Transfer Hashing: From Shallow to Deep

Joey Tianyi Zhou, Heng Zhao, Xi Peng, Meng Fang, Zheng Qin, and Rick Siow Mong Goh

***Abstract*— One major assumption used in most existing hashing approaches is that the domain of interest (i.e., the target domain) could provide sufficient training data, either labeled or unlabeled. However, this assumption may be violated in practice. To address this so-called data sparsity issue in hashing, a new framework termed transfer hashing with privileged information (THPI) is proposed, which marriages hashing and transfer learning (TL). To show the efficacy of THPI, we propose three variants of the well-known iterative quantization (ITQ) as a showcase. The proposed methods, ITQ+, LapITQ+, and deep transfer hashing (DTH), solve the aforementioned data sparsity issue from different aspects. Specifically, ITQ+ is a shallow model, which makes ITQ achieve hashing in a TL manner. ITQ+ learns a new slack function from the source domain to approximate the quantization error on the target domain given by ITQ. To further improve the performance of ITQ+, LapITQ+ is proposed by embedding the geometric relationship of the source domain into the target domain. Moreover, DTH is proposed to show the generality of our framework by utilizing the powerful representative capacity of deep learning. To the best of our knowledge, this could be one of the first DTH works. Extensive experiments on several popular data sets demonstrate the effectiveness of our shallow and DTH approaches comparing with several state-of-the-art hashing approaches.**

***Index Terms*— Deep transfer hashing (DTH), hashing, privileged information, transfer learning (TL).**

I. Introduction

HASHING is an efficient similarity searching method that has been successfully applied in many applications [1]–[6]. Hashing aims to design or learn a compact binary code for each data instance such that the similar/dissimilar instances in the original space are mapped to similar/dissimilar binary codes. In consequence, the cost of data storage can be largely reduced, thus could efficiently computing the similarity between instances with the hamming distance using binary operation (XOR).

Most existing data-dependent hashing methods require a large amount of data to learn a set of hash functions to construct binary codes [1]. However, in some scenarios, the data for the domain of interest (target domain) are likely to be insufficient to learn a precise hashing model. Such a data sparsity issue will limit the applications of hashing in many real-world scenarios. For example, taobao.com is an online platform for small businesses where individual shopkeepers can open their own shops each of which has to build a hashing system for potential customers to retrieve the sold products. Unfortunately, each individual shop generally has not enough images to build a good enough hashing system. A straightforward solution is augmenting data by crawling images of the same product from other websites such as amazon.com. However, these two data sources (taobao and amazon) are largely different. The images in taobao.com are usually amateur taken by shopkeepers. In contrast, the illustrations in amazon.com are usually taken by professional photographers. Furthermore, it is hard to crawling images of the exact object from different data sources. In consequence, a simple accumulation of these two image sources generally cannot give a desirable hashing system.

To solve the above-mentioned challenges, we introduce transfer learning (TL) [7] into hashing. We aim at extracting knowledge rather than simply accumulating raw data from auxiliary data sources, and then exploiting the knowledge to learn a hashing system for handling unobserved data from the target domain. To the best of our knowledge, the proposed framework, termed "transfer hashing with privileged information (THPI)," could be *one of the first transfer hashing works*. The concept of "privileged information" was proposed by Vapnik and Vashist [8], which is defined as the information $\tilde{\mathbf{x}}$ (related to the input $\mathbf{x}$) given by a teacher during training. In general, the privileged information is denoted by the pairwise correlation between the source and the target domain. Considering the aforementioned example, we aim to utilize the privileged information from a source domain (amazon) to obtain hash codes for a target domain instances (taobao). It should be noted that the privileged information can also be in different feature spaces. Considering one image contains multiple captions, we refer all captions of an image as the privileged information. To intuitively demonstrate the privileged information, we give an illustration example in Fig. 1 with a detailed descriptions.

The proposed THPI framework is different from existing works. First, THPI is a hashing approach, whereas learning using privileged information (LUPI) was proposed for classification. THPI extends LUPI into the scenario of hashing, and shows a feasible way to solve the data sparsity issue in hashing with the privileged information. Second, THPI is different from cross-modal hashing [9], [10]. The former aims

Manuscript received July 31, 2017; revised January 3, 2018; accepted April 9, 2018. This work was supported in part by the National Nature Science Foundation of China under Grant 61432012 and Grant U1435213, and in part by the Fundamental Research Funds for the Central Universities under Grant YJ201748. *(Corresponding author: Xi Peng.)*

J. T. Zhou, H. Zhao, Z. Qin, and R. S. M. Goh are with Institute of High Performance Computing, A*STAR, Singapore 138632 (e-mail: joey.tianyi.zhou@gmail.com; azurerain7@gmail.com; Qinz@ihpc.a-star.edu.sg; Gohsm@ihpc.a-star.edu.sg).

X. Peng is with the College of Computer Science, Sichuan University, Chengdu 610065, China (e-mail: pengx.gm@gmail.com).

M. Fang is with the Tencent AI Lab, Shenzhen 518057, China (e-mail: mfang@tencent.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNNLS.2018.2827036

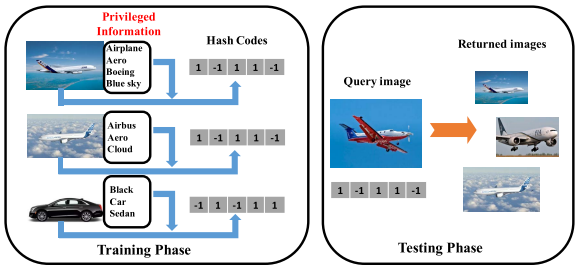


Fig. 1. THPI: the privileged information is generally defined as the pairwise correlation between the source and the target domain, which is only available during training. In the example, privileged information consists of the tags accompanied with images.

to address the data sparsity issue, whereas the latter assumes that the training data of different modalities are sufficient to learn reliable hash codes. Furthermore, given a query from one modality, cross-modal hashing aims to retrieve relevant data from other modalities; whereas THPI aims to utilize the privileged information to learn a good hashing model on the domain of interest.

To show the effectiveness of THPI, we propose three variants of the well-known iterative quantization (ITQ) [11] as showcases, i.e., ITQ+, LapITQ+, and deep transfer hashing (DTH). ITQ+ is a transfer version of ITQ, which makes the latter handling the data sparsity issue possible. More specifically, ITQ+ learns a new slack function from auxiliary data to approximate the quantization error given by ITQ. LapITQ+ further improves the performance of ITQ+ by using the local geometric property as an invariance and preserving it from the source domain into the target domain. Comparing with ITQ and LapITQ+, DTH is a deep hashing (DH) method that provides a feasible way to make transfer hashing benefiting from deep learning.

The paper is a substantial extension of our conference work [12] with following improvements. First, we propose a new algorithm (i.e., DTH) to show the generality of our framework. With the proposed three methods, we show that the privileged information is useful to address the data sparsity issue in either shallow or DH. It should be pointed out that it is nontrivial to develop our method from shallow to deep. To the best of our knowledge, DTH could be one of the first DH approaches. Second, we carry out more experimental analysis involving two new recently proposed deep learning-based hashing methods. The contribution and novelty of this paper are summarized as follows:

1) To address data sparsity issue on a target domain, we propose a novel framework termed THPI by transferring knowledge from the source domain into the target domain. To the best of our knowledge, it could be one of the first transfer hashing works.
2) Together with a novel slack function, we propose a new algorithm termed ITQ+ by incorporating the privileged information from the source domain into the target domain to assist hashing.
3) Based on ITQ+, LapITQ+ is proposed by embedding the underlying graph structure from the source domain into the target domain. LapITQ+ could give better hash codes thanks to the preservation of local geometric relationship.
4) To utilize the powerful representative capacity of DNNs, we further extend our method from shallow to deep model. In other words, we show a feasible way to incorporating advantages of deep learning and hashing.

II. Related Work

Our work is highly related to the following topics including learning to hash, TL, learning with privileged information, and DH. In this section, we give a brief discussion.

A. Learning to Hash

Existing hashing approaches can be grouped into two categories, namely, data-independent and data-dependent fashion. One typical data-independent method is locality-sensitive hashing (LSH) [13], which performs a set of random projections followed by thresholding.

Alternatively, the data-dependent methods learn discrete hash codes through minimizing the quantization error, which include spectral hashing (SH) [14], ITQ [11], and so on. The major difference among these methods lies in the ways of quantizing data. For example, SH considers the graph structure of data and reformulates the discrete quantization into the spectral graph partitioning [14] such that the graph geometry on the hash space resembles the original feature space. ITQ [11] reduces quantization error between the inputs and the hashing codes by refining initial projections.

One disadvantage of the data-dependent methods is requiring sufficient data to learn hashing functions for the target domain. This makes these methods failure in the case of encountering the data sparsity issue as the aforementioned. To address this problem, we propose THPI to improve the hashing performance by exploring and utilizing the knowledge from the source domains and the knowledge could be regarded as heterogeneous features.

Our proposed framework THPI is different from the cross-modal hashing [15], [16], which aims at learning binary codes from different modalities such that the information retrieval across modalities can be achieved. In addition, the performance of cross-modal hashing largely depends on available cross-domain correspondences, namely, it requires sufficient cross-domain correspondences to learn reliable hashing functions. In contrast, the proposed THPI performs hashing by exploiting all data from the source domain, thus could significantly alleviating the dependence of the correspondences between two domains. More recently, partial multi-modal hashing (PM^{2}H) [16] considers the situation with partial cross-domain correspondences and uses the graph to propagate the dependence of data points. Different from it, we only focus on improving the performance of the domain of interest rather than two domains.

B. Transfer Learning

TL [7] aims to transfer the knowledge from the source domain into the target domain, so that the rich source domain knowledge can be utilized to train a better classifier for the target domain. The transferred knowledge includes but

not limits to labels [17], [18], and cross-domain correspondences [19], [20]. Although TL has achieved huge success in many tasks, e.g., classification, regression, and clustering, few efforts have been devoted to developing transferable hashing approaches. To the best of our knowledge, only two works [21], [22] study this problem in recent. Comparing with [21], we focus on how to transfer knowledge across heterogeneous feature spaces in an unsupervised instead of supervised manner. Comparing with Domain Adaptive Hashing Networks [22], our proposed methods are more general and deal with knowledge transfer across heterogeneous domains, whereas [22] only deals with domain of same feature space.

C. Learning Using Privileged Information

LUPI was introduced by Vapnik and Vashist [8] Vapnik and Izmailov [8], [23], which uses the auxiliary privileged information to help training a better model with stronger generalization ability. It is noted that the privileged information will be unavailable in the testing phrase. Most existing LUPI-based works construct a correcting function to control the slack loss, as the traditional support vector machines (SVMs) does. Thus, these methods are termed SVM+. For a given training set $\{\mathbf{x}_i, \tilde{\mathbf{x}}_i\}_{i=1}^{l}$, SVM+ simultaneously learns a target classifier $f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x}$ and a slack approximation function $\epsilon(\tilde{\mathbf{x}}) = \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}$ from the original feature vectors and the privileged feature vectors, respectively. Here, $\tilde{\mathbf{x}}$ denotes the privileged feature of the original feature $\mathbf{x}$. Mathematically, SVM+ aims to solve

$$\begin{aligned} \min_{\mathbf{w},\tilde{\mathbf{w}},b} \ & \frac{1}{2}(\|\mathbf{w}\|^2 + \lambda\|\tilde{\mathbf{w}}\|^2) + C\sum_{i=1}^{l}\epsilon(\tilde{\mathbf{x}}) \\ \text{s.t. } & y_i\mathbf{w}^\top\mathbf{x}_i \geq 1 - \epsilon(\tilde{\mathbf{x}}_i), \vee i = 1, 2, \ldots, l \\ & \epsilon(\tilde{\mathbf{x}}) \geq 0, \vee i = 1, 2, \ldots, l \end{aligned}$$

Comparing with these LUPI works [24], [25], we aim to construct a slack function to address the data sparsity issue in hashing instead of classification. We believe that our work is complementary to LUPI in the scenario of hashing and unsupervised learning.

D. Deep Hashing

Deep neural networks (DNNs) [26], [27] has shown promising performance in learning features from scratch. In recent, there are emerging DH methods which integrates the deep learning into hashing for scalable image/document searching [2], [28], [29]. However, most of these works mainly focus on the traditional supervised hashing setting. For example, Lai *et al.* [30] proposed a supervised hashing method to jointly learn image representations and binary codes using a convolutional neural network (CNN). Semantic hashing [28] stacks a set of restricted Boltzmann machines to learn hash codes for document searching in unsupervised setting. DeepBit [31] was proposed specifically for image retrieval by using a CNN to learn binary descriptors in an unsupervised manner. Although, these works have achieved impressive performance with deep learning, they may suffer from following limitations. First, they usually ignore the data sparsity issue. If the target domain cannot provides sufficient data, these methods may achieve undesirable performance. Second, the methods such as DH cannot handle a very large scale problem since their loss is incompatible with stochastic gradient descent (SGD). To the best of our knowledge, there is no DH method proposed in the framework of transfer hashing.

III. Iterative Quantization With Privileged Information (ITQ+)

Let $\mathbf{X}_T = [\mathbf{x}_{T_1} \ldots, \mathbf{x}_{T_n}]^\top \in \mathbb{R}^{n\times d_T}$ be the collection of n given data points from the target domain with the dimension of d_T, hashing aims to learn a binary code matrix $\mathbf{B}_T \in \{-1, 1\}^{n\times c}$ for $\mathbf{X}_T$. Here, c denotes the length of each code. The key of hashing is learning a binary function $b_T^k = \text{sgn}(\mathbf{r}_T^{k\top}\mathbf{x}_T)$ for the target $\mathbf{x}_T$, where $\mathbf{r}_T^k \in \mathbb{R}^{d_T}$ is the hyperplane for the kth bit and $k \in \{1, \ldots, c\}$. Let $\mathbf{R}_T \in \mathbb{R}^{d_T\times c}$ denote the projection matrix, the hash code matrix could be computed by $\mathbf{B}_T = \text{sgn}(\mathbf{X}_T\mathbf{R}_T)$, where sgn is the sign function.

One major challenge of hashing is addressing the data sparsity issue, i.e., the observed data from the target domain are insufficient (n is small). To address this issue, various methods have been proposed but more efforts in this direction are still deserved. Recent developments in LUPI have proved that the amount of training data could be significantly reduced with the privileged information [8], [23], [32]. However, it still remains unknown whether LUPI is helpful to hashing and how to develop THPI. In this paper, we present a new learning-to-hash framework termed THPI. The proposed THPI shows a feasible way of exploiting the high-level idea of LUPI to address the data sparsity issue in hashing. Similar to LUPI, THPI assumes that the available data consist of insufficient observed data $\mathbf{x}_T$ from the target domain and sufficient data $\mathbf{x}_S \in \mathbb{R}^{d_S}$ from the source domain, as well as the pairwise correspondence between $\mathbf{x}_T$ and $\mathbf{x}_S$ (i.e., the privileged information). Formally, there are n data point pairs $\{(\mathbf{x}_{S_1}, \mathbf{x}_{T_1})(\mathbf{x}_{S_2}, \mathbf{x}_{T_2}), \ldots, (\mathbf{x}_{S_n}, \mathbf{x}_{T_n})\}$ for training. For simplicity, we denote $\mathbf{X}_{\text{SC}} = [\mathbf{x}_{S_1} \ldots \mathbf{x}_{S_n}]^\top \in \mathbb{R}^{n\times d_S}$ as the data matrix of n corresponding instances from the source domain, and $\mathbf{X}_{\text{SU}} = [\mathbf{x}_{S_{n+1}}, \mathbf{x}_{S_{n+2}}, \ldots, \mathbf{x}_{S_{n+n_S}}]^\top \in \mathbb{R}^{n_S\times d_S}$ as the matrix of the remaining n_S instances of the source domain.

A. Iterative Quantization

The ITQ algorithm [11] was proposed to learn hashing functions by minimizing the quantization error between the inputs and the binary codes. Specifically, it alternatively learns an orthogonal projection matrix $\mathbf{R}_T \in \mathbb{R}^{d_T\times c}$ and the code matrix $\mathbf{B}_T \in \{-1, 1\}^{n\times c}$ by optimizing the following problem:

$$\begin{aligned} \min_{\mathbf{B}_T,\mathbf{R}_T} \ & \|\mathbf{B}_T - \mathbf{X}_T\mathbf{R}_T\|_F^2 \\ \text{s.t. } & \mathbf{R}_T^\top\mathbf{R}_T = \mathbf{I} \end{aligned} \tag{1}$$

where the constraint is used to avoid trivial solutions.

B. Transferable Iterative Quantization (ITQ+)

Let $\mathbf{E}$ be the error matrix induced by the quantization process, we define it by $\mathbf{E} = \mathbf{B}_T - \mathbf{X}_T\mathbf{R}_T$. With the privileged data $\mathbf{X}_{\text{SC}} \in \mathbb{R}^{n\times c}$, we aim to learn a slack function $\mathbf{g}(\mathbf{X}_{\text{SC}}) = \mathbf{X}_{\text{SC}}\mathbf{P}$ to approximate the quantization error matrix $\mathbf{E}$, where

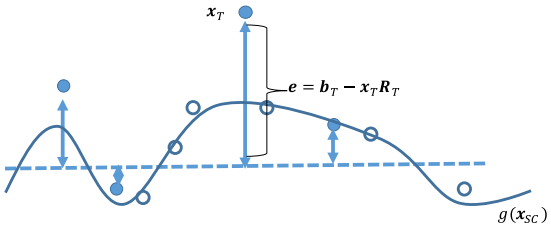


Fig. 2. Illustrative figure to show how the privileged information is used to model the quantization error, and thus enjoy better generalization ability. Solid dots: training data from the target domain. Hallowed dots: unseen data. The curve is the quantization error function constructed with privileged information from the source domain, which regularizes the quantization error.

$\mathbf{P} \in \mathbb{R}^{d_S \times c}$ denotes another orthogonal projection matrix to incorporate the privileged information from the source domain. With the above-mentioned definitions, the objective function of the proposed ITQ with privileged information (ITQ+) is as follows:

$$\min_{\mathbf{B}_T \in \mathbb{B}, \mathbf{R}_T, \mathbf{P}_T} \|\mathbf{E}\|_F^2 + \lambda_1 \|\mathbf{E} - \mathbf{g}(\mathbf{X}_{\mathrm{SC}})\|_F^2 \quad \text{s.t. } \mathbf{R}_T^\top \mathbf{R}_T = \mathbf{I}, \quad \text{and } \mathbf{P}^\top \mathbf{P} = \mathbf{I} \tag{2}$$

where $\lambda_1 > 0$ is a tradeoff parameter. In SVM+ [23], the slack variables is approximated by the privileged information, which can be regarded as tolerance functions to allow the margin constraints be violated. Different from [23], ITQ+ only borrows the high-level idea of LUPI to use the source domain information to approximate the target domain quantization error $\mathbf{E}$. Clearly, the motivation and objective of these two works are different. The constructed slack function also evaluates the difficulty in quantizing the target domain data with the privileged information from the source domain. Therefore, the constructed slack function can regularize the quantization error to avoid overfitting when the size of target domain training data is small. Fig. 2 illustrates how the privileged information is used to model the quantization error function, thus enjoying better generalization ability.

C. Optimization

To solve the optimization problem in (2), we alternatively optimize the binary code matrix $\mathbf{B}_T$, the projection matrix $\mathbf{R}_T$, and $\mathbf{P}$. The optimizing procedure is summarized in Algorithm 1, and the details are given in this section.

1) Update $\mathbf{B}_T$ by Fixing $\mathbf{R}_T$ and $\mathbf{P}$: With the fixed $\mathbf{R}_T$ and $\mathbf{P}$, we solve the following problem to obtain the binary code matrix $\mathbf{B}_T$:

$$\min_{\mathbf{B}_T \in \mathbb{B}} \|\mathbf{B}_T - \mathbf{X}_T \mathbf{R}_T\|_F^2 + \lambda_1 \|(\mathbf{B}_T - \mathbf{X}_T \mathbf{R}_T) - \mathbf{X}_{\mathrm{SC}} \mathbf{P}\|_F^2. \tag{3}$$

As $\mathbf{R}_T$ and $\mathbf{P}$ are fixed, we could rewrite (3) as follows:

$$\max_{\mathbf{B}_T \in \mathbb{B}} \mathrm{tr}\left(\mathbf{B}_T \left(\lambda_1 \mathbf{P}^\top \mathbf{X}_{\mathrm{SC}}^\top + (\lambda_1 + 1)\mathbf{R}_T^\top \mathbf{X}_T^\top\right)\right) \tag{4}$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a matrix.

The minimizer to (3) can be achieved by sorting the columns of $\mathbf{M}$ in descending order, where

$$\mathbf{M} = \lambda_1 \mathbf{P}^\top \mathbf{X}_{\mathrm{SC}}^\top + (\lambda_1 + 1)\mathbf{R}_T^\top \mathbf{X}_T^\top.$$

2) Update $\mathbf{R}_T$ by Fixing $\mathbf{P}$ and $\mathbf{B}_T$: With the fixed $\mathbf{B}_T$ and $\mathbf{P}$, the optimization problem with respect to $\mathbf{R}_T$ can be reformulated as follows:

$$\min_{\mathbf{R}_T} \|\mathbf{B}_T - \mathbf{X}_T \mathbf{R}_T\|_F^2 + \lambda_1 \|(\mathbf{B}_T - \mathbf{X}_T \mathbf{R}_T) - \mathbf{X}_{\mathrm{SC}} \mathbf{P}\|_F^2 \quad \text{s.t. } \mathbf{R}_T^\top \mathbf{R}_T = \mathbf{I}. \tag{5}$$

The above-mentioned optimization problem could be further reduced to

$$\min_{\mathbf{R}_T} \left\| \mathbf{X}_T \mathbf{R}_T - \left(\mathbf{B}_T - \frac{\lambda_1}{\lambda_1 + 1}(\mathbf{X}_{\mathrm{SC}} \mathbf{P})\right) \right\|_F^2 \quad \text{s.t. } \mathbf{R}_T^\top \mathbf{R}_T = \mathbf{I}. \tag{6}$$

The above-mentioned problem is an orthogonal procrustes problem [33] with an analytical solution. To be exact, the optimal solution is obtained by employing the singular value decomposition (SVD), that is,

$$\left(\mathbf{B}_T - \frac{\lambda_1}{\lambda_1 + 1}(\mathbf{X}_{\mathrm{SC}} \mathbf{P})\right)^\top \mathbf{X}_T = \hat{\mathbf{S}} \Sigma \mathbf{S}^\top$$

then

$$\mathbf{R}_T = \hat{\mathbf{S}} \mathbf{S}^\top. \tag{7}$$

3) Update $\mathbf{P}$ by Fixing $\mathbf{R}_T$ and $\mathbf{B}_T$: By fixing $\mathbf{R}_T$ and $\mathbf{B}_T$, the corresponding optimization with respect to $\mathbf{P}$ is as follows:

$$\min_{\mathbf{P}} \|(\mathbf{B}_T - \mathbf{X}_T \mathbf{R}_T) - \mathbf{X}_{\mathrm{SC}} \mathbf{P}\|_F^2 \quad \text{s.t. } \mathbf{P}^\top \mathbf{P} = \mathbf{I}. \tag{8}$$

The above-mentioned optimization is also a standard orthogonal procrustes problem, which has the following analytical solution:

$$\mathbf{P} = \hat{\mathbf{Q}} \mathbf{Q}^\top \tag{9}$$

where $\mathbf{Q}$ and $\hat{\mathbf{Q}}$ are results of SVD on $(\mathbf{B}_T - \mathbf{X}_T \mathbf{R}_T)^\top \mathbf{X}_{\mathrm{SC}}$, i.e., $(\mathbf{B}_T - \mathbf{X}_T \mathbf{R}_T)^\top \mathbf{X}_{\mathrm{SC}} = \hat{\mathbf{Q}} \Lambda \mathbf{Q}^\top$.

Algorithm 1 Alternating Optimization Procedure for ITQ+ (or LapITQ+)

1: Initialize $\mathbf{R}_T^0, \mathbf{P}^0$ to be the random orthogonal matrices, and set $\tau = 0$.
2: **While** not converge
3: Update $\mathbf{B}_T^{\tau+1}$ by solving (4) (or (11) for LapITQ+).
4: Update $\mathbf{R}_T^{\tau+1}$ via (7).
5: Update $\mathbf{P}^{\tau+1}$ via (9).
6: $\tau = \tau + 1$.
7: **End While**

IV. Laplacian Regularized ITQ+ (LapITQ+)

ITQ+ learns a hashing model for the target domain with $\mathbf{X}_{\mathrm{SC}}$, which may not fully explore the information in the source domain. In practice, apart from $\mathbf{X}_{\mathrm{SC}}$, one may have a large amount of training data $\mathbf{X}_{\mathrm{SU}}$ from the source domain. Although the corresponding feature vectors of $\mathbf{X}_{\mathrm{SU}}$ on the target domain are unknown, it is useful to describe

the latent structure of the whole data space. To exploit the underlying information existing in $\mathbf{X}_{\mathrm{SU}}$, we further extend ITQ+ to LapITQ+ with the incorporation of the transferred graph structure. Our motivation comes from the community of multiview analysis. To be specific, the latent graph structures in different views are generally believed to be similar [34]. Thus, when a large amount of training instances on the source domain are available, one can formulate the local geometry of the source domain into a graph and embed it into the target domain. Formally, ITQ is employed to obtain hashing codes $\mathbf{B}_S$ using all available source domain data $\mathbf{X}_S = [\mathbf{X}_{\mathrm{SC}}^\top, \mathbf{X}_{\mathrm{SU}}^\top]^\top$ by solving

$$\begin{aligned} &\min_{\mathbf{B}_S, \mathbf{R}_S} \ \|\mathbf{B}_S - \mathbf{X}_S\mathbf{R}_S\|_F^2 \\ &\text{s.t. } \mathbf{R}_S^\top \mathbf{R}_S = \mathbf{I} \end{aligned} \tag{10}$$

where $\mathbf{X}_S \in \mathbb{R}^{(n_S+n)\times d_S}$, $\mathbf{B}_S \in \{-1, 1\}^{(n_S+n)\times c}$ and $\mathbf{R}_S \in \mathbb{R}^{d_S\times c}$. Note that this problem could be solved offline in advance for time saving.

Next, an adjacency graph $\mathbf{G}$ is constructed based on the hash codes $\mathbf{B}_S$ with the following steps. For each code $\mathbf{B}_{S_i}$ (the row of $\mathbf{B}_S$), we seek k nearest neighbors for it and assign a weight of one to the connection. Note that the Euclidean distance can also be used to measure the affinity among data points. In our experiments, however, the used hamming distance has shown our LapITQ+ could achieve desirable performance. After obtaining the adjacency graph, we compute the graph Laplacian $\mathbf{L}_C \in \mathbb{R}^{n\times n}$ for the target domain data. In mathematical, the objective of LapITQ+ is as follows:

$$\begin{aligned} &\min_{\substack{\mathbf{B}_T\in\mathbb{B},\\ \mathbf{R}_T,\mathbf{P}_T}} \ \|\mathbf{E}\|_F^2 + \lambda_1\|\mathbf{E} - \mathbf{g}(\mathbf{X}_{\mathrm{SC}})\|_F^2 + \lambda_2 \mathrm{tr}\big(\mathbf{B}_T^\top \mathbf{L}_C \mathbf{B}_T\big) \\ &\text{s.t. } \mathbf{R}_T^\top \mathbf{R}_T = \mathbf{I}, \quad \text{and } \mathbf{P}^\top\mathbf{P} = \mathbf{I} \end{aligned}$$

where λ_1 and λ_2 are two nonnegative parameters. The third term is used to transfer the geometry structure from the source domain to the target domain. It should be pointed out that the graph Laplacian is obtained from the binary codes instead of the original space. In this way, the local geometric information on the source domain is quantified and transferred across domains.

LapITQ+ employs the similar optimization procedure with ITQ+. The only one difference between them is the updating scheme on $\mathbf{B}_T$. More specifically, there is the graph Laplacian in LapITQ+. Thus, we only present the details of this step for simplicity.

A. Updating $\mathbf{B}_T$ by Fixing $\mathbf{P}$ and $\mathbf{R}_T$

As the constraint $\mathbf{B}_T \in \{-1, 1\}^{n\times c}$ causes an NP-hard problem, we relax it to $\mathbf{B}_T \in [-1, 1]^{n\times c}$ on the feasible domain $\mathbb{B}$, thus leading to the following constrained quadratic programming (QP) optimization:

$$\min_{\mathbf{B}_T\in\mathbb{B}} \ -2\mathrm{tr}\,(\mathbf{B}_T\mathbf{K}) + \lambda_2\|\mathbf{B}_T\mathbf{L}\|_F^2 \tag{11}$$

where $\mathbf{K} = ((1+\lambda_1)\mathbf{R}^\top\mathbf{X}_T^\top) + \lambda_1\mathbf{P}^\top\mathbf{X}_{\mathrm{SC}}^\top$ and $\mathbf{L}_C = \mathbf{L}^\top\mathbf{L}$.

After obtaining the optimum of the above-mentioned problem, we binarize the codes by $\mathbf{B}_T = \mathrm{sgn}(\mathbf{B}_T)$.

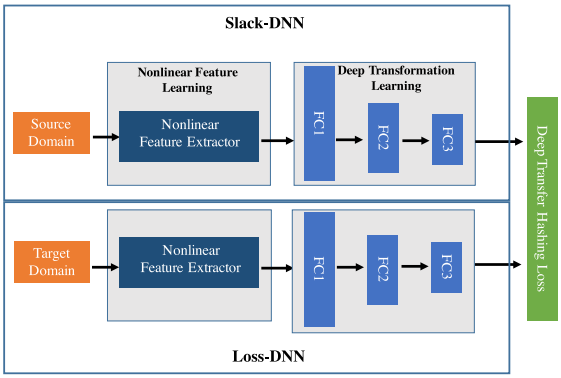


Fig. 3. Architecture of DTH.

V. Complexity Analysis

The time cost of the proposed algorithms mainly consists of two parts: 1) optimizing the binary codes $\mathbf{B}_T$ and 2) optimizing the orthogonal rotation matrices $\mathbf{R}_T$ and $\mathbf{P}$. To update $\mathbf{B}_T$, ITQ+ takes $O(n_T \log(n_T)c)$ for sorting, and LapITQ+ takes $O(n_T^3)$ to perform QP. To update the orthogonal rotation matrices $\mathbf{R}_T$ and $\mathbf{P}$, the time complexities of ITQ+ and LapITQ+ are bounded by $O(c^2\, d_T + d_T^3)$ and $O(c^2\, d_S + d_S^3)$, respectively. Here, n_T denotes the size of training data from the target domain, c is the length of code, and d_S and d_T are the dimensions of data from the source and target domains.

VI. Deep Transfer Hashing With Privileged Information

Recently, learning with DNNs has been widely proven a powerful tool on a variety of applications. In this section, we propose an new algorithm termed DTH under our framework of THPI. The proposed DTH (see Fig. 3) is flexible and can be build on the top of any existing deep feature extraction models, e.g., CNN and recurrent neural network. Instead of learning a single transformation matrix $\mathbf{R}_T$ in ITQ, DTH aims to learn a set of transformation matrices cascaded with the nonlinear mapping functions $h(\cdot)$ such that the quantization error is minimized. Specifically, DTH adopts two-stream DNN architecture wherein each stream consists of three modules, namely, nonlinear feature learning, deep transformation learning, and DTH loss. The DTH is motivated by the LUPI paradigm. Although LUPI paradigm has many good theoretical and practical merits, the existing methods focus on shallow architectures [35], [36] and less attention has been paid on unifying LUPI and deep learning.

A. Nonlinear Feature Learning

The traditional shallow hashing methods are based on the handcraft features by utilizing predefined project functions to transform the data from the input space into the feature space, such as scale-invariant feature transform [37] for images or bag-of-words (BOW) for texts. Different from these hand-craft features, deep learning aims to learn hierarchical features from raw data. With recent advances of DNNs, the much powerful features can be extracted for different applications. For example, CNN features have shown superior

TABLE I
CONFIGURATION OF THE DEEP TRANSFORMATION LEARNING

Layer	Configure
full1	Length of $\psi(\mathbf{x})$
	Tanh
full2	Length of $\psi(\mathbf{x})$
	Tanh
full3	Hash code length c
	Tanh
	Sgn

performances in various of computer vision tasks [26], [38]. word2vec [39] is built upon the long short-term memory [40], which could extract deep feature for each single word and has achieved the state-of-art performances in numerous natural language processing tasks [41].

To construct the SGD-compatible loss function, we reformulate ITQ as follows:

$$\min_{\psi_T} \sum_i \|\mathbf{b}_{T_i} - \psi_T(\mathbf{x}_{T_i})\mathbf{R}_T\|_2^2 \tag{12}$$

where $\psi_T(\mathbf{x}_{T_i})$ is the feature of the data $\mathbf{x}_{T_i}$. For images, $\psi_T(\mathbf{x}_{T_i})$ can be obtained by either hand-craft features like GIST [42] or pretrained deep CNN models like VGG-16 [43]. For textual data, $\psi_T(\mathbf{x}_{T_i})$ can be either traditional BOW features or deep features.

B. Deep Transformation Learning

In ITQ, linear transformation matrices $\mathbf{R}_T$ and $\mathbf{R}_S$ are learned to project the original target and source feature space into hashing-related space. However, such a transformation matrix is always inferior to neural network-based methods especially when the real-world data distribution is complex [44]. On the other hand, the universal approximation theorem [45] states that a feed-forward network with a single-hidden layer containing a finite number of neurons (i.e., a multilayer perceptron), can approximate any continuous functions on the compact subsets of $\mathbb{R}^n$, under some mild assumptions on the activation function.

With these merits, we replace the linear mapping $\mathbf{R}$ in ITQ by an nonlinear linear mapping $\phi_T(\cdot)$ learned through the neural networks. Based on (12), the problem can be further reformulated as

$$\min_{\phi_T} \sum_i \|\mathbf{b}_{T_i} - \phi_T(\psi_T(\mathbf{x}_{T_i}))\|_2^2 \tag{13}$$

where $\psi_T(\mathbf{x}_{T_i})$ is the nonlinear representation of $\mathbf{x}_{T_i}$ as described in Section VI-A.

The deep transformation mapping $\phi_T(\cdot)$ is implemented by using a neural network consisting of L fully connected layers, where the lth ($l = 1, 2, \ldots, L$) layer is with the weight of $\mathbf{W}_T^l$ and the nonlinear activation function of σ. The details of the used neural network is summarized in Table I. Particularly, $\mathbf{h}_{T_i}^l = \sigma(\mathbf{W}_T^l \mathbf{h}_{T_i}^{l-1})$ denotes the output of lth layer of the neural network for the data point $\mathbf{x}_{T_i}$ from the target domain, $\mathbf{h}_{T_i}^1 = \sigma(\mathbf{W}_T^1 \psi(\mathbf{x}_{T_i}))$, $\mathbf{W}_T^l$ is the weight and we use the tanh function as $\sigma(\cdot)$. By feeding the real-value output into the sign function, it gives the binary hash code $\mathbf{b}_{T_i} = \text{sgn}(\mathbf{h}_{T_i})$, where $\mathbf{h}_{T_i}$ denotes the output of deep transformation learning for $\mathbf{x}_{T_i}$ (i.e., $\mathbf{h}_{T_i}^L$).

C. Deep Transfer Hashing Loss

Similar to LapITQ+, the hashing loss used in DTH can be divided into two parts, namely, quantization loss and graph structure loss.

The quantization loss is defined by

$$L_Q = \sum_{i=1}^{n} \|\mathbf{b}_{T_i} - \mathbf{h}_{T_i}\|_2^2 \tag{14}$$

where $\mathbf{h}_{T_i}$ is the real-value output of deep transformation learning for the target data $\mathbf{x}_{T_i}$ and $\mathbf{b}_{T_i}$ denotes its corresponding hash code.

To utilize the local geometric relations on the manifold, we construct a similarity matrix $\mathbf{S} = (S_{ij})_{i,j=1}^N$ whose elements indicate the similarity between two instances i and j, where $S_{ij} = 1$ indicates similar samples and $S_{ij} = 0$ indicates dissimilar samples. To remain the graph structure between the discrete spaces and the real-valued feature space, we construct a soft similarity matrix as follows:

$$\mathbf{S}_1 = \text{cosine}(\psi(\mathbf{x}_{T_i}), \psi(\mathbf{x}_{T_j} x)) \tag{15}$$

where $\psi(\mathbf{x}_{T_i})$ can be obtained using original features or pretrained deep learning features.

Furthermore, we also consider the graph structure of hash codes $\mathbf{b}_{S_i}$ in the source domain. To transfer the graph from the source to the target domain, another similarity matrix is constructed as follows:

$$\mathbf{S}_2 = \text{cosine}(\mathbf{b}_{S_i}, \mathbf{b}_{S_j}). \tag{16}$$

Combining the above-mentioned equation, the similarity matrix $\mathbf{S}$ is expressed as $\mathbf{S} = \mathbf{S}_1 + \mathbf{S}_2$ with normalized vectors. Then, we define the structure loss by

$$L_G = \sum_{i,j} \left(\frac{1}{c} \mathbf{h}_{T_i}^\top \mathbf{h}_{T_j} - S_{ij} \right)^2 \tag{17}$$

where c is the length of hashing code and the similarity matrix S_{ij} models the neighborhood between instances from two domains. Although the construction of the similarity matrix is time-consuming, it can be done offline.

The overall objective for the unsupervised DH can be summarized to

$$\min L = L_Q + \lambda_1 L_G + \lambda_2 \text{reg}(\phi_T^L) \tag{18}$$

where L_Q is the quantization error, L_G is the loss in the graph structure, and $\text{reg}(\phi_T^L)$ is the ℓ_2 regularization enforced on the mapping ϕ_T^L of the Lth fully connected layer for the target domain.

For the data point $\mathbf{x}_T$ from the target domain, let $\mathbf{h}_T^L$ denote the corresponding feature that is learned through nonlinear feature learning and deep transformation learning. We use the second stream of our network (termed slack-DNN) to model the privileged information from the source domain. Moreover, $\phi_S^L(\mathbf{x}_S)$ consists of the features that are learned from the source domain data $\mathbf{x}_S$ by the slack DNN. Different from the existing cross-modality hashing methods, the second

stream of network, i.e., the loss DNN is not used to learn hashing codes, but to model the loss of the first stream. Let $F(\mathbf{x}_S)$ be the output of the second stream for an input privileged data $\mathbf{x}_S$, similar to ITQ+, the two streams share the same loss layer defined by

$$\min L_P = L + \lambda \sum_i \|L_Q(\mathbf{x}_{T_i}) - F(\mathbf{x}_{S_i})\|_2^2 \tag{19}$$

where L has been defined in (18), $L_Q(\mathbf{x}_{T_i}) = \mathbf{b}_{T_i} - \mathbf{h}_{T_i}$ is the quantization error for $\mathbf{x}_{T_i}$, and $F(\mathbf{x}_{S_i})$ is the output of slack DNN for $\mathbf{x}_{S_i}$. Compared with ITQ+/LapITQ+, DTH (19) approximates the quantization error using a neural network $F(\cdot)$ instead of a single mapping matrix, thus could enhancing the representative ability of the privileged information.

D. Optimization

DTH needs to optimize the parameters of two-stream DNNs (two-stream DNN) and binary hash code $\mathbf{b}_T$. Let $\mathbf{\Omega}_S$ and $\mathbf{\Omega}_T$ denote the parameters of slack DNN and loss DNN, respectively. We adopt an alternating learning strategy to learn $\mathbf{\Omega}_S$, $\mathbf{\Omega}_T$, and $\mathbf{b}_T$. At each time, we optimize one parameter by fixing the other two. The algorithm is outlined in Algorithm 2, and the detailed derivation are introduced as follows.

Algorithm 2 Optimization for DTH

1: **INPUT**: Source domain data $\mathbf{X}_S$, target domain data $\mathbf{X}_T$ and graph similarity matrix $\mathbf{S}$
2: Set the neural network parameters ψ_S and ψ_T for the source and the target domain with pretrain models.
3: **While** not converge
4: Randomly sample a mini-batch from $\mathbf{X}_{T_i}$ and compute the corresponding $\mathbf{h}_{T_i}$ by the forward propagation.
5: Calculate the derivate of $\mathbf{\Omega}_T$ via (21).
6: Update $\mathbf{\Omega}_T$ by the back-propagation.
7: **End While**
8: **While** not converge
9: Randomly sample a mini-batch from $\mathbf{X}_{S_i}$ and compute the corresponding $F(\mathbf{x}_{S_i})$ by the forward propagation.
10: Calculate the derivate of $\mathbf{\Omega}_S$ according to (22) in Section VI-D2.
11: Update the $\mathbf{\Omega}_S$ using back-propagation.
12: **End While**
13: Update $\mathbf{B}$ according to (24) in Section VI-D3.

1) Update $\mathbf{\Omega}_T$ by Fixing $\mathbf{\Omega}_S$ and $\mathbf{b}_T$: When $\mathbf{\Omega}_S$ and $\mathbf{b}$ are fixed, we learn the neural network $\mathbf{\Omega}_T$ from the target domain by using back-propagation (BP) with SGD. The gradient of the loss function L in (19) with respect to $\mathbf{h}_{T_i}$ is computed as follows:

$$\frac{\partial L}{\partial \mathbf{h}_{T_i}} = 2(\mathbf{h}_{T_i} - \mathbf{b}_{T_i}) + 2\lambda_1 \sum_j \left(\frac{1}{c}\mathbf{h}_{T_i}^\top \mathbf{h}_{T_j} - S_{ij}\right)\mathbf{h}_{T_j} \tag{20}$$

where $\mathbf{h}_{T_i} = \phi_T(\mathbf{x}_{T_i}, \mathbf{\Omega}_T)$. Furthermore, the gradient of the loss with the privileged information L_P is computed as follow:

$$\frac{\partial L_P}{\partial \mathbf{h}_{T_i}} = \frac{\partial L}{\partial \mathbf{h}_{T_i}} + 2\lambda(F(\mathbf{x}_{S_i}) + \mathbf{h}_{T_i} - \mathbf{b}_{T_i}). \tag{21}$$

Then, we can compute $(\partial L_P/\partial \mathbf{\Omega}_T)$ with $(\partial L_P/\partial \mathbf{h}_{T_i})$ by using the chain rule and update $\mathbf{\Omega}_T$ using BP.

2) Update $\mathbf{\Omega}_S$ by Fixing $\mathbf{\Omega}_T$ and $\mathbf{b}_T$: Similar to updating $\mathbf{\Omega}_T$, we can also learn the neural network parameter $\mathbf{\Omega}_S$ of the source domain with fixed $\mathbf{\Omega}_T$ and $\mathbf{b}_T$ using BP. The gradient of the loss L_P (19) with respect to $\mathbf{h}_{S_i}$ is computed as follows:

$$\frac{\partial L_P}{\partial \mathbf{h}_{S_i}} = 2\lambda(F(\mathbf{x}_{S_i}) + \mathbf{h}_{T_i} - \mathbf{b}_{T_i})\frac{\partial F}{\partial \mathbf{h}_{S_i}} \tag{22}$$

where $F(\mathbf{x}_{S_i}) = \phi_S(\mathbf{x}_{S_i}, \mathbf{\Omega}_S)$. Then we can compute $((\partial L_P)/(\partial \mathbf{\Omega}_S))^{(1/2)}$ with $((\partial L_P)/(\partial \mathbf{h}_{S_i}))^{(1/2)}$ by using the chain rule and update $\mathbf{\Omega}_S$ using BP.

3) Update $\mathbf{b}_T$ by Fixing $\mathbf{\Omega}_T$ and $\mathbf{\Omega}_S$: When $\mathbf{\Omega}_T$ and $\mathbf{\Omega}_S$ are fixed, the problem in (19) can be reformulated as follows:

$$\begin{aligned} &\max_{\mathbf{b}_{T_i}} \ \mathrm{tr}\big(\mathbf{b}_{T_i}^\top((\lambda+1)\mathbf{h}_{T_i} + \lambda F(\mathbf{x}_{S_i}))\big) \\ &\text{s.t. } \mathbf{b}_{T_i} \in \{-1, +1\}^c. \end{aligned} \tag{23}$$

One could obtain the optimal solution $\mathbf{b}_{T_i}$ as follows:

$$\mathbf{b}_{T_i} = \mathrm{sgn}((\lambda+1)\mathbf{h}_{T_i} + \lambda F(\mathbf{x}_{S_i})). \tag{24}$$

It is interesting to observe that the updating strategy of $\mathbf{b}_T$ in DTH is similar to that in ITQ+ (4). The major difference between them lies in the transformation and feature representation method, namely, linear mapping versus neural network.

E. Out-of-Sample Hash Codes Inference

In the setting of TL with the privileged information, the parallel privileged data will be unavailable during the testing phrase. Nevertheless, slack DNN helps to train a better loss DNN for inferring the hash codes. With the trained networks, we can easily obtain the binary hash codes for any data point $\mathbf{x}$, which does not appear in the training data set. Specifically, we pass $\mathbf{x}$ into the loss DNN network and perform a forward propagation as follows:

$$\mathbf{b}_i = \mathrm{sgn}(\phi_T(\mathbf{x}, \mathbf{\Omega}_T)). \tag{25}$$

VII. Experiments

A. Experimental Settings

1) Data Sets: In this section, we carry out experiments using three popular data sets, including British Broadcasting Corporation (BBC) Collection [46], multilingual Reuters [47], and NUS-WIDE [48].

BBC collection is a multiview data set of which each instance consists of three views and each view is constructed by splitting each article into different segments. In our experiment, we use View 2 as the source domain and View 1 as the target domain. In the deep learning setting, we feed the original features into the marginalizing stacked linear denoising autoencoders (mSDA) [19], [49].

Multilingual Reuters collection contains about 11 000 articles sampled from six topics in five different languages, e.g., English, French, and so on. We represent each document as a BOW vector and compute the term frequency-inverse document frequency (TF-IDF) as features. In our experiments, we follow the setting in [18] and [20] and use the documents

in English and French as the source and target domain, respectively. For computational efficiency, we reduce the dimension of the TF-IDF vectors by preserving 60% principal component analysis energy. For all the tested deep learning methods, we only preprocess the original TF-IDF features with the mSDA.

NUS-WIDE data set includes about 200 000 images sampled from 81 subjects with a total number of 5018 unique tags, which is downloaded from the Flickr websites. In our experiments, we directly use the features provided in [50]. For the shallow models, the image features (150-dimensional color moment) and the tag features (60-dimensional textual vector) are treated as data from the target and source domain, respectively. For the deep models, the documents are handled using the word2vec [31], and the images are passed through a VGG16 network [43].

2) Baselines: The proposed transfer hashing approach is compared with five shallow hashing methods including cross-view hashing (CVH) [9], canonical correspondence analysis (CCA)-ITQ [11], PM^{2}H [16], LSH [51], and data sensitive hashing (DSH) [52], where the last two are cross-modal hashing methods. Moreover, we compare the proposed DTH with five DH baselines including deepITQ (DITQ), deep cross-modal hashing (DCMH) [15], DH [53], DITQ+, and DeepLapITQ+ (DLapITQ+). Note that DITQ, DITQ+, and DLapITQ+ first extract features using the same neural network adopted in our DTH and then perform CCA-ITQ, ITQ+, and LapITQ+ to obtain results.

For fair comparisons, we adopt the evaluation protocols used in [11] and [13] to make a decision. Specifically, a nominal threshold of the average distance to the 50th nearest neighbor is used to determine whether a database point returned for a given query is considered a true positive. Moreover, the widely used criterion mean average precision (MAP) is used as the performance metric.

To verify the performance of our method in the case of the partial cross-domain correspondence, we randomly sample a subset from training data with a fix ratio $\alpha = n/(n+n_S)$ and use 10% of the rest for testing, where α increases from 0.1 to 0.7 with an interval of 0.2. We adopt the cross validation to tune the parameters for all the proposed methods. For the parameter analysis, please refer to Section VII-E for details. To remove the randomness due to sampling, we repeat each algorithm 10 times using different data partitions and report their mean of MAP.

B. Comparison With State of the Arts

We first evaluate the performance of different methods by varying the number of hashing bits in the range of $\{8, 16, 32, 64\}$ with the fixed $\alpha = 0.5$.

From Table II, one observes that the cross-modal hashing methods (CCA-ITQ, CVH, and PM^{2}H) perform better than single-modal hashing methods since the other modality gives additional information to give better hash codes on the target domain. Note that ITQ+ and LapITQ+ are the best shallow hashing approaches since they introduce the new slack function to use the privileged information to regularize the quantization loss, thus improving the generalization and robustness of model, especially when the target data encounter the data sparsity issue. The results of DH methods are summarized in Table III. The proposed DTH significantly outperforms other two single-modal-oriented DH approaches (DITQ and DH). The results verify the necessity of three components in DTH, namely, deep feature learning, deep transformation learning, and privileged loss construction. More specifically, all DH methods show a large advantage over shallow hashing methods, which indicates that the deep learning feature extraction is crucial to the performance boosting. Comparing with DH, the performance of DITQ is still inferior to it since DITQ learns a single transformation matrix based on deep features. In other words, the deep transformation learning plays an important role in improving the performance. Furthermore, DH can be deemed as a special case of DTH with the single-stream architecture and its performance is not as good as DTH. Therefore, we can conclude that the slack DNN can transfer the knowledge to the other stream of DNN such that the performance could be improved.

Although the latest DCMH performs comparable to DITQ+ and DLapITQ+ since it considers both two modalities, it is still inferior to DTH because DTH aims to optimize the codes of target domain instead two domains. From the result,

TABLE II

MAP (%) OVER 10 RUNS WITH $\alpha = 0.5$. THE BEST RESULTS ARE DENOTED IN BOLDFACE ON NONDEEP LEARNING SETTINGS

BBC							
Bit	LSH	DSH	CCA-ITQ	CVH	PM^{2}H	ITQ+	LapITQ+
8	14.00	20.28	21.58	17.76	21.84	24.32	**26.62**
16	16.79	23.25	24.69	24.61	25.12	27.69	**28.06**
32	21.71	25.00	27.09	25.36	28.03	29.00	**30.30**
64	24.61	28.36	26.51	27.62	29.55	29.58	**31.33**
Reuters							
Bit	LSH	DSH	CCA-ITQ	CVH	PM^{2}H	ITQ+	LapITQ+
8	6.37	7.10	8.75	8.46	8.65	9.67	**10.12**
16	6.55	8.12	10.00	9.49	10.34	10.96	**11.32**
32	7.17	8.15	11.93	9.91	12.23	12.85	**13.51**
64	7.24	7.81	13.06	12.97	13.80	13.95	**14.70**
NUS-wide							
Bit	LSH	DSH	CCA-ITQ	CVH	PM^{2}H	ITQ+	LapITQ+
8	14.86	23.82	33.18	21.15	33.93	35.07	**35.42**
16	20.49	25.69	36.99	24.49	37.67	38.51	**39.13**
32	25.58	33.10	40.68	27.30	41.50	42.16	**42.85**
64	28.50	35.42	42.45	30.57	43.14	44.01	**45.40**

TABLE III

MAP (%) OVER 10 RUNS WITH $\alpha = 0.5$. THE BEST RESULTS ARE DENOTED IN BOLDFACE ON DEEP LEARNING SETTINGS

BBC						
Bit	DITQ	DH	DCMH	DITQ+	DLapITQ+	DTH
8	30.10	32.56	34.70	33.52	34.20	**35.82**
16	32.45	34.12	36.21	35.29	36.88	**38.07**
32	34.56	35.27	37.86	37.35	38.94	**40.34**
64	35.12	36.73	39.55	38.61	40.13	**41.20**
Reuters						
Bit	DITQ	DH	DCMH	DITQ+	DLapITQ+	DTH
8	11.28	12.60	13.25	13.11	13.68	**14.02**
16	13.76	14.53	15.67	14.33	15.85	**16.30**
32	16.14	17.03	17.82	17.59	18.20	**19.38**
64	20.55	21.17	22.19	21.02	22.83	**23.29**
NUS-wide						
Bit	DITQ	DH	DCMH	DITQ+	DLapITQ+	DTH
8	52.01	51.22	54.28	53.19	55.48	**56.36**
16	54.17	53.28	57.17	56.13	57.80	**58.09**
32	57.58	56.29	58.16	57.86	59.33	**60.47**
64	58.23	57.36	59.90	59.44	60.07	**61.25**

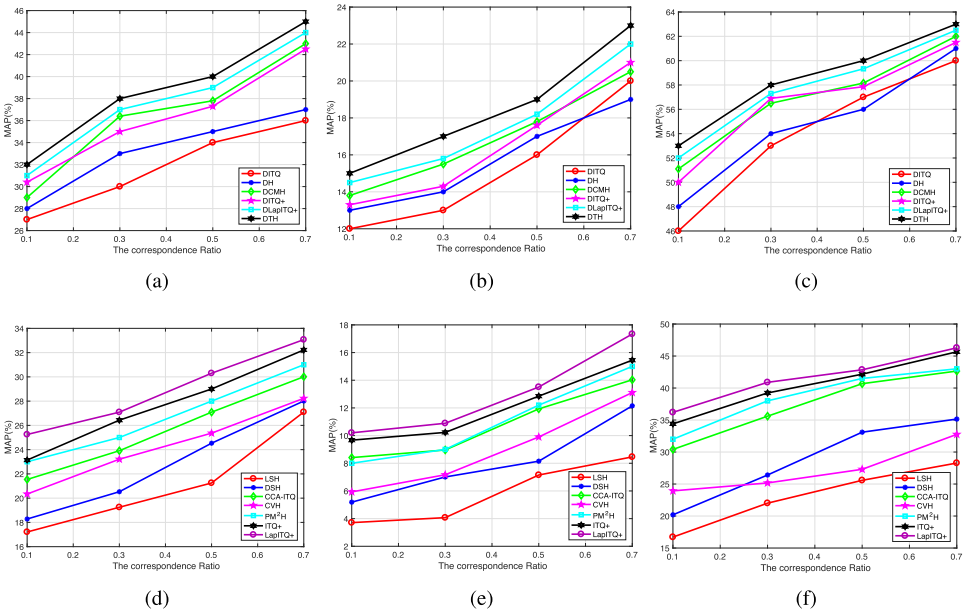


Fig. 4. Influence of privileged data size. (a) BBC (shallow). (b) Reuters (shallow). (c) NUS-Wide (shallow). (d) BBC (deep). (e) Reuters (deep). (f) NUS-Wide (deep).

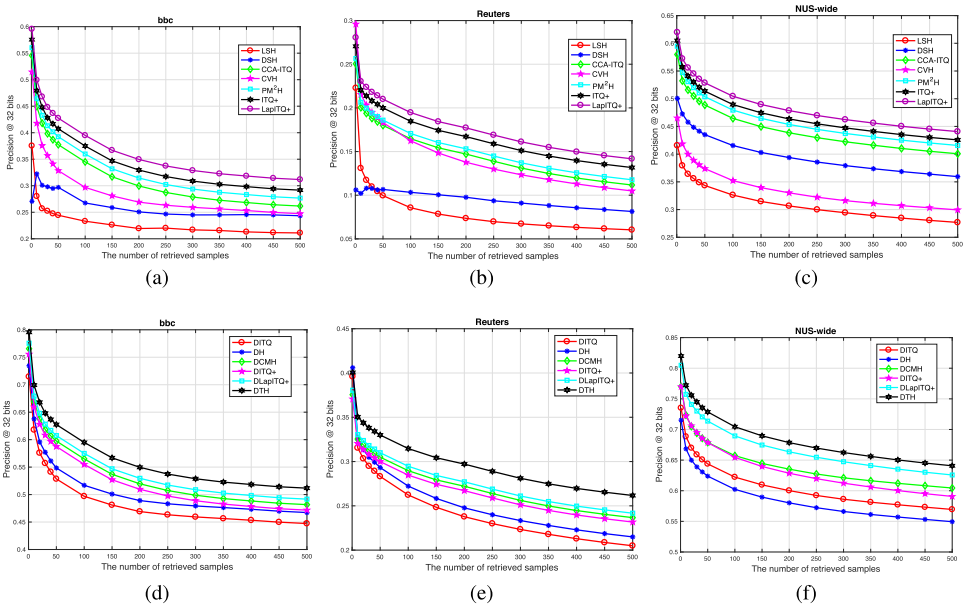
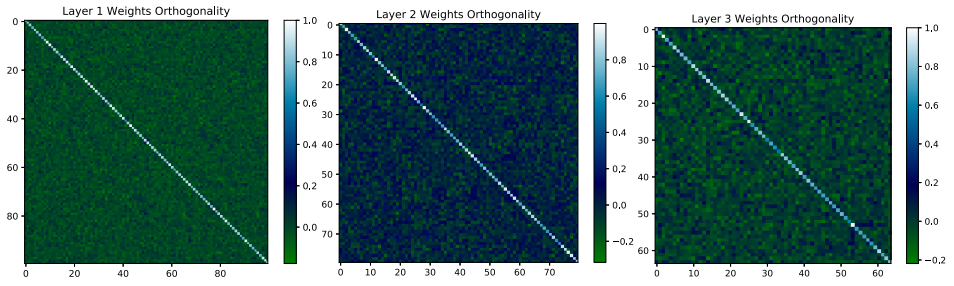


Fig. 5. Influence of the number of retrieved samples. (a) BBC (shallow). (b) Reuters (shallow). (c) NUS-Wide (shallow). (d) BBC (deep). (e) Reuters (deep). (f) NUS-Wide (deep).

we could conclude that the proposed slack function is a better way to transfer the source domain knowledge for hashing. Most cross-modal hashing methods such as CVH and DCMH require a lot of cross-domain data correspondences, and learn hashing functions only based on the correspondences. In contrast, LapITQ+ and DTH utilize all source domain data including unparalleled data to learn source-domain hash codes offline, and use the structure underlying these hash codes to regularize the learning of hash codes on the target domain. Finally, we also observe that LapITQ+ and PM^{2}H outperform ITQ+ by an improvement of 1%–2% in MAP since it incorporates the local geometric structure. Similarly, in the deep learning setting, we empirically observe that the incorporation of the similarity matrix $\mathbf{S}$ constructed from both two domains is crucial to get the nontrivial solution.

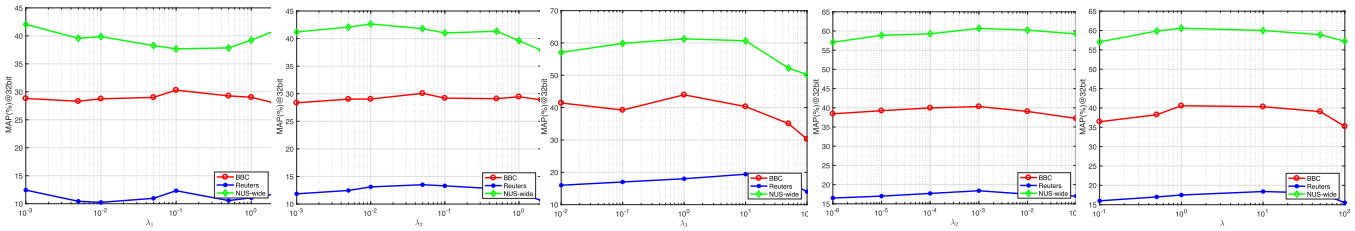
C. Training Data Size and Retrieved Sample Size

In this section, we vary the training data size by randomly selecting [10%, 30%, 50%, 70%] from the whole target domain data. The corresponding privileged data are available during training. Fig. 4 shows the result, from which one observes that the proposed methods ITQ+, LapITQ+, and DTH outperform other counterpart baselines by a considerable



(a) (b) (c)

Fig. 6. Weights orthogonality analysis. (a) Layer 1. (b) Layer 2. (c) Layer 3.



(a) (b) (c) (d) (e)

Fig. 7. Parameter analysis of LapITQ+/DTH. (a) λ_1 (LapITQ+). (b) λ_2 (LapITQ+). (c) λ_1 (DTH). (d) λ_2 (DTH). (e) λ (DTH).

performance margin, especially, when the target training data size is small.

One major application of hashing is information retrieval. Under such a scenario, it is more desirable to return K most similar results. Therefore, we also evaluate the performance of our methods for information retrieval using the Top-K precision [54]. To be specific, Fig. 5 reports the result of the tested methods with different K retrieved samples on the three data sets with the code length of 32 bits. Again, our algorithms achieve the highest precisions with different K. Regarding to different numbers of bits, the similar observations could also be obtained.

D. Orthogonality Analysis

In DTH, we remove the orthogonal constraint as it is unnecessary. In this section, we conduct experimental analysis to support this claim. Note that even though the orthogonal constraint is explicitly optimized by DTH, the obtained transformation matrix could only approximate an orthogonal matrix since the constraint will be relaxed during optimizing. Therefore, we will show that $(1/d_i)\mathbf{W}_T^{(l)\top}\mathbf{W}_T^{(l)}$ approximates an identity matrix, where $\mathbf{W}_T^{(l)}$ is the transformation matrix of the lth layer. To the end, Fig. 6 shows $(1/d_i)\mathbf{W}_T^{(l)\top}\mathbf{W}_T^{(l)}$ on the BBC data set. From the figure, one observes that the transformation matrix in each layer approximates an orthogonal matrix after convergence.

E. Parameter Analysis

In this section, we investigate the influence of parameters of our methods. As LapITQ+ is an extension of ITQ+, we only examine LapITQ+ which has two user-specified parameters λ_1 and λ_2. The value of parameters ranges in $[0, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 2]$. To investigate the influence of one parameter, we fix another one to 0.01. Fig. 7(a) and (b) reports the results of LapITQ+.[1] From results, one could observe that LapITQ+ is insensitive to λ_1 and λ_2. For the DH method DTH, the range of different parameters is at different scales and results are summarized in Fig. 7(c)–(e).

VIII. Conclusion

To address the data sparsity issue in hashing, this paper proposes a transfer hashing framework which exploits the privileged information from the source domain to learn a slack function for predicting hash codes of unobserved data from the target domain. Based on the proposed framework, three variants of ITQ are developed and have shown promising performance comparing with several state-of-the-art methods. One of our future work is to exploit the proposed method in other computer vision tasks, such as visual tracking [55], [56].

Acknowledgment

The authors would like to thank Associate Editor and anonymous reviewers for their valuable comments and constructive suggestions to improve the quality of this paper.

References

[1] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—A survey," *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Jan. 2016.

[2] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, Dec. 2015.

[3] 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 Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2018.2789887.

[4] Q. Wang, J. Wan, and Y. Yuan, "Locality constraint distance metric learning for traffic congestion detection," *Pattern Recognit.*, vol. 75, pp. 272–281, Mar. 2018.

[1]The results on the three data sets are 28.49, 11.52, and 41.78 with $\lambda_1 = 0$, and 27.56, 13.37, and 42.41 with $\lambda_2 = 0$, respectively.

[5] X. Peng, H. Tang, L. Zhang, Z. Yi, and S. Xiao, "A unified framework for representation-based subspace clustering of out-of-sample and large-scale data," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2499–2512, Dec. 2016.

[6] Y. Yang, C. Deng, S. Gao, W. Liu, D. Tao, and X. Gao, "Discriminative multi-instance multitask learning for 3D action recognition," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 519–529, Mar. 2017.

[7] L. Shao, F. Zhu, and X. Li, "Transfer learning for visual categorization: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 5, pp. 1019–1034, May 2014.

[8] V. Vapnik and A. Vashist, "A new learning paradigm: Learning using privileged information," *Neural Netw.*, vol. 22, nos. 5–6, pp. 544–557, 2009.

[9] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. IJCAI*, Jul. 2011, pp. 1360–1365.

[10] B. Wu, Q. Yang, W.-S. Zheng, Y. Wang, and J. Wang, "Quantized correlation hashing for fast cross-modal search," in *Proc. IJCAI*, Jul. 2015, pp. 3946–3952.

[11] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. CVPR*, Jun. 2011, pp. 817–824.

[12] J. T. Zhou, X. Xu, S. J. Pan, I. W. Tsang, Z. Qin, and R. S. M. Goh, "Transfer hashing with privileged information," in *Proc. IJCAI*, May 2016, pp. 2414–2420.

[13] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. NIPS*, 2009, pp. 1509–1517.

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

[15] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. CVPR*, Honolulu, HI, USA, Jul. 2017, pp. 3270–3278.

[16] Q. Wang, L. Si, and B. Shen, "Learning to hash on partial multi-modal data," in *Proc. IJCAI*, Buenos Aires, Argentina, Jul. 2015, pp. 3904–3910.

[17] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, Jun. 2013.

[18] J. T. Zhou, I. W. Tsang, S. J. Pan, and M. Tan, "Heterogeneous domain adaptation for multiple classes," in *Proc. AISTATS*, 2014, pp. 1095–1103.

[19] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid heterogeneous transfer learning through deep learning," in *Proc. AAAI*, 2014, pp. 2213–2220.

[20] J. T. Zhou, S. J. Pan, I. W. Tsang, and S.-S. Ho, "Transfer learning for cross-language text categorization through active correspondences construction," in *Proc. AAAI*, 2016, pp. 2400–2406.

[21] X. Ou, L. Yan, H. Ling, C. Liu, and M. Liu, "Inductive transfer deep hashing for image retrieval," in *Proc. ACM MM*, 2014, pp. 969–972.

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

[23] V. Vapnik and R. Izmailov, "Learning using privileged information: Similarity control and knowledge transfer," *J. Mach. Learn. Res.*, vol. 16, pp. 2023–2049, Jan. 2015.

[24] X. Xu, W. Li, and D. Xu, "Distance metric learning using privileged information for face verification and person re-identification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3150–3162, Dec. 2015.

[25] V. Sharmanska, N. Quadrianto, and C. H. Lampert, "Learning to rank using privileged information," in *Proc. ICCV*, Dec. 2013, pp. 825–832.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. NIPS*, 2012, pp. 1097–1105.

[27] H. Zhu *et al.*, "Yotube: Searching action proposal via recurrent and static regression networks," *IEEE Trans. Image Process*, vol. 27, no. 6, pp. 2609–2622, Jun. 2018.

[28] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Proc. RBM*, vol. 500, no. 3, p. 500, 2007.

[29] F. Shen, Y. Yang, L. Liu, W. Liu, D. Tao, and H. T. Shen, "Asymmetric binary coding for image search," *IEEE Trans. Multimedia*, vol. 19, no. 9, pp. 2022–2032, Sep. 2017.

[30] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. CVPR*, Jun. 2015, pp. 3270–3278.

[31] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proc. CVPR*, Jun. 2016, pp. 1183–1192.

[32] M. Lapin, M. Hein, and B. Schiele, "Learning using privileged information: SVM+ and weighted SVM," *Neural Netw.*, vol. 53, pp. 95–108, May 2014.

[33] P. H. Schönemann, "A generalized solution of the orthogonal Procrustes problem," *Psychometrika*, vol. 31, no. 1, pp. 1–10, 1966.

[34] J. He and R. Lawrence, "A graphbased framework for multi-task multi-view learning," in *Proc. ICML*, Jun. 2011, pp. 25–32.

[35] V. Sharmanska and N. Quadrianto, "Learning from the mistakes of others: Matching errors in cross-dataset learning," in *Proc. CVPR*, Jul./Dec. 2016, pp. 3967–3975.

[36] X. Wang, N. Thome, and M. Cord, "Gaze latent support vector machine for image classification improved by weakly supervised region selection," *Pattern Recognit.*, vol. 72, pp. 59–71, Dec. 2017.

[37] I. A. Adzhubei *et al.*, "A method and server for predicting damaging missense mutations," *Nature Methods*, vol. 7, no. 4, pp. 248–249, 2010.

[38] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.

[39] T. Mikolov, K. Chen, G. Corrado, and J. Dean. (2013). "Efficient estimation of word representations in vector space." [Online]. Available: https://arxiv.org/abs/1301.3781

[40] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[41] D. Bahdanau, K. Cho, and Y. Bengio. (2014). "Neural machine translation by jointly learning to align and translate." [Online]. Available: https://arxiv.org/abs/1409.0473

[42] M. Douze, H. Jégou, H. Sandhawalia, L. Amsaleg, and C. Schmid, "Evaluation of gist descriptors for Web-scale image search," in *Proc. ACM-CIVR*, 2009, p. 19.

[43] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[44] Z. Wang, Y. Yang, S. Chang, Q. Ling, and T. S. Huang, "Learning a deep l_∞ encoder for hashing," in *Proc. IJCAI*, 2016, pp. 2174–2180.

[45] B. C. Csáji, "Approximation with artificial neural networks," M.S. thesis, Faculty Sci., Eötvös Loránd Univ., Budapest, Hungary, 2001.

[46] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *Proc. ICML*, Jun. 2006, pp. 377–384.

[47] M. Amini, N. Usunier, and C. Goutte, "Learning from multiple partially observed views—An application to multilingual text categorization," in *Proc. NIPS*, 2009, pp. 28–36.

[48] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from National University of Singapore," in *Proc. CIVR*, Santorini, Greece, Jul. 2009, pp. 1–9.

[49] M. Chen, K. Q. Weinberger, Z. E. Xu, and F. Sha, "Marginalizing stacked linear denoising autoencoders," *J. Mach. Learn. Res.*, vol. 16, no. 12, pp. 3849–3875, 2015.

[50] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc. SIGMOD*, Jun. 2013, pp. 785–796.

[51] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. FOCS*, Oct. 2006, pp. 459–468.

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

[53] J. Lu, V. E. Liong, and J. Zhou, "Deep hashing for scalable image search," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2352–2367, May 2017.

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

[55] X. Lan, S. Zhang, P. C. Yuen, and R. Chellappa, "Learning common and feature-specific patterns: A novel multiple-sparse-representation-based tracker," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 2022–2037, Apr. 2018.

[56] X. Lan, A. J. Ma, P. C. Yuen, and R. Chellappa, "Joint sparse representation and robust feature-level fusion for multi-cue visual tracking," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5826–5841, Dec. 2015.

Joey Tianyi Zhou, photograph and biography not available at the time of publication.

Heng Zhao, photograph and biography not available at the time of publication.

Xi Peng, photograph and biography not available at the time of publication.

Meng Fang, photograph and biography not available at the time of publication.

Zheng Qin, photograph and biography not available at the time of publication.

Rick Siow Mong Goh, photograph and biography not available at the time of publication.