A Proposal-based Approach for Activity Image-to-Video Retrieval

Ruicong Xu, Li Niu,* Jianfu Zhang, Liqing Zhang*
MoE Key Lab of Artificial Intelligence,
Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, China.
{ranranxu, utscnewly, c.sis}@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract

Activity image-to-video retrieval task aims to retrieve videos containing the similar activity as the query image, which is a challenging task because videos generally have many background segments irrelevant to the activity. In this paper, we utilize R-C3D model to represent a video by a bag of activity proposals, which can filter out background segments to some extent. However, there are still noisy proposals in each bag. Thus, we propose an Activity Proposal-based Image-to-Video Retrieval (APIVR) approach, which incorporates multi-instance learning into cross-modal retrieval framework to address the proposal noise issue. Specifically, we propose a Graph Multi-Instance Learning (GMIL) module with graph convolutional layer, and integrate this module with classification loss, adversarial loss, and triplet loss in our cross-modal retrieval framework. Moreover, we propose geometry-aware triplet loss based on point-to-subspace distance to preserve the structural information of activity proposals. Extensive experiments on three widely-used datasets verify the effectiveness of our approach.

1 Introduction

Cross-modal retrieval task has attracted considerable research attention in the field of retrieval task. With the rapid development of video applications, a specific type of retrieval task, Activity Image-to-Video Retrieval (AIVR), comes into our sight. The goal of AIVR task is to retrieve the videos containing the similar activity as the image query, which expands its value in widespread applications. One daily-life example is news videos searching with a provided photo containing a particular activity. Another example is fitness videos recommendation based on a sports picture.

The key idea of cross-modal retrieval is to learn a common feature space, where cross-modal data of relevant semantic can be close to each other. Although there are abundant methods for cross-modal retrieval like text-image retrieval (Feng, Wang, and Li 2014; Hardoon, Szedmák, and Shawe-Taylor 2004; Peng, Huang, and Qi 2016; Wang et al. 2016; 2013), few methods (de Araújo and Girod 2018; Xu et al. 2017) are proposed for image-video retrieval. However, these methods are not specifically designed for AIVR task. AIVR task is in high demand of meaningful video representations, because a video may contain background segments irrelevant to the activity and poor video representations without considering noisy background segments will lead to inferior performance of AIVR task.

Recently, RNN (Ng et al. 2015; Srivastava, Mansimov, and Salakhutdinov 2015) and 3D CNN (Ji et al. 2013; Tran et al. 2015; Qiu, Yao, and Mei 2017) are used to extract deep learning-based video representations. As an extension of 3D CNN, R-C3D (Xu, Das, and Saenko 2017) can generate candidate temporal regions containing activities and filter out noisy background segments to obtain the superior activity video representations. Therefore, we take advantage of R-C3D model to generate temporal proposals that are most likely to contain activities and extract one feature vector for each proposal, leading to a bag of proposal features for each video. This paper is the first to target at AIVR task by utilizing activity proposals for videos.

In this paper, we propose an Activity Proposal-based Image-to-Video Retrieval (APIVR) approach for AIVR task. The major innovation in our paper is incorporating Graph Multi-Instance Learning (GMIL) module into cross-modal retrieval framework to address the proposal noise issue. As illustrated in Figure 1, our cross-modal retrieval framework is based on Adversarial Cross-Modal Retrieval (ACMR) proposed in (Wang et al. 2017), in which image features and activity proposal-based video features are projected into a common feature space steered by triplet loss, classification loss, and adversarial loss. To address the proposal noise issue, we treat each video as a bag and the activity proposals in each bag as multiple instances, which coincides with multi-instance learning (MIL) paradigm (Ilse, Tomczak, and Welling 2018). We assume that there is at least one clean instance in each bag, and employ self-attention mechanism to learn different weights for multiple instances, with higher weights indicating clean activity proposals. To further consider the relation among multiple instances in each bag, we insert graph convolutional layer into MIL module, yielding a novel Graph MIL (GMIL) module.

After learning weights based on our GMIL module, we

*Corresponding author

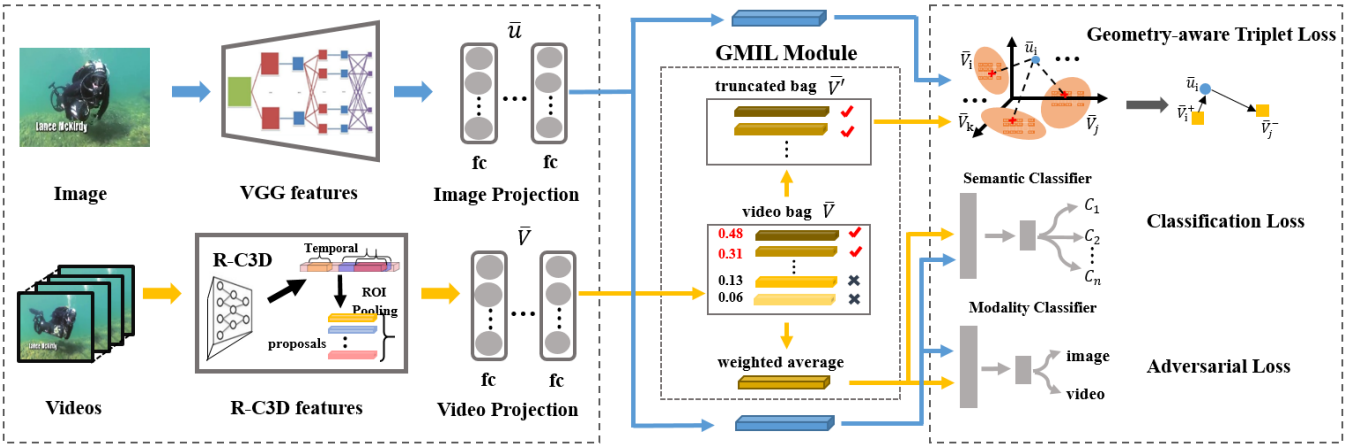

Figure 1: The flowchart of our proposed approach. The image features and bags of activity proposal features for videos are extracted by VGG (Simonyan and Zisserman 2014) and R-C3D (Xu, Das, and Saenko 2017) models respectively, and then projected into a common feature space. Our retrieval framework consists of triplet loss, classification loss, and adversarial loss. We incorporate Graph Multi-Instance Learning (GMIL) module into retrieval framework to address the proposal noise issue. We also design geometry-aware triplet loss based on truncated bag of activity proposals. Best viewed in color.

use weighted average of activity proposal features in each bag as input for the classification loss and adversarial loss in cross-modal retrieval framework, to suppress noisy activity proposals. For the remaining triplet loss, we propose a novel geometry-aware triplet loss, which calculates the point-to-subspace distance between image and bag of activity proposals. Considering that the noisy activity proposals may mislead the point-to-subspace distance, we use truncated bag of activity proposals based on the weights learnt by our GMIL module. Thus, our geometry-aware triplet loss can mitigate the proposal noise issue and simultaneously preserve the geometry property of activity proposals.

The contributions of our paper are summarized as follows:

- This work is the first activity proposal-based approach for activity image-to-video retrieval task. Our major contribution is incorporating multi-instance learning into cross-modal retrieval framework to address the proposal noise issue.
- Our two minor contributions are Graph Multi-Instance Learning (GMIL) module with graph convolutional layer and geometry-aware triplet loss based on truncated bag of activity proposals.
- Experiment results on three datasets, *i.e.*, action-based THUMOS'14 and ActivityNet datasets, event-based MED2017 Event dataset, demonstrate the superiority of our approach compared to state-of-the-art methods.

2 Related Work

In this section, we provide a brief overview of video representation, cross-modal retrieval, and multi-instance learning.

Video representations: Video representations play a crucial role in image-to-video retrieval task. Recently, deep learning-based models, *e.g.*, RNN (Jiang et al. 2018) and 3D CNN (Qiu, Yao, and Mei 2017), are proposed to fully exploit spatio-temporal information across consecutive frames. As an advanced 3D CNN model, R-C3D (Xu, Das, and Saenko 2017) can generate activity proposals across temporal dimension to filter out noisy background segments. Hence, we adopt R-C3D to generate video representations, which significantly facilitates the AIVR task.

Cross-modal retrieval methods: Cross-modal retrieval methods fall into two major categories: binary-value based methods (Yu et al. 2014; Lin, Shen, and van den Hengel 2014; Ye et al. 2017; Ding, Guo, and Zhou 2014; Xu et al. 2017) and real-value based retrieval methods (Zhai, Peng, and Xiao 2014; Wang et al. 2016; Peng, Huang, and Qi 2016; Peng et al. 2018; Wang, Li, and Lazebnik 2016; Wang et al. 2017; Zhen et al. 2019). Our cross-modal retrieval framework is based on ACMR (Wang et al. 2017), which consists of classification loss, triplet loss, and adversarial loss. Our contribution is incorporating graph multi-instance learning module into cross-modal retrieval framework together with geometry-aware triplet loss.

Multi-instance learning: Multi-instance learning (MIL) groups training samples into multi-instance bags, in which each bag contains at least one positive instance. Some early methods (Li et al. 2009) treat one bag as an entirety or infers instance labels within each bag. Recently, deep multi-instance learning methods (Zhu et al. 2017; Pappas and Popescu-Belis 2014; Ilse, Tomczak, and Welling 2018) employ pooling operators or trainable operators to aggregate multiple instances in each bag. Moreover, several graph MIL (Tu et al. 2019; Guo and Yi 2013) methods are proposed to exploit the graph structure of training bags in different ways, but their methods cannot be easily integrated into our cross-modal retrieval framework.

3 Methodology

In this section, we introduce our activity proposal-based image-to-video retrieval approach.

3.1 Problem Definition

For concise mathematical expression, we denote a matrix (*e.g.*, $\mathbf{A}$) and vector (*e.g.*, $\mathbf{a}$) using an uppercase and lowercase letter in boldface respectively, and denote a scalar (*e.g.*, a) using a lowercase letter. We use $\mathbf{I}_k$ and $\mathbf{A}^T$ to denote an identity matrix with size k and the transpose of $\mathbf{A}$ respectively. By using $\mathbf{vec}(\cdot)$, we perform column-wise concatenation to transform a matrix into a column vector. Moreover, we use $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ to denote the inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$.

In the AIVR task, our training process is based on mini-batches of video-image pairs $\{(\boldsymbol{V}_i, \boldsymbol{u}_i)|_{i=1}^n\}$, in which $(\boldsymbol{V}_i, \boldsymbol{u}_i)$ is a pair of video and image with the same category label, and n is the number of pairs in a mini-batch. Specifically, $\boldsymbol{V}_i = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_k\}$ with $\boldsymbol{h}_k \in \mathbb{R}^{d_1 \times 1}$ is a bag of proposal features in the i-th video and $\boldsymbol{u}_i \in \mathbb{R}^{d_2 \times 1}$ is the feature of the i-th image. Note that the dimensionalities of the image feature and activity proposal features are not equal in our problem, *i.e.*, $d_1 \neq d_2$. Each pair $(\boldsymbol{V}_i, \boldsymbol{u}_i)$ is associated with a one-hot label vector $\boldsymbol{y}_i$ with the entry corresponding to its category as one. In the testing stage, given an image query, the goal of the AIVR task is to retrieve the videos related to the activity in the image.

3.2 Activity Proposal-based Image-to-Video Retrieval (APIVR) Approach

As mentioned above, we represent each video as a bag of proposal features $\boldsymbol{V} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_k\}$ and each image as a feature vector $\boldsymbol{u}$. Considering the different statistical properties and data distributions of videos and images, we project video and image features into a common feature space with the mapping function $f_v(\cdot)$ and $f_u(\cdot)$ respectively. The mapping functions are defined as

$$\begin{aligned} f_v(\boldsymbol{V}) &= \{f_v(\boldsymbol{h}_1), f_v(\boldsymbol{h}_2), ..., f_v(\boldsymbol{h}_k)\} \\ &= \{\bar{\boldsymbol{h}}_1, \bar{\boldsymbol{h}}_2, ..., \bar{\boldsymbol{h}}_k\} = \bar{\boldsymbol{V}}, \quad (1) \\ f_u(\boldsymbol{u}) &= \bar{\boldsymbol{u}}, \quad (2) \end{aligned}$$

where $f_v : \mathbb{R}^{d_1 \times k} \rightarrow \mathbb{R}^{r \times k}$, $f_u : \mathbb{R}^{d_2 \times 1} \rightarrow \mathbb{R}^{r \times 1}$. The mapping functions $f_v(\cdot)$ (*resp.*, $f_u(\cdot)$) are three fully-connected layers with model parameters denoted as $\boldsymbol{\theta}_p$.

Based on the projected features $\bar{\boldsymbol{V}}$ and $\bar{\boldsymbol{u}}$, following ACMR (Wang et al. 2017), we employ three types of losses: triplet loss, classification loss, and adversarial loss. Concretely, triplet loss pulls an image close to the videos of the same category while pushing it far away from the videos of a different category. The classification loss targets at successfully separating the training samples from different categories regardless of modalities, which can preserve semantic information and simultaneously minimize the modality gap. The adversarial loss is involved in a minimax game by discriminating two modalities with a modality classifier and generating modality-agnostic representations to confuse the modality classifier, which can further reduce the modality gap. In summary, the above three types of losses jointly contribute to modality consistency and semantic distinguishability in the common feature space.

Graph Multi-Instance Learning Module In the common feature space, although we use R-C3D model to generate activity proposals from each video which are very likely to contain the activity, there still remain some noisy activity proposals irrelevant to the activity. Hence, each video is comprised of a mixture of clean and noisy proposals. If we utilize these noisy activity proposals based on the video label, the quality of semantic learning will be greatly degraded. In fact, this problem can be formulated as multi-instance learning, in which each video is treated as a bag and the activity proposals in each bag are treated as instances. Based on the assumption that there should be at least one clean instance in each bag, we expect to assign higher weights on the clean instances and lower weights on the noisy ones, so that the clean instances will play a dominant role in video bags.

Given a bag of instances $\bar{\boldsymbol{V}} = \{\bar{\boldsymbol{h}}_1, \bar{\boldsymbol{h}}_2, ..., \bar{\boldsymbol{h}}_k\}$, inspired by (Ilse, Tomczak, and Welling 2018), we employ self-attention mechanism to learn different weights for different instances in each bag as (3). In particular, we apply a fully-connected layer $\boldsymbol{L}_1 \in \mathbb{R}^{r \times r'}$ with non-linear operation $tanh(\cdot)$ to $\bar{\boldsymbol{V}}$, producing $tanh(\bar{\boldsymbol{V}}^T \boldsymbol{L}_1)$. Then, we apply another fully-connected layer $\boldsymbol{L}_2 \in \mathbb{R}^{r' \times 1}$ followed by softmax layer to obtain the k-dim weight vector $\boldsymbol{a}$ for $\bar{\boldsymbol{V}}$.

$$\boldsymbol{a} = softmax(tanh(\bar{\boldsymbol{V}}^T \boldsymbol{L}_1)\, \boldsymbol{L}_2). \quad (3)$$

However, the above process ignores the relation among multiple instances in each bag. To take such relation into consideration, we insert graph convolutional layer (Kipf and Welling 2017) into (3), which can leverage the graph structure of each bag. Graph convolutional layer (Kipf and Welling 2017) is originally proposed for semi-supervised learning and now we employ it for multi-instance learning. Following (Kipf and Welling 2017), we calculate the similarity graph $\boldsymbol{S}$ for each bag $\boldsymbol{V} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_k\}$ during preprocessing, in which S_{ij} is the cosine similarity between $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$. Besides, we define $\boldsymbol{S}' = \boldsymbol{S} + \boldsymbol{I}_k$ and a diagonal matrix $\boldsymbol{D}$ with $D_{ii} = \sum_j S'_{ij}$. Then, graph convolutional layer can be represented by a 1×1 convolution layer with parameters $\bar{\boldsymbol{S}} = \boldsymbol{D}^{-1/2} \boldsymbol{S}' \boldsymbol{D}^{-1/2}$. We insert two graph convolutional layers into (3) and arrive at

$$\hat{\boldsymbol{a}} = softmax(\bar{\boldsymbol{S}}\, tanh(\bar{\boldsymbol{S}} \bar{\boldsymbol{V}}^T \boldsymbol{L}_1)\, \boldsymbol{L}_2). \quad (4)$$

The generated $\hat{\boldsymbol{a}}$ is expected to be smoother than $\boldsymbol{a}$, *i.e.*, the weights of two instances in a bag should be close when these two instances are similar. The theoretical proof and more details can be found in (Kipf and Welling 2017).

At last, we obtain the weighted average of instance features as the bag-level feature $Z(\bar{\boldsymbol{V}}) = \sum_{j=1}^k \hat{a}_j \bar{\boldsymbol{h}}_j$. By assigning different weights on different activity proposals, we aim to focus more on the clean proposals and obtain discriminative video features.

Geometry-aware Triplet Loss with GMIL We use triplet loss to preserve the semantic relevance of similar training

samples across different modalities. As defined in (Schroff, Kalenichenko, and Philbin 2015), triplet loss is based on an anchor sample $\boldsymbol{x}$, a positive sample $\boldsymbol{p}$, and a negative sample $\boldsymbol{n}$, where $\boldsymbol{x}$ has the same category label as $\boldsymbol{p}$ yet a different category label from $\boldsymbol{n}$. Given a triplet $(\boldsymbol{x}, \boldsymbol{p}, \boldsymbol{n})$, triplet loss is used to enforce the distance between $\boldsymbol{x}$ and $\boldsymbol{p}$ to be smaller than that between $\boldsymbol{x}$ and $\boldsymbol{n}$ by a margin.

Since our objective is to retrieve videos by a given image query, anchor sample $\boldsymbol{x}$ is an image while positive sample $\boldsymbol{p}$ and negative sample $\boldsymbol{n}$ are videos. In a mini-batch of video-image pairs $\{(\bar{\boldsymbol{V}}_i, \bar{\boldsymbol{u}}_i)|_{i=1}^n\}$, with each image $\bar{\boldsymbol{u}}_i$ being an anchor sample, we use its paired video sample as the positive sample $\bar{\boldsymbol{V}}_i^+$ and one video from a different category as the negative sample $\bar{\boldsymbol{V}}_j^-$, leading to in total n triplets in a mini-batch. Then our triplet loss is formulated as

$$L_{triplet} = \sum_{i,j} \Big| \, d(\bar{\boldsymbol{u}}_i, \bar{\boldsymbol{V}}_i^+) - d(\bar{\boldsymbol{u}}_i, \bar{\boldsymbol{V}}_j^-) + m \, \Big|_+, \quad (5)$$

in which m is the margin set as 0.1 in our experiments, $d(\boldsymbol{x}, \boldsymbol{y})$ is the distance between $\boldsymbol{x}$ and $\boldsymbol{y}$, and $|x|_+ = x$ if $x > 0$ and 0 otherwise. For $d(\bar{\boldsymbol{u}}, \bar{\boldsymbol{V}})$, a straightforward approach is calculating the distance between $\bar{\boldsymbol{u}}$ and weighted average of activity proposal features $Z(\bar{\boldsymbol{V}})$, but that will cause serious loss of structural information in activity proposals. As shown in (Xu et al. 2017), point-to-subspace distance[1] is able to preserve the structural information and geometric property. In our problem, an image can be seen as a high-dimensional data point and video is a subspace spanned by activity proposals. Then the point-to-subspace distance is the Euclidean distance between an image point and its orthogonal projection on the subspace of videos.

Considering that noisy proposals may mislead point-to-subspace distance, we use truncated bag of proposals in lieu of intact bag of proposals. To be exact, we denote truncated bag as $\bar{\boldsymbol{V}}' = \bar{\boldsymbol{V}}[:, \mathcal{S}_b]$, in which $\mathcal{S}_b$ is the index set of proposals with top-b GMIL weights $\hat{a}_i$. That means, we use the top-b clean proposals in triplet loss. With simple mathematical derivation[1], the orthogonal projection of point $\bar{\boldsymbol{u}}$ on subspace $\bar{\boldsymbol{V}}'$ can be calculated as $\widetilde{\boldsymbol{V}}\bar{\boldsymbol{u}}$, where $\widetilde{\boldsymbol{V}} = \bar{\boldsymbol{V}}'((\bar{\boldsymbol{V}}')^T\bar{\boldsymbol{V}}')^{-1}(\bar{\boldsymbol{V}}')^T$. Then, the point-to-subspace distance between $\bar{\boldsymbol{u}}$ and $\bar{\boldsymbol{V}}'$, *i.e.*, Euclidean distance between $\bar{\boldsymbol{u}}$ and $\widetilde{\boldsymbol{V}}\bar{\boldsymbol{u}}$, can be simplified as

$$\begin{aligned} d\left(\bar{\boldsymbol{u}}, \bar{\boldsymbol{V}}'\right) &= \left\| \bar{\boldsymbol{u}} - \widetilde{\boldsymbol{V}}\bar{\boldsymbol{u}} \right\|_2^2 = Tr((\boldsymbol{I}_r - \widetilde{\boldsymbol{V}})^T(\boldsymbol{I}_r - \widetilde{\boldsymbol{V}})\bar{\boldsymbol{u}}\bar{\boldsymbol{u}}^T) \\ &= \bar{\boldsymbol{u}}^T\bar{\boldsymbol{u}} - \left\langle \mathbf{vec}(\widetilde{\boldsymbol{V}}), \mathbf{vec}(\bar{\boldsymbol{u}}\bar{\boldsymbol{u}}^T) \right\rangle \end{aligned} \quad (6)$$

By using $\widetilde{d}\left(\bar{\boldsymbol{u}}, \widetilde{\boldsymbol{V}}\right)$ to denote $\left\langle \mathbf{vec}(\widetilde{\boldsymbol{V}}), \mathbf{vec}(\bar{\boldsymbol{u}}\bar{\boldsymbol{u}}^T) \right\rangle$ and substituting (6) into (5), we can arrive at

$$\begin{aligned} L_{triplet} &= \sum_{i,j} \left| \, d(\bar{\boldsymbol{u}}_i, \bar{\boldsymbol{V}}_i'^+) - d(\bar{\boldsymbol{u}}_i, \bar{\boldsymbol{V}}_j'^-) + m \, \right|_+ \\ &= \sum_{i,j} \left| \, \widetilde{d}\left(\bar{\boldsymbol{u}}_i, \widetilde{\boldsymbol{V}}_j^-\right) - \widetilde{d}\left(\bar{\boldsymbol{u}}_i, \widetilde{\boldsymbol{V}}_i^+\right) + m \, \right|_+ . \end{aligned} \quad (7)$$

[1]https://en.wikipedia.org/wiki/Projection_(linear_algebra)

Following (Yao, Mei, and Ngo 2015), given an anchor sample $\bar{\boldsymbol{u}}_i$, we tend to select its hardest negative sample $\bar{\boldsymbol{V}}_j^-$ and the details are omitted here. Based on (7), we tend to minimize $L_{triplet}$ by optimizing GMIL module parameters $\boldsymbol{\theta}_m$ and projection module parameters $\boldsymbol{\theta}_p$.

Classification Loss with GMIL To ensure the training samples in each modality are semantically discriminative, we additionally use a semantic classifier to separate intra-modal training samples from different categories. To minimize the modality gap, we apply the same classifier for both images and videos. In particular, we add a softmax classification layer with model parameters $\boldsymbol{\theta}_c$ on top of the image features $\bar{\boldsymbol{u}}$ and the weighted average of proposal features $Z(\bar{\boldsymbol{V}})$. Given a mini-batch of video-image pairs $\{(\bar{\boldsymbol{V}}_i, \bar{\boldsymbol{u}}_i)|_{i=1}^n\}$ associated with one-hot label $\{\boldsymbol{y}_i|_{i=1}^n\}$, the classification loss is written as follows,

$$L_{class} = -\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{y}_i^T(log(p(Z(\bar{\boldsymbol{V}}_i))) + log(p(\bar{\boldsymbol{u}}_i))), \quad (8)$$

in which $p(\cdot)$ denotes the prediction scores by using the softmax classification layer. Defining GMIL module parameters $\boldsymbol{\theta}_m = \{\boldsymbol{L}, \boldsymbol{w}\}$, we tend to minimize L_{class} by optimizing semantic classifier parameters $\boldsymbol{\theta}_c$, GMIL module parameters $\boldsymbol{\theta}_m$, and projection module parameters $\boldsymbol{\theta}_p$.

Adversarial Loss with GMIL To further minimize the modality gap across videos and images, adversarial learning (Goodfellow et al. 2014; Wang et al. 2017) is implemented as an interplay between discriminating modalities by learning a modality classifier and learning representations to confuse the modality classifier. In the process of discriminating modalities, we learn a modality classifier to discriminate the video modality from the image modality. The modality classifier is implemented as a binary classifier with model parameters $\boldsymbol{\theta}_d$, in which we assume the label of video (*resp.*, image) modality is 1 (*resp.*, 0). In the process of learning representations to confuse the modality classifier, we expect the projected video/image features in the common feature space could fool the modality classifier. Considering that clean proposals have more representative feature distribution while the noisy proposals are scattered throughout the feature space, we apply the modality classifier on the weighted average of proposal features $Z(\bar{\boldsymbol{V}})$ for videos. Similar to the classification loss, the adversarial loss is formally defined as

$$L_{adv} = -\frac{1}{n}\sum_{i=1}^{n} log(\delta(Z(\bar{\boldsymbol{V}}_i)) + log(1 - \delta(\bar{\boldsymbol{u}}_i)), \quad (9)$$

where $\delta(\cdot)$ is the predicted probability of being from video modality. As adversarial learning is an interplay between discriminating modalities and learning representations, in the process of discriminating modalities, we tend to minimize L_{adv} by optimizing the modality classifier parameters $\boldsymbol{\theta}_d$. On the contrary, in the process of learning representations, we tend to maximize L_{adv} by optimizing projection module parameters $\boldsymbol{\theta}_p$ and GMIL module parameters $\boldsymbol{\theta}_m$.

The Whole Algorithm We collect $L_{triplet}$, L_{class}, and L_{adv} in (7), (8), (9) as the following total training loss:

$$L_{total} = \alpha \cdot L_{triplet} + \beta \cdot L_{class} - L_{adv}, \quad (10)$$

where α and β are trade-off parameters and empirically fixed as 0.1 and 10 respectively in our experiments.

Due to the adversarial loss L_{adv} in (10), we play a minimax game by learning representations and discriminating modalities alternatingly. By using $\boldsymbol{\theta}_g = \{\boldsymbol{\theta}_p, \boldsymbol{\theta}_m, \boldsymbol{\theta}_c\}$ to denote the model parameters in learning representations, our objective can be written as follows,

$$\min_{\boldsymbol{\theta}_g} \max_{\boldsymbol{\theta}_d} \ \alpha \cdot L_{triplet} + \beta \cdot L_{class} - L_{adv}, \qquad (11)$$

which can be optimized by updating $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_d$ in an alternating manner. We leave the summary of our training algorithm to Supplementary due to space limitation. In the testing, we pass the testing images and videos through our trained model, yielding the projected features $\bar{\boldsymbol{u}}$ (*resp.*, $Z(\bar{\boldsymbol{V}})$) for images (*resp.*, videos). Then, given a query image $\boldsymbol{u}_i$, we retrieve its relevant videos by ranking all the l_2 distances between $\bar{\boldsymbol{u}}_i$ and $Z(\bar{\boldsymbol{V}})$.

4 Experiments

In this section, we compare our APIVR approach with the state-of-the-art methods on three datasets and provide extensive ablation studies.

4.1 Datasets Construction

To the best of our knowledge, there are no publicly available datasets of activity video-image pairs specifically designed for the AIVR task. Therefore, we construct video-image datasets for the AIVR task based on public video datasets, *i.e.*, THUMOS'14[2], ActivityNet (Heilbron et al. 2015) and MED2017 Event[3] datatsets, in which THUMOS'14 and ActivityNet datasets are action-based datasets while MED2017 Event dataset is an event-based dataset. The difference between "action" and "event" lies in that an event generally consists of a sequence of interactive or stand-alone actions. The details of above three datasets are left to Supplementary. Based on the above three datasets, we aim to obtain activity images and activity video clips, which can be used to construct our datasets for AIVR task.

To obtain activity video clips, considering that long videos may belong to multiple activity categories, we divide each long video into multiple short videos based on the activity temporal annotations to ensure that each short video only belongs to one activity category. Then, we sample a fixed number of consecutive key frames in each short video as a video clip. The number of key frames used in our experiments is 768 for all datasets, which is large enough to cover at least one activity instance.

To obtain activity images, we first locate the activity intervals in long videos according to activity temporal annotations. Then, we sample images from those activity intervals so that each image should belong to one activity category.

With obtained activity images and activity video clips, we sample video clips and images from each category to form training pairs and testing pairs. Particularly, for THUMOS'14 dataset, we form 1500 training pairs and 406 testing pairs. For ActivityNet dataset, we form 4800 training pairs and 1200 testing pairs. For MED2017 Event dataset, we form 2200 training pairs and 404 testing pairs.

[2]http://crcv.ucf.edu/THUMOS14/

[3]https://www.nist.gov/itl/iad/mig/med-2017-evaluation/

Method	mean Average Precision (mAP)			
	@10	@20	@50	@100
APIVR (w/o TL)	0.3228	0.3096	0.2956	0.2875
APIVR (w/o AL)	0.3278	0.3146	0.3026	0.2905
APIVR (w/o CL)	0.2438	0.2389	0.2312	0.2306
APIVR (w/o GA)	0.3531	0.3376	0.3204	0.3145
APIVR (w/o $GMIL$)	0.3428	0.3368	0.3276	0.3102
APIVR (w/o $Graph$)	0.3618	0.3521	0.3326	0.3285
Full APIVR approach	**0.3812**	**0.3645**	**0.3459**	**0.3314**

Table 1: Comparision of our full APIVR approach and our special cases in terms of mAP@K on THUMOS'14. Best results are denoted in boldface.

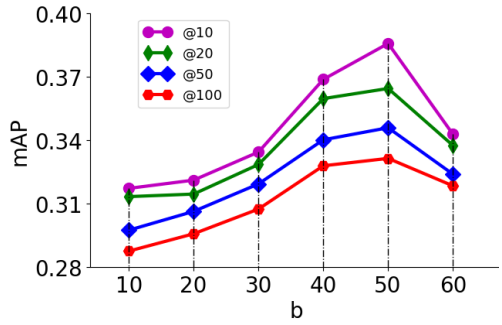

Figure 2: The effect of top-b proposals chosen from video bags to represent videos on THUMOS'14 dataset.

4.2 Implementation Details

For images, we employ VGG model (Simonyan and Zisserman 2014) to extract the fc7 layer features and then reduce the dimension from 4096-dim to 128-dim by PCA for the ease of memory and computation in our experiment.

For video clips, we use R-C3D model to generate activity proposals, which is pretrained on Sports-1M dataset and finetuned on UCF101 dataset (Tran et al. 2015). We extract a 4096-dim feature vector for each activity proposal and each video is represented by a bag of top-60 proposal features, *i.e.*, $k = 60$, by ranking the scores that may contain activities. In our geometry-aware triplet loss, we use top-50 proposals in each bag, *i.e.*, $b = 50$.

In the projection module, mapping functions $f_v(\cdot)$ (*resp.*, $f_u(\cdot)$) are implemented as three fully-connected layers as follows. $f_v : \boldsymbol{V}(d_1 = 4096) \rightarrow 500 \rightarrow 200 \rightarrow \bar{\boldsymbol{V}}(r = 64)$ and $f_u : \boldsymbol{u}(d_2 = 128) \rightarrow 100 \rightarrow 80 \rightarrow \bar{\boldsymbol{u}}(r = 64)$. In our experiments, we use mAP@K, *i.e.*, mean Aversion Precision based on top K retrieved results, as the evaluation metric.

4.3 Ablation Studies

In order to explore the effectiveness of different components in our approach, we investigate some special cases of our approach. Specifically, we study the contributions of three types of losses by comparing with APIVR (w/o TL), APIVR

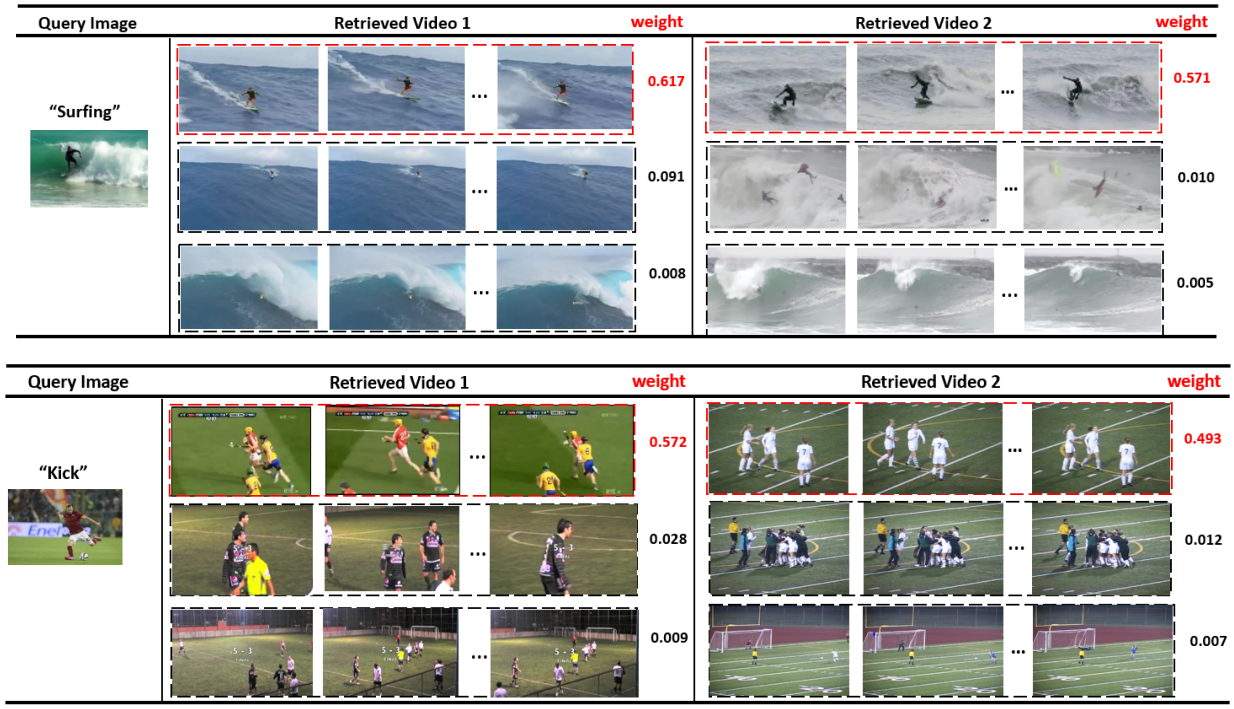

Figure 3: Illustration of activity proposal weights learnt by our GMIL module on the ActivityNet dataset. The clean proposal is assigned with the highest weight (marked in red) and the other two noisy proposals are assigned with the lowest weights.

(w/o AL), and APIVR (w/o CL), which are our three special cases by ablating Triplet Loss (TL), Adversarial Loss (AL), and Classification Loss (CL) respectively. Besides, to verify the benefit of geometry-aware triplet loss, we replace $d\left(\bar{\boldsymbol{u}}, \bar{\boldsymbol{V}}'\right)$ in (6) with $\left\|\bar{\boldsymbol{u}} - Z(\bar{\boldsymbol{V}})\right\|_2^2$ and refer to this special case as APIVR (w/o GA). To demonstrate the effectiveness of our Graph Multi-Instance Learning (GMIL) module, we replace GMIL module in (4) with MIL module in (3), and name this case as APIVR (w/o $Graph$). Furthermore, we also assign uniform weights to proposals in each video instead of learning weights using GMIL module and name this special case as APIVR (w/o $GMIL$), which means that $Z(\bar{\boldsymbol{V}}) = \frac{1}{k}\sum_{s=1}^{k} \bar{\boldsymbol{h}}_{\boldsymbol{s}}$ in (8) (9) and intact bags of activity proposals are used in (7).

By taking THUMOS'14 dataset as an example, experimental results are reported in Table 1. Obviously, we can see that APIVR (w/o TL), APIVR (w/o AL), and APIVR (w/o CL) are all inferior to our full APIVR approach, which indicates that each type of loss plays an essential role in our cross-modal framework and contributes to the overall performance. Based on the results of three losses, compared with adversarial loss and triplet loss, we can see that classification loss has more influence on the performance, which proves the significance of semantic classifier in our approach. When using standard triplet loss instead of geometry-aware triplet loss, APIVR (w/o GA) suffers from a drop in performance, which demonstrates that it is beneficial to preserve the structural information and geometric property of activity proposals. Moreover, we can also note that the results of APIVR (w/o $GMIL$) are worse than the full APIVR approach, which proves the benefit of paying more attention to clean proposals based on our GMIL module. More results of ablating GMIL for each loss are provided in Supplementary. Finally, we can observe that APIVR (w/o $graph$) underperforms the full APIVR approach, which shows the advantage of inserting graph convolutional layer into MIL module.

Recall that we use truncated bags of top-b clean proposals in our geometry-aware triplet loss. To investigate the impact of b, we vary b and report the performance of our full APIVR approach in Figure 2. We can observe that $b = 50$ achieves the best performance, and the intact bags of proposals, *i.e.*, $b = 60$, may harm the performance because of the included noisy proposals. When b is very small (*i.e.*, $b \leq 30$), too much useful information is discarded and thus the performance is also unsatisfactory.

4.4 Visualization of Retrieved Videos

To better demonstrate the effectiveness of our GMIL module for identifying clean proposals, we provide two representative qualitative results in Figure 3, in which the query image belongs to the category "surfing" (*resp.*, "kick") in the top (*resp.*, "bottom") row. We list top-2 retrieved videos for each query image. For each retrieved video, we show one proposal with the highest weight and another two proposals with the lowest weights. It is obvious that the proposals with the highest weight can capture the relevant activity while the other two proposals are less relevant or even background

Methods	THUMOS'14 dataset				MED2017 Event dataset			
	mAP@10	mAP@20	mAP@50	mAP@100	mAP@10	mAP@20	mAP@50	mAP@100
ITQ	0.2613	0.2572	0.2477	0.2340	0.2284	0.2168	0.2127	0.2034
SpH	0.2131	0.2080	0.2033	0.1914	0.2044	0.1926	0.1878	0.1611
SKLSH	0.2004	0.1974	0.1951	0.1847	0.1956	0.1924	0.1883	0.1774
CBE-opt	0.2687	0.2601	0.2554	0.2483	0.2268	0.2128	0.2051	0.1984
MFH	0.2402	0.2398	0.2188	0.2128	0.2246	0.2192	0.2108	0.1994
SCM	0.2661	0.2576	0.2484	0.2395	0.2113	0.2041	0.1962	0.1924
CMFH	0.2545	0.2513	0.2466	0.2331	0.2262	0.2169	0.2101	0.2088
BPBC	0.2724	0.2706	0.2684	0.2571	0.2488	0.2501	0.2451	0.2402
JRL	0.2770	0.2656	0.2526	0.2411	0.2347	0.2278	0.2203	0.2198
CCL	0.3222	0.3188	0.3072	0.2949	0.2454	0.2417	0.2321	0.2267
JFSSL	0.2367	0.2351	0.2325	0.2241	0.2292	0.2218	0.2131	0.2064
Corr-AE	0.2266	0.2178	0.2096	0.2104	0.2032	0.2011	0.1971	0.1918
DSPE	0.2632	0.2544	0.2443	0.2312	0.2312	0.2246	0.2161	0.2004
CMDN	0.2927	0.2892	0.2754	0.2714	0.2328	0.2342	0.2250	0.2171
ACMR	0.3361	0.3274	0.3107	0.3061	0.2518	0.2401	0.2373	0.2244
DSCMR	0.3621	0.3523	0.3251	0.3188	0.2665	0.2576	0.2470	0.2381
APIVR	**0.3812**	**0.3645**	**0.3459**	**0.3314**	**0.3049**	**0.2973**	**0.2867**	**0.2771**

Table 2: mAP@K of different methods on THUMOS'14 and MED2017 Event dataset. Best results are denoted in boldface.

Method	mean Average Precision (mAP)			
	@10	@20	@50	@100
ITQ	0.1851	0.1704	0.1598	0.1414
SpH	0.1885	0.1843	0.1617	0.1551
SKLSH	0.1638	0.1595	0.1556	0.1474
CBE-opt	0.2044	0.1970	0.1842	0.1768
MFH	0.2155	0.2048	0.1977	0.1932
SCM	0.2285	0.2230	0.2166	0.2011
CMFH	0.2334	0.2318	0.2205	0.2155
BPBC	0.2352	0.2296	0.2184	0.2071
JRL	0.2266	0.2182	0.2177	0.2096
CCL	0.2358	0.2208	0.2138	0.2082
JFSSL	0.2166	0.2087	0.1958	0.1929
Corr-AE	0.2024	0.2012	0.1924	0.1866
DSPE	0.2212	0.2107	0.2079	0.2055
CMDN	0.2422	0.2401	0.2288	0.2232
ACMR	0.2318	0.2224	0.2111	0.2091
DSCMR	0.2481	0.2344	0.2287	0.2122
APIVR	**0.2635**	**0.2545**	**0.2488**	**0.2319**

Table 3: Performance of different methods in terms of mAP@K on the ActivityNet dataset. Best results are denoted in boldface.

segments, which indicates the great advantages of our GMIL module in identifying clean proposals.

4.5 Comparisons with the State-of-the-art Methods

We compared our APIVR approach with the state-of-the-art methods including single modality hashing methods CBE-opt (Yu et al. 2014), ITQ (Gong and Lazebnik 2011), SKLSH (Raginsky and Lazebnik 2009), SpH (Heo et al. 2012), multiple modalities hashing methods MFH (Ye et al. 2017), SCM (Zhang and Li 2014), CMFH (Ding, Guo, and Zhou 2014), BPBC (Xu et al. 2017), and cross-modal retrieval methods Corr-AE (Feng, Wang, and Li 2014), CMDN (Peng, Huang, and Qi 2016), ACMR (Wang et al. 2017), DSPE (Wang, Li, and Lazebnik 2016), JRL (Zhai, Peng, and Xiao 2014), JFSSL (Wang et al. 2016), CCL (Peng et al. 2018), DSCMR (Zhen et al. 2019). Among them, BPBC is a hashing method targeting at image-to-video retrieval task. Although the method in (de Araújo and Girod 2018) also targets at image-to-video retrieval, but it focuses on improving video Fisher Vectors using bloom filters and thus cannot be directly applied to our problem. Besides, Corr-AE, CMDN, ACMR, CCL, DSPE and DSCMR are deep learning-based methods and have achieved remarkable results in cross-modal retrieval task. For all baselines, we take the average of proposal features extracted by R-C3D as the video features and VGG fc7 features as the image features for fair comparison. The encoding length in the hashing methods is set to 128-bit.

The experiment results are summarized in Table 2, 3. Compared with ACMR (Wang et al. 2017) method, which has a similar framework to ours, our superior performance confirms the advantages of preserving structural information using geometric projection and attending clean proposals using GMIL module. Obviously, we can see that our approach achieves significant improvement over all baselines in all scope of K on both action-based and event-based datasets. For example, on the ActivityNet dataset with the largest number of categories, APIVR approach outperforms the other methods by about 2% in all scope of K.

5 Conclusion

In this paper, we have proposed the first activity proposal-based image-to-video retrieval (APIVR) approach for the activity image-to-video retrieval task. We have incorporated graph multi-instance learning module into cross-modal retrieval framework to address the proposal noise issue, and also proposed geometry-aware triplet loss. Experiments on three datasets have demonstrated the superiority of our approach compared to the state-of-the-art methods.

6 Supplementary

6.1 Details of Datasets

We construct our datasets based on three public datasets: THUMOS'14[4], ActivityNet (Heilbron et al. 2015) and MED2017 Event[5] datatsets, in which THUMOS'14 and ActivityNet datasets are action-based datasets while MED2017 Event dataset is an event-based dataset. The details of the above three datasets are introduced as follows:

THUMOS'14 dataset: The THUMOS'14 dataset consists of 2765 trimmed training videos and 200 untrimmed validation videos from 20 different sport activities. We merge similar categories such as "cliff diving" and "diving", and obtain a total number of 18 categories.

ActivityNet dataset: The ActivityNet dataset(1.3) contains 200 activity categories. Due to the limit of the GPU memory and speed, we only use the validation set with 4926 videos. Similar to THUMOS'14 dataset, we merge similar categories such as "clean and jerk" and "snatch", leading to in total 156 categories.

MED2017 Event dataset: ActivityNet dataset The Multimedia Event Detection (MED) launched by TRECVID consists of more complicated events, which is also suitable to explore the AIVR task. We use the resources for the Pre-Specified Event portion of the MED2017 evaluation. The dataset has totally 200 trimmed videos distributed in 10 event categories.

6.2 Training Algorithm

Recall that our objective function is

$$\min_{\boldsymbol{\theta}_g} \max_{\boldsymbol{\theta}_d} \ \alpha \cdot L_{triplet} + \beta \cdot L_{class} - L_{adv}, \qquad (12)$$

in which α and β are trade-off parameters and empirically fixed as 10 and 0.01 respectively. The problem in (12) is a minimax problem, which can be optimized by updating $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_d$ in an alternating manner. The details of our training process are shown in Algorithm 1, in which we set the number of image-video pairs in a mini-batch $n = 64$, the number of generation steps $t = 50$, and the learning rate $\lambda = 0.0001$.

Algorithm 1 The training process of our APIVR approach.

Input: Mini-batches of video-image pairs $\{(\boldsymbol{V}_i, \boldsymbol{u}_i)|_{i=1}^n\}$ and the associated labels $\{\boldsymbol{y}_i|_{i=1}^n\}$. The number of steps t, learning rate λ, and trade-off parameters α, β.

update until convergence:

1: **for** t steps **do**
2: update $\boldsymbol{\theta}_g$ by **descending** stochastic gradients: $\boldsymbol{\theta}_g \leftarrow \boldsymbol{\theta}_g - \lambda \cdot \bigtriangledown_{\boldsymbol{\theta}_g}(\alpha \cdot L_{triplet} + \beta \cdot L_{class} - L_{adv})$.
3: **end for**
4: update parameter $\boldsymbol{\theta}_d$ by **ascending** stochastic gradients: $\boldsymbol{\theta}_d \leftarrow \boldsymbol{\theta}_d + \lambda \cdot \bigtriangledown_{\boldsymbol{\theta}_d}(\alpha \cdot L_{triplet} + \beta \cdot L_{class} - L_{adv})$.
5: **return** Model parameters $\boldsymbol{\theta}_g$ and $\boldsymbol{\theta}_d$

[4]http://crcv.ucf.edu/THUMOS14/

[5]https://www.nist.gov/itl/iad/mig/med-2017-evaluation/

References

de Araújo, A. F., and Girod, B. 2018. Large-scale video retrieval using image queries. *T-CSVT* 28(6):1406–1420.

Ding, G.; Guo, Y.; and Zhou, J. 2014. Collective matrix factorization hashing for multimodal data. In *CVPR*.

Feng, F.; Wang, X.; and Li, R. 2014. Cross-modal retrieval with correspondence autoencoder. In *MM*.

Gong, Y., and Lazebnik, S. 2011. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*.

Goodfellow, I. J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A. C.; and Bengio, Y. 2014. Generative adversarial networks. *CoRR* abs/1406.2661.

Guo, Z., and Yi, Y. 2013. Graph-based multiple instance learning for action recognition. In *ICIP*.

Hardoon, D. R.; Szedmák, S.; and Shawe-Taylor, J. 2004. Canonical correlation analysis: An overview with application to learning methods. *Neural Computation* 16(12):2639–2664.

Heilbron, F. C.; Escorcia, V.; Ghanem, B.; and Niebles, J. C. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*.

Heo, J.; Lee, Y.; He, J.; Chang, S.; and Yoon, S. 2012. Spherical hashing. In *CVPR*.

Ilse, M.; Tomczak, J. M.; and Welling, M. 2018. Attention-based deep multiple instance learning. In *Proceedings of the 35th International Conference on Machine Learning, ICML*.

Ji, S.; Xu, W.; Yang, M.; and Yu, K. 2013. 3d convolutional neural networks for human action recognition. *T-PAMI* 35(1):221–231.

Jiang, Y.; Wu, Z.; Wang, J.; Xue, X.; and Chang, S. 2018. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *T-PAMI* 40(2):352–364.

Kipf, T. N., and Welling, M. 2017. Semi-supervised classification with graph convolutional networks. In *ICLR*.

Li, Y.; Kwok, J. T.; Tsang, I. W.; and Zhou, Z. 2009. A convex method for locating regions of interest with multi-instance learning. In *Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD*.

Lin, G.; Shen, C.; and van den Hengel, A. 2014. Supervised hashing using graph cuts and boosted decision trees. *CoRR* abs/1408.5574.

Ng, J. Y.; Hausknecht, M. J.; Vijayanarasimhan, S.; Vinyals, O.; Monga, R.; and Toderici, G. 2015. Beyond short snippets: Deep networks for video classification. In *CVPR*.

Pappas, N., and Popescu-Belis, A. 2014. Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP*.

Peng, Y.; Qi, J.; Huang, X.; and Yuan, Y. 2018. CCL: cross-modal correlation learning with multigrained fusion by hierarchical network. *T-MM* 20(2):405–420.

Peng, Y.; Huang, X.; and Qi, J. 2016. Cross-media shared representation by hierarchical learning with multiple deep networks. In *IJCAI*.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*.

Raginsky, M., and Lazebnik, S. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *NIPS*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. Facenet: A unified embedding for face recognition and clustering. In *CVPR*.

Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

Srivastava, N.; Mansimov, E.; and Salakhutdinov, R. 2015. Unsupervised learning of video representations using LSTMs. In *ICML*.

Tran, D.; Bourdev, L. D.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*.

Tu, M.; Huang, J.; He, X.; and Zhou, B. 2019. Multiple instance learning with graph neural networks. *CoRR* abs/1906.04881.

Wang, K.; He, R.; Wang, W.; Wang, L.; and Tan, T. 2013. Learning coupled feature spaces for cross-modal matching. In *ICCV*.

Wang, K.; He, R.; Wang, L.; Wang, W.; and Tan, T. 2016. Joint feature selection and subspace learning for cross-modal retrieval. *T-PAMI* 38(10):2010–2023.

Wang, B.; Yang, Y.; Xu, X.; Hanjalic, A.; and Shen, H. T. 2017. Adversarial cross-modal retrieval. In *MM*.

Wang, L.; Li, Y.; and Lazebnik, S. 2016. Learning deep structure-preserving image-text embeddings. In *CVPR*.

Xu, R.; Yang, Y.; Shen, F.; Xie, N.; and Shen, H. T. 2017. Efficient binary coding for subspace-based query-by-image video retrieval. In *MM*.

Xu, H.; Das, A.; and Saenko, K. 2017. R-C3D: region convolutional 3d network for temporal activity detection. In *ICCV*.

Yao, T.; Mei, T.; and Ngo, C. 2015. Learning query and image similarities with ranking canonical correlation analysis. In *ICCV*.

Ye, D.; Li, Y.; Tao, C.; Xie, X.; and Wang, X. 2017. Multiple feature hashing learning for large-scale remote sensing image retrieval. *ISPRS* 6(11):364.

Yu, F. X.; Kumar, S.; Gong, Y.; and Chang, S. 2014. Circulant binary embedding. In *ICML*.

Zhai, X.; Peng, Y.; and Xiao, J. 2014. Learning cross-media joint representation with sparse and semisupervised regularization. *T-CSVT* 24(6):965–978.

Zhang, D., and Li, W. 2014. Large-scale supervised multimodal hashing with semantic correlation maximization. In *AAAI*.

Zhen, L.; Hu, P.; Wang, X.; and Peng, D. 2019. Deep supervised cross-modal retrieval. In *CVPR*.

Zhu, W.; Lou, Q.; Vang, Y. S.; and Xie, X. 2017. Deep multi-instance networks with sparse label assignment for whole mammogram classification. In *Medical Image Computing and Computer Assisted Intervention - MICCAI*.