Learning Deep Binary Descriptors via Bitwise Interaction Mining

Ziwei Wang, *Student Member, IEEE,* Han Xiao, Yueqi Duan, Jie Zhou, *Senior Member, IEEE* and Jiwen Lu, *Senior Member, IEEE,*

Abstract—In this paper, we propose a GraphBit method to learn unsupervised deep binary descriptors for efficient image representation. Conventional binary representation learning methods directly quantize each element according to the threshold without considering the quantization ambiguousness. The elements near the boundary dubbed as "ambiguous bits" fail to collect effective information for reliable binarization and are sensitive to noise that causes reversed bits. We argue that there are implicit inner relationships among bits in binary descriptors called bitwise interaction, where the related bits can provide extra instruction as prior knowledge for ambiguousness reduction. Specifically, we design a deep reinforcement learning model to learn the structure of the graph for bitwise interaction mining, and the uncertainty of binary codes is reduced by maximizing the mutual information with input and related bits. Consequently, the ambiguous bits receive additional instruction from the graph for reliable binarization. Moreover, we further present a differentiable search method (GraphBit+) that mines the bitwise interaction in continuous space, so that the heavy search cost caused by the training difficulties in reinforcement learning is significantly reduced. Since the GraphBit and GraphBit+ methods learn fixed bitwise interaction which is suboptimal for various input, the inaccurate instruction from the fixed bitwise interaction cannot effectively decrease the ambiguousness of binary descriptors. To address this, we further propose the unsupervised binary descriptor learning method via dynamic bitwise interaction mining (D-GraphBit), where a graph convolutional network called GraphMiner reasons the optimal bitwise interaction for each input sample. Extensive experimental results on the CIFAR-10, NUS-WIDE, ImageNet-100, Brown and HPatches datasets demonstrate the efficiency and effectiveness of the proposed GraphBit, GraphBit+ and D-GraphBit.

Index Terms—Binary descriptors, unsupervised learning, bitwise interaction, reinforcement learning, differentiable search, graph convolutional networks

1 Introduction

Extracting effective descriptors is one of the most active issues in computer vision, which is widely applicable in numerous tasks, such as face recognition [38], [42], image classification [25], [34], object recognition [37] and many others. Strong discriminative power and low computational cost are two essential properties for an effective descriptor. On one hand, strong discriminative power enables descriptors to be distinctive in image description and robust to various transformations. On the other hand, highly efficient descriptors present low memory cost and high computational speed, which are suitable for the scenarios of mobile devices with limited computational capabilities and real-time requirements. In recent years, a number of deep binary descriptors have been proposed due to their strong discriminative power and low computational cost [34], [18]. Binary descriptors substitute real-valued elements with binary codes which are efficient for storage and matching, while deep learning obtains high quality representation by training numerous parameters with large amount of data.

For most existing deep binary descriptor learning approaches, binarization is an essential step to quantize each real-valued element into zero or one, which enhances the efficiency of the descriptors at the cost of quantization loss [18]. However, to the best of our knowledge, these methods directly perform binarization on the real-valued elements to obtain binary codes, which ignores the descriptor ambiguousness. Binarizing real-valued elements that lie in the boundary of quantization usually suffers from the "ambiguous bits", which fails to receive effective information from the corresponding input for reliable binarization due to the high sensitivity to noise.

We argue that there are implicit relationship among bits for the learned binary codes dubbed as bitwise interaction, and the related bits can provide extra instruction for the ambiguous bits as prior knowledge. For example, it is ambiguous to decide whether a person is tall or short in 5 feet 9 inches. However, the answer becomes more certain if we consider an additional gender bit of female or an age bit of young child. In this paper we propose GraphBit, a method that eliminates the ambiguity through bitwise interaction mining in a directed acyclic graph. The nodes of the graph are the elements in binary descriptors and the directed edges represent bitwise connections. In Figure 1 (a), DeepBit [34] ignores the reliability during the training procedure and the learned binary descriptors are ambiguous. On the contrary, GraphBit maximizes the mutual information between the binary descriptors and input samples with the mined bitwise

- *Ziwei Wang, Han Xiao, Jie Zhou and Jiwen Lu are with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Automation, Tsinghua University, Beijing 100084, China. E-mail: wang-zw18@mails.tsinghua.edu.cn, h-xiao20@mails.tsinghua.edu.cn, jzhou@tsinghua.edu.cn, lujiwen@tsinghua.edu.cn.*
- *Yueqi Duan is with the Beijing National Research Center for Information Science and Technology (BNRist), Department of Electrical Engineering, Tsinghua University, Beijing 100084, China. E-mail: duanyueqi@tsinghua.edu.cn.*
- *Part of this work was presented in [19].*
- *Code: https://github.com/ZiweiWangTHU/GraphBit.git.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3161600, IEEE Transactions on Pattern Analysis and Machine Intelligence

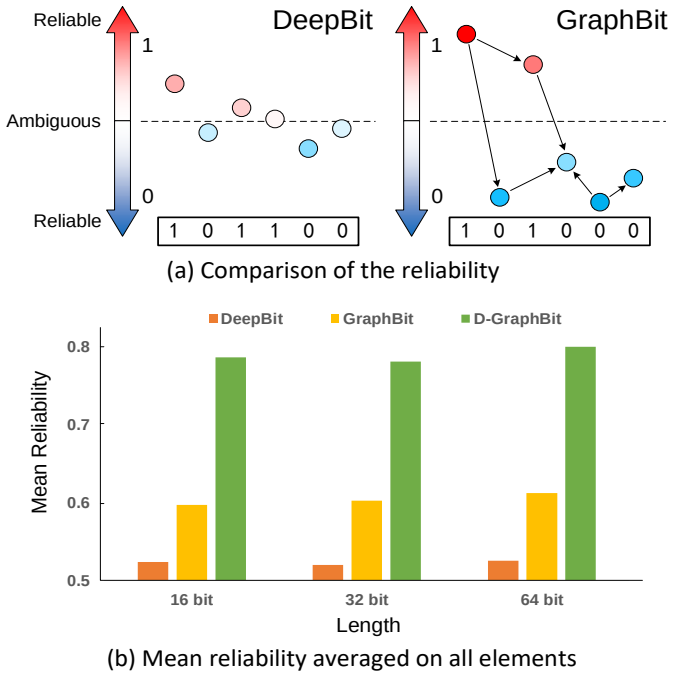


Figure 1. Comparison of the reliability between DeepBit [34] and GraphBit. We define the reliability of each bit as the Shannon entropy that measures the uncertainty of the binary descriptor according to (1). Binary codes with low reliability represent visual information in high uncertainty, which are sensitive to noise in binarization with bit reversion. In Figure 1 (a), the position and color of the dots demonstrate the reliability of binary codes, and the arrows represent the directed bitwise interaction. Figure 1 (b) shows the mean reliability averaged on all elements for 16-bit, 32-bit and 64-bit binary descriptors on CIFAR-10 [31]. DeepBit fails to consider the reliability of learned binary descriptors and obtains hash codes with ambiguous bits, while our GraphBit learns more confident binary codes due to the mined bitwise interaction. Moreover, D-GraphBit further improves the reliability with dynamic bitwise interaction that is optimal for different input samples. (Best viewed in color.)

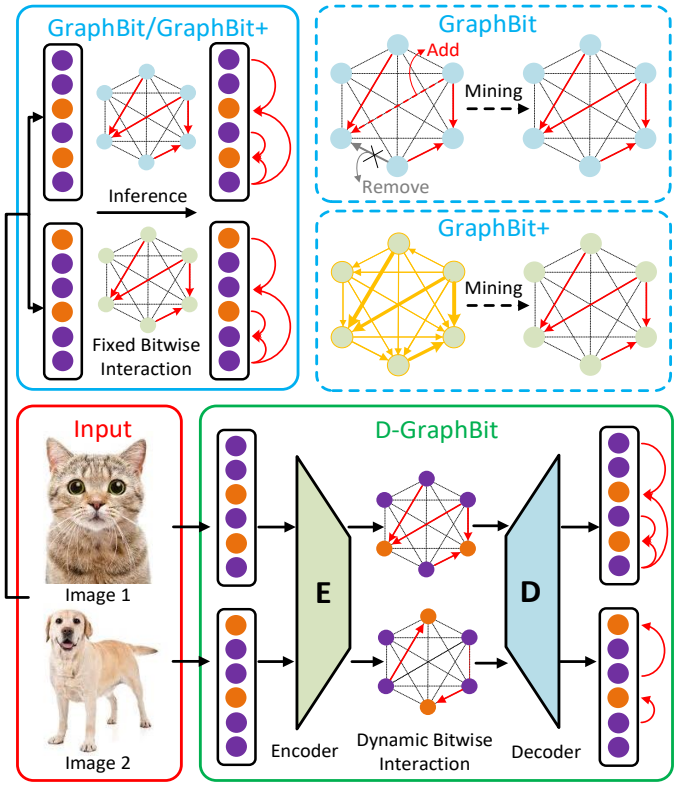


Figure 2. Comparison among the proposed GraphBit, GraphBit+ and D-GraphBit, where the rectangles with colorful circles represent the learned binary descriptors. The purple and orange circles mean reliable and ambiguous bits respectively, and the red arrows demonstrate instruction that eliminates ambiguity in the binary codes. GraphBit and GraphBit+ mine the fixed bitwise interaction for different input samples via non-differentiable reinforcement learning and differentiable hypergraph optimization respectively. In order to acquire the optimal solution for various input samples, D-GraphBit obtains the dynamic bitwise interaction for each instance via the efficient GraphMiner based on graph convolutional networks. (Best viewed in color.)

interaction. More specifically, we employ neural networks to parameterize the possibilities of elements being quantized into one in a binomial distribution, and define the reliability of each bit as the Shannon entropy that measures the uncertainty of the binary descriptor. We simultaneously train the parameters of CNN and the structure of the graph, maximizing the mutual information between binary descriptors and the observed input under the instruction from the related bits for ambiguity elimination. For bitwise interaction mining, deep reinforcement learning is leveraged to effectively explore the large search space, where we define the action to *add* or *remove* directed connections between nodes and apply the current graph structure as the state. Aiming to mine the bitwise interaction more efficiently without heavy search cost in reinforcement learning, we further present GraphBit+ that employs the gradient descent to optimize a hypergraph of bitwise interaction in differentiable search.

In fact, the optimal bitwise interaction of learned binary descriptors varies across different samples. While GraphBit mines the fixed bitwise interaction for binary codes of all input, the instruction from the mined bitwise interaction is usually suboptimal for ambiguity elimination. In order to address these limitations, we further present unsupervised binary descriptor learning via dynamic bitwise interaction mining (D-GraphBit). More specifically, we represent the dynamic bitwise interaction by the adjacency matrix which demonstrates the correlation among different bits, and the adjacency matrix is learned via the proposed GraphMiner based on graph convolutional networks. For each sample, GraphMiner encodes the original binomial distribution of binarization from the bit space to the interaction space containing the latent graph structure of bitwise interaction, reasons the bitwise interaction in the interaction space and decodes the bitwise interaction back to the bit space to acquire the distribution of reliable binary descriptors. Figure 1 (b) illustrates the reliability of each bit on CIFAR-10 for 16-bit binary descriptors, where GraphBit significantly outperforms DeepBit and D-GraphBit further enhances the reliability. Figure 2 depicts the comparison among GraphBit, GraphBit+ and D-GraphBit. Extensive experiments on CIFAR-10 [31], NUS-WIDE [15], ImageNet-100 [16], Brown [8] and HPatches [4] show that our GraphBit, GraphBit+ and D-GraphBit outperform the state-of-the-art unsupervised binary descriptors due to the strong reliability. Moreover, our method can be integrated to other unsupervised hash code learning techniques as a plug-and-play module to further strengthen the performance.

This paper is an extended version of our conference paper [19], where we make the following new contributions:

(1) We present a differentiable search method (GraphBit+) for efficient bitwise interaction mining, so that the heavy search cost of reinforcement learning caused by training difficulties is significantly reduced.
(2) We propose a D-GraphBit method by learning dynamic bitwise interaction for each instance, and the instruction from the mined bitwise interaction is optimal for all samples that eliminates the ambiguous bits accurately.
(3) We conduct extensive experiments on a wide variety of datasets to evaluate the proposed GraphBit, GraphBit+ and D-GraphBit, and the results show the effectiveness and the efficiency of the presented methods. Moreover, we combine our techniques with other unsupervised binary descriptor learning techniques to further enhance the vanilla models.

The presented GraphBit, GraphBit+ and D-GraphBit all aim to mine the optimal bitwise interaction that is modeled as a graph for reliable quantization in unsupervised binary descriptor learning, and they respectively leverage different search algorithms including reinforcement learning, differentiable search and graph neural networks with various advantages and limitations in accuracy, inference latency and training cost as shown in Table 5. D-GraphBit outperforms others with respect to accuracy due to the dynamic bitwise interactions, and slightly increase the inference latency resulted from the lightweight GraphMiner. GraphBit+ significantly reduces the training cost with slight accuracy degradation compared with GraphBit. Therefore, users can choose the appropriate one for unsupervised binary descriptor learning according to the accuracy requirement, training cost budget and the hardware configurations for deployment.

2 Related Work

In this section, we briefly review five related topics including: 1) binary descriptors, 2) unsupervised learning, 3) deep reinforcement learning, 4) differentiable search and 5) graph neural networks.

Binary Descriptors: Binary descriptors have attracted much attention in computer vision due to their efficiency for storage and matching for deployment, where early works can be traced back to binary robust independent elementary feature (BRIEF) [10], binary robust invariant scalable keypoint (BRISK) [32], oriented FAST and rotated BRIEF (ORB) [44] and fast retina keypoint (FREAK) [1]. BRIEF computed binary descriptors through the intensity different tests between pixels. BRISK obtained scale and rotation invariance by leveraging a circular sampling pattern. ORB improved BRIEF by applying scale pyramids and orientation operators. FREAK utilized retinal sampling grid for acceleration.

As hand-crafted binary descriptors are heuristics and usually require strong prior knowledge, a number of learning based approaches have been proposed and achieved outstanding performance [52], [55], [56], [21]. For example, Strecha *et al.* [52] proposed LDA-Hash by applying linear discriminant analysis (LDA) before binarization. Trzcinski *et al.* [55] presented D-BRIEF by learning discriminative projections through similarity relationships. They also learned hash functions with boosting to obtain BinBoost [56]. Fan *et al.* [21] proposed a receptive fields descriptor (RFD) by thresholding responses of two different receptive fields, rectangular pooling area and Gaussian pooling area.

More recently, several deep binary descriptor learning approaches have been proposed [36], [63], [22], which achieve the state-of-the-art accuracy performance. Liu *et al.* [36] encouraged the similar images to obtain closer binary descriptors and punished semantically dissimilar samples whose binary codes had short Hamming distance, and the learned binary representations could accurately preserve the topology of the semantic space. Zhang *et al.* [63] mined the semantic similarity between labeled and unlabeled images and generated pseudo labels for unlabeled images to effectively leverage the limited supervision. Ghasedi *et al.* [22] employed Generative Adversarial Networks (GANs) [24] to learn binary codes through which the reconstructed images were encouraged to have minimum semantic discrepancy with real ones. Nevertheless, these deep binary descriptors fail to exploit bitwise interactions, which suffer from ambiguous bits.

Unsupervised Learning: Unsupervised learning enables models to learn from numerous unlabeled data without expensive annotation cost. Clustering methods [12] utilize the cluster index as the pseudo class labels to train the representation model. Self-supervised learning approaches [17], [41] design the pretext tasks to provide hand-crafted auxiliary supervision with human priors, where the learned semantics are assumed to be transferred to the downstream tasks. Instance specificity analysis methods [27], [13] regard each sample as a independent class, and take the instance and the variant counterparts as positive pairs. These methods assume that the instance semantic similarity is automatically extracted by the instance-wise supervision. Neighborhood discovery approaches [59] progressively mine the instance-to-instance relationship in local regions with class consistency maximization. In this paper, we employ the energy constraint to learn discriminative binary descriptors in an unsupervised manner.

Deep Reinforcement Learning: Deep Reinforcement learning aims to learn the policy of sequential actions for decision-making problems with discriminative deep neural networks. Deep reinforcement learning algorithms have obtained very promising results on a wide variety of vision tasks such as object detection [43], visual tracking [28], network architecture search [65] and many others. More recently, deep reinforcement learning approaches have been employed in visual representation learning [54], [6], [7]. For example, Truong *et al.* [54] applied reward to compute the suitability of detected keypoints based on the final registration quality. Bhowmik *et al.* [6] directly optimized the high-level task loss of image matching with principles from reinforcement learning, so that the non-differentiable key point selection, descriptor matching and robust model fitting were incorporated in a complete pipeline. Tyszkiewicz *et al.* [57] bridged the training and inference stage of local feature learning via the probabilistic model, where analytical policy gradient was presented to enhance the optimization convergence. However, to our best knowledge, deep reinforcement learning has not been extended to binary representation extraction, which is of significant importance in visual analysis tasks.

Differentiable Search: In order to efficiently explore the large search space, differentiable search strategies have been utilized for network architecture search [35], mixed-precision quantization [9] and feature aggregation [26]. Differentiable search usually models each choice as a component in the superstructure, and optimizes the branch importance with gradient descent to acquire the optimal solution. Liu *et al.* [35] relaxed the architecture space to be continuous where component importance weights and hypernet parameters were jointly trained. Cai *et al.* [9] assigned different bitwidths for each parallel branches in the hypernet for mixed-precision quantization, and selected the bitwidth with the largest value during inference to achieve the optimal accuracy-complexity trade-off. Guan *et al.* [26] optimized the feature weights via the bridge loss that strengthened knowledge distillation via the bi-directional path between student and teachers. To further reduce the search cost caused by reinforcement learning in GraphBit, we generalized the differentiable search strategies to bitwise interaction mining.

Graph Neural Networks: Graph neural networks (GNN) can learn informative representations for non-Euclidean data in action recognition [49], person re-identification [62] and visual matching [46]. Sarlin *et al.* employed graph neural networks to predict the cost function of the optimal transportation, which was the alternative problem of the original feature matching. Mazur *et al.* [39] modeled the compact data representations as the differentiable weighted graphs and mapped the data into a non-vector space where the shortest path was used as the metric. Moreover, graph convolutional networks [30], [14] were further proposed to hierarchically aggregate information to learn discriminative representation for non-Euclidean data. For instance, Chen *et al.* [14] proposed the graph-based global reasoning networks that adaptively selected the receptive field of convolution based on pixel correlation. In this paper, we extend the graph convolutional networks to mine the dynamic bitwise interaction for binary descriptor learning in the interaction space.

3 Bitwise Interaction Mining for Binary Descriptor Learning

In this section, we first introduce the reliability of binary codes and present the objective function of GraphBit. Then we detail the deep reinforcement learning model for bitwise interaction mining. Finally, we present the differentiable search of Graphbit+ that significantly reduces the cost of interaction mining.

3.1 Reliability of Binary Codes

Feature elements lying in the quantization boundary are sensitive to noise in binary descriptor learning, and the ambiguous bits in low reliability result in weak discriminative power due to the lack of effective information from the inputs. Different from most existing methods, we explicitly learn reliable binary codes for accurate visual analysis. As shown in Figure 3, we utilize the architecture of VGG16 [50] as the backbone of the deep hashing model, and substitute the softmax layer with a fully-connected layer followed by an activation function of sigmoid. We first perform a sigmoid function at the end of CNN to normalize each element into the range $[0, 1]$ for reliability estimation, which parameterizes the possibility of being quantized to one in a binomial distribution. The binarization results are more certain for the probability close to one or zero, while are ambiguous for the value near 0.5. Hence, we define the reliability according to the Shannon Entropy:

$$r(b_{kn}) = 1 + \frac{1}{Z}[s_{kn}\log s_{kn} + (1 - s_{kn})\log(1 - s_{kn})] \quad (1)$$

where b_{kn} and s_{kn} respectively mean the k_{th} binary descriptor bit and k_{th} real-valued feature element after sigmoid for the n_{th} sample. $r(b_{kn})$ represents the reliability of b_{kn}, and Z is a constant that normalizes the reliability into the range $[0, 1]$. The Shannon Entropy demonstrates the uncertainty of binary descriptors, which can be applied to evaluate the reliability. Our goal is to maximize the reliability for binary codes through the mined bitwise interaction.

3.2 Objective Function

Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ be the N input samples of the image set. The objective of GraphBit is to simultaneously learn deep binary descriptors $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_N]$ and the adjacency matrix $\Phi \in [0, 1]^{K\times K}$ of the graph where K is the length of the binary descriptors. The element in the i_{th} row and j_{th} column of Φ is denoted as ϕ_{ij}, which equals to one if the i_{th} bit provides instruction for the j_{th} bit to eliminate ambiguity. In order to describe bitwise relationship, we denote the relationship between random variables X and Y by mutual information $I(X; Y)$, which describes the decrease of entropy of X when Y is tractable:

$$I(X; Y) = H(X) - H(X|Y), \quad (2)$$

The entropy is defined as follows:

$$H(X) = -\mathbb{E}_{x\sim p(X)}[\log p(x)] \quad (3)$$

$$H(X|Y) = -\mathbb{E}_{y\sim p(Y)}[\mathbb{E}_{x\sim p(X|Y)}[\log p(x|y)]] \quad (4)$$

where $H(X)$ and $H(X|Y)$ reveal the uncertainty of the variables. Inspired by the above motivations, we formulate the following objective function to learn GraphBit:

$$\begin{aligned}\min J &= J_1 + \alpha J_2 + \beta J_3 \\ &= \sum_{k=1}^{K} || \sum_{n=1}^{N} (b_{kn} - 0.5)||_2^2 - \alpha \sum_{n=1}^{N}\sum_{k=1}^{K} I(b_{kn}; \mathbf{x}_n, \hat{\boldsymbol{b}}_n) \\ &+ \beta \sum_{n=1}^{N}\sum_{k=1}^{K} ||p(b_{kn}|\mathbf{x}_n) - p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)||_1, \end{aligned} \quad (5)$$

where α and β are two parameters to balance the weights of different terms, and $\hat{\boldsymbol{b}}_n$ means the binary descriptor without considering the bitwise interaction. $||\cdot||_1$ and $||\cdot||_2$ stand for the L1 and L2 norm respectively. $p(b_{kn}|\mathbf{x}_n)$ represents the conditional distribution of b_{kn} given the input $\boldsymbol{x}_n$, and $p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)$ means the binomial distribution of b_{kn} under the condition $\mathbf{x}_n$ and $\hat{\boldsymbol{b}}_n$. The difference between $p(b_{kn}|\mathbf{x}_n)$ and $p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)$ is that the latter considers the related bits according to the mined bitwise interaction. The above two

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3161600, IEEE Transactions on Pattern Analysis and Machine Intelligence

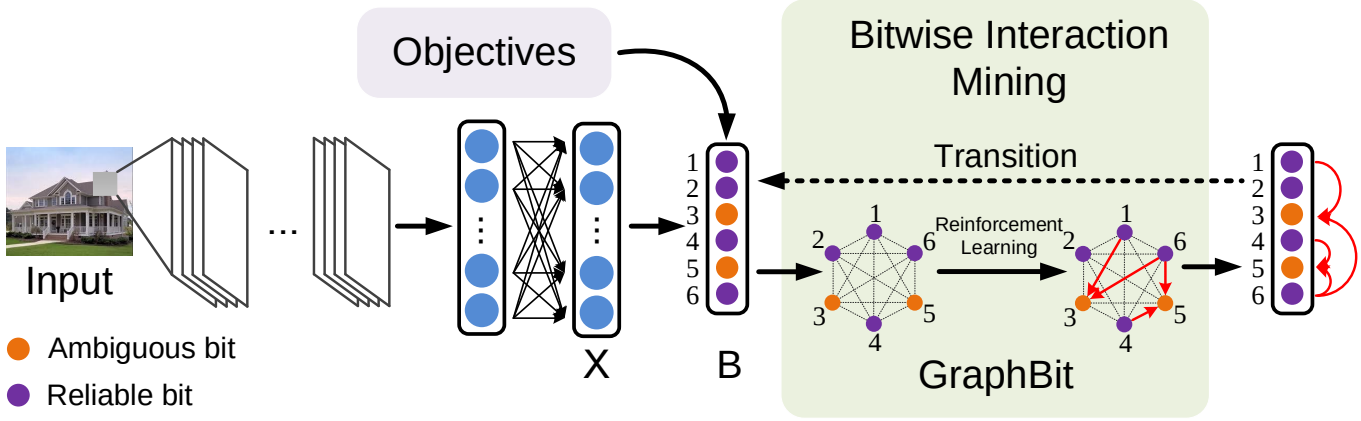


Figure 3. The flowchart of the proposed GraphBit. For each input image, we first learn a normalized feature by the deep hashing model with the VGG16 architecture where the softmax layer is substituted with a fully-connected layer followed by a sigmoid function. The normalized feature ranges from 0 to 1, parameterizing the possibility of being binarized into one. Then, we simultaneously mine the bitwise interaction via reinforcement learning and optimize the parameters of the deep hashing model with the mined bitwise interaction, which eliminates the ambiguity of the binary descriptors with enhanced reliability.

conditional distributions are parameterized by the backbone networks in the following:

$$p(b_{kn}|\mathbf{x}_n) \sim B(\sigma(t_{kn}))$$
$$p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n) \sim B(\sigma(t_{kn} + \sum_{i=1}^{K} w_{ik}^b \phi_{ik} t_{in})) \quad (6)$$

where t_{kn} means the k_{th} real-valued feature element of the k_{th} sample and σ represents the sigmoid function. $B(x)$ stands for the binomial distribution with the probability x being quantized into one. Meanwhile, w_{ik}^b is the element in the i_{th} row and j_{th} column of the learned interaction weight matrix $\boldsymbol{w}^b$, which demonstrates the influence on b_{kn} from b_{in} for all input samples according to the mined bitwise interaction. Meanwhile, the element ϕ_{ik} in the adjacency matrix that equals to one indicates the instruction from b_{in} to b_{kn}. We detail the physical meanings of the three terms in the objective function as follows:

1) J_1 is to make each bit in the learned GraphBit evenly distributed. If an element in the learned binary descriptors stay the same for all the samples, it would present no discriminative power. Instead, we encourage each bit to be zeros for half of the samples and ones for the others to convey more information.
2) J_2 expects the reliability of the learned binary descriptors to be enhanced under the instruction of the related bits and the corresponding input, where maximizing the mutual information is equivalent to reliability enhancement due to the consistent form. For the independent bits with no bitwise interaction, the uncertainty is reduced by maximizing the mutual information between only the input and the hash codes.
3) J_3 aims to prevent the interacted bits to become trivial with regularization. Under the guidance of J_2, those ambiguous bits that fail to collect effective information from the inputs may tend to receive extra directions from other reliable bits. However, they may become redundant as a repeat of the related bits if suffering from too strong instructions. Therefore, the constraint of J_3 is to guarantee the independence of the interacted bits.

We apply variational information maximization to simplify J_2 in (5) with the upper bounding, which is then approximated with Monte Carlo simulation [5]. J_2 can be rewritten in the following:

$$\begin{aligned}
&I(b_{kn}; \mathbf{x}_n, \hat{\boldsymbol{b}}_n) \\
&= H(b_{kn}) - H(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n) \\
&= H(b_{kn}) + \mathbb{E}_{\mathbf{x}_n \sim \mathbf{X}}[\mathbb{E}_{b'_{kn} \sim p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)}[\log p(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)]] \\
&= H(b_{kn}) + \mathbb{E}_{\mathbf{x}_n \sim \mathbf{X}}[D_{KL}(p(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)||q(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)) \\
&+ \mathbb{E}_{b'_{kn} \sim p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)}[\log q(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)]] \\
&\geq H(b_{kn}) + \mathbb{E}_{\mathbf{x}_n \sim \mathbf{X}}[\mathbb{E}_{b'_{kn} \sim p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)}[\log q(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)]]
\end{aligned}$$

where b'_{kn} means the variable sampled from the conditional distribution of b_{kn}, and $q(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)$ is the auxiliary distribution for the posterior distribution $p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)$. In this paper, we parametrize $q(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)$ based on the mined bitwise interaction defined in (6). When the auxiliary distribution approaches the true posterior distribution, the bound becomes tight because $D_{KL}(q(b'_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n)||q(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n))$ approaches 0. Since the priors for each bit are set to binomial distribution with equal possibility to be zero or one, $H(b_{kn})$ is regarded to be constant.

During the training stage, we simultaneously optimize the deep hashing model and the interaction weight matrix $\boldsymbol{w}^b$. In inference, we obtain the reliable binary descriptors according to the conditional distribution given the input and the original distribution of binary descriptors with the mined bitwise interaction.

3.3 Deep Reinforcement Learning for Bitwise Interaction Mining

In order to effectively explore the large search space, we employ the policy gradient for bitwise interaction mining. We denote the policy as $\pi_\theta(a|s)$ where θ, a and s represent the parameters of the policy network, the action and the state respectively. The policy takes the optimal action for a given state to achieve the goal, which maximizes the expected reward over the entire search process. Mining

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3161600, IEEE Transactions on Pattern Analysis and Machine Intelligence

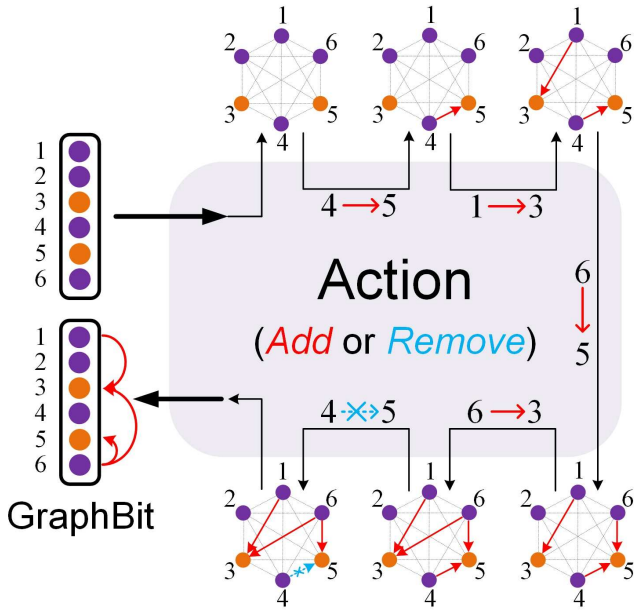


Figure 4. An example of deep reinforcement learning based bitwise interaction mining. We sequentially *add* directed connections of the 4th-5th bits, the 1st-3rd bits, the 6th-5th bits and the 6th-3rd bits, and then *remove* the connection of the 4th-5th bits. We repeat the process of bitwise interaction mining until finalizing the structure of graph.

bitwise interaction for ambiguity elimination can be viewed as a Markov Decision Process (MDP). At each step, the agent takes the action to *add* or *remove* a directional connection between bits based on the current structure of the graph, which iteratively explores the bitwise interaction to maximize the reward. The policy network recurrently *adds* and *removes* the edges until convergence or achieving the maximum step. At the end of the sequence, we retrain the parameters of CNN with the learned structure of graph under the guidance of the objective function.

States: The state space $\mathcal{S}$ represents the current structure of the graph, which can be defined as a binary matrix $W_s \in \{0,1\}^{K\times K}$. For the element $w_{ij}^s \in W_s$, it equals to one if a directed edge exists from the ith bit to the jth bit, and equals to zero otherwise.

Action: Given the current graph W_s, the agent aims to select one action from all possible connections and disconnections. $\mathcal{A}$ is the set of actions divided into three categories: $\mathcal{A}_c \bigcup \mathcal{A}_r \bigcup \{stop\}$. The action to *add* a bitwise edge is denoted as $\mathcal{A}_c$, while $\mathcal{A}_r$ represents to *remove* bitwise connections. The action of *stop* is executed for convergence or the maximum time step. Figure 4 shows an example of stage transition with the actions. $W_t \in [0,1]^{K\times K}$ parameterizes the probability of actions that adds edges on the graph, whose element w_{ij}^t in the i_{th} row and j_{th} column represents the probability of adding connection from the ith bit to the jth bit. We select the actions based on the following rules:

1) *Add*: We connect the ith bit to the jth bit if the sampling strategy based on W_t selects the element w_{ij}^t and the maximal element in W_t is larger than k_1.
2) *Remove*: We disconnect the original edge from the ith bit to the jth bit if $w_{ij}^t \leqslant k_2$.
3) *Stop*: We terminate the current epoch of bitwise interaction mining when achieving the maximum time step or convergent reward.

In the action selection criteria, the hyperparameter k_1 prevents excess connections in the graph to avoid redundant bits, where only bits with high connection probability are regarded to be interacted. The other hyperparameter k_2 decides the probability of interaction removal, because low probability to connect an edge indicates keeping it disconnected.

Reward Function: We define the reward function $\mathcal{R}(\mathcal{S},\mathcal{A})$ in round t as following:

$$r(s_t, a_t) = J(s_t) - J(s_{t+1}) \tag{7}$$

where $r(s_t,a_t) \in \mathcal{R}(\mathcal{S},\mathcal{A})$ is the reward for the action a_t in the state s_t, and $J(s_t)$ is objective function of sample batches in the state s_t. We consider the bitwise interaction enabling loss degradation in high quality, which enhances the discriminative power and the reliability of the learned GraphBit. We utilize the REINFORCE algorithm [60] to update parameters in the policy network in response to the rewards from the environment.

3.4 Differentiable Search for Bitwise Interaction Mining

Reinforcement learning based search strategies effectively explore the large search space for bitwise interaction mining, while still suffers from high training cost due to the following two reasons. First, computing the reward function requires to feed forward the sample batch two times for loss difference acquisition. Second, the agent training usually converges with a large number of epochs because of the reward fluctuation in the environment [40]. In order to decrease the search cost, we present differentiable search strategy called GraphBit+ for bitwise interaction mining. Differentiable search usually constructs a superstructure where various choices in the search space form different branches, and adds the output of all branches for inference. The model parameters and component importance weights are optimized via gradient descent, and the edges with the high importance weight are selected as the bitwise interaction. In order to mine the bitwise interaction with differentiable search, we rewrite the conditional distribution of binary descriptors given bitwise interaction and input in the following:

$$p(b_{kn}|\mathbf{x}_n, \hat{\boldsymbol{b}}_n) \sim B(\sigma(t_{kn} + \sum_{i=1}^{K} w_{ik}^b a_{ik} t_{in})) \tag{8}$$

where a_{ij} means the element in the i_{th} row and j_{th} column of normalized adjacency weight matrix. The normalization is implemented by $a_{ij} = \exp(a_{ij}^0)/\sum_{i=1}^{K}\sum_{j=1}^{K}\exp(a_{ij}^0)$, and a_{ij}^0 represents the original adjacency weights. Large a_{ij} represents high existence probability of the interaction from the i_{th} bit to the j_{th} one. On one hand, the adjacency weights should be reliable so that the interaction among bits clearly exists or not without obscurity. On the other hand, the adjacency weights are required to be sparse in order to avoid redundancy brought by excess bitwise interaction. To combine the above objectives, we formulate the loss function as follows despite of the objectives (5) in GraphBit:

$$J_4 = -\sum_{i=1}^{K}\sum_{j=1}^{K} ||a_{ij} - 0.5||_2^2 \tag{9}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3161600, IEEE Transactions on Pattern Analysis and Machine Intelligence

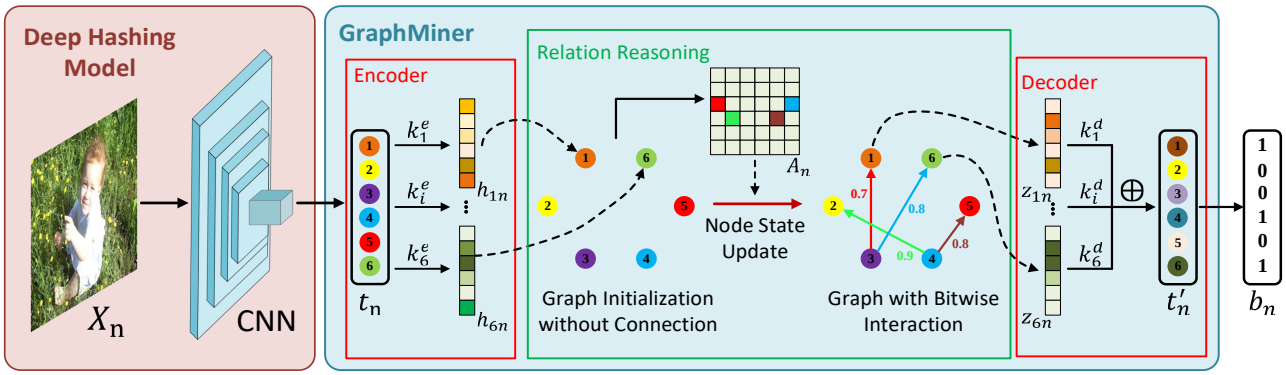


Figure 5. The pipeline of the presented D-GraphBit. The deep hashing model first learns the binary descriptor distribution without bitwise interaction, which is then mapped to the interaction space by the encoder. The interaction reasoning network mines the dynamic bitwise interaction in the interaction space and the decoder transforms the interacted node features to the distribution of reliable hash codes, so that the ambiguous bits are eliminated with the optimal bitwise interaction.

J_4 maximizes the distance between the elements in the normalized adjacency weight matrix and 0.5, so that the element is enforced to approach one or zero to obtain the reliable graph. Since the summation over elements in the normalized adjacency weight matrix equals to one, J_4 also sparsifies the connected edges with most elements remaining near zero. Therefore, only bitwise interaction with strong correlation is mined without bringing redundancy in binary descriptor learning. Combining all loss terms, we rewrite the overall learning objectives for GraphBit+ as follows:

$$J_{overall} = J_1 + \alpha J_2 + \beta J_3 + \gamma J_4 \tag{10}$$

For each epoch of bitwise interaction mining, the adjacency weights a_{ij} and the interaction weights w^b_{ij} are iteratively optimized. When adjacency weight update completes, the adjacency matrix element ϕ_{ij} is set to one for top-k element a_{ij} in adjacency weight matrix and is set to zero otherwise. Afterwards, the interaction weights are finetuned where the binary descriptors are sampled from the conditional distribution in (6) for loss computation. We obtain the reliable binary descriptors according to the conditional distribution given interacted bits and input in (6) during inference.

4 Learning Unsupervised Binary Descriptors via Dynamic Bitwise Interaction Mining

We first provide an overview of the proposed D-GraphBit. Then we detail the presented GraphMiner that consists of a encoder that maps the descriptors from the bit space to the interaction space, an graph convolutional layer that dynamically mines the bitwise interaction in the interaction space and a decoder that obtains the reliable binary codes. Finally, we present the training details of our D-GraphBit.

4.1 Overview

The optimal bitwise interaction of binary descriptors varies across different samples. While the GraphBit mines the bitwise interaction statically, the instruction from the fixed graph is suboptimal for the binary descriptors of all samples in ambiguity elimination. In order to address these limitations, we further propose D-GraphBit to learn the interaction graph for each sample via the graph convolutional network based dynamic bitwise interaction mining module called GraphMiner, so that the instruction from the optimal bitwise interaction is utilized to extract reliable binary descriptors.

Figure 5 demonstrates the overall pipeline of the proposed D-GraphBit. The binomial distribution of binary descriptors without bitwise interaction is parameterized by the deep hashing model, which is fed forward to the GraphMiner to generate binomial distribution of reliable binary descriptors with dynamic bitwise interaction mining. The presented GraphMiner includes three parts: an encoder that maps the distribution of binary descriptors without bitwise interaction from the bit space to the interaction space, an interaction reasoning network that mines the implicit bitwise interaction in the interaction space and a decoder that transforms the interacted node features in the interaction space to the distribution of reliable hash codes in the bit space. In the following subsections, we formulate each step of the GraphMiner in details.

4.2 Encoding from Bit Space to Interaction Space

The encoder aims to map the distribution of binary descriptors without bitwise interaction in the bit space to the node feature in the interaction space, where dynamic bitwise interaction can be efficiently mined by the graph convolutional networks. Since each bit is regarded as a node in the interaction graph, we first learn the independent node features $\boldsymbol{h}_{in}$ of the i_{th} bit for the n_{th} binary code that aggregates the correlation with other bits. We obtain the independent node features according to the following formulation:

$$\boldsymbol{h}_{in} = \boldsymbol{k}^e_i \boldsymbol{t}_n \tag{11}$$

where $\boldsymbol{t}_n$ means the real-valued features of the deep hashing model for the n_{th} sample and $\boldsymbol{k}^e_i \in \mathbb{R}^{K\times K}$ represents the encoding matrix for the i_{th} independent node feature. The global information of the binary descriptors without bitwise interaction is embedded into various independent node features by different encoding matrices, which informatively represent the node states for the following dynamic bitwise interaction mining.

We employ K fully-connected layers in parallel for the linear mapping, which attains many benefits including the following aspects. First, the optimization of the encoding

matrices implemented as fully-connected layers can be combined in the training pipeline of deep hashing models so that end-to-end learning is enabled. Second, linear mapping is simple for implementation and also efficient due to the low computational cost. Finally, the low model complexity avoids overfitting for the independent node feature learning, where low-dimensional real-valued features from the deep hashing model only contain limited information.

4.3 Mining Dynamic Bitwise Interaction in Interaction Space

The interaction space contains the latent graph structure of bitwise interaction for reliable binary code learning. Mining the dynamic bitwise interaction is equivalent to capturing the relation among different independent node features, which can be effectively and efficiently discovered by graph convolution. Graph convolutional networks are widely used techniques that have been proven to be effective to process non-Euclidean data such as point cloud [58] and social networks [33], because adjacency among nodes instead of the Euclidean distance among samples is used to measure the similarity. The graph convolution fuses information for each node from the adjacent nodes, and the relation among different nodes are learned via updating the adjacency matrix. Therefore, the graph convolution is compatible with other differentiable modules. We formulate the adjacency weight matrix of the n_{th} sample as $\boldsymbol{A}_n = \sigma(\boldsymbol{W}^a \boldsymbol{H}_n)$, where $\boldsymbol{W}^a$ is the learnable adjacency mining matrix and $\boldsymbol{H}_n$ is the concatenation of independent node features from the n_{th} sample. To remove the impact of the node degree on graph convolution, we scale the adjacency weight matrix to $\boldsymbol{A}_n^s = \boldsymbol{D}_n^{-0.5} \boldsymbol{A}_n \boldsymbol{D}_n^{-0.5}$. The degree matrix $\boldsymbol{D}_n$ is diagonal, where the element in the i_{th} row and column is the element summation in the i_{th} row of the adjacency weight matrix. The graph convolution acquires the interacted node feature $\boldsymbol{Z}_n$ for the n_{th} sample which contains the mined dynamic bitwise interaction [30]:

$$\boldsymbol{Z}_n = \sigma((\boldsymbol{I} + \hat{\boldsymbol{A}}_n)\boldsymbol{H}_n \boldsymbol{W}_g) \tag{12}$$

where $\boldsymbol{I}$ means the identity matrix considering self-correlation. Denoting the element in the i_{th} row and j_{th} column of $\boldsymbol{A}_n^s$ as $A_{n,ij}^s$, we represent the element in the same position of the normalized adjacency weight matrix $\hat{\boldsymbol{A}}_n$ by $\hat{A}_{n,ij} = \exp(A_{n,ij}^s) / \sum_{i=1}^{K} \sum_{j=1}^{K} \exp(A_{n,ij}^s)$. Moreover, the learnable state update matrix $\boldsymbol{W}_g$ projects the adjacency weight matrix to interacted node features. The graph convolution first performs Laplacian smoothing so that the independent node features are propagated across the graph for information fusion. According to the adjacency weight matrix where stronger bitwise interaction is represented by larger edge weights, the state of each node is updated for interacted node feature acquisition to demonstrate the mined dynamic bitwise interaction.

4.4 Decoding from Interaction Space to Bit Space

The updated node state represented by interacted node features $\boldsymbol{Z}_n$ considers the dynamic bitwise interaction, and the decoder aims to map the interacted node features from the interaction space back to the bit space for reliable binary descriptor acquisition. Similar to the encoder, we leverage the linear mapping to predict the distribution of binary descriptors with bitwise interaction as follows:

$$\boldsymbol{t}_n^{'} = \sum_{i=1}^{K} \boldsymbol{k}_i^d \boldsymbol{z}_{in} \tag{13}$$

where $\boldsymbol{t}_n^{'}$ means the parameters of the binomial distribution of the hash codes with bitwise interaction for the n_{th} sample. $\boldsymbol{z}_{in}$ is the i_{th} column of the interacted node features $\boldsymbol{Z}_n$ and $\boldsymbol{k}_i^d \in \mathbb{R}^{K \times K}$ represents the learnable decoding matrix from the interaction space to the bit space. In order to enable the end-to-end learning of the GraphMiner and the deep hashing model, we utilize K fully-connected layers in parallel as the decoding matrices, and then add output from all branches together to obtain the robust binary descriptors.

4.5 Training Details

Similar to the differentiable search for bitwise interaction mining, we expect the adjacency weight matrix to be reliable and sparse. Therefore, we modify J_4 shown in (9) as follows to adapt to the dynamic bitwise interaction:

$$J_4 = -\sum_{n=1}^{N} \sum_{i=1}^{K} \sum_{j=1}^{K} ||\hat{A}_{n,ij} - 0.5||_2^2 \tag{14}$$

By leveraging the overall objective function shown in (10), parameters in the deep hashing model and the GraphMiner are optimized jointly. During each epoch of dynamic bitwise interaction mining, we update the learnable parameters of GraphMiner in two stages. First, we learn the encoding matrix and the adjacency mining matrix simultaneously where the interacted node features are obtained via (12). In the second stage, we replace the normalized adjacency weight matrix with the adjacency matrix for interacted node feature acquisition to optimize the state update matrix and decoding matrix, which is written as follows:

$$\boldsymbol{Z}_n = \sigma((\boldsymbol{I} + \Phi_n)\boldsymbol{H}_n \boldsymbol{W}_g) \tag{15}$$

where Φ_n is the adjacency matrix of the n_{th} sample. The element in the i_{th} row and j_{th} column of Φ_n equals to one if the corresponding element of $\hat{\boldsymbol{A}}_n$ is among top-k and equals to zero otherwise. Moreover, we also leverage (15) to acquire binary descriptors with dynamic bitwise interaction in inference.

5 Experiments

In this section, we conducted comprehensive experiments to evaluate our method on CIFAR-10 [31], NUS-WIDE [15] and ImageNet-100 [16] for image retrieval, on Brown [8] for patch matching and on HPatches [4] for patch verification, image matching and patch retrieval. We first describe the implementation details and introduce the applied datasets. Secondly, we validate the effectiveness of bitwise interaction, reinforcement learning strategy and the differentiable search method, and further investigate the influence of the GCN based dynamic bitwise interaction mining via the ablation study. Thirdly, we compare our method with the state-of-the-art unsupervised binary descriptors to show

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3161600, IEEE Transactions on Pattern Analysis and Machine Intelligence

the superiority, and combine GraphBit, GraphBit+ and D-GraphBit with other unsupervised binary code learning techniques to further enhance the vanilla model. Finally, we compare the deployment efficiency and the training cost of different descriptor learning methods.

5.1 Datasets and Implementation Details

We first introduce the datasets we carried out experiments on and corresponding data preprocessing techniques:

CIFAR-10: The CIFAR-10 dataset includes $60,000$ images of size 32×32, which is categorized into 10 classes. We randomly selected $1,0000$ images ($1,000$ images per class) for the query set, and the rest $50,000$ images for the training set and also the retrieval database. Four pixels were padded on each side of the images which were cropped into the size of 32×32 randomly with normalization.

NUS-WIDE: The NUS-WIDE dataset contains $269,648$ images collected from Flicker with 81 manually classes. Two images are regarded as positive if they share at least one label and are negative otherwise. We only used the 21 most frequent classes, resulting in a total of $166,047$ images. We randomly chose $2,100$ images (100 images per class) as the query set and regarded the rest as the training set and the retrieval database. The images were warped and normalized to 64×64 before forward propagation.

ImageNet-100: ImageNet (ILSVRC12) contains approximately 1.2 million training and 50K validation images from $1,000$ categories, which is much more challenging because of its large scale and high diversity. ImageNet-100 dataset was collected by [11], where 100 classes were randomly selected from original ImageNet. Images in the training and validation sets of the selected classes were leveraged as the database and queries respectively. Meanwhile, 100 images per category were randomly chosen from the database to be the training samples. For fair comparison, we employed the same class selection and data split as those in [11]. Followed by data augmentation of bias subtraction applied in CIFAR-10, a 224×224 region was randomly cropped for training from the resized image whose shorter side was 256. For inference, we employed the 224×224 center crop.

Brown: Brown contains three subsets including Liberty, Notre Dame and Yosemite. Each of them consists of over 400K patches for training and 100K test pairs. Among the test pairs, half of them are positive and the rest are negative. We took one subset as the training set and others as the test sets, leading to six training-test combinations. Following the evaluation protocol of [8], we report the false positive rate at 95% recall (FPR95).

HPatches: HPatches provides visual analysis tasks including patch verification, image matching and patch retrieval. HPatches consists of 116 sequences in total, splitting into 57 with photometric changes and 59 with significant geometric deformations. Different levels of geometrical perturbations imposed on the images form the EASY, HARD and TOUGH patch groups. Following [4], we leveraged the mean average precision (mAP) as the evaluation metric.

We utilized VGG16 [50] as our deep hashing model, where the last softmax layer was substituted by a fully-connected layer for feature dimension reduction. We used a sigmoid function to normalize the real-valued feature into

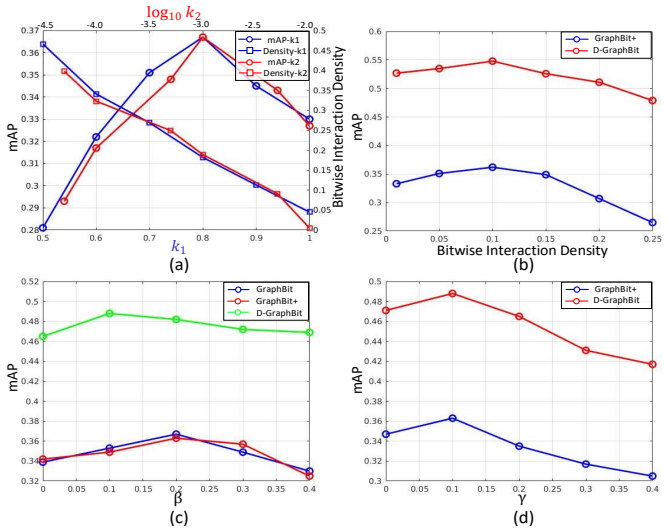


Figure 6. (a) shows the mAP variation of GraphBit with different interaction density, and (b) demonstrates that in GraphBit+ and D-GraphBit respectively. (c) illustrates the performance change of GraphBit, GraphBit+ and D-GraphBit with different β, and (d) depicts the mAP for GraphBit+ and D-GraphBit w.r.t. γ in the overall objective. The binary descriptor length was set to 32.

$[0, 1]$ before binarization to represent the probability of being one for each bit. The deep hashing model optimization and the bitwise interaction mining were alternatively implemented in each round, where we leveraged the Adam optimizer [29] with the batchsize 128. For the parameter update of the deep hashing model, the learning rate started from 0.001, 0.003, 0.01, 0.01 and 0.005 for CIFAR-10, NUS-WIDE, ImageNet-100K, Brown and HPatches respectively. The learning rate decayed twice at 50% and 80% of total epochs by multiplying 0.1, and the number of training epochs in each round for the above datasets were set as 10, 20, 20, 10 and 20 respectively. Rigid sign function was applied for feature binarization in order to prevent time-consuming sampling in inference.

For GraphBit, the policy network consisted of three convolutional layers, followed by two fully-connected layers and two deconvolutional layers. The hyperparameters α and β in (5) were set as 0.4 and 0.2 respectively. The state matrix was randomly initialized by a sparse matrix whose non-zero elements were one in order to avoid trivial action probability matrix W_t. For initialization, we generated a random number from the uniform distribution between zero and one for each element in the state matrix. We assigned the $ceil(0.01K^2)$ elements with the largest random number to one and remained others being zero, where $ceil(x)$ means the smallest integer larger than x. For the action selection strategy in the policy gradient, we gradually increased the parameters k_1 and k_2 during the iterations in the following strategy: $k_1 = min\{0.08T, 0.8\}$ and $k_2 = min\{\frac{T}{10K^2}, \frac{1}{K^2}\}$, where T means the index of rounds during the searching process and K represents the feature dimension. We only selected one combination of states, actions and rewards in the sample sequence to accelerate training. The total rounds for alternative bitwise interaction mining and deep hashing model learning were set to 10 with the learning rate 0.001. In each round, the policy networks were trained until reaching the maximal step $ceil(5 \times N/128)$ or reward convergence, where N demonstrates the number of training

Table 1
MAP (%) on CIFAR-10, NUS-WIDE and ImageNet-100 for image retrieval with binary descriptors in different code lengths, where the bitwise interaction types and the search methods were varied.

Bitwise Interaction	Search Methods	CIFAR-10			NUS-WIDE			ImageNet-100		
		16 bit	32 bit	64 bit	16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
None	–	27.95	32.77	36.16	44.15	45.50	49.17	9.70	11.02	15.67
Fixed	Random	30.16	33.25	37.57	45.01	45.66	49.35	9.89	12.05	16.36
	Reinforcement Learning	32.15	36.74	39.90	48.41	48.77	50.62	13.54	22.89	27.71
	Differentiable Search	31.68	36.25	39.51	48.18	49.20	50.57	13.31	22.66	27.50
Dynamic	Random	35.06	41.26	47.85	56.86	61.01	63.31	13.66	18.90	21.74
	Graph Neural Networks	**41.02**	**48.79**	**50.02**	**64.25**	**65.24**	**67.03**	**20.55**	**32.25**	**35.93**

samples. Reward convergence was achieved if the difference of average reward across five consecutive 100 steps was less than $1e$-4.

For GraphBit+, the number of rounds for iterative bitwise interaction mining and deep hashing model learning was set to 10. The adjacency weights and interaction weights were alternatively optimized for 10 epochs in each round, where α, β and γ in the overall loss were assigned to 0.4, 0.2 and 0.1 respectively with the learning rate 0.001. We sampled the binary descriptors with bitwise interaction according to (8) and (6) of the manuscript when training the adjacency weights and interaction weights respectively. For the discretization of adjacency matrix Φ, we selected the top-$\frac{K^2}{10}$ elements in adjacency weight matrix for bitwise interaction assignment.

For D-GraphBit, we applied K fully-connected layers in parallel for the encoder and decoder of GraphMiner respectively. The optimizer and batchsize of GraphMiner were set as the same as those of the deep hashing model, so that the GraphMiner and the deep hashing model could be trained jointly. The number of rounds for dynamic bitwise interaction mining and deep hashing mode optimization was assigned to 10. During each epoch of GraphMiner training, the encoding matrix and the adjacency mining matrix were optimized with the interacted node features obtained from (12) for 10 epochs in the first stage, and the state update matrix and the decoding matrix were updated with the interacted node features acquired via (15) for 10 epochs in the second stage. The learning rate was constantly set to 0.001. The discretization of adjacency matrix and the hyperpamaters in the overall loss shared the same settings of GraphBit+ except that β equaled to 0.1.

5.2 Ablation Study

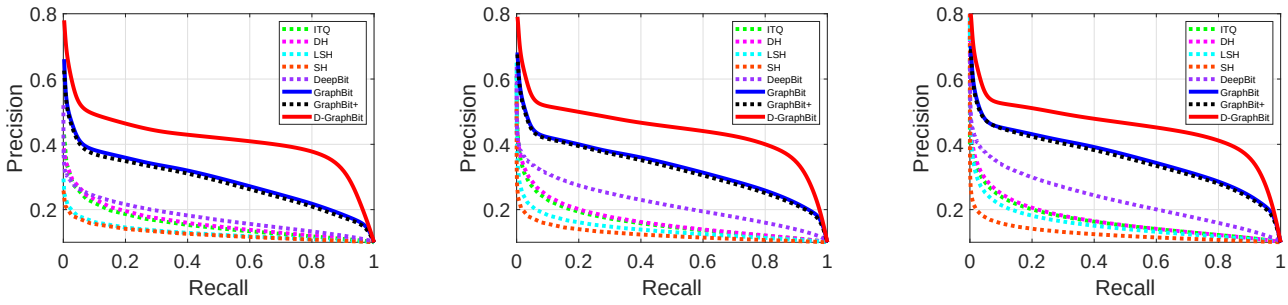
In this section, we analyze the effect of bit ambiguity elimination and the policy gradient search strategy in GraphBit, show the effectiveness of the differentiable search in GraphBit+, and demonstrate the superiority of GCN based dynamic bitwise interaction mining in D-GraphBit via the ablation study. Since reducing the uncertainty of the binary descriptors according to the bitwise interaction eliminates the ambiguity, we conducted extensive experiments by varying the bitwise interaction types and the search methods on CIFAR-10, NUS-WIDE and ImageNet-100 with binary descriptors in different code lengths. Meanwhile, we investigated the impact of the bitwise interaction density on the performance by changing the maximum of k_1 and k_2 in action sampling for Graphbit, and varied the k value of top-k in adjacency matrix discretization for GraphBit+ and D-GraphBit. Finally, we analyze the influence of the bitwise priors on mAP by varying the hyperparameter β and study performance variation with edge entropy in the interaction graph by changing γ in the overall objectives.

Performance w.r.t. the bitwise interaction types: Table 1 illustrates the mean average precision (mAP) w.r.t. different bitwise interaction types and search methods. The implementations of no bitwise interaction were maximizing mutual information between the input images and the learned binary descriptors without bitwise interaction. By comparing the results in the 3_{rd}, 5_{th}, 6_{th} and 8_{th} rows of Table 1, we conclude that the bitwise interaction significantly enhances the binary descriptors in various lengths across all three datasets. Moreover, the dynamic bitwise interaction achieves the optimal ambiguity elimination for different input samples, as the reliability of the hash codes is further strengthened.

Performance w.r.t. the search methods: In order to demonstrate the effectiveness of reinforcement learning and differentiable search for fixed bitwise interaction learning and GCN based dynamic bitwise interaction search, we also mined the fixed and dynamic bitwise interaction with random search. By observing the results from the 4_{th} to the 6_{th} rows of Table 1, we know that the efficiency of fixed bitwise interaction search is improved by our policy gradient and differentiable search methods. Without the discretization errors in bitwise interaction search, reinforcement learning slightly outperforms the differentiable search at the cost of lower training efficiency. Comparing the results in the 7_{th} and 8_{th} rows of Table 1, we draw the conclusion that the presented search strategy via the graph neural networks is highly effective as it outperforms random search by a large margin.

Performance w.r.t. the bitwise interaction density: The bitwise interaction density is defined as the L1 norm of the adjacency matrix divided by the number of elements, where higher density means more bitwise interaction. Figure 1(a) demonstrates the mAP variation and the corresponding bitwise interaction density for 32-bit binary descriptors obtained by GraphBit, where the horizontal axis depicts the maximum for k_1 or k_2 as we gradually increased them according to the implementation details. Meanwhile, the bitwise interaction density in GraphBit+ and D-GraphBit can be controlled by the k value of top-k in adjacency matrix discretization, and the resulted mAP is illustrated in Figure 1(b). For reinforcement learning based search, smaller k_1 and k_2 maximum in action sampling lead to high density. Large k of top-k in adjacency matrix discretization

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3161600, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) 16 bits (b) 32 bits (c) 64 bits

Figure 7. Comparison of Precision/Recall curves on the CIFAR-10 dataset under varying binary lengths (a) 16 bits, (b) 32 bits and (c) 64 bits with the state-of-the-art unsupervised binary descriptors.

Table 2
MAP (%) of top 1,000 returned images with different unsupervised binary descriptors on CIFAR-10, NUS-WIDE and ImageNet-100.

Category	Method	CIFAR-10			NUS-WIDE			ImageNet-100		
		16 bit	32 bit	64 bit	16 bit	32 bit	64 bit	16 bit	32 bit	64 bit
Hand-crafted	LSH	12.55	13.76	15.07	20.49	25.58	28.50	–	–	–
	PCA-ITQ	15.67	16.20	16.64	30.23	30.52	31.63	8.35	9.12	9.33
Unsupervised Learning	DeepBit	19.43	24.86	27.73	35.75	39.76	41.25	8.27	9.45	10.10
	DBD-MQ	21.53	26.50	31.85	37.12	39.98	42.15	9.33	10.01	11.76
	SADH	37.60	34.00	30.30	60.14	57.99	56.33	–	–	–
	DVB	35.30	37.20	39.60	55.40	56.20	59.50	–	–	–
	UDBD	32.24	36.17	39.60	52.30	57.90	59.80	–	–	–
	GraphBit	32.15	36.74	39.90	48.41	48.77	50.62	13.54	22.89	27.71
	GraphBit+	31.68	36.25	39.51	48.18	49.20	50.57	13.31	22.66	27.50
	D-GraphBit	41.02	48.79	50.02	64.25	65.24	67.03	20.55	32.25	35.93
	GreedyHash	44.80	47.20	50.10	60.92	65.33	69.53	30.44	39.87	47.42
	GreedyHash+GraphBit	48.64	53.35	62.02	68.28	69.71	70.45	33.68	46.74	56.15
	GreedyHash+GraphBit+	47.89	52.74	61.09	68.17	70.14	70.76	33.04	46.57	55.79
	GreedyHash+D-GraphBit	**54.10**	**59.54**	**63.75**	**71.73**	**72.92**	**74.07**	**42.22**	**57.52**	**61.61**

also increases the interaction density for GraphBit+ and D-GraphBit. The influence of density for fixed and dynamic bitwise interaction is similar, where medium density achieves the best performance. Low density fails to provide sufficient bitwise interaction for ambiguity elimination while high density connects bits with weak correlation, and both of them degrade the performance.

Performance w.r.t. the importance of bitwise priors: Figure 1(c) depicts the performance of hash codes in 32 bits with varying β. The hyperparameter β in the overall objectives controls the importance of priors provided by mined bitwise interaction on the learned binary code distribution, where smaller β shows that the bitwise priors affect the binary descriptors more significantly due to the less regularization. Figure 1(c) indicates that medium β utilizes bitwise interaction optimally, where small β ignores the knowledge obtained from the input images and large β fails to impose affective priors to eliminate ambiguity.

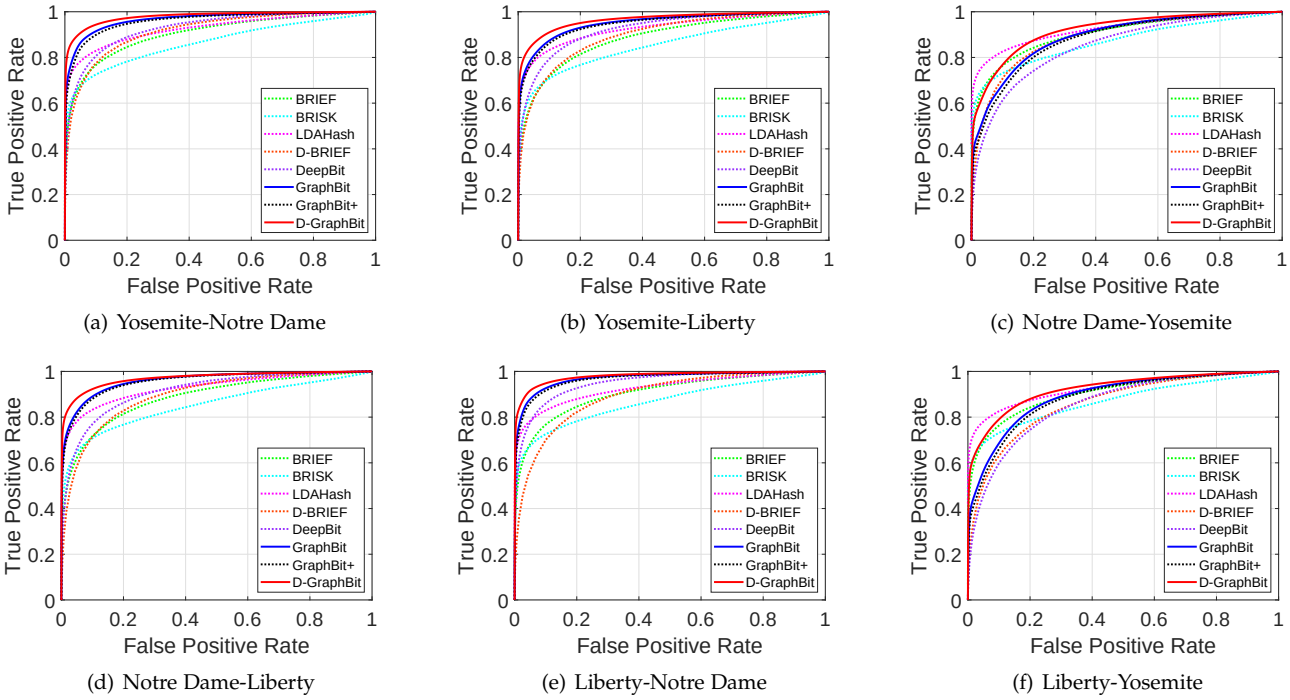
Performance w.r.t. edge entropy in the graph: Large γ in the overall objectives of GraphBit+ and D-GraphBit enlarges the contrast of adjacency weights, which enforces the mined bitwise interaction to be very reliable with low Shannon entropy of edges. In order to investigate the performance variation with the edge entropy in the interaction graph, we changed the hyperparameter γ for GraphBit+ and D-GraphBit where the results are demonstrated in Figure 1(d). Large γ only increases the adjacency weight of the most reliable interaction and remains others to be similarly low, while small γ fails to mine reliable interaction due to the insufficient contrast of adjacency weights.

5.3 Comparison with the State-of-the-art Methods

In this section, we compare the proposed GraphBit, GraphBit+ and D-GraphBit with the hand-crafted binary descriptors including SH [45], BRISK [32], BRIEF [10], ORB [44], LSH [2] and PCA-ITQ [23], the unsupervised binary descriptors including DH [20], DeepBit [34], DBD-MQ [18], SADH [47], BinGAN [64], BGAN+ [51], UDBD [61], DVB [48] and GreedyHash [53]. For reference, we listed the performance of real-valued features SIFT [37] and RootSIFT [3], and the supervised binary descriptors including LDAHash [52], D-BRIEF [55], BinBoost [56] and RFD [21]. The performance of the baseline methods was obtained by copying from the referenced paper or re-implementation.

Comparison on CIFAR-10, NUS-WIDE and ImageNet-100: Table 2 illustrates the mAP of different binary descriptors on CIFAR-10, NUS-WIDE and ImageNet-100 for image retrieval. Figure 7 demonstrates the precision/recall (PR) curve for various hash codes on CIFAR-10. Although DBD-MQ applied the multi-quantization to learn discriminative binary descriptors, it failed to utilize the bitwise interaction for ambiguity elimination that led to weak robustness. On the contrary, our GraphBit and GraphBit+ mine the bitwise interaction and then adopt the bitwise priors to enhance the reliability of binary codes. As a result, we improve the performance by a sizable margin especially on the largescale ImageNet that faces the challenge of high diversity. Moreover, the fixed bitwise interaction cannot achieve optimal for

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2022.3161600, IEEE Transactions on Pattern Analysis and Machine Intelligence



(a) Yosemite-Notre Dame (b) Yosemite-Liberty (c) Notre Dame-Yosemite

(d) Notre Dame-Liberty (e) Liberty-Notre Dame (f) Liberty-Yosemite

Figure 8. Comparison of ROC curves on the Brown dataset with several binary descriptors, where six train-test combinations were adopted.

Table 3
Comparison of 95% error rates (FPR95) on the Brown dataset with the state-of-the-art binary descriptors, where six train-test combinations were applied. Supervised binary descriptors include LDAHash, D-BRIEF, BinBoost and RFD, binary codes obtained without label information contains BRISK, BRIEF, DeepBit, DBD-MQ, BinGAN, BGAN+ and UDBD. The real-valued feature SIFT is provided for reference.

Train Test	Yosemite Notre Dame	Yosemite Liberty	Notre Dame Yosemite	Notre Dame Liberty	Liberty Notre Dame	Liberty Yosemite	Average FPR95
SIFT (128 Byte)	28.09	36.27	29.15	36.27	28.09	29.15	31.17
LDAHash (16 Byte)	51.58	49.66	52.95	49.66	51.58	52.95	51.40
D-BRIEF(4 Byte)	43.96	53.39	46.22	51.30	43.10	47.29	47.54
BinBoost(8 Byte)	14.54	21.67	18.96	20.49	16.90	22.88	19.24
RFD (50-70 Byte)	11.68	19.40	14.50	19.35	13.23	16.99	15.86
BRISK (64 Byte)	74.88	79.36	73.21	79.36	74.88	73.21	75.81
BRIEF (32 Byte)	54.57	59.15	54.96	59.15	54.57	54.96	56.23
DeepBit (32 Byte)	29.60	34.41	63.68	32.06	26.66	57.61	40.67
DBD-MQ (32 Byte)	27.20	33.11	57.24	31.10	25.78	57.15	38.59
BinGAN (32 Byte)	16.88	26.08	40.80	25.76	27.84	47.64	30.76
BGAN+ (32 Byte)	32.14	36.11	60.24	40.51	30.26	54.64	42.12
UDBD (32 Byte)	14.61	20.79	52.60	18.99	11.76	52.17	28.49
GraphBit (32 Byte)	17.78	24.72	49.94	21.18	15.25	49.64	29.75
GraphBit+ (32 Byte)	18.43	25.27	49.96	21.32	15.63	49.17	29.96
D-GraphBit (32 Byte)	**10.63**	**15.66**	**40.07**	**14.91**	**9.56**	**41.47**	**22.05**

all input images due to the semantics variation, and the presented D-GraphBit learns the dynamic bitwise interaction via the GCN based search strategy and further enhances the performance. Since the greedy discrete optimization proposed in GreedyHash [53] reduces the information loss for hash codes, we integrate our GraphBit, GraphBit+ and D-GraphBit with GreedyHash to further strengthen the discrimination ability of the learned binary descriptors, which outperform the vanilla GreedyHash by a large margin on all three datasets.

Comparison on Brown: Table 3 illustrates the 95% error rates (FPR95) of our method and the state-of-the-art binary descriptors on the Brown dataset, and Figure 8 shows ROC curves of all six train-test combinations. The length of the binary descriptors for our methods was set as 256 in the experiments on Brown. GraphBit and GraphBit+ learn reliable binary codes with ambiguity elimination through bitwise interaction mining, achieving an average FPR95 improvement of 8.84% and 8.63% compared with DBD-MQ. D-GraphBit further enhances the performance by 7.70% (22.05% vs. 29.75%) with the optimal bitwise interaction mined dynamically. Moreover, GraphBit, GraphBit+ and D-GraphBit obtain a lower average FPR95 compared to the widely used real-valued SIFT features with much smaller storage cost. As unsupervised methods, GraphBit, GraphBit+ and D-GraphBit obtain better average performance than the supervised LDAHash and D-BRIEF, which shows their applicability in scenarios where label information is difficult to collect.

Comparison on HPatches: We followed the standard evaluation protocol [4] to report the performance of mAP on the three visual analysis tasks including patch verification,

image matching and patch retrieval. The goal of verification is to classify whether two patches are matched or not. Matching is performed by comparing patch sets between the target and reference images, and retrieval aims to find similar patches for query images. Table 4 shows the results, where GraphBit outperforms DeepBit by 3.92%, 1.17% and 4.58% on each tested visual analysis task respectively. D-GraphBit further enhances the performance by 5.77%, 5.01% and 5.33% on patch verification, image matching and patch retrieval respectively, showing the effectiveness of instruction from dynamic bitwise interaction. Moreover, D-GraphBit outperforms the supervised binary descriptor BinBoost without using any label information.

Table 4
MAP (%) of unsupervised binary codes on HPatches.

Method	Verification	Matching	Retrieval
SIFT(128 Byte)	65.12	25.47	31.98
RootSIFT (128 Byte)	58.53	27.22	33.56
BinBoost (32 Byte)	66.67	14.77	22.45
BRIEF(32 Byte)	58.07	10.50	16.03
ORB (32 Byte)	60.15	15.32	18.85
DeepBit (32 Byte)	61.27	13.05	20.61
UDBD (32 Byte)	69.77	17.27	28.88
GraphBit (32 Byte)	65.19	14.22	25.19
GraphBit+ (32 Byte)	66.01	13.37	24.32
D-GraphBit (32 Byte)	**70.96**	**19.23**	**30.52**

5.4 Deployment Efficiency and Training Cost

The storage cost and the inference latency depict the efficiency of deploying the descriptor extraction model, and the time to obtain a well-trained deep hashing model illustrates the traiing cost. In order to show the efficiency in training and deployment of our methods, we conducted experiments to evaluate the storage cost, the inference latency and the training cost on CIFAR-10 with 32-bit binary descriptors, where Table 5 shows the results. Our hardware equipped with a 2.8-GHz CPU and a 32G RAM, and we utilized a GTX 1080 Ti GPU for acceleration. We evaluated the total time of extracting one probe feature and retrieving from gallery features as the inference latency, and the storage cost for each probe feature was also investigated. Meanwhile, The training cost is defined as the whole training time of each method. Compared with the real-valued HOG features, the proposed GraphBit (GraphBit+) and D-GraphBit both save the storage cost by 75% and decrease the inference latency by 52% and latency by 49% respectively. GraphBit+ reduces the training cost of GraphBit by 65% due to the efficient differentiable search strategy for bitwise interaction mining. The increase in inference latency for D-GraphBit compared with GraphBit and GraphBit+ is acceptable since the GraphMiner is very lightweight. Since the graph convolutional networks in the GraphMiner can be optimized via gradient descent, the training cost of D-GraphBit is also sizably decreased compared with GraphBit. Our GraphBit, GraphBit+ and D-GraphBit only require negligible extra inference latency than DeepBit, while obtain much higher performance across different visual tasks.

Table 5
Comparison of the storage cost, the inference latency and the training cost accross different descriptor extraction models.

Method	Storage Cost	Inference Latency	Training Cost
HOG	16 Byte	42.2 ms	-
DeepBit	4 Byte	20.2 ms	1.88h
GreedyHash	4 Byte	24.7 ms	2.52h
GraphBit	4 Byte	20.4 ms	5.73h
GraphBit+	4 Byte	20.4 ms	2.02h
D-GraphBit	4 Byte	21.6 ms	2.05h

GraphBit, GraphBit+ and D-GraphBit achieve different trade-offs among accuracy, training cost and inference latency. D-GraphBit outperforms others with respect to accuracy due to the dynamic bitwise interactions, and slightly increase the inference latency resulted from the lightweight GraphMiner. GraphBit+ significantly reduces the training cost with slight accuracy degradation compared with GraphBit. Therefore, users can choose the optimal one for unsupervised binary descriptor learning based on the accuracy requirement, training cost budget and the hardware configurations for deployment.

6 Conclusion

In this paper, we have proposed an unsupervised deep binary descriptor learning method called GraphBit for compact image representation. Our GraphBit models binary codes in binomial distributions and maximizes the mutual information with the observed inputs and the related bits to reduce the uncertainty. Moreover, GraphBit mines the bitwise interaction through deep reinforcement learning to enhance the reliability of the ambiguous bits. To reduce the heavy training cost caused by reinforcement learning, we have further presented GraphBit+ that leverages differentiable search strategy for bitwise interaction mining. We have also proposed D-GraphBit that learns dynamic bitwise interaction for each instance via the GCN based GraphMiner, so that the dynamic bitwise interaction provides optimal instruction to eliminate the binary descriptor ambiguity for each input. Extensive experimental results have demonstrated the effectiveness and efficiency of the proposed method, and the mined bitwise interaction in GraphBit, GraphBit+ and D-GraphBit can also be integrated with other unsupervised binary descriptors to further enhance the vanilla model.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant 62125603, Grant 61822603, Grant U1813218, and Grant U1713214, in part by a grant from the Beijing Academy of Artificial Intelligence (BAAI), and in part by a grant from the Institute for Guo Qiang, Tsinghua University.

References

[1] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. FREAK: Fast retina keypoint. In *CVPR*, pages 510–517, 2012.
[2] Alexandr Andoni and Piotr Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, pages 459–468, 2006.
[3] Relja Arandjelovic and Andrew Zisserman. Three things everyone should know to improve object retrieval. In *CVPR*, pages 2911–2918, 2012.

[4] Vassileios Balntas, Karel Lenc, Andrea Vedaldi, and Krystian Mikolajczyk. HPatches: A benchmark and evaluation of handcrafted and learned local descriptors. In *CVPR*, pages 5173–5182, 2017.

[5] David Barber and Felix V Agakov. The IM algorithm: A variational approach to information maximization. In *NIPS*, pages 201–208, 2003.

[6] Aritra Bhowmik, Stefan Gumhold, Carsten Rother, and Eric Brachmann. Reinforced feature points: Optimizing feature detection and description for a high-level task. In *CVPR*, pages 4948–4957, 2020.

[7] Eric Brachmann and Carsten Rother. Neural-guided ransac: Learning where to sample model hypotheses. In *ICCV*, pages 4322–4331, 2019.

[8] Matthew Brown, Gang Hua, and Simon Winder. Discriminative learning of local image descriptors. *TPAMI*, 33(1):43–57, 2011.

[9] Zhaowei Cai and Nuno Vasconcelos. Rethinking differentiable search for mixed-precision neural networks. In *CVPR*, pages 2349–2358, 2020.

[10] Micheal Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. BRIEF: Binary robust independent elementary features. In *ECCV*, pages 778–792, 2010.

[11] Zhangjie Cao, Mingsheng Long, Jianmin Wang, and Philip S Yu. Hashnet: Deep learning to hash by continuation. In *ICCV*, pages 5608–5617, 2017.

[12] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, pages 132–149, 2018.

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, pages 1597–1607, 2020.

[14] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, pages 433–442, 2019.

[15] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. Nus-wide: a real-world web image database from national university of singapore. In *ICMR*, page 48, 2009.

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[17] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, pages 1422–1430, 2015.

[18] Yueqi Duan, Jiwen Lu, Ziwei Wang, Jianjiang Feng, and Jie Zhou. Learning deep binary descriptor with multi-quantization. In *CVPR*, pages 1183–1192, 2017.

[19] Yueqi Duan, Ziwei Wang, Jiwen Lu, Xudong Lin, and Jie Zhou. Graphbit: Bitwise interaction mining via deep reinforcement learning. In *CVPR*, pages 8270–8279, 2018.

[20] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep hashing for compact binary codes learning. In *CVPR*, pages 2475–2483, 2015.

[21] Bin Fan, Qingqun Kong, Tomasz Trzcinski, Zhiheng Wang, Chunhong Pan, and Pascal Fua. Receptive fields selection for binary feature description. *TIP*, 23(6):2583–2595, 2014.

[22] Kamran Ghasedi Dizaji, Feng Zheng, Najmeh Sadoughi, Yanhua Yang, Cheng Deng, and Heng Huang. Unsupervised deep generative adversarial hashing network. In *CVPR*, pages 3664–3673, 2018.

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

[24] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[25] Kristen Grauman and Trevor Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, pages 1458–1465, 2005.

[26] Yushuo Guan, Pengyu Zhao, Bingxuan Wang, Yuanxing Zhang, Cong Yao, Kaigui Bian, and Jian Tang. Differentiable feature aggregation search for knowledge distillation. In *ECCV*, pages 469–484, 2020.

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020.

[28] Chen Huang, Simon Lucey, and Deva Ramanan. Learning policies for adaptive tracking with deep feature cascades. In *ICCV*, pages 105–114, 2017.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[30] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[31] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Master Thesis*, 2009.

[32] Stefan Leutenegger, Margarita Chli, and Roland Siegwart. BRISK: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011.

[33] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. In *NIPS*, pages 539–548, 2018.

[34] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *CVPR*, pages 1183–1192, 2016.

[35] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[36] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. Deep supervised hashing for fast image retrieval. In *CVPR*, pages 2064–2072, 2016.

[37] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[38] Jiwen Lu, Venice Erin Liong, Xiuzhuang Zhou, and Jie Zhou. Learning compact binary face descriptor for face recognition. *TPAMI*, 37(10):2041–2056, 2015.

[39] Denis Mazur, Vage Egiazarian, Stanislav Morozov, and Artem Babenko. Beyond vector spaces: Compact data representation as differentiable weighted graphs. In *NIPS*, pages 6906–6916, 2019.

[40] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, *et al.* Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533, 2015.

[41] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84, 2016.

[42] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. In *BMVC*, pages 1–12, 2015.

[43] Aleksis Pirinen and Cristian Sminchisescu. Deep reinforcement learning of region proposal networks for object detection. In *CVPR*, pages 6945–6954, 2018.

[44] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. ORB: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011.

[45] Ruslan Salakhutdinov and Geoffrey Hinton. Semantic hashing. *IJAR*, 50(7):969–978, 2009.

[46] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *CVPR*, pages 4938–4947, 2020.

[47] Fumin Shen, Yan Xu, Li Liu, Yang Yang, Zi Huang, and Heng Tao Shen. Unsupervised deep hashing with similarity-adaptive and discrete optimization. *TPAMI*, 40(12):3034–3044, 2018.

[48] Yuming Shen, Li Liu, and Ling Shao. Unsupervised binary representation learning with deep variational networks. *IJCV*, 127 (11-12):1614–1628, 2019.

[49] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with directed graph neural networks. In *CVPR*, pages 7912–7921, 2019.

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, pages 1–14.

[51] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, Alan Hanjalic, and Heng Tao Shen. Unified binary generative adversarial network for image retrieval and compression. *IJCV*, pages 1–22, 2020.

[52] Christoph Strecha, Alex Bronstein, Michael Bronstein, and Pascal Fua. LDAHash: Improved matching with smaller descriptors. *TPAMI*, 34(1):66–78, 2012.

[53] Shupeng Su, Chao Zhang, Kai Han, and Yonghong Tian. Greedy hash: Towards fast optimization for accurate hash coding in cnn. In *NIPS*, pages 798–807, 2018.

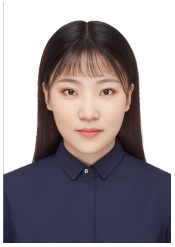
[54] Prune Truong, Stefanos Apostolopoulos, Agata Mosinska, Samuel Stucky, Carlos Ciller, and Sandro De Zanet. Glampoints: Greedily learned accurate match points. In *ICCV*, pages 10732–10741, 2019.

[55] Tomasz Trzcinski and Vincent Lepetit. Efficient discriminative

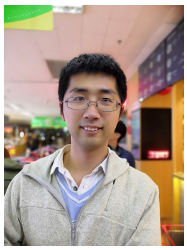
projections for compact binary descriptors. In *ECCV*, pages 228–242, 2012.
[56] Tomasz Trzcinski, Mario Christoudias, Pascal Fua, and Vincent Lepetit. Boosting binary keypoint descriptors. In *CVPR*, pages 2874–2881, 2013.
[57] Michał J Tyszkiewicz, Pascal Fua, and Eduard Trulls. Disk: Learning local features with policy gradient. *arXiv preprint arXiv:2006.13566*, 2020.
[58] Lei Wang, Yuchun Huang, Yaolin Hou, Shenman Zhang, and Jie Shan. Graph attention convolution for point cloud semantic segmentation. In *CVPR*, pages 10296–10305, 2019.
[59] Ziwei Wang, Yunsong Wang, Ziyi Wu, Jiwen Lu, and Jie Zhou. Instance similarity learning for unsupervised feature representation. *arXiv preprint arXiv:2108.02721*, 2021.
[60] Ronald J Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, pages 229–256, 1992.
[61] Gengshen Wu, Zijia Lin, Guiguang Ding, Qiang Ni, and Jungong Han. On aggregation of unsupervised deep binary descriptor with weak bits. *TIP*, 29:9266–9278, 2020.
[62] Yiming Wu, Omar El Farouk Bourahla, Xi Li, Fei Wu, Qi Tian, and Xue Zhou. Adaptive graph representation learning for video person re-identification. *TIP*, 2020.
[63] Jian Zhang and Yuxin Peng. Ssdh: semi-supervised deep hashing for large scale image retrieval. *TCSVT*, 29(1):212–225, 2017.
[64] Maciej Zieba, Piotr Semberecki, Tarek El-Gaaly, and Tomasz Trzcinski. Bingan: Learning compact binary descriptors with a regularized gan. In *NIPS*, pages 3608–3618, 2018.
[65] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

Jie Zhou (SM'04) received the BS and MS degrees both from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From then to 1997, he served as a postdoctoral fellow in the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a full professor in the Department of Automation, Tsinghua University. His research interests include computer vision, pattern recognition, and image processing. In recent years, he has authored more than 300 papers in peer-reviewed journals and conferences. Among them, more than 100 papers have been published in top journals and conferences such as the IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Image Processing, and CVPR. He is an associate editor for the IEEE Transactions on Pattern Analysis and Machine Intelligence and two other journals. He received the National Outstanding Youth Foundation of China Award. He is an IAPR Fellow.

Ziwei Wang received the B.S. degree from the Department of Physics, Tsinghua University, China, in 2018. He is currently working toward the PhD degree in the Department of Automation, Tsinghua University, China. His research interests include network compression and binary representation. He has published 10 scientific papers in TPAMI, CVPR, ICCV and ECCV. He serves as a regular reviewer member for TIP, CVPR, NeurIPS, ICML, ICLR, WACV and ICME.



Han Xiao is currently a first-year master student in the Department of Automation, Tsinghua University, China. Her research interests include computer vision, efficient inference and model compression. Previously, she got her bachelor's degree from the Department of Automation at Tsinghua University.



Yueqi Duan received the B.S. and Ph.D. degrees both in the Department of Automation, Tsinghua University, China, in 2014 and 2019, respectively. He is currently an assistant professor at the Department of Electrical Engineering, Tsinghua University. His research interests include 3D vision and binary representation. He has authored 16 scientific papers in TPAMI, TIP, CVPR and ECCV. He serves as an area chair of ICME 2020-2022, and as a reviewer member for TPAMI, IJCV, CVPR, ICCV, ECCV and NeurIPS.

Jiwen Lu (M'11-SM'15) received the B.Eng. degree in mechanical engineering and the M.Eng. degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, in 2003 and 2006, respectively, and the Ph.D. degree in electrical engineering from Nanyang Technological University, Singapore, in 2012. He is currently an Associate Professor with the Department of Automation, Tsinghua University, Beijing, China. His current research interests include computer vision and pattern recognition. He was/is a member of the Image, Video and Multidimensional Signal Processing Technical Committee, Multimedia Signal Processing Technical Committee and the Information Forensics and Security Technical Committee of the IEEE Signal Processing Society, a member of the Multimedia Systems and Applications Technical Committee and the Visual Signal Processing and Communications Technical Committee of the IEEE Circuits and Systems Society, respectively. He serves as the General Co-Chair for the International Conference on Multimedia and Expo (ICME) 2022, the Program Co-Chair for the International Conference on Multimedia and Expo 2020, the International Conference on Automatic Face and Gesture Recognition (FG) 2023, and the International Conference on Visual Communication and Image Processing (VCIP) 2022. He serves as the Co-Editor-of-Chief for Pattern Recognition Letters, an Associate Editor for the IEEE Transactions on Image Processing, the IEEE Transactions on Circuits and Systems for Video Technology, the IEEE Transactions on Biometrics, Behavior, and Identity Sciences, and Pattern Recognition. He was a recipient of the National Natural Science Funds for Distinguished Young Scholar. He is a Fellow of IAPR.