VRAG: Region Attention Graphs for Content-Based Video Retrieval

Kennard Ng,1 Ser-Nam Lim,2 Gim Hee Lee 1
1 Department of Computer Science, National University of Singapore
2 Facebook AI
kennard.ng.pool.hua@gmail.com, sernam@gmail.com, gimhee.lee@nus.edu.sg

Abstract

Content-based Video Retrieval (CBVR) is used on media-sharing platforms for applications such as video recommendation and filtering. To manage databases that scale to billions of videos, video-level approaches that use fixed-size embeddings are preferred due to their efficiency. In this paper, we introduce Video Region Attention Graph Networks (VRAG) that improves the state-of-the-art of video-level methods. We represent videos at a finer granularity via region-level features and encode video spatio-temporal dynamics through region-level relations. Our VRAG captures the relationships between regions based on their semantic content via self-attention and the permutation invariant aggregation of Graph Convolution. In addition, we show that the performance gap between video-level and frame-level methods can be reduced by segmenting videos into shots and using shot embeddings for video retrieval. We evaluate our VRAG over several video retrieval tasks and achieve a new state-of-the-art for video-level retrieval. Furthermore, our shot-level VRAG shows higher retrieval precision than other existing video-level methods, and closer performance to frame-level methods at faster evaluation speeds. Finally, our code will be made publicly available.

Introduction

The volume of videos on the Internet has grown exponentially with the inception of media-sharing websites such as Facebook, Twitch and Youtube. Content-based Video Retrieval (CBVR) is important on these platforms for applications such as video recommendation and video filtering.

In CBVR, evaluating the video similarity is a key component. There are predominantly two types of approaches for inferring video-to-video similarity: frame-level approaches and video-level approaches. Video-level methods encode videos into fixed size embeddings, and measure video similarity through the similarity of their embeddings. On the other hand, frame-level methods derive video similarity by aggregating pairwise similarities between video frames. These two approaches form a dichotomy. Video-level approaches offer faster evaluation speeds, while frame-level methods provide more accurate video similarities at significant overheads. Although practical implementations of frame-level methods may employ optimizations such as summarizing the video into a shorter sequence of frames (Ji et al. 2017; Fajtl et al. 2018; Zhou, Qiao, and Xiang 2018), video-level methods remain orders of magnitude more efficient than frame-level methods when the number of videos scale to the size of billions. Consequently, video-level methods remain highly relevant for real-world applications despite its inferior performance to frame-level methods.

Recently, several video-level CBVR works (Kordopatis-Zilos et al. 2017a; Baraldi et al. 2018; Kordopatis-Zilos et al. 2017b; Poullot et al. 2015) propose the same design principle of extracting local frame-level features, and then aggregating these features into fixed-size video embeddings. As a result, most of these works focus on proposing a better frame-level feature extractor and/or video-level feature pooling methods. For example, DML (Kordopatis-Zilos et al. 2017b) and LBoW (Kordopatis-Zilos et al. 2017a) extract Maximum Activation of Convolution (MAC) features from each frame and then apply mean pooling and bag-of-words, respectively, to get video-level representations. LAMV (Baraldi et al. 2018) later propose representing frames at a finer granularity using Region Maximum Activation of Convolutions (R-MAC) and learnable pooling of these frame descriptors in the Fourier Domain (Poullot et al. 2015). While prior works have made strides in frame-level feature extraction and video-level pooling, these approaches...
encode each frame separately, and do not model the spatial and temporal interactions inherent within videos.

In view that prior video-level approaches lack modelling of spatio-temporal interactions within videos, we propose VRAG: a region attention graph-based framework for content-based video retrieval. As shown in Figure 1, our VRAG models videos at the fine-grained region-level as a graph. Each node of the graph represents a R-MAC feature vector. To model interactions between region nodes, we encode spatial relations through complete subgraphs between regions in the same frame and temporal relations through fully connected edges across adjacent frames. We further augment these spatial and temporal connections through self-attention, which modulates the strength of these associations through the affinities between their region-level content. Consequently, we transform each region into context-aware embeddings by selectively aggregating features from neighboring regions via Graph Attention (Veličković et al. 2018) layers. To generate video-level embeddings, we learn attention weights for each region and selectively aggregate region-level embeddings into a fixed size representation. We model our attention pooling on the intuition that important video regions add significant context to other regions and derive the attention weights from the pairwise affinities between regions across multiple Graph Attention layers.

Our VRAG improves the state-of-the-art for video-level approaches across video retrieval tasks (Kordopatis-Zilos et al. 2019; Revaud et al. 2013; Wu, Hauptmann, and Ngo 2007), and reduces the performance gap between video-level and frame-level retrieval approaches. On Event Video Retrieval (Revaud et al. 2013), our VRAG also achieves higher retrieval scores over the state-of-the-art for frame-level retrieval, ViSiL (Kordopatis-Zilos et al. 2019), while being more than 50 times faster. We further propose to reduce the gap between video-level and frame-level approaches by segmenting videos into shots and representing videos over multiple shot embeddings using our VRAG, as illustrated in Figure 2. Our shot-level VRAG evaluates the similarity between videos by aggregating pairwise similarities between their shot embeddings. In our experiments, we show that our shot-level VRAG bridges the gap between video-level and frame-level approaches on most video retrieval tasks (Kordopatis-Zilos et al. 2019; Wu, Hauptmann, and Ngo 2007) at faster evaluation speeds than frame-level methods, and higher retrieval precision over video-level approaches.

Related Work

Video Retrieval

Multi-modal video retrieval performs video retrieval through multi-modal embeddings. Recent methods use pre-trained expert networks to extract video representations across different modalities, e.g. optical flow, audio, and introduce novel techniques to fuse these representations into a multi-modal embedding (Liu et al. 2019; Miech, Laptev, and Sivic 2018; Gabeur et al. 2020). (Liu et al. 2019) models pairwise interactions between modalities and aggregates across modalities via attention pooling while (Gabeur et al. 2020) use Transformers (Vaswani et al. 2017) to fuse features from different modalities.

Video-to-text retrieval is a popular instance of cross-modal video retrieval, where relevant videos are retrieved given a query sentence and vice versa. The video-to-text pipeline extracts video and text representations and map related video-text pairs into the same representation (Liu et al. 2019; Miech et al. 2019).

Content-based Video Retrieval (CBVR)

CBVR methods can be broadly classified into video-level (Poullot et al. 2015; Baraldi et al. 2018; Kordopatis-Zilos et al. 2017b,a) and frame-level (Tan et al. 2009; Chou, Chen, and Lee 2015; Kordopatis-Zilos et al. 2019) approaches.

Video-level approaches encode videos into fixed size embeddings, i.e. vectors. These methods extract features from video frames and temporally pool frame features into a fixed size representation. The similarity between videos is then determined through the similarities between their embeddings e.g. cosine similarity. Compared to prior video-level (Kordopatis-Zilos et al. 2017b,a; Poullot et al. 2015; Baraldi et al. 2018) methods that encode video frames independently, our VRAG models spatio-temporal relations between/within frames through Graph Convolution, and adds video-level context into the frame representations.

Frame-level approaches (Tan et al. 2009; Chou, Chen, and Lee 2015) represent videos as sequences of video frame features that scale with the video length. To evaluate video similarity, these methods aggregate pairwise similarities between video frames to derive the similarity between videos. Recently, ViSiL (Kordopatis-Zilos et al. 2019) propose using Convolutional Neural Networks (CNN) layers and the (Symmetric) Chamfer Similarity to aggregate pairwise frame similarities, and achieves state-of-the-art frame-level CBVR. However, ViSiL requires $n^2$ forward passes to evaluate pairwise similarities between $n$ videos, while our VRAG is orders of magnitude faster since it uses only $n$ forward passes to generate video embeddings.

Our Approach

We present our Video Region Attention Graph (VRAG) network in Figure 3.
Input Frame Representation. Following ViSil, we represent each video frame using concatenated Region Maximum Activation of Convolution (Tolias, Sicre, and Jégou 2016) (R-MAC) features from intermediate convolution layers. We pass each video frame through a pretrained ConvNet, i.e., ResNet50 (He et al. 2015) pre-trained on ImageNet, with $L$ layers to generate intermediate activation maps, $f_{\text{conv}} : \mathcal{T}^{(t)} \rightarrow \left\{ \mathcal{M}_1^{(t)}, \ldots, \mathcal{M}_L^{(t)} \right\}$. Given that each intermediate activation map $\mathcal{M}_i^{(t)} \in \mathbb{R}^{H_i \times W_i \times C_i}$ has varying spatial dimensions, we transform the activation maps into the same spatial dimension by defining different R-MAC kernel sizes for each layer. Subsequently, we concatenate the intermediate R-MAC features channel-wise to represent each video frame, i.e. $f_{\text{rmac}} : \left\{ \mathcal{M}_1^{(t)}, \ldots, \mathcal{M}_L^{(t)} \right\} \rightarrow \mathcal{X}^{(t)} \in \mathbb{R}^{R \times C}$, where $R$ refers to the number of R-MAC features in each frame, and $C = \sum_{i=1}^{L} C_i$ is the dimensions of the concatenated vector. We then encode a video of length $T$ into $N = T \times R$ R-MAC features, i.e. $\mathbf{X} = [\mathcal{X}^{(1)} | \mathcal{X}^{(2)} | \ldots | \mathcal{X}^{(T)}] \in \mathbb{R}^{N \times C}$.

Spatial and Temporal Graph Structure. We represent each R-MAC feature vector $\mathbf{x}_i \in \mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N] \in \mathbb{R}^{N \times C}$ as a node $\mathcal{V}_i$ in the graph $\mathcal{G} = \{ \mathcal{V}, \mathcal{E}_{\text{spatial}}, \mathcal{E}_{\text{temporal}} \}$ of our VRAG. We then add two sets graph edges: 1) $\mathcal{E}_{\text{spatial}}$ is a set of edges that connects region nodes in the same frame $\mathcal{T}^{(t)}$ to capture spatial relationships between these nodes with a frame and this includes a self referencing edge. 2) $\mathcal{E}_{\text{temporal}}$ refers to edges that connect the regions between consecutive frames $\mathcal{T}^{(t-1)}$ and $\mathcal{T}^{(t+1)}$ to capture temporal relations between video frames.

Learning Region Embeddings. From the video region graph $\mathcal{G} = \{ \mathcal{V}, \mathcal{E}_{\text{spatial}}, \mathcal{E}_{\text{temporal}} \}$ constructed earlier, we learn region-level embeddings using Graph Convolutional Network (Kipf and Welling 2017; Veličković et al. 2018) (GCN) layers. To increase the duration of videos that can be processed, we first reduce the dimensions of R-MAC region vectors, i.e. $\mathbf{X} \in \mathbb{R}^{N \times C}$ using a fully-connected layer with non-linear activation, i.e. $f_r : \mathbf{X} \rightarrow \mathbf{X}^{(0)} \in \mathbb{R}^{N \times C'}$. We then pass the region vectors through $K$ Graph Attention (Veličković et al. 2018) layers to generate intermediate region features, i.e. $g^{(k)} : \mathbf{X}^{(k-1)} \rightarrow \mathbf{X}^{(k)} \in \mathbb{R}^{N \times C'}$ for $k \in [1, \ldots, L]$. Specifically, we aggregate the features from neighboring regions as follows:

1) To enable interactions between neighboring regions with complementary representations, we use the key-query self-attention mechanism (Vaswani et al. 2017) to describe the similarity between regions. Specifically, we generate key and query embeddings using linear transformations from the input region vectors $\mathbf{x}_i^{(k-1)}$, i.e.:

$$f_{\text{query}} : \mathbf{x}_i^{(k-1)} \rightarrow \mathbf{a}_{i}^{(k)} \in \mathbb{R}^{C'}, \quad f_{\text{key}} : \mathbf{x}_j^{(k-1)} \rightarrow \mathbf{b}_{j}^{(k)} \in \mathbb{R}^{C'}.$$  

2) The similarity between a region $i$ and its neighboring region $j \in \mathcal{N}(i)$ is defined using the dot product of their query and key embeddings, i.e. $s_{ij}^{(k)} = \mathbf{a}_{i}^{(k)} \cdot \mathbf{b}_{j}^{(k)}$. The output of the Graph Attention layer $g^{(k)}$ is derived as:

$$u_{ij}^{(k)} = \frac{e_{ij}^{(k)}}{\sum_{n \in \mathcal{N}(i)} e_{in}^{(k)}}, \quad f^{(k)} : \sum_{j \in \mathcal{N}(i)} u_{ij}^{(k)} \mathbf{x}_j^{(k-1)} \rightarrow \mathbf{x}_i^{(k)},$$

where $f^{(k)}$ is a linear layer with a non-linear activation.

3) Finally, given that adding graph convolution layers may lead to over-smoothing of region representations (Cai
and Wang 2020; Chen et al. 2019), we propose concatenating region representations along the depth of our network to preserve discriminative features, i.e. $f_{concat}: \{X, X^{(0)}, \ldots, X^{(K)}\} \rightarrow R \in \mathbb{R}^{N \times (C + (K+1)C')}$.  

**Video Embedding.** We aggregate region embeddings $R = [r_1, \ldots, r_N] \in \mathbb{R}^{N \times (C + (K+1)C')}$ using weighted sum aggregation to generate a pooled region representation: $r = \sum_{i=1}^{N} \beta_i r_i$, where $\beta_i$ is the attention weight for region $r_i$. The pooled representation $r$ is fed into two linear layers to generate the video-level embedding $v$, i.e.: $f_{mlp}: R \rightarrow v \in \mathbb{R}^D$. We model our attention weights $\beta = [\beta_1, \ldots, \beta_N] \in \mathbb{R}^N$ from the intuition that important regions in a video act as anchor regions that add context to other regions, and should have high affinities with other regions in the video. Therefore, we compute the attention weights $\beta$ as follows:  
1) We reuse the key and query embeddings of each Graph Attention layer $g(k)$ from Equation 1 to compute $K$ pairwise affinity matrices $\mathcal{A}^{(k)}$ between regions, i.e.  
$$\mathcal{A}^{(k)} = A^{(k)} \times \left( B^{(k)} \right)^T \in \mathbb{R}^{N \times N},$$  
$$A^{(k)} = \left[ a_i^{(k)}, \ldots, a_N^{(k)} \right] \in \mathbb{R}^{N \times C'}, \quad (3)$$  
$$B^{(k)} = \left[ b_i^{(k)}, \ldots, b_N^{(k)} \right] \in \mathbb{R}^{N \times C'}.$$  
The row $\mathcal{A}_i^{(k)}$ returns the pairwise affinity between region $i$ and every other region $j \in [1, \ldots, N]$ at Graph Attention layer $k$. For each region $i$, we average the affinities in $\mathcal{A}_i^{(k)}$ and concatenate the averaged affinities over $K$ Graph Attention layers to obtain the affinity vector $\alpha_i \in \mathbb{R}^K$.  
2) We feed $\alpha_i$ into a single linear layer to derive the unnormalized attention weights i.e. $f_{att}: \alpha_i \rightarrow \beta_i$, and normalize the region attention weights via softmax, i.e.:  
$$\beta_i = \frac{e^{\beta_i}}{\sum_{j=1}^{N} e^{\beta_j}}, \quad (4)$$

**Training VRAG**  
We train VRAG using the triplet margin loss (Vassileios Baltas and Mikolajczyk 2016). From a triplet of video-level embeddings $(v, v^+, v^-)$ that correspond to the anchor, positive and negative video respectively, we compute the loss:  
$$L = [c(v, v^-) - c(v, v^+) + m]_+, \quad (5)$$
where $c(\cdot, \cdot)$ computes the cosine similarity between embeddings, $m$ refers to the margin. During training, we sample triplets using the triplet mining scheme consistent with (Kordopatis-Zilos et al. 2019).

**Shot Representations**  
In this section, we introduce an intermediate approach utilizing shot representations for video retrieval. We use the shot-boundary algorithm in Section to divide a video $[T^{(1)}, \ldots, T^{(7)}]$ into non-overlapping shots and use VRAG to encode each shot into embeddings. Using shots can reduce noise within VRAG embeddings by removing spurious edges across shot boundaries. In our shot-level approach, we evaluate video similarity by aggregating the pairwise similarities between shot-level embeddings.  

**Shot Boundary Algorithm.** In our shot boundary algorithm, we identify the boundaries between video shots by comparing the similarities between consecutive frames. Specifically, we flatten the R-MAC features from each frame $X(t) \in \mathbb{R}^{R \times C}$ into the frame representation vector $F(t) \in \mathbb{R}^{(R \times C) \times 1}$. The similarities between consecutive frames is then the cosine similarity between their representations vectors $F(t)$ and $F(t-1)$ is lower than the minimum cosine similarity threshold $\tau_s$. Generally, a higher $\tau_s$ creates more shots and resembles frame-level approaches while a smaller $\tau_s$ reduces the number of shots and closely approximates video-level methods.  

**Aggregating Shot Similarities.** Under our shot-level approach, we evaluate the similarity between videos $X$ and $X'$ from the pairwise similarities between their shot embeddings, i.e. $[o_1, \ldots, o_n]$ and $[o'_1, \ldots, o'_m]$. We compute the cosine similarity between pairs of shots and build a pairwise shot cosine similarity matrix $S \in [-1, 1]^{N \times M}$ where $S_{ij}$ returns the cosine similarity between shots $o_i$ and $o_j'$.  
We aggregate these pairwise similarities into a similarity metric between videos. We use two aggregation schemes proposed in (Kordopatis-Zilos et al. 2019): Chamfer similarity (CS) which takes the maximum similarity along the columns of $S$ followed by average pooling of the maximum similarity vector, i.e.  
$$s = CS(X, X') = \frac{1}{N} \sum_{i=1}^{N} \left( \max_{j \in 1, \ldots, M} S_{ij} \right), \quad (6)$$  
Symmetric Chamfer Similarity (SCS) which computes the average Chamfer similarity from $S$ and $S^T$, i.e.:  
$$s = SCS(X, X') = \frac{CS(X, X') + CS(X', X)}{2}. \quad (7)$$

**Experiments**  
We evaluate video retrieval performance using mean Average Precision (Wu, Hauptmann, and Ngo 2007) (mAP). Additionally, we provide qualitative results in our supplementary material.

**Experiment Settings**  
We evaluate VRAG and our shot-level approach over several video retrieval settings: Near-Duplicate Video Retrieval (NDVR); Event Video Retrieval (EVR); and Fine-grained Incident Video Retrieval (FIVR).

**Datasets.** For evaluation, we use CC_WEB_VIDEO (Wu, Hauptmann, and Ngo 2007) for NDVR, EVVE (Revaud et al. 2013) for EVR, and FIVR200K (Kordopatis-Zilos et al. 2019) and its subset FIVR5K (Kordopatis-Zilos et al. 2019; Kordopatis-Zilos et al. 2019) for FIVR. During training, we sample triplets of videos from VCDB (Jiang, Jiang, and Wang 2014).  
CC_WEB_VIDEO (Wu, Hauptmann, and Ngo 2007) contains 13,129 videos with 24 query videos. In this dataset,
near-duplicate videos correspond to videos stored in different formats e.g. .flv; .wmv; and videos with minor content differences e.g. photometric variations like lighting. We evaluate on the original annotations from (Wu, Hauptmann, and Ngo 2007) and the cleaned annotations from (Kordopatis-Zilos et al. 2019).

**EVVE** (Revaud et al. 2013) contains 2,995 videos from Youtube with 620 query videos over 13 events. **FIVR200K** (Kordopatis-Zilos et al. 2019) contains 100 query videos and a total of 225,960 videos from 4,687 Wikipedia events. The dataset groups similar videos into four categories: 1) Near-Duplicate (ND) videos which share all scenes with the query video; 2) Duplicate Scene (DS) videos that share at least one scene with the query video; 3) Complementary Scene (CS) videos that share at least segment with the query video, but from a different viewpoint; 4) Incident Scene (IS) videos that capture the same incident as the query video without sharing segments. From the video categories, the dataset introduces three video retrieval settings: 1) Duplicate Scene Video Retrieval (DSVR) which only considers ND and DS videos as positives; 2) Complementary Scene Video Retrieval (CSR) that extends from DSVR to include CS videos; and 3) Incident Scene Video Retrieval (ISVR) which marks all four categories of videos as similar videos.

**FIVR5K** (Kordopatis-Zilos et al. 2019; Kordopatis-Zilos et al. 2019) is a subset of FIVR200K, with 50 query videos and 5,000 database videos. To create FIVR5K, the authors (Kordopatis-Zilos et al. 2019) select the 50 hardest query videos on the DSVR task from FIVR200K using (Kordopatis-Zilos et al. 2017a) to measure difficulty.

**VCDB** (Jiang, Jiang, and Wang 2014) consists of a core dataset with 528 videos and a background dataset with 100,000 distractor videos. The core dataset videos has a total of 9,236 pairs of partial copies between the 528 videos in the core dataset.

**Implementation Details.** For each video, we sample frames at one second intervals. We extract R-MAC features from each bottleneck layer of ResNet50 (He et al. 2015), i.e. \( L = 4 \) and \( C = 3840 \). We set the number of regions per frame \( R = 9 \), intermediate dimensions \( C' = 512 \), video-level embedding dimensions \( D = 4096 \). We use \( K = 3 \) Graph Attention (Veličković et al. 2018) layers and the ELU (Clevert, Unterthiner, and Hochreiter 2016) non-linearity. We train VRAG on a single GTX 1080 Ti using a batch size of one over 120 epochs. The maximum duration of each video clip in the training triplet is \( W = 64s \), and we sample 1,000 triplets from each triplet pool, giving a total of 2,000 training iterations per epoch. We optimize VRAG using the triplet margin loss with \( m = 0.2 \) and the Adam (Kingma and Ba 2014) optimizer using a fixed learning rate at \( 3 \times 10^{-7} \). During inference, VRAG processes up to 2,000 frames on 11GB of GPU memory, which amounts to over 30 minutes of video.

**Ablation Studies**

**Our VRAG.** At the region-level, we use attention to aggregate features from neighboring regions. We compare our attention aggregation mechanism with alternatives such as max aggregation and average aggregation, i.e. GCN (Kipf and Welling 2017). In Table 1, we show that our choice of region-level attention aggregation improves performance across all FIVR5K settings.

| Method | Region Agg. | DSVR | CSR | ISVR |
|--------|-------------|------|-----|------|
| VRAG   | Max         | 0.520| 0.537| 0.508|
|        | Average     | 0.518| 0.532| 0.506|
|        | Attention   | 0.532| 0.548| 0.518|

Table 1: Comparison on FIVR5K over different choices of region-level aggregation. Attention-weighted summation is used to pool region embeddings for all choices of region-level aggregation.

At the video-level, we pool region embeddings to a fixed size vector \( \bar{r} \) using an attention-weighted summation. We compare our attention pooling with standard methods, i.e. max pooling and average pooling. In Table 2, we demonstrate that our learnable pooling gives better performance over conventional pooling techniques.

| Method | Region Pooling | DSVR | CSR | ISVR |
|--------|----------------|------|-----|------|
| VRAG   | Max            | 0.423| 0.435| 0.415|
|        | Average        | 0.508| 0.515| 0.499|
|        | Attention      | 0.532| 0.548| 0.518|

Table 2: Comparison on FIVR5K over different video-level pooling techniques.

We also compare our choice of self-attention to an alternative with reduced parameters. In Table 3, we observe that using separate key/query parameters allows for more complex relations between regions and significantly improves retrieval performance in FIVR5K.

| Method | Attention | DSVR | CSR | ISVR |
|--------|-----------|------|-----|------|
| VRAG   | \( f^{(k)}_{\text{query}} = f^{(k)}_{\text{key}} \) | 0.493| 0.510| 0.490|
|        | \( f^{(k)}_{\text{query}} \neq f^{(k)}_{\text{key}} \) | 0.532| 0.548| 0.518|

Table 3: Comparison on FIVR5K using different implementations of attention.

Finally, we propose concatenating the region-level embeddings to reduce over-smoothing effects observed in GCNs (Cai and Wang 2020; Chen et al. 2019). We compare against three baselines: 1) Using only the output from the final Graph Attention layer; 2) Concatenating the outputs from all Graph Attention layers; 3) Concatenating the outputs from all Graph Attention layers and \( f_\cdot \). In Table 4, we show that multi-layer concatenation of region-level embeddings greatly improves retrieval performance. Furthermore, we observe a monotonic increase in performance as we concatenate region representations across more layers.

**Shot-level Video Retrieval.** In shot-level video retrieval, i.e. VRAG-S, we compare different choices of \( \tau_s \) for our
Table 4: Comparison on FIVR5K over different levels of depth-wise concatenation.

| Method   | Concat. Layers | DSVR | CSVR | ISVR |
|----------|----------------|------|------|------|
| VRAG     | Final Graph Attn. | 0.452 | 0.463 | 0.442 |
|          | All Graph Attn.   | 0.481 | 0.497 | 0.477 |
|          | All Graph Attn. + \( J_r \) | 0.507 | 0.526 | 0.498 |
|          | All Layers        | 0.532 | 0.548 | 0.518 |

Table 5: Comparison on FIVR5K over different shot-level hyperparameters.

Comparison with Baseline Methods

We compare VRAG and our shot-level approach with existing approaches as baselines across several video retrieval tasks. We evaluate our performance on CC_WEB_VIDEO (Wu, Hauptmann, and Ngo 2007) for NDVR, EVVE (Revaud et al. 2013) for EVR, and use FIVR200K (Kordopatis-Zilos et al. 2019), and FIVR5K (Kordopatis-Zilos et al. 2019; Kordopatis-Zilos et al. 2019), for FIVR.

Video-level Methods. For LBoW (Kordopatis-Zilos et al. 2017a), we use the results from (Kordopatis-Zilos et al. 2019) for CC_WEB_VIDEO and FIVR200K. We re-implement LBoW, i.e. LBoW\(^1\), with 1000 codebooks built on VCDB using KMeans++, for EVVE and FIVR5K. For DML (Kordopatis-Zilos et al. 2017b), we obtain results for CC_WEB_VIDEO and FIVR200K from (Kordopatis-Zilos et al. 2019), and use the released source code\(^1\) for EVVE and FIVR5K, i.e. DML*. For TMK (Poullot et al. 2015) and LAMV (Baraldi et al. 2018) on EVVE, we found discrepancies between their evaluation script and the original script from (Revaud et al. 2013). We provide corrected results, i.e. TMK*, LAMV*, using the original EVVE evaluation script (Revaud et al. 2013). LAMV was not available in the released source code\(^2\), and we use TMK with frequency normalization (0.534 mAP on EVVE) as a close proxy for LAMV (0.536 mAP on EVVE (Baraldi et al. 2018)). We also evaluate TMK* and LAMV* on FIVR5K and FIVR200K. For other video-level methods, we report results from (Kordopatis-Zilos et al. 2019).

Frame-level Methods. On EVVE, the authors of ViSiL (Kordopatis-Zilos et al. 2019) were only able to download, process and evaluate on 80% of the dataset. We provide complete results for ViSiL using the released source code\(^3\), i.e. ViSiL\(_{sym}\) and ViSiL\(_{v}\). For other frame-level methods, we report the updated results using deep network features in (Kordopatis-Zilos et al. 2019).

Near-duplicate Video Retrieval.

In Table 6, we compare our approach with other video-level and frame-level approaches on CC_WEB_VIDEO (Wu, Hauptmann, and Ngo 2007), and demonstrate that our approach achieves state-of-the-art performance for video-level NDVR. Similar to FIVR5K, we observe that our shot-level approach has intermediate performance to frame-level and video-level approaches. We also compare the run times for video-level VRAG and shot-level VRAG with ViSiL\(_{v}\) (Kordopatis-Zilos et al. 2019) after extracting R-MAC features from each video frame. Video-level and shot-level VRAG takes 33mins and 52mins, respectively, while ViSiL uses 109mins to infer video similarities.

Table 6: Results on four different versions of CC_WEB_VIDEO. (*) denotes evaluation on the entire data set and the subscript \(c\) uses cleaned annotations.

Table 8: Comparison on FIVR5K over different shot-level approaches.
### Table 7: Comparison with state-of-the-art EVR approaches on EVVE. We use the same event class ordering as (Revaud et al. 2013). For video-level approaches, we also underline the result with the highest mAP, excluding results with discrepancies in the evaluation script, i.e. LAMV and LAMV+QE. We report results obtained from the original EVVE evaluation script.

| Approach | Method   | mAP | per event class |
|----------|----------|-----|-----------------|
| Video    | LBoW†    | 0.469 | 0.323 0.373 0.062 0.392 0.306 0.232 0.205 0.127 0.060 0.376 0.233 0.769 0.713 |
|          | DML*     | 0.472 | 0.437 0.368 0.052 0.385 0.242 0.275 0.205 0.105 0.085 0.414 0.245 0.783 0.656 |
|          | TMK*     | 0.469 | 0.508 0.306 0.139 0.366 0.294 0.244 0.208 0.125 0.152 0.287 0.213 0.810 0.614 |
|          | LAMV*    | 0.493 | 0.649 0.321 0.157 0.411 0.319 0.241 0.224 0.124 0.194 0.257 0.191 0.857 0.660 |
|          | LAMV+QE* | 0.541 | 0.795 0.413 0.160 0.546 0.376 0.297 0.235 0.124 0.236 0.257 0.185 0.907 0.754 |
|          | LAMV     | 0.536 | 0.715 0.383 0.158 0.461 0.387 0.227 0.247 0.138 0.222 0.273 0.273 0.908 0.691 |
|          | LAMV+QE  | 0.587 | 0.837 0.500 0.126 0.588 0.455 0.343 0.267 0.142 0.230 0.293 0.216 0.950 0.770 |
|          | Ours     | 0.623 | 0.792 0.675 0.072 0.496 0.329 0.292 0.256 0.241 0.497 0.692 0.378 0.928 0.770 |
|          | Ours+QE  | 0.653 | 0.888 0.743 0.042 0.505 0.342 0.304 0.247 0.280 0.489 0.782 0.410 0.943 0.835 |

| Shot     | ViSiL    | 0.589 | 0.889 0.570 0.169 0.432 0.345 0.393 0.297 0.181 0.479 0.564 0.369 0.885 0.799 |
|          | ViSiL_vf | 0.612 | 0.923 0.724 0.301 0.573 0.418 0.276 0.291 0.200 0.544 0.396 0.339 0.938 0.753 |
|          | ViSiL_vf | 0.618 | 0.920 0.713 0.222 0.589 0.350 0.345 0.276 0.169 0.444 0.567 0.375 0.909 0.842 |

| Frame    | Ours     | 0.709 | 0.704 0.656 0.042 0.596 0.406 0.575 10m 35s |
|          | Ours (SCS) | 0.709 | 0.704 0.656 0.042 0.596 0.406 0.575 10m 35s |

### Table 8: Results on FIVR5K

We found that most FIVR200K queries comprise of single shots while their Duplicate Scene (DS) videos are sequences of visually diverse shots. Additionally, most shots in the DS videos have low visual and semantic correspondence with the query segment. Consequently, video-level methods are likely to yield noisy representations that are distinct to the query. On the other hand, higher fidelity approaches, i.e. frame-level and shot-level retrieval, that segment the video into multiple representations can preserve the query video segment and obtain more accurate video similarities. We show examples in the supplementary material.

### Event Video Retrieval

In Table 7, we evaluate our performance on EVVE (Revaud et al. 2013). Without test augmentations, i.e. Query Expansion, VRAG outperforms state-of-the-art video-level and frame-level methods. We apply Average Query Expansion (Douze et al. 2013) on VRAG, i.e. VRAG+QE and it demonstrates further performance gains. Our shot-level approach is competitive with video-level and frame-level approaches and use 75mins of evaluation time. In contrast, video-level VRAG takes approximately 12mins for inference on EVVE, while ViSiL (Kordopatis-Zilos et al. 2019) uses 18hrs. We attribute the difference in efficiency to paradigm differences between video-level and frame-level approaches: Our video-level method requires $620 \times 2375 \approx 2995$ forward passes through VRAG to encode $n$ video embeddings. In contrast, ViSiL directly outputs the similarity between every query-candidate video pair. This requires $620 \times 2375 \approx 1.5 \times 10^6$ forward passes through ViSiL, which is orders of magnitude larger than the number of videos $n = 2995$.

### Conclusion

In this work, we introduce Video Region Attention Graph Network (VRAG) that utilizes self-attention during region aggregation and region pooling to generate video-level embeddings for efficient video retrieval. Specifically, we represent videos at a finer granularity through region-level features and encode video spatio-temporal dynamics through region-level relations. Our VRAG improves the state-of-the-art performance of video-level methods across multiple video retrieval datasets. We also introduce an intermediate approach to video-level and frame-level video retrieval that utilizes shots for video retrieval. We demonstrate that our shot-level approach bridges the gap in performance between video-level and frame-level approaches, where it obtains higher video retrieval performance at marginal increase in computational costs over video-level approaches.
References

Baraldi, L.; Douze, M.; Cucchiara, R.; and Jégou, H. 2018. LAMV: Learning to Align and Match Videos with Kernelized Temporal Layers. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7804–7813.

Cai, C.; and Wang, Y. 2020. A Note on Over-Smoothing for Graph Neural Networks. arXiv:2006.13318.

Chen, D.; Lin, Y.; Li, W.; Li, P.; Zhou, J.; and Sun, X. 2019. Measuring and Relieving the Over-smoothing Problem for Graph Neural Networks from the Topological View. arXiv:1909.03211.

Chou, C.; Chen, H.; and Lee, S. 2015. Pattern-Based Near-Duplicate Video Retrieval and Localization on Web-Scale Videos. *IEEE Transactions on Multimedia*, 17(3): 382–395.

Clevert, D.-A.; Unterthiner, T.; and Hochreiter, S. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *CoRR*, abs/1511.07289.

Douze, M.; Revaud, J.; Schmid, C.; and Jégou, H. 2013. Stable Hyper-pooling and Query Expansion for Event Detection. In *2013 IEEE International Conference on Computer Vision*, 1825–1832.

Fajtl, J.; Sokeh, H. S.; Argyriou, V.; Monekosso, D.; and Remagnino, P. 2018. Summarizing Videos with Attention. *CoRR*, abs/1812.01969.

Gabeur, V.; Sun, C.; Alahari, K.; and Schmid, C. 2020. Multi-modal Transformer for Video Retrieval. In *European Conference on Computer Vision (ECCV)*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Deep Residual Learning for Image Recognition. *CoRR*, abs/1512.03385.

Ji, Z.; Xiong, K.; Pang, Y.; and Li, X. 2017. Video Summarization with Attention-Based Encoder-Decoder Networks. *CoRR*, abs/1708.09545.

Jiang, Y.-G.; Jiang, Y.; and Wang, J. 2014. VCDB: A Large-Scale Database for Partial Copy Detection in Videos. In Fleet, D.; Pajdla, T.; Schiele, B.; and Tuytelaars, T., eds., *Computer Vision – ECCV 2014*, 357–371. Cham: Springer International Publishing. ISBN 978-3-319-10593-2.

Kingma, D.; and Ba, J. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of the 5th International Conference on Learning Representations*, ICLR ’17.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kom­patsiaris, I. 2019. Finding Near-Duplicate Videos in Large-Scale Collections. In Mezaris, V.; Nixon, L. J. B.; Papadopoulos, S.; and Teysou, D., eds., *Video Verification in the Fake News Era*, 91–126. Springer.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, I. 2019. FIVR: Fine-grained Incident Video Retrieval. *IEEE Transactions on Multimedia*.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kom­patsiaris, Y. 2017a. Near-Duplicate Video Retrieval by Aggregating Intermediate CNN Layers. In *MMM*.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kom­patsiaris, Y. 2017b. Near-Duplicate Video Retrieval with Deep Metric Learning. In *2017 IEEE International Conference on Computer Vision Workshop (ICCVW)*.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, Y. 2019. ViSiL: Fine-Grained Spatio-Temporal Video Similarity Learning. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 6350–6359.

Liu, Y.; Albanie, S.; Nagrani, A.; and Zisserman, A. 2019. Use What You Have: Video retrieval using representations from collaborative experts. In *BMVC*.

Miech, A.; Laptev, I.; and Sivic, J. 2018. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. *CoRR*, abs/1804.02516.

Miech, A.; Zhukov, D.; Alayrac, J.; Tapaswi, M.; Laptev, I.; and Sivic, J. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2630–2640.

Poullot, S.; Tsukatani, S.; Phuong Nguyen, A.; Jégou, H.; and Satoh, S. 2015. Temporal Matching Kernel with Explicit Feature Maps. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM ’15, 381–390. New York, NY, USA: Association for Computing Machinery. ISBN 9781450334594.

Revaud, J.; Douze, M.; Schmid, C.; and Jégou, H. 2013. Event Retrieval in Large Video Collections with Circulant Temporal Encoding. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2459–2466.

Tan, H.-K.; Ngo, C.-W.; Hong, R.; and Chua, T.-S. 2009. Scalable Detection of Partial Near-Duplicate Videos by Visual-Temporal Consistency. In *Proceedings of the 17th ACM International Conference on Multimedia*, MM ’09, 145–154. New York, NY, USA: Association for Computing Machinery. ISBN 9781605586083.

Tolias, G.; Sicre, R.; and Jégou, H. 2016. Particular object retrieval with integral max-pooling of CNN activations. *CoRR*, abs/1511.05879.

Vassileios Balntas, D. P.; Edgar Riba; and Nikolajczyk, K. 2016. Learning local feature descriptors with triplets and shallow convolutional neural networks. In Richard C. Wilson, E. R. H.; and Smith, W. A. P., eds., *Proceedings of the British Machine Vision Conference (BMVC)*, 119.1–119.11. BMVA Press. ISBN 1-901725-59-6.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. *CoRR*, abs/1706.03762.

Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; and Bengio, Y. 2018. Graph Attention Networks. arXiv:1710.10903.

Wu, X.; Hauptmann, A. G.; and Ngo, C.-W. 2007. Practical Elimination of Near-Duplicates from Web Video Search. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM ’07, 218–227. New York, NY, USA: Association for Computing Machinery. ISBN 9781595937025.
Zhou, K.; Qiao, Y.; and Xiang, T. 2018. Deep Reinforcement Learning for Unsupervised Video Summarization with Diversity-Representativeness Reward. arXiv:1801.00054.
Supplementary Material for
VRAG: Region Attention Graphs for Content-Based Video Retrieval

Qualitative Results

T-SNE Embeddings

In this section, we visualize our video-level representations into T-SNE embeddings for EVVE (Revaud et al. 2013) in Figure 1 and FIVR5K (Kordopatis-Zilos et al. 2019; Kordopatis-Zilos et al. 2019) in Figure 2.

![Figure 1: T-SNE embedding visualizations on EVVE. (Left) DML (Kordopatis-Zilos et al. 2017). (Right) Our VRAG.](image1)

![Figure 2: T-SNE embedding visualizations on FIVR-5K. (Left) DML (Kordopatis-Zilos et al. 2017). (Right) Our VRAG.](image2)

Across both datasets, we observe that our VRAG forms more visible and well-separated T-SNE embedding clusters compared to the existing video-level method DML (Kordopatis-Zilos et al. 2017).

Retrieval Results

Additionally, we include examples of retrieval results on the EVVE (Revaud et al. 2013) dataset. In our retrieval examples, we visualize the five closest database videos to the query videos. We demonstrate positive results in Figures 3 to 6. From Figures 4 and 6, we see that our VRAG can retrieve videos of similar events taken from different viewpoints. In Figures 3 and 5, VRAG retrieves videos of the same event taken from different points in time.

Figures 7 to 10 demonstrate some failure cases of our method with negatively-related videos marked in red borders. In Figures 7, 9 and 10, we see that our VRAG pulls negative videos that are semantically related to the query video. Specifically, Figure 7 includes a negative video from a different geyser, Figure 9 has a negative video from another riot, and Figure 6 shows a negative video from a separate live performance. However, our VRAG may confuse videos in instances where distractor videos contain similar elements to query videos as positives. For example in Figure 8, VRAG retrieves a negative video depicting a crowded sports stadium as a close positive to a query video of a protest with multiple scenes of crowds.

References

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatziaris, I. 2019. FIVR: Fine-grained Incident Video Retrieval. IEEE Transactions on Multimedia.

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, Y. 2017. Near-Duplicate Video Retrieval with Deep Metric Learning. In 2017 IEEE International Conference on Computer Vision Workshop (ICCVW).

Kordopatis-Zilos, G.; Papadopoulos, S.; Patras, I.; and Kompatsiaris, Y. 2019. ViSiL: Fine-Grained Spatio-Temporal Video Similarity Learning. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 6350–6359.

Revaud, J.; Douze, M.; Schmid, C.; and Jégou, H. 2013. Event Retrieval in Large Video Collections with Circulant Temporal Encoding. In 2013 IEEE Conference on Computer Vision and Pattern Recognition, 2459–2466.
Figure 3: Positive geyser event retrieval example.

Figure 4: Positive flooding event retrieval example.
Figure 5: Positive water park event retrieval example.

Figure 6: Positive live performance retrieval example.
Figure 7: Negative geyser event retrieval example. Negative videos have red borders.

Figure 8: Negative protests event retrieval example. Negative videos have red borders.
Figure 9: Negative riots event retrieval example. Negative videos have red borders.

Figure 10: Negative live performance retrieval example. Negative videos have red borders.