Video Representation Learning with Graph Contrastive Augmentation

Jingran Zhang[1], Xing Xu[1,*], Fumin Shen[1], Yazhou Yao[2], Jie Shao[1], and Xiaofeng Zhu[1]
[1]Center for Future Multimedia and School of Computer Science and Engineering,
University of Electronic Science and Technology of China, China
[2]School of Computer Science and Engineering, Nanjing University of Science and Technology, China

ABSTRACT

Contrastive-based self-supervised learning for image representations has significantly closed the gap with supervised learning. A natural extension of image-based contrastive learning methods to the video domain is to fully exploit the temporal structure presented in videos. We propose a novel contrastive self-supervised video representation learning framework, termed Graph Contrastive Augmentation (GCA), by constructing a video temporal graph and devising a graph augmentation that is designed to enhance the correlation across frames of videos and developing a new view for exploring temporal structure in videos. Specifically, we construct the temporal graph in the video by leveraging the relational knowledge behind the correlated sequence video features. Afterwards, we apply the proposed graph augmentation to generate another graph view by cooperating random corruption of the original graph to enhance the diversity of the intrinsic structure of the temporal graph. To this end, we provide two different kinds of contrastive learning methods to train our framework using temporal relationships concealed in videos as self-supervised signals. We perform empirical experiments on downstream tasks, action recognition and video retrieval, using the learned video representation, and the results demonstrate that with the graph view of temporal structure, our proposed GCA remarkably improves performance against or on par with the recent methods. Code is made available at https://github.com/ACMMM2021-Anonymous/video-graph-ssl.

CCS CONCEPTS

• **Computing methodologies** → **Computer vision problems**; **Activity recognition and understanding**.

KEYWORDS

Video representation learning, Self-supervised learning, Contrastive learning, Graph augmentation

ACM Reference Format:
Jingran Zhang, Xing Xu, Fumin Shen, Yazhou Yao, Jie Shao, Xiaofeng Zhu. 2021. Video Representation Learning with Graph Contrastive Augmentation. In *Proceedings of the 29th ACM International Conference on Multimedia (MM '21), October 20–24, 2021, Virtual Event, China.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3474085.3475510

*Corresponding author.

MM '21, October 20–24, 2021, Virtual Event, China

ACM ISBN 978-1-4503-8651-7/21/10…$15.00
https://doi.org/10.1145/3474085.3475510

1 INTRODUCTION

With the help of large-scale datasets, like ImageNet [5], Kinetics [17], *etc*, Convolutional Neural Networks (CNNs) have achieved great success in many computer vision tasks [14, 29, 41], owing to the refined representation learned from the proper-trained networks with strong supervision information. However, such success relies heavily on human annotations, which are laborious, time-consuming, and expensive but are necessary to enable the supervised training of the CNNs. Whereas at the same time, a tremendous amount of unlabeled data is instantly available on the Internet. To dig out inherent information and take advantage of such scale unlabeled data, the community of self-supervised learning (SSL) has been developed for utilizing the intrinsic characteristics of unlabeled data and improving the performance of CNNs. Moreover, learning from the data itself unleashes its potential of easy access property, and accelerates many applications in artificial intelligence where annotating data is difficult.

Recently, self-supervised learning has achieved rapid development on computer vision tasks for image representation learning. Specifically, the key idea of SSL in computer vision is that it typically first devises a specific pretext task to generate a label for each sample, like colorization [45], Jigsaw [25], and path shuffling [6] and then trains the model with this task in a way to acquire a well-trained model. Afterwards, the model outputs hierarchical representations of the input data which can fit into related tasks. The visual representations obtained in this manner have been closing the gap between those with supervised learning in ImageNet. Especially, contrastive learning-based self-supervised methods have even reached state-of-the-art performance for visual image classification [3, 8, 13]. Relying on multi-view sample instances, contrastive learning methods aim at learning representation by maximizing similarity of sample—bringing views of the same sample under a consistency and augmented way that exploit task-specific or data-specific structure, while minimizing similarity of views of different samples. The implementation of multi-view samples is drawn from the composition of data or feature transformations. Moreover, those transformations have the capacity of creating realistically rational samples without destroying the semantic information behind the data.

Meanwhile, SSL for video representation learning has attracted increasing attention in recent years [9, 16, 20, 22, 39]. Different from images, videos have an additional dimension characterized by the temporal evolution of appearance. Learning representations from

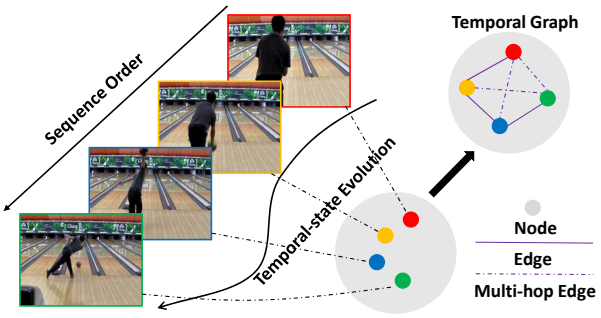

Figure 1: Illustration of Temporal-state Evolution. It is clear that the nearby frames are highly correlated and organized by a temporal ordering sequence. A *graph view* can well model those properties and further captures the semantic relation behind the clip.

such structure in an informative way is urgent for many related applications, as the additional information in videos is redundant and temporal correlated. Inspired by recent advances in contrastive learning, it is also crucial for videos to design a good contrastive formulation with a specific choice of the augmentation viewpoint. In particular, recent works [11, 20, 34] also devised disparate ways in different fashions for video representation learning. In the image domain, most formulations seek an invariance view to spatial transformations. However, for videos, these methods are not as well progressed as images, since they do not have a distinct view for video as image. Considering the complexity of videos, except utilizing frame-level spatial view, the additional temporal dimension determines that it is expected to devise a *clip-specific view* for video representation learning.

However, to fully exploit the temporal coherence of spatial feature, it remains a question about how to devise a distinctive view for videos like images. As shown in Figure 1, we can observe that the frames in a video are highly correlated and only a few pieces of information are evolving in the nearby neighbor clip. Hence, it is except to design a distinct video transformation that can reflect the relevant relation between temporal sequence and discover the hidden temporal semantic knowledge behind the appearance of objects in the frames. To address the above issue, we consider enhancing the temporal state correlation with a distinct *temporal graph view* for videos. Specifically, we introduce graph contrastive learning approach, dubbed Graph Contrastive Augmentation (GCA), to train the video encoder. It is different from recent approaches that merely learn correspondences across frames or clips.

A great challenge is that multi-view visual representation learning is not easily extended to graph data, as the structure in the graph is complex and not well defined. Moreover, simply aggregating information from the original graph cannot thoroughly take advance of the information from a similar graph under different views. Fortunately, the graph constructed from a video clip in our method is well organized and flexible to process. Specifically, the vertexes (nodes) are modeled by sequence features and the edges are built by the multi-hop sequence order of the clip, which leverages the choice of each graph as an instance is feasible, as nodes in the graph are highly correlated. Additionally, recent works [12, 27, 44, 46] on graph self-supervised representation also give us a reminder about how to augment graphs for multi-graph views. To this end, we can naturally exploit the contrastive graph learning for distinct video clip view designing and further capture the richer temporal structure of videos.

The general framework of our GCA approach is illustrated in Figure 2. It can be summarized as that 1) using a video encoder to capture the clip sequence features, 2) adopting graph augmentation from the clip features for multi-view representation learning, and 3) leveraging contrastive self-supervised task for whole framework training. Specifically, we first apply a specific sampling method and a set of traditional augmentation techniques to get the input clips $\mathbf{V}$ and $\mathbf{V}'$ for encoding information of spatial invariances. Afterwards, we adopt a video encoder $\mathcal{F}^v$ (3D ConvNets) to extract features $\mathbf{z}$ and $\mathbf{z}^p$ from clips $\mathbf{V}$ and $\mathbf{V}'$ respectively. Additionally, the graph contrastive augmentation learning proposed in our method is exploited to process features $\mathbf{z}$ and $\mathbf{z}^p$ for distinct multi-view temporal graph learning. During graph contrastive augmentation learning, we first construct the temporal graph $\mathcal{G}$ from input feature embeddings, in which the edge value assignment and weight diffusion step are proposed. We then propose the graph augmentation to generate another graph view $\mathcal{G}'$ for multi-view graph contrastive augmentation learning. Finally, we adopt Graph Convolution Networks (GCNs) to process the augmented graph for features aggregation and output the corresponding features $\tilde{\mathbf{z}}_1$ and $\tilde{\mathbf{z}}_2$ of $\mathbf{z}$ and $\mathbf{z}^p$, respectively. Notably, the whole framework takes two kinds of contrastive self-supervised learning tasks for training, which will be discussed in Section 3.4. To this end, we evaluate the video encoder $\mathcal{F}^v$ on various video-relevant downstream tasks, *e.g.,* action recognition, and video retrieval. Extensive experiments on those tasks about feature transfer, *e.g.,* as a feature extractor or used for fine-tuning, demonstrate that the superiority of our GCA model compared to a bundle of state-of-the-art self-supervised learning methods for video representation learning.

To summarize, we make the following three contributions:

- We propose an effective self-supervised video representation learning framework term GCA to better explore temporal information beyond spatial information in videos, in which two kinds of different contrastive learning methods are leveraged to train the whole framework.
- We further introduce a graph contrastive augmentation method to regularize the temporal coherence in contrastive learning of videos. Specifically, the temporal graph construction is developed for enhancing the temporal state correlation, and the graph augmentation is proposed for multi-view graph contrastive learning and temporal structure exploration.
- We conduct extensive experiments and thoroughly evaluate the quality of the learned representations to a variety of video downstream tasks, showing that our GCA method can lead to remarkable improvement of the performance on those downstream tasks.

2 RELATED WORK

Contrastive Learning. Contrastive learning methods currently have attracted significant attention in self-supervised learning [3, 13, 32], in which the models are trained to distinguish single instances from multi-view in a classification manner. It requires

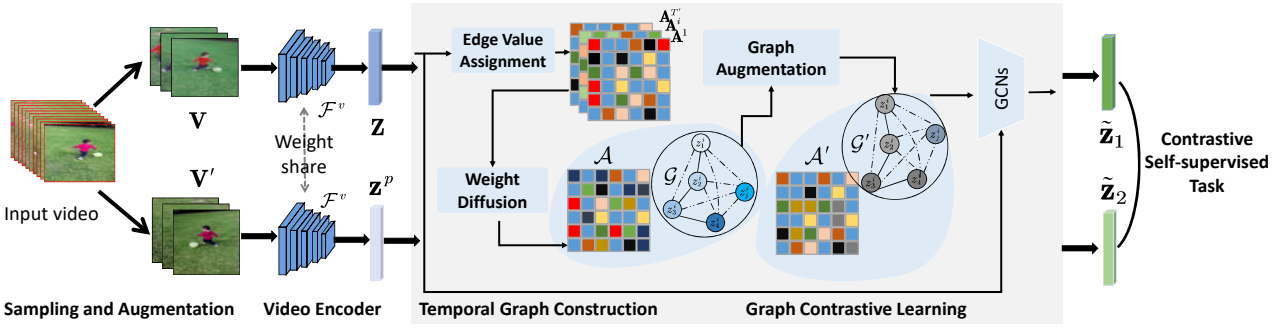

Figure 2: Illustration of the General Framework of Our GCA method for Video Representation Learning.

comparing each instance with large-scale numbers of examples. Previous works [3, 13, 37] prove that the use of rich negative samples can help the learning process. Wu *et al.* [37] proposed to use a memory bank to store latent embeddings of samples, while He *et al.* [13] proposed to replace the memory bank with a queue and apply a momentum encoder to update the queue. However, this is still an open question that whether large-scale negative samples are needed and what guarantees the model learning well without collapsing in a contrastive manner. Some contrastive methods [4, 8] also attempt to investigate on training the model without using negative pairs. BYOL uses a moving average network to produce prediction targets as a means of stabilizing the bootstrap step.

Following those successful image-based contrastive learning methods, there are also many attempts to extend it into the video domain. For instance, contrastive multi-view coding (CMC) [32] was proposed to utilize the optical flow field of video clips for video representation learning in a multi-view manner by maximizing the mutual information between RGB and optical flow data.

Self-supervised Video Representation Learning. Since many works prove the effectiveness of contrastive learning in self supervised image representation learning, some attempts [10, 35, 42] extend it to the video domain by considering the temporal coherent natural. Specifically, except for using the spatial information in each frame, those works also take the temporal continuity property of consecutive video frames into account. For example, Kong *et al.* [20] introduced the cycle-consistency checking progress together with contrastive learning to improve the discrimination of frame and video. Han *et al.* [11] proposed a self-supervised co-training scheme to improve the training regime of the popular contrastive loss. Tao *et al.* [31] introduced the intra-negative samples by breaking the original temporal relations in contrastive learning framework. Wang *et al.* [34] proposed to remove the background impact by adding the background.

Except for the contrastive learning-based methods for video representation learning, a vast range of proxy tasks-based methods also have been recently proposed. For instance, Xu *et al.* [40] proposed to predict the temporal sequence ordering of the frames in the clips. Other works [18, 23] attempted to use the space-time puzzle as a supervisory signal. Han *et al.* [10] proposed a pre-text by predicting the future state with a convex combination of the condensed representations. Benaim *et al.* [1] proposed to learn video representation by predicting the motion speed of the object.

Note that although existing works take the spatio-temporal structure into account and the substantial progress has achieved, they, unfortunately, ignore the temporal state relation behind the clip, which is an essential part of videos. Without the temporal relationship reasoning, the learned representation capability remains limited. Therefore, we further introduce a temporal contrastive graph to leverage such a relationship behind the consecutive clip, providing stronger supervision for video encoder training.

3 PROPOSED METHOD

3.1 Overview of Our GCA Approach

Suppose we have a large-scale video dataset $\mathcal{V} = \{\mathbf{V}_i\}_{i=1}^{N}$, where N is the number of videos in the dataset. We consider adopting a video encoder $\mathcal{F}^v$ to process the video clip $\mathbf{V}_i = \{\mathbf{v}_1^i, \mathbf{v}_2^i, \ldots, \mathbf{v}_T^i\}$ without any human annotation, where $\mathbf{v}_j^i$ is the sampled $j-$th frame from $\mathbf{V}_i$ and the T is the clip length. The video encoder $\mathcal{F}^v$ parameterized by θ typically transforms a video-level clip $\mathbf{V}_i$ into a powerful visual embedding $\mathbf{z}_i$. Our goal is to effectively train the video encoder $\mathcal{F}^v$ and make it possess the ability to generate compact and informative video representation $\mathbf{z}_i = \mathcal{F}^v(\mathbf{V}_i)$ by exploiting the structural knowledge and the consistency within each video clip. The resulting representations are close for similar videos while distinguishable for dissimilar videos and can be effectively used for performing various downstream tasks.

To achieve this, we extend the recently succeeded contrastive learning-based self-supervised methods in the image domain [8, 13, 32] to the video domain. To fully utilize the strong correlated temporal structure of videos, we propose a graph contrastive augmentation framework (GCA) for self-supervised video representation learning. In our GCA, the optimization of the whole framework is performed by maximizing the agreement between two augmented views of the same example and minimizing the similarity of negative examples via a contrastive loss in the feature space. Specifically, our choice of pre-training task and learning objective treats each instance as a distinct class of its own and learns to discriminate

among these instances. To this end, we adopt InfoNCE [33] loss as:

$$\mathcal{L} = -\mathbb{E}\left[\log \frac{\exp(\text{sim}(\mathbf{z}_i \cdot \mathbf{z}^p)/\tau)}{\exp(\text{sim}(\mathbf{z}_i \cdot \mathbf{z}^p)/\tau) + \sum_{n \in \mathcal{N}_i} \exp(\text{sim}(\mathbf{z}_i \cdot \mathbf{z}^n)/\tau)}\right], \tag{1}$$

where $\text{sim}(\cdot)$ is the similarity measure function (dot product is adopted in this paper), and $\mathbf{z}_p$ is the feature sampled from positive set $\mathcal{P}_i$ of $\mathbf{z}_i$ and $\mathbf{z}_n$ is the feature sampled from negative set $\mathcal{N}_i$, the $\mathcal{P}_i$ and $\mathcal{N}_i$ are defined as $\mathcal{P}_i = \{\mathbf{z}^p \mid \mathbf{z}^p \sim \mathcal{D}\}$ and $\mathcal{N}_i = \{\mathbf{z}^n \mid \mathbf{z}^p \sim \mathcal{D}\}$. $\mathcal{D}$ is the feature set of the augmented data of $\mathcal{V}$ and contains different views of any video $\mathbf{v}_i$. Notably, the implementation of different views of the given video in this paper is through a set of data augmentation functions.

3.2 Temporal Graph Construction

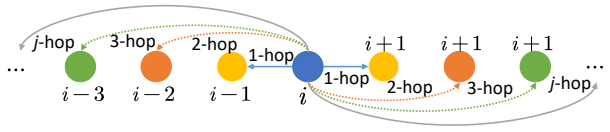

Figure 3: Illustration of the Multi-hop Structure in a Video Clip. Different nodes represent different frames in the clip, and the color represents the hop-connective relation.

A general option for contrastive video representation learning is to directly extend recent methods with video clip augmentation. In this paper, we devise a *"graph view"* for video representation learning to further consider the temporal structure in a clip. Given the extracted feature representation sequence $\mathbf{z}_i = \{z_1^i, \ldots, z_{T'}^i\}$, the temporal order $\ldots, i-1, i, i+1, \ldots$ and semantic correlation between z_j^i and z_k^i are the essential points needed to consider for video model building. Due to the effectiveness of graph network in relationship modeling, we integrate the graph construction of video data in our video encoder $\mathcal{F}^v$ to enhance the frame interaction with each clip. A simple example of a node and its multi-hop structure is illustrated in Figure 3.

Particularly, we construct a fully connected T' hop non-directional temporal graph $\mathcal{G} = (\mathbf{Z}, \mathcal{E})$ on a feature sequence of a T'-frames clip to better utilize video temporal structure, where $\mathbf{Z}$ represents the feature node set and $\mathcal{E}$ donates the edge set which connects the node in $\mathbf{Z}$.

Edge Value Assignment. Afterwards, we compute the correlation value to each edge e in $\mathcal{E}$. Specifically, we first project the original visual feature to an interaction latent space. Formally, given the visual feature $\mathbf{z}_i$. We calculate the similarity of each feature node by dot product. The whole process is expressed as:

$$a_{mn} = \frac{\exp(\mathbf{W}z_m^i \cdot \mathbf{W}z_n^i)}{\sum_{k=1}^{T'} \exp(\mathbf{W}z_m^i \cdot \mathbf{W}z_k^i)}, \tag{2}$$

where $\mathbf{W}$ is the projection matrix and a_{mn} is the attention score between z_m^i and z_n^i and will be assigned to edge e_{mn}. Consequently, we can obtain an attention score matrix $\mathbf{A}$ by performing element-wise operation listed in Equ. 2 as $\mathbf{A} = \{a_{mn}\}_{T' \times T'}$. To this end, a multi-hop temporal graph is constructed with edge value assignment in a specific way, *i.e.,* it is an edge selective way. For example, in the k-hop graph edge value assignment, the edge weight between z_m^i and z_n^i in attention score matrix $\mathbf{A}^k$ is set as a_{mn} only when they are connected in k-hop graph, otherwise, it will be set as 0.

Weight Diffusion in Temporal Graph. Having formulated the temporal graph $\mathcal{G}$ and obtained edge value assignment, we further advance to enable modeling the temporal dynamics within multi-hop connection frames. Recall that in the construction of the temporal graph, the multi-hop edge of the graph is constructed by temporal order connection across consecutive frames. This enables us to diffuse the graph attention score to not direct connected frames. We achieve this via extending the concept of neighborhood to multi-hop temporally connected frames. The diffusion procedure further assigns the range weight of multi-hop neighbors via graph diffusion on $\mathbf{A}$:

$$\mathcal{A} = \sum_{k=0}^{T'} \omega_k \mathbf{A}^k, \tag{3}$$

where $\mathcal{A}$ is the final adjacent matrix, ω_k is the similarity decay factor, and $\omega_k > \omega_{k+1}$. We set $\omega_k = \alpha(1-\alpha)^k$, where $0 < \alpha < 1$ is a hyper-parameter. Actually, this mechanism enables the model not only to depend on directly connected frames but also to take into account the multi-hop path between any frame in the video clip, effectively creating correlation shortcuts between frames that are not connected. Notably, the similarity value between two frames is weighted depending on attention score a_{ij} and path k.

3.3 Graph Contrastive Learning

In Section 3.2, we devise a temporal graph to fully explore temporal knowledge. However, we can not directly utilize the image domain transformation methods to build graph view for temporal coherence modeling, *i.e.,* the application of cropping, rotation, color transformation, etc, only acquire the knowledge that different views of an image encode the same semantic information. For a sequence of correlated frames in a video clip, the frame instances between the clip instance are inherently linked and dependent on each other. Therefore, the coherent temporal structure in videos determines that we need to devise a specific augmentation method to further explore the *graph view of temporal structure.* Particularly, multiple inherent information resources encoded by the temporal graph including node attributes, structure information like node topological are available and their interactions and combinations provide a way for us to design valuable graph augmentation method.

To generate different correlated graph views for the temporal graph $\mathcal{G}$, the most obvious approaches for the augmentation strategy involve adding or removing nodes (frames feature representations) or edges (connectivity of frames). However, the question remains which edges to change, nodes adding poses challenges in imputing features and perturbing the connectivity of new nodes, and node dropping will randomly discard the certain portion of vertices along with their connections in $\mathcal{G}$. Previous work [12] proves that sampling on each edge to generate another view of $\mathcal{G}$ prevents the graph from arbitrarily deviating from the original graph adjacency. It implies that the inherent structure of $\mathcal{G}$ has certain robustness to the random noise.

Graph Augmentation. Since the augmented graphs typically provide different views for the original graph, we randomly corrupt the original graph at attribute levels to enhance the diversity of

for intrinsic structure of $\mathcal{G}$. Specifically, for graph $\mathcal{G}$, in the edge sampling phase, we permute $\mathcal{A}$ with Bernoulli sampling trick on each edge to generate two different graph views $\mathcal{A}'$ and $\mathcal{A}''$. The edge in the original graph is randomly permuted with Bernoulli sampling to get the graph variant adjacency $\mathbf{A}'$. For cooperating the discrete sampling process with the back-propagation algorithm in training, we exploit a relaxed Bernoulli sampling procedure, Gumbel-Softmax reparameterization trick, to integrate it into the differentiable layers. Hence, the Gumbel-Softmax straight-through gradient estimator [15] is applied for approximating relaxed samples to guarantee their sparsity. Formally, the resulting adjacent matrix can be computed as:

$$\mathcal{A}'_{ij} = \frac{\exp((\log(\mathcal{A}_{ij}) + \mathbf{G}_{ij})/\tau')}{\sum_{k=1}^{T'} \exp((\log(\mathcal{A}_{ik}) + \mathbf{G}_{ik})/\tau')}, \tag{4}$$

where $\mathcal{A}'$ is the augmented adjacency matrix, τ' is the temperature parameter of Gumbel-Softmax distribution, $\mathbf{G}_{ij}$ is Gumbel random variate drawn from a Gumbel distribution $\mathbf{G} \sim \text{Gumbel}(0, 1)$. Another view $\mathcal{A}''$ of $\mathcal{G}$ is also generated in this way like $\mathcal{A}'$.

To this end, the feature representations of two augmented views are processed by a Graph Convolution Network (GCN) encoder $\text{gcn}(\cdot)$ and denoted as $\tilde{\mathbf{z}}_1 = \text{gcn}(\mathbf{z}, \mathcal{A}')$ and $\tilde{\mathbf{z}}_2 = \text{gcn}(\mathbf{z}^p, \mathcal{A}'')$.

3.4 Contrastive Self-supervised Task

The key idea of contrastive learning is to learn intrinsic structure which is invariant to any view of a sample. However, solving the objective of this approach typically suffers the issue of the existence of trivial constant solutions. Fortunately, there are two widely used tricks to avoid such collapsed solutions. The first one relies on a large number of negative pairs to avoid such collapse, like, Indis [37], SimCLR [3], MoCo [13]. The second type uses a mechanism dubbed "stop-gradient" to prevent collapsing, like BYOL [8], SimSiam [4]. In this paper, we adopt both the two types for solving our GCA training and show a fair experiment result.

Negative Pairs. Specifically, for the large-scale negative pairs case, we exploit MoCo [13] to train our method, which maintains a dictionary queue Q. Given the encoded query Q of $K+1$ encoded keys $Q = \{\mathbf{z}^p, \mathbf{z}_1^n, \ldots, \mathbf{z}_k^n\}$, the objective of our GCA is to look up the single key $\mathbf{z}^p$ that our query $\mathbf{z}^i$ matches. More formally, MoCo updates $\mathbf{z}_i^n$ by a momentum-based encoder $\mathcal{F}^{v\prime}$ whose parameters θ' are updating by:

$$\theta' \leftarrow m\theta' + (1-m)\theta, \tag{5}$$

where $m \in [0, 1)$ is a momentum updating hyper-parameter. Therefore, the final loss in this case can expressed as:

$$\mathcal{L}_1 = -\log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}^p/\tau)}{\sum_{\mathbf{z} \in Q} \exp(\mathbf{z}_i \cdot \mathbf{z}/\tau)}. \tag{6}$$

Stop Gradient. Additionally, for the "stop-gradient" case, we apply SimSiam [4] to optimize our framework, which designs an additional prediction MLP head [8] h to transform $\mathbf{z}_i$ to logist $\mathbf{p}_i = h(\mathbf{z}_i)$ for matching its positive pair $\mathbf{z}^p$. Equipped with "stop-gradient" operation $\text{stopgrad}(\cdot)$, the whole objective is shown as:

$$\mathcal{L}_2 = \frac{1}{2}\mathcal{S}(\mathbf{p}_i, \text{stopgrad}(\mathbf{z}^p)) + \frac{1}{2}\mathcal{S}(\mathbf{p}^p, \text{stopgrad}(\mathbf{z}_i)), \tag{7}$$

where $\mathcal{S}(\cdot, \cdot)$ measures the negative cosine similarity of two terms, and donates as $\mathcal{S}(\mathbf{p}_i, \mathbf{z}^p) = -\frac{\mathbf{p}_i}{\|\mathbf{p}_i\|_2} \cdot \frac{\mathbf{z}^p}{\|\mathbf{z}^p\|_2}$, $\|\cdot\|_2$ is ℓ_2- norm. The whole training process of our proposed GCA approach is summarized in Algorithm 1.

Algorithm 1: Graph contrastive augmentation learning procedure for video representations learning.

Input : An unlabeled video set $\mathcal{V} = \{\mathbf{V}_i\}_{i=1}^N$, where $\mathbf{V}_i = \{\mathbf{v}_1^i, \mathbf{v}_2^i, \ldots, \mathbf{v}_T^i\}$.
Output : The video encoder $\mathcal{F}^v$ with weight θ.

```
for epoch ← 1 do
    for V_i ∈ 𝒱 do
        Extract feature of V_i using video encoder 𝓕^v as
          z_i = 𝓕^v(V_i);
        Construct the temporal graph 𝒢 with adjacent
          matrix 𝒜;
        Generate the multi-view of 𝒢 to get 𝒜' and 𝒜'';
        Train the whole model with ℒ_1 or ℒ_2;
    end
end
Back-propagate and update 𝓕^v;
return The wegiht parameter θ
```

4 EXPERIMENT

4.1 Experiment Setup

We evaluate the performance of GCA in the downstream transfer-learning tasks following the common practice in self-supervised video representation learning [11, 20, 34]. Various experiments are conducted about *action recognition* and *video retrieval* for the learned video encoder $\mathcal{F}^v$ evaluation. The experiments cover fine-tuned and line probing settings of video encoder $\mathcal{F}^v$, which are also adopted in previous works [20, 34].

4.1.1 Dataset. We exploit the **Kinetics400** dataset [17] as the pre-training dataset for video representation learning. Only 240k video clips from training split are used in our self-supervised learning process, and the average duration of a video clip is around 10 seconds. For downstream tasks, we evaluate our encoder on UCF101 [30] and HMDB51 [21] datasets. UCF101 consists of 101 human action classes and around 13k videos. HMDB51 contains around 7k videos with 15 human action classes.

4.1.2 Video Encoder. We apply the **S3D** [38] architecture as the video encoder for all experiments to extract video features respectively. During GCA pre-training, we use MoCo [13] or Simsiam [4] to optimize our approach. We attach a non-linear projection head [13] at the bottom of S3D to produce 128-D embeddings. The 128-D embeddings are used for MoCo contrastive loss. As for Simsiam mode, except for the projection head, an extra non-linear prediction head [4, 8] is attached to generate the same size embeddings of the projection head.

4.1.3 Implementation Details. We sample 16 frames of a video clip at a sliding step 4 and then randomly crop it to 112×112 resolutions

Table 1: Overall comparison of our proposed GCA method and the compared approaches on the action recognition benchmarks of UCF101 and HMDB51. We also list four other groups of factors in the table, including pre-training dataset size, backbone, input size, and model complexity. This table has a better trade-off in terms of accuracy and those factors compared to previous methods. † refers to using use additional optical flow information, and ⋆ means our implementation for video.

Method	Venue	Pre-train dataset	Backbone	Size	#Parameters	#Flops	UCF101	HMDB51
From scratch			S3D	32x224x224	8.3M	36.3G	67.7	24.2
Supervised		Kinetics400	S3D	32x224x224	8.3M	36.3G	93.8	72.4
Shuffle & Learn [24]	ECCV 2016	UCF101	CaffeNet	1x227x227	58.3M	7.6G	50.2	18.1
Geometry [7]	CVPR 2018	UCF101	FlowNet	1x227x227	-	-	54.1	22.6
OPN [22]	ICCV 2017	UCF101	CaffeNet	1x227x227	58.3M	7.6G	56.3	23.8
ST order [2]	ECCV 2018	UCF101	CaffeNet	1x227x227	58.3M	7.6G	58.6	25.0
Cross & Learn [28]	ECCV 2018	UCF101	CaffeNet	1x227x227	58.3M	7.6G	58.7	27.2
IIC† [31]	ACM MM 2020	UCF101	3D-ResNet18	16x224x224	33.6M	48.4G	72.7	36.8
CMC [32]	ECCV 2020	UCF101	CaffeNet	11x227x227	58.3M	83.6G	59.1	26.7
VCOP [40]	CVPR 2019	UCF101	R(2+1)D-18	16x112x112	33.3M	8.3G	72.4	30.9
PRP [43]	CVPR 2020	UCF101	R(2+1)D-10	16x112x112	14.4M	5.9G	72.1	35.0
RotNet3D [16]	2018	Kinetics400	3D-ResNet18	16x112x112	33.6M	8.5G	62.9	33.7
3D-ST-Puzzle [19]	AAAI 2019	Kinetics400	3D-ResNet18	16x112x112	33.6M	8.5G	63.9	33.7
AoT [36]	CVPR 2018	Kinetics400	T-CAM	64x224x224	-	-	79.4	-
DPC [9]	ICCVW 2019	Kinetics400	Custom 3D-ResNet34	25x224x224	32.6M	85.9G	75.7	35.7
VTHCL [42]	2020	Kinetics400	3D-ResNet18	24x224x224	33.6M	72.6G	80.6	48.6
SpeedNet [1]	CVPR 2020	Kinetics400	I3D	64x224x224	12.7M	111.4G	66.7	43.7
SpeedNet [1]	CVPR 2020	Kinetics400	S3D-G	64x224x224	9.6M	74.0G	81.1	48.8
MemDPC† [10]	ECCV 2020	Kinetics400	Custom 3D-ResNet34	64x224x224	32.6M	219.9G	86.1	54.5
Pace [35]	ECCV 2020	Kinetics400	R(2+1)D-10	16x112x112	14.4M	5.9G	77.1	36.6
Pace [35]	ECCV 2020	Kinetics400	S3D-G	64x224x224	9.6M	74.0G	87.1	52.6
CoCLR† [11]	NIPS 2020	Kinetics400	S3D	64x128x128	8.3M	23.7G	87.9	54.6
CCL [20]	NIPS 2020	Kinetics400	Custom 3D-ResNet18	8x112x112	12.1M	8.3G	69.4	37.8
BE [34]	CVPR 2021	Kinetics400	I3D	16x224x224	12.7M	27.9G	86.8	55.4
BE [34]	CVPR 2021	Kinetics400	3D-ResNet34	16x224x224	63.7M	50.3G	87.1	56.2
MoCo-video⋆		Kinetics400	S3D	32x224x224	8.3M	36.3G	77.8	40.5
Simsiam-video⋆		Kinetics400	S3D	32x224x224	8.3M	36.3G	78.6	39.9
GCA-M		Kinetics400	S3D	32x224x224	8.3M	36.3G	87.4 (9.6↑)	55.3 (14.8↑)
GCA-S		Kinetics400	S3D	32x224x224	8.3M	36.3G	87.3 (8.7↑)	55.7 (15.8↑)

Table 2: Linear probe on UCF101 and HMDB51 datasets.

Method	Pre-train	UCF101	HMDB51
CCL [20]	Kinetics400	52.1	27.8
MemDPC [10]	Kinetics400	54.1	30.5
MoCo-video	Kinetics400	57.9	30.2
Simsiam-video	Kinetics400	57.4	30.5
GCA-M	Kinetics400	68.3 (10.4↑)	40.1 (9.9↑)
GCA-S	Kinetics400	69.1 (11.7↑)	39.7 (9.2↑)

as input to the video encoder. We also adopt image augmentation for each frame in a temporally consistent manner, *i.e.,* random cropping, horizontal flipping, Gaussian blurring, and color jittering are all with the same parameter for each frame. Moreover, we also adopt temporal jitter to fully utilize the natural that a random clip encodes the same pattern from the source video. The model is training with SGD using a linear warm-up scheme at an initial learning rate of 0.03. The SGD weight decay is 10^{-5} and the momentum is 0.9. The total epochs we used are 200 and the batch size is set as 128 with experiments on 8 RTX 2080ti GPU cards. We set the negative pairs K as 16,384, the temperature parameter τ as 0.07 for MoCo.

4.1.4 Downstream Tasks. To evaluate the learned encoder $\mathcal{F}^v$, we evaluate the quality of the pre-training video representations by transferring them to **action recognition** and **video retrieval** on UCF101 [30] and HMDB51 [21] datasets. we mainly offer two strategies for downstream tasks, fine-tuning and linear probe. In fine-tuning mode, the pre-trained encoder is trained end-to-end together with the classifier in a supervised manner. In linear probe mode, all the parameters except the linear classifier layer are frozen. Both evaluation tasks and settings are the common practice in self-supervised video representation learning.

4.2 Overall Results on Downstream Tasks

As video representation learning has a large variety of experimental setups, we compare with self-supervised video representation methods in terms of architecture, pre-training dataset, input resolution, and computation complexity. Note that we use **GCA-M** to represent the feature encoding method for negative pairs. Moreover, we apply **GCA-S** to denote stop-gradient trick we used.

4.2.1 Action Recognition. Once we complete the pre-training process, we apply the learned video encoder $\mathcal{F}^v$ (S3D) with a randomly initialized classification layer for action recognition. There are typically two types for evaluation: 1)end-to-end fine-tuning; 2) linear-probe.

End-to-end Fine-tuning. In this protocol, the whole model is trained in an end-to-end manner. We transfer the learned model from a large-scale Kinetics400 dataset to small-scale datasets, UCF101 and HMDB51 datasets. We average the results of 10 clips of a video to evaluate the samples following the common practices [11, 34, 35] for a fair comparison. Due to the large variability of experimental settings adopted by related methods [10, 11, 31, 34, 35], we also list those different factors for comparison in Table 1.

Table 3: Retrieval results on UCF101 dataset.

Method	Backbone	R@1	R@5	R@10	R@20	R@50
Random	S3D	11.3	20.5	29.2	36.5	49.9
Jigsaw [26]	AlexNet	19.7	28.5	33.5	40.0	49.4
OPN [22]	AlexNet	19.9	28.7	34.0	40.6	51.6
VCOP [40]	C3D	12.5	29.0	39.0	50.6	66.9
VCP [23]	C3D	17.3	31.5	42.0	52.6	67.7
PRP [43]	3D-ResNet18	22.8	38.5	46.7	55.2	69.1
CMC [32]	CaffeNet	26.4	37.7	45.1	55.3	66.3
IIC [31]	R3D	36.5	54.1	62.9	72.4	83.4
MemDPC [10]	C-3D-ResNet	20.2	40.4	52.4	64.7	-
Pace [35]	C3D	31.9	49.7	59.2	68.9	80.2
SpeedNet [1]	S3D-G	13.0	28.1	37.5	49.5	-
CCL [20]	C-3D-ResNet	22.0	39.1	44.6	56.3	70.8
MoCo-video	S3D	28.9	44.4	53.4	63.4	77.4
Simsiam-video	S3D	29.3	45.1	55.6	64.1	78.7
GCA-M	S3D	**39.2**	56.1	67.5	79.3	**87.4**
GCA-S	S3D	38.8	**56.3**	**68.9**	**79.9**	87.2

Table 4: Retrieval results on HMDB51 dataset.

Method	Backbone	R@1	R@5	R@10	R@20	R@50
Random	S3D	6.9	20.4	29.7	44.6	62.3
VCOP [40]	C3D	7.4	22.6	34.4	48.5	70.1
VCP [23]	C3D	7.8	23.8	35.5	49.3	71.6
PRP [43]	C3D	10.5	27.2	40.4	56.2	75.9
CMC [32]	CaffeNet	10.2	25.3	36.6	51.6	74.3
IIC [31]	R3D	13.4	32.7	46.7	61.5	**83.8**
MemDPC [10]	C-3D-ResNet	7.7	25.7	40.6	57.7	-
Pace [35]	C3D	12.5	32.2	45.4	61.0	80.7
BE [34]	R3D	11.9	31.3	44.5	60.5	81.4
MoCo-video	S3D	9.8	25.5	36.9	52.7	74.1
Simsiam-video	S3D	10.2	25.9	37.4	53.3	74.9
GCA-M	S3D	17.5	35.3	**47.9**	**63.5**	81.6
GCA-S	S3D	**17.7**	**36.4**	47.6	62.8	81.9

From Table 1, we list the following observation: 1) Equipped with GCA, our method brings 9.6% and 14.8% improvement on UCF101 and HMDB51 for MoCo setting, and has boosts of 8.7% and 15.8% on UCF101 and HMDB51 for Simsiam setting; 2) Even with a light-weight video encoder, those self-supervised methods can still have a comparable or even better performance with those with a deeper model, which means that the pre-trained methods are more important. Moreover, deeper models are computationally intractable and hard to deploy. 3) Compared with those self-supervision methods, our method yields better or comparable results. More specifically, compared with the pre-text task self-supervised methods, like [16, 22, 40], our method has at least 15.0% improvement on UCF101 and 18.9% improvement on HMDB51 dataset. While compared with contrastive learning-based self-supervised learning method, we obtain at least 0.3% gain on UCF101 and a comparable result on HMDB51 dataset. It indicates the effectiveness of our proposed GCA method. Note that the method CoCLR uses additional information of optical flow that can boost the performance of self-supervised representation learning.

Linear-probe. To further analyze the transferability of learned representation, we adopt the linear-probe protocol to evaluate our method, which is also widely used for action recognition [10, 20]. Specifically, the parameters of the whole model except the linear classifier are frozen, *i.e.,* we use the pre-trained model as a feature extractor and only train a SVM in a supervised manner to evaluate the extracted features.

Table 2 presents the top-1 classification precision on those two small-scale datasets under this protocol. Obviously, there exists a performance gap between our method and the previous. Specifically, GCA is higher than the baseline method, MoCo, SimSiam, with 10.4% and 11.7% improvement on UCF101 respectively, and also inferior to previous methods. The results show the superiority of our method and the effectiveness of the learned representation.

4.2.2 Video Retrieval. In this section, we evaluate our GCA on video retrieval task using K nearest-neighbor retrieval. For a fair comparison, we pre-train our model on UCF101 dataset and validate it on both UCF101 and HMDB51 datasets, which is also adopted in [11, 31, 34, 40]. Specifically, we use the training set as a gallery and the test set as the query to retrieve the most similar video in the gallery. The metric Recall K (R@K) that whether the top k nearest neighbors contain the clip of the same class of query are reported. We set $k = 1, 5, 10, 20, 50$ in this paper. Note that we use the pre-trained model to extract features of 16 frames and no further training is allowed.

The results on UCF101 and HMDB51 dataset are shown in Table 3 and 4 respectively. Comparison with others self-supervised methods, we can obtain the following observation: 1) In both benchmark datasets, our GCA exceeds all previous approaches by a significant margin; 2) Combining with the baseline, MoCo or Simsiam, our method can bring a 10.3% improvement to R@1 for MoCo and a 9.5% improvement to R@1 for Simsiam on UCF101 dataset, which significantly exceeds the it with the same backbone. 3) Compared with previous works, our GCA method gains at least 2.7% and 4.3% scores of R@1 on UCF101 and HMDB51 dataset respectively. We find that in obtaining strong labels, our method simultaneously learns robust, visual representations that can be used for other tasks without any fine-tuning.

4.3 Further Analysis

4.3.1 How does Graph Contrastive Augmentation Working?

Previous works [12, 27, 44, 46] of graph contrastive learning provide various ways to devise graph augmentations for multi-view graph representation learning. From those works, the most obvious augmentation methods contain node dropping and edge masking, where node dropping will randomly discard a certain portion of vertices along with their connections, and edge masking will randomly add or drop a fraction of edges. In this part, we analyze the role of different graph augmentations in our framework and examine whether those graph augmentations help video representation learning. Specifically, except for the graph augmentation method we used, we additionally discuss multi-view learning by removing nodes and perturbing edges. We apply those augmentation methods to pre-train the video encoder on UCF101 dataset without label and then fine-tune it on UCF101 dataset for action recognition. Note that, only the MoCo case is exploited here for our model pre-training.

The results are shown in Figure 4 (a), which contains five types of methods for comparison. We observe that dropping nodes boosts the performance against the baseline, which implies that missing part of vertices does not affect the semantic meaning of $\mathcal{G}$ as the frames in a clip are highly correlated. Additionally, masking edges also brings improvement against the baseline. The underlying prior demonstrates that the temporal graph $\mathcal{G}$ has certain robustness to

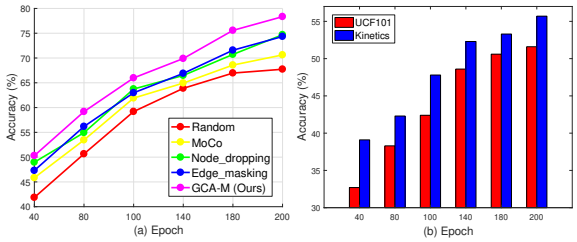

Figure 4: Analysis of our graph augmentation and effect of pre-training dataset scale.

the variation of edge connectivity. Compared to those, the augmentation we adopted can be thought of adding random noise to the graph, and weight diffusion can serve as a denoising filter, which allows information transfer between multi-hop neighbors. We speculate that randomly sampling on each edge together with weight diffusion between adjacent matrices helps in capturing the intrinsic structure of temporal graph. Moreover, maximizing the similarity between the augmented graphs allows the model to encode the invariant graph information.

4.3.2 Large Scale Dataset Help Self-Supervised Learning? Self-supervised representation learning benefits from large scale unlabeled datas. To figure out the effect the pre-training dataset scale, we conduct experiments that pre-train the model on Kinetics400 and UCF101 datasets and test it on HMDB51 dataset for action recognition. The Kinetics400 and UCF101 datasets are both collected from the Youtube website, which can be thought of sampling from the same distribution but with different scales.

The results are shown in Figure 4 (b). As it can be seen that the video encoder pre-trained on a larger unlabelled dataset have a remarkable boost in classification accuracy on small size dataset compared with the baseline model which fully trains the model from scratch. Moreover, the performance increase along with the epoch also indicates that a large training epoch helps for self-supervised contrastive learning. This partly demonstrates that a large-scale dataset with proper training time even without labels can help ConvNets training and produce a performance boost.

4.4 Qualitative Results

4.4.1 Visualization of Retrieval Results. We visualize the Nearest Neighbors (NN) retrieval results of a query clip and its R@1 neighbors on UCF101 and HMDB51 datasets. As shown in Figure 5, the frames in the left part are the query video clips, and the frames in the right part are the retried clips. We can observe that the representation learned by our GCA has the ability to cluster the same semantic meaning video and cares more about the motion.

4.4.2 GAM Visualization. To figure out the difference between our GCA and baseline methods in a qualitative manner, in this part, we visualize the salient region in the video clip using CAM [47], which will highlight the interesting part when classifying an action of the clip. Specifically, we pre-train the video encoder, S3D, on UCF101 dataset in a self-supervised (MoCo, Simsiam, GCA-M, GCA-S) manner. Additionally, we also train S3D on UCF101 dataset from scratch. It can also verify that whether the model can capture temporal information, like moving objects from videos, or

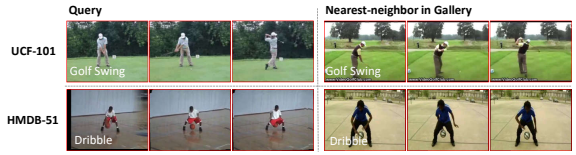

Figure 5: Video Retrieval with Nearest Neighbor on UCF101 and HMDB51 Datasets. The action label for each video clip is shown in the lower left corner.

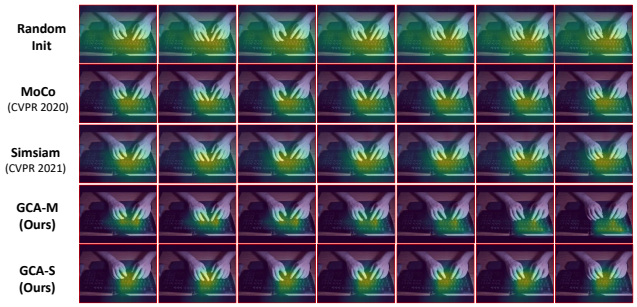

Figure 6: Class Activation Maps. The color in yellow part high the salient region for action classification.

whether the model purely relies on static objects in frames from a video. The visualization results are shown in Figure 6. Note that, we randomly sample an example that S3D trained in all manner can correctly predict the action. From the results, we can observe that pre-trained methods do help S3D focusing on the discriminative part (hand typing) in this example. Moreover, with the graph view in our method, the S3D concentrates more precisely on motion regions and capture the spatio-temporal information within videos.

5 CONCLUSION

In this paper, we present a novel self-supervised video representation learning method termed Graph Contrastive Augmentation (GCA) by devising a *graph view* to fully exploit the temporal structure presented in videos. Moreover, equipped with two widely used tricks in contrastive learning – *memory bank* for large scale negative samples and *stop-gradient*, we explore the temporal relationships as self-supervised signals to train our framework. We conducted extensive downstream experiments including action recognition and video retrieval to evaluate our method. The experiment results demonstrate that the proposed CGA can provide a distinct view for video augmentation and can learn powerful video representation. Note that, the graph view in our method proves that fully exploring the potential of temporal structure definitely benefits video representation learning.

6 ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (61976049, 62072080, 61632007, 61976116 and U20B2063); the Fundamental Research Funds for the Central Universities under Project (ZYGX2019Z015 and 30920021135); the Sichuan Science and Technology Program, China (2018GZDZX0032, 2019ZDZX0008, 2019YFG0003, 2019YFG0533 and 2020YFS0057).

REFERENCES

[1] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. 2020. SpeedNet: Learning the Speediness in Videos. In *Conference on Computer Vision and Pattern Recognition*. 9919–9928.

[2] Uta Büchler, Biagio Brattoli, and Björn Ommer. 2018. Improving Spatiotemporal Self-supervision by Deep Reinforcement Learning. In *European Conference on Computer Vision*, Vol. 11219. 797–814.

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *International Conference on Machine Learning*, Vol. 119. 1597–1607.

[4] Xinlei Chen and Kaiming He. 2021. Exploring Simple Siamese Representation Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.

[6] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. 2015. Unsupervised Visual Representation Learning by Context Prediction. In *IEEE International Conference on Computer Vision*. 1422–1430.

[7] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. 2018. Geometry guided convolutional neural networks for self-supervised video representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition*. 5589–5597.

[8] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *Advances in Neural Information Processing Systems*.

[9] Tengda Han, Weidi Xie, and Andrew Zisserman. 2019. Video Representation Learning by Dense Predictive Coding. In *International Conference on Computer Vision Workshops*. 1483–1492.

[10] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Memory-Augmented Dense Predictive Coding for Video Representation Learning. In *European Conference of Computer Vision*, Vol. 12348. 312–329.

[11] Tengda Han, Weidi Xie, and Andrew Zisserman. 2020. Self-supervised Co-Training for Video Representation Learning. In *Advances in Neural Information Processing Systems*.

[12] Kaveh Hassani and Amir Hosein Khas Ahmadi. 2020. Contrastive Multi-View Representation Learning on Graphs. In *International Conference on Machine Learning*, Vol. 119. 4116–4126.

[13] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Conference on Computer Vision and Pattern Recognition*. 9726–9735.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[15] Eric Jang, Shixiang Gu, and Ben Poole. 2017. Categorical Reparameterization with Gumbel-Softmax. In *International Conference on Learning Representations*.

[16] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. 2018. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387* (2018).

[17] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv:1705.06950* (2017).

[18] Dahun Kim, Donghyeon Cho, and In So Kweon. 2019. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In *AAAI Conference on Artificial Intelligence*. 8545–8552.

[19] Dahun Kim, Donghyeon Cho, and In So Kweon. 2019. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In *(AAAI) Conference on Artificial Intelligence*. 8545–8552.

[20] Quan Kong, Wenpeng Wei, Ziwei Deng, Tomoaki Yoshinaga, and Tomokazu Murakami. 2020. Cycle-Contrast for Self-Supervised Video Representation Learning. In *Advances in Neural Information Processing Systems*.

[21] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso A. Poggio, and Thomas Serre. 2011. HMDB: A large video database for human motion recognition. In *International Conference on Computer Vision*. 2556–2563.

[22] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. 2017. Unsupervised Representation Learning by Sorting Sequences. In *International Conference on Computer Vision*. 667–676.

[23] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. 2020. Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning. In *AAAI Conference on Artificial Intelligence*. 11701–11708.

[24] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. 2016. Shuffle and Learn: Unsupervised Learning Using Temporal Order Verification. In *European Conference on Computer Vision*, Vol. 9905. 527–544.

[25] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Vision*, Vol. 9910. 69–84.

[26] Mehdi Noroozi and Paolo Favaro. 2016. Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles. In *European Conference on Computer Science*, Vol. 9910. 69–84.

[27] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *Conference on Knowledge Discovery and Data Mining*. 1150–1160.

[28] Nawid Sayed, Biagio Brattoli, and Björn Ommer. 2018. Cross and learn: Cross-modal self-supervision. In *German Conference on Pattern Recognition*. 228–243.

[29] Heng Tao Shen, Luchen Liu, Yang Yang, Xing Xu, Zi Huang, Fumin Shen, and Richang Hong. 2020. Exploiting Subspace Relation in Semantic Labels for Cross-modal Hashing. *IEEE Transactions on Knowledge and Data Engineering* (2020), 10.1109/TKDE.2020.2970050.

[30] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402* (2012).

[31] Li Tao, Xueting Wang, and Toshihiko Yamasaki. 2020. Self-supervised Video Representation Learning Using Inter-intra Contrastive Framework. In *ACM International Conference on Multimedia*. 2193–2201.

[32] Yonglong Tian, Dilip Krishnan, and Phillip Isola. 2020. Contrastive Multiview Coding. In *European Conference of Computer Vision*, Vol. 12356. 776–794.

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation Learning with Contrastive Predictive Coding. arXiv:arXiv:1807.03748

[34] Jinpeng Wang, Yuting Gao, Ke Li, Yiqi Lin, Andy J Ma, Hao Cheng, Pai Peng, Rongrong Ji, and Xing Sun. 2021. Removing the Background by Adding the Background: Towards Background Robust Self-supervised Video Representation Learning. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[35] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. 2020. Self-supervised Video Representation Learning by Pace Prediction. In *European Conference on Computer Vision*, Vol. 12362. 504–521.

[36] Donglai Wei, Joseph J. Lim, Andrew Zisserman, and William T. Freeman. 2018. Learning and Using the Arrow of Time. In *IEEE Conference on Computer Vision and Pattern Recognition*. 8052–8060.

[37] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *Conference on Computer Vision and Pattern Recognition*. 3733–3742.

[38] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. 2018. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In *European Conference on Computer Vision*, Vol. 11219. 318–335.

[39] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. 2019. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*. 10334–10343.

[40] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. 2019. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *Conference on Computer Vision and Pattern Recognition*. 10334–10343.

[41] Xing Xu, Fumin Shen, Yang Yang, Heng Tao Shen, and Xuelong Li. 2017. Learning Discriminative Binary Codes for Large-scale Cross-modal Retrieval. *IEEE Trans. Image Processing* 26, 5 (2017), 2494–2507.

[42] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. 2020. Video Representation Learning with Visual Tempo Consistency. arXiv:arXiv:2006.15489

[43] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. 2020. Video Playback Rate Perception for Self-Supervised Spatio-Temporal Representation Learning. In *Conference on Computer Vision and Pattern Recognition*. 6547–6556.

[44] Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. 2020. Graph Contrastive Learning with Augmentations. In *Advances in Neural Information Processing Systems*.

[45] Richard Zhang, Phillip Isola, and Alexei A. Efros. 2016. Colorful Image Colorization. In *European Conference on Computer Vision*, Vol. 9907. 649–666.

[46] Tong Zhao, Yozen Liu, Leonardo Neves, Oliver J. Woodford, Meng Jiang, and Neil Shah. 2021. Data Augmentation for Graph Neural Networks. In *(AAAI) Conference on Artificial Intelligence*.

[47] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *IEEE Conference on Computer Vision and Pattern Recognition*. 2921–2929.