Spatio-Temporal Self-supervision for Few-Shot Action Recognition

Wanchuan Yu[1], Hanyu Guo[1], Yan Yan[1], Jie Li[2], and Hanzi Wang[1(✉)]

[1] Fujian Key Laboratory of Sensing and Computing for Smart City, School of Informatics, Xiamen University, Xiamen, China
{wanchuan,guohanyu}@stu.xmu.edu.cn, {yanyan,hanzi.wang}@xmu.edu.cn
[2] Video and Image Processing System Laboratory, School of Electronic Engineering, Xidian University, Xi'an, China
leejie@mail.xidian.edu.cn

Abstract. Few-shot action recognition aims to classify unseen action classes with limited labeled training samples. Most current works follow the metric learning technology to learn a good embedding and an appropriate comparison metric. Due to the limited labeled data, the generalization of embedding networks is limited when employing the meta-learning process with episodic tasks. In this paper, we aim to repurpose self-supervised learning to learn a more generalized few-shot embedding model. Specifically, a Spatio-Temporal Self-supervision (STS) framework for few-shot action recognition is proposed to generate self-supervision loss at the spatial and temporal levels as auxiliary losses. By this means, the proposed STS can provide a robust representation for few-shot action recognition. Furthermore, we propose a Spatio-Temporal Aggregation (STA) module that accounts for the spatial information relationship among all frames within a video sequence to achieve optimal video embedding. Experiments on several challenging few-shot action recognition benchmarks show the effectiveness of the proposed method in achieving state-of-the-art performance for few-shot action recognition.

Keywords: Few-shot learning · Action recognition · Self-supervised learning

1 Introduction

Deep learning has achieved remarkable success in the field of action recognition [10,13,18,22]. The main reason for the significant progress is the sufficiently large-scale labeled training data. However, the time-consuming and costly annotation process renders acquiring adequate data for network training an infrequent occurrence. Hence, recent research has placed greater attention on enhancing the generalization of the model to novel data with limited instances. Similar to the capacity of humans to transfer knowledge from only a few examples, few-shot learning (FSL) shows promise in mitigating data scarcity issues. While

Q. Liu et al. (Eds.): PRCV 2023, LNCS 14425, pp. 84–96, 2024.
https://doi.org/10.1007/978-981-99-8429-9_7

recent few-shot classification has made significant progress for images, progress in video classification has remained unsolved. Few-shot action recognition (FASR) is much more complicated due to the additional temporal dimension. Besides, video actions have different characteristics in terms of speed, duration, and occurrence scenarios.

To deal with the FASR problem, most existing metric-based FASR methods simply calculate the similarity between the embedding of a support class and a query video. Moreover, these methods mainly use frame-level embeddings [1,2], clip-level embeddings [12,16], or patch-level embeddings [24] for temporal alignment to obtain accurate video matching. Despite significant effects, these methods still need to address the key challenge of improving the generalization of the learned few-shot embedding model. Besides, unconstrained learning from training data will lead to the inductive bias of source classes and weaken the generalization performance of embeddings. ARN [23] has recently used spatial and temporal self-supervision to train a more robust encoder and attention. However, this method alone self-supervises support videos and query videos without fully capitalizing on their inherent connection, which is more suitable for few-shot learning. Therefore, using support and query videos together for self-supervision can help narrow the distances between the same categories and map instances of different categories to different clusters.

To address the above problems, a spatial-temporal self-supervision (STS) framework using self-supervised learning for few-shot action recognition is proposed. To be specific, we first propose a spatial cross self-supervision module (SCS) based on the spatial scale to enhance patch representations by establishing correlations between patches at different locations. This module effectively addresses the issue of key patches impacting correlation establishment in various video scenarios caused by displacement and indentation. Secondly, we develop a temporary cross self-supervision (TCS) module based on a temporary scale to fully enhance the temporality of the video, which can solve the problem of misclassification due to similar directionality of videos (e.g.,"moving something away from the camer" vs."moving something towards the camer"). Moreover, the spatio-temporal aggregation (STA) module is utilized to aggregate video representations along the spatial and temporal dimensions, reducing the emphasis on a specific frame during temporal matching. Our model achieves competitive performance on several action recognition benchmarks: Something v2 (SSv2) [5], Kinetics [7], HMDB51 [8], and UCF101 [15].

2 Related Work

2.1 Few-Shot Action Recognition

CMN [25] proposes a memory network structure to obtain an optimal video representation and a multi-saliency embedding algorithm to encode a variable-length video sequence into a fixed-size representation. OTAM [2] proposes a dynamic time-warping algorithm to enhance long-term temporal ordering information by ordered temporal alignment. ARN [23] constructs a C3D encoder

to capture short-term dependencies and leverage permutation-invariant pooling to learn discriminative action representations. The recent method TRX [12] compares the query samples to sub-sequences of all support samples with an attention mechanism to construct query-specific class prototypes for few-shot matching and achieves promising results. STRM [16] proposes spatio-temporal enrichment and temporal relationship modeling modules to measure query-class similarity. In contrast to previous works, our algorithm uses a self-supervised approach to construct separate spatial and temporal pretext tasks of the model, enhancing the generalization to novel classes of spatio-temporal modeling.

2.2 Self-supervised Learning (SSL)-Based Few-Shot Learning

The success of contrastive learning approaches like SimCLR [3] and MoCo [6] shows that the feature extraction network trained using self-supervised learning can have a robust representational capacity. Few-shot learning works [9,14] have achieved better results by combining the FSL framework with well-designed auxiliary self-supervised pretext tasks. This indicates that such methods can facilitate the transferability of learned feature representations. SLA [9] augments original labels through self-supervision of input transformation to relax invariant constraints during simultaneous learning of the original and self-supervised tasks. More recently, ESPT [14] proposes a new type of self-supervised pretext task for few-shot image classification that uses relations between local spatial features of multiple image samples in each episode to construct a supervision signal. However, most SSL augmented few-shot learning methods are not used in action recognition. Our algorithm is specifically designed for few-shot action recognition. By separately constructing self-supervised objective functions at the spatial and temporal levels, we combine them with the objective function of the original task to optimize the model parameters.

3 Method

3.1 Problem Definition

In the few-shot action recognition task, the goal is to classify an unlabeled video (query set) into one of the several classes represented by a limited number of labeled video samples (support set) that have not been seen during training. To this end, videos in a dataset are divided into two sets with disjoint classes: the meta-training set D_{train} and the meta-testing set D_{test}, i.e., $C_{train} \cap C_{test} = \emptyset$. Then, following previous work [2,23,25] using the episodic training strategy [17] to optimize the model with a meta-training set D_{train}. For each episode, we randomly sample N action classes each with K videos from D_{train} to construct the support set $S = \{(x_s, y_s), y_s \in C_{train}, s = 1, \ldots, n \times k\}$. And the query set $S = \{(x_q, y_q), y_q \in C_{train}, s = 1, \ldots, n \times p\}$ sampled from the rest of the videos of the N selected classes. To perform meta-learning, S and Q are completely disjoint, i.e., $S \cap Q = \emptyset$. Specifically, we use a large number of episodic tasks sampled from D_{train} for training to adapt to D_{test}. During the inference phase, episodic tasks are sampled on the D_{test} in a similar way as meta-training.

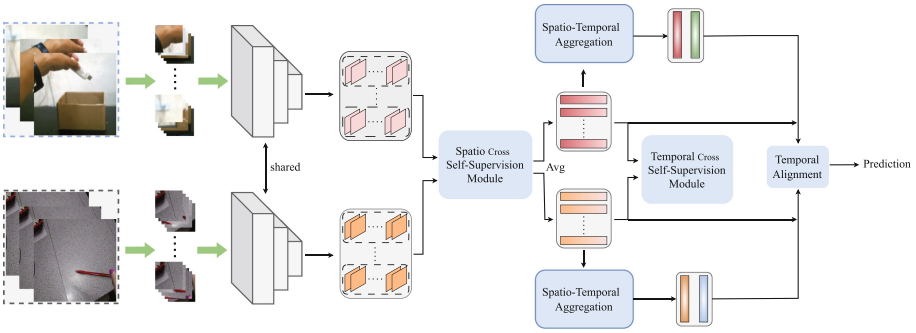

Fig. 1. Illustration of the proposed overall framework. Embedded video features are first fed into the spatial cross self-supervision (SCS) module to enhance patch representations. After that, the temporary cross self-supervision (TCS) module is used to boost the temporality of the video. Moreover, the spatiotemporal aggregation (STA) module incorporates the spatial relationships between frames to obtain optimal video embedding. Finally, we compute the similarity scores to make the final prediction.

3.2 Spatio-Temporal Self-supervision Framework

Overview. Figure 1 illustrates the overview of the proposed STS framework. The input support videos S and query videos Q with T frames are passed through an embedding network (e.g., ResNet-50) to extract support features $F_s = \{s_1, s_2, \ldots, s_{n\times k}\}$ and query features $F_q = \{q_1, q_2, \ldots, q_{n\times p}\}$, where $s_i = \{s_i^1, s_i^2, \cdots, s_i^T\}, q_i = \{q_i^1, q_i^2, \cdots, q_i^T\}$ and $s_i^j, q_i^j \in R^{P^2\times D}$. Each frame feature consists of $P \times P$ patch features with dimension D.

Spatial Cross Self-supervision Module. The position of the notable patches within a frame vary across different video scenes and motion postures. Thus, enhancing the correlation between associated patches is advantageous for capturing the precise frame-level appearance. Given the support feature F_s and query feature F_q, as illustrated in Fig. 2(a), we initially leverage self-attention to capture spatial relationships among patches within a frame. Let $x_i \in \mathbb{R}^{P^2\times D}$ denote the patch features s_i^j, q_i^j of a frame p_j $(j \in [1, T])$, where $P \times P$ is the number of patches. Then we use weights $W_q, W_k, W_v \in \mathbb{R}^{D\times D}$ to map patch features to x_i^q, x_i^k, and x_i^v, where $[x_i^q; x_i^k; x_i^v] = [W_q x_i; W_k x_i; W_v x_i]$ and D is the dimension of the input patch features. The attention matrix is computed by the dot-products between the query and key matrices. Then the value matrix and attention matrix are dot-products to reweight the correlations among all patches:

$$z_i = \lambda(\frac{x_i^q x_i^{k\mathsf{T}}}{D})x_i^v + x_i \tag{1}$$

where λ denotes the softmax function. Although this attention mechanism can establish correlations between patches, it cannot capture the relative positions and relationships of noteworthy patches in different video scenes. Meanwhile,

the absence of robust constraints may introduce the induction bias, ultimately resulting in incorrect relationships between the query and support sets. Therefore, we calculate ordered cross-attention a^{space} and unordered cross-attention $\hat{a}^{space}$ between the query and support sets separately and then use $\hat{a}^{space}$ to enhance the relative relationship between patches. Let Q_i denote the patch features z_i obtained from the query feature q_i^j and S_i from the support feature s_i^j. A sub-network $\phi(\cdot)$ is then used to approximately enhance Q_i before mapping with the parameter W_q. The $Q_i^{\mathsf{T}} \in \mathbb{R}^{D\times P^2}$ is mapped with the parameter $W_p \in \mathbb{R}^{P^2}$ to obtain patch-level enriched features $\hat{Q}_i$. Thus, $\phi(Q_i) = \hat{Q}_i$ can be defined as:

$$\hat{Q}_i = \sigma(Q_i^{\mathsf{T}} W_p)^{\mathsf{T}} + Q_i \tag{2}$$

where σ denotes the ReLU non-linearity. After that, Q_i^q and S_j^k are mapped from weights $\hat{W}_q, \hat{W}_k \in \mathbb{R}^{D\times D}$, where $[Q_i^q; S_j^k] = [\hat{W}_q \hat{Q}_i; \hat{W}_k S_j]$. Let $o^s \in \mathbb{N}^{P^2}$ represent randomly shuffling the order of patches, then the ordered cross-attention $a_{(i,j)}^{space}$ and the unordered cross-attention $\hat{a}_{(i,j)}^{space}$ can be defined as:

$$a_{(i,j)}^{space} = Q_i^q S_j^k \tag{3}$$

$$\hat{a}_{(i,j)}^{space} = \nu(\phi(Q_i^q), o^s) S_j^k \tag{4}$$

where $\nu(x, o^s)$ is a function that shuffles the patches in x based on the order o^s. For instance, let o^s be the shuffle order $[2, 1, 3]$. In this case, the first patch of $\hat{Q}_i$ is obtained by enhancing the second patch of Q_i. Therefore, we can propose a spatial cross self-supervised loss to enhance the correlation between related patches in a frame, and the loss can be defined as:

$$\mathcal{L}_{spa}^{self} = \frac{1}{TN^2} \sum_{t=1}^{T} \sum_{k=1}^{N} \sum_{p=1}^{N} (1 - \hat{a}_{(k,p)}^{space-t}) \varepsilon(a_{(o^s(k),p)}^{space-t}) \tag{5}$$

where $\varepsilon(a) = 1$ if $a > \theta$, otherwise $\varepsilon(a) = -1$. The default setting for the judgment value θ is the P-th largest value in a. By employing ordered cross attention to guide unordered cross attention, we effectively enhance spatial connections between patches, thereby facilitating the utilization of a reweighted value embedding to generate more discriminative support class-specific embedding. The generation process can be defined as:

$$\alpha_i = \lambda(\frac{a_i}{D}) S_i^v \tag{6}$$

where S_i^v is mapped from weights $\hat{W}_v \in \mathbb{R}^{D\times D}$, the query features Q_i^v also uses the same weights. Let $[Q_i^v; S_i^v] = [\hat{W}_v Q_v; \hat{W}_v S_v]$. Then we calculate the distances between spatial patches point-to-point based on the ground truth label between query features and support class-specific features to define the cross-entropy loss $\mathcal{L}_{spa}$ at the spatial level.

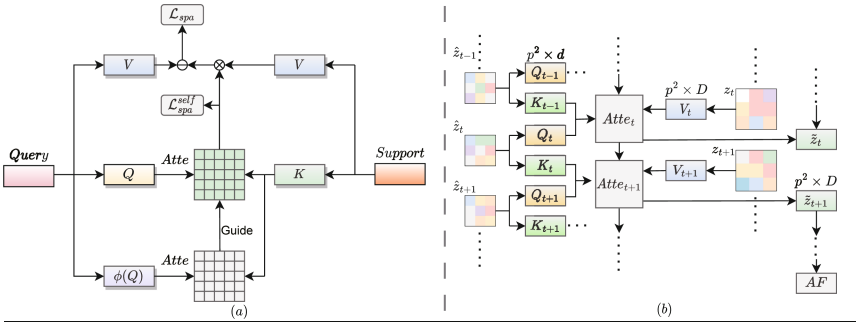

Fig. 2. (a) The spatial cross self-supervision (SCS) module, where the output self-supervision loss $\mathcal{L}_{spa}^{self}$ and the spatial cross-entropy loss $\mathcal{L}_{spa}$ are calculated by guiding the original cross-attention matrix using the disrupted cross-attention matrix. (b) The spatio-temporal aggregation (STA) module, where the output feature AF is aggregated by all the attention weights, which are computed by the cross-attention matrices for two adjacent frames in succession.

Temporal Cross Self-supervision Module. The temporary orders play an important role in the video, but using temporary matching directly does not reveal the difference, such as "moving something away from the camera" vs. "moving something towards the camera". Thus, learning about temporary orders is beneficial for processing more detailed temporal relationships. Similar to spatial processing, we use the Q_i^q, Q_i^v, S_i^k, and S_i^v for temporal cross self-supervision. Then spatial global-average pooling is applied to collapse the spatial dimension. Let $o^t \in \mathbb{N}^N$ represent randomly shuffling the order of video frames, and then the unordered cross-attention $\hat{a}_{(i,j)}^{time}$ can be defined as:

$$\hat{a}_{(i,j)}^{time} = \nu(\phi(\psi(Q_i^q)), o^t)\psi(S_j^k) \tag{7}$$

where $\nu(x, o^t)$ is a function that disrupts the video frames in x based on the order o^t, $\psi(\cdot)$ represents a spatial pooling function. Therefore, we can propose a temporal cross self-supervised loss to learn the temporary orders, and the loss can be defined as:

$$\mathcal{L}_{time}^{self} = \frac{1}{T^2}\sum_{t=1}^{T}\sum_{t'=1}^{T}(1 - \hat{a}_{(t,t')}^{time})\zeta((o^t(t), t')) \tag{8}$$

where $\zeta(i,j) = 1$ if $i > j$, otherwise $\zeta(i,j) = -1$. By supervising the order of videos, the directionality between frames can be enhanced to generate more discriminative temporal features.

Spatio-Temporal Aggregation Module. Many previous works [1,2] tend to ignore the long-term temporal relationships existing in the sequence of video. To address this limitation, our approach introduces a spatio-temporal aggregation

module (STA) that accounts for the spatial information relationship among all frames within a video sequence in order to achieve optimal video embedding. As illustrated in Fig. 2(b), we use $Z = \{z_1, z_2, \ldots, z_T\}$ to represent the original input, where $z_i \in \mathbb{R}^{P^2 \times D}$. To reduce the amount of computation, we utilize a 1×1 convolution operation to compute the reduced dimensional video sequence $\hat{Z} = \{\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_T\}$, where $\hat{z}_i \in \mathbb{R}^{P^2 \times d}$. We apply a linear transformation to both the dimensionality reduction feature $\hat{z}_t$ and the original feature z_t to obtain triplets of query-key-values:

$$z_t^q = \widetilde{W}_q \hat{z}_t, z_t^k = \widetilde{W}_k \hat{z}_t, z_t^v = \widetilde{W}_v z_t \tag{9}$$

where $\widetilde{W}_q$, $\widetilde{W}_k$, $\widetilde{W}_q$ denote the weight of the linear transformation layer. In a similar manner, other frame features are processed. The spatial relationship between adjacent frames is captured via the computation of the attention matrix of frame $t+1$ using the dot product between the query matrix of frame $t+1$ and the key matrix of frame t.

$$Atte_{t+1} = \frac{z_{t+1}^q z_t^{k\mathsf{T}}}{d} \tag{10}$$

However, Eq. (10) only accounts for the interaction between two adjacent frames. To capture the spatial relationships of all frames effectively, we combine all the preceding attention matrices leading up to a specific frame t. This aggregation facilitates to compute the spatio-temporal characteristics of frame $t+1$:

$$\tilde{z}_{t+1} = \lambda(\sum_{i=1}^{t+1} Atte_i) z_{t+1}^v \tag{11}$$

where λ denotes the softmax function. Finally, the spatial-temporal aggregation feature $\tilde{Z}$ and the original feature Z are simply pooled in the spatio-temporal dimension and added together to obtain the final feature AF:

$$AF = \frac{\sum_{i=1}^{T} \psi(\tilde{z}_i)}{T} + \frac{\sum_{j=1}^{T} \psi(z_i)}{T} \tag{12}$$

where $\psi(\cdot)$ represents a spatial pooling function. Simply by calculating the Euclidean distance between the aggregated features AF, we can obtain the distance D_c. However, only using this strategy will fail to capture some fine-grained action information. To address this, we use the existing fine-grained distance function TRX [12] to calculate the fine-grained distance D_f. Finally, the global distance D_g can be expressed as follows:

$$D_g = D_c + D_f \tag{13}$$

Then we calculate the distances based on the ground truth label using the video-to-class distance D_g to define the cross-entropy loss $\mathcal{L}_g$ as the main loss. With ω_i as hyper-weights, our STS is trained using the joint formulation given by:

$$\mathcal{L} = \omega_1 \mathcal{L}_g + \omega_2 \mathcal{L}_{spa} + \omega_3 \mathcal{L}_{spa}^{self} + \omega_4 \mathcal{L}_{time}^{self} \tag{14}$$

4 Experiments

4.1 Experimental Settings

Datasets. We evaluate our method on four widely used datasets, including Something v2 (SSv2) [5], Kinetics [7], HMDB51 [8], and UCF101 [15] for few-shot action recognition. For Kinetics, we follow the splits in CMN [25] to select 100 action classes from Kinetics-400, which contains various activities in daily life and is rich in scene context. The 100 classes with 100 video clips per class are divided into 64, 12, and 24 for training, validation, and testing. For SSv2, we follow the two widely used splits denoted as SSv2† and SSv2*, proposed by [26] and [2] respectively. Both splits adopt 64, 24, and 12 non-overlapping action classes as the training set, validation set, and testing set. But compared with SSv2†, the training set of SSv2* uses approximately 10x videos per class. For UCF101, we use the splits from [23], which sample 70, 10, and 21 non-overlapping action classes as the training set, validation set and testing set. For HMDB51 with 51 classes with at least 101 video clips per class, we also use the split from [23] and select 31 training, 10 validation, and 10 testing classes.

Implementation Details. We follow the sparse sampling strategy described in TSN [19], which divides each input video into $N = 8$ segments and then randomly samples one frame in each segment. We resize the each frame scale into 224×224. Then we use ResNet-50 pretrained on ImageNet as the feature extractor. With $D = 2048$, an adaptive max pooling operation reduces the spatial resolution to P, where $P = 4$. During training, the weight of L_g, L_{spa}, L_{spa}^{self}, and L_{time}^{self} is set to 1, 0.5, 0.1, and 0.1, respectively. We train our model for 75,000 randomly sampled training episodes for SSv2* and SSv2† dataset with a learning rate of 1×10^{-4}. For the other three datasets, we set the learning rate to 1×10^{-3} and trained for 50,000 episodes. To evaluate few-shot performance on each benchmark, we randomly construct 10,000 episodes from the test set and report the average classification accuracy.

4.2 Comparison with State-of-the-Art Methods

In Table 1, we compare our method with state-of-the-art algorithms on Kinetics, SSV2†, SSV2*, UCF101, and HMDB51. On the five datasets, we conduct experiments under 5-way 5-shot settings.

Results on Kinetics. Table 1 shows that our model significantly outperforms all competing methods under 5-shot settings. For instance, our STS achieves new state-of-the-art results with 87.5%. Compared with current state-of-the-art methods, such as TRX [12] and STRM [16], our STS outperforms these methods by 1.7% and 0.9% under the 5-shot setting, respectively. This demonstrates that the attributes of the spatio-temporal self-supervision framework surpass these traditional spatio-temporal modeling methods.

Results on SSV2. We also evaluate the proposed STS on the SSV2 dataset, which is more complex in temporal reasoning. The gains of our STS on SSv2 are more evident, further demonstrating the advantages of our temporary self supervision on this dataset. Our method achieves +2.0% improvements on SSV2† compared with STRM [16] under the 5-shot settings. For the SSV2*, which includes more training data, our method achievements +1.1% improvement in the 5-shot settings. These results indicates that more training data can generate more discriminative embeddings through self -supervision.

Table 1. Comparison with state-of-the-art methods on Kinetics, SSV2†, SSV2*, UCF101, and HMDB51 in terms of 5-shot classification accuracy. "-" stands for the result is not available in published works. The best results are in bold.

Method	Kinetics	SSV2†	SSV2*	UCF101	HMDB51
MAML [4]	75.3	41.9	-	-	-
CMN [25]	78.9	-	-	-	-
TARN [1]	78.5	-	-	-	-
OTAM [2]	85.8	48.0	52.3	88.9	68.0
TRX [12]	85.9	**59.1**	64.6	96.1	75.6
MTFAN [21]	87.4	-	60.4	95.1	74.6
STRM [16]	86.7	55.3	68.1	96.9	77.3
HyRSM [20]	86.1	56.1	69.0	94.7	76.0
Nguyen *et al.* [11]	87.4	-	61.1	95.9	76.9
HCL [24]	85.8	55.4	64.9	93.9	76.3
Ours	**87.5**	57.3	**69.2**	**97.1**	**77.5**

Results on UCF101 and HMDB51. In order to further verify our STS, we also compare it with state-of-the-art methods on the UCF101 and HMDB51 datasets, whose data is simpler compared with Kinetics and SSv2. And the results are shown in Table 1. Our method improves over TRX [12] on HMDB51 e.g., +1.9% for the 5-shot settings. Similarly, STS improves over TRX on UCF101 e.g., +1.0% for the 5-shot settings.

4.3 Ablation Studies

Influence of the Different Training Losses in the STS Framework. To examine the effect of various training losses in our proposed STS, we performed ablation studies on the SSV2†, Kinetics, and HMDB51 datasets under the 5-way 5-shot setting. The results are shown in Table 2. The coarse-grained loss $\mathcal{L}_c$ primarily focuses on the similarity between the aggregated video embedding, while the fine-grained loss $\mathcal{L}_f$ mainly considers the similarity between the clip-level embedding, the total loss is given by $\mathcal{L}_g = \mathcal{L}_c + \mathcal{L}_f$. As evidenced in lines

1-3, adopting both coarse-grained and fine-grained classification losses without a doubt leads to better outcomes than just relying on $\mathcal{L}_c$ or $\mathcal{L}_f$. Since $\mathcal{L}_f$ is more suited to the 5-way 5-shot setup, it obtains considerably better results than $\mathcal{L}_c$. On the Kinetics and HMDB51 datasets, incorporating a distance loss $\mathcal{L}_{spa}$ between spatial patches point-to-point effectively enhances the model performance. Self-supervised learning of temporal and spatial features has demonstrated noticeable benefits, as shown in rows 5 and 6 of Table 2, respectively. Integrating self-supervision in both spatial and temporal domains can significantly improve the generality of the acquired embedding. Finally, we achieve the best results by combining $\mathcal{L}_g + L_{spa} + \mathcal{L}_{spa}^{self} + \mathcal{L}_{time}^{self}$ in row 6.

Table 2. Influence of various training losses in the STS framework on Kinetics, HMDB51, and SSV2† under the 5-way 5-shot setup. The best result are in bold.

	STS						Kinetics	HMDB51	SSV2†
	$\mathcal{L}_c$	$\mathcal{L}_f$	$\mathcal{L}_g$	$\mathcal{L}_{spa}$	$\mathcal{L}_{spa}^{self}$	$\mathcal{L}_{time}^{self}$	5-shot	5-shot	5-shot
1	✓	×	×	×	×	×	84.3	68.5	47.5
2	×	✓	×	×	×	×	85.0	75.4	55.1
3	×	×	✓	×	×	×	85.5	75.9	55.9
4	×	×	✓	✓	×	×	86.3	76.5	56.2
5	×	×	✓	✓	✓	×	87.1	77.2	56.5
6	×	×	✓	✓	✓	✓	**87.5**	**77.5**	**57.3**

Results on Using only the TRX Loss Under the 5-Way 1-Shot Setting. Table 3 shows the performance of our STS framework only based on TRX [12] alignment metrics in terms of 1-shot setting. TRX is designed for 5-shot, so its performance is not ideal under the 1-shot setting. To prove the validity of our model, we removed the coarse-grained loss $\mathcal{L}_c$ and kept only the fine-grained loss $\mathcal{L}_f$, i.e., the TRX loss, and compared the performance of TRX and its subsequent improvement STRM [16] under the 5-way 1-shot setting.

Table 3. Results on using only the TRX loss on Kinetics, SSV2†, SSV2*, UCF101, and HMDB51 under the 5-way 1-shot setup. The best results are in bold.

Method	Kinetics	SSV2†	SSV2*	UCF101	HMDB51
TRX [12]	63.6	36.0	42.0	78.2	53.1
STRM [16]	62.9	37.1	43.1	80.5	52.3
STS($\mathcal{L}_f$)	64.3	37.9	43.7	81.0	54.8
STS($\mathcal{L}_f$, $\mathcal{L}_c$)	**65.1**	**38.2**	**44.0**	**81.6**	**55.6**

5 Conclusions

In this paper, we propose a novel Spatio-Temporal Self-supervision (STS) framework for few-shot action recognition that consists of a spatial cross self-supervision (SCS) module, a temporary cross self-supervision (TCS) module, and a spatio-temporal aggregation (STA) module. The SCS and TCS modules for few-shot action recognition are proposed to generate self-supervision loss at the spatial and temporal levels as auxiliary losses to facilitate the transferability of learned feature representations. The STA module accounts for the spatial information relationship among all frames within a video sequence to achieve optimal video embedding. Extensive experiments on five commonly used benchmarks verify the effectiveness of our method and demonstrate that STS achieves state-of-the-art performance under the 5-shot setting.

Acknowledgements. This work was supported by the National Key Research and Development Program of China under Grant 2022ZD0160402, by the National Natural Science Foundation of China under Grant U21A20514, 62176195, and Grants 62372388, 62071404, and by the FuXiaQuan National Independent Innovation Demonstration Zone Collaborative Innovation Platform Project under Grant 3502ZCQXT2022008.

References

1. Bishay, M., Zoumpourlis, G., Patras, I.: Tarn: temporal attentive relation network for few-shot and zero-shot action recognition. arXiv preprint arXiv:1907.09021 (2019)
2. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10618–10627 (2020)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: International Conference on Machine Learning, pp. 1597–1607. PMLR (2020)
4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning, pp. 1126–1135. PMLR (2017)
5. Goyal, R., et al.: The "something something" video database for learning and evaluating visual common sense. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 5842–5850 (2017)
6. He, K., Fan, H., Wu, Y., Xie, S., Girshick, R.: Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9729–9738 (2020)
7. Kay, W., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
8. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: a large video database for human motion recognition. In: 2011 International Conference on Computer Vision, pp. 2556–2563. IEEE (2011)
9. Lee, H., Hwang, S.J., Shin, J.: Self-supervised label augmentation via input transformations. In: International Conference on Machine Learning, pp. 5714–5724. PMLR (2020)

10. Liu, Z., et al.: Video swin transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3202–3211 (2022)
11. Nguyen, K.D., Tran, Q.H., Nguyen, K., Hua, B.S., Nguyen, R.: Inductive and transductive few-shot video classification via appearance and temporal alignments. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13680, pp. 471–487. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-20044-1_27
12. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal-relational crosstransformers for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 475–484 (2021)
13. Qiu, Z., Yao, T., Ngo, C.W., Tian, X., Mei, T.: Learning spatio-temporal representation with local and global diffusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 12056–12065 (2019)
14. Rong, Y., Lu, X., Sun, Z., Chen, Y., Xiong, S.: ESPT: a self-supervised episodic spatial pretext task for improving few-shot learning. arXiv preprint arXiv:2304.13287 (2023)
15. Soomro, K., Zamir, A.R., Shah, M.: UCF101: a dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
16. Thatipelli, A., Narayan, S., Khan, S., Anwer, R.M., Khan, F.S., Ghanem, B.: Spatio-temporal relation modeling for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19958–19967 (2022)
17. Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al.: Matching networks for one shot learning. In: Advances in Neural Information Processing Systems, vol. 29 (2016)
18. Wang, L., Zhu, S., Li, Z., Fang, Z.: Complementary temporal classification activation maps in temporal action localization. In: Ma, H., et al. (eds.) PRCV 2021. LNCS, vol. 13020, pp. 373–384. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-88007-1_31
19. Wang, L., et al.: Temporal segment networks: towards good practices for deep action recognition. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9912, pp. 20–36. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46484-8_2
20. Wang, X., et al.: Hybrid relation guided set matching for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 19948–19957 (2022)
21. Wu, J., Zhang, T., Zhang, Z., Wu, F., Zhang, Y.: Motion-modulated temporal fragment alignment network for few-shot action recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9151–9160 (2022)
22. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 305–321 (2018)
23. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P.H.S., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12350, pp. 525–542. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58558-7_31

24. Zheng, S., Chen, S., Jin, Q.: Few-shot action recognition with hierarchical matching and contrastive learning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV 2022. LNCS, vol. 13664, pp. 297–313. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-19772-7_18
25. Zhu, L., Yang, Y.: Compound memory networks for few-shot video classification. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 751–766 (2018)
26. Zhu, L., Yang, Y.: Label independent memory for semi-supervised few-shot video classification. IEEE Trans. Pattern Anal. Mach. Intell. **44**(1), 273–285 (2020)