Motion-Aware Graph Reasoning Hashing for Self-supervised Video Retrieval

Ziyun Zeng[1, 3]
zengzy21@mails.tsinghua.edu.cn
Jinpeng Wang[1, 3]
wjp20@mails.tsinghua.edu.cn
Bin Chen[2, 3, ✉]
chenbin2021@hit.edu.cn
Yuting Wang[1, 3]
huangmozhi9527@gmail.com
Shu-Tao Xia[1, 3]
xiast@sz.tsinghua.edu.cn

[1] Tsinghua Shenzhen International Graduate School, Tsinghua University, China
[2] Harbin Institute of Technology, Shenzhen, China
[3] Research Center of Artificial Intelligence, Peng Cheng Laboratory, China

Abstract

Unsupervised video hashing aims to learn a nonlinear hashing function to map videos into a similarity-preserving hamming space without label supervision. Different from static images, the motion information within videos is crucial for content understanding. However, most existing works merely extract general features from sparsely sampled frames and do not explore motion information adequately. On the other hand, directly extracting clip-wise motion features is not practical in inference because of the heavy computation overhead. In this paper, we propose ***M**otion-**A**ware **G**raph **R**easoning **H**ashing* (MAGRH), an end-to-end framework that utilizes the motion information explicitly while keeping inference efficiency. Specifically, we design a dual-branch architecture consisting of a main branch and an auxiliary branch. During training, the main (auxiliary) branch receives frame-wise (clip-wise) inputs and produces general (motion) hash codes via delicately designed graph reasoning modules and hash layers. On top of the two branches, we develop a combination of intra- and inter-branch contrastive objectives to simultaneously learn branch-specific hashing functions as well as transfer motion knowledge from the auxiliary branch to the main branch. In inference, the hash codes are solely produced by the main branch, which only requires frame-wise inputs. Benefiting from motion guidance, our MAGRH yields superior performance on two public benchmarks, *i.e.*, FCVID and ActivityNet, even with a small frame rate.

1 Introduction

Video retrieval aims to find videos relevant to a given query from a large-scale database. The rapid growth of video scale from the Internet and social media raises the concern of retrieval and storage efficiency, and hashing has become a reliable solution. Hashing aims to project

✉ Bin Chen is the corresponding author.

high-dimensional real-valued data into compact binary codes in a similarity-preserving way. Then the binary codes can be stored efficiently and support fast bitwise Hamming distance computation. In recent years, a bunch of learnable image hashing methods [4, 5, 6, 24, 33] which leverage label supervision or explore similarity structure among data have achieved promising performance. On the contrary, video hashing [13, 14, 20, 41] still yields poor performance. The reason is two-fold: **(i)** Lack of label supervision. Since annotating videos is more labor-intensive, it is hard to construct large-scale labeled datasets, *e.g.*, ImageNet [25]. **(ii)** Modeling the temporary dependency within frames is challenging. Therefore, efficient and effective ***U**nsupervised **V**ideo **H**ashing* (UVH) has become a valuable research topic.

Most state-of-the-art UVH methods take temporary dependency into consideration. They mainly adopt RNN [41], LSTM [21, 27], and Transformer [22] to handle sparsely sampled frames. Some latest works [8, 10] have shed light on *Graph Reasoning* (GR) in the field of video content understanding. They mainly focus on learning the semantic relation between objects within a single frame, while modeling the temporary dependency via GR is seldom considered. On the other hand, *modeling temporary dependency can be regarded as an implicit way to exploit the motion information within frames.* However, without explicit motion guidance, a model may be trapped on objects or scenes, as these semantic components are easier to learn in sparsely sampled frames. For example, when retrieving "Car Racing", other car-dominated videos such as "Parking Car" might be returned, if "Racing" is not emphasized. Nevertheless, directly extracting clip-wise motion features via specific models is not practical in inference due to largely increased computation overhead.

Table 1 lists the configuration of two frame-wise models in video hashing [21, 22] and one clip-wise model in action recognition [29]. The GFLOPs of the clip-wise model, *i.e.*, R(2+1)D-34, is ten times more than that of the other two models. Besides, clip-wise models are not as flexible as frame-wise models because they usually require a fixed number of input frames. Hence it is a challenge to exploit motion information as well as keep inference efficiency.

Table 1: The configuration of three feature extractors.

Model	Input Type	Input Size	Params	GFLOPs
VGG-16	Frame	$3 \times 224 \times 224$	138M	15.47
ResNet-50	Frame	$3 \times 224 \times 224$	25.6M	4.11
R(2+1)D-34	Clip	$3 \times 32 \times 112 \times 112$	63.7M	152.76

In this paper, we propose a self-supervised video hashing method, namely ***M**otion-**A**ware **G**raph **R**easoning **H**ashing* (MAGRH). Specifically, we design a dual-branch architecture consisting of: **(i)** A general branch to encode frame-wise general features. **(ii)** An auxiliary branch to encode clip-wise motion features. In each branch, we use two cascaded ***G**raph **R**easoning **M**odules* (GRMs) to model the temporary dependency between frames. We further develop a combination of intra- and inter-branch contrastive objectives. The intra-branch objective aims to improve branch-specific hash codes. And the inter-branch objective aims to transfer motion knowledge from the auxiliary branch to the main branch. As a result, the main branch learns a "motion-aware" hashing function that can dig motion information adequately in sparsely sampled frames. In inference, the auxiliary branch is removed, and the hash codes are solely produced by the main branch with frame-wise inputs. Therefore, the computation complexity is comparable to conventional video hashing methods.

To summarize, we make the following contributions:

- We propose a novel self-supervised video hashing method, *i.e.*, ***M**otion-**A**ware **G**raph **R**easoning **H**ashing* (MAGRH), in which we explicitly model the motion information and take an early step to adopt graph reasoning to model the temporary dependency.
- We design a dual-branch architecture to produce general and motion hash codes sep-

arately, as well as a combination of intra- and inter-branch contrastive objectives to simultaneously improve branch-specific hash codes and force the general hash codes to preserve more motion information within frame-wise inputs.

- Extensive experiments on two public benchmarks, FCVID and ActivityNet, show the superiority of the proposed MAGRH. Benefiting from motion guidance, our MAGRH surpasses state-of-the-art methods by a large margin with fewer inference frames.

2 Related Works

Video Hashing. In the early phase, image hashing has made great breakthroughs in image retrieval consistently. For example, DH [7] learns hash codes by seeking multiple hierarchical non-linear transformations. Video hashing can be a direct extension of image hashing by handling each frame independently and aggregating frame features as the global video representation [11, 26, 36]. But such a migration yields inferior performance because of the ignorance of video-specific structural information, *e.g.*, temporal dependency. VHDT [37] is the first work to explore video temporal information and achieves considerable performance gain over previous methods. NPH [21] further explores neighborhood information between videos to preserve the similarity structure. JTAE [19] jointly learns an appearance encoder and a temporal encoder by reconstructing the visual and temporal pattern separately.

Benefiting from the powerful representation ability of deep neural networks (DNNs), many DNN-based video hashing approaches emerge in the past few years. Among them, most state-of-the-art approaches typically sample frames within a video sparsely and leverage a sequence model to explore the temporal dependency, *e.g.*, SSTH [41] and SSVH [27]. Inspired by the recent success of Transformer [30], the newly proposed BTH [22] incorporates the hash layer into a bidirectional Transformer and surpasses conventional RNN- and LSTM-based approaches. However, none of the existing video hashing methods utilize the motion information explicitly. Besides, some of them rely on sophisticated multi-stage training, *e.g*. NPH and BTH, which is unfriendly for real-world applications.

On the other hand, some latest works have shown the great potential of ***G**raph **R**easoning* (GR) in video content understanding [8, 10]. But they merely adopt GR in a single frame to learn the object semantic relationship instead of exploiting the whole frame sequence. Compared to previous works, our MAGRH is a simple end-to-end framework that leverages motion information explicitly and efficiently, and we take an early step to adopt graph reasoning in video hashing.

Video Representation Learning. Beyond video hashing, there are a bunch of tries to learn better video representations in the past few years. For instance, TSM [23] captures temporal information by shifting part of the channels along the temporal dimension, which is parameter-free. However, the shift operation before each convolution still increases memory footprint, while our MARGH does not involve extra operations in inference. [32] propose a temporal consistency regularization (TCR), which maintains representation between the full-resolution video and its down-sampled version. Our MAGRH differs from TCR because TCR is a unidirectional regularization while our MARGH enables a bidirectional interaction between the motion and the general features, *i.e.*, the motion feature can also interact with general features to enhance themselves, leading to better guidance. It is favorable for videos that are not motion-centric, as we carefully avoid motion overwhelming other clues. Besides, the TCR requires label supervision, while our MARGH is fully self-supervised, which is more practical to be applied in the real world.

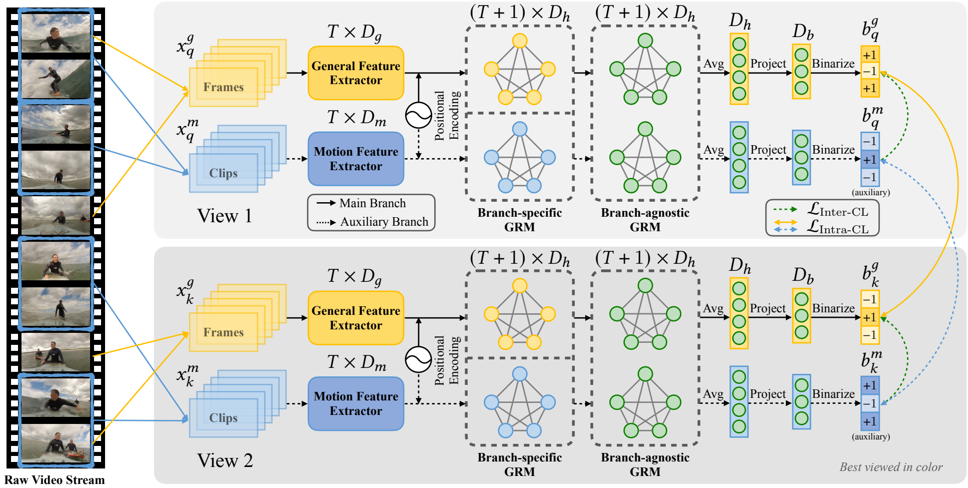


Figure 1: The framework of MAGRH. We first sample two sets of frames and two sets of clips to construct two correlated views. Each view consists of two branches: **(i)** A main branch to encode frame-wise inputs and produce general hash codes. **(ii)** An auxiliary branch to encode clip-wise inputs and produce motion hash codes. We develop two contrastive objectives to learn better representations within each branch and transfer motion knowledge from the auxiliary branch to the main branch. In inference, the auxiliary branch is removed.

3 Proposed Method

3.1 Problem Formulation and Model Overview

Given a training set $\mathcal{D}$ consisting of N_D videos that belong to N_C categories, *i.e.*, $\mathcal{D} = \{v_i, y_i\}_{i=1}^{N}$, where v_i is the i-th video and $y_i \in \{1, \cdots, N_c\}$ is the corresponding label of the i-th video. We aim to learn a neural hashing function $\mathcal{H} : v_i \mapsto \{-1, 1\}^K$ such that v_i can be encoded into a K bits binary code b_i for efficient retrieval. The learned hashing function $\mathcal{H}$ should preserve the similarity structure of $\mathcal{D}$, *i.e.*, the distance between b_i and b_j should be small if v_i and v_j share the same label, otherwise, it should be large.

To achieve this goal, we propose a self-supervised model named ***M**otion-**A**ware **G**raph **R**easoning **H**ashing* (MAGRH), which can be trained in an end-to-end manner. The framework of MAGRH is illustrated in Figure 1. For each video, we first sample two sets of frames (x_q^g and x_k^g) and two sets of clips (x_q^m and x_k^m) to construct two correlated views $\{x_q^g, x_q^m\}$ and $\{x_k^g, x_k^m\}$. Within each view, we design a dual-branch architecture consisting of: **(i)** A main branch to encode frame-wise inputs x_q^g (x_k^g) and produce general hash codes b_q^g (b_k^g). **(ii)** An auxiliary branch to encode clip-wise inputs x_q^m (x_k^m) and produce motion hash codes b_q^m (b_k^m). On top of the two branches, we develop a combination of two contrastive objectives, *i.e.*, $\mathcal{L}_{\text{Intra-CL}}$ and $\mathcal{L}_{\text{Inter-CL}}$. $\mathcal{L}_{\text{Intra-CL}}$ enables branch-specific hash learning while $\mathcal{L}_{\text{Inter-CL}}$ transfers motion knowledge from the auxiliary branch to the main branch. In inference, the auxiliary branch is removed. We only encode frame-wise inputs via the main branch.

3.2 Training Pipeline

As illustrated in Figure 1, the proposed MAGRH is a contrastive learning-based framework consisting of two correlated views. Since the two views share the same pipeline, we omit the subscripts q, k and describe the pipeline within a single view for simplicity.

3.2.1 General and Motion Feature Extraction

As for the main branch, given a video v composed of T^v frames, we first follow [22, 27] to sample T frames randomly. Then, we send these frames into a general feature extractor, *e.g.*, VGG and ResNet, and obtain the frame-wise general features $x^g \in \mathbb{R}^{T \times D_g}$, where D_g denotes the dimension of general features.

Similarly, for the auxiliary branch, we divide the video v into T^v/T_c clips, where T_c is the number of frames in one clip. In our implementation, we fix $T_c = 32$. Next, we randomly sample T clips, sending them to a motion feature extractor, *e.g.*, R(2+1)D-34, and obtain the clip-wise motion features $x^m \in \mathbb{R}^{T \times D_m}$, where D_m denotes the dimension of motion features.

3.2.2 Sequence-level Graph Reasoning

G*raph* ***R****easoning* (GR) has made great success in video analysis recently [8, 10]. Existing works mainly focus on mining object relations in a single frame. They treat objects as graph nodes and learn the representations via graph networks such as GCN [18] and GAT [31]. In our work, we construct a graph where each node represents a single frame (clip), and conduct sequence-level graph reasoning via two ***G****raph* ***R****easoning* ***M****odules* (GRMs).

To be specific, we first project x^g, x^m into the same dimension D_h via two linear layers $\phi^g(\cdot)$ and $\phi^m(\cdot)$. Then a learnable positional embedding $z_p \in \mathbb{R}^{T \times D_h}$ is added to each projected feature. Besides, we add a learnable branch-specific embedding $z_m \in \mathbb{R}^{D_h}$ and concatenate an extra "[AGG]" token $\varphi_{agg} = \text{maxpool}(\phi(x))$ at the beginning of the sequence.

Formally, the final input sequences are computed as:

$$\hat{x}^* = [\varphi^*_{agg};\ \phi^*(x^*) + z_p + z^*_m] \in \mathbb{R}^{(T+1) \times D_h},\ * \in \{g, m\} \tag{1}$$

where $[\cdot;\cdot]$ denotes the concatenation operator.

Once $\hat{x}^g, \hat{x}^m$ are prepared, we conduct graph reasoning with two consecutive GRMs, *i.e.*, *branch-specific* GRM and *branch-agnostic* GRM. As illustrated in Figure 1, the branch-specific GRM is composed of two independent GCNs corresponding to the main and auxiliary branches. While the branch-agnostic GRM only contains a shared GCN between the two branches. In each GCN, we first construct a graph by computing an adjacent matrix $A \in \mathbb{R}^{(T+1) \times (T+1)}$ followed by row-wise normalization (the superscripts g, m are omitted for simplicity):

$$A = g_1(\hat{x})^T g_2(\hat{x}), \quad A_{i,j} \to \frac{A^2_{i,j}}{\sum_{t=1}^{T+1} A^2_{i,t}} \tag{2}$$

where $g_1(\cdot), g_2(\cdot)$ denote two projections, $A_{i,j}$ denotes the element of A in the i-th row and j-th column. Next, L graph convolution layers are used to exploit certain (*i.e.*, general and motion) information within frames (clips). The l-th layer is implemented as [34, 35]:

$$\hat{x}_l = \text{ReLU}(\text{LN}(A\hat{x}_{l-1}W_l)), l \in \{1, \cdots, L\} \tag{3}$$

where $\mathrm{LN}(\cdot)$ denotes layer normalization [1], x_l denotes the output of the l-th layer, and $W_l \in \mathbb{R}^{D_h \times D_h}$ denotes the weight matrix of the l-th layer. $\hat{x}_0$ equals to $\hat{x}$ in Eqn.(1) and the output of the L-th layer, *i.e.*, x_L, is taken as the final representation.

This cascaded design makes training more efficient because we "decouple" the learning process of different clues, thus preventing information loss. For the main branch, the sub-important clues such as object and scene are easier to be preserved in the branch-specific GRM, while the model pays more attention to motion in the branch-agnostic GRM.

3.2.3 Hash Layers

After obtaining the output of branch-agnostic GRM, *i.e.*, $\hat{x}_L^g, \hat{x}_L^m$, we apply an average pooling along the temporal channel:

$$\bar{x}^* = \frac{1}{T+1} \sum_{i=1}^{T+1} \hat{x}_{L,i}^* \in \mathbb{R}^{D_h}, \; * \in \{g, m\} \tag{4}$$

where $\hat{x}_{L,i} \in \mathbb{R}^{D_h}$ denotes the i-th temporal channel of $\hat{x}_L$. Then we project $\bar{x}^g, \bar{x}^m$ into D_b-dimensional real-valued vectors h^g, h^m via a linear layer, where D_b equals to the hash code length. Finally, h^g, h^m are binarized through an $sgn(\cdot)$ function.

The overall process of hash layers is as follows:

$$\begin{aligned} h^* &= \Phi^*(\bar{x}^*) \in \mathbb{R}^{D_b}, \; * \in \{g, m\} \\ b^* &= \mathrm{sgn}(h^*) \in \{-1, 1\}^{D_b}, \; * \in \{g, m\} \end{aligned} \tag{5}$$

where $\mathrm{sgn}(x) = 1$ if $x > 0$, otherwise $\mathrm{sgn}(x) = -1$, $\Phi^g(\cdot), \Phi^m(\cdot)$ are two linear layers.

3.3 Learning Objectives

Inspired by the recent success of contrastive learning [3, 12, 15], we first propose a hash code-based contrastive objective, namely $\mathcal{L}_{\text{Intra-CL}}$, to learn the intra-branch hashing function for the two branches simultaneously:

$$\mathcal{L}_{\mathrm{CL}}^* = -\sum_{b_q^*, b_k^* \in \mathcal{B}} \log \frac{\exp(s(b_q^*, b_k^*)/\tau)}{\exp(s(b_q^*, b_k^*)/\tau) + \sum_{b_{k-}^* \in \mathcal{B} \setminus \{b_q^*, b_k^*\}} \exp(s(b_q^*, b_{k-}^*)/\tau)} \tag{6}$$

$$\mathcal{L}_{\text{Intra-CL}} = \frac{1}{2}\left(\mathcal{L}_{\mathrm{CL}}^g + \mathcal{L}_{\mathrm{CL}}^m\right) \tag{7}$$

where $\mathcal{B}$ denotes a mini-batch, b_q^*, b_k^* denote the hash codes corresponding to two correlated views, in which the inputs x_q^*, x_k^* are sampled from two non-overlapping frame (clip) sets, b_{k-}^* denotes the negative sample for b_q^* in $\mathcal{B}$, τ denotes the temperature parameter, and $s(\cdot, \cdot)$ denotes the cosine similarity function.

Beyond $\mathcal{L}_{\text{Intra-CL}}$, we hope the learned hashing function of the main branch focuses more on the crucial motion information within sparsely sampled frames. To achieve this goal, we take the insights from [28] and derive an inter-branch contrastive objective, namely $\mathcal{L}_{\text{Inter-CL}}$, to transfer motion knowledge from the auxiliary branch to the main branch explicitly. Due to the space limitation, we only present the formulation of $\mathcal{L}_{\text{Inter-CL}}$ here, and leave the full derivation in the appendix:

$$\hat{\mathcal{L}}^{*}_{\text{CL}} = -\sum_{b^g_*, b^m_* \in \mathcal{B}} \log \frac{\exp(s(b^g_*, b^m_*)/\hat{\tau})}{\exp(s(b^g_*, b^m_*)/\hat{\tau}) + \sum_{b^m_{*-} \in \mathcal{B} \setminus \{b^g_*, b^m_*\}} \exp(s(b^g_*, b^m_{*-})/\hat{\tau})} \tag{8}$$

$$\mathcal{L}_{\text{Inter-CL}} = \frac{1}{2}\left(\hat{\mathcal{L}}^{q}_{\text{CL}} + \hat{\mathcal{L}}^{k}_{\text{CL}}\right) \tag{9}$$

where $\hat{\tau}$ denotes the temperature parameter. Note that we only adopt an *intra-view* contrastive between general and motion hash codes.

Finally, we add an L_2 regularization term $\mathcal{L}_{\text{reg}}$ to reduce the quantization error between the real-valued vectors h and the hash codes b:

$$\mathcal{L}_{\text{reg}} = \sum_{i \in \{q,k\}} \sum_{j \in \{g,m\}} \|b^j_i - h^j_i\|^2_2 \tag{10}$$

The overall learning objective is the sum of $\mathcal{L}_{\text{Intra-CL}}$, $\mathcal{L}_{\text{Inter-CL}}$, and $\mathcal{L}_{\text{reg}}$ *i.e.*,

$$\mathcal{L}_{\text{MAGRH}} = \mathcal{L}_{\text{Intra-CL}} + \gamma \mathcal{L}_{\text{Inter-CL}} + \mathcal{L}_{\text{reg}} \tag{11}$$

where γ denotes the weight parameter for $\mathcal{L}_{\text{Inter-CL}}$.

We apply a Straight-Through Estimator [38] to resolve the intractable issue caused by binarization, thus enabling end-to-end training by back-propagation.

3.4 Inference Setting

In inference, the similarity between a given query and the database items can be calculated via bitwise Hamming distance efficiently. Notably, the hash codes are solely produced by the main branch, which is computationally efficient as only frame-wise inputs are required.

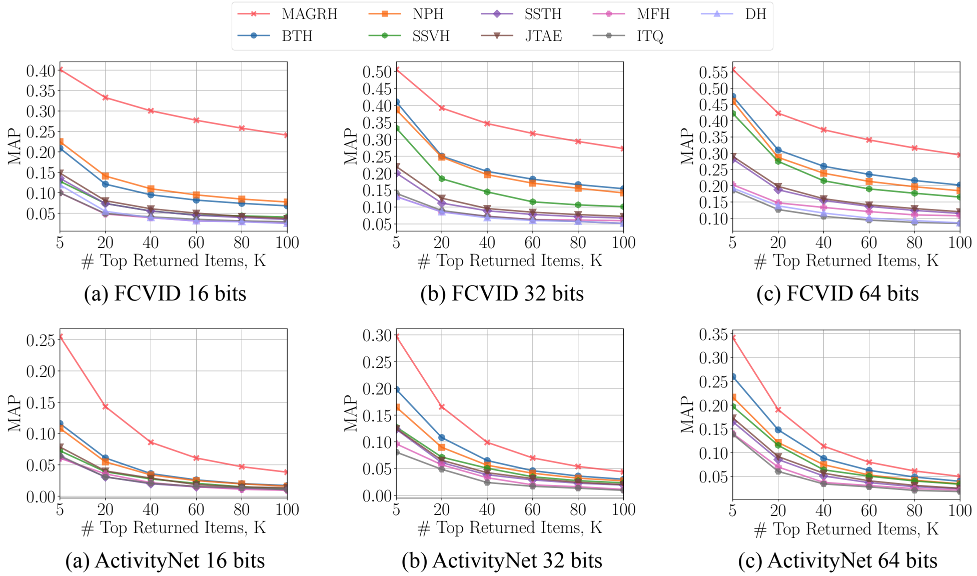
4 Experiments

4.1 Experimental Setup

Our proposed MAGRH is evaluated on two widely-used public benchmarks: FCVID [16] and ActivityNet [2]. **FCVID** consists of 91,223 videos belonging to 239 categories, these categories cover a wide range of topics, *e.g.*, scenes, objects, and activities. Following [22, 41], we pick 91,185 videos. The training set has 45,585 videos, and the rest 45,600 videos form the test set as well as the retrieval database. **ActivityNet** comprises various human activities, which are annotated into 200 categories. We follow [22] and select 9,722 videos as the training set. Since the test set is not publicly available, we randomly pick 1,000 and 3,760 videos in the validation set as the queries and the retrieval database respectively.

We follow previous video hashing works [21, 22, 27, 41] and adopt the **M**ean **A**verage **P**recision at top-K retrieved results (MAP@K) to evaluate the performance. In addition to MAP@K, we further draw the **P**recision-**R**ecall (PR) curve as an additional metric. The retrieved results are sorted by their Hamming ranking.

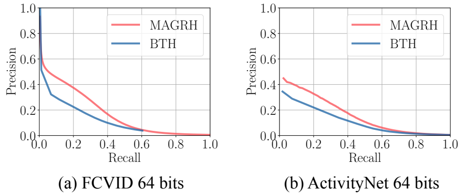
We use Adam [17] as the optimizer, the initial learning rate is 5×10^{-5}, the batch size is 64 and the training epoch is 200. Other hyper-parameters are listed as follows: **(i)** The dimension of hidden states, $D_h = 768$. **(ii)** The temperature parameter in Eqn.(6) and (8), $\tau = \hat{\tau} = 0.1$. **(iii)** The weight parameter in Eqn.(11), $\gamma = 1$. **(iv)** The input length for training and inference, $T = 10$.



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

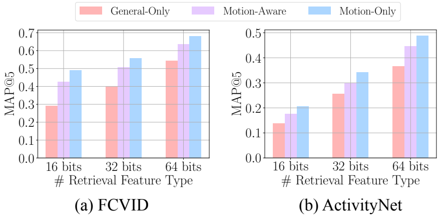
(a) ActivityNet 16 bits (b) ActivityNet 32 bits (c) ActivityNet 64 bits

Figure 2: The MAP@K results of different methods under 16, 32, and 64 bits.



(a) FCVID 64 bits (b) ActivityNet 64 bits

Figure 3: The PR curve of MARGH and BTH under 64 bits.



(a) FCVID (b) ActivityNet

Figure 4: The MAP@5 results *w.r.t.* different training protocols.

We use exactly the same frame-wise feature extractors as [22] for a fair *inference* comparison. For clips, we use an R(2+1)D-34 [29] pre-trained on IG-65M [9] to extract 2048-D features. The feature extractors are *frozen* during training.

4.2 Results and Analysis

4.2.1 Comparions with state-of-the-arts

We compared the proposed MAGRH with 8 state-of-the-art video hashing methods: BTH [22], NPH [21], SSVH [27], SSTH [41], JTAE [19], MFH [26], ITQ [11], and DH [7]. The MAP@K results are reported in Figure 2. Our MAGRH outperforms all previous methods under all code lengths by a large margin. Specifically, the MAP@K (K=5, 20, 60, 100) of MAGRH surpasses the most competitive method, *i.e.*, BTH, by **19.28%**, **21.21%**, **19.49%**, **17.28%** on FCVID and **13.87%**, **8.18%**, **3.48%**, **2.14%** on ActivityNet under 16 bits.

To show the superiority of our method sufficiently, we further draw the PR curve of MAGRH and BTH in Figure 3. It can be seen that MARGH always achieves higher precision under the same recall rate on all datasets, especially when the recall rate is low. We owe the performance gain to two aspects: (**i**) The cascaded GRMs, *i.e.*, the branch-specific and

Table 2: The MAP@K *w.r.t.* object and scene subsets on FCVID under 64 bits.

Method	Object		Scene	
	K=5	K=100	K=5	K=100
BTH	0.8608	0.6671	0.6840	0.3582
MAGRH	**0.9516**	**0.8406**	**0.8783**	**0.6841**

Table 3: The inference efficiency comparison between MAGRH and four state-of-the-art methods.

Method	SSTH	SSVH	NPH	BTH	MAGRH
Inference Frames	25	25	25	25	**10**
Encoding Time (ms)	0.88	1.03	1.42	1.18	**0.46**

Table 4: The MAP@K results *w.r.t.* different architectures under 16 bits.

Method	FCVID				ActivityNet			
	K=5	K=20	K=60	K=100	K=5	K=20	K=60	K=100
MAGRH	**0.4015**	**0.3329**	**0.2773**	**0.2408**	**0.2551**	**0.1430**	**0.0609**	**0.0383**
MAGRH$_{\text{linear}}$	0.2494	0.1803	0.1384	0.1188	0.1370	0.0716	0.0330	0.0217
MAGRH$_{\text{w/o specific}}$	0.3856	0.3144	0.2573	0.2221	0.2362	0.1326	0.0571	0.0358
MAGRH$_{\text{w/o agnostic}}$	0.3861	0.3130	0.2539	0.2185	0.2429	0.1343	0.0582	0.0364

branch-agnostic GRM, capture the temporal dependency precisely as well as prevent information loss in the interaction between general and motion features. (**ii**) With explicit motion knowledge transfer, *i.e.*, Eqn.(8), the model can dig crucial motion information within limited frames, thus generating high-quality "motion-aware" hash codes while other hashing methods are less efficient to capture such dynamic semantic components.

We also take the robustness into consideration, and evaluate our MARGH under 64 bits on two FCVID subsets where motion is not predominant: (**i**) Object, including 1,089 videos among cow, dolphin, elephant, laptop, and TabletPC. (**ii**) Scene, including 1,038 videos among beach, mountain, desert, river, and sunset. The retrieval results are in Table 2. The previous SOTA, *i.e.*, BTH, performs well on the object subset but badly on the scene subset, while our MAGRH performs well on both subsets. It implies that **as a video-specific characteristic, motion always helps video hashing.** Besides, the GRMs also contribute to capturing temporal consistency. Therefore, our MAGRH is robust in real scenarios.

It is worth noting that our MAGRH also has higher inference efficiency. We list the inference frames and the encoding time *w.r.t.* several state-of-the-art methods in Table 3. Benefit from the learned "motion-aware" hashing function, our MAGRH can yield superior performance with fewer frames, thus achieving 2× acceleration compared to other methods.

4.2.2 Component Analysis

To explore the contribution of each component, we design three variants of MAGRH: **(i)** MAGRH$_{\text{linear}}$ that removes the two GRMs. We directly send averaged frame (clip) features to the hash layers. **(ii)** MAGRH$_{\text{w/o specific}}$ that removes the branch-specific GRM. **(iii)** MAGRH$_{\text{w/o agnostic}}$ that removes the branch-agnostic GRM. The MAP@K results under 16 bits are reported in Table 4. Compared to MAGRH, the performance of MAGRH$_{\text{linear}}$ drops largely. It implies the proposed sequence-level graph reasoning, *i.e.*, the GRM, models the temporal dependency effectively and plays a key role in video understanding. Besides, MAGRH also outperforms MARGH$_{\text{w/o specific}}$ and MAGRH$_{\text{w/o agnostic}}$. It indicates the cascaded design of GRMs eases training by "decoupling" the learning process of clues from different branches, thus preventing information loss.

4.2.3 Ablation Study

The effectiveness of motion knowledge transfer scheme: To verify the effectiveness of the proposed motion knowledge transfer scheme, *i.e.*, $\mathcal{L}_{\text{Inter-CL}}$, we compared the performance with two training protocols: **(i)** General-Only that only takes T frames as the input. In infer-

Table 5: The MAP@K results *w.r.t.* different contrastive configuration under 16 bits.

Method	FCVID				ActivityNet			
	K=5	K=20	K=60	K=100	K=5	K=20	K=60	K=100
MAGRH	**0.4015**	**0.3329**	**0.2773**	**0.2408**	**0.2551**	**0.1430**	**0.0609**	**0.0383**
$\text{MAGRH}_{\text{full\_h}}$	0.2375	0.1446	0.0891	0.0689	0.1048	0.0453	0.0194	0.0125
$\text{MAGRH}_{\text{intra\_h}}$	0.3783	0.2964	0.2295	0.1917	0.2099	0.1085	0.0460	0.0290
$\text{MAGRH}_{\text{inter\_h}}$	0.3738	0.2863	0.2154	0.1765	0.1978	0.0977	0.0422	0.0268

ence, the hash codes are purely produced by general features without any motion guidance. **(ii)** Motion-Only that only takes T clips as the input. In inference, the hash codes are purely produced by motion features. For the two protocols, the model is trained without $\mathcal{L}_{\text{Inter-CL}}$. We report the MAP@5 results in Figure 4. It is not surprising that the Motion-Only setting achieves the best performance because the hash codes preserve most motion information. But such a clip-wise model is not practical in real scenarios due to the heavy computational burden. On the other hand, the General-Only setting achieves inferior performance because of a lack of motion guidance. Compared to the above two settings, our Motion-Aware setting makes a good balance between accuracy and efficiency, *i.e.*, it achieves comparable performance with Motion-Only and surpasses General-Only by a large margin, while the computation cost is affordable as the input only requires T frames in inference.

The optimal contrastive configuration in $\mathcal{L}_{\textbf{Intra-CL}}$ and $\mathcal{L}_{\textbf{Inter-CL}}$: Note that we directly align binary hash codes b in $\mathcal{L}_{\text{Intra-CL}}$ (Eqn.(7)) and $\mathcal{L}_{\text{Inter-CL}}$ (Eqn.(9)), while aligning the real-valued vectors h is also a reasonable choice. Therefore, we further design three variants to figure out the optimal contrastive configuration for learnable hashing: **(i)** $\text{MAGRH}_{\text{full\_h}}$ that replaces all hash codes b with real-valued vectors h in $\mathcal{L}_{\text{Intra-CL}}$ and $\mathcal{L}_{\text{Inter-CL}}$. **(ii)** $\text{MAGRH}_{\text{intra\_h}}$ that only replaces the hash codes b with real-valued vectors h in $\mathcal{L}_{\text{Intra-CL}}$. **(iii)** $\text{MAGRH}_{\text{inter\_h}}$ that only replaces the hash codes b with real-valued vectors h in $\mathcal{L}_{\text{Inter-CL}}$. We report the MAP@K results under 16 bits in Table 5. MAGRH outperforms the above three contrastive configurations by a large margin, especially for the $\text{MAGRH}_{\text{full\_h}}$. We imply that the optimization in float space is less effective for video hashing because binarization can be regarded as an extra regularization, and some image quantization works [39, 40] have presented similar conclusions.

5 Conclusions

In this paper, we propose ***M****otion-****A****ware* ***G****raph* ***R****easoning* ***H****ashing* (MAGRH) for self-supervised video retrieval. Different from existing frame-wise video hashing methods, we explicitly extract clip-wise motion features and transfer motion knowledge by designing a dual-branch architecture with a combination of intra- and inter-branch contrastive objectives. Besides, we make an early attempt to model temporary dependency via graph reasoning. Extensive experiments show the superiority of MAGRH over state-of-the-art methods, as the “motion-aware” hash codes yield a remarkable performance gain even with limited inference frames, and achieve 2× acceleration compared to previous works. In the future, we may explore training schemes to make motion knowledge transfer more efficient.

Acknowledgement

This work is supported in part by the National Natural Science Foundation of China under grant 62171248, the PCNL KEY project (PCL2021A07), the Guangdong Basic and Applied Basic Research

Foundation under grant 2021A1515110066, and the GXWD 20220811172936001.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

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

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[4] Zhixiang Chen, Xin Yuan, Jiwen Lu, Qi Tian, and Jie Zhou. Deep hashing via discrepancy minimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6838–6847, 2018.

[5] Ling-Yu Duan, Jie Lin, Zhe Wang, Tiejun Huang, and Wen Gao. Weighted component hashing of binary aggregated descriptors for fast visual search. *IEEE Transactions on multimedia*, 17(6):828–842, 2015.

[6] Yueqi Duan, Jiwen Lu, Jianjiang Feng, and Jie Zhou. Context-aware local binary feature learning for face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1139–1153, 2017.

[7] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. Deep hashing for compact binary codes learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2475–2483, 2015.

[8] Zerun Feng, Zhimin Zeng, Caili Guo, and Zheng Li. Exploiting visual semantic reasoning for video-text retrieval. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 1005–1011, 2021.

[9] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12046–12055, 2019.

[10] Nikolaos Gkalelis, Andreas Goulas, Damianos Galanopoulos, and Vasileios Mezaris. Objectgraphs: Using objects and a graph convolutional network for the bottom-up recognition and explanation of events in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3375–3383, 2021.

[11] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence*, 35(12): 2916–2929, 2012.

[12] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

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

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

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

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

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

[18] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview, 2017.

[19] Chao Li, Yang Yang, Jiewei Cao, and Zi Huang. 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*, pages 9–17, 2017.

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

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

[22] Shuyan Li, Xiu Li, Jiwen Lu, and Jie Zhou. Self-supervised video hashing via bidirectional transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13549–13558, 2021.

[23] Ji Lin, Chuang Gan, Kuan Wang, and Song Han. Tsm: Temporal shift module for efficient and scalable video understanding on edge devices. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[24] Wei Liu, Jun Wang, Rongrong Ji, Yu-Gang Jiang, and Shih-Fu Chang. Supervised hashing with kernels. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2074–2081. IEEE, 2012.

[25] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115 (3):211–252, 2015.

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

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

[28] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2019.

[29] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.

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

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

[32] Andrés Villa, Kumail Alhamoud, Victor Escorcia, Fabian Caba, Juan León Alcázar, and Bernard Ghanem. vclimb: A novel video class incremental learning benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19035–19044, 2022.

[33] Jinpeng Wang, Ziyun Zeng, Bin Chen, Tao Dai, and Shu-Tao Xia. Contrastive quantization with code memory for unsupervised image retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2468–2476, 2022.

[34] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018.

[35] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018.

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

[37] Guangnan Ye, Dong Liu, Jun Wang, and Shih-Fu Chang. Large-scale video hashing via structure learning. In *Proceedings of the IEEE international conference on computer vision*, pages 2272–2279, 2013.

[38] Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. In *International Conference on Learning Representations*, 2018.

[39] Tan Yu, Jingjing Meng, Chen Fang, Hailin Jin, and Junsong Yuan. Product quantization network for fast visual search. *International Journal of Computer Vision*, 128(8):2325–2343, 2020.

[40] Ziyun Zeng, Jinpeng Wang, Bin Chen, Tao Dai, and Shu-Tao Xia. Pyramid hybrid pooling quantization for efficient fine-grained image retrieval. *arXiv preprint arXiv:2109.05206*, 2021.

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