Self-supervised Video Representation Learning by Context and Motion Decoupling

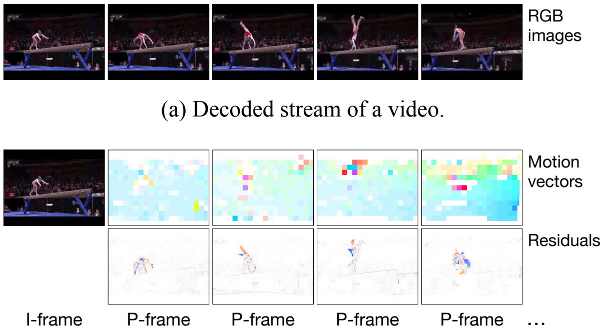
Lianghua Huang, Yu Liu, Bin Wang, Pan Pan, Yinghui Xu, Rong Jin
Machine Intelligence Technology Lab, Alibaba Group
xuangen.hlh,ly103369,ganfu.wb,panpan.pp,renji.xyh,jinrong.jr@alibaba-inc.com

Abstract

A key challenge in self-supervised video representation learning is how to effectively capture motion information besides context bias. While most existing works implicitly achieve this with video-specific pretext tasks (e.g., predicting clip orders, time arrows, and paces), we develop a method that explicitly decouples motion supervision from context bias through a carefully designed pretext task. Specifically, we take the key frames and motion vectors in compressed videos (e.g., in H.264 format) as the supervision sources for context and motion, respectively, which can be efficiently extracted at over 500 fps on CPU. Then we design two pretext tasks that are jointly optimized: a ***context matching*** *task where a pairwise contrastive loss is cast between video clip and key frame features; and a* ***motion prediction*** *task where clip features, passed through an encoder-decoder network, are used to estimate motion features in a near future. These two tasks use a shared video backbone and separate MLP heads. Experiments show that our approach improves the quality of the learned video representation over previous works, where we obtain absolute gains of* 16.0% *and* 11.1% *in video retrieval recall on UCF101 and HMDB51, respectively. Moreover, we find the motion prediction to be a strong regularization for video networks, where using it as an auxiliary task improves the accuracy of action recognition with a margin of* $7.4\% \sim 13.8\%$.

1. Introduction

Self-supervised representation learning from unlabeled videos has recently received considerable attention [18, 50]. Compared with static images, the redundancy, temporal consistency, and multi-modality of videos potentially provide richer sources of "supervision". Various methods have been proposed in this field that learn video representation by designing video-specific pretext tasks [33, 54, 49], adapting contrastive learning to videos [17, 50, 38, 18], cross-modal learning [23, 43, 36, 32], and contrastive clustering [2].



(a) Decoded stream of a video.

(b) Compressed stream of a video.

Figure 1: (a) and (b) show the decoded and compressed streams of a sample video, respectively. We notice that the **context** and **motion** information are roughly decoupled in I-frames and motion vectors of the compressed stream. We exploit these modalities as the supervision sources for self-supervised video representation learning.

This paper focuses on visual-only video representation learning, and we distinguish two orthogonal but complementary aspects of video representation: **context** and **motion**. Context depicts coarse-grained and relatively static environments, while motion represents dynamic and fine-grained movements or actions. Context information alone can be used to classify certain actions (*e.g.*, *swimming* is most likely to take place at *swimming pool*), but it also leads to background bias [40, 28]. For actions that heavily depend on movement patterns (*e.g.*, *breaststroke* and *frontcrawl*), motion information must be introduced. We aim to design a self-supervised video representation learning method that jointly learns these two complementary information. Our idea is to design a multi-task framework, where context and motion representation learning is decoupled in pretext tasks. One problem here is the source of supervision. Considering the scalability of our framework on larger datasets, we endeavor to avoid the use of computationally expensive features such as optical flow [59, 6] and dense trajectories [47].

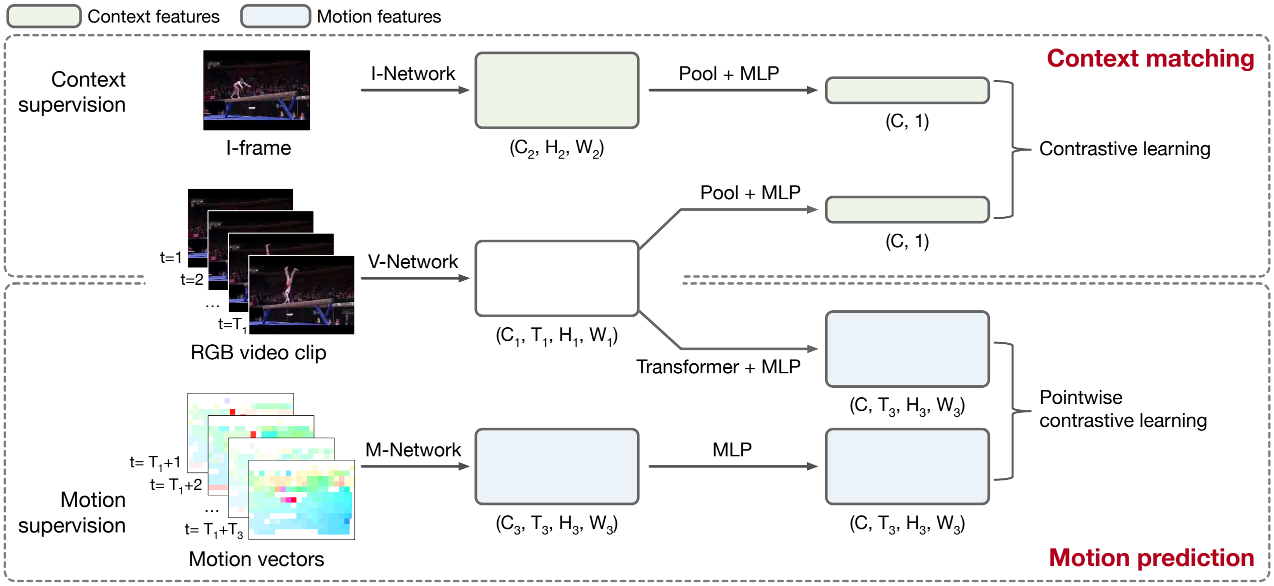


Figure 2: An overview of our framework. We decouple the context and motion supervision in two separate pretext tasks: context matching and motion prediction. The context matching task takes the relatively static I-frames in the compressed video as the source of supervision, and casts a contrastive loss between global features of I-frames and video clips. The motion prediction task takes the motion vector maps extracted from *future frames* of the compressed video as the source of supervision, and compares the predicted and "groundtruth" motion features in a pointwise way using the contrastive loss.

We notice that video in compressed format (such as H.264 and MPEG-4) roughly decouples the context and motion information in its I-frames and motion vectors. As shown in Figure 1, a compressed video stores only a few key frames (*i.e.*, I-frames) completely, and it reconstructs other frames based on motion vectors (*i.e.*, pixel offsets) and residual errors from the key frames. I-frames can represent relatively static and coarse-grained context information, while motion vectors depict dynamic and fine-grained movements. Moreover, both modalities can be efficiently extracted at over 500 fps on CPU [52].

Inspired by this, we present a self-supervised video representation learning method where two decoupled pretext tasks are jointly optimized: **context matching** and **motion prediction**. Figure 2 shows an overview of our framework. The *context matching* task aims to give the video network a rough grasp of the environment in which actions take place. It casts a noise contrastive estimation (NCE) loss [16, 35] between global features of video clips and I-frames, where clips and I-frames from the same videos are pulled together, while those from different videos are pushed away. The *motion prediction* task requires the model to predict pointwise motion dynamics *in a near future* based on visual information of the current clip. The assumption is that, in order to predict future motion, the video network needs to extract *low-level* movements from visual data and reorganize them into *high-level* trajectories. In this way, the learned representation should contain semantic and long-term motion information, helpful for downstream tasks. In our framework, instead of directly estimating the values of motion vectors, we use pointwise contrastive learning to compare predicted and real motion features at every spatial and temporal location (x, y, t). We find that this leads to more stable pretraining and better transferring performance.

We conduct extensive experiments on three network architectures, three datasets, and two downstream tasks (*i.e.*, action recognition and video retrieval) to assess the quality of our learned video representation. We achieve state-of-the-art performance on all these experiments. For example, we achieve R@1 video retrieval scores of 41.7% and 16.8% respectively on UCF101 [41] and HMDB51 [24] datasets, obtaining 16.0% and 11.1% absolute gains compared to existing works. We also validate several modeling options in our ablation studies, where we find that the *motion prediction* can serve as a strong regularization for video networks, and using it as an auxiliary task clearly improves the performance of supervised action recognition.

We summarize our contributions in the following:

- Unlike existing works where the source of supervision usually comes from the decoded raw video frames, we present a self-supervised video representation learning method that explicitly decouples the context and motion supervision in the pretext task.

- We present a *context matching* task for learning coarse-grained and relatively static context representation, and a *motion prediction* task for learning fine-grained and high-level motion representation.
- To the best of our knowledge, we present the first approach that exploits the modalities in compressed videos as the efficient supervision sources for visual representation learning.
- We achieve significant improvements over existing works on downstream tasks of action recognition and video retrieval. Extensive ablation studies also validate the effectiveness of our several modeling options.

2. Related Work

2.1. Self-supervised Video Representation Learning

There is an increasing interest in learning representation from unlabeled videos. Many works explore the intrinsic structure of videos and design video-specific pretext tasks, such as estimating video playback rates [5, 49, 57], verifying temporal orders of clips [33, 27, 54], predicting video rotations [21], solving space-time cubic puzzles [22], and dense predictive coding [17, 18].

Contrastive learning has recently shown great success in image representation learning [35, 19, 10], where self-supervised pretraining is approaching the performance of the supervised counterpart [12, 11]. More recently, contrastive learning has been introduced to video domain [42, 50, 38], where clips from the same video are pulled together while clips from different videos are pushed away. Another type of contrastive learning methods employ adaptive cluster assignment [8, 3, 2], where the representation and embedding clusters are simultaneously learned. However, since most methods apply contrastive learning on raw RGB clips without separating the motion information, the learned representation may suffer from the context bias problem. This work also follows the contrastive learning paradigm, but we explicitly decouple the motion supervision from the context bias in the pretext task.

Considering the multi-modality of videos, many works explore mutual supervision across modalities to improve the learned representation. For example, they regard the temporal or semantic consistency between videos and the corresponding audios [23, 2, 53, 37], narrations [32], or a combination of different modalities [36] as a natural source of supervision for representation learning.

A recent work named DSM [48] shares some similarities with our framework. It also tries to enhance the learned video representation by decoupling the scene and the motion. However, it achieves this by simply changing the construction of positive and negative pairs in contrastive learning, and it still learns on raw video clips; while our approach explicitly decouples the context and motion information in the source of supervision. Besides, the significantly better performance of our approach than DSM on downstream tasks also verifies the superiority of our work.

2.2. Action Recognition in Compressed Videos

Video compression techniques (*e.g.*, H.264 and MPEG-4) usually store only a few key frames completely, and reconstruct other frames using motion vectors and residual errors from the key frames. Taking advantage of this, many works propose to build video models directly on the compressed data to achieve faster inference and better performance [60, 61, 52, 39]. Pioneering works [60, 61] replace the optical flow stream in two-stream action recognition models [40] with a motion vector stream, thereby avoiding slow optical flow extraction. CoViAR [52] further utilizes all modalities (*i.e.*, I-frames, motion vectors, and residuals) in compressed video and bypasses the decoding of RGB frames for efficient video action recognition. DMC [39] improves the quality of the motion vector maps by jointly learning a generative adversarial network. More recently, Wang *et al.* [51] presents a method for fast object detection in compressed video, showing the potential of using compressed data for more fine-grained tasks. This work uses MPEG-2 Part2 [25] for video encoding, as practiced in CoViAR, where every I-frame is followed by 11 consecutive P-frames. We take the I-frames and motion vectors as proxies of the context and motion information for self-supervised video representation learning.

2.3. Motion Prediction

Motion prediction task usually refers to deduce the states (*e.g.*, position, orientation, speed, or posture) of an object in a near future [14, 55]. Example applications include human pose prediction [14, 31, 30] and traffic prediction [55, 13]. Typical models for solving this task include RNNs [26, 31], Transformers [1, 58], and graph neural networks [30]. In this work, we leverage a simple Transformer encoder-decoder network [46] for predicting future motion features based on current visual observations.

3. Methodology

3.1. Overall Framework

This work presents a self-supervised video representation learning method with two decoupled pretext tasks: a context matching task for learning coarse-grained context representation; and a motion prediction task for learning fine-grained motion representation. For efficiency, we use the I-frames and motion vectors in compressed video as the sources of supervision for context and motion tasks, respectively. Figure 2 shows an overview of our framework, where the V-Network, I-Network, and M-Network are used to extract video, context, and motion features, respectively.

In the context matching task, we compare global features of video clips and I-frames. An InfoNCE loss [35] is cast, where (*video clip, I-frame*) pairs from the same videos are pulled together, while pairs from different videos are pushed away. By matching still images to video clips, the learned representation is supposed to capture the global and coarse-grained contextual information of the video.

Unlike context, motion information is relatively sparse, localized and fine-grained. Therefore, we prefer to use a pointwise task to guide the V-Network to capture motion representation. A simple choice is to estimate motion vectors correspond to a video clip. However, this would lead the model to learn *low-level* offsets (*e.g.*, optical flow) instead of *high-level* trajectories or actions that are more preferred. In this work, we use *motion prediction* as the pretext task, where visual data of the current clip is used to predict motion information in a near future. The assumption is that, to predict motion dynamics in future frames, V-Network needs to extract *low-level* movements from the video and reorganize them into *high-level* trajectories. In this way, the learned representation can capture long-term motion information, which is beneficial for downstream tasks.

In our implementation, instead of directly predicting the motion vector values, we estimate the features of motion vector maps using an encoder-decoder network (*i.e.*, Transformer [46]), and compare the predicted and "groundtruth" motion features in a pointwise way. We find that leads to more stable pretraining and better transferring performance. The context matching and motion prediction tasks are jointly optimized in an end-to-end fashion. Next we will introduce these pretext tasks respectively in details.

3.2. Context Matching

The context of a video is relatively static in a period of time. It depicts a global environment in which the action takes place. We present a *context matching* task for the video model to capture such information, where the source of supervision comes from the I-frames extracted from the compressed video. A brief overview of the context matching process is shown in the upper half of Figure 2.

Specifically, we extract features $\mathbf{x}_i \in \mathbb{R}^{C_1 \times T_1 \times H_1 \times W_1}$ of a random clip in video i, and features $\mathbf{z}_i \in \mathbb{R}^{C_2 \times H_2 \times W_2}$ of a random I-frame surrounding the clip. Then we use global average pooling to obtain their global representation $\mathbf{x}'_i \in \mathbb{R}^{C_1}$ and $\mathbf{z}'_i \in \mathbb{R}^{C_2}$. Following the design improvements used in recent unsupervised frameworks [10, 12], we apply two-layer MLP heads g^V and g^I on the clip and I-frame features respectively to obtain $\mathbf{x}^*_i = g^V(\mathbf{x}'_i) \in \mathbb{R}^C$ and $\mathbf{z}^*_i = g^I(\mathbf{z}'_i) \in \mathbb{R}^C$. Then the InfoNCE loss is applied:

$$J_I = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp(\cos(\mathbf{z}^*_i, \mathbf{x}^*_i)/\tau)}{\sum_{k=1}^{B} \exp(\cos(\mathbf{z}^*_k, \mathbf{x}^*_i)/\tau)}, \qquad (1)$$

where B denotes the number of samples in the minibatch, $\cos(\mathbf{z}, \mathbf{x}) = (\mathbf{z}^{\mathrm{T}}\mathbf{x})/(\|\mathbf{z}\|_2 \cdot \|\mathbf{x}\|_2)$ denotes the cosine similarity between $\mathbf{z}$ and $\mathbf{x}$, and τ is a temperature adjusting the scale of cosine similarities. The loss function pulls video clips and I-frames from the same videos together, and pushes those from different videos far apart.

Period	Loss	Encoder-Decoder	UCF101	HMDB51
Current	InfoNCE	Transformer	29.6	10.4
Future	Cross-Ent.	Transformer	23.0	8.1
Future	MSE	Transformer	27.9	10.2
Future	InfoNCE	ConvGRU	39.8	15.2
Future	InfoNCE	Transformer	**41.7**	**16.8**

Table 1: Video retrieval performance (R@1) comparison when applying different settings in motion prediction. Words with gray background denote the setting changes with respect to our baseline. The Mean Square Error (*abbr.* MSE) and Cross Entropy (*abbr.* Cross-Ent.) losses are used when we directly predict motion vector values and value ranges after quantization, respectively.

3.3. Motion Prediction

Compared with contextual information, motion information is more fine-grained and position (in x, y, and t dimensions) sensitive. To encourage the video network to capture high-level and long-term motion information, we design a motion prediction task where visual data of current clip is used to predict motion dynamics in the near future. We use motion vectors extracted from the compressed video as the source of supervision. The lower half of Figure 2 shows a brief overview of the motion prediction process.

Specifically, we extract features $\mathbf{x}_i \in \mathbb{R}^{C_1 \times T_1 \times H_1 \times W_1}$ from a clip of video i, and features $\mathbf{v}_i \in \mathbb{R}^{C_3 \times T_3 \times H_3 \times W_3}$ from the motion vector maps in a near future after the clip. Subsequently, we feed $\mathbf{x}_i$ to an encoder-decoder network $\mathcal{T}$ (*i.e.*, Transformer [46] or ConvGRU [4, 18]) to predict the motion features $\hat{\mathbf{v}}_i = \mathcal{T}(\mathbf{x}_i) \in \mathbb{R}^{C_3 \times T_3 \times H_3 \times W_3}$. The "groundtruth" and predicted motion features are then flattened and fed into two-layer MLP heads g_1^M and g_2^M respectively to obtain projected embeddings $\mathbf{v}^*_i = g_1^M(\text{flatten}(\mathbf{v}_i)) \in \mathbb{R}^{C \times N}$ and $\hat{\mathbf{v}}^*_i = g_2^M(\text{flatten}(\hat{\mathbf{v}}_i)) \in \mathbb{R}^{C \times N}$, where $N = T_3 \cdot H_3 \cdot W_3$. The pointwise InfoNCE loss is then applied to $\mathbf{v}^*_i$ and $\hat{\mathbf{v}}^*_i$:

$$J_M = -\frac{1}{BN}\sum_{i,j} \log \frac{\exp(\cos(\mathbf{v}^*_{ij}, \hat{\mathbf{v}}^*_{ij})/\tau)}{\sum_{k,l} \exp(\cos(\mathbf{v}^*_{kl}, \hat{\mathbf{v}}^*_{ij})/\tau)}, \qquad (2)$$

where $i, k = 1 \sim B$, $j, l = 1 \sim N$, $\mathbf{v}^*_{ij} \in \mathbb{R}^C$ denotes the jth column of $\mathbf{v}^*_i$. In the loss function, only feature points corresponding to the same video i and at the same spatial and temporal position (x, y, t) are regarded as positive pairs, otherwise they are regarded as negative pairs.

The pointwise InfoNCE loss aims to lead the video network to learn fine-grained motion representation.

We test several modeling options and configurations of the motion prediction task, *e.g.*, whether to predict *current* or *future* motion information, whether to match features using the InfoNCE loss or to directly predict motion vector values, and the use of different encoder-decoder networks, *etc*. Comparison of these settings is shown in Table 1. To summarize, we find that: 1) Predicting *future* motion information leads to significantly better video retrieval performance compared with estimating *current* motion information; 2) Matching predicted and "groundtruth" motion features using the pointwise InfoNCE loss brings better results than directly estimating motion vector values; 3) Different encoder-decoder networks lead to similar results, while using Transformer performs slightly better.

In this work, we follow the optimal setting, where we predict future motion features using a Transformer network, and we employ the pointwise InfoNCE loss for training the model. When applying the Transformer network, we simply consider the input $\mathbf{x}_i \in \mathbb{R}^{C_1 \times T_1 \times H_1 \times W_1}$ as a 1-D sequence of length $T_1 \cdot H_1 \cdot W_1$, and the output $\hat{\mathbf{v}}_i \in \mathbb{R}^{C_3 \times T_3 \times H_3 \times W_3}$ as a 1-D sequence of length $T_3 \cdot H_3 \cdot W_3$.

3.4. Joint Optimization

We linearly combine the context matching loss and the motion prediction loss to obtain the final loss:

$$J = (1-\alpha)J_I + \alpha J_M, \tag{3}$$

where the α is a scalar hyper-parameter within $[0, 1]$. We simply set $\alpha = 0.5$ in our experiments, where the context and motion losses are equally weighted. The V-Network, I-Network, M-Network, Transformer $\mathcal{T}$, and all MLP heads (*i.e.*, g^V, g^I, g_1^M, and g_2^M) are jointly optimized with loss function (3) in an end-to-end fashion.

4. Experiments

4.1. Experiment Settings

Datasets. All experiments in this paper are conducted on three video classification datasets: UCF101 [41], HMDB51 [24], and Kinetics400 [9]. UCF101 consists of 13,320 videos belonging to 101 classes, while HMDB51 consists of 6,766 videos in 51 classes. Both datasets are divided into three train/test splits. We use their first split in all our experiments. Kinetics400 is a large-scale dataset containing 246K/20K videos in its train/val subsets. It populates 400 classes of human actions.

Networks. We evaluate the performance of our framework based on three video backbones (also the V-Network in Figure 2): C3D [44], R(2+1)D-26 [45], and R3D-26 [20, 45]. For the I-Network and M-Network, we use shallow R2D-10 and R3D-10 backbones, respectively, where each of them comprises 4 layers of ResNet BasicBlocks [20, 45]. The encoder-decoder network in our framework is implemented as a shallow Transformer, where 2 encoding layers and 4 decoding layers are used. Feature dimensions in the hidden layers of MLP heads are set to 2048, while dimensions in the hidden layers of the Transformer network and the output layers of MLP heads are set to 512.

Pretraining settings. Two datasets are used for pretraining: either UCF101 or Kinetics400. For UCF101, we pretrain our framework for 120 epochs on 4 GPUs, with a total batch size of 64. For Kinetics400, we pretrain our framework for 120 epochs on 32 GPUs, with a total batch size of 512. Following recent practices in large-batch training [15], we set the learning rate (lr) to scale up linearly with the batch size: $\text{lr} = 0.0005 \times B$. A cosine annealing rule is applied to decay the learning rate smoothly during training. We use SGD as the optimizer, where the weight decay and momentum are set to 0.005 and 0.9, respectively. Each video clip consists of 16 frames with a temporal stride of 4, and we predict motion dynamics in the next 8 consecutive frames. All clips are resized to $16 \times C_{\text{in}} \times 112 \times 112$, where $C_{\text{in}} = 3$ for video clips and $C_{\text{in}} = 2$ for motion vector maps. To introduce hard negatives for each video clip, we sample one positive clip and three negative clips of motion vector maps from the same video. We use random crop, random flip, Gaussian blur, and color jitter as the data augmentation for video clips and I-frames, and we use random crop and random flip as the data augmentation for motion vector maps. We ensure that the cropping and flipping parameters are consistent for positive (*video clip*, *motion vectors*) pairs.

Backbone	UCF101			HMDB51		
	Scratch	UCF	K400	Scratch	UCF	K400
C3D	60.5	78.6	**83.4**	29.2	46.9	**52.9**
R(2+1)D-26	65.0	79.7	**85.7**	32.5	48.6	**54.0**
R3D-26	58.0	76.6	**83.7**	28.9	47.2	**55.2**

Table 2: Action recognition performance (*i.e.*, top-1 accuracy) comparison between training from scratch and from our pretrained models. The three video backbones are unsupervised pretrained on either UCF101 (*abbr.* UCF) or Kinetics400 (*abbr.* K400) datasets.

Pretraining	Finetuning epoch					
	1	5	10	20	50	120
Scratch	3.2	17.5	22.4	28.9	42.7	65.0
UCF101	6.1	17.0	43.2	52.7	64.3	79.7
Kinetics400	**11.5**	**41.8**	**54.4**	**65.1**	**73.2**	**85.7**

Table 3: Convergence speed comparison between training from scratch and from pretrained models. The top-1 accuracy of the R(2+1)D-26 network on the UCF101 action recognition task is used as the indicator.

Modality	UCF101		HMDB51	
	R@1	R@5	R@1	R@5
I-frames	33.8	47.7	11.5	28.3
Motion vectors	30.4	50.9	14.0	37.5
I + Optical flow	**43.2**	**58.9**	**17.7**	**38.5**
I + Motion vectors	**41.7**	**57.4**	**16.8**	**37.2**

Table 4: Video retrieval performance comparison when pretraining C3D network on the UCF101 dataset using different modalities as the source of supervision. We also evaluate the results using optical flow here for a reference.

Network	UCF101		HMDB51	
	R@1	R@5	R@1	R@5
I-Network	**32.3**	**47.9**	**12.0**	27.8
M-Network	25.4	44.2	10.7	**32.1**
V-Network	41.7	57.4	16.8	37.2

Table 5: Video retrieval performance comparison when using the pretrained I-Network and M-Network for the retrieval. We list the results of V-Network as a reference. The pretraining experiment is conducted on the UCF101 dataset with C3D as the video backbone.

Finetuning settings. Pretrained models are finetuned on either UCF101 or HMDB51 to assess the transferring performance. For UCF101, we set the learning rate as $\text{lr} = 0.0001 \times B$ and the weight decay of the SGD optimizer as 0.003. For HMDB51, we set the learning rate as $\text{lr} = 0.0002 \times B$ and the weight decay as 0.002. For both datasets, we finetune the model for 120 epochs on 1 GPU with a batch size of 8. We use the cosine annealing scheduler to decay lr during training. A dropout of 0.3 is used before feeding backbone features to the classifier. We use random crop, random flip, Gaussian blur, and color jitter as the augmentation to improve the diversity of training data. As a common practice, we uniformly sample 10 clips in a video and average their scores for performance evaluation.

Video retrieval settings. Video retrieval experiments are conducted on either UCF101 or HMDB51 datasets, where videos in the test set are used to find videos in the training set. The *recall at top-k* (*abbr.* R@k) is used as the metric for evaluation – if one of the top-k searched videos has the same class label as the query video, a successful retrieval is count. Following the practice of [54, 18], we sample 10 video clips with a sliding window, and use the average of their global features as the representation of the video.

4.2. Ablation Study

This section validates several modeling and configuration options in our framework. The finetuning or video retrieval results on UCF101 and HMDB51 datasets are used as the performance indicators.

Method	UCF101		HMDB51	
	Top1	Top5	Top1	Top5
C3D	60.5	84.2	29.2	58.0
C3D with Reg.	**74.3**	**91.4**	**38.5**	**69.0**
R(2+1)D-26	65.0	85.9	32.5	66.1
R(2+1)D-26 with Reg.	**75.6**	**92.3**	**41.7**	**71.3**
R3D-26	58.0	83.1	28.9	60.4
R3D-26 with Reg.	**71.2**	**88.9**	**36.3**	**67.9**

Table 6: Action recognition performance comparison of three backbones trained with and without the auxiliary motion prediction task as regularization.

Task	Pretraining epoch					
	1	5	10	20	50	120
Context matching	25.8	42.5	48.0	50.0	53.7	**65.9**
Motion prediction	0.1	1.5	4.1	8.4	15.9	**33.1**
Video retrieval	6.0	6.3	8.5	9.5	14.8	**16.8**

Table 7: Correlation between pretraining and transferring tasks. The top-1 accuracy of the context matching and motion prediction tasks, as well as the R@1 score of the downstream video retrieval task are recorded during the pretraining process. The pretraining experiment is conducted on the UCF101 dataset with C3D as the video backbones, while the evaluation is performed on the HMDB51 dataset.

Ablation: scratch *vs.* pretraining. Table 2 compares the action recognition performance of three different backbones on UCF101 and HMDB51 datasets when training from scratch (*i.e.*, from randomly initialized parameters) and from our pretrained models. We observe that: 1) Even without introducing new training data, self-supervised pretraining followed by supervised finetuning on UCF101 leads to a remarkable $14.7\% \sim 18.6\%$ performance gain for all three backbones, suggesting that pretraining with context and motion decoupling significantly improves the quality of the learned representation; 2) Training on larger Kinetics400 ($\sim 25\times$ the scale of UCF101) further improves the accuracy with a margin of $4.8\% \sim 7.1\%$ on UCF101 and a margin of $5.4\% \sim 8.0\%$ on HMDB51, showing the scalability of our framework on larger-scale datasets. We also compare the convergence speed when training from scratch and from pretrained models on UCF101. Results are shown in Table 3. We observe that self-supervised pretraining clearly boosts the convergence, especially when pretrained on the larger Kinetics400 dataset, where the top-1 accuracy on epoch 5 surpasses that without pretraining by 24.3%.

Ablation: pretraining with different modalities. Table 4 compares the video retrieval performance when pretraining C3D network on UCF101 using different modalities as the source of supervision. As a reference, we also introduce the optical flow [59] as a substitute for the motion vector maps,

Method	Year	Pretrained	Resolution	Architecture	UCF101	HMDB51
Shuffle & Learn [33]	2016	UCF101	227 × 227	CaffeNet	50.2	18.1
OPN [27]	2017	UCF101	227 × 227	VGG-14	59.6	23.8
DPC [17]	2019	UCF101	128 × 128	R3D-18	60.6	-
VCOP [54]	2019	UCF101	112 × 112	R(2+1)D-26	72.4	30.9
PacePred [49]	2020	UCF101	112 × 112	R(2+1)D-18	**75.9**	35.9
VTDL [50]	2020	UCF101	112 × 112	C3D	73.2	**40.6**
PRP [57]	2020	UCF101	112 × 112	C3D	69.1	34.5
VCP [29]	2020	UCF101	112 × 112	C3D	68.5	32.5
DSM [48]	2020	UCF101	112 × 112	C3D	70.3	40.5
Ours		UCF101	112 × 112	**C3D**	**78.6**	**46.9**
Ours		UCF101	112 × 112	**R(2+1)D-26**	**79.7**	**48.6**
Ours		UCF101	112 × 112	**R3D-26**	**76.6**	**47.2**
3D-RotNet [21]	2018	Kinetics400	112 × 112	R3D-18	62.9	33.7
ST-Puzzle [22]	2019	Kinetics400	224 × 224	R3D-18	65.8	33.7
DPC [17]	2019	Kinetics400	128 × 128	R3D-18	68.2	34.5
DPC [17]	2019	Kinetics400	224 × 224	R3D-34	75.7	35.7
CBT [42]	2019	Kinetics600	112 × 112	S3D-G	79.5	44.6
SpeedNet [5]	2020	Kinetics400	224 × 224	S3D-G	81.1	48.8
MemoryDPC [18]	2020	Kinetics400	224 × 224	R2D3D-34	78.1	41.2
PacePred [49]	2020	Kinetics400	112 × 112	R(2+1)D-18	77.1	36.6
VTDL [50]	2020	Kinetics400	112 × 112	C3D	75.5	43.2
DSM [48]	2020	Kinetics400	224 × 224	I3D	74.8	52.5
DSM [48]	2020	Kinetics400	224 × 224	R3D-34	78.2	**52.8**
VTHCL [56]	2020	Kinetics400	-	R3D-50	**82.1**	49.2
Ours		Kinetics400	112 × 112	**C3D**	**83.4**	**52.9**
Ours		Kinetics400	112 × 112	**R(2+1)D-26**	**85.7**	**54.0**
Ours		Kinetics400	112 × 112	**R3D-26**	**83.7**	**55.2**

Table 8: Comparison with state-of-the-art self-supervised approaches on action recognition on UCF101 and HMDB51.

Method	Year	Pretrained	UCF101				HMDB			
			R@1	R@5	R@10	R@20	R@1	R@5	R@10	R@20
Jigsaw [34]	2016	UCF101	19.7	28.5	33.5	40.0	-	-	-	-
OPN [27]	2017	UCF101	19.9	28.7	34.0	40.6	-	-	-	-
Buchler [7]	2018	UCF101	**25.7**	36.2	42.2	49.2	-	-	-	-
VCOP [54]	2019	UCF101	14.1	30.3	40.4	51.1	7.6	22.9	34.4	48.8
VCP [29]	2020	UCF101	18.6	33.6	42.5	53.5	7.6	24.4	36.3	53.6
MemoryDPC [18]	2020	UCF101	20.2	**40.4**	**52.4**	**64.7**	**7.7**	**25.7**	**40.6**	**57.7**
SpeedNet [5]	2020	Kinetics400	13.0	28.1	37.5	49.5	-	-	-	-
Ours (C3D)		UCF101	**41.7**	**57.4**	**66.9**	**76.1**	**16.8**	**37.2**	50.0	64.3
Ours (R(2+1)D-26)		UCF101	38.4	55.4	65.2	74.6	14.3	36.0	48.5	64.2
Ours (R3D-26)		UCF101	38.9	56.1	65.8	75.6	15.2	36.0	**51.4**	**65.5**

Table 9: Comparison with state-of-the-art self-supervised approaches on video retrieval on UCF101 and HMDB51.

which is more accurate but also more computationally expensive to extract. We find that: 1) Context-only pretraining brings higher R@1 score on UCF101, while motion-only pretraining leads to better results on HMDB51. This is consistent with the observation that most classes in UCF101 can be distinguished using static context or pose cues, while classes in HMDB51 mainly differ in motion [24]; 2) Learning both context and motion information leads to much better results than learning any of them alone, suggesting the complementarity between the two modalities; 3) Replacing motion vectors with optical flow brings only slight performance gains, but motion vectors are several orders of magnitudes faster to extract.

Ablation: effectiveness of I-Network and M-Network. To assess if the co-trained I-Network and M-Network also learned effective representation, we evaluate the video retrieval performance on UCF101 and HMDB51 using the pretrained I-Network or M-Network. The pretraining experiment is conducted with C3D as the video backbone. Results are shown in Table 5. We observe that both networks can obtain reasonable video retrieval recalls. Interestingly, we find that I-Network performs much better than M-Network on UCF101 with a gain of 6.9%, and is comparable with M-Network on HMDB51 (with a gain of 1.3% in R@1 and a reduction of 4.3% in R@5). This may be because most classes in UCF101 can be classified by context information, while action classes in HMDB51 are mainly distinguished by the motion information [24].

Ablation: motion prediction as regularization. Spatial-temporal neural networks often learn context bias rather than motion dynamics [28], leading to worse results on action classes that heavily depend on motion patterns. We assess whether explicitly introducing an auxiliary motion prediction task improves the performance. Results are shown in Table 6. We train three video backbones from scratch on UCF101 and HMDB51, with or without introducing the motion prediction task. The weight of the pointwise InfoNCE loss (*i.e.*, Eq. (2)) is set to 0.2. We observe in Table 6 that introducing motion prediction as regularization consistently improves the action recognition performance, where the results on UCF101 are improved by $13.8\%, 10.6\%$, and 13.2% respectively for C3D, R(2+1)D-26, and R3D-26 backbones, and the results on HMDB51 are improved by $9.3\%, 9.2\%$, and 7.4% respectively.

Ablation: correlation between pretraining and transferring. To assess whether the pretraining and transferring tasks have high correlation, we record the top-1 accuracies of context matching, motion prediction, and the R@1 scores of video retrieval during the pretraining process. Results are shown in Table 7. In the experiment, we pretrain the C3D backbone using our method on UCF101, and evaluate the video retrieval recalls on HMDB51. We observe that, as the accuracies of the pretraining tasks increase, the transferring performance consistently improves. This validates the effectiveness of our pretext tasks in self-supervised video representation learning.

4.3. Comparison with State-of-the-art Approaches

Transfer learning. Table 8 compares our work with previous self-supervised video representation learning methods on UCF101 and HMDB51. The models are pretrained on either UCF101 or Kinetics400. To make the comparison as fair as possible, we test our framework with three different backbones (*i.e.*, C3D, R(2+1)D-26, and R3D-26), and we use a unified clip size of 112×112. As shown in the table, when pretrained on UCF101, our approach significantly outperforms state-of-the-art methods under all three backbones. Compared with previous best results, we achieve improvements of $2.7\%, 3.8\%$, and 0.7% respectively for the three backbones on UCF101, and we achieve improvements of $6.3\%, 8.0\%$, and 6.6% respectively on HMDB51. When pretrained on Kinetics400, we also observe absolute gains of $1.3\%, 3.6\%$, and 1.6% for the three backbones on UCF101, and gains of $0.1\%, 1.2\%$, and 2.4% on HMDB51 over previous methods. Besides, our models pretrained on Kinetics400 clearly outperforms those pretrained on UCF101, with nonnegligible improvements of $4.8\% \sim 7.1\%$ on UCF101 and $5.4\% \sim 8.0\%$ on HMDB51, indicating the scalability of our approach on larger datasets.

Video retrieval. Table 9 compares the video retrieval recalls of our work and state-of-the-art methods on UCF101 and HMDB51. All methods except SpeedNet [5] are pretrained on the UCF101 dataset, while SpeedNet is pretrained on the Kinetics400 dataset. Following [54, 18], we use videos in the test set as queries to search videos in the training set. The recall at top-k (R@k) as used as the performance indicator. As shown in the table, the results of our approach outperform previous methods by a very large margin. Specifically, on UCF101, our R@1 scores surpass state-of-the-art results by $16.0\%, 12.7\%$, and 13.2% when using C3D, R(2+1)D-26, and R3D-26 backbones, respectively; while on HMDB51, we outperform state-of-the-art results by $11.1\%, 6.7\%$, and 8.5% with the three backbones, respectively. The results validate the high quality of the video representation learned by our context-motion decoupled self-supervised learning framework.

5. Conclusion and Future Work

This paper presents a self-supervised video representation learning framework that explicitly decouples the context and motion supervision in the pretext task. We take the I-frames and motion vectors in compressed videos as the supervision sources for context and motion, respectively, which can be effectively extracted at more than 500 fps on CPU. We then present two pretext tasks that are jointly optimized: a *context matching* task that compares global features of I-frames and video clips under the contrastive learning framework; and a *motion prediction* task where current visual data in a video are used to predict the future motion information. The former aims to lead the video network to learn coarse-grained contextual representation, while the latter encourages the video network to capture fine-grained motion representation. Extensive experiments show that our decoupled learning framework achieves significantly better performance on downstream tasks of both action recognition and video retrieval. Various ablation studies also verify our several modeling and configuration options.

In the future, we would like to explore how the *residual errors* in compressed video can be used to improve the learned representation. Residual errors are supplementary information of motion vectors, which indicate the vanishing and emerging pixels and compensate for the estimation errors of motion vectors. This information may potentially improve the quality of the learned video representation.

References

[1] Emre Aksan, Peng Cao, Manuel Kaufmann, and Otmar Hilliges. Attention, please: A spatio-temporal transformer for 3d human motion prediction. *arXiv*, 2020. 3

[2] Humam Alwassel, Dhruv Mahajan, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. *arXiv*, 2019. 1, 3

[3] Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020. 3

[4] Nicolas Ballas, Li Yao, Chris Pal, and Aaron Courville. Delving deeper into convolutional networks for learning video representations. *arXiv*, 2015. 4

[5] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, pages 9922–9931, 2020. 3, 7, 8

[6] Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *European conference on computer vision*, pages 25–36. Springer, 2004. 1

[7] Uta Buchler, Biagio Brattoli, and Bjorn Ommer. Improving spatiotemporal self-supervision by deep reinforcement learning. In *Proceedings of the ECCV*, pages 770–786, 2018. 7

[8] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv*, 2020. 3

[9] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, pages 6299–6308, 2017. 5

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv*, 2020. 3, 4

[11] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv*, 2020. 3

[12] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv*, 2020. 3, 4

[13] Liangji Fang, Qinhong Jiang, Jianping Shi, and Bolei Zhou. Tpnet: Trajectory proposal network for motion prediction. In *CVPR*, pages 6797–6806, 2020. 3

[14] Katerina Fragkiadaki, Sergey Levine, Panna Felsen, and Jitendra Malik. Recurrent network models for human dynamics. In *ICCV*, pages 4346–4354, 2015. 3

[15] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv*, 2017. 5

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

[17] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *Workshop on Large Scale Holistic Video Understanding, ICCV*, 2019. 1, 3, 7

[18] Tengda Han, Weidi Xie, and Andrew Zisserman. Memory-augmented dense predictive coding for video representation learning. *arXiv*, 2020. 1, 3, 4, 6, 7, 8

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

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 5

[21] Longlong Jing, Xiaodong Yang, Jingen Liu, and Yingli Tian. Self-supervised spatiotemporal feature learning via video rotation prediction. *arXiv*, 2018. 3, 7

[22] Dahun Kim, Donghyeon Cho, and In So Kweon. Self-supervised video representation learning with space-time cubic puzzles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8545–8552, 2019. 3, 7

[23] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *Advances in Neural Information Processing Systems*, pages 7763–7774, 2018. 1, 3

[24] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, pages 2556–2563. IEEE, 2011. 2, 5, 7, 8

[25] Didier Le Gall. Mpeg: A video compression standard for multimedia applications. *Communications of the ACM*, 34(4):46–58, 1991. 3

[26] Donghan Lee, Youngwook Paul Kwon, Sara McMains, and J Karl Hedrick. Convolution neural network-based lane change intention prediction of surrounding vehicles for acc. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017. 3

[27] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *ICCV*, pages 667–676, 2017. 3, 7

[28] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proceedings of the ECCV*, pages 513–528, 2018. 1, 8

[29] Dezhao Luo, Chang Liu, Yu Zhou, Dongbao Yang, Can Ma, Qixiang Ye, and Weiping Wang. Video cloze procedure for self-supervised spatio-temporal learning. *arXiv*, 2020. 7

[30] Wei Mao, Miaomiao Liu, Mathieu Salzmann, and Hongdong Li. Learning trajectory dependencies for human motion prediction. In *ICCV*, pages 9489–9497, 2019. 3

[31] Julieta Martinez, Michael J Black, and Javier Romero. On human motion prediction using recurrent neural networks. In *CVPR*, pages 2891–2900, 2017. 3

[32] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, pages 9879–9889, 2020. 1, 3

[33] Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *European Conference on Computer Vision*, pages 527–544. Springer, 2016. 1, 3, 7

[34] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016. 7

[35] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv*, 2018. 2, 3, 4

[36] Mandela Patrick, Yuki M Asano, Ruth Fong, João F Henriques, Geoffrey Zweig, and Andrea Vedaldi. Multimodal self-supervision from generalized data transformations. *arXiv*, 2020. 1, 3

[37] AJ Piergiovanni, Anelia Angelova, and Michael S Ryoo. Evolving losses for unsupervised video representation learning. In *CVPR*, pages 133–142, 2020. 3

[38] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. *arXiv*, 2020. 1, 3

[39] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *CVPR*, pages 1268–1277, 2019. 3

[40] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1, 3

[41] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv*, 2012. 2, 5

[42] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv*, 2019. 3, 7

[43] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, pages 7464–7473, 2019. 1

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

[45] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018. 5

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 3, 4

[47] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Action Recognition by Dense Trajectories. In *IEEE Conference on Computer Vision & Pattern Recognition*, pages 3169–3176, Colorado Springs, United States, June 2011. 1

[48] Jinpeng Wang, Yuting Gao, Ke Li, Xinyang Jiang, Xiaowei Guo, Rongrong Ji, and Xing Sun. Enhancing unsupervised video representation learning by decoupling the scene and the motion. *arXiv*, 2020. 3, 7

[49] Jiangliu Wang, Jianbo Jiao, and Yun-Hui Liu. Self-supervised video representation learning by pace prediction. *arXiv*, 2020. 1, 3, 7

[50] Jinpeng Wang, Yiqi Lin, Andy J Ma, and Pong C Yuen. Self-supervised temporal discriminative learning for video representation learning. *arXiv*, 2020. 1, 3, 7

[51] Shiyao Wang, Hongchao Lu, and Zhidong Deng. Fast object detection in compressed video. In *ICCV*, pages 7104–7113, 2019. 3

[52] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *CVPR*, pages 6026–6035, 2018. 2, 3

[53] Fanyi Xiao, Yong Jae Lee, Kristen Grauman, Jitendra Malik, and Christoph Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv*, 2020. 3

[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*, pages 10334–10343, 2019. 1, 3, 6, 7, 8

[55] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *CVPR*, pages 7593–7602, 2018. 3

[56] Ceyuan Yang, Yinghao Xu, Bo Dai, and Bolei Zhou. Video representation learning with visual tempo consistency. *arXiv*, 2020. 7

[57] Yuan Yao, Chang Liu, Dezhao Luo, Yu Zhou, and Qixiang Ye. Video playback rate perception for self-supervised spatio-temporal representation learning. In *CVPR*, pages 6548–6557, 2020. 3, 7

[58] Cunjun Yu, Xiao Ma, Jiawei Ren, Haiyu Zhao, and Shuai Yi. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. *arXiv*, 2020. 3

[59] Christopher Zach, Thomas Pock, and Horst Bischof. A duality based approach for realtime tv-l 1 optical flow. In *Joint pattern recognition symposium*, pages 214–223. Springer, 2007. 1, 7

[60] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *CVPR*, pages 2718–2726, 2016. 3

[61] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with deeply transferred motion vector cnns. *IEEE Transactions on Image Processing*, 27(5):2326–2339, 2018. 3