MoDist: Motion Distillation for Self-supervised Video Representation Learning

Fanyi Xiao Joseph Tighe Davide Modolo

Amazon AI

{xfanyi, tighej, dmodolo}@amazon.com

Abstract

We present MoDist as a novel method to explicitly distill motion information into self-supervised video representations. Compared to previous video representation learning methods that mostly focus on learning motion cues implicitly from RGB inputs, we show that the representation learned with our MoDist method focus more on foreground motion regions and thus generalizes better to downstream tasks. To achieve this, MoDist enriches standard contrastive learning objectives for RGB video clips with a cross-modal learning objective between a Motion pathway and a Visual pathway. We evaluate MoDist on several datasets for both action recognition (UCF101/HMDB51/SSv2) as well as action detection (AVA), and demonstrate state-of-the-art self-supervised performance on all datasets. Furthermore, we show that MoDist representation can be as effective as (in some cases even better than) representations learned with full supervision. Given its simplicity, we hope MoDist could serve as a strong baseline for future research in self-supervised video representation learning.

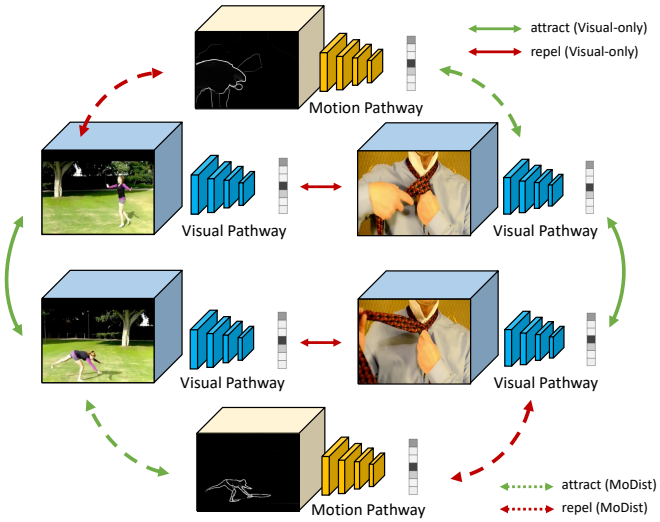

Figure 1: **Motion Distillation**. We propose Motion Distillation (MoDist) as an explicit method to learn motion-aware video representations without using any label.

1. Introduction

Supervised learning has enjoyed great successes in many computer vision tasks in the past decade. One of the most important fuel in this successful journey is the availability of large amount of high-quality labeled data. Notably, the ImageNet [17] dataset for image classification, was where it all started for the deep learning revolution in vision. In the video domain, the Kinetics dataset [40] has long been regarded as the "ImageNet for videos" and has enabled the "pretrain-then-finetune" paradigm for many video tasks. Interestingly, though years old, ImageNet and Kinetics are still the to-go datasets for pretraining, at least among those that are publicly available. This shows how much effort is needed to create these large-scale labeled datasets.

To mitigate the reliance on large-scale labeled datasets, *self-supervised learning* came with the promise to learn useful representations from large amount of *unlabeled* data. Following the recent success in NLP (e.g., BERT, GPT-3 [18, 7]), some research has attempted to find its counterpart in vision. Among them, pioneering research has been conducted in the image domain to produce successful methods like MoCo [35] and SimCLR [12]. Compared to images, large-scale video datasets induce even higher annotation costs, making it even more important to develop effective self-supervised methods to learn generalizable representations for videos. Some recent videos works attempted to learn such representations by training their models to solve pretext tasks, like predicting the correct temporal order clips [51, 26, 6, 77, 9], predict future frames [19] and predict whether a video is played at its intrinsic speed [4]. Though successful to a certain extent, these methods do not explicitly make use of motion information derived from the temporal sequence, which has been shown to be important for supervised action recognition tasks [63, 24, 73].

In this paper, we propose MoDist (Motion Distillation) as a novel self-supervised video representation learning

method, to explicitly train networks to model motion cues. Specifically, MoDist consists of two pathways: the main Visual pathway that is later used for downstream tasks, and a supporting Motion pathway that is only used during training (Fig. 1). We connect these two pathways and set-up a cross-modal contrastive learning objective to have the Motion pathway guide its Visual counterpart to focus on foreground motion regions for better motion modeling.

To evaluate MoDist, we perform self-supervised pretraining on Kinetics-400 and transfer its representation to 4 video datasets for both action recognition (UCF101 [65], HMDB51 [44], Something-Something [1]) and action detection (AVA [29]). Without bells and whistle, MoDist outperforms all previous video self-supervised methods on all datasets, under all evaluation settings. For example, MoDist improves top-1 accuracy by 17% and 16.9% on UCF101 and HMD51, over previous SOTA trained on Kinetics-400. Furthermore, on Something-Something and AVA, MoDist even outperforms its fully-supervised counterpart, demonstrating the strength of our approach. Finally, we ablate the components of MoDist both quantitatively and qualitatively.

2. Related Work

Self-supervised image representation learning. The goal of self-supervised image representation learning is to learn useful representations from large collection of unlabeled images. Early work focused on designing different pretext tasks in the intent of inducing generalizable semantic representations [20, 52, 53, 84]. Though producing promising results, these methods could not match the performance of fully-supervised trained representations [42], as it is hard to prevent the network from utilizing shortcuts to solve pretext tasks (e.g., "chromatic aberration" in context prediction [20]). This changed when researchers re-visited the decade-old technique of contrastive learning [30, 79]. Some of these recent work started to successfully produce results that were comparable to those of supervised learning on images [35, 12, 13, 50, 28, 14, 10]. Though related, these work were designed to learn from static images and thus cannot utilize the rich temporal information contained in videos.

Self-supervised video representation learning. Videos present unique opportunities to extract self-supervision by exploiting its temporal structure. In addition to the popular pretext task based works introduced in Sec. 1 [51, 26, 77, 19, 4], others attempted to learn video representations by tracking either patches [75], pixels [76] or colors [70] across neighboring frames. Closer to our work is the recent arXiv paper CVRL [59], which extends the image-based SimCLR into the video domain, achieving impressive results. Though successful to certain extent, none of above methods explicitly make use of motion information derived from the video temporal sequence, which we demonstrate in this work to greatly enhance the generalization ability of the learned representations. One line of work attempts to fix this by learning useful representations through exploiting the correspondences between RGB and motion pixels [48, 60, 37]. Our method is different in that we utilize visual-motion correspondence at a higher level than pixels, which we will show to be beneficial. Whereas in [72], the authors propose to regress pseudo labels generated from motion statistics as the pretext task. In contrast, we adopt the contrastive instance discrimination framework which is more generalizable. The recent work of CoCLR [33] also utilizes optical flow for representation learning. However, it only uses optical flow as a media to mine RGB images with similar motion cues, whereas we propose to directly distill motion information into visual representations and thus yield an much simpler and more robust method. Finally, there is also a line of work that utilizes audio-video synchronization for self-supervised learning [55, 43, 2, 58, 56]. Although we do not explore the use of audio in this paper, it can be easily incorporated into our approach.

Motion in video tasks. Motion information has been heavily studied for many video tasks. As a prominent motion representation, optical flow has been utilized in many video action classification methods, either in the form of classical hand-crafted spatiotemporal features [45, 16, 71], or serve as input to deep CNN systems trained with supervised learning [23, 24, 73]. In contrast, our method focuses on exploiting motion information in the context of self-supervised learning. Beyond video classification, motion has also been exploited in many other tasks like video object detection [85, 39, 25, 80], video frame prediction [61, 47], video segmentation [68, 3, 15], object tracking [36, 5, 57], and 3D reconstruction [69].

3. MoDist

We design MoDist as a two-branch network consisting of a Visual pathway and a Motion pathway (Fig. 1). The Visual pathway takes as input visual[1] clips and produces their visual embeddings. Similarly, the Motion pathway operates on motion clips (we will study different motion inputs in Sec. 3.2) and generates motion embeddings. MoDist is trained using three contrastive learning objectives (Sec. 3.1: (i) a visual-only loss that pulls together visual clip embeddings that are sampled from the same video (solid green arrow in Fig. 1) and pushes away that of different videos (solid red arrow); (ii) a motion-only loss that operates like (i), but on motion clips (omitted in Fig. 1 to avoid clutter) and (iii) a motion-to-visual loss to explicitly distill motion knowledge from the Motion pathway into the Visual path-

[1] We use "visual" and "RGB" interchangeably in this paper.

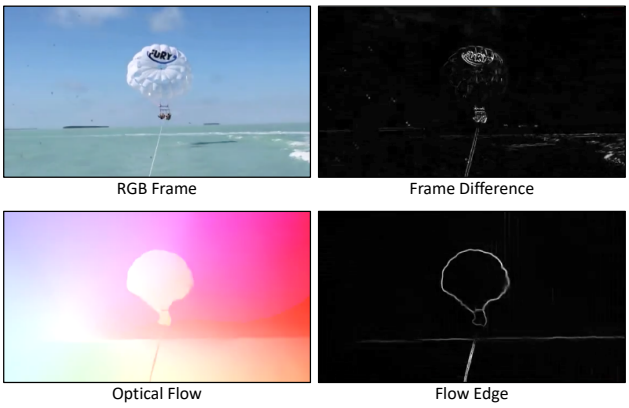

Figure 2: **Motion inputs**. Given the RGB input (top-left), we compare three options for motion inputs. Best viewed on screen.

way (dashed arrows). As shown in Fig. 1, we generate positive pairs from clips extracted from the same video (green arrows) and negative pairs from clips extracted from different videos (red arrows). After pretraining with MoDist, we then remove the Motion pathway and transfer the Visual pathway to target datasets for task-specific finetuning.

3.1. Training MoDist

Visual-only learning. We model this using a contrastive learning objective, similar to previous works on self-supervised learning for images [12, 35, 28]. However, different from these methods which take as input random spatial crops from an image, our model takes as input random clips with spatiotemporal jitterring. Specifically, as shown in Fig. 1, given a random clip we produce its embedding v^q (query), and sample a second positive clip from the same video and produce its embedding v^k (key), as well as N negative embeddings v_i^n, $i \in \{1, ..., N\}$ from other videos. Then, we train the Visual pathway with the InfoNCE objective $\mathcal{L}_v = \text{IN}(v^q, v^k, v^n)$ [54, 35]:

$$\mathcal{L}_v = -\log \frac{\exp(v^q \cdot v^k/\tau)}{\exp(v^q \cdot v^k/\tau) + \sum_{i=1}^{N} \exp(v^q \cdot v_i^n/\tau)}, \quad (1)$$

where τ is a temperature parameter. This objective ensures that our Visual pathway pulls together embeddings v^q and v^k, while pushing away those of all the negative clips v_i^n.

Motion-only learning. To improve the discriminativeness of the Motion pathway, we add another InfoNCE objective $\mathcal{L}_m = \text{IN}(m^q, m^k, m^n)$, which is trained in a similar way to $\mathcal{L}_v$ but this time on motion embeddings m^q, m^k (both are sampled from the same video as v_q) and m^n (which denotes a set of negative motion embeddings). This ensures that the Motion pathway is able to embed similar motion patterns close to each other.

Motion distillation: motion-to-visual learning. We model this also with a contrastive learning objective, but with a different purpose compared to the previous two. Here, we aim at distilling motion information from the Motion pathway into the Visual pathway. Specifically, we train the model using the following InfoNCE objectives:

$$\mathcal{L}_{mv} = \text{IN}(v^q, m^k, v^n) + \text{IN}(m^q, v^k, m^n). \quad (2)$$

Note that v^q is *not necessarily in temporal synchronization* with m^k, but rather just a motion clip sampled from the same video (same for v^k and m^q). In our ablation, we show that allowing for this misalignment encourages the embedding to better learn semantic abstraction of visual and motion patterns, which leads to better performance.

One key difference to visual-only contrastive learning is on how we sample motion clips for both motion-only and motion-to-visual learning. Instead of sampling randomly, we constrain to only sample in temporal regions with strong motion cues. Specifically, we compute the sum of pixels P_i on the motion input and only sample frames with $\sum_{i=1}^{K} P_i/K > \gamma$, where K is the total number of pixels in a frame and γ is the threshold. This process helps avoid sampling irrelevant regions with no motion and thus leads to better representations.

Final training objective. The final training objective for MoDist is the sum of all aforementioned loss functions:

$$\mathcal{L} = \mathcal{L}_v + \mathcal{L}_m + \mathcal{L}_{mv}. \quad (3)$$

Training MoDist end-to-end is non-trivial, as video representation are expensive to compute and to maintain (as contrastive learning requires large batch sizes [12]). Inspired by [79, 35], we avoid this problem by adopting the idea of memory bank for negative samples. Specifically, we construct two memory banks of negative samples for visual and motion inputs, and maintain a momentum version of the Motion and Visual pathways updated as a moving average of their online counterparts with momentum coefficient λ: $\theta' \leftarrow \lambda\theta' + (1-\lambda)\theta$, where θ and θ' are weights for the online and momentum version of the model respectively. One caveat is that when pushing negatives into the pool, we push the video index, along with the embedding, so that we can avoid sampling visual or motion clips that are from the same video as positive clips, which would otherwise confuse the network and hurt the representations. Similar to [34], we forward queries through the online model and keys through the momentum model to produce embeddings.

3.2. Designing motion inputs

There are many ways to represent a motion input. A straightforward way is to directly compute the difference of

stage	Visual pathway	Motion pathway
input clip	$3\times8\times224^2$ (stride 8)	$3\times16\times224^2$ (stride 4)
conv_1	5×7^2, 64 stride 2, 2^2	1×7^2, 8 stride 2, 2^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2
res_2	$\begin{bmatrix} 1\times1^2, 64 \\ 1\times3^2, 64 \\ 1\times1^2, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1^2, 8 \\ 1\times3^2, 8 \\ 1\times1^2, 32 \end{bmatrix}\times3$
res_3	$\begin{bmatrix} 1\times1^2, 128 \\ 1\times3^2, 128 \\ 1\times1^2, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1^2, 16 \\ 1\times3^2, 16 \\ 1\times1^2, 64 \end{bmatrix}\times4$
res_4	$\begin{bmatrix} 3\times1^2, 256 \\ 1\times3^2, 256 \\ 1\times1^2, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1^2, 32 \\ 1\times3^2, 32 \\ 1\times1^2, 128 \end{bmatrix}\times6$
res_5	$\begin{bmatrix} 3\times1^2, 512 \\ 1\times3^2, 512 \\ 1\times1^2, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1^2, 64 \\ 1\times3^2, 64 \\ 1\times1^2, 256 \end{bmatrix}\times3$
global average pool, projection		

Table 1: **MoDist network architecture** (ResNet-50). The dimensions of kernels are denoted by $\{T\times S^2, C\}$ for temporal, spatial, and channel sizes, while strides are denoted with {temporal stride, spatial stride^2}. Note that Motion pathway is $8\times$ more lightweight (in channel sizes) compared to the Visual pathway.

pixel values between two consecutive frames. While capturing motion to a certain extent, it also captures undesired signals like pixel value shifts caused by background motion (e.g., sea-wave in Fig. 2 top-right). A more appropriate representation might be optical flow [8, 78, 21, 67]. However, a disadvantage of feeding in raw optical flow (or flow vector magnitude, as used in [27]) is that it is heavily influenced by factors like illumination change (Fig. 2 bottom-left) and it also captures absolute flow magnitude, which is not very useful for learning general motion patterns. To overcome these limitations, in MoDist, we propose to use flow edge maps as inputs to the Motion pathway network. Specifically, we apply a Sobel filter [64] onto the flow magnitude map to produce the flow edges (Fig. 2 bottom-right). In our experiments, this simple operation turns out to produce significantly better motion representations that focus on foreground motion regions.

3.3. Visual and Motion pathway architectures

The MoDist architecture is presented in Table 1. Our Visual pathway is a 3D ResNet50 (R3D-50) with a structure similar to that of "Slow-only" in [22, 59], which features 2D convs in res_2, res_3 and non-degenerate 3D convs in res_4, res_5. Our Visual pathway takes as input a tensor of size $3\times8\times224^2$, capturing 8 frames of size 224×224. The sampling stride is 8, which means that the visual input clip spans 8×8 frames, corresponding to ~2 seconds for videos at 30 FPS. To have larger temporal receptive field, we set the temporal kernel size of $\texttt{conv}_1$ to 5 following [59].

Our Motion pathway is a 2D ResNet50. and it takes as input a tensor of size $3\times16\times224^2$, stacking 16 motion frames. We use a sampling stride of 4, so that it spans for the same time as the visual input (i.e., ~2 secs). Following the design philosophy of SlowFast Networks [22], we design our Motion pathway to be much more lightweight compared to our Visual pathway (1/8 channel sizes across the network), as motion inputs have intrinsically less variability (i.e., no variations on colors, illumination, etc.). Finally, we note that our motion distillation idea is general and can be applied to other architectures as well.

4. Experiments

4.1. Implementation Details

MoDist training details. We train MoDist on the Kinetics-400 (K400) dataset [40]. The dataset consists of ~240k video clips that span at most 10 seconds. These were originally annotated with 400 different action classes, but we *do not* use any of these labels. We train MoDist for 600 epochs on the whole 240k videos when we compare against the literature. For our ablation study, instead, we compare different variants of MoDist trained for 100 epochs on a subset of 60k videos ("K400-mini"). We use a pool size (N in Eq. 1) of 65536 negative samples for both visual and motion inputs. We set the momentum update coefficient $\lambda = 0.999$ and temperature τ to 0.1. The embedding dimension is set to 128 for both Visual and Motion pathways. For the visual inputs, we apply random spatial cropping, temporal jittering, p = 0.2 probability grayscale conversion, p = 0.5 horizontal flip, p = 0.5 Gaussian blur, and p = 0.8 color perturbation on brightness, contrast and saturation, all with 0.4 jittering ratio. For motion inputs, we randomly sample flow edge clips in high motion regions (with motion threshold γ set to 0.02) and skip other augmentations.

Flow Edge Maps. To compute flow edge map for frame t, we first compute optical flow from frame t to $t-5$, using `RAFT-things` model trained entirely on synthetic data without human annotations [67]. Then, we apply a Sobel filter onto the magnitude map of optical flow and clamp the resulting edge map in [0, 10] as the final flow edge map. We note that this is an offline pre-processing that only needs to be done once and reused throughout training.

Baselines. We compare against two baselines: (i) *Self-Supervised RGB-only* is a strong self-supervised representation trained from RGB inputs using only the contrastive learning objective of Eq. 1 (i.e., without our motion learning objectives $\mathcal{L}_{mv}$ and $\mathcal{L}_m$); and (ii) *Supervised* is a fully supervised model trained for action classification on K400. Both baselines use a R3D-50 backbone.

Method	Date	Dataset (duration)	Arch.	Size	Modality	Frozen	UCF	HMDB
CBT [66]	2019	K600+ (273d)	S3D	112^2	V	✓	54.0	29.5
MemDPC [32]	2020	K400 (28d)	R-2D3D-34	224^2	V	✓	54.1	30.5
MIL-NCE [49]	2020	HTM (15y)	S3D	224^2	V+T	✓	82.7	53.1
MIL-NCE [49]	2020	HTM (15y)	I3D	224^2	V+T	✓	83.4	54.8
XDC [2]	2020	IG65M (21y)	R(2+1)D	224^2	V+A	✓	85.3	56.0
ELO [58]	2020	Youtube8M (8y)	R(2+1)D	224^2	V+A	✓	–	64.5
AVSlowFast [81]	2020	K400 (28d)	AVSlowFast-50	224^2	V+A	✓	77.4	42.2
CoCLR [33]	2020	K400 (28d)	S3D	128^2	V	✓	74.5	46.1
CVRL [59]	2021	K400 (28d)	R3D-50	224^2	V	✓	89.8	58.3
MoDist		K400 (28d)	R3D-18	128^2	V	✓	90.4	57.5
MoDist		K400 (28d)	R3D-50	224^2	V	✓	**91.5**	**63.0**
w/o Pretrain		-	R3D-50	224^2	V	✗	69.0	22.7
OPN [46]	2017	UCF (1d)	VGG	227^2	V	✗	59.6	23.8
3D-RotNet [38]	2018	K400 (28d)	R3D-18	112^2	V	✗	62.9	33.7
ST-Puzzle [41]	2019	K400 (28d)	R3D-18	224^2	V	✗	63.9	33.7
VCOP [83]	2019	UCF (1d)	R(2+1)D	112^2	V	✗	72.4	30.9
DPC [31]	2019	K400 (28d)	R-2D3D-34	128^2	V	✗	75.7	35.7
CBT [66]	2019	K600+ (273d)	S3D	112^2	V	✗	79.5	44.6
DynamoNet [19]	2019	Youtube8M-1 (58d)	STCNet	112^2	V	✗	88.1	59.9
AVTS [43]	2018	K400 (28d)	I3D	224^2	V+A	✗	83.7	53.0
AVTS [43]	2018	AudioSet (240d)	MC3	224^2	V+A	✗	89.0	61.6
XDC [2]	2019	K400 (28d)	R(2+1)D	224^2	V+A	✗	84.2	47.1
XDC [2]	2019	IG65M (21y)	R(2+1)D	224^2	V+A	✗	94.2	67.4
AVSlowFast [81]	2020	K400 (28d)	AVSlowFast-50	224^2	V+A	✗	87.0	54.6
CVRL [59]	2020	K400 (28d)	R3D-50	224^2	V	✗	92.9	67.9
SpeedNet [4]	2020	K400 (28d)	S3D-G	224^2	V	✗	81.1	48.8
MemDPC [32]	2020	K400 (28d)	R-2D3D-34	224^2	V	✗	86.1	54.5
CoCLR [33]	2020	K400 (28d)	S3D	128^2	V	✗	87.9	54.6
GDT [56]	2020	K400 (28d)	R(2+1)D	112^2	V+A	✗	89.3	60.0
GDT [56]	2020	IG65M (21y)	R(2+1)D	112^2	V+A	✗	95.2	72.8
MIL-NCE [49]	2020	HTM (15y)	S3D	224^2	V+T	✗	91.3	61.0
ELO [58]	2020	Youtube8M-2 (13y)	R(2+1)D	224^2	V+A	✗	93.8	67.4
MoDist		K400 (28d)	R3D-18	128^2	V	✗	91.3	62.1
MoDist		K400 (28d)	R3D-50	224^2	V	✗	**94.0**	**67.4**
Supervised [82]		K400 (28d)	S3D	224^2	V	✗	96.8	75.9

Table 2: **Comparison with state-of-the-art approaches.** We report top-1 accuracy in this table. In parenthesis, we show the total video duration in time (**d** for day, **y** for year). Top half of the table contains results for Linear protocol (Frozen ✓), whereas the bottom half shows results for the Full end-to-end finetuning protocol (Frozen ✗). For modality, V refers to visual only, A is audio, T is text narration.

method	data	UCF	HMDB
RGB-only	K400-mini	63.6	33.7
MoDist	K400-mini	**78.1**	**47.2**
RGB-only	K400	74.6	46.3
MoDist	K400	**85.5**	**57.7**

(a) **Motion distillation**.

input	UCF	HMDB
RGB-only	63.6	33.7
Frame Diff	71.6	40.1
Optical Flow	74.1	44.2
Flow Edges	**78.1**	**47.2**

(b) **Motion inputs**.

components	UCF	HMDB
MoDist	**78.1**	**47.2**
− temp. jitter	77.4	47.1
− motion thresh	77.3	46.4
− $\mathcal{L}_m$	77.8	46.6

(c) **MoDist components**.

epoch	UCF	HMDB
100	85.5	57.7
200	87.3	59.0
400	88.6	61.8
600	**89.9**	**62.1**

(d) **Training epochs**.

Table 3: **Ablating MoDist**. We present top-1 classification accuracy using the Linear Layer Training evaluation protocol (sec. 4.2). Experiments in (b) and (c) are conducted on K400-mini, whereas (d) is on the full K400 dataset. We use 8×8 R3D-50 model for finetuning.

4.2. Action Recognition on UCF101 and HMDB51

Datasets and evaluation protocol. We first evaluate MoDist for action recognition on the two most popular datasets in the literature: UCF101 [65] and HMDB51 [44]. We follow the standard settings to perform self-supervised training on K400 and then transfer the learned weights to target datasets for evaluation. Two evaluation protocols are often employed in the literature to evaluate the quality of the self-supervised representation: (i) *Linear Layer Training* freezes the trained backbone and simply trains a linear classifier on the target dataset, while (ii) *Full Network Training* finetunes the entire network end-to-end on the target dataset. For completeness, we evaluate using both protocols and report action classification top-1 accuracy. For all experiments on UCF101 and HMDB51, we report results using `split1` for train/test split. In total, there are 9.5k/3.7k train/test videos with 101 action classes in UCF101, and 3.5k/1.5k train/test videos with 51 actions in HMDB51. We

use 32×8 inputs during finetuning and standard 10 (temporal) ×3 (spatial) crop testing [74, 22]. Finally, we also use these two datasets to conduct extensive ablation studies to investigate various design choices of MoDist using the Linear Layer Training evaluation protocol.

Ablation: motion distillation (Table 3a). First and foremost, we study the importance of enriching visual embeddings with motion cues using the proposed distillation method. Results show that MoDist improves substantially over the baseline on both datasets: +15 points when trained on K400-mini, and +11 when trained on K400. Notably, MoDist trained on K400-mini already outperforms the RGB baseline trained with $4\times$ more data (K400): +3.5/+0.9 on UCF/HMDB. This truly shows the importance of motion modelling to train strong and generalizable representations.

Ablation: motion representations (Table 3b). Despite the conceptual advantages of flow edge maps discussed in Sec. 3, we benchmark it against other motion representations presented in that same section: Frame Difference and Optical Flow. As shown in Table 3b, Flow Edges is indeed the best way to represent motion for self-supervised training, thanks to its ability to prune background motion noise and absolute motion magnitude. That being said, even the much weaker Frame Difference representation outperforms the RGB-only baseline by +8.0 top-1 accuracy on UCF and +6.4 on HMDB. This further confirms the importance of enriching video representations with motion cues.

Ablation: MoDist components (Table 3c). We now dissect MoDist to study the importance of its components.

Temporal Jittering. Unlike previous work that learn self-supervised representation by exploiting pixel-level correspondences between RGB and optical flow inputs [48, 60], we demonstrate that it's more effective to learn self-supervised representations by introducing temporal "misalignment" between them. Specifically, we compare MoDist, which trains on RGB and motion clips that are temporally jittered, against a variant that is trained on synchronized RGB and motion clips (i.e., sync pairs $[v^q, m^k]$ and $[m^q, v^k]$ in Eq. 2). Our results show that the misaligned inputs lead to better representations (+0.7 on UCF), as it prevents the model from exploiting the shortcut of finding pixel correspondences using low-level visual cues.

Motion thresholding. Another component we study is the motion input sampling strategy discussed in Sec. 3.1. We compare MoDist to a variant which randomly samples motion input clips, without removing those with little motion (i.e., setting threshold $\gamma = 0$, Sec. 4.1). Without this threshold, top-1 accuracy degrades by -0.8 on both datasets, due to the noise introduced by clips with too little motion.

Motion loss $\mathcal{L}_m$. Finally, we study whether it's necessary to have the extra contrastive objective $\mathcal{L}_m$ between motion inputs (Eq. 3), which is included to help training more discriminative motion embeddings. Results show that this motion discrimination objective is indeed useful as it improves top-1 acc by +0.3 and +0.6 on UCF101 and HMDB51.

Ablation: training epochs (Table 3d). Next, we vary the number of training epochs and study its impact on representation quality. It is clear from the results that the representation learned by MoDist benefits from a longer training schedule. This is in line to what is previously observed in the self-supervised learning literature [12, 35, 59].

Comparison to state-of-the-arts (Table 2). We now compare MoDist against previous self-supervised video representation learning methods in the literature using both evaluation protocols introduced at the beginning of Sec. 4.2: Linear (✓for column "Frozen") and Full (✗).

By only training a linear layer on top of our self-supervised learned representation, our method is able to achieve significantly better top-1 classification accuracy compared to the previous state-of-the-art trained on K400: +16.7 and +16.9 over CoCLR on UCF101 and HMDB51, respectively. Only the recent arXiv paper CVRL comes close to our results on UCF, but still lacks on HMDB (-4.7). Moreover, MoDist outperforms all previous methods, including those trained on 100×more data than K400 (e.g., IG65M and Youtube8M), and those that use extra modalities like audio and text (e.g., XDC, MIL-NCE).

Next, we compare against methods that adopt the end-to-end full training evaluation protocol. Similar to our observation with the linear evaluation protocol, MoDist again achieves SOTA results among the methods trained on K400. Among all methods, only XDC and GDT produce results comparable to MoDist, but are either trained on 270×more data (IG65M contains 21 years of video content *vs*. K400 only 28 days) or use extra audio modality as inputs.

Finally, towards making the best effort in enabling fair comparison against the literature, we also present the results of a weaker MoDist model trained with a smaller backbone (R18) and a smaller input size (128×128). Under this setting, our model still convincingly outperforms models with similar backbone and input resolutions (e.g., 3D-RotNet, CBT, GDT, CoCLR) using both evaluation protocols and even outperforms many methods that use larger backbones (e.g. XDC, AVSlowFast, etc.).

4.3. Action Recognition on Something-Something

To further demonstrate the effects of explicit motion distillation, we evaluate MoDist on Something-Something-v2 (SSv2) [1], which is a challenging action classification dataset heavily focused on motion. Different from

UCF101 and HMDB51 which contain action classes similar to K400, SSv2 contains a very different set of actions featuring complex human object interactions, like "Moving something up" and "Pushing something from left to right". The dataset consists of 168k training, 24k validation and 24k test videos, all annotated with 174 action classes. We finetune on SSv2 with a recipe that mostly follows the official implementation of [22]. We use a clip size of 16×8 and a batchsize of 16 (over 8 GPUs). We train for 22 epochs with an initial learning rate of 0.03 and decay it by 10× twice at 14 and 18 epochs. A learning rate warm-up is scheduled for 0.19 epochs starting from a learning rate of 0.0001.

We evaluate using both the Linear and Full finetune protocol. We compare methods that are pretrained in different ways: MoDist and the RGB-only baseline are pretrained self-supervisedly on K400, whereas R3D-50 [22] is pretrained with full supervision on K400. Rand Init is a randomly initialized network without pretraining (Table 4).

For the Full protocol evaluation, it's clear that pretraining on K400 is beneficial and improves by almost +10 top-1 accuracy. Next, MoDist outperforms the RGB-only baseline, showing once more the importance of motion distillation. Finally, when comparing to R3D-50 that is pretrained with full supervision on K400, MoDist not only closes the gap between self-supervised and fully-supervised methods, but it even outperforms the supervised pretraining (+1.9).

Furthermore, we test with the Linear protocol, which is much more challenging due to the large difference between the label spaces of K400 and SSv2. As shown in Table 4, unsurprisingly accuracy for all methods are much lower compared to Full finetune results. However, it's notable that the gap between MoDist and RGB-only significantly increases (+10.5 vs +2.5) compared to the Full protocol, which further demonstrates our method's strength in generalization. Moreover, it's interesting to see the supervised baseline underperform both self-supervised methods, as it's harder to overcome taxonomy bias under Linear protocol compared to the Full protocol for a representation pretrained with a fixed label taxonomy. We believe this is a promising example showing how self-supervised training can remove the label taxonomy bias that is inevitable under supervised settings, and lead to more general video representations that can be better transferred to new domains.

4.4. Action Detection on AVA

In previous sections we showed that self-supervised representation can generalize to new domains within the same downstream task (i.e., action recognition). However, we believe that self-supervised representations can go beyond that and also generalize to novel downstream tasks, since they do not optimize for any task specific objective. To test this, we transfer MoDist representations to the new task of action detection, which requires not only to recognize the action class, but also localize the person performing the action.

method	pretrain dataset	sup.	Full	Linear
R3D-50 [22]	K400	✓	55.5	16.3
Rand Init	-	✗	45.4	-
RGB-only	K400	✗	54.9	16.6
MoDist	K400	✗	57.4	27.1

Table 4: **Action classification on SSv2.** We pretrain on K400 and then transfer the representation to SSv2 for finetuning under both Linear and Full protocols. For finetuning, we use 16×8 clip as input following [22]. Results are reported as top-1 accuracy.

method	pretrain dataset	sup.	mAP
Faster-RCNN [22]	ImageNet	✓	15.3
Faster-RCNN [22]	K400	✓	21.9
Rand Init	-	✗	6.6
CVRL [59]	K400	✗	16.3
RGB-only	K400	✗	18.6
MoDist	K400	✗	22.1

Table 5: **Action detection on AVA.** We pretrain using different datasets (ImageNet and K400) and then transfer the representation to AVA for finetuning. We use 8×8 clip as input for finetuning [22]. CVRL numbers are taken from [59].

We evaluate action detection on the AVA dataset [29] which contains 211k training and 57k validation videos. Spatiotemporal labels (i.e., action classes and bounding boxes) are provided at 1 FPS rate. We follow the standard evaluation protocol and compute mean Average Precision over 60 action classes, using an IOU threshold of 0.5. We follow the Faster-RCNN detector design of [22] and use the Visual pathway architecture of Table 1 as the detector backbone. We fix the training schedule to 20 epochs with an initial learning rate of 0.1 and a batch size of 64 [22].

Results are shown in Table 5. Clearly, video pretraining plays a critical role in action detection, as demonstrated by the low mAP of 6.6 when training from scratch and the substantially lower AP achieved by supervised pretraining on ImageNet (pretrained 2D convs are inflated into 3D for fine-tuning [11]) compared to supervised pretraining on K400. As for self-supervised pretraining, both the RGB-only baseline and MoDist outperform ImageNet supervised pretraining, again demonstrating the importance of pretraining on videos. Moreover, MoDist again outperforms both the RGB-only baseline and the recent CVRL approach, which also only uses RGB inputs for pretraining.

Finally, we note that MoDist even outperforms the supervised Faster-RCNN pretrained on K400. To our best knowledge, we are the first to demonstrate that self-supervised video representation can transfer to action detection and match the performance of fully-supervised pretraining.

Figure 3: **Grad-CAM visualization for MoDist (left) and RGB-only (right) representations**. Predictions are overlaid on each frame. Best viewed in color, zoomed in. We encourage readers to check out the video version of these visualizations here: https://www.youtube.com/watch?v=TpOggtBN4yo.

4.5. Low-Shot Learning

We have demonstrated that the self-supervised representation learned with MoDist can perform very well on a variety of datasets and tasks when finetuned on the target domain. In this section, we investigate how its performance varies with respect to the amount of data available for finetuning. We evaluate using the Full Training protocol on the UCF101 dataset starting from just 1% of its training data (1 video per class) and gradually increase that to 100% (9.5k videos). We compare results against our two baselines: RGB-only and Supervised (Table 6). MoDist outperforms RGB-only across all training set sizes and it only requires 20% of the training videos to match the performance of RGB-only with 100% (89.1 vs 89.0). Another interesting observation is that the gap Δ between MoDist and RGB-only reaches its maximum with the smallest training set (1%), suggesting that motion distillation is particularly helpful for generalization in low-shot scenarios.

method	1%	5%	20%	40%	60%	80%	100%
Supervised	69.3	85.1	93.0	94.5	94.7	95.8	95.4
RGB-only	32.9	62.8	82.2	86.5	87.8	89.5	89.0
MoDist	42.8	71.9	89.1	91.3	92.9	93.4	94.0
Δ	+9.9	+9.1	+6.9	+4.8	+5.1	+3.9	+5.0

Table 6: **Low-shot learning on UCF101.** Rows indicate different pretrainings on K400, while columns vary the % of UCF training data used for finetuning. All results are top-1 accuracy.

4.6. Visualizing MoDist Representations

To gain deeper insights on what MoDist has learned in its representations, we adopt Grad-CAM [62] to visualize the spatiotemporal regions that contribute the most to the classification decisions on UCF101. In short, Grad-CAM works by flowing the gradients of a target class into the final conv layer to produce a coarse localization map highlighting the important regions responsible for the prediction. To avoid overwriting weights in finetuning, we freeze

the self-supervised pretrained weights of MoDist and RGB-only and train a linear classifier on top for visualization.

We show visualization for MoDist and RGB-only in Fig. 3. First, we observe that the representation learned by MoDist focuses more on the "motion-sensitive" regions (i.e., regions where object motion likely occur). For example, in A-1 (A indexing column, 1 indexing row), MoDist makes the correct prediction of "PommelHorse" by focusing its attention on the person carrying out the motion. The RGB-only model, on the other hand, incorrectly predicted "ParallelBars" as it finds "bar-like" straight lines in the background. This common pattern can also be observed in other examples, like A-2 (RGB-only model predicts "BlowDryHair" after finding hair textures) and A-3 (RGB-only model confuses ice with water surface). Furthermore, we can observe another type of behavior in B-3 and C-3. In both examples, the background scenes (gym) are associated with many fine-grained action classes (different gym activities), our model is able to distinguish them by focusing on the actual motion pattern. The baseline, instead, gets confused as it focuses too much on the background.

Finally, we present some failure cases in the last row of Fig. 3. For example, in A-4 MoDist correctly focuses on the right motion region (fingers), but confuses the finger motion of "Knitting" with "Typing". Similar patterns can also be seen in B-4 and C-4.

Conclusion

We presented MoDist to learn self-supervised video representations with explicit motion distillation. We demonstrated SOTA self-supervised performance with MoDist across various datasets and tasks. Moreover, we showed that MoDist representations can be as effective as representations learned with full supervision for SSv2 action recognition and AVA action detection. Given the simplicity of our method, we hope it will serve as a strong baseline for future research in self-supervised video representation learning.

References

[1] https://20bn.com/datasets/something-something/v2. 2, 6

[2] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020. 2, 5

[3] Linchao Bao, Baoyuan Wu, and Wei Liu. CNN in MRF: Video object segmentation via inference in a CNN-based higher-order spatio-temporal MRF. In *CVPR*, 2018. 2

[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 *CVPR*, 2020. 1, 2, 5

[5] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *ECCV*, 2016. 2

[6] Biagio Brattoli, Uta Buchler, Anna-Sophia Wahl, Martin E Schwab, and Bjorn Ommer. LSTM self-supervision for detailed behavior analysis. In *CVPR*, 2017. 1

[7] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020. 1

[8] Thomas Brox and Jitendra Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *T-PAMI*, 2011. 4

[9] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*, 2018. 1

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *NeurIPS*, 2020. 2

[11] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 7

[12] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 1, 2, 3, 6

[13] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 2

[14] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 2

[15] Jingchun Cheng, Yi-Hsuan Tsai, Wei-Chih Hung, Shengjin Wang, and Ming-Hsuan Yang. Fast and accurate online video object segmentation via tracking parts. In *CVPR*, 2018. 2

[16] N. Dalal, B. Triggs, and C. Schmid. Human detection using oriented histograms of flow and appearance. In *ECCV*, 2006. 2

[17] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 1

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 1

[19] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. DynamoNet: Dynamic action and motion network. In *ICCV*, 2019. 1, 2, 5

[20] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised Visual Representation Learning by Context Prediction. In *ICCV*, 2015. 2

[21] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Smagt, D. Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 4

[22] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *ICCV*, 2019. 4, 6, 7

[23] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NeurIPS*, 2016. 2

[24] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016. 1, 2

[25] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to track and track to detect. In *ICCV*, 2017. 2

[26] Basura Fernando, Hakan Bilen, Efstratios Gavves, and Stephen Gould. Self-supervised video representation learning with odd-one-out networks. In *CVPR*, 2017. 1, 2

[27] Katerina Fragkiadaki, Pablo Arbelaez, Panna Felsen, and Jitendra Malik. Learning to segment moving objects in videos. In *CVPR*, 2015. 4

[28] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. In *NeurIPS*, 2020. 2, 3

[29] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *CVPR*, 2018. 2, 7

[30] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 2

[31] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCV Workshops*, 2019. 5

[32] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. In *ECCV*, 2020. 5

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

[34] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 3

[35] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 3, 6

[36] João F Henriques, Rui Caseiro, Pedro Martins, and Jorge Batista. High-speed tracking with kernelized correlation filters. *T-PAMI*, 2014. 2

[37] Lianghua Huang, Yu Liu, Bin Wang, Pan Pan, Yinghui Xu, and Rong Jin. Self-supervised video representation learning by context and motion decoupling. In *CVPR*, 2021. 2

[38] Longlong Jing and Yingli Tian. Self-supervised spatiotemporal feature learning by video geometric transformations. *arXiv preprint arXiv:1811.11387*, 2018. 5

[39] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object detection in videos with tubelet proposal networks. In *CVPR*, 2017. 2

[40] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1, 4

[41] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *AAAI*, 2019. 5

[42] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *CVPR*, 2019. 2

[43] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018. 2, 5

[44] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 2, 5

[45] Ivan Laptev, Marcin Marszalek, Cordelia Schmid, and Benjamin Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 2

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

[47] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Flow-grounded spatial-temporal video prediction from still images. In *ECCV*, 2018. 2

[48] Aravindh Mahendran, James Thewlis, and Andrea Vedaldi. Cross pixel optical-flow similarity for self-supervised learning. In *ACCV*, 2018. 2, 6

[49] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 5

[50] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 2

[51] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *ECCV*, 2016. 1, 2

[52] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 2

[53] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017. 2

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

[55] Andrew Owens and Alexei A Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018. 2

[56] 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. *arXiv preprint arXiv:2003.04298*, 2020. 2, 5

[57] Federico Perazzi, Anna Khoreva, Rodrigo Benenson, Bernt Schiele, and Alexander Sorkine-Hornung. Learning video object segmentation from static images. In *CVPR*, 2017. 2

[58] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, 2020. 2, 5

[59] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv preprint arXiv:2008.03800*, 2020. 2, 4, 5, 6, 7

[60] Nawid Sayed, Biagio Brattoli, and Björn Ommer. Cross and learn: Cross-modal self-supervision. In *German Conference on Pattern Recognition*, 2018. 2, 6

[61] Nima Sedaghat, Mohammadreza Zolfaghari, and Thomas Brox. Hybrid learning of optical flow and next frame prediction to boost optical flow in the wild. *arXiv preprint arXiv:1612.03777*, 2016. 2

[62] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 8

[63] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. 1

[64] Irwin Sobel. History and definition of the sobel operator. 2014. 4

[65] Khurram Soomro, Amir Roshan Zamir, and M Shah. A dataset of 101 human action classes from videos in the wild. In *ICCV Workshops*, 2013. 2, 5

[66] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*, 2019. 5

[67] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, 2020. 4

[68] Yi-Hsuan Tsai, Ming-Hsuan Yang, and Michael J Black. Video segmentation via object flow. In *CVPR*, 2016. 2

[69] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *CVPR*, 2017. 2

[70] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 2

[71] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 2

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

[73] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 1, 2

[74] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 6

[75] Xiaolong Wang and Abhinav Gupta. Unsupervised Learning of Visual Representations using Videos. In *ICCV*, 2015. 2

[76] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 2

[77] Donglai Wei, Joseph J Lim, Andrew Zisserman, and William T Freeman. Learning and using the arrow of time. In *CVPR*, 2018. 1, 2

[78] Philippe Weinzaepfel, Jerome Revaud, Zaid Harchaoui, and Cordelia Schmid. Deepflow: Large displacement optical flow with deep matching. In *ICCV*, 2013. 4

[79] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 2, 3

[80] Fanyi Xiao and Yong Jae Lee. Video object detection with an aligned spatial-temporal memory. In *ECCV*, 2018. 2

[81] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2019. 5

[82] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 5

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

[84] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016. 2

[85] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-guided feature aggregation for video object detection. *ICCV*, 2017. 2