RSPNet: Relative Speed Perception for Unsupervised Video Representation Learning

Peihao Chen,[1,5*†] **Deng Huang,**[1†] **Dongliang He,**[2] **Xiang Long,**[2] **Runhao Zeng,**[1]
Shilei Wen,[2] **Mingkui Tan,**[1,4‡] **Chuang Gan**[3]

[1]School of Software Engineering, South China University of Technology, [2]Baidu Inc., [3]MIT-IBM Watson AI Lab
[4]Key Laboratory of Big Data and Intelligent Robot, Ministry of Education, [5]Pazhou Laboratory
{phchencs, im.huangdeng, runhaozeng.cs, ganchuang1990}@gmail.com,
{hedongliang01, longxiang, wenshilei}@baidu.com, mingkuitan@scut.edu.cn

Abstract

We study unsupervised video representation learning that seeks to learn both motion and appearance features from **unlabeled video** only, which can be reused for downstream tasks such as action recognition. This task, however, is extremely challenging due to 1) the highly complex spatial-temporal information in videos and 2) the lack of labeled data for training. Unlike representation learning for static images, it is difficult to construct a suitable self-supervised task to effectively model both motion and appearance features. More recently, several attempts have been made to learn video representation through *video playback speed prediction*. However, it is non-trivial to obtain precise speed labels for the videos. More critically, the learned models may tend to focus on motion patterns and thus may not learn appearance features well. In this paper, we observe that the relative playback speed is more consistent with motion patterns and thus provides more effective and stable supervision for representation learning. Therefore, we propose a new way to perceive the playback speed and exploit the **relative speed** between two video clips as labels. In this way, we are able to effectively perceive speed and learn better motion features. Moreover, to ensure the learning of appearance features, we further propose an **appearance-focused** task, where we enforce the model to perceive the appearance difference between two video clips. We show that jointly optimizing the two tasks consistently improves the performance on two downstream tasks (namely, *action recognition* and *video retrieval*) *w.r.t* the increasing pre-training epochs. Remarkably, for action recognition on the UCF101 dataset, we achieve 93.7% accuracy without the use of labeled data for pre-training, which outperforms the ImageNet supervised pre-trained model. Our code, pre-trained models, and supplementary materials can be found at https://github.com/PeihaoChen/RSPNet.

1 Introduction

Video analysis has been a prominent research topic in computer vision due to its vast potential applications, including action recognition (Wu et al. 2021; Long et al. 2018), event detection (Gan et al. 2015), action localization (Chen et al. 2020a; Zeng et al. 2019, 2020), audio-visual scene analysis (Gan et al. 2019; Chen et al. 2020b; Gan et al. 2020), video question answering (Huang et al. 2020), *etc*. Compared with static images, videos often contain more complex spatial-temporal content and have a larger data volume, making them very challenging to annotate and analyze. How to learn effective video representations with a few annotations or even without annotations represents an important yet challenging task (Gan et al. 2016a, 2018; Fan et al. 2018).

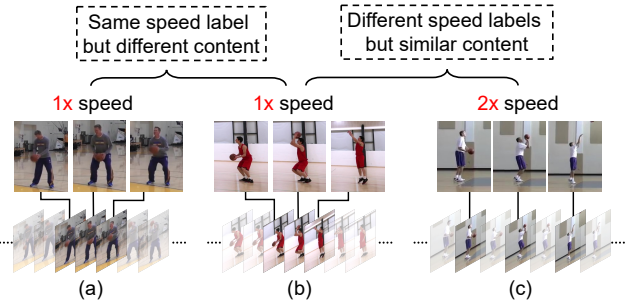

Figure 1: An illustrative example of content-label inconsistency. In existing speed perception-based methods (Benaim et al. 2020), 1) both video clips (a) and (b) are labeled as 1x speed, *i.e.*, sampled consecutively, but the contents of these two clips are dissimilar. The left player shoots the ball more slowly, while the middle player has finished shooting within the same time period. 2) Although clip (c) is labeled as 2x speed, *i.e.*, the sampling interval is set to 2 frames, it appears similar to the middle clip with different speed labels.

Recently, unsupervised video representation learning, which seeks to learn appearance and motion features from unlabeled videos, has attracted great attention (Cho et al. 2020; Benaim et al. 2020; Epstein, Chen, and Vondrick 2020; Gan et al. 2016b). This task, however, is very difficult due to several challenges: 1) The downstream video understanding tasks, such as action recognition, rely on both appearance features (*e.g.*, texture and shape of objects, background scene) and motion features (*e.g.*, the movement of

*This work was done when Peihao Chen was a research intern at Baidu.
†Equal contribution.
‡Corresponding author.

objects). It is difficult to learn representation for both appearance and motion simultaneously because of the complex spatial-temporal information in videos. 2) It is difficult to mine effective supervision from unlabeled video data for representation learning.

Existing methods attempt to solve these challenges by designing pretext tasks to obtain pseudo labels for video representation learning. The pretext tasks include context prediction (Han, Xie, and Zisserman 2019), playback speed perception (Benaim et al. 2020), temporal clip order prediction (Xu et al. 2019), *etc*. In particular, training models using the playback speed perception task achieves great success because models must focus on the moving objects to perceive the playback speed (Wang, Jiao, and Liu 2020). This focus helps models to learn representative motion features. Specifically, Benaim et al. (2020) train a model to determine whether videos are sped up or not. Epstein, Chen, and Vondrick (2020); Yao et al. (2020); Wang, Jiao, and Liu (2020) attempt to predict the specific playback speed for each video.

However, these works suffer from two limitations. **First**, the playback speed labels used for the pretext task can be imprecise due to inconsistency with the motion content in videos. As shown in Figure 1, the clips with different labels (*i.e.*, different playback speeds) may appear similar to each other. The underlying reason for this is that different people often implement the same action at different speeds. Using such inconsistent speed labels for training may make it difficult to learn discriminative features. **Second**, perceiving speed mainly relies on the motion content. The models are not explicitly encouraged to explore appearance features, which, however, are also important for video understanding. Recently, the instance discrimination task (Wu et al. 2018; He et al. 2020) has shown its effectiveness for learning appearance features in the image domain. However, how to extend it to the video domain and effectively combine it with motion features learning is non-trivial.

To address the imprecise label issue in the above methods, we observe that the relative playback speed can provide more precise supervision for training. To this end, we propose a new pretext task that exploits relative playback speed as labels for perceiving speed, namely **relative speed perception** (RSP). Specifically, we sample two clips from the same video and train a neural network to identify their relative playback speed instead of predicting the specific playback speed of each video clip. The relative playback speed label is obtained through the comparison between playback speeds of two clips from the same video (*e.g.*, 2x is faster than 1x). We observe that for the same video, the higher the playback speed is, the faster the objects will move. Consequently, such labels are independent of the original speed of objects in a video and can reveal the precise motion distinction between two clips. In this sense, the labels are more consistent with the motion content and can provide more effective supervision for representation learning.

Moreover, to encourage models to pay attention to learning appearance features, we follow the spirit of the instance discrimination task in image domain and design an **appearance-focused video instance discrimination** (A-VID) task. In this task, we require the model to find two clips sampled from the same video from numerous clips from other videos. Considering that different clips in the same video are often filmed at the same speed, we propose a speed augmentation strategy, *i.e.*, randomizing the playback speed of each clip. Consequently, models cannot finish this task by simply learning speed information. Instead, models tend to learn appearance features, such as background scene and the texture of objects, because these features are consistent throughout a video but vary among different videos. We train models to finish RSP and A-VID tasks jointly using a two-branch architecture such that models are expected to learn both motion and appearance features simultaneously. We name our model **RSPNet**. Experimental results show that the learned features perform well on two downstream tasks, *i.e.*, action recognition and video retrieval.

To summarize, our contributions are as follows:

- We propose a relative speed perception task for unsupervised video representation learning. In this task, the labels are consistent with the motion content and provide more effective supervision for representation learning.
- We extend the instance discrimination task to the video domain and propose a speed augmentation strategy to make it focus more on exploring the appearance content. In this way, we can effectively combine it with the relative speed perception task to learn representation for both motion and appearance contents simultaneously.
- We verify the effectiveness of RSP and A-VID tasks for learning video representation on two downstream tasks and three datasets. Remarkably, without the need of annotation for pre-training, the action recognition accuracy on UCF-101 significantly outperforms the models supervised pre-trained on ImageNet (93.7% *vs.* 86.6%).

2 Related Work

Unsupervised video representation learning. In recent years, unsupervised video representation learning, which uses video itself as supervision, has become a popular topic (Jing and Tian 2020). The existing methods learn representation through various carefully designed pretext tasks. Xu et al. (2019) proposed the video clip order prediction task to leverage the temporal order of image sequences. Luo et al. (2020) proposed the video cloze procedure task by predicting the spatio-temporal operation applied on the video clips. Instead of focusing on the RGB domain, Ng et al. (2018) proposed a multitask learning model trained by estimating optical flow to learn motion representation. Since the video contains multiple frames, predicting future frames in latent space (van den Oord, Li, and Vinyals 2018) is also an effective task to learn visual representation.

More recently, many works have been proposed to learn features by discriminating playback speeds. Epstein, Chen, and Vondrick (2020); Cho et al. (2020) try to predict whether a clip is sped up or not. Wang, Jiao, and Liu (2020); Yao et al. (2020); Jenni, Meishvili, and Favaro (2020) attempt to predict the specific playback speed of one clip. However, these works suffer from the imprecise speed label issue. Cho et al. (2020) design a method to sort video clips according to their

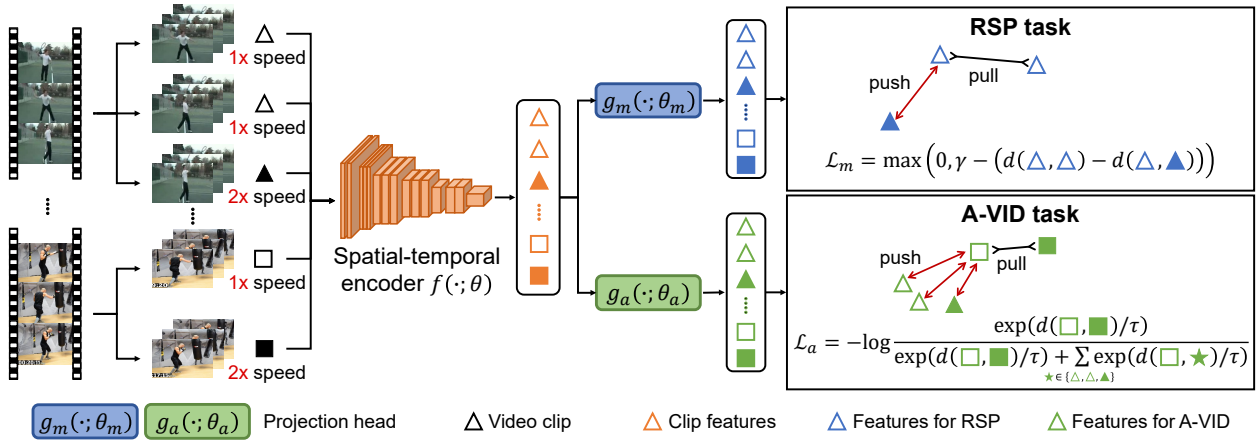

Figure 2: Illustration of the proposed self-supervised video representation learning scheme. Given a set of video clips with different playback speeds, we use a spatial-temporal encoder $f(\cdot;\theta)$ followed by two projection heads (*i.e.*, g_m and g_a) to extract clip features for two pretext tasks. In the relative speed perception (RSP) task, we identify the relative playback speed between clips instead of predicting their specific playback speeds. In the appearance-focused video instance discrimination (A-VID) task, we distinguish video clips relying on the appearance contents. We formulate two pretext tasks as a metric learning problem and use triplet loss $\mathcal{L}_m$ and InfoNCE loss $\mathcal{L}_a$ for training.

playback speeds. However, they do not explicitly encourage the model to learn appearance features. Our method uses relative speed to resolve the imprecise label issue. Moreover, we extend instance discrimination task (Wu et al. 2018) to the video domain to encourage appearance learning.

Metric learning. Metric learning (Xing et al. 2002) aims to automatically construct task-specific distance metrics that compare two samples from a specific aspect. Based on this metric, the similar pairs of samples are pulled together and the dissimilar pairs of samples are pushed apart. This method has achieved great success in many areas, *e.g.*, face recognition (Schroff, Kalenichenko, and Philbin 2015), music recommendation (McFee, Barrington, and Lanckriet 2012), and person reidentification (Yang, Wang, and Tao 2018). Recently, many works have successfully adopted metric learning for self-supervised representation learning (Wu et al. 2018; He et al. 2020; Tian, Krishnan, and Isola 2019). They usually generate positive pairs by creating multiple views of each data point and generate negative pairs by randomly choosing images/patches/videos. In this work, we aim to learn video representation by comparing two video clips using metric learning. Unlike the existing works, we propose to identify their speed distinction and appearance distinction to learn motion and appearance features from unlabeled data.

3 Proposed Method

Problem definition. Let $\mathcal{V} = \{v_i\}_{i=1}^N$ be a video set containing N videos. We sample a clip $\mathbf{c}_i$ from a video with s_i playback speed. Unsupervised video representation learning aims to learn a spatial-temporal encoder $f(\cdot;\theta)$ to map video clip $\mathbf{c}_i$ to features $\mathbf{x}_i$ that best describe the content in $\mathbf{c}_i$.

This task is challenging because of the complex spatial-temporal information in videos and the lack of annotations. It is difficult to construct supervision information from unlabeled videos $\mathcal{V}$ to train a model to learn the representation for both appearance and motion contents. Recently, some existing unsupervised learning methods have attempted to learn video representation through playback speed perception. However, most of these methods suffer from the imprecise speed label issue and do not explicitly encourage models to learn appearance features. Consequently, the learned features may not be suitable for downstream video understanding tasks such as action recognition and video retrieval.

3.1 General Scheme of RSPNet

In this paper, we observe that relative playback speed can provide more effective labels for representation learning. Thus, we propose a relative speed perception task, *i.e.*, predicting whether two clips have the same speed or not, to resolve imprecise label issues and learn motion features. Moreover, we extend the instance discrimination task to the video domain and propose a speed augmentation strategy to explicitly make models pay attention to exploring appearance features. Considering the success of metric learning in representation learning (Hadsell, Chopra, and LeCun 2006), we formulate these two tasks as metric learning, in which we seek to maximize the similarity of two clip features in positive pairs while minimizing that in negative pairs.

Formally, for the **relative speed perception** task, instead of directly predicting playback speed s_i for clip $\mathbf{c}_i$, we propose to compare the speeds of two clips $\mathbf{c}_i$ and $\mathbf{c}_j$ that are sampled from the same video. Since the actions in $\mathbf{c}_i$ (or $\mathbf{c}_j$) are often implemented by the same subject, the motions in these two clips are similar when $s_i = s_j$ and are dissimilar when $s_i \neq s_j$. In this sense, the relative speed labels are obtained through comparing s_i and s_j (*i.e.*, clips $\mathbf{c}_i$ and $\mathbf{c}_j$ are

labeled as a positive pair when $s_i = s_j$ and are negative otherwise). Such labels are more consistent with motion content in videos and reveal the precise motion distinction. For the **appearance-focused video instance discrimination** task, we enforce the model to predict whether two clips $\mathbf{c}_i$ and $\mathbf{c}_l$ are sampled from the same video. The intuition is that clips sampled from the same video often share similar appearance content, which can be used as an important clue for distinguishing videos. We also randomize the playback speed; *i.e.*, s_i can be equal or not equal to s_l. In this way, models are encouraged to pay more attention to learning appearance features instead of finishing this task by learning playback speed information.

We use two individual projection heads $g_m(\cdot;\theta_m)$ and $g_a(\cdot;\theta_a)$ to map spatial-temporal features $\mathbf{c}_i$ to $\mathbf{m}_i$ and $\mathbf{a}_i$ for two tasks, respectively. We train models on these two tasks jointly. The objective function is formulated as follows:

$$\mathcal{L}(\mathcal{V};\theta,\theta_a,\theta_m) = \mathcal{L}_m(\mathcal{V};\theta,\theta_m) + \lambda\mathcal{L}_a(\mathcal{V};\theta,\theta_a), \quad (1)$$

where $\mathcal{L}_m$ and $\mathcal{L}_a$ denote the loss functions of each task, respectively, and λ is a fixed hyperparameter to control the relative importance of each term. During inference for downstream tasks, we forward a video clip through the spatiotemporal encoder $f(\cdot;\theta)$ and obtain $\mathbf{x}_i$ as its spatiotemporal features. The schematic of our approach is shown in Figure 2. In the following, we will introduce more details about two pretext tasks in Section 3.2.

3.2 RSP and A-VID Tasks

Relative speed perception. This task aims to maximize the similarity of two clips with the same playback speed and minimize the similarity of two clips with different playback speeds. Given a video, we sample 3 clips $\mathbf{c}_i$, $\mathbf{c}_j$ and $\mathbf{c}_k$ with playback speeds s_i, s_j and s_k, respectively, where $s_i = s_j \neq s_k$. We feed each clip into the spatial-temporal encoder $f(\cdot;\theta)$ followed by a projection head $g_m(\cdot;\theta_m)$ to obtain their corresponding features $\mathbf{m}_i$, $\mathbf{m}_j$, $\mathbf{m}_k$. The dot product function $d(\cdot,\cdot)$ is used to measure the similarity between two clips. As the clips with the same playback speed share similar motion features, we expect that their features can be closer than the clips with different playback speeds. We achieve this object by using a triplet loss (Schroff, Kalenichenko, and Philbin 2015) as follows:

$$\mathcal{L}_m(\mathcal{V};\theta,\theta_m) = \max(0, \gamma - (p^+ - p^-)), \quad (2)$$

where $p^+ = d(\mathbf{m}_i, \mathbf{m}_j)$, $p^- = d(\mathbf{m}_i, \mathbf{m}_k)$ and $\gamma > 0$ is a certain margin. We desire that the similarity of a positive pair is larger than a negative pair by a margin γ.

Appearance-focused video instance discrimination. To explicitly encourage models to learn appearance features, we propose an A-VID task to further regularize the learning process. Motivated by the fact that different clips from the same video always exhibit similar spatial information, we extend the contrastive learning in the image domain (Wu et al. 2018) to the video domain. Specifically, we sample two clips $\mathbf{c}_i$ and $\mathbf{c}_j$ from the same randomly selected video v^+ and K clips $\{\mathbf{c}_n\}_{n=1}^K$ from K videos in subset $\mathcal{V} \setminus v^+$. Next, we feed each clip into the spatial-temporal encoder $f(\cdot;\theta)$ followed by a projection head $g_a(\cdot;\theta_a)$ and obtain their corresponding features. The encoder $f(\cdot;\theta)$ shares weights with the encoder in the RSP task, while the weights of projection head $g_a(\cdot;\theta_a)$ are independent of $g_m(\cdot;\theta_m)$. We consider ($\mathbf{c}_i$, $\mathbf{c}_j$) to be a positive pair and ($\mathbf{c}_i$, $\mathbf{c}_n$) to be a negative pair. We further apply the InfoNCE loss (He et al. 2020) as the training loss:

$$\mathcal{L}_a(\mathcal{V};\theta,\theta_a) = -\log\frac{q^+}{q^+ + \sum_{n=1}^K q_n^-}, \quad (3)$$

where $q^+ = \exp(d(\mathbf{a}_i, \mathbf{a}_j)/\tau)$, $q_n^- = \exp(d(\mathbf{a}_i, \mathbf{a}_n)/\tau)$, and τ is a temperature hyperparameter (Wu et al. 2018) which affects the concentration level of distribution. The reason we use InfoNCE loss is that it pushes away multiple negative samples at the same time, which is more efficient and stable for training (Sohn 2016). We do not use InfoNCE loss for RSP because for two same video clips with different playback speeds, A-VID tends to pull them together while RSP tends to push them away. Using a triple loss in Equation (2) to enforce their similarity larger than a margin is enough.

An underlying question is how to sample these video clips. A naive solution is to sample all clips at the same playback speed. In this sense, clips $\mathbf{c}_i$ and $\mathbf{c}_j$ will share similar motion features while the motion features in $\mathbf{c}_i$ and $\mathbf{c}_n$ are dissimilar. This approach may provide clues for models to determine whether any two clips are from the same video or not. To encourage models to pay more attention to learning appearance features, we propose a speed augmentation strategy. Concretely, we randomize the playback speed of each clip, *i.e.*, randomly selecting s_i, s_j, and s_n from possible playback speeds, such that the motion features cannot provide effective clues for this task. In this way, models have to focus on learning other informative features, including background and object appearance for discriminating video instances. The training method is shown in Algorithm 1.

Algorithm 1 Training method of RSPNet

Require: video set $\mathcal{V} = \{v_i\}_{i=1}^N$, # negative pair for A-VID K.
1: Initialize parameters $\theta, \theta_a, \theta_m$ for $f(\cdot;\theta), g_a(\cdot;\theta_a), g_m(\cdot;\theta_m)$, respectively
2: **while** no converge **do**
3: Randomly sample a video v^+ from $\mathcal{V}$, extract clips c_i, c_j, c_k from v^+ with speed s_i, s_j, s_k, where $s_i = s_j \neq s_k$.
4: Sample K clips $\{c_n\}_{n=1}^K$ from video set $\mathcal{V} \setminus \{v^+\}$.
5: Extract features $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k$, and $\{\mathbf{x}_n\}_{n=1}^K$ from video clips c_i, c_j, c_k, $\{c_n\}_{n=1}^K$ using encoder $f(\cdot;\theta)$.
6: *// RSP task*
7: Obtain features $\mathbf{m}_i, \mathbf{m}_j, \mathbf{m}_k$ from $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ using $g_m(\cdot;\theta_m)$.
8: Compute $\mathcal{L}_m$ using Equation (2).
9: *// A-VID task*
10: Obtain features $\mathbf{a}_i, \mathbf{a}_j, \{\mathbf{a}_n\}_{n=1}^K$ from $\mathbf{x}_i, \mathbf{x}_j, \{\mathbf{x}_n\}_{n=1}^K$ using $g_a(\cdot;\theta_a)$
11: Compute $\mathcal{L}_a$ and $\mathcal{L}$ using Equations (3) and (1), respectively.
12: Update parameters $\theta, \theta_a, \theta_m$ via stochastic gradient descent.
13: **end while**

Pre-training settings	UCF101			HMDB51		
	TSM-18	3DResNet-18	C3D	TSM-18	3DResNet-18	C3D
w/o pre-training	49.7	42.3	59.0	17.5	19.0	24.9
w/ RSP only	54.5	49.7	67.2	26.5	25.9	29.4
w/ A-VID only	60.8	57.2	68.1	30.2	31.1	35.1
SP + A-VID	59.8	57.8	70.9	29.7	30.7	35.1
RSP + VID	57.5	54.2	70.8	30.1	29.9	34.5
RSP + A-VID (Ours)	**61.2**	**60.2**	**71.5**	**32.2**	**32.6**	**36.3**

Table 1: Comparison of different pre-training settings on UCF101 and HMDB51 datasets. All models are pre-trained on the Kinetics-100 dataset except for the w/o pre-training setting. SP denotes speed prediction for each individual clip. VID denotes video instance discrimination without speed augmentation strategy.

4 Experiments

Datasets. We pre-train models on the training set of the Kinetics-400 dataset (Carreira and Zisserman 2017), which consists of approximately 240K training videos with 400 human action classes. Each video lasts approximately 10 seconds. To reduce training costs in ablation studies, we build a lightweight dataset, namely, Kinetics-100, by selecting 100 classes with the least disk size of videos from Kinetics-400. The UCF101 (Soomro, Zamir, and Shah 2012) dataset consists of 13,320 videos from 101 realistic action categories on YouTube. The HMDB51 (Kuehne et al. 2011) dataset consists of 6,849 clips from 51 action classes. Compared with UCF101 and HMDB51, the Something-Something-V2 (*Something-V2*) dataset (Goyal et al. 2017) contains 220,847 videos with 174 classes and focuses more on modeling temporal relationships (Lin, Gan, and Han 2019).

Pre-training details. We instantiate the projection head as a fully connected layer with 128 output dimensions. After pre-training, we use the features before the projection heads for downstream tasks. Unless otherwise stated, we sample 16 consecutive frames with 112×112 spatial size for each clip following Kim, Cho, and Kweon (2019). Clips are augmented by using random cropping with resizing, random color jitter and random Gaussian blur (Chen et al. 2020c). We use SGD as the optimizer with a minibatch size of 64. We train the model for 200 epochs by default. The learning rate policy is linear cosine decay starting from 0.1. Following He et al. (2020), we set $\tau = 0.07$, $K = 16384$, $\gamma = 0.15$ and $\lambda = 1$ for Equations (1), (2) and (3). All videos are played at 25 fps. The possible playback speed s for clips in this paper is set to 1x (*i.e.*, sampling frames consecutively) and 2x (*i.e.*, sampling interval is set 2 frames).

Fine-tuning details. We fine-tune our RSPNet on UCF101, HMDB51, and *Something-V2* with labeled videos for action recognition. We train for 30, 70 and 50 epochs on these datasets, respectively, with a learning rate of 0.01. Following (Xu et al. 2019), we initialize the models with the weights from the pre-trained RSPNet except for the newly appended fully connected layer with randomly initialized weights. Unless otherwise stated, the size of input video clips is the same as pre-training.

4.1 Ablation Studies

Effectiveness of two pretext tasks. In this paper, we propose two tasks, namely, RSP and A-VID, to learn video representation. To verify the effectiveness of each task, we pre-train models using either RSP or A-VID on three backbone networks.

From Table 1, compared with training from scratch, using the RSP or A-VID task for pre-training significantly improves the action recognition performance on the UCF101 and HMDB51 datasets, which demonstrates that models learn useful clues for action recognition through pre-training on our designed pretext task. The improvement brought about by the A-VID task is relatively larger than that of relative speed discrimination. The underlying reason is that the UCF101 and HMDB51 datasets focus more on modeling appearance information than temporal relationships (Lin, Gan, and Han 2019). The models pre-trained on A-VID are more sensitive to object appearance and background scene, while models pre-trained on RSP are more sensitive to the movement of objects. When we jointly pre-trained models on both tasks, we achieved the best results for all three models. Compared with the w/o pre-training setting, we achieve a relative improvement of 11.5%, 17.9%, and 12.5% on UCF101 and 14.7%, 13.6%, and 11.4% on HMDB51 in top-1 accuracy. These results demonstrate that the two pretext tasks are complementary to each other and are effective for learning video representation.

Does relative speed perception help? As discussed in Section 1, we train models to perceive the relative speed of two clips to resolve the imprecise speed label issue. Here, we implement a variant of our method by replacing RSP with directly predicting the speed of each clip (*i.e.*, 1x or 2x speed). We formulated this as a classification problem and use a cross-entropy loss function to optimize it following Wang, Jiao, and Liu (2020). We denote this task as speed prediction (SP). Table 1 shows that exploiting relative speed as labels consistently improves the performance on three backbone networks and on two datasets compared with directly using the playback speed of each clip (SP + A-VID *vs.* RSP + A-VID). These results demonstrate that relative speed labels are more consistent with the motion content and help models to learn more discriminative video features.

Method	Architecture	Pre-train Dataset	Frozen	UCF101	HMDB51
CBT (Sun et al. 2019)	S3D	Kinetics-600	Y	54.0	29.5
MemDPC (Han, Xie, and Zisserman 2020a)	3DResNet-34	Kinetics-400	Y	54.1	30.5
RSPNet (Ours)	3DResNet-18	Kinetics-400	Y	**61.78**	**42.81**
Fully supervised	S3D-G	ImageNet	N	86.6	57.7
	S3D-G	Kinetics-400	N	96.8	75.9
CMC (Tian, Krishnan, and Isola 2019)	CaffeNet	UCF101	N	59.1	26.7
VCP (Luo et al. 2020)	C3D	UCF101	N	68.5	32.5
PSP (Cho et al. 2020)	R(2+1)D	UCF101	N	74.8	36.8
ClipOrder (Xu et al. 2019)	R(2+1)D	UCF101	N	72.4	30.9
PRP (Yao et al. 2020)	R(2+1)D	UCF101	N	72.1	35.0
MAS (Wang et al. 2019)	C3D	Kinetics-400	N	61.2	33.4
RTT (Jenni, Meishvili, and Favaro 2020)	C3D	Kinetics-400	N	69.9	39.6
3D ST-Puzzle (Kim, Cho, and Kweon 2019)	3DResNet-18	Kinetics-400	N	65.8	33.7
3DRotNet (Jing et al. 2018)	3DResNet-18	Kinetics-400	N	66.0	37.1
DPC (Han, Xie, and Zisserman 2019)	3DResNet-18	Kinetics-400	N	68.2	34.5
MemDPC (Han, Xie, and Zisserman 2020a)	3DResNet-34	Kinetics-400	N	78.7	41.2
Pace (Wang, Jiao, and Liu 2020)	R(2+1)D	Kinetics-400	N	77.1	36.6
CBT (Sun et al. 2019)	S3D	Kinetics-600	N	79.5	44.6
CoCLR (Han, Xie, and Zisserman 2020b)	S3D	Kinetics-400	N	87.9	54.6
SpeedNet (Benaim et al. 2020)	S3D-G	Kinetics-400	N	81.1	48.8
RSPNet (Ours)	C3D	Kinetics-400	N	76.7	44.6
	3DResNet-18	Kinetics-400	N	74.3	41.8
	R(2+1)D	Kinetics-400	N	81.1	44.6
	S3D-G	Kinetics-400	N	89.9	59.6
	S3D-G	Kinetics-400	N	**93.7***	**64.7***

Table 2: Comparison with other unsupervised methods on UCF101 and HMDB51 datasets. We show the backbone architecture and the pre-training dataset of each method. *We pre-train the model for 1000 epochs.

Does speed augmentation help? Instead of naively extending the instance discrimination task from the image domain to video domain, we propose to randomize the speed of each clip. To verify its effectiveness, we implement a variant by dropping speed augmentation. We denote it as VID, as it is not appearance-focused. Table 1 shows that the speed augmentation strategy significantly improves the performance (RSP + VID *vs.* RSP + A-VID). The reason is that the speed augmentation strategy makes the VID task become speed-agnostic. In this way, models are encouraged to pay more attention to learning appearance features. Together with the motion features learned from the RSP task, models can extract more discriminative representation for appearance and motion, which are both important for action recognition.

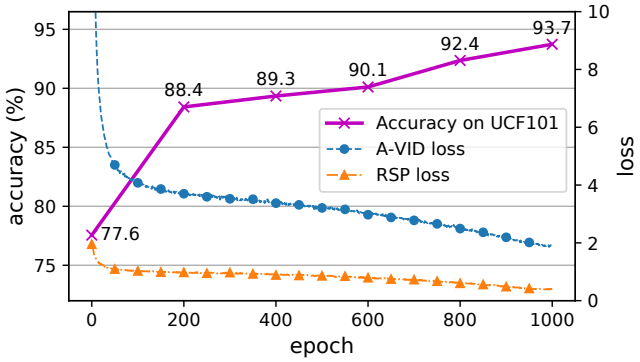

Figure 3: Pre-training losses of two pretext tasks and top-1 accuracy of UCF101 after fine-tuning. We pre-train S3D-G model on K-400 for 1000 epochs and report the results every 200 epochs.

4.2 Evaluation on Action Recognition Task

Performance on UCF101 and HMDB51. We compare our method with the state-of-the-art self-supervised learning methods in Table 2. We report top-1 accuracy on the UCF101 and HMDB51 datasets together with the backbone and pre-training dataset. For fair comparison, we evaluate two settings: 1) freeze the backbone and only train a classifier, *i.e.*, linear probe (denoted as Frozen=Y) and 2) fine-tuning the entire network (denoted as Frozen=N).

Our RSPNet achieves the best results on all backbone networks over the two datasets under two settings. Specifically, for the linear probe setting, even we pre-train RSPNet on a smaller dataset or use a smaller network, we outperform the previous methods on two datasets. For the fine-tuning setting, with C3D, our method outperforms RTT (76.7% *vs.* 69.9% on UCF101 and 44.6% *vs.* 39.6% on HMDB51). With 3DResNet-18, our method outperforms DPC by 6.1% and 7.3% in absolute improvement on the two datasets, respectively. With R(2+1)D, our RSPNet improves the accuracy from 77.1% to 81.1% on UCF101 and from 36.6% to 44.6% on HMDB51. For S3D-G, we follow SpeedNet (Benaim et al. 2020) to use 16 and 64 consecutive frames with size of 224 $\times$ 224 as input for pre-training and fine-tuning, respectively. Under the same settings, our RSPNet increases the accuracy from 81.1% to 89.9% on UCF101 and from 48.8% to 59.6% on HMDB51.

	3DResNet-18	C3D	S3D-G
w/o pre-training	42.1	45.8	51.2
Fully supervised	43.7	47.0	**56.8**
Unsupervised (Ours)	**44.0**	**47.8**	55.0

Table 3: Performance comparison on *Something-V2*.

Method	Architecture	Top-k		
		$k=1$	$k=10$	$k=50$
OPN	OPN	19.9	34.0	51.6
Buchler *et al.*	CaffeNet	25.7	42.2	59.5
ClipOrder	R3D	14.1	40.0	66.5
SpeedNet	S3D-G	13.0	37.5	65.0
VCP	R(2+1)D	19.9	42.0	64.4
Pace	C3D	31.9	59.2	80.2
RSPNet (Ours)	C3D	36.0	66.5	87.7
	3DResNet-18	**41.1**	**68.4**	**88.7**

Table 4: Video retrieval results on UCF101.

When we train longer (*i.e.*, 1000 epochs), we can further improve the top-1 accuracy to 93.7% and 64.7% on the UCF101 and HMDB51 datasets, respectively. In Figure 3, we show the curve of pre-training losses and the performance on UCF101 for the S3D-G model using different checkpoints. As the losses decrease, the performance for downstream tasks increases consistently. This observation demonstrates the effectiveness of the proposed RSP and A-VID tasks. The model does learn semantic representation to solve the tasks instead of learning trivial solutions. Remarkably, without the need for any annotation for pre-training, our RSPNet outperforms the ImageNet supervised pre-trained variant (93.7% *vs.* 86.6%, respectively) and achieves performance close to that of the Kinetics supervised pre-trained model (96.8%).

Performance on *Something-V2*. We compare our RSPNet with supervised learning methods on *Something-V2*, a challenging dataset in which temporal information is essential (Lin, Gan, and Han 2019). Note that RSPNet is unsupervised pre-trained on Kinetics-400 without any manual annotation. In Table 3, despite not using annotations, RSPNet consistently increases the accuracy compared with the random initialized models on three backbone architectures. Surprisingly, RSPNet even outperforms the supervised pre-trained model on 3DResNet-18 and C3D, increasing from 43.7% to 44.0% and from 47.0% to 47.8%, respectively. It shows the benefits of the discriminative features learned from the proposed two pretext tasks. More implementation details can be found in supplementary materials.

4.3 Evaluation on Video Retrieval Task

Given a query video with feature representation being $\mathbf{x}_i$, we use the nearest neighbor search to retrieve relevant videos based on the cosine similarity (Xu et al. 2019). We evaluate our method on split 1 of the UCF101 dataset and apply the top-k accuracies (k=1, 10, 50) as evaluation metrics.

From Table 4, our method outperforms state-of-the-art approaches by a large margin under different values of k.

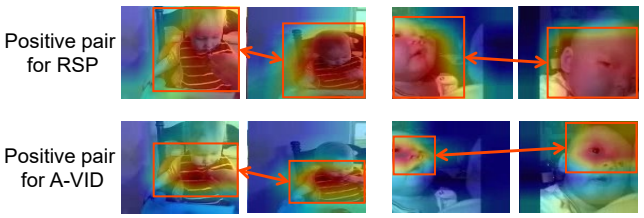

Figure 4: Visualization of RoI learned for RSP and A-VID. Our model focuses on the regions containing rich motion and appearance information for two pretext tasks, respectively. We outline the area where the heatmap is higher than a threshold with a rectangle.

For example, our method achieves much better performance than Pace (Wang, Jiao, and Liu 2020) under all values of k using the same C3D backbone. With 3DResNet-18 as backbone network, we can achieve better retrieval performance. This result implies that the proposed pretext tasks help us to learn more discriminative features for video retrieval tasks. More details can be found in supplementary materials.

4.4 RoI Visualization

From Section 3.1, we formulate two pretext tasks as metric learning, which seeks to maximize the similarity of the positive pair. To better understand the clues learned for the two pretext tasks, we visualize the region of interest (RoI) that contributes most to the similarity score using the class-activation map (CAM) technique (Zhou et al. 2016). More details are shown in the supplementary materials.

In Figure 4, we show the heatmaps of each clip in two positive pairs. We use the middle frame to represent a clip to visualize the heatmap. For the RSP task, the heatmaps tend to cover the whole region of actions, which provides rich information for perceiving the relative speed. For the A-VID task, models tend to focus on small but discriminative regions (*e.g.*, the striped clothes and the eyes of a baby in two pair sample) to identify two clips in the same video. One interesting finding is that the models are able to adaptively localize the same object even though they appear in different locations of a frame. This approach may provide a new perspective for person reidentification, which we leave for future work.

5 Conclusion

In this paper, we have proposed an unsupervised video representation learning framework named RSPNet. We train models to perceive relative playback speed for learning motion features by using relative speed labels to resolve the imprecise speed label issue. Additionally, we extend the instance discrimination task to the video domain and propose a speed augmentation strategy to make models focus on learning appearance features. Extensive experiments show that the features learned by RSPNet perform better in action recognition and video retrieval downstream tasks. Visualization of the RoI implies that RSPNet can focus on the discriminative area for two tasks.

Acknowledgments

This work was partially supported by National Natural Science Foundation of China (NSFC) 62072190, 61836003 (key project), Program for Guangdong Introducing Innovative and Entrepreneurial Teams 2017ZT07X183, International Cooperation Open Project of State Key Laboratory of Subtropical Building Science, South China University of Technology (2019ZA01), Fundamental Research Funds for the Central Universities D2191240, CCF-Baidu Open Fund.

References

Benaim, S.; Ephrat, A.; Lang, O.; Mosseri, I.; Freeman, W. T.; Rubinstein, M.; Irani, M.; and Dekel, T. 2020. SpeedNet: Learning the Speediness in Videos. In *CVPR*.

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

Chen, P.; Gan, C.; Shen, G.; Huang, W.; Zeng, R.; and Tan, M. 2020a. Relation Attention for Temporal Action Localization. *IEEE Transactions on Multimedia* 22: 2723–2733.

Chen, P.; Zhang, Y.; Tan, M.; Xiao, H.; Huang, D.; and Gan, C. 2020b. Generating Visually Aligned Sound From Videos. *IEEE Transactions on Image Processing* 29: 8292–8302.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. E. 2020c. A Simple Framework for Contrastive Learning of Visual Representations .

Cho, H.; Kim, T.; Chang, H. J.; and Hwang, W. 2020. Self-Supervised Spatio-Temporal Representation Learning Using Variable Playback Speed Prediction. *arXiv* abs/2003.02692.

Epstein, D.; Chen, B.; and Vondrick, C. 2020. Oops! Predicting Unintentional Action in Video. In *CVPR*.

Fan, L.; Huang, W.; Gan, C.; Ermon, S.; Gong, B.; and Huang, J. 2018. End-to-End Learning of Motion Representation for Video Understanding. In *CVPR*.

Gan, C.; Gong, B.; Liu, K.; Su, H.; and Guibas, L. J. 2018. Geometry Guided Convolutional Neural Networks for Self-Supervised Video Representation Learning. In *CVPR*.

Gan, C.; Huang, D.; Chen, P.; Tenenbaum, J. B.; and Torralba, A. 2020. Foley Music : Learning to Generate Music from Videos. In *ECCV*.

Gan, C.; Sun, C.; Duan, L.; and Gong, B. 2016a. Webly-Supervised Video Recognition by Mutually Voting for Relevant Web Images and Web Video Frames. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *ECCV*.

Gan, C.; Wang, N.; Yang, Y.; Yeung, D.; and Hauptmann, A. G. 2015. DevNet: A Deep Event Network for multimedia event detection and evidence recounting. In *CVPR*.

Gan, C.; Yao, T.; Yang, K.; Yang, Y.; and Mei, T. 2016b. You Lead, We Exceed: Labor-Free Video Concept Learning by Jointly Exploiting Web Videos and Images. In *CVPR*.

Gan, C.; Zhao, H.; Chen, P.; Cox, D. D.; and Torralba, A. 2019. Self-Supervised Moving Vehicle Tracking With Stereo Sound. In *ICCV*.

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

Hadsell, R.; Chopra, S.; and LeCun, Y. 2006. Dimensionality Reduction by Learning an Invariant Mapping. In *CVPR*.

Han, T.; Xie, W.; and Zisserman, A. 2019. Video Representation Learning by Dense Predictive Coding. In *ICCVW*.

Han, T.; Xie, W.; and Zisserman, A. 2020a. Memory-augmented Dense Predictive Coding for Video Representation Learning. In *ECCV*.

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

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

Huang, D.; Chen, P.; Zeng, R.; Du, Q.; Tan, M.; and Gan, C. 2020. Location-Aware Graph Convolutional Networks for Video Question Answering. In *AAAI*.

Jenni, S.; Meishvili, G.; and Favaro, P. 2020. Video Representation Learning by Recognizing Temporal Transformations. In *ECCV*.

Jing, L.; and Tian, Y. 2020. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* .

Jing, L.; Yang, X.; Liu, J.; and Tian, Y. 2018. Self-Supervised Spatiotemporal Feature Learning via Video Rotation Prediction. *arXiv* abs/1811.11387.

Kim, D.; Cho, D.; and Kweon, I. S. 2019. Self-Supervised Video Representation Learning with Space-Time Cubic Puzzles. In *AAAI*.

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

Lin, J.; Gan, C.; and Han, S. 2019. TSM: Temporal Shift Module for Efficient Video Understanding. In *ICCV*.

Long, X.; Gan, C.; de Melo, G.; Wu, J.; Liu, X.; and Wen, S. 2018. Attention Clusters: Purely Attention Based Local Feature Integration for Video Classification. In *CVPR*.

Luo, D.; Liu, C.; Zhou, Y.; Yang, D.; Ma, C.; Ye, Q.; and Wang, W. 2020. Video Cloze Procedure for Self-Supervised Spatio-Temporal Learning. In *AAAI*.

McFee, B.; Barrington, L.; and Lanckriet, G. R. G. 2012. Learning Content Similarity for Music Recommendation. *IEEE Transactions on Speech and Audio Processing* 20: 2207–2218.

Ng, J. Y.; Choi, J.; Neumann, J.; and Davis, L. S. 2018. ActionFlowNet: Learning Motion Representation for Action Recognition. In *WACV*.

Schroff, F.; Kalenichenko, D.; and Philbin, J. 2015. FaceNet: A unified embedding for face recognition and clustering. In *CVPR*.

Sohn, K. 2016. Improved Deep Metric Learning with Multi-class N-pair Loss Objective. In Lee, D. D.; Sugiyama, M.; von Luxburg, U.; Guyon, I.; and Garnett, R., eds., *NeurIPS*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *arXiv* abs/1212.0402.

Sun, C.; Baradel, F.; Murphy, K.; and Schmid, C. 2019. Learning Video Representations using Contrastive Bidirectional Transformer. *arXiv* abs/1906.05743.

Tian, Y.; Krishnan, D.; and Isola, P. 2019. Contrastive Multiview Coding. *arXiv* abs/1906.05849.

van den Oord, A.; Li, Y.; and Vinyals, O. 2018. Representation Learning with Contrastive Predictive Coding. *arXiv* abs/1807.03748.

Wang, J.; Jiao, J.; Bao, L.; He, S.; Liu, Y.; and Liu, W. 2019. Self-Supervised Spatio-Temporal Representation Learning for Videos by Predicting Motion and Appearance Statistics. In *CVPR*.

Wang, J.; Jiao, J.; and Liu, Y. 2020. Self-supervised Video Representation Learning by Pace Prediction. *arXiv* abs/2008.05861.

Wu, W.; He, D.; Lin, T.; Li, F.; Gan, C.; and Ding, E. 2021. MVFNet: Multi-View Fusion Network for Efficient Video Recognition. In *AAAI*.

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

Xing, E. P.; Ng, A. Y.; Jordan, M. I.; and Russell, S. J. 2002. Distance Metric Learning with Application to Clustering with Side-Information. In *NeurIPS*.

Xu, D.; Xiao, J.; Zhao, Z.; Shao, J.; Xie, D.; and Zhuang, Y. 2019. Self-Supervised Spatiotemporal Learning via Video Clip Order Prediction. In *CVPR*.

Yang, X.; Wang, M.; and Tao, D. 2018. Person Re-Identification With Metric Learning Using Privileged Information. *IEEE Transactions on Image Processing* 27: 791–805.

Yao, Y.; Liu, C.; Luo, D.; Zhou, Y.; and Ye, Q. 2020. Video Playback Rate Perception for Self-Supervised Spatio-Temporal Representation Learning. In *CVPR*.

Zeng, R.; Gan, C.; Chen, P.; Huang, W.; Wu, Q.; and Tan, M. 2019. Breaking Winner-Takes-All : Iterative-Winners-Out Networks for Weakly Supervised Temporal Action Localization. *IEEE Transactions on Image Processing* 28: 5797–5808.

Zeng, R.; Xu, H.; Huang, W.; Chen, P.; Tan, M.; and Gan, C. 2020. Dense Regression Network for Video Grounding. In *CVPR*.

Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2016. Learning Deep Features for Discriminative Localization. In *CVPR*.