Unsupervised Video Hashing with Multi-granularity Contextualization and Multi-structure Preservation

Yanbin Hao
University of Science and Technology of China
China
haoyanbin@hotmail.com

Jingru Duan
University of Science and Technology of China
China
duanjr@mail.ustc.edu.cn

Hao Zhang*
Singapore Management University
Singapore
zhanghaoinf@gmail.com

Bin Zhu
University of Bristol
United Kingdom
andrewzhu1216@gmail.com

Pengyuan Zhou
University of Science and Technology of China
China
pyzhou@ustc.edu.cn

Xiangnan He
University of Science and Technology of China
China
xiangnanhe@gmail.com

ABSTRACT

Unsupervised video hashing typically aims to learn a compact binary vector to represent complex video content without using manual annotations. Existing unsupervised hashing methods generally suffer from incomplete exploration of various perspective dependencies (e.g., long-range and short-range) and data structures that exist in visual contents, resulting in less discriminative hash codes. In this paper, we propose a ***M**ulti-granularity **C**ontextualized and **M**ulti-**S**tructure preserved **H**ashing (**MCMSH**)* method, exploring multiple axial contexts for discriminative video representation generation and various structural information for unsupervised learning simultaneously. Specifically, we delicately design three self-gating modules to separately model three granularities of dependencies (i.e., long/middle/short-range dependencies) and densely integrate them into MLP-Mixer for feature contextualization, leading to a novel model MC-MLP. To facilitate unsupervised learning, we investigate three kinds of data structures, including clusters, local neighborhood similarity structure, and inter/intra-class variations, and design a multi-objective task to train MC-MLP. These data structures show high complementarities in hash code learning. We conduct extensive experiments using three video retrieval benchmark datasets, demonstrating that our MCMSH not only boosts the performance of the backbone MLP-Mixer significantly but also outperforms the competing methods notably. Code is available at https://github.com/haoyanbin918/MCMSH.

CCS CONCEPTS

• **Computing methodologies** → **Visual content-based indexing and retrieval**.

* Hao Zhang is the corresponding author.

MM '22, October 10–14, 2022, Lisboa, Portugal

ACM ISBN 978-1-4503-9203-7/22/10…$15.00
https://doi.org/10.1145/3503161.3547836

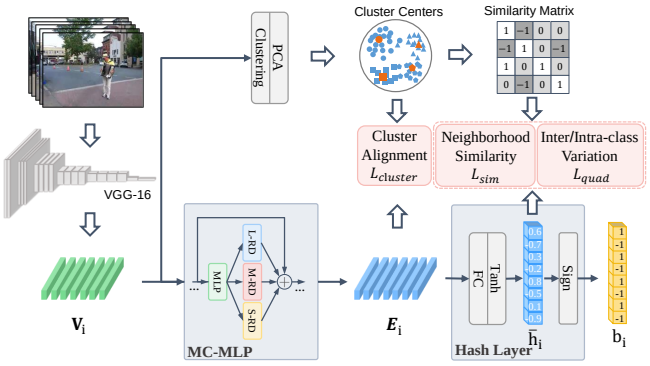


Figure 1: Overall structure of the proposed MCMSH. The MC-MLP module is designed to learn discriminative video representations by capturing multiple granularities of axial contexts. Three kinds of unsupervised learning objectives are developed to optimize the hash code generation.

KEYWORDS

Hashing, feature contextualization, unsupervised learning, video retrieval

ACM Reference Format:
Yanbin Hao, Jingru Duan, Hao Zhang*, Bin Zhu, Pengyuan Zhou, and Xiangnan He. 2022. Unsupervised Video Hashing with Multi-granularity Contextualization and Multi-structure Preservation. In *Proceedings of the 30th ACM International Conference on Multimedia (MM '22), October 10–14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3503161.3547836

1 INTRODUCTION

Hashing is a technique that generates low-dimensional, compact binary codes that convey the information within data. Its main advantages include low storage cost and high matching speed, greatly boosting the development of applications such as data retrieval [9, 11, 26, 28, 34, 37, 48, 54, 60], data indexing [8, 47], digital signatures [7, 10], etc. This paper focuses on learning compact hash codes for video data in an unsupervised manner. Since video contents are much richer and more complex due to 3-dimensional spatio-temporal variations, and the huge and rapidly increasing amount of

video data make manual labelling impossible, unsupervised video hashing, thereby, is becoming more challenging and continuing to engage research attentions.

The main challenge of unsupervised video hashing is twofold: (1) how to maximally convey the spatio-temporal contents presented in video to a limited number of binary codes and (2) how to accurately preserve the relevance structure within video data for a retrieval task. The first one is challenging because that video is a temporal sequence of image frames and the extra time dimension greatly enriches the visual variations. Current video hashing approaches [4, 25, 38, 49, 51] address this challenge by mainly following the pipeline that firstly uses a feature extractor, e.g., convolutional neural networks (CNNs), to generate frame-level embedding for all frames, and then applies a temporal calculational unit, e.g., recurrent neural networks (RNNs) or transformers [46], to model temporal relations among frames. For example, the works [24, 57] utilize long short-term memory (LSTM) network [18] to encode the temporal correlations. As RNNs process the video frame by frame, the influence of the current frame on the latter frames would degenerate with the growing of temporal distance, preferring the short-range context information. Recently, attention mechanisms, e.g., transformers [46], show promising performance on various research tasks [6, 56]. The work [26] adopts the bidirectional transformer to model correlations among frames. Transformers holistically process a video sequence and conduct a pair-wise comparison between every frame, building long-range context interactions. However, the pairwise comparison of transformers may also increase the computational complexity much.

Existing works [13, 23, 25, 26, 57] address the second issue by approximating the relevance structure within videos and preserving it in the hash code space, because the relevance or similarity information is crucial for a retrieval task. In general, these works mainly explore one or two types of structural information. For example, [23] explores both video appearance and temporal structures to learn hash codes with the use of autoencoder. [13, 25] turn to build the neighborhood similarities for all training video instances and preserve them through a structure reconstruction loss. [26] extends [25] by additionally introducing cluster alignment for hidden representation encoding. In terms of the retrieval performance, neighborhood preservation used by [13, 25, 26] shows more promising results than feature reconstruction [23, 57]. Nevertheless, it is non-trivial to completely express the relevance structures with one or two types of structural simulations. Consequently, the exploration of structural information is still an open question for unsupervised video hashing.

To jointly address the aforementioned two limitations for unsupervised video hashing, specifically hash code learning and unsupervised training, we propose a ***multi-granularity contextualized and multi-structure preserved hashing*** (**MCMSH**) method. Figure 1 illustrates the overall structure of MCMSH. Firstly, three self-gating modules are purposely designed to model various granularities of video feature contexts and incorporated densely into the multi-layer perceptron (MLP) based mixer (MLP-Mixer) [44] to derive accurate and compact video hash codes, resulting in a novel video hashing model multi-granularity contextualized MLP (MC-MLP). Particularly, the three self-gating modules separately capture long-range, middle-range and short-range axial dependencies (contexts) from video features, referred to as L-RD, M-RD, and S-RD modules, and refine parallelly the MLP features with element-wise multiplication. Contexts used by them are feature dynamics aggregated from different perspectives and potentially can model diverse contents of videos. Secondly, to facilitate unsupervised learning, we investigate various types of structural patterns within the data, including cluster information, neighborhood similarity, and inter/intra-class variations, to guide both the hidden representation encoding and hash code generation of MC-MLP. These structural patterns explicitly cover a broad range of relevance structures, such as local neighborhood structure, intra-class closeness and inter-class separation, and are maximally preserved in the hash code space through the multi-objective optimization, which thereby can boost the video retrieval task significantly.

We summarize the contributions as follows:

- **Multi-granularity contextualized MLP**. We propose a new model named multi-granularity contextualized MLP (MC-MLP) for compact hash codes learning for videos. MC-MLP contains three self-gating modules (L/M/S-RD) to explicitly model multi-granularity contexts and refine video features in parallel.
- **Multi-structure preservation**. We explore multiple structural patterns from video data to facilitate unsupervised learning. These structures accurately approximate the underlying similarity and group patterns, and are preserved as many as possible in the Hamming space.
- **Superior retrieval performance**. We verify that our MC-MLP can significantly improve the performance of the backbone MLP-Mixer on the hashing-based video retrieval task, and the proposed MCMSH with multi-structure preservation also achieves either state-of-the-art or comparable retrieval results on three commonly used video benchmark datasets.

2 RELATED WORK

Our work is mainly relevant to unsupervised video hashing literatures. Below, we organize the review in the order of classical machine learning based approaches and advanced deep learning based approaches primarily, clarifying both model architectures and learning strategies.

Classical machine learning based. The classical machine learning based hashing methods mainly extract static image-level features from video frames, such as global histogram (e.g., HSV [52]) and local pattern features (SIFT [33], LBP [58]), and then use a hash function to compute binary hash codes. Representative works include spectral hashing (SH) [50], self-taught hashing (STH) [55], multiple feature hashing (MFH) [40], unsupervised stochastic multi-view hashing (USMVH) [13] and its extension t-distributed USMVH (t-USMVH) [12]. SH and STH use a single feature to represent video frames and learn hash code or hash function by schemes such as the binarization of the eigenvectors of the graph. MFH, USMVH, and t-USMVH extract multiple features and learn hash codes by manually weighting the importance of different types of feature sources. For model optimization, MFH is also based on spectral factorization, while USMVH and t-USMVH use a composite Kullback-Leibler (KL)

divergence to achieve similarity structure preservation from feature space to Hamming space.

Advanced deep learning based. In recent years, deep learning technologies, e.g., MLPs [44], CNNs [17], RNNs and transformers [46], have gained significant breakthroughs and greatly boosted the development of various multimedia related research tasks. In the studied hashing field, there are also a lot of works [4, 5, 23–27, 41, 51, 57] that use deep neural networks for either/both feature extraction or/and hash code generation. The most previous approaches include, to name a few, [4] and [51], where [4] simply squeezes the CNN features over frames to obtain the video representation while [51] additionally extracts the optical flow features to enhance the recognition for temporal variations. Since video contents are dynamics presented in sequential order, both spatial and temporal clues should be considered jointly. Subsequently, benefiting from RNN or its variants [18, 59] which inherently possess the ability of modeling temporal relations, current video hashing approaches design their model architecture following the form of "CNN+RNN", for example, self-supervised temporal hashing (SSTH) [57], joint appearance and temporal encoding (JTAE) [23], self-supervised video hashing (SSVH) [41], neighborhood preserving hashing (NPH) [25], structure-adaptive neighborhood preserving hashing (SNPH) [27] and unsupervised variational video hashing (UVVH) [24]. To achieve more reliable temporal information capturing, BTH [26] employs the bidirectional transformers to fully exploit the long-range bidirectional correlations among frames. In terms of the model learning, most of the above works investigate one or two types of similarity structures. For example, SSTH [57] and JTAE [23] use the temporal order of the video sequence as a self-supervision for learning to hash. SSVH [41], NPH [25], SNPH [27] and BTH [26] learn to reconstruct the similarity structure built with the original features from the relaxed hash codes, and SNPH and BTH further consider the content reconstruction (SNPH) and cluster alignment (BTH). UVVH [24] applies a probabilistic latent loss to approximate the posterior distribution learned by the model to a predefined prior.

The proposed multi-granularity contextualized MLP (MC-MLP) model originates from MLP-Mixer [44], which uses the token-mixing layers to capture cross-token interactions. Recently, there are also some vision MLPs that improve the MLP-Mixer, such as MLP with gating (gMLP) [30], axial shifted MLP (AS-MLP) [29], spatial shift MLP (s^2-MLP) [53] and Permute-MLP [19]. These vision MLPs aim at building efficient token-mixing operations. Whereas, our MC-MLP turns to improve MLP-Mixer by introducing various axial contexts.

3 PROPOSED METHOD

An overview of the proposed unsupervised video hashing method (i.e., multi-granularity contextualized and multi-structure preserved hashing, MCMSH for abbreviation) is illustrated in Figure 1. The model architecture of MCMSH is based primarily on MLP-Mixer and further enhanced with diverse axial (frame and channel) contexts. To accomplish unsupervised learning, we explore various types of structural information within videos, including clusters analysis, neighborhood similarity, and inter/intra-class variation, and preserve them in the learnt video hash code space.

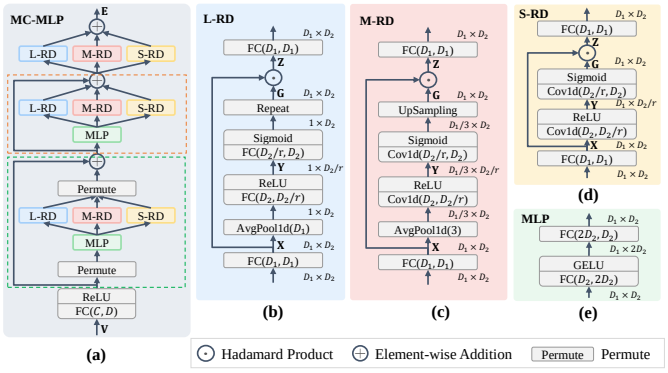


Figure 2: The schema of the proposed (a) MC-MLP model. It mainly consists of four blocks, i.e., (b) L-RD, (c) M-RD, (d) S-RD and (e) MLP.

3.1 Video Representation Encoding and Hash Code Generation

The notation for the video input is as follows. We are given a set of N training videos denoted by $\mathcal{V} = \{\mathbf{V}_i \in \mathbb{R}^{T\times C}\}_{i=1}^{N}$. Here, a video is represented by a $T \times C$ feature matrix $\mathbf{V}_i = \left[\boldsymbol{v}_1^i, \boldsymbol{v}_2^i, \cdots, \boldsymbol{v}_T^i\right]^T$, where T and C are the number of sampled frames (tokens in MLP-Mixer) and the channel dimensionality per frame respectively. In the implementation, we use the CNN features extracted from VGG [39] as the frame features. To generate hash codes for each video, we first introduce the proposed multi-granularity contextualized MLP (MC-MLP) to learn discriminative representation from the complex sequential video features and then design a hash layer on top of MC-MLP to generate compact hash codes. In addition, since the original video embeddings $\mathbf{V}_i$ generally possess a high channel dimensionality (e.g., 4,096), we use a fully connected (FC) layer activated by ReLU to reduce C to a relatively lower value D before feeding into the mixing layers of MC-MLP.

Multi-granularity contextualized MLP. MC-MLP improves the capability of modeling multi-granularity dependencies for MLP-Mixer by densely refining its per-mixing features with various dynamics aggregated from different feature axes. MLP-Mixer is originally proposed for image processing, which separately models the per-location (channel-mixing) interaction and cross-location (token-mixing) interaction by pure MLPs, achieving comparable performance with the advanced CNNs and transformers but requiring a lower computational cost. Since MLP-Mixer regards the data sample (image) as a set of sequential tokens with the tensor shape of $T \times D$, it can be naturally extended to facilitate the video representation learning from the $frame \times channel$ video feature map. However, although MLP-Mixer inherently models the interactions between channels or between tokens, it lacks of explicit exploration for various axial content dependencies which have been demonstrated to be promising in learning discriminative representation. To address this issue, we propose three kinds of self-gating modules, including, a **long-range dependency (L-RD)** module, a **middle-range dependency (M-RD)** module and a **short-range dependency (S-RD)** module, to jointly adjust the per-mixing feature maps. Details of the three module variants (L/M/S-RD) are presented below.

We derive the three self-gating modules following the squeeze-and-excitation paradigm, i.e., "{*Operators1*}-*ReLU*-{*Operators2*}-*Sigmoid*", proposed by SE-Net [20] and extended by [15, 16, 42]. Since MLP-Mixer contains two types of mixing layers, i.e., channel-mixing and token-mixing, we formulate the feature contextualization modules in a general fashion regardless of the mixing type. Their gating unit is defined as below. Formally, given the MLP-based input feature map $\mathbf{X} \in \mathbb{R}^{D_1 \times D_2}$, the refined feature $\mathbf{Z} \in \mathbb{R}^{D_1 \times D_2}$ is computed as follows

$$\mathbf{Y}_{U^{(B)}} = \text{ReLU}\left(f_{op1}\left(\mathbf{X}_{U^{(A)}}; \Phi_{op1}\right)\right), \quad (1)$$

$$\mathbf{G}_{U^{(A)}} = \text{Sigmoid}\left(f_{op2}\left(\mathbf{Y}_{U^{(B)}}; \Phi_{op2}\right)\right), \quad (2)$$

$$\mathbf{Z}_{U^{(A)}} = \mathbf{G}_{U^{(A)}} \circ \mathbf{X}_{U^{(A)}}, \quad (3)$$

$$U^{(A)} \subseteq \{(i,j)\}|_{i,j=1}^{D_1, D_2},$$

$$U^{(B)} \subseteq \{(i,j)\}|_{i,j=1}^{\frac{D_1}{R_1}, \frac{D_2}{R_2}},$$

where Φ_{op1} and Φ_{op2} are parameters for operators f_{op1} and f_{op2}, respectively; $\mathbf{Y}_{U^{(B)}}$ is the axial context and $\mathbf{G}_{U^{(A)}}$ is a gating mask; $U^{(A)}$ and $U^{(B)}$ are index sets for (i,j)-th element on the feature map; R_1, R_2 denote dimensionality reduction ratios caused by axial pooling and reduction. Through adjusting the dimensionality reduction ratios, multiple axial contexts can thus be obtained. For example, when instantiating f_{op1} with "AvgPool1d(D_1)" (i.e., $R_1 = D_1$) and "FC($D_2, \frac{D_2}{r}$)" (i.e., $R_2 = r$), the global context over D_1 axis is generated. "$\circ$" is the Hadamard product operator.

The **L-RD** module squeezes input along an axis by global 1D average pooling, representing axial context as a global vector, and uses two FC layers (f_{op1} and f_{op2}) to calculate the gating weights, as shown in Figure 2(b). Hereby, $R_1 = D_1$ and $R_2 = r$. The **M-RD** module uses a relative smaller pooling kernel (e.g., 3) to aggregate middle-scale dynamics, and facilitates f_{op1} and f_{op2} with 1D convolutions with a 3 kernel to assign the regional context, as shown in Figure 2(c), where $R_1 = 3$ and $R_2 = r$. Unlike L-RD and M-RD that adopt pooling operation to aggregate axial contexts, the **S-RD** module employs a convolutional operator (1D convolution with 3 kernel) to capture local contexts within a neighboring field, as shown in Figure 2(d). Hereby, $R_1 = 1$ and $R_2 = r$. Finally, three gating masks solely calculated by L/M/S-RD are summed up. It is worth noting that D_1 and D_2 could be the number of frames or channels. In addition, we also add a FC layer before and after the gating unit.

Hash layer. By inputting a video represented by $\mathbf{V}_i$ into the proposed MC-MLP, we can obtain its new $T \times D$ embedding matrix $\mathbf{E}_i$. To generate hash codes from $\mathbf{E}_i$, we firstly use a FC layer to project $\mathbf{E}_i$ to T K-dimensional real-valued vectors stored in $\mathbf{H}_i = \left[\boldsymbol{h}_1^i, \boldsymbol{h}_2^i, \cdots, \boldsymbol{h}_T^i\right]^T$, where K is the hash code length. Then, we fuse $\{\boldsymbol{h}_j^i\}_{j=1}^T$ via mean pooling and activate the fused values with $Tanh$ to the range of $(-1, 1)$, resulting in the relaxed hash code vector $\bar{\boldsymbol{h}}_i$. In the end, a $Sign$ function, given as $Sgn(x) = 1$ if $x \geq 0$ and $Sgn(x) = -1$ otherwise, is applied to convert the real-valued vector $\bar{\boldsymbol{h}}_i$ to a binary vector $\boldsymbol{b}_i$. The calculation is as follows:

$$\mathbf{H}_i = FC(\mathbf{E}_i, D \rightarrow K), \quad (4)$$

$$\bar{\boldsymbol{h}}_i = Tanh(\frac{1}{T}\sum_{t=1}^{T} \boldsymbol{h}_t^i), \quad (5)$$

$$\boldsymbol{b}_i = Sgn(\bar{\boldsymbol{h}}_i). \quad (6)$$

3.2 Structure Preservation for Unsupervised Learning

To facilitate unsupervised (or self-supervised) learning, we investigate various types of structural information from the video data, including cluster information, neighborhood similarity, and inter/intra-class variations.

Cluster approximation and alignment. The intrinsic cluster information plays an important role in various tasks, such as retrieval [28, 34] and classification [32]. Unlike the supervised training that cluster labels are given, there is no human-labeled semantics available for unsupervised learning. As a result, we need to seek a breakthrough from the data itself. Clustering algorithm is a potential way of producing pseudo labels, which learns statistical cohort characteristics without any human effort. In the implementation, we apply K-means on the training video instances $\mathcal{V}$ for cluster centers approximation. Specifically, we firstly adopt the average pooling on the T frame features $\{\boldsymbol{v}_t^i\}_{t=1}^T$ to obtain a video vector embedding $\bar{\boldsymbol{v}}_i$, and then conduct K-means to learn M cluster centers $\{\boldsymbol{u}_j\}_{j=1}^M$, where here PCA is also used for reducing the dimensionality of C to the same value D with $\boldsymbol{e}$. To align the learnt video representation $\bar{\boldsymbol{e}}_i = \frac{1}{T}\sum_{t=1}^T \boldsymbol{e}_t^i$ with its the nearest class center $\boldsymbol{u}_i^1$, we formulate the cluster loss $\mathcal{L}_{cluster}$ as MSE, that is

$$\mathcal{L}_{cluster} = \frac{1}{B}\sum_{i=1}^{B} \|\bar{\boldsymbol{e}}_i - \boldsymbol{u}_i^1\|_2^2, \quad (7)$$

where B is the training batch size.

Neighborhood similarity reconstruction. The cluster alignment brings video instances closer to its nearest class center but takes too little account of local neighborhood structure which is essential for the studied retrieval task [13, 14, 26]. Towards this goal, we firstly construct a pairwise similarity graph $\mathbf{S} \in \mathbb{R}^{N \times N}$ with each element representing the similarity of a video pair in the training data set, and then reconstruct the pairwise similarities in the embedded hash code space with the local neighborhood structure being wished to preserve.

We follow the works [26, 31] to accomplish the construction of $\mathbf{S}$. Specifically, for each video i represented by $\bar{\boldsymbol{v}}_i$, its m nearest cluster centers $\{\boldsymbol{u}_i^1, \boldsymbol{u}_i^2, \cdots, \boldsymbol{u}_i^m\}$ are picked out. Then, a truncated similarity matrix $\mathbf{P} \in \mathbb{R}^{N \times M}$ is calculated as

$$\mathbf{P}_{i,j} = \begin{cases} \dfrac{\exp(-\|\bar{\boldsymbol{v}}_i - \boldsymbol{u}_i^j\|_2^2/\sigma)}{\sum_{l=1}^m \exp(-\|\bar{\boldsymbol{v}}_i - \boldsymbol{u}_i^l\|_2^2/\sigma)}, & \forall \boldsymbol{u}_i^j \in \{\boldsymbol{u}_i^*\}, \\ 0, & \text{otherwise}, \end{cases} \quad (8)$$

where $\|\cdot\|_2$ denotes the l_2-norm (i.e., Euclidean distance), and σ is a bandwidth parameter. Afterwards, we compute an $N \times N$ adjacency matrix as $\mathbf{A} = \mathbf{P}\Lambda^{-1}\mathbf{P}^T$, where $\Lambda = diag(\mathbf{P}^T\mathbf{1}) \in \mathbb{R}^{M \times M}$. $\mathbf{A}$ is an nonnegative sparse symmetric matrix with elements of each row sum to 1, indicating a relevance structure computed based on the truncated similarities. Furthermore, we discretize $\mathbf{A}$ to a bi-level similarity matrix $\mathbf{A}'$ by setting $\mathbf{A}'_{ij} = 1$ if $\mathbf{A}_{ij} > 0$ and $\mathbf{A}'_{ij} = -1$ otherwise. The pairwise similarity matrices $\mathbf{A}$ and $\mathbf{A}'$ are determined by the number (i.e., m) of nearest cluster centers. Consequently, when setting m with different values, we can compute multiple similarity matrices that consider different relevance degrees among video instances. In practice, we follow [26] and set three numbers

for m, m_1, m_2, m_3 ($m_1 < m_2 < m_3$), resulting in three corresponding matrices $\mathbf{A}'^{(1)}$, $\mathbf{A}'^{(2)}$ and $\mathbf{A}'^{(3)}$. Based on them, we construct the ultimate similarity matrix $\mathbf{S}$ as

$$\mathbf{S}_{ij} = \begin{cases} 1, & \mathbf{A}'^{(1)} = 1, \\ -1, & \mathbf{A}'^{(2)} = -1 \text{ and } \mathbf{A}'^{(3)} = 1, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Compared with the bi-level similarity matrix $\mathbf{A}'$, the fused similarity matrix $\mathbf{S}$ adds a new similarity degree of 0 to separate boundary cases. Given a video query, boundary cases are mostly dissimilar within small neighborhoods (e.g., m_1, m_2) but possess similarity in a larger area (e.g., m_3). So, to enhance the cluster separability, we forcefully pull them away from each other by assigning the similarity score with -1.

In the hash code space, we compute the pairwise similarity between the relaxed hash code vectors $\bar{\boldsymbol{h}}_i$ and $\bar{\boldsymbol{h}}_j$ as dot product (cosine similarity) $\frac{1}{K}\bar{\boldsymbol{h}}_i^T\bar{\boldsymbol{h}}_j$, considering the fact that *Sgn* function makes the optimization intractable. However, to reduce the gap between the relaxed real-valued codes and the binary codes, we also involve a quantization error as a penalty term. The final neighborhood similarity reconstruction loss is as follows

$$\mathcal{L}_{sim} = \frac{1}{B}\sum_{s_{ij}\in\mathbf{S}}(\frac{1}{K}\bar{\boldsymbol{h}}_i^T\bar{\boldsymbol{h}}_j - s_{ij})^2 + 0.1\frac{1}{B}\sum_{i=1}^{B}\|\boldsymbol{b}_i - \bar{\boldsymbol{h}}_i\|_2^2. \tag{10}$$

Since the hash code vector b_i is computed by *Sgn* which is non-differentiable, we use straight-through estimator (STE) [1] to approximate the derivative calculation.

Inter/Intra-class variation control. Neighborhood similarity structure reflects the local pairwise relevance in the (relaxed) hash code space. As a complement to the structure, we additionally use a quadruplet ranking loss to achieve a large inter-class variation and a small intra-class variation among the binary representations. This loss has been demonstrated to be beneficial to the generalization from the training set to the testing set [3]. Its formulation is as below:

$$\begin{aligned}\mathcal{L}_{quad} = &\frac{1}{N}\sum_{i,j,k}^{N}\max(0, \|\bar{\boldsymbol{h}}_i - \bar{\boldsymbol{h}}_j\|_2^2 - \|\bar{\boldsymbol{h}}_i - \bar{\boldsymbol{h}}_k\|_2^2 + \alpha_1), \\ &+ \frac{1}{N}\sum_{i,j,k,l}^{N}\max(0, \|\bar{\boldsymbol{h}}_i - \bar{\boldsymbol{h}}_j\|_2^2 - \|\bar{\boldsymbol{h}}_l - \bar{\boldsymbol{h}}_k\|_2^2 + \alpha_2),\end{aligned} \tag{11}$$

where the (i, j) video pair is a positive pair and others such as (i, l), (i, k) and (l, k) are negative pairs, and α_1 and α_2 are margins set empirically following [35] ((α_1, α_2) is set as $\{(4, 0.25), (8, 0.5), (16, 1)\}$ for hash code length $\{16, 32, 64\}$ bits.). The quadruplet loss is modified based on the triplet loss [36]. The first term in $\mathcal{L}_{quad}$ shares the same idea with triplet loss that obtains the correct orders for positive (e.g., (i, j)) and negative pairs (e.g., (i, k)) w.r.t the same probe video. The additional second term brings a new constraint to push away negative pairs (e.g., (l, k)) from positive pairs (e.g., (i, j)) but w.r.t different probe videos. As a result, this quadruplet loss can control the minimum inter-class distance to be larger than the maximum intra-class distance regardless of whether pairs contain the same probe.

The studied three data structures (i.e., neighborhood, cluster, and inter/intra-class variation) show different structural patterns in hashing code learning. First, the neighborhood, which reflects pairwise similarities between videos, is a general consensus among existing hashing works [12, 13, 25, 27]. However, considering neighborhood alone has a drawback in that the distribution of the holistic samples could not be captured within a small training batch [26, 28, 32, 39]. As a complementary, the clustering technique captures statistics reflecting cohort characteristics of whole samples. Besides, the loss of cluster alignment would pull a sample point closer to its assigned cluster, encouraging the correct prediction of clusters. Finally, both the neighborhood and clustering do not focus on promoting high inter-class separability, which has been verified to be useful in recognition & re-identification tasks [3, 36]. Inspiringly, we adopt the quadruplet loss to maintain small intra-class and large inter-class variations, also known as ranking orders. Besides, we conduct experiments on pair-wise combinations of the three structures and observe their complementary nature.

Model optimization. The optimization of MCMSH is based on a multi-objective formulation combining the three proposed loss functions $\mathcal{L}_{cluster}$, $\mathcal{L}_{sim}$ and $\mathcal{L}_{quad}$. By arranging them with balancing parameters, we have the overall loss

$$\mathcal{L}_{all} = \alpha\mathcal{L}_{cluster} + \beta\mathcal{L}_{sim} + \gamma\mathcal{L}_{quad}. \tag{12}$$

4 EXPERIMENT

We conduct extensive experiments on three standard benchmarks for video retrieval and evaluate the performance with the mean average precision at top-k retrieved results (mAP@k) mainly.

4.1 Datasets

- **FCVID**. FCVID [21] consists of 91,223 videos for 239 manually annotated categories. Videos in this dataset show various activities, objects, events, etc. We follow [41] to use 91,185 videos, where 45,585 videos for training and the other 45,600 videos for evaluation.
- **ActivityNet**. ActivityNet [2] has 28k videos of 203 activity categories collected from YouTube. We follow [25] to use 9,722 videos for training, and 4,758 videos for evaluation of which 1,000 videos are as queries and the rest 3,758 videos are as retrieval database.
- **YFCC**. YFCC [43] consists of 0.8M video clips for 80 categories. We follow [57] to use 101,256 videos, where 409,788 videos for training and 101,256 videos for evaluation.

4.2 Implementation Details

During **feature preparation**, for FCVID and YFCC datasets, we sample 25 (i.e., $T = 25$) frames for each video, and use VGG-16 [39] network to extract 4096-D (i.e., $C = 4096$) frame-level features. For ActivityNet dataset, we follow [23] to sample 30 (i.e., $T = 30$) frames per video, and use ResNet-50 [17] to extract 2048-D (i.e., $C = 2048$) frame-level features. The above features are provided by [41] (FCVID and YFCC) and [26] (ActivityNet) respectively. For **model architecture** design, we set the reduced feature dimensionality as $D = 256$, and the reduction ratio r used in L/M/S-RD modules to 16 when integrating into channel-mixing block and 4 when integrating into token-mixing block. For **multi-objective loss functions**, we empirically fix the number of clusters produced by

K-means to 2,000 and the nearest neighbors $m_1 = 3, m_2 = 4, m_3 = 5$ following [26]. The balancing parameters α, β, γ used in Eq. (12) are fine-tuned and finally set as $\alpha = 0.8, \beta = 0.1, \gamma = 0.01$. For **model optimization**, the training settings are set as: 55 epochs with batch size 256, learning rate (lr) 0.0003, decaying lr by 0.9 at epoch 20 and 40, and Adam [22] optimizer with momentum 0.9.

4.3 Ablation Study

In this section, we study the impacts of hyperparameters, including self-gating modules (i.e., L/M/S-RD) of MC-MLP, the dimension reduction ratio r used in MC-MLP, and balancing factors $\{\alpha, \beta, \gamma\}$ in the ultimate loss function Eq. (12), using FCVID dataset.

Impact of L/M/S-RD modules. We firstly verify the impacts of the proposed self-gating module types. Table 1 demonstrates the performances of baseline and MC-MLP variants with different L/M/S-RD and combined modules. Here, results are obtained by fixing the objective balancing parameters $\alpha = 0.8, \beta = 0.1$ and $\gamma = 0.01$. We observe that the proposed self-gating modules, regardless of their types, consistently improve the base network (i.e., MLP-Mixer), indicating their effectiveness (e.g., 0.288→0.298 mAP@20 with L-RD, 0.288→0.298 mAP@20 with M-RD and 0.288→0.295 mAP@20 with S-RD). Moreover, compared to the single self-gating module, the MC-MLP, which combines the three L/M/S-RD modules in parallel, achieves much better performance with the same backbone (0.302 vs. 0.288 of backbone with ∼5% relative performance improvement).

Impact of dimension reduction ratio r. Secondly, we compare different MC-MLP nets with various r. The dimension reduction ratio r is introduced to ensure that a self-gating module's capacity and computational cost can be controlled. It is notable that there are two kinds of MLP blocks, i.e., token-mixing and channel-mixing, and the channel commonly has a larger dimension (e.g., 256) than the tokens (e.g., 25 or 30). In this case, we mainly tune the ratio used in the channel-mixing block. Table 2 shows the performance changes when setting $r = \{2, 8, 16, 32\}$. We can find that increasing r does not lead to monotonic performance degradation but significantly reduce the number of parameters (e.g., from 1.88M with $r = 2$ to 1.75M with $r = 32$). Considering the trade-off between the performance and the number of parameters, we set $r = 16$ for L/M/S-RD modules when integrating them into the channel-mixing block.

Impact of different structures and their combinations. Next, we test the impacts of different objectives by tuning their assigned factors $\{\alpha, \beta, \gamma\}$ in Eq. (12).

First, we assess the contribution of different structures by setting the studied loss factor as 1 and others as 0. Here, the hash code length $K = 64$ is adopted. As denoted in section 3.2, we build three types of structures within data for unsupervised learning, i.e., cluster ($\mathcal{L}_{cluster}$), neighborhood similarity ($\mathcal{L}_{sim}$) and inter/intra-class variation ($\mathcal{L}_{quad}$). Table 3 shows the performance changes with a single structure and their combinations. We observe that the cluster structural information can provide a higher performance than the other two, the combinations of two structures outperform that of a single structure, and adopting all three structures performs the best. This shows an evidence that different structures are complementary to each other and also proves the feasibility of the proposed multi-structure preservation mechanism.

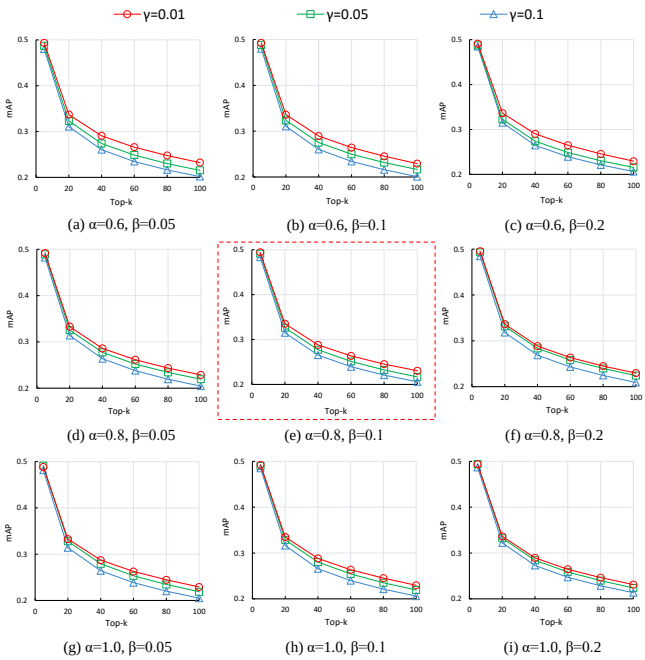


Figure 3: Performance changes varying $\{\alpha, \beta, \gamma\}$ for MCMSH. The hash code length is fixed as $K = 64$. The settings used for reporting the performance in the tables are as $\{\alpha = 0.8, \beta = 0.1, \gamma = 0.01\}$.

Then, we tune the values of $\{\alpha, \beta, \gamma\}$ with much finer granularities for a relative optimal setting using FCVID dataset. Figure 3 demonstrates the performance changes while varying $\{\alpha, \beta, \gamma\}$ with three settings, i.e., $\alpha \in \{1.0, 0.8, 0.6\}$, $\beta \in \{0.05, 0.1, 0.2\}$ and $\gamma \in \{0.01, 0.05, 0.1\}$. The selected value ranges are preliminarily narrowed down through empirically testing rough settings. As shown in the subfigure 3(e), the chosen settings of $\{\alpha = 0.8, \beta = 0.1, \gamma = 0.01\}$ provide a generally better performance across different mAP metrics.

4.4 Comparison with State-of-the-arts

We compare our MCMSH with several state-of-the-art unsupervised hashing methods, including both shallow learning method MFH [40], and deep learning methods DH [5], JTEA [23], SSTH [57], SSVH [41] and BTH [26]. Following [26], The image hashing method DH is extended to video hashing by using the same CNN features. Here, we test their results with hash code lengths of 16, 32 and 64 bits.

Results on FCVID. Figure 4(a-c) show the comparison between MCMSH and SOTAs on FCVID. Overall, our MCMSH, regardless of hash code length, achieves the best performance among all the competing methods. Compared to the more sophisticated transformer-based BTH, the absolute performance improvements of MCMSH are as high as 11.8%, 8.9%, 7.8%, 6.9%, 6.5% and 5.9% for mAP@k (k=5, 20, 40, 60, 80, 100) respectively with 16-bits. These considerable increases demonstrate strong proof of MCMSH's superiority in video hash code generation and structural information preservation. Particularly, as BTH adopts transformer as network model, it inherently possesses the ability of long-range dependency

Table 1: Performance (mAP@k) comparison with different MC-MLP variants on FCVID with 32 bits and 64 bits hash code lengths.

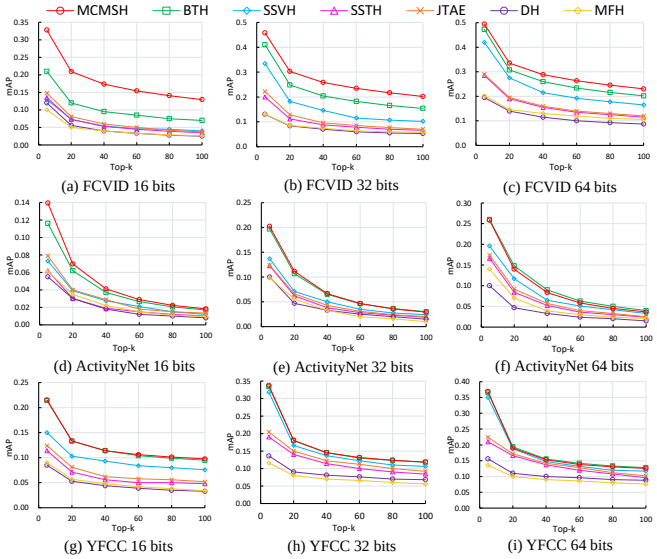
Model	32 bits					64 bits				
	k=20	k=40	k=60	k=80	k=100	k=20	k=40	k=60	k=80	k=100
MLP-Mixer	0.288	0.244	0.223	0.208	0.195	0.323	0.277	0.254	0.237	0.223
+L-RD	0.298	0.253	0.229	0.212	0.198	0.330	0.283	0.258	0.240	0.225
+M-RD	0.298	0.253	0.230	0.213	0.199	0.332	0.284	0.258	0.240	0.225
+S-RD	0.295	0.250	0.226	0.209	0.195	0.329	0.282	0.257	0.239	0.224
MC-MLP	**0.302**	**0.258**	**0.235**	**0.217**	**0.202**	**0.335**	**0.288**	**0.263**	**0.245**	**0.230**

Table 2: Performance (mAP@k) comparison of MCMSH with different dimension reduction ratio r on FCVID with hash code length 64 bits.

r	k=20	k=40	k=60	k=80	k=100	Param.
2	0.333	0.287	0.262	0.243	0.228	1.88M
8	**0.335**	**0.289**	**0.264**	**0.246**	**0.231**	1.78M
16	**0.335**	0.288	0.263	0.245	0.230	1.76M
32	0.331	0.286	0.262	0.244	0.229	**1.75M**

Table 3: Performance (mAP@k) comparison with a single data structure and their combination using FCVID dataset with 64 bits codes.

Loss	k=5	k=20	k=40	k=60	k=80	k=100
$L_{cluster}(\alpha=1)$	0.466	0.304	0.256	0.230	0.211	0.197
$L_{sim}(\beta=1)$	0.430	0.270	0.228	0.206	0.190	0.176
$L_{quad}(\gamma=1)$	0.441	0.262	0.211	0.185	0.167	0.154
$0.8L_{cluster}+0.1L_{sim}$	0.490	0.332	0.285	0.260	0.241	0.225
$0.8L_{cluster}+0.01L_{quad}$	0.486	0.328	0.282	0.257	0.239	0.224
$0.1L_{sim}+0.01L_{quad}$	0.464	0.290	0.239	0.213	0.195	0.181
MCMSH	**0.494**	**0.335**	**0.288**	**0.263**	**0.245**	**0.230**



(a) FCVID 16 bits (b) FCVID 32 bits (c) FCVID 64 bits
(d) ActivityNet 16 bits (e) ActivityNet 32 bits (f) ActivityNet 64 bits
(g) YFCC 16 bits (h) YFCC 32 bits (i) YFCC 64 bits

Figure 4: Performance (mAP@k) comparison with state-of-the-arts on FCVID (a-c), ActivityNet (d-f) and YFCC (g-i) datasets. The results of the competing methods are cited from [26].

modeling. As compared, our MLP-based MC-MLP model considers multi-granular dependencies, i.e., long-range, middle-range and short-range. Hence, it is not surprising that MCMSH obtains such significant performance improvements. This further demonstrates

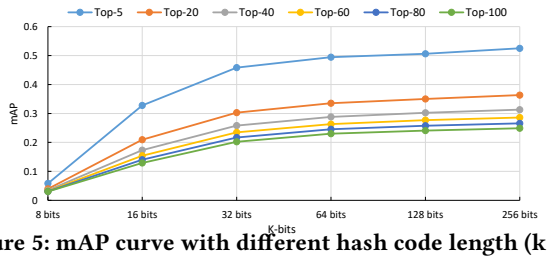


Figure 5: mAP curve with different hash code length (k bits) on FCVID dataset.

that the idea of contextualizing frame/video features with various dependencies is promising. In addition, we also draw the mAP curve with different hash code lengths for MCMSH in Figure 5 and observe the same trend as reported individual results.

Results on ActivityNet. Figure 4(d-f) present the performance comparison on ActivityNet. We observe similar performance trends with those on FCVID. That is, our MCMSH outperforms other methods with hash code lengths 16 and 32 bits by large margins (0.8%-4.0% for 16-bits, 0.5%-6.5% for 32-bits) and is competitive with code length 64 bits. We speculate that as videos in both FCVID and ActivityNet datasets contain rich human-object interactions, modeling multi-granular motion dependencies is especially essential for video content understanding. This further proves the feasibility of the hash code learning strategy of MCMSH.

Results on YFCC. Figure 4(g-i) list the results of different methods on YFCC. Except for BTH, MCMSH significantly outperforms all the other SOTAs under all evaluation configurations. While the results of MCMSH and BTH are very similar (actually MCMSH is slightly better than BTH with 16 and 32 code bits). The performance trend is slightly different from those on other two datasets. One possible reason might be that the discriminatory power of a smaller model (1.76M of MCMSH vs. 3.17M of BTH as shown in the table that follows) tends to saturate easily when processing large-scale datasets like YFCC (around 410k training samples). Considering the trade-off between effectiveness and efficiency, MCMSH is still superior to BTH.

Table 4: Cross-dataset mAP@20 gain when training on FCVID and test on YFCC with 64 bits. Results of competing methods are cited from [26].

Method	SSTH	SSVH	BTH	MCMSH
mAP@20	−6.3% ↓	−7.8% ↓	−5.7% ↓	−3.2% ↓

4.5 Cross-dataset Performance Comparison

We conduct cross-dataset retrieval to investigate the generalization of MCMSH by following the standard protocol, i.e., training on

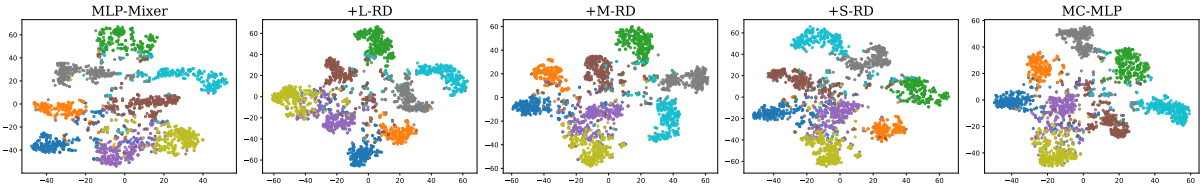


Figure 6: Visualizing feature distributions with different MC-MLP variants and the backbone MLP-Mixer via t-SNE.

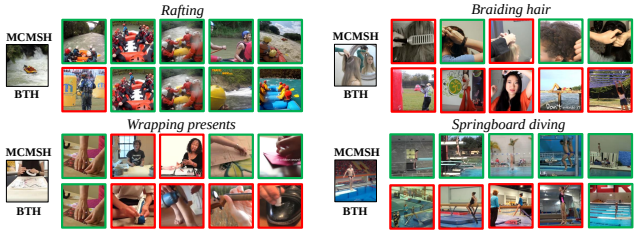


Figure 7: Top-5 retrieved results of BTH and MCMSH on ActivityNet dataset. The selected four action categories contain various human-object interactions and thus require strong spatio-temporal modeling. The video in a green square is correctly retrieved, while the video in a red square is an error.

a dataset and test on another dataset. In the implementation, we report the cross-dataset performance when training on FCVID and test on YFCC. As shown in Table 4, all methods suffer a considerable performance drop. This is probably because there is an obvious discrepancy between the training dataset (around 45k videos) and the test dataset (around 100k videos). In other words, when the training dataset is much smaller than the test dataset, the generalization of the data-driven hash methods is limited. In particular, the performance drop of MCMSH is less than the other methods, indicating a better generalization across datasets.

Table 5: Comparison of parameters, FLOPs and average encoding time between BTH and MCMSH. The average encoding time is computed in the same platform.

Method	Param.	FLOPs	Average Encoding Time
BTH	3.17M	0.05G	0.53ms
MCMSH	1.76M	0.05G	0.47ms

4.6 Model Complexity Comparison

Model complexity is another important measurement for method evaluation. We show the model complexity comparison between the most competitive method BTH and our MCMSH in Table 5. The number of parameters, computational FLOPs, and average encoding time are reported with settings of 64 bits and 25 frames. As observed, MCMSH has 1.76M parameters, which is only 55% of BTH (3.17M). The average encoding time is computed as the amount of time spent on hash code generation for a single instance. It can be found that our MCMSH requires less encoding time (0.47ms) than BTH (0.57ms). In terms of FLOPs, they have a similar number, i.e., 0.05G.

4.7 Qualitative Results

To contrast the features of different MC-MLP variants and the backbone MLP-Mixer, we apply t-SNE [45] to project the features (relaxed hash codes with 64 bits) for visualization. Figure 6 shows t-SNE visualization plots on videos randomly sampled on FCVID dataset. It is obvious that the features of different semantic classes learned from MC-MLP with a single self-gating module (e.g., L-RD, M-RD, or S-RD) are better separated compared to those from the backbone MLP-Mixer. The complete version of MC-MLP, i.e., w/ L-RD, M-RD, and S-RD, further increases the distinguishability of hash codes.

We also visually compare the retrieval results of the most competitive BTH and our MCMSH by showing their top-5 video samples retrieved from ActivityNet dataset. The results are based on the setting of hash code length 64 bits. The selected four video classes, e.g., "*Rafting*", "*Braiding hair*", "*Wrapping presents*" and "*Springboard diving*", contain rich human-object interactions and thus require strong spatio-temporal modeling. As shown in Figure 7, our MCMSH can generally return more accurate results than BTH. Particularly, for the class "*Braiding hair*" which needs to model the hand-hair interaction, our MCMSH successfully retrieves three correct videos while BTH fails to find any correct sample. Moreover, the other two incorrect samples ("*Brushing hair*") retrieved by MCMSH have higher semantic relevance to the query action "*Braiding hair*", compared to the samples retrieved by BTH. The above examples indicate that MCMSH performs better than the transformer based BTH for modeling temporal correlations within the video.

5 CONCLUSION

We have presented an unsupervised video hashing method, MCMSH, which explores multiple axial contexts for discriminative video representation and various structural information for unsupervised learning simultaneously. MCMSH builds a MLP based neural network model MC-MLP for hash code generation. MC-MLP enhances the backbone with three self-gating modules, including long-range, middle-range, and short-range dependency modules (i.e., L/M/S-RD). The three modules L/M/S-RD focus on different kinds of axial contexts to model multi-granular spatio-temporal interactions. To facilitate unsupervised learning, we investigate three kinds of structural information within data, including clusters, neighborhood similarity, and intra/inter-class variation. The three data structures are crucial for video retrieval. In the experiment, we thoroughly verify the impacts of the designed self-gating modules and the constructed structures. Experimental results conducted on three commonly used benchmark datasets demonstrate the effectiveness and efficiency of MCMSH compared with state-of-the-arts.

ACKNOWLEDGMENTS

The work was supported by the National Natural Science Foundation of China under Grant No. 62101524.

REFERENCES

[1] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

[2] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*. 961–970.

[3] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang. 2017. Beyond Triplet Loss: A Deep Quadruplet Network for Person Re-Identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[4] Yajiao Dong and Jianguo Li. 2018. Video retrieval based on deep convolutional neural network. In *Proceedings of the 3rd International Conference on Multimedia Systems and Signal Processing*. 12–16.

[5] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. 2015. Deep Hashing for Compact Binary Codes Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] Chaowei Fang, Dingwen Zhang, Liang Wang, Yulun Zhang, Lechao Cheng, and Junwei Han. 2022. Cross-Modality High-Frequency Transformer for MR Image Super-Resolution. *arXiv preprint arXiv:2203.15314* (2022).

[7] Praveen Gauravaram and Lars R Knudsen. 2009. On randomizing hash functions to strengthen the security of digital signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 88–105.

[8] Ivan Giangreco, Ihab Al Kabary, and Heiko Schuldt. 2014. Adam-a database and information retrieval system for big multimedia collections. In *2014 IEEE International Congress on Big Data*. IEEE, 406–413.

[9] Yun Gu, Chao Ma, and Jie Yang. 2016. Supervised recurrent hashing for large scale video retrieval. In *Proceedings of the 24th ACM international conference on Multimedia*. 272–276.

[10] Shai Halevi and Hugo Krawczyk. 2006. Strengthening digital signatures via randomized hashing. In *Annual International Cryptology Conference*. Springer, 41–59.

[11] Ning Han, Jingjing Chen, Guangyi Xiao, Hao Zhang, Yawen Zeng, and Hao Chen. 2021. Fine-grained cross-modal alignment network for text-video retrieval. In *Proceedings of the 29th ACM International Conference on Multimedia*. 3826–3834.

[12] Yanbin Hao, Tingting Mu, John Y Goulermas, Jianguo Jiang, Richang Hong, and Meng Wang. 2017. Unsupervised t-distributed video hashing and its deep hashing extension. *IEEE Transactions on Image Processing* 26, 11 (2017), 5531–5544.

[13] Yanbin Hao, Tingting Mu, Richang Hong, Meng Wang, Ning An, and John Y Goulermas. 2016. Stochastic multiview hashing for large-scale near-duplicate video retrieval. *IEEE Transactions on Multimedia* 19, 1 (2016), 1–14.

[14] Yanbin Hao, Chong-Wah Ngo, and Benoit Huet. 2019. Neighbourhood structure preserving cross-modal embedding for video hyperlinking. *IEEE Transactions on Multimedia* 22, 1 (2019), 188–200.

[15] Yanbin Hao, Shuo Wang, Pei Cao, Xinjian Gao, Tong Xu, Jinmeng Wu, and Xiangnan He. 2022. Attention in Attention: Modeling Context Correlation for Efficient Video Classification. *IEEE Transactions on Circuits and Systems for Video Technology* (2022).

[16] Yanbin Hao, Hao Zhang, Chong-Wah Ngo, and Xiangnan He. 2022. Group Contextualization for Video Recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 928–938.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

[18] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[19] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. 2022. Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2022).

[20] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[21] Yu-Gang Jiang, Zuxuan Wu, Jun Wang, Xiangyang Xue, and Shih-Fu Chang. 2017. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE transactions on pattern analysis and machine intelligence* 40, 2 (2017), 352–364.

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

[23] Chao Li, Yang Yang, Jiewei Cao, and Zi Huang. 2017. Jointly modeling static visual appearance and temporal pattern for unsupervised video hashing. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 9–17.

[24] Shuyan Li, Zhixiang Chen, Xiu Li, Jiwen Lu, and Jie Zhou. 2019. Unsupervised variational video hashing with 1d-cnn-lstm networks. *IEEE Transactions on Multimedia* 22, 6 (2019), 1542–1554.

[25] Shuyan Li, Zhixiang Chen, Jiwen Lu, Xiu Li, and Jie Zhou. 2019. Neighborhood preserving hashing for scalable video retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 8212–8221.

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

[27] Shuyan Li, Xiu Lia, Jiwen Lu, and Jie Zhou. 2021. Structure-adaptive Neighborhood Preserving Hashing for Scalable Video Search. *IEEE Transactions on Circuits and Systems for Video Technology* (2021).

[28] Yunqiang Li and Jan van Gemert. 2020. Deep unsupervised image hashing by maximizing bit entropy. *arXiv preprint arXiv:2012.12334* (2020).

[29] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. 2021. As-mlp: An axial shifted mlp architecture for vision. *arXiv preprint arXiv:2107.08391* (2021).

[30] Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to mlps. *Advances in Neural Information Processing Systems* 34 (2021), 9204–9215.

[31] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2011. Hashing with graphs. In *Icml*.

[32] Xiang Long, Chuang Gan, Gerard De Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. 2018. Attention clusters: Purely attention based local feature integration for video classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7834–7843.

[33] David G Lowe. 1999. Object recognition from local scale-invariant features. In *Proceedings of the seventh IEEE international conference on computer vision*, Vol. 2. Ieee, 1150–1157.

[34] Xiao Luo, Daqing Wu, Zeyu Ma, Chong Chen, Minghua Deng, Jinwen Ma, Zhongming Jin, Jianqiang Huang, and Xian-Sheng Hua. 2020. Cimon: Towards high-quality hash codes. *arXiv preprint arXiv:2010.07804* (2020).

[35] Xiushan Nie, Xin Zhou, Yang Shi, Jiande Sun, and Yilong Yin. 2021. Classification-enhancement deep hashing for large-scale video retrieval. *Applied Soft Computing* 109 (2021), 107467.

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[37] Ling Shen, Richang Hong, and Yanbin Hao. 2020. Advance on large scale near-duplicate video retrieval. *Frontiers of Computer Science* 14, 5 (2020), 1–24.

[38] Ling Shen, Richang Hong, Haoran Zhang, Xinmei Tian, and Meng Wang. 2019. Video retrieval with similarity-preserving deep temporal hashing. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 4 (2019), 1–16.

[39] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[40] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. 2011. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *Proceedings of the 19th ACM international conference on Multimedia*. 423–432.

[41] Jingkuan Song, Hanwang Zhang, Xiangpeng Li, Lianli Gao, Meng Wang, and Richang Hong. 2018. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Transactions on Image Processing* 27, 7 (2018), 3210–3221.

[42] Yi Tan, Yanbin Hao, Xiangnan He, Yinwei Wei, and Xun Yang. 2021. Selective dependency aggregation for action classification. In *Proceedings of the 29th ACM International Conference on Multimedia*. 592–601.

[43] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2015. The new data and new challenges in multimedia research. *arXiv preprint arXiv:1503.01817* 1, 8 (2015).

[44] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in Neural Information Processing Systems* 34 (2021).

[45] Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, 11 (2008).

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[47] Jun Wang, Wei Liu, Sanjiv Kumar, and Shih-Fu Chang. 2015. Learning to hash for indexing big data—A survey. *Proc. IEEE* 104, 1 (2015), 34–57.

[48] Shuo Wang, Dan Guo, Xin Xu, Li Zhuo, and Meng Wang. 2019. Cross-modality retrieval by joint correlation learning. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 15, 2s (2019), 1–16.

[49] Yingxin Wang, Xiushan Nie, Yang Shi, Xin Zhou, and Yilong Yin. 2019. Attention-based video hashing for large-scale video retrieval. *IEEE Transactions on Cognitive and Developmental Systems* 13, 3 (2019), 491–502.

[50] Yair Weiss, Antonio Torralba, and Rob Fergus. 2008. Spectral hashing. *Advances in neural information processing systems* 21 (2008).

[51] Gengshen Wu, Jungong Han, Yuchen Guo, Li Liu, Guiguang Ding, Qiang Ni, and Ling Shao. 2018. Unsupervised deep video hashing via balanced code for large-scale video retrieval. *IEEE Transactions on Image Processing* 28, 4 (2018), 1993–2007.

[52] Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo. 2007. Practical elimination of near-duplicates from web video search. In *Proceedings of the 15th ACM international conference on Multimedia*. 218–227.

[53] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. 2022. S2-mlp: Spatial-shift mlp architecture for vision. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 297–306.

[54] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. 2020. Central similarity quantization for efficient image and video retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3083–3092.

[55] Dell Zhang, Jun Wang, Deng Cai, and Jinsong Lu. 2010. Self-taught hashing for fast similarity search. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. 18–25.

[56] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. 2021. Token shift transformer for video classification. In *Proceedings of the 29th ACM International Conference on Multimedia*. 917–925.

[57] Hanwang Zhang, Meng Wang, Richang Hong, and Tat-Seng Chua. 2016. Play and rewind: Optimizing binary representations of videos by self-supervised temporal hashing. In *Proceedings of the 24th ACM international conference on Multimedia*. 781–790.

[58] Guoying Zhao and Matti Pietikainen. 2007. Dynamic texture recognition using local binary patterns with an application to facial expressions. *IEEE transactions on pattern analysis and machine intelligence* 29, 6 (2007), 915–928.

[59] Bin Zhu and Chong-Wah Ngo. 2020. CookGAN: Causality based text-to-image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5519–5527.

[60] Bin Zhu, Chong-Wah Ngo, Jingjing Chen, and Yanbin Hao. 2019. R2gan: Cross-modal recipe retrieval with generative adversarial network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11477–11486.