Separated Variational Hashing Networks for Cross-Modal Retrieval

Peng Hu[1,4,†], Xu Wang[1,†], Liangli Zhen[5], Dezhong Peng[1,2,3,*]
[1]Machine Intelligence Laboratory, College of Computer Science, Sichuan University, Chengdu, China [2]Chengdu Sobey Digital Technology Co., Ltd., Chengdu, China [3]Shenzhen Peng Cheng Laboratory, Shenzhen, China [4]Institute for Infocomm Research, A*STAR, Singapore [5]Institute of High Performance Computing, A*STAR, Singapore

ABSTRACT

Cross-modal hashing, due to its low storage cost and high query speed, has been successfully used for similarity search in multimedia retrieval applications. It projects high-dimensional data into a shared isomorphic Hamming space with similar binary codes for semantically-similar data. In some applications, all modalities may not be obtained or trained simultaneously for some reasons, such as privacy, secret, storage limitation, and computational resource limitation. However, most existing cross-modal hashing methods need all modalities to jointly learn the common Hamming space, thus hindering them from handling these problems. In this paper, we propose a novel approach called Separated Variational Hashing Networks (SVHNs) to overcome the above challenge. Firstly, it adopts a label network (LabNet) to exploit available and nonspecific label annotations to learn a latent common Hamming space by projecting each semantic label into a common binary representation. Then, each modality-specific network can separately map the samples of the corresponding modality into their binary semantic codes learned by LabNet. We achieve it by conducting variational inference to match the aggregated posterior of the hashing code of LabNet with an arbitrary prior distribution. The effectiveness and efficiency of our SVHNs are verified by extensive experiments carried out on four widely-used multimedia databases, in comparison with 11 state-of-the-art approaches.

CCS CONCEPTS

• **Information systems → Multimedia and multimodal retrieval**.

KEYWORDS

Cross-modal retrieval, cross-modal hashing, common Hamming space, separated variational hashing network.

*Corresponding author.
†First two authors contributed equally to this work.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351078

ACM Reference Format:
Peng Hu, Xu Wang, Liangli Zhen, Dezhong Peng. 2019. Separated Variational Hashing Networks for Cross-Modal Retrieval. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351078

1 INTRODUCTION

While moving ahead with big data, there are a tremendous growth of large-scale and high-dimensional data with various modalities (*e.g.*, image and text) on the Internet. To exploit these multimodal data, a large number of research efforts have devoted to the cross-modal retrieval in recent years [1–10]. Cross-modal retrieval is set to search relevant samples across heterogeneous media modalities. Cross-modal hashing, due to its low storage cost and high query speed, has been a hot topic in the research of cross-modal retrieval. It projects different types of high-dimensional data into a shared isomorphic Hamming space with similar binary codes for semantically-similar data. Two main difficulties in cross-modal hashing are 1) the semantic gap between low-level features and high-level semantics [11] and 2) the cross-modal heterogeneity gap induced by the inconsistent distributions of different modalities [12, 13]. Although the semantic gap can be reduced by deep learning, the heterogeneity gap remains a challenging problem which needs further studies to conquer.

Over the past few years, a considerable amount of works have raised to address the issue of heterogeneity [12–22]. A common strategy employed in these methods is to learn a common Hamming space, where the semantic-similarity among different modalities can be directly measured by employing Hamming distance. These approaches fall into two major groups, *i.e.*, the shallow models and the deep ones. The shallow methods [19, 20, 23, 24] project different modalities into a common Hamming space by learning linear single-layer transformations. On account of the limited representative capacity, these linear architectures cannot effectively narrow the heterogeneity gap. By exploiting the strong representation ability of deep neural networks, various deep cross-modal hashing methods [12, 15, 25–27] have been developed to learn more effective nonlinear transformations, and have achieved promising performance. Despite the great progress made by these methods, there are still several limitations: 1) They usually require training all modality-specific networks jointly, costing a lot of computing and storage resources; 2) Most of the methods are specifically designed for two modalities, and they need $\frac{m(m-1)}{2}$ runs for m modalities (usually $m > 2$) in a pairwise manner, which is time-consuming for multimodal datasets.

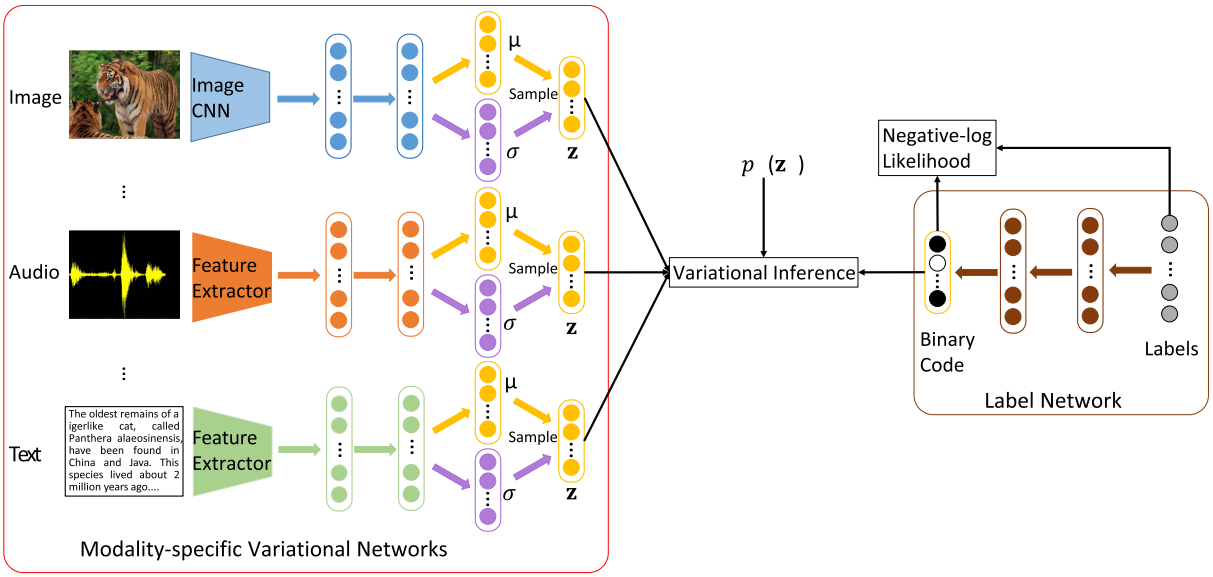


Figure 1: The basic idea of our SVHNs. Firstly, the label network (LabNet) is trained from all available labels. Then, the modality-specific variational networks (MVNs) separately approximate the binary semantics obtained by the pretrained LabNet with variational inference.

To address these issues and to improve the retrieval performance, we propose a novel Separated Variational Hashing Networks (SVHNs) approach for cross-modal retrieval. Our method is trained under two stages: 1) All available labels are employed to train the label network (LabNet), which is desired to generate discriminative hash codes. Due to the discriminative information in labels are "purer" than that in data, the generated hash codes are more discriminative than those produced by modalities; 2) The modality-specific variational networks (MVNs) are separately trained by variational inference, which makes the latent variables approximate to the generated hash codes from LabNet and be regularized by a prior distribution. Different from existing cross-modal hashing methods [12, 13, 15, 26], the proposed SVHNs can separately train each modality (in the second training stage), without requiring all modalities to be simultaneously available, thus reducing resource usages. Besides, the proposed SVHNs can transform different modalities into a common Hamming space with only a program run, which is less time-consuming for large datasets.

The contributions of our work can be summarized as follows:

- A novel hashing model is proposed to separately learn a common Hamming space for cross-modal retrieval, which can fully and independently exploit each modality without the pairwise limitation.
- We design a Label Network (LabNet) to independently extract the discriminative binary representations from the available single- or multi-label annotations. Then, the common discriminative Hamming space is pre-learned by our LabNet to separate each modality from the joint cross-modal training.
- We integrate variational inference with separated cross-modal hashing learning to separately preserve the semantic relevance across modalities as much as possible and be suitable for the out-of-sample extension. Therefore, our SVHNs can achieve better performance without relaxation, as demonstrated in our experimental studies.

2 RELATED WORK

Over the past few years, plentiful cross-modal hashing methods have come forth in the literature. These works are of two broad: the shallow settings and the deep ones. This section will briefly review the related approaches from these two categories.

Shallow Cross-modal Hashing: During the early research of cross-modal hashing, most of the efforts focus on shallow models, which learn linear or nonlinear single-layer transformations to project different modalities into a common Hamming space. These shallow methods can be further divided into two general classes: unsupervised and supervised. The unsupervised methods, represented by [19, 28, 29], make attempts to reduce the heterogeneity gap via maximizing pairwise statistical concomitant relationships. For example, Liu *et.al.* [19] propose a Fusion Similarity Hashing (FSH) to explicitly embed the graph-based fusion similarity across modalities into a common Hamming space. Ding *et.al.* [28] proposed a Collective Matrix Factorization Hashing (CMFH) method to learn unified hash codes by collective matrix factorization with latent factor model. By comparison, the supervised approaches, represented by [20, 23, 24, 30], exploit the discriminating information to enhance the closeness of various modalities, obtaining a more discriminative common Hamming space. For instance, Lin *et.al.* [23] proposed Semantics-preserving Hashing (SePH) to employ the semantic affinity of training data as the supervisor. Subsequently, SePH was extended by utilizing predictive models (*e.g.*, linear ridge regression, logistic regression, and kernel logistic regression) as the hash functions to project the corresponding modality-specific features into hash codes. Li *et.al.* [30] presented a supervised Linear Subspace Ranking Hashing framework (LSRH), which maps data from different modalities into a common Hamming space,

and employs Hamming distance as the cross-modal similarity. Different from the above-mentioned shallow methods, the proposed SVHNs method utilizes several hashing networks to nonlinearly transform the multimodal data into a common Hamming space, which provides stronger scalability, nonlinearity and representational capacity for cross-modal hashing.

Deep Cross-modal Hashing: Thanks to the great success of deep neural networks (DNNs) [13, 15, 31, 32], several deep cross-modal hashing approaches have presented in recent years. These deep models exploit the strong representation ability of DNNs, achieving favorable performance. Likewise, they can be divided into unsupervised and supervised methods. As an example of the unsupervised ones, Zhang *et.al.* [25] proposed an Unsupervised Generative Adversarial Cross-modal Hashing approach (UGACH) to employ GAN's ability to exploit the underlying manifold structure of cross-modal data. To utilize the discrimination information, some supervised methods have been developed. For example, Jiang and Li [15] developed a Deep Cross-modal Hashing (DCMH) to employ a negative log-likelihood loss to maintain cross-modal similarities. Motivated by the strong ability of adversarial learning in modeling data distribution, Li *et.al.* [12] presented a Self-Supervised Adversarial Hashing approach (SSAH), which aims to maintain the semantic correlation and consistency of the representations between different modalities. In addition, Liong *et.al.* [13] proposed a cross-modal deep variational hashing (CMDVH) which is the one most closely related to ours. Compared with our method, CMDVH also trained under two steps. However, in the first step, all modalities and labels should be jointly used to train the fusion network to learn the inferred binary codes by a coupled of DNNs, whereas our SVHNs method learns the semantic binary codes from available labels, which carry "purer" discriminant information than modalities. Furthermore, each modality can be separately used to train its corresponding MVN, leading to separated training for each modality in our model. Besides, our SVHNs can project multimodal data into a common single Hamming space, thus efficiently processing more than two modalities.

3 SEPARATED VARIATIONAL HASHING NETWORKS

3.1 Problem Formulation

For a clear description, we first give some definitions. The samples of the k-th modality are denoted as $\mathcal{X}^k = \{\mathbf{x}_i^k\}_{i=1}^{N_k}$, where N_k is the number of the instances from the k-th modality, and $\mathbf{x}_i^k$ is the i-th sample of the k-th modality. The corresponding labels are denoted as $\mathcal{Y}^k = \{\mathbf{y}_i^k\}_{i=1}^{N_k}$, where $\mathbf{y}_i^k \in \mathbb{R}^{c\times 1}$ is a binary-valued label annotation assigned to $\mathbf{x}_i^k$ and c is the number of classes. If the i-th instance of the k-th modality belongs to the j-th class $y_{ij}^k = 1$, otherwise $y_{ij}^k = 0$.

The goal of cross-modal hashing is to learn a unified binary representation for the multiple modalities: $\mathcal{B} = \{\mathbf{B}^k\}_{k=1}^m$, where m is the number of modalities, $\mathbf{B}^k = [\mathbf{b}_1^k, \cdots, \mathbf{b}_i^k, \cdots, \mathbf{b}_{N_k}^k]$ is the hash code matrix of the k-th modality, $\mathbf{b}_i^k \in \{-1, 1\}^L$ is the hash code of $\mathbf{x}_i^k$, and L is the length of the binary code. The similarity between two binary codes is evaluated using the Hamming distance. The relationship between the Hamming distance $d(\mathbf{b}_i^k, \mathbf{b}_j^l)$ and the inner product $\langle \mathbf{b}_i^k, \mathbf{b}_j^l \rangle$ can be formulated using $d(\mathbf{b}_i^k, \mathbf{b}_j^l) = \frac{1}{2}(L - \langle \mathbf{b}_i^k, \mathbf{b}_j^l \rangle)$. Therefore, we can use the inner product to quantify the similarity of two binary codes, *i.e.*, $S(\mathbf{b}_i^k, \mathbf{b}_j^l) = \frac{1}{2}\langle \mathbf{b}_i^k, \mathbf{b}_j^l \rangle$.

Since the multimodal data typically have different statistical properties and follow inconsistent distributions, they cannot be directly compared with each other for cross-modal retrieval. Multimodal hashing attempts to learn m modality-specific functions $\{f_k(\cdot)\}_{k=1}^m$ to project the corresponding modalities into a common Hamming space, where the hash codes of different modalities can be directly compared with each other. Then the similarities between different modalities can be computed from the obtained common hash codes for cross-modal retrieval. Furthermore, in the common Hamming space, the similarity of the samples from the same class is desired to be larger than the similarity of the samples from different categories.

3.2 Label Network

The labels from different modalities have indistinctive forms, and they are more readily available than pair-wise multimodal data in the real-world applications. The label is the direct carrier of semantics. Therefore, the accurate binary semantic codes can be directly learned from the available labels. Furthermore, the dimension of the labels is much smaller than the ones of the multimodal data. Thus, it will cost much less computational resources to train the label network on the labels. We merge all the available labels from all modalities as a whole label set denoted as $\mathcal{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$, where N is the total number of these available labels, $\mathbf{y}_i \in \{0, 1\}^c$, c is the number of classes, $y_{ij} = 1$ if $\mathbf{y}_i$ has the semantics of the j-th class and 0 otherwise. Then, we design a neural network to transform a semantic label into a discriminative binary feature vector, denoted as $g(\mathbf{y}, \theta) \in \mathbb{R}^L$ with parameters θ. The output of LabNet is defined as follows:

$$\mathbf{u}_i = g(\mathbf{y}_i) \in \mathbb{R}^L. \tag{1}$$

The hash codes of $\mathbf{y}_i$ can be obtained as follows:

$$\mathbf{h}_i = \operatorname{sgn}(g(\mathbf{y}_i)) \in \{-1, 1\}^L, \tag{2}$$

where $\operatorname{sgn}(\cdot)$ is the sign function that extracts the sign of a real number. The LabNet is desired to project the semantic labels into a Hamming space, in which the labels with the same semantics are compact and ones without the same semantics are scattered. Then, the objective function of LabNet is defined as follows:

$$\begin{aligned}\arg\min_{\mathbf{H},\theta} \mathcal{J} &= \mathcal{J}_1 + \lambda \mathcal{J}_2 \\ &= \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\log(1 + e^{S_{ij}}) - \Delta_{ij} S_{ij}\right) \\ &\quad + \frac{\lambda}{N}\sum_{i=1}^{N} \| |\mathbf{u}_i| - \mathbf{1} \|_2^2,\end{aligned} \tag{3}$$

where $\Delta_{ij} = \mathbf{1}\{S(\mathbf{y}_i, \mathbf{y}_j)\}$, $S_{ij} = S(\mathbf{u}_i, \mathbf{u}_j) = \frac{1}{2}\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \frac{1}{2}\mathbf{u}_i^T \mathbf{u}_j$, λ is a constant parameter to balance the effect of the two items and $\mathbf{1}\{\cdot\}$ is an indicator function whose value is 0 if the element is 0 otherwise 1, $\|\cdot\|_2$ is ℓ-2 norm, $|\cdot|$ is absolute value function, and $\mathbf{1} \in \mathbb{R}^L$ is a vector with all elements as 1. The first term of

Equation (3) is the negative-log likelihood of the label similarities with likelihood function defined as follows:

$$p(\Delta_{ij}|\mathbf{u}_i, \mathbf{u}_j) = \begin{cases} \delta(S_{ij}) & \text{if } \Delta_{ij} = 1; \\ 1 - \delta(S_{ij}) & \text{otherwise,} \end{cases} \tag{4}$$

where $\delta(S_{ij}) = \frac{1}{1+e^{-S_{ij}}}$ is the sigmoid function. It is easy to find that minimizing this negative-log likelihood function is equivalent to maximizing the likelihood. We can also see that, the larger similarity S_{ij} is, the larger $p(\Delta_{ij}|\mathbf{u}_i, \mathbf{u}_j)$ will be, and vice versa. Therefore, Equation (3) is a reasonable similarity measure for common representations and is a good criterion for learning discriminative features. The second term of Equation (3) aims to constrain the obtained features as binary codes.

With the obtained objective function in Equation (3), the proposed LabNet can be iteratively optimized with back-propagation in an end-to-end manner. Therefore, the overall network can be optimized using a stochastic gradient descent optimization algorithms, like ADAM [33]. The detailed optimization process is shown in Algorithm 1.

Algorithm 1 Optimization procedure of LabNet

Input: All available labels $\mathcal{Y}$, the length of the binary code L, batch size N_b, positive balance parameter λ, learning rate α

1: **while** not converge **do**
2: Randomly select N_b labels from $\mathcal{Y}$ to construct a mini-batch.
3: Compute the output of LabNet for the mini-batch according to Equation (1) as $\mathbf{u}_i = g(\mathbf{y}_i)$.
4: Compute the loss $\mathcal{J}$ for the obtained output according to Equation (3).
5: Update the parameters of LabNet by minimizing $\mathcal{J}$ with descending their stochastic gradient as $\theta = \theta - \alpha \frac{\partial \mathcal{J}}{\partial \theta}$.
6: **end while**

Output: Optimized LabNet model.

4 MODALITY-SPECIFIC VARIATIONAL NETWORKS

Inspired by the success of variational networks [34], we utilize a probabilistic approximation to learn common representations for cross-modal retrieval using m modality-specific variational networks (MVNs). In [34], Kingma *et al.* proposed a learned approximate posterior inference generative model with a neural network, called the Variational Auto-Encoder (VAE). This latent probabilistic generative manner tends to produce flexible general features and capture diversity from the inputs, which makes the model more general and suitable for the out-of-sample extensions [34, 35]. The latent variable is modeled by an approximate inference model with the given data points and a prior distribution.

In our model, we assume that the discrete semantics of a given datapoint $\mathbf{x}_i^k$ for the k-th modality and the corresponding latent representation $\mathbf{z}_i^k$ can be defined by a posterior distribution as $p(\mathbf{z}_i^k|\mathbf{x}_i^k)$, which is the intractable true posterior distribution [34]. Like [34], we introduce a recognition model $q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k)$, *i.e.*, our MVNs, to approximate the intractable true posterior distribution. In this case, we can let the variational approximate posterior distribution $q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k)$ be a multivariate Gaussian with a diagonal covariance structure as follows:

$$q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_i^k, \left(\boldsymbol{\sigma}_i^k\right)^2 \mathbf{I}), \tag{5}$$

where the mean $\boldsymbol{\mu}_i^k \in \mathbb{R}^L$ and the standard deviation $\boldsymbol{\sigma}_i^k \in \mathbb{R}^L$ are the outputs of the k-th nonlinear MVN $f_k(\cdot; \boldsymbol{\Theta}_k)$ for the datapoint $\mathbf{x}_i^k$ of the k-th modality, *i.e.*, $\mathbf{x}_i^k \xrightarrow{f_k} (\boldsymbol{\mu}_i^k, \boldsymbol{\sigma}_i^k)$, where $f_k(\cdot; \boldsymbol{\Theta}_k)$ is the nonlinear function of the k-th MVN for the k-th modality with parameters $\boldsymbol{\Theta}_k$.

To solve our problem by an alternative method for correlating the semantics of $\mathbf{x}_i^k$ and the representation $\mathbf{z}_{ij}^k$ obtained by $q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k)$, we adopt the reparameterization trick [34] to sample the random variable $\mathbf{z}_{ij}^k$. With the reparameterization trick, we sample $\mathbf{z}_{ij}^k$ from the posterior $\mathbf{z}_{ij}^k \sim q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k)$ using

$$\mathbf{z}_{ij}^k = \boldsymbol{\mu}_i^k + \boldsymbol{\sigma}_i^k \odot \boldsymbol{\epsilon}_j, \tag{6}$$

where $\boldsymbol{\epsilon}_j \in \mathbb{R}^L$ is a j-th auxiliary variable vector with $\boldsymbol{\epsilon}_j \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\odot$ is an element-wise product. Equation (6) makes the latent representation differentiable and capable of back-propagation [35].

Unlike the traditional VAE, our MVNs utilize a posterior inference model to encode the modality-specific data as the common semantic binary codes learned by the LabNet from their semantic labels. The encoders, *i.e.*, MVNs, aim at transforming the input to the latent variable with parameters $\{\Theta_k\}_{k=1}^m$, and the decoders attempt to transform the latent variable to the hash codes of the label input without any trainable parameters. Then, according to [34], the objective function of the k-th MVN can be formulated as follows:

$$\begin{aligned} \mathcal{L}_k &\simeq \frac{1}{N_k} \sum_{i=1}^{N_k} D_{KL}\left(q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k) \| p(\mathbf{z})\right) \\ &\quad - \frac{1}{N_k} \sum_{i=1}^{N_k} \mathbb{E}_{\mathbf{z}\sim q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k)}\left[\log p(\mathbf{x}_i^k|\mathbf{z})\right] \\ &= \frac{1}{2N_k} \sum_{i=1}^{N_k} \sum_{l=1}^{L} \left(\left(\mu_{il}^k\right)^2 + \left(\sigma_{il}^k\right)^2 - \log\left((\sigma_{il}^k)^2\right) - 1\right) \\ &\quad - \frac{1}{N_k J} \sum_{i=1}^{N_k} \sum_{j=1}^{J} \left(\log p(\mathbf{x}_i^k|\mathbf{z}_{ij}^k)\right), \end{aligned} \tag{7}$$

where $D_{KL}(\cdot)$ is the Kullback-Leibler divergence, J is the sampling number of each datapoint, and μ_{ij}^k and σ_{ij}^k denote the j-th element of vectors $\boldsymbol{\mu}_i^k$ and $\boldsymbol{\sigma}_i^k$, respectively. The first term of Equation (7) (the Kullback-Leibler divergence of the approximate posterior distribution from the prior one) is a regularizer to enforce the variational approximate posterior distribution $q_{\Theta_k}(\mathbf{z}|\mathbf{x}_i^k)$ to obey the prior distribution $p(\mathbf{z})$, where $p(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \mathbf{0}, \mathbf{I})$ is the centered isotropic multivariate Gaussian distribution. For the second item of Equation (7), the sample $\mathbf{z}_{ij}^k$ is then inputted to the function $\log p(\mathbf{x}_i^k|\mathbf{z}_{ij}^k)$, which equals the probability density of the latent semantics for the datapoint $\mathbf{x}_i^k$, thus we have $p(\mathbf{x}_i^k|\mathbf{z}_{ij}^k) = e^{-\beta\|\mathbf{h}_i^k - \mathbf{z}_{ij}^k\|_2^2}$ which is Heat

kernel with a positive parameter $\beta \in \mathbb{R}^+$, where $\mathbf{h}_i^k = \text{sgn}\left(g(\mathbf{y}_i^k)\right)$ with the trained LabNet $g(\cdot)$. This item aims to make the latent representation $\mathbf{z}$ approximate its hash semantics by the modality-specific network $f_k(\cdot, \Theta_k)$. Therefore, the loss function can be rewritten as follows:

$$\mathcal{L}_k \simeq \frac{1}{2N_k}\sum_{i=1}^{N_k}\sum_{l=1}^{L}\left(\left(\mu_{il}^k\right)^2 + \left(\sigma_{il}^k\right)^2 - \log\left((\sigma_{il}^k)^2\right) - 1\right) + \frac{\beta}{N_k J}\sum_{i=1}^{N_k}\sum_{j=1}^{J}\|\mathbf{h}_i^k - \mathbf{z}_{ij}^k\|_2^2. \tag{8}$$

This objective function allows training each MVN with back-propagation in an end-to-end manner. In this work, we optimize our each MVN model by using a stochastic gradient descent-based optimization algorithms, *e.g.*, ADAM [33]. The detailed optimization process for the k-th modality is summarized in Algorithm 2.

Algorithm 2 Optimization procedure of the k-th MVN

Input: The training data of the k-th modality $\mathcal{X}^k$, the corresponding labels $\mathcal{Y}_k$, the length of the binary code L, the batch size N_b, the positive parameter β, the number of samples J per datapoint, the learning rate α.

1: **while** not converge **do**
2: Randomly select N_b datapoints from $\mathcal{X}^k$ and $\mathcal{Y}_k$ to construct a mini-batch.
3: Compute the hash codes by the learned LabNet for the mini-batch according to Equation (2) as $\mathbf{h}_i^k = \text{sgn}\left(g(\mathbf{y}_i^k)\right), i = 1, 2, \cdots, N_b$.
4: Compute the outputs of the k-th MVN for the mini-batch.
5: Randomly sample the latent variable $\mathbf{z}$ from the obtained outputs for the mini-batch according to Equation (6) as $\mathbf{z}_{ij}^k = \boldsymbol{\mu}_i^k + \boldsymbol{\sigma}_i^k \odot \boldsymbol{\epsilon}_j, i = 1, 2, \cdots, N_b, j = 1, 2, \cdots, J$.
6: Compute the loss $\mathcal{L}_k$ with the obtained hash codes and variables according to Equation (2).
7: Update the parameters of the k-th MVN by minimizing $\mathcal{L}_k$ with descending their stochastic gradient as $\Theta_k = \theta - \alpha\frac{\partial \mathcal{L}_k}{\partial \Theta_k}$.
8: **end while**

Output: Optimized the k-th MVN model.

From Algorithm 2, we can see that each MVN can be trained separately with each other. It has the following four benefits: 1) Our SVHNs can tackle the separated modalities without combining them; 2) Separate training can save the computational resource; 3) Separate training can speed up the training stage; 4) Our SVHNs do not have the pairwise limitation of all modalities. Like other cross-modal hashing learning methods, different MVNs of our SVHNs are used to extract the unified hash code from different formats of the input data. The hash code of $\mathbf{x}_i^k$ can be obtained by the k-th learned MVN as follows:

$$\mathbf{b}_i^k = \text{sgn}(\boldsymbol{\mu}_i^k) \in \{-1, 1\}^L. \tag{9}$$

With the learned common hash codes, the multimodal data can be correlated by a common Hamming distance metric. The effectiveness of the proposed method is verified by extensive experiments carried out on the widely-used cross-modal datasets.

5 EXPERIMENTAL STUDY

To evaluate our SVHNs, we conduct experiments on four cross-modal datasets, namely, PKU XMedia [36], MIRFLICKR-25K [37], NUS-WIDE [38], and MS-COCO [39]. In the following experiments, the effectiveness of the proposed method is verified in comparison with 11 state-of-the-art cross-modal real-valued and hashing methods. Furthermore, additional evaluations are conducted to investigate the performance of our SVHNs in more detail.

5.1 Experimental Setup

Table 1: General statistics of the three datasets used in the experiments, where "*/*/*" in the "Instance" column stands for the number of total/training/query sets.

Dataset	Label	Modality	Instance	Feature
PKU XMedia	20	Image	5,000/4,000/1,000	4,096D VGG
		Text	5,000/4,000/1,000	3,000D BoW
		Audio clip	1,000/800/200	29D MFCC
		3D model	500/400/100	4,700D LightField
		Video	1,143/969/174	4,096D C3D
MIRFLICKR-25K	24	Image	20,015/10,000/2,000	4,096D VGG
		Text	20,015/10,000/2,000	1,386D BoW
NUS-WIDE	21	Image	190,421/10,500/2,100	4,096D VGG
		Text	190,421/10,500/2,100	1,000D BoW
MS-COCO	80	Image	122,218/10,000/5,000	4,096D VGG
		Text	122,218/10,000/5,000	300D Doc2Vec

5.1.1 Datasets and Compared Methods. For a fair comparison, we follow the partitions of the training and query subsets in [12, 36]. The statistics of the four datasets are summarized in Table 1. Furthermore, to our best knowledge, there is no cross-modal hashing method to transform more than two modalities to a learned single common space. Thus we just compare our proposed SVHNs with six hashing approaches (*i.e.*, SePH [23], SePH$_{lr}$ [24], RoPH [40], LSRH [30], DCMH [15], and SSAH [12]) on the two-modality datasets (*i.e.*, MIRFLICKR-25K, NUS-WIDE and MS-COCO), and with four real-valued multimodal methods (*i.e.*, MCCA [3], GMLDA [9], MvDA [18], and MvDA-VC [17]) on the five-modality dataset (*i.e.*, PKU XMedia) to evaluate the effectiveness of the single common space learned by these methods. Furthermore, to investigate the performance of traditional cross-modal hashing methods, *i.e.*, SePH, SePH$_{lr}$, RoPH and LSRH, for cross-modal retrieval on the PKU XMedia dataset, these methods were performed 10 times to learn 10 cross-modal pairwise spaces in a pairwise manner on the dataset. The results of CMDVH [13] and SSAH [12]) are provided by their authors on the MIRFLICKR-25K and NUS-WIDE datasets. In addition, it should be noted that the feature extractors (VGGNet [41] and Doc2Vec [42]) are not fine-tuned in our training stage for a fair comparison with other shallow methods.

5.1.2 Implementation Details. Our proposed SVHNs approach is trained on two GTX 1080Ti graphics cards and a 3.50GHz i7-7800X

Table 2: Performance comparison in terms of mAP scores on the PKU XMedia dataset. The best result is shown in boldface.

Method	Query	Image				Text				Audio				3D				Video				Avg.
	Database	Text	Audio	3D	Video	Image	Audio	3D	Video	Image	Text	3D	Video	Image	Text	Audio	Video	Image	Text	Audio	3D	
MCCA [3]		0.115	0.145	0.172	0.125	0.120	0.124	0.147	0.115	0.133	0.114	0.176	0.137	0.126	0.095	0.122	0.104	0.090	0.077	0.094	0.104	0.122
GMLDA [9]		0.614	0.150	0.855	0.622	0.625	0.131	0.747	0.504	0.255	0.178	0.228	0.157	0.489	0.422	0.121	0.433	0.372	0.305	0.102	0.443	0.388
MvDA [18]		0.623	0.295	0.882	0.669	0.616	0.242	0.767	0.564	0.287	0.237	0.416	0.270	0.456	0.393	0.229	0.426	0.322	0.266	0.153	0.432	0.427
MvDA-VC [17]		0.655	0.221	0.879	0.710	0.645	0.186	0.762	0.599	0.244	0.209	0.371	0.231	0.532	0.458	0.194	0.501	0.429	0.358	0.133	0.529	0.442
SePH (16 bits) [23]*		0.860	0.572	0.843	0.728	0.875	0.611	0.887	0.815	0.449	0.460	0.446	0.394	0.520	0.563	0.377	0.487	0.374	0.391	0.303	0.448	0.570
SePH$_{lr}$ (16 bits) [24]*		0.861	0.212	0.852	0.819	0.950	0.220	0.847	0.862	0.223	0.234	0.326	0.232	0.461	0.471	0.181	0.409	0.348	0.375	0.138	0.432	0.473
RoPH (16 bits) [40]*		0.770	0.717	0.841	0.751	0.711	0.535	0.328	0.497	0.426	0.331	0.477	0.397	0.452	0.186	0.374	0.384	0.382	0.291	0.397	0.250	0.475
LSRH (16 bits) [30]*		0.731	0.273	0.588	0.487	0.781	0.323	0.674	0.499	0.297	0.340	0.232	0.243	0.373	0.483	0.174	0.299	0.280	0.351	0.182	0.282	0.395
SVHNs (16 bits)		**0.910**	**0.759**	**0.910**	**0.909**	**0.962**	**0.800**	**0.962**	**0.962**	**0.563**	**0.563**	**0.573**	**0.580**	**0.617**	**0.617**	**0.491**	**0.626**	**0.553**	**0.553**	**0.446**	**0.555**	**0.695**
SePH (32 bits) [23]*		0.873	0.630	0.872	0.795	0.902	0.672	0.899	0.848	0.468	0.497	0.524	0.438	0.584	0.602	0.405	0.515	0.472	0.470	0.384	0.461	0.615
SePH$_{lr}$ (32 bits) [24]*		0.891	0.282	0.854	0.863	0.949	0.289	0.907	0.895	0.265	0.289	0.340	0.299	0.499	0.516	0.177	0.449	0.400	0.456	0.157	0.441	0.511
RoPH (32 bits) [40]*		0.817	0.775	0.865	0.803	0.773	0.581	0.341	0.559	0.457	0.371	0.499	0.457	0.448	0.223	0.385	0.420	0.409	0.334	0.406	0.284	0.510
LSRH (32 bits) [30]*		0.887	0.351	0.766	0.670	0.927	0.376	0.770	0.711	0.325	0.335	0.263	0.280	0.517	0.535	0.218	0.420	0.355	0.458	0.221	0.353	0.487
SVHNs (32 bits)		**0.908**	**0.781**	**0.908**	**0.908**	**0.970**	**0.832**	**0.970**	**0.971**	**0.579**	**0.579**	**0.583**	**0.577**	**0.660**	**0.660**	**0.556**	**0.665**	**0.586**	**0.586**	**0.479**	**0.585**	**0.717**
SePH (64 bits) [23]*		0.892	0.678	0.882	0.826	0.911	0.718	0.912	0.879	0.499	0.524	0.520	0.499	0.599	0.639	0.429	0.543	0.473	0.510	0.402	0.521	0.643
SePH$_{lr}$ (64 bits) [24]*		0.896	0.334	0.874	0.872	0.955	0.379	0.915	0.904	0.314	0.367	0.373	0.329	0.497	0.540	0.224	0.455	0.449	0.477	0.185	0.505	0.542
RoPH (64 bits) [40]*		0.852	0.802	0.880	0.823	0.807	0.615	0.361	0.623	0.490	0.398	0.523	0.500	0.500	0.236	0.413	0.454	0.451	0.358	0.463	0.332	0.544
LSRH (64 bits) [30]*		0.905	0.387	0.771	0.771	0.947	0.462	0.864	0.861	0.326	0.395	0.261	0.282	0.506	0.552	0.243	0.469	0.435	0.466	0.219	0.418	0.527
SVHNs (64 bits)		**0.916**	**0.803**	**0.916**	**0.915**	**0.969**	**0.848**	**0.970**	**0.971**	**0.616**	**0.616**	**0.619**	**0.620**	**0.675**	**0.675**	**0.570**	**0.671**	**0.580**	**0.580**	**0.484**	**0.576**	**0.730**

*These methods are two-modality methods.

CPU with PyTorch[1]. For training, we employ the ADAM optimizer [33] with a batch size of 100 and set the maximal number of epochs as 200 for LabNet and 100 for MVHs. The learning rate α is empirically set as 0.0001 for LabNet and MVHs. In our experiments, the parameters λ and β were set to 0.1 and 10, respectively, which were obtained by cross-validation on the NUS-WIDE dataset using 16 bits. For all datasets, the image features are extracted by pre-trained VGGNet [41]. Specifically, the image extractor model has the same configuration with 19-layer VGGNet pre-trained on the ImageNet, and 4,096-dimensional features vector from fc7 layer is extracted as the image original representations. For the MS-COCO, the 300-dimensional text features are extracted by the Doc2Vec model[2] [42], which is pre-trained on Wikipedia. For the text features of other datasets, each text is represented by a bag-of-words (BoW) vector. For other modalities, the features of each modality are provided by the authors. Then LabNet and MVNs with three fully-connected layers are adopted for all modalities to learn common hash codes. The number of hidden units are 2,048, 512 and L for LabNet, and 4,096, 512 and $2L$ for each MVN. Furthermore, the binary representations of the retrieval database are obtained from LabNet, and ones of query samples are computed by MVNs.

5.1.3 Evaluation Protocol. For all the datasets, the data points of the test (query) set are randomly sampled from the total set and the remaining points as the retrieval set (database) as Table 1 following [15]. The ground-truth neighbors are defined as those cross-modal samples which share at least one same class. To evaluate the performance of our SVHNs and other compared methods, $m(m-1)$ kinds of cross-modal retrieval tasks are performed in the common space learned by these methods on the above datasets. The testing is conducted in a pairwise manner, *i.e.*, the samples from one modality are used as the target database while the ones from another modality are used as the queries, which is defined as follows.

- $\mathcal{X}^k$-**query**-$\mathcal{X}^l$ ($\mathcal{X}^k \rightarrow \mathcal{X}^l$, $k \neq l$): for a query from the k-th modality, relevant instances from the l-th modality are retrieved in the target database ranked by calculated cross-modal similarity in the common space.

Take two-modality case (image and text) for an example. There are two kinds of retrieval tasks, *i.e.*, image-query-text (Image → Text) and text-query-image (Text → Image) in this case.

For cross-modal hashing-based retrieval, we use the widely-used Hamming ranking and hash lookup as the retrieval protocols to evaluate our SVHNs and other compared methods following [15]. Mean Average Precision (mAP) score is adopted as the evaluation metric to measure the scores of the Hamming ranking protocol on the four datasets. mAP is the mean value of Average Precision (AP) scores for each query. mAP score considers the ranking of returned retrieval results as well as precision simultaneously, which is extensively adopted in cross-modal retrieval tasks. It should be noted that the mAP score is calculated on the all returned results following [15] in our experiments. The hash lookup protocol returns all the points within a certain Hamming radius away from the query point [15]. The widely-used precision-recall curve is used as a metric to evaluate the performance of the hash lookup protocol.

5.2 Experimental Analysis

5.2.1 Hamming Ranking. To evaluate the performance of our SVHNs for more than two modalities, the comparison with some real-valued multimodal and hashing two-modality methods is conducted on the PK XMedia dataset. The mAP scores of 20 cross-modal retrieval tasks on the dataset are shown in Table 2. Because of the seriously unbalanced modalities of PKU XMedia, these multimodal real-valued methods cannot achieve satisfactory performance. Thanks to the separated training of each modality, our SVHNs can extract more discriminative information from each modality without the pairwise limitation. In the comparison with two-modality hashing methods, our method not only outperforms these cross-modal hashing methods but also costs much less time to learn a single common

[1]The PyTorch homepage: https://pytorch.org/
[2]The pre-trained Doc2Vec model is available at https://github.com/jhlau/doc2vec.

Table 3: Performance comparison in terms of mAP scores on MIRFLICKR-25K, NUS-WIDE and MS-COCO. The best result is shown in boldface.

Task	Method	MIRFLICKR-25K			NUS-WIDE			MS-COCO		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
Image → Text	SePH [23]	0.730	0.740	0.746	0.646	0.655	0.665	0.578	0.606	0.615
	SePH$_{lr}$ [24]	0.730	0.748	0.756	0.597	0.621	0.641	0.551	0.563	0.597
	RoPH [40]	0.733	0.743	0.748	0.640	0.652	0.664	0.608	0.635	0.639
	LSRH [30]	0.730	0.774	0.792	0.601	0.632	0.663	0.550	0.558	0.542
	DCMH [15]	0.737	0.754	0.763	0.581	0.603	0.628	0.557	0.558	0.586
	SSAH [12]	0.782	0.790	0.800	0.642	0.636	0.639	0.591	0.606	0.576
	CMDVH [13]	0.753	0.765	0.791	0.743	0.766	0.757	-	-	-
	SVHNs	**0.908**	**0.921**	**0.930**	**0.828**	**0.846**	**0.857**	**0.699**	**0.767**	**0.787**
Text → Image	SePH [23]	0.751	0.757	0.765	0.656	0.661	0.667	0.573	0.608	0.611
	SePH$_{lr}$ [24]	0.763	0.781	0.789	0.621	0.676	0.667	0.574	0.594	0.632
	RoPH [40]	0.748	0.765	0.768	0.650	0.664	0.669	0.603	0.629	0.635
	LSRH [30]	0.734	0.780	0.795	0.583	0.652	0.679	0.596	0.614	0.598
	DCMH [15]	0.753	0.760	0.763	0.587	0.605	0.637	0.586	0.609	0.624
	SSAH [12]	0.791	0.795	0.803	0.669	0.662	0.666	0.641	0.659	0.660
	CMDVH [13]	0.755	0.751	0.783	0.667	0.729	0.757	-	-	-
	SVHNs	**0.823**	**0.843**	**0.848**	**0.761**	**0.780**	**0.790**	**0.720**	**0.783**	**0.796**

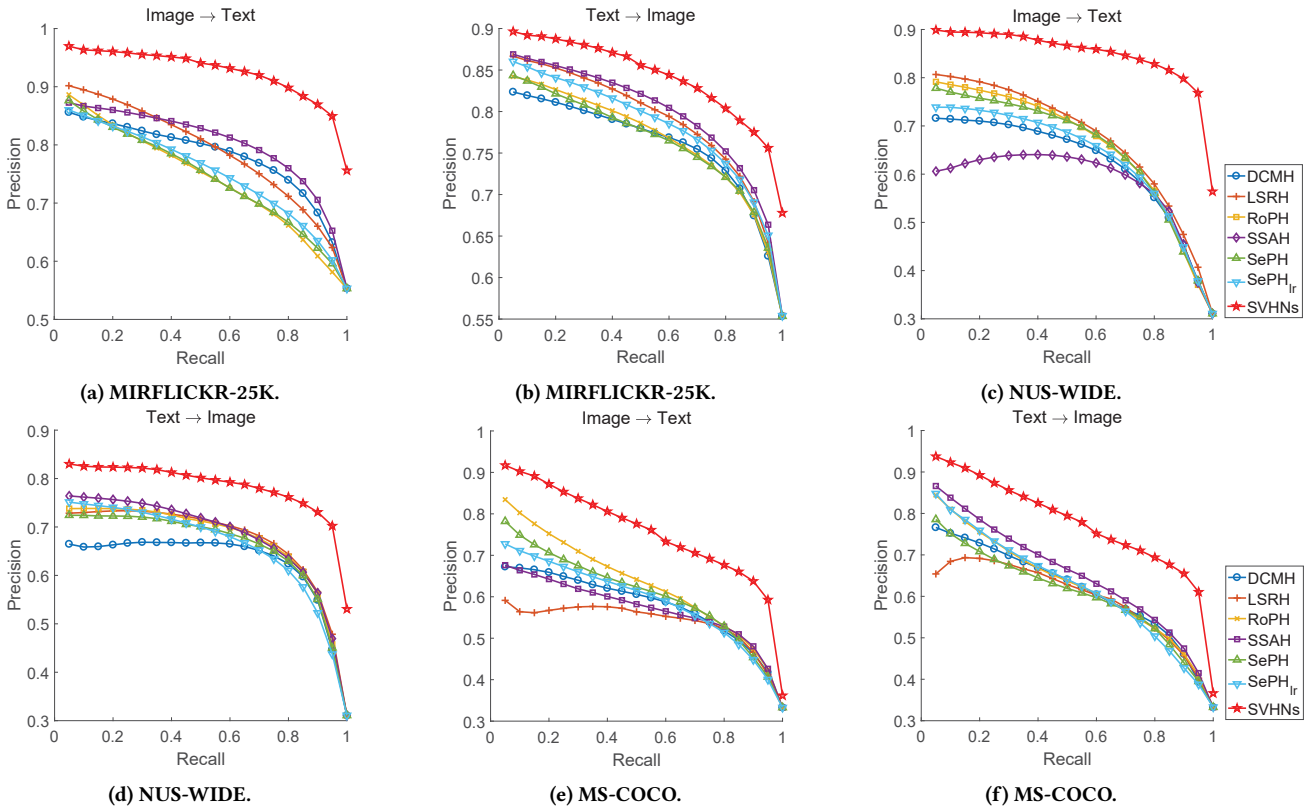
space for the five modalities. This is mainly due to the following two reasons: 1) Without the pairwise limitation, our SVHNs can fully and separately exploit the whole data of each modality to preserve the discrimination into the common Hamming space; 2) These cross-modal hashing methods should be conducted $\frac{m(m-1)}{2}$ times in a pairwise manner to learn $\frac{m(m-1)}{2}$ pairwise common Hamming spaces for m modalities, which will cost much more time than ours. In brief, our approach outperforms all compared methods even with 16 bits, which verifies the effectiveness of our SVHNs for more than two modalities.

For two-modality datasets, the mAP scores of two cross-modal retrieval tasks, *i.e.*, retrieving text by image query (Image → Text) and retrieving image by text query (Text → Image), on the MIRFLICKR-25K, NUS-WIDE and MS-COCO datasets, are shown in Table 3. Like Table 2, our approach also outperforms all cross-modal hashing methods in Table 3. For all experiments, we set the length of hash codes L as 16, 32 and 64 bits. From the experimental results of Tables 2 and 3, the following observations are given: 1) There is no obvious gap between the results of traditional and deep methods, and some shallow methods are even higher than the deep methods. This is because the used deep features contain much high-level semantic information, which may boost the performance of the traditional methods. 2) All the cross-modal hashing methods have the pairwise limitation, leading to inadequate usage of cross-modal data, which limits their performance. 3) All supervised methods are superior to the unsupervised ones. This is because the supervised approaches explicitly explore more discriminative information, *e.g.*, the class label, to boost the performance of cross-modal retrieval. 4) The existing cross-modal hashing methods are specially designed for two-modality cases. They cannot be used to project the multimodal data (more than two modalities) into a common Hamming space. However, our SVHNs can separately learn the common hashing representations from more than two modalities and achieve the best performance even compared with real-valued multimodal methods.

5.2.2 Hash Lookup. In the hash lookup protocol, the precision and recall are computed for the returned points given any Hamming radius following [15]. The precision-recall curves with code length 64 on the MIRFLICKR-25K, NUS-WIDE and MC-COCO datasets are drawn for additional comparison as shown in Fig. 2. The precision-recall evaluations are consistent with the mAP scores for cross-modal retrieval tasks, where our SVHNs can dramatically outperform all the compared methods. Our SVHNs can also achieve the best performance on other cases with different values of code length, *i.e.*, 32 bits and 64 bits, whose results are omitted due to space limitation. Overall speaking, our SVHNs has achieved the best performance compared with the existing cross-modal hashing methods.

5.2.3 Ablation Study. We also investigate the performance of the variants for our SVHNs to verify the effectiveness of our different modules. There are two variants of our SVHNs: 1) SVHNs-1 jointly learns the common Hamming space through m modality-specific networks with the loss of LabNet without LabNet, 2) SVHNs-2 is trained as SVHNs but all binary representations of retrieval and query sets are computed by MVNs, and 3) SVHNs-3 separately learns the common Hamming space with LabNet and m modality-specific networks without variational inference. Table 4 shows the mAP scores of these methods on the MS-COCO dataset. We can see that the LabNet is much important for our model to learn more effective binary representations. Furthermore, the variational inference also helps our model to obtain more effective hash codes.

5.2.4 Efficiency Study. To further investigate the efficiency of our SVHNs, we compare it with some baselines in terms of the cost of training time and GPU memory on the MS-COCO dataset. SVHNs$_j$, a variant of our SVHNs, whose MVNs are jointly trained on all modalities, is used to investigate the efficiency of the separated training. For a fair comparison, the maximal epochs of all methods are set to 10. Specifically, the maximum epochs of LabNet and



(a) MIRFLICKR-25K. (b) MIRFLICKR-25K. (c) NUS-WIDE.

(d) NUS-WIDE. (e) MS-COCO. (f) MS-COCO.

Figure 2: The precision-recall curves on the MIRFLICKR-25K, NUS-WIDE and MS-COCO datasets. The code length is 64.

Table 4: Ablation study of SVHNs in terms of mAP scores on the MS-COCO dataset. The best result is shown in boldface.

Method	Task	16 bits	32 bits	64 bits
SVHNs-1	Image → Text	0.624	0.657	0.656
SVHNs-2		0.644	0.678	0.697
SVHNs-3		0.697	0.762	0.774
SVHNs		**0.699**	**0.767**	**0.787**
SVHNs-1	Text → Image	0.621	0.649	0.649
SVHNs-2		0.639	0.672	0.670
SVHNs-3		0.719	0.777	0.795
SVHNs		**0.720**	**0.783**	**0.796**

Table 5: Comparison of the training time cost and GPU memory usage on the MS-COCO dataset. The code length is 64.

Method	Time Cost	Memory Usage
DCMH [15]	91.16s	1441MiB
SSAH [12]	3671.59s	1441MiB
SVHNs$_j$	25.24s	951MiB
SVHNs	**21.69s**	**911MiB**

each MVN are both set to 10 for SVHNs$_j$ and SVHNs. The performance verification is omitted from the training stage in all methods. The results are shown in Table 5, from which we can see that the separated training can improve training efficiency.

6 CONCLUSION

In this paper, we proposed a novel approach called Separated Variational Hashing Networks (SVHNs) to separately transform any number of modalities into a common Hamming space. SVHNs consists of a label network (LabNet) and multiple modality-specific networks. LabNet is used to exploit all available label annotations to learn a latent common Hamming space by projecting the semantic labels into the common binary codes. Then, the modality-specific variational networks can separately project multiple modalities into their common semantic binary representations learned by LabNet. It is achieved by conducting the variational inference that matching the aggregated posterior of the hashing code vector of LabNet with an arbitrary prior distribution. Extensive experimental results on four widely-used benchmark datasets and the comprehensive analysis have demonstrated the effectiveness of the designed LabNet and the variational inference, leading to superior cross-modal retrieval performance compared to current state-of-the-art methods.

ACKNOWLEDGMENT

This work is supported by the National Key Research and Development Project of China under contract No. 2017YFB1002201 and partially supported by the National Natural Science Foundation of China (Grants No. 61971296), and Sichuan Science and Technology Planning Projects (Grants No. 2018TJPT0031, 2019YFH0075, 2018GZDZX0030).

REFERENCES

[1] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *International Conference on Machine Learning*, pages 1247–1255, 2013.

[2] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. Deep supervised cross-modal retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 10394–10403, Long Beach, CA, USA, June 2019.

[3] Jan Rupnik and John Shawe-Taylor. Multi-view canonical correlation analysis. In *Conference on Data Mining and Data Warehouses*, pages 1–4, 2010.

[4] Weiran Wang, Raman Arora, Karen Livescu, and Jeff Bilmes. On deep multi-view representation learning. In *International Conference on Machine Learning*, pages 1083–1092, 2015.

[5] Peng Hu, Liangli Zhen, Dezhong Peng, and Pei Liu. Scalable deep multimodal learning for cross-modal retrieval. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 635–644, New York, NY, USA, 2019. ACM.

[6] Yuxin Peng, Xin Huang, and Jinwei Qi. Cross-media shared representation by hierarchical learning with multiple deep networks. In *International Joint Conference on Artificial Intelligence*, pages 3846–3853, 2016.

[7] Jinwei Qi and Yuxin Peng. Cross-modal bidirectional translation via reinforcement learning. In *International Joint Conference on Artificial Intelligence*, pages 2630–2636, 2018.

[8] Fabio Carrara, Andrea Esuli, Tiziano Fagni, Fabrizio Falchi, and Alejandro Moreo Fernández. Picture it in your mind: Generating high level visual representations from textual descriptions. *Information Retrieval Journal*, 21(2-3):208–229, 2018.

[9] Abhishek Sharma, Abhishek Kumar, Hal Daume, and David W Jacobs. Generalized multiview analysis: A discriminative latent space. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2160–2167. IEEE, 2012.

[10] Peng Hu, Dezhong Peng, Yongsheng Sang, and Yong Xiang. Multi-view linear discriminant analysis network. *IEEE Transactions on Image Processing*, 2019.

[11] Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, (12):1349–1380, 2000.

[12] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. Self-supervised adversarial hashing networks for cross-modal retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4242–4251, 2018.

[13] Venice Erin Liong, Jiwen Lu, Yappeng Tan, and Jie Zhou. Cross-modal deep variational hashing. pages 4097–4105, 2017.

[14] Jiwen Lu, Venice Erin Liong, Xiuzhuang Zhou, and Jie Zhou. Learning compact binary face descriptor for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(10):2041–2056, 2015.

[15] Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3232–3240, 2017.

[16] Peng Hu, Dezhong Peng, Jixiang Guo, and Liangli Zhen. Local feature based multi-view discriminant analysis. *Knowledge-Based Systems*, 149:34–46, 2018.

[17] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):188–194, 2016.

[18] Meina Kan, Shiguang Shan, Haihong Zhang, Shihong Lao, and Xilin Chen. Multi-view discriminant analysis. In *European Conference on Computer Vision*, pages 808–821, 2012.

[19] Hong Liu, Rongrong Ji, Yongjian Wu, Feiyue Huang, and Baochang Zhang. Cross-modality binary code learning via fusion similarity hashing. In *Computer Vision and Pattern Recognition*, pages 6345–6353, 2017.

[20] Devraj Mandal, Kunal N Chaudhury, and Soma Biswas. Generalized semantic preserving hashing for n-label cross-modal retrieval. In *Computer Vision and Pattern Recognition*, pages 2633–2641. IEEE, 2017.

[21] Jianfeng Dong, Xirong Li, Weiyu Lan, Yujia Huo, and Cees G. M. Snoek. Early embedding and late reranking for video captioning. In *ACM Multimedia*, 2016.

[22] Peng Hu, Dezhong Peng, Xu Wang, and Yong Xiang. Multimodal adversarial network for cross-modal retrieval. *Knowledge-Based Systems*, 180:38–50, 2019.

[23] Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. Semantics-preserving hashing for cross-view retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3864–3872, 2015.

[24] Zijia Lin, Guiguang Ding, Jungong Han, and Jianmin Wang. Cross-view retrieval via probability-based semantics-preserving hashing. *IEEE transactions on cybernetics*, 47(12):4342–4355, 2017.

[25] Jian Zhang, Yuxin Peng, and Mingkuan Yuan. Unsupervised generative adversarial cross-modal hashing. In *AAAI Conference on Artificial Intelligence*, 2018.

[26] Cheng Deng, Zhaojia Chen, Xianglong Liu, Xinbo Gao, and Dacheng Tao. Triplet-based deep hashing network for cross-modal retrieval. *IEEE Transactions on Image Processing*, 27(8):3893–3903, 2018.

[27] Lin Wu, Yang Wang, and Ling Shao. Cycle-consistent deep generative hashing for cross-modal retrieval. *IEEE Transactions on Image Processing*, 28(4):1602–1612, 2018.

[28] Guiguang Ding, Yuchen Guo, and Jile Zhou. Collective matrix factorization hashing for multimodal data. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2075–2082, 2014.

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

[30] Kai Li, Guo-Jun Qi, Jun Ye, and Kien A Hua. Linear subspace ranking hashing for cross-modal retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (9):1825–1838, 2017.

[31] Xi Peng, Shijie Xiao, Jiashi Feng, Wei-Yun Yau, and Zhang Yi. Deep subspace clustering with sparsity prior. In *International Joint Conference on Artificial Intelligence*, pages 1925–1931, New York, NY, USA, 9-15 July 2016.

[32] Joey Tianyi Zhou, Heng Zhao, Xi Peng, Meng Fang, Zheng Qin, and Rick Siow Mong Goh. Transfer hashing: From shallow to deep. *IEEE transactions on neural networks and learning systems*, 29(12):6191–6201, 2018.

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, pages 1–13, San Diego, USA, 2015.

[34] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. pages 1–14, 2014.

[35] Venice Erin Liong, Jiwen Lu, Ling-Yu Duan, and Yap-Peng Tan. Deep variational and structural hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[36] Xiaohua Zhai, Yuxin Peng, and Jianguo Xiao. Learning cross-media joint representation with sparse and semisupervised regularization. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(6):965–978, 2014.

[37] Mark J Huiskes and Michael S Lew. The MIR flickr retrieval evaluation. In *ACM International Conference on Multimedia Information Retrieval*, pages 39–43. ACM, 2008.

[38] Nikhil Rasiwasia, Jose Costa Pereira, Emanuele Coviello, Gabriel Doyle, Gert RG Lanckriet, Roger Levy, and Nuno Vasconcelos. A new approach to cross-modal multimedia retrieval. In *International Conference on Multimedia*, pages 251–260. ACM, 2010.

[39] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[40] Kun Ding, Bin Fan, Chunlei Huo, Shiming Xiang, and Chunhong Pan. Cross-modal hashing via rank-order preserving. *IEEE Transactions on Multimedia*, 19(3):571–585, 2017.

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[42] Jey Han Lau and Timothy Baldwin. An empirical evaluation of doc2vec with practical insights into document embedding generation. In *Workshop on Representation Learning for NLP*, pages 78–86. Association for Computational Linguistics, 2016.