Self-Supervised Video Similarity Learning

Giorgos Kordopatis-Zilos*[1] Giorgos Tolias[1] Christos Tzelepis[2] Ioannis Kompatsiaris[3]
Ioannis Patras[2] Symeon Papadopoulos[3]

[1]VRG, FEE, Czech Technical University in Prague
[2]Queen Mary University of London
[3]Information Technologies Institute, CERTH

{kordogeo,toliageo}@fel.cvut.cz {c.tzelepis,i.patras}@qmul.ac.uk {ikom,papadop}@iti.gr

Abstract

We introduce S^{2}VS, a video similarity learning approach with self-supervision. Self-Supervised Learning (SSL) is typically used to train deep models on a proxy task so as to have strong transferability on target tasks after fine-tuning. Here, in contrast to prior work, SSL is used to perform video similarity learning and address multiple retrieval and detection tasks at once with no use of labeled data. This is achieved by learning via instance-discrimination with task-tailored augmentations and the widely used InfoNCE loss together with an additional loss operating jointly on self-similarity and hard-negative similarity. We benchmark our method on tasks where video relevance is defined with varying granularity, ranging from video copies to videos depicting the same incident or event. We learn a single universal model that achieves state-of-the-art performance on all tasks, surpassing previously proposed methods that use labeled data. The code and pretrained models are publicly available at: https://github.com/gkordo/s2vs

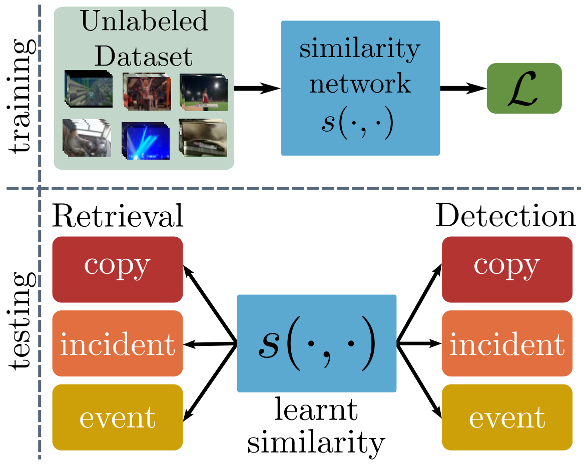

Figure 1. A video similarity network is trained with SSL to compare two videos. The resulting model is used, without any further training, for retrieval and detection of relevant videos in different tasks, where the definition of relevance ranges from video *copies* to videos capturing the same *incidents* and *events*.

1. Introduction

Self-supervised learning is a popular approach, especially for learning representations that are amenable to transfer to different tasks [9, 10, 24, 27, 61]. SSL allows to scale-up the dataset size by not relying on manual labeling and is known to obtain representations with high transferability. The commonly studied setup is to consider SSL for pre-training on a proxy task and then perform supervised fine-tuning on different target tasks [9, 10, 27]. In this work, we rather perform SSL and directly use the model on video similarity-related tasks.

Computing similarity between videos is a common objective across a number of video retrieval [43, 63, 78] and video detection [38, 46] problems. The definition of what is a relevant video to retrieve or detect may differ according to the task at hand. In this work, three cases are considered: i) video copies [38, 78], *i.e.*, edited versions of the same source video, ii) videos of the same incident [43], *i.e.*, videos capturing the same spatio-temporal span, and iii) videos of the same event [63], *i.e.*, videos capturing the same spatial or temporal span. In this work, we target both retrieval and detection. In the former, only ranking per query matters; therefore, the distribution of similarities can vary among queries. While in the latter, the ability to apply a similarity threshold and detect relevant videos matters.

*Research partially conducted at ITI-CERTH.

Supervised learning of specialised models per task is very demanding in terms of training data collection, especially in the video domain. Instead, in the proposed method, we are learning a single model via SSL to perform all retrieval and detection tasks (see Figure 1) without further fine-tuning. We inject self-supervision into video similarity learning by adopting the concept of instance-discrimination [10], where each video forms its own class, and any transformation of it preserves the class label.

In this work, we adopt the ViSiL [44] architecture for video similarity, which needs labeled video datasets for its development in prior works [44, 45], but we train it in a self-supervised way and argue that instance-discrimination through augmentations is well suited for all the aforementioned tasks. To pronounce the synergy, we develop an appropriate composition of video augmentations and propose a model-tailored loss combined with a standard SSL loss. By eliminating the need for video annotations, we are able to train on large video datasets and achieve state-of-the-art results on all target retrieval and detection tasks. Evaluation is performed on three standard benchmarks, namely, VCDB [38], FIVR [43], and EVVE [63].

In summary, our contributions include the following:

- We perform SSL via instance-discrimination for video similarity estimation and surpass existing results, obtained with fully supervised training, on three different retrieval and detection tasks.
- The performance of the InfoNCE loss [57] is improved by a proposed loss that acts jointly on self-similarity and hard-negative similarity of each video in the batch.
- We are the first to jointly benchmark retrieval and detection performance on a range of video-relevance granularities. Additionally, we repurpose the FIVR dataset, whose performance has almost reached saturation, and evaluate only on hard examples.

2. Related Work

Video similarity and self-supervised learning are the two research fields that are most relevant to our work.

2.1. Video similarity

Video similarity methods can be roughly classified into two general categories, *i.e.*, *global representation* and *matching* approaches.

Global representation approaches first design or learn a mapping of input examples to a vector space and then use standard distance metrics or similarity measures to compare pairs of examples. These methods reduce down to representation learning, typically called global representation or descriptor, in the sense that the input example is represented by a single vector. Early methods extract hand-crafted features [34, 56] from all video frames and use aggregation schemes, *e.g.*, mean pooling [35, 78], Bag-of-Words [6, 65, 67], to generate global video vectors. More recent approaches use deep features combined with learnable aggregation methods, *i.e.*, using unsupervised schemes [22, 41, 54] or training deep supervised models with metric learning [42, 47, 48]. In addition, several methods extract hash codes for the entire video and measure similarity in the Hamming space [68]. The latter typically train deep networks, such as LSTMs [33, 69, 80] or Transformers [49, 50, 74], with self-supervised schemes that optimize for the preservation of the video adjacencies from the initial feature space to the Hamming one.

Matching approaches represent videos with more than a single vector and involve elaborate similarity estimation schemes, leverage spatio-temporal representations, and exploit video alignment or fine-grained similarity functions. Early methods propose handcrafted solutions to assess similarity through video alignment using Temporal Networks [70, 76], temporal Hough Voting [18, 39], or Dynamic Programming [12]. Other methods build on the foundations of representation learning to generate spatio-temporal representations with transformer-based networks for temporal aggregation [31, 66], multi-attention networks [75], attention-based RNN architectures [5, 20] or Fourier-based representations [4, 60]. Recent work focuses on video similarity networks that design and learn matching functions to estimate the video-to-video similarity [25, 30, 37, 44, 45]. The matching function is parametric and learnable in this case. ViSiL [44] is among the first methods in this direction; it performs fully-supervised training of a video similarity network to capture fine-grained spatial and temporal structures. Also, Distill-and-Select (DnS) [45] leverages knowledge distillation to train students using ViSiL as the teacher network. In our work, we adopt the DnS variant of ViSiL and train it with self-supervised learning.

2.2. Self-supervised learning

SSL recently witnessed rapid growth and is leveraged in several vision-related problems. Many examples exist in the image domain for the training of representation models via solving explicit proxy tasks [16, 17, 55, 58, 82], discriminating instances through contrastive learning [10, 27, 32, 71], optimizing clustering and representation [2, 7, 8], bootstrapping knowledge with self-distillation [9, 11, 23] or image reconstruction with masked autoencoders [3, 26].

The video domain offers additional avenues for self-supervision, *e.g.*, by exploiting spatio-temporal information, such as frame ordering [21, 53], motion [1, 36], multimodal co-training [24], temporal field of view [62] or, more recently, video masking autoencoders [73, 77].The roadmap of augmentations designed for videos has been adopted by some approaches [24, 40, 61] that train a video representations network on a proxy task. Qian *et al.* [61] use temporally consistent spatial augmentation and contrastive learn-

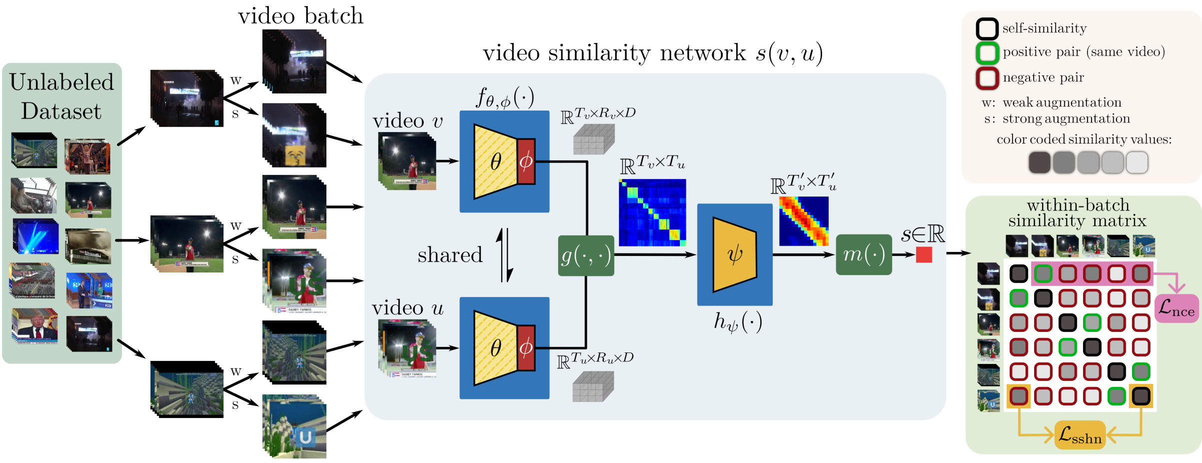

Figure 2. Overview of the proposed approach. Each video in a random batch is augmented twice, and all video pairs are processed by the video similarity network (includes feature representation, spatial matching, and a learnable temporal matching component) to estimate the video-to-video similarity values. Two losses are applied per row of the within-batch similarity matrix: the widely known InfoNCE loss ($\mathcal{L}_{\text{nce}}$) and the newly proposed $\mathcal{L}_{\text{sshn}}$ that maximizes self-similarity and minimizes the hardest negative similarity.

ing. These methods learn a mapping of videos to a vector space, and, most of the time, the goal is to perform fine-tuning on other tasks with good generalization.

In another line of research, when video-to-video similarity estimation is the objective, mapping to a vector space is not the most suitable choice; instead, a matching function for a video pair is typically the preferred choice. In contrast to this work, where we learn a parametric matching function, prior work uses hand-crafted matching and only learns the representation [4, 31]. This is the case for near-duplicate video retrieval [31] and video matching through alignment [4], where self-supervision comes in the form of pre-generated static training datasets through spatio-temporal augmentations. In the same way, He *et al.* [25] target video copy localization and, through self-supervision, generate ground truth masks at the level of frame-to-frame correspondences. In contrast to them, we optimize a more general video similarity model and effectively employ it to tackle multiple retrieval and detection tasks.

Lastly, a related work [59] in the image domain proposes a self-supervised method reflecting the objectives of the target task based on task-specific augmentations. Their method relies on contrastive image representation learning using advanced augmentations, *e.g.*, text and emoji overlays, strong blurring, and CutMix [81], and an adopted InfoNCE loss [57]. We use similar task-specific augmentations and losses in our work for videos, instead of images.

3. SSL for Video Similarity

Our aim is to learn a video similarity function $s\colon \mathcal{V} \times \mathcal{V} \to \mathbb{R}$, where $\mathcal{V}$ is the space of all videos. The goal is for two videos to have high similarity if they are relevant, and low otherwise. The definition of relevance is task-dependent. In our experiments, we consider several evaluation tasks, where relevance ranges from video copies to videos of the same physical event. Nevertheless, we perform training in a single universal way without video labels for supervision. We perform training with self-supervision in the spirit of instance-discrimination, *i.e.*, two augmented videos originating from the same original video are considered as positive to each other, or negative otherwise. In some parts, we follow the work of Pizzi *et al.* [59], who perform SSL for image copy detection. The overview of the proposed approach is illustrated in Figure 2.

3.1. Similarity network

We adopt the ViSiL variant proposed in DnS [45], namely the fine-grained attention student, as our similarity network architecture. It consists of a representation network, a hand-crafted spatial matching function, a learnable temporal matching function, and a final hand-crafted matching function that estimates the final video-level similarity.

The representation network $f_{\theta,\phi}\colon \mathcal{V} \to \mathbb{R}^{T \times R \times D}$ maps an input video to a D-dimensional vector per region, for R regions per frame, for T frames, where R and T vary according to the frames' size and video length, respectively. This network consists of a pre-trained backbone network and has a parameter set θ that is fixed in this work, similar to the prior ones [4, 37, 43]. The learnable part corresponds to the parameter set ϕ, a dot-attention scheme [79] that is applied to weigh region vectors based on their saliency.

Given two input videos v and u and their corresponding representations, the hand-crafted spatial matching is performed by the function $g\colon \mathbb{R}^{T_v \times R_v \times D} \times \mathbb{R}^{T_u \times R_u \times D} \to \mathbb{R}^{T_v \times T_u}$, that takes as input two video representations and

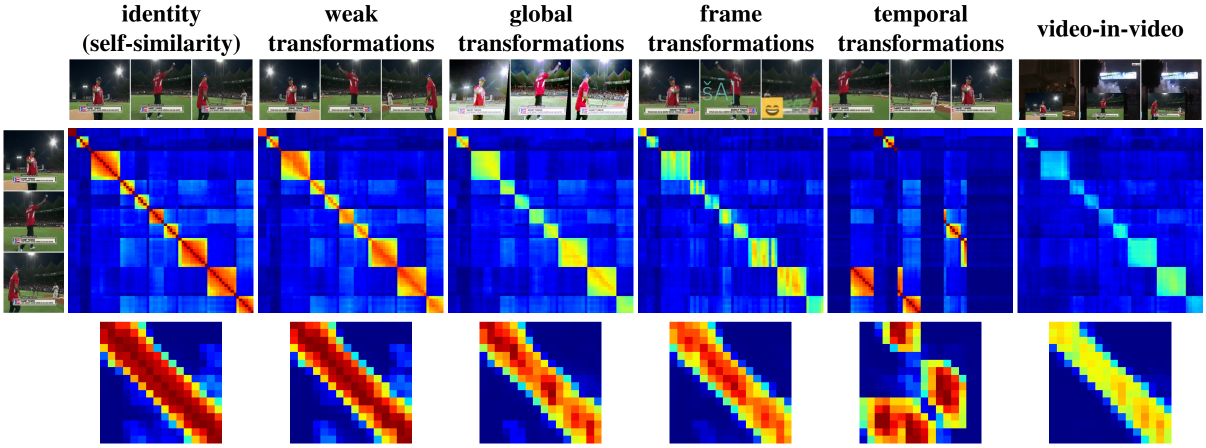

Figure 3. Temporal similarity matrices formed between the original video and its strongly augmented counterpart. Sampled frames (top), similarity matrices that are the input to (middle), and output of (bottom) the temporal matching function of the proposed S^{2}VS are shown. Only one type of augmentation is applied in each case. All matrices are scaled in [0,1] with blue (red), indicating similarity close to 0 (1).

estimates the temporal similarity matrix. It computes the $R_v \times R_u$ spatial similarity matrix for all frame pairs and then applies Chamfer similarity on each of them to estimate the frame-to-frame similarity.

The temporal matching is performed by function $h_\psi : \mathbb{R}^{T_v \times T_u} \rightarrow \mathbb{R}^{T'_v \times T'_u}$. This is a four-layer CNN that learns to capture temporal patterns in the input similarity matrices. It outputs a filtered temporal similarity matrix. It holds that $T_v = 4T'_v$, and similarly for u, due to the CNN design that contains strided max pooling operations. The parameters of the CNN, denoted by ψ, are learnable.

Chamfer similarity is applied and denoted by the function $m\colon \mathbb{R}^{T'_v \times T'_u} \rightarrow \mathbb{R}$, taking as input the filtered temporal similarity matrix and estimating the final video-level similarity, *i.e*., the scalar similarity between the two videos.

To summarize, similarity $s(v, u)$, for the video pair consisting of videos v and u, is equivalent to $s(v, u) = m\left(h_\psi\left(g(f_{\theta,\phi}(v), f_{\theta,\phi}(u))\right)\right)$, and the goal in this work is to learn ϕ and ψ with self-supervision on videos, while θ remains fixed and is obtained from supervised pre-training on ImageNet. The reader is referred to the original ViSiL work [44] for additional details.

3.2. Weak/strong video augmentations

We apply two sets of augmentations to generate two corresponding versions of a training video, *i.e*., one weakly and one strongly augmented version. Formally, given an original video v, the output of an augmentation function A is a video tensor $\tilde{v} = A(v) \in \mathbb{R}^{T_B \times H_B \times W_B \times 3}$, where T_B, H_B, and W_B correspond to the number of frames, height, and width of the video in the batch, respectively.

Weak augmentations consist of conventional geometric transformations (i.e., resized crop and horizontal flip), applied globally on the entire video, and temporal cropping to select T_B consecutive frames.

Strong augmentations consist of the weak augmentations and several other transformations grouped into the following four categories:

Global transformations are frame transformations applied to all frames in a consistent way. We use RandAugment [14], an automatic augmentation strategy that includes different geometric and photometric image transformations and requires two hyperparameters, namely N_{RAug} and M_{RAug}. These correspond to the number of randomly-applied consecutive transformations and their magnitude value that determines their severity, respectively.

Frame transformations are applied independently per frame. We use overlay and blurring transformation[1]. Following advanced augmentations from prior work [59], we add random emojis and text, each with probability $p_{overlay}$, and blur frames with probability p_{blur}. We opt for these operations to emulate common video copy transformations.

Temporal transformations act only on the temporal dimension and include five operations, with one applied per video. Following [44], we use fast forward, slow motion, reverse play, and frame pause, where a single frame is duplicated several times consecutively. In addition, we propose Temporal Shuffle-Dropout (TSD) to alter the global temporal structure but preserve the local one. The video is first split into short clips, each of them with length randomly chosen in $[4, \ldots, T_B/2]$. In the shuffling phase, applied with probability p_{shuf}, the clip order is shuffled. In the dropout phase, a clip is dropped with probability p_{drop}, where it is either discarded or filled with empty frames or Gaussian noise with probability p_{cont}.

[1]The RandAugment implementation we use does not contain blurring operations. Hence, global transformations do not blur videos.

Video-in-video randomly mixes two strongly augmented videos, the *host* and the *donor*, in the same batch. The donor video is randomly spatially down-sampled with a factor λ_{viv} and is overlaid in a random location within the host video. Each strongly augmented video is chosen as donor with probability p_{viv}. Then, a host video is randomly chosen, while the mixed output replaces the donor video. This process requires properly adjusting the instance-discrimination labels since the generated video is the outcome of two others. Video-in-video transformation is very common in real-life video cases.

Figure 3 presents the impact of each type of augmentation on the temporal similarity matrices. *Self-similarity*, *i.e.*, identity augmentation, is shown as a reference and hints about the temporal structure of the video. *Weak* augmentation only slightly affects, while *global* and *frame* augmentations noticeably affect the strength on the block diagonal structures. Such a structure is preserved with global transformations but not so much with the frame ones; observe some blue vertical lines indicating a significant impact on the frame representation. Nevertheless, the trained network handles both cases robustly, assigning large similarity values on the diagonal part, as seen on the filtered matrices. The *temporal* transformations significantly alter the global structure but partly preserve the local one, while the video-in-video transformation has a substantial impact on the intensity of the main diagonal, highlighting its challenging aspect; yet, the trained network effectively learns to handle such cases.

3.3. Loss on video similarity

A random set of N videos, where each video is augmented once with the weak and once with the strong augmentations, forms a training batch of size $B = 2N$ denoted by $\mathcal{B} = [v_1, \cdots, v_{2N}]$. We compute the similarity matrix $S \in [0, 1]^{B \times B}$, with elements $S_{i,j} = s(v_i, v_j)$, comprising all pairwise video similarities within the batch. Each row of S consists of the self-similarity on the diagonal, one positive-pair similarity, and $B - 2$ negative-pair similarities[2]. Note that S is not symmetric and that the diagonal elements are not equal to 1 because of h_ψ. For the i-th row of the similarity matrix, let $p(i)$ be the set of column indices of the positive pairs. Additionally for the i-th row, let $n(i)$ be the set of column indices of the negative pairs.

The total loss is a combination of two losses that optimize different parts of S: (i) the widely used InfoNCE [57] loss estimated per row excluding the self-similarity value, and (ii) a loss that maximizes the self-similarity, *i.e.*, main diagonal, and minimizes the similarity with the hardest negative, *i.e.*, the negative with the highest similarity, for each video in the batch.

[2] This is the case where video-in-video augmentation is not used; otherwise, there can be more (less) positives (negatives).

InfoNCE loss is estimated for each positive pair by

$$\mathcal{L}_{\text{nce}}(i,j) = -\log \frac{\exp(S_{i,j}/\tau)}{\exp(S_{i,j}/\tau) + \sum_{k \notin p(i) \cup i} \exp(S_{i,k}/\tau)}, \tag{1}$$

where τ is a temperature hyper-parameter and (i, j) is a positive pair. The final InfoNCE loss is given by the average over all positive pairs as

$$\mathcal{L}_{\text{nce}} = {}^1\!/_P \sum_i \sum_{j \in p(i)} \mathcal{L}_{\text{nce}}(i,j), \tag{2}$$

where P is the total number of positive pairs in the batch.

Self-similarity – hardest negative loss: Since the self-similarity is not equal to 1 by design, we add a loss term that is trying to push it to high values. Together with that, an additional term pushes the hardest negative of each row to have small similarity. For the i-th row, this loss is given by

$$\mathcal{L}_{\text{sshn}}(i) = -\underbrace{\log(S_{i,i})}_{self-sim} \underbrace{- \log \max_{j \notin p(i) \cup i} (1 - S_{i,j})}_{hard-negative\ sim}, \tag{3}$$

and the total loss is given by the average over rows as $\mathcal{L}_{\text{sshn}} = {}^1\!/_B \sum_i \mathcal{L}_{\text{sshn}}(i)$. Note that the hard-negative term resembles entropy maximization through the Kozachenko-Leononenko estimator and a consequent spreading of elements in the representation space [64]. Differently to them, we perform this directly on pairwise similarities and not on distances over a vector space.

To this end, we optimize a weighted sum of the losses presented above, as follows

$$\mathcal{L} = \mathcal{L}_{\text{nce}} + \lambda \mathcal{L}_{\text{sshn}}, \tag{4}$$

where λ is a hyperparameter that tunes the impact of $\mathcal{L}_{\text{sshn}}$.

4. Evaluation setup

Here, we present the training/evaluation datasets, the evaluation metrics, and some implementation details.

4.1. Datasets

DnS-100K [45] consists of 115,792 unlabeled videos. It is used for knowledge distillation in the original work, but we use it as a training set.

VCSL [29] is originally created for video copy localization. It contains 9,207 videos with more than 281K copied segments split into training, validation, and test set. Due to the unavailability of several videos, we managed to collect only 8,384 videos. We use this dataset to train our model in a supervised way, only to provide an indicative comparison with the proposed SSL approach.

VCDB [38] is created for partial video copy detection. The core dataset ($\mathcal{C}$) contains 528 videos from 28 discrete sets with over 9,000 copied segments. It also contains a set $\mathcal{D}$ of 100,000 distractor videos. We use this dataset for evaluation for detection and retrieval of video copies, considering as related the videos that share at least one copied segment. Moreover, we use the distractor set as an alternative unlabeled training set. We use VCDB, VCDB ($\mathcal{D}$), or VCDB ($\mathcal{C}$+$\mathcal{D}$) to indicate that only set $\mathcal{C}$, only set $\mathcal{D}$, or both sets are used, respectively.

FIVR-200K [43] is used as a benchmark for fine-grained incident video retrieval. It consists of 225,960 videos and 100 queries. FIVR-200K includes three different subtasks: a) Duplicate Scene Video Retrieval (DSVR), b) Complementary Scene Video Retrieval (CSVR), and c) Incident Scene Video Retrieval (ISVR). In this work, we use the same subsets to evaluate for the corresponding detection tasks, denoted by DSVD, CSVD, and ISVD. For quick comparisons, we also use **FIVR-5K** [44], a subset of FIVR-200K. We use it in our ablations, denoted by **FIVR**, where the average performance of the three subtasks is reported.

EVVE [63] is a dataset for video retrieval. It consists of 620 queries and 2,375 database videos. Due to the unavailability of several videos, we use only 504 queries and 1906 database videos [44], which is roughly ≈80% of the initial dataset. All reported methods are evaluated on this subset.

In summary, we train on DnS-100K, or VCDB($\mathcal{D}$), and evaluate on VCDB for video copies, on FIVR for video copies, and incidents, and on EVVE for video copies, incidents, and events.

4.2. Evaluation metrics

To evaluate methods for retrieval, we use mean Average Precision (mAP). AP is equivalent to the area under the precision-recall curve for a particular query, and mAP is obtained by simply averaging over all queries. To evaluate for detection, we use micro Average Precision (μAP) as a good indicator of detection performance also used in prior work [19, 46, 59]. This is equivalent to the area under the precision-recall curve for all queries jointly. The lists of similarities from all queries are merged, and the labels defining relevant/non-relevant videos (despite being defined with respect to different queries) are used to estimate precision and recall. All the metrics are re-scaled to the $[0, 100]$ range. For retrieval and mAP, only the ranking per query matters; therefore, the distribution of similarities can vary a lot among queries. This is not the case for detection and μAP, where the ranking among all queries jointly matters, reflecting the ability to apply a threshold and detect the relevant items.

4.3. Implementation details

Pre-trained backbone networks (θ): To implement our representation network $f_{\theta,\phi}$, we follow the literature [44, 45, 66] and employ a ResNet50 [28] network pre-trained on ImageNet [15]. We also extract region vectors applying regional max activation pooling [72] on intermediate layers [41], whitened through a PCA-whitening layer [13] learned from 1M region vectors sampled from the VCDB [38] dataset.

Training process: We train our network for 30K iterations with a batch size of 64. We employ AdamW [52] optimization with learning rate $5 \cdot 10^{-5}$ and weight decay 0.01. We use cosine learning rate decay with 1K iterations warm-up [51]. Other parameters are set to $T_B = 32$, $\tau = 0.03$, and $\lambda = 3$. Also, following the original ViSiL work [44], the similarity regularization is employed with a factor $r = 1$. The similarity network generates scores in $[-1, 1]$, and we rescale them to $[0, 1]$ for the loss calculation. For further implementation details and the complete list of hyperparameter values, we point readers to the supplementary material.

5. Experiments

We evaluate the performance of the proposed approach on different retrieval and detection tasks related to video similarity, compare its performance to the state-of-the-art methods, and conduct an ablation study.

5.1. Comparison with the state-of-the-art

We compare the proposed S^{2}VS method with the following approaches. **DML** [42] extracts a video embedding based on a network trained with supervised deep metric learning. **LAMV** [4] trains a video representation using a generated dataset while relying on kernel-based temporal alignment. **TCA**$_f$ [66] is a transformer-based architecture trained with supervised contrastive learning. **VRL** [31] is a CNN and transformer-based network trained end-to-end with no labeled data. **ViSiL**$_f$ [44] is a baseline without any training on videos that corresponds to the frame-to-frame similarity part of ViSiL combined with Chamfer similarity. **ViSiL**$_v$ is the full similarity model trained with supervision. **DnS** [45] is a ViSiL-based student network trained with distillation from a teacher trained with supervision; we compare with the best-performing fine-grained attention student $\mathbf{S}^f_{\mathcal{A}}$. For TCA and VRL, the reported results are taken from the original papers. For the remaining approaches, we run the provided pretrained networks, and following **DnS** [45], we implement LAMV and DML with the same features provided in the official repository[3].

Table 1 presents the performance comparison on video retrieval and detection. S^{2}VS is among the top two performing methods in all cases despite not requiring labels. This holds for both training sets used for our method. The best-performing competitor is DnS, which requires a man-

[3]https://github.com/mever-team/distill-and-select

Approach	Lab.	Trainset	Retrieval VCDB ($\mathcal{C}$+$\mathcal{D}$)	Retrieval FIVR-200K DSVR	Retrieval FIVR-200K CSVR	Retrieval FIVR-200K ISVR	Retrieval EVVE	Detection VCDB ($\mathcal{C}$+$\mathcal{D}$)	Detection FIVR-200K DSVD	Detection FIVR-200K CSVD	Detection FIVR-200K ISVD	Detection EVVE
DML [42]	✓	VCDB ($\mathcal{C}$+$\mathcal{D}$)	-	52.8	51.4	44.0	61.1	-	39.0	36.5	30.0	75.5
LAMV [4]	✗	YFCC100M	78.6	61.9	58.7	47.9	62.0	62.0	55.4	50.0	38.8	<u>80.6</u>
TCA$_f$ [66]	✓	VCDB ($\mathcal{C}$+$\mathcal{D}$)	-	87.7	83.0	70.3	-	-	-	-	-	-
VRL$_f$ [31]	✗	internal	-	90.0	85.8	70.9	-	-	-	-	-	-
ViSiL$_f$ [44]	✗	-	82.0	89.0	84.8	72.1	62.7	40.9	66.9	59.5	45.9	74.6
ViSiL$_v$ [44]	✓	VCDB ($\mathcal{C}$+$\mathcal{D}$)	-	89.9	85.4	72.3	65.8	-	75.8	69.0	53.0	79.1
DnS [45]	✓	DnS-100K	**87.9**	92.1	87.5	<u>74.1</u>	65.1	**74.0**	79.7	69.5	54.2	74.3
S^{2}VS (Ours)	✗	VCDB ($\mathcal{D}$)	-	**92.7**	**87.9**	**74.6**	**67.2**	-	<u>85.7</u>	<u>76.9</u>	<u>62.8</u>	**80.7**
S^{2}VS (Ours)	✗	DnS-100K	**87.9**	<u>92.5</u>	<u>87.8</u>	73.9	<u>65.9</u>	<u>73.0</u>	**89.3**	**80.2**	**64.9**	78.9

Table 1. State-of-the-art comparison via retrieval mAP (%) and detection μAP (%) on three evaluation datasets. **Bold** and <u>underline</u> indicate the best and second best approach, respectively. Missing values are either due to unavailability or unfair comparison due to leak of evaluation data during training.

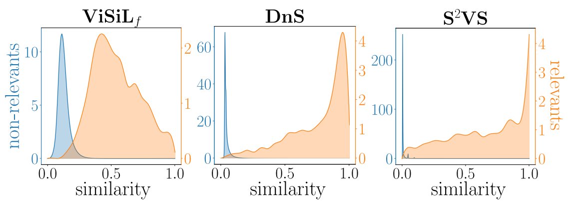

Figure 4. Similarity distribution of S^{2}VS and ViSiL$_f$ and DnS competitors on the DSVD set of FIVR-200K. For S^{2}VS and DnS the similarities are rescaled to $[0, 1]$.

Appr.	Retrieval DSVR	Retrieval CSVR	Retrieval ISVR	Detection DSVD	Detection CSVD	Detection ISVD
ViSiL$_f$ [44]	$52.3_{-36.6}$	$49.4_{-35.4}$	$43.3_{-28.9}$	$16.0_{-50.9}$	$14.0_{-45.5}$	$11.1_{-34.8}$
DnS [45]	$63.1_{-29.0}$	$58.8_{-28.7}$	$\mathbf{48.4}_{-25.7}$	$34.5_{-45.2}$	$25.3_{-44.2}$	$19.6_{-34.6}$
S^{2}VS (Ours)	$\mathbf{64.4}_{-28.1}$	$\mathbf{60.0}_{-27.8}$	$47.1_{-26.8}$	$\mathbf{52.5}_{-36.7}$	$\mathbf{42.6}_{-37.7}$	$\mathbf{35.0}_{-29.9}$

Table 2. Retrieval mAP (%) and detection μAP (%) comparison on FIVR-200K$^{\mathcal{H}}$, which is harder than the original FIVR-200K due to easy video removal. **Bold** indicates the best approach. Red subscripts indicate the performance drop in comparison to the original FIVR-200K.

ually labeled dataset of several thousand copied video segments to train a teacher network. Compared to the baseline ViSiL$_f$, S^{2}VS consistently improves performance across all cases by a noticeable margin. The performance improvement is larger for detection due to better similarity calibration across queries, which is demonstrated in Figure 4. S^{2}VS achieves the best separation between relevant and non-relevant samples, compared with ViSiL$_f$ and DnS, with the two distributions not significantly overlapping.

In addition, we evaluate our proposed approach on the CC_WEB_VIDEO [78] dataset, but we do not provide detailed results since the achieved performance is saturated. The mAP performance is 98.6% / 97.5% / 99.6% / 99.5% when trained with DnS-100K for the different versions as listed in DnS paper [45]. They are on par or slightly better than the competing methods.

Making FIVR harder: FIVR-200K contains a large number of easy examples that dominate the estimation of average performance, giving the impression that the task is nearly solved due to the very high and almost saturated performance. To mitigate this, we discard such easy examples from the dataset to make it more challenging. In particular, we remove all database videos that, as duplicate to a query, are ranked in a position of perfect precision, *i.e.*, before all negatives, by ViSiL$_f$, DNS, and S^{2}VS. This process results in 4,828 videos identified as easy database examples, which is almost 40% of the total videos labeled as relevant to any query. We denote the harder version as FIVR-200K$^{\mathcal{H}}$.

Table 2 presents results for the harder version and the differences compared to the original version in Table 1. Compared with the initial results, the performance significantly drops, with the difference ranging from ≈25% up to 36% of mAP for retrieval and ≈30% up to 50% of μAP for detection. Our approach achieves very similar performance to DnS in the three retrieval tasks with a difference of about 1.3-1.8%, but with a substantial difference of up to 12% from the baseline. The performance gap is much larger in the case of detection. The proposed method surpasses the other two by a clear margin of more than 15% from the second best for all tasks, highlighting its effectiveness even in challenging settings.

Similarity normalization: To delve further into the detection performance comparison, we apply similarity normalization [59], initially proposed for image copy detection, on top of all approaches. For each query video, the top-k neighbors in a background set are estimated. Then, their

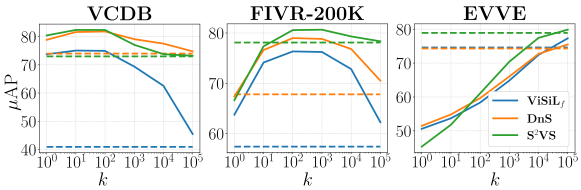

Figure 5. Detection performance measured via μAP with similarity normalization for varying values of k for top-k neighbors. The average performance over the three subtasks is reported on FIVR-200K. Dashed lines indicate performance without normalization.

similarity to the query is averaged to form the query's bias term, which is subtracted from the initial similarities. This impacts the global ranking of video pairs, as each query in the dataset has a different bias term. In this experiment, we use the DnS-100K as the background set and calculate the bias term based on the average similarity of the top-k neighbors. Figure 5 presents the results with and without similarity normalization. All methods benefit from a wide range of k on VCDB and FIVR-200K, but only for very large values on EVVE, while the baseline benefits the most as there is more space for improvements. Our method achieves the best performance after normalization too. Nevertheless, k needs to be tuned independently per test set, which is a major drawback. Therefore, the good detection performance of our approach, even without normalization, is important.

5.2. Ablation study

We perform ablations for different augmentation types and loss functions. For further ablations, we point readers to the supplementary material.

Impact of augmentations: Table 3 presents the performance of the proposed approach for different training augmentation strategies. Each newly added augmentation gives a performance boost, while performance improves by a large margin compared to no use of strong augmentations. We conclude that the variety and diversity of the augmentations are the key ingredients for high performance.

Impact of $\mathcal{L}_{sshn}$ loss: Table 4 reports the mAP and μAP of our method trained with and without the proposed $\mathcal{L}_{sshn}$ loss. It is evident that without the use of the proposed loss, the network does not work effectively. In all evaluation cases, the performance difference is a least 3%, with the most notable discrepancies on detection runs where it reaches almost 10% on VCDB. This highlights that the proposed loss is necessary for the effective training of such a video similarity network.

Label supervision vs self-supervision: Table 5 presents the results of our approach while positives and negatives are drawn from VCSL, which is an annotated dataset. For this supervised variant, the same losses and augmentations are used except for the video-in-video because we found several such cases in the annotated pairs of VCSL. During batch construction, we make sure that the selected video segments for the positive videos overlap with the dataset ground truth. With our SSL training scheme, the model achieves significantly better results, especially for detection, where it outperforms the supervised counterpart by a margin of up to 9%. This highlights the need for a large and diverse dataset to effectively train these video similarity learning models.

	Retrieval			**Detection**		
Augmentations	**VCDB**	**FIVR**	**EVVE**	**VCDB**	**FIVR**	**EVVE**
no strong aug.	88.8	78.5	50.6	79.8	66.9	65.5
+ global trans.	94.3	83.4	59.5	89.5	74.7	76.3
+ frame trans.	95.1	86.2	64.1	90.0	79.8	76.1
+ temporal trans.	95.2	86.5	64.8	89.9	80.8	76.8
+ video-in-video	95.2	87.0	65.9	90.1	81.7	78.9

Table 3. Retrieval mAP (%) and detection μAP (%) of S^{2}VS with different augmentation strategies.

	Retrieval			**Detection**		
$\mathcal{L}_{\text{sshn}}$	**VCDB**	**FIVR**	**EVVE**	**VCDB**	**FIVR**	**EVVE**
✗	89.2	81.6	62.6	80.3	72.3	75.4
✓	95.2	87.0	65.9	90.1	81.7	78.9

Table 4. Retrieval mAP (%) and detection μAP (%) of S^{2}VS with and without $\mathcal{L}_{sshn}$ loss.

		Retrieval			**Detection**		
Train	**Dataset**	**VCDB**	**FIVR**	**EVVE**	**VCDB**	**FIVR**	**EVVE**
Sup.	VCSL	92.4	83.8	64.0	85.6	72.7	73.5
SSL	DnS-100K	95.2	87.0	65.9	90.1	81.7	78.9
SSL	VCDB ($\mathcal{D}$)	-	87.3	67.2	-	78.1	80.7

Table 5. Retrieval mAP (%) and detection μAP (%) for the proposed method with self-supervision on two different datasets and a variant that uses labeled positives and negatives.

6. Conclusions

In this paper, we propose a self-supervised learning approach for training video similarity networks. Eliminating the need for labels allows us to train on large-scale video corpora, which, together with a diverse set of video augmentations, form the key ingredient for achieving top performance. The obtained single model is evaluated on several target retrieval and detection tasks. It manages to perform on par or outperform existing models that exploit labeled datasets, especially for detection due to better similarity calibration across queries.

Acknowledgments: This work was supported by Junior Star GACR under grant No. GM 21-28830M, and the EU partially funded projects H2020 MediaVerse, H2020 AI4Media and Horizon Europe vera.ai under contract No. 957252, 951911 and 101070093, respectively.

References

[1] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *ICCV*, 2015. 2

[2] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *ICLR*, 2019. 2

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2021. 2

[4] Lorenzo Baraldi, Matthijs Douze, Rita Cucchiara, and Hervé Jégou. LAMV: Learning to align and match videos with kernelized temporal layers. In *CVPR*, 2018. 2, 3, 6, 7

[5] Mina Bishay, Georgios Zoumpourlis, and I. Patras. TARN: Temporal attentive relation network for few-shot and zero-shot action recognition. In *BMVC*, 2019. 2

[6] Yang Cai, Linjun Yang, Wei Ping, Fei Wang, Tao Mei, Xian-Sheng Hua, and Shipeng Li. Million-scale near-duplicate video retrieval system. In *ACM MM*, 2011. 2

[7] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *ECCV*, 2018. 2

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020. 2

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 2

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2

[11] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2

[12] Chien-Li Chou, Hua-Tsung Chen, and Suh-Yin Lee. Pattern-based near-duplicate video retrieval and localization on web-scale videos. *IEEE TMM*, 2015. 2

[13] Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*, 2007. 6

[14] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. RandAugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 4

[15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[16] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015. 2

[17] Carl Doersch and Andrew Zisserman. Multi-task self-supervised visual learning. In *ICCV*, 2017. 2

[18] Matthijs Douze, Hervé Jégou, and Cordelia Schmid. An image-based approach to video copy detection with spatio-temporal post-filtering. *IEEE TMM*, 2010. 2

[19] Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, Maxim Maximov, Laura Leal-Taixé, Ismail Elezi, et al. The 2021 image similarity dataset and challenge. In *arXiv:2106.09672*, 2021. 6

[20] Yang Feng, Lin Ma, Wei Liu, Tong Zhang, and Jiebo Luo. Video re-localization. In *ECCV*, 2018. 2

[21] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 2

[22] Zhanning Gao, Gang Hua, Dongqing Zhang, Nebojsa Jojic, Le Wang, Jianru Xue, and Nanning Zheng. ER3: A unified framework for event retrieval, recognition and recounting. In *CVPR*, 2017. 2

[23] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *NeurIPS*, 2020. 2

[24] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *NeurIPS*, 2020. 1, 2

[25] Zhen Han, Xiangteng He, Mingqian Tang, and Yiliang Lv. Video similarity and alignment learning on partial video copy detection. In *ACM MM*, 2021. 2, 3

[26] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2

[27] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[29] Sifeng He, Xudong Yang, Chen Jiang, Gang Liang, Wei Zhang, Tan Pan, Qing Wang, Furong Xu, Chunguang Li, JinXiong Liu, et al. A large-scale comprehensive dataset and copy-overlap aware evaluation protocol for segment-level video copy detection. In *CVPR*, 2022. 5

[30] Sifeng He, He Yue, Minlong Lu, et al. TransVCL: Attention-enhanced video copy localization network with flexible supervision. In *AAAI*, 2023. 2

[31] Xiangteng He, Yulin Pan, Mingqian Tang, Yiliang Lv, and Yuxin Peng. Learn from unlabeled videos for near-duplicate video retrieval. In *ACM SIGIR*, 2022. 2, 3, 6, 7

[32] Olivier Henaff. Data-efficient image recognition with contrastive predictive coding. In *ICML*, 2020. 2

[33] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997. 2

[34] Jing Huang, S Ravi Kumar, Mandar Mitra, Wei-Jing Zhu, and Ramin Zabih. Spatial color indexing and applications. *IJCV*, 1999. 2

[35] Zi Huang, Heng Tao Shen, Jie Shao, Bin Cui, and Xiaofang Zhou. Practical online near-duplicate subsequence detection for continuous video streams. *IEEE TMM*, 2010. 2

[36] Dinesh Jayaraman and Kristen Grauman. Learning image representations tied to ego-motion. In *ICCV*, 2015. 2

[37] Chen Jiang, Kaiming Huang, Sifeng He, Xudong Yang, Wei Zhang, Xiaobo Zhang, Yuan Cheng, Lei Yang, Qing Wang, Furong Xu, et al. Learning segment similarity and alignment in large-scale content based video retrieval. In *ACM MM*, 2021. 2, 3

[38] Yu-Gang Jiang, Yudong Jiang, and Jiajun Wang. VCDB: A large-scale database for partial copy detection in videos. In *ECCV*, 2014. 1, 2, 5, 6

[39] Yu-Gang Jiang and Jiajun Wang. Partial copy detection in videos: A benchmark and an evaluation of popular methods. *IEEE TBD*, 2016. 2

[40] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019. 2

[41] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Near-duplicate video retrieval by aggregating intermediate cnn layers. In *MMM*, 2017. 2, 6

[42] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Near-duplicate video retrieval with deep metric learning. In *ICCVW*, 2017. 2, 6, 7

[43] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. FIVR: Fine-grained Incident Video Retrieval. *IEEE TMM*, 2019. 1, 2, 3, 6

[44] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. ViSiL: Fine-grained spatio-temporal video similarity learning. In *ICCV*, 2019. 2, 4, 6, 7

[45] Giorgos Kordopatis-Zilos, Christos Tzelepis, Symeon Papadopoulos, Ioannis Kompatsiaris, and Ioannis Patras. DnS: Distill-and-Select for Efficient and Accurate Video Indexing and Retrieval. *IJCV*, 2022. 2, 3, 5, 6, 7

[46] Julien Law-To, Li Chen, Alexis Joly, Ivan Laptev, Olivier Buisson, Valerie Gouet-Brunet, Nozha Boujemaa, and Fred Stentiford. Video copy detection: a comparative study. In *ACM CIVR*, 2007. 1, 6

[47] Hyodong Lee, Joonseok Lee, Joe Yue-Hei Ng, and Paul Natsev. Large scale video representation learning via relational graph clustering. In *CVPR*, 2020. 2

[48] Joonseok Lee, Sami Abu-El-Haija, Balakrishnan Varadarajan, and Apostol Natsev. Collaborative deep metric learning for video understanding. In *ACM SIGKDD*, 2018. 2

[49] Pandeng Li, Hongtao Xie, Jiannan Ge, Lei Zhang, Shaobo Min, and Yongdong Zhang. Dual-stream knowledge-preserving hashing for unsupervised video retrieval. In *ECCV*, 2022. 2

[50] Shuyan Li, Xiu Li, Jiwen Lu, and Jie Zhou. Self-supervised video hashing via bidirectional transformers. In *CVPR*, 2021. 2

[51] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *ICLR*, 2016. 6

[52] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 6

[53] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 2

[54] Kennard Ng, Ser-Nam Lim, and Gim Hee Lee. VRAG: Region attention graphs for content-based video retrieval. In *arXiv:2205.09068*, 2022. 2

[55] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[56] Timo Ojala, Matti Pietikäinen, and David Harwood. A comparative study of texture measures with classification based on featured distributions. *PR*, 1996. 2

[57] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. In *arXiv:1807.03748*, 2018. 2, 3, 5

[58] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[59] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *CVPR*, 2022. 3, 4, 6, 7

[60] Sébastien Poullot, Shunsuke Tsukatani, Anh Phuong Nguyen, Hervé Jégou, and Shin'Ichi Satoh. Temporal matching kernel with explicit feature maps. In *ACM MM*, 2015. 2

[61] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021. 1, 2

[62] Adria Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Pătrăucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. In *ICCV*, 2021. 2

[63] Jérôme Revaud, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Event retrieval in large video collections with circulant temporal encoding. In *CVPR*, 2013. 1, 2, 6

[64] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Spreading vectors for similarity search. In *ICLR*, 2018. 5

[65] Lifeng Shang, Linjun Yang, Fei Wang, Kwok-Ping Chan, and Xian-Sheng Hua. Real-time large scale near-duplicate web video retrieval. In *ACM MM*, 2010. 2

[66] Jie Shao, Xin Wen, Bingchen Zhao, and Xiangyang Xue. Temporal context aggregation for video retrieval with contrastive learning. In *WACV*, 2021. 2, 6, 7

[67] Josef Sivic and Andrew Zisserman. Video Google: A text retrieval approach to object matching in videos. In *CVPR*, 2003. 2

[68] Jingkuan Song, Yi Yang, Zi Huang, Heng Tao Shen, and Richang Hong. Multiple feature hashing for real-time large scale near-duplicate video retrieval. In *ACM MM*, 2011. 2

[69] Jingkuan Song, Hanwang Zhang, Xiangpeng Li, Lianli Gao, Meng Wang, and Richang Hong. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE TIP*, 2018. 2

[70] Hung-Khoon Tan, Chong-Wah Ngo, Richard Hong, and Tat-Seng Chua. Scalable detection of partial near-duplicate videos by visual-temporal consistency. In *ACM MM*, 2009. 2

[71] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020. 2

[72] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. In *ICLR*, 2016. 6

[73] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *NeurIPS*, 2022. 2

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[75] Kuan-Hsun Wang, Chia-Chun Cheng, Yi-Ling Chen, Yale Song, and Shang-Hong Lai. Attention-based deep metric learning for near-duplicate video retrieval. In *ICPR*, 2021. 2

[76] Ling Wang, Yu Bao, Haojie Li, Xin Fan, and Zhongxuan Luo. Compact cnn based video representation for efficient video copy detection. In *MMM*, 2017. 2

[77] Rui Wang, Dongdong Chen, Zuxuan Wu, Yinpeng Chen, Xiyang Dai, Mengchen Liu, Yu-Gang Jiang, Luowei Zhou, and Lu Yuan. Bevt: Bert pretraining of video transformers. In *CVPR*, 2022. 2

[78] Xiao Wu, Alexander G Hauptmann, and Chong-Wah Ngo. Practical elimination of near-duplicates from web video search. In *ACM MM*, 2007. 1, 2, 7

[79] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *nAC-ACL*, 2016. 3

[80] Li Yuan, Tao Wang, Xiaopeng Zhang, Francis EH Tay, Zequn Jie, Wei Liu, and Jiashi Feng. Central similarity quantization for efficient image and video retrieval. In *CVPR*, 2020. 2

[81] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 3

[82] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2