Removing the Background by Adding the Background: Towards Background Robust Self-supervised Video Representation Learning

Jinpeng Wang[1,2] * Yuting Gao[2] * Ke Li[2] Yiqi Lin[1] Andy J. Ma [1] †
Hao Cheng [2] Pai Peng [2] Feiyue Huang [2] Rongrong Ji[3,4] Xing Sun[2]

[1]Sun Yat-sen University [2]Tencent Youtu Lab [3]Xiamen University [4]Peng Cheng Laboratory

Abstract

Self-supervised learning has shown great potentials in improving the video representation ability of deep neural networks by getting supervision from the data itself. However, some of the current methods tend to cheat from the background, i.e.*, the prediction is highly dependent on the video background instead of the motion, making the model vulnerable to background changes. To mitigate the model reliance towards the background, we propose to* remove the background impact by adding the background. *That is, given a video, we randomly select a static frame and add it to every other frames to construct a distracting video sample. Then we force the model to pull the feature of the distracting video and the feature of the original video closer, so that the model is explicitly restricted to resist the background influence, focusing more on the motion changes. We term our method as* Background Erasing *(BE). It is worth noting that the implementation of our method is so simple and neat and can be added to most of the SOTA methods without much efforts. Specifically, BE brings 16.4% and 19.1% improvements with MoCo on the severely biased datasets UCF101 and HMDB51, and 14.5% improvement on the less biased dataset Diving48.*

1. Introduction

Convolutional neural networks (CNNs) have achieved competitive accuracy on a variety of video understanding tasks, including action recognition [20], temporal action detection [63] and spatio-temporal action localization [55]. Such success relies heavily on manually annotated datasets, which are time-consuming and expensive to obtain. Meanwhile, there are numerous unlabeled data that are instantly available on the Internet, drawing more and more researchers' attention from the community to utilize off-the-shelf unlabeled data to improve the performance of CNNs by self-supervised learning.

Figure 1: **Illustration of the *background cheating***. In the real open world, *an* action can happen at *various* locations. Current models trained on the mainstream datasets tend to give predictions simply because it sees some background cues, neglecting the fact that motion pattern is what actually defines an "action".

Recently, self-supervised learning methods have been developed from the image field to the video field. However, there are big differences between the mainstream video dataset and the mainstream image dataset. Li et al.[29] and Girdhar et al.[14] point out that the current commonly used video datasets usually exist large *implicit biases* over scene and object structure , making the temporal structure become less important and the prediction tends to have a high dependence on the video background. We name this phenomenon as *background cheating*, as is shown in Figure 1. For example, a trained model may classify an action as *playing soccer* simply because it sees the field, without really understanding the *cartwheel* motion. As a result, the model is easily to overfit the training set, and the learned feature representation is likely to be scene-biased. Li et al.[29] reduce the bias by resampling the training set, and Wang et al.[53] propose to pull actions out of the context by training a binary classifier to explicitly distinguish action samples and conjugate samples that are contextually similar to action samples but contains different action.

In this work, to hinder the model from *background cheating* and make the model generalize better, we present to

*The first two authors contributed equally. This work was done when Jinpeng was in Tencent Youtu Lab.

†Corresponding Author. Email: majh8@mail.sysu.edu.cn.

reduce the impact of the background by adding the background and encourage the model to learn consistent feature w/ or w/o the operation. Specifically, given a video, we randomly select a static frame and add it to every other frames to construct a distracting video, as is shown in Figure 3. Then we force the model to pull the feature of the distracting video and the feature of the original video together by consistency regularization. In this way, we made a disturbance to the video background and require its feature to be consistent with the original video, achieving the purpose of making the model not be excessively dependent on the background, thereby alleviating the *background cheating* problem.

Experimental results demonstrate that the proposed method can effectively reduce the influence of the *background cheating*, and the extracted representation is more robust to the background bias and have stronger generalization ability. Our approach is simple and incorporate it into existing self-supervised video learning methods can bring significant gains.

In summary, our main contributions are twofold:

- We propose a simple yet effective video representation learning method that is robust to the background.
- The proposed approach can be easily incorporated with existing self-supervised video representation learning methods, bringing further gains on UCF101[41], HMDB51 [27] and Diving48[30] datasets.

2. Related Work

2.1. Self-supervised Learning for Image

Self-supervised learning is a generic learning framework which gets supervision from the data itself. Current methods can be grouped into two types of paradigms, *i.e.*, constructing pretext tasks or constructing contrastive learning.

Pretext tasks. These methods focus on solving surrogate classification tasks with surrogate labels, including predicting the rotation angle of image[13], solving the jigsaw puzzle[35], coloring image[62] and predicting relative patches[35], etc. Recently, the type of image transformation also be used as a surrogate[61].

Contrastive learning. Another mainstream method is based on contrastive learning, which regards each instance as a category. Early work [11] directly used each sample in the dataset as a category to learn a linear classifier, but this method will become infeasible when the number of samples increases. To alleviate this problem, Wu et al. [56] replace the classifier with a memory bank storing previously computed representations and then use a noise contrastive estimation [15] to compare instances. MoCo [21] stores the representations from a momentum encoder and achieves great success. In contrast, Ye et al. [59] propose to use a mini batch to replace the memory bank. SimCLR [8] shows that the memory bank can be entirely replaced by a large batch size.

2.2. Self-supervised Video Representation Learning

Recent years, self-supervised learning has been expanded into the video domain and attracts a lot interests.

Pretext tasks. The majority of the prior work explore natural video properties as supervision signal. Among them, temporal order is one of the most widely-used property, such as, the arrow of time [54], the order of shuffled frames [34], the order of video clip [57] and the playback rate of the video [1, 58]. Besides the temporal order, the spatio-temporal statistics are also used as supervision. For example, pxiel-wise geometry information [12], space-time cubic puzzles [26, 32] and the optical-flow and the appearance statistics [49]. In addition, DynamoNet[10] predicts future frames by learning dynamic motion filter, which is pre-trained on a large-scale dataset Youtube-8M. More recently, Buchler et al. [6] and ELO [38] propose to ensemble multiple pretext task based on reinforcement learning.

Contrastive learning. Contrastive learning is introduced into the field of video representation learning by TCN [40], which uses different camera views as positive samples. IIC [43] proposes an inter-intra mulit-modal contrastive framework based on the Contrastive Multiview Coding [44]. CoCLR [19] takes the advantage of the natural correlation between the RGB and the Optical Flow modalities to select the negative samples in the memory bank. GDT [33] achieves great success by using tens of millions data for pre-training with multi-modal constrastive leanring.

It is worth to mention that while all methods mentioned above focus on designing specific tasks, we present a generalized constraint term that can be integrated into any existing self-supervised video representation learning approach.

2.3. Background Biases in Video

Current widely used video datasets have serious bias towards the background[29, 14], which may misleads the model using just the static cues to achieve good results. For example, only using three frames during training, TSN[52] can achieve 85% accuracy on UCF101. Therefore, using these datasets for training can easily cause the model making background biased predictions.

In order to mitigate the background bias, Li et al.[30] resample the original dataset to generate a less biased dataset Diving48 for the action recognition task. Wang et al.[53] use conjugate samples that are contextually similar to human action samples but do not contain the action to train a classifier to deliberately separate the action from the context. Choi et al.[9] propose to detect and mask actors with a human detector and further present a novel adversarial loss for debasing. In this work, we try to debias through consis-

tency constraint, which is simple but effective and does not need additional costs.

3. Methodology

In this section we introduce the proposed *Background Erasing* (BE) method. We first give an overall description of BE, and then introduce how to integrate BE into existing self-supervised methods.

3.1. Overall Architecture

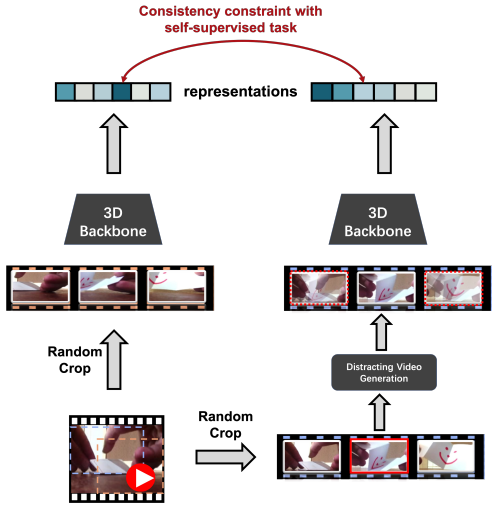

Figure 2: **The framework of the proposed method BE.** A video is first randomly cropped spatially, then we generate the distracting video by adding a static frame upon other frames. The model is trained by a existing self-supervised task together with a consistency constraint, with the goal of pulling the feature of the original video and that of the distracting video closer. (Best viewed in color).

The framework of the proposed BE is shown in Figure 2. For each input video x, we first randomly crop two fixed-length clips from different spatial locations, denoted as x^o and x^v. Suppose we have a basic data augmentation set $\mathcal{A}$, from which we sample two specific operations a^1 and a^2, and operate on x^o and x^v respectively. In this way, the input clips have different distribution in the pixel level but are consistent in the semantic level. Afterwards, x^o is directly fed into the 3D backbone to extract the feature representation and we denote this procedure as $F(x^o;\theta)$, where θ represents the backbone parameters. For x^v, we first generate a distracting counterpart x^d for it, which has the interference of added static frame noise but the semantics remains the same. The output feature maps of x^o and x^d are represented by $f_{x^o}, f_{x^d} \in \mathbb{R}^{C\times T\times H\times W}$. C is the number of channel and T is the length of time dimension. W and H are spatial size. At last, the extracted features f_{x^o}, f_{x^d} are pulled closer within the existing self-supervised methods.

3.2. Background Erasing.

In the video representation learning, sometimes the statistical characteristics of the background will drown out the motion features of the moving subject. Thus it is easy for the model to make predictions based only on the background information. And the model is easy to overfit to the training set and has poor generalization on the new dataset.

Background Erasing(BE) is proposed to remove the negative impact of the background by adding the background. Specifically, for a video sequence x, we randomly select one static frame and add it as a spatial background noise to every other frames to generate a distracting video, in which each frame $\hat{x}$ is obtained by the following formula:

$$\hat{x} = (1-\lambda)\cdot x^{(j)} + \lambda \cdot x^{(k)}, j \in [1, T] \tag{1}$$

where λ is sampled from the uniform distribution $[0, \gamma]$, $x^{(j)}$ means the j-th frame of x, k denotes the index of the randomly selected frame and T is the length of the video sequence x. BE operation is applied to x^v, and the generated distracting video clip x^d has a background perturbation on the spatial dimension, but the motion pattern is basically not changed, as shown in Figure 3.

Furthermore, it is easy to prove that the *time derivative* of x^d is a linear transformation of the *time derivative* of x^v, formally:

$$\frac{\mathrm{d}((1-\lambda)x^v + \lambda\delta)}{\mathrm{d}t} = (1-\lambda)\frac{\mathrm{d}x^v}{\mathrm{d}t} \tag{2}$$

where δ represents the result of repeating the selected frame $x^{(k)}$ T times along the time dimension. Previous works[3, 5, 4, 51] have shown that the *time derivative* of a video clip is an important information for action classification, thus, the property that BE maintains the linear transformation of such information is very crucial.

Afterwards, we force the model to pull the feature of x^o and the feature of x^d closer, which will be introduced in details later. Since x^o and x^d resemble each other in the motion pattern but differentiate each other in spatial, when the features of x^o and x^d are brought closer, the model will be promoted to suppress the background noise, yielding video representations that are more sensitive to motion changes. We have tried a variety of ways to add background noise, results are shown in Table 4. Experimental results demonstrate that the intra-video static frame, *i.e.*, BE, works best overall. Meanwhile, we have also tried to add various data augmentations to the selected intra-video static frame to introduce more disturbance, but there is no positive gain.

3.3. Plug-and-Play

Using BE solely for optimization will make the model fall into a trivial solution easily. Therefore, we integrate BE

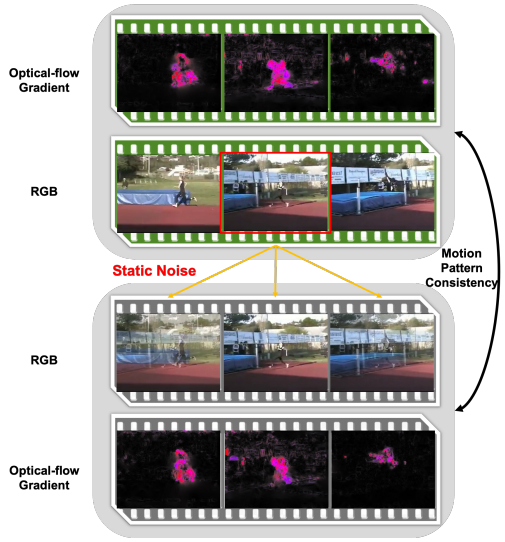

Figure 3: **Distracting Video Generation**. One intra-video static frame is randomly selected and added to other frames as *Noise*. The background of the generated distracting video has changed, but the optical flow gradient is basically not changed, indicating that the motion pattern is retained.

into the existing self-supervised methods, specifically, we adopt two paradigms, handcrafted pretext and contrastive.

3.3.1 Pretext Task

Most pretext tasks can be formulated as a multi-category classification task and optimized with the cross-entropy loss. Specifically, each pretext will define a transformation set R with M operations. Given an input x, a transformation $r \in R$ is performed, then the convolutional neural network with parameters θ is required to distinguish which operation it is. The loss function is as follows:

$$\mathcal{L}_p = -\frac{1}{M}\sum_{r\in R}\mathcal{L}_{ce}(F(r(x);\theta), r), \tag{3}$$

where $\mathcal{L}_{ce}$ is Cross Entropy.

Plugged-in BE. For handcrafted pretext task, we use a consistency regularization term to pull the feature of x^o closer to the feature of x^d, and make them consistent in the temporal dimension. Formally,

$$\mathcal{L}_{be} = ||\psi(f_{x^o}) - \psi(f_{x^d})||^2 \tag{4}$$

where ψ is an explicit feature mapping function that project features from $C \times T \times H \times W$ to $C \times T$. We use spatial global max pooling since x^o and x^d have different pixel distribution due to random cropping. In this fashion, we force the max response at each time dimension being consistent. And the final loss is:

$$\mathcal{L} = \mathcal{L}_p + \beta\mathcal{L}_{be} \tag{5}$$

where β is a hyperparameter that controls the importance of the regularization term. In our experiments, β is set to 1.

3.3.2 Contrastive Learning

Contrastive learning [16] aims to learn an invariant representation for each sample, which is achieved by maximizing similarity of similar pairs over dissimilar pairs.

Plugged-in BE. Given a video dataset $\mathcal{D}$ with N videos $\mathcal{D} = \{x_1, x_2, ..., x_N\}$, for each video x_i, we randomly sample once in each epoch, obtaining x_i^o and x_i^d. In order to add a consistency constraint between x^o and x^d, we directly treat their features $f(x^o)$ and $f(x^d)$ as positive pairs instead of using MSE loss. Specifically, assume there is a projection function ϕ, which consists of a spatio-temporal max pooling and a fully connected layer with D dimension. Then the high level feature can be encoded by $z_x = \phi(f(x))$. Given a particular video x_i and clip sampling function s, the negative set $\mathcal{N}_{1i}$ is defined as: $\mathcal{N}_{1i} = \{s(x_n)|\forall n \neq i\}$, each element in $\mathcal{N}_{1i}$ is a clip and represents an identity, then the InfoNCE[36] loss is improved as follows:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(z_{x_i^o}\cdot z_{x_i^d})}{exp(z_{x_i^o}\cdot z_{x_i^d}) + \sum_{n\in\mathcal{N}_{1i}} exp(z_{x_i^o}\cdot z_n)} \tag{6}$$

where $\cdot$ denotes the dot product. In this way, the optimization goal is *video-level* discrimination in essence.

However, in order to discriminate each instance in $\mathcal{D}$, there may exist many spatial details. In order to make the objective more challenge, we introduce *hard negatives*, the different video clips with augmentation a^1 but from the same video. In this way, the optimization goal changes from *video-level* into *clip-level*, which is based on the observation that different clips of the same video contain different motion patterns but similar background. The hard negative set $\mathcal{N}_{2i}$ for x_i is defined as: $\mathcal{N}_{2i} = \{x_i^h | x_i^h \neq x_i^o, x_i^h \in x_i\}$, and the overall negative set is $\mathcal{N}_i = \{\mathcal{N}_{1i} \cup \mathcal{N}_{2i}\}$. Then the final objective function is:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} log\frac{exp(z_{x_i^o}\cdot z_{x_i^d})}{exp(z_{x_i^o}\cdot z_{x_i^d}) + \sum_{n\in\mathcal{N}_i} exp(z_{x_i^o}\cdot z_n)} \tag{7}$$

For efficiency, we randomly select one hard negative sample from $\mathcal{N}_{2i}$ each iteration and we find more hard negative samples have a similar result experimentally.

4. Experiments

4.1. Implementation Details

Datasets. All the experiments are conducted on four video datasets, UCF101 [41], HMDB51 [27], Kinetics [25] and Diving48 [30]. The first three contain prominent bias, while Diving48 is less biased. UCF101 is a realistic video dataset

Method			Pretrain					Fine-tune	
Method(year)	**Backbone**	**Depth**	**Dataset(duration)**	**Frame**	**Res**	**Single-Mod**	***C/P***	**UCF101**	**HMDB51**
Supervised									
Random Init	I3D	22	✗	-	224	✓	-	60.5	21.2
ImageNet Supervised	I3D	22	ImageNet	-	224	✓	-	67.1	28.5
K400 Supervised	I3D	22	K400(28d)	-	224	✓	-	96.8	74.5
Self-supervised									
Shuffle [34] [ECCV, 2016]	AlexNet	8	UCF101(1d)	-	112	✓	*P*	50.2	18.1
VGAN [47] [NeurIPS, 2016]	VGAN	22	UCF101(1d)	-	112	✓	*P*	52.1	-
OPN [28] [ICCV, 2017]	Caffe Net	14	UCF101(1d)	-	112	✓	*P*	56.3	22.1
Geometry [12] [CVPR, 2018]	Flow Net	56	UCF101(1d)	16	112	✗	*P*	55.1	23.3
IIC [43] [ACM MM, 2020]	C3D	10	UCF101(1d)	16	112	✗	*C*	72.7	36.8
Pace [50] [ECCV, 2020]	R(2+1)D	23	K400(28d)	16	112	✓	*C*	77.1	36.6
3D RotNet [23] [2018]	C3D	10	K400(28d)	16	112	✓	*P*	62.9	33.7
3D RotNet + BE	C3D	10	K400(28d)	16	112	✓	*P*	**65.4**(2.5↑)	**37.4**(3.7↑)
ST Puzzles [26] [AAAI, 2019]	C3D	10	UCF101(1d)	48	112	✓	*P*	60.6	28.3
ST Puzzles + BE	C3D	10	UCF101(1d)	48	112	✓	*P*	**63.7**(3.1↑)	**30.8**(2.5↑)
Clip Order [57] [CVPR, 2019]	C3D	10	UCF101(1d)	64	112	✓	*P*	65.6	28.4
Clip Order + BE	C3D	10	UCF101(1d)	64	112	✓	*P*	**68.5**(2.9↑)	**32.8**(4.4↑)
MoCo [21] [CVPR, 2020]♢	C3D	10	UCF101(1d)	16	112	✓	*C*	60.5	27.2
MoCo + BE	C3D	10	UCF101(1d)	16	112	✓	*C*	**72.4**(11.9↑)	**42.3**(14.1↑)
CoCLR[19] [NeuIPS, 2020]	R3D	23	K400(28d)	32	128	✗	*C*	87.9	54.6
DPC [17][ICCW, 2019]	R3D	34	K400(28d)	64	224	✓	*P*	75.7	35.7
AoT [54] [CVPR, 2018]	T-CAM	-	K400(28d)	64	224	✓	*P*	79.4	-
Pace [50] [ECCV, 2020]	S3D-G	23	K400(28d)	64	224	✓	*C*	87.1	52.6
SpeedNet [1] [CVPR, 2020]	S3D-G	23	K400(28d)	64	224	✓	*P*	81.1	48.8
SpeedNet [1] [CVPR, 2020]	I3D	22	K400(28d)	64	224	✓	*P*	66.7	43.7
MoCo [21] [CVPR, 2020]♢	I3D	22	K400(28d)	16	224	✓	*C*	70.4	36.3
MoCo + BE	I3D	22	K400(28d)	16	224	✓	*C*	**86.8**(16.4↑)	**55.4**(19.1↑)
MoCo + BE	I3D	22	UCF101(1d)	16	224	✓	*C*	82.4	52.9
MoCo + BE	R3D	34	UCF101(1d)	16	224	✓	*C*	83.4	53.7
MoCo + BE	R3D	34	K400(28d)	16	224	✓	*C*	87.1	56.2

Table 1: Top-1 accuracy (%) of integrating BE as a regularization term to four existing approaches and compared with previous methods on the UCF101 and HMDB51 dataset. Single-Mod denotes Single-Modality, C/P represents Contrastive/Pretext task, ♢ means our implementation, K400 is short for Kinetics-400 and d represents day.

with 13,320 videos of 101 action categories. HMDB51 contains 6,849 clips of 51 action categories. Kinetics is a large scale action recognition dataset that contains 246k/20k train/val video clips of 400 classes. Diving48 consists of 18k trimmed video clips of 48 diving sequences.

Networks. We use C3D [45], R3D [20] and I3D [7] as base encoders followed by a spatio-temporal max pooling layer.

Default Settings. All the experiments are conducted on 8 Tesla V100 GPUs with a batch size of 64 under PyTorch[37] framework. We adopt SGD as our optimizer with momentum of 0.9 and weight decay of 5e-4.

Self-supervised Pre-training Settings. We pre-train the network for 50 epochs with the learning rate initialized as 0.01 and decreased to 1/10 every 10 epochs. The input clip consists of 16 frames, which is uniformly sampled from the original video with a temporal stride of 4. Then the sampled clip is resized to $16\times3\times112\times112$ or $16\times3\times224\times224$. The γ of Background Erasing is experimentally set to 0.3, and a larger value may result in excessive blur. The choice of temporal stride and γ is analysed in the supplementary. The basic augmentation set $\mathcal{A}$ contains random rotation less than 10 degrees and color jittering, and all these operations are applied in a temporal consistent way, that is, each frame of a video uses the same augmentation. The vector dimension D is 128.

Supervised Fine-tuning Settings. After pre-training, we transfer the weights of the base encoder to two downstream tasks, *i.e.*, action recognition and video retrieval, with the last fully connected layer randomly initialized. We fine-tune the network for 45 epochs. The learning rate is initialized as 0.05 and decreases to 1/10 every 10 epochs.

Evaluation Settings. For action recognition, following common practice[57], the final result of a video is the average of the results of 10 clips that are uniformly sampled from it during testing time.

4.2. Action Recognition

Comparison on common datasets. In this section, we integrate BE into three pretext tasks, *i.e.*, 3D RotNet, ST Puzzles and Clip Order, and one contrastive task, *i.e.*, MoCo[21], to verify the performance gains brought by BE. All the results shown in Table 1 are averaged over 3 dataset splits. We also report the result of the random initialized model and the result of the model pre-trained with all labels

Method	Pretrain	Single-Mod	Diving48
Supervised Learning			
R(2+1)D [46][CVPR, 2018]	✗	✓	21.4
R(2+1)D [46] [CVPR, 2018]	Sports1M	✓	28.9
I3D[7]◊[CVPR, 2017]	ImageNet	✓	20.5
I3D[7]◊[CVPR, 2017]	K400	✓	27.4
TRN [64] [ECCV, 2018]	ImageNet	✗	22.8
DIMOFS [2] [2018]	K400+Track	✗	31.4
GST [31] [ICCV, 2019]	ImageNet	✓	38.8
Att-LSTM [24] [CVPRW, 2019]	ImageNet	✓	35.6
GSM [42] [CVPR, 2020]	ImageNet	✓	40.3
CorrNet [48] [CVPR, 2020]	Sports1M	✓	44.7
Self-supervised Learning			
MoCo + BE (I3D)	Diving48	✓	58.3
MoCo + BE (R3D-18)	UCF101	✓	46.6
MoCo [21] ◊ (I3D)	UCF101	✓	43.2
MoCo + BE (I3D)	UCF101	✓	**58.8**(15.6↑)
MoCo [21] ◊ (I3D)	K400	✓	47.9
MoCo + BE (I3D)	K400	✓	**62.4**(14.5↑)

Table 2: Top-1 accuracy (%) of integrating BE into MoCo and compared to previous method on Diving48.

of ImageNet and Kinetics in a supervised manner for reference. It can be observed that plugging BE into three handcrafted pretext tasks can all bring improvements. Specifically, BE brings 2.5%/3.7% improvement with 3D RotNet, 3.1%/2.5% gain with ST Puzzle and 2.9%/4.4% improvement with Clip Order on UCF101/HMDB51. Further, when BE is introduced into MoCo, using the same backbone I3D and the same pretrain dataset Kinetics, it can bring 16.4% and 19.1% improvements on UCF101 and HMDB51 respectively, which is significant and nonnegligible.

Comparison on a less biased dataset. In this section, we fine-tune and test on a less biased Diving48, and the results are shown in Table 2. It can be observed that without using additional videos during pre-training, *i.e.*, pre-training and fine-tuning both on Diving48, MoCo enhanced with BE can achieve 58.3% top-1 accuracy using I3D backbone, which is far beyond the result of Kinetics supervised pre-training (27.4%). When Kinetics is also used in a self-supervised manner, the accuracy of our method can be further improved from 58.3% to 62.4%, which achieves state-of-the-art. It proves that our method can well alleviate the negative impact of scene bias in the training set, prevent the model from overfitting to the training set, *i.e.*, hinder the model from background cheating and obtain a more robust representation towards the motion. At the same time, it also indicates that given a dataset with less bias, the benefit from supervised pre-training on a large biased dataset is very small.

4.3. Video Retrieval

In this section, we evaluate BE on video retrieval tasks. Following the convention [49, 1], the network is fixed as a feature extractor after pre-training on the split 1 of UCF101. Then the videos from HMDB51 are divided into clips in units of 16 frames. All the clips in the training set constitute a *Gallery*, and each clip in the test set is used as a *query* to retrieve the most similar clip in the *Gallery* with cosine distance. If the category of the query appears in the K-nearest neighbors retrieved, then it is considered as a hit. It should be noted that in order to keep the scale of representations generated by each 3D architecture consistent, we replaced the original global average pooling with an adaptive max pooling, yielding representations with a fixed scale of $1024 \times 2 \times 7 \times 7$. We show the accuracy when $K = 1, 5, 10, 20, 50$ and compare with other self-supervised methods on HMDB51 in Table 3. It can be seen that when using the backbone C3D, combining BE with MoCo can bring a 0.7% improvement to top1 acc and a 2.2% improvement to top5 acc, which significantly exceeds the Clip Order and VCP with the same backbone. In addition, when using R3D as the backbone, our results surpass the current mainstream method Pace, which proves that the extracted representations are more discriminative.

Method	Net	1	5	10	20	50
Clip Order [57]	C3D	7.4	22.6	34.4	48.5	70.1
Clip Order [57]	R3D	7.6	22.9	34.4	48.8	68.9
VCP [32]	C3D	7.8	23.8	35.3	49.3	71.6
MemDPC [18]	R3D	7.7	25.7	40.6	57.7	-
Pace [50]	R3D	9.6	26.9	41.1	56.1	76.5
MoCo [21] ◊	C3D	9.5	25.4	38.3	52.2	72.4
MoCo + BE	C3D	10.2	27.6	40.5	56.2	76.6
MoCo + BE	I3D	9.3	28.8	41.4	57.9	78.5
MoCo + BE	R3D	**11.9**	**31.3**	**44.5**	**60.5**	**81.4**

Table 3: **Recall-at-topK (%).** Accuracy under different K values on HMDB51.

4.4. Variants of Distracting Video Generation

In this section, we conduct experiments to explore the effectiveness of different distracting video generation methods. We employ MoCo with I3D as the baseline and optimized with Eq. 7, all the experiments are pre-trained on the split 1 of the UCF101.

One main operation in the background erasing is to generate a distracting video with background noise while retaining the temporal semantics. In order to explore whether adding a static frame is the most effective operation, we compare it with another four common ways: (a).Gaussian Noise: add an identical White Gaussian Noise on each frame. (b).Video Mixup [22]: interpolate two videos frame by frame. (c).Video CutMix [60]: randomly replace one region of each frame with a patch from another frame. (d).Inter-Video Frame: randomly select one frame from another video, and add this static frame as noise to each frame of this video. (e).Our Intra-Video Frame: randomly select one frame from the video itself, and add this static frame as noise to each frame of this video. The results are shown in Table 4 and three observations can be obtained:

Method	UCF101	HMDB51
baseline	72.7	42.1
Gaussian Noise	73.2(0.5↑)	42.4(0.3↑)
Video Mixup	68.3(4.4↓)	38.1(4.0↓)
Video CutMix	71.2(1.5↓)	40.5(1.6↓)
Inter-Video Frame	77.4(4.7↑)	46.5(4.4↑)
Intra-Video Frame	**82.4**(9.7↑)	**52.9** (10.8↑)

Table 4: Top-1 accuracy (%) of different distracting video generation methods on UCF101 and HMDB51.

i. Video Mixup and Video CutMix perform worse than the baseline. Notice that these two ways destroy the motion pattern of the original video, which demonstrates the importance of keeping semantics consistency.

ii. Gaussian Noise, Inter-Video Frame and Intra-Video Frame give positive improvement and are more suitable for action modeling since all of them preserve the motion semantics. Therefore, the idea of removing noise by adding noise is effective, but it is essential to make sure the introduced noise does not affect the motion pattern.

iii. Interestingly, we find that Intra-Video Frame leads to 5% and 6.4% improvement on UCF101 and HMDB51 respectively compared to the Inter-Video Frame. The only difference between them is the source of the static frame, *i.e.*, the former one is selected from the same video that has a more similar background while the latter one is selected from another video that has more discrepancy. Generally, the background in the video is basically unchanged relative to the motion area. Compared to inter-frame, the scene information added by the intra-frame has the same pixel distribution as most other frames in the video. When the convolutional neural network pulls the feature of the distracting video and that of the original video closer, the model needs to remove static intra-frame noise, which is equivalent to remove the background information in the video, making the extracted feature more robust to the background bias.

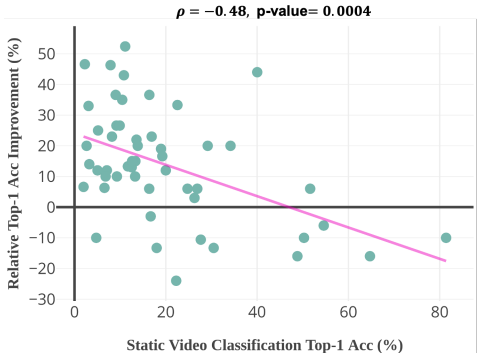

Figure 4: **Relative top-1 acc improvement has a strong negative correlation with the static video classification top-1 acc .** Each dot represents a class in HMDB51 dataset and BE brings more significant improvements in categories that rely less on static information.

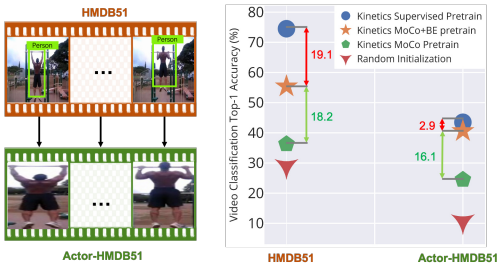

Figure 5: Fine-tuning on the actor dominated dataset actor-HMDB51, our method is very close to the result of Kinetics fully supervised, with only 2.9% difference. Meanwhile the improvement brought by BE over MoCo baseline has only a small drop compared to HMDB51, from 18.2% to 16.1%.

4.5. How does Background Erasing Work?

In this section, we explore how does the Background Erasing works. To this end, we study the relationship between relative performance improvement (%) from the proposed Background Erasing and static video classification top-1 accuracy (%) to see which classes benefit more from our method. *Static* video is generated by randomly selecting one frame and then copying it multiple times. We first trained a randomly initialized I3D model with *static* video generated from HMDB51, which means only *static* information is used. Then two I3D models are pretrained on Kinetics and fine-tuned on HMBD51 using MoCo w/ or w/o BE. At last, we calculate the relative performance improvement brought by BE *w.r.t.* the MoCo baseline, as is shown in Figure 4. The Pearson Correlation is $\rho = -0.48$ with a p-value 0.0004, indicating a *strong negative correlation* between relative performance improvement and static scene bias. Thus, BE works by bringing a significant improvement in categories that rely less on static information.

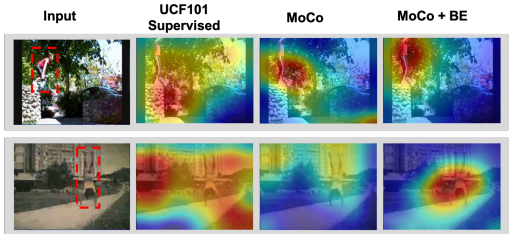

Figure 6: **Generalization ability on novel classes**. Supervised model is severely affected by the scene bias, while after pre-training with MoCo+BE, the model can precisely focus more on moving areas.

4.6. Is Background Really Removed?

In order to verify whether BE has really achieved the purpose of removing the background and paying more attention to the moving subject, we tried to cut the background in the real dataset to test the robustness of our work. We first use HMDB51 to generate an actor dominated dataset Actor-HMDB51. Specifically, we detect the actor in each video frame by frame with public implementation [1] of Faster R-CNN [39] and then crop actor regions out. Considering that some actions in HMDB51 contain two or more persons, we crop a minimum area that contains all persons for these cases. The dataset Actor-HMDB51 obtained in this way has small scene bias thus requires more attention towards the motion information to be well distinguished. Then, we select biased Kinetics for supervised and self-supervised pre-training, and fine-tune on small scale Actor-HMDB51 using I3D backbone. Figure 5 illustrates the result of different methods on HMDB51 and actor-HMDB51. The performance gap between supervised pre-training and self-supervised MoCo+BE on HMDB51 is 19.1%(74.5%-55.4%), while on Actor-HMDB51 is only 2.9% (43.5%-40.6%), which manifests that the advantages of supervised pre-training heavily rely on background information. However, the improvement brought by BE over the MoCo baseline only slightly decreased, from 18.2% to 16.1%. This phenomenon indicates regardless of whether the fine-tuning and test dataset have significant scene bias, BE can steadily bring significant improvement, which demonstrates that BE can indeed make the model pay more attention to the motion pattern. More details about the generation and evaluation of Actor-HMDB51 are provided in the supplementary.

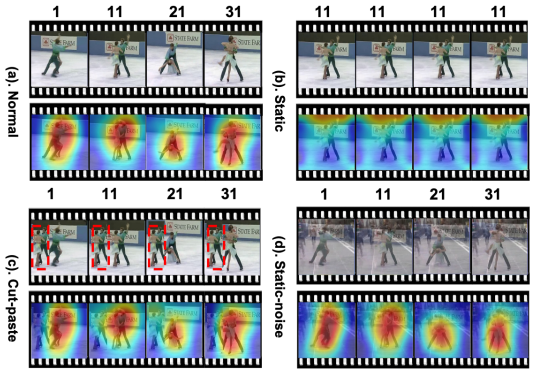

Figure 7: **Does the model really learn to focus on motion pattern?** *i.* When the input is a *static* video, our model doesn't show high activation as expected. *ii.* When pasting a static human body, the model still focus on the moving persons. *iii.* Using a static frame as background noise does not affect the model focus.

[1] https://github.com/endernewton/tf-faster-rcnn

4.7. Visualization Analysis

In this part, we visualize the salient regions of the extracted representations with small modifications on CAM[65]. Specially, we select some videos with significant movements of shape $3 \times 16 \times 224 \times 224$ and the extracted feature representations before global average pooling layer is of shape $512 \times 4 \times 4 \times 4$. Then we average these features over the channel dimension to get the compressed features of shape $4 \times 4 \times 4$. Afterwards, the compressed features are resized to the size of original videos and masked to them.

Novel class transfer capability. To verify the transfer ability of our model on novel class, we visualize some new classes that have never been seen during the training procedure. Specifically, we train three I3D models on UCF101 in both supervised and self-supervised (MoCo and MoCo+BE) manner, and further evaluate on another dataset HMDB51. The visualizations are shown in Figure 6. It can be observed that the supervised model is severely affected by the scene bias and falsely focus on the static background. On the contrary, that the model focus more on motion areas after pre-training with BE and suffer less from scene bias.

Adversarial samples. In this part, we construct some adversarial samples to verify whether our model can really focus on motion pattern, as shown in Figure 7. We use MoCo combined with BE, with I3D as backbone and Kinetics as pretrain dataset. First, using a *static* video as input, our model has a low response to the overall area. Then we paste another static human body, our method can correctly focus on the moving actor, which indicates that our model does not only focus on the human body. In addition, we introduce a static frame from *ride bike* action as noise, which will not affect our model. These experiments prove that feature representations extracted by our method have a fully understanding of space-time.

5. Conclusion

In this paper, we propose a novel Background Erasing (BE) method for self-supervised learning. The proposed method minimizes the feature distance between the sample and sample variation constructed by BE. The proposed method is evaluated using different CNN backbones on three benchmark datasets. Experimental results show that the proposed BE can be well integrated into both the *pretext task* and *contrastive learning* and outperforms existing methods for action recognition notably, especially on a less biased dataset.

Acknowledgement

This work was supported partially by NSFC (No. 61906218), Guangdong Basic and Applied Basic Research Foundation (No. 2020A1515011497) and Science and Technology Program of Guangzhou (No. 202002030371).

References

[1] 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 *CVPR*, pages 9922–9931, 2020. 2, 5, 6

[2] Gedas Bertasius, Christoph Feichtenhofer, Du Tran, Jianbo Shi, and Lorenzo Torresani. Learning discriminative motion features through detection. *arXiv preprint arXiv:1812.04172*, 2018. 6

[3] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, volume 2, pages 1395–1402 Vol. 2, 2005. 3

[4] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *TPAMI*, 23(3):257–267, 2001. 3

[5] A. Briassouli and I. Kompatsiaris. Robust temporal activity templates using higher order statistics. *TIP*, 18(12):2756–2768, 2009. 3

[6] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *Proceedings of the European conference on computer vision (ECCV)*, pages 770–786, 2018. 2

[7] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 4724–4733. IEEE, 2017. 5, 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2

[9] Jinwoo Choi, Chen Gao, Joseph CE Messou, and Jia-Bin Huang. Why can't i dance in the mall? learning to mitigate scene bias in action recognition. In *NeurIPS*, pages 853–865, 2019. 2

[10] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *ICCV*, pages 6192–6201, 2019. 2

[11] Alexey Dosovitskiy, Philipp Fischer, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with exemplar convolutional neural networks. *TPAMI*, 38(9):1734–1747, 2015. 2

[12] Chuang Gan, Boqing Gong, Kun Liu, Hao Su, and Leonidas J Guibas. Geometry guided convolutional neural networks for self-supervised video representation learning. In *CVPR*, pages 5589–5597, 2018. 2, 5

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 2

[14] Rohit Girdhar and Deva Ramanan. CATER: A diagnostic dataset for Compositional Actions and TEmporal Reasoning. In *ICLR*, 2020. 1, 2

[15] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304, 2010. 2

[16] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, volume 2, pages 1735–1742. IEEE, 2006. 4

[17] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCVW*, pages 0–0, 2019. 5

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *ECCV*, 2020. 6

[19] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. *NeurIPS*, 33, 2020. 2, 5

[20] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *CVPR*, pages 6546–6555, 2018. 1, 5

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, pages 9729–9738, 2020. 2, 5, 6

[22] Yann N. Dauphin Hongyi Zhang, Moustapha Cisse and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2018. 6

[23] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv preprint arXiv:1811.11387*, 2018. 5

[24] Gagan Kanojia, Sudhakar Kumawat, and Shanmuganathan Raman. Attentive spatio-temporal representation learning for diving classification. In *CVPRW*, pages 0–0, 2019. 6

[25] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[26] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, volume 33, pages 8545–8552, 2019. 2, 5

[27] Hilde Kuehne, Hueihan Jhuang, Rainer Stiefelhagen, and Thomas Serre. Hmdb51: A large video database for human motion recognition. In *High Performance Computing in Science and Engineering ?12*, pages 571–582. Springer, 2013. 2, 4

[28] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676, 2017. 5

[29] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, September 2018. 1, 2

[30] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, pages 513–528, 2018. 2, 4

[31] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *ICCV*, pages 5512–5521, 2019. 6

[32] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *AAAI*, 2020. 2, 6

[33] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, pages 9879–9889, 2020. 2

[34] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, pages 527–544. Springer, 2016. 2, 5

[35] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, pages 69–84. Springer, 2016. 2

[36] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 4

[37] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 5

[38] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, pages 133–142, 2020. 2

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, pages 91–99, 2015. 8

[40] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, Sergey Levine, and Google Brain. Time-contrastive networks: Self-supervised learning from video. In *ICRA*, pages 1134–1141. IEEE, 2018. 2

[41] 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. 2, 4

[42] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Gate-shift networks for video action recognition. In *CVPR*, pages 1102–1111, 2020. 6

[43] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Self-supervised video representation learning using inter-intra contrastive framework. *ACM MM*, 2020. 2, 5

[44] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. *arXiv preprint arXiv:1906.05849*, 2019. 2

[45] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015. 5

[46] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 6

[47] Carl Vondrick, Hamed Pirsiavash, and Antonio Torralba. Generating videos with scene dynamics. In *NeurIPS*, pages 613–621, 2016. 5

[48] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, pages 352–361, 2020. 6

[49] Jiangliu Wang, Jianbo Jiao, Linchao Bao, Shengfeng He, Yunhui Liu, and Wei Liu. Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics. In *CVPR*, pages 4006–4015, 2019. 2, 6

[50] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. *ECCV*, 2020. 5, 6

[51] 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 *ECCV*, pages 20–36. Springer, 2016. 3

[52] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *TPAMI*, 2018. 2

[53] Yang Wang and Minh Hoai. Pulling actions out of context: Explicit separation for effective combination. In *CVPR*, pages 7044–7053, 2018. 1, 2

[54] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, pages 8052–8060, 2018. 2, 5

[55] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *ICCV*, pages 3164–3172, 2015. 1

[56] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, pages 3733–3742, 2018. 2

[57] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *CVPR*, pages 10334–10343, 2019. 2, 5, 6

[58] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *CVPR*, pages 6548–6557, 2020. 2

[59] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *CVPR*, pages 6210–6219, 2019. 2

[60] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *ICCV*, 2019. 6

[61] Liheng Zhang, Guo-Jun Qi, Liqiang Wang, and Jiebo Luo. Aet vs. aed: Unsupervised representation learning by auto-encoding transformations rather than data. In *CVPR*, pages 2547–2555, 2019. 2

[62] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, pages 649–666. Springer, 2016. 2

[63] Yue Zhao, Yuanjun Xiong, Limin Wang, Zhirong Wu, Xiaoou Tang, and Dahua Lin. Temporal action detection with structured segment networks. In *ICCV*, pages 2914–2923, 2017. 1

[64] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, pages 803–818, 2018. 6

[65] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 8