

Efficient Cross-Modal Retrieval via Deep Binary Hashing and Quantization

Yang Shi
yang.shi@rakuten.com

Young-joo Chung
youngjoo.chung@rakuten.com

Rakuten, Inc.
San Mateo, USA

Abstract

Cross-modal retrieval aims to search for data with similar semantic meanings across different content modalities. However, cross-modal retrieval requires huge amounts of storage and retrieval time since it needs to process data in multiple modalities. Existing works focused on learning single-source compact features such as binary hash codes that preserve similarities between different modalities. In this work, we propose a jointly learned deep hashing and quantization network (HQ) for cross-modal retrieval. We *simultaneously* learn binary hash codes and quantization codes to preserve semantic information in multiple modalities by an end-to-end deep learning architecture. At the retrieval step, binary hashing is used to retrieve a subset of items from the search space, then quantization is used to re-rank the retrieved items. We theoretically and empirically show that this two-stage retrieval approach provides faster retrieval results while preserving accuracy. Experimental results on the NUS-WIDE, MIR-Flickr, and Amazon datasets demonstrate that HQ achieves boosts of more than 7% in precision compared to supervised neural network-based compact coding models.

1 Introduction

Cross-modal retrieval aims to search for data with similar semantic meanings across different content modalities, such as audio-text tag search in music recommender systems [1] and image-text search in web search [2, 3]. In cross-modal retrieval scenarios, features from different modalities are mapped to the shared space to be compared using searching and ranking algorithms. However, storing and comparing features from multiple modalities requires huge computational resources because of the size of the dataset and the number of modalities [4]. To overcome these challenges, approximate nearest neighbor search (ANN) [5] across different content modalities has gained huge attention [6]. ANN search based on compact coding methods, including binary hashing and quantization, is known to achieve an order of magnitude speed-ups compared to exact Nearest Neighbor (NN) search while preserving near-optimal accuracy [7, 8].

Compact coding methods learn efficient codes (usually binary values) that can substitute original continuous features. Binary hashing aims to learn binary codes that provide

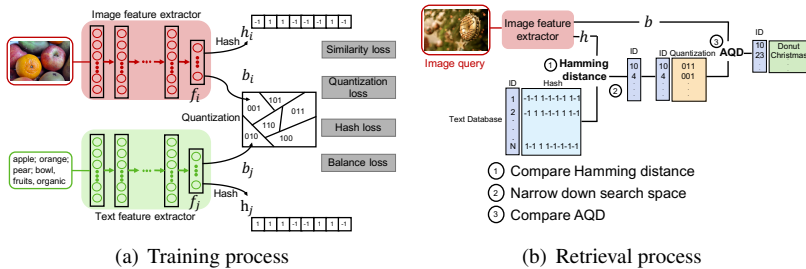


Figure 1: Deep cross-modal binary Hashing and Quantization network (HQ). In the training step, we jointly learn binary codes and quantization codes for data from different modalities. In the retrieval step, we first narrow down the search space by comparing the Hamming distance of binary codes, then measure the similarity between the query and the data in the smaller search space using quantization distance.

competitive accuracy as well as continuous features while reducing the space and the retrieval time. Following the success in single-modal retrieval [15, 24, 52, 65, 40], methods for creating binary hash codes that can connect multiple modalities have been proposed [9, 12, 17, 18, 26, 69, 47]. Furthermore, to overcome the limitation of the expressive power of binary hashing, quantization-based methods have been explored [8, 68, 46]. Quantization learns a shared lookup table (a dictionary) consisting of continuous values as well as binary codes for each data point to indicate which dictionary entry the data point is hashed to. As a result, it obtains more accurate approximations compared to simple binary hash codes. However, such quantization-based methods sacrifice computational efficiency compared to binary hashing.

In this paper, we propose a new supervised cross-modal retrieval model via deep binary Hashing and Quantization (HQ). It combines binary hashing and quantization to fully explore each other’s strengths via simultaneously learning an end-to-end deep learning network. To fully utilize both codes, we also leverage a two-stage retrieval framework that is widely used in practice [9, 10, 47]. Thus, HQ can provide more accurate results than binary hashing-based methods and require fewer computational resources than quantization-based methods. The overview of HQ is shown in Figure 1.

Our contributions are as follows:

- We propose a cross-modal retrieval model via deep binary hashing and quantization (HQ). It *simultaneously* learns binary hash codes and quantization codes using an end-to-end deep learning architecture to preserve semantic information in multiple modalities. Both codes are fully utilized by a two-stage retrieval, which first narrows down search space by comparing the Hamming distance of binary codes, and then measures the similarity between the query and the data in the smaller search space using quantization distance measurement.
- We analyze the computation and memory complexity of HQ retrieval process. We show $O(n)$ times speed-up using HQ compared to the quantization method while maintaining the same level of memory usage and retrieval accuracy (n is the feature length before quantization or binary hash).
- We demonstrate the retrieval accuracy of HQ through extensive experiments on the NUS-WIDE, MIR-Flickr, and Amazon datasets. HQ achieved boosts of more than 7%

in precision compared to supervised neural network-based models that utilize either binary hashing or quantization.

2 Preliminaries

Binary hashing Given input $x \in \mathbb{R}^n$, we compute binary code $h_x = H(x) \in \mathbb{R}^n$, where $H(\cdot)$ is a function that maps continuous value to $\{+1, -1\}$. The similarity measurement is *Hamming distance*: $\text{dist}(h_x, h_y) = \text{sum}(h_x \text{ XOR } h_y)$. The larger the value, the greater the dissimilarity between the two.

Quantization Given input $x \in \mathbb{R}^n$, we compute the quantization $x \approx \sum_{l=1}^m C^l b_x^l$, where $C^l \in \mathbb{R}^{n \times k}$ is the l -th dictionary book, $b_x^l \in \mathbb{R}^k$ is the l -th dictionary index indicator for x . We assume the input is assigned to only one entry of the dictionary: $\|b_x^l\|_0 = 1$. Different inputs will share the same C^l but will have different index indicators. We use *Asymmetric Quantizer Distance (AQD)* to measure quantization similarities: $AQD(x, y) = x^T (\sum_{l=1}^m C^l b_y^l)$. A greater value correlates with a greater similarity between the two. Generally, quantization preserves more information than binary codes, though it is slower at searching step since AQD requires more computation than Hamming distance.

3 Related Work

3.1 Cross-modal Hashing

With the increase of multi-modal data, the need for cross-modal retrieval increased, and creating compact codes for cross-modal retrieval has been extensively explored to provide efficient retrieval. Hashing methods such as Cross Modality Similarity Sensitive Hashing (CMSSH) [4], Semantic Correlation Maximization (SCM) [47], and Semantics Preserving Hashing (SePH) [29] mapped data into a common Hamming space and compared their similarities using Hamming distance. Probabilistic frameworks were also introduced to learn both the binary codes and their dimensions [14, 54]. Recently, hashing methods utilizing deep learning to create features have been explored. Starting with deep visual-semantic hashing [4] that utilized Convolutional Neural Network and Recurrent Neural Network to learn text-image modal-shared features to be hashed, Correlation Hashing Network (CHN) [5] jointly learned data representation from each modality and used a structured max-margin loss to learn similarity-preserving and high-quality hash codes. Deep Cross-modal hashing (DCMH) [21] utilized two deep neural networks, one for each modality, and learned the discrete hash codes directly through a sign function. Deep Joint-Semantics Reconstructing Hashing (DJSRH) [53] introduced a joint-semantics affinity matrix to combine the semantic similarity in each modality and improve cross-modal coding similarity. Joint-modal Distribution-based Similarity Hashing (JDSH) [30] further simplified DJSRH and proposed a sampling and weighting scheme to strengthen the discriminative ability in hash codes. UGACH [55], UCH [27] and DADH [2] used generative adversarial networks (GAN) to learn better representations from different modalities.

3.2 Cross-modal Quantization

To solve the limitation of the expressive power of binary hashing, quantization has been introduced for single/multi-modal retrieval [16, 49]. Quantized correlation hashing (QCH) [42] is the first attempt to integrate hash function learning and quantization for cross-modal retrieval. Composite Correlation Quantization (CCQ) [31] mapped both paired images and texts to shared latent space and learned composite quantizers that convert the shared latent features into compact binary codes while preserving both intra-modal similarity and inter-modal correlation. Collective Deep Quantization (CDQ) [6] introduced a deep neural network to quantization for cross-modal retrieval. Shared Predictive Deep Quantization (SPDQ) [46] created a shared subspace across different modalities and private subspaces for individual modalities. Representations are learned from both subspaces.

Little research has been conducted in combining binary hashing and quantization. Quantization-based hashing (QBH) [37] is the first attempt to incorporate the quantization-based method into the similarity-preserve hashing. It learned hashing and quantization codes *sequentially*. It minimized a similarity-preserving error to learn binary hash codes. The quantization error was calculated based on the assumption that data with the same hash code should share the same dictionary. Thus, the quantization was restricted by hashing. This approach was applied to single-modal retrieval based on hand-crafted features.

To the best of our knowledge, our work is the first attempt to integrate binary hashing and quantization in end-to-end feature learning for efficient cross-modal retrieval. Unlike previous methods, HQ learns binary hash codes and quantization codes simultaneously. We theoretically and empirically demonstrate that our proposed architecture can achieve faster retrieval time while preserving retrieval accuracy.

4 Proposed Architecture

4.1 Model Formulation

Model HQ is shown in Figure 1. We use different neural networks to extract continuous features from different content modalities. For the image network, we can train AlexNet [22] or VGG-Net [46], and for the text network, we can use the multi-layer perceptron (MLP). Given the continuous features, we learn binary codes and quantization codes by minimizing hash loss and quantization loss simultaneously.

Querying can be done in three steps: (1) compute the Hamming distance between a query item and all items in the database using the binary hash code, (2) select αN -out-of- N closest items (assume the database size is N , $0 < \alpha < 1$), and (3) compute distances between a query item and the chosen αN items using quantization codes. We use Asymmetric Quantizer Distance (AQD) to measure quantization similarities. This two-stage query is faster than directly computing quantization distances using AQD for all N points, and it is more accurate than only computing Hamming distances using binary codes.

4.2 Loss Functions

There are four parts in the loss function: (1) similarity loss, (2) hashing loss, (3) balance loss and (4) quantization loss. For simplicity, we define i, j as two data points that come from different modalities. They have continuous representation features f_i, f_j and the similarity label s_{ij} ($s_{ij} = 1$ means i and j are similar, $s_{ij} = 0$ means they are dissimilar).

Similarity loss To measure the similarity of features in different modalities, we adopt the Maximum a Posteriori (MAP) estimation. The logarithm MAP is:

$$\log p(f_i, f_j | s_{ij}) \propto \log p(s_{ij} | f_i, f_j) p(f_i) p(f_j) \quad (1)$$

In Equation 1, the conditional likelihood for similarity label s_{ij} is:

$$p(s_{ij} | f_i, f_j) = \sigma(\langle f_i, f_j \rangle)^{s_{ij}} (1 - \sigma(\langle f_i, f_j \rangle))^{1-s_{ij}} \quad (2)$$

where $s_{ij} \in \{0, 1\}$, $\sigma(x) = 1/(1 + e^{-x})$ is the sigmoid function. Assuming the prior for f_i and f_j is known, the cross entropy loss term is:

$$L_{sim} = \sum_{i,j} (\log(1 + e^{\langle f_i, f_j \rangle}) - s_{ij} \langle f_i, f_j \rangle). \quad (3)$$

Hash loss We create a binary code that has the same length as the dimension of the continuous feature. To minimize binary code error, we define the hash loss. Given the binary code $h \in \{-1, +1\}^n$, the hash loss is:

$$L_h = \sum_{i,j} (\|f_i - h_i\|_F^2 + \|f_j - h_j\|_F^2). \quad (4)$$

Balance loss To ensure we use the bit information maximally, we follow previous works [17, 18] and minimize balance loss defined as follows:

$$L_b = \sum_{i,j} (\|f_i \mathbf{1}\|_F^2 + \|f_j \mathbf{1}\|_F^2) \quad (5)$$

where $\mathbf{1}$ is a vector of 1s. It balances the number of $+1$ and -1 in a generated binary code for each training sample.

Quantization loss To minimize quantization error, we sum over the results of multiple dictionary books to approximate the original continuous features: $f \approx \sum_{l=1}^m C^l b^l$. Each dictionary book $C^l \in \mathbb{R}^{n \times k}$, binary indicator $b^l \in \{0, 1\}^k$, and $\|b^l\|_0 = 1$. Then, we can define the quantization loss as follows:

$$L_q = \sum_{i,j} (\|f_i - \sum_{l=1}^m C^l b_i^l\|_F^2 + \|f_j - \sum_{l=1}^m C^l b_j^l\|_F^2) \quad (6)$$

Combining these four loss functions, we obtain the final loss function L :

$$L = \lambda_{sim} L_{sim} + \lambda_h L_h + \lambda_b L_b + \lambda_q L_q \quad (7)$$

where $\lambda_{sim}, \lambda_h, \lambda_b, \lambda_q$ are the hyper-parameters to balance between different losses.

4.3 Learning Algorithm

We need to learn neural network parameters Θ , binary code h , dictionary book C and 1-of- k binary indicator b . Θ can be optimized by performing back propagation given loss function in Eq. 7. The optimal solution for minimizing $\|f - h\|_F^2$ is $h = \text{sgn}(f)$, where $\text{sgn}(x) = 1$ if $x \geq 0$, $\text{sgn}(x) = -1$ if $x < 0$. since $h \in \{-1, +1\}$.

We follow [18] when updating C and b . C has a closed form solution if we fix other parameters in Equation 6: $C = [F_i B_i^T + F_j B_j^T] [B_i B_i^T + B_j B_j^T]^{-1}$. F, B are the batched features and indicators for all training samples. b can be updated by thoroughly checking all possible indicators since we restrict $\|b\|_0 = 1$. In each epoch, we alternatively update C and b for a fixed number of times until they converge.

Model	Memory	Computation
Lossless	$O(32Nn)$	$O(Nn)$
Quantization	$O(32mkn + Nm \log_2 k)$	$O(mkn + Nm)$
Binary hash	$O(Nn)$	$O(Nn)$
HQ	$O(Nn + 32mkn + Nm \log_2 k)$	$O(Nn + mkn + \alpha Nm)$

Table 1: Memory and computation analysis for different models. We assume the total number of items in the database is N , dense feature dimension n , each of the m quantization dictionary book has length k .

5 Retrieval Efficiency Analysis

In this section, we analyze the computation and memory complexity of HQ and other hashing methods in the cross-modal retrieval task. We show the complexity comparison between the lossless model, the quantization model, and the hashing model in Table 1. We include detailed explanations of the computation and memory complexity and approximation error analysis in the supplemental material.

Now assume two models: (1) A model with hashing and quantization as described in Figure 1(a) (HQ); it consists of n -length binary code and $m_1 n \times k_1$ dictionaries. After hamming distance search, it narrows down the database size N to αN ($0 < \alpha \ll 1$). (2) A model with only quantization codes; it consists of $m_2 n \times k_2$ dictionaries (CDQ [8]).

Claim. *If Model one and two use the same amount of memory, retrieval time for Model one will be $O(n)$ times faster, where n is the continuous feature dimension, with some assumptions.*

Proof. The dominant terms in memory storage for Model one and two are $O(Nn + Nm_1 \log_2 k_1)$ and $O(Nm_2 \log_2 k_2)$. To make them equal, we have $O(n + m_1 \log_2 k_1) = O(m_2 \log_2 k_2)$. We assume $\log_2 k_1 \approx p_1 k_1$ and $\log_2 k_2 \approx p_2 k_2$ and $p_1 = p_2 = p$, where p is a very small constant number ($0 < p \ll 1$). Therefore, $O(m_2 k_2) = O(n/p + m_1 k_1)$.

In terms of computation complexity, the dominant terms for Model one and two require $O(Nn + \alpha Nm_1 + m_1 nk_1)$ and $O(Nm_2 + m_2 nk_2)$. The computation complexity of case two over case one is:

$$\frac{O(Nm_2 + m_2 nk_2)}{O(Nn + \alpha Nm_1 + m_1 nk_1)} = \frac{O(Nm_2 + m_1 nk_1 + n^2/p)}{O(\alpha Nm_1 + m_1 nk_1 + Nn)}. \quad (8)$$

The right side is derived by assuming Model one and two use the same amount of memory, i.e. $O(m_2 k_2) = O(n/p + m_1 k_1)$.

We can see that the retrieval time advantage is proportional to α . The more we narrow down the search space, the faster the retrieval speed is. Also, the advantage is proportional to the feature size n : we gain more benefit if feature length is large. If we assume $O(Np) = 1$, Equation 8 becomes $O(n)$. \square

6 Experiments

6.1 Dataset

We used NUS-WIDE dataset [8], MIR-Flickr dataset [20] and Amazon Review [63] for experiments. NUS-WIDE public web image dataset contains images associated with textual tags. Besides, each image-tag pair is annotated with one or multiple labels from 81 concept labels. Following the prior work [42], we used a subset of 195,834 image-tag pairs that

belong to at least one of the 21 most frequent labels. We randomly sampled 10,500 pairs as training data, 1,050 pairs as validation data, and 2,100 pairs as test data. **MIR-Flickr** consists of 25,000 images collected from the Flickr website. Each image-tag sample belongs to at least one of the 24 labels. We randomly sampled 5,000 pairs as training data, 1,000 pairs as validation data, and 2,000 pairs as test data. For **Amazon** dataset, we used image-title pairs from Grocery and Gourmet Food category. Each image-title belongs to at least one of the 14 categories. We randomly sampled 10,000 pairs as training data, 320 pairs as validation data, and 2,000 pairs as test data.

The similarity measurement is based on the concept labels associated with the images and tags. Each pair is similar if they share at least one same label, otherwise they are dissimilar. We randomly shuffled the tags and re-paired them with images to create dissimilar pairs. After the shuffle process, we have around 31%, 53%, and 21% similar pairs for NUS-WIDE, MIR-Flickr, and Amazon datasets respectively.

6.2 Implementation Details

To extract image features in NUS-WIDE and MIR-Flickr, we initialized image feature extraction networks with pre-trained VGGNet-19 [56] for Section 5.3 and with AlexNet [27] pre-trained on the ImageNet dataset [11] for Section 5.4. Amazon image features are provided by the original paper [53]. In all datasets, final image features are 128-length vectors. For text features, we followed previous work [27] and extracted 1,000, 1,386 and 1,000 most frequently used tags/words for NUS-WIDE, MIR-Flickr and Amazon, respectively, and created bag-of-word vectors for text inputs. These vectors were converted to the final 128-length text features using a two-layer MLP.

We cross-validated the hyper parameters and finally set $\lambda_q = 0.0001$, $\lambda_{sim} = 70$, $\lambda_h = 0.01$, $\lambda_b = 0.01$ for NUS-WIDE experiments, $\lambda_q = 0.0001$, $\lambda_{sim} = 50$, $\lambda_h = 0.01$, $\lambda_b = 0.01$ for MIR-Flickr experiments and $\lambda_q = 0.001$, $\lambda_{sim} = 10$, $\lambda_h = 0.01$, $\lambda_b = 0.001$ for Amazon experiments. λ_q was set especially small because we observed quantization loss decreasing much faster than other losses. $\alpha N = 100$ is set for all experiments. All experiments were performed on a NVIDIA GeForce GTX 1080 Ti. All codes were written with PyTorch 1.1.0.

6.3 Evaluation Metrics

We performed two tasks: (1) $I \rightarrow T$: retrieve relevant texts given an image query and (2) $T \rightarrow I$: retrieve relevant images given a text query. We followed prior works [6, 23] and used *Mean Average Precision* (MAP)@50 to evaluate the performances. We also computed the *Harmonic mean* of MAP@50 to measure the balanced performance between both tasks.

6.4 Results

In this section, we compare our model with 7 previous models: unsupervised cross-modal hashing methods CVH [24], LCMH [60], QBH [57], DJSRH [58], JDSH [60], and supervised methods CMSSH [9], CDQ [6] and DCMH [27]. We trained and tuned DJSRH and JDSH using the code provided by authors¹. We carefully implemented and tuned QBH, CMSSH, CDQ and DCMH. We also provided Lossless model results as references. Lossless model only uses a similarity loss and returns continuous-valued features. At the retrieval time, it

¹<https://github.com/zs1994/DJSRH>, <https://github.com/KaiserLew/JDSH>

Task	Method	NUS-WIDE				MIR-Flickr				Amazon			
		16	32	64	128	16	32	64	128	16	32	64	128
$I \rightarrow T$	Lossless	0.521	0.525	0.526	0.528	0.621	0.623	0.623	0.617	0.308	0.309	0.345	0.376
	CVH [CVH]	0.372	0.363	0.404	0.390	0.606	0.599	0.596	0.589	-	-	-	-
	LCMH [LCMH]	0.354	0.361	0.389	0.383	0.559	0.569	0.585	0.593	-	-	-	-
	QBH [QBH]	0.524	0.439	0.266	0.329	0.573	0.495	0.401	0.602	0.259	0.259	0.260	0.291
	DJSRH [DJSRH]	0.479	0.493	0.494	0.464	0.582	0.580	0.575	0.572	0.287	0.300	0.305	0.334
	JDSH [JDSH]	0.370	0.371	0.372	0.370	0.558	0.553	0.556	0.561	0.272	0.277	0.273	0.280
	CMSSH [CMSSH]	0.351	0.353	0.353	0.370	0.605	0.602	0.579	0.536	0.283	0.270	0.280	0.288
	DCMH [DCMH]	0.351	0.467	0.413	0.389	0.628	0.633	0.609	0.595	0.274	0.292	0.294	0.305
	CDQ [CDQ]	0.498	0.491	0.508	0.515	0.638	0.615	0.604	0.585	0.265	0.279	0.286	0.312
	HQ	0.542	0.449	0.508	0.411	0.588	0.583	0.610	0.583	0.293	0.301	0.332	0.354
$T \rightarrow I$	Lossless	0.489	0.485	0.552	0.548	0.641	0.630	0.648	0.643	0.309	0.310	0.324	0.354
	CVH [CVH]	0.401	0.384	0.442	0.432	0.591	0.583	0.576	0.576	-	-	-	-
	LCMH [LCMH]	0.376	0.387	0.408	0.419	0.561	0.569	0.582	0.582	-	-	-	-
	QBH [QBH]	0.250	0.294	0.447	0.337	0.604	0.614	0.523	0.521	0.273	0.274	0.274	0.281
	DJSRH [DJSRH]	0.365	0.367	0.378	0.384	0.578	0.589	0.583	0.583	0.280	0.275	0.273	0.272
	JDSH [JDSH]	0.435	0.449	0.458	0.460	0.616	0.598	0.602	0.612	0.287	0.285	0.282	0.302
	CMSSH [CMSSH]	0.393	0.393	0.383	0.381	0.560	0.588	0.603	0.607	0.274	0.271	0.273	0.304
	DCMH [DCMH]	0.392	0.358	0.367	0.406	0.619	0.621	0.647	0.603	0.298	0.284	0.290	0.289
	CDQ [CDQ]	0.419	0.422	0.525	0.433	0.606	0.601	0.626	0.623	0.283	0.299	0.311	0.345
	HQ	0.493	0.473	0.547	0.432	0.628	0.640	0.630	0.625	0.305	0.309	0.322	0.332
Harmonic mean	Lossless	0.504	0.504	0.539	0.538	0.631	0.626	0.635	0.632	0.308	0.309	0.334	0.365
	CVH [CVH]	0.386	0.373	0.422	0.410	0.598	0.591	0.586	0.582	-	-	-	-
	LCMH [LCMH]	0.365	0.374	0.398	0.400	0.560	0.569	0.583	0.587	-	-	-	-
	QBH [QBH]	0.339	0.352	0.334	0.333	0.588	0.548	0.454	0.559	0.266	0.266	0.267	0.286
	DJSRH [DJSRH]	0.414	0.421	0.428	0.420	0.580	0.584	0.579	0.577	0.283	0.287	0.288	0.300
	JDSH [JDSH]	0.400	0.406	0.411	0.410	0.586	0.575	0.578	0.585	0.279	0.281	0.277	0.291
	CMSSH [CMSSH]	0.371	0.372	0.367	0.375	0.582	0.595	0.591	0.569	0.278	0.270	0.276	0.296
	DCMH [DCMH]	0.370	0.405	0.389	0.397	0.623	0.627	0.627	0.599	0.285	0.288	0.292	0.297
	CDQ [CDQ]	0.455	0.454	0.516	0.470	0.622	0.608	0.615	0.603	0.274	0.289	0.298	0.328
	HQ	0.516	0.461	0.527	0.421	0.606	0.610	0.620	0.603	0.299	0.305	0.327	0.343

Table 2: The MAP@50 of our approach and previous models varying compact code dimension. Harmonic mean is based on $I \rightarrow T$ and $T \rightarrow I$ tasks. (CVH and LCMH results are cited from the previous paper [45].)

computes cosine similarity between features. The performance results are listed in Table 2. **Comparison to unsupervised models.** Compared to the unsupervised method DJSRH, HQ achieved a relative improvement of 14.37% of average Harmonic mean with different code dimensions on NUS-WIDE, 5.05% on MIR-Flickr, and 9.95% on Amazon. DJSRH outperformed other unsupervised and some supervised methods. This is because DJSRH measures similarities in both single and cross modalities with more parameters. For example, DJSRH has almost twice the parameters of HQ² when both models use 64-length codes. In addition, we noticed that the MAP improvements on NUS-WIDE and Amazon were higher than the ones on MIR-Flickr when comparing supervised and unsupervised models. This is because NUS-WIDE and Amazon datasets have larger variances in image and text (In NUS-WIDE and Amazon, the average number of labels for one image were 2.2 and 1.5, while it was 3.7 for MIR-Flickr.), which makes retrieval difficult without supervision.

Comparison to supervised models. CMSSH uses a boosting algorithm to learn a linear projection of the original feature and then converts it to binary codes. It does not utilize any deep neural nets. Thus, it is less accurate than the others. When we compared CDQ (the quantization model) with DCMH (the binary hashing model), CDQ boosted the average Harmonic mean on NUS-WIDE and Amazon by 21.5% and 3.4%, while decreasing 1.1% on MIR-Flickr. Compared to the second best model CDQ, HQ achieved an average Harmonic mean improvement of 1.75% and 7.24% on NUS-WIDE and Amazon.

Comparison to the model that uses both hashing and quantization. Compared to QBH,

²DJSRH requires 120M parameters while HQ requires 62M. The number of parameters was obtained by PyTorch’s model.parameters().

Task	Method	Amazon			
		16	32	64	128
$I \rightarrow T$	QBH	0.259	0.259	0.260	0.291
	HQ-b	0.286	0.298	0.295	0.310
$T \rightarrow I$	QBH	0.273	0.274	0.274	0.281
	HQ-b	0.289	0.290	0.288	0.306
H-Mean	QBH	0.266	0.266	0.267	0.286
	HQ-b	0.287	0.294	0.291	0.308

Table 3: MAP@50 of HQ-with-Hamming-distance evaluation.

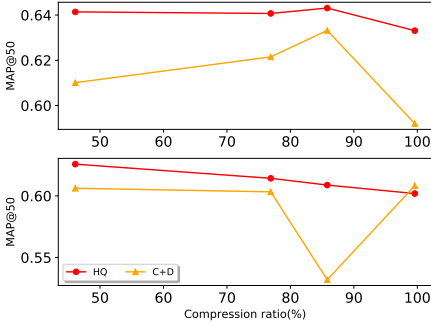


Figure 2: MAP@50 of models with different code-learning strategy on MIR-Flicker. ($I \rightarrow T$ on the top, $T \rightarrow I$ on the bottom)

which learns hashing and quantization codes sequentially, HQ achieved better performances. There are two possible reasons: (1) QBH cannot fully utilize the quantizer since it is restricted by hash codes; accordingly, the quantization error was calculated based on the assumption that data with the same hash code should share the same dictionary, and (2) QBH uses Hamming distance while HQ uses asymmetric quantizer distance which is more accurate, especially when the hash code length is short. To remove the measurement bias, we evaluate the performance of **HQ-b** which is HQ using the same distance measurement with QBH: Hamming distance between the query’s and database’s quantization codes in the second-stage retrieval. In Table 3, we observe that with the same distance measurement, HQ-b still outperforms QBH.

6.5 HQ Effectiveness Analysis

In this section, we demonstrate the effectiveness of HQ in terms of learning strategy, retrieval strategy, and retrieval efficiency.

Impact of learning two compact codes. We assume that learning binary code acts as a regularizer for learning quantization code, and vice versa. To verify this assumption, we implemented CDQ with L1 and L2 regularizers on the feature extraction neural networks.

Task	Method	Amazon			
		16	32	64	128
$I \rightarrow T$	CDQ+L1	0.259	0.289	0.259	0.264
	CDQ+L2	0.284	0.292	0.282	0.318
	HQ	0.293	0.301	0.332	0.354
$T \rightarrow I$	CDQ+L1	0.296	0.292	0.318	0.297
	CDQ+L2	0.288	0.293	0.305	0.296
	HQ	0.305	0.309	0.322	0.332
H-Mean	CDQ+L1	0.276	0.290	0.285	0.280
	CDQ+L2	0.286	0.292	0.293	0.307
	HQ	0.299	0.305	0.327	0.343

Table 4: MAP@50 of HQ and CDQ-with-regularizer models varying compact code dimension.

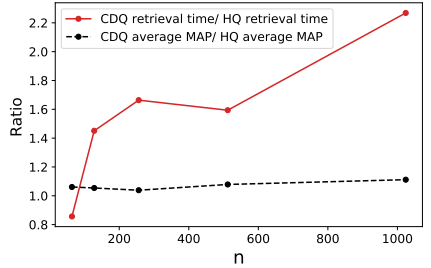


Figure 3: Retrieval efficiency and quality of HQ and CDQ on NUS-WIDE varying n . Average MAP is the average of $I \rightarrow T$ and $T \rightarrow I$ MAP@50.

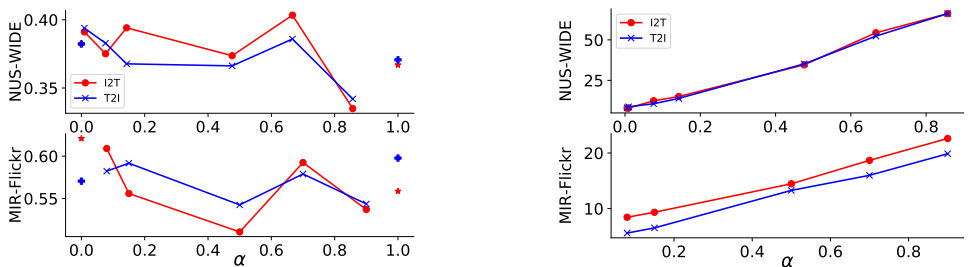


Figure 4: MAP@50 (left) and retrieval time [e^{-4} sec] (right) per query varying α .

The result is shown in Table 4. Indeed, CDQ with L1/L2 regularizer underperformed HQ and thus verified our assumption.

Impact of simultaneous code-learning. In HQ, we learn quantization codes and binary codes together. We compare HQ with another model which uses a different learning strategy while keeping the same loss target and the retrieval method: **C+D**. C+D trains a CDQ and DCMH separately, and uses the two-stage retrieval process; it uses DCMH to narrow down search space and then retrieve final results using CDQ. Unlike HQ, this model learns quantization codes and binary hash codes separately. We show MAP@50 of both models varying compression ratio in Figure 2. The compression ratio is calculated based on the memory required for storing the features. The compression ratio is $1 - \frac{P_*}{P}$ where P_* is the size of memory we need for retrieval in the compressed models, and P is the memory needed in the lossless model (See Table 1). From Figure 2, C+D did not achieve equally good results compared to HQ. This proves that learning binary hash codes and quantization codes simultaneously achieves a better result than learning them separately.

Impact of the two-stage retrieval. In HQ, we select αN -out-of- N data points at the first retrieval step. The size of the subset is important as it balances retrieval accuracy and retrieval speed. In Figure 4, we show the MAP@50 and retrieval time while varying α . $\alpha = 0.0$ indicates that we compute MAP@50 only using binary codes, and $\alpha = 1.0$ indicates we only use quantization codes to compute distance scores. We found that (1) the combination of hashing and quantization achieved the best results, (2) the best performance was achieved when α was around $0.1 \sim 0.3$ and $0.6 \sim 0.8$, (3) retrieval time was always proportional to α . Since we learn both hash and quantization codes simultaneously, the information is diffused to both codes. Therefore, using only quantization for retrieval does not yield best performance.

Retrieval efficiency In Section 5, we proved that HQ retrieval process is $O(n)$ times faster than CDQ, where n is the continuous feature dimension. We computed the retrieval time of the two models varying n while keeping the same memory usage amount in Figure 3. It showed HQ is almost linearly faster than CDQ while keeping similar retrieval accuracy.

7 Conclusions

We proposed HQ, a new cross-modal retrieval structure that combines binary hashing and quantization. It simultaneously learns both codes to preserve semantic information in multiple modalities using an end-to-end deep learning architecture. We also leveraged a two-stage retrieval method for faster and more accurate retrieval results. The experimental results demonstrated that HQ outperformed supervised neural network-based compact coding models.

References

- [1] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Commun. ACM*, pages 117–122, 2008.
- [2] Cong Bai, Chao Zeng, Qing Ma, Jinglin Zhang, and Shengyong Chen. Deep adversarial discrete hashing for cross-modal retrieval. In *Proceedings of the 2020 International Conference on Multimedia Retrieval, ICMR '20*, page 525–531. Association for Computing Machinery, 2020. ISBN 9781450370875.
- [3] Michael M. Bronstein, Alexander M. Bronstein, Fabrice Michel, and Nikos Paragios. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3594–3601, June 2010.
- [4] Yue Cao, Mingsheng Long, Jianmin Wang, Qiang Yang, and Philip S. Yu. Deep visual-semantic hashing for cross-modal retrieval. *KDD*, 2016.
- [5] Yue Cao, Mingsheng Long, Jianmin Wang, and Philip Yu. Correlation hashing network for efficient cross-modal retrieval. *CoRR*, 2016.
- [6] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. Collective deep quantization for efficient cross-modal retrieval. *AAAI*, pages 3974–3980, 2017.
- [7] Yue Cao, Bin Liu, Mingsheng Long, and Jianmin Wang. Cross-modal hamming hashing. In *The European Conference on Computer Vision (ECCV)*, September 2018.
- [8] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, and Yantao Zheng Zhiping Luo. A real-world web image database from national university of singapore. *CIVR*, 2009.
- [9] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, New York, NY, USA, 2016.
- [10] Van Dang, Michael Bendersky, and W. Bruce Croft. Two-stage learning to rank for information retrieval. In Pavel Serdyukov, Pavel Braslavski, Sergei O. Kuznetsov, Jaap Kamps, Stefan Rüger, Eugene Agichtein, Ilya Segalovich, and Emine Yilmaz, editors, *Advances in Information Retrieval*, pages 423–434, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [11] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.
- [12] Guiguang Ding, Yuchen Guo, and Jile Zhou. Collective matrix factorization hashing for multimodal data. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14*, pages 2083–2090, Washington, DC, USA, 2014. IEEE Computer Society.
- [13] Federico Avanzini Federico Simonetta, Stavros Ntalampiras. Multimodal Music Information Processing and Retrieval: Survey and Future Challenges. In *Multilayer Music Representation and Processing (MMRP) 2019 International Workshop*, 2019.

- [14] Behnam Gholami and Abolfazl Hajisami. Probabilistic semi-supervised multi-modal hashing. In Edwin R. Hancock Richard C. Wilson and William A. P. Smith, editors, *Proceedings of the British Machine Vision Conference (BMVC)*, pages 28.1–28.12. BMVA Press, September 2016.
- [15] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824, 2011.
- [16] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(12):2916–2929, December 2013. ISSN 0162-8828.
- [17] Hengtong Hu, Lingxi Xie, Richang Hong, and Qi Tian. Creating something from nothing: Unsupervised knowledge distillation for cross-modal hashing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [18] Peng Hu, Xu Wang, Liangli Zhen, and Dezhong Peng. Separated variational hashing networks for cross-modal retrieval. In *Proceedings of the 27th ACM International Conference on Multimedia*, MM '19, pages 1721–1729, New York, NY, USA, 2019. ACM.
- [19] Peng Hu, Liangli Zhen, Dezhong Peng, and Pei Liu. Scalable deep multimodal learning for cross-modal retrieval. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR'19, page 635–644, New York, NY, USA, 2019. Association for Computing Machinery.
- [20] Mark J. Huiskes and Michael S. Lew. The mir flickr retrieval evaluation. In *MIR '08: Proceedings of the 2008 ACM International Conference on Multimedia Information Retrieval*, New York, NY, USA, 2008. ACM.
- [21] Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. *CVPR*, pages 3232–3240, 2017.
- [22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Neurips*, 2012.
- [23] Brian Kulis and Kristen Grauman. Kernelized locality-sensitive hashing for scalable image search. In *ICCV*, page 2130–2137, 2009.
- [24] Shaishav Kumar and Raghavendra Udupa. Learning hash functions for cross-view similarity search. *IJCAI*, pages 1360–1365, 2011.
- [25] Saar Kuzi, ChengXiang Zhai, Yin Tian, and Haichuan Tang. Figexplorer: A system for retrieval and exploration of figures from collections of research articles. *SIGIR'20*, pages 2133–2136, 2020.
- [26] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. Self-supervised adversarial hashing networks for cross-modal retrieval. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 4242–4251, 2018.

- [27] Chao Li, Cheng Deng, Lei Wang, De Xie, and Xianglong Liu. Coupled cyclegan: Unsupervised hashing network for cross-modal retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 176–183, 2019.
- [28] Xuelong Li, Di Hu, and Feiping Nie. Deep binary reconstruction for cross-modal hashing. *IEEE Transactions on Multimedia*, pages 973–985, 2018.
- [29] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. Semantics-preserving hashing for cross-view retrieval. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3864–3872, June 2015.
- [30] Song Liu, Shengsheng Qian, Yang Guan, Jiawei Zhan, and Long Ying. Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval. *SIGIR '20*, page 1379–1388, New York, NY, USA, 2020. Association for Computing Machinery.
- [31] Mingsheng Long, Yue Cao, Jianmin Wang, and Philip S. Yu. Composite correlation quantization for efficient multimodal retrieval. *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR '16*, 2016.
- [32] Changyi Ma, Chonglin Gu, Wenye Li, and Shuguang Cui. Large-scale image retrieval with sparse binary projections. *SIGIR'20*, pages 1817–1820, 2020.
- [33] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton van den Hengel. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, page 43–52, New York, NY, USA, 2015. Association for Computing Machinery.
- [34] Bahadır Ozdemir and Larry S Davis. A probabilistic framework for multimodal retrieval using integrative indian buffet process. *Advances in Neural Information Processing Systems*, 27:2384–2392, 2014.
- [35] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. Supervised discrete hashing. In *CVPR*, pages 37–45, 2015.
- [36] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- [37] Jingkuan Song, Lianli Gao, Li Liu, Xiaofeng Zhu, and Nicu Sebe. Quantization-based hashing: a general framework for scalable image and video retrieval. *Pattern Recognition*, 75:175 – 187, 2018. Distance Metric Learning for Pattern Recognition.
- [38] Shupeng Su, Zhisheng Zhong, and Chao Zhang. Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.
- [39] Changchang Sun, Xuemeng Song, Fuli Feng, Wayne Xin Zhao, Hao Zhang, and Liqiang Nie. Supervised hierarchical cross-modal hashing. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*,

- SIGIR'19, page 725–734, New York, NY, USA, 2019. Association for Computing Machinery.
- [40] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431, 2010.
- [41] Lidan Wang, Jimmy Lin, and Donald Metzler. A cascade ranking model for efficient ranked retrieval. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 105–114, New York, NY, USA, 2011. Association for Computing Machinery.
- [42] Wei Wang, Beng Chin Ooi, , Xiaoyan Yang, Dongxiang Zhang, and Yueting Zhuang. Effective multi-modal retrieval based on stacked auto-encoders. *VLDB*, 2014.
- [43] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21, pages 1753–1760. Curran Associates, Inc., 2009.
- [44] Botong Wu, Qiang Yang, Wei-Shi Zheng, Yizhou Wang, and Jingdong Wang. Quantized correlation hashing for fast cross-modal search. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 3946–3952. AAAI Press, 2015.
- [45] Gengshen Wu, Zijia Lin, Jungong Han, Li Liu, Guiguang Ding, Baochang Zhang, and Jialie Shen. Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval. *IJCAI*, pages 2854–2860, 2018.
- [46] Erkun Yang, Cheng Deng, Chao Li, Wei Liu, Jie Li, and Dacheng Tao. Shared predictive cross-modal deep quantization. *IEEE Transaction On Neural Networks And Learning Systems*, 2019.
- [47] Dongqing Zhang and Wu-Jun Li. Large-scale supervised multimodal hashing with semantic correlation maximization. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, pages 2177–2183. AAAI Press, 2014.
- [48] Jian Zhang, Yuxin Peng, and Mingkuan Yuan. Unsupervised generative adversarial cross-modal hashing. In *Proceedings of the Thirty-second AAAI Conference on Artificial Intelligence*, AAAI'18. AAAI Press, 2018.
- [49] Ting Zhang and Jingdong Wang. Collaborative quantization for cross-modal similarity search. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016.
- [50] Xiaofeng Zhu, Zi Huang, Heng Tao Shen, and Xin Zhao. Linear cross-modal hashing for efficient multimedia search. *ACM Multimedia*, pages 143–152, 2013.