Optimal Projection Guided Transfer Hashing for Image Retrieval

Lei Zhang, *Senior Member, IEEE*, Ji Liu, Yang Yang, Fuxiang Huang, Feiping Nie, David Zhang, *Fellow, IEEE*

***Abstract*—Recently, learning to hash has been widely studied for image retrieval thanks to the computation and storage efficiency of binary codes. Most existing learning to hash methods have yielded significant performance. However, for most existing learning to hash methods, sufficient training images are required and used to learn precise hashing codes. In some real-world applications, there are not always sufficient training images in the domain of interest. In addition, some existing supervised approaches need a amount of labeled data, which is an expensive process in terms of time, labor and human expertise. To handle such problems, inspired by transfer learning, we propose a simple yet effective unsupervised hashing method named Optimal Projection Guided Transfer Hashing (GTH) where we borrow the images of other different but related domain i.e., source domain to help learn precise hashing codes for the domain of interest i.e., target domain. In GTH, we aim to learn domain-invariant hashing functions. To achieve that, we propose to minimize the error matrix between two hashing projections of target and source domains. We seek for the maximum likelihood estimation (MLE) solution of the error matrix between the two hashing projections due to the domain gap. Furthermore, an alternating optimization method is adopted to obtain the two projections of target and source domains. By doing so, two projections can be progressively aligned. Extensive experiments on various benchmark databases for cross-domain visual recognition verify that our method outperforms many state-of-the-art learning to hash methods. The source code is available at https://github.com/liuji93/GTH**

***Index Terms*—Projection Alignment, Maximum Likelihood Estimation, Transfer Learning, Learning to Hash, Image Retrieval**

I. Introduction

With the development of Internet and multimedia, the quantity of web data has increased explosively, which comes to two important problems to be issued in visual retrieval. On one hand, how to store such a large-scale data with limited storage space is an urgent problem. On the other hand, how to efficiently match the query data embedded in a high-dimensional space is another fundamental but intractable problem in real-world retrieval tasks. In big data era, a new research field, small-data challenges with unsupervised and semi-supervised methods beyond deep neural networks, has emerged and attracted a number of researchers on this topic. Qi and Luo [1] contributed a high-rise building survey towards addressing small data challenges in big data era.

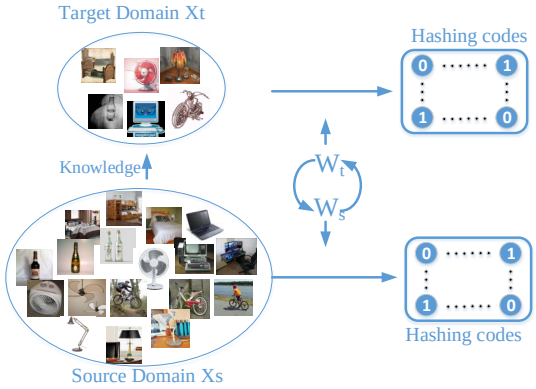


Fig. 1: Overview of our GTH. The images from the relevant but different source domain are used to help learn hashing codes for target domain where there are insufficient images that can be used to learn effective hashing codes.

In recent years, hashing algorithms have been proposed to handle the large-scale information retrieval problems in machine learning, computer vision, and big data communities [2], [3], [4], [5], [6]. The main goal of hashing techniques is to encode documents, images, and videos into a set of compact binary codes that preserve the feature similarity/dissimilarity in Hamming space. As a result, there will be less storage cost and faster computational speed by using binary features.

The early hashing methods aim to preserve similarity/dissimilarity by obtaining random projections and permutations from the raw high-dimensional Euclidean space to a low-dimensional Hamming space to generate the binary codes e.g., local sensitive hashing (LSH) [7] and min-wise hashing (Min-Hash) [8]. However, LSH and Min-Hash do not depend on the training data, and thus require longer hashing codes to guarantee better retrieval performance.

LSH and Min-Hash can be sorted as data-independent hashing methods. Recently, the data-dependent hashing, also known as *learning to hash*, has witnessed a number of studies. Generally, the data-dependent learning to hash approaches can be divided into two groups: unsupervised and supervised

This work is supported by National Natural Science Fund of China (Grant 61771079), Chongqing Natural Science Fund (No. cstc2018jcyjAX0250) and Chongqing Youth Talent Program. (*Corresponding author: Lei Zhang*)

Lei Zhang, Ji Liu and Fuxiang Huang are with the School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China (E-mail: leizhang@cqu.edu.cn, jiliu@cqu.edu.cn, huangfuxiang@cqu.edu.cn).

Yang Yang is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China. (E-mail: dlyyang@gmail.com)

Feiping Nie is with the Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, Xi'an 710072, China. (E-mail: feipingnie@gmail.com)

David Zhang is with the School of Science and Engineering, Chinese University of Hong Kong (Shenzhen), Shenzhen 518172, China. (E-mail: csdzhang@comp.polyu.edu.hk)

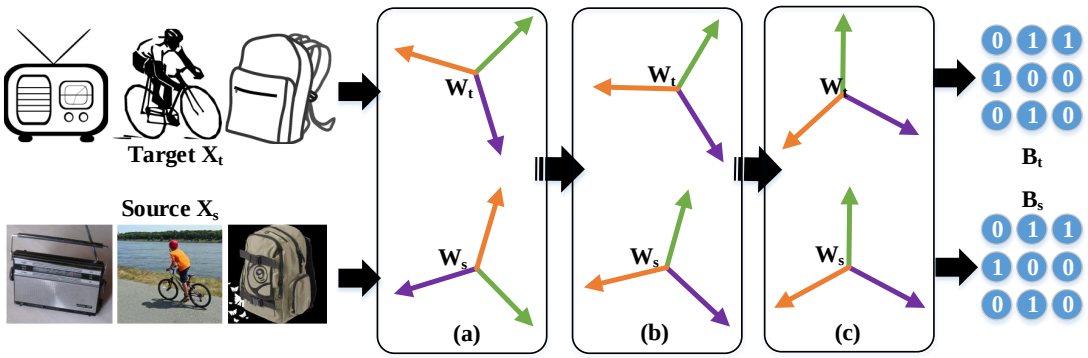


Fig. 2: Diagram of GTH. The main idea of GTH is to align hashing projections of source and target domains. The hashing projections are aligned gradually by using an alternating optimization method.

hashing methods. *Unsupervised* methods such as Spectral Hashing (SH) [9], Anchor Graph Hashing (AGH) [10], Iterative Quantization (ITQ) [5], Density Sensitive Hashing (DSH) [11], Circulant Binary Embedding (CBE) [12], Scalable Graph Hashing (SGH) [13], and Ordinal Constraint Hashing (OCH) [14] aim to explore the intrinsic structure of data to preserve the similarity of neighbors without any supervised information. *Supervised* methods such as LDA hashing [15], Minimal Loss Hashing [16], FastHash [17], Kernel-based Supervised Hashing (KSH) [18], Supervised Discrete Hashing (SDH) [19], Kernel-based Supervised Discrete Hashing (KSDH) [20], and Supervised Quantization for similarity search (SQ) [21] preserve similarity/dissimilarity of intra-class/inter-class images by using semantic information. Recently, Do et al. [22] proposed a deep learning to hashing model, Binary Deep Neural Network (BDNN), for both unsupervised and supervised hashing. The advantages of BDNN lie in the multi-layer perceptron based deep reconstruction for binary code learning and the efficient backpropagation solver for the non-smooth objective function caused by binarization.

Although most of existing learning to hash methods have achieved significant performance, they are faced with two problems for both unsupervised and supervised hashing methods. On one hand, most of existing learning to hash methods usually require a large amount of data instances to learn a set of binary hashing codes. However, in some real-world applications, for a domain of interest, i.e., the target domain, the data instances may not be sufficient enough to learn a precise hashing model. Some supervised methods need a large number of labeled images to learn hashing codes. It is well-known that labeling a number of images takes a lot of time, labor and human expertise. On the other hand, they assume that the distributions of training and testing data are similar, which may not hold in many real-world applications, due to the impact of pose, illumination, camera resolution, etc.

To handle the above problems, inspired by transfer learning, we propose a simple yet effective Optimal Projection Guided Transfer Hashing (GTH) method in this paper. Due to the distribution disparity between source and target domains, by only leveraging the source domain as auxiliary data for training the existing hashing methods, the performance may be insignificant and even worse caused by negative transfer. The claim that traditional hashing baselines trained with source data may be sometimes slightly worse than themselves trained without using source data has also been experimentally demonstrated in Section IV. Therefore, in our GTH, we propose to learn two hashing projections for target and source domains, respectively. Moreover, the knowledge from source domain can be easily used to promote target domain to learn precise hashing codes. In transfer hashing, it is important to guarantee similar images between target and source domains have similar hashing codes. In our GTH, we assume that similar images between target and source domains should mean small discrepancy between hashing projections. To this end, we let the hashing projection (functions) of target domain close to the hashing projection of source domain. The overview of our GTH is shown as Fig. 1.

It is easy to adopt minimizing l_2 or l_1 loss between the two hashing projections of source and target domains directly. In other words, in the term of maximum likelihood estimation, we actually assume that errors between two projections of source and target domains obey Gaussian or Laplacian distribution with the l_2 or l_1 loss. However, the data distributions of source and target domains are not similar due to the existence of cross pose, cross camera, and illumination variation, etc. Therefore, the distribution of errors may be far from Gaussian or Laplacian distribution. To improve the above problem, we propose the GTH model from the view of maximum likelihood estimation in this paper. Inspired by [23], we design an iteratively weighted l_2 loss for the errors between the projections of source and target domains, which makes our GTH more adaptive to cross-domain case.

Besides, an alternating optimization method is adopted to obtain the two projections of target and source domain such that the projections of source and target domains can be aligned gradually. The two different domains can share the Hamming space for each other. In other words, the target projection learning is guided by source projection and, in return, the source projection learning is guided by target projection. Finally, the optimal projections of target and source domains will be obtained. The core idea of our GTH is shown as Fig. 2. This paper is a substantial extension of our previous AAAI conference work [24]. Comparing to our previous con-

ference work, further contributions are resorted in this paper. Specifically, more insight analysis and theoretical discussion about our model are presented and more experimental results on benchmark datasets are conducted. The main contributions and novelties of this paper are summarized as follows.

- Guided by transfer learning, we propose a simple Optimal Projection Guided Transfer Hashing (GTH) method. To the best of our knowledge, there are few methods proposed to handle the problem that there are insufficient training images to learn precise model. We first develop a total unsupervised transfer hashing method to solve cross-domain hashing problem for image retrieval based on conventional machine learning.
- We first propose to learn hashing projections of source and target domains respectively for characterizing the domain disparity. The domain gap is reduced by modeling on hashing projections with Gaussian prior based l_2 loss rather than explicitly reducing the domain discrepancy in data-level. This is the GTH-g model.
- Beyond the Gaussian prior, we further propose a novel GTH-h model to seek for the maximum likelihood estimation (MLE) solution of the hashing functions of target and source domains. An iteratively weighted l_2 loss is designed for characterizing the domain gap between the projections of source and target domains, such that high errors can be punished. Besides, the projections of both domains are optimized simultaneously, so that the domain adaptive hashing projection can be obtained.
- Extensive experiments on various benchmark databases are conducted. The experimental results verify that our method outperforms many state-of-the-art unsupervised learning to hash methods.

The rest of this paper is organized as follows. In Section II, we review the related work. Section III presents the proposed GTH model. The experimental results and discussions are shown in Section IV. Finally, Section V concludes this paper.

II. Related Work

In this section, we present related works on learning to hash and transfer learning.

A. Learning to hash

In the past 10 years, various hashing methods have been proposed. Based on whether priori semantic information is used, they can categorized into two major groups: supervised hashing and unsupervised hashing. Also, we introduce a new category of transfer hashing, which leverages an auxiliary domain for learning to hash on target domain.

Supervised hashing. There are a lot of supervised hashing methods, which can be categorized into three subgroups: pairwise similarity preserving hashing, multiwise similarity preserving hashing, and classification loss based hashing. The pairwise similarity preservation methods consist of LDA hashing [15], Minimal Loss Hashing [16],[17], Kernel-based Supervised Hashing (KSH) [18], and the Kernel-based Supervised Discrete Hashing (KSDH) [20]. Those methods aim to align the pair-wise similarity computed in the hash codes such that pairwise semantic similarity is preserved by maximizing the inter-class variations and minimizing the intra-class variations of the hash codes. Multiwise similarity preserving hashing aims to maximize the agreement of the similarity orders over more than two items between the original space and the hamming space, which means multiwise hashing preserves groundtruth orders of ranking lists. The representative methods include concomitant min-hashing (CMH) [25], Winner-Takes-All Hashing (WTAH) [26], Triplet Loss Hashing [6], and Listwise Supervision Hashing [27], etc. Classification loss based supervised hashing formulates the problem by imposing the semantic information to the binary codes, which can make the final learned binary codes more discriminative. The latest Supervised Discrete Hashing (SDH) [19] takes advantage of the supervised information in the hashing code learning phase. By imposing the label information over the learned binary codes, the classification performance is improved. Subsequently, the improved classification hashing approach, Supervised Quantization for similarity search (SQ) [21], is proposed to enhance the retrieval performance. The overlapping of data points in different clusters is reduced. However, for supervised hashing, there always lacks label information for model learning due to the high cost of labour and finance in some real-world application situation.

Unsupervised hashing. Unsupervised methods aim to explore the intrinsic structure of data to preserve the similarity of neighbors without any supervised information. A number of unsupervised hashing methods have been developed in recent years. Locality-sensitive Hashing (LSH) [7], a typical data-independent method, uses a set of randomly generating projection to transform the image features to hashing codes. To solve the random projections selection problem of LSH, Density Sensitive Hashing (DSH) [11] uses projective functions which best agree with the distribution of the data to explore latent geometric structure. Spectral Hashing (SH) [9] maintains the manifold structure of original high-dimension space in the learned Hamming space. However, when the number of samples is becoming bigger, it is memory-consuming to construct similarity matrix and time-consuming to make direct eigen-decomposition. To avoid that problem, Anchor Graph Hashing (AGH) [10] uses K-means clustering to obtain K cluster centers as anchor to approximate the original similarity matrix. Iterative Quantization (ITQ) [5] learns similarity-preserving binary codes by finding a rotation of zero-centered data so as to minimize the quantization error of mapping this data to the vertices of a zero-centered binary hypercube. Circulant Binary Embedding (CBE) [12] generates binary codes by projecting the data with a circulant matrix. The circulant structure enables the use of Fast Fourier Transformation to speed up the computation. Scalable Graph Hashing (SGH) [13] construct the similarity matrix between samples in unsupervised manner. The inner product between binary features of samples is used to approximate that similarity matrix. Very recently, Do et al. [28] proposed two novel hashing models, i.e. relaxed binary autoencoder (RBA) and simultaneous feature aggregating and hashing (SAH), for avoiding the suboptimal hash codes due to the independent design between feature aggregation and hashing. By combining the powerful feature representation of

deep networks, Shen et al. [29] proposed an unsupervised deep hashing model, i.e. similarity-adaptive and discrete hashing (SADH), and achieved great performance. Several ranking-preserved hashing algorithms were proposed recently to learn more discriminative binary codes, e.g., Ordinal Constraint Hashing (OCH) [14] and Deep Ordinal Hashing (DOH) [30]. The merit of DOH [30] lies in the joint integration of global semantic structure and local spatial information with spatial attention maps, observed by FCN and CNN networks.

Transfer hashing. This is a relatively new topic in learning to hashing inspired by transfer learning and domain adaptation. There is a few work on this challenging topic. Formally, transfer hashing tends to leverage an auxiliary domain (source domain) of sufficient data for learning effective hashing codes with the domain of interest (i.e., target domain). Zhou et al. [31] first proposed a transfer hashing framework with privileged information (THPI), which includes two transferable variants of ITQ, i.e., ITQ+ and LapITQ+. An inherent issue of both methods is that the same number of source data as the target domain is constrained for matrix computation. Further, Zhou et al. [32] first proposed a deep transfer hashing (DTH) and shows great performance. Venkateswara et al. [33] proposed a deep hashing network (DHN) for unsupervised domain adaptation. Zhu et al. [34] proposed a discrete semantic transfer hashing (DSTH), which aims to exploiting auxiliary contextual modalities to augment the semantics of hashing codes of images with visual similarity of images preserved. In our work, a optimal projection guided transfer hashing (GTH) is induced from the viewpoint of maximum likelihood estimation for completely unsupervised learning to hash.

B. Transfer learning

Unsupervised and semi-supervised learning for addressing small data challenges play an vital role in big data era [1]. Transfer learning (TL) [35], a newly developed learning perspective for small data challenges in recent years, aims to transfer knowledge across two different domains such that rich source domain knowledge can be utilized to generate better classifiers on a target domain. In transfer learning, the transferred knowledge can be labels [36], [37], features [38], [39], [40], [41], [42] and cross domain correspondences [43], [44]. Shu et al. [45] and Tang et al. [46] proposed deep transfer networks (DTNs) for heterogeneous domain knowledge propagation from text domain to image domain and addressing a challenging cross-modal transfer problem. Shu et al. [47] also proposed a personalized aging transfer model, bi-level dictionary learning based personalized age progression (BDL-PAP), towards render photo-realistic aging faces. Zhou et al. [48] proposed a Sparse Heterogeneous Feature Representation (SHFR) to learn feature mapping across domain with multiple classes, inspired by language translation. Transfer learning has shown promising results in many machine learning tasks, such as classification and regression, and more recent advances in transfer learning and domain adaptation methodologies can be referred to as [49]. To the best of our knowledge, there are few works on studying transfer learning in learning to hashing, except [31], [34], [32], [33] from shallow to deep. Different from their works, we focus on knowledge transfer across hashing projections induced from the perspective of maximum likelihood estimation. It is worth noting that our GTH is a completely unsupervised model, therefore both the labels in target and source domains will not be used in our GTH.

TABLE I: Notations and descriptions

Notation	Description	Notation	Description
$\mathbf{X}_t$	target samples	$\mathbf{M}$	weight matrix
$\mathbf{X}_s$	source samples	N_t	#target sample
$\mathbf{B}_t$	target codes	N_s	#source sample
$\mathbf{B}_s$	source codes	d	#dimension
$\mathbf{W}_t$	target projection	r	code length
$\mathbf{W}_s$	source projection	λ_1,λ_2	parameters

III. Optimal Projection Guided Transfer Hashing

In this section, we elaborate our proposed Optimal Projection Guided Transfer Hashing (GTH). We firstly present a formal description of the problem in the scenario where there are insufficient training samples on the target domain. Then, we present the detailed deduction of our proposed GTH model from the viewpoint of maximum likelihood estimation (MLE). Finally, the detailed optimization process is presented.

A. Problem Description

Suppose that we have N_t target data points $\mathbf{X}_t = [\mathbf{x}_{t_1}, \mathbf{x}_{t_2}, \cdots, \mathbf{x}_{t_{N_t}}] \in \mathbb{R}^{d\times N_t}$. We aim to learn a set of binary code $\mathbf{B}_t = \{\mathbf{b}_{t_i}\}_{i=1}^{N_t} \in \{-1, 1\}^{r\times N_t}$ to well preserve feature information of the original dataset. $\mathbf{b}_{t_i}$ is the corresponding binary codes of $\mathbf{x}_{t_i}$. N_t, d, and r denote the number of the target domain samples, the dimension of each sample, and the code length of binary feature, respectively. Similar with most of learning to hash methods, we also learn hashing projection to map and quantize each $\mathbf{x}_{t_i}$ into a binary codes $\mathbf{b}_{t_i}$. However, when the available target training data is limited, i.e., N_t is small, the binary codes learned by existing learning to hash methods can not perform well. In our GTH, we take advantage of the knowledge (i.e., features) of another known domain (i.e., source domain). Suppose that we have already obtained N_s source data points $\mathbf{X}_s = [\mathbf{x}_{s_1}, \mathbf{x}_{s_2}, \cdots, \mathbf{x}_{s_{N_s}}] \in \mathbb{R}^{d\times N_s}$. The main problem is how to transfer the knowledge of source domain into target domain. In our GTH, we propose to learn hashing projection of target domain $\mathbf{W}_t$ and hashing projection of source domain $\mathbf{W}_s$, respectively. We assume that similar images between target and source domains should entail small discrepancy between their hashing projections. Therefore, we let the projections of target and source domain close to each other, such that the similar instances between those two domains will be encoded and transformed into similar hashing codes. Some frequently used notations and their associated descriptions are summarized in Table I.

B. Model Formulation of GTH

We denote $\mathbf{B}_t = \mathrm{H}(\mathbf{W}_t^{\mathrm{T}}\mathbf{X}_t)$ and $\mathbf{B}_s = \mathrm{H}(\mathbf{W}_s^{\mathrm{T}}\mathbf{X}_s)$ where $\mathbf{W}_t \in \mathbb{R}^{d\times r}$ is hashing projection of target domain and $\mathbf{W}_s \in$

$\mathbb{R}^{d\times r}$ is hashing projection of source domain. $\mathrm{H}(v) = sgn(v)$ is the symbolic function, which equals to 1 if $v \geq 0$ and -1 otherwise. In our GTH, to learn domain-invariant hashing projections, we let the hashing projection of target domain be close to the source domain:

$$\min_{\mathbf{W}_t,\mathbf{W}_s} \|\mathbf{W}_t - \mathbf{W}_s\|^2 \tag{1}$$

We denote that $\mathbf{E} = \mathbf{W}_t - \mathbf{W}_s$ represents the error matrix. E_{ij} is one element in the error matrix. As discussed above, from the view of maximum likelihood estimation (MLE), the error matrix follows Gaussian distribution by using the Eq. (1). However, the different data distributions of source and target domains may lead to that the probability distribution of error matrix is far from Gaussian distribution. Without loss of generality, we let $\mathbf{e} = [E_{11}, E_{21}, \cdots, E_{d1}, \cdots, E_{1r}, E_{2r}, \cdots, E_{dr}]^{\mathrm{T}}$. Assume that $e_1, e_2, \cdots, e_N$ are independently and identically distributed according to some probability density function (PDF) $f_\theta(e_n)$ where $N = d \times r$ and θ denotes the parameter set that characterizes the distribution. The likelihood estimation can be represented as $L_\theta = \prod_{n=1}^{N} f_\theta(e_n)$ and MLE aims to maximize this likelihood function or minimize the negative log likelihood function: $-\log L_\theta = \sum_{n=1}^{N} \rho_\theta(e_n)$ where $\rho_\theta(e_n) = -\log f_\theta(e_n)$.

With the above analysis, the Eq. (1) with uncertain probability density function can be transformed into the following minimization problem:

$$\min_{\mathbf{W}_t,\mathbf{W}_s} \sum_{n=1}^{N} \rho_\theta(e_n) \tag{2}$$

In general, we assume that the unknown PDF $f_\theta(e_n)$ is symmetric, and the bigger error will assign a low probability value, i.e., $f_\theta(e_i) < f_\theta(e_j)$ if $|e_i| > |e_j|$. Therefore, $\rho_\theta(e_n)$ has the following properties: 1) $\rho_\theta(0)$ is the global minimal of $\rho_\theta(e_n)$; 2) if we denote $\rho_\theta(0) = 0$, $\rho_\theta(e_n) = \rho_\theta(-e_n)$; 3) $\rho_\theta(e_i) < \rho_\theta(e_j)$ if $|e_i| < |e_j|$.

Denote that $F_\theta(\mathbf{e}) = \sum_{n=1}^{N} \rho_\theta(e_n)$. $F_\theta(\mathbf{e})$ can be approximated by using its first order Taylor expansion in the neighborhood $\mathbf{e}_0$:

$$\widetilde{F}_\theta(\mathbf{e}) = F_\theta(\mathbf{e}_0) + (\mathbf{e} - \mathbf{e}_0)^{\mathrm{T}} F'_\theta(\mathbf{e}_0) + R_1(\mathbf{e}) \tag{3}$$

where $R_1(\mathbf{e})$ is the higher-order remained term, and $F'_\theta(\mathbf{e}_0)$ is the derivative of $F_\theta(\mathbf{e}_0)$.

$$R_1(\mathbf{e}) = 0.5(\mathbf{e} - \mathbf{e}_0)^{\mathrm{T}} \mathbf{\Omega} (\mathbf{e} - \mathbf{e}_0) \tag{4}$$

$\mathbf{\Omega}$ is a diagonal matrix and we denote

$$\Omega_{nn} = \rho'_\theta(\Lambda_n)/\Lambda_n = \omega_\theta(\Lambda_n) \tag{5}$$

where we randomly assign a value to Λ_n that satisfies $\Lambda_n \in (0, e_n)$ if $e_n > 0$, otherwise $\Lambda_n \in (e_n, 0)$, for improving the flexibility and robustness of the model to the weights. $\rho'_\theta(\Lambda_n)$ represents the first-order derivative. Because $\rho_\theta(0)$ is the global minimal of $\rho_\theta(e_n)$, we can get $\rho'_\theta(0) = 0$. We denote $\mathbf{e}_0 = \mathbf{0}$ such that the following objective function is obtained,

$$\widetilde{F}_\theta(\mathbf{e}) = R_1(\mathbf{e}) = 0.5\|\mathbf{\Omega}^{\frac{1}{2}}\mathbf{e}\|^2 \tag{6}$$

It is obvious that each element Ω_{nn} in the diagonal matrix $\mathbf{\Omega}$ can be regarded as a weight coefficient with respect to each error value e_n. We expect that the higher value $|e_n|$ will be assigned a lower weight coefficient Ω_{nn}.

According to [23] and [50], we also choose the sigmoid function as the weight function,

$$\omega_\theta(\Lambda_n) = \exp(\mu\delta - \mu\Lambda_{\mathrm{n}}^2)/(1 + \exp(\mu\delta - \mu\Lambda_{\mathrm{n}}^2)) \tag{7}$$

where μ and δ are positive scalars. Parameter μ controls the decreasing rate from 1 to 0, and δ controls the location of demarcation point. For the choice of μ and δ, we just follow [23]. Considering the Eq. (5), Eq. (7), and $\rho_\theta(0) = 0$, we obtain $\rho_\theta(\Lambda_n)$ as follows,

$$\rho_\theta(\Lambda_n) = \frac{-1}{2\mu}\big(ln(1+\exp(\mu\delta-\mu\Lambda_{\mathrm{n}}^2)) - \ln(1+\exp(\mu\delta))\big) \tag{8}$$

Therefore, we can transform Eq. (6) into matrix form as following objective function,

$$\min_{\mathbf{W}_t,\mathbf{W}_s} \frac{1}{2}\|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 \tag{9}$$

where $\odot$ denotes element-wise product. We denote $M_{ij} = \omega_\theta(\widetilde{E}_{ij})$, and we randomly choose a value as $\widetilde{E}_{ij}$ that satisfies $\widetilde{E}_{ij} \in (0, E_{ij})$ if $E_{ij} > 0$, otherwise $\widetilde{E}_{ij} \in (E_{ij}, 0)$. Note that $\mathbf{M}$ is the matrix form of all diagonal elements in $\mathbf{\Omega}$.

It is worth noting that the Eq. (9) can be viewed as an inductive model. If we let $\omega_\theta(\widetilde{E}_{ij}) = 2$, then Eq. (9) is degenerated into Eq. (1), which assumes that the errors obey Gaussian distribution. Specially, in this paper, GTH-h refers to Eq. (9) with $\omega_\theta(\widetilde{E}_{ij})$ defined as Eq. (7) and GTH-g refers to Eq. (9) with $\omega_\theta(\widetilde{E}_{ij}) = 2$. The proof is shown in *Appendix*.

The quantization loss between hashing codes and its magnitude is used as regularization term in GTH. Besides, we impose orthogonality constraints to the hashing projections $\mathbf{W}_s$ and $\mathbf{W}_t$, such that the discrimination of hashing codes $\mathbf{B}_s$ and $\mathbf{B}_t$ can be guaranteed. The overall objective function of GTH is formulated as

$$\begin{aligned}\min_{\mathbf{W}_t,\mathbf{W}_s,\mathbf{B}_t,\mathbf{B}_s} & \frac{1}{2}\|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 \\ & + \frac{\lambda_1}{2}\|\mathbf{B}_t - {\mathbf{W}_t}^{\mathrm{T}}\mathbf{X}_t\|^2 + \frac{\lambda_2}{2}\|\mathbf{B}_s - {\mathbf{W}_s}^{\mathrm{T}}\mathbf{X}_s\|^2 \\ s.t. \quad & {\mathbf{W}_t}^{\mathrm{T}}\mathbf{W}_t = \mathbf{I}, {\mathbf{W}_s}^{\mathrm{T}}\mathbf{W}_s = \mathbf{I}, \\ & \mathbf{B}_t = \mathrm{H}({\mathbf{W}_{\mathrm{t}}}^{\mathrm{T}}\mathbf{X}_{\mathrm{t}}), \mathbf{B}_{\mathrm{s}} = \mathrm{H}({\mathbf{W}_{\mathrm{s}}}^{\mathrm{T}}\mathbf{X}_{\mathrm{s}})\end{aligned} \tag{10}$$

where $\lambda_1$and λ_2 denote the regularization coefficients.

C. Solving Algorithm

In this paper, we propose a weighted l_2 loss for the errors between the projections of source and target domains, and update the weight coefficients by using the errors from the last iteration. As the non-convex $sgn(\cdot)$ function makes Eq. (10) a NP-hard problem, we relax the $sgn(x)$ function as its signed magnitude x [51]. Therefore, the Eq. (10) can be rewritten as

$$\begin{aligned}\min_{\mathbf{W}_t,\mathbf{W}_s,\mathbf{B}_t,\mathbf{B}_s} & \frac{1}{2}\|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 \\ & + \frac{\lambda_1}{2}\|\mathbf{B}_t - {\mathbf{W}_t}^{\mathrm{T}}\mathbf{X}_t\|^2 + \frac{\lambda_2}{2}\|\mathbf{B}_s - {\mathbf{W}_s}^{\mathrm{T}}\mathbf{X}_s\|^2 \\ s.t. \quad & {\mathbf{W}_t}^{\mathrm{T}}\mathbf{W}_t = \mathbf{I}, {\mathbf{W}_s}^{\mathrm{T}}\mathbf{W}_s = \mathbf{I}, b^s_{i,j}, b^t_{i,k} \in \{1, -1\}\end{aligned} \tag{11}$$

As mentioned above, we will adopt a relax way to solve problem (10). The solutions for optimization problem (11) can be calculated by alternatingly updating the variables, $\mathbf{W}_t$, $\mathbf{W}_s$, $\mathbf{B}_t$, $\mathbf{B}_s$, and $\mathbf{M}$.

$\mathbf{W}_t$-Step. By fixing $\mathbf{W}_s$, $\mathbf{B}_t$, $\mathbf{B}_s$, and $\mathbf{M}$, the projection of target domain $\mathbf{W}_t$ can be obtained by solving the following subproblem

$$\begin{aligned} \min_{\mathbf{W}_t} \|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 + \lambda_1 \|\mathbf{B}_t - {\mathbf{W}_t}^{\mathrm{T}} \mathbf{X}_t\|^2 \\ s.t. \quad {\mathbf{W}_t}^{\mathrm{T}} \mathbf{W}_t = \mathbf{I} \end{aligned} \tag{12}$$

Updating $\mathbf{W}_t$ is a typical optimization problem with orthogonality constraints. We apply the optimization procedure in [52] to update $\mathbf{W}_t$. Let $\mathbf{G}_t$ be the partial derivative of the objective function with respect to $\mathbf{W}_t$. $\mathbf{G}_t$ is represented as

$$\mathbf{G}_t = \mathbf{M} \odot (\mathbf{W}_t - \mathbf{W}_s) + \lambda_1 (\mathbf{X}_t \mathbf{X}_t^{\mathrm{T}} \mathbf{W}_t - \mathbf{X}_t \mathbf{B}_t^{\mathrm{T}}) \tag{13}$$

To preserve the orthogonality constraint on $\mathbf{W}_t$, we first define the skew-symmetric matrix $\mathbf{Q}_t$ [53] as $\mathbf{Q}_t = \mathbf{W}_t^{\mathrm{T}} \mathbf{G}_t - \mathbf{G}_t^{\mathrm{T}} \mathbf{W}_t$. Then, we adopt Crank Nicolson like scheme [52] to update the orthogonal matrix $\mathbf{W}_t$:

$$\mathbf{W}_t^{(k+1)} = \mathbf{W}_t^{(k)} - \frac{\tau}{2} (\mathbf{W}_t^{(k+1)} + \mathbf{W}_t^{(k)}) \mathbf{Q}_t \tag{14}$$

where τ denotes the step size. We empirically set $\tau = 0.1$. By solving Eq. (14), we can get

$$\mathbf{W}_t^{(k+1)} = \mathbf{W}_t^{(k)} \mathbf{Q}_t \tag{15}$$

and $\mathbf{Q}_t^{(k+1)} = (\mathbf{I} + \frac{\tau}{2} \mathbf{Q}_t)^{-1} (\mathbf{I} - \frac{\tau}{2} \mathbf{Q}_t)$. We iteratively update $\mathbf{W}_t$ several times based on Eq. (15) with the Barzilai-Borwein (BB) method [52]. Note that the orthogonality of $\mathbf{W}$ can be guaranteed with this solver.

$\mathbf{W}_s$-Step. By fixing $\mathbf{W}_t$, $\mathbf{B}_t$, $\mathbf{B}_s$, and $\mathbf{M}$, the projection of source domain $\mathbf{W}_s$ can be solved as:

$$\begin{aligned} \min_{\mathbf{W}_s} \|\mathbf{M}^{\frac{1}{2}} \odot (\mathbf{W}_t - \mathbf{W}_s)\|^2 + \lambda_2 \|\mathbf{B}_s - {\mathbf{W}_s}^{\mathrm{T}} \mathbf{X}_s\|^2 \\ s.t. \quad {\mathbf{W}_s}^{\mathrm{T}} \mathbf{W}_s = \mathbf{I} \end{aligned} \tag{16}$$

Updating $\mathbf{W}_s$ is the same as $\mathbf{W}_t$. Let $\mathbf{G}_s$ be the partial derivative of the objective function with respect to $\mathbf{W}_s$, then $\mathbf{G}_s$ can be represented as

$$\mathbf{G}_s = \mathbf{M} \odot (\mathbf{W}_s - \mathbf{W}_t) + \lambda_2 (\mathbf{X}_s \mathbf{X}_s^{\mathrm{T}} \mathbf{W}_s - \mathbf{X}_s \mathbf{B}_s^{\mathrm{T}}) \tag{17}$$

To preserve the orthogonality constraint on $\mathbf{W}_s$, we define the skew-symmetric matrix $\mathbf{Q}_s$ as $\mathbf{Q}_s = \mathbf{W}_s^{\mathrm{T}} \mathbf{G}_s - \mathbf{G}_s^{\mathrm{T}} \mathbf{W}_s$. Then, we adopt Crank Nicolson like scheme to update the orthogonal matrix $\mathbf{W}_s$:

$$\mathbf{W}_s^{(k+1)} = \mathbf{W}_s^{(k)} - \frac{\tau}{2} (\mathbf{W}_s^{(k+1)} + \mathbf{W}_s^{(k)}) \mathbf{Q}_s \tag{18}$$

where τ denotes the step size. We set $\tau = 0.1$ which is the same as that of $\mathbf{W}_t$. By solving Eq. (18), we can get

$$\mathbf{W}_s^{(k+1)} = \mathbf{W}_s^{(k)} \mathbf{Q}_s \tag{19}$$

where $\mathbf{Q}_s^{(k+1)} = (\mathbf{I} + \frac{\tau}{2} \mathbf{Q}_s)^{-1} (\mathbf{I} - \frac{\tau}{2} \mathbf{Q}_s)$. We can iteratively update $\mathbf{W}_s$ for several times based on Eq. (19) with the Barzilai-Borwein (BB) method.

Algorithm 1 Optimal Projection Guided Transfer Hashing

Input: Target samples $\mathbf{X}_t$, source samples $\mathbf{X}_s$, parameters $\lambda_1 = 0.1$, $\lambda_2 = 1$, and the identity matrix $\mathbf{I}$;

Output: $\mathbf{W}_t$, $\mathbf{B}_t$, $\mathbf{W}_s$, and $\mathbf{B}_s$.

Initialize: Initialize $\mathbf{W}_t^{(0)}$ and $\mathbf{W}_s^{(0)}$ as the top r eigenvectors of $\mathbf{X}_t \mathbf{X}_t^{\mathrm{T}}$ and $\mathbf{X}_s \mathbf{X}_s^{\mathrm{T}}$ corresponding to the first r largest eigenvalues, respectively. $\mathbf{B}_t^{(0)}$ and $\mathbf{B}_s^{(0)}$ are random matrices, and $k = 1$.
repeat
  Compute the error matrix $\mathbf{E}^{(k-1)} = \mathbf{W}_t^{(k-1)} - \mathbf{W}_s^{(k-1)}$. The $\widetilde{\mathbf{E}}$ is obtained, which satisfies that $\widetilde{E}_{ij} \in (0, E_{ij})$ if $E_{ij} > 0$ otherwise $\widetilde{E}_{ij} \in (E_{ij}, 0)$.
  update $\mathbf{M}^{(k)}$ with Eq. (22): Compute the weight as $M_{ij}^k = \frac{exp(\mu^{(k-1)} \delta^{(k-1)} - \mu^{(k-1)} \widetilde{E}_{ij}^2)}{1 + exp(\mu^{(k-1)} \delta^{(k-1)} - \mu^{(k-1)} \widetilde{E}_{ij}^2)}$, where the choice of parameter $\mu^{(k-1)}$ and $\delta^{(k-1)}$ can be found in experimental setting part.
  update $\mathbf{W}_t^{(k)}$: by solving Eq. (15);
  update $\mathbf{W}_s^{(k)}$: by solving Eq. (19);
  update $\mathbf{B}_t^{(k)}$: by solving Eq. (20);
  update $\mathbf{B}_s^{(k)}$: by solving Eq. (21);
  k=k+1;
until maximum iterations

$\mathbf{B}_t$-Step and $\mathbf{B}_s$-Step. Because $\mathbf{B}_t$ and $\mathbf{B}_s$ are two binary matrices, the solutions can be directly obtained as:

$$\mathbf{B}_t = sgn(\mathbf{W}_t^{\mathrm{T}} \mathbf{X}_t) \tag{20}$$

$$\mathbf{B}_s = sgn(\mathbf{W}_s^{\mathrm{T}} \mathbf{X}_s) \tag{21}$$

M-Step. The weight matrix $\mathbf{M}$ can be directly computed as follows:

$$\mathbf{M} = \omega_\theta (\mathbf{W}_t - \mathbf{W}_s) \tag{22}$$

where $\omega_\theta(\cdot)$ is denoted as Eq. (7).

Specifically, the overall solving process of our GTH model is summarized in Algorithm 1.

D. Computation Complexity

The time cost of the proposed GTH in Algorithm 1 consists of three parts: 1) optimizing the hashing projections $\mathbf{W}_t$ and $\mathbf{W}_s$, 2) optimizing the binary codes $\mathbf{B}_t$ and $\mathbf{B}_s$, and 3) optimizing the weight matrix $\mathbf{M}$. First, the computational cost of $\mathbf{W}_t$ involves the computation of Eq. (13), (14) and (15), which requires $\mathcal{O}(drn_t + d^2(r + n_t))$, $\mathcal{O}(r^2 d)$ and $\mathcal{O}(r^2 d)$, respectively. Therefore, the updating $\mathbf{W}_t$ requires $\mathcal{O}(drn_t + d^2(r + n_t) + r^2 d)$. Similar to that of $\mathbf{W}_t$, updating of $\mathbf{W}_s$ requires $\mathcal{O}(drn_s + d^2(r + n_s) + r^2 d)$. Second, the computational cost of $\mathbf{B}_t$ and $\mathbf{B}_s$ involves the computation of Eq. (20) and (21), which requires $\mathcal{O}(dr(n_s + n_t))$. Third, the computational cost of $\mathbf{M}$ involves Eq. (22), which requires $\mathcal{O}(dr)$. Therefore, with K iterations for convergence, the total computation complexity of GTH is $\mathcal{O}(K(d^2 r + nd^2 + ndr + r^2 d))$. Note that d and r denote the feature dimensionality and binary code length, respectively. n_s and n_t denote the number of samples in source and target domain, respectively. $n = n_s + n_t$ is the total number of samples.

TABLE II: Experimental tasks constructed based on the Multi-PIE, Office, VLCS and ImageNet databases.

Task	Task 1	Task 2	Task 3	Task 4	Task 5	Task 6
Source domain	PIE-C05	Office (Amazon)	VLCS (VOC2007)	ImageNet	ImageNet	ImageNet
Target domain	PIE-C29	Office (Dslr)	VLCS (Caltech101)	VLCS (LabelMe)	VLCS (VOC2007)	VLCS (SUN09)

IV. Experiment

In this section, extensive experiments are conducted to evaluate the proposed hashing method on image retrieval performance. We first perform the experiments on three groups benchmark datasets: PIE-C05&PIE-C29 from **Multi-PIE** [54], Dslr&Amazon from **Office** [55], and VOC2007&Caltech101 from **VLCS** [56]. Besides, we use the LabelMe, VOC2007, and SUN09 in **VLCS** as the target domain, respectively, and the related subset of ImageNet [57] as the source domain to test the proposed GTH model. For comparisons, we choose to compare with 10 state-of-the-art learning-to-hash methods, including LSH [7], ITQ [5], CBE [12], DSH [11], SpH [58], SGH [13], OCH [14], ITQ+ [31], and LapITQ+ [31]. Note that ITQ+ and LapITQ+ are baselines of transfer hashing.

A. Datasets, Settings, and Implementation Details

Description of Datasets: In experiments, four databases, including Multi-PIE, Office, VLCS and ImageNet, are exploited to verify the effectiveness of the proposed method.

- The **Multi-PIE** dataset consists of 41,368 face images from 68 subjects. The images are under five near-frontal poses (C05, C07, C09, C27 and C29). We use the two subsets selected from poses C05 and C29. Each image is resized to 32×32 and represented by a 1024-dimensional vector. We use the pose session C29 that contains 1632 images as the target domain and the pose session C05 that contains 3332 images as the source domain, i.e., C05&C29. Additionally, for the target domain, we randomly select 500 samples as testing images and the rest samples are used as training images.
- The **Office** dataset is the most popular benchmark object dataset for evaluating domain adaptation models in computer vision community. The dataset consists of daily objects in office environment. **Office** has 3 domains: Amazon (A), Dslr (D), and Webcam (W). The *Amazon* dataset with 2817 images is used as the source domain and the *Dslr* with 498 images is used as target domain, i.e., Amazon&Dslr. 100 images from target domain are randomly selected as testing set and the rest images are used as training set. Each image is represented by a 4096-d deep feature vector extracted with a well-trained convolutional neural network (CNN) [59].
- The **VLCS** dataset aggregates photos from Caltech, LabelMe, Pascal VOC 2007 and SUN09. It provides a 5-way multi-class benchmark on five common classes: *bird*, *car*, *chair*, *dog* and *person*. The *VOC 2007* dataset that contains 3376 images is used as the source domain and the *Caltech* dataset that contains 1415 images is used as the target domain, i.e., VOC2007&Caltech. 100 images from target domain are randomly selected as testing set and the rest images are used as the training set. Each image is represented by a 4096-d CNN feature vector [59].
- The **ImageNet** dataset contains over 14 million labeled data, and we adopt the ILSVRC 2012 subset, which has more than 1.2 million images of totally 1000 object categories. The images related to VLCS are selected to form a subset and worked as the target domain. To be specific, 3376, 2656, and 3282 samples are included in VOC2007, LabelMe and SUN09 subset, respectively. The subset of ImageNet that contains 7341 images is used as source domain. Therefore, three tasks including ImageNet&LabelMe, ImageNet&VOC2007, and ImageNet&SUN09 are experimented. Each image is represented by a 4096-d CNN feature vector [59].

In summary, there are total 6 tasks with regard to source and target domain constructed by the four databases described above. The details of the 6 tasks are presented in Table V.

Parameter settings: There are two trade-off parameters in the objective function (10), i.e., λ_1 and λ_2, which are used to penalize the loss between the binary codes and its signed magnitude. For our GTH, we empirically set λ_1 as 1 and λ_2 as 0.1. In the weight function Eq. (7), there are two parameters δ and μ, which need to be calculated in the algorithm. δ is the parameter of demarcation point. When the square of residual is larger than δ, the weight value is less than 0.5. In order to make the model robust to outliers, we compute the value of δ as follows. Denote that $\psi = [E_{11}^2, E_{21}^2, \cdots, E_{d1}^2, \cdots, E_{1r}^2, E_{2r}^2, \cdots, E_{dr}^2]$. By sorting ψ in an ascending order, we get the re-ordered array ψ_a. Let $k = \lfloor \tau N \rfloor$, where the scalar $\tau \in (0, 1]$. The $\lfloor \tau N \rfloor$ outputs the largest integer but smaller than τN. We set δ as $\psi_a(k)$. Parameter μ controls the decreasing rate of weight value from 1 to 0. Here we simply let $\mu = c/\delta$, where c is a constant. In the experiments, if there is no specific instruction, c is set as 10 and τ is set as 0.8.

Implementation details: The compared baseline methods are proposed under no domain adaption assumption. For fair comparisons, two settings are considered for the traditional hashing baselines. 1) Target only, in which we only use the target training data for learning to hashing without using source domain data. 2) Source+target, in which we intuitively use all the source domain data and target domain training data (except the testing queries of the target domain) for training the compared models. During test phase, we only focus on the retrieval performance of the target domain and report the mean average precision (MAP) scores based on the Hamming MAP used in [31]. For each query, the first K nearest neighbors are considered as true positives, where K is determined as the 2% of the number of target domain samples.

TABLE III: The MAP scores (%) on the Multi-PIE (PIE-C05&PIE-C29), Office (Amazon&Dslr), and VLCS (VOC2007&Caltech101) databases with varying code length from 16 to 64. Notably, *target only* means that the model is trained on target domain data without using source domain data. For others, both the source and target data are used for training without special indication.

	PIE-C05&PIE-C29					Amazon&Dslr					VOC2007&Caltech101				
Bit	16	24	32	48	64	16	24	32	48	64	16	24	32	48	64
LSH(target only)	17.11	20.80	24.84	29.77	19.92	28.08	35.60	44.82	51.45	52.21	11.13	16.41	21.58	29.28	33.89
LSH(source+target)	18.23	21.79	25.26	29.91	32.96	19.69	28.92	35.12	46.72	53.07	11.06	16.51	20.61	27.41	33.12
ITQ(target only)	19.81	23.26	26.51	30.39	33.53	41.29	49.62	53.44	59.37	62.61	29.60	39.28	42.61	48.14	51.50
ITQ(source+target)	18.17	21.63	23.74	26.82	28.86	43.15	51.74	56.80	62.47	65.84	21.69	28.52	33.46	39.50	42.34
CBE(target only)	16.83	22.19	25.56	30.55	33.15	18.73	27.94	34.16	44.68	52.35	11.37	16.85	21.34	29.25	33.66
CBE(source+target)	16.31	22.13	27.10	30.06	32.51	20.82	27.60	36.21	47.52	51.96	11.04	15.64	20.68	26.97	33.84
DSH(target only)	17.62	21.45	26.24	31.21	34.13	26.89	34.51	39.29	50.02	53.97	15.64	20.20	22.09	26.73	29.81
DSH(source+target)	17.05	19.60	22.01	25.65	28.12	26.51	32.34	37.39	48.29	50.12	8.69	6.23	13.40	15.56	20.21
SpH(target only)	25.55	29.92	32.74	35.12	36.85	31.70	40.61	45.51	52.94	56.87	25.49	31.28	35.44	39.84	42.98
SpH(source+target)	19.55	23.76	27.35	33.41	36.02	33.80	34.37	37.30	47.84	51.88	12.10	17.76	21.76	24.09	36.57
SGH(target only)	9.12	12.44	14.96	19.62	22.48	47.13	53.46	58.82	64.55	67.52	30.75	39.52	47.56	55.16	60.26
SGH(source+target)	10.72	14.65	15.99	20.10	20.66	42.66	45.06	51.30	59.38	64.33	27.35	34.24	36.43	49.48	53.72
OCH(target only)	21.06	24.76	26.51	32.11	32.34	41.64	51.96	57.21	63.29	65.63	30.77	34.81	36.95	40.78	41.80
OCH(source+target)	20.75	26.29	28.96	33.33	34.39	41.77	52.41	56.00	62.38	65.45	32.94	35.45	38.00	41.46	42.25
ITQ+	24.07	26.71	26.56	26.48	27.09	44.71	50.60	55.24	59.39	60.14	**34.10**	**45.99**	**49.47**	56.47	58.60
LapITQ+	24.62	28.91	**32.43**	34.29	34.95	43.47	52.13	56.57	62.06	63.39	32.28	42.47	47.64	54.10	56.52
GTH-g	24.16	28.40	31.69	34.95	35.70	44.16	**53.57**	**57.59**	**63.91**	**66.96**	28.62	41.20	46.42	56.59	63.10
GTH-h	**25.45**	**29.42**	31.76	**35.25**	**36.56**	**45.23**	52.36	57.26	63.17	65.63	30.05	39.70	48.14	**57.33**	**63.53**

B. Experimental Evaluation on Image Retrieval

In the Table III, we report the MAP scores of all the compared methods and our GTH on PIE-C05&PIE-C29, Amazon&Dslr, and VOC2007&Caltech101 databases. To further demonstrate the effectiveness of GTH, we perform the experiments on ImageNet&VOC2007, ImageNet&LabelMe, and ImageNet&SUN09 databases, and report the MAP scores in Table IV. The code length is varying from 16 to 64. Besides, the Precision-Recall curve, Precision and Recall are also shown. Specifically, the results and analysis of 6 tasks are analyzed as follows.

1) Results on Multi-PIE (PIE-C05&PIE-C29) Dataset: From Table III, we see that our GTH outperforms these compared methods in most cases. More specifically, our GTH outperforms the compared transfer hashing method, i.e. ITQ+ and LapITQ+ on PIE-C05&PIE-C29 datasets. We also show the PR-curve, Precision and Recall for PIE-C05&PIE-C29 datasets in Fig. 3, with the code length setting as 16 and 48, respectively. We can see from Fig. 3 that our GTH always presents the competitive retrieval performance compared to baselines and state-of-the-arts, which demonstrates the efficiency of our GTH. Additionally, in order to have a better insight about the role of source data, we present the results of those baselines without transferability by training the model on target only data. We see that the removal of the source data can produce different performance variation for different methods, which is sometimes better than the results with source data training. The reason is that due to the distribution discrepancy between source and target domains, negative transfer may deteriorate the performance by simply leveraging the source data for training.

2) Results on Office (Amazon&Dslr) Dataset: On Amazon&Dslr datasets, our GTH outperforms all the compared methods as presented in Table III. Our GTH-h outperforms the best LapITQ+ method almost $2\%-4\%$. We show the PR-curve, Precision and Recall for Amazon&Dslr dataset in Fig. 4. The code lengths are set to 16 and 48. From Fig. 4, we can see that our GTH shows competitive retrieval performance by comparing to the baselines.

3) Results on VOC2007&Caltech101 Dataset: On the VOC2007&Caltech101 databases, from Table III, we can see that our GTH outperforms the compared methods when the code length is set as 24, 32, 48, and 64. We show the PR-curve, Precision and Recall for VOC2007&Caltech101 dataset in Fig. 5, by setting the code length as 16 and 48, respectively.

4) Results on ImageNet&LabelMe Dataset: On ImageNet&LabelMe dataset, our GTH also achieves competitive MAP scores. We also show the PR-curve, Precision and Recall for ImageNet&LabelMe dataset in Fig. 6. The code length is set to 16. From the Fig. 6, we can see that our GTH always presents more competitive retrieval performance by comparing to the baselines of learning to hashing.

5) Results on ImageNet&VOC2007 Dataset: From Table IV, we can see that our GTH outperforms the compared methods on all databases in most cases. Especially, when the code length is set to 16, our GHT-g outperforms the second best method NoDA over 1% on MAP. We further present the PR-curve, Precision and Recall for ImageNet&VOC2007 dataset in Fig. 7, by setting the code length as 16. As shown in Fig. 7, our GTH always presents more competitive retrieval performance compared to the baselines, which demonstrates the efficiency of our GTH.

TABLE IV: The MAP scores (%) on the ImageNet&LabelMe, ImageNet&VOC2007, and ImageNet&SUN09 databases with varying code length from 16 to 64. Notably, *target only* means that the model is trained on target domain data without using source domain data. For others, both the source and target data are used for training without special indication.

	ImageNet&LabelMe					ImageNet&VOC2007					ImageNet&SUN09				
Bit	16	24	32	48	64	16	24	32	48	64	16	24	32	48	64
LSH(target only)	4.05	4.90	6.12	7.75	9.97	3.43	3.93	4.56	5.93	7.28	3.60	4.21	4.74	6.22	8.20
LSH(source+target)	4.01	4.90	5.82	8.07	9.74	3.43	4.21	4.54	5.70	6.91	3.79	4.42	4.92	6.20	8.51
ITQ(target only)	10.52	14.95	17.86	23.77	27.96	18.12	22.84	26.40	31.84	36.16	12.60	15.96	19.65	25.29	29.52
ITQ(source+target)	17.90	23.94	**28.39**	**34.19**	**39.05**	22.23	28.42	**33.49**	**38.24**	**42.20**	19.22	**25.51**	**27.50**	**31.96**	34.85
CBE(target only)	3.84	4.91	6.05	7.92	9.53	3.46	4.01	4.63	5.71	6.84	3.56	4.04	4.93	6.38	7.88
CBE(source+target)	4.08	4.92	6.09	7.38	10.87	3.46	3.86	4.58	5.75	7.16	3.69	4.34	4.93	6.32	8.35
DSH(target only)	12.59	15.36	18.67	21.41	24.15	12.35	14.79	16.99	21.10	24.36	11.15	13.56	16.70	19.58	23.30
DSH(source+target)	6.46	8.33	8.73	11.53	13.51	10.03	10.40	14.29	17.29	20.35	7.61	9.48	11.08	14.09	16.63
SpH(target only)	12.06	14.59	17.31	21.37	23.93	11.00	14.57	17.28	21.86	25.68	10.14	12.55	16.25	19.76	23.79
SpH(source+target)	7.50	9.90	10.75	15.20	17.63	8.99	12.20	15.40	19.66	22.43	7.28	9.88	11.94	15.85	18.87
SGH(target only)	20.67	23.48	27.28	30.66	34.37	22.51	26.29	28.99	33.11	35.46	19.27	22.61	24.41	28.89	31.55
SGH(source+target)	17.52	22.11	25.69	30.51	34.21	19.37	22.30	27.25	31.37	33.95	17.28	22.08	25.66	28.40	31.12
OCH(target only)	18.12	23.92	26.99	30.54	35.43	20.48	25.77	30.97	35.21	38.39	17.97	23.81	26.58	30.49	33.44
OCH(source+target)	17.05	24.00	26.23	30.41	34.99	20.54	27.05	30.99	35.55	37.80	17.85	24.27	26.79	30.63	33.50
ITQ+	17.71	20.87	23.05	25.37	27.45	21.30	25.74	27.89	31.08	33.78	17.33	21.47	24.09	26.22	27.81
LapITQ+	18.86	23.61	26.27	30.44	33.44	22.19	26.52	29.74	35.20	37.96	19.73	23.74	26.24	30.32	33.31
GTH-g	**19.99**	**24.74**	27.51	32.30	37.76	21.96	28.09	32.07	37.78	41.29	**20.15**	24.09	27.48	31.84	**36.04**
GTH-h	19.26	24.49	27.32	33.13	38.09	**22.51**	**28.94**	32.40	37.72	41.62	19.81	24.17	27.12	31.94	35.80

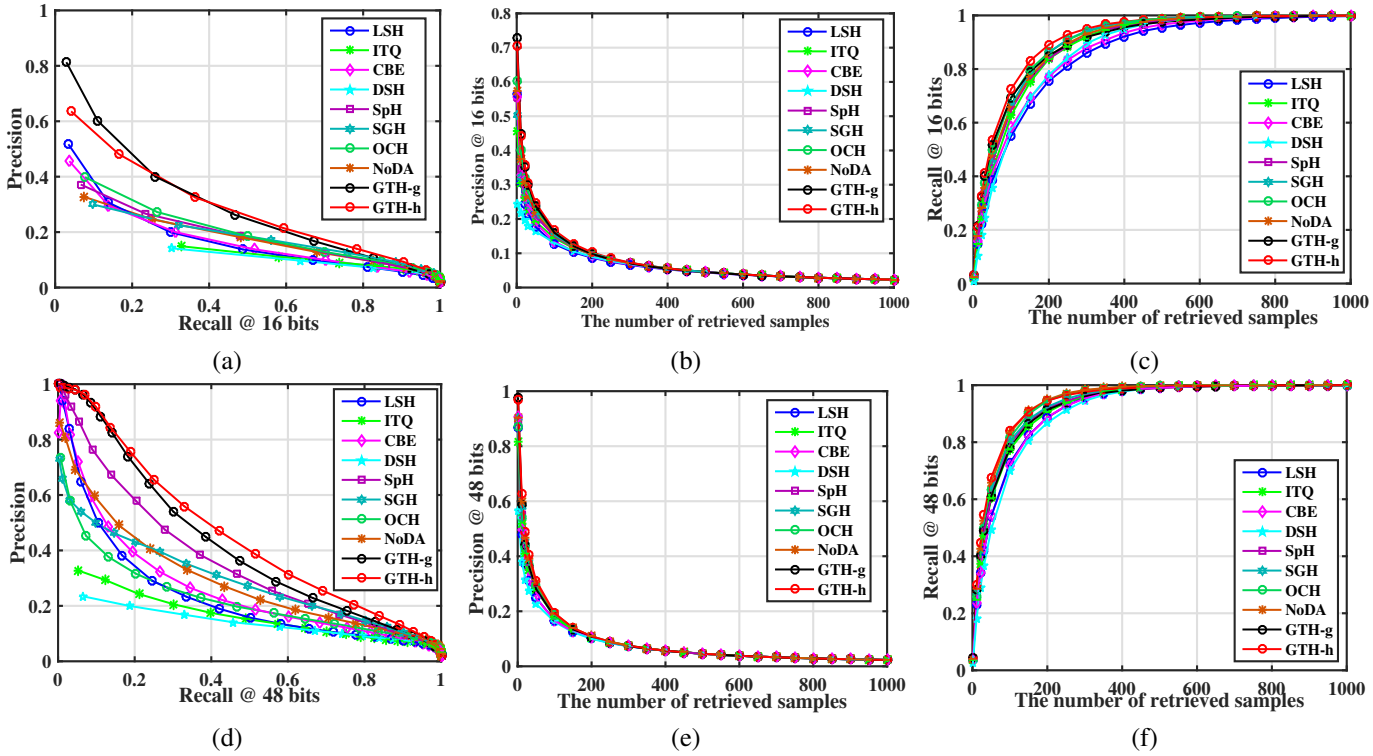


Fig. 3: Retrieval performance on PIE-C05&PIE-C29 datasets. (a) Precision and Recall curve @16 bit; (b) Precision @16 bit; (c) Recall@16 bit; (d) Precision and Recall curve @48 bit; (e) Precision @48 bit; (f) Recall@48 bit

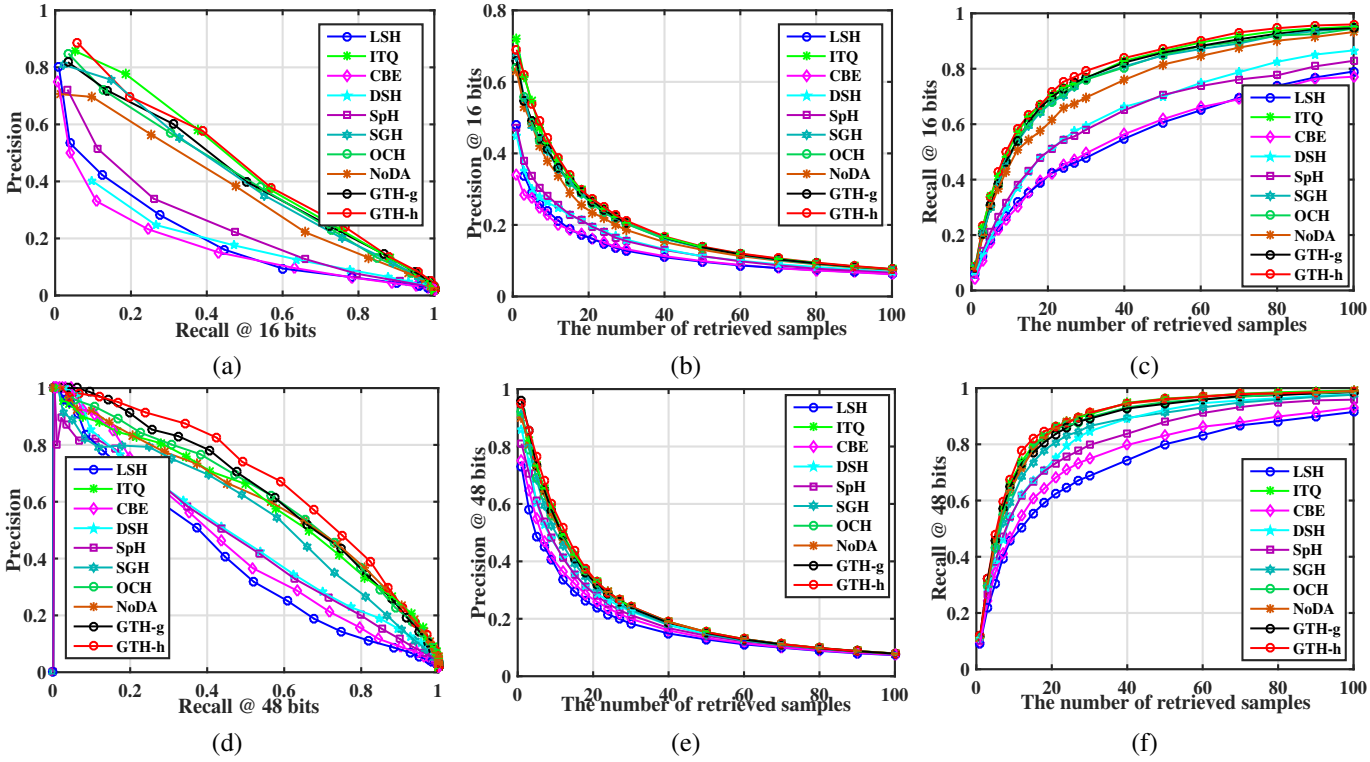


Fig. 4: Retrieval performance on Amazon&Dslr datasets. (a) Precision and Recall curve @16 bits; (b) Precision @16 bits; (c) Recall@16 bits; (d) Precision and Recall curve @48 bits; (e) Precision @48 bits; (f) Recall@48 bits.

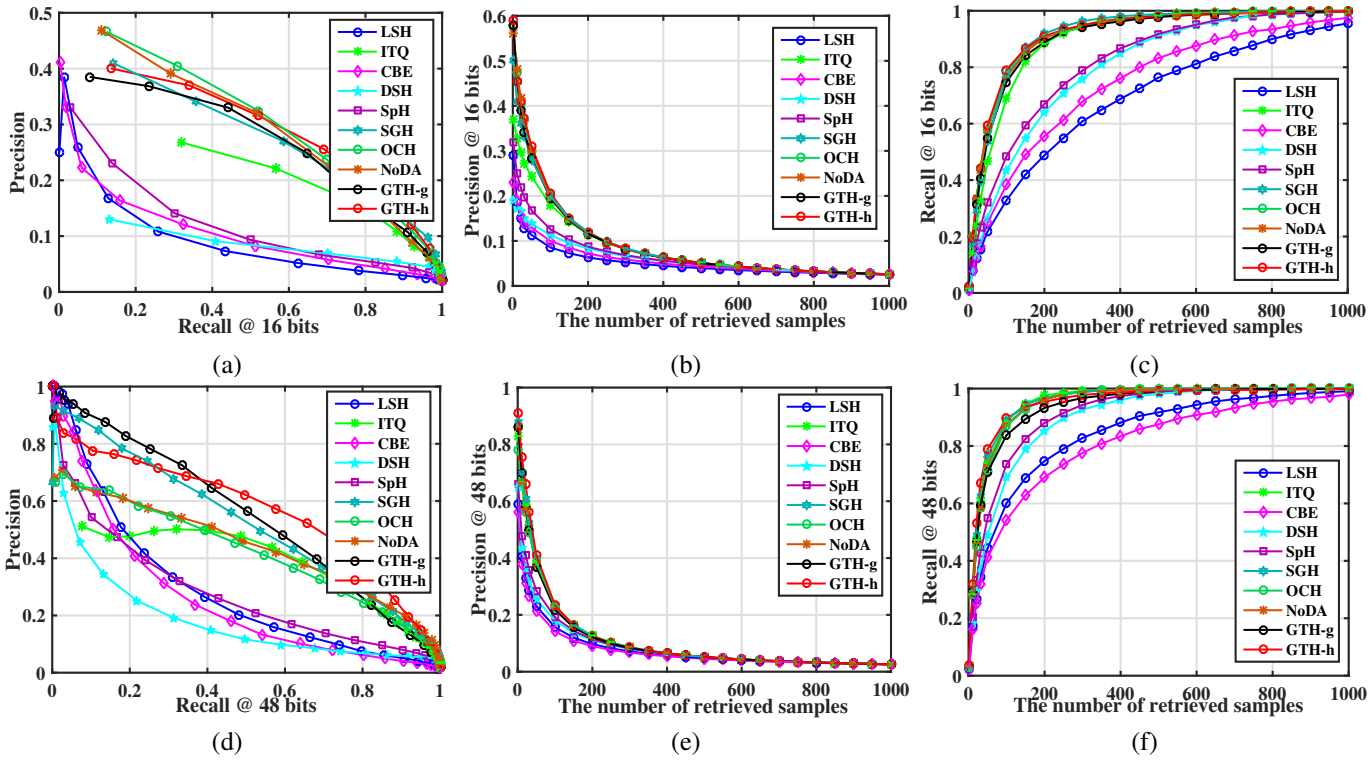
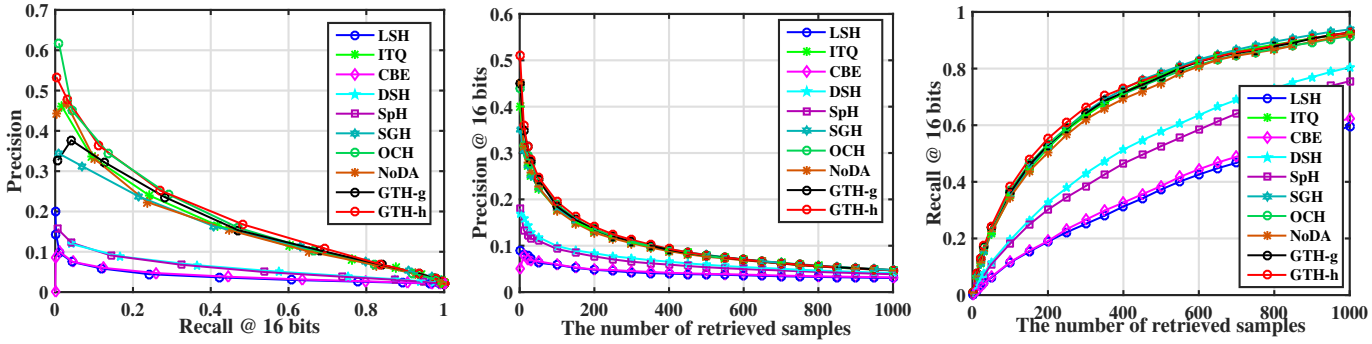
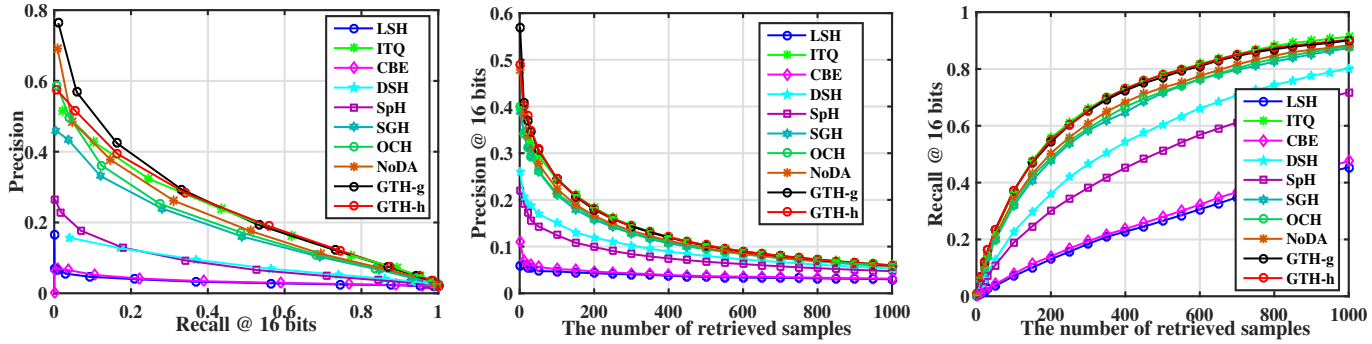


Fig. 5: Retrieval performance on VOC2007&Caltech101 datasets. (a) Precision and Recall curve @16 bits; (b) Precision @16 bits; (c) Recall@16 bits; (d) Precision and Recall curve @48 bits; (e) Precision @48 bits; (f) Recall@48 bits.



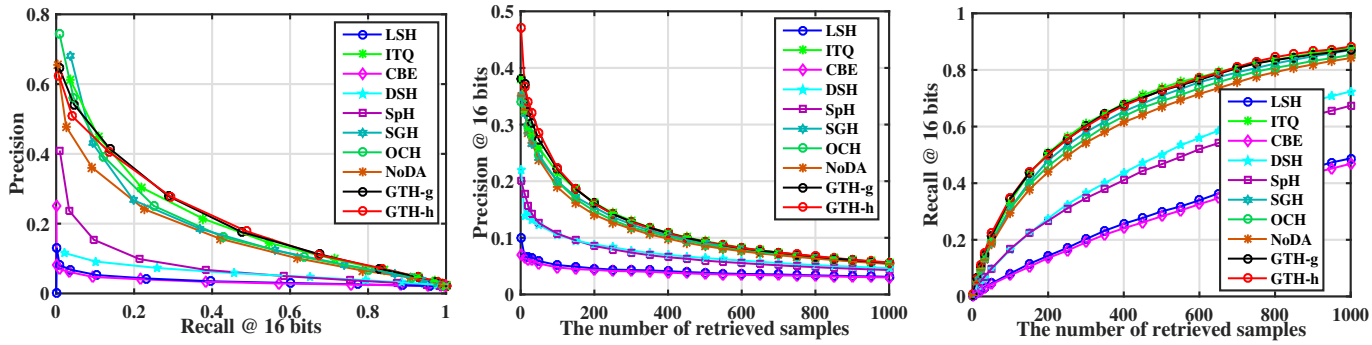
(a) (b) (c)

Fig. 6: Retrieval performance on ImageNet&LabelMe databases @16 bits. (a) Precision and Recall curve; (b) Precision with different number of retrieved samples; (c) Recall rate with different number of retrieved samples.



(a) (b) (c)

Fig. 7: Retrieval performance on ImageNet&VOC2007 databases @16 bits. (a) Precision and Recall curve; (b) Precision with different number of retrieved samples; (c) Recall rate with different number of retrieved samples.

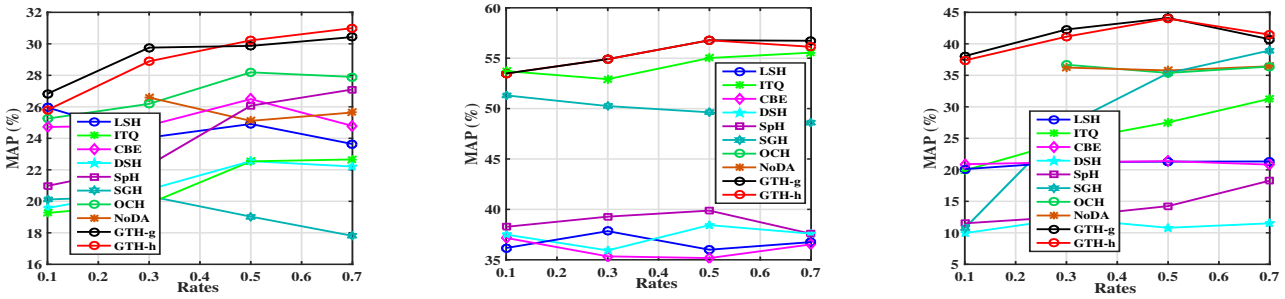


(a) (b) (c)

Fig. 8: Retrieval performance on ImageNet&SUN09 databases @16 bits. (a) Precision and Recall curve; (b) Precision with different number of retrieved samples; (c) Recall rate with different number of retrieved samples.

6) Results on ImageNet&SUN09 Dataset: On ImageNet&SUN09 dataset, our GTH achieves competitive MAP scores. We give the PR-curve, Precision and Recall for ImageNet&SUN09 dataset in Fig. 8. The code length is set to 16. From the subfigures, we can observe that our GTH always presents more competitive retrieval performance compared to the baselines. The effectiveness of our GTH is verified, by transfer hashing from source to target domain. From Table III and IV, we could find that the traditional hashing models show less improvement by comparing the performance with source data and that without source data during training process. This is due to that the models lack of transferability and the domain discrepancy can produces negative effect, which therefore demonstrates the necessity of inducing generalized hashing from the perspective of transfer learning and domain adaptation.

The above experimental results demonstrate the effectiveness of our GTH model. Our GTH is more suitable to the scenario where there are no enough training images used to learn precise hashing codes on the domain of interest.



(a) (b) (c)

Fig. 9: MAP scores @32 bits with the varying number of training images in target domain. (a) PIE-C05&PIE-C29; (b) Amazon&Dslr; (c) VOC2007&Caltech101. Note that the x-axis denotes the ratio rate, i.e., 0.1, 0.3, 0.5 and 0.7, which demonstrates the proportion of the selected training samples from the target training set.

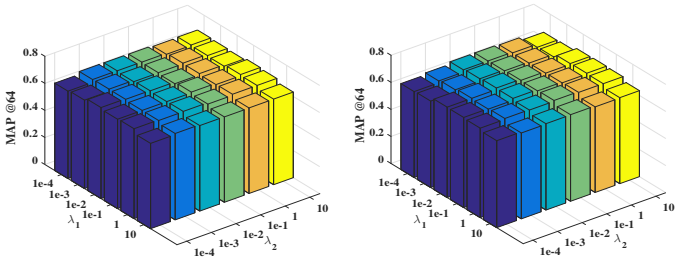
Comparing to the ITQ+ based transfer hashing that constrains the number of source samples to be the same as that of target samples, our GTH is more flexible and free to the domain size. In model aspect, the proposed GTH is easy to understand and follow, but effective in retrieval tasks.

C. Performance variation w.r.t. target training samples

In order to further demonstrate the efficiency of our GTH by using less target training data, we use different numbers of training data on target domain to learn the hashing functions. Specially, we choose 10%, 30%, 50%, and 70% images from training data of target domain as training data. Then, in testing phase, we also use hashing codes of the testing queries to search the most similar hashing codes in the whole training samples. The experiments are conducted on PIE-C05&PIE-C29, Amazon&Dslr, and VOC2007&Caltech101 databases, respectively. The MAP scores of all the compared methods and our GTH are shown in Fig. 9. Due to the intrinsic limitation of OCH method with regard to the number of training data, there are some empty MAP scores in several cases. In this experiment, the code length is set as 32. It is worth noting that our GTH always outperforms all the compared methods, which further demonstrates the efficiency of our GTH when there are less target domain samples. The goal of our GTH for learning to hash based on scarcely labeled domain of interest is achieved, by leveraging an external source domain.

D. Parameters Sensitivity

In order to further investigate the properties of the proposed method, the retrieval performances versus the different values of regularization parameters, λ_1 and λ_2, are explicitly explored. To clearly show the results, we perform experiments on Amazon&Dslr databases to verify the parameters sensitivity. Specifically, we tune the value of both parameters from the pool $\{0.0001, 0.001, 0.01, 0.1, 1, 10\}$. The MAP scores with the code length set as 64 are shown in Fig. 10. We can observe that the performance of our GTH-g and GTH-h models are not very sensitive to the settings of λ_1 and λ_2. Apparently, when the parameters are not very large, the MAP scores of our methods can not be severely influenced. This also demonstrates that both regularization terms are indispensable



(a) (b)

Fig. 10: Parameters sensitivity analysis based on Amazon&Dslr dataset. (a) GTH-g; (b) GTH-h.

for superior performance. Overall, the proposed models are not sensitive to the parameters in a reasonable range.

E. Computational Time Analysis

The computation complexity of GTH is presented in Section III. This section presents the computation time analysis by comparing to other baselines in Table V. We see that GTH shows competitive efficiency in training stage. Experiments are run on MATLAB operated on a computer with CPU E3-1231 v3 3.40GHz and 16G RAM.

F. Convergence Analysis

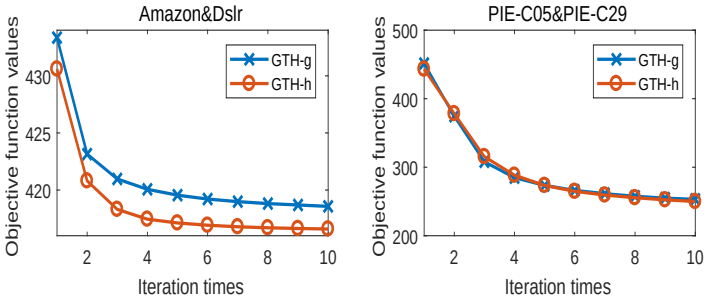
The proposed GTH is solved with a variable alternating strategy, and the convergency can be guaranteed. We present the convergence curves of the objective function in Fig. 11, from which we see that GTH can quickly converge to an optimal solution within several iterations.

V. Conclusion

In this paper, we propose a simple but effective transfer hashing method named Optimal Projection Guided Transfer Hashing (GTH). Inspired by transfer learning, we propose to borrow the knowledge from a semantic related but distribution different auxiliary domain (i.e., source domain). We assume that the semantic similar images between target and source

TABLE V: Computation time (s) of all methods @64 bits on PIE-C05&PIE-C29 task.

Method	LSH	ITQ	CBE	DSH	SpH	SGH	OCH	ITQ+	LapITQ+	GTH-g	GTH-h
Time (s)	**0.01**	0.60	0.16	0.16	3.82	15.02	1.92	1.45	9.12	2.77	2.66



(a) (b)

Fig. 11: Convergence analysis of GTH @64 bits based on Amazon&Dslr (a) and PIE-C05&PIE-C29 (b) tasks.

domains should have small discrepancy between their domain-specific hashing projections. Therefore, we let the projections of target and source domain close to each other, so that the similar instances between two domains can be imposed with similar hashing codes. We propose the GTH model from the viewpoint of maximum likelihood estimation (MSE) in this paper, and an iteratively weighted l_2 loss modeling the error between the hashing projections of source and target domains is proposed. The error modeling can well reduce the bias of hashing functions between source and target domain, which makes our GTH more adaptive to cross-domain case. Extensive experiments on four groups of benchmark databases with 6 tasks are conducted. The experimental results show that our GTH always outperforms other baselines in retrieval performance when there are much less target samples, and the superiority of GTH over many state-of-the-art learning to hash methods is validated.

Most of existing research focus on close-set retrieval, i.e. the query is included in the gallery set, which has a distance to open-set retrieval. In our future work, a new challenge for cross-domain open-world retrieval instead of cross-modal retrieval is an interesting but challenging research direction and have to be addressed. The combination of transfer learning, domain adaptation and learning to hashing can be a potential for this challenge.

APPENDIX A
PROOF OF DEGENERATION FROM GTH-H TO GTH-G

As stated in Section III, if we let $\omega_\theta(\widetilde{E}_{ij}) = 2$, then Eq. (9) (GTH-h) is degenerated into Eq. (1) (GTH-g), which assumes that the errors obey Gaussian distribution. We present the detailed mathematical proof as follows.

We define that $\mathbf{E} = \mathbf{W}_t - \mathbf{W}_s$ represents the error matrix, E_{ij} is the element in the error matrix. Without loss of generality, we let $\mathbf{e} = [E_{11}, E_{21}, \cdots, E_{d1}, \cdots, E_{1r}, E_{2r}, \cdots, E_{dr}]^{\mathrm{T}}$, $\mathbf{w_t} = [W^t_{11}, W^t_{21}, \cdots, W^t_{d1}, \cdots, W^t_{1r}, W^t_{2r}, \cdots, W^t_{dr}]^{\mathrm{T}}$, and $\mathbf{w_s} = [W^s_{11}, W^s_{21}, \cdots, W^s_{d1}, \cdots, W^s_{1r}, W^s_{2r}, \cdots, W^s_{dr}]^{\mathrm{T}}$. Assume that $e_1, e_2, \cdots, e_N$ are independently and identically distributed, with the the following Gaussian probability density function $f_\theta(e_n)$, shown by

$$f_\theta(e_n) = \frac{1}{\sqrt{2\pi}\delta} exp(-\frac{e_n^2}{2\delta^2}) \tag{23}$$

The Eq. (23) can also be written as

$$f_\theta(e_n) = \frac{1}{\sqrt{2\pi}\delta} exp\big(-\frac{(w_t^n - w_s^n)^2}{2\delta^2}\big) \tag{24}$$

According to the maximum likelihood estimation (MLE), the following joint likelihood function is presented as

$$\begin{aligned} L(\mathbf{w_t}, \mathbf{w_s}) &= \prod_{n=1}^{N} f_\theta(e_n) \\ &= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\delta} exp\big(-\frac{(w_t^n - w_s^n)^2}{2\delta^2}\big) \end{aligned} \tag{25}$$

For computing the MLE solution, the negative log-likelihood function of Eq. (25) is written as

$$\begin{aligned} &-\log L(\mathbf{w_t}, \mathbf{w_s}) \\ &= -\log \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi}\delta} exp\big(-\frac{(w_t^n - w_s^n)^2}{2\delta^2}\big) \\ &= -\sum_{n=1}^{N} \log \frac{1}{\sqrt{2\pi}\delta} exp\big(-\frac{(w_t^n - w_s^n)^2}{2\delta^2}\big) \\ &= \frac{1}{2\delta^2}\sum_{n=1}^{N}(w_t^n - w_s^n)^2 - N\log\frac{1}{\sqrt{2\pi}\delta} \end{aligned} \tag{26}$$

Maximizing the likelihood function is amount to minimizing the negative log likelihood function Eq. (26). Then, the MLE problem in Eq. (25) can be transformed into

$$\min_{w_t, w_s} \sum_{n=1}^{N}(w_t^n - w_s^n)^2 \tag{27}$$

Finally, the model (27) can be reformulated as Eq. (1) in matrix form, which is a degenerated case of Eq. (9). The proof that GTH-g is a special case of GTH-h is done.

ACKNOWLEDGEMENT

We would like to thank Fangyi Liu and Shanshan Wang for their help in coding for partial comparison experiments. We would also like to thank the associate editor and reviewers for their comments in improving the quality of this work.

REFERENCES

[1] G.-J. Qi and J. Luo, "Small data challenges in big data era: A survey of recent progress on unsupervised and semi-supervised methods," *arXiv:1903.11260*, 2019.

[2] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2018.

[3] J. Tang, Z. Li, M. Wang, and R. Zhao, "Neighborhood discriminant hashing for large-scale image retrieval." *IEEE Trans. Image Processing*, vol. 24, no. 9, pp. 2827–2840, 2015.

[4] J. Wang, W. Liu, S. Kumar, and S. Chang, "Learning to hash for indexing big datała survey," *Proceedings of the IEEE*, vol. 104, no. 1, pp. 34–57, 2016.

[5] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[6] M. Norouzi, D. J. Fleet, and R. Salakhutdinov, "Hamming distance metric learning," in *NIPS*, 2012, pp. 1061–1069.

[7] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *International Conference on Very Large Data Bases*, 1999, pp. 518–529.

[8] P. Indyk, "A small approximately min-wise independent family of hash functions," *Journal of Algorithms*, vol. 38, no. 1, pp. 84–90, 2001.

[9] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *NIPS*, 2008, pp. 1753–1760.

[10] W. Liu, J. Wang, S. Kumar, and S. F. Chang, "Hashing with graphs," in *ICML*, 2011, pp. 1–8.

[11] Z. Jin, C. Li, Y. Lin, and D. Cai, "Density sensitive hashing," *IEEE Trans. Cybernetics*, vol. 44, no. 8, pp. 1362–1371, 2014.

[12] F. Yu, S. Kumar, Y. Gong, and S.-F. Chang, "Circulant binary embedding," in *ICML*, 2014, pp. 946–954.

[13] Q. Y. Jiang and W. J. Li, "Scalable graph hashing with feature transformation," in *International Conference on Artificial Intelligence*, 2015, pp. 2248–2254.

[14] H. Liu, R. Ji, J. Wang, and C. Shen, "Ordinal constraint binary coding for approximate nearest neighbor search," *IEEE Trans. Pattern Analysis & Machine Intelligence*, vol. PP, no. 99, pp. 1–1, 2018.

[15] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "Ldahash: Improved matching with smaller descriptors." *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 34, no. 1, pp. 66–78, 2011.

[16] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *ICML*, 2011, pp. 353–360.

[17] G. Lin, C. Shen, Q. Shi, A. V. D. Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *CVPR*, 2014, pp. 1971–1978.

[18] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *CVPR*, 2012, pp. 2074–2081.

[19] F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *CVPR*, 2015, pp. 37–45.

[20] X. Shi, F. Xing, J. Cai, Z. Zhang, Y. Xie, and L. Yang, "Kernel-based supervised discrete hashing for image retrieval," in *ECCV*. Springer, 2016, pp. 419–433.

[21] X. Wang, T. Zhang, G. J. Qi, J. Tang, and J. Wang, "Supervised quantization for similarity search," in *CVPR*, 2016, pp. 2018–2026.

[22] T.-T. Do, A.-D. Doan, and N.-M. Cheung, "Learning to hash with binary deep neural network," in *ECCV*, 2016, pp. 219–234.

[23] M. Yang, L. Zhang, J. Yang, and D. Zhang, "Robust sparse coding for face recognition," pp. 625–632, 2011.

[24] J. Liu and L. Zhang, "Optimal projection guided transfer hashing for image retrieval," in *AAAI*, 2019.

[25] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," *Journal of Computer & System Sciences*, vol. 60, no. 3, pp. 630–659, 2000.

[26] B. Chen and A. Shrivastava, "Revisiting winner take all (wta) hashing for sparse datasets," *arXiv*, 2016.

[27] J. Wang, W. Liu, A. X. Sun, and Y. G. Jiang, "Learning hash codes with listwise supervision," in *ICCV*, 2013, pp. 3032–3039.

[28] T.-T. Do, K. Le, T. Hoang, H. Le, T. V. Nguyen, and N.-M. Cheung, "Simultaneous feature aggregating and hashing for compact binary code learning," *IEEE Trans. Image Processing*, 2019.

[29] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H.-T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 3034–3044, 2018.

[30] L. Jin, X. Shu, K. Li, Z. Li, G.-J. Qi, and J. Tang, "Deep ordinal hashing with spatial attention," *IEEE Trans. Image Processing*, vol. 28, no. 5, pp. 2173–2186, 2019.

[31] J.-T. Zhou, X. Xu, S.-J. Pan, I. W. Tsang, Z. Qin, and R.-S.-M. Goh, "Transfer hashing with privileged information," in *IJCAI*, 2016.

[32] J. T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, and R. S. M. Goh, "Transfer hashing: From shallow to deep," *IEEE Trans. Neural Networks and Learning Systems*, 2018.

[33] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, "Deep hashing network for unsupervised domain adaptation," in *CVPR*, 2017, pp. 5385–5394.

[34] L. Zhu, Z. Huang, L. Xie, and H.-T. Shen, "Exploring auxiliary context: Discrete semantic transfer hashing for scalable image retrieval," *IEEE Trans. Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5264–5276, 2018.

[35] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[36] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *AAAI*, 2014, pp. 2213–2219.

[37] X. Yang, M. Wang, R. Hong, Q. Tian, and Y. Rui, "Enhancing person re-identification in a self-trained subspace," *ACM Trans. Multimedia Computing, Communications, and Applications*, vol. 13, no. 3, p. 27, 2017.

[38] L. Zhang and D. Zhang, "Robust visual knowledge transfer via extreme learning machine based domain adaptation," *IEEE Trans. Image Processing*, vol. 25, no. 10, pp. 4959–4973, 2016.

[39] Y. Xu, Y. Yang, F. Shen, X. Xu, Y. Zhou, and H. T. Shen, "Attribute hashing for zero-shot image retrieval," in *ICME*, 2017, pp. 133–138.

[40] Y. Yang, Y. Luo, W. Chen, F. Shen, J. Shao, and H. T. Shen, "Zero-shot hashing via transferring supervised knowledge," in *ACM MM*, 2016, pp. 1286–1295.

[41] S. Wang, L. Zhang, and W. Zuo, "Class-specific reconstruction transfer learning via sparse low-rank constraint," in *ICCVW*, 2017, pp. 949–957.

[42] L. Zhang, S. Wang, G.-B. Huang, W. Zuo, J. Yang, and D. Zhang, "Manifold criterion guided transfer learning via intermediate domain generation," *IEEE Trans. Neural Networks and Learning Systems, DOI: 10.1109/TNNLS.2019.2899037*, 2019.

[43] L. Zhang, W. Zuo, and D. Zhang, "Lsdt: Latent sparse domain transfer learning for visual adaptation," *IEEE Trans. Image Process*, vol. 25, no. 3, pp. 1177–1191, 2016.

[44] B. Wang, Y. Yang, X. Xu, A. Hanjalic, and H. T. Shen, "Adversarial cross-modal retrieval," in *ACM MM*, 2017, pp. 154–162.

[45] X. Shu, G.-J. Qi, J. Tang, and J. Wang, "Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation," in *ACM MM*, 2015, pp. 35–44.

[46] J. Tang, X. Shu, Z. Li, G.-J. Qi, and J. Wang, "Generalized deep transfer networks for knowledge propagation in heterogeneous domains," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 12, no. 4s, pp. 68:1–68:22, 2016.

[47] X. Shu, J. Tang, Z. Li, H. Lai, L. Zhang, and S. Yan, "Personalized age progression with bi-level aging dictionary learning," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 905–917, 2018.

[48] J.-T. Zhou, I. W. Tsang, and S. J. Pan, "Multi-class heterogeneous domain adaptation," *Journal of Machine Learning Research*, vol. 20, pp. 1–31, 2019.

[49] L. Zhang, "Transfer adaptation learning: A decade survey," *arXiv:1903.04687*, 2019.

[50] J. Zhang, R. Jin, Y. Yang, and A. G. Hauptmann, "Modified logistic regression: an approximation to svm and its applications in large-scale text categorization," in *ICML*, 2003, pp. 888–895.

[51] S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *CVPR*, 2011, pp. 817–824.

[52] Z. Wen and W. Yin, "A feasible method for optimization with orthogonality constraints," *Mathematical Programming*, vol. 142, no. 1-2, pp. 397–434, 2013.

[53] A. H. Armstrong, "Numerical solution of partial differential equations. by smith g. d. . pp. viii, 179. 25s. 1965. (oxford university press)," *Mathematical Gazette*, vol. 50, no. 374, pp. 179–449, 2005.

[54] T. Sim, S. Baker, and M. Bsat, "The cmu pose, illumination, and expression (pie) database," in *AFGR*, 2002, pp. 46–51.

[55] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, "Adapting visual category models to new domains," 2010.

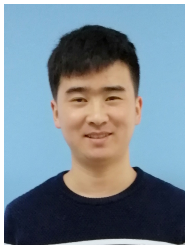
[56] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *CVPR*, 2011, pp. 1521–1528.

[57] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "Imagenet: A large-scale hierarchical image database," in *CVPR*, 2009, pp. 248–255.

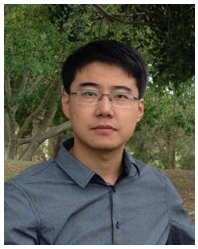
[58] J. P. Heo, Y. Lee, J. He, S. F. Chang, and S. E. Yoon, "Spherical hashing: binary code embedding with hyperspheres," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 37, no. 11, pp. 2304–2316, 2015.

[59] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: a deep convolutional activation feature for generic visual recognition," in *ICML*, 2013.

Lei Zhang (M'14-SM'18) received his Ph.D degree in Circuits and Systems from the College of Communication Engineering, Chongqing University, Chongqing, China, in 2013. He worked as a Post-Doctoral Fellow with The Hong Kong Polytechnic University, Hong Kong, from 2013 to 2015. He is currently a Professor/Distinguished Research Fellow with Chongqing University. He has authored more than 90 scientific papers in top journals, such as IEEE T-NNLS, IEEE T-IP, IEEE T-MM, IEEE T-IM, IEEE T-SMCA, and top conferences such as ICCV, AAAI, ACM MM, ACCV, etc. His current research interests include machine learning, pattern recognition, computer vision and intelligent systems. Dr. Zhang was a recipient of the Best Paper Award of CCBR2017, the Outstanding Reviewer Award of many journals such as Pattern Recognition, Neurocomputing, Information Sciences, etc., Outstanding Doctoral Dissertation Award of Chongqing, China, in 2015, Hong Kong Scholar Award in 2014, Academy Award for Youth Innovation in 2013 and the New Academic Researcher Award for Doctoral Candidates from the Ministry of Education, China, in 2012. He is a Senior Member of IEEE.



Ji Liu received his Bachelor degree in Lamps and Lighting in 2016 from Hubei Engineering University, China. Since September 2016, he is currently studying for his Master degree in Chongqing University. His current research interests include machine learning and image retrieval.



Yang Yang Yang Yang received the bachelors degree from the Jilin University, and the masters degree from the Peking University, in 2006 and 2009 respectively, and the PhD degree from the University of Queensland, Australia, in 2012. He is currently with the University of Electronic Science and Technology of China. He was a research fellow under the supervision of Prof. Tat-Seng Chua in National University of Singapore during 2012-2014. During the PhD study, Yang Yang was supervised by Prof. Heng Tao Shen and Prof. Xiaofang Zhou.

Fuxiang Huang received her Bachelor degree in Applied Electronic Technology Education in 2018 from China West Normal University. Since September 2018, she is currently studying for her Master degree in Chongqing University. Her current research interests include machine learning to hashing and image retrieval.

Feiping Nie received the Ph.D. degree in computer science from Tsinghua University, Beijing, China, in 2009. He is currently a Professor with the Center for Optical Imagery Analysis and Learning, Northwestern Polytechnical University, China. His research interests are machine learning and its applications fields, such as pattern recognition, data mining, computer vision, image processing, and information retrieval. He currently serves as an Associate Editor or a PC Member for several prestigious journals and conferences in the related fields.

David Zhang (F'09) graduated in Computer Science from Peking University in 1974. He received his MSc in 1982 and his PhD in 1985 in Computer Science from the Harbin Institute of Technology (HIT), respectively. From 1986 to 1988 he was a Postdoctoral Fellow at Tsinghua University and then an Associate Professor at the Academia Sinica, Beijing. In 1994 he received his second PhD in Electrical and Computer Engineering from the University of Waterloo, Ontario, Canada. He is a Chair Professor since 2005 at the Hong Kong Polytechnic University where he is the Founding Director of the Biometrics Research Centre (UGC/CRC) supported by the Hong Kong SAR Government in 1998. He also serves as Visiting Chair Professor in Tsinghua University, and Adjunct Professor in Peking University, Shanghai Jiao Tong University, HIT, and the University of Waterloo. He is the Founder and Editor-in-Chief, International Journal of Image and Graphics (IJIG); Book Editor, Springer International Series on Biometrics (KISB); Organizer, the International Conference on Biometrics Authentication (ICBA); Associate Editor of more than ten international journals including IEEE TRANSACTIONS and so on; and the author of more than 10 books, over 300 international journal papers and 30 patents from USA/Japan/HK/China. Professor Zhang is a Croucher Senior Research Fellow, Distinguished Speaker of the IEEE Computer Society, and a Fellow of both IEEE and IAPR.