Bit-aware Semantic Transformer Hashing for Multi-modal Retrieval

Wentao Tan
Shandong Normal University
Jinan, China
tan.wt.lucky@gmail.com

Lei Zhu*
Shandong Normal University
Jinan, China
leizhu0608@gmail.com

Weili Guan
Monash University
Clayton, Australia
weili.guan@monash.edu

Jingjing Li
University of Electronic Science and Technology of China
Chengdu, China
lijin117@yeah.net

Zhiyong Cheng
Shandong Artificial Intelligence Institute
Jinan, China
jason.zy.cheng@gmail.com

ABSTRACT

Multi-modal hashing learns binary hash codes with extremely low storage cost and high retrieval speed. It can support efficient multi-modal retrieval well. However, most existing methods still suffer from three important problems: 1) Limited semantic representation capability with shallow learning. 2) Mandatory feature-level multi-modal fusion ignores heterogeneous multi-modal semantic gaps. 3) Direct coarse pairwise semantic preserving cannot effectively capture the fine-grained semantic correlations. For solving these problems, in this paper, we propose a *Bit-aware Semantic Transformer Hashing* (BSTH) framework to excavate bit-wise semantic concepts and simultaneously align the heterogeneous modalities for multi-modal hash learning on the concept-level. Specifically, the bit-wise implicit semantic concepts are learned with the transformer in a self-attention manner, which can achieve implicit semantic alignment on the fine-grained concept-level and reduce the heterogeneous modality gaps. Then, the concept-level multi-modal fusion is performed to enhance the semantic representation capability of each implicit concept and the fused concept representations are further encoded to the corresponding hash bits via bit-wise hash functions. Further, to supervise the bit-aware transformer module, a label prototype learning module is developed to learn prototype embeddings for all categories that capture the explicit semantic correlations on the category-level by considering the co-occurrence priors. Experiments on three widely tested multi-modal retrieval datasets demonstrate the superiority of the proposed method from various aspects.

CCS CONCEPTS

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

*Lei Zhu (leizhu0608@gmail.com) is the corresponding author.

SIGIR '22, July 11–15, 2022, Madrid, Spain

ACM ISBN 978-1-4503-8732-3/22/07…$15.00
https://doi.org/10.1145/3477495.3531947

KEYWORDS

Concept-aware, Transformer, Multi-modal Retrieval, Hashing Technology, Fine-grained Semantic

ACM Reference Format:
Wentao Tan, Lei Zhu, Weili Guan, Jingjing Li, and Zhiyong Cheng. 2022. Bit-aware Semantic Transformer Hashing for Multi-modal Retrieval. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '22), July 11–15, 2022, Madrid, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3477495.3531947

1 INTRODUCTION

The massive multimedia data require more accurate and efficient multi-modal retrieval frameworks, which can support large-scale retrieval scenarios. In recent years, hashing-based methods have attracted increasing attention compared with real value-based retrieval systems [34, 35, 40]. They have low storage cost by binary hash codes and high retrieval speed by XOR operator in Hamming space. Generally, the hashing retrieval system can handle three scenarios, i.e. uni-modal search, cross-modal search and multi-modal search. The hashing-based uni-modal methods [25, 29, 39] only learn the binary hash codes of the original samples based on single-view features. The cross-modal retrieval [9, 12, 17, 43] aims to alleviate heterogeneous semantic gaps among different modalities and search relevant samples in other modalities under a given modality-specific query. Different from the above tasks, the multi-modal retrieval methods need to pay more attention to how to collaboratively fuse heterogeneous multi-modal data and perform the retrieval among different samples based on fused features [22]. Recently, several hashing-based methods [18, 22, 27, 37, 45] are proposed to handle the multi-modal retrieval task. Although these methods have achieved impressive success, there are three important bottlenecks that have not been solved well yet as follows:

Limited semantic representation capability with shallow learning. Multi-modal hashing methods can be roughly divided into shallow and deep-based ones based on the learning framework. As shown in Table 1, most existing multi-modal hashing methods are shallow architectures. These shallow methods [18, 27] generally optimize hash codes and functions based on shallow learning models, so they can not fully capture the complex semantic information in multi-modal data [45]. Benefiting from the development of deep

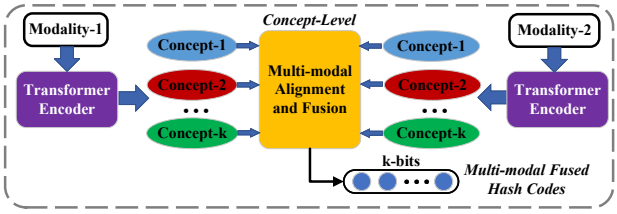

Figure 1: The key idea of our proposed bit-aware semantic transformer hashing, which can align the heterogeneous modalities by the learned bit-wise implicit semantic concepts and perform multi-modal fusion on the fine-grained concept-level for multi-modal hash learning. For the limited space here, we take two modalities for illustration.

neural networks, some methods [22, 45] introduce pre-trained networks as the backbone structures to first extract the representation features of the original multi-modal data, and further design specific strategies to fuse the extracted features and generate the binary codes. However, these deep-based methods ignore the modeling of the latent fine-grained semantic information and cannot fully exploit the powerful potential capability of deep neural networks.

Mandatory feature-level multi-modal fusion ignores heterogeneous multi-modal semantic gaps. As shown in Table 1, all the existing methods perform mandatory multi-modal fusion on the coarse-grained feature-level without considering the fine-grained implicit semantic correlations among different modalities, which cannot fully bridge heterogeneous multi-modal semantic gaps. Principally, different modalities can describe a sample from the multiple perspectives and they share fine-grained implicit semantic concepts among the multi-modal representations. These heterogeneous modalities can be potentially aligned based on these shared implicit semantic concepts, which can further assist the multi-modal fusion on the fine-grained concept-level. In addition, these implicit semantic concepts can be encoded to the hash bits, which can effectively enhance the discrimination of hash codes.

Direct coarse pairwise semantic preserving cannot effectively capture the fine-grained semantic correlations. Most existing multi-modal hashing methods directly use the annotated label information to supervise the hash code learning process by the well-designed pairwise semantic preserving objective functions, but they fail to consider fine-grained explicit semantic correlations among different samples. In the real-world scenario, multiple different objects often simultaneously appear in a picture, a sentence, or a piece of audio. The different-frequent co-occurrences of these objects indicate the semantic relevance among them. These co-occurrence priors should contribute to exploiting the explicit semantic correlations on the category-level, and further to capturing the fine-grained semantic correlations among different samples.

Motivated by the above analyses, in this paper, we propose a *Bit-aware Semantic Transformer Hashing* (BSTH) framework to excavate bit-wise semantic concepts and simultaneously align the heterogeneous modalities for multi-modal hash learning on the concept-level. Specifically: 1) The whole proposed framework is performed under the deep neural network architecture, which can capture more intrinsic multi-modal semantics. 2) To better reduce the heterogeneous multi-modal semantic gaps, we propose a bit-aware semantic transformer module to align and fuse multi-modal data on the fine-grained concept-level. Concretely, we first introduce a transformer encoder [33] to excavate bit-wise implicit semantic concepts in a self-attention manner, which can achieve implicit semantic alignment among heterogeneous modalities. Then we perform multi-modal fusion on the fine-grained concept-level to enhance the semantic representation capability of each implicit concept. Finally, we adopt bit-wise hash functions to encode the fused semantic concepts to the corresponding hash bits. 3) To supervise the bit-aware transformer module, we propose a label prototype learning module to model the explicit semantic correlations on the category-level by considering the co-occurrence priors. This module learns prototype embeddings for all categories and utilizes them to generate the supervising hash codes, which can preserve the fine-grained semantic correlations among different samples. These supervising hash codes are used to guide the learning process of the bit-aware semantic transformer module. The main contributions of our method can be summarized as follows:

- We propose a *Bit-aware Semantic Transformer Hashing* (BSTH) framework to excavate bit-wise semantic concepts and simultaneously align the heterogeneous modalities for multi-modal hash learning on the concept-level. Specifically, we introduce the transformer encoder to extract bit-wise implicit semantic concepts in a self-attention manner and achieve implicit semantic alignment among different modalities on the fine-grained concept-level. Then we perform the concept-level multi-modal fusion to enhance the semantic representation capability of each implicit concept and design bit-wise hash functions to encode these fused concept representations to the corresponding hash bits.

Table 1: The main differences between our proposed method and all the compared methods. "Model" represents the type of model architecture. "Fusion Level" shows the multi-modal fusion level with two typical branches, i.e. feature-level and concept-level. $*$ represents the corresponding method adopts a bit-by-bit optimization strategy but still fuses the multi-modal data on the feature-level. "Learning" indicates whether to use labels for supervision.

Methods	Model	Fusion Level	Learning
MFH [30]	shallow	feature-level	unsupervised
MAH [15]	shallow	feature-level	unsupervised
MVLH [27]	shallow	feature-level $*$	unsupervised
MvDH [26]	shallow	feature-level $*$	unsupervised
MFKH [18]	shallow	feature-level	supervised
DMVH [37]	shallow	feature-level	supervised
SDMH [21]	shallow	feature-level	supervised
FOMH [20]	shallow	feature-level	supervised
FDMH [16]	shallow	feature-level	supervised
DCMVH [45]	deep	feature-level	supervised
SAPMH [41]	shallow	feature-level	supervised
FGCMH [22]	deep	feature-level	supervised
BSTH (ours)	deep	concept-level	supervised

- To supervise the bit-aware transformer module, a label prototype learning module is developed to learn prototype embeddings for all categories that capture the explicit semantic correlations on the category-level by considering the co-occurrence priors. This module generates the supervising hash codes by the learned label prototype embeddings, which can better guide the learning process of the bit-aware semantic transformer module.
- Experiments on three widely tested multi-modal retrieval datasets demonstrate the superiority of the proposed method from various aspects. Particularly, in the large-scale scenario, our method outperforms the second best methods on the mAP metric by 6.81% and 7.92% on NUS-WIDE and MS-COCO, with 128 bits, respectively.

2 RELATED WORK

Multi-modal hashing. Compared with uni-modal hashing, the multi-modal hashing methods [15, 26, 27, 30, 36, 37, 42, 45] generate binary representations for the efficient multimedia retrieval system by capturing the correlations among different modalities and performing multi-modal fusion. Generally, multi-modal hashing methods can be divided into supervised and unsupervised ones. Unsupervised methods directly models multi-modal correlations to learn hash codes without any supervised semantic information. For example, Multiple Feature Hashing (MFH) [30] exploits the local structure of each modality and considers these structures together to learn multi-modal fused hash codes for near-duplicate video retrieval. Multiview Alignment Hashing (MAH) [15] introduces kernelized nonnegative matrix factorization to optimize the hash functions and learns the binary codes by exploiting the latent semantics in different views and preserving joint probability distribution. Multi-view Latent Hashing (MVLH) [27] learns the hash codes by the shared latent factors and designs an adaptive weighting strategy to fuse all views. Multiview Discrete Hashing (MvDH) [26] ensures the consistence between the hash codes learning by matrix factorization and the cluster labels with spectral clustering to enhance the discrimination of the hash codes.

Different from the unsupervised multi-modal hashing methods, several supervised methods [16, 22, 37, 41, 45] have been proposed to generate more discriminative hash codes by exploiting explicit semantic labels. Compact Kernel Hashing with Multiple Features (MFKH) [18] formulates the multi-view feature learning framework as a similarity preserving hashing problem with optimal linearly-combined multiple kernels. Discrete Multi-view Hashing (DMVH) [37] exploits Locally Linear Embedding (LLE) [24] to construct the affinity matrix, which can preserve local similarity structure and the semantic similarity between samples. Supervised Discrete Multi-view Hashing (SDMH) [21] learns the shared hash codes by exploiting the complementary features among different views and removing the redundant information. Flexible Online Multi-modal Hashing (FOMH) [20] proposes an online multi-modal hashing framework, which can adaptively fuse heterogeneous multi-modal data in a self-weighted manner and learn the discriminative hash codes with an asymmetric supervision strategy. Flexible Discrete Multi-view Hashing (FDMH) [16] develops an adaptive dictionary learning strategy, which combines the shared collective latent embeddings of multi-view data and semantic information to learn the discriminative hash codes. Supervised Adaptive Partial Multi-view Hashing (SAPMH) [41] introduces a parameter-free learning strategy to adjust the fusion weights of multi-modal features under the partial multi-view scenario.

Recently, several deep multi-modal hashing methods [22, 36, 45] are proposed to generate multi-modal fused hash codes with deep learning. Deep Collaborative Multi-view Hashing (DCMVH) [45] first proposes a deep multi-modal hashing method to perform multi-view feature fusion and learn the hash codes with the supervision of pairwise semantic matrix. Flexible Graph Convolutional Multi-modal Hashing (FGCMH) [22] introduces multiple Graph Convolutional Networks (GCNs) [11] to exploit the intra-modality and the fusion-modality structural similarity, and capture the complementary correlations among the different views when learning the hash codes.

The main differences between our proposed method and existing multi-modal hashing methods are summarized in Table 1. All existing multi-modal hashing methods directly perform multi-modal fusion on the feature-level, but our method can capture the implicit bit-wise semantic concepts to align the heterogeneous multiple modalities and fuse them on the fine-grained concept-level. MVLH and MvDH are marked with $*$, which indicates they adopt a time-consuming bit-by-bit optimization strategy but still fuse the multi-modal data on the feature-level. In addition, these supervised multi-modal hashing baselines simply preserve the pairwise semantic information into hash codes without considering the fine-grained semantic correlations of different categories.

Transformer. In recent years, transformer architectures achieve great success in computer vision [1, 5, 19], natural language processing [4, 23, 38] and pre-trained models [6, 31, 32]. The transformer can model sequence relationships better than Convolutional Neural Networks (CNNs) [28] and Recurrent Neural Networks (RNNs) [7]. Compared with GCNs [11], it can adaptively capture global relationships of input sequences rather than explicitly constructing affinity graphs. There are several attempts to explore the transformer in hashing. TransHash [2] develops a siamese transformer backbone based on Vision Transformer (ViT) [5] to extract the visual feature representations and performs hashing-based image retrieval. It is the first attempt to handle the deep hash learning problem without using CNNs. Bidirectional Transformers Hashing (BTH) [13] designs a video hashing model based on the bidirectional transformers to capture correlations among video frames in a self-supervised manner. These attempts consider the transformer as a feature extractor to replace CNNs. Different from them, in this paper, we exploit the transformer encoder to perform the downstream multi-modal hash learning task, which excavates bit-wise implicit semantic concepts and aligns the heterogeneous modalities to better fuse multi-modal data.

3 NOTATIONS AND PROBLEM DEFINITION

Notations. In this paper, we utilize boldface uppercase letters, e.g., $\mathbf{Y}$, and boldface lowercase letters, e.g., $\boldsymbol{y}$, to denote matrices and vectors, respectively. $||\cdot||_2$ denotes the L2-norm and $||\cdot||_F$ denotes the Frobenius norm. For the convenience of description, we present

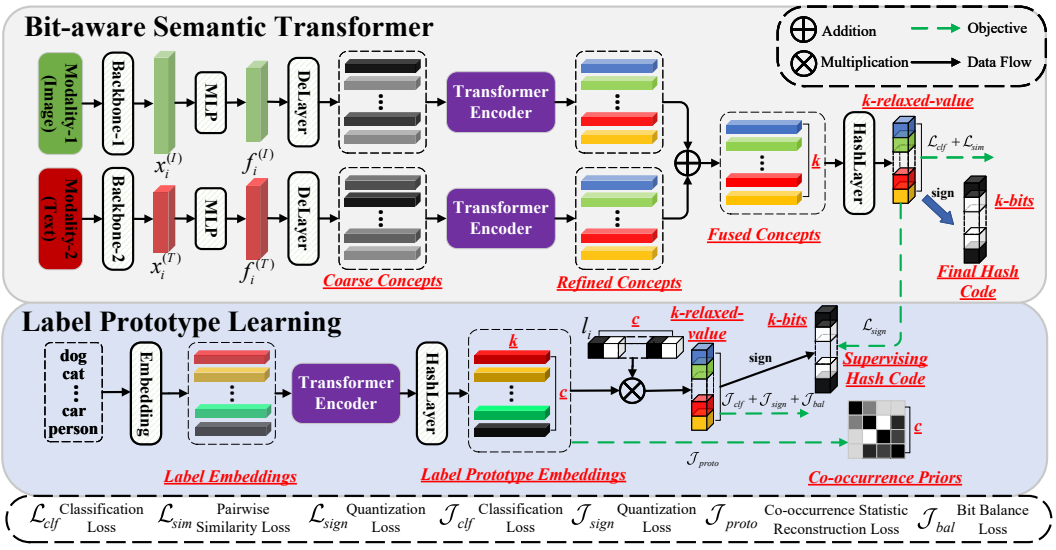

Figure 2: The framework of our proposed BSTH. BSTH mainly contains the Bit-aware Semantic Transformer module and the Label Prototype Learning module. In the Bit-aware Semantic Transformer module, the modality-specific feature representation is first decomposed to the coarse implicit semantic concepts by the feature decoupling layer, and then, the transformer encoder refines them in a self-attention manner. Finally, the multi-modal fusion is performed on the fine-grained concept-level and these fused concept representations are independently coded to the corresponding hash bits by bit-wise hash functions. The label prototype learning module learns the prototype embeddings for all categories by considering the co-occurrence priors and these embeddings are further used to generate the supervising hash codes to guide the learning process of the bit-aware semantic transformer module.

the subsequent explanations under the bi-modal scenario, i.e. image and text. As shown in the Figure 2, the modality-specific branches in the bit-aware semantic transformer module are structurally same, except for the corresponding feature extractors. Therefore, our proposed framework can be easily extended to three or more modalities cases.

Suppose that the training dataset $\mathbf{O}_{tr} = \{\boldsymbol{o}_i\}_{i=1}^{n}$ consists of n samples with image and text modalities. We utilize the modality-specific feature extractors, e.g., VGGNet [28] and Bag-of-Words (BoW), to extract the feature representations for original visual and textual data, denoted as $\mathbf{X}^{(I)} = [\boldsymbol{x}_1^{(I)}, \boldsymbol{x}_2^{(I)}, ..., \boldsymbol{x}_n^{(I)}] \in \mathbb{R}^{n \times d^{(I)}}$ and $\mathbf{X}^{(T)} = [\boldsymbol{x}_1^{(T)}, \boldsymbol{x}_2^{(T)}, ..., \boldsymbol{x}_n^{(T)}] \in \mathbb{R}^{n \times d^{(T)}}$, where $d^{(I)}$ and $d^{(T)}$ indicate the corresponding dimensionalities of the extracted visual feature representations and textual feature representations, respectively. $\mathbf{L} = [\boldsymbol{l}_1, \boldsymbol{l}_2, ..., \boldsymbol{l}_n] \in \{0, 1\}^{n \times c}$ denotes the label matrix of the training data, where $l_{ij} = 1$ indicates the i-th sample belongs to the j-th category and vice versa, c is the number of the categories. $\mathbf{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_n] \in \{-1, 1\}^{n \times k}$ represents the k-bits hash code matrix.

Problem Definition. When multi-modal samples arrive, the hashing-based multi-modal retrieval system needs to integrate multi-modal heterogeneous information and generate joint feature representations. Then these representations are quantified to hash codes to support efficient search via Hamming distance.

4 METHODOLOGY

The overview of our proposed Bit-aware Semantic Transformer Hashing (BSTH) framework is shown in Figure 2, which is mainly composed of a bit-aware semantic transformer module and a label prototype learning module. The bit-aware semantic transformer module learns the implicit semantic concept corresponding to each hash bit in a self-attention manner and performs the multi-modal fusion on the fine-grained concept-level. These fused concept representations are encoded to the corresponding hash bits via bit-wise hash functions. The label prototype learning module learns the prototype embeddings for all categories by considering the co-occurrence priors, which can capture the explicit semantic correlations on the category-level. This module generates the supervising hash codes by linear combinations of these embeddings, which can guide the learning process of the bit-aware semantic transformer module. In the next sections, we will further introduce the motivation and details of these two modules.

4.1 Bit-aware Semantic Transformer

Motivation and Discussion. Generally, existing multi-modal hashing methods [22, 27, 45] adopt mandatory coarse-grained feature-level multi-modal fusion, e.g., addition and concatenation fusion strategies, which is hard to bridge heterogeneous semantic gaps and capture fine-grained semantic correlations among different modalities. Principally, each hash bit can represent a corresponding implicit semantic concept. Hence, we can determine bit-wise

implicit semantic concepts as a bridge to perform the fine-grained semantic alignment and concept-level fusion among multiple modalities, which can simultaneously reduce the multi-modal semantic gaps and enhance concept representation capability.

Based on the above discussion, we first design a feature decoupling layer to decompose the modality-specific feature representations to the corresponding coarse implicit semantic concepts. The transformer encoder is further introduced to adaptively capture the potential correlations among these implicit semantic concepts and refine them in a self-attention manner. These refined implicit semantic concepts can align multiple modalities and reduce the heterogeneous multi-modal semantic gaps. Then we perform multi-modal fusion on the fine-grained concept-level to generate more representative semantic concepts. Finally, these fused semantic concepts are independently encoded to the corresponding hash bits by bit-wise hash functions, which can generate the more discriminative hash codes.

Details. As shown in Figure 2, the extracted modality-specific feature representations $\boldsymbol{x}_i^{(*)}$ are mapped to the feature representations $\boldsymbol{f}_i^{(*)}$ with the same dimensionality via the corresponding Multi-Layer Perceptron (MLP) architectures $\mathbf{MLP}^{(*)}(\cdot;\theta_{mlp}^{(*)})$ as follows:

$$\boldsymbol{f}_i^{(*)} = \mathbf{MLP}^{(*)}(\boldsymbol{x}_i^{(*)};\theta_{mlp}^{(*)}), s.t. * \in \{I,T\}, \tag{1}$$

where $\theta_{mlp}^{(*)}$ is the trainable parameters. Then, we design feature decoupling layers $\mathbf{DeLayer}^{(*)}(\cdot;\theta_{de}^{(*)})$ to decompose the projected feature representations $\boldsymbol{f}_i^{(*)} \in \mathbb{R}^{1\times d_c}$ into a sequence $\mathbf{C}_i^{(*)}$ with k coarse implicit semantic concepts:

$$\begin{aligned}\mathbf{C}_i^{(*)} &= [\boldsymbol{c}_{i1}^{(*)}, \boldsymbol{c}_{i2}^{(*)}, ..., \boldsymbol{c}_{ik}^{(*)}] \\ &= \mathbf{DeLayer}^{(*)}(\boldsymbol{f}_i^{(*)};\theta_{de}^{(*)}),\end{aligned} \tag{2}$$

where $\boldsymbol{c}_{ik}^{(*)} \in \mathbb{R}^{1\times d_c}$ represents the k-th coarse implicit semantic concept representation of the i-th sample with the specific modality, d_c denotes the dimensionality of the concept representation, $\theta_{de}^{(*)}$ is the trainable parameters. Specifically, we implement this layer by a linear projection layer to map the $\boldsymbol{f}_i^{(*)}$ to $\tilde{\boldsymbol{f}}_i^{(*)} \in \mathbb{R}^{1\times(k\times d_c)}$, and a reshape operator to convert the $\tilde{\boldsymbol{f}}_i^{(*)}$ to $\mathbf{C}_i^{(*)} \in \mathbb{R}^{k\times d_c}$, denoting k coarse implicit semantic concept representations. Intuitively, these concepts are decoupled from an individual modality-specific feature representation $\boldsymbol{f}_i^{(*)}$, so they can collaboratively represent the original i-th sample with the specific modality.

We further introduce the modality-specific transformer encoder to capture the potential correlations among these k coarse implicit semantic concepts in a self-attention manner and generate the refined implicit semantic concepts. It can be formulated as follows:

$$\tilde{\mathbf{C}}_i^{(*)} = \mathbf{TransformerEncoder}^{(*)}(\mathbf{C}_i^{(*)};\theta_{enc}^{(*)}), \tag{3}$$

where $\tilde{\mathbf{C}}_i^{(*)} \in \mathbb{R}^{k\times d_c}$ represents the refined semantic concept sequence of i-th sample and $\theta_{enc}^{(*)}$ is the trainable parameters. The transformer encoder is composed of three submodules: positional encoding submodule, multi-head self-attention submodule, and feed-forward network. The details of these submodules can refer to [33]. These implicit semantic concepts can align the heterogeneous modalities to reduce the semantic gaps. Then we can better perform multi-modal fusion on the fine-grained concept-level as follows:

$$\tilde{\mathbf{C}}_i^{f} = \sum_{*\in\{I,T\}} \tilde{\mathbf{C}}_i^{(*)}, \tilde{\mathbf{C}}_i^{f} \in \mathbb{R}^{k\times d_c}, \tag{4}$$

$$\tilde{\mathbf{C}}_i^{f} = [\tilde{\boldsymbol{c}}_{i1}^{f}, \tilde{\boldsymbol{c}}_{i2}^{f}, ..., \tilde{\boldsymbol{c}}_{ik}^{f}], \tag{5}$$

where $\tilde{\mathbf{C}}_i^{f}$ represents the fused implicit semantic concept sequence of i-th sample, denoted in Eq.(5). Finally, we adopt bit-wise hash functions to encode these fused semantic concept representations into the corresponding hash bits as follows:

$$\boldsymbol{h}_i = \mathbf{Concat}(\mathcal{H}_1(\tilde{\boldsymbol{c}}_{i1}^{f};\theta_{h1}), \mathcal{H}_2(\tilde{\boldsymbol{c}}_{i2}^{f};\theta_{h2}), ..., \mathcal{H}_k(\tilde{\boldsymbol{c}}_{ik}^{f};\theta_{hk})), \tag{6}$$

$$\boldsymbol{b}_i = \mathrm{sign}(\boldsymbol{h}_i),\ \boldsymbol{h}_i \in \mathbb{R}^{1\times k},\ \boldsymbol{b}_i \in \{-1,1\}^{1\times k}, \tag{7}$$

where $\mathcal{H}_k(\cdot;\theta_{hk})$ with the trainable parameters θ_{hk} represents the bit-wise hash function corresponding to the k-th hash bit, which is implemented by linear projections with the activation function $\tanh(\cdot)$. $\mathcal{H}_k(\cdot;\theta_{hk})$ projects k-th fused implicit semantic concept representation $\tilde{\boldsymbol{c}}_{ik}^{f} \in \mathbb{R}^{1\times d_c}$ into a relaxed value of the corresponding hash bit. Then we concatenate these k relaxed values into $\boldsymbol{h}_i$ and perform an element-wise sign function, i.e. $\mathrm{sign}(\cdot)$, to generate the final hash code $\boldsymbol{b}_i$. The bit-wise encoding process can effectively enhance the independence of each hash bit.

For preserving the label semantic information, we utilize a fully-connected layer $\mathbf{FC}(\cdot;\theta_{fc})$ with the trainable parameters θ_{fc} to predict the pseudo labels based on the learned relaxed representation $\boldsymbol{h}_i$ as follows:

$$\tilde{\boldsymbol{l}}_i = \mathrm{sigmoid}(\mathbf{FC}(\boldsymbol{h}_i;\theta_{fc})). \tag{8}$$

4.2 Label Prototype Learning

Motivation and Discussion. As we all know, in the real world, several objects often appear simultaneously in a scene, and the statistical co-occurrence probabilities between paired objects are different. The explicit semantic correlations among different objects can be captured by considering these statistical co-occurrence priors. However, the existing methods regard the label vector as a whole for directly guiding the learning processes of their models, which cannot model explicit semantic correlations among different categories.

Therefore, we design a label prototype learning module to capture these explicit semantic correlations on the category-level by considering the co-occurrence priors on the dataset. This module learns the prototype embeddings for all categories to generate the supervising hash codes by linear combinations of the learned embeddings. Therefore, these supervising hash codes can better preserve the fine-grained semantic correlations and guide the learning process of the bit-aware semantic transformer module.

Details. Firstly, all categories are projected into the feature embeddings with an $\mathbf{Embedding}(\cdot;\theta_{el})$ layer as follows:

$$\mathbf{E}^{L} = \mathbf{Embedding}(\mathcal{S}^{L};\theta_{el}),\ \mathbf{E}^{L} \in \mathbb{R}^{c\times k}, \tag{9}$$

where $\mathcal{S}^{L}$ is the category sequence, e.g., ["dog", "cat", "person", ...], and θ_{el} is the trainable parameters. To learn the label prototype embeddings with preserving the explicit semantic correlations among

different categories, we introduce a transformer encoder to model these correlations as follows:

$$\tilde{\mathbf{E}}^L = \mathbf{TransformerEncoder}^{(L)}(\mathbf{E}^L; \theta^L_{enc}),\ \tilde{\mathbf{E}}^L \in \mathbb{R}^{c\times k}, \quad (10)$$

$$\tilde{\mathbf{E}}^L = [\tilde{\boldsymbol{e}}^L_1, \tilde{\boldsymbol{e}}^L_2, ..., \tilde{\boldsymbol{e}}^L_c], \quad (11)$$

where θ^L_{enc} is the trainable parameters. Then we can learn the final label prototype embeddings by the category-wise hash function $\mathcal{H}^p_c(\cdot)$ as follows:

$$\begin{aligned}\mathbf{P}^L &= [\boldsymbol{p}^L_1, \boldsymbol{p}^L_2, ..., \boldsymbol{p}^L_c] \\ &= [\mathcal{H}^p_1(\tilde{\boldsymbol{e}}^L_1; \theta^p_{h1}), \mathcal{H}^p_2(\tilde{\boldsymbol{e}}^L_2; \theta^p_{h2}), ..., \mathcal{H}^p_c(\tilde{\boldsymbol{e}}^L_c; \theta^p_{hc})],\end{aligned} \quad (12)$$

where $\mathbf{P}^L \in \mathbb{R}^{c\times k}$ denotes all the label prototype embeddings and θ^p_{hc} is the trainable parameters. To avoid the quantization errors caused by sign$(\cdot)$ operator, we adopt the relaxed label prototype embeddings rather than binary representations. When the label prototype embeddings are learned, we use the annotated vector to generate the supervising hash code by linear combination of these embeddings as follows:

$$\boldsymbol{h}^L_i = \boldsymbol{l}_i\mathbf{P}^L,\ \boldsymbol{h}^L_i \in \mathbb{R}^{1\times k}, \quad (13)$$

$$\boldsymbol{b}^L_i = \text{sign}(\boldsymbol{h}^L_i),\ \boldsymbol{b}^L_i \in \{-1, 1\}^{1\times k}. \quad (14)$$

So this supervising hash code can preserve the explicit semantic correlations and is used to guide the learning process of the bit-aware semantic transformer module.

Finally, similar to Eq.(8), we predict the pseudo label to preserve the label semantic information into the supervising hash code with $\mathbf{FC}^{(L)}(\cdot; \theta^L_{fc})$ as follows:

$$\tilde{\boldsymbol{l}}^L_i = \text{sigmoid}(\mathbf{FC}^{(L)}(\boldsymbol{h}^L_i; \theta^L_{fc})), \quad (15)$$

where θ^L_{fc} is the trainable parameters.

4.3 Objective Function

Bit-aware Semantic Transformer. Under the supervised scenario, we utilize the classification loss $\mathcal{L}_{clf}$ to preserve the annotated semantic information into the predicted pseudo label:

$$\mathcal{L}_{clf} = ||\tilde{\boldsymbol{l}}_i - \boldsymbol{l}_i||^2_2. \quad (16)$$

To minimize the quantization errors by sign$(\cdot)$ operator, we utilize the supervising hash code $\boldsymbol{b}^L_i$ output from the label prototype learning module in Eq.(14) to guide the learning process of the relaxation representation $\boldsymbol{h}_i$ in Eq.(6):

$$\mathcal{L}_{sign} = ||\boldsymbol{h}_i - \boldsymbol{b}^L_i||^2_2. \quad (17)$$

To preserve the pairwise correlations between samples, the following loss function is designed with cosine similarity:

$$\mathcal{L}_{sim} = ||\cos(\boldsymbol{h}_i, \boldsymbol{h}_j) - \mathbf{S}_{ij}||^2_2, \quad (18)$$

where $\mathbf{S}$ is the affinity matrix, which can model the fine-grained associations between relevant samples. $\mathbf{S}_{ij}$ is constructed as follows:

$$\mathbf{S}_{ij} = \frac{2}{1 + e^{-\boldsymbol{l}_i\boldsymbol{l}^T_j}} - 1, \quad (19)$$

where $\mathbf{S}_{ij} \in [0, \frac{2}{1+e^{-c}} - 1]$.

Finally, we can obtain the whole objective function in the learning stage of the bit-aware semantic transformer module as follows:

$$\min_{\Theta_{BaT}} \mathcal{L} = \beta_1\mathcal{L}_{clf} + \beta_2\mathcal{L}_{sign} + \beta_3\mathcal{L}_{sim}, \quad (20)$$

where $\beta_1, \beta_2, \beta_3$ are the trade-off hyper-parameters, Θ_{BaT} denotes all the above trainable parameters in the bit-aware semantic transformer module.

Label Prototype Learning. Similar to Eq. (16), we also utilize the label information to guide the training process of the label prototype learning module, that is:

$$\mathcal{J}_{clf} = ||\tilde{\boldsymbol{l}}^L_i - \boldsymbol{l}_i||^2_2. \quad (21)$$

The quantization loss can be formulated as follows:

$$\mathcal{J}_{sign} = ||\boldsymbol{h}^L_i - \boldsymbol{b}^L_i||^2_2. \quad (22)$$

To preserve the explicit semantic correlations among different categories into the label prototype embeddings by considering the co-occurrence priors, we develop the co-occurrence statistic reconstruction loss as follows:

$$\mathcal{J}_{proto} = ||\cos(\mathbf{P}^L, \mathbf{P}^L) - \tilde{\mathcal{R}}||^2_F, \quad (23)$$

where $\tilde{\mathcal{R}} \in \mathbb{R}^{c\times c}$ is the normalized co-occurrence statistic matrix. $\tilde{\mathcal{R}}_{ij}$ represents the co-occurrence frequency on the training dataset between the i-th and j-th category.

In addition, the bit balance loss is introduced to maximize the information entropy [44] of the hash code, that is:

$$\mathcal{J}_{bal} = ||\boldsymbol{h}^L_i\mathbf{1}||^2_2, \quad (24)$$

where $\mathbf{1}$ is a column vector filled with 1.

Finally, the whole objective function of the label prototype learning module can be formulated as follows:

$$\min_{\Theta_{LPL}} \mathcal{J} = \gamma_1\mathcal{J}_{clf} + \gamma_2\mathcal{J}_{sign} + \gamma_3\mathcal{J}_{proto} + \gamma_4\mathcal{J}_{bal}, \quad (25)$$

where $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ are the trade-off hyper-parameters, Θ_{LPL} denotes all the above trainable parameters in the label prototype learning module.

4.4 Optimization and Out-of-Sample Extension

We first optimize the label prototype learning module and generate the supervising hash codes for all the training samples. Then, we use these supervising hash codes to guide the learning process of the bit-aware semantic transformer module. When a new multimodal query arrives, we can generate the corresponding hash code by the bit-aware semantic transformer module.

5 EXPERIMENTS

5.1 Experimental Datasets

We conduct the experiments on three benchmark datasets: MIR Flickr [8], NUS-WIDE [3] and MS COCO [14]. They all contain visual and textual data and are widely tested in recent multi-modal hashing methods [22, 41, 45]. Similar to [22, 45], we utilize the popular VGGNet [28] to extract the feature representations of images and Bag-of-Words (BoW) vectors as the text feature representations. The statistics of the above three datasets are summarized in Table 3, and the detailed settings are as follows:

Table 2: mAP comparison results on MIR Flickr, NUS-WIDE and MS COCO. The best result in each column is marked with bold. The second best result in each column is underlined.

Methods	Ref.	MIR Flickr				NUS-WIDE				MS COCO			
		16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
MFH [30]	TMM13	0.5795	0.5824	0.5831	0.5836	0.3603	0.3611	0.3625	0.3629	0.3948	0.3966	0.3960	0.3980
MAH [15]	TIP15	0.6488	0.6649	0.6990	0.7114	0.4633	0.4945	0.5381	0.5476	0.3967	0.3943	0.3966	0.3988
MVLH [27]	MM15	0.6541	0.6421	0.6044	0.5982	0.4182	0.4092	0.3789	0.3897	0.3993	0.4012	0.4065	0.4099
MvDH [26]	TIST18	0.6828	0.7210	0.7344	0.7527	0.4947	0.5661	0.5789	0.6122	0.3978	0.3966	0.3977	0.3998
MFKH [18]	MM12	0.6369	0.6128	0.5985	0.5807	0.4768	0.4359	0.4342	0.3956	0.4216	0.4211	0.4230	0.4229
DMVH [37]	ICMR17	0.7231	0.7326	0.7495	0.7641	0.5676	0.5883	0.6092	0.6279	0.4123	0.4288	0.4355	0.4563
SDMH [21]	TMM19	0.7316	0.7400	0.7568	0.7723	0.6321	0.6346	0.6626	0.6648	-	-	-	-
FOMH [20]	MM19	0.7557	0.7632	0.7654	0.7705	0.6329	0.6456	0.6678	0.6791	0.5008	0.5148	0.5172	0.5294
FDMH [16]	NPL20	0.7802	0.7963	0.8094	0.8181	0.6575	0.6665	0.6712	0.6823	0.5404	0.5485	0.5600	0.5674
DCMVH [45]	TIP20	0.8097	0.8279	0.8354	0.8467	0.6509	0.6625	0.6905	0.7023	0.5378	0.5427	0.5490	0.5576
SAPMH [41]	TMM21	0.7657	0.8098	0.8188	0.8191	0.6503	0.6703	0.6898	0.6901	0.5467	0.5502	0.5563	0.5672
FGCMH [22]	MM21	**0.8173**	**0.8358**	0.8377	0.8406	0.6677	0.6874	0.6936	0.7011	0.5641	0.5723	0.5797	0.5862
BSTH (ours)	-	0.8145	0.8340	**0.8482**	**0.8571**	**0.6990**	**0.7340**	**0.7505**	**0.7704**	**0.5831**	**0.6245**	**0.6459**	**0.6654**

Table 3: The statistics of the experimental datasets. 'D' denotes the feature dimensionality of specific modality.

Datasets	Training	Retrieval	Query	Categories	Visual	Textual
MIR Flickr [8]	5,000	17,772	2,243	24	4,096-D	1,386-D
NUS-WIDE [3]	21,000	193,749	2,085	21	4,096-D	1,000-D
MS COCO [14]	18,000	82,783	5,981	80	4,096-D	2,000-D

Table 4: mAP results on the extended MIR Flickr.

Methods	Ref.	16 bits	32 bits	64 bits	128 bits
FDMH [16]	NPL20	0.7897	0.8131	0.8314	0.8414
DCMVH [45]	TIP20	0.6379	0.7490	0.7759	0.7946
SAPMH [41]	TMM21	0.7595	0.8125	0.8166	0.8302
BSTH (ours)	-	**0.8791**	**0.8998**	**0.9141**	**0.9205**

MIR Flickr[1] consists of 25,000 image-text pairs collected from the Flickr website[2]. 20,015 pairs are selected and each pair belongs to at least one label from the 24 annotated categories. Following the commonly used experimental settings [22, 45], we randomly select 100 samples of each category and obtain 2,243 samples as the query set after removing the duplicates. The remaining 17,772 samples are served as the retrieval set and we randomly choose 5,000 samples from the retrieval set as the training set.

NUS-WIDE[3] is a web dataset containing 269,648 instances and each instance belongs to one of 81 semantic categories. We choose 195,834 instances from the 21 most common categories. Similarly, we randomly choose 100 instances of each category and obtain 2,085 samples as the query set after removing the duplicates. The remaining 193,749 instances are constructed as the retrieval set and 21,000 instances are randomly selected from the retrieval set for training.

MS COCO[4] is also a multi-label dataset and each sample is associated with at least one of the 80 categories. In our experiments, we adopt MS COCO 2014 dataset with 82,783 training samples and 40,504 validation samples. Referring to [22, 45], we randomly select 80 samples of each category from the validation set and remove the duplicate samples to obtain the query set with 5,981 samples. We use all the training samples to form our retrieval set and randomly choose 18,000 samples from the retrieval set as our training set.

5.2 Implementation Details

The implementation details of all the submodules mentioned in Section 4 are summarized as follows:

$\mathbf{MLP}^{(*)}(\cdot)$: Both $\mathbf{MLP}^{(I)}(\cdot)$ and $\mathbf{MLP}^{(T)}(\cdot)$ for visual and textual modalities consist of two linear projection layers ($d^{(I)} \rightarrow 2,048 \rightarrow d_c$) and ($d^{(T)} \rightarrow 1,024 \rightarrow d_c$), respectively, where $d_c = 128$.

$\mathbf{TransformerEncoder}^{(*)}(\cdot)$: The visual and textual modalities adopt same settings, i.e. single-head and two encoder layers.

$\mathbf{Embedding}(\cdot)$: The *Embedding* layer in PyTorch[5] is used to transfer all categories into the corresponding embeddings.

$\mathbf{TransformerEncoder}^{(L)}(\cdot)$: It adopts the setting with single-head and one encoder layer.

$\mathcal{H}(\cdot)$: All the bit-wise hash functions in Eq.(6) consist of two linear projection layers ($d_c \rightarrow \frac{d_c}{2} \rightarrow 1$), which project the concept representations into relaxed values of the corresponding hash bits.

$\mathcal{H}^p(\cdot)$: All the hash functions in Eq.(12) contain one linear projection layer ($k \rightarrow k$).

The values of all the trade-off hyper-parameters are same on three datasets: $\{\beta_1 = 1, \beta_2 = 0.01, \beta_3 = 1, \gamma_1 = 0.001, \gamma_2 = 100, \gamma_3 = 1, \gamma_4 = 0.01\}$. At the training stage, we set the batch size as 1,024 and optimize the whole network by a standard back-propagation (BP) algorithm with Adam optimizer [10]. The learning rate is empirically set to 0.001 for both the bit-aware semantic transformer module and the label prototype learning module. Our method is implemented via PyTorch and all the experiments are conducted on a server with a single GPU (NVIDIA RTX 2080Ti) and a CPU (2.20GHz Intel (R) Xeon (R) Silver 4214).

5.3 Evaluation Metric and Baselines

Following the existing multi-modal hashing retrieval methods [22, 41, 45], we adopt a standard evaluation metric, i.e. the mean Average Precision (mAP), to quantitatively evaluate the hashing performance of our proposed method. A higher mAP value indicates better performance.

[1] https://press.liacs.nl/mirflickr/
[2] https://www.flickr.com/
[3] https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html
[4] https://cocodataset.org/
[5] https://pytorch.org/

Table 5: The ablation study results on MIR Flickr, NUS-WIDE and MS COCO.

Vars.	MIR Flickr				NUS-WIDE				MS COCO			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
BSTH-T	0.7050	0.7128	0.7222	0.7271	0.5890	0.6197	0.6411	0.6538	0.5773	0.6219	0.6390	0.6571
BSTH-I	0.7986	0.8259	0.8318	0.8326	0.6655	0.6936	0.7194	0.7326	0.4007	0.4004	0.4043	0.4018
BSTH-FMLP	0.7903	0.8032	0.8103	0.8270	0.6601	0.6696	0.6979	0.6952	0.5305	0.5513	0.5620	0.5738
BSTH-SH	0.8076	0.8153	0.8224	0.8149	0.6888	0.7154	0.7174	0.7291	0.5617	0.5610	0.5487	0.5417
BSTH-NC	0.8054	0.8165	0.8224	0.8212	0.6936	0.7168	0.7457	0.7479	0.5623	0.5817	0.6024	0.6090
BSTH-PMLP	0.8048	0.8195	0.8383	0.8450	0.6900	0.7119	0.7339	0.7525	0.5299	0.5717	0.5959	0.6189
BSTH-SHARE	0.8130	0.8272	0.8372	0.8499	0.6931	0.7131	0.7488	0.7632	0.5371	0.5604	0.6029	0.6201
BSTH	**0.8145**	**0.8340**	**0.8482**	**0.8571**	**0.6990**	**0.7340**	**0.7505**	**0.7704**	**0.5831**	**0.6245**	**0.6459**	**0.6654**

We compare our proposed method with twelve state-of-the-art (SOTA) multi-modal hashing methods, including four unsupervised methods: MFH [30], MAH [15], MVLH [27], MvDH [26], and eight supervised methods: MFKH [18], DMVH [37], SDMH [21], FOMH [20], FDMH [16], DCMVH [45], SAPMH [41], FGCMH [22]. All the compared methods are summarized in Section 2. Related Work.

5.4 Quantitative Results and Comparisons

The mAP results of our method and all the compared baselines varying with the different hash code lengths (i.e. 16, 32, 64, and 128 bits) on three datasets are reported in Table 2. Note that, for fair comparisons, the mAP results of the compared baselines refer to the results provided in the original papers directly. From the Table 2, we can obtain the following observations and analyses.

Compared with all the state-of-the-art baselines, our method consistently achieves significant performance improvements with all hash code lengths on NUS-WIDE and MS COCO. Specifically, our method outperforms the second best methods by 3.13%, 4.66%, 5.69%, 6.81% on NUS-WIDE, and 1.9%, 5.22%, 6.62%, 7.92% on MS-COCO, with 16, 32, 64 and 128 bits, respectively. On MIR Flickr, our method achieves the comparable performances compared with competitive DCMVH and FGCMH. We can find that our method performs better on the large-scale NUS-WIDE and MS COCO datasets. In addition, most compared baselines and ours achieve different mAP performance improvements as the hash code length increases, and our method improves mAP performance more obviously with longer hash code. The main reason for the above observations is that our proposed method can excavate more accurate and comprehensive bit-wise semantic concepts with more training samples and longer hash code, which can better assist the fine-grained concept-level multi-modal fusion. Hence, our method can be effectively extended to large-scale retrieval scenarios. In addition, the compared methods adopt directly coarse semantic preserving strategies based on label vectors without considering fine-grained semantic correlations. In contrast, our method learns the prototype embeddings for all categories by considering the co-occurrence priors and generates the supervising hash codes, which can preserve the fine-grained semantic correlations among different samples. These supervising hash codes can fully exploit the annotated semantic information and guide the learning process of the bit-aware semantic transformer module.

To further validate the above analysis, we design a supplementary experiment on MIR Flickr, which serves the entire retrieval set as the training set and compares our method with several most competitive baselines, i.e. FDMH, DCMVH and SAPMH. The comparison results are shown in Table 4. We can find that FDMH, SAPMH and our method achieve different mAP performance improvements by enlarging the training set, and our method is more significant. Our method outperforms the second best methods by 8.94%, 8.67%, 8.27%, and 7.91% with 16, 32, 64 and 128 bits, respectively. However, the performance of DCMVH shows a decreasing trend, which demonstrates the limited ability of DCMVH to be extended to large-scale retrieval scenarios.

Table 6: Comparisons of training time and testing time in seconds with 64 bits on three datasets.

Methods	MIR Flickr		NUS-WIDE		MS COCO	
	Train	Test	Train	Test	Train	Test
MFH [30]	32.05	0.52	866.17	4.59	426.19	2.17
MVLH [27]	30.14	1.63	418.04	13.99	123.29	6.57
MvDH [26]	77.28	313.52	279.68	2813.03	345.74	1847.18
MFKH [18]	150.40	0.04	235.44	0.17	155.92	0.09
SDMH [21]	13.96	2.55	46.99	19.39	35.29	8.72
FOMH [20]	3.75	2.26	12.64	20.17	12.82	11.13
FDMH [16]	0.29	0.04	0.76	0.27	0.72	0.13
DCMVH [45]	298.52	0.14	490.29	0.80	882.52	0.51
SAPMH [41]	42.55	2.10	40.92	33.19	54.29	16.79
BSTH (ours)	135.11	1.61	259.46	16.81	231.99	9.71

As shown in Table 2, compared with the shallow methods, the deep-based methods (i.e. DCMVH and FGCMH) achieve different but limited mAP performance improvements. In contrast, our method outperforms them more significantly on NUS-WIDE and MS COCO, which indicates that our method can fully exploit the powerful representation capability of deep neural networks and be more effectively extended to large-scale scenarios. In addition, the supervised methods perform better than the unsupervised methods except MFKH on MIR Flickr and NUS-WIDE. Compared with unsupervised methods, supervised methods can exploit semantic information and improve retrieval performance.

To verify the scalability of our proposed method, we report the training time and testing time of several multi-modal hashing baselines on three benchmark datasets, shown in Table 6. Compared with the shallow methods, our approach consumes more time in the training stage. The main reason is that we need more training iterations to fully excavate bit-wise implicit semantic concepts to align heterogeneous modalities and reduce multi-modal semantic gaps in the bit-aware semantic transformer module. As the training stage is in the off-line mode, this time consumption is acceptable. In addition, compared with the deep-based method, i.e. DCMVH, we take more time in the testing stage because our method needs to perform multi-modal fusion on the fine-grained concept-level

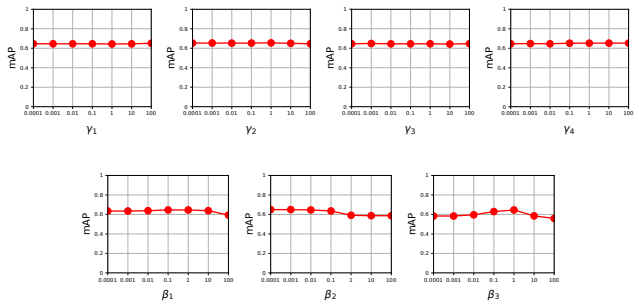

Figure 3: The parameter sensitivity results on MS COCO with 64 bits.

rather than feature-level. Within acceptable testing time increases, our method significantly improves mAP performance.

5.5 Ablation Study

To evaluate the effectiveness of the key components in our framework, we design seven variants to conduct the ablation study with the different hash code lengths (i.e. 16, 32, 64, and 128 bits) on three datasets, and the comparison results are shown in Table 5. These variants are summarized as follows: **BSTH-T**: It only utilizes the textual modality to perform hashing-based retrieval. **BSTH-I**: It only utilizes the visual modality to perform hashing-based retrieval. **BSTH-FMLP**: We remove the implicit concept learning submodule, containing the decoupling layer and transformer structure, and replace it with a MLP architecture. It fuses the heterogeneous multi-modal data on the feature-level rather than the fine-grained concept-level. **BSTH-SH**: To verify the effectiveness of bit-wise hash code generation process, we remove the bit-wise hash functions. The k implicit semantic concept representations are fused via sum operation and generate the final hash codes with a single hash function. **BSTH-NC**: It removes the guidance of co-occurrence priors in the label prototype learning module. **BSTH-PMLP**: It removes the label prototype learning module, and utilizes the original label vectors to learn the supervising hash codes in the label prototype learning module. **BSTH-SHARE**: We replace the modality-specific transformer encoder with a shared transformer encoder. Specifically, the modality-specific coarse implicit semantic concepts are jointly input into the shared transformer encoder. As shown in Table 5, we can get the following analyses:

BSTH-T and **BSTH-I**: Multi-modal fusion can effectively improve the retrieval performance compared with uni-modal.

BSTH-FMLP and **BSTH-SH**: Our proposed method achieves fairly acceptable performance by excavating bit-wise implicit semantic concepts to align the heterogeneous modalities and performing multi-modal fusion on the fine-grained concept-level. The bit-wise hash functions can effectively enhance the discrimination of the hash codes.

BSTH-NC and **BSTH-PMLP**: The results of them validate the effectiveness of our proposed label prototype learning module. The co-occurrence priors can model the explicit semantic correlations on the category-level. Compared with directly semantic preserving by label vectors, the learned label prototype embeddings can better preserve the fine-grained semantic correlations among samples into the supervising hash codes.

Table 7: The positional encoding on MS COCO.

Image	Text	MS COCO			
		16 bits	32 bits	64 bits	128 bits
		0.5707	0.6136	0.6329	0.6462
	✓	0.5806	0.6235	0.6337	0.6527
✓		0.5799	0.6163	0.6347	0.6541
✓	✓	**0.5831**	**0.6245**	**0.6459**	**0.6654**

BSTH-SHARE: We can observe that adopting the shared transformer encode leads to a decrease of mAP performance. The reason is that the modality heterogeneity interferes with the correlation modeling of implicit semantic concepts in the transformer encoder.

In addition, we observe the influence of the position encoding on mAP performance in different modalities. The comparison results on MS COCO are shown in Table 7. We can observe that the position encodings for both visual modality and textual modality achieve different performance improvements. The potential reason is that the position encoding can enhance the diversity of the implicit semantic concepts.

5.6 Parameter Sensitivity

There are seven hyper-parameters in our method to trading off all the loss function terms: β_1, β_2, β_3, γ_1, γ_2, γ_3 and γ_4, as shown in Eq.(20) and Eq.(25). To observe mAP performance variations with these hyper-parameters, we conduct the parameter experiments on MS COCO with 64 bits and tune them in the range of $\{0.0001, 0.001, 0.01, 0.1, 1, 10, 100\}$. As shown in Figure 3, we can find that the mAP performance is relatively stable in a wide range of γ_1, γ_2, γ_3 and γ_4, and in a certain range of β_1, β_2. When $\beta_3 = 1$, the best performance can be achieved.

6 CONCLUSION

In this paper, we propose a *Bit-aware Semantic Transformer Hashing* (BSTH) framework to excavate bit-wise semantic concepts and simultaneously align the heterogeneous modalities for multi-modal hash learning on the concept-level. We design bit-wise hash functions to encode the implicit semantic concepts to the corresponding hash bits. To supervise the bit-aware transformer module, we develop a label prototype learning module to learn prototype embeddings for all categories that capture the explicit semantic correlations on the category-level by considering the co-occurrence priors. This module generates the supervising hash codes for guiding the bit-aware semantic transformer module. Experiments on three widely tested multi-modal retrieval datasets demonstrate the superiority of the proposed method from various aspects.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under Grant 62172263 and Grant U1836216, in part by the Natural Science Foundation of Shandong, China, under Grant ZR2020YQ47 and Grant ZR2019QF002, in part by the Major Fundamental Research Project of Shandong, China, under Grant ZR2019ZD03, in part by the Youth Innovation Project of Shandong Universities, China, under Grant 2019KJN040.

REFERENCES

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-End Object Detection with Transformers. In *Proceedings of the European Conference on Computer Vision*. 213–229.

[2] Yongbiao Chen, Sheng Zhang, Fangxin Liu, Zhigang Chang, Mang Ye, and Zhengwei Qi. 2021. TransHash: Transformer-based Hamming Hashing for Efficient Image Retrieval. *arXiv preprint arXiv:2105.01823* (2021).

[3] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of the ACM International Conference on Image and Video Retrieval*. 1–9.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*.

[6] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. 2020. Multi-modal Transformer for Video Retrieval. In *Proceedings of the European Conference on Computer Vision*. 214–229.

[7] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural computation* 9, 8 (1997), 1735–1780.

[8] Mark J. Huiskes, Bart Thomee, and Michael S. Lew. 2010. New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative. In *Proceedings of the ACM SIGMM International Conference on Multimedia Information Retrieval*. 527–536.

[9] Qing-Yuan Jiang and Wu-Jun Li. 2017. Deep Cross-Modal Hashing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3270–3278.

[10] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *Proceedings of International Conference on Learning Representations*.

[11] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the International Conference on Learning Representations*.

[12] Chao Li, Cheng Deng, Ning Li, Wei Liu, Xinbo Gao, and Dacheng Tao. 2018. Self-Supervised Adversarial Hashing Networks for Cross-Modal Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4242–4251.

[13] Shuyan Li, Xiu Li, Jiwen Lu, and Jie Zhou. 2021. Self-Supervised Video Hashing via Bidirectional Transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13549–13558.

[14] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *Proceedings of the European Conference on Computer Vision*. 740–755.

[15] Li Liu, Mengyang Yu, and Ling Shao. 2015. Multiview Alignment Hashing for Efficient Image Search. *IEEE Transactions on Image Processing* 24, 3 (2015), 956–966.

[16] Luyao Liu, Zheng Zhang, and Zi Huang. 2020. Flexible Discrete Multi-view Hashing with Collective Latent Feature Learning. *Neural Processing Letters* 52, 3 (2020), 1765–1791.

[17] Song Liu, Shengsheng Qian, Yang Guan, Jiawei Zhan, and Long Ying. 2020. Joint-modal Distribution-based Similarity Hashing for Large-scale Unsupervised Deep Cross-modal Retrieval. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1379–1388.

[18] Xianglong Liu, Junfeng He, Di Liu, and Bo Lang. 2012. Compact kernel hashing with multiple features. In *Proceedings of the ACM International Conference on Multimedia*. 881–884.

[19] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030* (2021).

[20] Xu Lu, Lei Zhu, Zhiyong Cheng, Jingjing Li, Xiushan Nie, and Huaxiang Zhang. 2019. Flexible Online Multi-modal Hashing for Large-scale Multimedia Retrieval. In *Proceedings of the ACM International Conference on Multimedia*. 1129–1137.

[21] Xu Lu, Lei Zhu, Jingjing Li, Huaxiang Zhang, and Heng Tao Shen. 2019. Efficient Supervised Discrete Multi-View Hashing for Large-Scale Multimedia Search. *IEEE Transactions on Multimedia* 22, 8 (2019), 2048–2060.

[22] Xu Lu, Lei Zhu, Li Liu, Liqiang Nie, and Huaxiang Zhang. 2021. Graph Convolutional Multi-modal Hashing for Flexible Multimedia Retrieval. In *Proceedings of the ACM International Conference on Multimedia*. 1414–1422.

[23] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).

[24] Sam T Roweis and Lawrence K Saul. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 5500 (2000), 2323–2326.

[25] Fumin Shen, Yan Xu, Li Liu, Yang Yang, Zi Huang, and Heng Tao Shen. 2018. Unsupervised Deep Hashing with Similarity-Adaptive and Discrete Optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 12 (2018), 3034–3044.

[26] Xiaobo Shen, Fumin Shen, Li Liu, Yunhao Yuan, Weiwei Liu, and Quan-Sen Sun. 2018. Multiview Discrete Hashing for Scalable Multimedia Search. *ACM Transactions on Intelligent Systems and Technology* 9, 5 (2018), 53:1–53:21.

[27] Xiao-Bo Shen, Fumin Shen, Quan-Sen Sun, and Yunhao Yuan. 2015. Multi-view Latent Hashing for Efficient Multimedia Search. In *Proceedings of the ACM International Conference on Multimedia*. 831–834.

[28] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of International Conference on Learning Representations*.

[29] Jingkuan Song, Tao He, Lianli Gao, Xing Xu, Alan Hanjalic, and Heng Tao Shen. 2018. Binary Generative Adversarial Networks for Image Retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*. 394–401.

[30] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Jiebo Luo. 2013. Effective Multiple Feature Hashing for Large-Scale Near-Duplicate Video Retrieval. *IEEE Transactions on Multimedia* 15, 8 (2013), 1997–2008.

[31] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2020. VL-BERT: Pre-training of Generic Visual-Linguistic Representations. In *Proceedings of the International Conference on Learning Representations*.

[32] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. 2019. VideoBERT: A Joint Model for Video and Language Representation Learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7463–7472.

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Proceedings of the Advances in Neural Information Processing Systems*. 5998–6008.

[34] Bokun Wang, Yang Yang, Xing Xu, Alan Hanjalic, and Heng Tao Shen. 2017. Adversarial Cross-Modal Retrieval. In *Proceedings of the ACM International Conference on Multimedia*. 154–162.

[35] Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. 2019. CAMP: Cross-Modal Adaptive Message Passing for Text-Image Retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5763–5772.

[36] Chenggang Yan, Biao Gong, Yuxuan Wei, and Yue Gao. 2021. Deep Multi-View Enhancement Hashing for Image Retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 4 (2021), 1445–1451.

[37] Rui Yang, Yuliang Shi, and Xin-Shun Xu. 2017. Discrete Multi-view Hashing for Effective Image Retrieval. In *Proceedings of the ACM International Conference on Multimedia Retrieval*. 175–183.

[38] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. XLNet: Generalized Autoregressive Pretraining for Language Understanding. In *Proceedings of the Advances in Neural Information Processing Systems*. 5754–5764.

[39] Jian Zhang and Yuxin Peng. 2019. SSDH: Semi-Supervised Deep Hashing for Large Scale Image Retrieval. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 1 (2019), 212–225.

[40] Liangli Zhen, Peng Hu, Xu Wang, and Dezhong Peng. 2019. Deep Supervised Cross-Modal Retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10394–10403.

[41] Chaoqun Zheng, Lei Zhu, Zhiyong Cheng, Jingjing Li, and An-An Liu. 2021. Adaptive Partial Multi-View Hashing for Efficient Social Image Retrieval. *IEEE Transactions on Multimedia* 23 (2021), 4079–4092.

[42] Chaoqun Zheng, Lei Zhu, Xu Lu, Jingjing Li, Zhiyong Cheng, and Hanwang Zhang. 2020. Fast Discrete Collaborative Multi-Modal Hashing for Large-Scale Multimedia Retrieval. *IEEE Transactions on Knowledge and Data Engineering* 32, 11 (2020), 2171–2184.

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

[44] Xiang Zhou, Fumin Shen, Li Liu, Wei Liu, Liqiang Nie, Yang Yang, and Heng Tao Shen. 2020. Graph Convolutional Network Hashing. *IEEE Transactions on Cybernetics* 50, 4 (2020), 1460–1472.

[45] Lei Zhu, Xu Lu, Zhiyong Cheng, Jingjing Li, and Huaxiang Zhang. 2020. Deep Collaborative Multi-View Hashing for Large-Scale Image Search. *IEEE Transactions on Image Processing* 29 (2020), 4643–4655.