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

1 Appendix

1.1 The Derivation of $\mathcal{L}_{\text{Inter-CL}}$

Given the motion hash code $\mathcal{M}$ and the general hash code $\mathcal{G}$ of video v, we first define a distribution q with a latent variable C as follows:

$$\begin{aligned} q(\mathcal{M},\mathcal{G}|C=1) &= p(\mathcal{M},\mathcal{G}) \\ q(\mathcal{M},\mathcal{G}|C=0) &= p(\mathcal{M})p(\mathcal{G}) \end{aligned} \tag{1}$$

where $p(\mathcal{M}), p(\mathcal{G})$ denote the distribution of $\mathcal{M},\mathcal{G}$ respectively, $p(\mathcal{M},\mathcal{G})$ denotes the joint distribution of $\mathcal{M},\mathcal{G}$. According to Eqn.(1), $C=1$ implies the two hash codes $\mathcal{M},\mathcal{G}$ are correlated, while $C=0$ implies they are independent.

For a batch consisting of video v and another N videos, we have:

$$q(C=1) = \frac{1}{N+1},\ q(C=0) = \frac{N}{N+1} \tag{2}$$

Then the posterior probability $q(C=1|\mathcal{M},\mathcal{G})$ can be derived via *Bayes Rule*:

$$\begin{aligned} q(C=1|\mathcal{M},\mathcal{G}) &= \frac{q(\mathcal{M},\mathcal{G},C=1)}{q(\mathcal{M},\mathcal{G},C=1)+q(\mathcal{M},\mathcal{G},C=0)} \\ &= \frac{q(\mathcal{M},\mathcal{G}|C=1)q(C=1)}{q(\mathcal{M},\mathcal{G}|C=1)q(C=1)+q(\mathcal{M},\mathcal{G}|C=0)q(C=0)} \\ &= \frac{p(\mathcal{M},\mathcal{G})}{p(\mathcal{M},\mathcal{G})+Np(\mathcal{M})p(\mathcal{G})} \end{aligned} \tag{3}$$

✉ Bin Chen is the corresponding author.

Next, we take the log form of Eqn.(3) and have:

$$\begin{aligned}\log q(C=1|\mathcal{M},\mathcal{G}) &= \log\frac{p(\mathcal{M},\mathcal{G})}{p(\mathcal{M},\mathcal{G})+Np(\mathcal{M})p(\mathcal{G})} = -\log\left(1+\frac{Np(\mathcal{M})p(\mathcal{G})}{p(\mathcal{M},\mathcal{G})}\right)\\ &\leq -\log\frac{Np(\mathcal{M})p(\mathcal{G})}{p(\mathcal{M},\mathcal{G})} = -\log N+\log\frac{p(\mathcal{M},\mathcal{G})}{p(\mathcal{M})p(\mathcal{G})}\end{aligned} \tag{4}$$

note Eqn.(4) can be rewritten as:

$$\log\frac{p(\mathcal{M},\mathcal{G})}{p(\mathcal{M})p(\mathcal{G})} \geq \log N+\log q(C=1|\mathcal{M},\mathcal{G}) \tag{5}$$

then we can derive the *Mutual Information Bound* by multiplying $p(\mathcal{M},\mathcal{G})$ on both sides of Eqn.(5) and taking the expectation form:

$$\begin{aligned}I(\mathcal{M};\mathcal{G}) &= \mathbb{E}_{p(\mathcal{M},\mathcal{G})}\log\frac{p(\mathcal{M},\mathcal{G})}{p(\mathcal{M})p(\mathcal{G})} \geq \log N+\mathbb{E}_{p(\mathcal{M},\mathcal{G})}\log q(C=1|\mathcal{M},\mathcal{G})\\ &= \log N+\mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=1)}\log q(C=1|\mathcal{M},\mathcal{G})\end{aligned} \tag{6}$$

where $I(\mathcal{M};\mathcal{G})$ denotes the mutual information between $\mathcal{M}$ and $\mathcal{G}$. Eqn.(6) can be easily expanded by adding a negative term $\mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=0)}\log q(C=0|\mathcal{M},\mathcal{G})$, *i.e.*,

$$\begin{aligned}I(\mathcal{M};\mathcal{G}) &\geq \log N+\mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=1)}\log q(C=1|\mathcal{M},\mathcal{G})+N\mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=0)}\log q(C=0|\mathcal{M},\mathcal{G})\\ &= \log N+\mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=1)}\log q(C=1|\mathcal{M},\mathcal{G})+N\mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=0)}\log\left(1-q(C=1|\mathcal{M},\mathcal{G})\right)\end{aligned} \tag{7}$$

On the other hand, transferring knowledge between $\mathcal{M}$ and $\mathcal{G}$ means maximizing their mutual information $I(\mathcal{M};\mathcal{G})$, which is equivalent to maximizing the lower bound as proposed in Eqn.(7). Since we do not know the real probability of $q(C=1|\mathcal{M},\mathcal{G})$, we replace it with a distance function $d(\mathcal{M},\mathcal{G})$ such that $d(\mathcal{M},\mathcal{G})$ should be small when $\mathcal{M},\mathcal{G}$ are derived from the same video v, otherwise, $d(\mathcal{M},\mathcal{G})$ should be large.

Therefore, the learning objective is:

$$\mathcal{L}_d(\mathcal{M},\mathcal{G}) = \mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=1)}\log d(\mathcal{M},\mathcal{G})+N\mathbb{E}_{q(\mathcal{M},\mathcal{G}|C=0)}\log\left(1-d(\mathcal{M},\mathcal{G})\right) \tag{8}$$

[2] has proven that Eqn.(8) is an equivalent form of InfoNCE [1]. Finally, we can replace $\mathcal{M},\mathcal{G}$ with b^g_*, b^m_* in our manuscript and derive $\mathcal{L}_{\text{Inter-CL}}$:

$$\hat{\mathcal{L}}^*_{\text{CL}} = -\sum_{b^g_*,b^m_*\in\mathcal{B}}\log\frac{\exp(s(b^g_*,b^m_*)/\hat{\tau})}{\exp(s(b^g_*,b^m_*)/\hat{\tau})+\sum_{b^m_{*^-}\in\mathcal{B}\setminus\{b^g_*,b^m_*\}}\exp(s(b^g_*,b^m_{*^-})/\hat{\tau})} \tag{9}$$

$$\mathcal{L}_{\text{Inter-CL}} = \frac{1}{2}\left(\hat{\mathcal{L}}^q_{\text{CL}}+\hat{\mathcal{L}}^k_{\text{CL}}\right) \tag{10}$$

where $\hat{\tau}$ denotes the temperature parameter.

1.2 Additional Experiments

1.2.1 Ablation Study

The training scheme: In our default training scheme, we optimize intra-branch representation and transfer motion knowledge synchronously. But the motion feature itself may not be

Table 1: The MAP@K results *w.r.t.* different training schemes under 16 bits.

Variant	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**
$\text{MAGRH}_{\text{asyn}}$	**0.4027**	**0.3369**	**0.2798**	**0.2426**	0.2539	0.1370	0.0584	0.0366

Table 2: The MAP@K *w.r.t.* 2 weight sharing strategies on ActivityNet under 16 bits.

Variant	K=5	K=20	K=60	K=100
MAGRH	**0.2551**	**0.1430**	**0.0609**	**0.0383**
$\text{MAGRH}_{\text{all specific}}$	0.2480	0.1411	0.0604	0.0383
$\text{MAGRH}_{\text{all agnostic}}$	0.2454	0.1348	0.0590	0.0370

Table 3: The MAP@K *w.r.t.* different input embeddings on ActivityNet under 16 bits.

Variant	K=5	K=20	K=60	K=100
MAGRH	**0.2551**	**0.1430**	**0.0609**	**0.0383**
$\text{MAGRH}_{\text{no pos}}$	0.2452	0.1378	0.0576	0.0347
$\text{MAGRH}_{\text{no bs}}$	0.2487	0.1403	0.0590	0.0371
$\text{MAGRH}_{\text{no AGG}}$	0.2472	0.1392	0.0581	0.0366
$\text{MAGRH}_{\text{feats only}}$	0.2444	0.1361	0.0562	0.0356

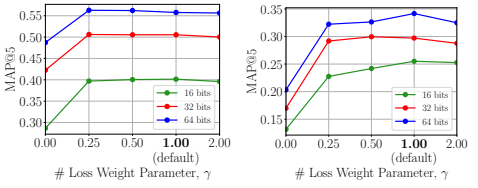
good enough in the early stage, thus hurting the learning process of the motion-aware hashing function. Therefore, we design an asynchronous training scheme, namely $\text{MAGRH}_{\text{asyn}}$, by first training the auxiliary branch alone, then fixing it and transferring motion knowledge to the main branch. Table 1 shows the MAP@K results under 16 bits. The $\text{MAGRH}_{\text{asyn}}$ does not bring much performance gain. Therefore, we choose the synchronous training scheme since it does not require an extra training stage and leaves the training scheme efficiency for future study.

The weight sharing strategy: We use branch-specific/agnostic graphs because it eases training difficulty, as we decouple the learning stage. The frame-wise clues are easier to be preserved in the branch-specific graph, while the branch-agnostic graph focuses on motion learning. We conduct experiments on ActivityNet under 16 bits with different weight sharing strategies to verify the effectiveness of our design intuitively. Specifically, we develop two variants: **(i)** $\text{MAGRH}_{\text{all specific}}$ that adopts two branch-specific graphs, *i.e.*, all weights are private. **(ii)** $\text{MAGRH}_{\text{all agnostic}}$ that adopts two branch-agnostic graphs, *i.e.*, all weights are shared. As shown in Table 2, both of them perform worse than MARGH.

The impact of different input embeddings: Note that besides the origin feature embedding, we add three additional embeddings for the input of MAGRH in our implementation: (**i**) The learnable positional embedding that helps to build the topology relation of the graph. (**ii**) The branch-specific embedding that helps the modality-agnostic GRM to distinguish the graph type, *i.e.* frame and clip graph, which benefits intra-branch representation learning. (**iii**) The [AGG] token performs like the "readout" operation in graph reasoning, which aggregates the information over the full graph and prevents information loss. To exploit their effectiveness, we conduct experiments with four variants: **(i)** $\text{MAGRH}_{\text{no pos}}$ that removes the positional embedding. **(ii)** $\text{MAGRH}_{\text{no bs}}$ that removes the branch-specific embedding. **(iii)** $\text{MAGRH}_{\text{no AGG}}$ that removes the [AGG] token. **(iv)** $\text{MAGRH}_{\text{feats only}}$ that removes all extra embeddings. The MAP@K results are shown in Table 3, both of the four variants underperform, implying their contribution to the graph reasoning modules.

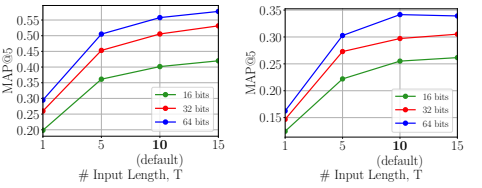
1.2.2 Parameter Sensitivity

We report the MAP@5 results *w.r.t.* different loss weight parameter γ in $\mathcal{L}_{\text{MAGRH}}$ and input length T in Figure 1 and 2 respectively. It shows γ is more sensitive on ActivityNet than FCVID because ActivityNet is a motion-centric dataset. Setting $\gamma = 0$ removes $\mathcal{L}_{\text{Inter-CL}}$,



(a) FCVID (b) ActivityNet

Figure 1: The sensitivity of the loss weight parameter γ in $\mathcal{L}_{\text{MAGRH}}$.



(a) FCVID (b) ActivityNet

Figure 2: The sensitivity of the input length (frame rate) T.

Figure 3: The top-5 retrieved results *w.r.t.* MAGRH (the upper row) and BTH (the bottom row) under 64 bits on FCVID and ActivityNet.

thus leading to inferior performance. On the other hand, when setting γ to a large value, *e.g.*, $\gamma = 2$, the performance degrades because of ignorance of sub-important clues, *e.g.*, objects and scenes. As for input length T, it is reasonable that the model achieves better performance with larger T. Since the gain is limited when $T > 10$, we set $T = 10$ to balance performance and computation overhead.

1.2.3 Visualization

To show the superiority of the proposed MAGRH intuitively, we illustrate the top-5 retrieved results by MAGRH and the state-of-the-art hashing method, *i.e.*, BTH, under 64 bits in Figure 3. The relevance of the top-5 retrieved videos returned by MARGH is consistently higher than that of BTH, especially when retrieving motion-centric videos. For instance, BTH falsely returned "Car Parking" when the query video is "Car Racing" possibly because it only focuses on the object "Car". Instead, our MAGRH stresses "Racing" and returns videos with higher relevance.

References

[1] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

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