Redundancy-resistant Generative Hashing for Image Retrieval

Changying Du[1], **Xingyu Xie**[2], **Changde Du**[3], **Hao Wang**[1*]
[1] 360 Search Lab, Beijing 100015, China
[2] College of Automation, Nanjing University of Aeronautics and Astronautics, Nanjing, China
[3] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
ducyatict@gmail.com, nuaaxing@gmail.com, duchangde2016@ia.ac.cn, cashenry@126.com

Abstract

By optimizing probability distributions over discrete latent codes, Stochastic Generative Hashing (SGH) bypasses the critical and intractable binary constraints on hash codes. While encouraging results were reported, SGH still suffers from the deficient usage of latent codes, i.e., there often exist many uninformative latent dimensions in the code space, a disadvantage inherited from its auto-encoding variational framework. Motivated by the fact that code redundancy usually is severer when more complex decoder network is used, in this paper, we propose a constrained deep generative architecture to simplify the decoder for data reconstruction. Specifically, our new framework forces the latent hashing codes to not only reconstruct data through the generative network but also retain minimal squared L2 difference to the last real-valued network hidden layer. Furthermore, during posterior inference, we propose to regularize the standard auto-encoding objective with an additional term that explicitly accounts for the negative redundancy degree of latent code dimensions. We interpret such modifications as Bayesian posterior regularization and design an adversarial strategy to optimize the generative, the variational, and the redundancy-resistanting parameters. Empirical results show that our new method can significantly boost the quality of learned codes and achieve state-of-the-art performance for image retrieval.

1 Introduction

Image retrieval aims to retrieve relevant images to a given query image from a very large image corpus. Due to the huge amount of computation and storage cost required for similarity evaluation in the original feature space, the core problem in this area has been how to re-represent each image using sufficiently compact codes while preserving the original similarity structure as more as possible.

Learning to hash (LTH) [Wang *et al.*, 2017] is a popular kind of techniques developed for the aforementioned approximate nearest neighbor search problem. By using binary hash codes to represent the original data, the storage cost can be dramatically reduced. Furthermore, search with such binary representations can be efficiently conducted because (1) we can perform Hamming distance computation that is supported via POPCNT on modern CPU/GPU; and (2) we can achieve a constant or sub-linear time complexity for search by using hash codes to construct an index [Kong and Li, 2012].

Since the expressiveness of the learned hash codes can directly affect the final retrieval performance, improving code quality by capturing preferred properties of the hash function is a long-standing topic in LTH studies. Supervised hashing [Xia *et al.*, 2014], semi-supervised hashing [Wang *et al.*, 2012; Zhu *et al.*, 2016], and cross-modal hashing [Jiang and Li, 2016] achieve this goal by assuming the images are labeled or tagged, which requires many human efforts or even be unrealistic for modern applications. In contrast, unsupervised hashing methods do not utilize any semantic relations explicitly, and thus are more challenging to be effective in practice. Representative work along this line include spectral hashing [Weiss *et al.*, 2008], isotropic hashing [Kong and Li, 2012], iterative quantization [Gong *et al.*, 2013], K-means hashing [He *et al.*, 2013], discrete graph hashing [Liu *et al.*, 2014], spherical hashing [Heo *et al.*, 2015], graph hashing [Jiang and Li, 2015], binary autoencoder (BAE) hashing [Carreira-Perpinán and Raziperchikolaei, 2015], stochastic generative hashing (SGH) [Dai *et al.*, 2017], etc. Among them, SGH and BAE enjoy the theoretical advantage of minimum description length (MDL) owing to their auto-encoding architecture. This property is critical to achieve not only fast but also accurate retrieval. Besides, due to its probabilistic nature, SGH avoids commonly used relaxation of the binary constraints (which often leads to inferior results) by optimizing probability distributions over discrete hash codes.

The apparent benefits claimed by SGH are inherited from the auto-encoding variational inference framework [Kingma and Welling, 2014; Rezende *et al.*, 2014]. We note, however, that the MDL principle may be unsatisfied due to the prior constraints in this framework. In fact, recent studies show that such a variational autoencoder (VAE) framework suffers from the deficient usage of latent codes, i.e. there often exist many uninformative latent dimensions in the code space [Sønderby *et al.*, 2016], especially when the decoder network is complex. This phenomenon is caused by the variational

[*] Corresponding author

pruning [Hoffman, 2017; Yeung *et al.*, 2017], a problem occurs when many latent units collapse early in training before they learned a useful representation.

In this paper, we design a new deep generative architecture which forces the latent hashing codes to not only reconstruct data through the generative (decoder) network but also retain minimal squared L2 difference to the output of the last real-valued decoder hidden layer. We show that such a constrained deep structure has the effect of simplifying the decoder while providing sufficiently good generation/reconstruction quality. A simplified decoder network together with a deep inference (encoder) network are expected to alleviate variational pruning and hence code redundancy. Furthermore, during posterior inference, we propose to regularize the standard auto-encoding objective with an additional term that explicitly accounts for the negative redundancy degree of latent code dimensions. We interpret such modifications as Bayesian posterior regularization and design an adversarial strategy to optimize the generative, the variational, and the redundancy-resistanting parameters. Experimental results show that our redundancy-resistant generative hashing framework can significantly boost the quality of learned codes and achieve state-of-the-art retrieval performance on image data sets.

2 Model

As a powerful unsupervised representation learning framework, VAE [Kingma and Welling, 2014; Rezende *et al.*, 2014] defines a latent variable generative model as follows

$$p(\mathbf{Z}) = \prod_{i=1}^{N} p(\mathbf{z}_i), \quad p_\theta(\mathbf{X}|\mathbf{Z}) = \prod_{i=1}^{N} p_\theta(\mathbf{x}_i|\mathbf{z}_i)$$

where θ denotes the likelihood is parameterized by a decoder Deep Neural Network (DNN), and $\mathbf{Z} \in \mathbb{R}^{K\times N}$ and $\mathbf{X} \in \mathbb{R}^{D\times N}$ denote the latents and the observed data, respectively. Specifying $q_\phi(\mathbf{Z}|\mathbf{X}) = \prod_{i=1}^{N} q_\phi(\mathbf{z}_i|\mathbf{x}_i)$ as the DNN-parameterized inference model (encoder), VAE seeks to maximize the evidence lower bound (ELBO) w.r.t. ϕ and θ

$$\sum_{i=1}^{N} \log p_\theta(\mathbf{x}_i) \geq \sum_{i=1}^{N} \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)}[\log p_\theta(\mathbf{x}_i|\mathbf{z}_i)] - \sum_{i=1}^{N} D_{KL}(q_\phi(\mathbf{z}_i|\mathbf{x}_i)||p(\mathbf{z}_i)) \equiv ELBO.$$

2.1 Stochastic Generative Hashing

As an instance of VAE, SGH [Dai *et al.*, 2017] uses a linear Gaussian likelihood and imposes the Bernoulli prior on $\mathbf{Z}$:

$$p(\mathbf{Z}) = \prod_{i=1}^{N}\prod_{j=1}^{K} \rho_j^{z_{ij}}(1-\rho_j)^{1-z_{ij}}, \tag{1}$$

where $\rho = [\rho_j]_{j=1}^K \in [0,1]^K$. Correspondingly, the inference model is parameterized as

$$q_\phi(\mathbf{Z}|\mathbf{X}) = \prod_{i=1}^{N}\prod_{j=1}^{K} (\phi_j(\mathbf{x}_i))^{z_{ij}}(1-\phi_j(\mathbf{x}_i))^{1-z_{ij}}, \tag{2}$$

where $\phi_j(\mathbf{x}_i) = q_\phi(z_{ij} = 1|\mathbf{x}_i)$, and ϕ_j represents a scalar-valued linear or deep nonlinear transformation.

2.2 Redundancy-resistant Generative Hashing

To ensure efficient similarity search, we assume the latent hash code $\mathbf{z}$ of each data to be binary, thus our Redundancy-resistant SGH (R-SGH) model shares the same Bernoulli prior $p(\mathbf{Z})$ as in (1). Compared to SGH, one distinct feature of R-SGH lies in the new deep generative architecture for likelihood computation. Suppose the decoder DNN (generative network) has $M+1$ layers, where 1) the first layer is the hash code layer, which has prior $p(\mathbf{Z})$ and is the latent variable of our Bayesian model; 2) the last layer is the data reconstruction layer, whose output is used to form the mean and covariance parameters of our Gaussian likelihood on data; and 3) the M-th layer is the last real-valued decoder hidden layer, whose output is denoted by $\mathbf{H}^{(M)} = [\mathbf{h}_i^{(M)}]_{i=1}^N$). Our idea is to force $\mathbf{Z}$ to not only reconstruct data through the decoder network but also retain minimal squared L2 difference to the output of the M-th decoding layer. Thus we propose to add the regularization term $-\|\mathbf{Z} - \mathbf{H}^{(M)}\|_F^2$ into the VAE objective, where $\|\cdot\|_F$ denotes the Frobenius norm of a matrix, and we assume $\mathbf{z}$ and $\mathbf{h}^{(M)}$ have the same dimension.

Consider the extreme case that the above regularization term is perfectly optimized ($\mathbf{Z}$ exactly equals to $\mathbf{H}^{(M)}$), i.e., the hash code layer is identical to the last decoder hidden layer, then our decoder model is equivalent to a single layer network. Thus, such a constrained deep structure has the effect of simplifying the decoder while providing sufficiently good generation/reconstruction quality. As studied in RNN-based variational language models [Yang *et al.*, 2017], where the overly flexible LSTM decoder in VAE does not make effective use of the latent representation, a simplified decoder network together with a deep unconstrained inference (encoder) network are expected to alleviate variational pruning and hence code redundancy. For our R-SGH, we assume that the inference model has the same form as in (2), and ϕ is implemented by an encoder DNN (which transforms the input data into Bernoulli parameter for hash code through M hidden layers) with mirrored layer structure as the decoder.

Furthermore, during posterior inference, we propose to regularize the standard auto-encoding variational objective with an additional term that explicitly accounts for the negative redundancy degree of latent code dimensions. The key insight is that, if a latent dimension can be approximated by the linear combination of the other latent dimensions, then this dimension is useless for data generation. To eliminate such redundancy and encourage all latent dimensions to explain the data, we formulate our overall model as follows:

$$\max_{\theta,\phi}\min_{\mathbf{A}} \; \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}[\delta\|\mathbf{Z}^\top - \mathbf{Z}^\top\mathbf{A}\|_F^2 - \eta\|\mathbf{Z} - \mathbf{H}^{(M)}\|_F^2] + ELBO, \quad s.t. \; a_{kk} = 0, \; k = 1, ..., K, \tag{3}$$

where $\mathbf{A} \in \mathbb{R}^{K\times K}$ denotes a combination coefficient matrix. Intuitively, we always find the best possible combination coefficients to account for the redundancy degree of latent dimensions, and $\delta > 0$ and $\eta > 0$ are regularization parameters that balance ELBO and redundancy-related terms.

Before we design specific adversarial algorithm to optimize θ, ϕ and $\mathbf{A}$ alternatingly, we first interpret the formulation in (3) as Bayesian posterior regularization [Zhu *et al.*, 2014]. We first define an auxiliary function

$l(\mathbf{Z}|\eta, \delta, \mathbf{A}) = \exp\left\{\eta\|\mathbf{Z} - \mathbf{H}^{(M)}\|_F^2 - \delta\|\mathbf{Z}^\top - \mathbf{Z}^\top\mathbf{A}\|_F^2\right\}$,

then given $\mathbf{A}$ the formulation in (3) can be rewritten as

$$\max_{\theta,\phi}\ ELBO - \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[\log l(\mathbf{Z}|\eta, \delta, \mathbf{A})\right]. \tag{4}$$

Solving problem (4), we can get the posterior distribution

$$q_\phi(\mathbf{Z}|\mathbf{X}) = \frac{p_\theta(\mathbf{X}|\mathbf{Z})p(\mathbf{Z})l(\mathbf{Z}|\eta, \delta, \mathbf{A})}{p(\mathbf{X})}.$$

As a direct way to impose constraints or incorporate knowledge in Bayesian models, Bayesian posterior regularization is more natural and general than specially designed priors.

2.3 Adversarial Training

The VAE parameters θ, ϕ aim to maximize (3), while the redundancy-resistanting parameter $\mathbf{A}$ plays an adversarial role to minimize it. To optimize them in turn, we have to formulate each sub-problem separately.

Optimizing θ Given ϕ and $\mathbf{A}$, problem (3) simplifies to

$$\max_\theta\ \sum_{i=1}^N \mathbb{E}_{q_\phi(\mathbf{z}_i|\mathbf{x}_i)} \log p_\theta(\mathbf{x}_i|\mathbf{z}_i), \tag{5}$$

which is straightforward to solve utilizing the reparameterization trick introduced in [Dai *et al.*, 2017]. Specifically, to deal with the discrete variational distribution, we use the Doubly Stochastic Neuron (DSN) that achieves the hash code variable by $z_{ij} = f(\phi_j(\mathbf{x}_i), \xi_j, \varepsilon_j)$, where $\xi_j, \varepsilon_j \sim \mathcal{U}(0,1)$ and

$$f(e, \xi_j, \varepsilon_j) = \begin{cases} 1, & \text{if } \ e > \xi_j, \\ \mathbf{1}_{e>\varepsilon_j}, & \text{if } \ e = \xi_j, \\ 0, & \text{if } \ e < \xi_j. \end{cases}$$

Hence, the objective in (5) can be rewritten as

$$\max_\theta\ \sum_{i=1}^N \mathbb{E}_{\xi,\varepsilon} \log p_\theta(\mathbf{x}_i|f(\phi(\mathbf{x}_i), \xi, \varepsilon)), \tag{6}$$

where $\mathbf{z}_i = f(\phi(\mathbf{x}_i), \xi, \varepsilon) = [f(\phi_j(\mathbf{x}_i), \xi_j, \varepsilon_j)]_{j=1}^K$. Then it is similar as in the vanilla VAE to derive a Monte Carlo estimator for the gradient of the objective in (6) w.r.t. θ.

Optimizing ϕ Given θ and $\mathbf{A}$, problem (3) simplifies to

$$\max_\phi\ \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[\delta\|\mathbf{Z}^\top - \mathbf{Z}^\top\mathbf{A}\|_F^2 - \eta\|\mathbf{Z} - \mathbf{H}^{M+1}\|_F^2\right] + ELBO, \tag{7}$$

which can also be solved using the reparameterization trick. However, we empirically note that such direct treatment often leads to poor performance since a optimum found this way is easily dominated by the first term. The underlying reason is that the ELBO is upper-bounded while the first term can be arbitrarily large. To this end, we introduce the ϵ-insensitive idea, and reformulate the problem as

$$\max_\phi\ ELBO + \delta \cdot \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[\sum_{k=1}^K \min\left(\frac{\|\mathbf{r}_k\|^2}{N} - \epsilon, 0\right)\right] - \eta \cdot \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[\|\mathbf{Z} - \mathbf{H}^{M+1}\|_F^2\right], \tag{8}$$

where $\mathbf{r}_k$ is the k-th column of $\mathbf{R} = \mathbf{Z}^\top - \mathbf{Z}^\top\mathbf{A}$. Following [Dai *et al.*, 2017], we employ the distributional derivative of DSN to derive the gradient of the objective in (8) w.r.t. ϕ.

Optimizing A Given ϕ and θ, problem (3) simplifies to

$$\min_{\mathbf{A}}\ \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[\|\mathbf{Z}^\top - \mathbf{Z}^\top\mathbf{A}\|_F^2\right], \quad s.t.\ \ a_{kk} = 0,\ k = 1, ..., K. \tag{9}$$

Applying the reparameterization trick again, we can solve this problem via gradient decent. Alternatively, here we show we can get an analytical solution by adding an extra term $\lambda\|\mathbf{A}\|_F^2$ to the objective, then (9) is converted into

$$\min_{\mathbf{A}}\ \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[\|\mathbf{Z}^\top - \mathbf{Z}^\top\mathbf{A}\|_F^2\right] + \lambda\|\mathbf{A}\|_F^2, \quad s.t.\ \ a_{kk} = 0,\ k = 1, ..., K. \tag{10}$$

where $\lambda > 0$. It can be shown that (10) can be solved in close-form [Lu *et al.*, 2012] with

$$\mathbf{A}^* = -\mathbf{D}(\mathrm{diag}(\mathbf{D}))^{-1},\ \ \mathrm{diag}(\mathbf{A}^*) = 0, \tag{11}$$

where $\mathbf{D} = \mathbb{E}_{q_\phi(\mathbf{Z}|\mathbf{X})}\left[(\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I})^{-1}\right]$ and $\mathrm{diag}(\mathbf{A})$ denotes a vector with its i-th element being the i-th diagonal element of $\mathbf{A}$. By setting $\lambda > 0$, $\mathbf{Z}\mathbf{Z}^\top + \lambda\mathbf{I}$ is ensured to be invertible. When $\lambda \to 0$, (10) degenerates to (9).

Remarks To reasonably optimize (7) and (9), we alter the terms related to the redundancy-resistanting parameter $\mathbf{A}$ and optimize (8) and (10) instead. This leads to different overall objectives for maximization and the adversarial minimization. Inconsistent objectives during block coordinate descent/ascent may lead to instability and misconvergence of optimization. However, in our adversarial framework, we empirically didn't find this is a problem. As will be seen in Figure 1, our algorithm converges steadily. We conjecture that it is due to the equilibrium condition difference between adversarial training and maximum likelihood training.

2.4 Complexity Analysis

Using the gradient estimators of $\{\theta, \phi\}$ and the analytical solution of $\mathbf{A}$, the computational complexity of a single joint update of the parameters $\{\theta, \phi, \mathbf{A}\}$ for the proposed model is $O(\tilde{N}C_{nn} + K^3)$, where $\tilde{N}$ is the batch size during our mini-batch training, and C_{nn} is the cost of evaluating the encoder/decoder networks. After training, all parameters are frozen, and we are interested in encoding a query image to its hash code, a procedure of the same complexity $O(C_{nn})$ as other deep hashing methods.

3 Experiments

We evaluate the proposed method on two computer vision tasks: 1) Image generation/reconstruction on MNIST [Oliva and Torralba, 2001]; 2) Image retrieval on CIFAR-10 [Krizhevsky, 2009] and Caltech-256 [Griffin *et al.*, 2007].

3.1 Experimental Setup

Compared Methods (1) Spectral Hashing (SH) [Weiss *et al.*, 2008]; (2) Iterative Quantization (ITQ) [Gong *et al.*, 2013]; (3) PCA Hashing (PCAH), which projects the input via PCA and performs binarization according to the sign of each dimension; (4) Stochastic Generative Hashing (SGH) [Dai *et al.*, 2017]; and (5) Deep-SGH, a variant of SGH that adopts DNN as the encoder and decoder.

Parameter Settings For the compared methods, we use the implementations provided by their authors (Deep-SGH is implemented directly based on SGH) and set the parameters according to their original papers. Without explicit statement, 1) for our R-SGH, the prior parameter ρ_j is set to 0.5 for any $j \in \{1, .., K\}$, the threshold parameter ϵ is set to 0.05, and both δ and η are set to 0.01; and 2) for R-SGH and Deep-SGH, the encoder and decoder network structures are set as $[D\text{-}K\text{-}K\text{-}K]$ and $[K\text{-}K\text{-}K\text{-}D]$ respectively, where D and K are the dimensions of input data and hash code respectively. Note, the 4-layer encoder and decoder share the same hash code layer, and all hidden layers have the same dimension.

Evaluation Metrics (1) L2 reconstruction error; (2) **precision** at S: the percentage of true relevant instances among top S retrieved ones; and (3) **recall** at S: the percentage of true relevant instances that are retrieved as the top S similar ones; (4) mean average precision (**mAP**), which computes the area under the entire precision-recall curve and evaluates the overall retrieval performance of different hashing algorithms. Note that, in evaluating these metrics for a given query image, we use a fixed small set of Euclidean nearest neighbors to the query as its ground-truth relevant images.

3.2 Performance Evaluation

Generation on MNIST To study the discriminative power of learned representations, we re-generate the images with 64 bit hash code on MNIST dataset. Specifically, we first randomly sample an $\mathbf{x}$ from the dataset and encode it into binary representation via the learned $q_\phi(\mathbf{z}|\mathbf{x})$, and then, we re-generate an artificial sample from $p_\theta(\mathbf{x}|\mathbf{z})$ by the mean.

We compare the proposed method with SGH. For fairness, we only consider the redundancy-resistant constraint on shallow network, where both the encoder and decoder are single layer neural network. The quantitative comparison is shown in Figure 1 where R-SGH achieves lower reconstruction error than that of SGH eventually, indicating that the hash codes of R-SGH contain richer information for reconstruction owing to their low redundancy merit. The visualization results are shown in Figure 2, which verify the advantage of our model again.

Image Retrieval on CIFAR-10 The CIFAR-10 dataset contains 60,000 color images from 10 object classes. Each original image is of size 32 × 32 and is represented as a 512-D GIST feature vector [Oliva and Torralba, 2001]. Following the same setting as in [Erin Liong *et al.*, 2015], we randomly sample 1000 samples, 100 per class, as the query data, and use the remaining 590,00 images as the gallery set.

The parameter δ in R-SGH is set to 0.1 on this dataset. For all considered methods, we repeat the experiments 20 trials and take the averages as the final results. Table 1 lists the mAP results of different hashing methods. Figure 3 shows the recall and precision curves as functions of the retrieved number of instances under different number of hash bits.

First, the clear advantages of our R-SGH over the other compared unsupervised hashing methods can be easily observed in terms of mAP performance, which demonstrate that our redundancy-resistant generative hashing framework can consistently improve the overall retrieval performance. Second, R-SGH keeps a greater growth of mAP than that of SGH when increasing number of hash bits are used, this is because the redundancy-resistant terms in R-SGH can effectively activate the uninformative bit in the generative model. Third, R-SGH outperforms the other compared methods by a large margin when 128 bits of hash code are used. We attribute this to the fact that higher dimensional latent representations come with higher information redundancy. Finally, we note that SGH and ITQ achieve better recall and precision performances when 32 bits hash code are used. This is due to very little information redundancy contained in the hash code when the number of hash bits is small. In this case, the redundancy-resistant regularization influences the reconstruction procedure and may degrade the performance. Fortunately, if low redundancy is expected in advance, we can prevent the negative impact by setting a small threshold parameter ϵ.

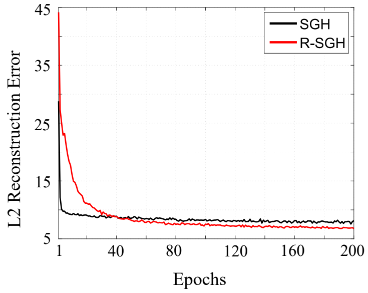


Figure 1: Reconstruction error vs. the number of iterating epochs.

Method	32 bits	64 bits	128 bits
ITQ	22.34	27.45	29.28
SH	6.97	12.23	19.39
PCAH	7.79	11.33	15.72
SGH	23.86	30.56	35.61
Deep-SGH	21.96	29.98	34.77
R-SGH	**24.66**	**33.62**	**44.12**

Table 1: mAP (%) results on the CIFAR-10 dataset.

Image Retrieval on Caltech-256 The Caltech-256 dataset consists of 29,780 images associated with 256 object categories. We randomly choose 1000 images as the query set and the rest of the dataset is regarded as the training set. We use the VGG network (VGG-fc7 [Szegedy *et al.*, 2015]) pre-trained on ILSVRC [Russakovsky *et al.*, 2015] as the feature extractor, which transforms each image into a 1024-dimensional vector.

We followed the same settings as in the CIFAR-10 experiment and used the same DNN structures. Table 2 displays the mAP performance of different hashing methods on Caltech-256 dataset. Figure 4 shows the precision-recall curves for different hashing methods under 32, 64 and 128 bits, respectively. We can see that R-SGH significantly outperforms the other compared methods on this dataset, even under 32 bits hash code. It is probably due to the well known fact that the CNN features their self have some information redundan-

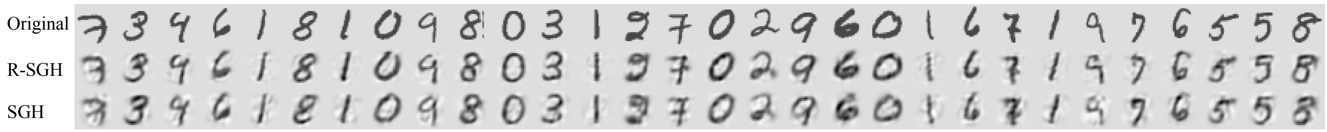
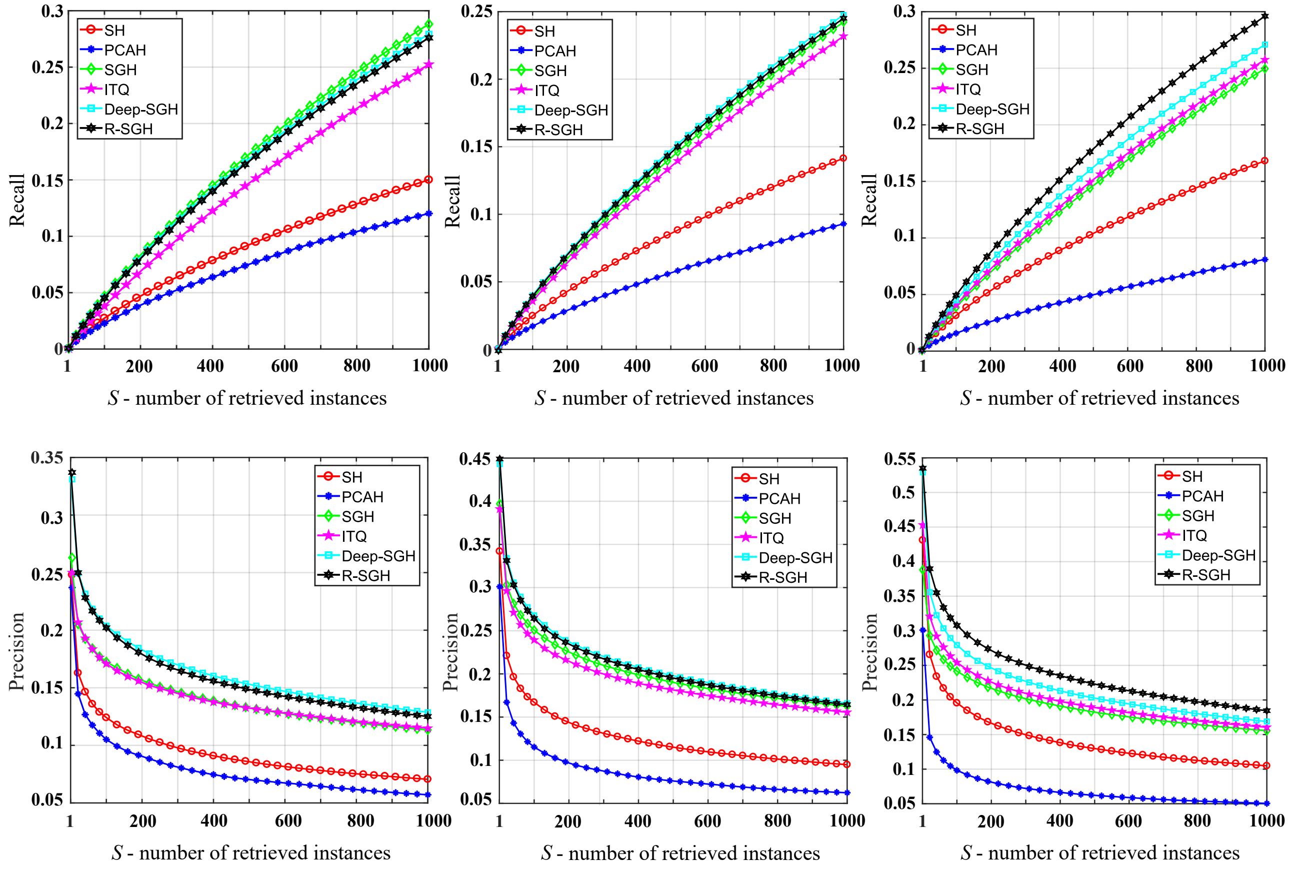


Figure 2: Illustration of MNIST images re-generated by R-SGH and SGH with 64 bit hash code. The hash codes of R-SGH contain richer information for reconstruction owing to their low redundancy merit.



(a) 32 bit

(b) 64 bit

(c) 128 bit

Figure 3: Recall and precision curves on CIFAR-10 dataset. The advantage of R-SGH is more evident when more hash bits are used, which verifies our conjecture that higher dimensional latent representations come with higher information redundancy.

cy. The improvements of R-SGH also show that the negative influence of information redundancy is even greater than that caused by dimensionality reduction for image retrieval, hence R-SGH is more suitable for the CNN features or other information redundant features than the compared methods.

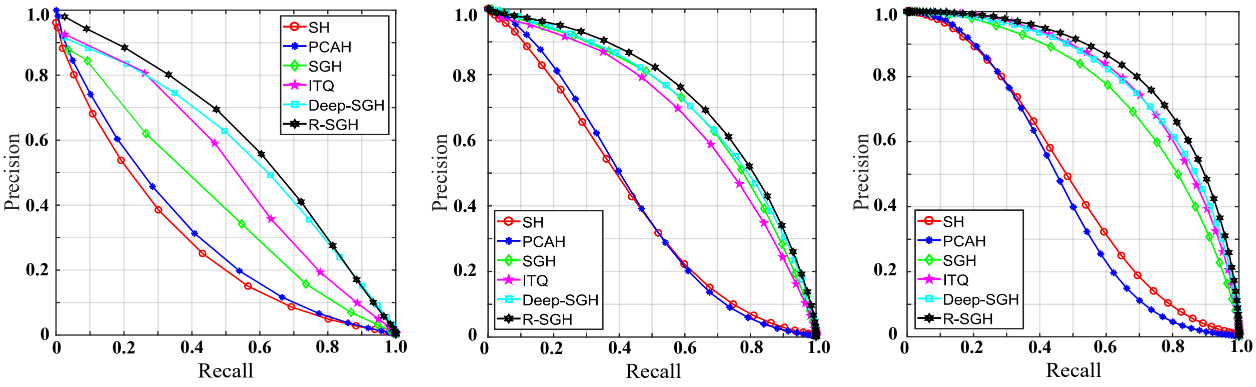
Method	32 bits	64 bits	128 bits
ITQ	50.12	68.56	76.88
SH	41.64	52.32	62.93
PCAH	43.62	55.65	66.02
SGH	47.12	71.09	78.61
Deep-SGH	51.39	70.34	79.39
R-SGH	**59.02**	**74.18**	**84.96**

Table 2: mAP (%) results on the Caltech-256 dataset.

3.3 Sensitivity Analysis

For sensitivity analysis of the proposed model, we use the Caltech-256 data set to evaluate the influences of regularization parameters δ and η. Here, the number of hash bits for each image is fixed to 128. When η is fixed to 0.01, Figure 5 (a) shows the mAP performance of R-SGH with varying δ. When δ is fixed to 0.01, Figure 5 (b) shows the mAP performance of R-SGH with varying η.

It is worth to mention that R-SGH seems to be more sensitive to δ. Unsuitable δ can degrade the mAP performance severely, while η has little impact on the mAP in a relatively large range, i.e., $[0.002, 0.07]$, as shown in Figure 5 (b). Both terms we proposed improves the discriminative power of the learned hash codes.

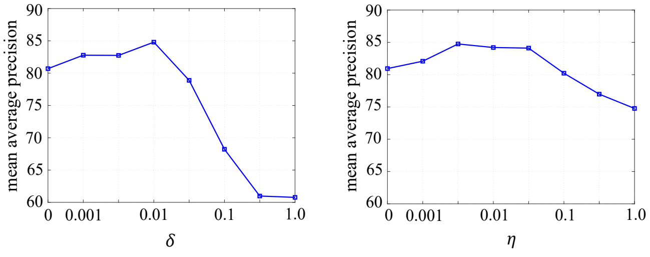


(a) 32 bit

(b) 64 bit

(c) 128 bit

Figure 4: Precision-recall curves on Caltech-256 dataset under 32, 64 and 128 bits hash code, respectively. The proposed R-SGH significantly outperforms the other compared methods even under 32 bits hash code, which may be due to the well known fact that CNN features their self have information redundancy.



(a)

(b)

Figure 5: (a) mAP vs. δ curve with fixed $\eta = 0.01$. (b) mAP vs. η curve with fixed $\delta = 0.01$.

4 Related Work

Learning-based hashing is an effective solution to accelerate similarity search by designing compact binary codes in a low-dimensional space. To enhance the expressiveness of fixed-length hash codes, there already exist some hash learning methods that pursue the independence (or low redundancy) between the hash bits. For example, the iterative quantization method [Gong *et al.*, 2013] maximizes the variance of each binary bit to keep the non-relevance and minimizes the binarization loss to obtain a high performance for image retrieval. Later, Liong et.al [Erin Liong *et al.*, 2015] proposed to use a DNN to learn hash codes with three objectives: (1) the loss between the real-valued feature descriptor and the learned binary codes is minimized; (2) binary codes distribute evenly on each bit; and (3) different bits are as independent as possible. They pursue the independence between hash bits by constraining the weight matrix in the encoders to be high rank. A more recent work named DeepBit [Lin *et al.*, 2016] learns compact binary code based on similar rules, i.e., minimal loss quantization, uncorrelated and evenly distributed bits. Bypassing quantization, our redundancy-resistant generative hashing method clearly differs from these methods.

Recent studies of VAEs for representation learning have revealed the decoupling phenomenon of latent code dimensionality from expressive power. Caused by the variational pruning [Hoffman, 2017; Yeung *et al.*, 2017], this phenomenon can be seen as a virtue of automatic relevance determination if efficiency is not a problem (but it is not the case for hashing-based similarity search), but also as a problem when many units collapse early in training before they learned a useful representation. [Sønderby *et al.*, 2016] observed that such units remain uninformative for the rest of the training, presumably trapped in a local minima or saddle point, with the optimization algorithm unable to re-activate them. In [Bowman *et al.*, 2015; Sønderby *et al.*, 2016], the authors used warm-up strategy to alleviate it by initializing training using the reconstruction error only. As more principled approaches, [Hoffman, 2017] proposed to learn a shared shearing matrix to rotate the variational distribution, and [Tomczak and Welling, 2017] proposed a "Variational Mixture of Posteriors" prior for the latent codes. The most similar strategy to ours is to restrict the power of the decoder by word dropout [Bowman *et al.*, 2015], lossy coding [Chen *et al.*, 2016] and dilated CNN [Yang *et al.*, 2017]. Nevertheless, our adversarial learning framework is more general than these works.

5 Conclusion

We designed a new deep generative architecture which forces the latent hashing codes to not only reconstruct data through the generative network but also retain minimal difference to the last real-valued hidden layer of the network. Furthermore, during posterior inference, we proposed to regularize the standard auto-encoding variational objective with an additional term that explicitly accounts for the negative redundancy degree of latent code dimensions. We designed an adversarial strategy to optimize the generative, the variational, and the redundancy-resistanting parameters. Experimental results show that our redundancy-resistant generative hashing framework can significantly boost the quality of learned codes and achieve state-of-the-art retrieval performance on image data.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61602449, 61672501, 61603372).

References

[Bowman *et al.*, 2015] Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*, 2015.

[Carreira-Perpinán and Raziperchikolaei, 2015] Miguel A Carreira-Perpinán and Ramin Raziperchikolaei. Hashing with binary autoencoders. In *CVPR*, 2015.

[Chen *et al.*, 2016] Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*, 2016.

[Dai *et al.*, 2017] Bo Dai, Ruiqi Guo, Sanjiv Kumar, Niao He, and Le Song. Stochastic generative hashing. In *ICML*, 2017.

[Erin Liong *et al.*, 2015] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep hashing for compact binary codes learning. In *CVPR*, 2015.

[Gong *et al.*, 2013] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on PAMI*, 35(12):2916–2929, 2013.

[Griffin *et al.*, 2007] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[He *et al.*, 2013] Kaiming He, Fang Wen, and Jian Sun. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *CVPR*, 2013.

[Heo *et al.*, 2015] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon. Spherical hashing: Binary code embedding with hyperspheres. *IEEE Transactions on PAMI*, 37(11):2304–2316, 2015.

[Hoffman, 2017] Matthew D Hoffman. Learning deep latent gaussian models with markov chain monte carlo. In *ICML*, 2017.

[Jiang and Li, 2015] Qing-Yuan Jiang and Wu-Jun Li. Scalable graph hashing with feature transformation. In *IJCAI*, 2015.

[Jiang and Li, 2016] Qing-Yuan Jiang and Wu-Jun Li. Deep cross-modal hashing. *arXiv preprint arXiv:1602.02255*, 2016.

[Kingma and Welling, 2014] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.

[Kong and Li, 2012] Weihao Kong and Wu-Jun Li. Isotropic hashing. In *NIPS*, 2012.

[Krizhevsky, 2009] Alex Krizhevsky. Learning multiple layers of features from tiny images. *Technical report, Univ. of Toronto*, 2009.

[Lin *et al.*, 2016] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *CVPR*, 2016.

[Liu *et al.*, 2014] Wei Liu, Cun Mu, Sanjiv Kumar, and Shih-Fu Chang. Discrete graph hashing. In *NIPS*, 2014.

[Lu *et al.*, 2012] Can-Yi Lu, Hai Min, Zhong-Qiu Zhao, Lin Zhu, De-Shuang Huang, and Shuicheng Yan. Robust and efficient subspace segmentation via least squares regression. *ECCV*, pages 347–360, 2012.

[Oliva and Torralba, 2001] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

[Rezende *et al.*, 2014] Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *ICML*, 2014.

[Russakovsky *et al.*, 2015] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[Sønderby *et al.*, 2016] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *NIPS*, 2016.

[Szegedy *et al.*, 2015] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[Tomczak and Welling, 2017] Jakub M Tomczak and Max Welling. Vae with a vampprior. *arXiv preprint arXiv:1705.07120*, 2017.

[Wang *et al.*, 2012] Jun Wang, Sanjiv Kumar, and Shih Fu Chang. Semi-supervised hashing for large-scale search. *IEEE Transactions on PAMI*, 34(12):2393–2406, 2012.

[Wang *et al.*, 2017] Jingdong Wang, Ting Zhang, Nicu Sebe, Heng Tao Shen, et al. A survey on learning to hash. *IEEE Transactions on PAMI*, 2017.

[Weiss *et al.*, 2008] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *NIPS*, 2008.

[Xia *et al.*, 2014] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, 2014.

[Yang *et al.*, 2017] Zichao Yang, Zhiting Hu, Ruslan Salakhutdinov, and Taylor Berg-Kirkpatrick. Improved variational autoencoders for text modeling using dilated convolutions. In *ICML*, 2017.

[Yeung *et al.*, 2017] Serena Yeung, Anitha Kannan, Yann Dauphin, and Li Fei-Fei. Tackling over-pruning in variational autoencoders. *arXiv preprint arXiv:1706.03643*, 2017.

[Zhu *et al.*, 2014] Jun Zhu, Ning Chen, and Eric P Xing. Bayesian inference with posterior regularization and applications to infinite latent SVMs. *JMLR*, 15(1):1799–1847, 2014.

[Zhu *et al.*, 2016] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, 2016.