SCVRL: Shuffled Contrastive Video Representation Learning

Michael Dorkenwald[1]*, Fanyi Xiao[2], Biagio Brattoli[2], Joseph Tighe[2], Davide Modolo[2]
[1]Heidelberg University [2]AWS AI Labs

Abstract

We propose SCVRL, a novel contrastive-based framework for self-supervised learning for videos. Differently from previous contrast learning based methods that mostly focus on learning visual semantics (e.g., CVRL), SCVRL is capable of learning both semantic and motion patterns. For that, we reformulate the popular shuffling pretext task within a modern contrastive learning paradigm. We show that our transformer-based network has a natural capacity to learn motion in self-supervised settings and achieves strong performance, outperforming CVRL on four benchmarks.

1. Introduction

In recent years, self-supervised approaches have shown impressive results in the area of unsupervised representation learning [10, 19]. These methods are not bound to categorical descriptions given by labels (e.g., action classes) and thus are able to learn a representation that is not biased towards a particular target task. In this context, rich representations can be learned using pretext tasks [7, 17, 34, 47, 50, 51, 54] or, more recently, using contrastive learning objectives [9–12, 19–21, 32, 38, 52] that led self-supervised performance to outperform the fully-supervised counterpart in the image domain [10, 19].

While contrastive learning in images has advanced the state-of-the-art, it remains relatively under-explored in the video domain, where the few existing works applied contrastive learning to videos [22,38,40] in a way that primarily captures the semantics of the scene and disregards motion. For example, CVRL [38] adapted the SimCLR image-based contrastive framework [10] to videos by forcing two clips from the same video to have similar representations, while pushing apart clips from different videos (Fig. 2, right). This leads to strong visual features that are particularly effective on datasets such as UCF [44], HMDB [25], and Kinetics [23], where context and object appearance matter more than motion information. However, forcing the representations of two clips from a single video to be the same induces invariance to temporal information. For example, in a "high jump" video, this will force the video encoder to embed the "running phase" at the start of the video to use the same representation of the "jumping phase" at the end, even though these contain very different motion patterns. Instead, we propose a new framework that can learn semantic-rich and motion-aware representations.

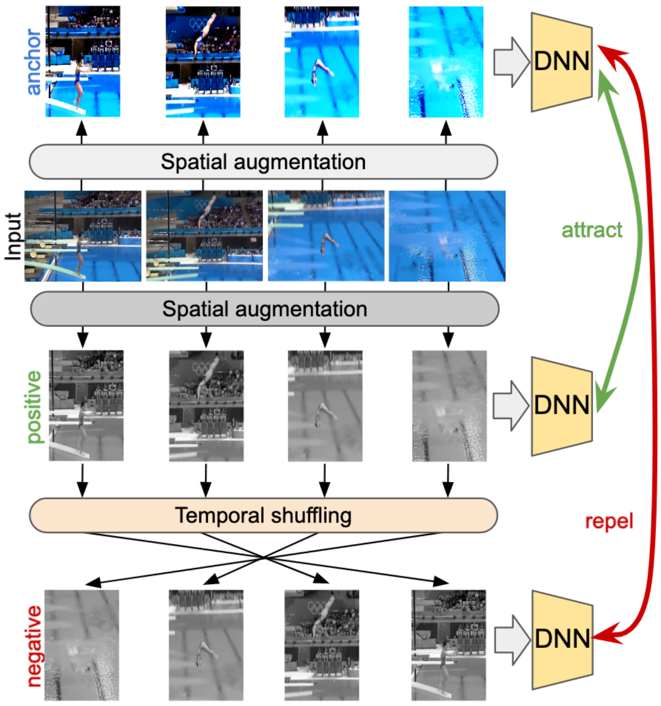

Figure 1. **Our temporal contrastive loss.** We reformulate the popular pretext task of frame shuffling within a novel contrastive learning framework, where all the contrasted samples (anchor, positive and negative) are augmentations of the same clip (input).

To accomplish this task, we borrow inspiration from the literature of self-supervised pretraining using pretext tasks for videos, like shuffle detection [7, 8, 27, 54], order verification [22] or frame order prediction [26, 54]. These methods learn by predicting a property of the video transformation (*e.g*. the original order of the shuffled frames). While achieving competitive results, they learn representations that are covariant to their transformations [32], hindering their generalization potential for downstream tasks.

[*] This work was done during an internship at AWS.

Instead, we reformulate the previous shuffle detection pretext task in a contrastive learning formulation that yields stronger temporal-aware representations which learn motion beyond just frame shuffling and can therefore better generalize to new tasks.

In detail, we propose SCVRL, a new contrastive video representation learning method that combines two contrastive objectives: a novel shuffled contrastive learning and a visual contrastive learning similar to [38]. In order to encourage the learning of motion-sensitive features, our shuffled contrastive approach forces two augmentations of the same clip (anchor and positive, Fig. 1) to have similar representations, while pushing apart negative clips generated by temporally shuffling the positive clip. Since all these samples come from the same original clip, during training the model cannot just look at their visual semantics to solve the contrastive objective, but is instead forced to reason about their temporal evolution, which is the key to the proposed learning. Thanks to the rich combination of two contrastive objectives, SCVRL learns representations that are both motion and semantic-aware. SCVRL trains in an end-to-end fashion and its design consists of a single feature encoder with two dedicated MLP heads, one for each of the contrastive objectives.

As feature encoder, SCVRL adopts a Multiscale Vision Trasformer [55], which is well suited for learning the rich SCVRL objective. We evaluate SCVRL on four popular benchmarks: Diving-48 [28], UCF101 [44] and HMDB [25] and Something-Something-v2 [18]. We show strong performance, consistently higher than the CVRL baseline on all datasets and metrics. Additionally, we also conduct an extensive ablation study to show the importance of our design choices and investigate the extent to which SCVRL learns motion and semantic representations.

2. Related Work

Self-supervised learning for images. Early works on self-supervised image representation learning used various pretext tasks such as image rotation prediction [17], auto-encoder learning [37,43,46], or solving jigsaw puzzles [34] to learn semantic generalizable representations. That led to promising results, unfortunately still far away from those of fully-supervised models, mostly due to the difficulties of preventing the network from learning short-cuts. That changed with the (re-)emergence of approaches based on contrastive learning [9–12, 19–21, 32, 52]. The underlying idea behind these approaches is to attract representations of different augmentations of the same image (positive pair) while repelling against those of different instances (negative pair) [2, 36]. Thanks to this training paradigm, recent works were able to produce results on par with those of fully-supervised approaches [9–12, 19, 21, 32].

Self-supervised learning for videos. In the video domain, many works exploited the temporal structure of videos by designing specific pretext tasks such as pace prediction [3,48], order prediction [7,8,26,33,51,54], future frame prediction [5, 6, 14, 15] or by tracking patches [49], pixels [50] or color [47] across neighboring frames. More recently, several approaches [22,38,40,53] adopted contrastive learning objectives from the image domain to learn stronger video representations. Among these, *CVRL* [38] was the first to propose a video-specific solution for sampling pairs (positives and negatives) for contrastive video learning. In detail, CVRL proposed to sample positive pairs of clips from within the same video (their only constraint is that they cannot be too close to each other temporally) and negatives from other videos. This achieves impressive results on downstream tasks and it inspired this work (SCVRL). Specifically, it inspired us to design a novel sampling strategy that is even more suitable for contrastive video learning and that encourages the representation to learn *both* semantic (as CVRL) and motion cues (using our novel shuffling contrastive learning). Finally, note how CVRL also inspired other very recent works [22, 40]. [40] tried to match the representation of a short clip to the one of a long clip, while [22] tried to incorporate temporal context using an order verification pretext task. We argue that none of these approaches explicitly enforce the representation to learn motion information like our SCVRL.

Vision Transformers. In recent years, transformer models [13, 45] have achieved unprecedented performance on many NLP tasks. Dosovitskiy et al. [16] adapted them to the image-domain with a convolution-free architecture (ViT) achieving competitive results on image classification and sparking a new research trend in the field. Since then, many works have been proposed, including some in the video domain [1,4,55,56]. Among these, MViT [55] is of particular interest to this paper, as we use it as the feature encoder for SCVRL. MViT proposes a multiscale feature hierarchy for transformers to effective model dense visual inputs without the need for external data. It produces state-of-the-art results on various video classification benchmarks and we believe its architecture design is well suited to learn the contrastive objective of SCVRL.

3. Method

SCVRL is a novel contrastive learning framework that learns representations which are *both* rich in semantics and sensitive to motion cues. To achieve this, it leverages two objectives: a classic *visual* contrastive objective to learn the semantics of the video (e.g., CVRL [38]) and a novel objective which compares a video clip and the same clip *temporally shuffled* (Fig. 1), such that the learnt representation is aware of the temporal order of an action and can distinguish

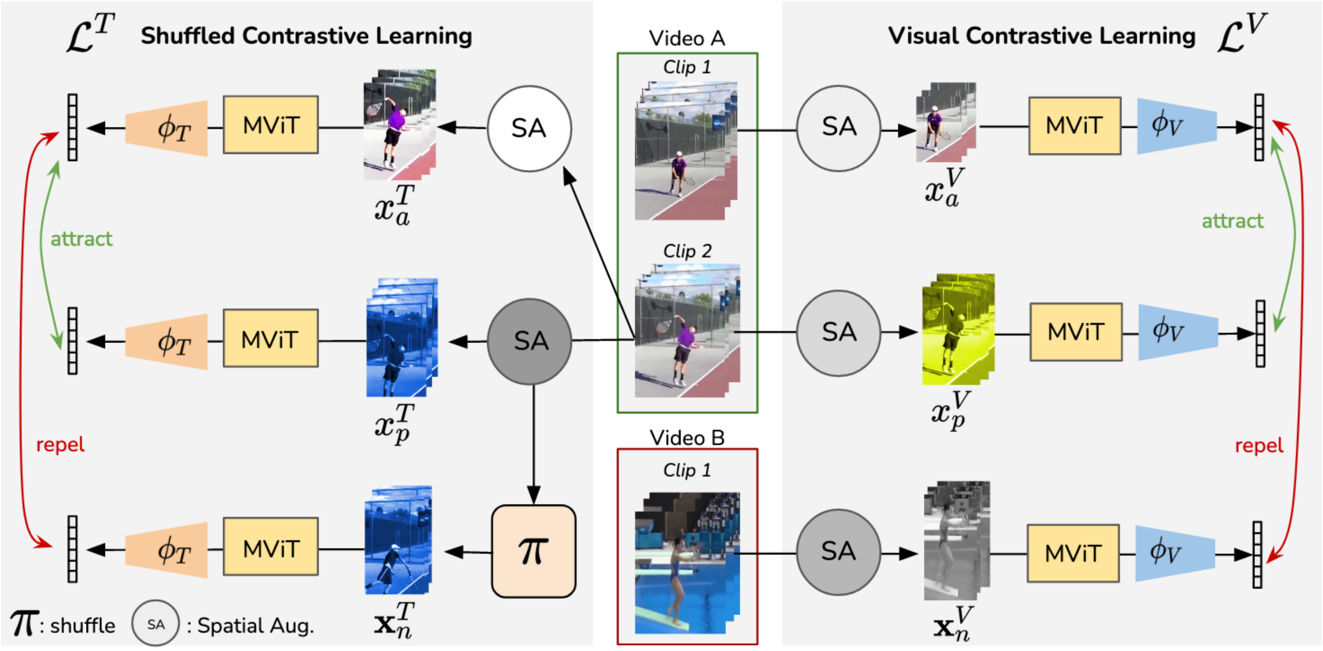

Figure 2. **Method overview.** Our framework consists of two contrastive objectives: *(i)* our novel shuffled contrastive learning (left) is computed using the same clip (Video A, Clip 2) for all three samples, in fact anchor (top) and positive (middle) only differ by their spatial augmentation, then the positive is **shuffled** (π) over the time dimension to produce the negative sample (bottom); *(ii)* visual contrastive learning (right), similarly to CVRL [38] uses two clips from the same video (Video A) are used as anchor (Clip 1, top) and positive (Clip 2, middle) and a clip from a different video (Video B) constitutes the negative sample (bottom). Each sample is then run through the respective network composed of a shared backbone and two different heads, one for each loss. The visual contrastive loss extracts only semantic information and forces temporal invariance. The shuffled contrastive loss encourages the network to learn temporal information since it is the only characteristic that distinguishes positive from negative.

its different phases.

3.1. Preliminaries on Contrastive Learning

The objective of contrastive learning is to produce an embedding space by attracting positive pairs, x_a (anchor) and x_p (positive), while pushing away a set of N negatives $\mathbf{x_n} = \{x_{n_1}, \dots, x_{n_N}\}$. This is achieved by training an encoder f which embeds a given video clip x to a $\mathbb{L}_2$ normalized feature vector using the InfoNCE [21, 36] loss $\mathcal{L} = \text{IN}(f; x_a, x_p, \mathbf{x}_n)$ which is formulated as follows:

$$\mathcal{L} = -\log \frac{e^{f(x_a)^T \cdot f(x_p)/\tau}}{e^{f(x_a)^T \cdot f(x_p)/\tau} + \sum_{i=1}^{N} e^{f(x_a)^T \cdot f(x_{n_i})/\tau}}, \quad (1)$$

with $\tau > 0$ as a temperature parameter. It is clear from this formulation that what the model learns largely depends on how the positive and negative pairs are sampled. In the next section we delve into how we construct training samples for our shuffled contrastive learning objective and in the following one we then present our SCVRL framework.

3.2. Shuffled Contrastive Learning

The key to ensure that $\mathcal{L}$ learns a motion-aware representation space lies in how positive, negative, and anchor clips are sampled during training. Differently from previous approaches (e.g. CVRL [38]), that sample positive pairs from one video and negatives from other videos, we use a single clip to produce anchor, positive and negatives and only change the temporal order of its frames as a mean to learn temporal sensitive features.

Specifically, we extract a single clip from a video and apply two different spatial augmentations to obtain our positive pair x_a^T and x_p^T (Fig. 2). Our negatives are then generated by applying a series of temporal permutations $\pi_i \in \Pi$ to the positive clip: $x_{n_i}^T = \pi_i(x_p^T)$ (i.e., *shuffling*). In the context of Eq. 1, we define our temporal shuffled contrastive objective as $\mathcal{L}^T = \text{IN}(f; x_a^T, x_p^T, \mathbf{x}_n)$. This objective forces the encoder to push apart the representations of the anchor clip x_a^T (which is in normal order) and the shuffled clip $\mathbf{x}_n^T$. This makes the learned representation sensitive to frame ordering (*i.e.*, motion-aware), as these pairs x_a^T and $\mathbf{x}_n^T$ share the same semantics and the model can only repel them using motion information. Furthermore, in order to avoid trivial solutions, an important design choice is to make sure $x_{n_i}^T$ and x_a^T do not share the same spatial extent. To achieve this, we generate $x_{n_i}^T$ by permuting x_p^T rather than x_a^T. This critical design avoids the shortcut where the encoder solves the

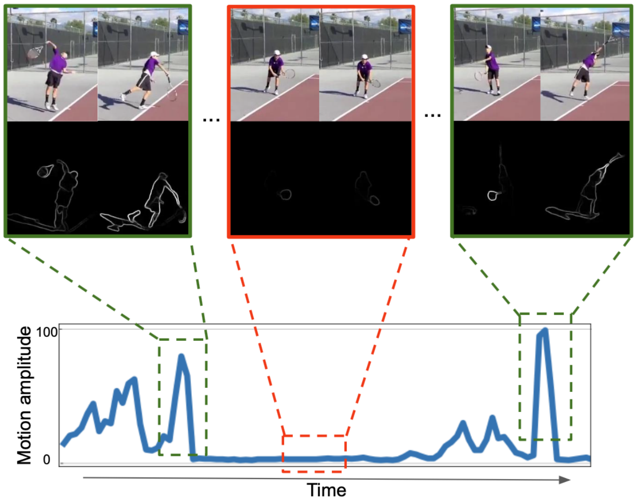

Figure 3. **Targeted Sampling.** During self-supervised training, we select clips with high motion. For each video, we compute the frame difference (middle), select the top 4K pixels and calculate the media over 1 second (bottom plot). In this example, the first and last clip (green boundaries) are selected given that the tennis player is moving (hitting the ball). The middle clip (red boundaries) is excluded from self-supervised training since the subject is only preparing the ball and not moving, therefore shuffling these frames would not produce different enough samples to learn motion information.

task with a simple pixel-level comparison between frames.

At the same time, this design also carries a potential risk: if x_p^T is a clip containing almost no motion (e.g., static scenes), then its shuffled representation $x_{n_i}^T$ would look identical to it and the encoder would not be able to push them apart and learn correctly. To overcome this problem and take full advantage of our shuffling contrastive learning method, we propose a new targeted sampling approach that aims at selecting the right clips for shuffling.

Probabilistic targeted sampling. Given a video, we want to sample clips that have strong motion for shuffling, as visualized in Fig. 3. To measure the motion of each clip, we use optical flow edges [53], which have been shown to be more robust to global camera motion compared to raw optical flow. We estimate flow edges by applying a Sobel filter [42] onto the flow magnitude map and take the median over the highest 4k flow edge pixels values in each frame. Then, we aggregate them in time by taking the median over all the frames in a 1-second temporal window (m_i). Rather than deterministically sampling the clip with the highest motion (which would disregard a lot of useful clips), we propose to sample from a multinomial distribution based on the probability p_i computed from window i by inserting its motion amplitude m_i into a softmax function:

stages	operators	output sizes
data layer	stride $4 \times 1 \times 1$	$3 \times 16 \times 224^2$
cube$_1$	$\mathbf{2} \times 7 \times 7, 96$ stride $2 \times 4 \times 4$	$96 \times 8 \times 56^2$
scale$_2$	$\begin{bmatrix} \text{MHPA(96)} \\ \text{MLP(384)} \end{bmatrix} \times 1$	$96 \times 8 \times 56^2$
scale$_3$	$\begin{bmatrix} \text{MHPA(192)} \\ \text{MLP(768)} \end{bmatrix} \times 2$	$192 \times 8 \times 28^2$
scale$_4$	$\begin{bmatrix} \text{MHPA(384)} \\ \text{MLP(1536)} \end{bmatrix} \times 11$	$384 \times 8 \times 14^2$
scale$_5$	$\begin{bmatrix} \text{MHPA(768)} \\ \text{MLP(3072)} \end{bmatrix} \times 2$	$768 \times 8 \times 7^2$
	Output: CLS token	

Table 1. **SCVRL backbone architecture**. The network is the same as MViT-B [55] except for the first layer (cube$_1$) where the temporal kernel size is reduced to avoid shortcuts (see Sec. 4.1). The dimensions of the output size is denoted as $C \times T \times H \cdot W$.

$$p_i = \frac{e^{m_i/\beta}}{\sum_j^C e^{m_j/\beta}}, \tag{2}$$

where β is the temperature which regularizes the entropy of the distribution and C is the number of temporal windows inside the video.

3.3. SCVRL Framework

Our self-supervised SCVRL framework learns a video representation that is both semantic-rich and motion-aware. To achieve this, it combines two contrastive objectives: our **T**emporal shuffled contrastive learning objective $\mathcal{L}^T$ and a **V**isual contrastive learning objective $\mathcal{L}^V$:

$$\mathcal{L}^F = \mathcal{L}^T + \lambda \mathcal{L}^V, \tag{3}$$

where λ is a weighting parameter and $\mathcal{L}^V$ is the contrastive objective used in CVRL [38] that helps SCVRL learn semantic features. Its objective is illustrated in Fig. 2 (right) and is defined as: $\mathcal{L}^V = \text{IN}(f; x_a^V, x_p^V, \mathbf{x}_n^V)$, where $x_a^V = x_a^T$, x_p^V is a different clip from the same video of x_a^V and the negative clips $\mathbf{x}_n^V$ are sampled from entirely different videos. To the best of our knowledge, SCVRL is the first work that explicitly models both motion and visual cues within the same self-supervised contrastive learning framework. Finally, we note that one cannot naively combine these two competing objectives as $\mathcal{L}^V$ wants to pull together the representation of all the clips within a video, while $\mathcal{L}^T$ wants to push them away when shuffled. To circumvent this, we design SCVRL with a shared backbone encoder f, but two independent MLP heads ϕ_V and ϕ_T, one for each of the objectives (Table 8b, Fig. 2) [30, 31]. For the backbone video encoder, we adopt Multiscale Vision Transformer [55] (MViT), which has a large temporal receptive

Method	Shuffle	UCF	Diving-48
Rand Init	✗	13.4	9.5
Rand Init	✓	12.8 (–0.6)	8.7 (–0.8)
CVRL	✗	67.4	12.0
CVRL	✓	62.9 (–3.5)	9.5 (–2.5)
SCVRL	✗	68.0	11.9
SCVRL	✓	56.2 **(–11.8)**	8.1 **(–3.8)**

Table 2. **Dependency on motion information.** We evaluate the performance drop when shuffling the input during inference as introduced by [41,55]. A large drop indicates that the network relies on temporal information to solve the task. A small drop, on the other hand, the model is basing the decision only on per-frame semantic. The models are trained using the linear evaluation protocol for the corresponding dataset and we report top-1 accuracy. The performance drop is indicated in red brackets.

field that makes it particularly suitable for learning motion cues using our shuffled contrastive learning objective.

4. Experiments

In this section, we first describe the implementation details and the benchmark datasets we use in our experiments. Then, we investigate the motion information extracted by our SCVRL compared to the baseline. Finally, we evaluate our model on the downstream task of action recognition and perform several ablation studies to better understand the impact of our design choices.

4.1. Implementation Details

Category	Motion	Δ Acc. $\uparrow$
Turning the camera downwards while filming sth	0.69	22.5
Turning the camera left while filming sth	0.95	17.8
Digging sth out of sth	0.64	12.1
Turning the camera upwards while filming sth	0.89	11.6
Uncovering sth	0.65	10.8
Showing a photo of sth to the camera	0.25	-1.9
Showing sth on top of sth	0.08	-2.6
Scooping sth up with sth	0.43	-2.9
Pulling two ends of sth so that it gets stretched	0.43	-3.3
Throwing sth against sth	0.42	-5.0

Table 3. **Performance gain related to motion.** For each SSv2 action class, we compare the absolute difference in accuracy between SCVRL and CVRL (Δ Acc.) using linear evaluation and the normalized motion. For this table, we show five of the highest and lowest performing classes, sorted by Δ Acc. The result shows that SCVRL improves on actions with high motion by a large margin. At the same time, CVRL outperforms on classes with low motion by only a marginal difference.

Training protocol. We pretrain SCVRL on the Kinetics-400 (K400) dataset [24], which contains around 240k 10-seconds videos, *without* the use of the provided action annotations. Our final model is pretrained on K400 for 500 epochs. Since the training of transformers is computationally very expensive, we validate the strong performance of our model by comparing to the CVRL baseline on a training scheme using 100 epochs. For efficiency, we only train for 50 epochs for our ablation study on a subset of K400 containing 60k videos that we call *Kinetics400-mini*. We train SCVRL with a learning rate of 1×10^{-4} with linear warm-up and cosine annealing using AdamW optimizer [29] with a batch size of 4 per GPU and weight decay of 0.05. We set the warm-up and the end learning rate to 1×10^{-6}. We set λ in Eq. 3 to 1. The temperature β for our targeted sampling is set to 5 as ablated in Fig. 8a. The spatial augmentations are generated with 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. The same augmentation is applied to all frames within a clip. To effectively train SCVRL with $\mathcal{L}^V$ we follow [21,52] and construct a *memory bank* of N^V negative samples. To train with $\mathcal{L}^T$, instead, we generate negatives on the fly by randomly permuting the positive clip N^T times. For the visual contrastive objective, we use a memory bank size N^V of 65536 negative samples, while for the temporal counterpart we set N^T to 12. For both objective we set the temperature τ to 0.1. As suggested in [21] we maintain a momentum version of our model and process anchor clips with our online model while positive and negat clips are processed with the momentum version.

Architecture. We use the *MViT-Base* (MViT-B) version of MViT and operate on clips of 16 frames which are extracted with a stride of 4. Our architecture is shown Tab. 1. As in previous works [1, 55] we use a cube projection layer to map the input video to tokens. This layer is designed as a 3D convolution with a temporal kernel size of 3 and stride of 2 with padding of 1. However, when the input sequence is shuffled, the projection layer has direct access to the seams between the shuffled frames. This would allow the network to learn shortcuts by directly detecting if a sequence is shuffled. We ablated this behavior in Tab. 8e. To overcome this issue, we propose to shuffle groups of two frames and use a temporal kernel size to 2 with a stride of 2, thus avoiding convolutional kernels overlapping across shuffled tokens. Finally, each MLP head in SCVRL is a two-layer network that consists of a linear layer that transforms the CLS token to a 2048 dimensional feature vector, followed by a ReLU activation function, and a second linear layer that maps to an embedding of 128-D.

Evaluation protocol and Baselines. Following [38,40], we employ two evaluation protocols to quantify self-supervised representations: (i) *Linear*: Training a linear classification layer on top of the frozen pretrained backbone (ii) *Full*: finetuning the whole network in an end-to-end fashion on the target dataset. For all evaluations we report the top1 ac-

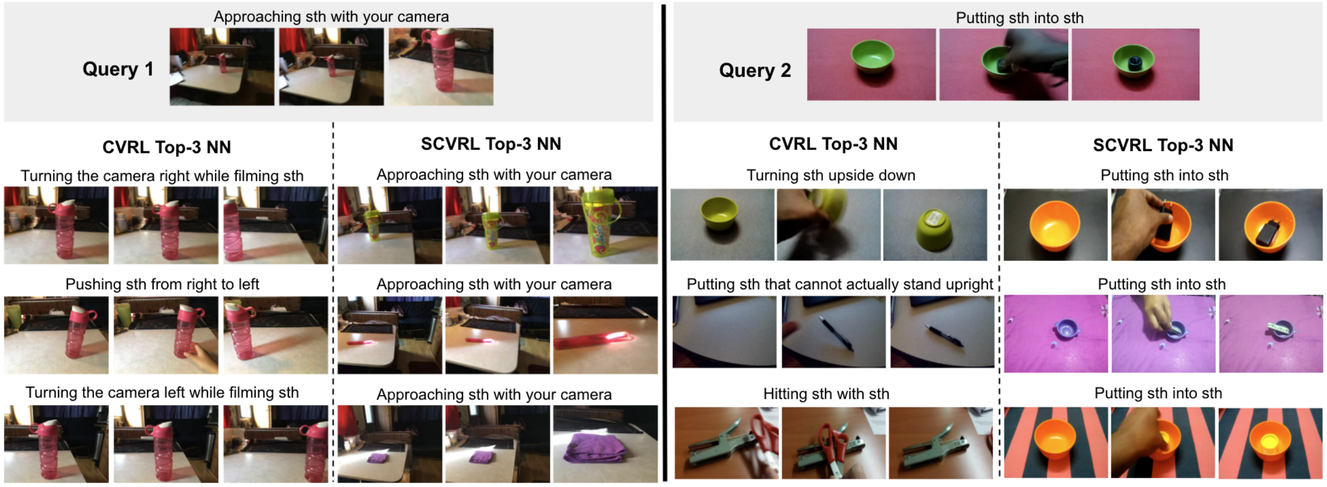
Approaching sth with your camera
Query 1
Putting sth into sth
Query 2
CVRL Top-3 NN
SCVRL Top-3 NN
CVRL Top-3 NN
SCVRL Top-3 NN
Turning the camera right while filming sth
Approaching sth with your camera
Turning sth upside down
Putting sth into sth
Pushing sth from right to left
Approaching sth with your camera
Putting sth that cannot actually stand upright
Putting sth into sth
Turning the camera left while filming sth
Approaching sth with your camera
Hitting sth with sth
Putting sth into sth

Figure 4. **Video retrieval comparison.** For each query, we show the top three nearest-neighbor based on CVRL and SCVRL representations. For each video, we show the first, middle and last frame. The ground-truth class is provided on top. The figure illustrates that CVRL is biased towards appearance while SCVRL is aware of motion patterns. For example, in Query 1 CVRL retrieves videos of the same scene even if the motion (i.e. action class) is different, while SCVRL retrieves nearest neighbors with different appearance but similar motion.

curacy. We follow previous works and use a standard of 10 temporal and 3 spatial crops during testing. We compare against two baselines: (i) *CVRL* [38], a state-of-the-art self-supervised contrastive learning method trained using the visual contrastive loss $\mathcal{L}^V$ of Eq. 3; and (ii) *Supervised*, which is a fully supervised model trained for actions on K400. All baselines use the same MViT-B backbone as SCVRL.

4.2. Datasets

Diving-48 [28] is a fine-grained action dataset capturing 48 unique diving classes. It has around 18k trimmed video clips, each containing a diving sequence that consists of a mix of takeoff, the motion during the dive and water entering. It is a challenging dataset due to the semantic similarity across all diving classes (e.g., similar foreground and background, similar diving outfits, etc.). As the original Diving-48 dataset had some annotation issues, we follow [4] and use their cleaned annotations.

Something-Something-v2 [18] (SSv2), similarly to Diving-48, is a benchmark that was specifically developed to evaluate a model's capability to learn temporal dynamics. SSv2 consists of video clips showing complex human-object interactions, such as "Moving something up" and "Pushing something from left to right". It contains a total of 174 unique actions, 168k training videos, 24k validation videos and 24k test videos.

UCF101 [44] and HMDB [25] are both standard benchmark datasets for the classification of sport activities and daily human actions. UCF contains 13320 YouTube videos labelled with 101 classes, while HMDB contains 6767 video clips labelled with 51 actions. Differently from the previous two datasets, these can reliably be solved using mostly semantic cues, as their action classes are quite distinct. For all experiments on UCF101 and HMDB51, we report results using split1 for train/test split.

4.3. Analysis of Motion Information

The goal of our shuffled contrastive learning objective is to learn stronger temporal features. We now quantify the amount of temporal information learned by SCVRL in comparison to CVRL.

Dependency on motion information. First, we investigate to what extend SCVRL has learnt to distinguish temporally coherent sequences compared to shuffled ones. For this, we follow previous works [41, 55] and compute the performance change between running inference on a test clips *vs.* its shuffled version. Results using the Linear evaluation protocols are presented in Tab. 2. These show that SCVRL is very sensitive to the temporal order of the frames and its performance drops by $\sim$ 12 points Top-1 accuracy on UCF. This validates the effectiveness of our shuffled contrastive learning strategy. On the contrary, CVRL's performance on UCF is barely affected (-3.5), confirming that CVRL's representations are temporal invariant when trained on datasets that mostly rely on semantic cues (i.e., UCF).

Correlation between motion quantity and performance. We now compare per-class performance change between SCVRL and the baseline CVLR in Tab. 3 and correlate it to the amount of motion each action contains. For this, we calculate the median motion magnitude in the pixel space for all videos of each class. We show five high (top) and five low performing (bottom) classes sorted by their

	Method	Pretrain data	Sup.	Full	Linear
100 Epochs	Rand Init	–	✗	18.4	6.9
	CVRL	K400	✗	52.6	**12.1**
	SCVRL	K400	✗	**53.8**	11.9
	MViT	K400	✓	68.8	22.3
	SCVRL	K400	✗	66.4	18.1

Table 4. **Action classification on Diving-48.** We pretrain on K400 and then transfer the representation to Diving48 for finetuning under both Linear and full protocols. Results are reported as top-1 accuracy. SCVRL outperforms the CVRL baseline by a significant margin. For context, we also compare to supervised video models pre-trained on Imagenet-1K and show that our model outperforms them.

	Method	Pretrain	Arch.	Sup.	UCF		HMDB	
					Full	Linear	Full	Linear
100 Epochs	Rand Init	K400	MViT-B	✗	68.0	18.7	32.4	13.6
	CVRL	K400	MViT-B	✗	83.0	67.4	54.6	**42.4**
	SCVRL	K400	MViT-B	✗	**85.7**	**68.0**	**55.4**	40.8
	Supervised	K400	MViT-B	✓	93.7	93.4	68.8	67.6
	SCVRL	K400	MViT-B	✗	89.0	74.4	62.6	50.1
	CVRL [38]	K400	R3D-50	✗	92.9	89.8	67.9	58.3

Table 5. **Action classification on UCF and HMDB.** We pretrain on K400 and then transfer the representation to UCF101 and HMDB for Linear and Full finetuning. Results are reported as top-1 accuracy.

performance gains. First and foremost, the improvement brought by SCVRL is substantial (top: 10-20 Top-1 accuracy points) compared to its loss (bottom: 2-5 points). Second, SCVRL improves particularly on actions with large motion, like "Turning the camera left", showing that it is important to model motion cues in video representations and that SCVRL is capable to do that well.

Video retrieval comparison. In Fig. 4, we compare the nearest neighbors obtained using the representation space learned by SCVRL to the one from CVRL on the SSv2 dataset. For each query we show the top-3 nearest neighbors. For both queries we observe that SCVRL better captures the temporal information in the query and the top-3 nearest neighbor are very similar in motion while having a large variation in semantics. In Query 1, the nearest neighbor from CVRL have all the same appearance, yet, completely different progressions. Also in Query 2 CVRL top-1 nearest neighbor displays a similar object as the query (green cup) but different motion, while SCVRL correctly retrieves a clip with the same motion even if the cup has a different color. This illustrates that the learned features from SCVRL better captures the missing temporal information in CVRL.

4.4. Downstream Evaluation

Next, we evaluate SCVRL on four datasets introduced in Sec. 4.2 for action recognition, using the two evaluation protocols of Sec. 4.1.

	Method	Arch.	Pretrain	Sup.	Full	Linear
100 Epochs	Rand Init	MViT-B	–	✗	38.7	3.2
	CVRL	MViT-B	K400	✗	45.3	11.4
	SCVRL	MViT-B	K400	✗	**46.8**	**13.8**
	MViT	MViT-B	K400	✓	53.7	19.4
	SCVRL	MViT-B	K400	✗	53.5	19.4
	MViT [55]	MViT-B	K400	✓	64.7	–

Table 6. **Action classification on SSv2.** We pretrain on K400 and then transfer the representation to SSv2 for finetuning under both Linear and Full protocols. Top-1 accuracy is reported in the table.

Method	1%	5%	10%	25%	50%	100%
Rand Init	1.5	2.9	7.8	18.7	28.5	40.0
CVRL	4.4	15.4	23.3	31.0	40.1	45.3
SCVRL	6.1	19.4	26.3	34.4	43.0	46.8
rel. Δ%	+27.9	+20.6	+11.4	+9.9	+6.7	+3.2

Table 7. **Low-shot learning on SSv2.** Rows indicate different pretraining method on K400, while columns vary the % of SSv2 training data used for finetuning. All results are top-1 accuracy. Our method (SCVRL) consistently provides an higher gain than CVRL and, in particular, achieves higher gain when the amount of annotations are lower.

Action Classification on Diving-48. Tab. 4 shows that our SCVRL achieves better performance than baseline for full finetuning on Diving-48 while being on par for the linear evaluation protocol.

Action Classification on UCF and HMDB. In Table 5 we show the top-1 accuracy on UCF and HMDB for both evaluation protocols after pretraining on Kinetics400 for 100 epochs (details in sec. 4.1). SCVRL outperforms its corresponding baseline CVRL on UCF and HMDB when finetuning the full model. For the linear evaluation settings, we observe a marginal gain on UCF and a small loss on HMDB. We argue this is likely due to the semantic-heavy nature of these datasets (as opposed to motion, like Diving-48 and SSv2). Our fully trained model achieves inferior performance when comparing to CVRL trained using a ResNet R3D-50 architecture. Note, this gap is not caused by the method, instead, it is due to the architecture choice.

Action Classification on SSv2. In Table 6 we compare SCVRL and the baseline CVRL, both pretrained on Kinetics-400. SCVRL consistently outperforms its baseline, by 1.5% on full-finetuning and 2.4% on linear. Finally, note how we omit the recent state-of-the-art CORP [22] from the table since it is not compatible on SSv2 (i.e., differently from previous methods, it directly pretrains on the *target dataset*).

Temp. β	UCF	SSv2
∞	75.0	43.8
10	75.1	43.9
5	**75.5**	**44.3**
3	74.1	43.1

(a) **Targeted sampling**

Method	UCF	SSv2
CVRL	70.3	41.9
SCVRL shared	69.6	40.7
SCVRL separate	**75.5**	**44.3**

(b) **Contrastive heads**

Method	UCF	SSv2
CVRL	70.3	41.9
+ cls shuffle	70.3	42.6
SCVRL	**75.5**	**44.3**

(c) **Ablate $\mathcal{L}_T$**

Method	UCF	SSv2
CVRL	70.3	41.9
SCVRL CLS	**75.5**	**44.3**
SCVRL AVG	74.2	43.9

(d) **CLS vs AVG pooling**

Method	k_t	UCF	SSv2
CVRL	2	70.3	41.9
SCVRL	2	**75.5**	**44.3**
CVRL	3	72.6	41.7
SCVRL	3	72.7	41.9

(e) **Proj. layer kernel size**

Table 8. **Ablating SCVRL**. We present top-1 classification accuracy using the Linear evaluation protocol on UCF and SSv2. For efficiency, we train the corresponding model on K400-mini since we are looking at relative improvements.

4.5. Low-Shot Learning on SSv2

In Table 7 we evaluate our model in the context of low-shot (or semi-supervised) learning, i.e. given only a fraction of the available train-set during finetuning. Both CVRL and SCVRL are pretrained on the full Kinetics400 training set. The train-data ratio (top of the table) only applies to SSv2 finetuning. We perform the finetuning for 1%, 5%, 10%, 25%, 50% and 100% of SSv2 train data and show that SCVRL consistently outperforms CVRL by a significant margin. In particular, the relative gain of SCVRL is higher when the amount of supervised data for the target domain (i.e., SSv2) is lower. This shows that learning representations that capture both semantic and motion information, as in SCVRL, leads to representations that are more generalize and more transferable.

4.6. Ablation studies

We now provide detailed ablation studies on different components and design choices. For these, we pretrain on the Kinetics-400-mini and finetune on UCF and SSv2.

Targeted sampling (Table 8a). We evaluate the effect of our probabilistic targeted sampling on the final performance. We observe a performance boost with targeted sampling (temperature $\beta = 5$) compared to uniform sampling (temperature $\beta = \infty$). Interestingly, with a low temperature value (3), which corresponds to a low entropy of the distribution of weights w_i, we experience a significant loss in performance. This is likely caused by the fact that this choice excludes a lot of valuable training clips and mostly focuses on a (relatively) fixed subset.

Head configurations (Table 8b). In Sec. 3.3 we conjectured about the importance in SCVRL of having two MLP heads, one dedicated to each contrastive objective. Here we now evaluate this choice and compare against SCVRL trained using a single head ($\phi_v = \phi_t$). The results validate our design, as they show that the performance degrades considerably when we use a single head, to the point where it is even worse than the CVRL baseline.

Importance of contrastive formulation (Table 8c). In this ablation we compare our proposed objective $\mathcal{L}^T$ against a traditional pretext task approach [7,35] based on a classifier trained using a cross-entropy loss to detect if a sequence is shuffled. This loss objective is combined with the visual contrastive objective $\mathcal{L}^V$ as in SCVRL. The results show that the proposed objective achieves much higher performance for both UCF and SSv2, validating the importance of reformulating pretext tasks using contrastive learning.

Used output feature representation (Table 8d). As studied in the supervised setting [39], the design choice between using a CLS token or a representation computed from spatial average pooling induces different model behaviors, especially on how localized each tokens are. However, it is not clear how that affects representations in a self-supervised setting. Hence, we compare SCVRL, which is trained on the output CLS token from MViT-B, against a different version trained using the average pooled token over the time and space dimensions (AVG). We obtain stronger results for the model which operates on the CLS token.

Shortcuts in projection layer (Table 8e). In Sec. 4.1 we discussed how for SCVRL we modify the temporal kernel size of MViT to avoid the network from learning shortcuts. We now ablate two potential choices: 2 and 3. The results show that SCVRL only significantly improves upon CVRL only for $k_t = 2$. This is likely caused by the projection layer having directly access to the gaps between shuffled frames and with that, can detect if a sequence is shuffled. Thus, the remaining part of the transformer is not challenged by the loss. That is why it is crucial for our framework to be trained with a kernel size which is aligned with the number of frames shuffled in each group.

5. Conclusion

We presented SCVRL, a novel shuffled contrastive learning framework to learn self-supervised video representations that are both motion and semantic-aware. We reformulated the previously used shuffle detection pretext task in a contrastive fashion and combined it with a standard visual contrastive objective. We validated that our method better captures temporal information compared to CVRL which led to improved performances on various action recognition benchmarks.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lucic, and Cordelia Schmid. Vivit: A video vision transformer. In *ICCV*, 2021. 2, 5

[2] Philip Bachman, R. Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. In *NeurIPS*, 2019. 2

[3] 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. 2

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In Marina Meila and Tong Zhang, editors, *ICML*, 2021. 2, 6

[5] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Bjorn Ommer. Behavior-driven synthesis of human dynamics. In *CVPR*, 2021. 2

[6] Andreas Blattmann, Timo Milbich, Michael Dorkenwald, and Björn Ommer. ipoke: Poking a still image for controlled stochastic video synthesis. In *ICCV*, 2021. 2

[7] Biagio Brattoli, Uta Buchler, Anna-Sophia Wahl, Martin E Schwab, and Bjorn Ommer. LSTM self-supervision for detailed behavior analysis. In *CVPR*, 2017. 1, 2, 8

[8] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *ECCV*, 2018. 1, 2

[9] 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. 1, 2

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

[11] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 2

[12] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 1, 2

[13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*, pages 4171–4186. Association for Computational Linguistics, 2019. 2

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

[15] Michael Dorkenwald, Timo Milbich, Andreas Blattmann, Robin Rombach, Konstantinos G. Derpanis, and Bjorn Ommer. Stochastic image-to-video synthesis using cinns. In *CVPR*, 2021. 2

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[17] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *ICLR*, 2018. 1, 2

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The ”something something” video database for learning and evaluating visual common sense. In *ICCV*, 2017. 2, 6

[19] 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. 1, 2

[20] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006. 1, 2

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020. 1, 2, 3, 5

[22] Kai Hu, Jie Shao, Yuan Liu, Bhiksha Raj, Marios Savvides, and Zhiqiang Shen. Contrast and order representations for video self-supervised learning. In *ICCV*, 2021. 1, 2, 7

[23] 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

[24] 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. 5

[25] Hildegard Kuehne, Hueihan Jhuang, Estibaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 1, 2, 6

[26] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. 1, 2

[27] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, 2017. 1

[28] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: towards action recognition without representation bias. In *ECCV*, 2018. 2, 6

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*. OpenReview.net, 2019. 5

[30] Timo Milbich, Karsten Roth, Homanga Bharadhwaj, Samarth Sinha, Yoshua Bengio, Björn Ommer, and Joseph Paul Cohen. Diva: Diverse visual feature aggregation for deep metric learning. In *ECCV*, 2020. 4

[31] Timo Milbich, Karsten Roth, Biagio Brattoli, and Björn Ommer. Sharing matters for generalization in deep metric learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2022. 4

[32] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *CVPR*, 2020. 1, 2

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

[34] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016. 1, 2

[35] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *ICCV*, 2017. 8

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

[37] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *CVPR*, 2016. 2

[38] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021. 1, 2, 3, 4, 5, 6, 7

[39] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *arXiv preprint arXiv:2108.08810*, 2021. 8

[40] Adrià Recasens, Pauline Luc, Jean-Baptiste Alayrac, Luyu Wang, Florian Strub, Corentin Tallec, Mateusz Malinowski, Viorica Patraucean, Florent Altché, Michal Valko, et al. Broaden your views for self-supervised video learning. In *ICCV*. 1, 2, 5

[41] Laura Sevilla-Lara, Shengxin Zha, Zhicheng Yan, Vedanuj Goswami, Matt Feiszli, and Lorenzo Torresani. Only time can tell: Discovering temporal data for temporal modeling. In *WACV*, pages 535–544. IEEE, 2021. 5, 6

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

[43] Jingkuan Song, Hanwang Zhang, Xiangpeng Li, Lianli Gao, Meng Wang, and Richang Hong. Self-supervised video hashing with hierarchical binary auto-encoder. *IEEE Trans. Image Process.*, 2018. 2

[44] Khurram Soomro, Amir Roshan Zamir, and M Shah. A dataset of 101 human action classes from videos in the wild. In *ICCV Workshops*, 2013. 1, 2, 6

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2

[46] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *ICML*, 2008. 2

[47] Carl Vondrick, Abhinav Shrivastava, Alireza Fathi, Sergio Guadarrama, and Kevin Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018. 1, 2

[48] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. In *ECCV*. 2

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

[50] Xiaolong Wang, Allan Jabri, and Alexei A Efros. Learning correspondence from the cycle-consistency of time. In *CVPR*, 2019. 1, 2

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

[52] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *CVPR*, 2018. 1, 2, 5

[53] Fanyi Xiao, Joseph Tighe, and Davide Modolo. Modist: Motion distillation for self-supervised video representation learning. *CoRR*, abs/2106.09703, 2021. 2, 4

[54] 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. 1, 2

[55] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. *CoRR*, abs/2103.15358, 2021. 2, 4, 5, 6, 7

[56] Jiaojiao Zhao, Xinyu Li, Chunhui Liu, Bing Shuai, Hao Chen, Cees G. M. Snoek, and Joseph Tighe. Tuber: Tubetransformer for action detection. *CoRR*, abs/2104.00969, 2021. 2