Complex & Intelligent Systems (2024) 10:8249–8272
https://doi.org/10.1007/s40747-024-01571-4

ORIGINAL ARTICLE

# Hybrid attentive prototypical network for few-shot action recognition

Zanxi Ruan[1] · Yingmei Wei[1] · Yanming Guo[1] · Yuxiang Xie[1]

Received: 7 August 2023 / Accepted: 17 July 2024 / Published online: 19 August 2024

## Abstract

Most previous few-shot action recognition works tend to process video temporal and spatial features separately, resulting in insufficient extraction of comprehensive features. In this paper, a novel hybrid attentive prototypical network (HAPN) framework for few-shot action recognition is proposed. Distinguished by its joint processing of temporal and spatial information, the HAPN framework strategically manipulates these dimensions from feature extraction to the attention module, consequently enhancing its ability to perform action recognition tasks. Our framework utilizes the R(2+1)D backbone network, coupling the extraction of integrated temporal and spatial features to ensure a comprehensive understanding of video content. Additionally, our framework introduces the novel Residual Tri-dimensional Attention (ResTriDA) mechanism, specifically designed to augment feature information across the temporal, spatial, and channel dimensions. ResTriDA dynamically enhances crucial aspects of video features by amplifying significant channel-wise features for action distinction, accentuating spatial details vital for capturing the essence of actions within frames, and emphasizing temporal dynamics to capture movement over time. We further propose a prototypical attentive matching module (PAM) built on the concept of metric learning to resolve the overfitting issue common in few-shot tasks. We evaluate our HAPN framework on three classical few-shot action recognition datasets: Kinetics-100, UCF101, and HMDB51. The results indicate that our framework significantly outperformed state-of-the-art methods. Notably, the 1-shot task, demonstrated an increase of 9.8% in accuracy on UCF101 and improvements of 3.9% on HMDB51 and 12.4% on Kinetics-100. These gains confirm the robustness and effectiveness of our approach in leveraging limited data for precise action recognition.

**Keywords** Few-shot action recognition · Few-shot learning · Video understanding · Metric learning

## Introduction

Action recognition is an essential subtask in the field of video understanding, which aims to classify a video containing human actions [1, 2]. This task possesses significant potential and broad application value across various practical areas, such as robotic operations [3–6], public safety monitoring of violent behaviors [7], and traffic flow detection [8, 9]. Unlike tasks in the image domain, the main challenge in action recognition lies in the variability of human activities, which poses significant demands on how to deal with the video's temporal information. With the continuous improvement of large-scale datasets [10, 11] in the video field and the development of excellent deep-learning large models [12–14], the task of action recognition has witnessed notable advancements in recent years. However, regular action recognition tasks need significant quantities of annotated video data as task support, which is time-consuming and difficult to transfer to other dataset tasks. Few-shot action recognition occurs when the need to reduce reliance on large-scale datasets emerges.

The goal of the few-shot action recognition task is to precisely categorize videos without labels into defined video categories using just a tiny amount of data. The task of few-shot action recognition is exceedingly challenging because of the restricted quantity of available data. Existing methodologies for few-shot action recognition tasks can typically be classified into two distinct categories, one is based on gener-

✉ Yingmei Wei
weiyingmei@nudt.edu.cn

Zanxi Ruan
zanxiruan@nudt.edu.cn

Yanming Guo
guoyanming@nudt.edu.cn

Yuxiang Xie
yxxie@nudt.edu.cn

[1] Laboratory for Big Data and Decision, National University of Defense Technology, Changsha, Hunan, China

ation [15, 16], and the other is based on metric learning [17, 18]. The main objective of generation-based approaches is to enhance recognition accuracy by augmenting the dataset's sample size. The metric-based approach derives a sample point vector space by processing the features, and the distance between the query vector and the support vector determines the classification result [19, 20]. The idea of metric learning is used in the bulk of current mainstream approaches. For instance, the Spatial–Temporal Relation Model (STRM) [21] builds upon the foundation laid by the Temporal-Relation CrossTransformer (TRX) [22], enhancing it with a spatio-temporal enrichment module and a temporal relationship modeling module. These additions aim to augment feature representation, resulting in STRM achieving the current state-of-the-art performance in the domain. However, like many preceding approaches [17, 21–24], STRM relies on a 2D network as its core feature extraction network. The main limitation of 2D networks in video processing is that they handle spatial and temporal information separately. Even though STRM introduced Patch-Level Enrichment (PLE) and Frame-Level Enrichment (FLE) to improve temporal and spatial processing, these processes are still decoupled. PLE focuses on enhancing local patch features within a frame, which often fails to capture the dynamic progression of actions over time. The action unfolds over time, and merely enhancing patches within a frame can neglect the overarching temporal dynamics. On the other hand, FLE averages global frame features, potentially resulting in the loss of essential local spatiotemporal details. This decoupled approach to enhancement struggles to effectively integrate spatiotemporal information, thereby impeding the model's capacity to fully grasp the complexity and continuity of actions in videos. The widespread adoption of 2D networks in few-shot action recognition methods, which typically involve the separate extraction of features and a decoupled approach to processing spatiotemporal information, significantly restricts the ability to preserve the intrinsic continuity and complex dynamics of actions within videos. This limitation hampers the overall effectiveness and accuracy of these models.

In summary, current research in few-shot action recognition (FSAR) faces two primary challenges: **(1) Insufficient integration of information.** Traditional few-shot action recognition methods often process spatial and temporal information separately in videos, which fails to effectively capture the continuity and complexity of actions over time. **(2) Few-shot learning generalization issues.** Given the reliance on limited training samples typical of few-shot settings, models are prone to overfitting, making accurate recognition of unseen categories or actions challenging. In this research background, we propose the Hybrid Attentive Prototypical Network (HAPN).

Addressing the first challenge, HAPN employs the R(2+1)D backbone network, ingeniously integrating the extraction of spatial and temporal features. This design allows the network to delve deeply into the video content, precisely capturing the subtle complexities of actions across both time and space. Simultaneously, we introduce the Residual Tri-dimensional Attention mechanism, significantly bolstering our framework's capability to integrate spatial, channel, and temporal information comprehensively. ResTriDA adopts a unified processing strategy, treating spatial, channel, and temporal data as interconnected entities rather than isolated dimensions, fundamentally transforming action recognition. Within the spatial dimension, ResTriDA maps the context of each frame, spotlighting areas of interest crucial for identifying specific actions. Concurrently, it refines feature channels along the channel dimension. By embedding a temporal convolution layer into this architecture, ResTriDA extends its influence into the temporal dimension, empowering the model to comprehend the continuity and evolution of actions across frames. ResTriDA enables a deep understanding of videos by closely linking enhancements in space and channel aspects with time changes. This means it can catch the fine details and the flow of movements within videos, leading to a richer and more accurate analysis of actions. In response to the second challenge, to address the risk of overfitting associated with limited data samples in few-shot learning tasks, we introduce the Prototype Attention Matching (PAM) module. This module employs metric learning principles to enhance the model's ability to generalize across different action categories, leveraging inherent similarities and ensuring robust action recognition performance.

In general, the contributions of our research can be summarized as follows:

- We propose a novel Hybrid Attention Prototyping Network framework, which is highlighted by the joint processing of temporal and spatial information. This joint processing strategy starts from feature extraction and extends to the final attention module.
- We incorporate the ResTriDA module and the PAM module into our framework. The ResTriDA module adeptly amplifies feature information throughout the three-dimensional space, enriching the model's representational capacity. Concurrently, the PAM module skillfully navigates the constraints of limited sample sizes by leveraging inherent similarities across various action categories, thereby enhancing the model's ability to generalize and discern nuanced differences in actions.
- We evaluate HAPN on three classical few-shot action recognition datasets, Kinetics-100 [11], HMDB51 [25], and UCF101 [26]. Comprehensive experiments demonstrate that our model performs remarkably better than the state-of-the-art on both 1-shot and 5-shot tasks.

## Related work

In this section, we provide a concise overview of the research areas relevant to this paper, namely video understanding, prototype network, and few-shot action recognition.

### Video understanding

The research of video understanding has expanded significantly in recent years [13, 14, 27, 28]. Initially, utilizing convolutional neural networks for video understanding tasks does not yield better results than traditional methods relying on manual feature extraction because videos possess temporal information and 2D networks are not adept at effectively learning and utilizing this temporal information. To address the issue of processing timing, researchers propose the two-stream network [13] and the 3D network [14]. Two-stream network imports the optical stream to process the temporal features of the video, and the 3D network directly performs convolutional processing on the input time axis. However, the calculation takes time for both the two-stream and the 3D network. The R(2+1)D [28] network is derived from the 3D network. The R(2+1)D architecture splits the 3D convolutional layer into two parts: a 2D convolutional layer that handles spatial information, and a 1D convolutional layer that handles temporal information. This operation reduces the optimization difficulty of the network and achieves superior performance in video understanding. The development in video understanding marks a significant transition from the initial separate processing of spatial and temporal information to later, more integrated approaches aimed at comprehensively understanding video features. However, in the research of few-shot action recognition, most existing methods still rely on strategies that treat spatial and temporal features separately, not fully exploiting the potential value of the continuity in video data. Notably, although the R(2+1)D network has made significant progress in handling spatial and temporal information, its potential in scenarios requiring high data efficiency, such as few-shot action recognition, has not been fully explored. Future research should aim to further optimize the video understanding network to suit the needs of few-shot learning. Our work introduces the concept of integrated processing for continuous video into the task of few-shot action recognition. *By employing the R(2+1)D network as the feature extractor, we consider the joint extraction of spatial and temporal information right from the first step. Our method not only addresses the shortcomings of traditional methods, which typically handle spatial and temporal features in isolation but also offers a new perspective for exploration in the domain of few-shot action recognition.*

### Prototype network

The prototype network is a straightforward and efficient method for learning from few-shot samples. The goal of a prototype network is to learn a vector space to achieve a sample classification task [29, 30]. ProtoNet [19] is based on Matching Net [20], which uses a cosine function in the embedding space to measure the degree of the match after feature extraction. ProtoNet calculates the vector mean based on the Euclidean distance metric and then determines the distance from the test sample to each prototype. RelationNet [31] is also a prototype network, and the relationship module structure is utilized in place of the cosine and Euclidean distance metrics used in MatchingNet and ProtoNet. RelationNet employs a learnable nonlinear classifier to identify the relationship between sample points. Although traditional prototype networks offer a straightforward and efficient approach to few-shot learning, they struggle with the complexity of data features and variability within and between classes. These networks typically compute the distance between samples and prototypes directly in the feature space without fully leveraging the relational information among samples. Moreover, when handling temporal-spatial data such as videos, their ability to integrate temporal and spatial features is limited, potentially failing to capture subtle differences in actions or events. *In response to these issues, our Hybrid Attentive Prototypical Network framework introduces an innovative improvement with the Prototypical Attentive Matching module. By incorporating a multi-head self-attention mechanism, the PAM module enhances the model's ability to capture relationships between features, allowing it to consider the dynamic interactions among sample features when computing distances between samples and prototypes. PAM not only addresses the inadequacies of traditional prototype networks in handling variability within and between classes but also enhances the network's ability to integrate and utilize complex data features.*

### Few-shot action recognition

Action recognition is a branch task in video understanding [32]. In contrast to the standard action recognition task, the few-shot action recognition task only provides a tiny sample size for training [33]. Some approaches use generation-based ideas to carry out the task. For example, ProtoGAN [16] creates a conditional generative adversarial network by incorporating class prototype vectors to generate additional instances of novel classes. AmeFu-Net [15] mainly proposes introducing depth information as additional details regarding the scene to alleviate the problem of a severe lack of

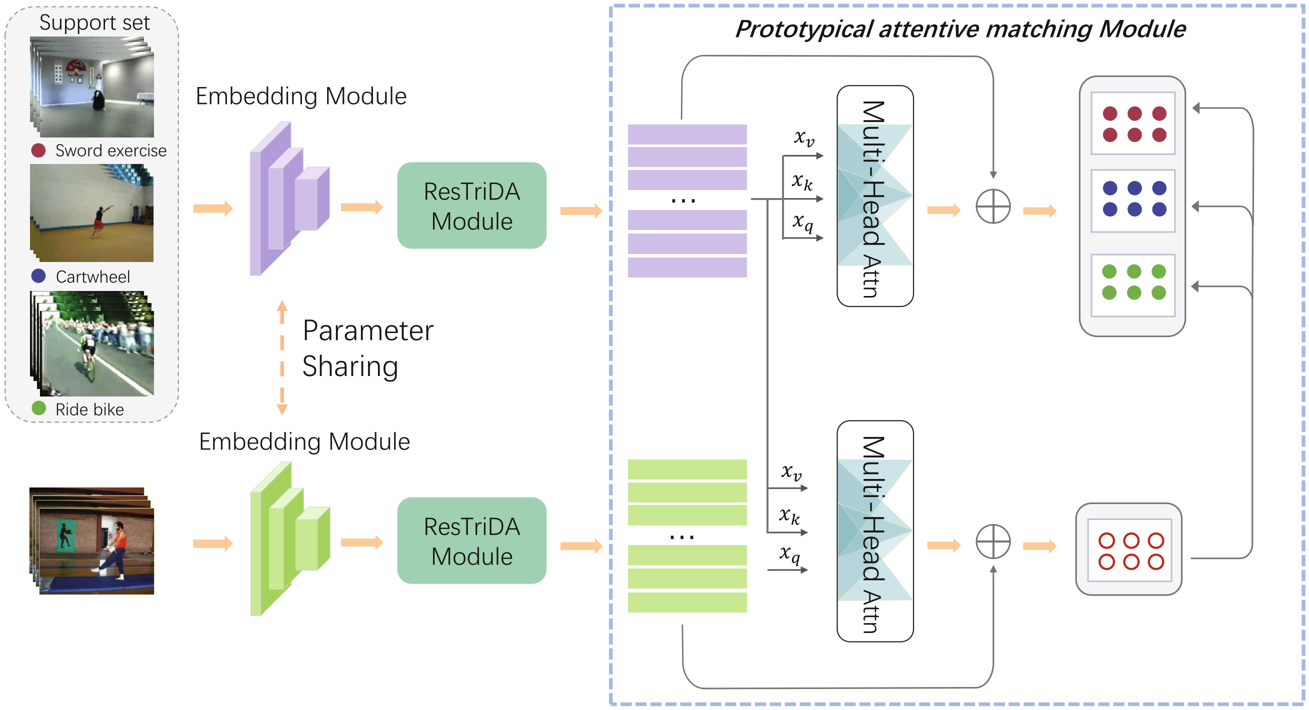


**Fig. 1** Schematic illustration of the proposed Hybrid Attentive Prototypical Network method for solving a 3-way 1-shot problem. The temporal feature embedding module analyzes the training video data, which consists of a support set (3-way 1-shot) and a query set, to extract the support features from the video data in the support set. The video frame data in the query set is also processed by the embedding module of the same structure to obtain the query feature. The ResTriDA module then enhances the spatial and temporal features. Then, before matching, we apply the standard multi-head self-attention layer to feature $x$, enhancing its representation. Finally, the prototype network acquires knowledge of a metric space where classification is accomplished by calculating the distance between samples in the query set and their corresponding prototype representations in the support set

annotation information under few-shot learning from a multimodal perspective. Most methods are still based on metrics, such as OTAM [23] using sparse sampling to get a sequence fragment and utilizing a temporal alignment to leverage the temporal information of features for metric matching. TRX [22] proposes a tuple base to process the temporal sequences separately while comparing the query and supporting video subsequences in a partial-based manner. STRM [21], similar to TRX [22], also utilizes a tuple base and uses the mlp-mixer to process the temporal sequences. HyRSM [17] adds an intra-relation function to the 2D network after extracting features to capture the intra-relation function to capture the temporal dependencies of video frames and proposes a new metric. Current few-shot action recognition methods typically process temporal and spatial features separately, overlooking the continuity and interdependence between time and space. This separation can lead to critical information loss, particularly detrimental in few-shot learning scenarios, impairing the model's ability to accurately recognize actions. Thus, future research should focus more on the integration of temporal and spatial features, drawing inspiration from the video understanding field's deep exploration of the continuous relationships in spatio-temporal data to explore more efficient ways of utilizing information. *Our approach is based on this insight, proposing an end-to-end joint coupling strategy. By employing the R(2+1)D network as our backbone for feature extraction, our model avoids the issue of information loss caused by separating temporal and spatial processing right from the start. Additionally, the hybrid attention mechanism we introduce integrates across spatial, temporal, and channel dimensions, ensuring comprehensive use of all available information at every step. This all-dimensional joint processing significantly enhances action recognition accuracy under few-shot conditions.*

## Method

In this section, we start with the problem description of few-shot action recognition. Then the proposed Hybrid Attentive Prototypical Network framework is introduced.

### Problem definition

Our task builds on the standard definition of few-shot action recognition [23, 34]. We divide the data set on this basis into a training set $D_{train}$, a validation set $D_{val}$, and a test set $D_{test}$. There is no data category overlap in any of these three sets. Few-shot action recognition can be considered an $N$ ways $K$ shots video classification problem. Using the training process as an example, we sample from the dataset $D_{train}$ according to the rules of episode training [35]. A support set $S$ is created by randomly picking $N$ data categories and sampling $K$ videos from each type. The query set $Q$ is formed by selecting $C$ videos from the remaining unsampled data of the selected $N$ categories. The goal of the few-shot action recognition task is to accurately classify a query set $Q\{q_1, \ldots, q_C\}$ without labels into one of the $N$ categories in the support set $S$.

## Overall architecture

Figure 1 illustrates the overall architecture of HAPN. To clearly illustrate our approach, we set $N = 3$, $K = 1$, and discuss only for a single query video. The support set $S\{s_1, \ldots, s_N\}$ contains $N$ kinds of video actions. We extract each video clip into $L$ video frames and use the video frames as input to the network. The temporal feature embedding module processes the video frames in the support set to obtain the support features $E(s_i)$. The video frame data in the query set also undergoes processing by the embedding module with the same structure to obtain the query feature $E(q)$. Note that the two embedding modules of the same structure share parameters and weights. Then, the features are processed by the ResTriDA module and the prototypical attentive matching module. The support set's features are mapped to the vector space $(D_1, cdots, D_N)$. The same mapping function is used to map the features in the query set to the vector space $G$. Finally, we calculate the distance between $G$ and $(D_1, \cdots, D_N)$. Then we select the category corresponding to the closest vector as the category of the query feature.

## Embedding module

The most significant difference between video action and standard picture data is that video incorporates temporal information. One of the important aspects and challenges in few-shot action recognition is extracting temporal information from video data. The majority of available few-shot action recognition methods extract video action features using 2D CNN [17, 21–24]. Although the method for feature extraction using 2D CNN is concise and convenient, it ignores the temporal elements of video actions, which causes many previous methods to add additional modules to analyze the temporal features, increasing the network's complexity. Some previous approaches [34, 36] use 3D CNN as a module for feature extraction, but they often suffer from the problems of slow network operation and insufficient recognition accuracy.

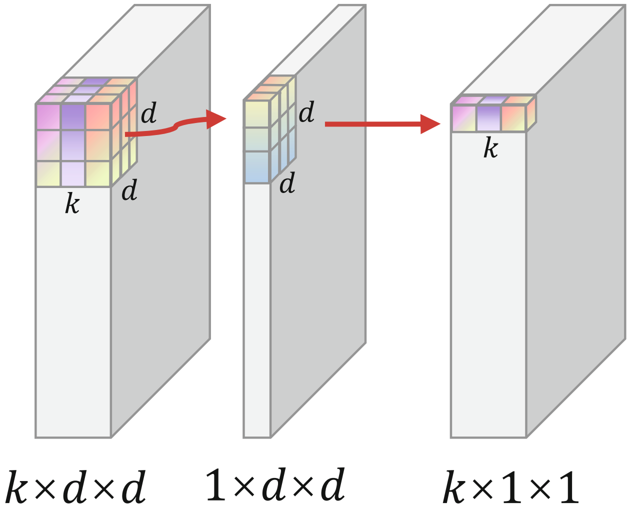


**Fig. 2** R(2+1)D main module structure. R(2+1)D is obtained by evolving based on 3D CNN. The complete 3D convolution of size $k \times d \times d$ in the 3D CNN is split into a 2D convolution of size $1 \times d \times d$ and a 1D convolution of size $k \times 1 \times 1$. The 2D convolution deals with the spatial information of the input data and the 1D convolution deals with the temporal information of the data

After considering the above issues, our embedding module adopts R(2+1)D [28]. As shown in Fig. 2, R(2+1)D is obtained by evolving based on 3D CNN. The goal is to divide the entire 3D convolution specification in 3D CNN into one 2D convolution and one 1D convolution. The spatial information of the input data is processed using 2D convolution, while the temporal information is processed using 1D convolution. The R(2+1)D convolution separates the processes, making optimizing the network for better outcomes easier. We obtain features with temporal information by utilizing R(2+1)D to extract the features.

## Residual tri-dimensional attention module

In this section, we dive into the ResTriDA module, which is an integral component of our proposed model. ResTriDA is designed to operate collaboratively to enhance spatial, channel, and temporal aspects of feature representations. This comprehensively enriching approach enables the network to concentrate specifically on the most important details in the input video features. ResTriDA acts as a single integrated unit that enhances spatial channel characterization while strengthening the spatio-temporal relationships in the video sequence. This holistic operation ensures a wider and more complex understanding of the video content, which greatly improves performance in action recognition tasks.

To more comprehensively extract spatial context information from videos and discern category differences across different spatial locations, we begin by integrating the Polarized Self-Attention mechanism (PSA) [37] into our model.

PSA serves to enhance feature extraction within both the spatial context and the channel of each video frame. However, acknowledging the limitation of PSA in capturing only spatial and channel-specific information, we further innovate by introducing ResTriDA. ResTriDA not only maintains the benefits of PSA but also integrates a temporal convolution operation, allowing us to capture crucial temporal dynamics within the video sequences. This innovative augmentation to the PSA transforms it into a tri-dimensional feature extractor, with robust capabilities across spatial, channel, and, critically, temporal dimensions. As a result, ResTriDA provides a more comprehensive view of the video content and also enables joint processing of the video information.

The processing outputs $E(s_i), E(q) \in \mathbb{R}^{B \times T \times C \times H \times W}$ obtained from the embedding module are given as the input to our innovative ResTriDA. Here, $B$ represents the size of the batch, $T$ represents the temporal dimension of the feature, $C$ represents the number of channels, and $H$ and $W$ represent the height and width of the feature, respectively.

As shown in Fig. 3, the ResTriDA module is a tripartite system, with distinct channel, spatial, and temporal branches. The channel and spatial branch are designed to compress information along one dimension while maintaining high-resolution features across the remaining orthogonal dimensions. This approach ensures that while the model condenses critical information to a more manageable form, it simultaneously preserves the granularity and richness of the data along other axes.

First, we introduce the Channel Attention Branch. This branch begins by transforming the input feature map, denoted as $\mathbf{x}$, through two different convolution layers, generating two sets of features $\mathbf{V}_{ch}$ and $\mathbf{Q}_{ch}$, which materialize as follows:

$$\mathbf{V}_{ch} = \mathcal{C}_{\theta}^{v}(\mathbf{x}), \mathbf{Q}_{ch} = \mathcal{C}_{\phi}^{q}(\mathbf{x}), \tag{1}$$

where $\mathcal{C}_{\theta}^{v}$ and $\mathcal{C}_{\phi}^{q}$ represent the convolutional operations with parameters $\theta$ and $\phi$. $\mathbf{V}_{ch}$ denotes the process of converting the input $\mathbf{x}$ by applying the convolutional layer, resulting in the extraction of features with half the number of channels. $\mathbf{Q}_{ch}$ represents the transformation of input $\mathbf{x}$ through the convolutional layer, compressing the number of channels to 1, which is used to compute the attention weights. Next, a Softmax operation is performed on the reshaped $\mathbf{Q}_{ch}$ to obtain the attentional weight $\Omega_{ch}$ on the channel.

$$\Omega_{ch} = \mathscr{S}\left(\mathcal{T}(\mathbf{Q}_{ch}, B, *)\right), \tag{2}$$

where $\mathcal{T}$ denotes the reshape transformation. $\mathscr{S}$ represents the Softmax function, the formula can be expressed in the following way:

$$\mathscr{S}(\mathbf{z})_i = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{K} \exp(\mathbf{z}_j)}, \tag{3}$$

here, $\mathbf{z}$ is a vector of real numbers. $K$ is the total number of elements in vector $\mathbf{z}$. $\exp(\cdot)$ denotes the exponential function.

Then reshape $\mathbf{V}_{ch}$ for matrix multiplication, which is noted as $\mathbf{Z}_{ch}$. The computed attention weights are then applied to $\mathbf{Z}_{ch}$ to obtain the weighted feature $\mathbf{A}_{ch}$. Finally, the weighted features are convolved and layer normalized, and then the final channel attention output $X_c$ is generated by the Sigmoid function $\sigma$. The above process can be represented as:

$$\mathbf{Z}_{ch} = \mathcal{T}(\mathbf{V}_{ch}, B, \frac{C}{2}, *), \tag{4}$$

$$\mathbf{A}_{ch} = \mathbf{Z}_{ch} \odot \mathbf{W}_{ch}, \tag{5}$$

$$\mathbf{X}_c = \sigma\left(\mathcal{LN}\left(\mathcal{C}_{\phi}^{q}\left(\mathbf{A}_{ch}\right)\right)\right). \tag{6}$$

The formula for the Sigmoid function $\sigma$ is:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{7}$$

In summary, the whole process of channel attention branching can be summarized in a more compact formula:

$$\mathbf{X}_c = \sigma\left(\mathcal{LN}\left(\mathcal{C}_{5\times5}\left(\mathcal{R}\left(\mathcal{C}_{wv}(\mathbf{x}) \odot \mathscr{S}\left(\mathcal{R}\left(\mathcal{C}_{wq}(\mathbf{x}), B, *, 1\right)\right),\right.\right.\right.\right.$$
$$\left.\left.\left.\left. B, \frac{C}{2}, H, W\right)\right)\right)\right). \tag{8}$$

Next, we introduce the Spatial Attention Branch. Based on the feature $\mathbf{X}_c$, the spatial attention branch further performs a sequence of operations similar to the channel branch but focusing on the spatial dimension.

$$\mathbf{V}_{sp} = \mathcal{C}_{\theta}^{v}(\mathbf{x}), \mathbf{Q}_{sp} = \mathcal{C}_{\phi}^{q}(\mathbf{x}), \tag{9}$$

where $\mathbf{V}_{sp}$ represents the feature $\mathbf{X}_c$ after transforming the channel attention processing through the convolutional layer $\mathcal{C}_{\theta}^{v}$, generating spatial features with half the number of channels. And $\mathbf{Q}_{sp}$ represents the base features for generating spatial attention weights by transforming $\mathbf{X}_c$ through the convolutional layer $\mathcal{C}_{\phi}^{q}$.

The spatial attention weight $\mathbf{W}_{sp}$ is then obtained by applying adaptive average pooling $\mathcal{G}$ and Softmax function to $Q_{sp}$.

$$\mathbf{W}_{sp} = \mathscr{S}\left(\mathcal{G}\left(\mathcal{T}(\mathbf{Q}_{sp}, B, *, 1)\right)\right). \tag{10}$$

The formula for $\mathcal{G}$ is as follows:

$$\mathcal{G}(\mathbf{Q}_{sp})_k = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{Q}_{sp}(i, j, k), \tag{11}$$

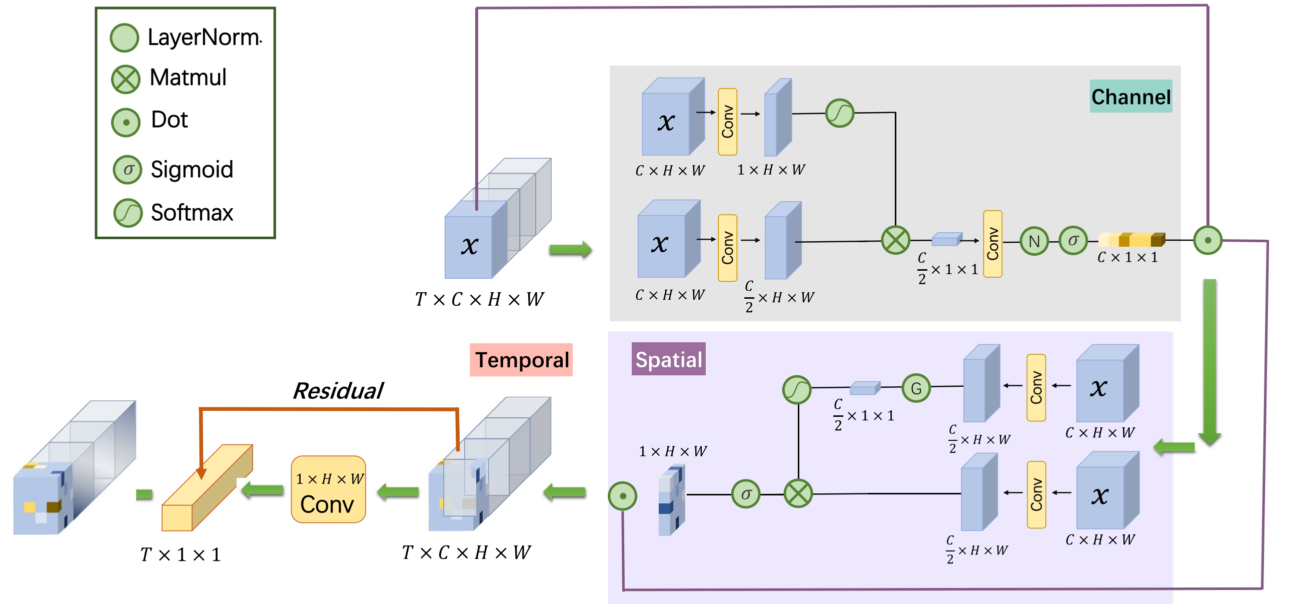


**Fig. 3** Residual tri-dimensional attention module. The ResTriDA module is composed of three branches, the channel branch, the spatial branch, and the temporal branch. Built upon the Polarized Self-Attention (PSA) mechanism, ResTriDA expands the feature extraction to a tri-dimensional level, covering spatial, channel, and temporal information. It not only retains the spatial and channel feature enhancement of PSA but also captures essential temporal dynamics in video sequences, offering a more holistic understanding of video content

where $\mathcal{G}(\mathbf{Q}_{sp})_k$ is the $k$-th element of the output of the adaptive average pooling operation, corresponding to the $k$-th channel. $\mathbf{Q}_{sp}(i, j, k)$ is the value at position $(i, j, k)$ in the input feature map $\mathbf{Q}_{sp}$. The sum is taken over all spatial locations $(i, j)$ for each channel $k$, and then divided by the total number of spatial locations $H \times W$ to compute the average.

Finally, spatial attention is applied and the formula is shown below:

$$\mathbf{A}_{sp} = \mathcal{T}\left(\mathbf{V}_{sp}, B, \frac{C}{2}, *\right) \odot \mathbf{W}_{sp}, \tag{12}$$

$$\mathbf{X}_s = \sigma\left(\mathcal{LN}\left(\mathcal{C}_{5\times 5}\left(\mathbf{A}_{sp}\right)\right)\right). \tag{13}$$

where $\mathbf{A}_{sp}$ is the weighted feature obtained by applying $\mathbf{W}_{sp}$ to $\mathbf{V}_{sp}$, a process that includes matrix multiplication and reshaping operations. The $\mathbf{X}_s$ is the final spatial attention output, obtained by convolution and layer normalization followed by Sigmoid function processing.

In short, the whole process of branching spatial attention can be represented by a complete formula:

$$\mathbf{X}_s = \sigma\left(\mathcal{LN}\left(\mathcal{C}_{5\times 5}\left(\mathbf{V}_{sp} \odot \mathbf{W}_{sp}\right)\right)\right). \tag{14}$$

The temporal residual branch in our model is a key component in capturing and enhancing temporal dynamics in video sequences. This branch operates on the spatially enhanced feature mapping $\mathbf{X}_s$ and applies a series of convolution and normalization operations to extract and refine temporal features. First, an initial spatial convolution is performed on $\mathbf{X}_s$:

$$\mathbf{Y}_{temp} = \mathcal{C}_{1\times d\times d}(\mathbf{X}_s), \tag{15}$$

this step focuses on the spatial dimension while keeping the temporal dimension constant.

Then layer normalization is performed and processed with the ReLU activation function:

$$\mathbf{Y}_{norm} = \mathcal{R}\left(\mathcal{LN}\left(\mathbf{Y}_{temp}\right)\right). \tag{16}$$

Next, temporal convolution is performed to emphasize the temporal dimension of the feature mapping, and layer normalization and ReLU activation are performed again to refine the temporal features further. The temporal information is then integrated using adaptive average pooling to obtain a compact and efficient representation of the temporal dynamics. The process mentioned above can be expressed as:

$$\mathbf{Y}_{final} = \mathcal{G}\left(\mathcal{R}\left(\mathcal{LN}\left(\mathcal{C}_{k\times 1\times 1}\left(\mathbf{Y}_{\text{norm}}\right)\right)\right)\right). \tag{17}$$

Finally, the original input of this branch $\mathbf{X}_s$ is added back to the output of the adaptive pooling step to establish a residual connection that mitigates the potential degradation in performance as the network depth increases. This connection facilitates the integration of original spatial features with the newly refined temporal features, thereby enriching the feature representation without losing the initial information. This step helps to preserve the original spatial channel information while integrating the newly extracted temporal features.

$$\mathbf{X}_{out} = \mathbf{X}_s + \mathbf{Y}_{final}. \tag{18}$$

The computation process of channel attention is similar to that of spatial attention. Channel attention is realized by utilizing two convolutional layers to create two sets of feature maps, and then using the Softmax function to compute the channel attention weights. Spatial attention is realized by applying similar convolution and attention mechanisms to the features processed by channel attention. The difference is that it uses adaptive global pooling to generate spatial attention weights. The channel attention branch focuses on which channels are more important, the spatial attention branch focuses on which regions in the image are more important, and the temporal residual branch focuses on which temporal information in the video is more important. These three mechanisms work together to make the network more efficient and accurate in processing complex visual information.

In conclusion, our ResTriDA module presents an innovative approach to attention mechanisms by catering to channel, spatial, and temporal dimensions, addressing the shortcomings of existing systems, and ultimately, providing superior feature recognition (Fig. 4).

## Prototypical attentive matching

General classification algorithms display overfitting in few-shot classification situations because of limited training data, resulting in large discrepancies between classification predictions and real results [38]. To mitigate the effect of overfitting caused by insufficient data, we employ a metric-based prototype network [19] to metrically classify the characteristics learned by the prior network which is called the prototypical attentive matching. We denote the feature obtained after processing by the above module as $x \in \mathbb{R}^{n \times d}$, where $n$ denotes the length of the input sequence and $d$ represents the hidden dimension. Before the feature vector is classified, we process the feature $x$ by using the standard multi-head self-attention layer [39], as shown in Fig . 4. We use the projection matrices $W_q$, $W_k$, $W_v$ of the three learnable parameters to compute the query, key, and value in the multi-head self-attention. The mapping relationship is expressed as:

$$x_q = W_q x, x_k = W_k x, x_v = W_v x. \tag{19}$$

After obtaining the output of the mapping of query, key, and value, the results $x_q$, $xk$, $x_v$ are further divided into $h$ equal parts according to the number of heads used. Each head calculates the self-attention separately and then concatenates the results. Then the spliced results are fused by the learnable parameter matrix $W^o$. After that, we get the attention scores between the samples and then concat the initial feature inputs and the attention scores. The above process can be expressed as Eqs. (20) and (21), where $d_k$ denotes the dimension of $x_k$, and $\mathscr{S}$ denotes the Softmax function.

$$Attn(x_q, x_k, x_v) = \mathscr{S}\left(\frac{x_q \cdot x_k^{\mathrm{T}}}{\sqrt{d_k}}\right) x_v, \tag{20}$$

$$x_{mha} = x + Concat(Attn(x_i), \ldots, Attn(x_h))W^o. \tag{21}$$

The multi-head self-attention module described above processes the query and support sets. It is worth mentioning that the support set provides key and value input during query set processing, which can be defined as:

$$x_q^{qry} = W_q x^{qry}, \quad x_k^{qry} = W_k x^{spt}, \quad x_v^{qry} = W_v x^{spt}. \tag{22}$$

After the treatment of multi-head attention, the prototype network acquires knowledge of a metric space $G$ in which the features in the query set can be classified by computing the cosine distance between the features in the query set and the corresponding prototype representations of the features in the support set of classes. The prototype representation is judged by the closest distance to classes. Specifically, the distance function $\mathcal{D}$ can be formed as:

$$\mathcal{D} = \frac{1}{\Pi} \sum \max \left| \frac{Q_G \bullet S_G}{\|Q_G\| \, \|S_G\|} \right|, \tag{23}$$

where $\Pi$ refers to the total number of sample points, $Q_G$ denotes the sample point in the query set, and $S_G$ denotes the sample points in the support set. The cross-entropy loss function [40] is used to calculate the loss of the network, where $\hat{y}$ is the predicted label and $y$ is the true label. The loss function $\mathcal{L}$ of the whole model can be simply formulated as:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{M} y_{ic} \log(\hat{y}_{ic}), \tag{24}$$

where $N$ represents the sample number in the dataset or batch and $M$ is the class number. $y_{ic}$ serves as a bool indicator (0 or 1) whether class label $c$ is correctly classified for observation $i$. $\hat{y}_{ic}$ is the predicted probability of observation $i$ being of class $c$.

Compared with current few-shot learning methods [15, 17, 18, 21, 23, 24, 34], our method reflects a more superficial induction bias that facilitates using such limited data ranges with excellent results. The main algorithm of HAPN is presented in Algorithm 1.

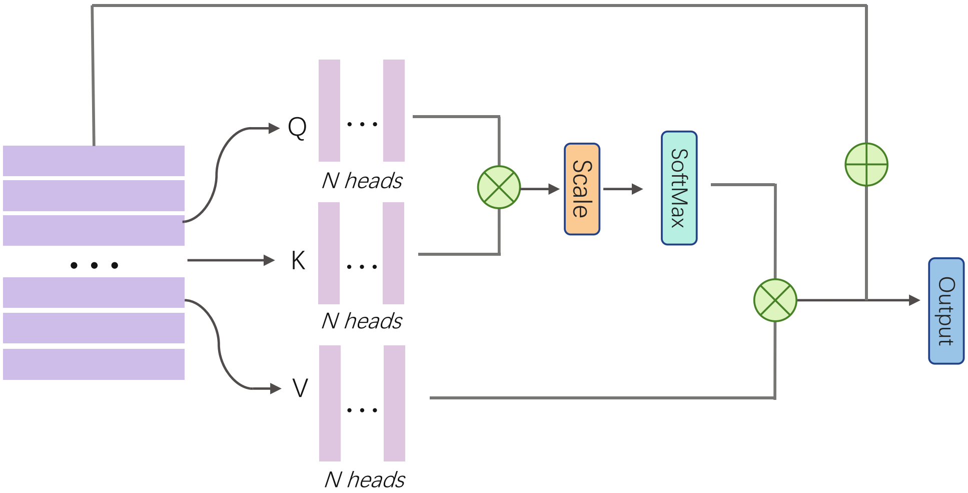


**Fig. 4** Multi-head self-attention (MHA) module. We use the projection matrices to compute the $Q$, $K$, and $V$ in the attention. After obtaining the output of the mapping of query, key, and value, the results are further divided into $h$ equal parts according to the number of heads used. Each head calculates the self-attention separately and then concatenates the results. Then the spliced results are fused by the learnable parameter matrix. After that, we obtain the attention scores between the samples and then concat the initial feature inputs and the attention scores

**Algorithm 1** Main algorithm of HAPN

```
Require: $S = \{s_1, \ldots, s_N\}$: Set of N video clips, each representing a different action category
Require: $Q$: A query video clip
Require: $\theta$: Parameters of R(2+1)D network
Require: $\alpha$: Learning rate for both task-level and meta-level updates
Initialize $\theta$ randomly
while not converged do
    Sample batch of tasks $\mathcal{T}_i$ from $p(\mathcal{T})$
    for each task $\mathcal{T}_i$ do
        $E(S) \leftarrow$ R(2+1)D$_\theta(S)$    ▷ Embed support set
        $E(Q) \leftarrow$ R(2+1)D$_\theta(Q)$    ▷ Embed query set
        ▷ Apply Residual Tri-Dimensional Attention (ResTriDA)
        $E'(S) \leftarrow$ ResTriDA$(E(S), \theta)$
        $E'(Q) \leftarrow$ ResTriDA$(E(Q), \theta)$
        ▷ Apply Prototypical Attentive Matching (PAM)
        $D_i \leftarrow$ VectorSpaceMapping$(E'(S))$    ▷ Map support features
        $G \leftarrow$ VectorSpaceMapping$(E'(Q))$    ▷ Map query features
        Category $\leftarrow \arg\min_i \mathcal{D}(G, D_i)$    ▷ Assign category to query
        Evaluate gradient $\nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta)$
        Adapt $\theta$ using gradient descent: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}_i}(\theta)$
    end for
    Update $\theta$ using aggregated gradients: $\theta \leftarrow \theta - \alpha \nabla_\theta \sum_{\mathcal{T}_i} \mathcal{L}_{\mathcal{T}_i}(\theta)$
end while
```

## Experiments

### Datasets

To validate the effectiveness of our method, experiments are conducted on three popular standard datasets: Kinetics-100 [11], HMDB51 [25], and UCF101 [26]. These datasets are widely recognized benchmarks within the few-shot action recognition field and are particularly well-suited for validating the effectiveness of our method for several reasons. Firstly, Kinetics-100, derived from the comprehensive Kinetics-400, presents a distilled challenge with a varied assortment of action classes, each represented by 100 videos. This balanced composition enables a robust assessment of our model's ability to learn and generalize from a consistent number of examples per class. Secondly, the HMDB51 dataset, with its 51 action classes spanning 6849 videos, is valued for its complex, real-world action scenarios. This dataset challenges our HAPN framework to accurately discern subtle nuances of human actions across a breadth of less controlled, more naturally occurring settings. Lastly, UCF101 is included for its extensive volume and diversity of actions, encapsulating a broad spectrum of human activities and presenting varied scenarios in terms of lighting, background, and camera angles. This variability is crucial to evaluate the adaptability and robustness of the hybrid attention mechanism at the core of HAPN, ensuring that the framework is not only learning specific patterns but also adapting to different visual contexts. We utilize the dataset splits proposed by CMN [33] for Kinetics-100 and ARN [34] for both HMDB51 and UCF101 to ensure that our evaluation is grounded in a recognized and reproducible experimental setup. The non-overlapping nature of the classes in the training, validation, and test sets across these datasets further guarantees that our evaluation is stringent and that the model's performance truly reflects its generalization capabilities to unseen data.

### Details of implementation

Following the episode training strategy [19, 35], we employ the pre-trained R(2+1)D-18 [28] as the model's backbone network. We employ a random sampling technique to select 8 frames from each video, which are then used as input to the network. Each frame is resized to 112 × 112, cropped, and subjected to random brightness and contrast adjustments. We employ a 70-epoch training assignment and train 200 episodes in each epoch cycle. It is worth noting that after each training epoch, the model is validated with 200 validation rounds. Before the feature data enters the ResTriDA module, we set a random dropout with a parameter of 0.1, mindful of the delicate balance between overfitting and underfitting risks inherent in few-shot tasks. Overfitting restricts the model's ability to generalize beyond the training data, while underfitting limits its capacity to capture underlying data patterns. A high dropout rate risks further underfitting, reducing the learnable information, and possibly degrading training performance. Thus, a lower dropout rate of 0.1 helps retain sufficient model capacity for learning from limited data, ensuring an optimal balance between model complexity and generalizability.

Regarding the learning rate, we employ distinct settings for the Kinetics-100 dataset versus UCF101 and HMDB51 to address the varying characteristics and challenges posed by these datasets. For UCF101 and HMDB51, where the datasets include a broader range of simpler and more complex motions, a lower learning rate of 0.0001 is utilized. This conservative approach helps in navigating the optimization landscape smoothly, avoiding the pitfalls of rapid convergence to suboptimal local minima, which is crucial given the diverse and noisy nature of these datasets. In contrast, Kinetics-100, being a subset of a larger dataset, is more homogeneous and has been structured to focus on specific types of actions. Therefore, a higher learning rate of 0.1, combined with a decay factor of 0.1, is appropriate as it allows for faster convergence without bypassing the global minimum.

The SGD optimizer is adopted to optimize the network, decaying the learning rate every 10 epochs. In the matching classification stage, we find that using the cosine function to calculate the distance between the query set and support set can make the model achieve optimal performance. We train HAPN on three Tesla P100 GPUs.

## Experimental results

We conduct a series of experiments on three different datasets, UCF101, HMDB51, and Kinetics-100, and thoroughly evaluate the model performance under the settings of 5-way 1-shot, 5-way 3-shot, and 5-way 5-shot. Here, 5-way means that each round of testing involves 5 different categories, while 1-shot, 3-shot, and 5-shot refer to the number of instances—1, 3, and 5-used for testing each category, respectively. To fully evaluate the model performance, we record six key performance metrics: Accuracy, Recall, Precision, F1 Score, AUC, and mAP.

Taking the 5-way 1-shot task as an example, assume that the model generates a probability matrix $P$ for $N$ query samples, where $P_{i,j}$ represents the predicted probability that the $i$-th query sample belongs to the $j$-th class. First, define $Y$ as the true label vector, where $Y_i$ indicates the true class index of the $i$-th query sample. Let $\hat{Y}$ represent the predicted label vector, where $\hat{Y}_i = \arg\max_j P_{i,j}$, indicating that the model predicts the $i$-th query sample is most likely to belong to the class index $j$. First, we calculate the Accuracy metric. Accuracy directly reflects the model's correct prediction rate across all query samples, serving as a key indicator of the model's generalization ability. The formula for calculating accuracy is as follows:

$$\text{Accuracy} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{Y}_i = Y_i], \tag{25}$$

where $\mathbf{1}[\cdot]$ is the indicator function that takes the value 1 when the predicted class matches the true class, and 0 otherwise. The recall for class $j$, $R_j$, measures the proportion of query samples correctly predicted as class $j$ against the total number of query samples that actually belong to class $j$. The macro-averaged recall $R_{macro}$ provides an overall performance indicator:

$$R_j = \frac{\sum_{i=1}^{N} \mathbf{1}[\hat{Y}_i = j \wedge Y_i = j]}{\sum_{i=1}^{N} \mathbf{1}[Y_i = j]}. \tag{26}$$

$$R_{macro} = \frac{1}{5}\sum_{j=1}^{5} R_j. \tag{27}$$

For class $j$, precision $P_j$ is defined as the ratio of the number of query samples correctly predicted as class $j$ to the total number of query samples predicted as class $j$. The macro-averaged precision $P_{macro}$, evaluates the model's overall performance across all classes:

$$P_j = \frac{\sum_{i=1}^{N} \mathbf{1}[\hat{Y}_i = j \wedge Y_i = j]}{\sum_{i=1}^{N} \mathbf{1}[\hat{Y}_i = j]}. \tag{28}$$

$$P_{macro} = \frac{1}{5}\sum_{j=1}^{5} P_j. \tag{29}$$

The F1 score combines precision and recall through their harmonic mean for each class $j$. The macro-averaged F1 score $F1_{macro}$ averages the F1 scores across all classes:

$$F1_j = 2 \times \frac{P_j \times R_j}{P_j + R_j}. \tag{30}$$

$$F1_{macro} = \frac{1}{5}\sum_{j=1}^{5} F1_j. \tag{31}$$

For each class $j$ in the 5-way classification setup, the task is treated as a binary classification problem to calculate the area under the Receiver Operating Characteristic (ROC) curve, denoted as $AUC_j$. This metric assesses the model's ability to distinguish between class $j$ (positive class) and all other classes (negative class). The average AUC, which provides a summary measure of the model's binary classification per-

**Table 1** Multi-metric performance analysis of the HAPN method

| Dataset | Setting | Accuracy (%) | Recall | Precision | F1 score | AUC | mAP |
|---|---|---|---|---|---|---|---|
| UCF101 | 5-way 1-shot | 93.1 | 0.931 | 0.929 | 0.930 | 0.992 | 0.977 |
| | 5-way 3-shot | 96.6 | 0.966 | 0.966 | 0.966 | 0.998 | 0.994 |
| | 5-way 5-shot | 97.9 | 0.979 | 0.981 | 0.980 | 0.999 | 0.996 |
| HMDB51 | 5-way 1-shot | 64.2 | 0.642 | 0.644 | 0.643 | 0.874 | 0.712 |
| | 5-way 3-shot | 74.0 | 0.739 | 0.740 | 0.740 | 0.941 | 0.839 |
| | 5-way 5-shot | 77.9 | 0.775 | 0.778 | 0.776 | 0.937 | 0.853 |
| Kinetics-100 | 5-way 1-shot | 86.1 | 0.872 | 0.873 | 0.873 | 0.978 | 0.933 |
| | 5-way 3-shot | 94.2 | 0.942 | 0.942 | 0.941 | 0.994 | 0.980 |
| | 5-way 5-shot | 97.4 | 0.974 | 0.974 | 0.974 | 0.997 | 0.982 |

Experiments are conducted on three mainstream video recognition datasets, UCF101, HMDB51 and Kinetics-100, and record six evaluation metrics

formance across all classes, is computed as follows:

$$AUC_{avg} = \frac{1}{5}\sum_{j=1}^{5} AUC_j. \tag{32}$$

The mean average precision (mAP) for each class $j$ is calculated by averaging precision across different recall thresholds. The model's mAP is the average of the APs across all classes:

$$mAP = \frac{1}{5}\sum_{j=1}^{5} AP_j. \tag{33}$$

These detailed metrics enable a comprehensive understanding of the model's performance.

The experimental results for the HAPN model are shown in the Table 1. For all three datasets, UCF101, HMDB51, and Kinetics-100, we observe a common trend: as the number of shots in the experimental setup increases, the model shows a significant improvement in almost all performance metrics. This phenomenon indicates that increasing the number of samples can significantly improve the performance of the model. In particular, HAPN performs well on the UCF101 and Kinetics-100 datasets, compared to the slightly inferior performance of HMDB51. This difference may stem from the characteristics of the dataset itself, such as the higher diversity of samples in HMDB51 and different lighting conditions, which increase the difficulty of identification.

In addition, we note that both AUC and mAP values are relatively high regardless of dataset or experimental setup, especially on Kinetics-100 and UCF101, suggesting that the HAPN model has a strong overall classification ability on these datasets.

It is worth mentioning that experimental results also indicate that the values of the metrics Accuracy, Recall, F1 Score, and Precision are almost the same, which means that the HAPN model achieves a balanced performance in recognizing the positive classes (True Positives) and avoiding the misclassification of the negative classes (False Positives) and avoiding misclassification of negative classes (False Positives). Ideally, an efficient model should have high precision-reducing misclassification of negative samples as well as high recall-being able to accurately identify most of the positive samples. HAPN is just such a model that performs well with the different categories shows good balance and no significant bias, nor does it exhibit any extreme performance bias. Although we discuss multiple metrics in this section to thoroughly understand the model's multi-faceted performance, previous FSAR methods mainly used accuracy as the sole performance indicator. Therefore, to maintain a fair comparison with existing literature, most of our subsequent experiments primarily focus on accuracy as the benchmark metric.

## Robustness analysis

To assess the robustness of HAPN, various robustness experiments under the HMDB51 dataset using the 5-way 1-shot setting, using classification accuracy as a comparison metric are constructed. Figure 5 accurately shows the fluctuation of the model's accuracy performance under a series of condition changes. Under standard environmental conditions, the HAPN model demonstrates a baseline accuracy of 64.2%, providing a solid foundation for subsequent comparisons.

To verify the adaptability and stability of the model under complex lighting conditions, a series of simulation experiments are specifically designed and implemented. First, we simulate scenarios of bright lighting, where the intensity of light far exceeds that of the standard environment. The results indicate that under bright light conditions, the accuracy of the model decreases slightly to 62.80%. This minor decline suggests that although strong light can interfere with image features, the model has a certain capacity to resist intense

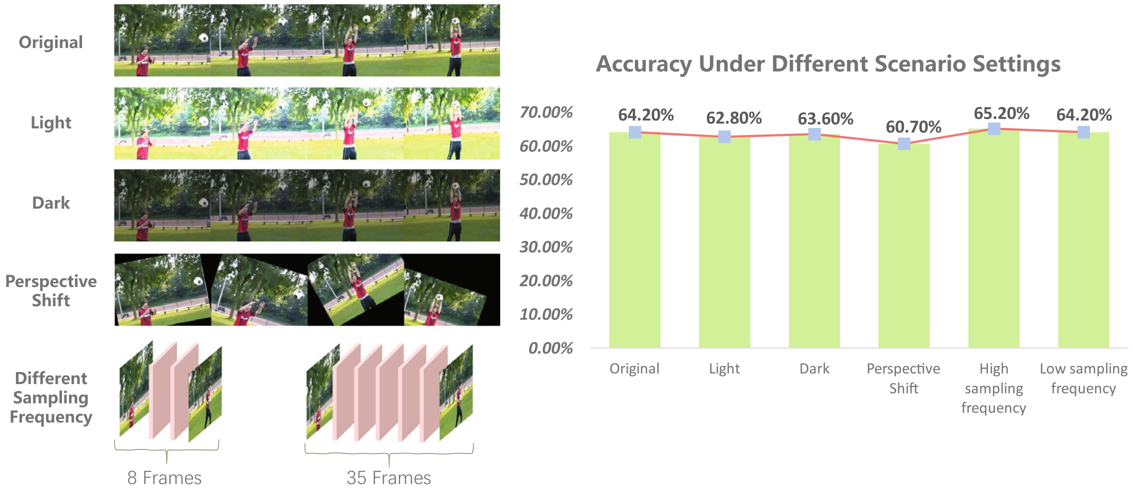


**Fig. 5** Comparison of recognition accuracy in different scenarios. We construct robustness experiments under the HMDB51 dataset using the 5-way 1-sho t setting. The HAPN model maintains a stable recognition in different scenarios

light disturbance, managing to suppress noise from overexposure and thus maintaining stable recognition performance. Subsequently, we simulate low-light environments, with illumination intensity falling below standard levels. The results show a slight increase in accuracy to 63.60% under low-light conditions. This implies that the HAPN model retains inherent robustness in extracting action features from video content, capable of adapting to feature extraction even with insufficient lighting.

In practical applications, such as surveillance cameras and autonomous vehicles, the actions captured in videos often face random changes in perspective due to factors like the positioning of the devices and the trajectories of moving targets, posing significant challenges for action recognition tasks. To investigate the robustness of our model against such scenarios of perspective transformation, we subject video frames to random rotations and affine transformations, simulating these complex conditions. The experimental result shows that even under such extreme and challenging conditions of perspective distortion, the accuracy of the model only drops slightly but still maintains a level of 60.70%. Although random changes in perspective indeed increase the difficulty of recognition, the model does not experience a significant decline in performance, effectively resisting the adverse effects brought about by perspective distortion.

To comprehensively evaluate our model's adaptability to changes in action speed, we design experiments that vary the sampling frequency of video frames. During the experiments, we use a high sampling frequency of 35 frames to simulate fast actions, incorporating more motion details and shorter intervals within the same timeframe, to test whether the model can accurately capture the rapidly changing information stream. The results show that at this high sampling frequency, the model maintains an accuracy of 65.20%, demonstrating its ability to keep a high recognition efficiency in fast dynamic scenes and effectively address the challenges brought by increased action speeds. On the other hand, we also reduce the video frame sampling frequency to 8 frames to simulate slow-motion scenarios, where each action unit spans a longer duration, placing higher demands on the model's long-term dependencies and understanding of action coherence. Under this low sampling frequency condition, the model's accuracy stabilizes at 64.20%, close to the accuracy under the original baseline conditions. Combining the experimental results under these two conditions, we observe that the model's accuracy fluctuates only slightly, exhibiting good robustness. This indicates that whether the action speed is fast or slow, the model can adapt well and maintain stable recognition performance. However, it is important to note that most current mainstream methods typically sample only 8 frames. To make a fair comparison and improve experimental efficiency, we also adopt the same frame sampling number in our experiments.

In summary, after a series of detailed and in-depth experimental analyses, we have observed that although changes in external environmental conditions such as light intensity, perspective variations, and the speed of action impact the performance of the model to varying degrees, the overall performance of the model still demonstrates strong adaptability and robustness.

## Comparison with state-of-the-art

Our model is evaluated in comparison to seven excellent algorithms, and our method demonstrates outstanding performance on three selected datasets, achieving state-of-the-art performance for each setting. As shown in Table 2, we highlight in red the improvement over the state-of-the-art

**Table 2** Accuracy comparison with state-of-the-art few-shot action recognition approaches on three standard datasets

| Method | Reference | UCF101 | | HMDB51 | | Kinetics-100 | |
|---|---|---|---|---|---|---|---|
| | | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| ProtoGAN [16] | *ECCV2018* | 57.8 ± 3.0 | 80.2 ± 1.3 | 34.7 ± 9.2 | 54.0 ± 3.9 | – | – |
| ARN [34] | *ECCV2020* | 62.1 ± 1.0 | 84.8 ± 0.8 | 44.6 ± 0.9 | 59.1 ± 0.8 | 63.7 | 82.4 |
| OTAM [23] | *CVPR2020* | 79.9 | 88.9 | 54.5 | 68.0 | 73.0 | 85.8 |
| HF-AR [24] | *IJCAI2021 Workshop* | 58.6 ± 1.2 | 86.4 ± 1.6 | 43.4 ± 0.6 | 62.2 ± 0.9 | – | – |
| TRX [22] | *CVPR2021* | 78.2 | 96.1 | – | 75.6 | 63.6 | 85.9 |
| HyRSM [17] | *CVPR2022* | 83.9 | 94.7 | 60.3 | 76.0 | 73.7 | 86.1 |
| STRM [21] | *CVPR2022* | – | 96.9 | – | 77.3 | – | 86.7 |
| **HAPN** | – | **93.1 (+ 9.8)** | **97.9 (+ 1.0)** | **64.2 (+ 3.9)** | **77.9 (+ 0.6)** | **86.1 (+ 12.4)** | **97.4 (+ 10.7)** |

The experiments are conducted in a 5-way setup and report results under 1-shot and 5-shot tasks. Our method outperforms the previous state-of-the-art methods. The underline indicates the best results of the previous methods in the available records

Bold font represents the best result

method. In the 1-shot setting on the UCF101 dataset, the HyRSM model previously achieved the best performance with an accuracy of 83.9%. Our HAPN model surpasses this, achieving an accuracy of 93.1% in the same setting, marking a significant increase of 9.8%. In the 5-shot setting on UCF101, HAPN outperforms the best-performing STRM model with an accuracy of 97.9%, an improvement of 1.0%. On the HMDB51 and Kinetics-100 datasets, HAPN similarly demonstrates exceptional performance, further affirming the effectiveness and generalizability of our model. The ARN [34] in Table 2 also uses a 3D network to extract features, but its subsequent temporal attention and spatial attention still treat the features separately. Our proposed hybrid attention module jointly enhances feature information across multiple dimensions. We incorporate an attention mechanism for selectively sampling points to learn inter-class relationships before the prototype network classifies, resulting in improved classification performance. Note that our model performs exceptionally well on Kinetics-100 because it uses the official R(2+1)D model provided by PyTorch that is pre-trained on Kinetics-400. Our model demonstrates superior performance on the 1-shot task, showing that it can effectively identify with a limited set of examples.

## Ablation study

### Impact of the backbone

ResNet [41] is the backbone of most current few-shot action recognition algorithms [17, 21, 22]. To investigate the influence of various backbone networks on the final recognition results, we replace the backbone network of the model. No gain modules are added for the experiments. As shown in Table 3, We can observe that the performance of all three datasets is superior when using a ResNet50 backbone compared to a ResNet18 backbone. However, the accuracy of model recognition is lower than when the backbone is R(2+1)D [28] network. As an example, consider the UCF101 dataset results. The discrepancy can be explained by the presence of temporal correlations in videos, which the ResNet architecture does not take into account when processing information. Previous studies [21, 22] have also attempted to address this limitation by incorporating additional methods for processing temporal information. In contrast, R(2+1)D includes a processing step for the temporal information, which extracts features from the input information in time and space in one step.

**Table 3** Performance comparison when varying the backbone

| Method | Backbone | 1-shot | 5-shot |
|---|---|---|---|
| UCF101 | ResNet18 | 61.9 | 81.5 |
| | ResNet50 | 68.5 | 83.1 |
| | **R(2+1)D** | **72.9** | **90.7** |
| HMDB51 | ResNet18 | 33.8 | 52.5 |
| | ResNet50 | 35.0 | 54.4 |
| | **R(2+1)D** | **40.9** | **59.3** |
| Kinetics-100 | ResNet18 | 41.4 | 64.2 |
| | ResNet50 | 45.7 | 69.7 |
| | **R(2+1)D** | **54.5** | **76.1** |

We conduct the experiment without incorporating any gain modules

Bold font represents the best result

**Table 4** Impact of the proposed components on HMDB51

| ResTriDA | MHA | 1-shot | 5-shot |
|---|---|---|---|
| ✗ | ✗ | 40.9 | 59.3 |
| ✗ | ✓ | 41.6 | 61.2 |
| ✓ | ✗ | 61.5 | 77.3 |
| ✓ | ✓ | **64.2 (+ 23.3)** | **77.9 (+ 18.6)** |

The results of the ablation experiments show the effectiveness of the components presented in our HAPN

Bold font represents the best result

**Table 5** Impact of the ResTriDA module parameters

(a) Activation function ablation

| Branch | Activation function | Top-1 accuracy (%) | Top-3 accuracy (%) |
|---|---|---|---|
| Channel&Spatial | ReLU | 27.60 | 68.30 |
| | LeakyReLU | 33.30 | 75.30 |
| | PReLU | 24.20 | 64.50 |
| | ELU | 32.10 | 71.60 |
| | Tanh | 33.60 | 72.00 |
| | Sigmoid | 61.20 | 89.20 |
| Temporal | ReLU | 61.20 | 89.20 |
| | LeakyReLU | 47.85 | 82.11 |
| | PReLU | 39.24 | 73.07 |
| | ELU | 28.55 | 69.43 |
| | Tanh | 37.10 | 75.80 |
| | Sigmoid | 44.83 | 79.50 |

(b) Kernel size ablation

| Branch | Kernel size | Top-1 accuracy (%) | Top-3 accuracy (%) |
|---|---|---|---|
| Channel&Spatial | 1 × 1 | 59.20 | 89.20 |
| | 2 × 2 | 54.00 | 86.40 |
| | 3 × 3 | 57.20 | 88.00 |
| | 5 × 5 | 61.20 | 89.20 |
| | 7 × 7 | 53.60 | 84.80 |
| Temporal | 1 × 1 | 61.20 | 89.20 |
| | 2 × 2 | 60.00 | 87.30 |
| | 3 × 3 | 55.80 | 87.50 |
| | 5 × 5 | 55.10 | 86.10 |
| | 7 × 7 | 51.70 | 78.33 |

(c) Normalization ablation

| Branch | Normalization | Top-1 accuracy (%) | Top-3 accuracy (%) |
|---|---|---|---|
| Channel&Spatial | Non | 57.60 | 86.00 |
| | LayerNorm | 61.20 | 89.20 |
| | InstanceNorm | 60.80 | 88.70 |
| Temporal | Non | 53.10 | 75.80 |
| | LayerNorm | 61.20 | 89.20 |
| | InstanceNorm | 59.50 | 85.40 |

(d) Pooling ablation

| Branch | Pooling | Top-1 accuracy (%) | Top-3 accuracy (%) |
|---|---|---|---|
| Channel&Spatial | AdaptiveAvgPool | 61.20 | 89.20 |
| | AdaptiveMaxPool | 60.00 | 88.40 |
| Temporal | AdaptiveAvgPool | 61.20 | 89.20 |
| | AdaptiveMaxPool | 59.30 | 86.72 |

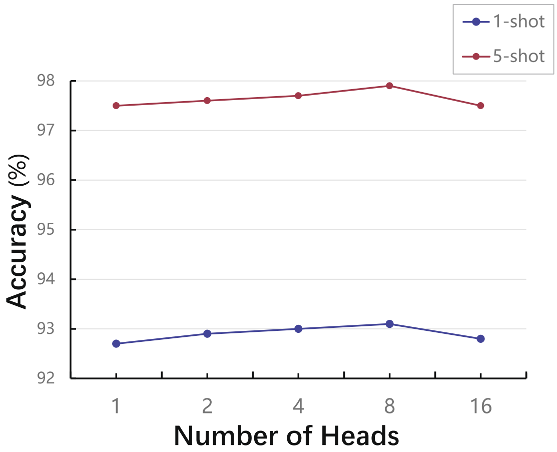
(e) Attention branch ablation

| Attention branch | Top-1 accuracy (%) | Top-3 accuracy (%) |
|---|---|---|
| Channel only | 46.40 | 79.60 |
| Spatial only | 48.80 | 83.60 |

**Table 5** continued

| (e) Attention branch ablation | | |
|---|---|---|
| Attention branch | Top-1 accuracy (%) | Top-3 accuracy (%) |
| Temporal only | 51.20 | 84.27 |
| Channel + spatial | 49.20 | 82.10 |
| Channel + temporal | 52.30 | 83.00 |
| Spatial + temporal | 56.00 | 85.20 |
| ResTriDA (no residual) | 58.30 | 86.50 |
| ResTriDA | 61.20 | 89.20 |

We investigate the impact of various parameters on the ResTriDA module, with the underline indicating the optimal results. The experiments use the HMDB51-Small dataset with a 5-way 1-shot setting



| Heads | 1-shot | 5-shot |
|---|---|---|
| 1 | 92.7 | 97.5 |
| 2 | 92.8 | 97.6 |
| 4 | 93 | 97.7 |
| 8 | 93.1 | 97.9 |
| 16 | 92.8 | 97.5 |

**Fig. 6** Impact of the head number on UCF101. Experiment results demonstrate that the model performs optimally when utilizing eight attention heads

### Impact of the proposed components

Here, The impact of the residual tri-dimensional module and the prototypical attentive matching module on recognition performance are analyzed. The prototypical attentive matching module contains both multi-head attention [39] and prototypical matching modules. Since the prototypical matching module involves the final classification of the network and cannot be removed separately for the experiments, we only chose multi-head attention (MHA) for the ablation experiments. The ablation experiments of the components are performed based on HMBD51. The base backbone of the network is a prototype R(2+1)D network. As shown in Table 4, the results show that each of our proposed components is effective. In particular, the residual tri-dimensional module has an extremely significant gain effect on the network. This is because the action recognition task is highly dependent on temporal information. Our ResTriDA module precisely processes the temporal information and enhances spatial and channel features. We observe that jointly processing temporal and spatial features can substantially increase the recognition accuracy. The final HAPN framework achieves 23.3% higher recognition accuracy than the baseline under 1-shot and 18.6% higher under 5-shot. Our proposed components greatly improve recognition accuracy.

### Impact of the ResTriDA module parameters

In this section, exhaustive parameter ablation experiments on the ResTriDA module are conducted to explore in depth the specific effects of various parameters on the performance of the module. The results of the experiment are shown in Table 5. To optimize the computational efficiency, and consider the similarity between the Channel branch and the Spatial branch in processing spatial context information, we choose to merge these two branches for the ablation experiments. We uniformly extract 30% of the data in the original HMDB51 dataset to form the HMDB51-Small dataset, on which we construct the experiments. For performance estimation, we adopt Top-1 Accuracy and Top-3 Accuracy as the key metrics. Top-1 Accuracy reflects the accuracy of the model in predicting the most probable categories, while Top-3 Accuracy evaluates the probability that the model contains the true categories among the top three most probable categories in its prediction.

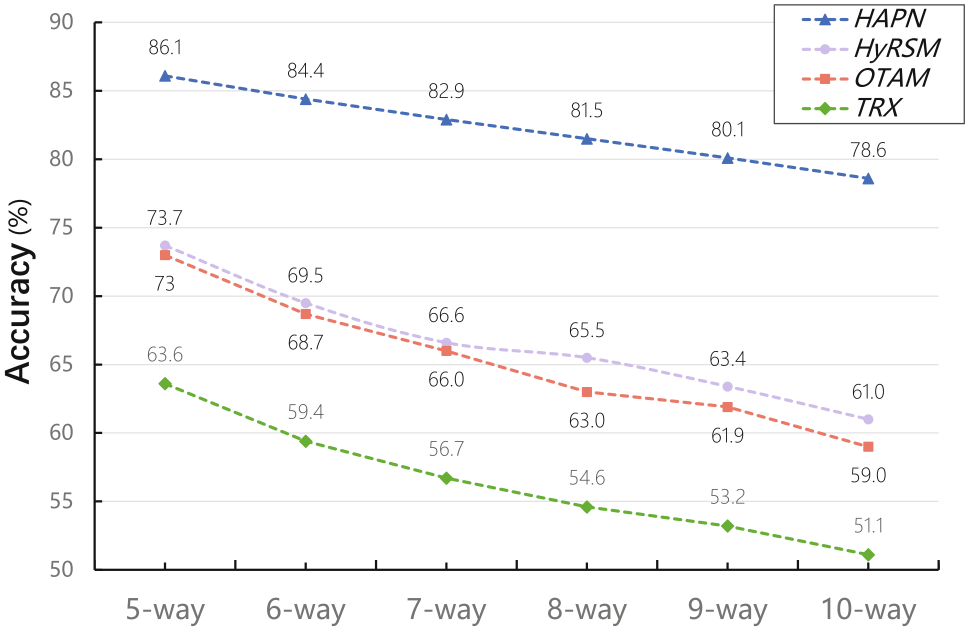


**Fig. 7** Comparison at N-way task on Kinetics-100. Through a series of experiments conducted under N-way 1-shot conditions, we demonstrate that our model consistently outperforms other algorithms in recognition accuracy, thereby showcasing the effectiveness of our approach

The 5 × 5 convolutional kernel size in the Channel& Spatial branch achieves the highest performance, showing that a slightly larger convolutional kernel can strike a good balance between capturing detail and preserving context when dealing with spatial-level features. The 1x1 convolutional kernel size in the Temporal branch achieves the best performance, suggesting that a smaller convolutional kernel is more effective when dealing with temporal-series data, and can acutely capture subtle changes in the action.

Finally, through ablation experiments with the attention branch, we found that the highest accuracy can be achieved using the full ResTriDA branch containing the channel, spatial, and temporal attention mechanisms. In particular, there is a notable enhancement in model performance with the incorporation of the Temporal branch, highlighting the importance of temporal features in few-shot action recognition tasks. These findings not only confirm the necessity of integrating multiple attentional mechanisms to improve model performance but also emphasize the value of appropriately tuning model parameters according to different data features.

### Impact of the head number

To verify the effect of different head numbers in multi-head self-attention [39] on the experimental effects, relevant experiments are conducted on the UCF101 and the results are shown in Fig. 6. Our results show an increase in recognition accuracy with more attention heads in both 1-shot and 5-shot tasks. The use of multiple attention heads allows for parallel processing of input data, where each head independently examines the input from different angles. This enhances the model's understanding of the data, leading to an improvement in accuracy. Note that while utilizing multiple attention heads improves recognition accuracy, an excessive number of heads can have a detrimental effect on performance. Specifically, when the number of attention heads exceeds 8, the recognition accuracy tends to decline. This could be due to the fact that a large number of heads increases the complexity of the model, leading to longer training times, reduced generalization ability, and a higher likelihood of overfitting. Experiments show that when the number of attention heads is 8, the model has the highest recognition accuracy, reaching 97.9% at the 5-shot task.

### Comparison at N-way task

As shown in Fig. 7, we set different N ways to test the robustness of the proposed method. More ways mean more kinds of samples for each task and more difficulty to recognize. We can observe from the results that recognition accuracy decreases as the way number increases. Our result shows that regardless of the specific conditions and settings used, the accuracy of our proposed method consistently surpasses the results of HyRSM [17], OTAM [23], and TRX [22], which demonstrates the robustness and reliability of our results under various conditions and settings.

## Visualization and analysis

To further validate the effect of the ResTriDA module, the experimental process of ablation of the ResTriDA module is visualized during the classification process. Figure 8 illustrates a series of confusion matrices to compare the effect on the model classification accuracy before and after adding

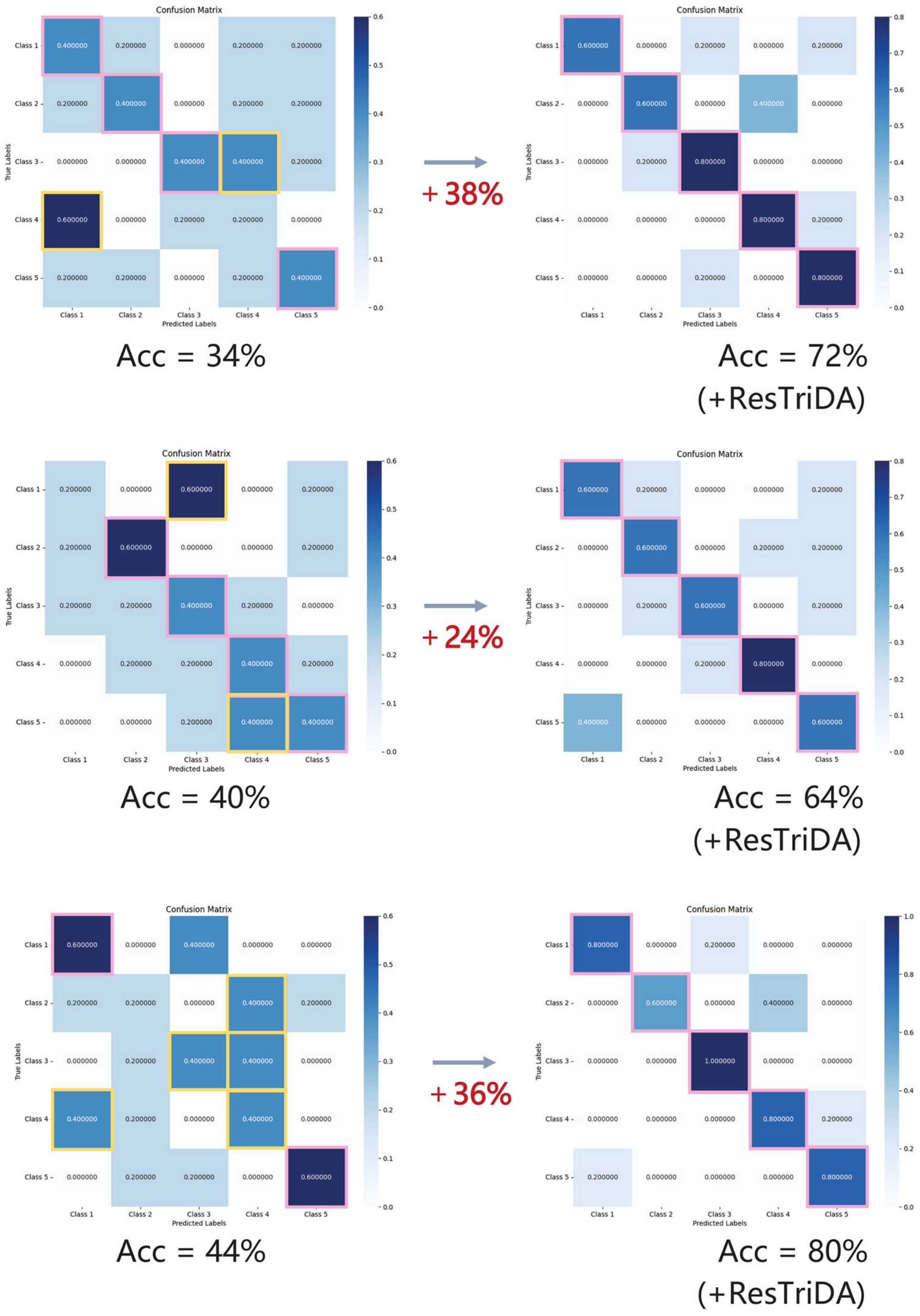


**Fig. 8** Visualization of the confusion matrix before and after adding the ResTriDA module. We fix the input video category and perform three sets of tests. The red boxes represent accurate matches, while the yellow boxes represent inaccurate matches

the ResTriDA module. The experiments are constructed on the HMDB51 dataset using the 5-way 1-shot setting. In these matrices, the values on the diagonal represent correctly recognized instances, while the other elements indicate misclassification. We can see a significant increase in correct classifications after adding the ResTriDA module.

Figure 9 provides an illustrative way to visualize the effect of the ResTriDA module. By observing the class activation map (CAM), we can understand intuitively how the model focuses on specific regions of the video at different points in time. For the Fencing example on the left side of Fig. 9, the class activation mapping shows that the baseline model spreads its attention over the time series, with attention including not only the athletes but also the audience and the background. This shows that the baseline model is more generalized in capturing features spatially and does not clearly distinguish between key features of the action. However, when the MHA mechanism is introduced, attention begins to focus on the interactions between the athletes, in particular the fencing gestures and the movement of the tip of

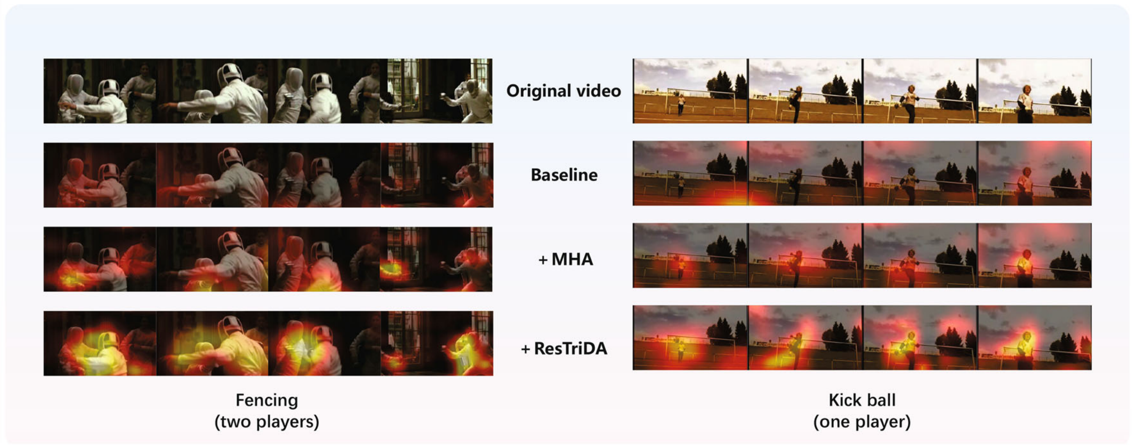


**Fig. 9** Class activation graph visualization. By looking at the class activation graph visualization, we can visualize the effect of the module. We give single and two-player examples to show that our model can handle multi-target character scenarios

the sword. Further, when ResTriDA is added, we notice that the class activation mapping not only continuously focuses on specific movement regions of the athletes when changing over time, but this focus becomes more obvious and consistent between continuous frames.

For example, when an athlete performs an attacking action, the model's focus can move with the tip of the sword, capturing the start, middle, and end positions of the attacking action. It demonstrates that our model can identify spatially critical features and capture the dynamics of these features over time, which is crucial for understanding the flow of the entire fencing action. For the Kickball example, the class activation mapping reveals that the baseline model recognizes the kicking action while also incorrectly focusing on the rest of the field. Augmented with MHA and ResTriDA, the CAM shows that the model's attention is more focused on the player's legs and the ball about to be touched, i.e., spatially critical points of the action. In addition, as the sequence proceeds, the model's attention moves following the ball's trajectory, showing sensitivity to temporal dynamics.

Overall, the detailed visualization of these class activation mappings allows us to explicitly see how the model accurately integrates the spatial localization and temporal evolution of actions in a few-shot learning setting. These visualizations not only demonstrate the model's high sensitivity to identifying nuances in actions but also highlight the important role of temporal continuity in understanding and tracking critical phases of actions. It is this nuanced detection and analysis, even with limited samples, that underpin the model's robust performance in action recognition.

## Training performance curve

Figure 10 shows the Loss and Accuracy curves of the model throughout the training process. From these two graphs, the training process of the model is effective and the loss decreases with time while the accuracy increases accordingly. The smoothness of the curves and their gradual convergence to a steady state indicate good learning progress and stable convergence of the model. There is no overfitting situation where the loss rises or the accuracy drops significantly. Therefore, it can be concluded that the training process is healthy and the model exhibits a strong capacity to fit the training data.

## Exploring the impact of GFPGAN super-resolution on experimental performance

Within this subsection, we first explore the possibility of using generative models to expand the sample set in a few-shot learning environment. As shown in Fig. 11, our approach first generates action descriptions for a specific category (e.g., ‘fencing’) using GPT-3 [42], and then feeds these descriptions into the Text2Video [43] model to generate new video samples. With this strategy, the number of samples is gradually increased from 5-way 1-shot to 5-way 5-shot. As shown in Table 6, the model's performance in distinguishing different categories is significantly improved after increasing the number of generated samples. The accuracy increased from 44.0 to 60.0%, while the AUC and mAP values also increased. However, it is worth mentioning that the cost of generating these additional samples is relatively high. For example, generating a video containing only 8 frames takes 200 to 300 s. In addition, we note that the quality of the sample generation process is somewhat randomized, which means that manual screening is required to obtain high-quality samples. Therefore, although the generative model is theoretically effective in enhancing samples, this approach still has its limitations in real-world application scenarios that require fast responses or limited resources. Nevertheless, the

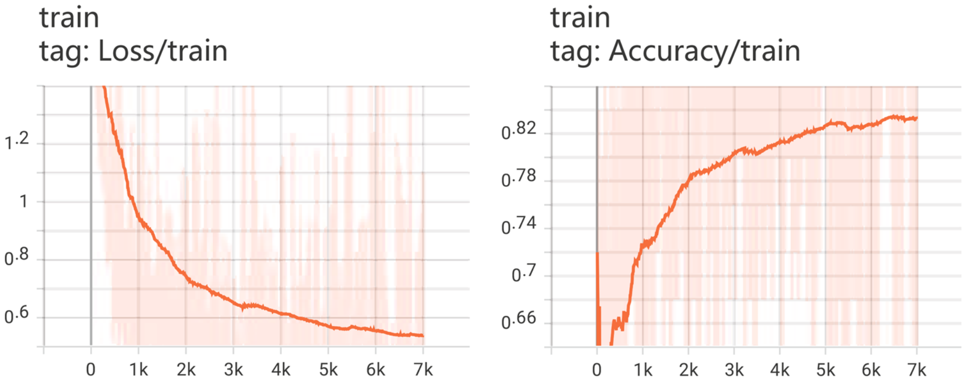


**Fig. 10** The learning curve of the training process. The smooth and gradual convergence of the curves to a steady state indicates a good learning state and stable convergence of the model

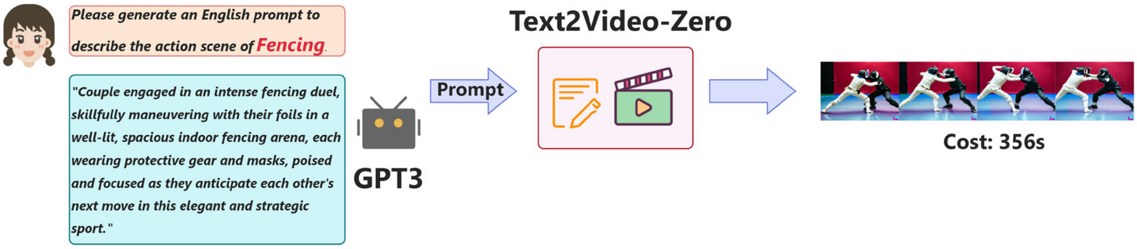

Please generate an English prompt to describe the action scene of Fencing
"Couple engaged in an intense fencing duel, skillfully maneuvering with their foils in a well-lit, spacious indoor fencing arena, each wearing protective gear and masks, poised and focused as they anticipate each other's next move in this elegant and strategic sport."
GPT3
Prompt
Text2Video-Zero
Cost: 356s

**Fig. 11** The process of generating samples. Since the output of a generative model may sometimes deviate from the target category or fail to accurately reflect the desired action scenario, we must carefully filter the generated content to ensure that it is consistent with the original labels and matches the actual scenario

**Table 6** Comparison of recognition results after adding generated samples as support set

| Setting | Accuracy | AUC | mAP | Precision | Recall | F1 score |
|---|---|---|---|---|---|---|
| 5-way 1-shot | 44.0 | 0.6815 | 0.4743 | 0.4912 | 0.4400 | 0.4472 |
| 5-way 2-shot | 48.2 | 0.6865 | 0.4471 | 0.4884 | 0.4800 | 0.4768 |
| 5-way 3-shot | 48.0 | 0.7115 | 0.5419 | 0.4969 | 0.4800 | 0.4823 |
| 5-way 4-shot | 54.0 | 0.8430 | 0.6546 | 0.5769 | 0.5400 | 0.5465 |
| 5-way 5-shot | 60.0 | 0.8000 | 0.7500 | 0.5000 | 0.6000 | 0.5333 |

As the support sample increases, the model shows significant improvement in all metrics

exploration of this approach is still important for few-shot action recognition tasks.

Insights are provided into the GFPGAN (Generative Facial Prior-Generative Adversarial Network) super-resolution algorithm [44] as a strategy to enhance the picture quality of the dataset. The aim is to enhance the image quality through this technique, thus potentially enhancing the performance of the dataset. Experiments are conducted using the HMDB51 dataset and a visual comparison of image quality before and after using GFPGAN is shown in Fig. 12. The numerical results are shown in Table 7, revealing that the performance of the super-resolution processed dataset does improve at the 5-shot setting, but the enhancement is limited. This limitation may stem from the fact that GFPGAN mainly targets spatial quality enhancement of images, whereas temporal features may be more critical for few-shot action recognition

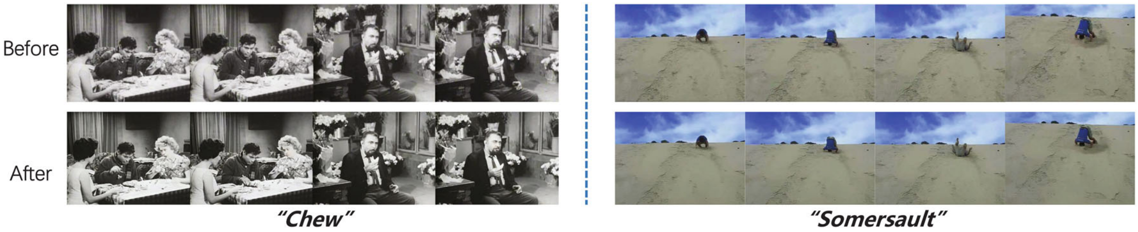


**Fig. 12** Comparison of the effects of super-resolution reconstruction using GFPGAN. Although GFPGAN focuses on the detail enhancement of faces, it also has the function of background reconstruction, making it suitable for action recognition datasets where people are the main subject

**Table 7** Comparison of recognition accuracy using the original dataset and the super-resolution reconstructed dataset

| Metric | Original | | Super-resolution | |
|---|---|---|---|---|
| | 5-way 1-shot | 5-way 5-shot | 5-way 1-shot | 5-way 5-shot |
| Accuracy | 64.2 | 77.9 | 63.6 (− 0.6%) | 78.8 (+ 0.9%) |
| Recall | 0.642 | 0.775 | 0.636 | 0.788 |
| Precision | 0.644 | 0.778 | 0.645 | 0.8 |
| F1 score | 0.643 | 0.776 | 0.640 | 0.793 |
| AUC | 0.874 | 0.937 | 0.87442 | 0.93794 |
| mAP | 0.712 | 0.853 | 0.712 | 0.8537 |

Although a super-resolution reconstruction technique is used to enhance the frame quality of the dataset, the observed results show that this treatment does not bring significant improvement in recognition performance

**Table 8** Detailed results of 10-fold cross-validation under different settings for each dataset

| Dataset | Setting | Fold | | | | | | | | | | Mean | SD | Variance | Min | Max |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | | | | | |
| UCF101 | 5-way 1-shot | 90.2 | 91.7 | 94.5 | 94.1 | 90.7 | 94.2 | 93.4 | 92.8 | 93.9 | 92.9 | 92.9 | 1.4 | 2.1 | 90.2 | 94.5 |
| | 5-way 5-shot | 97.6 | 98.0 | 98.2 | 98.3 | 97.4 | 98.1 | 98.2 | 98.0 | 98.5 | 97.5 | 98.0 | 0.4 | 0.1 | 97.4 | 98.5 |
| HMDB51 | 5-way 1-shot | 64.4 | 65.6 | 63.9 | 64.6 | 65.2 | 61.6 | 63.6 | 62.1 | 64.9 | 65.5 | 64.1 | 1.3 | 1.7 | 61.6 | 65.6 |
| | 5-way 5-shot | 77.9 | 78.2 | 77.5 | 77.5 | 78.5 | 78.3 | 77.5 | 78.8 | 78.1 | 78.1 | 78.0 | 0.4 | 0.2 | 77.5 | 78.8 |
| Kinetics-100 | 5-way 1-shot | 86.1 | 87.1 | 87.6 | 84.7 | 84.9 | 85.1 | 84.5 | 87.8 | 86.4 | 82.7 | 85.7 | 1.5 | 2.3 | 82.7 | 87.8 |
| | 5-way 5-shot | 97.5 | 98.1 | 96.5 | 97.5 | 96.3 | 97.1 | 96.8 | 97.9 | 98.1 | 96.8 | 97.3 | 0.6 | 0.4 | 96.3 | 98.1 |

We only record the accuracy to compare the previous method fairly

tasks that focus on temporal analysis. Although the results fall short of the expected significant improvement, this work provides valuable insights and a basis for exploring more suitable dataset enhancement methods in the future.

## Cross-validation experiments

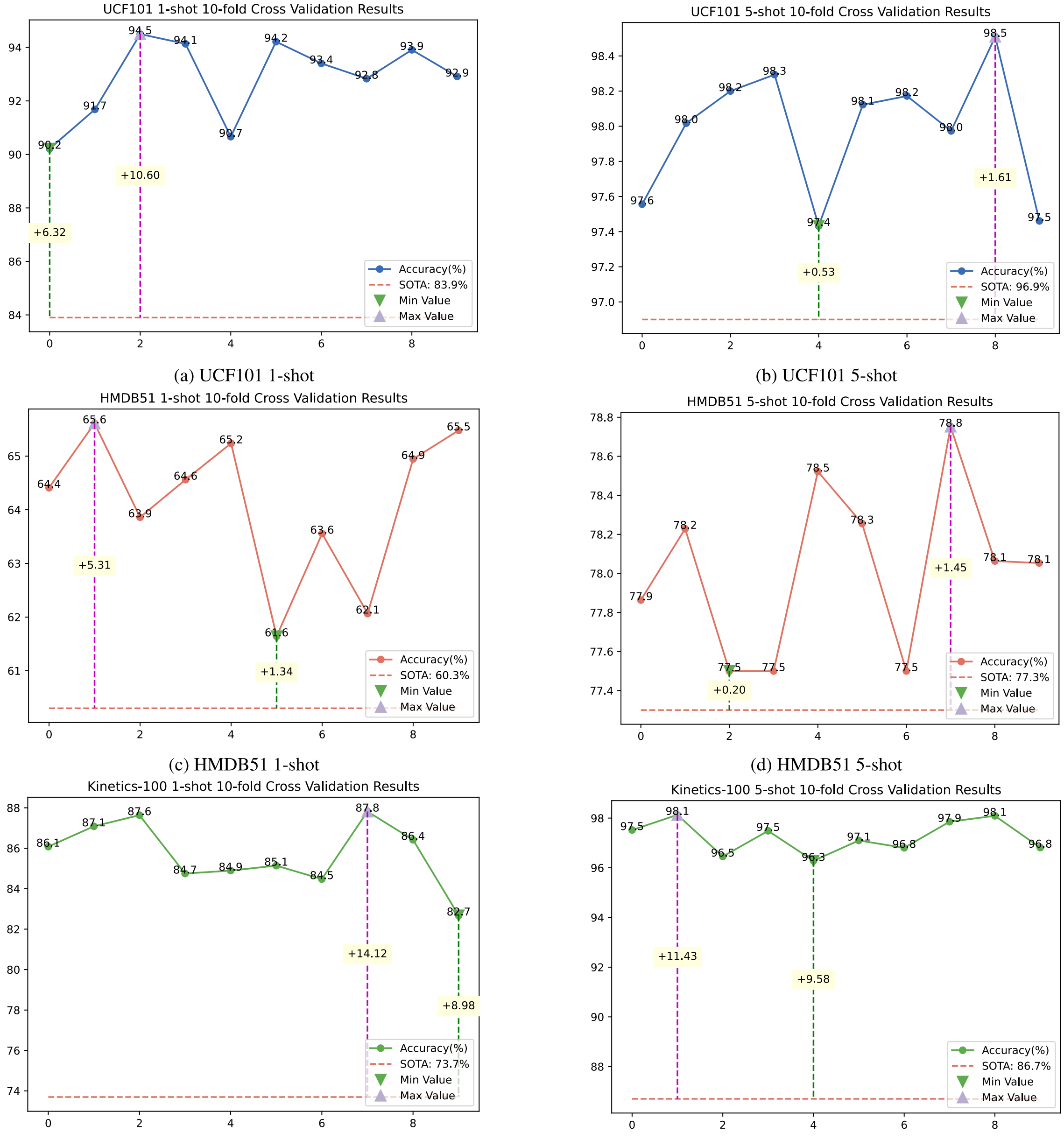
To comprehensively assess the robustness of the overall performance of the proposed model, we conduct extended experiments on three datasets: UCF101, HMDB51, and Kinetics-100. Specifically, we set up two experimental modes for each dataset-5-way 1-shot and 5-way 5-shot-and employ 10-fold cross-validation to ensure the comprehensiveness and reliability of the evaluations. This method involves randomly dividing the dataset into ten subsets, using nine for training and the remaining one for testing. This process cycles ten times, each time selecting a different test subset.

Table 8 shows the 10-fold cross-validation experimental results under different datasets and settings, including the accuracy rate of each experiment and statistical analysis metrics such as mean, standard deviation, variance, minimum, and maximum values. For example, under the 5-way 1-shot setting of the UCF101 dataset, our model HAPN achieves an average accuracy of 92.85% with a standard deviation of 1.44%, demonstrating high consistency and low volatility. Overall, the variances and standard deviations obtained are relatively small, indicating that the proposed model maintains good stability and reliability across various experimental settings. As shown in Fig. 13, by comparing the performance under different datasets and settings, it is evident that the maximum and minimum values in all experiments exceed the performance of the existing state-of-the-art (SOTA) models, highlighting the superiority of HAPN. Additionally, the gap between the maximum and minimum values, although present, is not significant, further confirming the model's robustness.

Based on the above experimental design and statistical analysis, it is clear that the proposed model not only outperforms existing SOTA models in standard benchmark tests but also demonstrates excellent stability and reliability across various test configurations.

## Discussion

Few-shot action recognition tasks draw inspiration from the human ability to quickly grasp and categorize new things, successfully freeing themselves from reliance on large-scale annotated datasets. Some existing advanced methods, such as STRM, often process the temporal and spatial features of videos separately. While this decoupling strategy simplifies the computation process, it weakens the intrinsic connection between time and space in the video data, thereby leading to loss of information. To address this issue, we



(a) UCF101 1-shot

(b) UCF101 5-shot

(c) HMDB51 1-shot

(d) HMDB51 5-shot

(e) Kinetics-100 1-shot

(f) Kinetics-100 5-shot

**Fig. 13** Visualization of accuracy trends and SOTA comparisons across datasets in 10-fold cross-validation. We mark the maximum and minimum values, and the improvements of these values over the SOTA method are highlighted with dashed lines

propose the Hybrid Attentive Prototypical Network model. The core innovation of the HAPN model lies in its integrated approach to processing time and space information. It couples the handling of temporal and spatial features from feature extraction to prototype classification, significantly enhancing the model's comprehensive understanding of video content. Specifically, in the feature extraction phase, we use the R(2+1)D network to extract integrated spatiotemporal features, followed by the ResTriDA network that comprehensively enhances spatial channels and temporal dimensions. Finally, in the classification phase, the PAM module skillfully leverages the inherent similarities of limited samples, enhancing the model's ability to generalize and discern subtle differences.

We systematically test the HAPN model on three standard few-shot action recognition datasets, covering various shot settings from 1-shot to 5-shots, and employing six different evaluation metrics to comprehensively assess the model's performance. Experimental results confirm that HAPN consistently outperforms existing top methods across all test datasets, demonstrating its exceptional performance. Additionally, we assess the model's robustness under various environmental conditions, including changes in lighting, angles, and sampling frequencies. Through module ablation studies, we validate the importance of each component in enhancing model performance. We also explore the potential of using generative models to expand the sample set within the few-shot learning environment. While this method significantly boosts model performance, the high cost and randomness of generating high-quality video samples may pose challenges in practical applications requiring rapid response or limited resources. Moreover, we attempt to enhance the image quality of the dataset using the GFPGAN super-resolution algorithm, aiming to improve the model's performance with this technology. Although there is some improvement in the 5-shot setting, the impact remains limited because GFPGAN primarily enhances the spatial quality of images, whereas few-shot action recognition tasks require a more detailed analysis of temporal features. Furthermore, we thoroughly investigate how the parameters of the ResTriDA module affect experimental outcomes and further demonstrate the model's interpretability and effectiveness through the visualization of confusion matrices and attention modules.

Our HAPN model is inspired by the evolutionary trends of traditional action recognition tasks and introduces the concept of deep coupled processing of videos into the domain of few-shot action recognition for the first time. This pioneering strategy not only showcases the model's forward-looking vision but also highlights its innovative value, providing new research perspectives for subsequent studies in few-shot action recognition.

## Conclusion and future work

In this work, we introduce a novel and innovative framework named HAPN for few-shot action recognition. Uniquely, our proposed network does not separate the processing of temporal and spatial information. Instead, it focuses on the joint handling of these dimensions, enhancing the model's capacity for action recognition tasks. Our framework employs the R(2+1)D backbone network to extract rich features from video sequences and integrates our uniquely designed ResTriDA module, which enriches feature representation across three crucial dimensions: channel, spatial, and temporal. To overcome the overfitting issue prevalent in scenarios with limited samples, we bring in the concept of metric learning and present the prototypical attentive matching module. This module employs the architecture of the prototype network and further integrates a multi-head attention mechanism to discern correlation across sample points of different classes within the vector space, thereby improving the classification task. Numerous experiments confirm that our model sets a new benchmark for robust state-of-the-art performance across three classical datasets used in few-shot action recognition: Kinetics-100, HMDB51, and UCF101. Specifically, in the 5-way 1-shot configuration, our model achieves substantial performance improvements of 9.8%, 3.9%, and 12.4% for UCF101, HMDB51, and Kinetics-100, respectively. These significant enhancements highlight the model's effectiveness and versatility, demonstrating its ability to adeptly handle varied and complex video data with minimal training samples.

However, some limitations require further exploration. Firstly, the model's effectiveness is reliant on the R(2+1)D backbone network, which might limit its capability to address more complex action recognition tasks. Secondly, the computation-intensive nature of the prototypical attentive matching module may affect efficiency when processing large-scale datasets. Despite these limitations, our research showcases a unique approach to addressing few-shot action recognition challenges by leveraging the domain knowledge of video understanding. In our future work, we intend to refine and expand our few-shot action recognition approach by leveraging the cross-modal capabilities of the CLIP model [45]. We plan to initially create detailed textual descriptions (prompts) based on video categories, then use the text encoder of the CLIP model to encode these textual prompts. Leveraging the zero-shot learning capabilities of CLIP, we will identify textual descriptions that best match the video features. Subsequently, we will integrate these textual descriptions with video features to obtain enhanced feature representations. Following this, we will design corresponding models based on these richly integrated feature representations. Through this approach, we aim to not only utilize visual information but also extensively exploit the rich semantic information contained in the text, effectively addressing the challenges of scalability and limited labeled data inherent in few-shot learning scenarios.

**Acknowledgements** This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61873274

**Data Availability** The generated data in this study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** No conflict of financial interests or personal relationships exit in the submission of this manuscript, and it is approved by all authors for publication.

## References

1. Vishwakarma S, Agrawal A (2013) A survey on activity recognition and behavior understanding in video surveillance. Vis Comput 29(10):983–1009
2. Yao G, Lei T, Zhong J (2019) A review of convolutional-neural-network-based action recognition. Pattern Recogn Lett 118:14–22
3. Bilal H, Yao W, Guo Y, Wu Y, Guo J (2017) Experimental validation of fuzzy PID control of flexible joint system in presence of uncertainties. In: 2017 36th Chinese control conference (CCC), pp 4192–4197. https://doi.org/10.23919/ChiCC.2017.8028015
4. Liu Z, Lu X, Liu W, Qi W, Su H (2024) Human-robot collaboration through a multi-scale graph convolution neural network with temporal attention. IEEE Robot Autom Lett 9(3):2248–2255. https://doi.org/10.1109/LRA.2024.3355752
5. Bilal H, Yin B, Aslam MS, Anjum Z, Rohra A, Wang Y (2023) A practical study of active disturbance rejection control for rotary flexible joint robot manipulator. Soft Comput 27(8):4987–5001
6. Bilal H, Yin B, Kumar A, Ali M, Zhang J, Yao J (2023) Jerk-bounded trajectory planning for rotary flexible joint manipulator: an experimental approach. Soft Comput 27(7):4029–4039
7. Ullah FUM, Obaidat MS, Ullah A, Muhammad K, Hijji M, Baik SW (2023) A comprehensive review on vision-based violence detection in surveillance videos. ACM Comput Surv 55(10):1–44
8. Wu Q, Li X, Wang K, Bilal H (2023) Regional feature fusion for on-road detection of objects using camera and 3D-lidar in high-speed autonomous vehicles. Soft Comput 27(23):18195–18213
9. Dou H, Liu Y, Chen S, Zhao H, Bilal H (2023) A hybrid CEEMD-GMM scheme for enhancing the detection of traffic flow on highways. Soft Comput 27(21):16373–16388
10. Caba Heilbron F, Escorcia V, Ghanem B, Carlos Niebles J (2015) Activitynet: a large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 961–970
11. Carreira J, Zisserman A (2017) Quo vadis, action recognition? A new model and the kinetics dataset. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6299–6308
12. Bertasius G, Wang H, Torresani L (2021) Is space-time attention all you need for video understanding? In: ICML, vol 2, p 4
13. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. Adv Neural Inform Process Syst 27:1
14. Tran D, Bourdev L, Fergus R, Torresani L, Paluri M (2015) Learning spatiotemporal features with 3D convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 4489–4497
15. Fu Y, Zhang L, Wang J, Fu Y, Jiang YG (2020) Depth guided adaptive meta-fusion network for few-shot video recognition. In: Proceedings of the 28th ACM international conference on multimedia, pp 1142–1151
16. Kumar Dwivedi S, Gupta V, Mitra R, Ahmed S, Jain A (2019) Protogan: towards few shot learning for action recognition. In: Proceedings of the IEEE/CVF international conference on computer vision workshops
17. Wang X, Zhang S, Qing Z, Tang M, Zuo Z, Gao C, Jin R, Sang N (2022) 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
18. Zhu X, Toisoul A, Perez-Rua J-M, Zhang L, Martinez B, Xiang T (2021) Few-shot action recognition with prototype-centered attentive learning. Preprint arXiv:2101.08085
19. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. Adv Neural Inform Process Syst 30:1
20. Vinyals O, Blundell C, Lillicrap T, Wierstra D et al (2016) Matching networks for one shot learning. Adv Neural Inform Process Syst 29:1
21. Thatipelli A, Narayan S, Khan S, Anwer RM, Khan FS, Ghanem B (2022) 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
22. Perrett T, Masullo A, Burghardt T, Mirmehdi M, Damen D (2021) Temporal–relational cross transformers for few-shot action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 475–484
23. Cao K, Ji J, Cao Z, Chang C-Y, Niebles JC (2020) Few-shot video classification via temporal alignment. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 10618–10627
24. Kumar N, Narang S (2021) Few shot activity recognition using variational inference. Preprint arXiv:2108.08990
25. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: a large video database for human motion recognition. In: International conference on computer vision. IEEE, pp 2556–2563
26. Soomro K, Zamir AR, Shah M (2012) Ucf101: a dataset of 101 human actions classes from videos in the wild. Preprint arXiv:1212.0402
27. Feichtenhofer C (2020) X3d: expanding architectures for efficient video recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 203–213
28. Tran D, Wang H, Torresani L, Ray J, LeCun Y, Paluri M (2018) A closer look at spatiotemporal convolutions for action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6450–6459
29. Song Y, Wang T, Cai P, Mondal SK, Sahoo JP (2022) A comprehensive survey of few-shot learning: evolution, applications, challenges, and opportunities. ACM Comput Surv 2022:1
30. Yang J, Guo X, Li Y, Marinello F, Ercisli S, Zhang Z (2022) A survey of few-shot learning in smart agriculture: developments, applications, and challenges. Plant Methods 18(1):1–12
31. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: relation network for few-shot learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition, vol 1, pp 199–1208
32. Poppe R (2010) A survey on vision-based human action recognition. Image Vis Comput 28(6):976–990
33. Zhu L, Yang Y (2018) Compound memory networks for few-shot video classification. In: Proceedings of the European conference on computer vision (ECCV), pp 751–766
34. Zhang H, Zhang L, Qi X, Li H, Torr PH, Koniusz P (2020) Few-shot action recognition with permutation-invariant attention. In: European conference on computer vision, vol 1. Springer, London, pp 525–542
35. Laenen S, Bertinetto L (2021) On episodes, prototypical networks, and few-shot learning. Adv Neural Inform Process Syst 34:24581–24592
36. Bishay M, Zoumpourlis G, Patras I (2019) Tarn: temporal attentive relation network for few-shot and zero-shot action recognition. Preprint arXiv:1907.09021

37. Liu H, Liu F, Fan X, Huang D (2021) Polarized self-attention: towards high-quality pixel-wise regression. Preprint arXiv:2107.00782
38. Sun Q, Liu Y, Chua T-S, Schiele B (2019) Meta-transfer learning for few-shot learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp 403–412
39. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Adv Neural Inform Process Syst 30:1
40. De Boer P-T, Kroese DP, Mannor S, Rubinstein RY (2005) A tutorial on the cross-entropy method. Ann Oper Res 134(1):19–67
41. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
42. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. arXiv:2005.14165
43. Khachatryan L, Movsisyan A, Tadevosyan V, Henschel R, Wang Z, Navasardyan S, Shi S (2023) Text2video-zero: text-to-image diffusion models are zero-shot video generators. Preprint arXiv:2303.13439
44. Wang X, Li Y, Zhang H, Shan Y (2021) Towards real-world blind face restoration with generative facial prior. In: The IEEE conference on computer vision and pattern recognition (CVPR)
45. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, Sastry G, Askell A, Mishkin P, Clark J et al (2021) Learning transferable visual models from natural language supervision. In: International conference on machine learning, PMLR, pp 8748–8763

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.