SELF-SUPERVISED TEMPORAL LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Self-supervised learning (SSL) has shown its powerful ability in discriminative representations for various visual, audio, and video applications. However, most recent works still focus on the different paradigms of spatial-level SSL on video representations. How to self-supervised learn the inherent representation on the *temporal* dimension is still unrevealed. In this work we propose self-supervised temporal learning (SSTL), aiming at learning spatial-temporal-invariance. Inspired by spatial-based contrastive SSL, we show that significant improvement can be achieved by a proposed temporal-based contrastive learning approach, which includes three novel and efficient modules: temporal augmentations, temporal memory bank and SSTL loss. The temporal augmentations include three operators – temporal crop, temporal dropout, and temporal jitter. Besides the contrastive paradigm, we observe the temporal contents vary between each layer of the temporal pyramid. The SSTL extends the upper-bound of the current SSL approaches by $\sim$6% on the famous video classification tasks and surprisingly improves the current state-of-the-art approaches by $\sim$100% on some famous video retrieval tasks. The code of SSTL is released with this draft, hoping to nourish the progress of the booming self-supervised learning community.

1 INTRODUCTION

With the explosive growth of unlabeled video data, self-supervised representation learning of videos has been drawing increasing attention from the vision community, aims at leveraging the information from unlabelled data to learn discriminative video representations. Recent advances (He et al. (2020); Tian et al. (2019); Doersch et al. (2015); Zhang et al. (2016); Noroozi & Favaro (2016)) in self-supervised representation learning for images have demonstrated promising developments. But progress on learning from the videos is less investigated in depth, especially for learning from the temporal information. Compared with single images, videos consist of a temporal sequence of image frames, and are more suitable sources for self-supervised representation learning as the contiguous frames are naturally correlated along the temporal dimension.

Over the past few years, researchers try to tackle the problem of self-supervised learning of videos along two research directions: (1) pretext tasks, such as future frame prediction (Han et al. (2020)), rotation transformation prediction (Jing et al. (2018)), and video order prediction (Xu et al. (2019)); (2) contrastive learning, including (Zhuang et al. (2020); Qian et al. (2020)). However, designing proper training supervisions for videos (especially for temporal information) is challenging and non-trivial, and many previous works still focus on variants of the spatial-level supervisions.

In this work, we analyze the characteristics of feature distribution extracted from a fully-supervised model in videos, and would like our unsupervised model to follow the similar characteristics. We make three observations: (1) As shown in Figure 1, the features from a fully-supervised model are insensitive to such sight temporal disturbances. This phenomenon also accords with our perception: when humans watch short videos, some slight temporal disturbances would not cause noticeable impacts to the correct understanding of the videos. (2) In Figure 2(a), the inter-video similarities are significantly higher than those of intra-video ones. (3) As the temporal distance increases, the similarity between pairs of clips within the same video (Figure 2(b)) gradually decreases.

Based on the above observations, we propose a contrastive learning framework learning feature representations to follow the same characteristics of a fully-supervised video representation model, which includes three important modules – temporal-level augmentation, temporal memory bank,

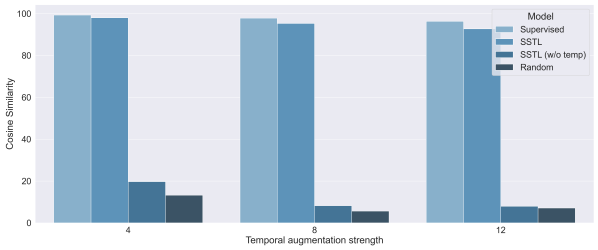

Figure 1: The extracted feature similarity of clips which applied with different degrees of temporal augmentations from the same input. The number of strength mean the buffer frames for potential frame cropping or dropping. As it becomes bigger, the augmentation will be stronger. "SSTL (w/o temp)" denotes our model trained without temporal augmentations. "Random" represents the model with random initialization.

and SSTL loss. We hope that the similarity between two clips which have a little difference on the temporal dimension becomes higher. So some novel temporal augmentations (i.e., temporal crop, dropout, and jitter) are introduced to create more difficulties for distinguishing the positive and negative pairs, which can improve the quality of learned representations. Considering the observation that the similarity between clips within the same videos but are temporally far-away usually relatively low, we propose the temporal memory bank which can store more than one slot for each video, and each slot only stores the feature of a video segment. The temporal memory bank not only provides more representative and accurate features of negative samples, but also help the model capture long-range temporal information across the entire video. With the statistical priors that the inter-video similarities are significantly higher than those of intra-video ones, we define the clip-to-video loss to encourage the distance between the clip and the video belonging to the same clip become closer in the latent space, which is the key component of the SSTL loss.

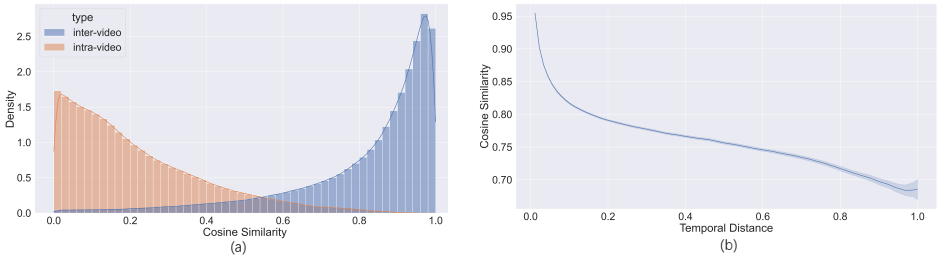

Figure 2: (a) The distribution of cosine similarities of intra-video and inter-video clips. (b) The relative temporal distances vs. feature similarities between clips with the same video. The shaded region shows the 95% confidence interval.

In summary, our main contributions are: 1) We introduce three simple but effective temporal augmentations to improve the quality of learned features; 2) We develop a novel temporal memory bank to capture long-range temporal information and provide more informative representations; 3) We design the SSTL loss function which includes two levels: clip-to-clip and clip-to-video; 4) Exhaustive experiments further demonstrate the proposed framework achieve significant performance on downstream tasks, especially on video retrieval tasks we obtain 100% relative improvement.

2 Related Work

Self-supervised learning from images Self-supervised learning or unsupervised learning leverages information from unlabeled data to train models. Recently, a number of self-supervised learning tasks are based on pretext tasks. These methods designed auxiliary tasks with pseudo training labels from the data itself to supervise the training of the network. Spyros *et al.*designed a pretext task which recognizes the rotation angle of the transformed images (Gidaris et al. (2018)). Another line of self-supervised learning is known as contrastive learning. These methods learn representations

by contrasting between congruent and incongruent representations of the same image or scene. The instance discrimination method (Wu et al. (2018)) presented an unsupervised feature learning approach to maximize distinction between instances via a novel non-parametric softmax formulation. CPC (Oord et al. (2018)) learns from continuous views, like the past and future or the sequential data in space. CMC (Tian et al. (2019)) apply contrastive learning to the multiview setting. Moco (He et al. (2020)) builds a dynamic dictionary with a queue and a moving-averaged encoder to facilitate contrastive unsupervised learning.

Self-supervised learning from videos Compared with 2D static images, the large number of videos with a temporal dimension can provide extra supervision signals. Meantime, a series of deep learning methods were proposed to focus on video understanding(Wang et al. (2016); Lin et al. (2019); Shao et al. (2020); Feichtenhofer et al. (2019); Feichtenhofer (2020)); Most of the prior work still focuses on various pretext task to learn self-supervised spatiotemporal representations. The rotation angles of the transformed video clips could be used as a supervisory signal(Jing et al. (2018)). In the work of Xu et al. (2019); Kim et al. (2019), the learning of spatiotemporal representation driven by predicting orders of video clips at temporal or spatial dimensions. Benaim et al. (2020) trains a deep network by detecting if a video is playing at a normal rate, or it is sped up. As shown in Han et al. (2019), 3D CNN models are trained by recurrently predicting future representations of spatiotemporal blocks. Yao et al. (2020) proposes to produce self-supervision signals about video playback rates from representation model learning. The approach of Han et al. (2020) trains the unsupervised model with a predictive attention mechanism over the set of compressed memories.

3 Our Approach

Our proposed framework SSTL learns representations in a self-supervised manner to make the learned feature representations have the desired characteristics that we observed from a fully-supervised model. In recent image-based self-supervised learning methods, contrastive learning has made a great success. We also choose to adopt contrastive learning in our framework but the learning is conducted by distinguishing the positive samples from a group of negative samples from the temporal dimension to capture temporal variations of videos. The challenge is how to create appropriate training sample pairs to provide supervisory signals. To achieve this goal, we propose three novel modules: temporal augmentations, temporal memory bank, and SSTL loss. Based on our observations on fully-supervised features in Figure 1, the learned feature similarity between two clips that are temporally close within the same video should large, while those of other clip pairs should have small similarities. We therefore design three novel temporal augmentations and require the cosine similarity between pairs of augmented clips from the same videos to be larger. The traditional memory bank in image-based contrastive learning usually adopted a container to store negative samples. As videos introduce the additional temporal dimension, the feature similarity decrease as the inter-clip temporal distance increases in Figure 2(b). With this observation, we propose the temporal memory bank that stores each video as a series of memory features in the memory slots, and each slot only stores the features of a clip covering a short time span of the videos. The features from the memory are more representative for creating negative training samples. We also find that the inter-video similarities are significantly higher than those of intra-video ones, as shown in Figure 2(a). Two possible strategies are adopted to define positive samples in our work: the pair with different data augmentations applied to the same clip, the clip and the video to which the same clip belongs. We call the combination of the above two losses as our SSTL loss.

3.1 Temporal Data augmentation

Data augmentation aims at transforming the clips without changing its semantic contents. It creates positive and negative pairs of clips as training supervisions. We consider two types of augmentations in our framework. The first type of augmentation involves transformations on the temporal dimension. We take three kinds of temporal transforms in our framework: temporal random cropping, temporal random drop, and playback rate distortion:

Playback rate distortion: Similar to random spatial resizing of images, we introduce the playback rate distortion ("resizing") along the temporal dimension. A random distortion factor from 1.0 to 1.25 is chosen as the target playback rate p. Then we randomly drop frames with probability $1 - 1/p$ and the maximum number of dropped frames is also limited.

Temporal random dropout. For each input clip, we randomly drop out a fixed number of non-continuous frames. This augmentation is inspired by our observation on temporal disturbance, which shows that dropping out some non-continuous frames would not much affect the semantic contents.

Temporal random crop. For each input clip, we choose a random subset of consequent 16 frames from all input frames, and this is the last part of the three temporal augmentations.

The second type of augmentation conducts spatial transformations of the video frames, which are similar to those of existing unsupervised learning approaches for still images and include random resizing, random cropping, color distortion, Gaussian blur, and random horizontal flip. To avoid different augmentations damage the underlying smoothness image structures of consecutive frames in videos, we make the spatial augmentation hyper-parameters to be fixed for all clips of a video.

3.2 TEMPORAL CONTRASTIVE LEARNING

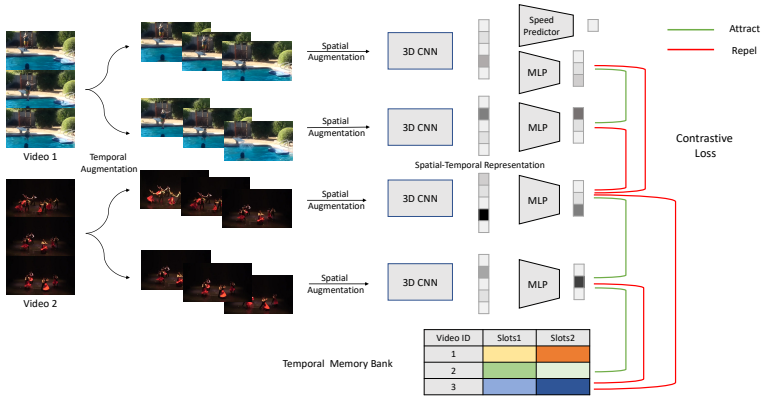

Figure 3: **Overview of the proposed self-supervised temporal learning (SSTL) framework.** We randomly one clip from a short video, and then get two processed clips through applying twice temporal augmentation and spatial augmentation for each clip. Finally, we feed it to a 3D backbone with an MLP head and a speed predictor head. The contrastive loss is used to train the network to maximize agreement between differently augmented views of the same clip and minimize agreement between different clips. Some repeated connection lines are not drew in the figure.

The framework of the proposed approach is illustrated in Fig 3. Given a video dataset $V = \{v^i | i = 1, \ldots, N\}$, we randomly sample some clips of shape $\mathbb{R}^{(T+e)\times H\times W}$ from different videos to create a training mini-batch. Here T, H, W, e denote clip temporal span, frame height, frame weight, and the buffer frames for potential frame cropping or dropping, which will be used in the following augmentations. Given each sampled clip from the videos, we apply a series of specifically designed temporal and spatial augmentations to the clip and obtain pairs of training clips, denoted v_x^i and v_y^i. The clip's temporal range becomes T because temporal augmentations will drop some frame of it and make the number of remaining frames equal to the number of model input. Our encoder $f(\cdot)$ is composed of a base video backbone (e.g., R3D-18 (Tran et al., 2018; Xie et al., 2018)) and a non-linear projection head. It encodes the input augmented clips into representation vectors: $\boldsymbol{h}_{i,x} = f(v_x^i)$. Other than the contrastive loss which is computed with the extracted features and our temporal memory bank, we also introduce a playback rate prediction branch for the encoder to estimate the input clip's playback rate as an auxiliary task.

Temporal Memory bank Unlike the common memory bank with the "key-value" structure, where one video corresponds to only one feature vector. Considering the temporal smoothness of the contents in videos, we design a temporal-oriented memory bank for creating more informative training pairs. Our memory bank is a two-dimensional matrix, which contains N rows and L columns. Each row means one video in the dataset and each column denotes a fixed temporal interval in one video (e.x. from 0 frames to 60 frames), and each memory cell stores the representation of a video segment. The l-th column is to represent the temporal interval $((l-1)/LT, l/LT)$ in the T-frames video. We define t/T as the clip's temporal position where t represents the position of the center frame and T denotes the total frames of the video. When the memory bank is updated with the new extracted feature from the clip, there are two updating strategies: 1) If the temporal interval of the cell contains the clip's temporal position, it will update with the fixed momentum. 2) The second

strategy conducts memory feature updating not only at the input clip's corresponding temporal interval but also neighboring intervals' features with an updating rate inversely proportionally to the distance to the clip. The memory updating rule for l-th column in our memory bank is defined as:

$$\theta_l^i \leftarrow \frac{\tilde{m}\theta_l^i + (1-\tilde{m})h_i}{\left\|\tilde{m}\theta_l^i + (1-\tilde{m})h_i\right\|_2} \tag{1}$$

$$\tilde{m} = \begin{cases} m & , \text{if } t/T \in ((l-1)/L, l/L) \\ (1-|(2l-1)/2L - t/T|)^c m & , \text{if } t/T \notin ((l-1)/L, l/L), \end{cases} \tag{2}$$

where $m \in [0,1]$ is a momentum coefficient, $c \in (0,1)$ denotes attenuation coefficient, θ_l^i refers to the feature stored in n-th row and l-th column of the memory bank. We empirically set c to be 0.6, and L to be 2 when trained in K-400. Apart from the input videos, the features from the rest videos in any column of the memory bank can be used as a negative sample when calculating contrastive loss.

Contrastive Learning We propose to adopt SSTL loss for supervising the learning of the video representations: clip-to-clip and clip-to-video. For clip-to-clip contrastive loss, we adopt InfoNCE, which is defined as:

$$\mathcal{L}_{i,x} = -\log \frac{\exp\left(\text{sim}\left(\boldsymbol{h}_{i,x}, \boldsymbol{h}_{i,y}\right)/\tau\right)}{\sum_{k=1}^{N} \mathbb{1}_{[k\neq i]} \exp\left(\text{sim}\left(\boldsymbol{h}_{i,x}, \boldsymbol{h}_{k,x}\right)/\tau\right) + \exp\left(\text{sim}\left(\boldsymbol{h}_{i,x}, \boldsymbol{h}_{k,y}\right)/\tau\right)}, \tag{3}$$

$$\mathcal{L}_{ctr} = \frac{1}{2N}\left(\sum_{i}^{N} \mathcal{L}_{i,x} + \sum_{i}^{N} \mathcal{L}_{i,y}\right), \tag{4}$$

where $\mathbb{1}_{[k\neq i]} \in \{0,1\}$ is an indicator function, which outputs 1 if $k \neq i$, and τ is a temperature parameter. The cosine similarity is adopted to measure inter-clip similarity $sim\left(\boldsymbol{a}, \boldsymbol{b}\right) = \boldsymbol{a}^\top \boldsymbol{b} / \left\|\boldsymbol{a}\right\| \left\|\boldsymbol{b}\right\|$. The overall loss L_{ctr} is computed with all positive pairs in a mini-batch, both (x, y) and (y, x).

The video-level feature can be obtained by average pooling the features from all cells in one row of the temporal memory bank. It can be seen as a positive sample to provide a supervision signal for the extracted feature from the same video. The clip-to-video loss is defined as:

$$\mathcal{L}_{video} = 1 - \frac{h_i\left(\frac{1}{L}\sum_{n=1}^{L} \theta_n^i\right)^\top}{\left\|h_i\right\| \left\|\frac{1}{L}\sum_{n=1}^{L} \theta_n^i\right\|} \tag{5}$$

The loss function is based on cosine similarity, and it encourages the distance between the clip and the video belonging to the same clip become closer in the latent space. Another intuitive aspect of optimizing this loss is to group representations of the clips sampled from the same video.

Playback rate prediction We also propose an auxiliary task on predicting the playback rate s of the input video clips, which varies from 1.0× to 1.25× and we have the ground-truth playback rate of the augmented video clips. Such a playback rate prediction task encourages the neural network to learn generic transferable video representations. We assume that the network can only recognize the actual video speed when it understands the potential video information. Our playback rate prediction branch is composed of two fully-connected layers and one ReLU non-linearity layer in-between. We use the flatten features of the backbone's last layer as the module's input, and the output is the predicted playback rate. The branch is trained with an MSE loss as $\mathcal{L}_{spd} = \frac{1}{2}(\tilde{s} - s)^2$

The overall loss By jointly optimizing the video speed regression objective and the contrastive objective, the final training loss is defined as

$$\mathcal{L} = \lambda_{ctr}\mathcal{L}_{ctr} + \lambda_{spd}\mathcal{L}_{spd} + \lambda_{video}\mathcal{L}_{video} \tag{6}$$

Where λ_{ctr}, λ_{spd}, λ_{video} are the hyperparameter to balance different training objectives. In this work, λ_{ctr}, λ_{spd} and λ_{video} are set to 1.0 , 0.1 and 0.2 respectively.

4 EXPERIMENTS

In this section, we conduct experiments to evaluate our proposed SSTL's effectiveness for action recognition and the quality of the learned spatiotemporal feature representations.

Method	Backbone	Input Size	Training Data	UCF101	HMDB51
Scratch	R3D-18	112×112	-	51.6	19.3
3DRotNet *[2018 Arxiv]*	R3D-18	112×112	Kinetics-400	62.9	33.7
VCOP *[2019 CVPR]*	R3D-18	112×112	UCF101	64.9	29.5
ST-Puzzle *[2019 AAAI]*	R3D-18	112×112	Kinetics-400	65.8	33.7
DPC *[2019 Arxiv]*	R3D-18(*)	128×128	Kinetics-400	68.2	34.5
PRP *[2020 CVPR]*	R3D-18	112×112	Kinetics-400	<u>66.5</u>	<u>29.7</u>
VIE *[2020 CVPR]*	R3D-18(†)	112×112	Kinetics-400	75.5	44.6
Speednet *[2020 CVPR]*	I3D	224×224	Kinetics-400	66.7	43.7
MemDPC *[2020 ECCV]*	R2D3D-34	224×224	Kinetics-400	78.1	41.2
VRLPP *[2020 ECCV]*	R(2+1)D-18	112×112	Kinetics-400	77.1	36.6
SSTL	R3D-18	112×112	Kinetics-400	**79.1**	**49.7**

Table 1: Performance comparison of self-supervised methods for Spatio-temporal representation learning on UCF101 and HMDB51 dataset (Pre-trained on RGB modality only). *The input video clips contain 64 frames. † The backbone is Resnet(V2)3D-18.

4.1 DATASETS

We conduct experiments on four action recognition datasets, Kinetics-400 (Carreira & Zisserman (2017)), UCF101 (Soomro et al. (2012)), HMDB51 (Kuehne et al. (2011)). Kinetics-400 (K-400) is a large-scale action dataset that has about 300k videos of 400 classes. In this work, we use the training split (around 240k video samples) without any annotation as the pre-training dataset, to validate the proposed approach. UCF101 contains 13K videos spanning of 101 human action classes. The dataset is divided into 3 splits and following prior works, we use split 1 for pre-training and evaluation in this paper. HMDB51 is a relatively small action recognition dataset, which consists of about 7k videos of 51 classes, we still use split 1 for downstream task evaluation.

4.2 IMPLEMENTATION DETAILS

Self-Supervised Learning In our experiment, we use R3D-18 (Tran et al. (2018)) as the backbone. When pre-training on the K-400 dataset, we sample consecutive 16-frame video clips from the videos. For data augmentation, we firstly resize the short size of the input frames to 112. Then we randomly crop the video clip to 112×112. The batch size is 2048 on 32 GPUs and we use SGD as optimizer with an initial learning rate 2.0, momentum 0.9 and weight decay 10^{-6}. A cosine learning rate decay strategy (Loshchilov & Hutter (2016)) with one annealing cycle was used to pre-train for 300 epochs. Note that, we apply a max pooling operation with a stride of 2 after the first convolution layer to have a batchsize of 64 which is $8 \times$ than before. The temperature used in the contrastive loss is 0.1.

Transfer Learning During the fine-tuning stage for action recognition, weights of convolutional layers are retained from the self-supervised learning networks and weights of fully-connected layers are randomly initialized. The way of image pre-processing is the same as the previous self-supervised pre-training stage. We also use SGD with an initial learning rate of 0.02 and a weight decay of 0.01. We train the models for 120 epochs, with the learning rate multiplied by 0.15 at the 50, 80 and 100 epochs on 16 GPUs.

Evaluation We sample 5 clips uniformly from each video in the testing set of UCF101 and HMDB51. For each clip, the center crop is applied. The predicted label of each video is generated by averaging the softmax probabilities of all clips in the video.

4.3 ACTION RECOGNITION

In this section, We compare our approach with other methods on the action recognition task. We performed self-supervised pre-training on Kinetics-400, and then we fine-tune all the layers for action recognition on target datasets. As shown in Table 1, our model achieves state-of-the-art results with the same amount of un-annotated data and RGB-only clips on both UCF101 and HMDB51 datasets. With the R3D-18 backbone, our method achieves 22.9% (74.5% vs. 51.6%) and 26.6% (44.5% vs. 19.3%) improvement over the random initialization. Our approach also obtains 6.1% compared with the current best-performing method PRP(Yao et al. (2020)).

Methods	Network	Top1	Top5	Top10	Top20	Top50
VCOP	R3D-18	14.1	30.3	40.4	51.1	66.5
PRP	R3D-18	22.8	38.5	46.7	55.2	69.1
VRLPP	R3D-18	23.8	38.1	46.4	56.6	69.8
Speednet	S3D-G	13	28.1	37.5	49.5	65
MemDPC	R2D3D-34	20.2	40.4	52.4	64.7	-
Scratch	R3D-18	9.9	18.9	26	35.5	51.9
SSTL (w/o temp)	R3D-18	25.4	47.3	57.9	68.7	69.5
SSTL	R3D-18	**44.5**	**57.4**	**63.5**	**70.0**	**79.0**

Table 2: Comparison with state-of-the-art methods for nearest neighbor retrieval on the UCF101 dataset.

Methods	Network	Top1	Top5	Top10	Top20	Top50
VCOP	R3D-18	7.6	22.9	34.4	48.8	68.9
PRP	R3D-18	8.2	25.8	38.5	53.5	75.9
VRLPP	R3D-18	9.6	26.9	41.1	**56.1**	**76.5**
MemDPC	R2D3D-34	7.7	25.7	40.6	57.7	-
Scratch	R3D-18	6.7	18.3	28.3	43.1	67.9
SSTL (w/o temp)	R3D-18	10.9	29.1	41.1	54.3	68.4
SSTL	R3D-18	**21.8**	**35.7**	**44.2**	54.7	69.4

Table 3: Comparison with state-of-the-art methods for nearest neighbor retrieval task on HMDB51 dataset.

4.4 Nearest Neighbor Retrieval

Our model SSTL can also be directly used as a feature extractor to encode the representation of the video clips. To evaluate the learned representations, we validate them via nearest-neighbor video retrieval. As it does not need annotations for network finetuning, its performance mainly depends on the pretrained models. We follow the experimental settings in (Xu et al. (2019); Luo et al. (2020); Wang et al. (2020)). We conduct our experiments on UCF101 and HMDB51 and sample 10 clips per video, in which each clip contains 16 consecutive frames. Unlike other methods that used the max-pooling on the last convolution layer's output, we directly take the features from the last full-connected layer as our encoded representations. For each clip in the testing set, we calculate the cosine distance between the representations of the input clip and all clips in the training set. If the class of the test clip exists in the classes of top-k nearest training clips, it is considered to be correctly retrieval. We consider varying k of 1, 5, 10, 20, 50 in the following experiments. As shown in Tables 2 and 3, our method achieves about **100%** relative improvement compared with previous PRP (Yao et al. (2020)) and VRLPP (Wang et al. (2020)) in most cases on the two datasets.

4.5 Ablation Study

In this section, we conduct extensive experiments to validate the effectiveness of individual components of proposed SSTL framework, including temporal augmentation, memory bank, playback rate prediction, and other training settings. Note that, we keep the setting identical to our final solution and change one component at a time in each experiment.

Spatial-Temporal Augmentations & Playback Rate Prediction. To study the impact of temporal data augmentation, we consider augmentations used in our framework. Our data augmentations mainly consist of two types: spatial/appearance transformation and temporal transformation (including temporal cropping, temporal dropping, playback rate distortion). Table 4 shows that stronger augmentation substantially improves the transfer learning evaluation of the learned unsupervised models. The Temporal augmentations greatly boosts the discriminativeness of learned representations, a 4.6% boost can be observed from 69.5% to 74.5%. Among the three types of temporal augmentations, we observe that the temporal random cropping leads to most significant performance improvement of about 3.5%. From the Tabel 2 and Tebel 3, we can also find that the application of temporal augmentations boosts up the Top1 retrieval accuracy by about 19.1% in UCF101 and 10.9% in HMDB51. It forces the model to extract similar high-level semantic features no matter how the playback rate is updated. But the too strong temporal augmentation might also damage the performance by about 2%. We suppose that a difference in input content has changed the seman-

Spatial	Temporal			Speed Pred.	UCF101
Color jittering	Input frames	Dropped frames	Speed distortion		
×	16 + 0	0	×	×	66.3
✓	16 + 0	0	×	×	69.5
✓	16 + 8	0	×	×	73.0
✓	16 + 12	4	×	×	74.1
✓	16 + 16	6	×	×	72.1
✓	16 + 12	4	✓	×	74.5
✓	16 + 12	4	✓	✓	**75.2**

Table 4: Evaluation about different augmentation configurations of R3D-18 on UCF101 datasets. The "Input frames" = A + B represents that the number of original input frames is A and the buffer frames for potential frame cropping or dropping is B. So we actually sample a consecutive A + B frames clip for training.

Network	Memory Size	Temporal-related module	Clip-to-video loss	UCF101
R3D-18	-	×	×	70.8
R3D-18	2048	×	×	72.0
R3D-18	4096	×	×	69.9
R3D-18	2048	✓	×	72.8
R3D-18	2048	✓	✓	**73.9**

Table 5: Results of our model with different memory bank settings. All models are trained on K400 with the same iterations. The representation is evaluated by training an action classifier on UCF101.

tic information, but we still let the representations to be similar. Table 4 also shows that with the playback rate prediction auxiliary task, the performance is further improved by 0.7%.

Memory Bank. As shown in Table 5, the following phenomena can be observed: First, the proposed memory bank boost the learned representation by around 1.2% and the optimal memory size for UCF101 is 2048; Second, the temporal memory bank provides a 0.8% improvement compared with the common memory bank; Third, as a result of our memory bank can provide video-level feature, it can be used to calculate clip-to-video loss which boost the accuracy by about 1%. To sum up, the designed memory bank can improve the performance by a large margin – about 3%.

Projection output dimension and Batch Size. Figure 4 shows the impact of batch size when the models are trained with different batch sizes and projection output dimensions. We observe that with larger batch sizes have significant advantages over the smaller ones. In Fig. 4 the optimal output dimension is 512, which is the same as representation dimension before the projection layer. We find that the larger output dimension cannot bring more information and might result in overfitting; on the other hand, the smaller output may lose some spatial-temporal information.

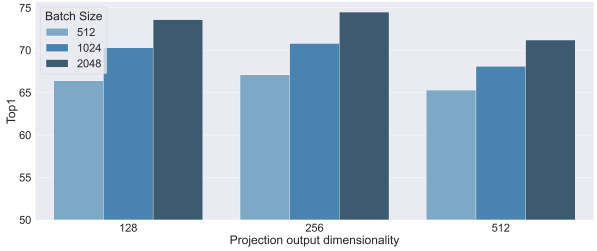

Figure 4: Transfer learning evaluation of representations with different output dimensions and batch size. Each bar is a single run from scratch. The representation before the projection is 512-dimensional here.

5 Conclusion

In this work we propose a novel framework for temporal representation learning. We carefully study its components and show the effects of different choices by conducting extensive ablation experiments. By combining our findings, we improve considerably over the previous stoa methods on the downstream tasks. Our results show that learning high-quality temporal representation is very helpful for video understanding tasks to achieve good performance. In future work, we plan to apply SSTL to a larger set of unlabeled videos and conduct the experiments with deeper networks.

REFERENCES

Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9922–9931, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Carl Doersch, Abhinav Gupta, and Alexei A Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE international conference on computer vision*, pp. 1422–1430, 2015.

Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 203–213, 2020.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pp. 6202–6211, 2019.

Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *arXiv preprint arXiv:2008.01065*, 2020.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

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

Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 8545–8552, 2019.

Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International Conference on Computer Vision*, pp. 2556–2563. IEEE, 2011.

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7083–7093, 2019.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *arXiv preprint arXiv:2001.00294*, 2020.

Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pp. 69–84. Springer, 2016.

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

Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020.

Hao Shao, Shengju Qian, and Yu Liu. Temporal interlacing network. In *AAAI Conference on Artificial Intelligence*, pp. 11966–11973, 2020.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019.

Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pp. 6450–6459, 2018.

Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. *arXiv preprint arXiv:2008.05861*, 2020.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pp. 20–36. Springer, 2016.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, pp. 305–321, 2018.

Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 10334–10343, 2019.

Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6548–6557, 2020.

Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *European conference on computer vision*, pp. 649–666. Springer, 2016.

Chengxu Zhuang, Tianwei She, Alex Andonian, Max Sobol Mark, and Daniel Yamins. Unsupervised learning from video with deep neural embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9563–9572, 2020.

A Appendix

A.1 Details of Figure 2

We randomly sample 50k videos from Kinetics-600, and uniformly sample 40 clips from each video. Then we use the model pretrained in Kinetics-600 to extract the features for the sampled clips. When we compute the inter-video similarity, we calculate the mean value of the cosine similarity for all features within the same video. The intra-video similarity is computed between the features from one video and one random different video.

A.2 UCF classes with largest gain using temporal augmentations

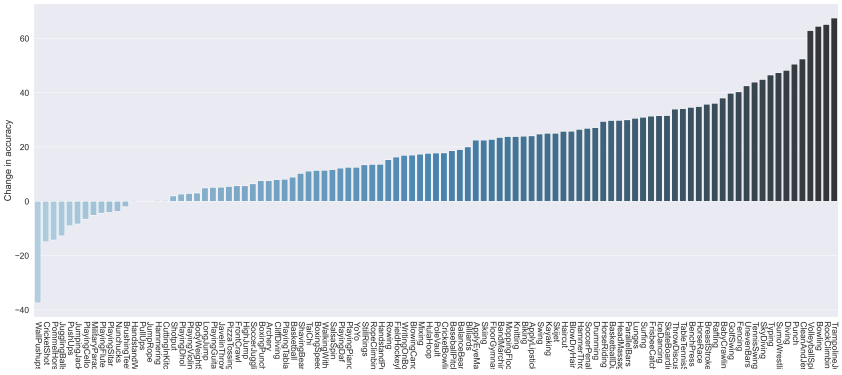

Figure 5: UCF classes with different gain using temporal augmentations. The plot shows UCF per-class accuracy difference between the two pretrained models tested. It suggests that our temporal augmentations are most useful for classes that require understanding motion, such as “Diving” and “VolleyballSpiking.”.