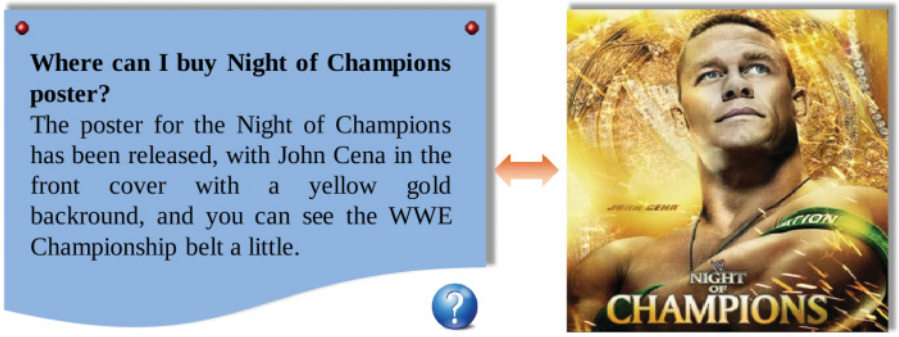



Fig. 1. An example of using images to help better question—answering. With a specific poster of "Night of Champions", the answer to where to buy can be more precise.

paragraph of lyrics, querying a picture with specific text descriptions, and recommending products based on social activities including sending messages, checking in at places, adding friends, and posting pictures. Figure 1 shows a simple example of leveraging images to provide more accurate answers in Question–Answering systems. All these applications, involving search, matching, and comparison, boil down to a fundamental problem: *similarity search across heterogeneous modalities*.

The challenges of similarity search across heterogeneous modalities are two-fold: (1) how to efficiently perform the computation to meet the large amount of data available; and (2) how to effectively compare the similarity with the existence of heterogeneity. A brute force similarity comparison between the examples from different media is prohibitively expensive for large-scale datasets. Traditional space partitioning methods which accelerate similarity search, such as KD-trees [Bentley 1975] and Metric trees [Uhlmann 1991], have poor performance in high-dimensional spaces [Weber et al. 1998]. Nowadays, initiated by locality sensitive hashing (LSH) [Andoni and Indyk 2006; Gionis et al. 1999], a line of research on hashing, a fast approximate nearest neighbour (ANN) technique, has thrived. Despite of the loss of accuracy, hashing-based algorithms enjoy constant or sub-linear query speed and low storage cost, and thereby have aroused more and more interest and become a main-stream technique. The key principle of hashing is to learn compact binary codes capable of preserving similarity. In other words, similar data points in the original feature space are projected to similar or same hash codes in the Hamming space. However, these methods all work with homogeneous data points.

It is non-trivial to put in hashing for similarity search across heterogeneous media. First, data from different media sources have incommensurable representation structures. For example, we could represent an image using raw real-valued pixels or more semantic Scale-Invariant Feature Transform (SIFT) features, while an article is featured by a vector with each element being a word from a dictionary. Second, the hashing algorithms are expected to not only preserve homogeneous media similarity in the way as traditional hashing does, but also simultaneously preserve heterogeneous media similarity. Heterogeneous media similarity is defined as semantic relatedness between a pair of instances in different modalities. For instance, a query image and a document in database are similar if they derive from the same topic, e.g., "computer" and closely semantic related. Such heterogeneous correspondence data that are labelled as similar or dissimilar play the role of "bridge" to search across heterogeneous media. However, in the task of illustrating questions with pictures as Figure 1 shows, questions as queries do not have any correspondence with the prepared database of

images. Therefore, the third challenge arises. The explicit relationships between query entities (in one domain) and the database entities (in another domain) probably do not apparently exist in the majority of practical applications.

So far, there have been limited attempts towards hashing across heterogeneous media. The existing works such as Bronstein et al. [2010], Kumar and Udupa [2011], Ou et al. [2013], Zhen and Yeung [2012a], and Zhou et al. [2014] all focus on the situation where explicit relationships are given. For example, the methods proposed in Kumar and Udupa [2011] and Zhou et al. [2014] assume that the data is formatted in a multi-view fashion, i.e., each data instance in the database should have a representation in each view. A Wikipedia page containing both illustrative pictures and describing paragraphs is a typical example of such multi-view data instance. In this case, explicit relationship is clearly given for each data instance. Particularly, the method proposed in Ou et al. [2013] even relies on the explicit relationships between queries and the database in the testing phase.

Moreover, almost all existing approaches embed multiple media data into a common Hamming space, thus generating hash codes with the same number of bits for all modalities. However, such an embedding is unreasonable because different media types usually have different dimensionality and distributions. In fact, researchers have argued that in uni-modal hashing using the same number of bits for all projected dimensions is unsound because dimensions of larger variances carry more information [Gong and Lazebnik 2011; Liu et al. 2011]. Analogously, heterogeneous media data with incommensurable representations and distributions also carry different amounts of information so that they should not be collectively hashed into binary codes of the same length. Otherwise, the equal treatment of different modalities can deteriorate the hashing performance. To the best of our knowledge, the works of Ou et al. [2013] and Wei et al. [2014] are among the first to adopt different bits for different modalities and correlate these bits with mapping functions. However, as mentioned above, Ou et al. [2013] re-learns hash codes for out-of-sample data highly reliant on the given relationships between queries and the database, which is neither practical nor efficient.

In this paper, we propose a novel learning method to enable translation-based hashing across heterogeneous media called *Heterogeneous Translated Hashing* (HTH) to address these limitations. Given a heterogeneous media search task, we observe that some multi-modal data are available on the Web which can serve as a bridge to preserve heterogeneous media similarity, while massive uncorrelated examples in each individual modality can be incorporated to enhance homogeneous media similarity preservation. Learning from such auxiliary heterogeneous correspondence data and homogeneous unlabelled data, HTH generates a set of hash functions for each modality that can project entities of each media type onto an individual Hamming space. All of the Hamming spaces are aligned with a learned translator. In practice, we formulate the above learning procedure as a joint optimization model. Despite the non-convex nature of the learning objective, we express it as a difference of two convex functions to enable the application of the constrained concave–convex procedure (CCCP) [Yuille et al. 2002] iteratively. Then, we employ the stochastic sub-gradient strategy [Shalev-Shwartz et al. 2007] to efficiently find the local optimum in each CCCP iteration. We also give a comprehensive analysis on the convergence of the proposed algorithm. Finally, we conduct extensive experiments on two real-world large-scale datasets and demonstrate our proposed method to be both effective and efficient.

The remainder of this paper is organized as follows. We review the related work in Section 2. In Section 3, we present the formulation, optimization details, and convergence analysis of the proposed method. Experimental results and analysis on two real-world datasets are shown in Section 4. Finally, Section 5 concludes the paper.

## 2. RELATED WORK

In this section, we briefly review the related work in two categories. We first introduce the recently developed learning to hash methods, which is the background of our approach. Then, we review several current state-of-the-art hashing methods across heterogeneous modalities.

### 2.1. Learning to Hash

LSH [Andoni and Indyk 2006; Gionis et al. 1999] and its variations [Datar et al. 2004; Kulis and Grauman 2009; Kulis et al. 2009; Raginsky and Lazebnik 2009], as the earliest exploration of hashing, generate hash functions from random projections or permutations. Nevertheless, these data-independent hash functions may not confirm to every application, and hence require very long hash codes, which increases costs of storage and online query, to achieve acceptable performances. Recently, data-dependent learning to hash methods attempt to alleviate the problem via learning hash functions from data. Researchers have explored learning to hash in all of the three settings, unsupervised, supervised, and semi-supervised hashing. (1) Unsupervised hashing: spectral hashing (SH) [Weiss et al. 2008], self-taught hashing (STH) [Zhang et al. 2010], and anchor graph hashing (AGH) [Liu et al. 2011] do not rely on any label information. Instead, they employ the intrinsic data structure and distribution to infer hash codes. (2) Supervised hashing: provided by labels of examples, boosting [Shakhnarovich et al. 2003], semantic hashing [Salakhutdinov and Hinton 2009], LDAHash [Strecha et al. 2012] seek hash codes which are capable of preserving label similarity. (3) Semi-supervised hashing: semi-supervised hashing [Wang et al. 2010] simultaneously takes advantages of both labelled and unlabelled examples, and deduces better hash codes therefore. These approaches have significantly improved the hashing results for many specific tasks.

### 2.2. Hash Acorss Heterogeneous Modalities

To the best of our knowledge, only a few research attempts towards multi-modal hashing have been made to speed up similarity search across different feature spaces or modalities.

Bronstein et al. [2010] first explored the cross-modality similarity search problem and proposed cross-modal similarity sensitive hashing (CMSSH). It embeds multi-modal data into a common Hamming space. Later, several works [Kumar and Udupa 2011; Wu et al. 2014; Zhai et al. 2013; Zhen and Yeung 2012a, 2012b; Zhu et al. 2013] were proposed. Both cross-view hashing (CVH) [Kumar and Udupa 2011] and inter-media hashing (IMH) [Song et al. 2013] extend SH to preserve intra-media and inter-media similarity simultaneously. Nevertheless, CVH enables cross-view similarity search given multi-view data whereas IMH adds a linear regression term to learn hash functions for efficient code generation of out-of-sample data. Zhen and Yeung [2012a] extended label-regularized max-margin partition (LAMP) [Mu et al. 2010] to the multi-modal case. Multi-modal latent binary embedding (MLBE) [Zhen and Yeung 2012b] presents a probabilistic model to learn binary latent factors which are regarded as hash codes in the common Hamming space. Parametric local multi-modal hashing (PLMH) [Zhai et al. 2013] extends MLBE and learns a set of local hash functions for each modality. Siddiquie et al. [2014] applies multi-modal hashing on a very interesting application of multi-modal image retrieval, in which the proposed method supports three kinds of queries, images, text descriptions, and sketches.

Recently, Zhu et al. [2013] and Wu et al. [2014] presented two new techniques for obtaining hash codes in multi-modal hashing. Zhu et al. [2013] obtains $k$-bit hash codes of a specific data point via thresholding its distances to $k$ cluster centres while [Wu et al.

2014] thresholds the learned sparse coefficients for each modality as binary codes. Wu et al. [2014] adopts the idea in Wu et al. [2014], which thresholds sparse coefficients to obtain hash codes, but performs the coupled dictionary learning across different modalities using sub-modular dictionary learning techniques different from Hypergraph Laplacian sparse coding in Wu et al. [2014]. Similarly, the work of Zhou et al. [2014] argues the superiority of hashing high-level latent semantic information over raw low-level features. So it first applies sparse coding and matrix factorization on raw image and text features, respectively, and then hashes the learned high-level sparse coefficients and latent representations. Unfortunately, this work is limited to perform similarity search among multi-view instances. [Ding et al. 2014], which applies Collective Matrix Factorization (CMF) [Singh and Gordon 2008] onto multi-modal hashing, faces the same problem. Deep learning has been applied to fields like computer vision, natural language processing, and so on where it has been proved to produce state-of-the-art results on various tasks. Wang et al. [2014] introduces stacked auto-encoders, a kind of deep learning architecture, into multi-modal similarity retrieval. Besides, the work provides a solution of hashing to enable the efficiency of retrieval. A disadvantage of this line of works is the inefficient online hash code computation. Either lasso optimization (in sparse coding) or forward-propagation (in deep learning) has to be conducted to obtain hash codes, which is unlikely to meet the real-time requirement.

All these methods assume that the hashed data reside in a common Hamming space. However, this may be inappropriate especially when the modalities are quite different. Relational-aware heterogeneous hashing (RaHH) [Ou et al. 2013] and HTH [Wei et al. 2014] address this problem by generating hash codes with different lengths (one for each modality) together with a mapping function. Unfortunately, RaHH has to (1) adopt a fold-in scheme to generate hash codes for out-of-sample data, which is time-consuming and devastating in real-time retrieval, because it learns codes directly instead of explicit hash functions; (2) require explicit relationship between queries and the database in both the training and testing phases, which are neither reasonable nor practical. The limitation (2) of RaHH also hampers the application of abundant auxiliary bridge data on the Web. In the next section, we elaborate our approach to eliminate these restrictions.

## 3. HETEROGENEOUS TRANSLATED HASHING

In this section, we present our approach in detail. We first introduce the general framework which consists of an offline training phase and an online querying phase. After introducing the notations and problem definitions, we show that HTH can be achieved by solving a novel optimization problem, and we develop an effective and efficient algorithm accordingly. The whole algorithm and the complexity analysis are given at the end of this section.

### 3.1. Overview

We illustrate the HTH framework in Figure 2. HTH involves two phases: an offline training phase (left) and an online querying phase (right). For the simplicity of presentation, we focus on two heterogeneous media types, namely, images and text documents. Nevertheless, it is straightforward to extend HTH to more general cases with three or more types of media. During the offline training phase, HTH learns (1) the hash functions for each media type to map the data to their individual Hamming space, which has dimensions of the number equalling the code length; and (2) a translator to align the two Hamming spaces. Since effective hash codes should simultaneously preserve homogeneous and heterogeneous media similarity, the correspondence between different domains is needed. In this work, we use auxiliary tagged images crawled from Flickr as the "bridge" shown in the central pink bounding box in Figure 2 which

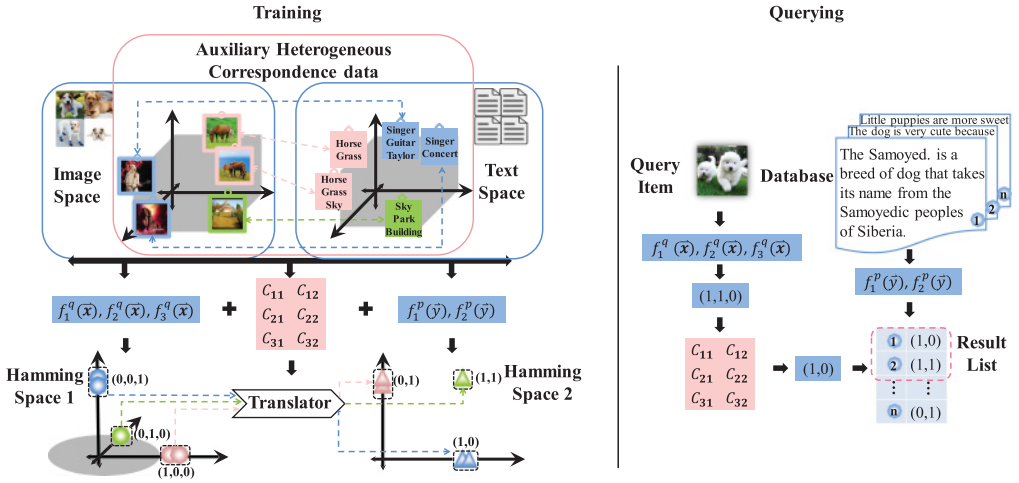



Fig. 2. The flowchart of the proposed Heterogeneous Translated Hashing framework. The left corresponds to the offline training of hash functions and the translator, and the right summarizes the process of online querying.

encloses images, documents, and their relationships. Meanwhile, a proportion of queries, e.g., images, are incorporated to enhance intra-similarity preservation together with auxiliary images as the left blue-bounding box which encloses all images shows. The same applies to text documents. With hash functions, similar homogeneous instances of each media type should be hashed into the same or close bucket in its Hamming space as displayed in Figure 2. Moreover, hash codes of one media type can be translated into the other Hamming space so that mutually correlated data points across different domains are expected to have small Hamming distances.

In the online querying phase, the database, e.g., a pile of text documents, is pre-encoded into a hash table via applying corresponding learned hash functions. When a new query instance comes, we first generate its hash codes using the domain specific hash functions. Subsequently, the hash codes are translated to the Hamming space of the database via the learned translator. Using existing hardware techniques such as bit operations, we can compute the Hamming distances between the query and all database instances, and retrieve its nearest neighbours efficiently.

### 3.2. Notations and Problem Definition

Suppose, we are given a few query data instances $\tilde{\mathbf{X}}^q = \{\tilde{\mathbf{x}}_i\}_{i=1}^N$ and a large database $\tilde{\mathbf{Y}}^p = \{\tilde{\mathbf{y}}_j\}_{j=1}^M$, where $\tilde{\mathbf{x}}_i \in \mathbb{R}^{d_q}$ is a $d_q$ dimensional feature vector in the query domain and $\tilde{\mathbf{y}}_j \in \mathbb{R}^{d_p}$ represents a $d_p$-dimensional vector in the feature space of the database. In addition, we are given a set of auxiliary data points from both modalities and their relationships which are expressed as a triple set $\mathcal{A}_{xy} = \cup_{i=1}^{N_1} \cup_{j=1}^{N_2} \{\mathbf{x}_i^*, \mathbf{y}_j^*, s_{ij}\}$, in which $s_{ij} = 1$ indicates that the instances $\mathbf{x}_i^*$ and $\mathbf{y}_j^*$ are correlated while $s_{ij} = 0$ otherwise.

We construct the training set $\mathcal{T}_x = \{\mathbf{x}_i\}_{i=1}^{N_x}$ of the query domain as follows: randomly sample $n$ instances from $\tilde{\mathbf{X}}^q$ and select all auxiliary data points corresponding to the query domain, i.e., $N_x = n + N_1$. Similarly, $\mathcal{T}_y = \{\mathbf{y}_j\}_{j=1}^{N_y}$ with $N_y = m + N_2$.

Our goal is to learn two sets of hash functions and a translator from the training set $\mathcal{A}_{xy}$, $\mathcal{T}_x$, and $\mathcal{T}_y$. The two sets of hash functions, $\mathcal{F}^q(\mathbf{x}) = \{f_k^q(\mathbf{x})\}_{k=1}^{k_q}$ and $\mathcal{F}^p(\mathbf{y}) = \{f_l^p(\mathbf{y})\}_{l=1}^{k_p}$, project the query and database domain into a $k_q$ dimensional and

Table I. Definition of Notations

| Notation | Description | Number | Set Notation |
|---|---|---|---|
| | Input | | |
| $\tilde{\mathbf{x}}_i$ | $i$th query instance | $N$ | $\tilde{\mathbf{X}}^q = \{\tilde{\mathbf{x}}_i\}_{i=1}^{N}$ |
| $\tilde{\mathbf{y}}_j$ | $j$th database instance | $M$ | $\tilde{\mathbf{Y}}^p = \{\tilde{\mathbf{y}}_j\}_{j=1}^{M}$ |
| $\{\mathbf{x}_i^*, \mathbf{y}_j^*, s_{ij}\}$ | a triple set of auxiliary pairs | $N_{xy} = N_1 \times N_2$ | $\mathcal{A}_{xy} = \cup_{i=1}^{N_1} \cup_{j=1}^{N_2} \{\mathbf{x}_i^*, \mathbf{y}_j^*, s_{ij}\}$ |
| $\mathbf{x}_i$ | $i$th training instance in query domain | $N_x$ | $\mathcal{T}_x = \{\mathbf{x}_i\}_{i=1}^{N_x}$ |
| $\mathbf{y}_j$ | $j$th training instance in database domain | $N_y$ | $\mathcal{T}_y = \{\mathbf{y}_j\}_{j=1}^{N_y}$ |
| | Output | | |
| $f_k^q(\mathbf{x})$ | $k$th hash function in query domain | $k_q$ | $\mathcal{F}^q(\mathbf{x}) = \{f_k^q(\mathbf{x})\}_{k=1}^{k_q}$ |
| $f_l^p(\mathbf{y})$ | $l$th hash function in database domain | $k_p$ | $\mathcal{F}^p(\mathbf{y}) = \{f_l^p(\mathbf{y})\}_{l=1}^{k_p}$ |
| $\mathbf{C}^{k_q \times k_p}$ | the $k_q \times k_p$ translator | | |
| $\mathbf{h}_k^q$ | $k$th hash code in query domain | $k_q$ | $\mathbf{H}^q = \{\mathbf{h}_k^q\}_{k=1}^{k_q}$ |
| $\mathbf{h}_l^p$ | $l$th hash code in database domain | $k_p$ | $\mathbf{H}^p = \{\mathbf{h}_l^p\}_{l=1}^{k_p}$ |

a $k_p$ dimensional Hamming space, respectively. The translator $\mathbf{C}^{k_q \times k_p}$ aligns the two Hamming spaces in a bitwise manner. Based on the hashing functions and the translator, we can generate hash codes $\mathbf{H}^q = \{\mathbf{h}_k^q\}_{k=1}^{k_q} \in \{-1, +1\}^{N \times k_q}$ in the query domain and $\mathbf{H}^p = \{\mathbf{h}_l^p\}_{l=1}^{k_p} \in \{-1, +1\}^{M \times k_p}$ in the database domain and perform accurate nearest neighbour retrieval across different media types. For brevity, we summarize these notations in Table I.

### 3.3. Learning Hash Functions and Translators

In this section, we introduce the objective function of our proposed HTH integrating both the homogeneous similarity preservation term and the heterogeneous similarity preservation term.

*3.3.1. Homogeneous Media Similarity Preservation.* A core criterion to preserve homogeneous media similarity is that similar data points in the original space should share similar hash codes within each single media type. To meet this criterion, in this work we first define the $k$th ($l$th) bit hash function of the query domain (the database domain) $f_k^q(\mathbf{x})$ ($f_l^p(\mathbf{y})$) as a linear projection which has been widely adopted in existing related works [Song et al. 2013; Zhang et al. 2010; Zhen and Yeung 2012a]:

$$f_k^q(\mathbf{x}) = sgn\left(\left(\mathbf{w}_k^q\right)^T \mathbf{x}\right) \text{ and } f_l^p(\mathbf{y}) = sgn\left(\left(\mathbf{w}_l^p\right)^T \mathbf{y}\right), \tag{1}$$

where $sgn(\cdot)$ is the sign function, and $\mathbf{w}_k^q$ and $\mathbf{w}_l^p$ denote projection vectors for the $k$th and $l$th bit hash codes in the query and database domain, respectively.

In each domain, we can treat each hash function above as a binary classifier and each bit $h_k^{q(i)} \in \{-1, +1\}$ of the $i$th query data point as a binary class label. The goal is to learn binary classifiers $f_1^q, \ldots, f_{k_q}^q$ to predict $k_q$ labels (bits) $h_1^q, \ldots, h_{k_q}^q$ for any query item $\mathbf{x}$. Moreover, we train binary classifiers for all the bits independently because different bits $h_1^q, \ldots, h_{k_q}^q$ should be uncorrelated. We propose to learn the hash function for the $k$th bit by solving the following optimization problem:

$$\mathcal{J}_{\mathbf{w}_k^q}^{ho} = \frac{1}{N_x} \sum_{i=1}^{N_x} \ell\big((\mathbf{w}_k^q)^T \mathbf{x}_i\big) + \gamma_q \Omega\big(\|\mathbf{w}_k^q\|_{\mathcal{H}}\big), \tag{2}$$

where $\ell(\cdot)$ denotes the loss function on one data point and $\Omega$ is a regularization term about functional norm $\|\mathbf{w}_k^q\|_{\mathcal{H}}$ in Hilbert spaces. Inspired by the large-margin criterion adopted by Support Vector Machine (SVM), we define $\ell$ using the hinge loss function $\ell((\mathbf{w}_k^q)^T \mathbf{x}_i) = [1 - h_k^{q(i)} (\mathbf{w}_k^q)^T \mathbf{x}_i]_+$, where $[a]_+$ returns $a$ if $a >= 0$ and 0 otherwise. $\Omega$ is commonly defined as the $L_2$-norm $\frac{1}{2}\|\mathbf{w}_k^q\|^2$. Note that $h_k^{q(i)} = f_k^q(\mathbf{x}_i) = sgn((\mathbf{w}_k^q)^T \mathbf{x}_i)$, the optimization objective (2) can be rewritten as

$$\mathcal{J}_{\mathbf{w}_k^q}^{ho} = \frac{1}{N_x} \sum_{i=1}^{N_x} \big[1 - \big|(\mathbf{w}_k^q)^T \mathbf{x}_i\big|\big]_+ + \frac{\gamma_q}{2} \|\mathbf{w}_k^q\|^2 + \left[\frac{1}{N_x} \left| \sum_{i=1}^{N_x} (\mathbf{w}_k^q)^T \mathbf{x}_i \right| - \delta \right]_+, \tag{3}$$

where $\gamma_q$ is a balancing parameter controlling the impact of the regularization, and the last term is to avoid a trivially optimal solution which assigns all $N_x$ data points to the same bit. Without the last constraint, the data points may be classified into the same side with large $|(\mathbf{w}_k^q)^T \mathbf{x}_i|$ value so that $[1 - |(\mathbf{w}_k^q)^T \mathbf{x}_i|]_+$ equals to 0 for all $N_x$ data points which is meaningless in hashing. Thus, we enforce $-\delta \le \frac{1}{N_x} |\sum_{i=1}^{N_x} (\mathbf{w}_k^q)^T \mathbf{x}_i| \le \delta$ with a pre-defined constant $\delta$.

Similarly, we can learn the hash functions for the database domain by minimizing the following objective function:

$$\mathcal{J}_{\mathbf{w}_l^p}^{ho} = \frac{1}{N_y} \sum_{j=1}^{N_y} \big[1 - \big|(\mathbf{w}_l^p)^T \mathbf{y}_j\big|\big]_+ + \frac{\gamma_p}{2} \|\mathbf{w}_l^p\|^2 + \left[\frac{1}{N_y} \left| \sum_{j=1}^{N_y} (\mathbf{w}_l^p)^T \mathbf{y}_j \right| - \delta \right]_+, \tag{4}$$

where $\gamma_p$ controls the impact of regularization. To learn hash functions from both query and database domains that preserve the homogeneous similarity, we combine (3) and (4) and derive the following objective function:

$$\begin{aligned} \mathcal{J}^{ho}(\mathbf{W}^q, \mathbf{W}^p) = & \sum_{k=1}^{k_q} \left\{ \frac{1}{N_x} \sum_{i=1}^{N_x} \big[1 - \big|(\mathbf{w}_k^q)^T \mathbf{x}_i\big|\big]_+ + \left[\frac{1}{N_x} \left| \sum_{i=1}^{N_x} (\mathbf{w}_k^q)^T \mathbf{x}_i \right| - \delta \right]_+ \right\} \\ & + \sum_{l=1}^{k_p} \left\{ \frac{1}{N_y} \sum_{j=1}^{N_y} \big[1 - \big|(\mathbf{w}_l^p)^T \mathbf{y}_j\big|\big]_+ + \left[\frac{1}{N_y} \left| \sum_{j=1}^{N_y} (\mathbf{w}_l^p)^T \mathbf{y}_j \right| - \delta \right]_+ \right\} \\ & + \frac{\gamma_q}{2} \|\mathbf{W}^q\|_F^2 + \frac{\gamma_p}{2} \|\mathbf{W}^p\|_F^2, \end{aligned} \tag{5}$$

where $\mathbf{W}^q = \{\mathbf{w}_1^q, \ldots, \mathbf{w}_{k_q}^q\}$, $\mathbf{W}^p = \{\mathbf{w}_1^p, \ldots, \mathbf{w}_{k_p}^p\}$, and $\|\cdot\|_F^2$ denotes the Frobenius norm. Using maximum margin-based classifiers, it is easy to project close data points to the

same side and thereby preserve the intrinsic similarity among different items in a single media type. Moreover, a larger margin between hash-generated partitions usually indicates better generalization ability to out-of-sample data [Mu et al. 2010]. Another notable advantage of utilizing SVM classifiers is its convenience of extending from linear projections to non-linear mappings, although we do not explore this potential in detail in this paper.

*3.3.2. Heterogeneous Media Similarity Preservation.* In the last subsection, we learn the hash codes that can preserve homogeneous media similarity for each media type. For the sake of flexibility and discrimination between two modalities, we adopt hash codes with different numbers of bits for different domains. To perform similarity search across different Hamming spaces, in this subsection, we introduce a translator $\mathbf{C}^{k_q \times k_p}$ to map the hash codes from a $k_q$-dimensional Hamming space to a $k_p$-dimensional Hamming space or vice versa. We also show that $\mathbf{C}$ can be learned from auxiliary heterogeneous pairs $\mathcal{A}_{xy} = \cup_{i=1}^{N_1} \cup_{j=1}^{N_2} \{\mathbf{x}_i^*, \mathbf{y}_j^*, s_{ij}\}$.

A good translator should have the following three properties: (1) semantically related points across different domains should have similar hash codes after translation; (2) semantically uncorrelated points across different domains should be far away from each other in the translated Hamming space; (3) it should have good generalization power. To obtain such a good translator, we propose to minimize the following heterogeneous loss function:

$$\mathcal{J}^{he} = \sum_{i,j}^{N_{xy}} \left[ s_{ij} d_{ij}^2 + (1 - s_{ij}) \tau(d_{ij}) \right] + \frac{\gamma_C}{2} \|\mathbf{C}\|_F^2, \tag{6}$$

where $d_{ij} = \sum_{l=1}^{k_p} [\sum_{k=1}^{k_q} C_{kl} (\mathbf{w}_k^q)^T \mathbf{x}_i^* - (\mathbf{w}_l^p)^T \mathbf{y}_j^*]^2$ represents the distance in the Hamming space of the database between the $i$th translated hash code from the query domain and the $j$th code string from the database domain. $\tau(\cdot)$ is the SCISD [Quadrianto and Lampert 2011] function specified by two parameters $a$ and $\lambda$:

$$\tau(d_{ij}) = \begin{cases} -\frac{1}{2} d_{ij}^2 + \frac{a\lambda^2}{2} & \text{if} \quad 0 \le |d_{ij}| \le \lambda \\ \frac{d_{ij}^2 - 2a\lambda |d_{ij}| + a^2\lambda^2}{2} & \text{if} \quad \lambda < |d_{ij}| \le a\lambda \cdot \\ 0 & \text{if} \quad |d_{ij}| > a\lambda \end{cases} \tag{7}$$

Note that if two data points are semantically similar, that is, $s_{ij} = 1$, we require that they have small $d_{ij}$; if they are semantically dissimilar, we require that they have small SCISD value which implies that they are far apart in the Hamming space. It is worth noting that $s_{ij}$ here is not limited to be binary, but could be real numbers according to specific applications.

*3.3.3. Overall Optimization Problem.* Combining the objective functions introduced in the previous two subsections, the overall optimization problem of HTH can be written as follows:

$$\min_{\mathbf{W}^q, \mathbf{W}^p, \mathbf{C}} \quad \mathcal{J}^{ho} + \beta \mathcal{J}^{he}, \tag{8}$$

where $\beta$ is a tradeoff parameter between homogeneous and heterogeneous loss functions.

### 3.4. Optimization

Problem (8) is non-trivial to solve because it is discrete and non-convex *w.r.t.* $\mathbf{W}^q$, $\mathbf{W}^p$, and $\mathbf{C}$. In the following, we develop an alternating algorithm to solve this problem which converges to a local minimum very quickly.

We first describe how to learn the projection vector $\mathbf{w}_k^q$ for the $k$th bit while fixing other variables. Note that projection vectors for different bits can be learned independently using the same algorithm. The objective function $w.r.t.$ $\mathbf{w}_k^q$ is as follows:

$$\mathcal{J}_{\mathbf{w}_k^q} = \frac{1}{N_x}\sum_{i=1}^{N_x}\left[1-\left|\left(\mathbf{w}_k^q\right)^T\mathbf{x}_i\right|\right]_+ + \left[\frac{1}{N_x}\left|\sum_{i=1}^{N_x}\left(\mathbf{w}_k^q\right)^T\mathbf{x}_i\right| - \delta\right]_+ + \beta\sum_{i,j}^{N_{xy}}\left[s_{ij}d_{ij}^2 + (1-s_{ij})\tau(d_{ij})\right] + \frac{\gamma_q}{2}\left\|\mathbf{w}_k^q\right\|^2. \tag{9}$$

Although (9) is not convex, it can be expressed as the differences of two convex functions, and hence can be minimized efficiently using CCCP [Yuille et al. 2002].

We briefly introduce the idea of CCCP here. Given an optimization problem in the form of $\min_x f(x) - g(x)$, where $f$ and $g$ are real-valued convex functions, the key idea of CCCP is to iteratively evaluate an upper bound of the objective function by replacing $g$ with its first-order Taylor expansion around the current solution, $x_t$, i.e., $\mathcal{R}(g(x_t)) = g(x_t) + \partial_x g(x_t)(x - x_t)$. Then, the relaxed sub-problem $f(x) - \mathcal{R}(g(x_t))$ is in convex form and can be solved by off-the-shelf convex solvers. The solution sequence $\{x_t\}$ obtained by CCCP is guaranteed to reach a local optimum.

Specifically, the upper bound of (9) in the $t$th CCCP iteration is as follows:

$$\mathcal{J}_{\mathbf{w}_k^q}^{(t)} = \frac{1}{N_x}\sum_{i=1}^{N_x}\left[f_1\left(\mathbf{w}_k^q\right) - \mathcal{R}\left(g_1\left(\mathbf{w}_k^{q(t)}\right)\right)\right] + \left[\frac{1}{N_x}\left|\sum_{i=1}^{N_x}\left(\mathbf{w}_k^q\right)^T\mathbf{x}_i\right| - \delta\right]_+ + \beta\sum_{i,j}^{N_{xy}}s_{ij}d_{ij}^2 + \beta\sum_{i,j}^{N_{xy}}(1-s_{ij})\left[\tau_1(d_{ij}) - \mathcal{R}\left(\tau_2\left(d_{ij}^{(t)}\right)\right)\right] + \frac{\gamma_q}{2}\left\|\mathbf{w}_k^q\right\|^2, \tag{10}$$

where $f_1(\mathbf{w}_k^q) = 1 + [|(\mathbf{w}_k^q)^T\mathbf{x}_i| - 1]_+$, $g_1(\mathbf{w}_k^{q(t)}) = |(\mathbf{w}_k^{q(t)})^T\mathbf{x}_i|$, $\tau_2(d_{ij}) = \frac{1}{2}d_{ij}^2 - \frac{a\lambda^2}{2}$, and

$$\tau_1(d_{ij}) = \begin{cases} 0 & \text{if} \quad 0 \le |d_{ij}| \le \lambda \\ \frac{ad_{ij}^2 - 2a\lambda|d_{ij}| + a\lambda^2}{2(a-1)} & \text{if} \quad \lambda < |d_{ij}| \le a\lambda. \\ \frac{1}{2}d_{ij}^2 - \frac{a\lambda^2}{2} & \text{if} \quad |d_{ij}| > a\lambda \end{cases} \tag{11}$$

The Taylor expansion of $g_1(\cdot)$ and $\tau_2(\cdot)$ around the value of $\mathbf{w}_k^q$ in the $t$th iteration are $\mathcal{R}(g_1(\mathbf{w}_k^{q(t)})) = |(\mathbf{w}_k^{q(t)})^T\mathbf{x}_i| + sgn((\mathbf{w}_k^{q(t)})^T\mathbf{x}_i)\cdot\mathbf{x}_i(\mathbf{w}_k^q - \mathbf{w}_k^{q(t)})$ and $\mathcal{R}(\tau_2(d_{ij}^{(t)})) = \frac{1}{2}d_{ij}^{(t)2} - \frac{a\lambda^2}{2} + d_{ij}^{(t)}\frac{\partial d_{ij}^{(t)}}{\mathbf{w}_k^q}(\mathbf{w}_k^q - \mathbf{w}_k^{q(t)})$, respectively. Note that

$$d_{ij}^{(t)} = \sum_{l=1}^{k_p}\left[\sum_{k=1}^{k_q}C_{kl}\left(\mathbf{w}_k^{q(t)}\right)^T\mathbf{x}_i^* - \left(\mathbf{w}_l^p\right)^T\mathbf{y}_j^*\right]^2. \tag{12}$$

However, minimizing (10) is time-consuming if the data dimensionality is high. As a result, we employ Pegasos [Shalev-Shwartz et al. 2011] which is a sub-gradient-based solver and reported to be one of the fastest gradient-based solvers, to solve the problem. In each Pegasos iteration, the key step is to evaluate the sub-gradient of $\mathcal{J}_{\mathbf{w}_k^q}^{(t)}$ $w.r.t.$ $\mathbf{w}_k^q$

**ALGORITHM 1:** Heterogeneous Translated Hashing (HTH)

**Require:**
$\mathcal{T}_x = \{\mathbf{x}_i\}_{i=1}^{N_x}$ – query training set
$\mathcal{T}_y = \{\mathbf{y}_j\}_{j=1}^{N_y}$ – database training set
$\mathcal{A}_{xy} = \cup_{i=1}^{N_1} \cup_{j=1}^{N_2} \{\mathbf{x}_i^*, \mathbf{y}_j^*, s_{ij}\}$ – auxiliary heterogeneous data
$\beta$, $\gamma_q$, $\gamma_p$, $\gamma_C$ – regularization parameters
$\delta$ – partition balance parameter
$a$, $\lambda$ – SCISD function parameter
$k_q$; $k_p$ – length of hash codes
**Ensure:**
$\mathbf{W}^q$, $\mathbf{W}^p$, $\mathbf{C}$
Initialize $\mathbf{W}^q$, $\mathbf{W}^p$ with CVH and $\mathbf{C} = \mathbf{I}$;
**while** $\mathbf{W}^q$, $\mathbf{W}^p$ and $\mathbf{C}$ are not converged **do**
  Fix $\mathbf{W}^p$ and $\mathbf{C}$, optimize $\mathbf{W}^q$:
  **for** $k = 1 \cdots k_q$ **do**
    **for** $t = 1 \cdots t_{max}$ **do**
      $\mathbf{w}_k^{q(t+1)} = \arg\min \mathcal{J}_{\mathbf{w}_k^q}^{(t)}$;
    **end for**
  **end for**
  Fix $\mathbf{W}^q$ and $\mathbf{C}$, optimize $\mathbf{W}^p$
  **for** $l = 1 \cdots k_p$ **do**
    **for** $t = 1 \cdots t_{max}$ **do**
      $\mathbf{w}_l^{p(t+1)} = \arg\min \mathcal{J}_{\mathbf{w}_l^p}^{(t)}$;
    **end for**
  **end for**
  Fix $\mathbf{W}^q$ and $\mathbf{W}^p$, optimize $\mathbf{C}$:
  **for** $t = 1 \cdots t_{max}$ **do**
    solve $\mathbf{C}^{(t+1)} = \arg\min \mathcal{J}_{\mathbf{C}}^{(t)}$;
  **end for**
**end while**

from $l_1$ random homogeneous data points and $l_2$ random heterogeneous pairs:

$$\frac{\partial \mathcal{J}_{\mathbf{w}_k^q}^{(t)}}{\mathbf{w}_k^q} = \frac{1}{N_x}\sum_{i=1}^{N_x}\left[\frac{\partial(f_1(\mathbf{w}_k^q))}{\mathbf{w}_k^q} - sgn\left(\left(\mathbf{w}_k^{q(t)}\right)^T\mathbf{x}_i\right)\mathbf{x}_i\right] + \eta_{\mathbf{w}_k^q} + \gamma_q\mathbf{w}_k^q + 2\beta\sum_{i,j}^{N_{xy}} s_{ij}d_{ij}\frac{\partial d_{ij}}{\mathbf{w}_k^q} + \beta\sum_{i,j}^{N_{xy}}(1-s_{ij})\left(\frac{\partial\tau_1(d_{ij})}{\mathbf{w}_k^q} - d_{ij}^{(t)}\frac{\partial d_{ij}^{(t)}}{\mathbf{w}_k^q}\right), \tag{13}$$

where

$$\frac{\partial(f_1(\mathbf{w}_k^q))}{\mathbf{w}_k^q} = \begin{cases} 0 & \text{if} \quad |(\mathbf{w}_k^q)^T\mathbf{x}_i| \le 1 \\ sgn((\mathbf{w}_k^q)^T\mathbf{x}_i)\mathbf{x}_i & \text{otherwise,} \end{cases} \tag{14}$$

$$\eta_{\mathbf{w}_k^q} = \begin{cases} 0 & \text{if} \quad \frac{1}{N_x}|(\sum_{i=1}^{N_x}\mathbf{w}_k^q)^T\mathbf{x}_i| \le \delta \\ sgn(\frac{1}{N_x}\sum_{i=1}^{N_x}(\mathbf{w}_k^q)^T\mathbf{x}_i - \delta)\cdot\frac{1}{N_x}\sum_{i=1}^{N_x}\mathbf{x}_i & \text{otherwise,} \end{cases} \tag{15}$$

$$\frac{\partial d_{ij}}{\mathbf{w}_k^q} = \sum_{l=1}^{k_p} \left\{ 2 \cdot \left[ \sum_{k=1}^{k_q} C_{kl} \left(\mathbf{w}_k^q\right)^T \mathbf{x}_i^* - \left(\mathbf{w}_l^p\right)^T \mathbf{y}_j^* \right] \cdot C_{kl} \mathbf{x}_i^* \right\}, \tag{16}$$

$$\frac{\partial d_{ij}^{(t)}}{\mathbf{w}_k^q} = \sum_{l=1}^{k_p} \left\{ 2 \cdot \left[ \sum_{k=1}^{k_q} C_{kl} \left(\mathbf{w}_k^{q(t)}\right)^T \mathbf{x}_i^* - \left(\mathbf{w}_l^p\right)^T \mathbf{y}_j^* \right] \cdot C_{kl} \mathbf{x}_i^* \right\}, \tag{17}$$

$$\frac{\partial \tau_1(d_{ij})}{\mathbf{w}_k^q} = \frac{\partial d_{ij}}{\mathbf{w}_k^q} \cdot \begin{cases} 0 & \text{if} \quad 0 \le |d_{ij}| \le \lambda \\ \frac{a d_{ij} - a\lambda sgn(d_{ij})}{a-1} & \text{if} \quad \lambda < |d_{ij}| \le a\lambda. \\ d_{ij} & \text{if} \quad |d_{ij}| > a\lambda \end{cases} \tag{18}$$

Similarly, the objective function and sub-gradient $w.r.t.$ $\mathbf{w}_l^p$ at the $t$th CCCP iteration are as follows:

$$\begin{aligned} \mathcal{J}_{\mathbf{w}_l^p}^{(t)} = & \frac{1}{N_y} \sum_{j=1}^{N_y} \left[ f_1\left(\mathbf{w}_l^p\right) - \mathcal{R}\left(g_1\left(\mathbf{w}_l^{p(t)}\right)\right) \right] + \left[ \frac{1}{N_y} \left| \sum_{j=1}^{N_y} \left(\mathbf{w}_l^p\right)^T \mathbf{y}_j \right| - \delta \right]_+ \\ & + \beta \sum_{i,j}^{N_{xy}} s_{ij} d_{ij}^2 + \beta \sum_{i,j}^{N_{xy}} (1 - s_{ij}) \left[ \tau_1(d_{ij}) - \mathcal{R}\left(\tau_2\left(d_{ij}^{(t)}\right)\right) \right] + \frac{\gamma_p}{2} \left\| \mathbf{w}_l^p \right\|^2, \end{aligned} \tag{19}$$

and

$$\begin{aligned} \frac{\partial \mathcal{J}_{\mathbf{w}_l^p}^{(t)}}{\mathbf{w}_l^p} = & \frac{1}{N_y} \sum_{j=1}^{N_y} \left[ \frac{\partial \left( f_1(\mathbf{w}_l^p) \right)}{\mathbf{w}_l^p} - sgn\left( \left(\mathbf{w}_l^{p(t)}\right)^T \mathbf{y}_j \right) \mathbf{y}_j \right] + \eta_{\mathbf{w}_l^p} + \gamma_p \mathbf{w}_l^p \\ & + 2\beta \sum_{i,j}^{N_{xy}} s_{ij} d_{ij} \frac{\partial d_{ij}}{\mathbf{w}_l^p} + \beta \sum_{i,j}^{N_{xy}} (1 - s_{ij}) \left( \frac{\partial \tau_1(d_{ij})}{\mathbf{w}_l^p} - d_{ij}^{(t)} \frac{\partial d_{ij}^{(t)}}{\mathbf{w}_l^p} \right), \end{aligned} \tag{20}$$

where

$$\frac{\partial d_{ij}}{\mathbf{w}_l^p} = 2 \cdot \left[ \sum_{k=1}^{k_q} C_{kl} \left(\mathbf{w}_k^q\right)^T \mathbf{x}_i^* - \left(\mathbf{w}_l^p\right)^T \mathbf{y}_j^* \right] \cdot \mathbf{y}_j^*, \tag{21}$$

$$\frac{\partial d_{ij}^{(t)}}{\mathbf{w}_l^p} = 2 \cdot \left[ \sum_{k=1}^{k_q} C_{kl} \left(\mathbf{w}_k^q\right)^T \mathbf{x}_i^* - \left(\mathbf{w}_l^{p(t)}\right)^T \mathbf{y}_j^* \right] \cdot \mathbf{y}_j^*, \tag{22}$$

$$\frac{\partial \tau_1(d_{ij})}{\mathbf{w}_l^p} = \frac{\partial d_{ij}}{\mathbf{w}_l^p} \cdot \begin{cases} 0 & \text{if} \quad 0 \le |d_{ij}| \le \lambda \\ \frac{a d_{ij} - a\lambda sgn(d_{ij})}{a-1} & \text{if} \quad \lambda < |d_{ij}| \le a\lambda. \\ d_{ij} & \text{if} \quad |d_{ij}| > a\lambda \end{cases} \tag{23}$$

To update the translator $\mathbf{C}$, we also use CCCP. The objective function and sub-gradient $w.r.t.$ every element $C_{kl}$ in the $t$th CCCP iteration

$$\mathcal{J}_{C_{kl}}^{(t)} = \beta \sum_{i,j}^{N_{xy}} s_{ij} d_{ij}^2 + \beta \sum_{i,j}^{N_{xy}} (1 - s_{ij}) \left[ \tau_1(d_{ij}) - \mathcal{R}\left(\tau_2(d_{ij}^{(t)})\right) \right] + \frac{\gamma_C}{2} \|\mathbf{C}\|_F^2, \tag{24}$$

and

$$\frac{\partial \mathcal{J}_{C_{kl}}^{(t)}}{C_{kl}} = \beta \sum_{i,j}^{N_{xy}} (1 - s_{ij}) \left( \frac{\partial \tau_1(d_{ij})}{C_{kl}} - d_{ij}^{(t)} \frac{\partial d_{ij}^{(t)}}{C_{kl}} \right) + \gamma_C C_{kl} + 2\beta \sum_{i,j}^{N_{xy}} s_{ij} d_{ij} \frac{\partial d_{ij}}{C_{kl}}, \tag{25}$$

where

$$\frac{\partial d_{ij}}{C_{kl}} = 2 \cdot \left[ \sum_{k=1}^{k_q} C_{kl} \left(\mathbf{w}_k^q\right)^T \mathbf{x}_i^* - \left(\mathbf{w}_l^p\right)^T \mathbf{y}_j^* \right] \cdot \left(\mathbf{w}_k^q\right)^T \mathbf{x}_i^*. \tag{26}$$

Noting that $\frac{\partial d_{ij}^{(t)}}{C_{kl}}$ and $\frac{\partial \tau_1(d_{ij})}{C_{kl}}$ can be obtained in the same way as in updating $\mathbf{w}_q^k$ and $\mathbf{w}_p^l$, we omit them here. The overall procedure of the HTH method, alternating learning $\mathbf{W}^q$, $\mathbf{W}^p$, and the translator $\mathbf{C}$ with CCCP and Pegasos, is presented in Algorithm 1.

### 3.5. Convergence Analysis

In this section, we provide a comprehensive convergence analysis of Algorithm 1. We analyze its convergence from the outermost loop through to the innermost one. The outermost loop corresponds to alternate between $\mathbf{W}^q$, $\mathbf{W}^p$, and $\mathbf{C}$. Such alternating optimization approach is widely adopted, and shown to be locally, q-linearly convergent, and to also exhibit a type of global convergence in Bezdek and Hathaway [2003]. Within each loop of alternating, Algorithm 1 performs CCCP iterations, each of which solves $\mathbf{w}_k^{q(t+1)} = \arg\min \mathcal{J}_{\mathbf{w}_k^q}^{(t)}$ in (10), $\mathbf{w}_l^{p(t+1)} = \arg\min \mathcal{J}_{\mathbf{w}_l^p}^{(t)}$ in (19), and $\mathbf{C}^{(t+1)} = \arg\min \mathcal{J}_{\mathbf{C}}^{(t)}$ in (24). Although the work of Lanckriet and Sriperumbudur [2009] gives a complete convergence analysis on the CCCP, there exist some assumptions to draw the final conclusion. Here, we follow the idea in Lanckriet and Sriperumbudur [2009], but specify the proof in our problem and therefore eliminate all the assumptions.

As in Lanckriet and Sriperumbudur [2009], we analyze the CCCP's convergence using Zangwill's global convergence theorem [Zangwill 1969]. Note that the global convergence here does not actually mean "globally convergent", but indicates that the algorithm must converge to a local optimum or a stationary point and not exhibit nonlinear behaviours such as divergence and oscillation.

THEOREM 3.1 ([ZANGWILL 1969]). *Let $\mathcal{A} : X \to \mathcal{P}(X)$ be a point-to-set map (an algorithm) that given a point $x_0 \in X$ generates a sequence $\{x_k\}_{k=0}^{\infty}$ through the iteration $x_{k+1} \in A(x_k)$. Also, let a solution set $\Gamma \subset X$ be given. Suppose*

*(1) All points $x_k$ are in a compact set $S \subset X$.*
*(2) There is a continuous function $\phi : X \to R$ such that:*
  *(a) $x \notin \Gamma \Rightarrow \phi(y) < \phi(x), \forall y \in \mathcal{A}(x)$,*
  *(b) $x \in \Gamma \Rightarrow \phi(y) \leq \phi(x), \forall y \in \mathcal{A}(x)$.*
*(3) $\mathcal{A}$ is closed at $x$ if $x \notin \Gamma$.*

*Then, the limit of any convergent sequence of $\{x_k\}_{k=0}^{\infty}$ is in $\Gamma$. Furthermore, $\lim_{k\to\infty} \phi(x_k) = \phi(x_*)$ for all limit points $x_*$.*

PROOF. We start our proof from optimizing $\mathbf{w}_k^q$ with CCCP. The proof applies to $\mathbf{w}_l^p$ and $\mathbf{C}$. Let $\mathcal{A}_{\text{cccp}}$ be the point-to-set map, $u(\cdot)$, $v(\cdot)$ be two convex functions decomposed to meet CCCP (i.e., $\mathcal{J}_{\mathbf{w}_k^q} = u - v$ in (9)), such that $\mathbf{w}_k^{q(t+1)} \in \mathcal{A}_{\text{cccp}}(\mathbf{w}_k^{q(t)})$, and

$$\mathcal{A}_{\text{cccp}}\left(\mathbf{w}_k^{q(t)}\right) = \arg\min \left\{ u\left(\mathbf{w}_k^q\right) - \left(\mathbf{w}_k^q\right)^T \bigtriangledown v\left(\mathbf{w}_k^{q(t)}\right) \right\}, \tag{27}$$

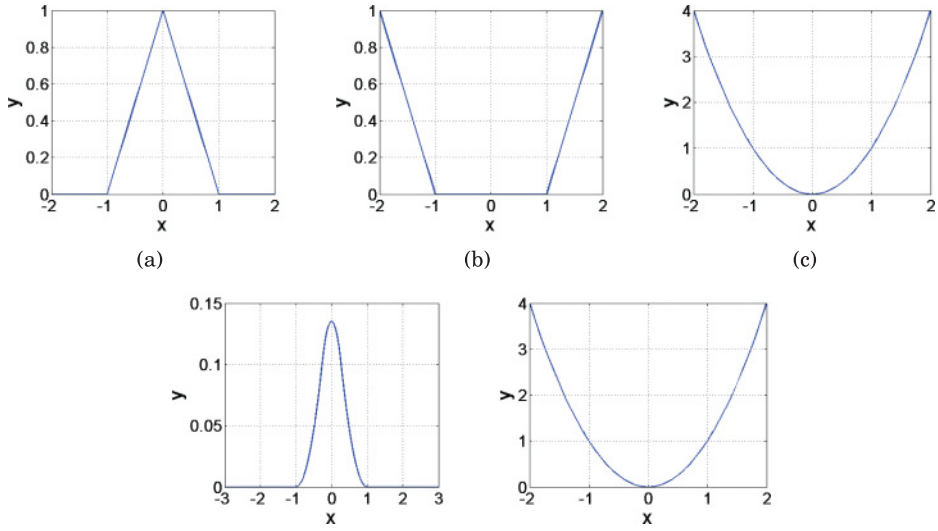



Fig. 3. Functions decomposed in Equation (9).

where

$$u(\mathbf{w}_k^q) = \frac{1}{N_x}\sum_{i=1}^{N_x} f_1(\mathbf{w}_k^q) + \left[\frac{1}{N_x}\left|\sum_{i=1}^{N_x}(\mathbf{w}_k^q)^T\mathbf{x}_i - \delta\right|\right]_+ + \beta\sum_{i=1}^{N_{xy}} s_{ij}d_{ij}^2 + \beta\sum_{i=1}^{N_{xy}}(1-s_{ij})\tau_1(d_{ij}) + \frac{\gamma_q}{2}\left\|\mathbf{w}_k^q\right\|^2, \tag{28}$$

and

$$(\mathbf{w}_k^q)^T \bigtriangledown v(\mathbf{w}_k^{q(t)}) = (\mathbf{w}_k^q)^T\left[\frac{1}{N_x}\sum_{i=1}^{N_x} sgn\left((\mathbf{w}_k^{q(t)})^T\mathbf{x}_i\right)\cdot\mathbf{x}_i + \beta\sum_{ij}^{N_{xy}}(1-s_{ij})d_{ij}^{(t)}\frac{\partial d_{ij}^{(t)}}{\mathbf{w}_k^q}\right]. \tag{29}$$

Equations (27), (28), and (29) are easily deduced from (10).

First of all, we prove that (1) in Theorem 3.1 holds in our optimization problem. Obviously, the fact that for every $\mathbf{w}_k^{q(t)} \in \mathbb{R}$, the set $H(\mathbf{w}_k^{q(t)}) = \{\mathbf{w}_k^{q(t+1)} : u(\mathbf{w}_k^{q(t+1)}) - v(\mathbf{w}_k^{q(t+1)}) \leq u(\mathbf{w}_k^{q(t)}) - v(\mathbf{w}_k^{q(t)}), \mathbf{w}_k^{q(t+1)} \in \mathcal{A}_{\text{cccp}}(\mathbb{R})\}$ is bounded is sufficient to prove that all points $\mathbf{w}_k^{q(t)}$ are bounded. On the one hand, given an initial vector $\mathbf{w}_{k(0)}^q$, $H$ is upper bounded according to the monotonically decreasing constraint. On the other hand, we observe that $\mathcal{J}_{\mathbf{w}_k^q} = u - v$ in (9) can be decomposed into the summation of five parts, each of which is plotted in Figure 3. Obviously all of them are lower bounded by zero. Therefore, we prove that H is bounded, followed by the fact that all $\mathbf{w}_k^{q(t)}$ are in a bounded set. Besides, all $\mathbf{w}_k^{q(t)}$ are discrete, and thereby constitute a closed set. In this case, we prove that (1) in Theorem 3.1 is valid, i.e., all points $\mathbf{w}_k^{q(t)}$ are in a compact set.

Secondly, we prove (2) in Theorem 3.1. Here, we define $\phi(\mathbf{w}_k^q) = \mathcal{J}_{\mathbf{w}_k^q} = u(\mathbf{w}_k^q) - v(\mathbf{w}_k^q)$. We introduce majorization-minimization algorithms first. The general idea of

majorization-minimization algorithms is to construct such a majorization function $g$

$$\begin{cases} f(x) \le g(x, y) < 0, & \forall x, y \in \Omega \\ f(x) = g(x, x), & \forall x \in \Omega \end{cases} \tag{30}$$

as the upper bound of the objective function $f$ to minimize, and minimize $g$ instead. CCCP clearly falls into the category of majorization-minimization algorithms. In our problem, the original function $f$ to minimize corresponds to $\mathcal{J}_{\mathbf{w}_k^q}$ in (9). The majorization function $g$[1] is $\mathcal{J}^{(t)}_{\mathbf{w}_k^q}$ and we minimize it instead at iteration $t$ by $\mathbf{w}_k^{q(t+1)} = \arg\min g(\mathbf{w}_k^q, \mathbf{w}_k^{q(t)})$. Consequently, it is easy to show that the value of $f$ is monotonically decreasing in each iteration:

$$f\big(\mathbf{w}_k^{q(t+1)}\big) \le g\big(\mathbf{w}_k^{q(t+1)}, \mathbf{w}_k^{q(t)}\big) \le g\big(\mathbf{w}_k^{q(t)}, \mathbf{w}_k^{q(t)}\big) = f\big(\mathbf{w}_k^{q(t)}\big). \tag{31}$$

Because $u$ and $v$ are strictly convex, the "equality" holds if and only if $g(\mathbf{w}_k^{q(t+1)}, \mathbf{w}_k^{q(t)}) = g(\mathbf{w}_k^{q(t)}, \mathbf{w}_k^{q(t)})$, i.e., $\mathbf{w}_k^{q(t)} = \arg\min g(\mathbf{w}_k^q, \mathbf{w}_k^{q(t)})$ and $\mathbf{w}_k^{q(t)} \in \Gamma$. Therefore, given $\phi(\mathbf{w}_k^q) = f(\mathbf{w}_k^q) = \mathcal{J}_{\mathbf{w}_k^q}$ and the above inequality, we conclude that (2) in Theorem 3.1 holds.

Before proceeding to prove (3) in Theorem 3.1, we first show the following Lemma 3.2.

LEMMA 3.2 ([GUNAWARDANA AND BYRNE 2005]). *Given a real-valued continuous function $h$ on $X \times Y$, define the point-to-set map $\Psi : X \to \mathcal{P}(Y)$ by*

$$\Psi(x) = \arg\min_{y' \in Y} h(x, y') = \{y : h(x, y) \le h(x, y'), \forall y' \in Y\}. \tag{32}$$

*Then, $\Psi$ is closed at $x$ if $\Psi(x)$ is non-empty.*

In our problem, $\Psi$ is $\mathcal{A}_{\text{cccp}}$ and $h$ is $g$ which is clearly continuous. Then,

$$\mathcal{A}_{\text{cccp}}(\mathbf{w}_k^q) = \arg\min g(\mathbf{w}_k^q, \mathbf{w}_k^{q(t)}) \ge \arg\min f(\mathbf{w}_k^q, \mathbf{w}_k^{q(t)}) \ge 0, \tag{33}$$

where the first inequality follows from the fact that $g$ is the upper bound of $f$ while the second inequality follows from the fact that $f$ is lower bounded by zero as we proved earlier. Besides, g is continuous. Therefore, $\mathcal{A}_{\text{cccp}}(\mathbf{w}_k^q)$ is non-empty. By Lemma 3.2, the non-emptiness of $\mathcal{A}_{\text{cccp}}(\mathbf{w}_k^q)$ ensures that $\mathcal{A}_{\text{cccp}}(\mathbf{w}_k^q)$ is closed at $\mathbf{w}_k^q$ and so satisfies (3) in Theorem 3.1.

Conditions (1), (2), and (3) are all satisfied, and we conclude from Theorem 3.1 that the limit of any convergent subsequence of $\{\mathbf{w}_k^q\}_{k=0}^{\infty}$ is in $\Gamma$. Furthermore, $\lim_{t\to\infty} f(\mathbf{w}_k^{q(t)}) = f(\mathbf{w}_k^{q(*)})$. □

However, the convergence of $f(\mathbf{w}_k^{q(t)})$ to $\mathbf{w}_k^{q(*)}$ does not necessarily imply that $\mathbf{w}_k^{q(t)}$ converges to $\mathbf{w}_k^{q(*)}$. To complete the proof, we introduce the other Theorem 3.3.

THEOREM 3.3 ([MEYER 1976]). *Let $\mathcal{A} : X \to \mathcal{P}(X)$ be a point-to-set map such that $\mathcal{A}$ is uniformly compact, closed, and strictly monotone on X, where X is a closed subset of $\mathbb{R}^n$. If $\{x_k\}_{k=0}^{\infty}$ is any sequence generated by $\mathcal{A}$, then all limit points will be fixed points of $\mathcal{A}$, $\phi(x_k) \to \phi(x_*) =: \phi^*$ as $k \to \infty$, where $x_*$ is a fixed point, $\|x_{k+1} - x_k\| \to 0$, and either $\{x_k\}_{k=0}^{\infty}$ converges or the set of limit points of $\{x_k\}_{k=0}^{\infty}$ is connected. Define $\mathcal{F}(a) := \{x \in \mathcal{F} : \phi(x) = a\}$, where $\mathcal{F}$ is the set of fixed points of $\mathcal{A}$. If $\mathcal{F}(\phi^*)$ is finite, then any sequence $\{x_k\}_{k=0}^{\infty}$ generated by $\mathcal{A}$ converges to some $x_*$ in $\mathcal{F}(\phi^*)$.*

[1]It is worth noting that optimizing $g$ is equivalent to optimizing $\mathcal{A}_{\text{cccp}}$ we propose before. $\mathcal{A}_{\text{cccp}}$ ignores constant terms irrelevant to the optimizing variable.

Because $u$ and $v$ are strictly convex, then a strict descent can be achieved in (31) i.e., $f(\mathbf{w}_k^{q(t+1)}) < g(\mathbf{w}_k^{q(t+1)}, \mathbf{w}_k^{q(t)}) < g(\mathbf{w}_k^{q(t)}, \mathbf{w}_k^{q(t)}) = f(\mathbf{w}_k^{q(t)})$. Therefore, $\mathcal{A}_{\text{cccp}}$ is strictly monotonically decreasing with respect to $f$ and so the theorem holds.

The innermost loop is the stochastic sub-gradient descent solver, Pegasos, to iteratively evaluate (13) and update $\mathbf{w}_k^q$ (similar to $\mathbf{w}_l^p$ and $C_{kl}$). In Shalev-Shwartz et al. [2007], the authors provide a detailed proof towards the convergence of Pegasos.

### 3.6. Complexity Analysis

The computational cost of the proposed algorithm comprises three parts: updating $\mathbf{W}^q$, $\mathbf{W}^p$, and $\mathbf{C}$. Hence, the total time complexity of training HTH is $O(k_q d_q(l_1 + l_3) + k_p d_p(l_2 + l_3) + l_3 k_p k_q)$, where $l_1$ and $l_2$ are the numbers of stochastically selected training data points in the query domain and database domain by the Pegasos solver. $l_3$ is the number of randomly sampled auxiliary data pairs from $N_{xy}$ auxiliary heterogeneous co-occurrence data pairs. Clearly, the time complexity for our algorithm scales linearly with the number of training data points and quadratic with the length of hash codes. In practice, the code length is short, otherwise, the technique of "hashing" will be meaningless. Hence, our algorithm is very computationally efficient.

During the online query phase, given a query instance $\tilde{\mathbf{x}}_i \in \tilde{\mathbf{X}}^q$, we apply our learned hash functions for the query domain to it by performing two dot-product operations, $\mathbf{H}_i^q = \mathbf{x}_i \cdot \mathbf{W}^q$ and translation $\mathbf{H}_i^q \cdot \mathbf{C}$, which are quite efficient. The translated query hash codes are then compared with the hash codes of the database by quick XOR and bit count operations. These operations enjoy the sub-linear time-complexity $w.r.t.$ the database size.

## 4. EXPERIMENTS

In this section, we evaluate the performance of HTH on two real-world datasets and compare it with the state-of-the-art multi-modal hashing algorithms.

### 4.1. Experimental Settings

*4.1.1. Datasets.* In this work, we use two real-world datasets, NUS-WIDE,[2] and MIRFLICKR-Yahoo Answers.

*NUS-WIDE* is a Flickr dataset containing 269,648 tagged images [Chua et al. 2009]. The annotation for 81 semantic concepts is provided for evaluation. We prune this dataset via keeping the image-tag pairs that belong to the ten largest concepts. For image features, 500 dimensional SIFT vectors are used. On the other side, a group of tags for an image composes a single text document. For each text document, we use the probability distribution of 100 Latent Dirichlet Allocation (LDA) [Blei et al. 2003] topics as the feature vector. Therefore, NUS-WIDE is a multi-view dataset. Each data instance has an image view and a text view. When searching images using text query or searching text documents using image query, the ground truth is derived by checking whether an image and a text document share at least one of the ten selected largest concepts.

*MIRFLICKR-Yahoo Answers* is a heterogeneous media dataset consisting of images from MIRFLICKR-25000 [Huiskes and Lew 2008] and QAs from Yahoo Answers. MIRFLICKR-25000 is another Flickr collection consisting of 25,000 images. We utilize 5,018 tags provided by NUS-WIDE to filter irrelevant pictures in MIRFLICKR-25000 by cross-checking tags of each image with these 5,018 tags. The 500 dimensional SIFT feature vector is also applied. Yahoo Answers are crawled from a public API of Yahoo

[2]http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm.

Query Language (YQL)[3]. The 5,018 tags are taken as keywords to search relevant QAs on Yahoo Answers. For each keyword, we extract top 100 results returned by YQL. Finally, we obtain a pool of about 300,000 QAs, each of which is regarded as a text document in the experiment. Each QA is represented in a 100 dimensional LDA-based feature vector. For the task using image query to retrieve QAs in the database, those QAs which share at least two words with tags corresponding to the image query (images in MIRFLICKR-25000 are also tagged) are labelled as the ground truth. The ground truth for the task using QA as query to retrieve the image database is obtained similarly. More importantly, we randomly select a number of multi-view instances, e.g., 2,000, in the NUS-WIDE dataset as the "bridge". As a result, we obtain $2,000^2 = 4 \times 10^6$ auxiliary heterogeneous pairs.

*4.1.2. Baselines.* We compare our method with the following four baseline algorithms.

*Cross-modality similarity-sensitive hashing (CMSSH)* [Bronstein et al. 2010], to the best of our knowledge, is the first approach that tackles hashing across multi-modal data. CMSSH uses Adaboost to construct a group of hash functions sequentially for each modality while only preserving inter-modality similarity.

*Cross-view hashing (CVH)* [Kumar and Udupa 2011] extends SH to the multi-view case via a CCA (canonical correlation analysis) alike procedure. In our implementation, CVH learns two hash functions which can directly be applied to out-of-sample data.

*Co-regularized hashing (CRH)* [Zhen and Yeung 2012a] proposes a boosted co-regularization framework to learn two sets of hash functions for both the query and database domain.

*Relation-aware heterogeneous hashing (RaHH)* [Ou et al. 2013] adopts uneven bits for different modalities and a mapping function between them. During testing, we add no heterogeneous relationship between queries and the database in our setting. In Ou et al. [2013], however, they used the explicit relationship and attained higher accuracies.

*4.1.3. Evaluation Metric.* In this paper, Mean Average Precision (MAP), precision, and recall are adopted as our evaluation metrics of effectiveness.

MAP stands out among performance measures in virtue of its competitive stability and discrimination. To compute MAP, Average Precision (AP) of top R retrieved documents for a single query is first calculated. $AP = \frac{1}{L}\sum_{r=1}^{R} P(r)\delta(r)$, where $L$ is the number of ground truth in the R retrieved set, $P(r)$ indicates the precision of top-$r$ retrieved documents, and $\delta(r) = 1$ denotes whether the $r$th retrieved document is a true neighbour otherwise $\delta(r) = 0$. MAP is then averaged over all queries' APs. The larger the MAP score, the better the retrieval performance. In our experiments, we set $R = 50$. The precision and recall scores reported in this paper are averaged over all queries. The larger the area under the curves, the better the achieved performance.

### 4.2. Results and Discussions

*4.2.1. Results on NUS-WIDE Dataset.* We perform two kinds of tasks on the NUS-WIDE dataset: (1) retrieving text documents by using images as queries; (2) retrieving images by using text documents as queries. In either task, we randomly select $300^2 = 90,000$ image-tag pairs from the NUS-WIDE dataset to be our training pairs. For the task of retrieving texts by image queries (retrieving images by text queries), we select 2,000 images (text documents) as queries and 10,000 text documents (images) to be the database. We perform our experiment on four such randomly sampled datasets and the average MAP results for all the compared algorithms are reported in Table II. To

[3]http://developer.yahoo.com/yql/.

Table II. MAP Comparison on NUS-WIDE

| Task | Algorithm | Code Length ($k_q = k_p$) | | | |
|---|---|---|---|---|---|
| | | 8 | 16 | 24 | 32 |
| Image→Text | CVH | 0.4210 | 0.4085 | 0.4066 | 0.4112 |
| | CMSSH | 0.4447 | 0.4209 | 0.4109 | 0.4123 |
| | CRH | 0.4645 | 0.5003 | 0.5255 | 0.3207 |
| | RaHH | 0.4120 | 0.4122 | 0.4098 | 0.4182 |
| | HTH | **0.5013** | **0.5357** | **0.5362** | **0.5151** |
| Text→Image | CVH | 0.4483 | 0.4323 | 0.4296 | 0.4361 |
| | CMSSH | 0.4779 | 0.4485 | 0.4378 | 0.4373 |
| | CRH | 0.4986 | 0.5327 | **0.5774** | 0.3378 |
| | RaHH | 0.4595 | 0.4396 | 0.4351 | 0.4315 |
| | HTH | **0.5398** | **0.5688** | 0.5508 | **0.5525** |

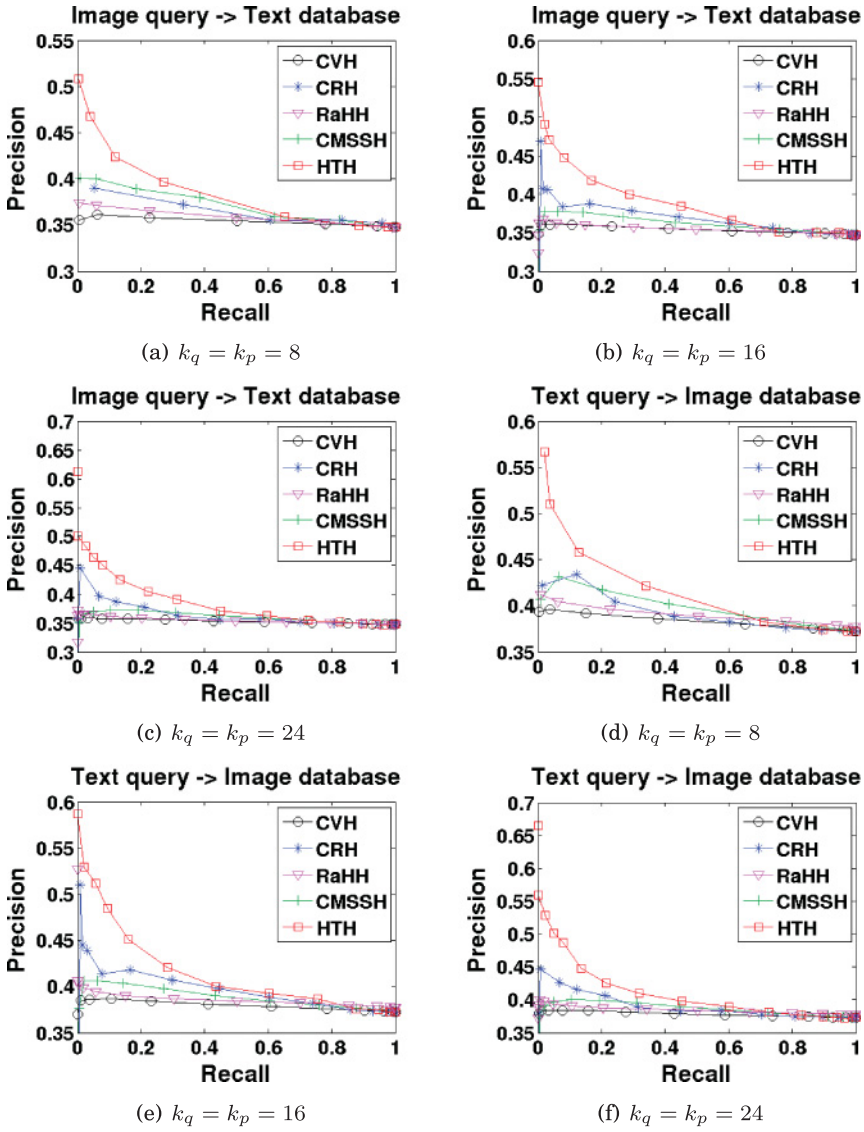
be comparable with CVH, CMSSH, and CRH, HTH adopts the same code length for different domains. From Table II, we have the following observation. HTH outperforms all state-of-the-art methods in almost all settings at most 30%. The great improvement is due to the flexibility of translation provided by our method. All the baselines simply project the data from different modalities into a shared Hamming space. However, the shared Hamming space fails to differentiate different modalities with different distributions and dimensionality. By contrast, our HTH projects each modality into an individual Hamming space and aligned them with a translator, capturing more important and dominant information in each modality with limited bits. When the lengths of codes get smaller, our HTH shows more superiority because even though the number of bits is so small that all baselines lose quite a lot of information, our method preserves the dominant information in each modality.

The precision-recall curves for 8, 16, and 24 bits are plotted in Figure 4. The superior performance of HTH in precision-recall curves agrees with the results of MAP in Table II. Figure 5 shows the recall candidate ratio curves, where the $x$-axis indicates the ratio of retrieved data points from the whole test database while the $y$-axis is simply recall. Our proposed method HTH outperforms all the baselines in all settings with different code lengths.

The time costs of HTH and other baselines are shown in Figure 6 as the code length changes. Since the training complexity of HTH is quadratic with respect to the code length, it takes more training time when the codes are longer. However, hashing with less bits is expected, thereby HTH is practical. In online querying phase, since CVH, CMSSH, and CRH have the same time complexity as HTH, we only compare HTH with RaHH. The average query search time of HTH is much less than RaHH because RaHH does not learn explicit hash functions and has to adopt the fold-in scheme for out-of-sample data. In real-time similarity search and retrieval, the improvements are meaningful and important enough.

*4.2.2. Results on MIRFLICKR-Yahoo Answers Dataset.* We also report the results of the two tasks (using images to search text documents and using text documents to search images) on the MIRFLICKR-Yahoo Answers dataset which contains a larger number of images and text documents.

In this experiment, $N_{xy} = 2,000^2 = 4 \times 10^6$ image-tags pairs from NUS-WIDE dataset, 500 randomly selected images from MIRFLICKR as well as 500 sampled QAs from Yahoo Answers are chosen for training. In this case, these image-tag pairs from NUS-WIDE serve as the auxiliary bridge while queries and the database have no direct correspondence since MIRFLICKR images and Yahoo Answers are obtained independently.



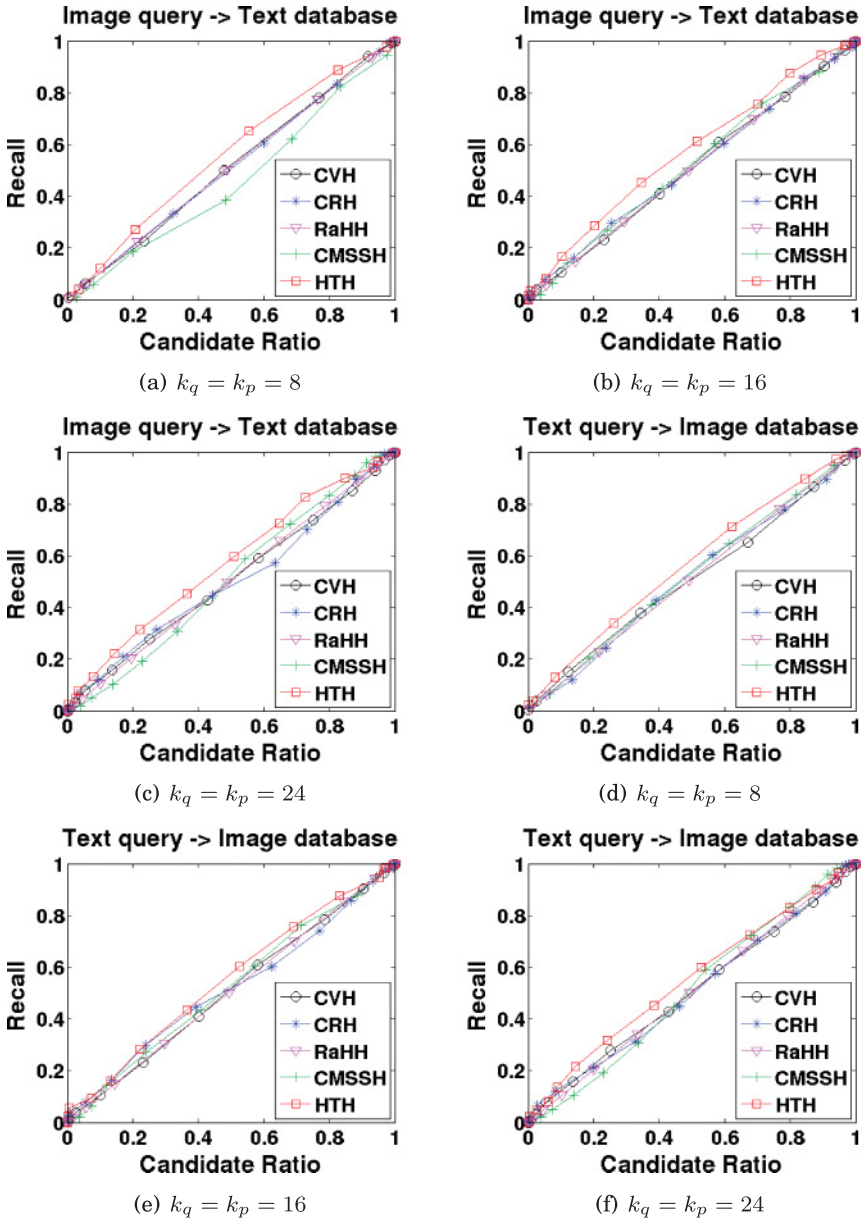

(a) $k_q = k_p = 8$ (b) $k_q = k_p = 16$

(c) $k_q = k_p = 24$ (d) $k_q = k_p = 8$

(e) $k_q = k_p = 16$ (f) $k_q = k_p = 24$

Fig. 4. Precision-recall curves on NUS-WIDE dataset.

To apply CVH, CMSSH, CRH to this dataset, we simply train hash functions for images and texts using the image-tag pairs from NUS-WIDE and generate hash codes of images from MIRFLICKR and QAs from Yahoo Answers by directly applying corresponding hash functions. For RaHH, in the training phase, the data is the same as those used in CVH, CMSSH, and CRH. In the testing phase, we do not add any relationships between queries and database entities when applying the fold-in algorithm to generate hash codes for out-of-sample data. For HTH, we train the hash functions with the auxiliary heterogeneous pairs and unlabelled homogeneous data. In the testing phase, it is easy to apply the learned hash functions to out-of-sample data without any correspondence information.




(a) $k_q = k_p = 8$ (b) $k_q = k_p = 16$

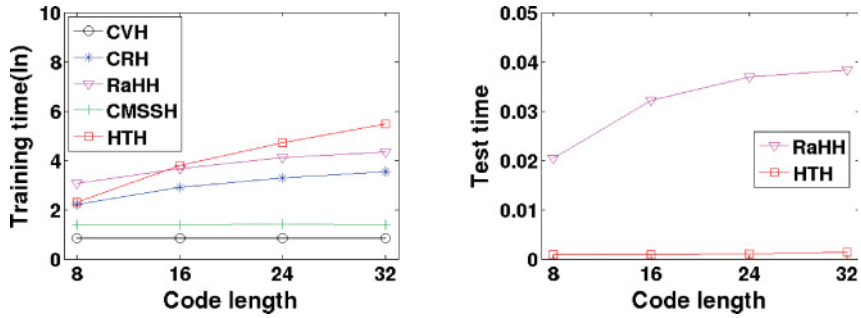
(c) $k_q = k_p = 24$ (d) $k_q = k_p = 8$

(e) $k_q = k_p = 16$ (f) $k_q = k_p = 24$

Fig. 5. Recall-candidate ratio curves on NUS-WIDE dataset.

The MAP results are summarized in Table III with various code length settings. It shows that HTH outperforms all the other algorithms under all settings. This demonstrates that HTH shows more superiority in situations where queries and the database do not inter-relate. Similarly, the precision-recall curves and the recall candidate ratio curves are plotted in Figures 7 and 8, respectively.

More importantly, our proposed HTH method adopts uneven bits for different modalities so that it discriminates between the query and database domain flexibly. In




(a) Training time (b) Test time

Fig. 6. Time cost of training and testing on NUS-WIDE dataset with different code lengths. The time is measured in seconds. The $y$-axis in (a) is the natural logarithm of training time.

Table III. MAP Comparison on MIRFLICKR-Yahoo Answers

| Task | Algorithm | Code Length0($k_q = k_p$) | | | |
|---|---|---|---|---|---|
| | | 8 | 16 | 24 | 32 |
| Image→QA | CVH | 0.1331 | 0.1443 | 0.1460 | 0.1506 |
| | CMSSH | 0.1373 | 0.1268 | 0.1210 | 0.1295 |
| | CRH | 0.0979 | 0.1419 | 0.1317 | 0.1216 |
| | RaHH | 0.1346 | 0.1437 | 0.1474 | 0.1275 |
| | HTH | **0.1577** | **0.1738** | **0.1824** | **0.1617** |
| QA→Image | CVH | 0.1515 | 0.1758 | 0.1694 | 0.1721 |
| | CMSSH | 0.1735 | 0.1483 | 0.1518 | 0.1544 |
| | CRH | 0.1048 | 0.2234 | 0.1793 | 0.1862 |
| | RaHH | 0.1669 | 0.1731 | 0.1686 | 0.1452 |
| | HTH | **0.1785** | **0.2460** | **0.2055** | **0.2324** |

Table IV, we compare HTH with RaHH, which also supports different code lengths. The row represents code length of images while the column is for that of QAs. HTH and RaHH both attain the best MAP results at $k_q = 16$ and $k_p = 24$. This code length combination is regarded as the best tradeoff between effective translator learning and original information preservation. Moreover, images require less bits to achieve comparable performance compared to QAs because instances in text domain are more dispersed so that more bits are called for encoding all the instances.

The time costs of HTH on MIRFLICKR-Yahoo Answers dataset is also reported in Figure 9. The results are similar to Figure 6 except that HTH is more efficient by contrast. This demonstrates that HTH is less sensitive to the size of the training data, which can be further proved in Figure 13 later.

*4.2.3. Parameter Sensitivity Analysis.* We conduct empirical analysis on parameter sensitivity on all datasets and study the effect of different parameter settings on the performance of HTH.

On the NUS-WIDE dataset, we examine four tradeoff parameters, $\beta$, $\gamma_q$, $\gamma_p$, and $\gamma_C$ as shown in objective function (8). We fix the code lengths of both modalities to be 16. We perform grid search on $\beta$ and $\gamma_C$ in the range of $\{10^{-6},10^{-3},10^{0},10^{3},10^{6}\}$ by fixing $\gamma_q$ and $\gamma_p$. HTH gains the best MAP at $\beta = 1,000$, $\gamma_C = 1$ as Figure 10(a) shows. When fixing the $\beta$ and $\gamma_C$, grid search of $\gamma_q$ and $\gamma_p$ in the range of $\{10^{-4},10^{-2},10^{0},10^{2},10^{4}\}$ shows that $\gamma_q = \gamma_p = 0.01$ performs the best. We adopt $\gamma_C = 1$, $\beta = 1,000$, $\gamma_q = 0.01$, $\gamma_p = 0.01$ in our experiments.

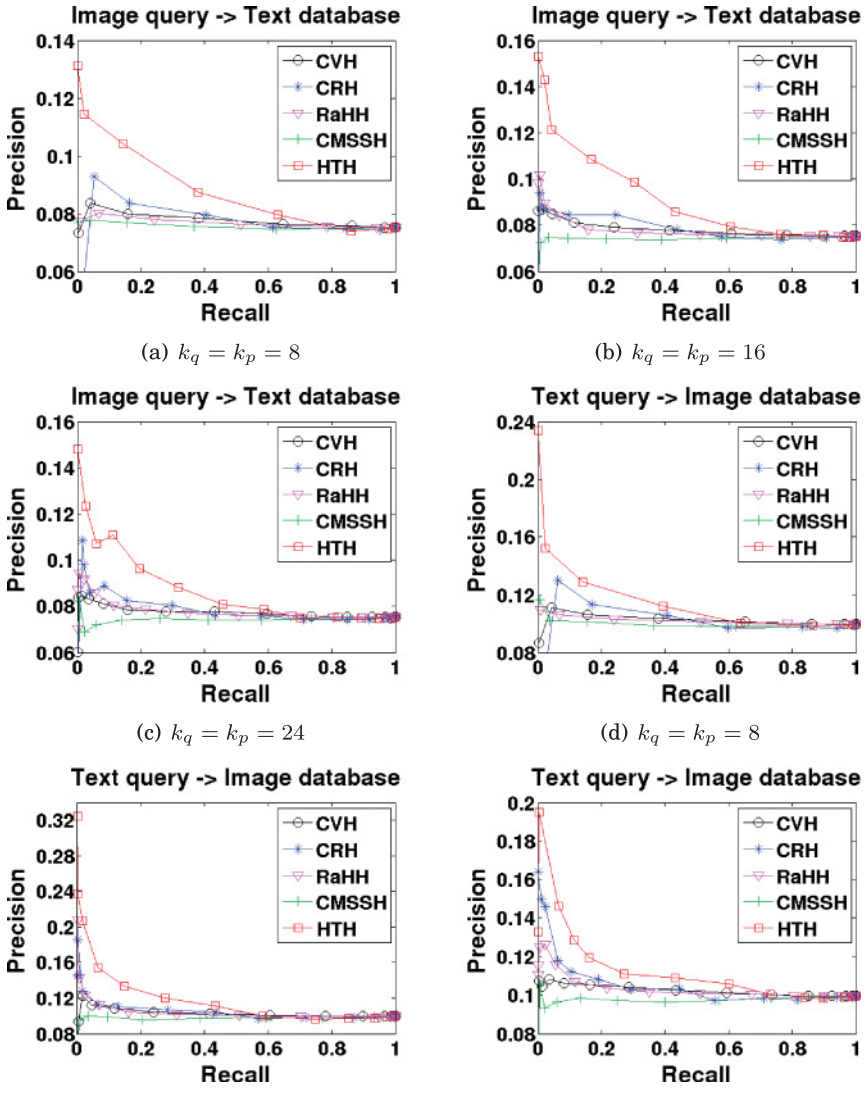
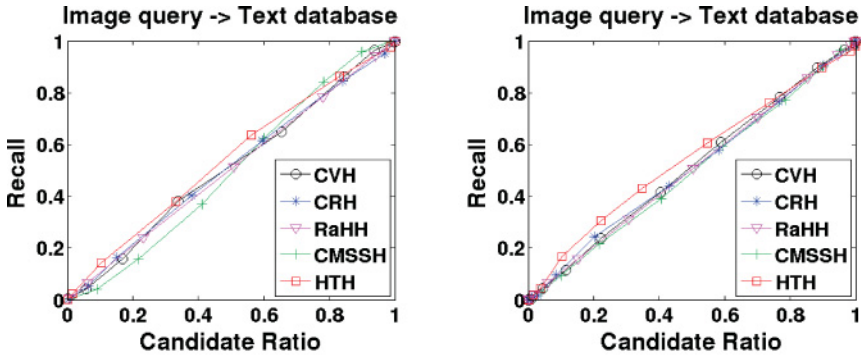



Fig. 7. Precision-recall curves on MIRFLICKR-Yahoo Answer dataset.

Table IV. MAP of RaHH and HTH on MIRFLICKR-Yahoo Answer with Different Combinational Code Length

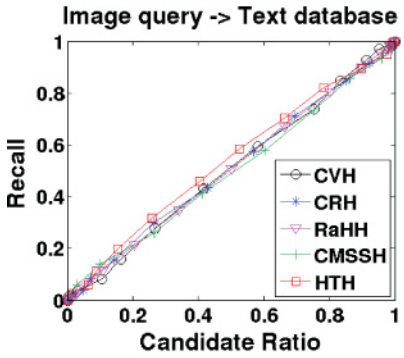
| $k_q$ \ $k_p$ | 8 | | 16 | | 24 | | 32 | |
|---|---|---|---|---|---|---|---|---|
| | RaHH | HTH | RaHH | HTH | RaHH | HTH | RaHH | HTH |
| 8 | 0.1346 | 0.1577 | 0.1315 | 0.1410 | 0.1442 | 0.1583 | 0.1366 | 0.1725 |
| 16 | 0.1346 | 0.1525 | 0.1437 | 0.1738 | **0.1545** | **0.1894** | 0.1274 | 0.1638 |
| 24 | 0.1346 | 0.1692 | 0.1437 | 0.1736 | 0.1474 | 0.1824 | 0.1378 | 0.1625 |
| 32 | 0.1346 | 0.1761 | 0.1437 | 0.1626 | 0.1474 | 0.1701 | 0.1275 | 0.1617 |



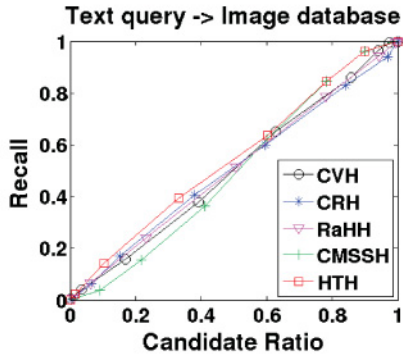

(a) $k_q = k_p = 8$



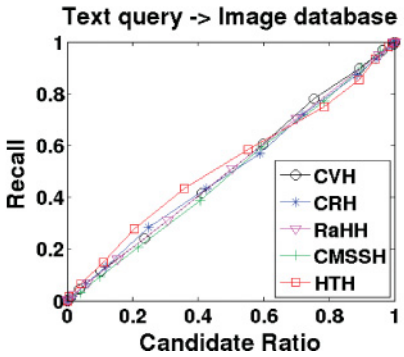

(b) $k_q = k_p = 16$



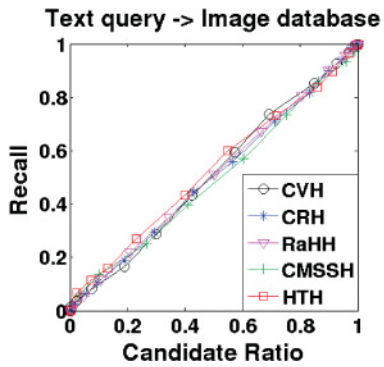

(c) $k_q = k_p = 24$




(d) $k_q = k_p = 8$



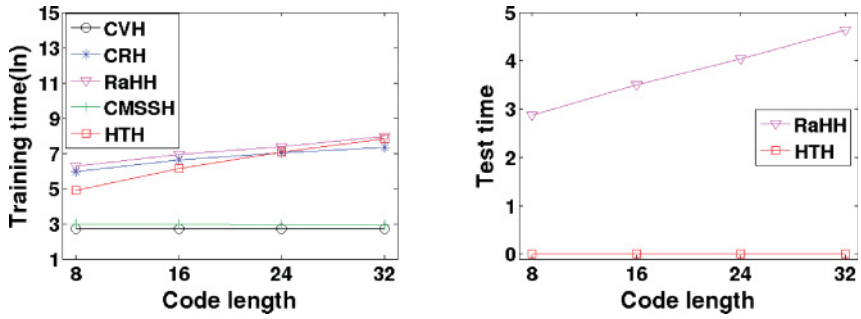

(e) $k_q = k_p = 16$




(f) $k_q = k_p = 24$

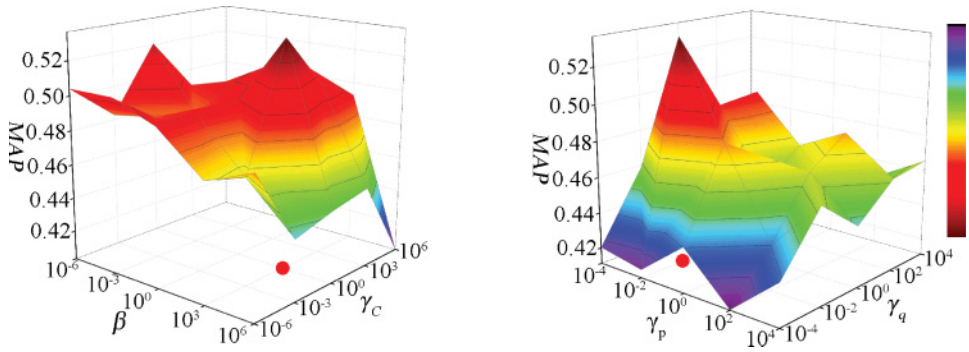
Fig. 8. Recall-candidate ratio curves on MIRFLICKR-Yahoo Answer dataset.

Especially, the parameter $\beta$ leverages the power of heterogeneous similarity preservation. When $\beta$ is too small, e.g., $\beta < 10^{-3}$, our model just ignores heterogeneous similarity preservation and only focuses on homogeneous similarity preservation. But when $\beta$ is too large, $\beta > 10^3$, the model ignores the homogeneous similarity preservation and cannot preserve the original structure of data. So, we can easily choose a proper value for $\beta$. Since $\gamma_C$ is contained in the term which $\beta$ controls, so they are coupled closely. Proper combinations of $\beta$ and $\gamma_C$ lead to superior performances of HTH.




(a) Training time (b) Test time

Fig. 9. Time cost of training and testing on MIRFLICKR-Yahoo Answers dataset with code lengths. The time is measured in seconds. The $y$-axis in (a) is the natural logarithm of training time.




(a) (b)

Fig. 10. Study of parameter sensitivity on NUS-WIDE dataset. Parameter settings in our experiments are labelled in red dots and correspond to the best performances.

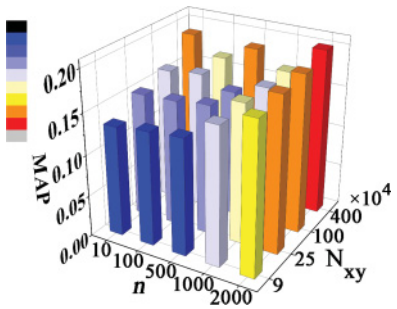
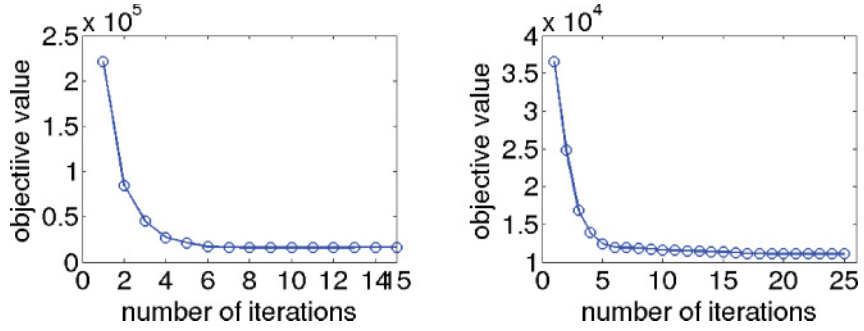



Fig. 11. The influence of varying $N_{xy}$, the number of auxiliary heterogeneous pairs, and $n$, the number of added unlabelled images/QAs, on MAP performance.

On the MIRFLICKR-Yahoo Answers dataset, we investigate the influence of $N_{xy}$, the number of auxiliary heterogeneous training pairs, and $n$, the number of added unlabelled images/QAs, on MAP performance in Figure 11. Reasonably, larger $n$ and $N_{xy}$ result in better MAP performance. In practice, we choose $N_{xy} = 4 \times 10^6$ and $n = 500$, which is competitive with $N_{xy} = 4 \times 10^6$ and $n = 2{,}000$, to be more efficient during training.

*4.2.4. Convergence Study.* Since our proposed HTH in Algorithm 1 is solved by an iterative procedure, we empirically examine its convergence performances. As Figure 12(a)




(a) $k_q = k_p = 8$ (b) $k_q = k_p = 16$

Fig. 12. The convergence of the proposed Heterogeneous Translated Hashing in Algorithm 1.

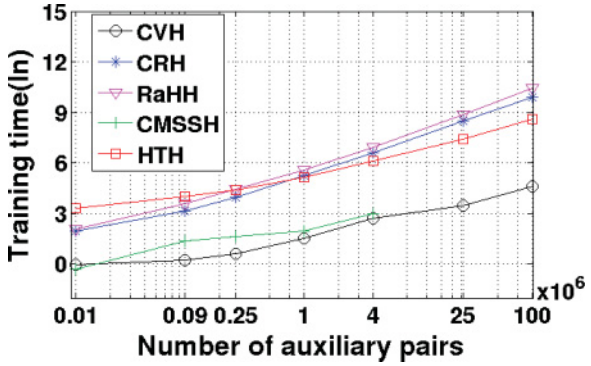



Fig. 13. Scalability of training on MIRFLICKR-Yahoo Answers dataset as the number of auxiliary heterogeneous pairs, $N_{xy}$, increases. The time is measured in seconds. The $x$-axis is in logarithmic scale. The $y$-axis is the natural logarithm of training time.

shows, after about 10 iterations the value of objective function converges. When the code lengths get longer as Figure 12(b) plots, the algorithms take more iterations, about 20 iterations, to converge.

*4.2.5. Scalability.* We check the scalability of HTH when the number of training data is increasing on the MIRFLICKR-Yahoo Answers dataset.

As Figure 13 shows, although HTH takes more training time than CRH and RaHH when the number of auxiliary heterogeneous pairs, $N_{xy}$, is small, it shows more efficiency compared with CRH and RaHH as $N_{xy}$ increases. Therefore, HTH has good scalability and can be applied to large-scale datasets. CVH and CMSSH rely on eigen decomposition operations which are efficient especially when the dimensions of the dataset are comparatively low. However, they do not consider homogeneous similarity or the regularization of parameters, thus resulting in less accurate out-of-sample testing performance. Note that we do not report results of CMSSH when $N_{xy} = 2.5 \times 10^7, 10^8$ since the algorithm has "out-of-memory" at these scales.

*4.2.6. A Case Study.* In Figure 14, we provide a similarity search example, randomly chosen from the MIRFLICKR-Yahoo Answers dataset, to visually show the effectiveness of HTH. Given an image from MIRFLICKR, we compare the relevance of top-5 nearest questions to the image by HTH and CVH. We choose CVH because it gives almost the best results in this case study among all baseline methods. The image is

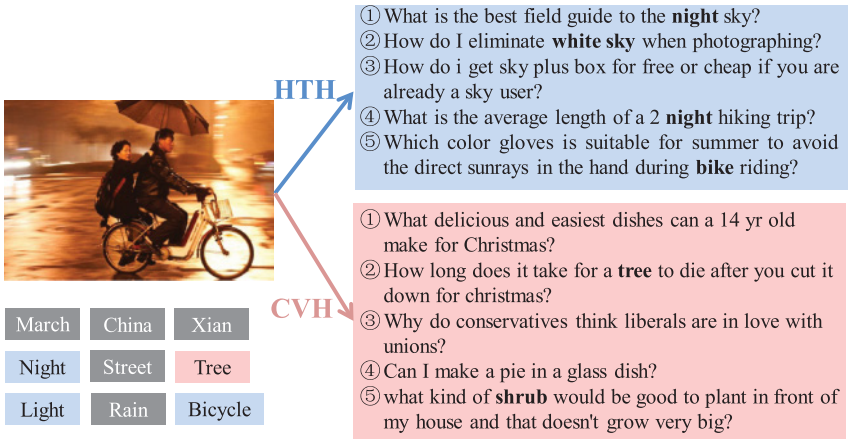



Fig. 14. Given a picture in the MIRFLICKR dataset as query, we retrieve top-5 nearest questions from the Yahoo Answers dataset by HTH and CVH. Whether there exist corresponding keywords in a retrieved question to the labels of the picture indicates the relevance of this question to the picture.

tagged as "march", "China", and so on on Flickr. Among the questions retrieved by our HTH, some of these tags, like "night", occur. And some of them, such as "bicycle", have synonyms occur. Other tags, like "light", have visually correlated keywords, "white sky", in the retrieved result list. However, the retrieved list by CVH only has one keyword, "shrub", as the hyponym of "tree". The retrieved questions via HTH are more relevant to this picture.

## 5. CONCLUSIONS

In this paper, we propose a novel HTH model to perform similarity search across heterogeneous media. In particular, by leveraging auxiliary heterogeneous relationship on the web as well as massive unlabelled instances in each modality, HTH learns a set of hash functions to project instances of each modality to an individual Hamming space and a translator aligning these Hamming spaces. Extensive experimental results demonstrate the superiority of HTH over state-of-the-art multi-modal hashing methods. In the future, we plan to apply HTH to other modalities from social media and mobile computing, and to devise a more appropriate scheme to translate between different Hamming spaces, thereby further improving HTH.

## ACKNOWLEDGMENTS

We thank the reviewers for their valuable comments to improve this paper.

## REFERENCES

A. Andoni and P. Indyk. 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*. ACM, 459–468. DOI:http://dx.doi.org/10.1109/FOCS.2006.49

Jon Louis Bentley. 1975. Multidimensional binary search trees used for associative searching. *Commun. ACM* 18, 9 (1975), 509–517.

James C. Bezdek and Richard J. Hathaway. 2003. Convergence of alternating optimization. *Neural, Parallel & Scientific Computations* 11, 4 (2003), 351–368.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3 (2003), 993–1022.

M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios. 2010. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *CVPR*. IEEE Computer Society, 3594–3601. DOI:http://dx.doi.org/10.1109/CVPR.2010.5539928

Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao. Zheng. 2009. NUS-WIDE: A real-world web image database from national university of Singapore. In *VLDB*. ACM, 48:1–48:9.

Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. 2004. Locality-sensitive hashing scheme based on P-stable distributions. In *SCG*. ACM, 253–262. DOI:http://dx.doi.org/10.1145/997817.997857

Guiguang Ding, Yuchen Guo, and Jile Zhou. 2014. Collective matrix factorization hashing for multimodal data. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*. IEEE Computer Society, 2083–2090.

Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity search in high dimensions via hashing. In *VLDB*. Morgan Kaufmann Publishers Inc., 518–529. http://dl.acm.org/citation.cfm?id=645925.671516

Yunchao Gong and S. Lazebnik. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*. IEEE Computer Society, 817–824. DOI:http://dx.doi.org/10.1109/CVPR.2011.5995432

Asela Gunawardana and William Byrne. 2005. Convergence theorems for generalized alternating minimization procedures. *J. Mach. Learn. Res.* 6 (2005), 2049–2073.

Mark J. Huiskes and Michael S. Lew. 2008. The MIR Flickr retrieval evaluation. In *MIR*. ACM, 39–43. DOI:http://dx.doi.org/10.1145/1460096.1460104

B. Kulis and K. Grauman. 2009. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*. IEEE Computer Society, 2130–2137. DOI:http://dx.doi.org/10.1109/ICCV.2009.5459466

B. Kulis, P. Jain, and K. Grauman. 2009. Fast similarity search for learned metrics. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 31, 12 (2009), 2143–2157. DOI:http://dx.doi.org/10.1109/TPAMI.2009.151

Shaishav Kumar and Raghavendra Udupa. 2011. Learning hash functions for cross-view similarity search. In *IJCAI*. AAAI Press, 1360–1365. DOI:http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-230

Gert R. Lanckriet and Bharath K. Sriperumbudur. 2009. On the convergence of the concave-convex procedure. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 1759–1767.

Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2011. Hashing with graphs. In *ICML*. ACM, 1–8.

R. R. Meyer. 1976. Sufficient conditions for the convergence of monotonic mathematicalprogramming algorithms. *J. Comput. Syst. Sci.* 12, 1 (1976), 108–121.

Yadong Mu, Jialie Shen, and Shuicheng Yan. 2010. Weakly-supervised hashing in kernel space. In *CVPR*. IEEE Computer Society, 3344–3351. DOI:http://dx.doi.org/10.1109/CVPR.2010.5540024

Mingdong Ou, Peng Cui, Fei Wang, Jun Wang, Wenwu Zhu, and Shiqiang Yang. 2013. Comparing apples to oranges: A scalable solution with heterogeneous hashing. In *KDD*. ACM, 230–238. DOI:http://dx.doi.org/10.1145/2487575.2487668

N. Quadrianto and C. Lampert. 2011. Learning multi-view neighborhood preserving projections. In *ICML*. ACM, 425–432.

Maxim Raginsky and Svetlana Lazebnik. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS*. Curran Associates, Inc., 1509–1517.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *Int. J. Approx. Reason.* 50, 7 (2009), 969–978. DOI:http://dx.doi.org/10.1016/j.ijar.2008.11.006

G. Shakhnarovich, P. Viola, and T. Darrell. 2003. Fast pose estimation with parameter-sensitive hashing. In *ICCV*. IEEE Computer Society, 750–757. DOI:http://dx.doi.org/10.1109/ICCV.2003.1238424

Shai Shalev-Shwartz, Yoram Singer, and Nathan Srebro. 2007. Pegasos: Primal estimated sub-gradient solver for SVM. In *ICML*. ACM, 807–814. DOI:http://dx.doi.org/10.1145/1273496.1273598

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. 2011. Pegasos: Primal estimated sub-gradient solver for SVM. *Math. Program.* 127, 1 (2011), 3–30. DOI:http://dx.doi.org/10.1007/s10107-010-0420-4

Behjat Siddiquie, Brandyn White, Abhishek Sharma, and Larry S. Davis. 2014. Multi-modal image retrieval for complex queries using small codes. In *ICMR*. ACM, 321.

Ajit P. Singh and Geoffrey J. Gordon. 2008. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 650–658.

Jingkuan Song, Yang Yang, Yi Yang, Zi Huang, and Heng Tao Shen. 2013. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In *SIGMOD*. ACM, 785–796. DOI:http://dx.doi.org/10.1145/2463676.2465274

C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua. 2012. LDAHash: Improved matching with smaller descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 34, 1 (2012), 66–78. DOI:http://dx.doi.org/10.1109/TPAMI.2011.103

Jeffrey K. Uhlmann. 1991. Satisfying general proximity/similarity queries with metric trees. *Inf. Process. Lett.* 40, 4 (1991), 175–179.

Jun Wang, S. Kumar, and Shih-Fu Chang. 2010. Semi-supervised hashing for scalable image retrieval. In *CVPR*. IEEE Computer Society, 3424–3431. DOI: http://dx.doi.org/10.1109/CVPR.2010.5539994

Wei Wang, Beng Chin Ooi, Xiaoyan Yang, Dongxiang Zhang, and Yueting Zhuang. 2014. Effective multi-modal retrieval based on stacked auto-encoders. *Proceedings of the VLDB Endowment* 7, 8 (2014).

Roger Weber, Hans-Jörg Schek, and Stephen Blott. 1998. A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces. In *VLDB*. Morgan Kaufmann Publishers Inc., 194–205. http://dl.acm.org/citation.cfm?id=645924.671192

Ying Wei, Yangqiu Song, Yi Zhen, Bo Liu, and Qiang Yang. 2014. Scalable heterogeneous translated hashing. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 791–800.

Yair Weiss, Antonio Torralba, and Rob Fergus. 2008. Spectral hashing. In *NIPS*. Curran Associates, Inc., 1753–1760.

F. Wu, Z. Yu, Y. Yang, S. Tang, Y. Zhang, and Y. Zhuang. 2014. Sparse multi-modal hashing. *IEEE Transactions on Multimedia,* 16, 2 (2014), 427–439. DOI: http://dx.doi.org/10.1109/TMM.2013.2291214

Alan L. Yuille, Anand Rangarajan, and A. L. Yuille. 2002. The concave-convex procedure (CCCP). *Adv. Neural Inf. Process. Syst.* 2 (2002), 1033–1040.

W. I. Zangwill. 1969. *Nonlinear Programming: A Unified Approach*. Prentice-Hall. http://books.google.com.hk/books?id=TWhxLcApH9sC.

Deming Zhai, Hong Chang, Yi Zhen, Xianming Liu, Xilin Chen, and Wen Gao. 2013. Parametric local multimodal hashing for cross-view similarity search. In *IJCAI*. AAAI Press, 2754–2760. http://dl.acm.org/citation.cfm?id=2540128.2540525

Dell Zhang, Jun Wang, Deng Cai, and Jinsong Lu. 2010. Self-taught hashing for fast similarity search. In *SIGIR*. ACM, 18–25. DOI: http://dx.doi.org/10.1145/1835449.1835455

Yi Zhen and Dit Yan Yeung. 2012a. Co-Regularized Hashing for Multimodal Data. In *NIPS*. Curran Associates, Inc., 1385–1393.

Yi Zhen and Dit Yan Yeung. 2012b. A probabilistic model for multimodal hash function learning. In *KDD*. ACM, 940–948. DOI: http://dx.doi.org/10.1145/2339530.2339678

Jile Zhou, Guiguang Ding, and Yuchen Guo. 2014. Latent semantic sparse hashing for cross-modal similarity search. In *Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 415–424.

Xiaofeng Zhu, Zi Huang, Heng Tao Shen, and Xin Zhao. 2013. Linear cross-modal hashing for efficient multimedia search. In *MM*. ACM, 143–152. DOI: http://dx.doi.org/10.1145/2502081.2502107

Received October 2014; accepted March 2015