Frequency Selective Augmentation for Video Representation Learning

Jinhyung Kim[1*], Taeoh Kim[2], Minho Shim[2], Dongyoon Han[3], Dongyoon Wee[2], Junmo Kim[4]

[1] LG AI Research
[2] NAVER CLOVA Video
[3] NAVER AI Lab
[4] KAIST

Abstract

Recent self-supervised video representation learning methods focus on maximizing the similarity between multiple augmented views from the same video and largely rely on the quality of generated views. However, most existing methods lack a mechanism to prevent representation learning from bias towards static information in the video. In this paper, we propose frequency augmentation (FreqAug), a spatio-temporal data augmentation method in the frequency domain for video representation learning. FreqAug stochastically removes specific frequency components from the video so that learned representation captures essential features more from the remaining information for various downstream tasks. Specifically, FreqAug pushes the model to focus more on dynamic features rather than static features in the video via dropping spatial or temporal low-frequency components. To verify the generality of the proposed method, we experiment with FreqAug on multiple self-supervised learning frameworks along with standard augmentations. Transferring the improved representation to five video action recognition and two temporal action localization downstream tasks shows consistent improvements over baselines.

Introduction

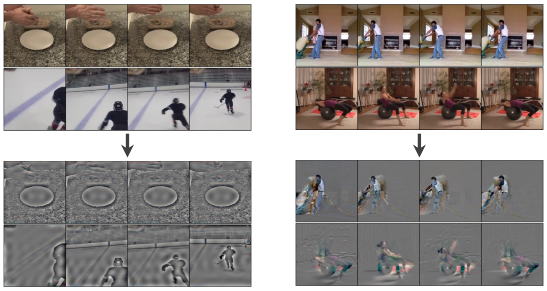
There has been growing attention on transferring knowledge from large-scale unsupervised learning to various downstream tasks in natural language processing (Devlin et al. 2018; Radford et al. 2019) and computer vision (Chen et al. 2020a; He et al. 2020; Grill et al. 2020) communities. Considering data accessibility and possible applications, video representation learning has great potential as a tremendous amount of videos with diverse contents are created, shared, and consumed every day. In fact, unsupervised or self-supervised learning (SSL) of video via learning invariance between multimodal or multiple augmented views of an instance is being actively studied (Han, Xie, and Zisserman 2020; Alayrac et al. 2020; Recasens et al. 2021; Huang et al. 2021b; Qian et al. 2021; Feichtenhofer et al. 2021).

Recent studies in image SSL indicate that a careful selection of data augmentation is crucial for the quality of the feature (Wen and Li 2021) or for improving performance in downstream tasks (Tian et al. 2020; Zhao et al. 2021). However, augmentations for video SSL have not been sufficiently explored yet. For videos, in terms of spatial dimension, the standard practice is adopting typical image augmentations in a temporally consistent way, *i.e.*, applying the same augmentation to every frame (Qian et al. 2021). Meanwhile, a few previous works have investigated augmentations in the temporal dimension, including sampling a distant clip (Feichtenhofer et al. 2021), sampling clips with different temporal scales (Dave et al. 2021) or playback speeds (Chen et al. 2021; Huang et al. 2021a; Duan et al. 2022), and dropping certain frames (Pan et al. 2021). Although effective, sampling-based augmentations in the temporal dimension inevitably modulate a video as a whole regardless of signals in a clip varying at different rates. These methods are limited in resolving the spatial bias problem (Li, Li, and Vasconcelos 2018) of video datasets which requires distinguishing motion-related features from static objects or scenes. Adding a static frame (Wang et al. 2021b) is a simple heuristic to attenuate the temporally stationary signal, but it is hard to generalize to the real world's non-stationary signal in the spatio-temporal dimension. The need for a more general way to selectively process a video signal depending on the spatial and temporal changing rates motivates us to consider frequency domain analysis.

In digital signal processing, converting a signal to the frequency domain using discrete Fourier transform (DFT), then processing the signal is widely used in many applications. Filtering in the frequency domain is one example that attenuates a specific frequency range to remove undesirable components, such as noise, from the signal. With its effectiveness in mind, we propose filtering video signals in the frequency domain to discard unnecessary information while keeping desired features for the SSL model to learn.

Fig. 1 shows the outcome of filtering out low-frequency components (LFC) from videos. When the spatial filter is applied, plain surfaces of objects in the scene are erased while their boundary or shapes are remained. As for the temporal filter, stationary parts of the video, *e.g.*, static objects or the background, are removed while dynamic parts, *e.g.*, a person moving, are retained. These results are aligned with the previous discoveries that high-frequency components (HFC) carry essential information for image and video understanding (Wang et al. 2020; Kim et al. 2020).

[*] This work is done during Ph.D. program at KAIST.



(a) Remove Spatial LFC (b) Remove Temporal LFC

Figure 1: Impact of removing low-frequency components (LFC). (a) Filtering spatial LFC can attenuate spatially redundant information, *e.g.* colors, while keeping the shape patterns. (b) Removing temporal LFC filters out temporally stationary information, *e.g.* the background, while keeping the motion pattern.

In this work, we propose frequency augmentation (FreqAug), a novel spatio-temporal augmentation in the frequency domain by randomly applying a filter to remove selective frequency bands from the video. Specifically, we aim to alleviate representation bias for better transferability by filtering out spatially and temporally static components from the video signal. FreqAug is composed of a 2D spatial filter and a 1D temporal filter, and their frequency band can be determined by the filter type and its cutoff frequency. In video SSL, FreqAug can be applied to each view independently on top of other video augmentations so that the model learns invariance on LFC (Fig. 4). In particular, applying FreqAug with high-pass filter results in obtaining the representation with less static bias via learning invariant features between the instance and its HFC. Note that what we are claiming is not that only HFC are important but rather a matter of relative importance. Since FreqAug is applied randomly, LFC still get involved in the invariance learning.

We demonstrate the effectiveness of the proposed method by presenting transfer learning performance on five action recognition datasets: coarse-grained (UCF101 and HMDB51) and fine-grained (Diving48, Gym99, and Something-Something-v2) datasets. Additionally, the learned features are evaluated via the temporal action segmentation task on Breakfast dataset and the action localization task on THUMOS'14 dataset. Empirical results show that FreqAug enhances the performance of multiple SSL frameworks and backbones, which implies the learned representation has significantly improved transferability. We also make both quantitative and qualitative analyses of how FreqAug can affect video representation learning.

Related Work

Frequency Domain Augmentations

Lately, several studies on frequency domain augmentation have been proposed for the 1D speech and 2D image domains. For speech or acoustic signals, a few works incorporated augmentations that are masking (Park et al. 2019) or filtering (Nam, Kim, and Park 2021) a spectrogram or mixing that of two samples (Kim, Han, and Ko 2021). In the image domain, Xu *et al.* (Xu et al. 2021b) tackled the domain generalization problem by mixing spectrum amplitude of two images. A concurrent work (Nam and Lee 2021) introduced randomly masking a certain angle of Fourier spectrum based on the spectrum intensity distribution for X-ray image classification. These methods are relevant to ours in that they randomly alter a spectrum for data augmentation, but differs in the following two respects. First, to the best of our knowledge, our work is the first 3D spatio-temporal augmentation in the frequency domain for video representation learning and investigates the transferability to various downstream tasks. Second, our method differs from the existing frequency domain augmentations in that it selectively filters out a certain frequency band, *e.g.*, low-frequency components, rather than random frequency components through the entire range. We empirically show the superiority of selective filtering over the random filtering strategy (Table 2).

Video Self-Supervised Learning

Self-supervised learning (SSL) through multi-view invariance learning has been widely studied for image recognition and other downstream tasks (Wu et al. 2018; Chen et al. 2020a; He et al. 2020; Chen et al. 2020b; Grill et al. 2020; Caron et al. 2020; Chen and He 2021). In video SSL, previous works exploited the view invariance-based approaches from the image domain and explored ways to utilize unique characteristics of the video including additional modalities, *e.g.*, optical flow, audio, and text (Wang et al. 2021a; Huang et al. 2021b; Xiao, Tighe, and Modolo 2021; Han, Xie, and Zisserman 2020; Miech et al. 2020; Alayrac et al. 2020; Alwassel et al. 2020; Recasens et al. 2021; Behrmann et al. 2021). However, we focus more on the RGB-based video SSL methods in this study. CVRL (Qian et al. 2021) proposed a temporally consistent spatial augmentation and temporal sampling strategy, which samples two positive clips more likely from near time. RSPNet (Chen et al. 2021) combined relative speed perception and video instance discrimination tasks to learn both motion and appearance features from video. Empirical results in (Feichtenhofer et al. 2021) show four image-based SSL frameworks (Chen et al. 2020b; Grill et al. 2020; Chen et al. 2020a; Caron et al. 2020) can be generalized well to the video domain. MoCo-BE (Wang et al. 2021b) and FAME (Ding et al. 2022) introduced a regularization that reduces background influences on SSL by adding a static frame to the video or mixing background, respectively. Suppressing static cues (Zhang, Wang, and Ma 2022) in the latent space is another way to reduce spatial bias. Our work is also a study on data augmentation for video SSL, but we propose to modulate the video signal in the frequency domain in a more general and simpler way.

Method

Preliminary

In this work, we aim to augment spatio-temporal video signals in a frequency domain by filtering particular frequency components. Discrete Fourier transform (DFT), a widely used technique in many digital signal processing applications, provides appropriate means of converting a finite discrete signal into the frequency domain for computers. For simplicity, let us consider 1D discrete signal $x[n]$ of length N, then 1D DFT is defined as,

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn}, \tag{1}$$

where $X[k]$ is the spectrum of $x[n]$ at frequency $k = 0, 1, ..., N-1$. Since DFT is a linear transformation, the original signal can be reconstructed by inverse discrete Fourier transform (iDFT):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j(2\pi/N)kn}. \tag{2}$$

1D-DFT can be extended to the multidimensional DFT by simply calculating a series of 1D-DFT along each dimension. One can express d-dimensional DFT in a concise vector notation as,

$$X_{\mathbf{k}} = \sum_{\mathbf{n}=0}^{\mathbf{N}-1} x_{\mathbf{n}} e^{-j2\pi \mathbf{k}(\mathbf{n}/\mathbf{N})}, \tag{3}$$

where $\mathbf{k} = (k_1, k_2, ..., k_d)$ and $\mathbf{n} = (n_1, n_2, ..., n_d)$ are d-dimensional indices from $\mathbf{0}$ to $\mathbf{N} = (N_1, N_2, ..., N_d)$ and $\mathbf{n}/\mathbf{N}$ is defined as $(n_1/N_1, n_2/N_2, ..., n_d/N_d)$. We omit the equation of d-dimensional iDFT as it is a straightforward modification from Eq. 2.

Filtering Augmentation in Frequency Domain

Filtering in signal processing often denotes a process of suppressing certain frequency bands of a signal. Filtering in frequency domain can be described as an element-wise multiplication $\odot$ between a filter F and a spectrum X as,

$$\hat{X} = F \odot X, \tag{4}$$

where $\hat{X}$ is a filtered spectrum. A filter can be classified based on the frequency band that the filter passes or rejects: low-pass filter (LPF), high-pass filter (HPF), band-pass filter, band-reject filter, and so on. LPF passes a low-frequency band while it filters out high-frequency components from the signal; HPF works oppositely. Let us consider a simple 1D binary filter, also known as an ideal filter, then LPF and HPF can be defined as,

$$F_{lpf}[k] = \begin{cases} 1 & \text{if } |k| < f_{co} \\ 0 & \text{otherwise,} \end{cases} \tag{5}$$

$$F_{hpf}[k] = 1 - F_{lpf}[k], \tag{6}$$

where f_{co} is the cutoff frequency which controls the frequency band of the filter.

In this work, we propose frequency augmentation (FreqAug, Fig. 2), which utilizes 3D-DFT with the binary filter approach to augment video data in the frequency domain by stochastically removing certain frequency components. Since video signals have three dimensions, *i.e.*, T, H, and W, the filter also can be 3D and have three independent cutoff frequencies. We introduce a single spatial cutoff frequency f^s_{co} that handles both H and W dimension, and one temporal cutoff frequency f^t_{co} for T dimension. Then 1D temporal filters are identical to Eq. 5 and Eq. 6, and 2D spatial LPF can be defined as,

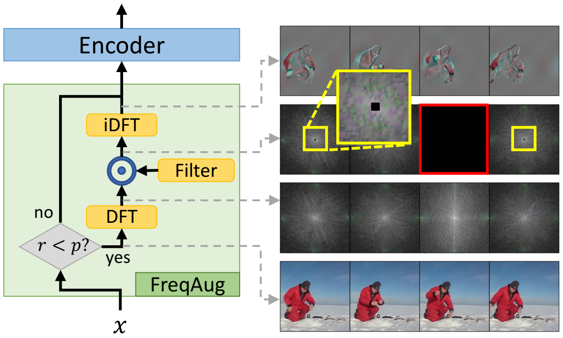


Figure 2: Frequency augmentation (FreqAug). Filtering in the frequency domain is a sequential process of 1) transforming a video to a spectrum by DFT; 2) applying the desired filter by element-wise multiplication; 3) transforming the filtered spectrum back to the video domain by iDFT. Figures on the right are an example of applying spatio-temporal high-pass filters. In filtered spectrum (2nd row), low-frequency components of spatial (small black regions inside yellow boxes, see the first one for the close-up) and temporal (the red box) axis are removed. FreqAug is placed after other augmentations and randomly applied when $r{\sim}U(0,1)$ is less than the augmentation probability p.

$$F^s_{lpf}[k_h, k_w] = \begin{cases} 1 & \text{if } |k_h| < f^s_{co} \text{ and } |k_w| < f^s_{co} \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

and F^s_{hpf} is obtained in the same way as Eq. 6. Finally, the spatio-temporal filter $\mathbf{F}$ can be obtained by outer product between the temporal filter F^t and the spatial filter F^s as,

$$\mathbf{F} = F^{st}[k_t, k_h, k_w] = F^t[k_t] \otimes F^s[k_h, k_w], \tag{8}$$

where $\otimes$ is outer product. The final 3D filtered spectrum $\hat{\mathbf{X}}$ can be represented as an element-wise multiplication between $\mathbf{F}$ and the spectrum $\mathbf{X}$ as Eq. 4.

Additionally, FreqAug has one more hyperparameter, the augmentation probability p, which determines how frequently the augmentation is applied. FreqAug processes the input only when the random scalar r, sampled from uniform distribution $U(0, 1)$, is less than p.

Fig. 2 presents a block diagram of FreqAug and a visualization of a video sample and its spectrum at each stage of FreqAug. Note that FreqAug blocks are located after other augmentations or noramlization, and operate with independent r for each view. For the spectrum, lower absolute spatial frequencies are located near the center of the spectrum at each column ($(k_h, k_w) = (0, 0)$) and lower absolute temporal frequencies are located near the third spectrum ($k_t = 0$). For visualization, we apply spatial and temporal HPF with $f^s_{co} = 0.01$ and $f^t_{co} = 0.1$, respectively. In the filtered spectrum (2nd row), spatial low-frequency (small black region inside yellow boxes) and temporal low-frequency (red box) components are removed.

Experiment

Experiment Settings

Here, we provide essential information to understand the following experiments. Refer to Appendix A1 for more details.

Datasets. For pretraining the model, we use Kinetics-400 (K400) (Carreira and Zisserman 2017) and Mini-Kinetics (MK200) (Xie et al. 2018). With the limited resources, we choose MK200 as a major testbed to verify our method's effectiveness. For evaluation of the pretrained models, we use five different action recognition datasets: UCF101 (Soomro, Zamir, and Shah 2012), HMDB51 (Kuehne et al. 2011), Diving48 (DV48) (Li, Li, and Vasconcelos 2018), Gym99 (Shao et al. 2020), and Something-something-v2 (SSv2) (Goyal et al. 2017). Following the standard practice, we report the finetuning accuracy on the three datasets: UCF101, HMDB51, and Diving48. Note that we present split-1 accuracy for UCF101 and HMDB51 by default unless otherwise specified. For Gym99 and SSv2, we evaluate the models on the low-shot learning protocol using only 10% of training data since they are relatively large-scale (especially the number of samples in SSv2 is about twice larger than that of our main testbed MK200). For temporal action localization, Breakfast (Kuehne, Arslan, and Serre 2014) and THUMOS'14 (Idrees et al. 2017) dataset are used.

Self-supervised Pretraining. For self-supervised pretraining, all the models are trained with SGD for 200 epochs. Regarding spatial augmentation, augmentations described in (Chen et al. 2020b) are applied as our baseline. For temporal augmentation, randomly sampled clips from different timestamps compose the positive instances. Also, two clips are constrained to be sampled within a range of 1 second. Each clip consists of T frames sampled from $T \times \tau$ consecutive frames with the stride τ. In terms of FreqAug, we use the following two default settings: 1) FreqAug-T (temporal) uses temporal HPF with a cutoff frequency 0.1; 2) FreqAug-ST (spatio-temporal) is a combination of spatial HPF with a cutoff frequency 0.01 alongside with FreqAug-T.

Finetuning and Low-shot Learning. We train the models for 200 epochs with the initial learning rate 0.025 without warm-up and zeroed weight decay for supervised finetuning and low-shot learning. Only fundamental spatial augmentations (Feichtenhofer et al. 2021) are used.

Temporal Action Segmentation and Localization. We train an action segmentation model, MS-TCN (Farha and Gall 2019) following (Behrmann et al. 2021), and a localization model, G-TAD (Xu et al. 2020) for evaluating the learned representation of pretrained encoders.

Evaluation. For Kinetics, UCF101, and HMDB51, we report average accuracy over 30-crops following (Feichtenhofer et al. 2019). In the case of Diving48, Gym99, and SSv2, we report the spatial 3-crop accuracy with segment-based temporal sampling. For temporal action segmentation, frame-wise accuracy, edit distance, and F1 score at overlapping thresholds 10%, 25%, and 50% are used. For temporal action localization, we measure mean average precision (mAP) with intersection-over-union (IoU) from 0.3 to 0.7.

Backbone	Augment.	Finetune			Low-shot (10%)	
		UCF101	HMDB51	DV48	Gym99	SSv2
SO-50	Baseline	87.0	56.5	67.8	29.9	25.3
	+ **FA-ST**	**90.0**	**61.6**	**71.0**	34.8	**28.1**
	+ **FA-T**	89.8	60.8	70.3	**35.2**	**28.1**
SO-18	Baseline	84.5	55.2	74.9	30.3	23.9
	+ **FA-ST**	88.5	57.8	**75.8**	**35.3**	25.7
	+ **FA-T**	**88.7**	**58.8**	75.7	34.7	**26.1**
R(2+1)D	Baseline	86.2	60.4	64.6	42.5	29.2
	+ **FA-ST**	**90.0**	**65.9**	67.7	**48.4**	**31.5**
	+ **FA-T**	89.5	65.2	**70.2**	48.3	30.5
S3D-G	Baseline	89.0	59.5	70.1	42.1	30.5
	+ **FA-ST**	90.2	**63.6**	**71.0**	**44.5**	31.1
	+ **FA-T**	**90.4**	62.2	68.8	44.3	**31.5**

Table 1: Evaluation results on Mini-Kinetics. We evaluate FreqAug (FA) with diverse backbones, including SlowOnly-50 (SO-50), SlowOnly-18 (SO-18), R(2+1)D and S3D-G, via finetuning and low-shot learning protocols.

Backbone. Our default encoder backbone is SlowOnly-50 (SO-50), a variant of 3D ResNet originated from the slow branch of SlowFast Network (Feichtenhofer et al. 2019). We evaluate our method on R(2+1)D (Tran et al. 2018) and S3D-G (Xie et al. 2018) models as well.

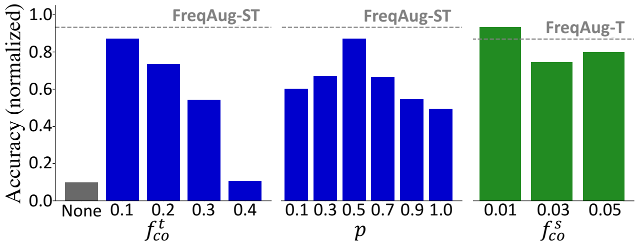
SSL Methods. We implement MoCo (Chen et al. 2020b) and BYOL (Grill et al. 2020) for pretraining the video model. We set MoCo as our default SSL method.

Action Recognition Evaluation Results

In Table 1, we present the evaluation results of MoCo with FreqAug pretrained on MK200. We validate on four different backbones: SlowOnly-50 (SO-50), SlowOnly-18 (SO-18), R(2+1)D, and S3D-G, which have various input resolutions (number of frames T, stride τ), depth, and network architecture. First, MoCo pretrained SO-50 with FreqAug significantly improves the baseline in all five downstream tasks. The absolute increments of top-1 accuracy range from 2.5% to 5.3% depending on the task. We observe that FreqAug-ST shows comparable or better accuracy than FreqAug-T in four out of five tasks, indicating the synergy between spatial and temporal filters. The results of the other three backbones show that FreqAug boosts the performance in almost all cases regardless of temporal input resolutions and the network architecture. Please refer to Appendix A3.1 for results with other SSL methods, A3.2 for the detailed setup of each backbone and results of 3D-ResNet-18 and other input resolutions, and A4.6 for comparison with other augmentations.

Ablation Study

On Hyperparameters. We conduct three types of ablation studies on MK200 to search for proper hyperparameters in Fig. 3. SO-50 pretrained with MoCo is used as the baseline model. For ease of visualization, we first min-max normalize top-1 accuracies for each task using all ablation models, then present average accuracy over five action recognition tasks. We also mark the accuracy of models with de-



(a) FreqAug-T (b) FreqAug-ST

Figure 3: Hyperparameter ablations on Mini-Kinetics. (a) temporal cutoff frequency (f_{co}^{t}) and augmentation probability (p) for FreqAug-T, and (b) spatial cutoff frequency (f_{co}^{s}) for FreqAug-ST. Other parameters set fixed. Min-max normalized accuracies of 5 tasks are averaged.

Filter	Finetune			Low-shot (10%)	
	UCF101	HMDB51	Diving48	Gym99	SSv2
No filter	87.0	56.5	67.8	29.9	25.3
HPF (default)	**89.8**	**60.8**	**70.3**	**35.2**	**28.1**
LPF (f_{co}^{t}=0.2)	84.1	51.3	66.3	26.2	22.2
LPF (f_{co}^{t}=0.3)	85.8	54.4	67.9	28.8	24.2
LPF (f_{co}^{t}=0.4)	87.9	56.1	69.2	30.3	25.5
Random (M=2)	88.9	59.0	69.1	33.4	26.9
Random (M=3)	89.1	58.0	69.1	33.3	25.9
Random (M=5)	88.2	56.5	69.5	31.5	25.2

Table 2: Temporal filtering strategy comparison on Mini-Kinetics: 1) LPF with cutoff frequency (f_{co}^{t}) and 2) random masking policy with mask parameter (M).

fault FreqAug-ST or FreqAug-T in dotted line for a better comparison. Note that the cutoff frequencies are searched in consideration of the minimum interval between each component: $1/T$ for temporal and $1/H$ (or $1/W$) for spatial dimension. Fig. 3 shows that FreqAug with default hyperparameters, (a) f_{co}^{t}=0.1 and p=0.5 for FreqAug-T, and (b) f_{co}^{s}=0.01 for FreqAug-ST, achieves the best performance. Detailed description and more ablation studies can be found in Appendix A3.4 and A3.5.

On Filtering Strategy. In Table 2, we compare two variants of temporal filtering strategy on MoCo-pretrained SO-50: LPF and random masking. LPF strategy is masking frequency components less than f_{co}^{t} as opposed to default HPF. We tested $f_{co}^{t}\in\{0.2, 0.3, 0.4\}$ and observe that the performance becomes worse than the baseline as more high-frequency components are filtered out. The results show a clear contrast between HPF and LPF strategies, and choosing a proper frequency band for the filter is essential. We also tested temporal random mask, like SpecAugment (Park et al. 2019), with mask parameter M. Larger M indicates that a larger mask size can be sampled. Refer to Appendix A2.2 for the detail. The scores for random policy ($M\in\{2, 3, 5\}$) are better than the baseline but cannot reach the HPF policy's score, which confirms the validity of selective augmentation. Refer to Appendix A4.5 for filtering in video domain.

Model	BB	T	Epochs	Finetune		
				UCF101	HMDB51	DV48
RSPNet‡	S3D-G	64	200	89.9	59.6	N/A
MoCo-BE	I3D	16	50	86.8	55.4	62.4
FAME †	I3D	16	200	88.6	61.1	72.9
ASCNet †	S3D-G	64	200	90.8	60.5	N/A
ρMoCo (ρ=2)†	SO-50	8	200	91.0	N/A	N/A
ρBYOL (ρ=2)†	SO-50	8	200	92.7	N/A	N/A
CVRL	SO-50	32	800	92.2	66.7	N/A
RSPNet‡	S3D-G	64	1000	93.7	64.7	N/A
MoCo (ours)	SO-50	8	200	90.6	62.8	72.9
MoCo + **FA-ST**	SO-50	8	200	92.1	65.6	74.0
MoCo + **FA-T**	SO-50	8	200	91.8	65.1	73.8
BYOL (ours)	SO-50	8	200	92.9	67.7	71.9
BYOL + **FA-ST**	SO-50	8	200	**93.7**	**68.3**	**74.4**
BYOL + **FA-T**	SO-50	8	200	93.2	67.7	72.2

Table 3: Comparison with RGB-based models pretrained on Kinetics-400. Backbone (BB), number of frames (T), and pretrain epochs are specified. The UCF101 and HMDB51 accuracies are averaged over 3 splits. †: evaluated on split-1; ‡: ambiguous or not specified which splits are used.

Method	Pretrain	Acc.	Edit	F1@{0.10, 0.25, 0.50}		
SO-50 †	Sup.	59.0	59.5	54.7	49.2	37.6
LSFD, N	Self-sup.	60.6	60.0	52.0	42.8	35.3
MoCo†	Self-sup.	59.9	60.4	57.2	52.0	40.2
+ **FreqAug-ST**†		65.2	63.9	61.7	56.6	45.2
+ **FreqAug-T**†		**65.9**	**64.8**	**62.5**	**57.1**	**45.3**

Table 4: Temporal action segmentation on Breakfast. All features are evaluated with MS-TCN. 'Edit' denotes edit distance. †: scores are averaged over 10 evaluations on split-1.

Comparison with Previous Models

Table 3 presents K400 experiments with FreqAug compared to previous video SSL works. For a fair comparison, SSL models are chosen based on three criteria: augmentation-based, RGB-only (without multimodality including optical flow), and spatial resolution of 224 × 224. We report the average accuracy of 3 splits for UCF101 and HMDB51. We set p=0.3 for BYOL + FreqAug-ST. Note that ρ of ρMoCo and ρBYOL indicates the number of views from different timestamps, so models with ρ=2 are directly comparable to our models. First, both FreqAug-ST and FreqAug-T consistently outperform the baseline MoCo and BYOL on UCF101, HMDB51, and Diving48. Compared with other models trained with similar epochs, MoCo and BYOL with FreqAug outperform all the others with similar training epochs. Interestingly, FreqAug demonstrates its training efficiency by defeating RSPNet on HMBD51 and surpassing CVRL; they are pretrained for 1000 and 800 epochs, respectively. We expect training with more powerful SSL methods and longer epochs can be complementary to our approach.

Other Downstream Evaluation Results

In Table 4, we report the results of temporal action segmentation task on the Breakfast dataset. We experiment with the features extracted from MoCo pretrained SO-50 on K400.

Method	Pretrain	mAP@{0.3, 0.4, 0.5, 0.6, 0.7}					Avg
TSM	Sup.	46.6	39.5	30.1	20.1	12.2	29.7
TSM + BSP		52.3	46.3	39.8	**30.8**	**21.1**	38.1
TSN†	Sup.	45.7	36.8	28.2	19.0	11.3	28.2
SO-50†		51.1	44.2	34.2	24.7	15.3	33.9
MoCo†	Self-sup.	52.2	45.6	37.3	28.0	18.2	36.3
+ **FreqAug-ST**†		54.1	47.4	39.4	29.6	19.8	38.1
+ **FreqAug-T**†		**55.4**	**48.7**	**40.3**	30.3	20.2	**39.0**

Table 5: Temporal action localization on THUMOS'14. Features are pretrained on K400 and evaluated with G-TAD. †: scores are mean over 5 runs.

Method	Sup.	Finetune			Low-shot (10%)	
	MK200	UCF101	HMDB51	Diving48	Gym99	SSv2
SO-50	77.4	91.0	61.0	72.3	36.4	25.5
+ **FA-ST**	78.6	91.3	62.9	73.2	39.2	27.1
+ **FA-T**	78.0	91.5	65.4	71.0	40.0	26.2

Table 6: Supervised pretraining with FreqAug (FA). SlowOnly-50 (SO-50) pretrained on MK200. Sup. denotes supervised action recognition accuracy.

In addition, we report the performance of the extracted feature by officially released SO-50 (Fan et al. 2020) pretrained on K400 by supervised learning. The results show that MoCo-pretrained with FreqAug substantially improves the baseline on all metrics. We conjecture that foreground motion can easily be separated in the videos with static backgrounds by the FreqAug-enhanced feature. Furthermore, MoCo with FreqAug surpasses its supervised counterpart and LSFD (Behrmann et al. 2021) in all metrics, which is the only video SSL method evaluated on this task.

In Table 5, we report the results of temporal action localization on THUMOS'14 dataset. We use the features extracted from MoCo pretrained SO-50 on K400. The results show that MoCo features outperform supervised features from RGB-only TSN (Wang et al. 2016) and SO-50. Moreover, adding FreqAug to MoCo improves the localization performance even further than the baseline. We also compare our results to BSP (Xu et al. 2021a), a localization-specific pre-training method, showing similar or better localization performances. Note that BSP is pre-trained in supervised manners while our encoders are pre-trained with fully unsupervised. For more results and analysis, please refer to Appendix A3.6 and A4.8.

Discussion

FreqAug for Supervised Learning

One may wonder whether using FreqAug in supervised learning is still effective; here, we evaluate FreqAug in a supervised scenario to demonstrate the versatility of our method. Table 6 shows the performance of MK200 pretrained SlowOnly-50 by supervised learning for 250 epochs. Note that p=0.3 is used since we observed lower accuracy with a too large p. When we applied FreqAug on top of basic augmentation, we observe overall performance improvements, including the performance of the five downstream tasks and the MK200 pretraining task.

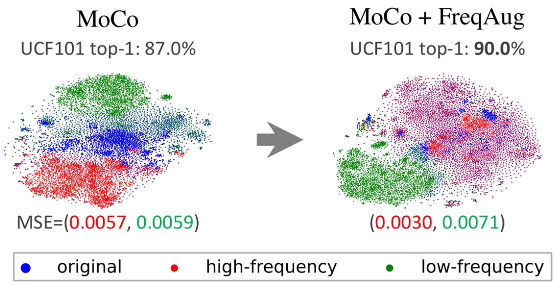


Figure 4: t-SNE visualization of the output features from original frames (blue) and its temporal HFC (red) or LFC (green). Mean squared error (MSE) between original features with HFC/LFC are presented under each plot. MoCo pretrained SlowOnly-50 models with or without FreqAug (and UCF101 finetuning acccuracies) are compared. FreqAug makes features of HFC close to that of original clips which results in better downstream performance. If red and blue dots are too close, they can be perceived as purple.

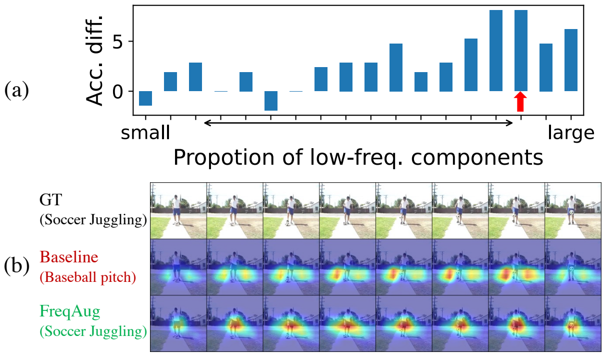


Figure 5: Comparing downstream models pretrained with or without FreqAug on UCF101. (a) Accuracy difference according to the LFC ratio of the sample. (b) Grad-CAM of a clip from a bin with large LFC (red arrow in (a)).

Influence on Video Representation Learning

We take a closer look at how the downstream performance of the features learned through FreqAug can be improved compared to the baseline. Fig. 4 shows t-SNE (van der Maaten and Hinton 2008) plots of features from original clips (blue) with both high-frequency components (HFC) and low-frequency components (LFC) and either high-pass or low-pass filtered clips (red/green) in temporal dimension. The distance between two features is measured using mean squared error (MSE). We compare features from MoCo pretrained SlowOnly-50 on MK200 with or without FreqAug-ST. The samples are from the validation set of MK200, and f^t_{co}=0.2 are set for both HPF and LPF. We observe that the distance between original clips and its temporal HFC substantially decreased when the model is pretrained with Fre-

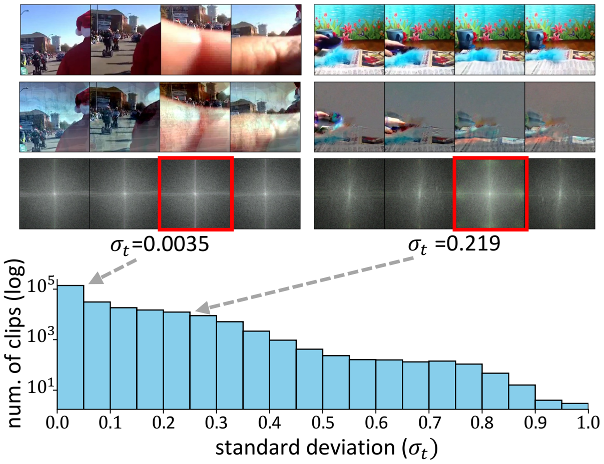


Figure 6: Standard deviation of spectrum intensity over temporal axis. The histogram illustrates the standard deviation (std) distribution of clips in the K400 training set. Top figures show examples of small std (left) and large std (right) videos; top, middle, and bottom rows denote original frames, filtered frames, and spectrum with its std, respectively. Red box indicates where the temporal frequency is zero.

qAug while there are relatively small changes in the distance between the clip and its LFC; which means FreqAug does not reduce the overall distance between features. It indicates that FreqAug makes the model extract relatively more features from HFC via invariance learning between HFC and all frequency components in the video. We believe the feature learned via FreqAug whose HFC has been enhanced, leads to better transferability of the model as shown empirically. Refer to Appendix A4.1 for more t-SNE analysis.

To analyze the effect of FreqAug on the downstream task, we group data instances in UCF101 according to the amount of temporal LFC each video has and present accuracy increment in each group caused by FreqAug in Fig. 5 (a); refer to Sec. A4.2 for the detailed description. The result shows that the effectiveness of FreqAug tends to be amplified even more on videos with a higher proportion of temporal LFC; those videos are expected to have a large portion of static scenes, background, or objects. In Fig. 5(b), we visualize a sample from a bin with a large LFC (red-arrowed in (a)); original frames, GradCAM (Selvaraju et al. 2017) of MoCo baseline (Baseline) and MoCo+FreqAug (FreqAug) models from top to bottom. We observed that FreqAug correctly focuses on the person juggling a soccer ball while Baseline fails to recognize the action because it focuses on the background field. Refer to Appendix A4.3 for more samples. In conclusion, FreqAug helps the model focus on motion-related areas in the videos with static backgrounds.

Analysis on Temporal Filtering

As aforementioned, FreqAug can help the model focus on motion-related information by randomly removing the background with a temporal high-pass filter. However, one may doubt whether FreqAug is only effective with videos whose background can be easily removed. In order to resolve the doubt above, we conduct further analysis by applying FreqAug on different subsets of the training dataset according to the spectrum intensity over the temporal axis.

Std.		Finetune			Low-shot(10%)	
< 0.05	> 0.05	UCF101	HMDB51	Diving48	Gym99	SSv2
no FreqAug		87.0	56.5	67.8	29.9	25.3
	✓	89.4	59.3	**70.9**	32.4	27.4
✓		88.8	58.3	69.9	32.4	27.0
✓	✓	**89.8**	**60.8**	70.3	**35.2**	**28.1**

Table 7: Impact of samples with large temporal variations to the temporal filter. Samples of temporal spectrum std. below or above threshold (0.05) are rejected to apply the temporal filter in FreqAug-T. Tested on MK200.

As in the top left clip of Fig. 6, videos with a low standard deviation of spectrum intensity over the temporal frequency (σ_t) tend to have temporally varying backgrounds due to rapid camera moving or abrupt scene change, which makes a naive filter hard to remove background. The spectrum intensity will be concentrated on the temporal zero-frequency (red boxes) when the scene change is small over time (right). Otherwise, the spectrum spreads across all temporal frequencies (left). In other words, σ_t gets decreased if many scene transitions exist. For quantitative analysis, we take the logarithm and mean over spatial frequency to the spectrum and then calculate std. over time. As we expected, the background of videos with a small σ_t is often not eliminated, and some traces of other frames are mixed.

The histogram in Fig. 6 shows that around half of the clips in K400 have relatively small σ_t. Then, a question naturally arises about how those clips with small σ_t affect the learning with FreqAug. We argue that the clips with a visually remaining background are also helpful for FreqAug. To support our claim, we conduct a quantitative experiment in Table 7 to confirm the impact of the temporal filter on videos with small σ_t. We study with two variants of FreqAug-T, which exclude the clips of either small σ_t (*i.e.*, under 0.05) or large σ_t (*i.e.*, over 0.05) when applying the filter. The result demonstrates that FreqAug outperforms the baseline with a large margin, even in the case of clips with small σ_t. This implies that the temporal filter enhances the representation of clips with both small and large temporal variations. Therefore, this experiment validates our claim that the role of temporal filtering is not limited to background erasing.

Conclusion

In this paper, we have proposed a simple and effective frequency domain augmentation for video representation learning. FreqAug augments multiple views by randomly removing spatial and temporal low-frequency components from videos so that a model can learn from the essential features. Extensive experiments have shown the effectiveness of FreqAug for various SSL frameworks and diverse backbones on *seven* downstream tasks. Lastly, we analyze the influence of FreqAug on both video SSL and its downstream tasks.

Acknowledgements

This work was supported by NAVER Corporation. The computational work in this study was mostly conducted on NAVER Smart Machine Learning (NSML) platform (Sung et al. 2017; Kim et al. 2018).

References

Alayrac, J.-B.; Recasens, A.; Schneider, R.; Arandjelović, R.; Ramapuram, J.; De Fauw, J.; Smaira, L.; Dieleman, S.; and Zisserman, A. 2020. Self-Supervised MultiModal Versatile Networks. In *NeurIPS*, 25–37.

Alwassel, H.; Mahajan, D.; Korbar, B.; Torresani, L.; Ghanem, B.; and Tran, D. 2020. Self-Supervised Learning by Cross-Modal Audio-Video Clustering. In *NeurIPS*.

Behrmann, N.; Fayyaz, M.; Gall, J.; and Noroozi, M. 2021. Long Short View Feature Decomposition via Contrastive Video Representation Learning. In *ICCV*, 9244–9253.

Caron, M.; Misra, I.; Mairal, J.; Goyal, P.; Bojanowski, P.; and Joulin, A. 2020. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. In *NeurIPS*.

Carreira, J.; and Zisserman, A. 2017. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 4724–4733.

Chen, P.; Huang, D.; He, D.; Long, X.; Zeng, R.; Wen, S.; Tan, M.; and Gan, C. 2021. RSPNet: Relative Speed Perception for Unsupervised Video Representation Learning. In *AAAI*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020a. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML*, 1597–1607.

Chen, X.; Fan, H.; Girshick, R. B.; and He, K. 2020b. Improved Baselines with Momentum Contrastive Learning. *arXiv preprint arXiv:2003.04297*.

Chen, X.; and He, K. 2021. Exploring Simple Siamese Representation Learning. In *CVPR*, 15750–15758.

Dave, I.; Gupta, R.; Rizve, M. N.; and Shah, M. 2021. TCLR: Temporal Contrastive Learning for Video Representation. *arXiv preprint arXiv:2101.07974*.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, S.; Li, M.; Yang, T.; Qian, R.; Xu, H.; Chen, Q.; Wang, J.; and Xiong, H. 2022. Motion-Aware Contrastive Video Representation Learning via Foreground-Background Merging. In *CVPR*, 9716–9726.

Duan, H.; Zhao, N.; Chen, K.; and Lin, D. 2022. TransRank: Self-supervised Video Representation Learning via Ranking-based Transformation Recognition. In *CVPR*.

Fan, H.; Li, Y.; Xiong, B.; Lo, W.-Y.; and Feichtenhofer, C. 2020. PySlowFast. https://github.com/facebookresearch/slowfast. Accessed: 2022-08-01.

Farha, Y. A.; and Gall, J. 2019. MS-TCN: Multi-Stage Temporal Convolutional Network for Action Segmentation. In *CVPR*, 3575–3584.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. SlowFast Networks for Video Recognition. In *ICCV*.

Feichtenhofer, C.; Fan, H.; Xiong, B.; Girshick, R.; and He, K. 2021. A Large-Scale Study on Unsupervised Spatiotemporal Representation Learning. In *CVPR*, 3299–3309.

Goyal, R.; Kahou, S. E.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fründ, I.; Yianilos, P.; Mueller-Freitag, M.; Hoppe, F.; Thurau, C.; Bax, I.; and Memisevic, R. 2017. The ”Something Something” Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*, 5843–5851.

Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; Piot, B.; kavukcuoglu, k.; Munos, R.; and Valko, M. 2020. Bootstrap Your Own Latent - A New Approach to Self-Supervised Learning. In *NeurIPS*, 21271–21284.

Han, T.; Xie, W.; and Zisserman, A. 2020. Self-supervised Co-training for Video Representation Learning. In *NeurIPS*.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR*.

Huang, D.; Wu, W.; Hu, W.; Liu, X.; He, D.; Wu, Z.; Wu, X.; Tan, M.; and Ding, E. 2021a. ASCNet: Self-Supervised Video Representation Learning With Appearance-Speed Consistency. In *ICCV*, 8096–8105.

Huang, L.; Liu, Y.; Wang, B.; Pan, P.; Xu, Y.; and Jin, R. 2021b. Self-supervised Video Representation Learning by Context and Motion Decoupling. In *CVPR*, 13886–13895.

Idrees, H.; Zamir, A. R.; Jiang, Y.-G.; Gorban, A.; Laptev, I.; Sukthankar, R.; and Shah, M. 2017. The THUMOS challenge on action recognition for videos “in the wild”. *Computer Vision and Image Understanding*, 155: 1–23.

Kim, G.; Han, D. K.; and Ko, H. 2021. SpecMix : A Mixed Sample Data Augmentation method for Training with Time-Frequency Domain Features. In *Proc. Interspeech*.

Kim, H.; Kim, M.; Seo, D.; Kim, J.; Park, H.; Park, S.; Jo, H.; Kim, K.; Yang, Y.; Kim, Y.; Sung, N.; and Ha, J. 2018. NSML: Meet the MLaaS platform with a real-world case study. *arXiv preprint arXiv:1810.09957*.

Kim, J.; Cha, S.; Wee, D.; Bae, S.; and Kim, J. 2020. Regularization on Spatio-Temporally Smoothed Feature for Action Recognition. In *CVPR*.

Kuehne, H.; Arslan, A. B.; and Serre, T. 2014. The Language of Actions: Recovering the Syntax and Semantics of Goal-Directed Human Activities. In *CVPR*.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *ICCV*, 2556–2563.

Li, Y.; Li, Y.; and Vasconcelos, N. 2018. RESOUND: Towards Action Recognition without Representation Bias. In *ECCV*.

Miech, A.; Alayrac, J.-B.; Smaira, L.; Laptev, I.; Sivic, J.; and Zisserman, A. 2020. End-to-End Learning of Visual Representations from Uncurated Instructional Videos. In *CVPR*.

Nam, H.; Kim, S.-H.; and Park, Y.-H. 2021. FilterAugment: An Acoustic Environmental Data Augmentation Method. *arXiv preprint arXiv:2110.03282*.

Nam, J.-H.; and Lee, S.-C. 2021. Frequency Filtering for Data Augmentation in X-Ray Image Classification. In *ICIP*, 81–85.

Pan, T.; Song, Y.; Yang, T.; Jiang, W.; and Liu, W. 2021. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *CVPR*, 11205–11214.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. In *Proc. Interspeech*, 2613–2617.

Qian, R.; Meng, T.; Gong, B.; Yang, M.-H.; Wang, H.; Belongie, S.; and Cui, Y. 2021. Spatiotemporal Contrastive Video Representation Learning. In *CVPR*, 6964–6974.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Recasens, A.; Luc, P.; Alayrac, J.; Wang, L.; Strub, F.; Tallec, C.; Malinowski, M.; Patraucean, V.; Altché, F.; Valko, M.; Grill, J.; van den Oord, A.; and Zisserman, A. 2021. Broaden Your Views for Self-Supervised Video Learning. In *ICCV*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. In *ICCV*.

Shao, D.; Zhao, Y.; Dai, B.; and Lin, D. 2020. FineGym: A Hierarchical Video Dataset for Fine-grained Action Understanding. In *CVPR*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Sung, N.; Kim, M.; Jo, H.; Yang, Y.; Kim, J.; Lausen, L.; Kim, Y.; Lee, G.; Kwak, D.; Ha, J.; and Kim, S. 2017. NSML: A Machine Learning Platform That Enables You to Focus on Your Models. *arXiv preprint arXiv:1712.05902*.

Tian, Y.; Sun, C.; Poole, B.; Krishnan, D.; Schmid, C.; and Isola, P. 2020. What Makes for Good Views for Contrastive Learning? In *NeurIPS*, volume 33, 6827–6839.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A Closer Look at Spatiotemporal Convolutions for Action Recognition. In *CVPR*, 6450–6459.

van der Maaten, L.; and Hinton, G. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9(86): 2579–2605.

Wang, H.; Wu, X.; Huang, Z.; and Xing, E. P. 2020. High-Frequency Component Helps Explain the Generalization of Convolutional Neural Networks. In *CVPR*.

Wang, J.; Gao, Y.; Li, K.; Jiang, X.; Guo, X.; Ji, R.; and Sun, X. 2021a. Enhancing unsupervised video representation learning by decoupling the scene and the motion. In *AAAI*.

Wang, J.; Gao, Y.; Li, K.; Lin, Y.; Ma, A. J.; Cheng, H.; Peng, P.; Ji, R.; and Sun, X. 2021b. Removing the Background by Adding the Background: Towards Background Robust Self-supervised Video Representation Learning. In *CVPR*.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 20–36.

Wen, Z.; and Li, Y. 2021. Toward Understanding the Feature Learning Process of Self-supervised Contrastive Learning. In *ICML*, 11112–11122.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised Feature Learning via Non-Parametric Instance Discrimination. In *CVPR*, 3733–3742.

Xiao, F.; Tighe, J.; and Modolo, D. 2021. MoDist: Motion Distillation for Self-supervised Video Representation Learning. *arXiv preprint arXiv:2106.09703*.

Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-off in Video Classification. In *ECCV*, 318–335.

Xu, M.; Pérez-Rúa, J.-M.; Escorcia, V.; Martinez, B.; Zhu, X.; Zhang, L.; Ghanem, B.; and Xiang, T. 2021a. Boundary-sensitive pre-training for temporal localization in videos. In *CVPR*, 7220–7230.

Xu, M.; Zhao, C.; Rojas, D. S.; Thabet, A.; and Ghanem, B. 2020. G-tad: Sub-graph localization for temporal action detection. In *CVPR*, 10156–10165.

Xu, Q.; Zhang, R.; Zhang, Y.; Wang, Y.; and Tian, Q. 2021b. A Fourier-Based Framework for Domain Generalization. In *CVPR*, 14383–14392.

Zhang, M.; Wang, J.; and Ma, A. J. 2022. Suppressing Static Visual Cues via Normalizing Flows for Self-Supervised Video Representation Learning. In *AAAI*, volume 36, 3300–3308.

Zhao, N.; Wu, Z.; Lau, R. W.; and Lin, S. 2021. What Makes Instance Discrimination Good for Transfer Learning? In *ICLR*.