applied sciences

MDPI

Article

Multi-Grained Similarity Preserving and Updating for Unsupervised Cross-Modal Hashing

Runbing Wu, Xinghui Zhu, Zeqian Yi, Zhuoyang Zou, Yi Liu and Lei Zhu *

College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, China; runbingwu@stu.hunau.edu.cn (R.W.); zhuxh@hunau.edu.cn (X.Z.); yzq@stu.hunau.edu.cn (Z.Y.); zzy@stu.hunau.edu.cn (Z.Z.); yiliu@hunau.edu.cn (Y.L.)
* Correspondence: leizhu@hunau.edu.cn

Abstract: Unsupervised cross-modal hashing is a topic of considerable interest due to its advantages in terms of low storage costs and fast retrieval speed. Despite the impressive achievements of existing solutions, two challenges remain unaddressed: (1) Semantic similarity obtained without supervision is not accurate enough, and (2) the preservation of similarity structures lacks effectiveness due to the neglect of both global and local similarity. This paper introduces a new method, Multi-Grained Similarity Preserving and Updating (MGSPU), to tackle these challenges. To overcome the first challenge, MGSPU employs a newly designed strategy to update the semantic similarity matrix, effectively generating a high-confidence similarity matrix by eliminating noise in the original cross-modal features. For the second challenge, a novel multi-grained similarity preserving method is proposed, aiming to enhance cross-modal hash code learning by learning consistency in multi-grained similarity structures. Comprehensive experiments on two widely used datasets with nine state-of-the-art competitors validate the superior performance of our method in cross-modal hashing.

Keywords: unsupervised cross-modal hashing; attention mechanism; similarity preserving

Citation: Wu, R.; Zhu, X.; Yi, Z.; Zou, Z.; Liu, Y.; Zhu, L. Multi-Grained Similarity Preserving and Updating for Unsupervised Cross-Modal Hashing. *Appl. Sci.* **2024**, *14*, 870. https://doi.org/10.3390/app14020870

Academic Editor: José Salvador Sánchez Garreta

Received: 18 December 2023
Revised: 15 January 2024
Accepted: 17 January 2024
Published: 19 January 2024

1. Introduction

As an efficient information retrieval paradigm for big multimedia data [1–3], cross-modal retrieval [4–9] utilizes one modality as a query to search another modal data. Among the existing techniques [10–18], cross-modal hashing [16–20] is popular for its fast retrieval speed and low storage cost. The core of cross-modal hashing is to map high-dimensional data to a low-dimensional common Hamming space, in which multi-modal instances (text, image, audio and video) with similar semantics are closer, and dissimilar instances are far apart. Notwithstanding the strong practical significance, this task is extremely challenging due to the heterogeneity gap between different modalities.

A number of studies are striving to promote the progress of cross-modal hashing. In general, they can be simply divided into two groups: supervised learning methods [21–25] and unsupervised learning methods [26–30]. The supervised learning methods use human-annotated label to guide model learning, which preserves semantic discrimination well to obtain great performance. However, directly obtaining the category information is really difficult in real-world scenarios, while the cost of manual labeling is so huge as to limit the wide application. In contrast, without relying on category annotations, unsupervised learning methods are great at capturing the underlying correlation patterns between different modalities to bridge the heterogeneity gap. Thus, they have gained considerable attention currently.

Motivation. Although existing researches have shown remarkable results on large-scale datasets, unsupervised cross-modal hashing is yet a Herculean task that await further exploration due to the following challenges. *The first challenge is how to accurately measure inter-modal and intra-modal similarity without category information* . In the early days, several

unsupervised learning methods [31,32] used discrete models to calculate binary similarity, namely, defining the similarity between two instances with small distances as "1", otherwise "−1". Unfortunately, the naive similarity measurement is far from accurate. To break this limitation, researchers [12,15,33,34] began to use original feature distance to measure the continuous similarity. However, such similarity is still not accurate enough to represent the real relationship between instances due to the noise existing in their original features. *The second challenge is how to preserve similarity structure consistency from both local and global view during hash learning process*. Undisputedly, using multi-grained similarity relationships across modalities is crucial to narrow cross-modal heterogeneity. However, current methods [35–37] either overlook multi-grained similarity learning across various modalities or struggle to capture a consistent distribution of relations within the content details. Additionally, some state-of-the-art methods [38–41] do not account for the semantic correlation between feature vectors from one modality and the corresponding hash codes from another modality. To conquer these challenges, this paper has made the following two efforts: (1) designing a novel and effective similarity updating strategy to rise the accuracy of underlying similarity measurement, and (2) combining multi-grained similarity information from local and global view to preserve similarity consistency.

Our Method. We propose a novel cross-modal hashing framework, named **M**ulti-**G**rained **S**imilarity **P**reserving and **U**pdating (**MGSPU**). As shown in Figure 1, this framework consists four module, i.e., feature extraction, semantic similarity matrix updating, hash code learning and hash function learning. To overcome the first challenge, a novel semantic similarity matrix update strategy is developed, which removes noises from the similarity matrix so as to obtain high-confidence supervisory signal. To face the second challenge, we capture the multi-grained similarity structure information from global and local view to preserve similarity consistency better. For global view, we use graph convolutional network (GCN for short) to aggregate the similarity structure information from the neighbors of each instance to enrich the coarse-grained similarity information. For local view, cross-modal attention mechanisms are used to perform cross-modal interaction to enhance similarity learning between instance pairs. Furthermore, we use a similarity consistency reconstruction method to ensure the similarity consistency of hash codes.

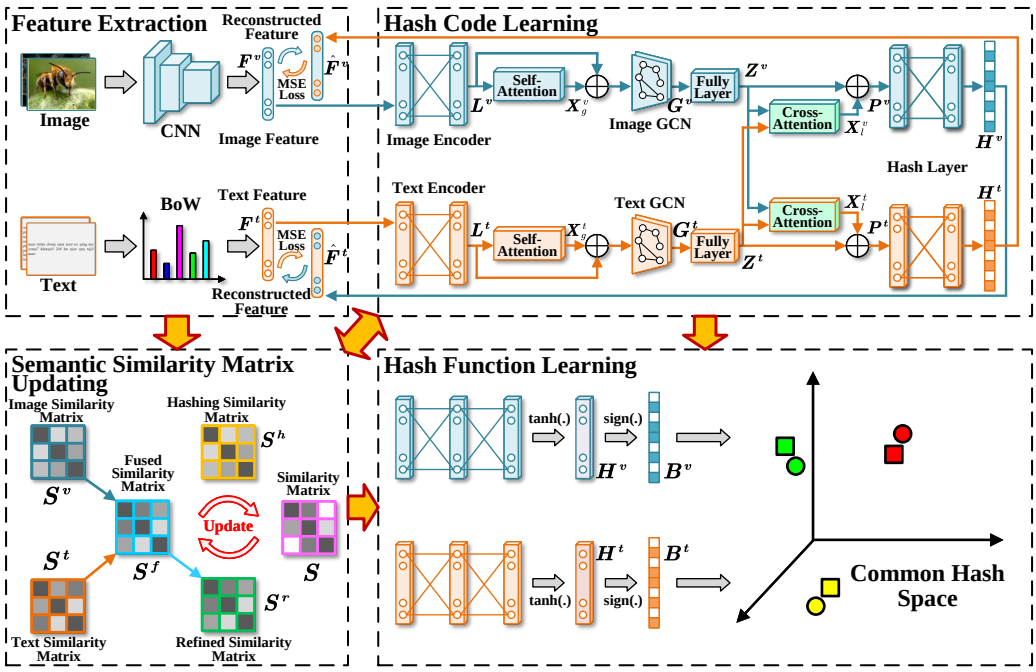


Figure 1. The framework of MGSPU. It comprises four main modules: feature extraction, semantic similarity matrix updating, hash code learning and hash function learning module.

Contributions. In summary, the contributions of this paper can be summarized as follows:

- We propose an effective unsupervised learning framework, called Multi-Grained Similarity Preserving and Updating, which learns high-quality hash codes by comprehensively improving cross-modal similarity learning.
- We propose a novel semantic similarity matrix updating strategy to effectively remove noises in the original similarity matrix, which produces high-confidence supervisory signal for cross-modal hashing learning.
- We propose a novel multi-grained similarity preserving method to enhance similarity consistency preserving of the cross-modal hash codes.
- We conducted extensive experiments on the widely used datasets MIRFLICKR-25K and NUS-WIDE to validate the superiority of the proposed approach and evaluate the effectiveness of each component.

Roadmap. The paper is structured as follows: Section 2 provides a summary of related works. Section 3 outlines the problem definition and details the MGSPU approach. Section 4 presents the experimental results and analysis. Lastly, Section 5 concludes the paper.

2. Related Work

According to literatures, cross-modal hashing techniques are generally categorized into two groups: supervised and unsupervised cross-modal hashing. This section reviews the prevailing solutions related to this paper, which are summarized in Table 1 to ease reading.

Table 1. Related works summary.

Type	Reference	Source	Authors	Description
Supervised	SCM [42]	AAAI (2014)	Zhang, D., et al.	Integrate semantic tags into the data modeling process.
	SePH [43]	CVPR (2015)	Lin, Z., et al.	The semantic similarity given by the training data is converted into a probability distribution, and then the hash code is learned by minimizing the KL divergence.
	DCMH [17]	CVPR (2017)	Jiang, Q.Y., et al.	Feature learning and hash learning are integrated for the first time.
	CMHH [44]	ECCV (2018)	Cao, Y., et al.	Hamming distance replaces the inner product and uses pairwise loss based on exponential distribution to solve the problem of uneven positive and negative samples.
	SSAH [45]	AAAI (2020)	Jin, S., et al.	Derive semantic ranking information using data features and tags and integrate it into cross-modal hashes.
	RDCMH [46]	AAAI (2019)	Liu, X., et al.	A multi-level semantic similarity matrix is constructed by considering the bidirectional relation, the relation between the whole instances.
	Bi_NCMH [47]	CVPR (2022)	Sun, C., et al.	The potential semantic relationships of data are captured by constructing a joint semantic matrix, but redundant information is introduced.

Table 1. *Cont.*

Type	Reference	Source	Authors	Description
Unsupervised	DJSRH [12]	ICCV (2019)	Su, S., et al.	The potential semantic relationships of data are captured by constructing a joint semantic matrix, but redundant information is introduced.
	JDSH [36]	ACM (2020)	Liu, S., et al.	The proposed sampling weighting scheme can generate more efficient hash codes.
	DAEH [27]	TCSVT (2022)	Shi, Y., et al.	Design information mixed similarity estimation integrates distance distribution and similarity rate information.
	DGCPN [48]	AAAI (2021)	Yu, J., et al.	Using a graph model to integrate neighborhood information, the overall loss is maintained by designing three types of data similarity.
	AGCH [26]	TMM (2021)	Zhang, P., et al.	The semantic structure of data is mined by GCN, and the structural information is explored from multiple perspectives by using a variety of similarity measures.
	DRNPH [49]	Mathematics (2022)	Yang, X., et al.	Cross-modal triples are used to maintain data similarity, and the distance between similar samples is required to be smaller than that between dissimilar pairs.

2.1. Supervised Cross-Modal Hashing

To obtain common binary representations, supervised cross-modal hashing methods use category information to maintain semantic discrimination. For example, semantic correlation maximization (SCM [42]) learns hash functions by constructing and maintaining semantic similarity matrices. Semantics-preserving hashing (SePH [43]) converts the semantic matrix into a probability distribution and minimizes the Kullback-Leibler divergence so that the learned hash codes are approximately distributed in Hamming space. Thanks to the prosperity of deep learning, researchers began to realize cross-modal retrieval by deep models that explore more discriminative features. Deep cross-modal hashing (DCMH [17]) is a pioneering achievement in this field, which perfectly combines hash function learning with deep feature extraction, and cleverly exploits the labelling information to construct similarity matrices, thus preserving the subtle similarity relationships in cross-modal data. Meanwhile, cross-modal hamming hashing (CMHH [44]) introduces a pairwise focusing loss based on exponential distribution. It penalises instances with similar semantic content but the Hamming distance exceeds a predefined threshold. Semi-supervised adversarial deep hashing (SSAH [45]) is a groundbreaking innovation that shifts the focus to self-supervised methods and integrates adversarial learning into cross-modal hashing, thus making significant progress in the field. Ranking-based deep cross-modal hashing (RDCMH [46]) uses maximum marginal loss to learn uniform Hamming representations. In addition, deep normalized cross-modal hashing with bi-direction relation reasoning (Bi-NCMH [47]) achieved excellent retrieval performance by constructing high-quality similarity matrices to capture similarity relations between instances with multiple labels. Despite the impressive performance of these techniques, their inherent limitations cannot be ignored: they rely heavily on manual annotation to obtain supervision.

2.2. Unsupervised Cross-Modal Hashing

Unsupervised cross-modal hashing methods do not rely on labels so as to apply in real-world scenarios easily. In the existing solutions, constructing a semantic similarity matrix is the key to guide cross-modal relationship learning.Among them, an ingenious method is Deep Joint Semantic Reconstruction Hash (DJSRH [12]), which proposes a joint semantic similarity matrix to simultaneously integrate multi-modal similarity information

of cross-modal instances. However, the similarity matrix in DJSRH introduces redundant information from intra-modal fusion items. As an improvement, joint-modal distribution-based similarity hashing (JDSH [36]) involves distribution-based similarity decision and weighting to learn more discriminative hash codes. Deep adaptively-enhanced hashing (DAEH [27]) utilizes distance distributions and similarity ratio information to estimate comparable similarity relationships as complementarity of simple feature distance-based metrics. Although competitive performance is achieved, lack of accurate similarity measurement are these methods since the similarity matrix they rely on are constructed from the original features with noises.

Other unsupervised learning studies make efforts on narrowing the heterogeneity gap by preserving similarity structure consistency. For example, deep graph-neighbor coherence preserving network (DGCPN [48]) explores the relationship information between data and their neighbors to capture neighborhood structure coherence. Aggregation-based graph convolutional hashing (AGCH [26]) employs a GCN to deeply explore the underlying neighborhood structure. Deep relative neighbor relationship preserving hashing (DRNPH [49]) method excavates deep relative neighbor relationships in common Hamming space via binary feature vector based intra- and inter-modal neighbor matrix reconstruction. Impressive progress had been made by these works, however, a common weakness they suffer from is lack of both global and local similarity learning from multi-modal contents.

To address the above shortcomings, we attempt to boost cross-modal hashing learning from two aspects: (1) trying to rise the quality of similarity matrix by reducing noises from original features, and (2) inviting multi-grained similarity learning strategy to capture both global and local similarity relationships. As a results, a novel unsupervised cross-modal hashing technique called multi-grained similarity preserving and updating is developed. The detailed details of this approach will be explored in depth in the next section.

3. Methodology

This section details the proposed MGSPU method across six aspects. Initially, we introduce the notations and problem definition in the Preliminary. Next is an overview of MGSPU, feature extraction, semantic similarity matrix updating, hashing learning (including hash code learning and hashing function learning), and finally a discussion of the optimization algorithm. All abbreviations involved are summarized in Table 2 for easier reading.

Table 2. Abbreviation summary.

Abbreviation	Meaning
MGSPU	Multi-Grained Similarity Preserving and Updating
CNN	Convolutional Neural Network
BoW	Bag of words
GCNs	Graph Convolutional Networks
ResNet	Residual Networks
LSTM	Long Short-Term Memory Network
MGSP	Multi-Grained Similarity Preserving
GSA	Global Similarity Aggregation
LSI	Local Similarity Interaction
SCR	Similarity Consistency Reconstruction

3.1. Preliminary

Notations. For clarity and simplicity, calligraphy uppercase letters, such as $\mathcal{O}$, denote sets. Bold uppercase letters, such as $\boldsymbol{W}$, represent matrices. Bold lowercase letters, such as $\boldsymbol{w}$, indicate vectors. In addition, the ij-th element of $\boldsymbol{W}$ is represented as $\boldsymbol{W}_{ij}$, the i-th row of $\boldsymbol{W}$ is represented as $\boldsymbol{W}_{i*}$, the j-th column of $\boldsymbol{W}$ is represented as $\boldsymbol{W}_{*j}$, $\boldsymbol{W}^T$ is the transpose of $\boldsymbol{W}$, $\boldsymbol{I}$ represents identity matrix. $\|.\|$ represents the Frobenius norm of a matrix. $sgn(.)$ represents a sign function, shown as below:

$$sgn(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases}. \quad (1)$$

For ease of comprehension, commonly utilized mathematical notations have been compiled in Table 3 to assist in reference.

Table 3. Notation summary.

Notation	Definition
$\boldsymbol{v}_i$	The i-th image instance
$\boldsymbol{t}_i$	The i-th text instance
$\boldsymbol{F}^v$	The original image features after feature extraction
$\boldsymbol{F}^t$	The original text features after feature extraction
f^v	Image hash function
f^t	Text hash function
$\boldsymbol{S}^v$	Image similarity matrix
$\boldsymbol{S}^t$	Text similarity matrix
$\boldsymbol{S}^f$	Fused similarity matrix
$\boldsymbol{S}^r$	Refined similarity matrix
$\boldsymbol{S}^h$	Hashing similarity matrix
$\boldsymbol{\Gamma}$	The indicator matrix
$\boldsymbol{\Lambda}$	The indicator matrix
$\boldsymbol{S}$	Unified similarity matrix
$\boldsymbol{L}^*$	The corresponding latent representations generated by the image or text encoder
$\boldsymbol{X}_g^*$	Image or text global similarity aggregation representations by self-attention mechanism
$\boldsymbol{G}^*$	Final outputs of image or text GCN
$\boldsymbol{Z}^*$	Image or text share the outputs of a fully connected network through a two-layer of weights
$\boldsymbol{X}_l^v$	Image attention-weighted representations by cross-attention mechanism
$\boldsymbol{X}_l^t$	Text attention-weighted representations by cross-attention mechanism
$\boldsymbol{P}^v$	Image features output by the local similarity interaction sub-module
$\boldsymbol{P}^t$	Text features output by the local similarity interaction sub-module
$\hat{\boldsymbol{F}}^v$	Reconstructed image feature representations
$\hat{\boldsymbol{F}}^t$	Reconstructed text feature representations
$\boldsymbol{H}^v$	Real-valued hash codes for image instances
$\boldsymbol{H}^t$	Real-valued hash codes for text instances
$\boldsymbol{B}^v$	Binary code for image instances
$\boldsymbol{B}^t$	Binary code for text instances

Problem Definition. This paper concentrates on cross-modal retrieval between two prevalent modalities, i.e., image v and text t. Let $\mathcal{O} = \{(\boldsymbol{v}_i, \boldsymbol{t}_i)\}_{i=1}^n$ be a cross-modal dataset, $\boldsymbol{v}_i \in \mathbb{R}^{d_v}$ and $\boldsymbol{t}_i \in \mathbb{R}^{d_t}$ refer to the image and text in the pairwise instance respectively, d_v and d_t represent the dimension of the corresponding features, n denotes the number of instances. Our study aims to learn two hash functions $f^v = (\cdot; \boldsymbol{\theta}_v)$ and $f^t = (\cdot; \boldsymbol{\theta}_t)$ to map images and texts to a common Hamming space: $\boldsymbol{B}^v = sgn(f^v = (\{\boldsymbol{v}_i\}_{i=1}^n; \boldsymbol{\theta}_v))$, $\boldsymbol{B}^t = sgn(f^t = (\{\boldsymbol{t}_i\}_{i=1}^n; \boldsymbol{\theta}_t))$, where $\boldsymbol{B}^v, \boldsymbol{B}^t \in \{-1, 1\}^{n \times C}$, C represents the length of the binary code, $\boldsymbol{\theta}_v$ and $\boldsymbol{\theta}_t$ are the learnable parameters. In addition, the similarity between two binary codes $\boldsymbol{B}_i^v$ and $\boldsymbol{B}_j^t$ is measured by Hamming distance $D\left(\boldsymbol{B}_i^v, \boldsymbol{B}_j^t\right) = \frac{1}{2}\left(C - < \boldsymbol{B}_i^v, \boldsymbol{B}_j^t >\right)$, which is the base for implementing cross-modal hashing retrieval.

3.2. Overview of MGSPU

Figure 1 shows the overview of the MGSPU framework. Specifically, it mainly consists of four modules: (1) *feature extraction module*, (2) *semantic similarity matrix updating module*, (3) *hash code learning module*, and (4) *hash function learning module*. The feature extraction module maps images and texts into feature subspaces by corresponding encoders: deep CNN for image features and Bag-of-Word (BoW) model for text features. The semantic

similarity matrix updating module realizes a novel similarity updating strategy to construct a high-confidence similarity matrix, which effectively enhance the hash code learning and hash function learning. To support hash function learning, the hash code learning module aims at generating real-value hash representations with a novel multi-grained semantic preserving learning strategy. It utilizes a pair of GCNs (each for one modality) to capture global semantic correlation, following by a cross-attention mechanism to enhance cross-modal local feature learning. This technique allows for sufficient semantic correlations preserving of both intra- and inter-modality. The hash function learning module is to learn two hash functions that are implemented by two fully-connected neural networks for image and text, respectively.

In a nutshell, the goal of this method is to optimize the hash function module with other three parts to generate high-quality binary hash codes for cross-modal retrieval task. Thereinafter, we discuss the technical details module by module.

3.3. Feature Extraction

The first step of cross-modal hash code learning is feature extraction from both images and texts. For images, we follow previous works [12,37,48] to extract deep features from CNN model (pre-trained on ImageNet). Specifically, we extract the the 4096-dimensional features from the first fc7 layer (after ReLU) of AlexNet [50] as the original image features $\boldsymbol{F^v} \in \mathbb{R}^{m \times 4096}$ for the batch-input images $\{\boldsymbol{v}_i\}_{i=1}^m$. While for texts $\{\boldsymbol{t}_i\}_{i=1}^m$ we directly adopt the BoW [51] vectors as their original features $\boldsymbol{F^t} \in \mathbb{R}^{m \times d_t}$. It is noted that the proposed MGSPU framework can also be compatible with other deep feature extraction models, such as ResNet [52] for images or LSTM [53] for texts. To better validate the superiority of the proposed technique, we use relatively naive but effective feature extraction models (AlexNet and BoW) when implementing MGSPU. The implementation details are presented in Section 4.3.

3.4. Semantic Similarity Matrix Updating

Unlike supervised hashing learning that do not easily suffer from noises in features due to accurate category annotation, the unsupervised method cannot ensure the high accuracy of latent semantic correlations between instances without category information. To address this issue, we focus on constructing a high-confidence unified similarity matrix $\boldsymbol{S}$ as a supervisory signal to filter out noises as much as possible. Accordingly, a novel semantic similarity matrix updating strategy is designed.

3.4.1. Similarity Matrix Construction

To comprehensively consider cross-modal similarity relationships, we first fuse the similarity matrices from image modality and text modality:

$$\boldsymbol{S}^f = \alpha_1 \boldsymbol{S}^v + (1 - \alpha_1)\boldsymbol{S}^t, \tag{2}$$

where $\boldsymbol{S}^f$ is the fused similarity matrix, $\boldsymbol{S}^v$ and $\boldsymbol{S}^t$ represent the image and text similarity matrices, respectively. $\alpha_1 \in [0, 1]$ is a hyperparameter to balance them. Each element in $\boldsymbol{S}^v$ and $\boldsymbol{S}^t$ is computed by Cosine similarity. Taking image modality as an example:

$$\boldsymbol{S}^v_{ij} = cos(\boldsymbol{v}_i, \boldsymbol{v}_j) = \frac{\boldsymbol{v}_i^T \boldsymbol{v}_j}{\|\boldsymbol{v}_i\|_F^2 \|\boldsymbol{v}_j\|_F^2}, \tag{3}$$

where v_i and v_j denote the feature of the i-th and j-th image. To rise the accuracy of similarity measurement without supervision, we design a filter-and-augment strategy to refine the similarity matrix, shown as follows:

$$S^r = \begin{cases} sgn(S^f), & \left|S^f_{ij}\right| > \eta_1 \\ 2 \times sigmoid(2 \times S^f) - 1 + I, & otherwise \end{cases}, \tag{4}$$

where S^r denotes the refined similarity matrix, η_1 is similarity threshold. the sign function $sgn(S^f)$ indicates if $S^f_{ij} > \eta_1$, $S^f_{ij} = 1$; if $S^f_{ij} < -\eta_1$, $S^f_{ij} = -1$. Meanwhile, inspired by reference [28], we use the non-linear function $sigmoid(\cdot)$ to compress the remaining values to further expand the distance between similar and non-similar instances. The coefficient 2 of S^f is to control the linear compression, then $2 \times sigmoid(\cdot) - 1$ maps the compressed value to $[-1, 1]$. I is an identity matrix to maintain the similarity between instance pairs.

3.4.2. Dual Instruction Fusion Updating

In order to preserve semantic similarity consistency in cross-modal hash codes, a reliable similarity matrix for unsupervised hash function learning is indispensable. During the training process, unfortunately, we found that there is a evident gap between the similarity relationships of cross-modal hash codes and the real relationships of original instances due to the noises existing in the flawed similarity matrix S^r. In other words, the value of $|S^r_{ij} - S^h_{ij}|$ is not small enough to correctly guide cross-modal hash function learning. Intuitively, if we rectify these incorrect similarity values, it will effectively embed the correct similarity relationship information into cross-modal hash codes. For this reason, we attempt to construct dual instruction to further denoise the similarity matrix S^r: the one is the difference of similarity value of S^r_{ij} and S^h_{ij}, the other is difference of the sign of them. Beyond all doubt, these two instructions actually measure the difference between S^r and S^h from two different perspectives, which could be fused to narrow the gap between S^r and S^h.

From the analysis mentioned, we develop a novel semantic similarity matrix updating strategy, named dual instruction fusion updating, to gradually eliminate the noises. Firstly, we construct the hashing similarity matrix S^h via the real-valued hash code H^v and H^t:

$$S^h = cos(H^v, H^v) + cos(H^t, H^t) + cos(H^v, H^t). \tag{5}$$

Thereafter, we utilize S^h to update S^r according to the similarity relationship among them. For the convenience of formal description and implementation, we introduce two indicator matrices Λ and Γ at first. The former is defined as $\Lambda_{ij} = S^r_{ij} \oplus S^h_{ij}$, therein $\oplus$ is XOR operation, which means if $S^r_{ij} \times S^h_{ij} > 0$, $\Lambda_{ij} = 0$; otherwise $\Lambda_{ij} = 1$. The latter indicates the numerical difference between S^h_{ij} and S^r_{ij}, therein the elements $|S^r_{ij} - S^h_{ij}| > \eta_2$ are set to 1 and the others are set to 0. $\eta_2 \in [0, 1]$ be a threshold to measure the difference between two elements in similar matrices. Accordingly, two cases are considered: (1) Not updating: if $\Lambda_{ij} = 0$ and $\Gamma_{ij} = 0$, then the updating do not be conducted since S^h and S^r are similar enough. (2) Updating: according to the value of Λ_{ij}, we introduce two updating rules: if $\Lambda_{ij} = 0$ and $\Gamma_{ij} = 1$, we update S^r_{ij} by S^h_{ij}; if $\Lambda_{ij} = 1$, we update S^r_{ij} by "0", which is a way with maximizing entropy to enhance generalization capability. The reason behind this rule is intuitive: if the difference between S^r_{ij} and S^h_{ij} is too large, a wise way is not to biased towards either side. To clearly show the update rules, we list a truth table for Λ

and $\boldsymbol{\Gamma}$ in Table 4. Thereby, the updated semantic similarity matrix $\boldsymbol{S}$ is generated by the following updating process:

$$\boldsymbol{S}_{ij} = \begin{cases} \boldsymbol{S}^r_{ij}, & \boldsymbol{\Lambda}_{ij} = 0 \cap \boldsymbol{\Gamma}_{ij} = 0 \\ \alpha_2 \boldsymbol{S}^r_{ij} + (1-\alpha_2)\boldsymbol{S}^h_{ij}, & \boldsymbol{\Lambda}_{ij} = 0 \cap \boldsymbol{\Gamma}_{ij} = 1, \\ 0, & otherwise \end{cases} \tag{6}$$

where $\alpha_2 \in [0,1]$ is a parameter. Finally, we use $\boldsymbol{S}$ to guide the cross-modal hash function learning avoiding the interference from similarity noise. Figure 2 illustrates the updating process.

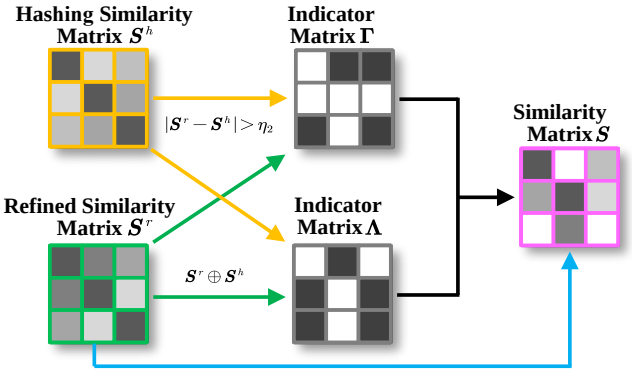


Figure 2. The process of dual instruction fusion updating. $\boldsymbol{\Gamma}$ is an indicator matrix that tells us where values need to be updated, when $|\boldsymbol{S}^r - \boldsymbol{S}^h| > \eta_2$, $\boldsymbol{\Gamma}_{ij} = 1$; otherwise $\boldsymbol{\Gamma}_{ij} = 0$. $\boldsymbol{\Lambda}_{ij} = \boldsymbol{S}^r_{ij} \oplus \boldsymbol{S}^h_{ij}$, $\oplus$ is XOR operation. If $\boldsymbol{\Gamma}_{ij} = 1$ and $\boldsymbol{\Lambda}_{ij} = 0$, $\boldsymbol{S}_{ij}$ is weighted by $\boldsymbol{S}^r_{ij}$ and $\boldsymbol{S}^h_{ij}$; if $\boldsymbol{\Gamma}_{ij} = 0$ and $\boldsymbol{\Lambda}_{ij} = 0$, then $\boldsymbol{S}_{ij} = \boldsymbol{S}^r_{ij}$; the rest of $\boldsymbol{S}$ is consistent with 0.

Table 4. Indicator matrices truth table for dual instruction fusion updating strategy.

$\boldsymbol{\Lambda}_{ij}$	$\boldsymbol{\Gamma}_{ij}$	Update	Rules
0	0		$\boldsymbol{S}^r_{ij}$
0	1	✓	$\alpha_2 \boldsymbol{S}^r_{ij} + (1-\alpha_2)\boldsymbol{S}^h_{ij}$
1	0	✓	0
1	1	✓	0

3.5. MGSP for Hash Code Learning

To enhance the cross-modal similarity consistency preserving for hash code learning, a novel technique, called Multi-Grained Similarity Preserving (MGSP) is developed, as shown in Figure 3a. It consists two key sub-modules, i.e., global similarity aggregation (GSA) and local similarity interaction (LSI), which explore multi-grained similarity information: global and local similarity structure information. Furthermore, another sub-module, named Similarity Consistency Reconstruction (SCR) is involved to narrow heterogeneity gap between original features and reconstructed features from hash codes.

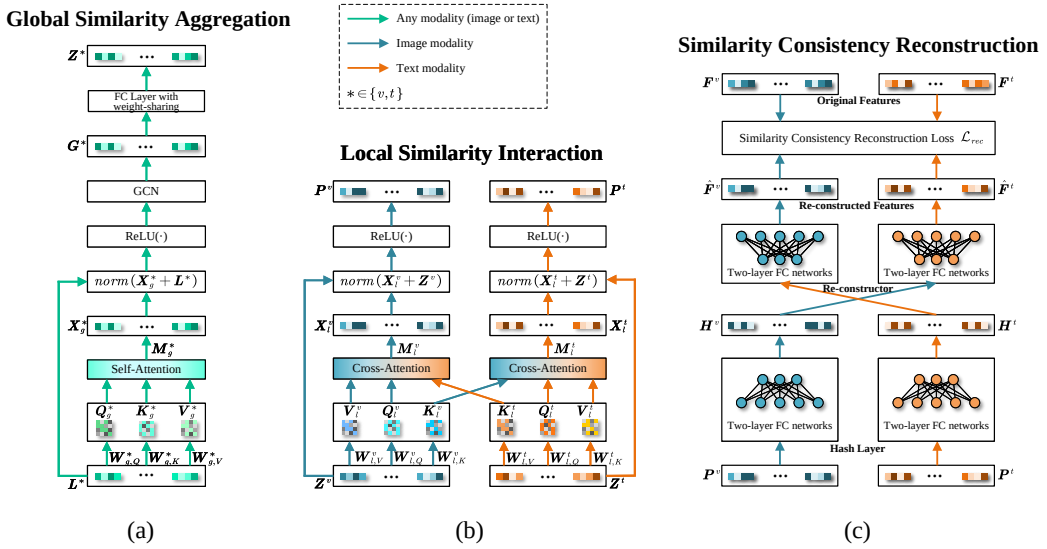


Figure 3. The details of MGSP. (**a**–**c**) illustrate the pipeline of global similarity aggregation, local similarity interaction, and similarity consistency reconstruction, respectively. Best view in color.

3.5.1. Global Similarity Aggregation

The similarity relationship between instances of any modality (image or text) is essential for similarity preserving, which reflects the latent feature distributions of each kind of data. As we known, the intra-modal similarity relationships can be represented as a graph therein each node is an instance and the similarity relationships are represented by edges. In such a graph, each node may be related to others through complex linked structure that can be captured to learn latent global similarity relationships within each modality. To this end, we developed a global similarity aggregation (GSA) sub-module involving the following process: two graphs firstly are constructed, each per modality. Then, we aggregate global similarity information of each node from its neighbors via a graph convolutional network (GCN). Specifically, let $* \in \{v, t\}$, for a batch of samples $\boldsymbol{F}^*$, we feed them into an encoder to generate the latent representations $\boldsymbol{L}^*$. To further focus on the crucial features, the key-value self-attention mechanism is used to obtain global similarity aggregation representations $\boldsymbol{X}_g^*$. The *query* $\boldsymbol{Q}_g^*$, *key* $\boldsymbol{K}_g^*$ and *value* $\boldsymbol{V}_g^*$ are calculated as:

$$\boldsymbol{Q}_g^* = \boldsymbol{L}^* \boldsymbol{W}_{g,Q}^*, \quad \boldsymbol{K}_g^* = \boldsymbol{L}^* \boldsymbol{W}_{g,K}^*, \quad \boldsymbol{V}_g^* = \boldsymbol{L}^* \boldsymbol{W}_{g,V}^*, \tag{7}$$

where $\boldsymbol{W}_{g,Q}^*, \boldsymbol{W}_{g,K}^*, \boldsymbol{W}_{g,V}^*$ are the learnable weight matrices. The attention map $\boldsymbol{M}_g^*$ is calculated as:

$$\boldsymbol{M}_g^* = softmax\left(\frac{\boldsymbol{Q}_g^* (\boldsymbol{K}_g^*)^T}{\sqrt{d}}\right). \tag{8}$$

Thus, the global similarity enhanced representations is $\boldsymbol{X}_g^* = \boldsymbol{M}_g^* \boldsymbol{V}_g^*$. To avoid the issue of smaller feature value caused by attention weighting, we add the latent representation $\boldsymbol{L}^*$ with $\boldsymbol{X}_g^*$, then normalize it as the input of the GCN:

$$\boldsymbol{G}^{*(0)} = \sigma\Big(norm(\boldsymbol{L}^* + \boldsymbol{X}_g^*)\Big). \tag{9}$$

where the superscript (0) denotes the 0-th layer of convolution, namely the input of GCN, $norm(\cdot)$ is normalization function, $\sigma(\cdot)$ represents a nonlinear activation function, usually the $ReLU(\cdot)$ function. The layer-by-layer propagation rule of GCN is formulated as:

$$\mathcal{G}^{*(l)} = \sigma\Big(\tilde{\boldsymbol{D}}^{-\frac{1}{2}} \tilde{\boldsymbol{A}} \tilde{\boldsymbol{D}}^{-\frac{1}{2}} \mathcal{G}^{*(l-1)} \boldsymbol{W}^{*(l)}\Big), \tag{10}$$

where $\tilde{\boldsymbol{D}}_{ii} = \sum_j \tilde{\boldsymbol{A}}_{ij}$, $\tilde{\boldsymbol{A}}$ is an adjacency matrix constructed from the refined similarity matrix $\boldsymbol{S}^r$ via kNN algorithm, k represents the number of neighbors, $\boldsymbol{G}^{*(l-1)}$ and $\boldsymbol{G}^{*(l)}$ represent the input and output of the l-th layer, $\boldsymbol{W}^{*(l)}$ denotes the learnable parameters of the l-th layer, $\sigma(\cdot)$ is $ReLU(\cdot)$ function. The final outputs of the GCN are denoted by $\boldsymbol{G}^*$. Finally, to further mitigate the cross-modal heterogeneity, a two-layer fully-connected network with weigh-sharing is integrated on the top of GCN, and its output are denoted as $\boldsymbol{Z}^*$.

3.5.2. Local Similarity Interaction

Capturing rich similarity structure information among each image-text pair $(\boldsymbol{v}_i, \boldsymbol{t}_i)$ is another desiderata to further enhance cross-modal similarity preserving. This kind of information is mainly reflected in the latent local feature relationships between modalities. To this end, we develop a local similarity interaction (LSI) sub-module to improve similarity consistency preserving within each image-text pair. Specifically, a key-value cross-attention mechanism is used on the top of GSA sub-module to learn fine-grained cross-modal similarity associations. We obtain *query* $\boldsymbol{Q}_l^v$ and $\boldsymbol{Q}_l^t$, *key* $\boldsymbol{K}_l^v$ and $\boldsymbol{K}_l^t$, as well as *value* $\boldsymbol{V}_l^v$ and $\boldsymbol{V}_l^t$ through linear transformation to obtain the visual and textual attention map $\boldsymbol{M}_l^v$ and $\boldsymbol{M}_l^t$:

$$\begin{aligned} \boldsymbol{Q}_l^v &= \boldsymbol{Z}^v \boldsymbol{W}_{l,Q}^v, \quad \boldsymbol{K}_l^v = \boldsymbol{Z}^v \boldsymbol{W}_{l,K}^v, \quad \boldsymbol{V}_l^v = \boldsymbol{Z}^v \boldsymbol{W}_{l,V}^v, \\ \boldsymbol{Q}_l^t &= \boldsymbol{Z}^t \boldsymbol{W}_{l,Q}^t, \quad \boldsymbol{K}_l^t = \boldsymbol{Z}^t \boldsymbol{W}_{l,K}^t, \quad \boldsymbol{V}_l^t = \boldsymbol{Z}^t \boldsymbol{W}_{l,V}^t. \end{aligned} \tag{11}$$

$$\begin{aligned} \boldsymbol{M}_l^v &= softmax\left(\frac{\boldsymbol{Q}_l^v (\boldsymbol{K}_l^t)^T}{\sqrt{d}}\right), \\ \boldsymbol{M}_l^t &= softmax\left(\frac{\boldsymbol{Q}_l^t (\boldsymbol{K}_l^v)^T}{\sqrt{d}}\right). \end{aligned} \tag{12}$$

Thus, the attention-weighted representations are $\boldsymbol{X}_l^v = \boldsymbol{M}_l^v \boldsymbol{V}_l^v$ and $\boldsymbol{X}_l^t = \boldsymbol{M}_l^t \boldsymbol{V}_l^t$. Similar to the above self-attention, the local are calculated as follows:

$$\begin{aligned} \boldsymbol{P}^v &= \sigma(norm(\boldsymbol{X}_l^v + \boldsymbol{Z}^v)), \\ \boldsymbol{P}^t &= \sigma\big(norm(\boldsymbol{X}_l^t + \boldsymbol{Z}^t)\big), \end{aligned} \tag{13}$$

where $\sigma(\cdot)$ is still $ReLU(\cdot)$.

3.5.3. Similarity Consistency Reconstruction

The similarity consistency preserving should not only be reflected in cross-modal features, but more importantly, in hash codes. Deep feature reconstruction, namely reducing the heterogeneity between original features and reconstructed features from continuous hash codes, has been verified to be an effective technique. Current literatures indicate that, two different reconstruction strategies, i.e., intra-modal [28] and inter-modal reconstruction [37], are developed for similarity consistency preserving. In this work, we integrate inter-modal reconstruction into our method to narrow cross-modal gap. Specifically, the continuous hash codes $\boldsymbol{H}^v$ and $\boldsymbol{H}^t$ are generated by a couple of two-layer fully-connected networks from $\boldsymbol{P}^v$ and $\boldsymbol{P}^t$, which then are fed into two re-constructors to output reconstructed features. Accordingly, inspired by [37], a similarity consistency reconstruction loss function $\mathcal{L}_{rec}$ is introduced to preserve the similarity structure information into hash codes:

$$\mathcal{L}_{rec} = \left\|\hat{\boldsymbol{F}}^v - \boldsymbol{F}^v\right\|_F^2 + \left\|\hat{\boldsymbol{F}}^t - \boldsymbol{F}^t\right\|_F^2, \tag{14}$$

where $\hat{\boldsymbol{F}}^v$ and $\hat{\boldsymbol{F}}^t$ denote the reconstructed feature representations.

3.5.4. Hash Code Learning Loss

Furthermore, to preserve similarity consistency of cross-modal hash codes from both intra-modal and inter-modal perspectives, we introduce hash code similarity consistency loss:

$$\begin{aligned}\mathcal{L}_{mod} =& \left\| cos(\boldsymbol{H}^v, \boldsymbol{H}^v) - cos(\boldsymbol{H}^t, \boldsymbol{H}^t) \right\|_F^2 \\ &+ \left\| cos(\boldsymbol{H}^v, \boldsymbol{H}^t) - cos(\boldsymbol{H}^t, \boldsymbol{H}^t) \right\|_F^2 \\ &+ \left\| cos(\boldsymbol{H}^v, \boldsymbol{H}^t) - cos(\boldsymbol{H}^v, \boldsymbol{H}^v) \right\|_F^2.\end{aligned} \tag{15}$$

Meanwhile, to ensure the accuracy of the similarity matrix, we designed a new loss function $\mathcal{L}_{sim}$ as follows:

$$\begin{aligned}\mathcal{L}_{sim} =& \left\| \boldsymbol{S} - cos(\boldsymbol{H}^v, \boldsymbol{H}^v) \right\|_F^2 \\ &+ \left\| \boldsymbol{S} - cos(\boldsymbol{H}^t, \boldsymbol{H}^t) \right\|_F^2 \\ &+ \left\| \boldsymbol{S} - cos(\boldsymbol{H}^v, \boldsymbol{H}^t) \right\|_F^2 \\ &+ \boldsymbol{\Gamma} \left\| \boldsymbol{S} - \boldsymbol{S}^r \right\|_F^2,\end{aligned} \tag{16}$$

where $\boldsymbol{\Gamma}$ is an indicator function, when $\left| \boldsymbol{S}_{ij}^r - \boldsymbol{S}_{ij}^h \right| > \eta_2$, $\Gamma_{ij} = 1$, otherwise $\Gamma_{ij} = 0$.

3.6. Hashing Function Learning

Using the semantic similarity matrix $\boldsymbol{S}$ and the hash codes $\boldsymbol{H}^v, \boldsymbol{H}^t$, we learn two modality-specific hash functions $sgn(f^v(.;\boldsymbol{\theta}_v))$ and $sgn(f^t(.;\boldsymbol{\theta}_t))$ to project deep features $\boldsymbol{F}^v$ and $\boldsymbol{F}^t$ to Hamming space. Specifically, to preserving the similarity structure between hash codes and cross-modal features, we introduce similarity loss function $\mathcal{L}_{f_1}$ as follows:

$$\begin{aligned}\mathcal{L}_{f_1} =& \left\| \boldsymbol{S} - cos(f^v(\boldsymbol{F}^v;\boldsymbol{\theta}_v), f^v(\boldsymbol{F}^v;\boldsymbol{\theta}_v)) \right\|_F^2 \\ &+ \left\| \boldsymbol{S} - cos(f^t(\boldsymbol{F}^t;\boldsymbol{\theta}_t), f^t(\boldsymbol{F}^t;\boldsymbol{\theta}_t)) \right\|_F^2 \\ &+ \left\| \boldsymbol{S} - cos(f^v(\boldsymbol{F}^v;\boldsymbol{\theta}_v), f^t(\boldsymbol{F}^t;\boldsymbol{\theta}_t)) \right\|_F^2,\end{aligned} \tag{17}$$

where $f^v(\boldsymbol{F}^v;\boldsymbol{\theta}_v)$ and $f^t(\boldsymbol{F}^t;\boldsymbol{\theta}_t)$ denote the relaxed hash codes generated by the hash functions, $\boldsymbol{\theta}_v$ and $\boldsymbol{\theta}_t$ denote the parameters. In addition, in order to numerically align the hash codes generated by the hash function with the hash codes $\boldsymbol{H}^v$ and $\boldsymbol{H}^t$ obtained during the training process, we also introduced loss function $\mathcal{L}_{f_2}$:

$$\begin{aligned}\mathcal{L}_{f_2} =& \left\| \boldsymbol{H}^v - f^v(\boldsymbol{F}^v;\boldsymbol{\theta}_v) \right\|_F^2 \\ &+ \left\| \boldsymbol{H}^t - f^t(\boldsymbol{F}^t;\boldsymbol{\theta}_t) \right\|_F^2 \\ &+ \left\| f^v(\boldsymbol{F}^v;\boldsymbol{\theta}_v) - f^t(\boldsymbol{F}^t;\boldsymbol{\theta}_t) \right\|_F^2.\end{aligned} \tag{18}$$

Finally, the quantization loss $\mathcal{L}_{f_3}$ transfers semantic information from relaxed hash codes to binary hash codes:

$$\mathcal{L}_{f_3} = \left\| \boldsymbol{H}^v - sign(\boldsymbol{H}^v) \right\|_F^2 + \left\| \boldsymbol{H}^t - sign\left(\boldsymbol{H}^t\right) \right\|_F^2. \tag{19}$$

3.7. Optimization

The learning process of MGSPU can be divided into two stages: (1) hash code learning stage, and (2) hash function learning stage. During the hash code learning stage, the optimization of the objective function $\mathcal{L}_{code}$ is performed by minimizing Equations (14)–(16):

$$\min_{\boldsymbol{\theta}^c} \quad \mathcal{L}_{code} = \lambda_1 \mathcal{L}_{rec} + \lambda_2 \mathcal{L}_{mod} + \lambda_3 \mathcal{L}_{sim}. \tag{20}$$

During the hash function learning stage, the optimization of the objective function $\mathcal{L}_{hash}$ entails minimizing Equations (17)–(19):

$$\min_{\boldsymbol{\theta}^v, \boldsymbol{\theta}^t} \quad \mathcal{L}_{hash} = \beta_1 \mathcal{L}_{f_1} + \beta_2 \mathcal{L}_{f_2} + \beta_3 \mathcal{L}_{f_3}. \tag{21}$$

The entire algorithm process of MGSPU is outlined in Algorithm 1, which is implemented by Adam adaptive algorithm [54].

Algorithm 1 Algorithm of MGSPU.

Input: Training set size n; number of epochs e; hash code length C; batch size m; learning rate of the network lr_{code} and lr_{hash}; hyperparameter $\alpha_1, \alpha_2, \eta_1, \eta_2, k, \lambda_1, \lambda_2, \lambda_3, \beta_1, \beta_2, \beta_3$;
Output: Network parameters for the optimal hash function learning part;

Initialize network parameters $\boldsymbol{\theta}^c, \boldsymbol{\theta}^v, \boldsymbol{\theta}^t$;
Construct a unified semantic matrix $\boldsymbol{S}^r$ using image features $\boldsymbol{F}^v$ and text features $\boldsymbol{F}^t$;
repeat
  Iterative training e times;
  for $i = 1, 2, ..., \lceil \frac{n}{m} \rceil$ **do**
    Arbitrarily select m training data;
    Generate continuous hash codes $\boldsymbol{H}^v, \boldsymbol{H}^t$ by forward propagation algorithm;
    Generate the semantic matrix $\boldsymbol{S}^h$ through the hash code $\boldsymbol{H}^v, \boldsymbol{H}^t$, and update the original semantic matrix $\boldsymbol{S}^r$ to generate a unified semantic matrix $\boldsymbol{S}$;
    Calculate the loss function Equations (14)–(16) and update the parameters through backpropagation $\boldsymbol{\theta}^c$;
  end for
  for $i = 1, 2, ..., \lceil \frac{n}{m} \rceil$ **do**
    Select m training image-text pairs and hash code $\boldsymbol{H}^v, \boldsymbol{H}^t$;
    Map the original feature $\boldsymbol{F}^v, \boldsymbol{F}^t$ into a hash code through f^v and f^t;
    Calculate the loss Equations (17)–(19), and update the parameter $\boldsymbol{\theta}^v, \boldsymbol{\theta}^t$ with backpropagation;
  end for
until Convergence.

4. Experiments

This section presents experiments and analysis to assess the retrieval performance of the proposed method. We begin by introducing the experimental settings, including datasets, evaluation metrics, baselines, and implementation details. Subsequently, we provide a performance comparison between our method and several baselines, along with an ablation analysis to validate the impact of each component.

4.1. Datasets

We conduct thorough experiments on two prominent multimedia benchmark datasets, namely MIRFLICKR-25K [55] and NUS-WIDE [56], widely employed for cross-modal retrieval evaluation. A concise introduction to these datasets is provided below.

MIRFLICKR-25K. The MIRFLICKR-25K dataset includes 25,000 image-text pairs from the popular photo-sharing platform Flickr. Each image is accompanied by multiple text labels. In our experiment, we specifically selected instances with at least 20 text tags. With AlexNet [50], we converted each image into a depth feature of 4096 dimensions, while the text labels were converted into a BoW [51] vector of 1386 dimensions. In addition, each instance is manually annotated with at least one of 24 unique tags. We experimented with 20,015 examples selected from the dataset.

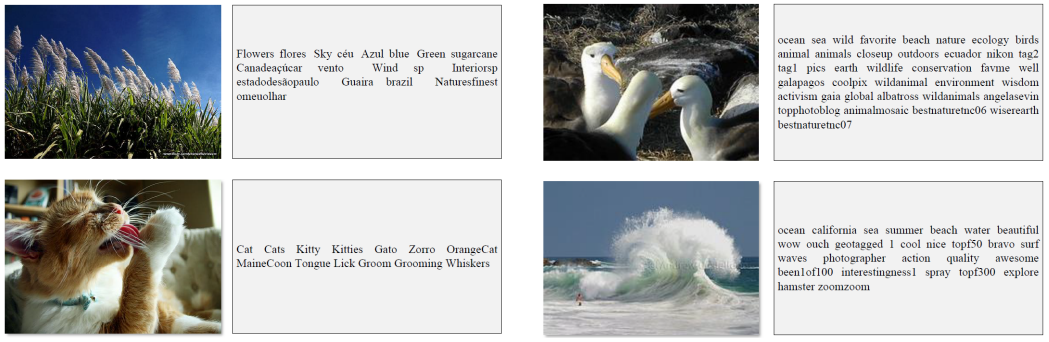
NUS-WIDE. The NUS-WIDE dataset is a substantial real-world web image collection, featuring more than 269,000 images accompanied by over 5000 user-provided tags and 81 concepts across the entire dataset. Using AlexNet [50], each image instance is represented as a 4096-dimensional deep feature, while the textual content is condensed into

a 1000-dimensional BoW [51] vector. For our experiment, we excluded instances lacking labels and focused on those associated with the 10 most frequent categories, resulting in a curated set of 186,577 image-text pairs.

Table 5 presents the statistics of the above two datasets, and some samples of these two datasets are shown in Figure 4.

Table 5. Details about the datasets used in our experiments.

Dataset	Train	Query	Retrieval	Text Feature
MIRFLICKR-25K	5000	2000	18,015	1386 d
NUS-WIDE	5000	2000	184,577	1000 d



(**a**) (**b**)

Figure 4. Some examples from the MIRFLICKR-25K and NUS-WIDE datasets. (**a**) MIRFLICKR-25K. (**b**) NUS-WIDE.

4.2. Evaluation Metrics

In our experiments, we conducted two types of cross-modal retrieval tasks: retrieving texts using image queries (denoted as "I2T") and retrieving images using text queries (denoted as "T2I"). Next, we utilized two standard hashing performance protocols, Hamming ranking and hash lookup [57], to assess the effectiveness of our method and its competitors.For the Hamming ranking protocol, we utilized mean average precision (mAP) to measure accuracy, while precision-recall curves (P-R curves) were employed for the Hash lookup protocol. For mAP and P-R curves, we considered images and texts to be similar if they shared at least one label; otherwise, they were considered dissimilar. Specifically, given a query q_i, the average precision (AP) of the top-N results is defined as:

$$AP(q_i) = \frac{1}{N}\sum_{r=1}^{R} p(r)d(r), \tag{22}$$

where N is the number of relevant instances in the result set, R represents the total amount of data. $p(r)$ denotes the precision of the top- r results. If the r-th retrieved result is relevant to the query instances, $d(r) = 1$; otherwise, $d(r) = 0$. The mAP value is defined as the average AP across all queries q_i:

$$mAP = \frac{1}{M}\sum_{i=1}^{M} AP(q_i), \tag{23}$$

where M represents the number of queries.

4.3. Baselines and Implementation Details

Baselines. we compare the proposed MGSPU method with nine baselines, including CMFH [31], DBRC [58], UDCMH [35], DJSRH [12], JDSH [36], DSAH [37], AGCH [26], DAEH [27], DRNPH [49], which are briefly described as follows:

- CMFH: This method learns uniform binary feature vectors for different modalities through collective matrix factorization of latent factor models.
- DBRC: This approach proposes a deep binary reconstruction model to preserve inter-modal correlation.
- UDCMH: This method utilizes deep learning and matrix factorization with binary latent factor models for multi-modal data search.
- DJSRH: This approach integrates original neighborhood information from different modalities into a joint-semantics affinity matrix to extract latent intrinsic semantic relations.
- JDSH: This method introduces a distribution-based similarity decision and weighting scheme for generating a more discriminative hash code.
- DSAH: This approach explores similarity information across modalities and incorporates a semantic-alignment loss function to align features' similarities with those between hash codes.
- AGCH: This method utilizes GCNs to uncover semantic structures, coupled with a fusion module for correlating different modalities.
- DAEH: This approach attempts to train hash functions with discriminative similarity guidance and an adaptively-enhanced optimization strategy.
- DRNPH: This method implements unsupervised deep relative neighbor relationship preserving cross-modal hashing for achieving cross-modal retrieval in a common Hamming space.

Except for CMFH, All other approaches use deep features to generate cross-modal hash codes.

Implementation Details. As discussed above, the learning process is divided into two stages. In hash code learning stage, three hyperparameters $\lambda_1, \lambda_2, \lambda_3$ are used to weight $\mathcal{L}_{rec}, \mathcal{L}_{mod}, \mathcal{L}_{sim}$, respectively. In hash function learning stage, three other hyperparameter $\beta_1, \beta_2, \beta_3$ are used to adjust the ratio between $\mathcal{L}_{f_1}, \mathcal{L}_{f_2}, \mathcal{L}_{f_3}$. On the MIRFLICKR-25K dataset, we set $\lambda_1 = 0.1, \lambda_2 = 1, \lambda_3 = 10, \beta_1 = 1, \beta_2 = 0.01, \beta_3 = 1$. On NUS-WIDE dataset, we set $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 10, \beta_1 = 10, \beta_2 = 0.01, \beta_3 = 10$. In the process of semantic similarity matrix construction, we set $\alpha_1 = 0.6, \eta_1 = 0.8, \alpha_2 = 0.4, \eta_2 = 0.7$ on the MIRFLICKR-25K, and set $\alpha_1 = 0.4, \eta_1 = 0.5, \alpha_2 = 0.7, \eta_2 = 0.6$ on the NUS-WIDE. In GSA sub-module, kNN algorithm is used to aggregate nodes in a certain neighborhood for each modality. We set $k = 40$ and $k = 60$ for MIRFLICKR-25K and NUS-WIDE, respectively. The optimization algorithm used is the Adam optimization algorithm [54]. For MIRFLICKR-25K, we set the learning rates for hash code learning and hash function learning to 0.001 and 0.0001, respectively, and for NUS-WIDE, both are set to 0.0001. The batch size is consistently set at 512. The number of iterations is defined as 60 for MIRFLICKR-25K and 100 for NUS-WIDE. It's worth noting that, under the same experimental setup, we directly utilize mAP@50 results provided in the original papers of the baseline methods.

Experimental Environment. All the experiments are performed on a workstation with Intel(R) Core i9-12900K 3.9 GHz CPU, 128 GB RAM, 1 TB SSD storage, 2TB HDD storage, and 1 NVIDIA GeForce RTX 3090Ti GPU with ubuntu-22.04.1 operating system. All the techniques are implemented by Python 3.9 on PyTorch 2.0.1.

4.4. Performance Evaluation

We compare the proposed method with nine baselines on MIRFLICKR-25K and NUS-WIDE datasets. The performance of all these methods are evaluated by Hamming Ranking protocol and hash lookup protocol. Tables 6 and 7 illustrates the mAP@50 results of our method and the competitors varying hash code lengths (16, 32, 64, 128 bits) on MIRFLICKR-

25K and NUS-WIDE. Figures 5 and 6 show the P-R curves on these two datasets in various code length. The detailed analysis and observation are presented as follows.

Table 6. mAP@50 score of our method and the baselines at various code lengths (bits) on MIRFLICKR-25K.

Methods	MIRFLICKR-25K							
	I2T				T2I			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CMFH	0.621	0.624	0.625	0.627	0.462	0.662	0.676	0.685
DBRC	0.617	0.619	0.620	0.621	0.618	0.622	0.626	0.628
UDCMH	0.689	0.698	0.714	0.717	0.692	0.704	0.718	0.733
DJSRH	0.810	0.843	0.862	0.876	0.786	0.822	0.835	0.847
JDSH	0.832	0.853	0.882	0.892	0.825	0.864	0.878	0.880
DSAH	0.863	0.877	0.895	0.903	0.846	0.860	0.881	0.882
AGCH	0.865	0.887	0.892	0.912	0.829	0.845	0.852	0.880
DAEH	0.812	0.835	0.847	0.845	0.778	0.819	0.825	0.831
DRNPH	0.876	0.902	0.914	0.933	0.860	0.872	0.885	0.897
Ours	**0.898**	**0.915**	**0.927**	**0.936**	**0.876**	**0.883**	**0.889**	**0.900**

Table 7. mAP@50 score of our method and the baselines at various code lengths (bits) on NUS-WIDE.

Methods	NUS-WIDE							
	I2T				T2I			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CMFH	0.455	0.459	0.465	0.467	0.529	0.577	0.614	0.652
DBRC	0.424	0.459	0.447	0.447	0.455	0.459	0.468	0.473
UDCMH	0.511	0.519	0.524	0.558	0.637	0.653	0.695	0.716
DJSRH	0.724	0.773	0.798	0.817	0.712	0.744	0.771	0.789
JDSH	0.736	0.793	0.832	0.835	0.721	0.785	0.794	0.804
DSAH	0.775	0.805	0.818	0.827	0.770	**0.790**	0.804	**0.815**
AGCH	0.809	**0.830**	0.831	0.852	0.769	0.780	0.798	0.802
DAEH	0.766	0.789	0.809	0.822	0.718	0.751	0.766	0.767
DRNPH	0.790	0.811	0.826	0.837	0.780	0.795	0.804	0.811
Ours	**0.811**	0.826	**0.844**	**0.858**	**0.780**	0.786	**0.806**	0.813

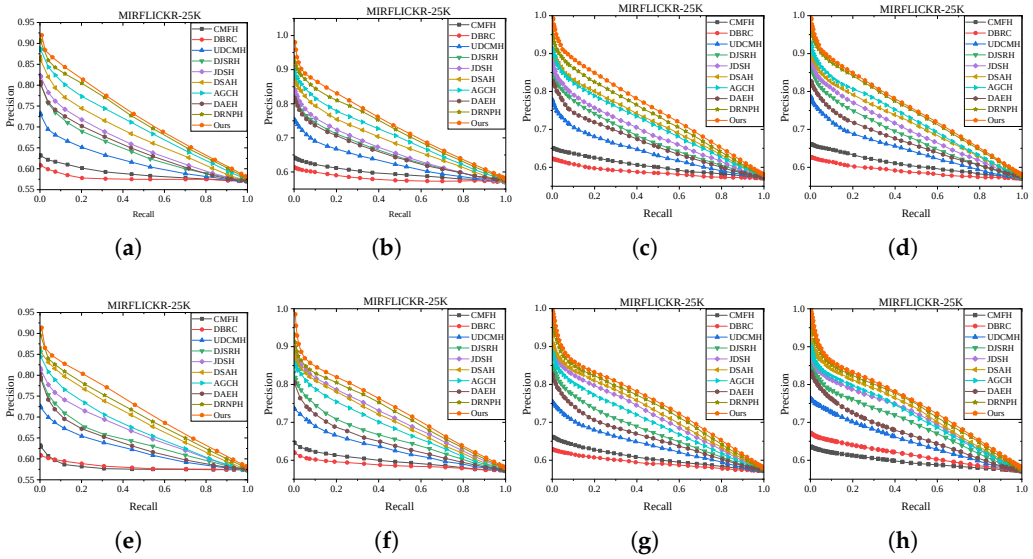


Figure 5. P-R curves of different models at different lengths on MIRFLICKR-25K dataset. (**a**) I2T:16bit. (**b**) I2T:32bit. (**c**) I2T:64bit. (**d**) I2T:128bit. (**e**) T2I:16bit. (**f**) T2I:32bit. (**g**) T2I:64bit. (**h**) T2I:64bit.

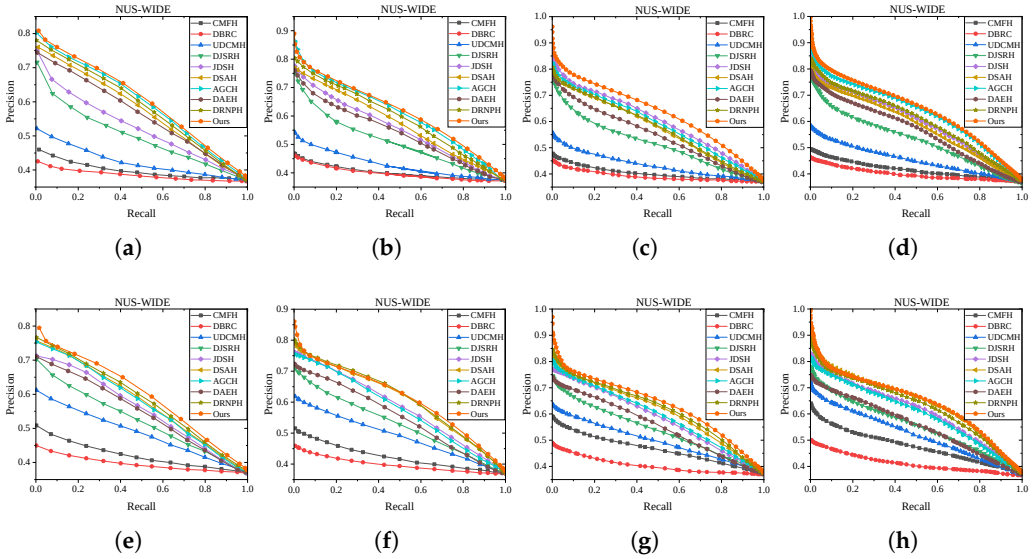


Figure 6. P-R curves of different models at different lengths on MIRFLICKR-25K dataset. (**a**) I2T:16bit. (**b**) I2T:32bit. (**c**) I2T:64bit. (**d**) I2T:128bit. (**e**) T2I:16bit. (**f**) T2I:32bit. (**g**) T2I:64bit. (**h**) T2I:128bit.

Hamming Ranking. It is clearly from the Tables 6 and 7 that the proposed method performs better than the baselines. Specifically, on MIRFLICKR-25K, we found that our method achieved the highest mAP@50 score for both retrieval tasks (I2T: mAP@50 = 0.898 (16 bits), mAP@50 = 0.915 (32 bits), mAP@50 = 0.927 (64 bits), mAP@50 = 0.936 (128 bits); T2I: mAP@50 = 0.876 (16 bits), mAP@50 = 0.883 (32 bits), mAP@50 = 0.889 (64 bits), mAP@50 = 0.900 (128 bits)). For example, our method beats out the strongest competitor, DRNPH, by a significant margin on both two tasks, especially in shorter hash code length: 0.022 (16 bits), 0.013 (32 bits), 0.013 (64 bits) on I2T task, and 0.016 (16 bits), 0.011 (32 bits) on T2I task. The reason behind these results is obvious: with the engagement of the proposed similarity matrix updating strategy, our method can gradually eliminate the noise of the original features used in the similarity relationship construction so as to improve similarity consistency preserving, which is unfortunately ignored by DRNPH. In all but a few case (32 bits code length on I2T and I2T task, MGSPU was defeated by AGCH and DSAH marginally), our method won the competition again on NUS-WIDE by stand-out performance: mAP@50 = 0.811 (16 bits), 0.826 (32 bits), 0.844 (64 bits), 0.858 (128 bits) on I2T task, and mAP@50 = 0.780 (16 bits), 0.786 (32 bits), 0.806 (64 bits), 0.813 (128 bits) on T2I task. Compared with mainstream solutions, complex similarity correlations can be greatly mined through the semantic similarity matrix update strategy, and the MGSP module further retains the potential similarity structure between data.

Hash Lookup. To comprehensively showcase the comprehensive performance comparison of MGSPU with baselines, we draw P-R curves in Figures 5 and 6 with different code lengths on both two datasets. As expected, in addition to dramatically defeating hand-crafted feature based method CMFH, MGSPU outperforms state-of-the-art competitors DRNPH, DSAH and AGCH on various hash code length. This observation is mainly due to the search performance boosting from the interplay of similarity updating and multi-grained similarity preserving of hash codes.

Discussion. It is no secret that, the main reason of poor performance of early models (such as CMFH and DBRC) is their shallow feature extraction techniques that cannot obtain feature representations with rich semantic information. With the help of powerful deep learning techniques, deep neural networks based methods such as DJSRH, JDSH, AGCH and DAEH achieved good results. Among them, DJSRH is equiped with a reconstruction framework for training, which is more competitive than batch training. AGCH uses GCN to aggregate neighborhood information and enhance feature expression. DAEH leverages teacher networks to enhance weaker hashing networks. However, all of them build sim-

ilarity matrices based on original features, which inevitably bring noises into semantic relationships so as to introduce biases. Furthermore, these methods either maintain local or global similarities to preserve the semantic relationships. For example, the similarity matrix in DJSRH contains redundant information from intra-modal fusion items, while DAEH ignores the semantic relationships of intra-modal details. Comparing with these solutions, therefore, we argue that stepwise denoising through a similarity matrix update strategy can greatly mine complex similarity correlations, thereby generating high-confidence supervision signals. In addition, the MGSP method can effectively improve the hash code quality due to further preserving the potential similar structures within and between modalities. Both the mAP@50 score on the hash ranking protocol and the area under the P-R curves on the hash lookup procotol strongly support our view.

4.5. Ablation Study

To verify the validity of each design in MGSPU, we conducted ablation experiments on the MIRFLICKR-25K and NUS-WIDE datasets, serveral variations were considered for this purpose:

- **MGSPU-1:** it removes semantic similarity matrix updating from MGSPU.
- **MGSPU-2:** it removes the similarity consistency reconstruction from MGSPU.
- **MGSPU-3:** it modifies similarity consistency reconstruction by replacing inter-modal reconstruction with intra-modal reconstruction.
- **MGSPU-4:** it removes the GCN module from MGSPU.

From Tables 8 and 9, the following observations can be obtained: firstly, the comparison of MGSPU-1 with our the full MGSPU method verifies that the proposed dual instruction fusion updating strategy can improve the quality of instance similarity matrix to enhance the retrieval performance. Specifically, the retrieval accuracy of MGSPU-1 for both I2T and T2I task show decrease in some extent: on MIRFLICKR-25K, mAP@50 results of I2T task drop from 0.898 (16 bits), 0.915 (32 bits), 0.927 (64 bits), 0.936 (128 bits) to 0.894 (16 bits), 0.912 (32 bits), 0.924 (64 bits), 0.933 (128 bits), respectively; mAP@50 results of T2I task drop from 0.876 (16 bits), 0.883 (32 bits), 0.889 (64 bits), 0.900 (128 bits) to 0.872 (16 bits), 0.876 (32 bits), 0.883 (64 bits), 0.892 (128 bits), respectively. It indicates that without the semantic similarity matrix updating, complex similarity relationship learning suffers from disturbance by noise. Secondly, we can clearly observe that MGSPU-2 has a remarkably performance degradation compared with the full version of MGSPU. This phenomenon confirms that similarity consistency reconstruction is beneficial to preserve semantic information into hash code. Thirdly, with intra-modal reconstruction, MGSPU-3 performs better than MGSPU-2 especially on long hash codes (e.g., 64 or 128 bits). However, compared with the inter-modal reconstruction used in our method, the performance of MGSPU-3 is slightly weaker, which indicates that the inter-modal reconstruction is more helpful to reduce the heterogeneity between the original feature and the hash code. Lastly, after removing GCN module, MGSPU-4 achieves lower retrieval accuracy than ours. these results show that the structural similarity aggregated from neighborhoods by the GCN module is essential to enrich the similarity relationship information of each instance.

Table 8. mAP@50 score for ablation study on MIRFLICKR-25K.

Methods	MIRFLICKR-25K							
	I2T				T2I			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
MGSPU-1	0.894	0.912	0.924	0.933	0.872	0.876	0.883	0.892
MGSPU-2	0.886	0.902	0.909	0.913	0.849	0.857	0.866	0.876
MGSPU-3	0.897	0.913	0.926	0.934	0.872	0.885	0.889	0.894
MGSPU-4	0.895	0.904	0.924	0.926	0.868	0.873	0.886	0.887
Ours	**0.898**	**0.915**	**0.927**	**0.936**	**0.876**	**0.883**	**0.889**	**0.900**

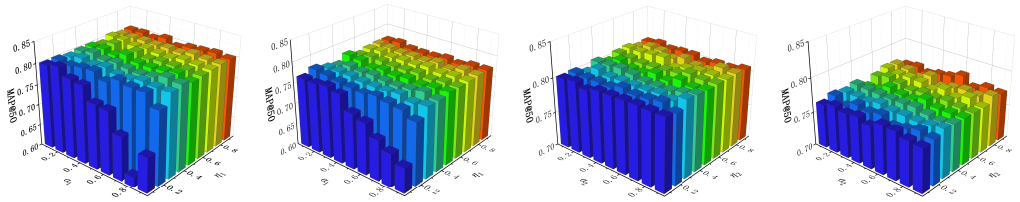
Table 9. mAP@50 score for ablation study on NUS-WIDE.

Methods	NUS-WIDE							
	I2T				T2I			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
MGSPU-1	0.803	0.821	0.837	0.849	0.763	0.783	0.804	0.805
MGSPU-2	0.796	0.809	0.819	0.821	0.746	0.775	0.777	0.776
MGSPU-3	0.801	0.821	0.827	0.851	0.768	**0.791**	0.799	0.812
MGSPU-4	0.794	0.826	0.843	0.847	0.773	0.786	0.798	0.797
Ours	**0.811**	**0.826**	**0.844**	**0.858**	**0.780**	0.786	**0.806**	**0.813**

4.6. Sensitivity to Hyperparameters

This section, we conduct an analysis of the sensitivity of all hyperparameters used in the model: $\alpha_1, \eta_1; \alpha_2, \eta_2; k; \lambda_1, \lambda_2, \lambda_3; \beta_1, \beta_2, \beta_3$. To explore the comprehensive impact of them, the accuracy of I2T and T2I task is used to visualize the trend of cross-modal hash performance. All these analysis are carried out in 16 bit hash code length on NUS-WIDE dataset.

Hyperparameter $\alpha_1, \eta_1, \alpha_2, \eta_2$. In semantic similarity matrix updating module, α_1, η_1 are used to construct the refined similarity matrix $\boldsymbol{S}^r$, α_2, η_2 are used to execute the updating strategy to generate semantic similarity matrix $\boldsymbol{S}$. We observe the performance change of MGSPU by varying α_1, η_1 and α_2, η_2. According to the experimental results in Figure 7, it is clearly that the retrieval accuracy is more susceptible to changes of α_1 when the value of η_1 is small. We speculate that this phenomenon is caused by the noises that are injected in refined similarity matrix when the value of η_1 is small. On the other hand, when we change the α_2 and η_2, the fluctuations in model performance are relatively less severe, which still shows that our model performs a bit better if η_2 is set to a large value (r.g., $\eta_2 = 0.70$). The reason behind this results is understandable: if a larger threshold η_2 is taken, the discrimination of whether $\boldsymbol{S}^r_{ij}$ and $\boldsymbol{S}^h_{ij}$ are dissimilar will be more rigorous. Under this circumstance, the semantic similarity matrix updating will be executed more cautiously to preserve robust.



(**a**) (**b**) (**c**) (**d**)

Figure 7. Sensitivity analysis of α_1, η_1 and α_2, η_2 on NUS-WIDE dataset. (**a**) I2T: α_1 and η_1. (**b**) T2I: α_1 and η_1. (**c**) I2T: α_2 and η_2. (**d**) T2I: α_2 and η_2.

Hyperparameter k. We recorded the performance change by varying the value of k on NUS-WIDE dataset to evaluate the effect by the the number of neighbors in kNN algorithm. As demonstrated in Figure 8, when $k \in [40, 80]$, the curve changes sharply, while the curve changes more modestly in the rest of the interval. We conjecture that when too many or too few neighbors we selected, noise will be introduced into the intra-modal similarity relationship representation, thereby affecting the latent similarity relationship learning within modality. Particularly, if k is set to 60, MGSPU achieves the highest mAP@50 score for both I2T and T2I task. It indicates that by selecting an appropriate number of neighbors, high-quality intra-modal similarity structure information can be aggregated by GCN to improve intra-modal similarity consistency preserving.

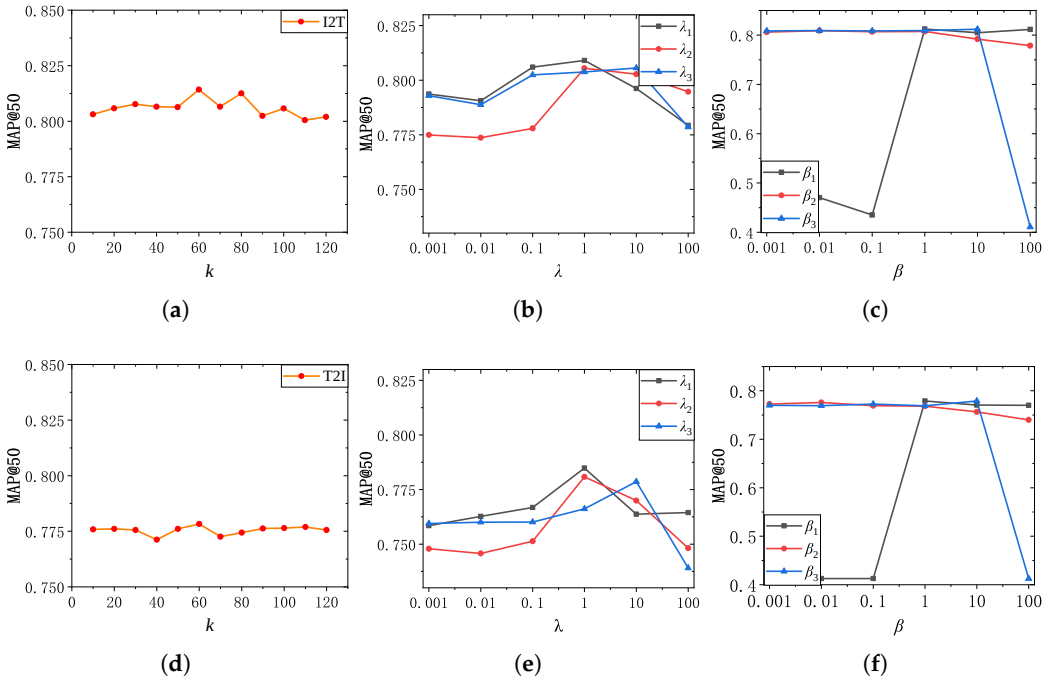


Figure 8. Sensitivity analysis of k; $\lambda_1, \lambda_2, \lambda_3$; $\beta_1, \beta_2, \beta_3$ on NUS-WIDE dataset. (**a**) I2T: k. (**b**) I2T: λ. (**c**) I2T: β. (**d**) T2I: k. (**e**) T2I: λ. (**f**) T2I: β.

Hyperparameter $\lambda_1, \lambda_2, \lambda_3$. As presented in Equation (20), $\lambda_1, \lambda_2, \lambda_3$ are used to balance three components, i.e., $\mathcal{L}_{re}$, $\mathcal{L}_{mod}$ and $\mathcal{L}_{sim}$ of hash code learning loss function. To analyze the effect by these three losses, we recorded the performance change of our method in Figure 8 by varying $\lambda_1, \lambda_2, \lambda_3$ from 0.001 to 100 with a 10-fold increase. It is noteworthy that our method obtains the best performance if we set $\lambda_1 = 1, \lambda_2 = 1, \lambda_3 = 10$. Among these three losses, we found that the new designed loss $\mathcal{L}_{sim}$ has a relatively greater effect on the learning of hash code. We argue that this is mainly due to the indispensability of a reliable similarity matrix for unsupervised hash learning.

Hyperparameter $\beta_1, \beta_2, \beta_3$. As depicted in Equation (21), $\beta_1, \beta_2, \beta_3$ serves as weight factor to balance $\mathcal{L}_{f_1}$, $\mathcal{L}_{f_2}$, and $\mathcal{L}_{f_3}$. The observations in Figure 8 indicates that $\mathcal{L}_{f_1}$ and $\mathcal{L}_{f_3}$ contributes more than $\mathcal{L}_{f_2}$. We infer that although the duty of $\mathcal{L}_{f_2}$ is to ensure numerical consistency between the generated hash codes $\boldsymbol{H}^*$ and trained hash codes and $sgn(f^*(\boldsymbol{F}^*; \boldsymbol{\theta}_*))$, the main goal of cross-modal hash learning is still to eliminate cross-modal heterogeneity, which is only achieved by $\mathcal{L}_{f_1}$. Besides, the quantization error is cannot reduced by other losses but $\mathcal{L}_{f_3}$. In addition, by setting $\beta_1 = 10, \beta_2 = 0.01, \beta_3 = 10$, our method achieves the best performance.

5. Conclusions

In this paper, we propose a novel unsupervised hashing learning framework, called multi-grained similarity preserving and updating to improve cross-modal hashing performance. To obtain a high-confidence similarity matrix, we develop an update strategy that corrects the similarity values after the original feature fusion using a matrix constructed in Hamming space, and a loss function is designed to guide the similarity update. Also to learn high-quality hash codes, we co-model from multiple granularity to preserve semantic correlations within and between modalities. Specifically, GCNs is used to capture the global similarity relationship within each modality, and a cross-attention mechanism is used to perform interactions between modalities to bridge the heterogeneous gap. In addition, deep feature reconstruction further enhances inter-modal correlations and reduces modal gaps.

In our experiments, we use the hamming ranking protocol and the hash lookup protocol to compare with other benchmark models, and the experimental results on both

datasets show that our approach achieves impressive performance. In addition, we set up four different ablation experiments to verify the performance of the designed module, and the results also validate the effectiveness of the designed module.

Although our experimental results achieved excellent results, for text data, we only used a simple bag-of-words model, and the gap between the results of image-retrieval text and text-retrieval image is relatively large. As an exploratory work, we plan to further improve the feature extraction stage in the future to achieve a more accurate alignment of the semantic information of images and text, which will lead to a more balanced cross-modal retrieval.

Author Contributions: R.W. and L.Z. designed the methodology, wrote the original draft, designed and prepared all figures. X.Z. contributed to conceptualization, project administration. X.Z. and L.Z. acquired funding, reviewed and edited the manuscript. R.W. and L.Z. conceived the experiments. R.W., Z.Y. and Y.L. conducted the experiments and acquired experimental results. R.W., L.Z. and Z.Z. analyzed the experimental results. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (62202163, 62072166), the Natural Science Foundation of Hunan Province (2022JJ40190), the Scientific Research Project of Hunan Provincial Department of Education (22A0145), the Key Research and Development Program of Hunan Province (2020NK2033), the Hunan Provincial Department of Education Scientific Research Outstanding Youth Project (21B0200), and the Hunan Provincial Natural Science Foundation Youth Fund Project (2023JJ40333).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: This paper uses the publicly available dataset MIRFLICKR-25K, NUS-WIDE.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, Y. Survey on deep multi-modal data analytics: Collaboration, rivalry, and fusion. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2021**, *17*, 1–25.
2. Zhu, L.; Zhang, C.; Song, J.; Liu, L.; Zhang, S.; Li, Y. Multi-graph based hierarchical semantic fusion for cross-modal representation. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Shenzhen, China, 5–9 July 2021; pp. 1–6.
3. Zhang, B.; Hu, H.; Sha, F. Cross-modal and hierarchical modeling of video and text. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 374–390.
4. Xie, L.; Shen, J.; Zhu, L. Online cross-modal hashing for web image retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30; Number 1.
5. Tian, Y.; Zhou, L.; Zhang, Y.; Zhang, T.; Fan, W. Deep cross-modal face naming for people news retrieval. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 1891–1905. [CrossRef]
6. Zhen, L.; Hu, P.; Wang, X.; Peng, D. Deep supervised cross-modal retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 10394–10403.
7. Wang, K.; Yin, Q.; Wang, W.; Wu, S.; Wang, L. A comprehensive survey on cross-modal retrieval. *arXiv* **2016**, arXiv:1607.06215.
8. Huang, X.; Peng, Y.; Yuan, M. MHTN: Modal-adversarial hybrid transfer network for cross-modal retrieval. *IEEE Trans. Cybern.* **2018**, *50*, 1047–1059. [CrossRef] [PubMed]
9. Yu, T.; Yang, Y.; Li, Y.; Liu, L.; Fei, H.; Li, P. Heterogeneous attention network for effective and efficient cross-modal retrieval. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Montreal, QC, Canada, 11–15 July 2021; pp. 1146–1156.
10. Chun, S.; Oh, S.J.; De Rezende, R.S.; Kalantidis, Y.; Larlus, D. Probabilistic embeddings for cross-modal retrieval. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 8415–8424.
11. Gao, D.; Jin, L.; Chen, B.; Qiu, M.; Li, P.; Wei, Y.; Hu, Y.; Wang, H. Fashionbert: Text and image matching with adaptive loss for cross-modal retrieval. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 2251–2260.
12. Su, S.; Zhong, Z.; Zhang, C. Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 3027–3035.

13. Gu, J.; Cai, J.; Joty, S.R.; Niu, L.; Wang, G. Look, imagine and match: Improving textual-visual cross-modal retrieval with generative models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7181–7189.
14. Cheng, M.; Jing, L.; Ng, M.K. Robust unsupervised cross-modal hashing for multimedia retrieval. *ACM Trans. Inf. Syst. (TOIS)* **2020**, *38*, 1–25.
15. Yao, H.L.; Zhan, Y.W.; Chen, Z.D.; Luo, X.; Xu, X.S. Teach: Attention-aware deep cross-modal hashing. In Proceedings of the z, Taipei, Taiwan, 21–24 August 2021; pp. 376–384.
16. Zhang, C.; Song, J.; Zhu, X.; Zhu, L.; Zhang, S. Hcmsl: Hybrid cross-modal similarity learning for cross-modal retrieval. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2021**, *17*, 1–22. [CrossRef]
17. Jiang, Q.Y.; Li, W.J. Deep cross-modal hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3232–3240.
18. Yang, E.; Deng, C.; Liu, W.; Liu, X.; Tao, D.; Gao, X. Pairwise relationship guided deep hashing for cross-modal retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31; Number 1.
19. Ma, L.; Li, H.; Meng, F.; Wu, Q.; Ngan, K.N. Global and local semantics-preserving based deep hashing for cross-modal retrieval. *Neurocomputing* **2018**, *312*, 49–62. [CrossRef]
20. Shen, X.; Zhang, H.; Li, L.; Yang, W.; Liu, L. Semi-supervised cross-modal hashing with multi-view graph representation. *Inf. Sci.* **2022**, *604*, 45–60.
21. Li, C.; Deng, C.; Li, N.; Liu, W.; Gao, X.; Tao, D. Self-supervised adversarial hashing networks for cross-modal retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–20 July 2018; pp. 4242–4251.
22. Zhang, D.; Wu, X.J.; Yu, J. Label consistent flexible matrix factorization hashing for efficient cross-modal retrieval. *ACM Trans. Multimed. Comput. Commun. Appl. (TOMM)* **2021**, *17*, 1–18. [CrossRef]
23. Chen, Z.D.; Li, C.X.; Luo, X.; Nie, L.; Zhang, W.; Xu, X.S. SCRATCH: A scalable discrete matrix factorization hashing framework for cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 2262–2275. [CrossRef]
24. Hu, P.; Zhen, L.; Peng, D.; Liu, P. Scalable deep multimodal learning for cross-modal retrieval. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 635–644.
25. Dong, X.; Liu, L.; Zhu, L.; Nie, L.; Zhang, H. Adversarial graph convolutional network for cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2021**, *32*, 1634–1645. [CrossRef]
26. Zhang, P.F.; Li, Y.; Huang, Z.; Xu, X.S. Aggregation-based graph convolutional hashing for unsupervised cross-modal retrieval. *IEEE Trans. Multimed.* **2021**, *24*, 466–479.
27. Shi, Y.; Zhao, Y.; Liu, X.; Zheng, F.; Ou, W.; You, X.; Peng, Q. Deep adaptively-enhanced hashing with discriminative similarity guidance for unsupervised cross-modal retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2022**, *32*, 7255–7268.
28. Zhu, L.; Wu, X.; Li, J.; Zhang, Z.; Guan, W.; Shen, H.T. Work together: Correlation-identity reconstruction hashing for unsupervised cross-modal retrieval. *IEEE Trans. Knowl. Data Eng.* **2022**, *35*, 8838–8851. [CrossRef]
29. Li, C.; Deng, C.; Wang, L.; Xie, D.; Liu, X. Coupled cyclegan: Unsupervised hashing network for cross-modal retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, January 27–1 February 2019; Volume 33; Number 01; pp. 176–183.
30. Wang, W.; Shen, Y.; Zhang, H.; Yao, Y.; Liu, L. Set and rebase: Determining the semantic graph connectivity for unsupervised cross-modal hashing. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, YoKohama, Japan, 7–15 January 2021; pp. 853–859.
31. Ding, G.; Guo, Y.; Zhou, J. Collective matrix factorization hashing for multimodal data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 2075–2082.
32. Song, J.; Yang, Y.; Yang, Y.; Huang, Z.; Shen, H.T. Inter-media hashing for large-scale retrieval from heterogeneous data sources. In Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 22–27 June 2013; pp. 785–796.
33. Tu, R.C.; Mao, X.L.; Lin, Q.; Ji, W.; Qin, W.; Wei, W.; Huang, H. Unsupervised Cross-modal Hashing via Semantic Text Mining. *IEEE Trans. Multimed.* **2023**, *25*, 8946–8957. [CrossRef]
34. Zhao, Y.; Zhu, Y.; Liao, S.; Ye, Q.; Zhang, H. Class concentration with twin variational autoencoders for unsupervised cross-modal hashing. In Proceedings of the Asian Conference on Computer Vision, Macau, China, 4–8 December 2022; pp. 349–365.
35. Wu, G.; Lin, Z.; Han, J.; Liu, L.; Ding, G.; Zhang, B.; Shen, J. Unsupervised Deep Hashing via Binary Latent Factor Models for Large-scale Cross-modal Retrieval. In Proceedings of the IJCAI, Stockholm, Sweden, 13–19 July 2018; Volume 1; Number 3; p. 5.
36. Liu, S.; Qian, S.; Guan, Y.; Zhan, J.; Ying, L. Joint-modal distribution-based similarity hashing for large-scale unsupervised deep cross-modal retrieval. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Xi'an, China, 25–30 July 2020; pp. 1379–1388.
37. Yang, D.; Wu, D.; Zhang, W.; Zhang, H.; Li, B.; Wang, W. Deep semantic-alignment hashing for unsupervised cross-modal retrieval. In Proceedings of the 2020 International Conference on Multimedia Retrieval, Dublin, Ireland, 26–29 October 2020; pp. 44–52.

38. Zhang, Z.; Lin, Z.; Zhao, Z.; Xiao, Z. Cross-modal interaction networks for query-based moment retrieval in videos. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2019; pp. 655–664.
39. Wang, Z.; Liu, X.; Li, H.; Sheng, L.; Yan, J.; Wang, X.; Shao, J. Camp: Cross-modal adaptive message passing for text-image retrieval. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Long Beach, CA, USA, 16–20 June 2019; pp. 5764–5773.
40. Yu, Y.; Xiong, Y.; Huang, W.; Scott, M.R. Deformable siamese attention networks for visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, DC, USA, 14–19 June 2020; pp. 6728–6737.
41. Gu, W.; Gu, X.; Gu, J.; Li, B.; Xiong, Z.; Wang, W. Adversary guided asymmetric hashing for cross-modal retrieval. In Proceedings of the 2019 on International Conference on Multimedia Retrieval, Ottawa, Canada, 10–13 June 2019; pp. 159–167.
42. Zhang, D.; Li, W.J. Large-scale supervised multimodal hashing with semantic correlation maximization. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec, QC, Canada, 27–31 July 2014; Volume 28; Number 1.
43. Lin, Z.; Ding, G.; Hu, M.; Wang, J. Semantics-preserving hashing for cross-view retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3864–3872.
44. Cao, Y.; Liu, B.; Long, M.; Wang, J. Cross-modal hamming hashing. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 202–218.
45. Jin, S.; Zhou, S.; Liu, Y.; Chen, C.; Sun, X.; Yao, H.; Hua, X.S. SSAH: Semi-supervised adversarial deep hashing with self-paced hard sample generation. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34; Number 07; pp. 11157–11164.
46. Liu, X.; Yu, G.; Domeniconi, C.; Wang, J.; Ren, Y.; Guo, M. Ranking-based deep cross-modal hashing. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February July 2019; Volume 33; Number 01; pp. 4400–4407.
47. Sun, C.; Latapie, H.; Liu, G.; Yan, Y. Deep normalized cross-modal hashing with bi-direction relation reasoning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 4941–4949.
48. Yu, J.; Zhou, H.; Zhan, Y.; Tao, D. Deep graph-neighbor coherence preserving network for unsupervised cross-modal hashing. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021; Volume 35; Number 5; pp. 4626–4634.
49. Yang, X.; Wang, Z.; Wu, N.; Li, G.; Feng, C.; Liu, P. Unsupervised Deep Relative Neighbor Relationship Preserving Cross-Modal Hashing. *Mathematics* **2022**, *10*, 2644. [CrossRef]
50. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
51. Ko, Y. A study of term weighting schemes using class information for text classification. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, OR, USA, 12–16 August 2012; pp. 1029–1030.
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, June 26–1 July 2016; pp. 770–778.
53. Memory, L.S.T. Long short-term memory. *Neural Comput.* **2010**, *9*, 1735–1780.
54. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
55. Huiskes, M.J.; Lew, M.S. The mir flickr retrieval evaluation. In Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, Vancouver, BC, Canada, 30–31 October 2008; pp. 39–43.
56. Chua, T.S.; Tang, J.; Hong, R.; Li, H.; Luo, Z.; Zheng, Y. 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, Santorini Island, Greece, 8–10 July July 2009; pp. 1–9.
57. Liu, W.; Mu, C.; Kumar, S.; Chang, S.F. Discrete graph hashing. *Adv. Neural Inf. Process. Syst.* **2014**, *27*, 3419–3427.
58. Li, X.; Hu, D.; Nie, F. Deep binary reconstruction for cross-modal hashing. In Proceedings of the 25th ACM International Conference on Multimedia, Mountain View, CA, USA, 23–27 October 2017; pp. 1398–1406.