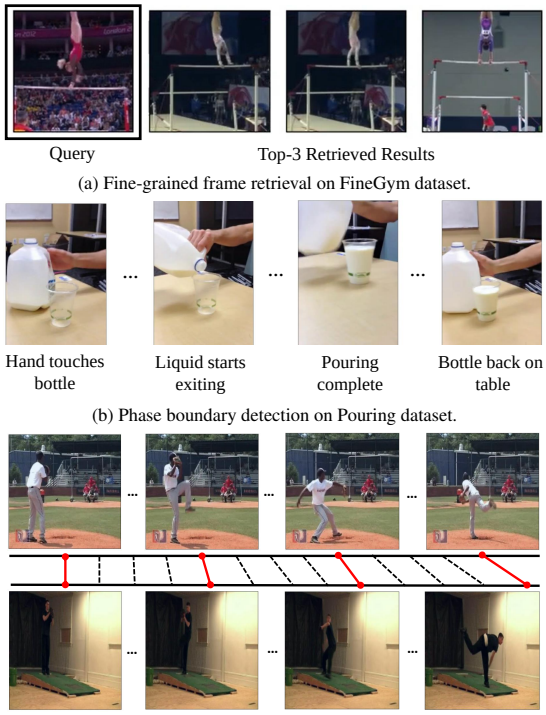
Frame-wise Action Representations for Long Videos via Sequence Contrastive Learning

Minghao Chen[1*] Fangyun Wei[2†] Chong Li[2] Deng Cai[1]
[1]State Key Lab of CAD&CG, College of Computer Science, Zhejiang University
[2]Microsoft Research Asia
minghaochen01@gmail.com {fawe, chol}@microsoft.com dengcai@cad.zju.edu.cn

Abstract

Prior works on action representation learning mainly focus on designing various architectures to extract the global representations for short video clips. In contrast, many practical applications such as video alignment have strong demand for learning dense representations for long videos. In this paper, we introduce a novel contrastive action representation learning (CARL) framework to learn frame-wise action representations, especially for long videos, in a self-supervised manner. Concretely, we introduce a simple yet efficient video encoder that considers spatio-temporal context to extract frame-wise representations. Inspired by the recent progress of self-supervised learning, we present a novel sequence contrastive loss (SCL) applied on two correlated views obtained through a series of spatio-temporal data augmentations. SCL optimizes the embedding space by minimizing the KL-divergence between the sequence similarity of two augmented views and a prior Gaussian distribution of timestamp distance. Experiments on FineGym, PennAction and Pouring datasets show that our method outperforms previous state-of-the-art by a large margin for downstream fine-grained action classification. Surprisingly, although without training on paired videos, our approach also shows outstanding performance on video alignment and fine-grained frame retrieval tasks. Code and models are available at https://github.com/minghchen/CARL_code.



(a) Fine-grained frame retrieval on FineGym dataset.

(b) Phase boundary detection on Pouring dataset.

(c) Temporal video alignment on PennAction dataset.

Figure 1. Multiple applications of our frame-wise representation learning on various datasets: (a) Fine-grained frame retrieval on FineGym [37]. (b) Phase boundary detection on Pouring [36]. (c) Temporal video alignment on PennAction [49]. As shown in the figures, the representations obtained through our method (CARL) are invariant to the appearance, viewpoint and background.

1. Introduction

In the last few years, deep learning for video understanding [1, 9, 17, 33, 39, 41, 44, 47] has achieved great success on video classification task [9, 19, 40]. Networks such as I3D [9] and SlowFast [17] always take short video clips (e.g., 32 frames or 64 frames) as input and extract global representations to predict the action category. In contrast, many practical applications, e.g., sign language translation [4, 5, 13], robotic imitation learning [29, 36], action alignment [6, 21, 23] and phase classification [16, 27, 37, 49] require algorithms having ability to model long videos with hundreds of frames and extract frame-wise representations rather than the global features (Fig. 1).

*Accomplished during Minghao Chen's internship at MSRA.
†Corresponding author.

Previous methods [27, 35, 37] have made an effort to learn frame-wise representations via supervised learning, where sub-actions or phase boundaries are annotated. However, it is time-consuming and even impractical to manually label each frame and exact action boundaries [21] on large-scale datasets, which hinders the generalization of models trained with fully supervised learning in realistic scenarios. To reduce the dependency of labeled data, some methods such as TCC [16], LAV [23] and GTA [21] explored weakly-supervised learning by using either cycle-consistency loss [16] or soft dynamic time warping [21, 23]. All these methods rely on video-level annotations and the training is conducted on the paired videos describing the same action. This setting obstructs them from applying on more generic video datasets where no labels are available.

The goal of this work is to learn frame-wise representations with spatio-temporal context information for long videos in a self-supervised manner. Inspired by the recent progress of contrastive representation learning [8, 11, 12, 20], we present a novel framework named contrastive action representation learning (CARL) to achieve our goal. We assume no labels are available during training, and videos in both training and testing sets have long durations (hundreds of frames). Moreover, we do not rely on video pairs of the same action for training. Thus it is practical to scale up our training set with less cost.

Modeling long videos with hundreds of frames is challenging. It is non-trivial to directly use off-the-shelf backbones designed for short video clip classification, since our task is to extract frame-wise representations for long videos. In our work, we present a simple yet efficient video encoder that consists of a 2D network to encode spatial information per frame and a Transformer [42] encoder to model temporal interaction. The frame-wise features are then used for representation learning.

Recently, SimCLR [11] uses instance discrimination [46] as the pretext task and introduces a contrastive loss named NT-Xent, which maximizes the agreement between two augmented views of the same data. In their implementation, all instances other than the positive reference are considered as negatives. Unlike image data, videos provide more abundant instances (each frame is regarded as an instance), and the neighboring frames have high semantic similarities. Directly regarding these frames as negatives may hurt the learning. To avoid this issue, we present a novel sequence contrastive loss (SCL), which optimizes the embedding space by minimizing the KL-divergence between the sequence similarity of two augmented video views and a prior Gaussian distribution.

The main contributions of this paper are summarized as follows:

- We propose a novel framework named contrastive action representation learning (CARL) to learn frame-wise action representations with spatio-temporal context information for long videos in a self-supervised manner. Our method does not rely on any data annotations and has no assumptions on datasets.
- We introduce a Transformer-based network to efficiently encode long videos and a novel sequence contrastive loss (SCL) for representation learning. Meanwhile, a series of spatio-temporal data augmentations are designed to increase the variety of training data.
- Our framework outperforms the state-of-the-art methods by a large margin on multiple tasks across different datasets. For example, under the linear evaluation protocol on FineGym [37] dataset, our framework achieves 41.75% accuracy, which is +13.94% higher than the existing best method GTA [21]. On Penn-Action [49] dataset, our method achieves 91.67% for fine-grained classification, 99.1% for Kendall's Tau, and 90.58% top-5 accuracy for fine-grained frame retrieval, which all surpass the existing best methods.

2. Related Works

Conventional Action Recognition. Various challenging video datasets [9, 25, 32, 38, 40] have been constructed to reason deeply about diverse scenes and situations. These datasets provide labels of high-level concepts or detailed physical aspects for short videos or trimmed clips. To tackle video recognition, large amounts of architectures have been proposed [1, 3, 9, 17, 33, 39, 41, 43, 44]. Most networks are based on 3D Convolution layers and combined with the techniques in image recognition [9, 17, 41], e.g., residual connections [24] and ImageNet pre-training [14]. Some works [33, 44] find that 3D ConvNets have insufficient receptive fields and become the bottleneck of the computational budget.

Recently, Transformers [42] achieved great success in the field of computer vision, e.g., ViT [15] and DETR [7]. There are also several works that extend Transformers to video recognition, such as TimeSformer [3] and ViViT [1]. Due to the strong capacity of Transformers and the global receptive field, these methods have become new state-of-the-art. Combining 2D backbones and Transformers, VTN [33] can efficiently process long video sequences. However, these architecture are all designed for video classification and predict one global class for a video.

Fine-grained Action Recognition. There are also some datasets [27, 35, 37, 49] that investigate fine-grained action recognition. They decompose an action into some action units, sub-actions, or phases. As a result, each video contains multiple simple stages, e.g., wash the cucumber, peel the cucumber, place the cucumber, take a knife, and make a slice in preparing cucumber [35]. However, these fine-level labels are more expensive to collect, resulting in a

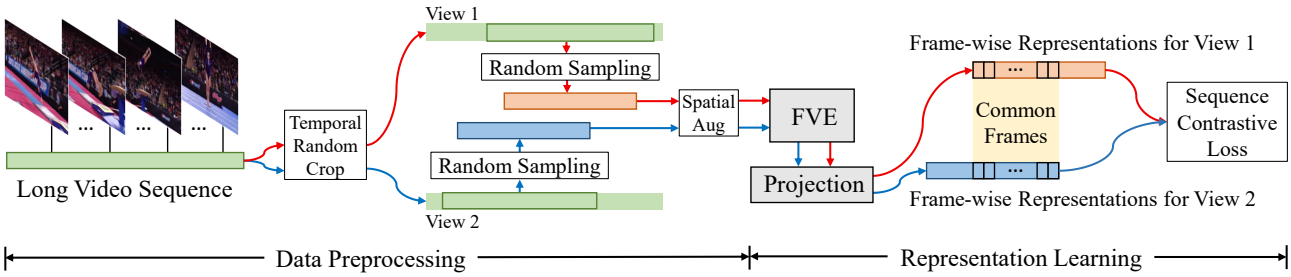


Figure 2. Overview of our framework (CARL). Two augmented views are constructed from a training video through a series of spatio-temporal data augmentations. The frame-level video encoder (FVE) and the projection head are optimized by minimizing the proposed sequence contrastive loss (SCL) between two views.

limited size of these datasets. GTA [21] argues that these boundary of manual annotations are subjective. Therefore, self-supervised learning for fine-level representations is a promising direction.

Self-supervised Learning in Videos. Previous methods of self-supervised learning in videos construct pretext tasks, including inferring the future [22], discriminating shuffled frames [31] and predicting speed [2]. There are also some alignment-based methods, where a pair of videos are trained with cycle-consistent loss [16] or soft dynamic time warping (DTW) [10, 21, 23]. Recently, the contrastive learning methods [11, 12, 20, 45] based on instance discrimination have shown superior performance on 2D image tasks. Some works [18, 26, 34, 36, 48] also use this contrastive loss for video representation learning. They treat different frames in a video [26, 36, 48] or different clips [18, 34] in other videos as negative samples. Different from these methods, our goal is fine-grained temporal understanding of videos and we treat a long sequence of frames as input data. The most relevant work to ours is [28], which utilizes 3D human keypoints for self-supervised acton discovery in long kinematic videos.

3. Method

In this section, we introduce a novel framework named contrastive action representation learning (CARL) to learn frame-wise action representations in a self-supervised manner. In particular, our framework is designed to model long video sequences by considering spatio-temporal context. We first present an overview of the proposed framework in Section 3.1. Then we introduce the details of view construction and data augmentation in Section 3.2. Next, we describe our frame-level video encoder in Section 3.3. Finally, the proposed sequence contrastive loss (SCL) and its design principles are introduced in Section 3.4.

3.1. Overview

Figure 2 displays an overview of our framework. We first construct two augmented views for an input video through a series of spatio-temporal data augmentations. This step is named data preprocessing. Then we feed two augmented views into our frame-level video encoder (FVE) to extract dense representations. Following SimCLR [11], FVE is appended with a small projection network which is a two-layer MLP for obtaining latent embeddings. Due to the fact that temporally adjacent frames are highly correlated, we assume that the similarity distribution between two augmented views should follow a prior Gaussian distribution. Based on the assumption, we propose a novel sequence contrastive loss (SCL) to optimize frame-wise representations in the embedding space.

3.2. View Construction

We first introduce the view construction step of our method, as shown in the 'data preprocessing' part in Figure 2. Data augmentation is crucial to avoid trivial solutions in self-supervised learning [11, 12]. Different from prior methods designed for image data which only require spatial augmentations, we introduce a series of spatio-temporal data augmentations to further increase the variety of videos.

Concretely, for a training video $\boldsymbol{V}$ with S frames, we aim to construct two augmented videos with T frames independently through a series spatio-temporal data augmentations. For temporal data augmentation, we first perform temporal random crop on $\boldsymbol{V}$ to generate two randomly cropped clips with the length of $[T, \alpha T]$ frames, where α is a hyper-parameter controlling maximum crop size. During this process, we guarantee at least β percent of overlapped frames existing between two clips. Then we randomly sample T frames for each video sequence, and obtain $\boldsymbol{V}^1 = \{\boldsymbol{v}_i^1 \mid 1 \leq i \leq T\}$, and $\boldsymbol{V}^2 = \{\boldsymbol{v}_i^2 \mid 1 \leq i \leq T\}$, where $\boldsymbol{v}_i^1$ and $\boldsymbol{v}_i^2$ represent i-th frame from $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$, respectively. We set $T = 240$ by default. For the videos with less than T frames, empty frames are padded before cropping. Finally, we apply several temporal-consistent spatial data augmentations, including random resize and crop, horizontal flip, random color distortions, and random Gaussian blur, on $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$ independently.

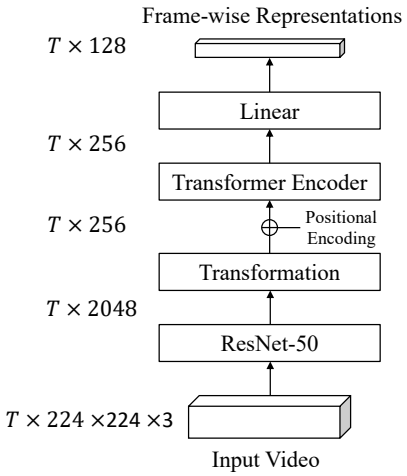


Figure 3. Architecture of the proposed frame-level video encoder (FVE). The input is a long video with T frames and the outputs are frame-wise representations. ResNet-50 is pre-trained on ImageNet. We freeze the first four residual blocks of ResNet-50 and only finetune the last block.

3.3. Frame-level Video Encoder

It is non-trivial to directly apply video classification backbones [9, 17, 41] to model long video sequences with hundreds of frames due to the huge computational cost. TCC [16] presents a video encoder that combines 2D ResNet and 3D Convolution to generate frame-wise features. However, stacking too many 3D Convolutional layers leads to unaffordable computational costs. As a result, this kind of design may have limited receptive fields to capture temporal context. Recently, Transformers [42] achieved great progress in computer vision [7, 15]. Transformers utilize the attention mechanism to solve sequence-to-sequence tasks while handling long-range dependencies with ease. In our network implementation, we adopt the Transformer encoder as an alternative to model temporal context.

Figure 3 shows our frame-level video encoder (FVE). To seek the tradeoff between representation performance and inference speed, we first use a 2D network, e.g., ResNet-50 [24], along temporal dimension to extract spatial features for the RGB video sequence of size $T \times 224 \times 224 \times 3$. Then a transformation block that consists of two fully connected layers with batch normalization and ReLU is applied to project the spatial features to the intermediate embeddings of size $T \times 256$. Following common practice, we add the sine-cosine positional encoding [42] on top of the intermediate embeddings to encode the order information. Next, the encoded embeddings are fed into the 3-layer Transformer encoder to model temporal context. At last, a linear layer is adopted to obtain the final frame-wise representations $\boldsymbol{H} \in \mathbb{R}^{T\times 128}$. We use $\boldsymbol{h}_i$ $(1 \leq i \leq T)$ to denote the representation of i-th frame.

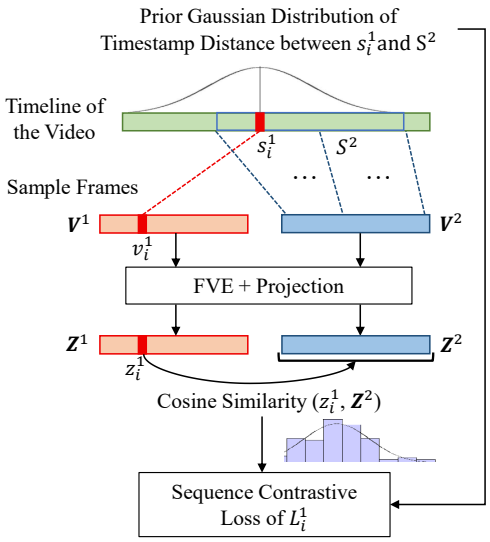


Figure 4. Illustration of the proposed sequence contrastive loss. We use the loss computation of $\boldsymbol{v}_i^1 \in \boldsymbol{V}^1$ as the example. We first compute a prior Gaussian distribution of timestamp distance $(s_i^1 - s_1^2, \cdots, s_i^1 - s_T^2)$. Then the embedding similarity distribution between $\boldsymbol{z}_i^1$ and $\boldsymbol{Z}^2$ is calculated. We minimize the KL-divergence of two distributions in the embedding space.

The 2D ResNet-50 network is pre-trained on ImageNet [14]. Considering the limited computational budget, we freeze the first four residual blocks since they already learned favorable low-level visual representations by pre-training. This simple design ensures that our network can be trained and tested on videos with more than 500 frames. VTN [33] adopt a similar hybrid Transformer-based network to perform video classification. They use the $[CLS]$ token to generate a global feature, while our network is designed to extract frame-wise representations by considering the spatio-temporal context. In addition, our network explores modeling much more prolonged video sequences.

3.4. Sequence Contrastive Loss

SimCLR [11] introduces a contrastive loss named NT-Xent by maximizing agreement between augmented views of the same instance. Unlike self-supervised learning for images, videos provide abundant sequential information, which is a vital supervisory signal. For typical instance discrimination, all instances other than the positive reference are considered as negatives. However, the neighboring frames around the reference frame are highly correlated. Directly regarding these frames as negatives may hurt the learning. Learning principles should be carefully designed to avoid this issue. To optimize frame-wise representations, we propose a novel sequence contrastive loss (SCL) which minimizes the KL-divergence between the embedding similarity of two augmented views and the prior Gaussian distribution, as shown in Figure 4.

Concretely, following SimCLR, we use a small projection network $g(\cdot)$ which is a two-layer MLP to project frame-wise representations $\boldsymbol{H}$ encoded by the proposed FVE to the latent embeddings $\boldsymbol{Z} = g(\boldsymbol{H})$. Let $\boldsymbol{Z}^1 = \{\boldsymbol{z}_i^1 \mid 1 \leq i \leq T\}$ and $\boldsymbol{Z}^2 = \{\boldsymbol{z}_i^2 \mid 1 \leq i \leq T\}$ denote the latent embeddings of $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$, where $\boldsymbol{z}_i^1$ and $\boldsymbol{z}_i^2$ represent the latent embedding of i-th frame in $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$ respectively. Let $S^1 = \{s_i^1 \mid 1 \leq i \leq T\}$ denote timestamp vector of $\boldsymbol{V}^1$, where s_i^1 is the corresponding raw video timestamp of the i-th frame in $\boldsymbol{V}^1$ (see Figure 4). In the same way, we can define $S^2 = \{s_i^2 \mid 1 \leq i \leq T\}$.

Given the i-th reference frame in $\boldsymbol{V}^1$ and its corresponding latent embedding $\boldsymbol{z}_i^1$, due to the fact that temporally adjacent frames are more highly correlated than those faraway ones, we assume the embedding similarity between $\boldsymbol{z}_i^1$ and $\boldsymbol{Z}^2 = \{\boldsymbol{z}_i^2 \mid 1 \leq i \leq T\}$ should follow a prior Gaussian distribution of timestamp distance between s_i^1 and $S^2 = \{s_i^2 \mid 1 \leq i \leq T\}$. This assumption motivates us to use KL-divergence to optimize the embedding space. Specifically, let $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$ denote cosine similarity, and $G(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp(-\frac{x^2}{2\sigma^2})$ denote the Gaussian function, where σ^2 is the variance. We formulate the loss of i-th reference frame in $\boldsymbol{V}^1$ as follows:

$$\mathcal{L}_i^1 = -\sum_{j=1}^{T} w_{ij} \log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i^1, \boldsymbol{z}_j^2)/\tau)}{\sum_{k=1}^{T} \exp(\mathrm{sim}(\boldsymbol{z}_i^1, \boldsymbol{z}_k^2)/\tau)}, \quad (1)$$

$$w_{ij} = \frac{G(s_i^1 - s_j^2)}{\sum_{k=1}^{T} G(s_i^1 - s_k^2)}, \quad (2)$$

where w_{ij} is the normalized Gaussian weight and τ is the temperature parameter. Then the overall loss for $\boldsymbol{V}^1$ can be computed across all frames:

$$\mathcal{L}^1 = \frac{1}{T} \sum_{i=1}^{T} \mathcal{L}_i^1. \quad (3)$$

Similarly, we can calculate the loss $\mathcal{L}^2$ for $\boldsymbol{V}^2$. Our sequence contrastive loss is defined as $\mathcal{L}_{\text{SCL}} = \mathcal{L}^1 + \mathcal{L}^2$. Noticeably, our loss does not rely on frame-to-frame correspondence between $\boldsymbol{V}^1$ and $\boldsymbol{V}^2$, which supports the diversity of spatial-temporal data augmentation.

4. Experiments

4.1. Datasets and Metrics

We use three video datasets, namely *PennAction* [49], *FineGym* [37] and *Pouring* [36] to evaluate the performance of our method. We compare our method with sate-of-the-arts on all three datasets. Unless otherwise specified, all ablation studies on conducted on PennAction dataset.

PennAction Dataset. Videos in this dataset show humans doing different kinds of sports or exercise. Following TCC [16], we use 13 actions of PennAction dataset. In total, there are 1140 videos for training and 966 videos for testing. Each action set has 40-134 videos for training and 42-116 videos for testing. We obtain per-frame labels from LAV [23]. The video frames are from 18 to 663.

FineGym Dataset. FineGym is a recent large-scale fine-grained action recognition dataset that requires representation learning methods to distinguish different sub-actions within the same video. We chunk the original YouTube videos according to the action boundaries so that each trimmed video data only describes a single action type (Floor Exercise, Balance Beam, Uneven Bars, or Vault-Women). Finally, we obtained 3182 videos for training and 1442 videos for testing. The video frames vary from 140 to 5153. FineGym provides two data splits according to the category number, namely FineGym99 with 99 sub-action classes and FineGym288 with 288 sub-action classes.

Pouring Dataset. In this dataset, videos record the process of hand pouring water from one object to another. The phase labels (5 phase classes) are obtained from TCC [16]. Following TCC [16], we use 70 videos for training and 14 videos for testing. The video frames are from 186 to 797.

Evaluation Metrics. For each dataset, We first optimize our network on the training set, without using any labels, and then use the following four metrics to evaluate the frame-wise representations:

- *Phase Classification* (or *Fine-grained Action Classification*) [16] is the averaged per-frame classification accuracy on testing set. Before testing, we fix the network and train a linear classifier by using per-frame labels (phase class or sub-action category) of the training set.
- *Phase Progression* [16] measures the representation ability to predict the phase progress. We fix the network and train a linear regressor to predict the phase progression values (timestamp distance between a query frame and phase boundaries) for all frames. Then it is computed as the average R-squared measure.
- *Kendall's Tau* [16] is calculated over every pair of testing videos by sampling two frames in the first video and retrieving the corresponding nearest frames in the second video, and checking whether their orders are shuffled. It measures how well-aligned two sequences are in time. No more training or finetuning is needed.
- *Average Precision@K* [23] is computed as how many frames in the retrieved K frames have the same phase labels as the query frame. It measures the fine-grained frame retrieval accuracy. $K = 5, 10, 15$ are evaluated. No more training or finetuning is needed.

Following [16, 23, 36], *Phase Classification*, *Phase Progression* and *Kendall's Tau* are evaluated on Pouring

Method	Training Strategy	Annotation	Classification	Progress	τ
TCC [16]	Per-action	Weakly	81.35	0.664	0.701
LAV [23]			84.25	0.661	0.805
TCC [16]	Joint	Weakly	74.39	0.591	0.641
LAV [23]			78.68	0.625	0.684
GTA [21]			-	0.789	0.748
SaL [31]	Joint	None	68.15	0.390	0.474
TCN [36]			68.09	0.383	0.542
Ours			**93.07**	**0.918**	**0.985**

Table 1. Comparison with state-of-the-art methods on PennAction, using various evaluation metrics: *Phase Classification* (Classification), *Phase Progression* (Progress) and *Kendall's Tau* (τ). The top row results are from per-action models, i.e., separate models are trained for different actions. The results in middle and bottom row are obtained from training a single model for all actions.

dataset. For PennAction, all four metrics are evaluated within each action category, and the final results are averaged across the 13 action categories. Following [21], we use *Fine-grained Action Classification* to evaluate our method on FineGym dataset.

4.2. Implementation Details

In our network, we adopt ResNet-50 [24] pre-trained by BYOL [20] as frame-wise spatial encoder. Unless otherwise specified, we use a 3-layer Transformer encoder [42] with 256 hidden size and 8 heads to model temporal context. We train the model using Adam optimizer with learning rate 10^{-4} and weight decay 10^{-5}. We decay the learning rate with cosine decay schedule without restarts [30]. In our loss, we set $\sigma^2 = 10$ and $\tau = 0.1$ as default. Following SimCLR [11], random image cropping, horizontal flipping, random color distortions, and random Gaussian blur are employed as the spatial augmentations. For our temporal data augmentations described in Section 3.2, we set hyper-parameters $\alpha = 1.5$ and $\beta = 20\%$. The video batch size is set as 4 (8 views), and our model is trained on 4 Nvidia V100 GPUs for 300 epochs. During training, we sample $T = 240$ frames for Pouring and FineGym, $T = 80$ frames for PennAction. During testing, we feed the whole video into the model at once, without any temporal downsampling. We L2-normalize the frame-wise representations for evaluation.

4.3. Main Results

Results on PennAction Dataset. In Table 1, our method is compared with state-of-the-art methods on PennAction. TCC [16] and LAV [23] train a separate model for each action ('Per-action' in the table), which results in 13 expert models for 13 action classes correspondingly. In contrast, we train only one model for all 13 action classes ('Joint' in the table). Noticeably, our approach not only outperforms the methods using joint training, but also outperforms the methods adopting per-action training strategy by a large margin under different evaluation metrics. In Table 2, we report the results under the *Average Precision@K* metric, which measures the performance of fine-grained frame retrieval. Surprisingly, although our model is not trained on paired data, it can successfully find frames with similar semantics from other videos. For all AP@K, our method is at least +11% better than previous methods.

Method	AP@5	AP@10	AP@15
TCN [36]	77.84	77.51	77.28
TCC [16]	76.74	76.27	75.88
LAV [23]	79.13	78.98	78.90
Ours	**92.28**	**92.10**	**91.82**

Table 2. Fine-grained frame retrieval results on PennAction.

Method	FineGym99	FineGym288
D^{3}TW [10]	15.28	14.07
SpeedNet [2]	16.86	15.57
TCN [36]	20.02	17.11
SaL [31]	21.45	19.58
TCC [16]	25.18	20.82
GTA [21]	27.81	24.16
Ours	**41.75**	**35.23**

Table 3. Comparison with state-of-the-art methods on FineGym, under the evaluation of *Fine-grained Action Classification*.

Results on FineGym Dataset. Table 3 summarizes the experimental results of *Fine-grained Action Classification* on FineGym99 and FineGym288. Our method outperforms the other self-supervised [2, 31, 36] and weakly supervised [10, 16, 21] methods. The performance of our method surpasses the previous state-of-the-art method GTA [21] by +13.94% on FineGym99 and +11.07% on FineGym288. The weakly supervised methods, i.e., TCC [16], D^{3}TW [10] and GTA [21], assume there exists an optimal alignment between two videos from the training set. However, for FineGym dataset, even in two videos describing

Method	Classification	Progress	τ
TCN [36]	89.53	0.804	0.852
TCC [16]	91.53	0.837	0.864
LAV [23]	92.84	0.805	0.856
Ours	**93.73**	**0.935**	**0.992**

Table 4. Comparison with state-of-the-art methods on Pouring.

Architecture	Classification	Progress	τ
ResNet-50 only	68.63	0.296	0.440
ResNet-50+C3D	83.96	0.705	0.778
ResNet-50+ Transformer	**93.07**	**0.918**	**0.985**

Table 5. Ablation study on different architectures.

the same action, the set and order of sub-actions may differ. Therefore, the alignment found by these methods can be incorrect, which impedes learning. The great improvement verifies the effectiveness of our framework.

Results on Pouring Dataset. As shown in Table 4, our method also achieves the best performance on a relatively small dataset, Pouring. These results further demonstrate the great generalization ability of our approach.

Visualization Results. We present the visualization of fine-grained frame retrieval and video alignment in Section A.

4.4. Ablation Study

In this section, we perform multiple experiments to analyze the different components of our framework. Unless otherwise specified, experiments are conducted on the *PennAction* dataset.

Network Architecture. In Table 5, we investigate the network architecture. 'ResNet-50+Transformer' denotes our default frame-level video encoder introduced in Section 3.3. 'ResNet-50 only' means we remove the Transformer encoder in our network, and only use 2D ResNet-50 and linear transformation layers to extract representations per frame. 'ResNet-50+C3D' represents that two 3D convolutional layers [41] are added on top of the ResNet-50 before the spatial pooling, which is the same as the model adopted in TCC [16] and LAV [23]. These models are all trained with the proposed sequence contrastive loss. Our default network outperforms the other two networks, which attributes to the long-range dependency modeling ability of Transformers.

Layer Number of Transformer Encoder. Table 6 shows studies using different numbers of layers in Transformers. We find that *Phase Classification* increases with more layers. However, *Phase Progression* slightly drops when there are too many layers. We use 3 layers by default.

#Layers	Classification	Progress	τ
1	92.15	0.909	0.985
2	92.61	0.913	**0.990**
3	**93.07**	**0.918**	0.985
4	92.81	0.910	**0.990**

Table 6. Study on the effects of using different number of layers in Transformer encoder.

Learnable Blocks	Classification	Progress	τ
None	90.63	0.907	**0.994**
Block5	**93.07**	0.918	0.985
Block4+Block5	92.98	**0.919**	0.989

Table 7. Ablation study on learnable blocks of ResNet-50.

Method	Classification	Progress	τ
TCN†	86.31	0.898	0.832
TCC†	86.35	0.899	0.980
Ours	**93.07**	**0.918**	**0.985**

Table 8. Applying our network to TCN and TCC. † denotes we re-implement the method and replace the network with ours. "Contrastive baseline" uses the corresponding frame at the other view as the positive sample.

Training Different Blocks of ResNet. In our implementation, ResNet-50 is pre-trained on ImageNet. In Table 7, we study the effects of finetuning different blocks of ResNet-50. The standard ResNet contains 5 blocks, namely *Block1-Block5*. 'None' denotes that all layers of ResNet are frozen. 'Block5' denotes we freeze the first four residual blocks of ResNet and only make the last residual block learnable, which is our default setting. Similarly, 'Block4+Block5' means we freeze the first three blocks and only train the last two blocks. Table 7 shows that encoding dataset-related spatial information is important ('None' vs. 'Block5'), and training more blocks does not lead to improvement ('Block5' vs. 'Block4+Block5').

Applying Our Network to Other Methods. We study whether our frame-level video encoder (FVE) introduced in Section 3.3 can boost the performances of TCC [16] and TCN [36]. We replace the C3D-based network with ours. Table 8 shows the results. We find that the proposed network can dramatically improve the performance of their methods (compared with the results in Table 3). In addition, our method still keeps a large performance gain, which attributes to the proposed sequence contrastive loss.

Hyper-parameters of Sequence Contrastive Loss. We study the hyper-parameters, i.e., temperature parameter τ and Gaussian variance σ^2 in our sequence contrastive loss (see Eq. 2). The variance σ^2 of the prior Gaussian distribution controls how the adjacent frames are semantically sim-

Hyper-parameters	Classification	Progress	τ
τ=0.1, σ^2=1	92.95	0.903	0.963
τ=0.1, σ^2=25	92.03	**0.922**	**0.993**
τ=1.0, σ^2=10	91.57	0.889	0.993
τ=0.3, σ^2=10	92.13	0.903	0.992
τ=0.1, σ^2=10	**93.07**	0.918	0.985

Table 9. Ablation study on Gaussian variance σ^2 and the temperature τ in sequence contrastive loss.

α	Sampling	β (%)	FineGym99
0	Random	20	36.72
1.5			**41.75**
1			39.03
1.5	Even	20	38.44
1.5	Random	0	38.15
		20	**41.75**
		50	39.14
		80	37.94
		100	35.53

Table 10. Ablation study on hyper-parameters of temporal data augmentations. Effects of maximum crop size α, overlap ratio β and random sampling strategy are studied. The experiments are conducted on FineGym99 dataset.

ilar to the reference frame, on the assumption. As Table 9 shows, too small variance ($\sigma^2 = 1$) or too large variance ($\sigma^2 = 25$) degrades the performance. We use $\sigma^2 = 10$ by default. In addition, we observe an appropriate temperature ($\tau = 0.1$) facilitates the learning from hard negatives, which is consistent with the conclusion in SimCLR [11].

Study on Different Temporal Data Augmentations. We study the different temporal data augmentations described in Section 3.2, including maximum crop size α, overlap ratio β between views, and different sampling strategies, namely random sampling and even sampling. Table 10 shows the results. From the table, we can see that the performance drops dramatically when we crop the video with a fixed length ($\alpha = 1$). The performance also decreases when we perform even sampling on the cropped clips. As described in Section 3.4, our sequence contrastive loss does not rely on frame-to-frame correspondence between two augmented views. Experimentally, constructing two views with $\beta = 100\%$ percent of overlapped frames degrades the performance, since the variety of augmented data decreases. In addition, we also observe the performance drops when two views are constructed independently ($\beta = 0\%$). The reason is that in this setting, the training may bring the representations of temporally distant frames closer, which hinders the optimization.

% of Labeled Data →	10	50	100
Number of training frames:			
80	27.10	32.78	34.02
160	30.28	36.46	38.06
240	**33.53**	39.89	41.75
480	31.46	37.92	39.45
Supervised	24.51	**48.75**	**60.37**

Table 11. Ablation studies on number of training frames under different data protocols. Study is conducted on FineGym99 *Fine-grained Action Classification* task. 'Supervised' means all layers are trained with supervised learning.

Number of Training Frames and Linear Evaluation Under Different Data Protocols. As described in Section 3.2, our network takes augmented views with T frames as input. We study the effects of different frame numbers T on FineGym99. Table 11 shows the results. We observe that taking long sequences as input is essential for frame-wise representation learning. However, a too large frame number degrades the performance. We thus set $T = 240$ by default. We also conduct linear evaluation under different data protocols. Concretely, we use 10%, 50% and 100% labeled data to train the linear classifier. Compared with the supervised model (all layers are learnable), our method achieves better performance when the labeled data is limited (10% data protocol).

5. Conclusion

In this paper, we present a novel framework named contrastive action representation learning (CARL) to learn frame-wise action representations, especially for long videos, in a self-supervised manner. To model long videos with hundreds of frames, we introduce a simple yet efficient network named frame-level video encoder (FVE), which considers spatio-temporal context during training. In addition, we propose a novel sequence contrastive loss (SCL) for frame-wise representation learning. SCL optimizes the embedding space by minimizing the KL-divergence between the sequence similarity of two augmented views and a prior Gaussian distribution. Experiments on various datasets and tasks show effectiveness and generalization of our method.

Acknowledgments

This work was supported in part by The National Key Research and Development Program of China (Grant Nos: 2018AAA0101400), in part by The National Nature Science Foundation of China (Grant Nos: 62036009, 61936006), in part by Innovation Capability Support Program of Shaanxi (Program No. 2021TD-05).

References

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

[2] Sagie Benaim, Ariel Ephrat, Oran Lang, Inbar Mosseri, William T. Freeman, Michael Rubinstein, Michal Irani, and Tali Dekel. Speednet: Learning the speediness in videos. In *CVPR*, 2020. 3, 6

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *ArXiv*, 2021. 2

[4] Necati Cihan Camgoz, Simon Hadfield, Oscar Koller, Hermann Ney, and Richard Bowden. Neural sign language translation. In *CVPR*, 2018. 1

[5] Necati Cihan Camgoz, Oscar Koller, Simon Hadfield, and Richard Bowden. Sign language transformers: Joint end-to-end sign language recognition and translation. In *CVPR*, 2020. 1

[6] Kaidi Cao, Jingwei Ji, Zhangjie Cao, C. Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *CVPR*, 2020. 1

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *ArXiv*, 2020. 2, 4

[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. 2

[9] João Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 1, 2, 4

[10] C. Chang, De-An Huang, Yanan Sui, Li Fei-Fei, and Juan Carlos Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation. In *CVPR*, 2019. 3, 6

[11] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 2, 3, 4, 6, 8

[12] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *ArXiv*, 2020. 2, 3

[13] Yutong Chen, Fangyun Wei, Xiao Sun, Zhirong Wu, and Stephen Lin. A simple multi-modality transfer learning baseline for sign language translation. *arXiv preprint arXiv:2203.04287*, 2022. 1

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 2, 4

[15] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 4

[16] Debidatta Dwibedi, Yusuf Aytar, Jonathan Tompson, Pierre Sermanet, and Andrew Zisserman. Temporal cycle-consistency learning. In *CVPR*, 2019. 1, 2, 3, 4, 5, 6, 7

[17] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 1, 2, 4

[18] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross B. Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *CVPR*, 2021. 3

[19] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter N. Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 1

[20] Jean-Bastien Grill, Florian Strub, Florent Altch'e, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Ávila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. *ArXiv*, 2020. 2, 3, 6

[21] Isma Hadji, Konstantinos G. Derpanis, and Allan D. Jepson. Representation learning via global temporal alignment and cycle-consistency. In *CVPR*, 2021. 1, 2, 3, 6

[22] Tengda Han, Weidi Xie, and Andrew Zisserman. Video representation learning by dense predictive coding. In *ICCVW*, 2019. 3

[23] Sanjay Haresh, Sateesh Kumar, Huseyin Coskun, Shahram Najam Syed, Andrey Konin, M. Zeeshan Zia, and Quoc-Huy Tran. Learning by aligning videos in time. In *CVPR*, 2021. 1, 2, 3, 5, 6, 7

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 4, 6

[25] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 2

[26] Haofei Kuang, Yi Zhu, Zhi Zhang, Xinyu Li, Joseph Tighe, Sören Schwertfeger, C. Stachniss, and Mu Li. Video contrastive learning with global context. *ArXiv*, 2021. 3

[27] Hilde Kuehne, Ali Bilgin Arslan, and Thomas Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *CVPR*, 2014. 1, 2

[28] Kenneth Li, Xiao Sun, Zhirong Wu, Fangyun Wei, and Stephen Lin. Towards tokenized human dynamics representation. *arXiv preprint arXiv:2111.11433*, 2021. 3

[29] Yuxuan Liu, Abhishek Gupta, P. Abbeel, and Sergey Levine. Imitation from observation: Learning to imitate behaviors from raw video via context translation. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 1

[30] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv: Learning*, 2017. 6

[31] Ishan Misra, C. Lawrence Zitnick, and Martial Hebert. Shuffle and learn: Unsupervised learning using temporal order verification. In *ECCV*, 2016. 3, 6

[32] Mathew Monfort, Bolei Zhou, Sarah Adel Bargal, Alex Andonian, Tom Yan, Kandan Ramakrishnan, Lisa M. Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, and Aude Oliva. Moments in time dataset: One million videos for event understanding. *PAMI*, 2020. 2

[33] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *ArXiv*, 2021. 1, 2, 4

[34] Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, H. Wang, Serge J. Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *CVPR*, 2021. 3

[35] Marcus Rohrbach, Anna Rohrbach, Michaela Regneri, Sikandar Amin, Mykhaylo Andriluka, Manfred Pinkal, and Bernt Schiele. Recognizing fine-grained and composite activities using hand-centric features and script data. In *IJCV*, 2015. 2

[36] Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018. 1, 3, 5, 6, 7

[37] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *CVPR*, 2020. 1, 2, 5

[38] Gunnar A. Sigurdsson, Gül Varol, X. Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *ECCV*, 2016. 2

[39] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 1, 2

[40] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *ArXiv*, 2012. 1, 2

[41] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015. 1, 2, 4, 7

[42] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 2, 4, 6

[43] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks for action recognition in videos. *PAMI*, 2019. 2

[44] X. Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 1, 2

[45] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *Advances in Neural Information Processing Systems*, 34, 2021. 3

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

[47] Yinghao Xu, Fangyun Wei, Xiao Sun, Ceyuan Yang, Yujun Shen, Bo Dai, Bolei Zhou, and Stephen Lin. Cross-model pseudo-labeling for semi-supervised action recognition. *arXiv preprint arXiv:2112.09690*, 2021. 1

[48] Ting Yao, Yiheng Zhang, Zhaofan Qiu, Yingwei Pan, and Tao Mei. Seco: Exploring sequence supervision for unsupervised representation learning. In *AAAI*, 2021. 3

[49] Weiyu Zhang, Menglong Zhu, and Konstantinos G. Derpanis. From actemes to action: A strongly-supervised representation for detailed action understanding. In *ICCV*, 2013. 1, 2, 5