Supervised Hierarchical Online Hashing for Cross-modal Retrieval

KAI HAN, YU LIU, RUKAI WEI, KE ZHOU, and JINHUI XU, Huazhong University of Science and Technology, China
KUN LONG, Huawei Technologies Co., Ltd., China

Online cross-modal hashing has gained attention for its adaptability in processing streaming data. However, existing methods only define the hard similarity between data using labels. This results in poor retrieval performance, as they fail to exploit the semantic structure information of labels and miss the high-quality hash codes guided by the hierarchical relevance between labels. In addition, they ignore the bit-flipping problem, which leads to sub-optimal cross-modal retrieval performance. To address these issues, we propose Supervised Hierarchical Online Hashing (SHOH) for cross-modal retrieval. Our approach acquires hierarchical similarity via cross-layer affiliation of labels and explores its application to online hashing. We design a hierarchical similarity learning method in the online learning framework, which includes virtual center learning and hierarchical similarity embedding. Labels with soft similarity bridge the label hierarchy and cross-modal hash embedding. Furthermore, we propose a Weighted Retrieval Strategy (WRS) to mitigate the impact caused by bit-flipping errors. Extensive experiments and verification on hierarchical and non-hierarchical datasets demonstrate that SHOH preserves accurate inter-class distances and achieves performance improvements compared to state-of-the-art methods. The source code is available at https://github.com/HUST-IDSM-AI/SHOH.

CCS Concepts: • **Computing methodologies** → **Supervised learning**; • **Information systems** → **Similarity measures;**

Additional Key Words and Phrases: Online hashing, cross-modal retrieval, label hierarchy

ACM Reference format:
Kai Han, Yu Liu, Rukai Wei, Ke Zhou, Jinhui Xu, and Kun Long. 2024. Supervised Hierarchical Online Hashing for Cross-modal Retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 4, Article 103 (January 2024), 23 pages.
https://doi.org/10.1145/3632527

1 INTRODUCTION

The increasing volume of data on the Internet has heightened interest in hashing retrieval, which is one of the major solutions to the approximate nearest neighbor retrieval. By mapping data from

This work is supported by the National Natural Science Foundation of China (Grant No. 62232007, No. 61902135, No. 61821003) and the Natural Science Foundation of Hubei Province (Grant No. 2022CFB060).
Authors' addresses: K. Han, Y. Liu (corresponding author), R. Wei, K. Zhou, and J. Xu, Huazhong University of Science and Technology, Wuhan, China; e-mails: kylehan@hust.edu.cn, liu_yu@hust.edu.cn, weirukai@hust.edu.cn, zhke@hust.edu.cn, xjh@hust.edu.cn; K. Long, Huawei Technologies Co., Ltd., Shenzhen, China; e-mail: longkun2@huawei.com.

1551-6857/2024/01-ART103 $15.00
https://doi.org/10.1145/3632527

high-dimensional feature space to low-dimensional Hamming space, it can effectively tackle the expensive storage and computing overhead caused by processing high-dimensional data. Due to these advantages, hashing has become increasingly popular for similarity search tasks involving multimedia data, e.g., image retrieval [16, 19], video retrieval [12, 25], and audio retrieval [8]. These tasks can be classified into single-modal retrieval [34, 37, 51] and multi-modal retrieval [41, 52]. The former involves only features of a single modality during training and querying, while the latter can further integrate features of various modalities to improve retrieval performance. Their commonality is that the query and retrieval results have at least one of the same modalities.

However, from the perspective of real-world retrieval, it is common for the query and retrieval results to belong to different modalities, e.g., image-text retrieval [33] and video-text retrieval [53]. Collectively, these tasks are known as cross-modal retrieval, which faces challenges of the heterogeneity gap caused by inconsistencies in information density and differences in feature dimensions across different modalities. Hash-based methods, specifically cross-modal hashing [24], are effective in addressing the challenges posed in cross-modal retrieval. Recent studies [17, 44, 49] have provided strong evidence for their suitability. MCSCH [44] utilizes multi-scale features to efficiently discover potential associations in multimedia data. This enhances the learning efficiency of hash models. LFMH [49] captures both intra- and inter-modality similarities by learning modality-specific features. It achieves this while minimizing storage and computational costs by directly leveraging supervised information. MTFH [17] is a pioneering study that successfully learns hash codes for heterogeneous data with varying code lengths. It delivers impressive performance and demonstrates seamless applicability across various settings. Other approaches [4, 42, 46] focus on mapping different modal features into a unified Hamming space to preserve semantic similarity. Furthermore, in application scenarios, hashing holds significant potential to empower online learning, which has broad applications in video delivery systems [1, 10], transaction monitoring systems [6], and recommender systems [22]. Online learning by hashing, i.e., online hashing, focuses on encoding continuously arriving data into hash codes and learning real-time changes in data distribution.

As a result, cross-modal online hashing is gaining popularity due to the above reasons, embedding multimedia streaming data (e.g., images and texts) into similarity hash codes for online learning and retrieval. Existing methods [18, 35, 48] address the problem of inconsistent semantic distribution of rounds during online learning, which lays the foundation for accurate streaming data mapping. However, they suffer from colossal information loss, because they are designed for non-hierarchical data. They can only use non-hierarchical labels, which are either the concatenation of fine and coarse labels or the finest-grained labels. The former results in poor retrieval performance, because it cannot distinguish between hierarchical classes. The latter results in a significant loss of semantic information, as it discards labels other than the finest-grained class. Furthermore, these methods fail to provide fine similarity labels in hash learning, since they only discriminate between two objects based on whether they share the same labels. These hard similarity labels (0 or 1) are coarse for online learning and result in inferior retrieval performance in real-world recommender systems.

We believe that labels with semantic hierarchical relevance also share a fine similarity. As shown in Figure 1, we use pairs of images and texts with labels to map hash codes, where I and T represent the image and text, respectively. The labels for pairs a, b, c, and d are *Gloves*, *Socks*, *Backpack*, and *Tote bag*, respectively. The pair e has the same label as d. Conventional methods consider these labels and concepts equivalent and map them to the Hamming space with almost equal distances. Nevertheless, in the hierarchy, *Accessory* is the parent class for *Gloves* and *Socks*, while *Backpack* and $Totebag$ are child classes for *Bag*, indicating that these labels should maintain different semantic distances. For instance, *Gloves* is close to *Socks* but relatively far from *Backpack*. We can map

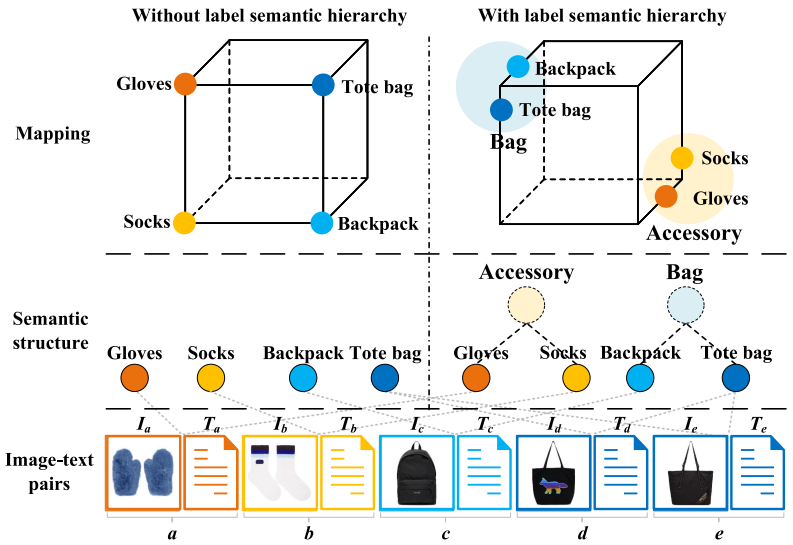

Fig. 1. Diagram of the label hierarchy. The label hierarchy can enhance the preservation of semantic information in the hash code, thereby increasing the accuracy of retrieval results. Mapping using only the finest-grained labels can optimize average Hamming distances. However, this approach may lead to a loss of semantic information.

the parent classes, i.e., *Accessory* and *Bag*, to Hamming space by the longest distances, while their child classes can be mapped within a limited range around them. This yields fine similarities by distinguishing inter-class distances.

To apply hierarchical similarity in online learning, we calculate *soft similarity labels* and *virtual centers* in each round. We first build cross-layer affiliation matrices from top to bottom according to the label set, where each matrix records the affiliation between parent and child labels. Then, to replace traditional hard similarity labels, we calculate similarity based on the hierarchical affiliation relationships, i.e., soft similarity labels. Meanwhile, we learn the virtual centers by cross-layer affiliation matrices, where the center is the hash code mapped from classes, and target hash codes should be close to corresponding centers. Finally, we iterate on updating the virtual centers and target hash codes using the soft similarity labels. With this design, the hash codes learned in each round are compatible with the obtained ones, resulting in overhead related to the size of the current round data instead of the sum of the prior data. Based on these steps, we propose **Supervised Hierarchical Online Hashing (SHOH)**. Compared to state-of-the-art online cross-modal hashing methods, SHOH achieves optimal results in **Mean Average Precision (MAP)** for retrieval.

Our contributions are summarized below.

- SHOH is the first attempt to apply the label semantic hierarchy in cross-modal online hashing.
- We design soft similarity labels and virtual centers to implement a hash learning scheme that adapts to online learning and learns by controlling inter-class distances.
- We design hierarchical center-wise and instance-wise mappings to capture the commonality and individuality of instances in each class, respectively.
- We design a **Weighted Retrieval Strategy (WRS)** to judge the confidence level of each classifier during the retrieval process, which can effectively improve the retrieval performance.

The remainder of this article is structured as follows: Section 2 briefly reviews existing online and closely related hierarchical offline hashing methods. The design of SHOH is introduced in Section 3. Section 4 presents experimental results and analyses. Section 5 discusses and presents other verification on hierarchical and non-hierarchical datasets and introduces our future work. Section 6 presents the conclusion.

2 RELATED WORK

2.1 Online Hashing

Online hashing aims to solve the problem that traditional offline hashing fails to perform incremental learning in streaming data. The earlier research focuses on uni-modal retrieval, i.e., retrieving images similar to the image query. For unsupervised single-modal online hashing methods, representative studies include SketchHash [11] and FROSH [3]. For supervised single-modal online hashing method, representative researches include MIHash [2], HMOH [14], OHWEU [36], FCOH [15], ATHOH [9], and RPT-WOH [7]. However, these methods are designed for single-modal retrieval, unsuitable for multimedia data. Therefore, they can not meet a wide range of retrieval needs.

Multi-modal online hashing can use multi-modal data for retrieval and adapt to the online scene during training. Due to the limited demand for multi-modal retrieval in online scenarios, there are few studies on this type. The existing multi-modal online hashing methods include DMVH [39], FOMH [20], and OASIS [38]. These methods can handle multi-modal data, but they still rely on information from the same modality with the query during retrieval.

In the realm of cross-modal online hashing, except for the change in streaming data distribution, the semantic disparities among heterogeneous modalities pose a new challenge. These challenges are further compounded by the absence of supervision in the learning process. While unsupervised cross-modal online hashing has made progress, including OCMH [40] and OCMFH [31], it struggles to achieve optimal performance. Supervised cross-modal online hashing methods mostly use supervised information to generate unified hash codes to solve heterogeneous gap problems, and their retrieval performance is better than unsupervised methods. Existing studies include OLSH [43], LEMON [35], OLCH [45], ORSCH [13], DOCH [48], and OSCMFH [26]. However, these methods ignore the hierarchy in the label set, resulting in a massive loss of semantic information during online hash learning.

2.2 Hierarchical Offline Hashing

The label hierarchy in the dataset has gained attention in recent years. Several emerging studies have proposed and explored ways to use the information from the label hierarchy, resulting in significant improvements in retrieval performance (including SHDH [30], HiCHNet [29], SHDCH [47], and HSSPH [32]). However, these methods are not specifically designed for online scenarios, making it challenging to meet the requirements of online learning.

SHDCH is an inspiration for the design of SHOH. SHDCH learns hash codes based on label hierarchy information, consisting of similarity at each layer and relatedness across layers. Its overall objective function is defined as:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{C}^k,\mathbf{C}^K} & \sum_{k=1}^{K} \alpha^k (\|r\mathbf{L}^k - \mathbf{F}^\top \mathbf{C}^k\|^2 + \|r\mathbf{L}^k - \mathbf{G}^\top \mathbf{C}^k\|^2) \\ & + \eta \sum_{k=1}^{K-1} \beta^k \|(r\mathbf{A}^{k,K} - (\mathbf{C}^k)^\top \mathbf{C}^K\|^2 + \mu(\|\mathbf{B}-\mathbf{F}\|^2 + \|\mathbf{B}-\mathbf{G}\|^2), \\ s.t. \quad & \mathbf{B} \in \{-1,1\}^{r\times N}, \quad \mathbf{C}^k \in \{-1,1\}^{r\times|\mathbf{C}^k|}, \quad \mathbf{C}^K \in \{-1,1\}^{r\times|\mathbf{C}^K|},\end{aligned} \tag{1}$$

where $\mathbf{A}^{k,K}$ represents the cross-layer affiliation of classes, K is the finest-grained layer. $\mathbf{C}^k$ is the hash representation of the kth layer's classes. $\mathbf{L}^k$ is the kth layer's label matrix. $\mathbf{F}$, $\mathbf{G}$ are the feature matrices for image and text modality, respectively. $\mathbf{B}$ is a matrix of hash codes, each consisting of r bits. N represents the number of instances, and $|\mathbf{C}^k|$ represents the number of classes in the kth layer. α, β, η and μ are parameters.

Although SHOH is inspired by these methods, it differs from them in several significant ways:

(1) These methods are mainly intended for offline scenarios with stationary and unchanging data. Therefore, they are not suitable for handling streaming data, which is common in real-world applications. In contrast, SHOH is specifically designed for processing streaming data and has demonstrated superior performance while maintaining high efficiency to meet the requirements of online scenarios. To generate informative hash codes, SHOH incorporates soft similarity labels and optimization algorithms into the online hash code generation process. To accurately map streaming data, SHOH combines hierarchical center-wise and instance-wise mappings in its online hash function learning.
(2) SHOH does not solely rely on hard labels used by traditional methods. We believe that these labels are too broad to capture the semantic information of the hierarchy effectively. Instead, we propose using soft similarity labels to accurately express multimedia data through refined hierarchical labels/classes.
(3) While these methods often use the (half-)relaxation strategy, which replaces hash codes with feature projections from each modality, this approach results in information loss. SHOH takes a different approach to tackle this issue. We develop efficient discrete optimization algorithms that directly learn hash codes using refined labels, instead of relying on the suboptimal (half-)relaxation strategy.
(4) These methods primarily focus on mapping instances individually, but they neglect the mappings of hierarchical classes. This oversight could result in mapping bias and instability. SHOH addresses this limitation by considering mappings from both perspectives. We combine hierarchical center-wise mapping with traditional instance-wise mapping to capture commonalities and differences among instances, respectively.

3 METHOD

3.1 Notations and Problem Definition

In this article, the bold uppercase letter represents a matrix, e.g., $\mathbf{X}$, while the bold lowercase letter denotes a vector, e.g., $\mathbf{x}$. $\mathbf{X}^\top$ indicates the transpose of $\mathbf{X}$. $\mathbf{I}$, $\mathbf{1}$, and $\mathbf{0}$ represent an identify matrix, an all-one column vector, and an all-zero column vector, respectively. $\|\cdot\|$ is the Frobenius-norm of a matrix or the 2-norm of a vector. $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix.

Assume that the multimedia streaming data only contain image-text pairs. At the tth round, a new data chunk $\{\vec{\mathbf{X}}_1^{(t)}, \vec{\mathbf{X}}_2^{(t)}\}$ is arriving. $\vec{\mathbf{X}}_m^{(t)} \in \mathbb{R}^{d_m \times n_t}$ represents the feature matrix of mth modality, where d_m is the dimension of the corresponding feature, $m \in \{1, 2\}$ corresponds to the modalities of image and text, respectively, and n_t is the size of the chunk. The mth modality feature of old data is indicated by $\tilde{\mathbf{X}}_m^{(t)} \in \mathbb{R}^{d_m \times N_{t-1}}$, where $N_{t-1} = \sum_{i=1}^{t-1} n_i$ is the size of old data. The total data for the mth modality is represented by $\mathbf{X}_m^{(t)} = [\tilde{\mathbf{X}}_m^{(t)}, \vec{\mathbf{X}}_m^{(t)}] \in \mathbb{R}^{d_m \times N_t}$, where $N_t = N_{t-1} + n_t$.

The online cross-modal hashing methods aim to incrementally learn the binary codes $\mathbf{B}^{(t)} = [\tilde{\mathbf{B}}^{(t)}, \vec{\mathbf{B}}^{(t)}]$ of length r for received data, where $\tilde{\mathbf{B}}^{(t)}$ and $\vec{\mathbf{B}}^{(t)}$ denote the old hash codes and the new ones in the current batch, respectively. The online hashing methods learn the hash function $H_m(\mathbf{X}_m^{(t)}) = sign(\mathbf{W}_m^{(t)}\mathbf{X}_m^{(t)})$ to preserve semantic similarity between instances from the original space to Hamming space. Table 1 lists the notations and their descriptions. The following content

Table 1. Summary of Main Notations

Notation	Explanation
$\mathbf{X}_m^{(t)}$	the mth modality features of all instances at the tth round
$\mathbf{B}^{(t)}$	hash codes of all instances at the tth round
$\mathbf{L}^{k(t)}$	the kth layer hard similarity labels of all instances at the tth round
$\mathbf{S}^{k(t)}$	the kth layer soft similarity labels of all instances at the tth round
$\mathbf{A}^{k,K}$	cross-layer affiliation between the kth and Kth layer classes
$\mathbf{C}^{k(t)}$	the kth layer virtual centers at the tth round
$\mathbf{W}_m^{(t)}$	hash functions for the mth modality at the tth round
r	length of hash codes
d_m	dimension of the mth modality features
n_t	number of newly arriving instances at the tth round
N_t	total number of all instances at the tth round
K	depth of the label hierarchy
$\lvert\mathbf{C}^k\rvert$	number of classes at the kth layer

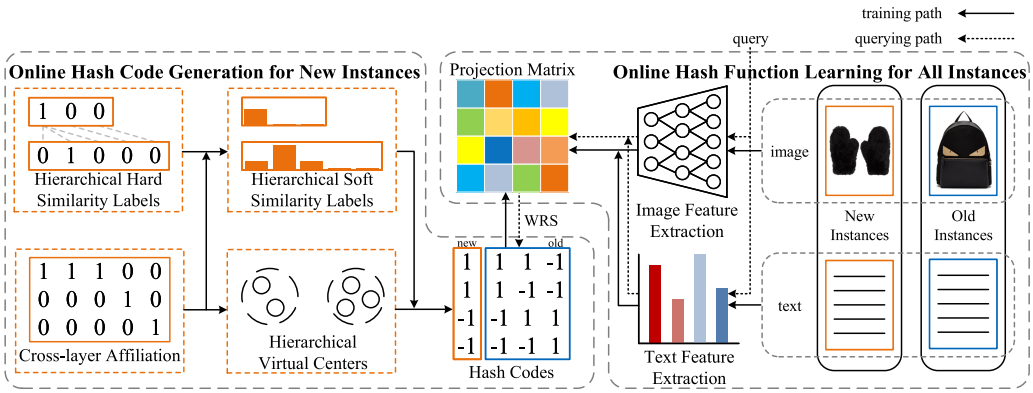

Fig. 2. Framework of SHOH. It contains two data paths: training and querying. During training, SHOH first generates hash codes for new instances. Then, it learns hash functions based on the features and hash codes of all the instances received. During querying, SHOH utilizes WRS to further improve the retrieval performance.

focuses on the learning process of the t-t round. Therefore, the superscript (t) is omitted when only the variables of the tth round are involved.

3.2 Framework Overview

The framework of SHOH consists of two parts, i.e., online hash code generation for new instances and online hash function learning for all instances, as illustrated in Figure 2.

There are two main processes in the online hash code generation, both of which use cross-layer affiliation. On the one hand, we transform the hard similarity labels into soft similarity labels by cross-layer affiliation, where the soft similarity labels are fixed properties for new instances. On the other hand, we acquire hierarchical virtual centers according to cross-layer affiliation and corresponding instances' hash codes. With the soft similarity labels, the hash codes for new instances and the hierarchical virtual centers are optimized iteratively until convergence or iteration up-bound is reached.

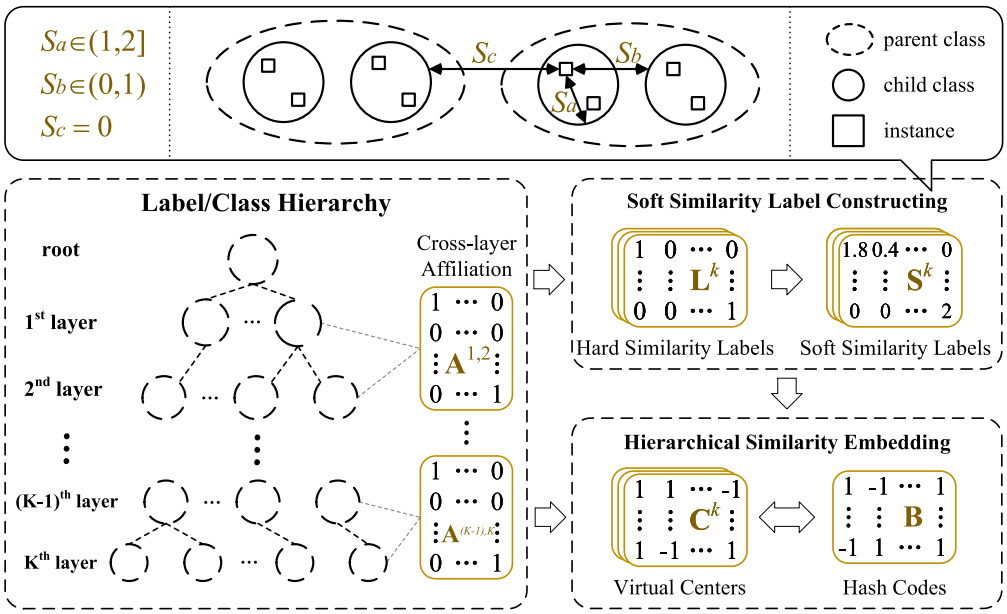

Fig. 3. Process of online hash code generation.

In the online hash function learning, both the features and hash codes of new and old instances are utilized to update the projection matrices of each modality, thereby achieving precise mapping from the feature space to the Hamming space. When a query is received, its features are projected into the Hamming space via the same modality projection matrix to obtain its corresponding hash codes. The hash code matrix of the database is then searched to retrieve data similar to the query.

3.3 Online Hash Code Generation

In this process, SHOH first explores label hierarchy information to generate soft similarity labels. Then, SHOH learns hierarchical virtual centers and embeds the hierarchical information into hash codes via the soft similarity labels. The process is shown in Figure 3.

3.3.1 Soft Similarity Labels Constructing. Assume that K denotes the layer number of label hierarchy and the classes at the Kth layer are the most fine-grained. i^k denotes the ith class at the kth layer, where $k \in \{1, 2, \ldots, K-1\}$. We define the cross-layer affiliation of classes between two adjacent layers as follows:

$$\mathbf{A}_{i,j}^{k,k+1} = \begin{cases} 1, & \text{if } j^{k+1} \in i^k, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

In fact, $\mathbf{A}^{k,k+1}$ is transitive by multiplication, i.e., $\mathbf{A}^{k,K}$ can be acquired by $\mathbf{A}^{k,K} = \Pi_{i=k}^{K-1} \mathbf{A}^{i,i+1}$.

To get hierarchical similarity between instances and classes at each layer, we first calculate the similarity layer-by-layer from top to bottom according to the affiliation. We define

$$\mathbf{V}^{k+1} = \mathbf{L}^k \mathbf{A}^{k,k+1} + \mathbf{L}^{k+1}, \tag{3}$$

where $\mathbf{V}^1 = \mathbf{L}^1$ and $\mathbf{L}^{k+1}$ represents hard similarities between instances and classes at the $(k+1)$-th layer. $\mathbf{V}^{k+1}$ achieves a mask for all dissimilar instances. Then, we normalize $\mathbf{V}^{k+1}$ by 2-norm. We

define $\mathbf{u}_i^{k+1} = \mathbf{v}_i^{k+1}/\|\mathbf{v}_i^{k+1}\|$, where $\mathbf{v}_i^{k+1}$ is the ith row of $\mathbf{V}^{k+1}$, while $\mathbf{u}_i^{k+1}$ is the ith row of $\mathbf{U}^{k+1}$. As a result, we acquire the soft similarity labels at $(k+1)$-th layer by

$$\mathbf{S}^{k+1} = \mathbf{U}^{k+1} + \gamma^{k+1}\mathbf{L}^{k+1}, \tag{4}$$

where $\gamma^{k+1} \in (0, 1]$ is used for emphasizing the classes to which instances belong at $(k+1)$-th layer. In general, we set $\gamma^{k+1} = 1$. The ranges of similarity values are shown at the top of Figure 3, where S_a denotes the similarity between an instance and the class to which it belongs, and S_b denotes the similarities between the instance and the sibling classes of the class, and S_c represents the similarities between the instance and the non-sibling classes of the class, at the $(k+1)$-th layer.

3.3.2 Hierarchical Virtual Centers Learning. It is challenging in online environments, since virtual centers play a crucial role in maintaining the semantic consistency of hash codes for both new and old data. To achieve online learning without updating old hash codes, we resort to virtual centers at each layer and force hash codes close to the appropriate centers. We define the virtual centers as $\mathrm{C} = \{\mathbf{C}^1, \cdots, \mathbf{C}^K\}$ and learn them by preserving the cross-layer affiliation via the distances between the centers. As a result, we define the loss function for centers learning as follows:

$$\begin{aligned} &\min_{\mathbf{C}^k, \mathbf{C}^K} \sum_{k=1}^{K-1} \beta^k \|(r\mathbf{A}^{k,K} - (\mathbf{C}^k)^\top \mathbf{C}^K\|^2, \\ &s.t. \quad \mathbf{C}^k \in \{-1, 1\}^{r\times|\mathbf{C}^k|}, \quad \mathbf{C}^K \in \{-1, 1\}^{r\times|\mathbf{C}^K|}, \end{aligned} \tag{5}$$

where $|\mathbf{C}^k|$ and $|\mathbf{C}^K|$ denote the numbers of classes at kth layer and Kth layer, respectively, and β^k is the parameter. β^k reflects the importance of affiliation between kth layer and Kth layer, and $\sum_{k=1}^{K-1} \beta^k = 1$.

3.3.3 Hierarchical Similarity Embedding. With learned virtual centers, we use soft similarity labels to learn the hash codes for instances. The hierarchical similarity embedding can be achieved by

$$\begin{aligned} &\min_{\mathbf{B}, \mathbf{C}^k} \sum_{k=1}^{K} \alpha^k \|r\mathbf{S}^k - \mathbf{B}^\top \mathbf{C}^k\|^2, \\ &s.t. \mathbf{B} \in \{-1, 1\}^{r\times n_t}, \mathbf{C}^k \in \{-1, 1\}^{r\times|\mathbf{C}^k|}, \end{aligned} \tag{6}$$

where α^k implies the importance of the similarity of the kth layer, $\sum_{k=1}^{K-1} \alpha^k = 1$, and n_t denotes the number of instances in the current batch.

Different from most methods adopting a suboptimal (half-)relaxation strategy that replaces $\mathbf{B}$ with $\mathbf{W}_m\mathbf{X}_m$, SHOH embeds hierarchical semantics directly into $\mathbf{B}$ to generate discriminative hash codes, bridging the semantic gap between the features of various modalities using consistent hash codes.

3.3.4 Online Generation for Hash Codes. To generate the hash codes for new instances in the online framework, we consider the consistency of old hash codes and the hash codes in the current round. According to the virtual centers learned in Equation (5), $\tilde{\mathbf{B}}$ remains unchanged and only $\vec{\mathbf{B}}$ is generated. As a result, we define the loss function as follows:

$$\begin{aligned} &\min_{\vec{\mathbf{B}}, \mathbf{C}^k} \sum_{k=1}^{K} \alpha^k (\|r\tilde{\mathbf{S}}^k - \tilde{\mathbf{B}}^\top \mathbf{C}^k\|^2 + \|r\vec{\mathbf{S}}^k - \vec{\mathbf{B}}^\top \mathbf{C}^k\|^2), \\ &s.t. \quad \vec{\mathbf{B}} \in \{-1, 1\}^{r\times n_t}, \mathbf{C}^k \in \{-1, 1\}^{r\times|\mathbf{C}^k|}, \end{aligned} \tag{7}$$

where $\mathbf{B} = [\tilde{\mathbf{B}}, \vec{\mathbf{B}}]$. Finally, by integrating Equations (7) and (5), online generation for hash codes can be achieved by

$$\begin{aligned} &\min_{\vec{\mathbf{B}}, \mathbf{C}^k, \mathbf{C}^K} \sum_{k=1}^{K} \alpha^k (\|r\tilde{\mathbf{S}}^k - \tilde{\mathbf{B}}^\top \mathbf{C}^k\|^2 + \|r\vec{\mathbf{S}}^k - \vec{\mathbf{B}}^\top \mathbf{C}^k\|^2) + \eta \sum_{k=1}^{K-1} \beta^k \|(r\mathbf{A}^{k,K} - \mathbf{C}^{k\top} \mathbf{C}^K\|^2 \\ &s.t. \quad \vec{\mathbf{B}} \in \{-1, 1\}^{r \times n_t}, \mathbf{C}^k \in \{-1, 1\}^{r \times |\mathbf{C}^k|}, \quad \mathbf{C}^K \in \{-1, 1\}^{r \times |\mathbf{C}^K|}, \end{aligned} \tag{8}$$

where η is the parameter to adjust the weights for two losses. In general, we set $\eta = 1$.

3.4 Discrete Optimization

When optimizing variables subjected to binary constraints, using the conventional relaxation strategy may lead to quantization error and semantic loss, which is unacceptable in online scenarios, as the loss increases with the volume of data. To address this issue, we propose an iterative algorithm to discretely solve the problem described by Equation (8). Specifically, it learns one variable at a time while keeping other variables constant. We use auxiliary matrices to store intermediate variables, which can avoid access to the old data effectively.

$\vec{\mathbf{B}}$-step: To optimize $\vec{\mathbf{B}}$, we keep other variables unchanged and rewrite the Equation (8) as follows:

$$\begin{aligned} &\min_{\vec{\mathbf{B}}} \sum_{k=1}^{K} \alpha^k \|\vec{\mathbf{B}}^\top \mathbf{C}^k\|^2 - 2tr(\vec{\mathbf{B}}^\top \mathbf{P}), \\ &s.t. \quad \mathbf{C}^k \in \{-1, 1\}^{r \times |\mathbf{C}^k|}, \quad \vec{\mathbf{B}} \in \{-1, 1\}^{r \times n_t}, \end{aligned} \tag{9}$$

where

$$\mathbf{P} = r \sum_{k=1}^{K} \alpha^k \mathbf{C}^k \vec{\mathbf{S}}^{k\top}. \tag{10}$$

For this optimization, leveraging the discrete cyclic coordinate descent algorithm, we can deduce the closed-form solution for each row of $\vec{\mathbf{B}}$ sequentially. Let $\vec{\mathbf{b}}$ denote the lth row of $\vec{\mathbf{B}}$, $l \in \{1, 2, \cdots, r\}$. $\vec{\mathbf{B}}'$ represents the matrix of $\vec{\mathbf{B}}$ except for $\vec{\mathbf{b}}$. Since $\vec{\mathbf{b}}$ is the lth bit of the hash codes of the new data, let $\mathbf{p}$ denote the lth row of $\mathbf{P}$. $\mathbf{P}'$ is the matrix of $\mathbf{P}$ excluding $\mathbf{p}$, $\mathbf{c}^k$ represents the lth row of $\mathbf{C}^k$ and $(\mathbf{C}^k)'$ is the matrix of $\mathbf{C}^k$ excluding $\mathbf{c}^k$.

To optimize $\vec{\mathbf{b}}$, $\vec{\mathbf{B}}'$ is regarded as a constant, and Equation (9) can be reduced to the following formulation, i.e.,

$$\begin{aligned} &\min_{\vec{\mathbf{b}}} \left(\sum_{k=1}^{K} \alpha^k \mathbf{c}^k (\mathbf{C}^k)'^\top \vec{\mathbf{B}}' - \mathbf{p} \right) \vec{\mathbf{b}}^\top, \\ &s.t. \quad \vec{\mathbf{B}} \in \{-1, 1\}^{r \times n_t}, \mathbf{C}^k \in \{-1, 1\}^{r \times |\mathbf{C}^k|}. \end{aligned} \tag{11}$$

This problem has an optimal closed-form solution, i.e.,

$$\vec{\mathbf{b}} = \text{sign}\left(\mathbf{p} - \sum_{k=1}^{K} \alpha^k \mathbf{c}^k (\mathbf{C}^k)'^\top \vec{\mathbf{B}}' \right). \tag{12}$$

$\mathbf{C}^K$-step: To optimize $\mathbf{C}^K$, we fix other variables, and then the optimization can be described below.

$$\begin{aligned} &\min_{\mathbf{C}^K} \eta \sum_{k=1}^{K-1} \beta^k \|\mathbf{C}^{k\top} \mathbf{C}^K\|^2 - 2tr(\mathbf{C}^{K\top} \mathbf{Q}^K) + \alpha^K (\|\tilde{\mathbf{B}}^\top \mathbf{C}^K\|^2 + \|\vec{\mathbf{B}}^\top \mathbf{C}^K\|^2), \\ &s.t. \quad \mathbf{C}^K \in \{-1, 1\}^{r \times |\mathbf{C}^K|}, \end{aligned} \tag{13}$$

where

$$\mathbf{Q}^K = r\left(\alpha^K \mathbf{D}^{K(t)} + \eta \sum_{k=1}^{K-1} \beta^k \mathbf{C}^k \mathbf{A}^{k,K}\right), \tag{14}$$

and

$$\mathbf{D}^{K(t)} = \mathbf{D}^{K(t-1)} + \vec{\mathbf{B}}\vec{\mathbf{S}}^K, \mathbf{D}^{K(t-1)} = \tilde{\mathbf{B}}\tilde{\mathbf{S}}^K. \tag{15}$$

We solve this problem in the same way in the $\vec{\mathbf{B}}$-step. Let $\mathbf{e}^{(t)}$ denote the lth row of $\mathbf{E}^{(t)}$. Then, we acquire the solution as follows:

$$\mathbf{c}^K = \text{sign}\left(\mathbf{q}^K - \left(\alpha^K \mathbf{e}^{(t)} + \eta \sum_{k=1}^{K-1} \beta^k \mathbf{c}^k (\mathbf{C}^k)'^\top\right)(\mathbf{C}^K)'\right), \tag{16}$$

where

$$\mathbf{e}^{(t)} = \mathbf{e}^{(t-1)} + \vec{\mathbf{b}}\vec{\mathbf{B}}'^\top, \mathbf{e}^{(t-1)} = \tilde{\mathbf{b}}\tilde{\mathbf{B}}'^\top. \tag{17}$$

$\mathbf{C}^k$-step: To optimize $\mathbf{C}^k$, we fix other variables and then get the following formula, i.e.,

$$\begin{aligned} &\min_{\mathbf{C}^k} \eta\beta^k \|\mathbf{C}^{k\top}\mathbf{C}^K\|^2 - 2tr(\mathbf{C}^{k\top}\mathbf{Q}^k) + \alpha^k(\|\tilde{\mathbf{B}}^\top \mathbf{C}^k\|^2 + \|\vec{\mathbf{B}}^\top \mathbf{C}^k\|^2), \\ &s.t. \quad \mathbf{C}^k \in \{-1, 1\}^{r\times|\mathbf{C}^k|}, \end{aligned} \tag{18}$$

where

$$\mathbf{Q}^k = r(\alpha^k \mathbf{D}^{k(t)} + \eta\beta^k \mathbf{C}^K \mathbf{A}^{k,K\top}), \tag{19}$$

and

$$\mathbf{D}^{k(t)} = \mathbf{D}^{k(t-1)} + \vec{\mathbf{B}}\vec{\mathbf{S}}^k, \mathbf{D}^{k(t-1)} = \tilde{\mathbf{B}}\tilde{\mathbf{S}}^k. \tag{20}$$

The optimal solution to this problem can be calculated by

$$\mathbf{c}^k = \text{sign}\Big(\mathbf{q}^k - (\alpha^k \mathbf{e}^{(t)} + \eta\beta^k \mathbf{c}^K (\mathbf{C}^K)'^\top)(\mathbf{C}^k)'\Big). \tag{21}$$

3.5 Online Hash Function Learning

To accurately map the query to Hamming space, we utilize a linear regression model that integrates hierarchical center-wise and instance-wise mappings.

3.5.1 Hierarchical Center-wise Mapping. By mapping classes' feature centers $\overline{\mathbf{X}}_m^{k(t)}$ to their virtual centers $\mathbf{C}^k$, layer-by-layer, the commonality of instances in each class can be acquired. The objective function is defined as follows:

$$\min_{\mathbf{W}_m} \sum_{k=1}^{K} \alpha^k \|\mathbf{C}^k - \mathbf{W}_m \overline{\mathbf{X}}_m^{k(t)}\|^2, \tag{22}$$

where $\overline{\mathbf{X}}_m^{k(t)}$ is the means of features of instances labeled by the kth layer's ith class, which is updated on-the-fly at each round. Let $\overline{\mathbf{x}}_{m;i}^{k(t)}$ be the ith column of $\overline{\mathbf{X}}_m^{k(t)}$, i.e., the feature center of the ith class, which is calculated as follows:

$$\overline{\mathbf{x}}_{m;i}^{k(t)} = \frac{N_{t-1;i}^k \overline{\mathbf{x}}_{m;i}^{k(t-1)} + \text{sum}\left(\vec{\mathbf{X}}_{m;i}^{k(t)}\right)}{N_{t;i}^k}, \tag{23}$$

where $N_{t;i}^k$ is the total number of instances belonged to the kth layer's ith class at the tth round, and $N_{0;i}^k = 0$. $\vec{\mathbf{X}}_{m;i}^{k(t)}$ denotes the feature matrix of corresponding new instances, and sum$(\cdot)$ is an operator for column-by-column summation of the matrix.

3.5.2 Instance-wise Mapping. By mapping each instance's feature to the hash code to achieve instance-wise mapping, it is beneficial for the model to get differences between instances, i.e., the individuality of each instance. The objective function is defined as follows:

$$\min_{\mathbf{W}_m} \|\mathbf{B} - \mathbf{W}_m \mathbf{X}_m\|^2. \tag{24}$$

According to $\mathbf{B} = [\tilde{\mathbf{B}}, \vec{\mathbf{B}}]$, Equation (24) can be transformed to

$$\min_{\mathbf{W}_m} \|\vec{\mathbf{B}} - \mathbf{W}_m \vec{\mathbf{X}}_m\|^2 + \|\tilde{\mathbf{B}} - \mathbf{W}_m \tilde{\mathbf{X}}_m\|^2. \tag{25}$$

3.5.3 Online Learning for Hash Function. By integrating Equation (22), Equation (25), and the regularization term, the hash functions for online learning can be achieved as follows:

$$\min_{\mathbf{W}_m} \|\vec{\mathbf{B}} - \mathbf{W}_m \vec{\mathbf{X}}_m\|^2 + \|\tilde{\mathbf{B}} - \mathbf{W}_m \tilde{\mathbf{X}}_m\|^2 + \mu \sum_{k=1}^{K} \alpha^k \|\mathbf{C}^k - \mathbf{W}_m \overline{\mathbf{X}}_m^{k(t)}\|^2 + \xi \|\mathbf{W}_m\|^2, \tag{26}$$

where μ and ξ are the tradeoff parameters. As a result, the solution can be given by

$$\mathbf{W}_m = \left(\mathbf{F}_m^{(t)} + \mu \sum_{k=1}^{K} \alpha^k \mathbf{C}^k \overline{\mathbf{X}}_m^{k(t)\top}\right)\left(\mathbf{G}_m^{(t)} + \mu \sum_{k=1}^{K} \alpha^k \overline{\mathbf{X}}_m^{k(t)} \overline{\mathbf{X}}_m^{k(t)\top} + \xi \mathbf{I}\right)^{-1}, \tag{27}$$

where

$$\mathbf{F}_m^{(t)} = \mathbf{F}_m^{(t-1)} + \vec{\mathbf{B}} \vec{\mathbf{X}}_m^\top, \mathbf{F}_m^{(t-1)} = \tilde{\mathbf{B}} \tilde{\mathbf{X}}_m^\top, \tag{28}$$

and

$$\mathbf{G}_m^{(t)} = \mathbf{G}_m^{(t-1)} + \vec{\mathbf{X}}_m \vec{\mathbf{X}}_m^\top, \mathbf{G}_m^{(t-1)} = \tilde{\mathbf{X}}_m \tilde{\mathbf{X}}_m^\top. \tag{29}$$

3.6 Weighted Retrieval Strategy

Most traditional hashing methods treat all bits of a hash code as equal, ignoring that different bits contribute differently [50]. During retrieval, we have noticed that classifiers with projection values close to zero may be ineffective and cause bit-flipping errors [14, 36], leading to decreased retrieval performance. To handle this issue, we propose a **Weighted Retrieval Strategy (WRS)** that evaluates the credibility of each classifier and assigns a weight to its corresponding bit.

When a new query with the mth modal feature $\mathbf{x}_m$ arrives, its hash code $\mathbf{b}_q$ is generated by

$$\mathbf{b}_q = \text{sign}(\mathbf{W}_m \mathbf{x}_m). \tag{30}$$

The traditional method of calculating Hamming distance, i.e., the number of inconsistent bits between the query hash code $\mathbf{b}_q$ and the database instance hash code $\mathbf{b}_d$, is defined as follows:

$$D = \frac{r - \mathbf{b}_q^\top \mathbf{b}_d}{2}. \tag{31}$$

We believe that the reliability of the projection value near zero is lower than that far away from zero. As a result, we define the lth bit's weight $\mathbf{t}_l$ with the corresponding projection value $\mathbf{w}_m \mathbf{x}_m$ as follows:

$$\mathbf{t}_l = \begin{cases} \text{abs}(\mathbf{w}_m \mathbf{x}_m), & \text{abs}(\mathbf{w}_m \mathbf{x}_m) < 1, \\ 1, & otherwise, \end{cases} \tag{32}$$

where $\mathbf{w}_m$ is the lth classifier, i.e., the lth row of $\mathbf{W}_m$, and abs$(\cdot)$ returns the absolute value. After all bits' weights $\mathbf{t}$ calculated on-the-fly, the **Weighted Distance (*WD*)** can be obtained by

$$WD = \frac{\mathbf{t}^\top \mathbf{1} - (\mathbf{t} \odot \mathbf{b}_q)^\top \mathbf{b}_d}{2}, \tag{33}$$

ALGORITHM 1: Workflow of SHOH at the tth round

Input: new instances' features $\vec{\mathbf{X}}_m^{(t)}$ and hierarchical labels $\vec{\mathbf{L}}^{k(t)}$; parameters α, β, γ, η, μ, and ξ; iteration number $iter$
Output: new instances' hash codes $\vec{\mathbf{B}}^{(t)}$, hash functions $H_m^{(t)}(\cdot)$
if at the first round **then**
  Initialize $\mathbf{C}^{K(t)}$ and $\mathbf{C}^{k(t)}$ randomly
  Initialize $\mathbf{D}^{k(0)}$, $\mathbf{E}^{(0)}$, $\mathbf{F}_m^{(0)}$, $\mathbf{G}_m^{(0)}$, and $\overline{\mathbf{X}}_m^{k(0)}$ to zero matrices
end if
Initialize $\vec{\mathbf{B}}^{(t)}$ randomly
Calculate $\vec{\mathbf{S}}^{k(t)}$ according to Equation (4)
for $i = 1$ to $iter$ **do**
  Update $\vec{\mathbf{B}}^{(t)}$ according to Equation (12)
  Update $\mathbf{C}^{K(t)}$ according to Equation (16)
  Update $\mathbf{C}^{k(t)}$ according to Equation (21)
end for
Update $\mathbf{D}^{K(t)}$, $\mathbf{D}^{k(t)}$, $\mathbf{E}^{(t)}$, $\mathbf{F}_m^{(t)}$, $\mathbf{G}_m^{(t)}$, and $\overline{\mathbf{X}}_m^{k(t)}$ according to Equations (15) (20) (17) (28) (29) (23)
Update $\mathbf{W}_m^{(t)}$ according to Equation (27)

where $\odot$ denotes the element-wise product operation.

Note that this strategy only works in the query process and is orthogonal to the online hash learning framework. In addition, the weight $\mathbf{t}$ is calculated on-the-fly based on the query $\mathbf{x}_m$, which does not bring additional storage overhead. The workflow of SHOH at the tth round is presented in Algorithm 1.

3.7 Complexity Analysis

In this section, we analyze the time complexity of SHOH. It prevents secondary access to old instances by retaining intermediate variables via auxiliary matrices, such as $\mathbf{D}^{K(t-1)}$, $\mathbf{D}^{k(t-1)}$, $\mathbf{E}^{(t-1)}$, $\mathbf{F}_m^{(t-1)}$, and $\mathbf{G}_m^{(t-1)}$, enabling it to be appropriate for large-scale datasets and to perform incremental learning. The time complexity of SHOH in each round is proportional to the size of the new instances, i.e., n_t. Specifically, for each round, the computational complexity of updating $\vec{\mathbf{B}}$ is $O(r^2 \sum_{k=1}^{K} |\mathbf{C}^k| iter)$, where $iter$ is the number of iterations. Updating $\mathbf{C}^K$ requires $O(r^2|\mathbf{C}^K|iter)$. Updating $\mathbf{C}^k$ requires $O(r^2|\mathbf{C}^k|iter)$, where $k = \{1, \cdots, K-1\}$. Updating $\mathbf{W}_m$ requires $O(d_m^3 + d_m^2(n_t + r + \sum_{k=1}^{K} |\mathbf{C}^k|) + d_m r(n_t + \sum_{k=1}^{K} |\mathbf{C}^k|))$.

4 EXPERIMENT

4.1 Datasets

FashionVC [28] contains 20,726 image-text pairs with hierarchical labels. The label hierarchy is designed based on clothing styles. The first layer includes eight parent classes: *Activewear*, *Dress*, *Jeans*, *Outerwear*, *Pants*, *Short*, *Skirt*, and *Top*. By further subdividing, the second layer consists of 27 child classes. Each parent class has at least 1 child class and at most 7 child classes, and we list the details in Table 2. Following References [29, 47], we select 19,862 instances to evaluate the retrieval performance. 2,000 instances are selected randomly as the query set, and the remainder is divided into 7 data chunks, with the first 6 chunks consisting of 2,000 instances and the last chunk totaling 3,696 instances. This section presents and analyzes the results obtained from the shallow features. Each image is represented by a 512-dimensional GIST feature vector, while each text is represented by a 2,685-dimensional bag-of-words vector. Additionally, we use 4,096-dimensional

Table 2. Label Hierarchy of FashionVC Dataset

Parent Class	Child Class
Activewear (Fp1)	Active Wear Pants (Fc1)
Dress (Fp2)	Cocktail Dress (Fc2); Day Dress (Fc3); Grown (Fc4)
Jeans (Fp3)	Bootcut Jeans (Fc5); Flared Jeans (Fc6); Wide Leg Jeans (Fc7); Boyfriend Jeans (Fc8); Skinny Jeans (Fc9); Straight Leg Jeans (Fc10); Topshop Jeans (Fc11)
Outerwear (Fp4)	Coat (Fc12); Jacket (Fc13); Vest (Fc14)
Pants (Fp5)	Cropped Pants (Fc15); Leggings (Fc16)
Short (Fp6)	Short Pants (Fc17)
Skirt (Fp7)	Knee-length Skirt (Fc18); Long Skirt (Fc19); Mini Skirt (Fc20)
Top (Fp8)	Blouse (Fc21); Cardigan (Fc22); Sweater (Fc23); Sweat & Hoodyshirt (Fc24); Tank Top (Fc25); T-shirt (Fc26); Tunic (Fc27)

Table 3. Label Hierarchy of Ssense Dataset

Parent Class	Child Class
Accessory (Sp1)	Belt (Sc1); Eyewear (Sc2); Gloves (Sc3); Keychain (Sc4); Necklace (Sc5); Socks (Sc6)
Bag (Sp2)	Backpack (Sc7); Duffle Bag (Sc8); Pouch (Sc9); Shoulder Bag (Sc10) ; Tote Bag (Sc11)
Clothing (Sp3)	Bikini (Sc12); Blazer (Sc13); Boxers (Sc14); Dress (Sc15); Jacket (Sc16); Jeans (Sc17); Long/mid Length Skirt (Sc18); Mini Skirt (Sc19); Shirt (Sc20); Shorts (Sc21); T-shirt (Sc22)
Shoes (Sp4)	Boots (Sc23); Espadrilles (Sc24); Flats (Sc25); Loafers (Sc26); Sandals (Sc27); Sneakers (Sc28)

features extracted from GIST and VGG-F models as input. Section 5 presents and discusses these results as part of the verification process.

Ssense [29] consists of 25,947 image-text pairs with hierarchical labels. Its label hierarchy is built based on the purpose of various fashion items. The hierarchy comprises 4 parent classes in the first layer, namely, *Accessory*, *Bag*, *Clothing*, and *Shoes*. In the second layer, there are 28 child classes. Each parent class has between 5 and 11 child classes, and we provide the details in Table 3. Following References [29, 47], 15,696 instances are selected for evaluation. 2,000 instances are selected randomly as the query set, and the remaining 17,862 instances are divided into 8 data chunks, with the first 7 chunks comprising 2,000 instances and the last chunk containing 3,862 instances. This section presents and analyzes the results obtained using shallow features. Each image is represented by a 512-dimensional GIST feature vector, while each text is represented by a 2,685-dimensional bag-of-words vector. Additionally, we use 4,096-dimensional feature vectors extracted from GIST and VGG-F models as input. Section 5 presents and discusses these results for further verification.

NUS-WIDE [5] consists of 269,648 non-hierarchical labeled image-text pairs of 81 categories. We select the 21 most frequent labels and their corresponding 195,834 pairs from the original dataset. Based on these labels, we construct a hierarchical structure referring to the hypernyms provided by WordNet [21], as shown in Figure 4. Subsequently, we randomly extract 5,000 pairs whose images and texts served as queries, while the remaining data are divided into 18 chunks. Specifically, each of the first 17 chunks contains 10,000 pairs, while the last consists of 10,834 pairs. We utilize 4,096-dimensional VGG-F feature vectors and 1,000-dimensional binary tagging vectors as input. Section 5 presents and discusses these results as other verification.

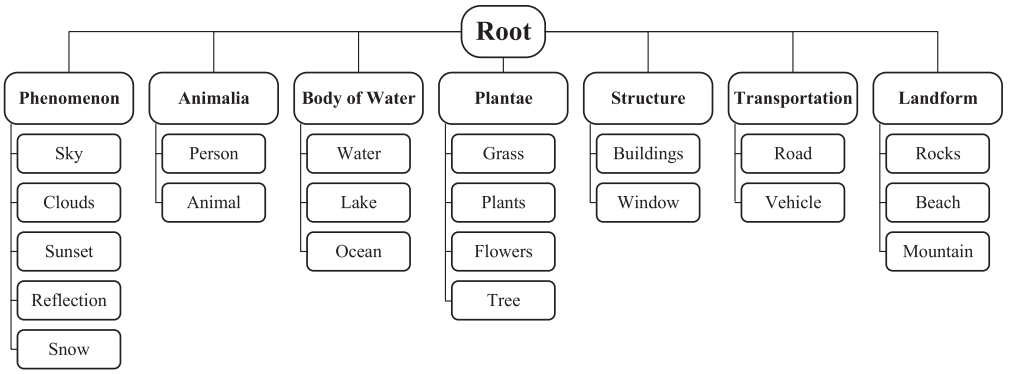

Fig. 4. Semantic hierarchical structure of NUS-WIDE dataset.

4.2 Experimental Settings

4.2.1 Evaluation Metrics. Two kinds of retrieval tasks are performed to assess cross-modal retrieval capacity. *Image-to-Text* represents retrieving the text by the given image, and *Text-to-Image* retrieving the images by the specified text. For a fair assessment, we adopted two widely used evaluation metrics, i.e., **Mean Average Precision (MAP)** and **Precision-Recall (PR)** curves. The finest-grained label set is considered ground truth.

4.2.2 Baselines. We select state-of-the-art online cross-modal hashing methods as baselines, including LEMON, DOCH, OMGH, and OSCMFH.

LEMON [35] is a supervised non-hierarchical online hashing method that builds a label embedding framework, including label similarity preserving and label reconstructing, leveraging the inner production between new and old data labels to keep similarities.

DOCH [48] is a supervised non-hierarchical online hashing method based on the offline method DLFH. It can generate consistent and high-quality hash codes exploring similarities between new and new data and employing semantic information.

OMGH [18] is a non-hierarchical online hashing method, developing a matrix tri-factorization framework to decompose features into modality-specific semantic representation and hash codes. By manifold embedding, OMGH can perform both unsupervised and supervised learning.

OSCMFH [26] is a supervised non-hierarchical online hashing method based on OCMFH. It utilizes semantic labels to obtain the latent semantic representation of new and old data.

Since all these methods use shallow hand-crafted features as initial features in their original experiments, we use GIST [23] to acquire initial features in the comparison experiments. In addition, we run SHOH on hierarchical and non-hierarchical datasets using deep features extracted from VGG-F [27]. We acquired some interesting results and discussed them in Section 5.

4.2.3 Experimental Details. SHOH has parameters as follows: α, β, γ, η, μ, and ξ. Our method achieves the best results on both FashionVC and Ssense when $\alpha^1 = 0.2$, $\alpha^2 = 0.8$, $\beta^1 = 1$, $\gamma^2 = 1$, $\eta = 10$, $\mu = 1,000$ and $\xi = 1$. Moreover, we set *iter* as 7. Note that we input labels hierarchically during the training process of SHOH. For each baseline, we input the finest-grained labels and the concatenation of hierarchical labels, and we differentiate between them using the "-fine" or "-full" suffix.

4.3 Retrieval Performance

The MAP results and PR curves are used to evaluate the retrieval performance. As shown in Tables 4 and 5, SHOH outperforms the state-of-the-art baselines in most cases, and our WRS

Table 4. MAP Results of Various Methods on FashionVC

Method	Image-to-Text				Text-to-Image			
	16-bit	32-bit	64-bit	128-bit	16-bit	32-bit	64-bit	128-bit
OSCMFH-full	0.0749	0.0749	0.0749	0.0749	0.0749	0.0749	0.0749	0.0749
OSCMFH-fine	0.2167	0.2528	0.2673	0.2662	0.3557	0.4120	0.4174	0.4190
OMGH-full	0.3460	0.3902	0.4553	0.4846	0.5830	0.6613	0.7908	0.8247
OMGH-fine	0.3259	0.4394	0.4772	0.5089	0.5927	0.7828	0.8101	0.8315
DOCH-full	0.2891	0.2959	0.2972	0.3026	0.4010	0.4048	0.4058	0.4079
DOCH-fine	0.3922	0.4506	0.4688	0.4684	0.6995	0.7737	0.7950	0.7968
LEMON-full	0.4681	0.5175	0.5335	0.5619	0.8738	0.9259	0.9296	0.9325
LEMON-fine	0.5302	0.5667	0.5764	0.5882	0.9070	0.9285	0.9297	0.9310
SHOH	0.5217	0.5757	0.6006	0.6123	0.9277	0.9391	0.9410	0.9419
SHOH-WRS	**0.5784**	**0.6122**	**0.6209**	**0.6255**	**0.9367**	**0.9446**	**0.9444**	**0.9445**
Relative Gains	9.09%	8.03%	7.72%	6.34%	3.27%	1.73%	1.58%	1.29%

The best results are in bold, and the best baselines' results are underlined.

Table 5. MAP Results of Various Methods on Ssense

Method	Image-to-Text				Text-to-Image			
	16-bit	32-bit	64-bit	128-bit	16-bit	32-bit	64-bit	128-bit
OSCMFH-full	0.0678	0.0678	0.0678	0.0678	0.0678	0.0678	0.0678	0.0678
OSCMFH-fine	0.6145	0.7754	0.8274	0.8567	0.8333	0.9606	0.9803	0.9836
OMGH-full	0.4749	0.5173	0.6239	0.6719	0.5456	0.6036	0.7172	0.8080
OMGH-fine	0.4849	0.6160	0.6860	0.7035	0.5770	0.7525	0.8178	0.8428
DOCH-full	0.2327	0.2343	0.2341	0.2361	0.2364	0.2370	0.2376	0.2395
DOCH-fine	0.5541	0.6350	0.6439	0.6529	0.5982	0.7092	0.7238	0.7331
LEMON-full	0.6597	0.7423	0.7812	0.7964	0.8370	0.9641	0.9822	0.9838
LEMON-fine	0.7738	0.7978	0.8070	0.8110	0.9724	0.9797	0.9817	0.9820
SHOH	0.8018	0.8512	0.8638	0.8747	0.9735	0.9843	0.9832	0.9845
SHOH-WRS	**0.8414**	**0.8750**	**0.8781**	**0.8806**	**0.9763**	**0.9848**	**0.9843**	**0.9852**
Relative Gains	8.74%	9.68%	6.13%	2.79%	0.40%	0.52%	0.21%	0.14%

The best results are in bold, and the best baselines' results are underlined.

further improves retrieval performance. On FashionVC, the average MAP values of SHOH-WRS overtake those of the best baselines by 7.8% and 1.97% in *Image-to-Text* and *Text-to-Image* tasks, respectively. On Ssense, the counterparts are 6.83% and 0.32%. In addition, based on the PR curves shown in Figure 5, it can be observed that the performance of SHOH surpasses all the baselines.

4.4 Time Cost Analysis

In the previous section, the qualitative analysis reveals that the time complexity of SHOH has a linear correlation with the volume of new data n_t. To validate this finding, we conduct experiments to evaluate the time cost of all methods on FashionVC at code lengths of 32-bit and 64-bit. As shown in Figure 6, we draw the following conclusions: (1) Our method may not have the highest training efficiency compared to LEMON, but it produces optimal performance results. Training time can be decreased by reducing the number of iterations with acceptable MAP degradation. (2) The bit-by-bit optimization implemented by SHOH requires more time as the number of bits increases. Therefore, longer code lengths do not necessarily result in better performance.

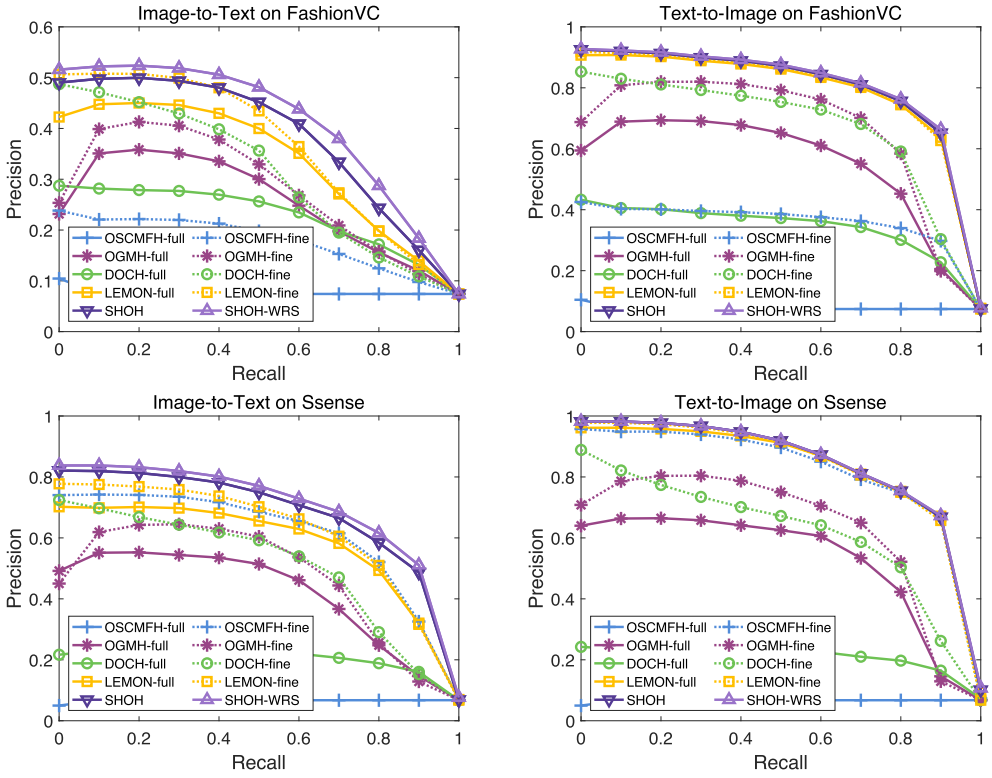

Fig. 5. PR curves in 32-bit hash codes on two datasets.

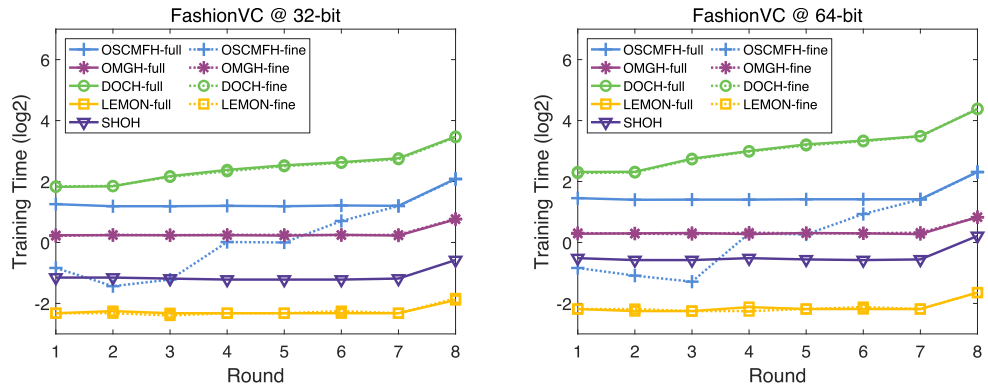

Fig. 6. Training time-round curves on FashionVC.

4.5 Ablation Studies

We conduct comprehensive ablation studies to verify the effectiveness of hierarchy, soft similarity labels, and weighted retrieval strategy. As shown in Table 6, three diverse derivatives of SHOH are designed to elaborate the effects of the hierarchy and soft similarity labels on retrieval performance.

Table 6. Design of Ablation Studies

structure	labels (notation)	method
hierarchical	soft similarity labels ($\mathbf{S}^1, \mathbf{S}^2$)	SHOH
	hard similarity labels ($\mathbf{L}^1, \mathbf{L}^2$)	SHOH-1
non-hierarchical	concatenation labels ($\mathbf{L}^{1+2}$)	SHOH-2
	the finest-grained labels ($\mathbf{L}^2$)	SHOH-3

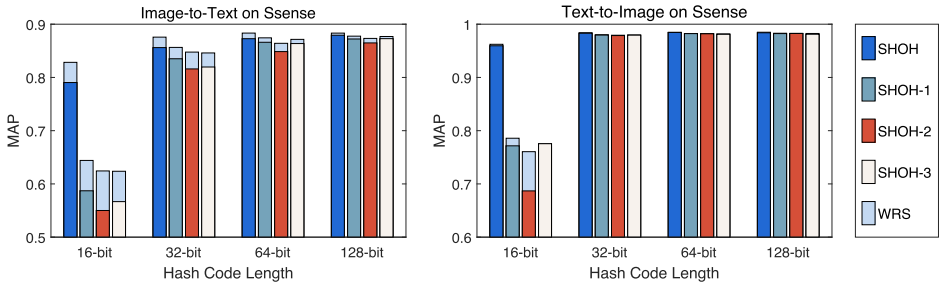

Fig. 7. MAP results of SHOH and derivatives on Ssense.

For the first derivative SHOH-1, its objective function is defined as follows:

$$\begin{aligned} &\min_{\vec{\mathbf{B}}, \mathbf{C}^1, \mathbf{C}^2} \eta\beta^1 \|(r\mathbf{A}^{1,2} - (\mathbf{C}^1)^\top \mathbf{C}^2\|^2 + \sum_{k=1}^{2} \alpha^k (\|r\tilde{\mathbf{L}}^k - \tilde{\mathbf{B}}^\top \mathbf{C}^k\|^2 + \|r\vec{\mathbf{L}}^k - \vec{\mathbf{B}}^\top \mathbf{C}^k\|^2), \\ &s.t. \quad \vec{\mathbf{B}} \in \{-1, 1\}^{r \times n_t}, \mathbf{C}^1 \in \{-1, 1\}^{r \times |\mathbf{C}^1|}, \mathbf{C}^2 \in \{-1, 1\}^{r \times |\mathbf{C}^2|}, \end{aligned} \tag{34}$$

and

$$\min_{\mathbf{W}_m} \|\vec{\mathbf{B}} - \mathbf{W}_m \vec{\mathbf{X}}_m\|^2 + \|\tilde{\mathbf{B}} - \mathbf{W}_m \tilde{\mathbf{X}}_m\|^2 + \mu \sum_{k=1}^{2} \alpha^k \|\mathbf{C}^k - \mathbf{W}_m \overline{\mathbf{X}}_m^{k(t)}\|^2 + \xi \|\mathbf{W}_m\|^2. \tag{35}$$

For the second derivative SHOH-2, its objective function is defined as follows:

$$\begin{aligned} &\min_{\vec{\mathbf{B}}, \mathbf{C}^{1+2}} \|r\tilde{\mathbf{L}}^{1+2} - \tilde{\mathbf{B}}^\top \mathbf{C}^{1+2}\|^2 + \|r\vec{\mathbf{L}}^{1+2} - \vec{\mathbf{B}}^\top \mathbf{C}^{1+2}\|^2, \\ &s.t. \quad \vec{\mathbf{B}} \in \{-1, 1\}^{r \times n_t}, \quad \mathbf{C}^{1+2} \in \{-1, 1\}^{r \times |\mathbf{C}^{1+2}|}, \end{aligned} \tag{36}$$

and

$$\min_{\mathbf{W}_m} \|\vec{\mathbf{B}} - \mathbf{W}_m \vec{\mathbf{X}}_m\|^2 + \|\tilde{\mathbf{B}} - \mathbf{W}_m \tilde{\mathbf{X}}_m\|^2 + \mu \|\mathbf{C}^{1+2} - \mathbf{W}_m \overline{\mathbf{X}}_m^{1+2(t)}\|^2 + \xi \|\mathbf{W}_m\|^2. \tag{37}$$

For the third derivative SHOH-3, its objective function is defined as follows:

$$\begin{aligned} &\min_{\vec{\mathbf{B}}, \mathbf{C}^2} \|r\tilde{\mathbf{L}}^2 - \tilde{\mathbf{B}}^\top \mathbf{C}^2\|^2 + \|r\vec{\mathbf{L}}^2 - \vec{\mathbf{B}}^\top \mathbf{C}^2\|^2, \\ &s.t. \quad \vec{\mathbf{B}} \in \{-1, 1\}^{r \times n_t}, \quad \mathbf{C}^2 \in \{-1, 1\}^{r \times |\mathbf{C}^2|}, \end{aligned} \tag{38}$$

and

$$\min_{\mathbf{W}_m} \|\vec{\mathbf{B}} - \mathbf{W}_m \vec{\mathbf{X}}_m\|^2 + \|\tilde{\mathbf{B}} - \mathbf{W}_m \tilde{\mathbf{X}}_m\|^2 + \mu \|\mathbf{C}^2 - \mathbf{W}_m \overline{\mathbf{X}}_m^{2(t)}\|^2 + \xi \|\mathbf{W}_m\|^2. \tag{39}$$

The MAP results are shown in Figure 7. It is clear that SHOH outperforms its derivatives in all cases. The light blue columns in the graph represent the performance gains brought by WRS, which enhances retrieval accuracy during querying and is independent of the training process.

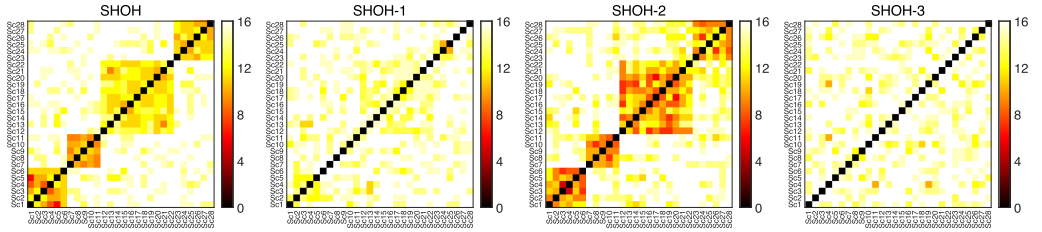

Fig. 8. Heatmaps of SHOH and derivatives at 32-bit hash codes on Ssense.

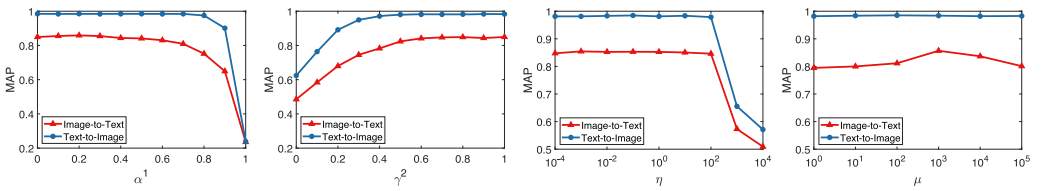

Fig. 9. Parameter sensitivity analysis of α^1, γ^2, η, and μ.

Furthermore, we employ heatmaps to exhibit the quality of generated hash codes, which indicates the semantic information preserved in the hash codes. Figure 8 shows the evaluation of hash code quality for SHOH and its derivatives. The color of each block in the heatmap represents the depth and darkness based on the semantic distance (measured by Hamming distance) between the centers of two classes. The closer the centers are, the darker the color and vice versa. The horizontal and vertical coordinates are provided in Table 3. We calculate the average hash codes for instances of each class, binarize the values, and obtain hash codes for each class center. We observe that SHOH and SHOH-2 effectively preserve the semantic information between sibling classes (having the same parent class). However, SHOH-2 suffers from a problem of too-close Hamming distance between sibling classes, which results in poor retrieval performance, as shown in Figure 7. However, SHOH-1 retains some inter-class semantic information, while SHOH-3 contains almost none.

Based on the above MAP results and heatmaps, we draw the following conclusions: (1) SHOH-1 benefits from the hierarchical structure, but its hard similarity labels restrict performance. As a result, it can achieve better fine-grained retrieval performance compared to other derivatives. However, the hash codes generated by SHOH-1 preserve only a small amount of semantic information. (2) SHOH-2 can utilize all labels, but its flat structure prevents it from distinguishing between classes in different layers. Consequently, it tends to have poor retrieval performance. (3) SHOH-3 can only use the finest-grained labels, resulting in its hash codes containing minimal semantic information compared to the others. In summary, SHOH outperforms its derivatives, demonstrating the effectiveness of hierarchies, soft similarity labels, and WRS.

4.6 Parameter Sensitivity Analysis

To assess the impact of α, β, γ, η, μ, and ξ, we conduct a sensitivity analysis for them. Each parameter is adjusted individually while keeping the other parameters constant. The experiment is conducted on Ssense, using 32-bit hash codes. $\beta^1 = 1$ is due to the constraints $\sum_{k=1}^{K-1} \beta^k = 1$ and $K = 2$. In addition, we find that the regularization parameter ξ exhibits minimal influence, except when it is extremely large or small. Consequently, to conserve space, we present the performance of the remaining parameters in Figure 9.

The value of $\alpha^k \in [0, 1]$ expresses the confidence for the kth layer. Since $\alpha^1 + \alpha^2 = 1$, we simply adjust α^1. We believe that as α^1 approaches 0, the hash code converges closely to the virtual center

Table 7. MAP Results of *Image-to-Text* Performed by SHOH and LEMON

FEM (Dim)	Method	FashionVC				Ssense			
		16-bit	32-bit	64-bit	128-bit	16-bit	32-bit	64-bit	128-bit
GIST (4,096)	LEMON-full	0.5673	0.6081	0.6307	0.6515	0.7230	0.8718	0.8952	0.9129
	LEMON-fine	0.6260	0.6477	0.6666	0.6716	0.8878	0.9129	0.9176	0.9217
	SHOH	0.6128	0.6693	0.6982	0.7046	0.9023	0.9284	0.9349	0.9386
	SHOH-WRS	0.6726	0.7043	0.7133	0.7137	0.9218	0.9390	0.9412	0.9406
VGG-F (4,096)	LEMON-full	0.6169	0.6914	0.7216	0.7344	0.7710	0.9349	0.9569	0.9611
	LEMON-fine	0.6840	0.7143	0.7288	0.7399	0.9398	0.9576	0.9616	0.9646
	SHOH	0.6872	0.7351	0.7491	0.7579	0.9407	0.9630	0.9667	0.9682
	SHOH-WRS	0.7277	0.7573	0.7610	0.7650	0.9535	0.9670	0.9692	0.9702

FEM is short for Feature Extraction Model, and Dim is short for Dimension.

Table 8. MAP Results on NUS-WIDE

Method	Image-to-Text				Text-to-Image			
	16-bit	32-bit	64-bit	128-bit	16-bit	32-bit	64-bit	128-bit
OSCMFH-full	0.4469	0.4752	0.5052	0.5005	0.6151	0.7197	0.7428	0.7483
OSCMFH-fine	0.4292	0.4732	0.4861	0.5012	0.5877	0.7171	0.7360	0.7393
OMGH-full	0.7026	0.7189	0.7236	0.7313	0.6721	0.6849	0.6895	0.6986
OMGH-fine	0.7287	0.7517	0.7685	0.7704	0.6941	0.7165	0.7321	0.7324
DOCH-full	0.6642	0.6769	0.6957	0.6912	0.6338	0.6444	0.6655	0.6627
DOCH-fine	0.7883	0.8066	0.8173	0.8249	0.7511	0.7679	0.7797	0.7872
LEMON-full	0.7656	0.7763	0.7834	0.7869	0.7213	0.7311	0.7373	0.7398
LEMON-fine	0.7921	0.8037	0.8116	0.8153	0.7454	0.7580	0.7628	0.7686
SHOH	0.7770	0.8092	0.8251	**0.8301**	0.7368	0.7678	0.7830	0.7857
SHOH-WRS	**0.7973**	**0.8190**	**0.8287**	0.8295	**0.7558**	**0.7779**	**0.7871**	**0.7878**

The best results are in bold, and the best baselines' results are underlined.

of its child class. γ^{k+1} emphasizes the class to which instances belong at the $(k+1)$-th layer. When γ is close to 0, the learned hash codes tend to preserve similar semantic relationships. However, they cannot guarantee the correct distinction of semantically similar instances. The MAP results for parameters η and μ are also presented in Figure 9. It shows that the performance remains satisfactory when η ranges from 10^{-4} to 10^2 and $\mu = 10^3$.

5 DISCUSSION AND FUTURE WORK

5.1 Discussion

5.1.1 Other Verification on Hierarchical Datasets. As features extracted by deep methods are more accurate than those extracted by shallow ones, we are interested in the performance of SHOH when working with deep features. We evaluated the performance of SHOH with deep features extracted by the VGG-F model and shallow features extracted by the GIST model. Table 7 shows the results on two datasets for different initial features with the same dimensions. Since VGG-F model is initially designed for image processing, we only present the results for *Image-to-Text*. The results are consistent with our expectation that, even in the same dimension, the results under deep features show higher MAP. This phenomenon is also observed in LEMON.

5.1.2 Other Verification on Non-hierarchical Datasets. After confirming the effectiveness of deep features, we are curious to see how SHOH will perform on non-hierarchical datasets. As

a result, we evaluated the performance of SHOH and all baselines on a widely used dataset, i.e., NUS-WIDE. The MAP results are shown in Table 8. We believe that SHOH can achieve outstanding retrieval performance even on non-hierarchical datasets by constructing a reasonable label hierarchy.

5.2 Future Work

By comparing the results of SHOH and baselines, we are convinced that SHOH outperforms existing methods with the same initial features. In addition, deep features lead to improved performance, which has been taken into consideration. However, the method used in our experiments for deep feature extraction and hash mapping does not integrate them into a unified learning model. This indicates that there is still room for improvement in the accuracy of feature extraction. Research on deep online hashing methods is the focus of our future work.

6 CONCLUSION

In this article, we present a novel online cross-modal hashing method that incorporates hierarchical labels and semantic structure information into the online hash learning process. We construct cross-layer affiliations and use this relationship to generate soft similarity labels, which supervise the hashing in the online learning framework. Our method, SHOH, optimizes the interclass distances and achieves superior retrieval performance compared to state-of-the-art methods, as demonstrated by extensive experiments and verification on hierarchical and non-hierarchical datasets. It provides a more comprehensive and diverse cross-modal retrieval approach for the recommender systems.

ACKNOWLEDGMENTS

This work is achieved in Key Laboratory of Information Storage System, Ministry of Education of China.

REFERENCES

[1] Abdelhak Bentaleb, Ali C. Begen, and Roger Zimmermann. 2018. ORL-SDN: Online reinforcement learning for SDN-Enabled HTTP adaptive streaming. *ACM Trans. Multim. Comput. Commun. Appl.* 14, 3 (2018), 71:1–71:28. DOI: https://doi.org/10.1145/3219752

[2] Fatih Çakir, Kun He, Sarah Adel Bargal, and Stan Sclaroff. 2017. MIHash: Online hashing with mutual information. In *IEEE International Conference on Computer Vision*. IEEE Computer Society, 437–445. DOI: https://doi.org/10.1109/ICCV.2017.55

[3] Xixian Chen, Irwin King, and Michael R. Lyu. 2017. FROSH: FasteR online sketching hashing. In *33rd Conference on Uncertainty in Artificial Intelligence*, Gal Elidan, Kristian Kersting, and Alexander Ihler (Eds.). AUAI Press. Retrieved from http://auai.org/uai2017/proceedings/papers/12.pdf

[4] Yudong Chen, Sen Wang, Jianglin Lu, Zhi Chen, Zheng Zhang, and Zi Huang. 2021. Local graph convolutional networks for cross-modal hashing. In *ACM Multimedia Conference*, Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo César, Florian Metze, and Balakrishnan Prabhakaran (Eds.). ACM, 1921–1928. DOI: https://doi.org/10.1145/3474085.3475346

[5] 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 *8th ACM International Conference on Image and Video Retrieval*, Stéphane Marchand-Maillet and Yiannis Kompatsiaris (Eds.). ACM. DOI: https://doi.org/10.1145/1646396.1646452

[6] Roxane Desrousseaux, Gilles Bernard, and Jean-Jacques Mariage. 2021. Predicting financial suspicious activity reports with online learning methods. In *IEEE International Conference on Big Data*, Yixin Chen, Heiko Ludwig, Yicheng Tu, Usama M. Fayyad, Xingquan Zhu, Xiaohua Hu, Suren Byna, Xiong Liu, Jianping Zhang, Shirui Pan, Vagelis Papalexakis, Jianwu Wang, Alfredo Cuzzocrea, and Carlos Ordonez (Eds.). IEEE, 1595–1603. DOI: https://doi.org/10.1109/BigData52589.2021.9671716

[7] Chen-Lu Ding, Xin Luo, Xiao-Ming Wu, Yu-Wei Zhan, Rui Li, Hui Zhang, and Xin-Shun Xu. 2022. Weakly-supervised online hashing with refined pseudo tags. In *31st ACM International Conference on Information & Knowledge Management*, Mohammad Al Hasan and Li Xiong (Eds.). ACM, 375–385. DOI: https://doi.org/10.1145/3511808.3557488

[8] Arindam Jati and Dimitra Emmanouilidou. 2020. Supervised deep hashing for efficient audio event retrieval. In *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 4497–4501. DOI: https://doi.org/10.1109/ICASSP40776.2020.9053766

[9] Sheng Jin, Qin Zhou, Hongxun Yao, Yao Liu, and Xian-Sheng Hua. 2021. Asynchronous teacher guided bit-wise hard mining for online hashing. In *35th AAAI Conference on Artificial Intelligence, 33rd Conference on Innovative Applications of Artificial Intelligence, 11th Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 1717–1724. DOI: https://doi.org/10.1609/aaai.v35i2.16265

[10] Theodoros Karagkioules, Georgios S. Paschos, Nikolaos Liakopoulos, Attilio Fiandrotti, Dimitrios Tsilimantos, and Marco Cagnazzo. 2022. Online learning for adaptive video streaming in mobile networks. *ACM Trans. Multim. Comput. Commun. Appl.* 18, 1 (2022), 2:1–2:22. DOI: https://doi.org/10.1145/3460819

[11] Cong Leng, Jiaxiang Wu, Jian Cheng, Xiao Bai, and Hanqing Lu. 2015. Online sketching hashing. In *IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2503–2511. DOI: https://doi.org/10.1109/CVPR.2015.7298865

[12] Shuyan Li, Xiu Li, Jiwen Lu, and Jie Zhou. 2021. Self-supervised video hashing via bidirectional transformers. In *IEEE Conference on Computer Vision and Pattern Recognition*. Computer Vision Foundation/IEEE, 13549–13558. DOI: https://doi.org/10.1109/CVPR46437.2021.01334

[13] Haitao Lin, Min Meng, and Jigang Wu. 2022. Online robust specific and consistent hashing. In *IEEE International Conference on Multimedia and Expo*. IEEE, 1–6. DOI: https://doi.org/10.1109/ICME52920.2022.9859620

[14] Mingbao Lin, Rongrong Ji, Hong Liu, Xiaoshuai Sun, Shen Chen, and Qi Tian. 2020. Hadamard matrix guided online hashing. *Int. J. Comput. Vis.* 128, 8 (2020), 2279–2306. DOI: https://doi.org/10.1007/s11263-020-01332-z

[15] Mingbao Lin, Rongrong Ji, Xiaoshuai Sun, Baochang Zhang, Feiyue Huang, Yonghong Tian, and Dacheng Tao. 2022. Fast class-wise updating for online hashing. *IEEE Trans. Pattern Anal. Mach. Intell.* 44, 5 (2022), 2453–2467. DOI: https://doi.org/10.1109/TPAMI.2020.3042193

[16] Shiguang Liu and Ziqing Huang. 2020. Efficient image hashing with geometric invariant vector distance for copy detection. *ACM Trans. Multim. Comput. Commun. Appl.* 15, 4 (2020), 106:1–106:22. DOI: https://doi.org/10.1145/3355394

[17] Xin Liu, Zhikai Hu, Haibin Ling, and Yiu-Ming Cheung. 2021. MTFH: A matrix tri-factorization hashing framework for efficient cross-modal retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* 43, 3 (2021), 964–981. DOI: https://doi.org/10.1109/TPAMI.2019.2940446

[18] Xin Liu, Jinhan Yi, Yiu-ming Cheung, Xing Xu, and Zhen Cui. 2023. OMGH: Online manifold-guided hashing for flexible cross-modal retrieval. *IEEE Trans. Multim.* 25 (2023), 3811–3824. DOI: https://doi.org/10.1109/TMM.2022.3166668

[19] Yu Liu, Yangtao Wang, Jingkuan Song, Chan Guo, Ke Zhou, and Zhili Xiao. 2020. Deep self-taught graph embedding hashing with pseudo labels for image retrieval. In *IEEE International Conference on Multimedia and Expo*. IEEE, 1–6. DOI: https://doi.org/10.1109/ICME46284.2020.9102819

[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 *27th ACM International Conference on Multimedia*, Laurent Amsaleg, Benoit Huet, Martha A. Larson, Guillaume Gravier, Hayley Hung, Chong-Wah Ngo, and Wei Tsang Ooi (Eds.). ACM, 1129–1137. DOI: https://doi.org/10.1145/3343031.3350999

[21] George A. Miller. 1995. WordNet: A lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41. DOI: https://doi.org/10.1145/219717.219748

[22] Daan Odijk and Anne Schuth. 2017. Online learning to rank for recommender systems. In *11th ACM Conference on Recommender Systems*, Paolo Cremonesi, Francesco Ricci, Shlomo Berkovsky, and Alexander Tuzhilin (Eds.). ACM, 348. DOI: https://doi.org/10.1145/3109859.3109925

[23] Aude Oliva and Antonio Torralba. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.* 42, 3 (2001), 145–175. DOI: https://doi.org/10.1023/A:1011139631724

[24] Heng Tao Shen, Luchen Liu, Yang Yang, Xing Xu, Zi Huang, Fumin Shen, and Richang Hong. 2021. Exploiting subspace relation in semantic labels for cross-modal hashing. *IEEE Trans. Knowl. Data Eng.* 33, 10 (2021), 3351–3365. DOI: https://doi.org/10.1109/TKDE.2020.2970050

[25] Ling Shen, Richang Hong, Haoran Zhang, Xinmei Tian, and Meng Wang. 2020. Video retrieval with similarity-preserving deep temporal hashing. *ACM Trans. Multim. Comput. Commun. Appl.* 15, 4 (2020), 109:1–109:16. DOI: https://doi.org/10.1145/3356316

[26] Zhenqiu Shu, Li Li, Jun Yu, Donglin Zhang, Zhengtao Yu, and Xiaojun Wu. 2023. Online supervised collective matrix factorization hashing for cross-modal retrieval. *Appl. Intell.* 53, 11 (2023), 14201–14218. DOI: https://doi.org/10.1007/s10489-022-04189-6

[27] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, Yoshua Bengio and Yann LeCun (Eds.). Retrieved from http://arxiv.org/abs/1409.1556

[28] Xuemeng Song, Fuli Feng, Xianjing Han, Xin Yang, Wei Liu, and Liqiang Nie. 2018. Neural compatibility modeling with attentive knowledge distillation. In *41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, Kevyn Collins-Thompson, Qiaozhu Mei, Brian D. Davison, Yiqun Liu, and Emine Yilmaz (Eds.). ACM, 5–14. DOI: https://doi.org/10.1145/3209978.3209996

[29] Changchang Sun, Xuemeng Song, Fuli Feng, Wayne Xin Zhao, Hao Zhang, and Liqiang Nie. 2019. Supervised hierarchical cross-modal hashing. In *42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Benjamin Piwowarski, Max Chevalier, Éric Gaussier, Yoelle Maarek, Jian-Yun Nie, and Falk Scholer (Eds.). ACM, 725–734. DOI: https://doi.org/10.1145/3331184.3331229

[30] Dan Wang, Heyan Huang, Chi Lu, Bo-Si Feng, Guihua Wen, Liqiang Nie, and Xianling Mao. 2018. Supervised deep hashing for hierarchical labeled data. In *32nd AAAI Conference on Artificial Intelligence, 30th Innovative Applications of Artificial Intelligence, and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence*, Sheila A. McIlraith and Kilian Q. Weinberger (Eds.). AAAI Press, 7388–7395. DOI: https://doi.org/10.1609/aaai.v32i1.12247

[31] Di Wang, Quan Wang, Yaqiang An, Xinbo Gao, and Yumin Tian. 2020. Online collective matrix factorization hashing for large-scale cross-media retrieval. In *43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (Eds.). ACM, 1409–1418. DOI: https://doi.org/10.1145/3397271.3401132

[32] Di Wang, Caiping Zhang, Quan Wang, Yumin Tian, Lihuo He, and Lin Zhao. 2023. Hierarchical semantic structure preserving hashing for cross-modal retrieval. *IEEE Trans. Multim.* 25 (2023), 1217–1229. DOI: https://doi.org/10.1109/TMM.2022.3140656

[33] Song Wang, Huan Zhao, and Keqin Li. 2022. Discrete joint semantic alignment hashing for cross-modal image-text search. *IEEE Trans. Circ. Syst. Video Technol.* 32, 11 (2022), 8022–8036. DOI: https://doi.org/10.1109/TCSVT.2022.3186714

[34] Xiaoqin Wang, Chen Chen, Rushi Lan, Licheng Liu, Zhenbing Liu, Huiyu Zhou, and Xiaonan Luo. 2022. Binary representation via jointly personalized sparse hashing. *ACM Trans. Multim. Comput. Commun. Appl.* 18, 3s (2022), 137:1–137:20. DOI: https://doi.org/10.1145/3558769

[35] Yongxin Wang, Xin Luo, and Xin-Shun Xu. 2020. Label embedding online hashing for cross-modal retrieval. In *28th ACM International Conference on Multimedia*, Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann (Eds.). ACM, 871–879. DOI: https://doi.org/10.1145/3394171.3413971

[36] Zhenyu Weng and Yuesheng Zhu. 2020. Online hashing with efficient updating of binary codes. In *34th AAAI Conference on Artificial Intelligence, 32nd Innovative Applications of Artificial Intelligence Conference, 10th AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 12354–12361. DOI: https://doi.org/10.1609/aaai.v34i07.6920

[37] Dayan Wu, Qi Dai, Bo Li, and Weiping Wang. 2023. Deep uncoupled discrete hashing via similarity matrix decomposition. *ACM Trans. Multim. Comput. Commun. Appl.* 19, 1 (2023), 22:1–22:22. DOI: https://doi.org/10.1145/3524021

[38] Xiao-Ming Wu, Xin Luo, Yu-Wei Zhan, Chen-Lu Ding, Zhen-Duo Chen, and Xin-Shun Xu. 2022. Online enhanced semantic hashing: Towards effective and efficient retrieval for streaming multi-modal data. In *36th AAAI Conference on Artificial Intelligence, 34th Conference on Innovative Applications of Artificial Intelligence, 12th Symposium on Educational Advances in Artificial Intelligence*. AAAI Press, 4263–4271. DOI: https://doi.org/10.1609/aaai.v36i4.20346

[39] Liang Xie, Jialie Shen, Jungong Han, Lei Zhu, and Ling Shao. 2017. Dynamic multi-view hashing for online image retrieval. In *26th International Joint Conference on Artificial Intelligence*, Carles Sierra (Ed.). ijcai.org, 3133–3139. DOI: https://doi.org/10.24963/ijcai.2017/437

[40] Liang Xie, Jialie Shen, and Lei Zhu. 2016. Online cross-modal hashing for web image retrieval. In *30th AAAI Conference on Artificial Intelligence*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 294–300. DOI: https://doi.org/10.1609/aaai.v30i1.9982

[41] Yanzhao Xie, Yu Liu, Yangtao Wang, Lianli Gao, Peng Wang, and Ke Zhou. 2020. Label-attended hashing for multi-label image retrieval. In *29th International Joint Conference on Artificial Intelligence*, Christian Bessiere (Ed.). ijcai.org, 955–962. DOI: https://doi.org/10.24963/ijcai.2020/133

[42] Cheng Yan, Xiao Bai, Shuai Wang, Jun Zhou, and Edwin R. Hancock. 2019. Cross-modal hashing with semantic deep embedding. *Neurocomputing* 337 (2019), 58–66. DOI: https://doi.org/10.1016/j.neucom.2019.01.040

[43] Tao Yao, Gang Wang, Lianshan Yan, Xiangwei Kong, Qingtang Su, Caiming Zhang, and Qi Tian. 2019. Online latent semantic hashing for cross-media retrieval. *Pattern Recog.* 89 (2019), 1–11. DOI: https://doi.org/10.1016/j.patcog.2018.12.012

[44] Zhaoda Ye and Yuxin Peng. 2020. Sequential cross-modal hashing learning via multi-scale correlation mining. *ACM Trans. Multim. Comput. Commun. Appl.* 15, 4 (2020), 105:1–105:20. DOI: https://doi.org/10.1145/3356338

[45] Jinhan Yi, Xin Liu, Yiu-ming Cheung, Xing Xu, Wentao Fan, and Yi He. 2021. Efficient online label consistent hashing for large-scale cross-modal retrieval. In *IEEE International Conference on Multimedia and Expo*. IEEE, 1–6. DOI: https://doi.org/10.1109/ICME51207.2021.9428323

[46] En Yu, Jianhua Ma, Jiande Sun, Xiaojun Chang, Huaxiang Zhang, and Alexander G. Hauptmann. 2022. Deep discrete cross-modal hashing with multiple supervision. *Neurocomputing* 486 (2022), 215–224. DOI: https://doi.org/10.1016/j.neucom.2021.11.035

[47] Yu-Wei Zhan, Xin Luo, Yongxin Wang, and Xin-Shun Xu. 2020. Supervised hierarchical deep hashing for cross-modal retrieval. In *28th ACM International Conference on Multimedia*, Chang Wen Chen, Rita Cucchiara, Xian-Sheng Hua, Guo-Jun Qi, Elisa Ricci, Zhengyou Zhang, and Roger Zimmermann (Eds.). ACM, 3386–3394. DOI: https://doi.org/10.1145/3394171.3413962

[48] Yu-Wei Zhan, Yongxin Wang, Yu Sun, Xiao-Ming Wu, Xin Luo, and Xin-Shun Xu. 2022. Discrete online cross-modal hashing. *Pattern Recog.* 122 (2022), 108262. DOI: https://doi.org/10.1016/j.patcog.2021.108262

[49] Donglin Zhang, Xiaojun Wu, and Jun Yu. 2021. Label consistent flexible matrix factorization hashing for efficient cross-modal retrieval. *ACM Trans. Multim. Comput. Commun. Appl.* 17, 3 (2021), 90:1–90:18. DOI: https://doi.org/10.1145/3446774

[50] Jian Zhang and Yuxin Peng. 2018. Query-adaptive image retrieval by deep-weighted hashing. *IEEE Trans. Multim.* 20, 9 (2018), 2400–2414. DOI: https://doi.org/10.1109/TMM.2018.2804763

[51] Zheng Zhang, Jianning Wang, Lei Zhu, and Guangming Lu. 2022. Discriminative visual similarity search with semantically cycle-consistent hashing networks. *ACM Trans. Multim. Comput. Commun. Appl.* 18, 2s (2022), 114:1–114:21. DOI: https://doi.org/10.1145/3532519

[52] Lei Zhu, Xu Lu, Zhiyong Cheng, Jingjing Li, and Huaxiang Zhang. 2020. Deep collaborative multi-view hashing for large-scale image search. *IEEE Trans. Image Process.* 29 (2020), 4643–4655. DOI: https://doi.org/10.1109/TIP.2020.2974065

[53] Yaoxin Zhuo, Yikang Li, Jenhao Hsiao, Chiuman Ho, and Baoxin Li. 2022. CLIP4Hashing: Unsupervised deep hashing for cross-modal video-text retrieval. In *International Conference on Multimedia Retrieval*, Vincent Oria, Maria Luisa Sapino, Shin'ichi Satoh, Brigitte Kerhervé, Wen-Huang Cheng, Ichiro Ide, and Vivek K. Singh (Eds.). ACM, 158–166. DOI: https://doi.org/10.1145/3512527.3531381

Received 17 May 2023; revised 19 September 2023; accepted 26 October 2023