Tubelet-Contrastive Self-Supervision for Video-Efficient Generalization

Fida Mohammad Thoker, Hazel Doughty, Cees G. M. Snoek
University of Amsterdam

Abstract

We propose a self-supervised method for learning motion-focused video representations. Existing approaches minimize distances between temporally augmented videos, which maintain high spatial similarity. We instead propose to learn similarities between videos with identical local motion dynamics but an otherwise different appearance. We do so by adding synthetic motion trajectories to videos which we refer to as tubelets. By simulating different tubelet motions and applying transformations, such as scaling and rotation, we introduce motion patterns beyond what is present in the pretraining data. This allows us to learn a video representation that is remarkably data efficient: our approach maintains performance when using only 25% of the pretraining videos. Experiments on 10 diverse downstream settings demonstrate our competitive performance and generalizability to new domains and fine-grained actions. Code is available at https://github.com/fmthoker/tubelet-contrast.

1. Introduction

This paper aims to learn self-supervised video representations, useful for distinguishing actions. In a community effort to reduce the manual, expensive, and hard-to-scale annotations needed for many downstream deployment settings, the topic has witnessed tremendous progress in recent years [18, 31, 62, 79], particularly through contrastive learning [15,56,58,60]. Contrastive approaches learn representations through instance discrimination [55], by increasing feature similarity between spatially and temporally augmented clips from the same video. Despite temporal differences, such positive video pairs often maintain high spatial similarity (see Figure 1), allowing the contrastive task to be solved by coarse-grained features without explicitly capturing local motion dynamics. This limits the generalizability of the learned video representations, as shown in our prior work [70]. Furthermore, prior approaches are constrained by the amount and types of motions present in the pretraining data. This makes them data-hungry, as video data has high redundancy with periods of little to no motion. In this work, we address the need for data-efficient and generalizable self-supervised video representations by proposing a contrastive method to learn local motion dynamics.

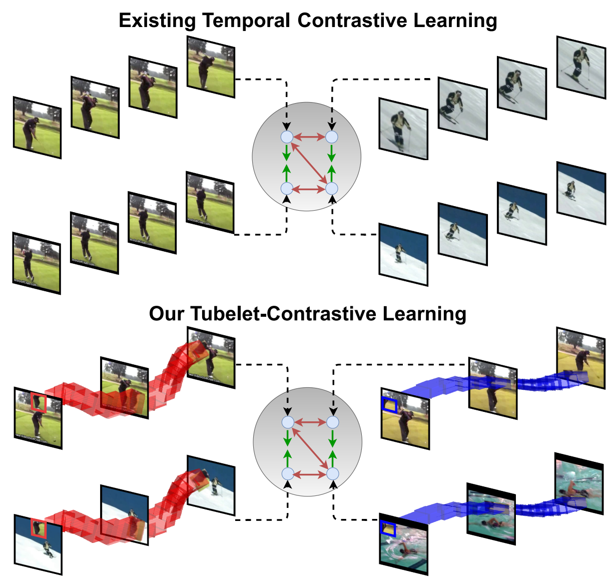

Figure 1: **Tubelet-Contrastive Positive Pairs** (bottom) only share the spatiotemporal motion dynamics inside the simulated tubelets, while temporal contrastive pairs (top) suffer from a high spatial bias. Contrasting tubelets results in a data-efficient and generalizable video representation.

We take inspiration from action detection, where tubelets are used to represent the motions of people and objects in videos through bounding box sequences *e.g.*, [29, 32, 42]. Typically, many tubelet proposals are generated for a video, which are processed to find the best prediction. Rather than finding tubelets in video data, we simulate them. In particular, we sample an image patch and 'paste' it with a randomized motion onto two different video clips as a shared tubelet (see Figure 1). These two clips form a positive pair for contrastive learning where the model has to rely on the spatiotemporal dynamics of the tubelet to learn the similarity. With such a formulation, we can simulate a large variety of motion patterns that are not present in the original videos. This allows our model to be data-efficient while improving generalization to new domains and fine-grained actions.

We make four contributions. First, we explicitly learn from local motion dynamics in the form of synthetic tubelets and design a simple but effective tubelet-contrastive framework. Second, we propose different ways of simulating tubelet motion and transformations to generate a variety of motion patterns for learning. Third, we reveal the remarkable data efficiency of our proposal: on five action recognition datasets our approach maintains performance when using only 25% of the pretraining videos. What is more, with only 5-10% of the videos we still outperform the vanilla contrastive baseline with 100% pretraining data for several datasets. Fourth, our comparative experiments on 10 downstream settings, including UCF101 [67], HMDB51 [37], Something Something [20], and FineGym [63], further demonstrate our competitive performance, generalizability to new domains, and suitability of our learned representation for fine-grained actions.

2. Related Work

Self-Supervised Video Representation Learning. The success of contrastive learning in images [6, 21, 23, 52] inspired many video contrastive works [27,45,56,58,60,69]. Alongside spatial invariances, these works learn invariances to temporal crops [56, 60, 61] and video speed [27, 45, 58]. Some diverge from temporal invariances and encourage equivariance [8, 57] to learn finer temporal representations. For instance, TCLR [8] enforces within-instance temporal feature variation, while TE [30] learns equivariance to temporal crops and speed with contrastive learning. Alternatively, many works learn to predict temporal transformations such as clip order [18, 39, 50, 79], speed [4, 7, 82] and their combinations [31, 47]. These self-supervised temporal representations are effective for classifying and retrieving coarse-grained actions but are challenged by downstream settings with subtle motions [62, 70]. Other works utilize the multimodal nature of videos [1, 2, 19, 22, 48, 51, 57] and learn similarity with audio [1, 2, 51] and optical flow [19, 22, 54, 77]. We contrast motions of synthetic tubelets to learn a video representation from only RGB data that can generalize to tasks requiring fine-grained motion understanding.

Other self-supervised works learn from the spatiotemporal dynamics of video. Both BE [75] and FAME [9] remove background bias by adding static frames [75] or replacing the background [9] in positive pairs. Several works instead use masked autoencoding to learn video representations [13, 71]. However, these works are all limited to the motions present in the pretraining dataset. We prefer to be less dataset-dependent and generate synthetic motion tubelets for contrastive learning, which also offers a considerable data-efficiency benefit. CtP [74] and MoSI [28] both aim to predict motions in pretraining. CtP [74] learns to track image patches in video clips while MoSI [28] learns to predict the speed and direction of added pseudo-motions. We take inspiration from these works and contrast synthetic motions from tubelets which allows us to learn generalizable and data-efficient representations.

Supervised Fine-Grained Motion Learning. While self-supervised works have mainly focused on learning representations to distinguish coarse-grained actions, much progress has been made in supervised learning of motions. Approaches distinguish actions by motion-focused neural network blocks [36, 38, 43, 48], decoupling motion from appearance [40, 68], aggregating multiple temporal scales [14, 53, 80], and sparse coding to obtain a mid-level motion representation [49, 59, 64]. Other works exploit skeleton data [12, 24] or optical flow [16, 66]. Alternatively, several works identify motion differences within an action class, by repetition counting [26, 84, 85], recognizing adverbs [10, 11] or querying for action attributes [83]. Different from all these works, we learn a motion-sensitive video representation with self-supervision. We do so by relying on just coarse-grained video data in pretraining and demonstrate downstream generalization to fine-grained actions.

Tubelets. Jain *et al.* defined tubelets as class-agnostic sequences of bounding boxes over time [29]. Tubelets can represent the movement of people and objects and are commonly used for object detection in videos [17, 33, 34], spatiotemporal action localization [25, 29, 32, 42, 81, 86] and video relation detection [5]. Initially, tubelets were obtained by supervoxel groupings and dense trajectories [29, 73] and later from 2D CNNs [32, 42], 3D CNNs [25, 81] and transformers [86]. We introduce (synthetic) tubelets of pseudo-objects for contrastive video self-supervised learning.

3. Tubelet Contrast

We aim to learn motion-focused video representations from RGB video data with self-supervision. After revisiting temporal contrastive learning, we propose tubelet-contrastive learning to reduce the spatial focus of video representations and instead learn similarities between spatio-temporal tubelet dynamics (Section 3.1). We encourage our representation to be motion-focused by simulating a variety of tubelet motions (Section 3.2). To further improve data efficiency and generalizability, we add complexity and variety to the motions through tubelet transformations (Section 3.3). Figure 2 shows an overview of our approach.

Temporal Contrastive Learning. Temporal contrastive learning learns feature representations via instance discrimination [55]. This is achieved by maximizing the similarity between augmented clips from the same video (positive pairs) and minimizing the similarity between clips from different videos (negatives). Concretely given a set of videos V, the positive pairs (v, v') are obtained by sampling different temporal crops of the same video [56, 58] and applying spatial augmentations such as cropping and color jittering. Clips sampled from other videos in the training set act as negatives. The extracted clips are passed through a video encoder and projected on a representation space by a non-linear projection head to obtain clip embeddings $(Z_v, Z_{v'})$.

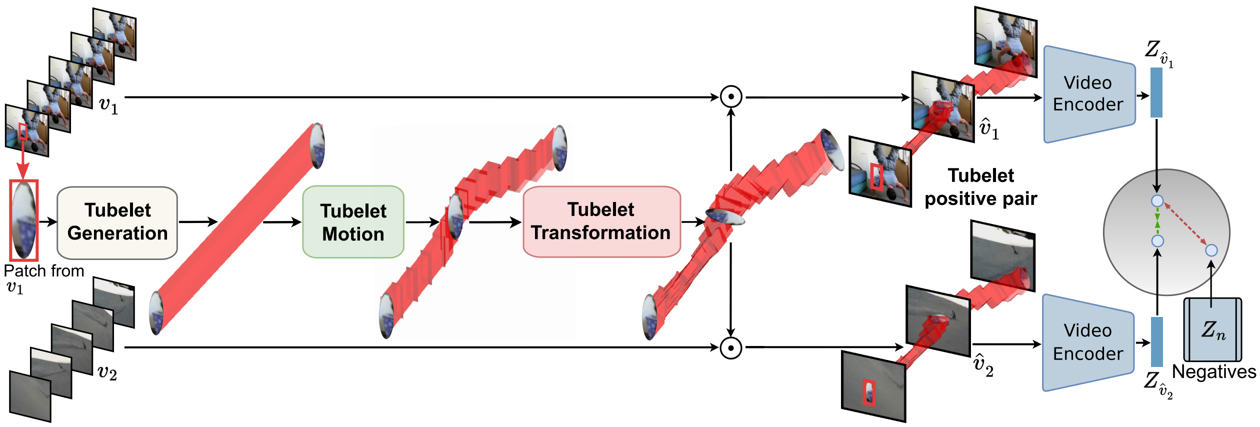

Figure 2: **Tubelet-Contrastive Learning.** We sample two clips (v_1, v_2) from different videos and randomly crop an image patch from v_1. We generate a tubelet by replicating the patch in time and add motion through a sequence of target locations for the patch. We then add complexity to these motions by applying transformations, such as rotation, to the tubelet. The tubelet is overlaid $\odot$ onto both clips to form a positive tubelet pair $(\hat{v}_1, \hat{v}_2)$. We learn similarities between clips with the same tubelets (positive pairs) and dissimilarities between clips with different tubelets (negatives) using a contrastive loss.

The noise contrastive estimation loss InfoNCE [55] is used for the optimization:

$$\mathrm{L}_{contrast}(v, v') = -\log \frac{h(Z_v, Z_{v'})}{h(Z_v, Z_{v'}) + \sum_{Z_n \sim \mathcal{N}} h(Z_v, Z_n)} \quad (1)$$

where $h(Z_v, Z_{v'})$=exp$(Z_v \cdot Z_{v'}/\tau)$, τ is the temperature parameter and $\mathcal{N}$ is a set of negative clip embeddings.

3.1. Tubelet-Contrastive Learning

Different from existing video contrastive self-supervised methods, we explicitly aim to learn motion-focused video representations while relying only on RGB data. To achieve this we propose to learn similarities between simulated tubelets. Concretely, we first generate tubelets in the form of moving patches which are then overlaid onto two different video clips to generate positive pairs that have a high motion similarity and a low spatial similarity. Such positive pairs are then employed to learn video representations via instance discrimination, allowing us to learn more generalizable and motion-sensitive video representations.

Tubelet Generation. We define a tubelet as a sequence of object locations in each frame of a video clip. Let's assume an object p of size $H' \times W'$ moving in a video clip v of length T. Then the tubelet is defined as follows:

$$\mathrm{Tubelet}_p = [(x^1, y^1), .., (x^T, y^T)], \quad (2)$$

where (x^i, y^i) is the center coordinate of the object p in frame i of clip v. For this work, a random image patch of size $H' \times W'$ acts as a pseudo-object overlaid on a video clip to form a tubelet. To generate the tubelet we first make the object appear static, *i.e.*, x^1=x^2=...=x^T and y^1=y^2 = ...=y^T, and explain how we add motion in Section 3.2.

Tubelet-Contrastive Pairs. To create contrastive tubelet pairs, we first randomly sample clips v_1 and v_2 of size $H \times W$ and length T from two different videos in V. From v_1 we randomly crop an image patch p of size $H' \times W'$, such that $H' \ll H$ and $W' \ll W$. From the patch p, we construct a tubelet $\mathrm{Tubelet}_p$ as in Eq. (2). Then, we overlay the generated tubelet $\mathrm{Tubelet}_p$ onto both v_1 and v_2 to create two modified video clips $\hat{v}_1$ and $\hat{v}_2$:

$$\hat{v}_1 = v_1 \odot \mathrm{Tubelet}_p \qquad \hat{v}_2 = v_2 \odot \mathrm{Tubelet}_p, \quad (3)$$

where $\odot$ refers to pasting patch p in each video frame at locations determined by $\mathrm{Tubelet}_p$. Eq. (3) can be extended for a set of M tubelets $\{\mathrm{Tubelet}_{p_1}, ..., \mathrm{Tubelet}_{p_M}\}$ from M patches randomly cropped from v_1 as:

$$\hat{v}_1 = v_1 \odot \{\mathrm{Tubelet}_{p_1}, ..., \mathrm{Tubelet}_{p_M}\}$$
$$\hat{v}_2 = v_2 \odot \{\mathrm{Tubelet}_{p_1}, ..., \mathrm{Tubelet}_{p_M}\}. \quad (4)$$

As a result, $\hat{v}_1$ and $\hat{v}_2$ share the spatiotemporal dynamics of the moving patches in the form of tubelets and have low spatial bias since the two clips come from different videos. Finally, we adapt the contrastive loss from Eq. (1) and apply $\mathcal{L}_{contrast}(\hat{v}_1, \hat{v}_2)$. Here the set of negatives $\mathcal{N}$ contains videos with different tubelets. Since the only similarity in positive pairs is the tubelets, the network must rely on temporal cues causing a motion-focused video representation.

3.2. Tubelet Motion

To learn motion-focused video representations, we need to give our tubelets motion variety. Here, we discuss how to simulate motions by generating different patch movements in the tubelets. Recall, Eq. (2) defines a tubelet by image patch p and its center coordinate in each video frame. We consider two types of tubelet motion: linear and non-linear.

Linear Motion. We randomly sample the center locations for the patch in K keyframes: the first frame (i=1), the last frame (i=T), and $K-2$ randomly selected frames. These patch locations are sampled from uniform distributions $x \in [0, W]$ and $y \in [0, H]$, where W and H are the video width and height. Patch locations for the remaining frames $i \notin K$ are then linearly interpolated between keyframes so we

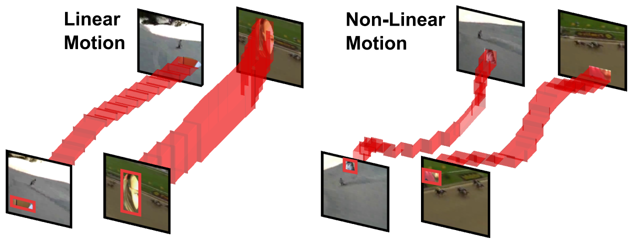

Figure 3: **Tubelet Motion.** Examples for *Linear* (left) and *Non-Linear* (right). Non-linear motions enable the simulation of a larger variety of motion patterns to learn from.

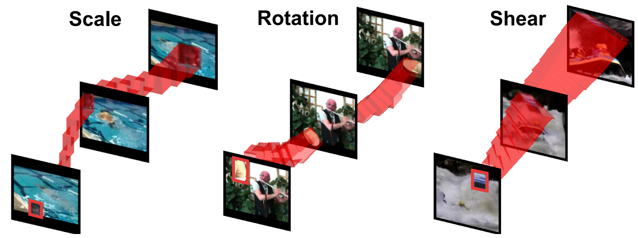

Figure 4: **Tubelet Transformation.** Examples for *Scale* (left), *Rotation* (middle), and *Shear* (right). The patch is transformed as it moves through the tubelet.

obtain the following linear motion definition:

$$\text{Tubelet}^{\text{Lin}} = [(x^1, y^1), (x^2, y^2), ..., (x^T, y^T)], \text{ s.t.} \quad (5)$$

$$(x^i, y^i) = \begin{cases} (\mathcal{U}(0, W), \mathcal{U}(0, H)), & \text{if } i \in K \\ \text{Interp}((x^k, y^k), (x^{k+1}, y^{k+1})), & \text{otherwise} \end{cases}$$

where $\mathcal{U}$ is a function for uniform sampling, k and $k+1$ are the neighboring keyframes to frame i and Interp gives a linear interpolation between keyframes. To ensure smoothness, we constrain the difference between the center locations in neighboring keyframes to be less than Δ pixels. This formulation results in tubelet motions where patches follow linear paths across the video frames. The left of Figure 3 shows examples of such linear tubelet motions.

Non-Linear Motion. Linear motions are simple and limit the variety of motion patterns that can be generated. Next, we simulate motions where patches move along more complex non-linear paths, to better emulate motions in real videos. We create non-linear motions by first sampling N 2D coordinates ($N \gg T$) uniformly from $x \in [0, W]$ and $y \in [0, H]$. Then, we apply a $1D$ Gaussian filter along x and y axes to generate a random smooth nonlinear path as:

$$\text{Tubelet}^{\text{NonLin}} = [(g(x^1), g(y^1)), ..., (g(x^N), g(y^N))]$$
$$\text{s.t.} \quad g(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{-z^2/2\sigma^2} \quad (6)$$

where σ is the smoothing factor for the gaussian kernels. Note the importance of sampling $N \gg T$ points to ensure a non-linear path. If N is too small then the path becomes linear after gaussian smoothing. We downsample the resulting non-linear tubelet in Eq. (6) from N to T coordinates resulting in the locations for patch p in the T frames. The right of Figure 3 shows examples of non-linear tubelet motions.

3.3. Tubelet Transformation

The tubelet motions are simulated by changing the position of the patch across the frames in a video clip, *i.e.* with translation. In reality, the motion of objects in space may appear as other transformations in videos, for instance, scale decreasing as the object moves away from the camera or motions due to planer rotations. Motivated by this, we propose to add more complexity and variety to the simulated motions by transforming the tubelets. In particular, we propose scale, rotation, and shear transformations. As before, we sample keyframes K with the first (i=0) and last frames (i=T) always included. Transformations for remaining frames are linearly interpolated. Formally, we define a tubelet transformation as a sequence of spatial transformations applied to the patch p in each frame i as:

$$\text{Trans}_F = [p, F(p, \theta^2), ..., .., F(p, \theta^T)], \text{ s.t.}$$
$$\theta^i = \begin{cases} \mathcal{U}(\text{Min}, \text{Max}), & \text{if } i \in K \\ \text{Interp}(\theta^k, \theta^{k+1}), & \text{otherwise} \end{cases} \quad (7)$$

where $F(p, \theta^i)$ applies the transformation to patch p according to parameters θ^i, $\mathcal{U}$ samples from a uniform distribution and θ^k and θ^{k+1} are the parameters for the keyframes neighboring frame i. For the first keyframe, no transformation is applied thus representing the initial state of the patch p. We instantiate three types of such tubelet transformations: scale, rotation, and shear. Examples are shown in Figure 4.

Scale. We scale the patch across time with $F(p, \theta^i)$ and horizontal and vertical scaling factors θ^i=(w^i, h^i). To sample w^i and h^i, we use Min=0.5 and Max=1.5.

Rotation. In this transformation $F(p, \theta^i)$ applies in-plane rotations to tubelet patches. Thus, θ^i is a rotation angle sampled from Min=$-90°$ and Max=$+90°$.

Shear. We shear the patch as the tubelet progresses with $F(p, \theta^i)$. The shearing parameters are θ^i=(r^i, s^i) which are sampled using Min=-1.5 and Max=1.5.

With these tubelet transformations and the motions created in Section 3.2 we are able to simulate a variety of subtle motions in videos, making the model data-efficient. By learning the similarity between the same tubelet overlaid onto different videos, our model pays less attention to spatial features, instead learning to represent these subtle motions. This makes the learned representation generalizable to different domains and action granularities.

4. Experiments

4.1. Datasets, Evaluation & Implementation

Pretraining Datasets. Following prior work [8, 27, 56–58, 74] we use **Kinetics-400** [35] for self-supervised pretraining. Kinetics-400 is a large-scale action recognition dataset containing 250K videos of 400 action classes. To show data

Evaluation Factor	Experiment	Dataset	Task	#Classes	#Finetuning	#Testing	Eval Metric
Standard	UCF101	UCF 101 [67]	Action Recognition	101	9,537	3,783	Top-1 Accuracy
	HMDB51	HMDB 51 [37]	Action Recognition	51	3,570	1,530	Top-1 Accuracy
Domain Shift	SSv2	Something-Something [20]	Action Recognition	174	168,913	24,777	Top-1 Accuracy
	Gym99	FineGym [63]	Action Recognition	99	20,484	8,521	Top-1 Accuracy
Sample Efficiency	UCF (10^3)	UCF 101 [67]	Action Recognition	101	1,000	3,783	Top-1 Accuracy
	Gym (10^3)	FineGym [63]	Action Recognition	99	1,000	8,521	Top-1 Accuracy
Action Granularity	FX-S1	FineGym [63]	Action Recognition	11	1,882	777	Mean Class Acc
	UB-S1	FineGym [63]	Action Recognition	15	3,511	1,471	Mean Class Acc
Task Shift	UCF-RC	UCFRep [84]	Repetition Counting	-	421	105	Mean Error
	Charades	Charades [65]	Multi-label Recognition	157	7,985	1,863	mAP

Table 1: **Benchmark Details** for the downstream evaluation factors, experiments, and datasets we cover. For non-standard evaluations, we follow the SEVERE benchmark [70]. For self-supervised pretraining, we use Kinetics-400 or Mini-Kinetics.

efficiency, we also pretrain with **Mini-Kinetics** [78], a subset containing 85K videos of 200 action classes.

Downstream Evaluation. To evaluate the video representations learned by our tubelet contrast, we finetune and evaluate our model on various downstream datasets summarized in Table 1. Following previous self-supervised work, we evaluate on standard benchmarks: **UCF101** [67] and **HMDB51** [37]. These action recognition datasets contain coarse-grained actions with domains similar to Kinetics-400. For both, we report top-1 accuracy on split 1 from the original papers. We examine the generalizability of our model with the **SEVERE** benchmark proposed in our previous work [70]. This consists of eight experiments over four downstream generalization factors: *domain shift*, *sample efficiency*, *action granularity*, and *task shift*. *Domain shift* is evaluated on Something-Something v2 [20] (SSv2) and FineGym [63] (Gym99) which vary in domain relative to Kinetics-400. *Sample efficiency* evaluates low-shot action recognition on UCF101 [67] and FineGym [63] with 1,000 training samples, referred to as UCF (10^3) and Gym (10^3). *Action granularity* evaluates semantically similar actions using FX-S1 and UB-S1 subsets from FineGym [63]. In both subsets, action classes belong to the same element of a gymnastic routine, *e.g.*, FX-S1 is types of jump. *Task shift* evaluates tasks beyond single-label action recognition. Specifically, it uses temporal repetition counting on UCFRep [84], a subset of UCF-101 [84], and multi-label action recognition on Charades [65]. The experimental setups are detailed in Table 1 and all follow SEVERE [70].

Tubelet Generation and Transformation. Our clips are 16 112×112 frames with standard spatial augmentations: random crops, horizontal flip, and color jitter. We randomly crop 2 patches to generate M=2 tubelets (Eq. 4). The patch size $H^{'} \times W^{'}$ is uniformly sampled from $[16{\times}16, 64{\times}64]$. We also randomly sample a patch shape from a set of predefined shapes. For linear motions, we use Δ=[40−80] displacement difference. For non-linear motion, we use N=48 and a smoothing factor of σ=8 (Eq. 6). For linear motion and all tubelet transformations, we use K=3 keyframes.

	UCF (10^3)	Gym (10^3)	SSv2-Sub	UB-S1
Temporal Contrast				
Baseline	57.5	29.5	44.2	84.8
Tubelet Contrast				
Tubelet Generation	48.2	28.2	40.1	84.1
Tubelet Motion	63.0	45.6	47.5	90.3
Tubelet Transformation	65.5	48.0	47.9	90.9

Table 2: **Tubelet-Contrastive Learning** considerably outperforms temporal contrast on multiple downstream settings. Tubelet motion and transformations are key.

Networks, Pretraining and Finetuning. We use R(2+1)D-18 [72] as the video encoder, following previous self-supervision works [8, 9, 56–58, 76]. The projection head is a 2-layer MLP with 128D output. We use momentum contrast [23] to increase the number of negatives $|\mathcal{N}|$ (Eq. 1) to 16,384 for Mini-Kinetics and 65,536 for Kinetics. We use temperature τ=0.2 (Eq. 1). The model is optimized using SGD with momentum 0.9, learning rate 0.01, and weight decay 0.0001. We use a batch size of 32 for Mini-Kinetics and 128 for Kinetics, a cosine scheduler [46], and pretrain for 100 epochs. After pretraining, we replace the projection head with a task-dependent head as in SEVERE [70] and finetune the whole network with labels for the downstream task. We provide finetuning details in the supplementary.

4.2. Ablation Studies & Analysis

To ablate the effectiveness of individual components we pretrain on Mini-Kinetics and evaluate on UCF (10^3), Gym (10^3), Something-Something v2 and UB-S1. To decrease the finetuning time we use a subset of Something Something (SSv2-Sub) with 25% of the training data (details in supplementary). Unless specified otherwise, we use non-linear motion and rotation to generate tubelets.

Tubelet-Contrastive Learning. Table 2 shows the benefits brought by our tubelet-contrastive learning. We first observe that our full tubelet-contrastive model improves considerably over the temporal contrastive baseline, which uses MoCo [23] with a temporal crop augmentation. This

Tubelet Motion	UCF (10^3)	Gym (10^3)	SSv2-Sub	UB-S1
No motion	48.2	28.2	40.1	84.1
Linear	55.5	34.6	45.3	88.5
Non-Linear	63.0	45.6	47.5	90.3

Table 3: **Tubelet Motions.** Learning from tubelets with non-linear motion benefits multiple downstream settings.

Transformation	UCF (10^3)	Gym (10^3)	SSv2-Sub	UB-S1
None	63.0	45.6	47.5	90.5
Scale	65.1	46.5	47.0	90.5
Shear	65.2	47.5	47.3	90.9
Rotation	65.5	48.0	47.9	90.9

Table 4: **Tubelet Transformation.** Adding motion patterns to tubelet-contrastive learning through transformations improves downstream performance. Best results for rotation.

#Tubelets	UCF (10^3)	Gym (10^3)	SSv2-Sub	UB-S1
1	62.0	39.5	47.1	89.5
2	65.5	48.0	47.9	90.9
3	66.5	46.0	47.5	90.9

Table 5: **Number of Tubelets.** Overlaying two tubelets in positive pairs improves downstream performance.

improvement applies to all downstream datasets but is especially observable with Gym (10^3) (+18.5%) and UB-S1 (+6.1%) where temporal cues are crucial. Our model is also effective on UCF (10^3) (+8.0%) where spatial cues are often as important as temporal ones. These results demonstrate that learning similarities between synthetic tubelets produces generalizable, but motion-focused, video representations required for finer temporal understanding.

It is clear that the motion within tubelets is critical to our model's success as contrasting static tubelets obtained from our tubelet generation (Section 3.1) actually decreases the performance from the temporal contrast baseline. When tubelet motion is added (Section 3.2), performance improves considerably, *e.g.*, Gym (10^3) +17.4% and SSv2-Sub +7.4%. Finally, adding more motion types via tubelet transformations (Section 3.3) further improves the video representation quality, *e.g.*, UCF (10^3) +2.5% and Gym (10^3) +2.4%. This highlights the importance of including a variety of motions beyond what is present in the pretraining data to learn generalizable video representations.

Tubelet Motions. Next, we ablate the impact of the tubelet motion type (Section 3.2) without transformations. We compare the performance of static tubelets with no motion, linear motion, and non-linear motion in Table 3. Tubelets with simple linear motion already improve performance for all four datasets, *e.g.*, +6.4% on Gym (10^3). Using non-linear motion further improves results, for instance with an additional +11.0% improvement on Gym (10^3). We conclude that learning from non-linear motions provides more generalizable video representations.

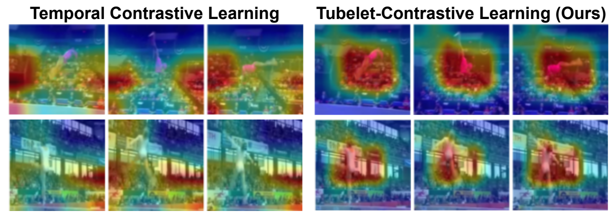

Figure 5: **Class-Agnostic Activation Maps without Finetuning** for the temporal contrastive baseline and our tubelet-contrast. Our model better attends to regions with motion.

	Linear Classification		Finetuning	
	UCF101	Gym99	UCF101	Gym99
Temporal Contrast	**58.9**	19.7	87.1	90.8
Tubelet Contrast	30.0	**34.1**	**91.0**	**92.8**

Table 6: **Appearance vs Motion**. Our method learns to capture motion dynamics with pretraining and can easily learn appearance features with finetuning.

Tubelet Transformation. Table 4 compares the proposed tubelet transformations (Section 3.3). All four datasets benefit from transformations, with rotation being the most effective. The differences in improvement for each transformation are likely due to the types of motion present in the downstream datasets. For instance, Gym (10^3) and UB-S1 contain gymnastic videos where actors are often spinning and turning but do not change in scale due to the fixed camera, therefore rotation is more helpful than scaling. We also experiment with combinations of transformations in supplementary but observe no further improvement.

Number of Tubelets. We investigate the number of tubelets used in each video in Table 5. One tubelet is already more effective than temporal contrastive learning, *e.g.*, 29.5% vs. 39.5% for Gym (10^3). Adding two tubelets improves accuracy on all datasets, *e.g.*, +8.5% for Gym (10^3).

Analysis of Motion-Focus. To further understand what our model learns, Figure 5 visualizes the class agnostic activation maps [3] without finetuning for the baseline and our approach. We observe that even without previously seeing any FineGym data, our approach attends better to the motions than the temporal contrastive baseline, which attends to the background regions. This observation is supported by the linear classification and finetuning results on UCF101 (appearance-focused) and Gym99 (motion-focused) in Table 6. When directly predicting from the learned features with linear classification, our model is less effective than temporal contrast for appearance-based actions in UCF101, but positively affects actions requiring fine-grained motion understanding in Gym99. With finetuning, our tubelet-contrastive representation is able to add spatial appearance understanding and maintain its ability to capture temporal motion dynamics, thus it benefits both UCF101 and Gym99.

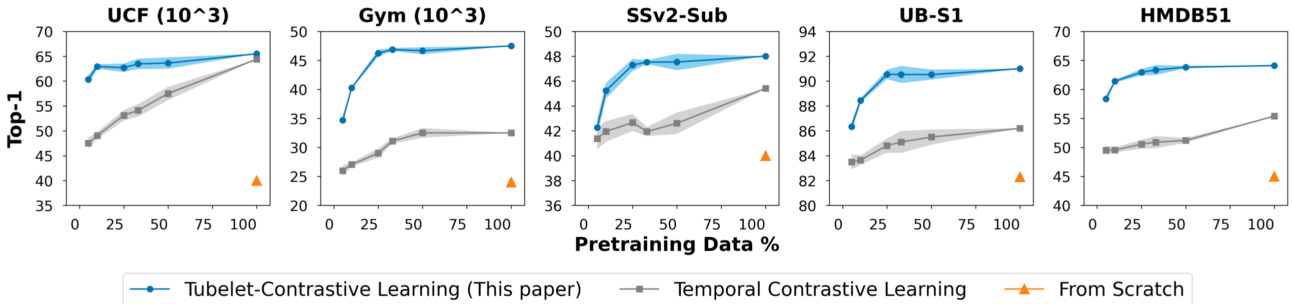

Figure 6: **Video-Data Efficiency of Tubelet-Contrastive Learning.** Our approach maintains performance when using only 25% of the pretraining data. When using 5% of the pretraining data, our approach is still more effective than using 100% with the baseline for Gym (10^3), UB-S1, and HMDB51. Results are averaged over three pretraining runs with different seeds.

4.3. Video-Data Efficiency

To demonstrate our method's data efficiency, we pretrain using subsets of the Kinetics-400. In particular, we sample $5\%, 10\%, 25\%, 33\%$ and 50% of the Kinetics-400 training set with three random seeds and pretrain our model and the temporal contrastive baseline. We compare the effectiveness of these representations after finetuning on UCF (10^3), Gym(10^3), SSv2-Sub, UB-S1, and HMDB51 in Figure 6. On all downstream setups, our method maintains similar performance when reducing the pretraining data to just 25%, while the temporal contrastive baseline performance decreases significantly. Our method is less effective when using only 5% or 10% of the data, but remarkably still outperforms the baseline trained with 100% data for Gym (10^3), UB-S1, and HMDB. We attribute our model's data efficiency to the tubelets we add to the pretraining data. In particular, our non-linear motion and transformations generate a variety of synthetic tubelets that simulate a greater variety of fine-grained motions than are present in the original data.

4.4. Standard Evaluation: UCF101 and HMDB51

We first show the effectiveness of our proposed method on standard coarse-grained action recognition benchmarks UCF101 and HMBD51, where we compare with prior video self-supervised works. For a fair comparison, we only report methods in Table 7 that use the R(2+1)D-18 backbone and Kinetics-400 as the pretraining dataset.

First, we observe that our method obtains the best results for UCF101 and HMDB51. The supplementary material shows we also achieve similar improvement with the R3D and I3D backbones. In particular, with R(2+1)D our method beats CtP [74] by 2.6% and 2.4%, TCLR [8] by 2.8% and 4.1%, and TE [30] by 2.8% and 1.9% all of which aim to learn finer temporal representations. This confirms that explicitly contrasting tubelet-based motion patterns results in a better video representation than learning temporal distinctiveness or prediction. We also outperform FAME [9] by 6.2% and 9.6% on UCF101 and HMDB51. FAME aims to learn a motion-focus representation by pasting the foreground region of one video onto the background of another to construct positive pairs for contrastive learning. We however are not limited by the motions present in the set of pretraining videos as we simulate new motion patterns for learning. We also outperform prior multi-modal works which incorporate audio or explicitly learn motion from optical flow. Since our model is data-efficient, we can pretrain on Mini-Kinetics and still outperform all baselines which are trained on the 3x larger Kinetics-400.

Method	**Modality**	UCF101	HMDB51
VideoMoCo [56]	RGB	78.7	49.2
RSPNet [58]	RGB	81.1	44.6
SRTC [45]	RGB	82.0	51.2
FAME [9]	RGB	84.8	53.5
MCN [44]	RGB	84.8	54.5
AVID-CMA [51]	RGB+Audio	87.5	60.8
TCLR [8]	RGB	88.2	60.0
TE [30]	RGB	88.2	62.2
CtP [74]	RGB	88.4	61.7
MotionFit [19]	RGB+Flow	88.9	61.4
GDT [57]	RGB+Audio	89.3	60.0
This paper †	RGB	**90.7**	**65.0**
This paper	RGB	**91.0**	**64.1**

Table 7: **Standard Evaluation: UCF101 and HMDB51** using R(2+1)D. Gray lines indicate use of additional modalities during self-supervised pretraining. Note that our method pretrained on Mini-Kinetics (†) outperforms all methods which pretrain on the $3\times$ larger Kinetics-400.

4.5. SEVERE Generalization Benchmark

Next, we compare to prior works on the challenging SEVERE benchmark [70], which evaluates video representations for generalizability in *domain shift*, *sample efficiency*, *action granularity*, and *task shift*. We follow the same setup as in the original SEVERE benchmark and use an R(2+1)D-18 backbone pretrained on Kinetics-400 with our tubelet-contrast before finetuning on the different downstream settings. Results are shown in Table 8.

Domain Shift. Among the evaluated methods our proposal achieves the best results on SSv2 and Gym99. These datasets differ considerably from Kinetics-400, particularly

		Domains		Samples		Actions		Tasks			
	Backbone	SSv2	Gym99	UCF (10^3)	Gym (10^3)	FX-S1	UB-S1	UCF-RC↓	Charades	**Mean**	**Rank↓**
SVT [61]	ViT-B	59.2	62.3	83.9	18.5	35.4	55.1	0.421	35.5	51.0	8.9
VideoMAE [71]	ViT-B	69.7	85.1	77.2	27.5	37.0	78.5	0.172	12.6	58.1	8.3
Supervised [72]	R(2+1)D-18	60.8	92.1	86.6	51.3	79.0	87.1	0.132	23.5	70.9	3.9
None	R(2+1)D-18	57.1	89.8	38.3	22.7	46.6	82.3	0.217	7.9	52.9	11.6
SeLaVi [2]	R(2+1)D-18	56.2	88.9	69.0	30.2	51.3	80.9	0.162	8.4	58.6	11.0
MoCo [23]	R(2+1)D-18	57.1	90.7	60.4	30.9	65.0	84.5	0.208	8.3	59.5	9.1
VideoMoCo [56]	R(2+1)D-18	59.0	90.3	65.4	20.6	57.3	83.9	0.185	10.5	58.6	9.1
Pre-Contrast [69]	R(2+1)D-18	56.9	90.5	64.6	27.5	66.1	86.1	0.164	8.9	60.5	9.0
AVID-CMA [51]	R(2+1)D-18	52.0	90.4	68.2	33.4	68.0	87.3	0.148	8.2	61.6	9.0
GDT [57]	R(2+1)D-18	58.0	90.5	**78.4**	45.6	66.0	83.4	**0.123**	8.5	64.8	8.6
RSPNet [58]	R(2+1)D-18	59.0	91.1	74.7	32.2	65.4	83.6	0.145	9.0	62.6	8.0
TCLR [8]	R(2+1)D-18	59.8	91.6	72.6	26.3	60.7	84.7	0.142	**12.2**	61.7	7.6
CtP [74]	R(2+1)D-18	59.6	92.0	61.0	32.9	79.1	88.8	0.178	9.6	63.2	5.6
This paper †	R(2+1)D-18	59.4	92.2	65.5	**48.0**	78.3	90.9	0.150	9.0	66.0	5.4
This paper	R(2+1)D-18	**60.2**	**92.8**	65.7	47.0	**80.1**	**91.0**	0.150	10.3	**66.5**	**4.1**

Table 8: **SEVERE Generalization Benchmark.** Comparison with prior self-supervised methods for generalization to downstream domains, fewer samples, action granularity, and tasks. ↓ indicates lower is better. Results for baselines are taken from SEVERE [70]. Our method generalizes best, even when using the 3x smaller Mini-Kinetics dataset (†) for pretraining.

in regard to the actions, environment and viewpoint. Our improvement demonstrates that the representation learned by our tubelet-contrast is robust to various domain shifts.

Sample Efficiency. For sample efficiency, we achieve a good gain over all prior works on Gym (10^3), *e.g.*, +20.7% over TCLR [8] and +14.1% over CtP [74]. Notably, the gap between the second best method GDT [57] and all others is large, demonstrating the challenge. For UCF (10^3), our method is on par with VideoMoCo [56] and CtP but is outperformed by GDT and RSPNet [58]. This is likely due to most actions in UCF101 requiring more spatial than temporal understanding, thus it benefits from the augmentations used by GDT and RSPNet. Our motion-focused representation requires more finetuning samples on such datasets.

Action Granularity. For fine-grained actions in FX-S1 and UB-S1, our method achieves the best performance, even outperforming supervised Kinetics-400 pretraining. We achieve a considerable improvement over other RGB-only models, *e.g.*, +19.6% and +6.3% over TCLR, as well as audio-visual models, *e.g.*, +14.1% and +7.6% over GDT. These results demonstrate that the video representation learned by our method are better suited to fine-grained actions than existing self-supervised methods. We additionally report results on Diving48 [41] in the supplementary.

Task Shift. For the task shift to repetition counting, our method is on par with AVID-CMA [51] and RSPNet, but worse than GDT. For multi-label action recognition on Charades, our approach is 3rd, comparable to VideoMoCo but worse than TCLR. This suggests the representations learned by our method are somewhat transferable to tasks beyond single-label action recognition. However, the remaining gap between supervised and self-supervised highlights the need for future work to explore task generalizability further.

Comparison with Transformers. Table 8 also contains recent transformer-based self-supervised works SVT [61] and VideoMAE [71]. We observe that both SVT and VideoMAE have good performance with large amounts of finetuning data (SSv2), in-domain fine-tuning (UCF(10^3)), and multi-label action recognition (Charades). However, they considerably lag in performance for motion-focused setups Gym99, FX-S1, UB-S1, and repetition counting compared to our tubelet contrast with a small CNN backbone.

Overall SEVERE Performance. Finally, we compare the mean and the average rank across all generalizability factors. Our method has the best mean performance (66.5) and achieves the best average rank (4.1). When pretraining with the 3x smaller Mini-Kinetics our approach still achieves impressive results. We conclude our method improves the generalizability of video self-supervised representations across these four downstream factors while being data-efficient.

5. Conclusion

This paper presents a contrastive learning method to learn motion-focused video representations in a self-supervised manner. Our model adds synthetic tubelets to videos so that the only similarities between positive pairs are the spatiotemporal dynamics of the tubelets. By altering the motions of these tubelets and applying transformations we can simulate motions not present in the pretraining data. Experiments show that our proposed method is data-efficient and more generalizable to new domains and fine-grained actions than prior self-supervised methods.

Acknowledgements. This work is part of the research programme Perspectief EDL with project number P16-25 project 3, which is financed by the Dutch Research Council (NWO) domain Applied and Engineering/Sciences (TTW). We thank Piyush Bagad for help with experiments and Artem Moskalev for useful discussions.

References

[1] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[2] Yuki M. Asano, Mandela Patrick, Christian Rupprecht, and Andrea Vedaldi. Labelling unlabelled videos from scratch with multi-modal self-supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2, 8

[3] Kyungjune Baek, Minhyun Lee, and Hyunjung Shim. Psynet: Self-supervised approach to object localization using point symmetric transformation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 6

[4] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[5] Shuo Chen, Zenglin Shi, Pascal Mettes, and Cees GM Snoek. Social fabric: Tubelet compositions for video relation detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[6] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning (ICML)*, 2020. 2

[7] Hyeon Cho, Taehoon Kim, Hyung Jin Chang, and Wonjun Hwang. Self-supervised visual learning by variable playback speeds prediction of a video. *IEEE Access*, 9:79562–79571, 2021. 2

[8] Ishan Dave, Rohit Gupta, Mamshad Nayeem Rizve, and Mubarak Shah. Tclr: Temporal contrastive learning for video representation. *Computer Vision and Image Understanding (CVIU)*, 219:103406, 2022. 2, 4, 5, 7, 8

[9] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. Motion-aware contrastive video representation learning via foreground-background merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 5, 7

[10] Hazel Doughty, Ivan Laptev, Walterio Mayol-Cuevas, and Dima Damen. Action modifiers: Learning from adverbs in instructional videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[11] Hazel Doughty and Cees G M Snoek. How do you do it? fine-grained action understanding with pseudo-adverbs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[12] Haodong Duan, Yue Zhao, Kai Chen, Dahua Lin, and Bo Dai. Revisiting skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[13] Christoph Feichtenhofer, Haoqi Fan, Yanghao Li, and Kaiming He. Masked autoencoders as spatiotemporal learners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[15] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[16] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[17] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[18] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[19] Kirill Gavrilyuk, Mihir Jain, Ilia Karmanov, and Cees G M Snoek. Motion-augmented self-training for video recognition at smaller scale. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 7

[20] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2, 5

[21] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[22] Tengda Han, Weidi Xie, and Andrew Zisserman. Self-supervised co-training for video representation learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[23] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5, 8

[24] James Hong, Matthew Fisher, Michaël Gharbi, and Kayvon Fatahalian. Video pose distillation for few-shot, fine-grained sports action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[25] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[26] Huazhang Hu, Sixun Dong, Yiqun Zhao, Dongze Lian, Zhengxin Li, and Shenghua Gao. Transrac: Encoding multi-scale temporal correlation with transformers for repetitive action counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[27] Deng Huang, Wenhao Wu, Weiwen Hu, Xu Liu, Dongliang He, Zhihua Wu, Xiangmiao Wu, Mingkui Tan, and Errui Ding. Ascnet: Self-supervised video representation learning with appearance-speed consistency. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 4

[28] Ziyuan Huang, Shiwei Zhang, Jianwen Jiang, Mingqian Tang, Rong Jin, and Marcelo H Ang. Self-supervised motion learning from static images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[29] Mihir Jain, Jan Van Gemert, Hervé Jégou, Patrick Bouthemy, and Cees G M Snoek. Action localization with tubelets from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1, 2

[30] Simon Jenni and Hailin Jin. Time-equivariant contrastive video representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 7

[31] Simon Jenni, Givi Meishvili, and Paolo Favaro. Video representation learning by recognizing temporal transformations. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[32] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 2

[33] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[34] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, et al. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *IEEE Transactions on Circuits and Systems for Video Technology*, 28(10):2896–2907, 2017. 2

[35] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[36] Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. Relational self-attention: What's missing in attention for video understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. 2

[37] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011. 2, 5

[38] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for video action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2

[39] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 2

[40] Tianjiao Li, Lin Geng Foo, Qiuhong Ke, Hossein Rahmani, Anran Wang, Jinghua Wang, and Jun Liu. Dynamic spatio-temporal specialization learning for fine-grained action recognition. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[41] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *European Conference on Computer Vision (ECCV)*, 2018. 8

[42] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *European Conference on Computer Vision (ECCV)*, 2020. 1, 2

[43] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[44] Yuanze Lin, Xun Guo, and Yan Lu. Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 7

[45] Zhang Lin, She Qi, Shen Zhengyang, and Wang Changhu. Inter-intra variant dual representations for self-supervised video recognition. In *British Machine Vision Conference (BMVC)*, 2021. 2, 7

[46] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017. 5

[47] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 2

[48] Khoi-Nguyen C Mac, Dhiraj Joshi, Raymond A Yeh, Jinjun Xiong, Rogerio S Feris, and Minh N Do. Learning motion in feature space: Locally-consistent deformable convolution networks for fine-grained action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 2

[49] Effrosyni Mavroudi, Divya Bhaskara, Shahin Sefati, Haider Ali, and René Vidal. End-to-end fine-grained action segmentation and recognition using conditional random field models and discriminative sparse coding. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018. 2

[50] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision (ECCV)*, 2016. 2

[51] Pedro Morgado, Nuno Vasconcelos, and Ishan Misra. Audio-visual instance discrimination with cross-modal agreement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 7, 8

[52] Artem Moskalev, Ivan Sosnovik, Fischer Volker, and Arnold Smeulders. Contrasting quadratic assignments for set-based representation learning. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[53] Bingbing Ni, Vignesh R Paramathayalan, and Pierre Moulin. Multiple granularity analysis for fine-grained action detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[54] Jingcheng Ni, Nan Zhou, Jie Qin, Qian Wu, Junqi Liu, Boxun Li, and Di Huang. Motion sensitive contrastive learning for self-supervised video representation. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[55] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 1, 2, 3

[56] Tian Pan, Yibing Song, Tianyu Yang, Wenhao Jiang, and Wei Liu. Videomoco: Contrastive video representation learning with temporally adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2, 4, 5, 7, 8

[57] Mandela Patrick, Yuki M. Asano, Polina Kuznetsova, Ruth Fong, João F. Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multi-modal self-supervision from generalized data transformations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. 2, 4, 5, 7, 8

[58] Chen Peihao, Huang Deng, He Dongliang, Long Xiang, Zeng Runhao, Wen Shilei, Tan Mingkui, and Gan Chuang. Rspnet: Relative speed perception for unsupervised video representation learning. In *The AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 1, 2, 4, 5, 7, 8

[59] AJ Piergiovanni and Michael S Ryoo. Fine-grained activity recognition in baseball videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018. 2

[60] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1, 2

[61] Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael S Ryoo. Self-supervised video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 8

[62] Madeline C Schiappa, Yogesh S Rawat, and Mubarak Shah. Self-supervised learning for videos: A survey. *ACM Computing Surveys*, 2022. 1, 2

[63] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5

[64] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Intra-and inter-action understanding via temporal action parsing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[65] Gunnar A. Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *European Conference on Computer Vision (ECCV)*, 2016. 5

[66] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014. 2

[67] 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, 5

[68] Baoli Sun, Xinchen Ye, Tiantian Yan, Zhihui Wang, Haojie Li, and Zhiyong Wang. Fine-grained action recognition with robust motion representation decoupling and concentration. In *Proceedings of the ACM International Conference on Multimedia (ACMMM)*, 2022. 2

[69] Li Tao, Xueting Wang, and Toshihiko Yamasaki. Pretext-contrastive learning: Toward good practices in self-supervised video representation leaning. *arXiv preprint arXiv:2010.15464*, 2021. 2, 8

[70] Fida Mohammad Thoker, Hazel Doughty, Piyush Bagad, and Cees G M Snoek. How severe is benchmark-sensitivity in video self-supervised learning? In *European Conference on Computer Vision (ECCV)*, 2022. 1, 2, 5, 7, 8

[71] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2, 8

[72] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 5, 8

[73] Jan van Gemert, Mihir Jain, Ella Gati, and Cees G M Snoek. APT: Action localization proposals from dense trajectories. In *British Machine Vision Conference (BMVC)*, 2015. 2

[74] Guangting Wang, Yizhou Zhou, Chong Luo, Wenxuan Xie, Wenjun Zeng, and Zhiwei Xiong. Unsupervised visual representation learning by tracking patches in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 4, 7, 8

[75] Jinpeng Wang, Yuting Gao, Ke Li, Yiqi Lin, Andy J Ma, Hao Cheng, Pai Peng, Feiyue Huang, Rongrong Ji, and Xing Sun. Removing the background by adding the background: Towards background robust self-supervised video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[76] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *European Conference on Computer Vision (ECCV)*, 2020. 5

[77] Fanyi Xiao, Joseph Tighe, and Davide Modolo. Maclr: Motion-aware contrastive learning of representations for videos. In *European Conference on Computer Vision (ECCV)*, 2022. 2

[78] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *European Conference on Computer Vision (ECCV)*, 2018. 5

[79] Dejing Xu, Jun Xiao, Zhou Zhao, Jian Shao, Di Xie, and Yueting Zhuang. Self-supervised spatiotemporal learning via video clip order prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[80] Ceyuan Yang, Yinghao Xu, Jianping Shi, Bo Dai, and Bolei Zhou. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[81] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[82] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[83] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[84] Huaidong Zhang, Xuemiao Xu, Guoqiang Han, and Shengfeng He. Context-aware and scale-insensitive temporal repetition counting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 5

[85] Yunhua Zhang, Ling Shao, and Cees G M Snoek. Repetitive activity counting by sight and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[86] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, Ivan Marsic, Cees G M Snoek, and Joseph Tighe. Tuber: Tubelet transformer for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2