Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.053625

ARTICLE

Abnormal Action Detection Based on Parameter-Efficient Transfer Learning in Laboratory Scenarios

Changyu Liu[1], Hao Huang[1], Guogang Huang[2,*], Chunyin Wu[1] and Yingqi Liang[3]

[1]College of Mathematics and Informatics, South China Agricultural University, Guangzhou, 510642, China

[2]School of Oceanography, Shanwei Institute of Technology, Shanwei, 516600, China

[3]Technical Department, Kerric (Guangdong) Laboratory Equipment Research and Manufacture Co., Ltd., Foshan, 528139, China

*Corresponding Author: Guogang Huang. Email: hxianjin@mail3.sysu.edu.cn

Received: 06 May 2024 Accepted: 02 August 2024 Published: 12 September 2024

ABSTRACT

Laboratory safety is a critical area of broad societal concern, particularly in the detection of abnormal actions. To enhance the efficiency and accuracy of detecting such actions, this paper introduces a novel method called TubeRAPT (Tubelet Transformer based on Adapter and Prefix Training Module). This method primarily comprises three key components: the TubeR network, an adaptive clustering attention mechanism, and a prefix training module. These components work in synergy to address the challenge of knowledge preservation in models pre-trained on large datasets while maintaining training efficiency. The TubeR network serves as the backbone for spatio-temporal feature extraction, while the adaptive clustering attention mechanism refines the focus on relevant information. The prefix training module facilitates efficient fine-tuning and knowledge transfer. Experimental results demonstrate the effectiveness of TubeRAPT, achieving a 68.44% mean Average Precision (mAP) on the CLA (Crazy Lab Activity) small-scale dataset, marking a significant improvement of 1.53% over the previous TubeR method. This research not only showcases the potential applications of TubeRAPT in the field of abnormal action detection but also offers innovative ideas and technical support for the future development of laboratory safety monitoring technologies. The proposed method has implications for improving safety management systems in various laboratory environments, potentially reducing accidents and enhancing overall workplace safety.

KEYWORDS

Parameter-efficient transfer learning; laboratory scenarios; TubeRAPT; abnormal action detection

1 Introduction

Laboratory safety has always been a focal point of widespread societal concern, with the safety of individual actions serving as a core element in ensuring laboratory safety. As technology advances, detecting abnormal actions within laboratories has gradually become an essential method for enhancing lab safety management. How to efficiently identify and monitor the various abnormal actions occurring in the laboratory is a key problem in the detection of abnormal actions in the laboratory. Traditional methods rely on lab managers to conduct behavioral monitoring through direct observation or by using surveillance equipment, which are insufficient for effective detection of abnormal actions on a continuous and extensive basis. Therefore, there is an urgent need to develop an automated and real-time method for detecting abnormal actions to strengthen laboratory safety management.

Among these laboratory abnormal actions, some similar actions are difficult to distinguish. For example, drinking water and smelling reagents are highly similar actions; except for the difference in the object held and the distance from the mouth, other motions are nearly identical, as illustrated in Fig. 1. Effective detection of abnormal laboratory actions requires precise differentiation of these similar actions. Models pre-trained on large datasets possess knowledge about similar actions, but how to retain this knowledge in training for laboratory abnormal action detection tasks on smaller datasets requires further research.

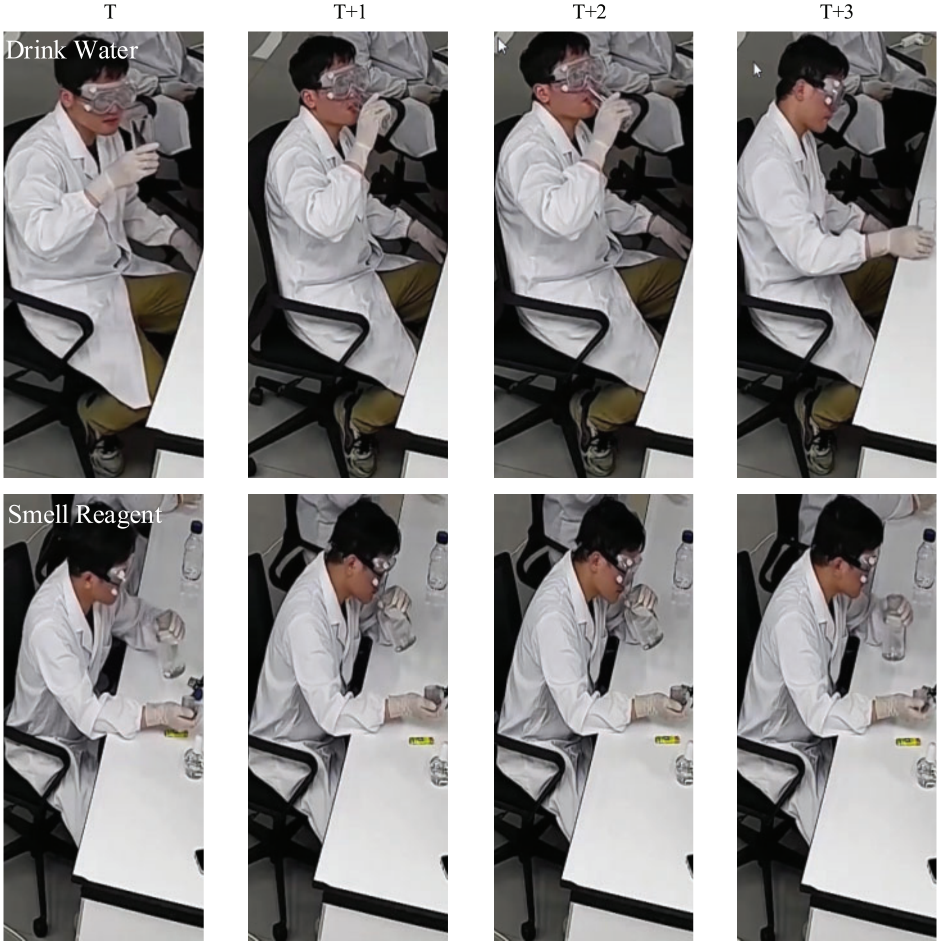


Figure 1: Comparative diagram of similar actions

Additionally, when large-scale pre-trained models are fully fine-tuned for downstream tasks, they face two major challenges: one is the potential degradation of the pre-trained model parameters; the other is the increasing size of pre-trained models, which makes it difficult for researchers to manage the training burden. In light of this, parameter-efficient fine-tuning techniques have been developed. Currently, these techniques are mainly focused on classification models in visual tasks, with relatively less research on more complex tasks like detection. The high cost of data acquisition and comprehensive fine-tuning on small datasets can lead to degradation of pre-trained model parameters and impair their generalization ability [1]. In response to this issue, this paper proposes a parameter-efficient fine-tuning technique for the detection of abnormal laboratory actions, aimed at addressing these challenges.

2 Related Work

This section provides an overview of methods for detecting abnormal actions and techniques for parameter-efficient fine-tuning.

2.1 Abnormal Action Detection Methods

Abnormal action detection methods have been extensively researched and developed across various domains, particularly in intelligent security, video surveillance, healthcare, and action analysis in specific environments. Traditional action detection methods are mostly limited to detecting local actions and often fall short in terms of detection efficiency and accuracy [2,3]. The emergence of deep learning methods offers new options for the task of abnormal action detection. These advanced technologies utilize big data and robust computational power to learn complex patterns, thereby improving the accuracy and efficiency of detection. From the perspective of learning strategies, abnormal action detection methods are mainly divided into supervised and unsupervised learning approaches.

Supervised learning methods rely on pre-labeled data, where these annotations are crucial for the model to learn and understand what constitutes "normal" and "abnormal" actions. Literature [4] introduced a video abnormal action detection method based on motion examples, which uses human skeleton and optical flow information to protect privacy while effectively detecting abnormal events. This method incorporates a support set containing diverse motion examples from a large-scale human action database, to deconstruct roughly defined abnormalities. By employing a non-maximum suppression strategy, it adaptively emphasizes the relevance of abnormal pairs, enhancing detection accuracy. Literature [5] introduced a novel framework called DeepSegmenter, aimed at detecting unedited abnormal actions in natural driving videos. This method addresses the issue by combining activity segmentation and classification within a unified framework, overcoming the limitations of traditional methods.

Unsupervised learning methods do not require any prior annotation information. They identify abnormalities by analyzing the intrinsic structure and patterns within the data itself, enabling them to autonomously complete detection tasks. Literature [6] proposed an abnormal action detection method that uses a pre-trained, domain-agnostic skeletal feature extractor, which is robust against skeletal errors and does not require direct observation of abnormal samples or training. This method is an unsupervised detection approach, capable of deriving anomaly scores without using abnormal samples, thereby detecting abnormal human actions in videos. Literature [7] introduced an attention-based residual autoencoder for video anomaly detection. This method effectively utilizes spatial and temporal information in video data by combining spatial and temporal branches, employing deep convolutional neural networks as encoders, and a multi-stage channel attention mechanism for unsupervised learning. Temporal shift methods are used to capture temporal features, while the channel attention module extracts contextual dependencies. This model significantly improves the accuracy and efficiency of video anomaly detection by leveraging adversarial learning and attention mechanisms.

Current abnormal action detection methods are primarily limited to classifying actions into broad categories of normal and abnormal, without finer differentiation of anomaly types, and these methods do not effectively utilize temporal information. In contrast, laboratory safety managers urgently need a method that can provide fine-grained categories of abnormal actions to more accurately identify and respond to potential risks. Therefore, researching a detection method that can annotate both temporal and spatial information and provide detailed categories of abnormal actions is particularly important. From a technical implementation perspective, these action detection methods are mainly divided into two types: one based on action tubes, and the other based on keyframes.

Action tube-based action detection methods first detect the target boxes of human actors in each frame, then use tracking and linking algorithms to connect these target boxes according to the subject of the action, forming a series of action tubes, which are then input into classifiers for action recognition. Literature [8], building on the dual-stream network and the R-CNN network [9], used the Viterbi algorithm to link target boxes with high confidence and overlap, using the dual-stream network as a feature extractor and SVM as the classifier, successfully introducing deep learning methods into action detection and refining the content and evaluation methods of action detection. Consistent with the evolution from R-CNN to Faster R-CNN [10], literature [11] replaced the selective search algorithm of literature [8] with a Region Proposal Network (RPN), speeding up the model‘s operational efficiency. Unlike dual-stream networks that separately process spatial and temporal features, literature [12] proposed a unified framework, T-CNN, to process both spatial and temporal features together, dividing the video into segments, directly producing action tubes from the segments, connecting the tubes, and then recognizing actions.

Keyframe-based action detection methods process multiple video frames but only detect action subjects in keyframes, extracting feature maps from multiple video frames, then mapping detected action subject target boxes back to the feature maps of multiple video frames to obtain feature maps of the action subjects, which are then used for action recognition. With the introduction of the AVA dataset [13], keyframe-based methods have rapidly developed. Based on the AVA dataset, literature [13] proposed an I3D dual-stream convolutional network. Multi-frame video frames and optical flow frames are input into this network, using Faster R-CNN to detect keyframes, which are then recognized after RPN and ROI Pooling. Optical flow calculation is time-consuming, and researchers have been trying to eliminate usage of optical flow, allowing deep learning methods to directly learn temporal information from video frames. Inspired by the rods and cones in human eyes, literature [14] introduced a dual-stream network called SlowFast, which does not require usage of optical flow, allowing the network to learn spatial information from slow frame rate video frames and motion information from fast frame rate video frames. Since it does not require usage of optical flow, SlowFast not only surpasses previous methods based on optical flow in terms of accuracy but also improves in speed. Given that the AVA dataset contains a large amount of data on human and human interaction actions, literature [15] proposed ACAR-Net, using a feature bank to model the actor-context-actor relationships, then detecting actions, achieving the best results that year. Literature [16] introduced a low-cost point-supervised temporal action detection method that generates pseudo-labels through prototype learning and contrastive constraints, improving detection accuracy and reducing error accumulation. The keyframe-based action detection method is advantageous for fulfilling the laboratory's needs for anomaly detection while requiring minimal annotation effort for the dataset.

Whether based on action tubes or keyframes, action detection methods rely on the development of target detection algorithms. Previous action detection methods were mainly based on Faster R-CNN. Vision Transformer (ViT) is the first pure Transformer method to be applied in computer vision, surpassing the effects of convolutional neural networks and becoming a new milestone [17,18]. As Transformer models are increasingly applied in computer vision, current target detection methods have seen the introduction of DETR [19], a completely end-to-end Transformer framework that does not require post-processing for maximum suppression. TubeR [20] is similar to DETR, transforming the action detection problem into a sequence to sequence problem, thereby easily achieving end-to-end implementation.

2.2 Parameter-Efficient Fine-Tuning

As the size of pre-trained models continues to grow, the training costs associated with traditional full-parameter fine-tuning methods have become increasingly prohibitive. To address this issue, the technique of parameter-efficient fine-tuning (PEFT) has been proposed [1,21]. Initially applied in the field of natural language processing, PEFT has recently expanded its application to computer vision, demonstrating broad adaptability and efficiency.

The focus of parameter-efficient fine-tuning lies in freezing most of the pre-trained model's parameters and training only a small portion, significantly reducing the hardware and time requirements for model training. This process may or may not involve adding extra parameters to the pre-trained model. PEFT techniques are mainly divided into additive methods and partial methods [21]. Additive methods include Adapter Tuning [1,22–25], Prefix Tuning [26–28], and Prompt Tuning [29], while partial methods include Specification Tuning [30] and Reparameter Tuning [31].

Adapter Tuning involves adding Adapter modules to the pre-trained model and training only these modules. These Adapter modules are added to different locations in the Transformer network depending on the downstream task and learning objectives. Literature [32] applied prompt and adapter tuning to self-supervised encoder-decoder speech models, significantly improving performance on sequence generation tasks like automatic speech recognition (ASR) and slot filling. Literature [33] introduced the COMPACTER method, which fine-tunes large-scale pre-trained language models by integrating low-rank hypercomplex adapter layers, achieving task performance comparable to or better than full-parameter fine-tuning while maintaining a minimal number of trainable parameters.

Prefix Tuning adjusts the model by adding a short learnable prefix to each layer of the Transformer network without changing the original model's parameters. Direct optimization of prefix vectors can lead to training instability, thus requiring other methods to generate prefix vectors to avoid instability issues. Literature [34] addressed this problem by optimizing a multi-layer perceptron. VQT [35] introduces a small number of learnable "query" tokens at each layer to aggregate intermediate features of the Transformer base model, effectively used for linear probing, achieving parameter and memory-efficient transfer learning. EFFT [36] fine-tunes the pre-trained visual Transformer model in a parameter-efficient manner, primarily addressing internal and inter-layer redundancy, and achieves efficient information extraction while preserving the model's intermediate features.

Prompt Tuning is a simplified form of Prefix Tuning and an improvement over hard prompts, employing a soft prompt approach. Similar to Prefix Tuning in purpose, soft prompts are carefully optimized to adapt to downstream tasks. During the entire training process, other network parameters are frozen, only adjusting these prompts, thus exemplifying a method of task-adaptive network fine-tuning [37,38].

The partial methods in parameter-efficient fine-tuning aim to train only a small portion of the network's parameters without changing the internal structure of the model, to adapt to specific downstream tasks. Specification Tuning [30] focuses on directly fine-tuning a small subset of key parameters in the pre-trained model, adapting to downstream tasks through this refined adjustment. Reparameter Tuning [31] introduces new learnable parameters during the training phase and integrates these parameters into the original model during the inference phase through a technique known as reparameterization. These two partial methods provide a strategy for efficiently leveraging pre-trained models for rapid adaptation to downstream tasks while minimizing the demand for computational resources, opening new avenues for the flexible application of large-scale pre-trained models.

Overall, parameter-efficient fine-tuning techniques offer an effective and resource-saving solution for adapting large-scale pre-trained models. The advantages of this technology are significant, including a substantial reduction in computational resources and storage space required, effectively preventing catastrophic forgetting. Moreover, parameter-efficient fine-tuning enhances parameter sharing and can compete with traditional full-parameter fine-tuning methods without sacrificing performance, particularly evident when the network adapts to downstream tasks with smaller data scales.

3 The Proposed Method

In this section, we introduce our proposed TubeRAPT. Section 3.1 introduces TubeRAPT, and Section 3.2 discusses the training methods for TubeRAPT.

3.1 Introduction of TubeRAPT

Inspired by literature [20], this paper proposes an end-to-end method for laboratory abnormal action detection, named Tubelet Transformer based on Adapter and Prefix Training module, abbreviated as TubeRAPT. As shown in Fig. 2, TubeRAPT primarily consists of three components: feature extraction, a Transformer, and target box generation along with category prediction. A distinctive feature of the TubeRAPT network is the use of the Adapter and Prefix Training module (APT module) along with an adaptive clustering attention mechanism, aimed at enhancing the network's performance and efficiency.

Feature extraction encodes each video segment through a convolutional neural network, obtaining feature representations for each frame to be used in subsequent time-series modeling and classification. In this paper, the CSN network [39] is used as the feature extractor. This network includes a regularization function, which, while achieving lower accuracy on the training set, delivers higher accuracy on the test set.

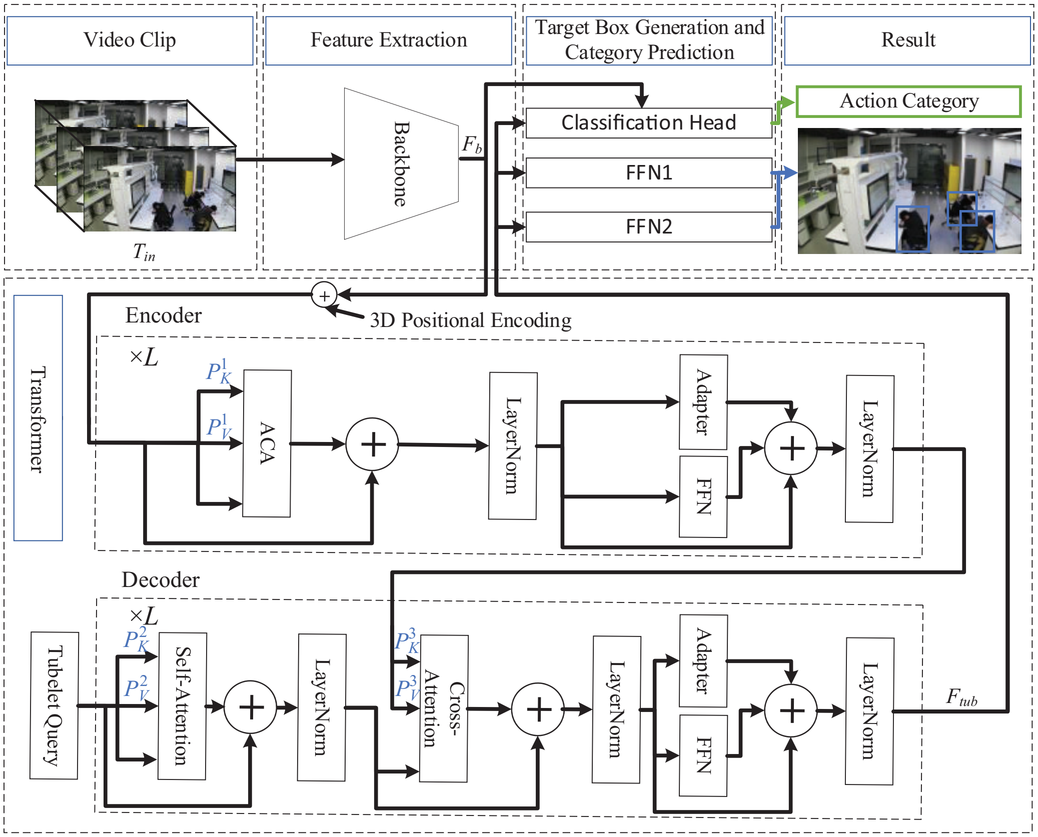


Figure 2: The architecture of TubeRAPT

The Transformer consists of several attention layers and feedforward neural network layers. It encodes and models the frame sequences, capturing long-term dependencies and contextual information within the time series. The data processing involves the TubeRAPT network feeding video segments of T_{in} frames into the Backbone network to extract video features F_b, shaped as $T_b WH \times$ C, which after adding 3D positional encoding, are inputted into the Transformer and transformed into action tube features F_{tub}, shaped as $T_o \times N \times C$. Here, T_b and C represent the time dimension and feature dimension, respectively, W and H are the feature width and height, T_o is the output time dimension, and N is the number of actions. Action tubes are three-dimensional structures spanning multiple frames in the video sequence, used to represent the spatial and temporal continuity of an action.

The design for target box generation and category prediction draws from literature [20]. Action tube features F_{tub} are processed by the regression head to determine the locations of actions and exclude queries where no action exists, while the classification head determines the category of each query. The classification head can be formalized as:

$$AH\left(F_{tub}, F_b\right) = Linear\left(LN\left(CA\left(pool_t\left(F_{tub}\right), LN\left(SA\left(F_b\right)\right)\right) + pool_t\left(F_{tub}\right)\right)\right) \quad (1)$$

where LN represents Layer Normalization, and $pool_t$ refers to Temp Pooling, which is actually a fully connected layer that reduces the dimensions of F_{tub} from $T_o \times N \times C$ to $N \times C$. N represents the number of action tubes, indicating the upper limit of detected human figures. The regression head includes two FFNs, i.e., feedforward neural networks, where FFN1 is responsible for predicting target boxes, and FFN2 maps the features output by the Transformer to scores or probabilities for each target box.

3.1.1 Attention

As shown in Fig. 2, TubeRAPT employs self-attention, cross-attention, and clustering attention. In the Transformer network, Self-Attention and Cross-Attention are crucial operations. The formulas for Self-Attention are shown in Eqs. (2) and (3):

$$SA\left(F\right)=softmax\left(\left(\sigma_{q}^{s}\left(F\right)\times\sigma_{k}^{s}\left(F\right)^{T}\right)/\sqrt{C}\right)\times\sigma_{v}^{s}\left(F\right) \tag{2}$$

$$\sigma_{i}^{s}\left(*\right)=Linear_{i}^{s}\left(*\right),i\in\{q,k,v\} \tag{3}$$

where F is the token sequence input to Self-Attention, and *Linear* refers to a linear mapping, achieved through a single fully connected layer. Cross-Attention is utilized in the decoder of the Transformer, where it decodes action tube features from the encoder output memory features, F_{en}, and the action tube queue features, F_q, as illustrated in Eqs. (4) and (5):

$$CA\left(F_q,F_{en}\right)=softmax\left(\left(F_q\times\sigma_{k}^{c}\left(F_{en}\right)^{T}\right)/\sqrt{C}\right)\times\sigma_{v}^{c}\left(F_{en}\right) \tag{4}$$

$$\sigma_{i}^{c}\left(*\right)=Linear_{i}^{c}\left(*\right),i\in\{q,k,v\} \tag{5}$$

Inspired by literature [40], this paper employs Adaptive Clustering Attention (ACA) in place of the self-attention layers in the encoder to enhance the computational speed of self-attention. ACA is an attention mechanism for Transformer models that uses a small number of prototypes to represent queries and computes the attention mapping only between prototypes and keys. The number of prototypes is automatically determined based on the distribution of the queries, and finally, the attention output is broadcasted to the queries represented by the prototypes. The aim of this method is to reduce computational complexity and improve the scalability of the model.

Using ACA involves the following steps: first, the attention between queries and keys is calculated using a multi-head attention mechanism. Then, the values are weighted by the attention map to produce the weighted output features. Through the adaptive clustering approach, the attention output is estimated by calculating the attention mapping between the prototypes and the keys, updating the features of the prototypes. This estimation method reduces computational complexity compared to methods that compute attention precisely.

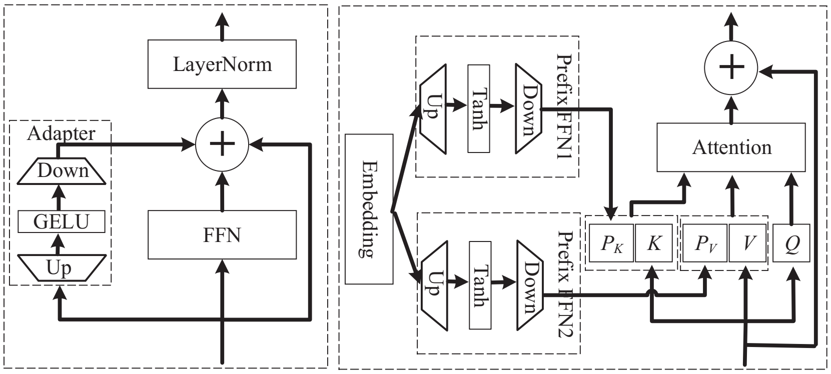
3.1.2 Adapter and Prefix Training Module

Inspired by the work of [41], this paper introduces a parameter-efficient fine-tuning technique module for abnormal action detection, termed the Adapter and Prefix Training (APT) module, while the TubeR network utilizing the APT module is referred to as the TubeRAPT network. This module consists of two parts: Adapter and Prefix, as illustrated in Fig. 3.

As depicted in Fig. 3a, the Adapter part of the APT module is formalized as shown in Eq. (6).

$$X_o=GELU\left(X_i\cdot W_{up}\right)\cdot W_{Down} \tag{6}$$

where $X_i \in \mathbb{R}^{N\times d}$ represents the input of the Adapter module. After being multiplied with the weight parameters $W_{up}\in\mathbb{R}^{d\times d1}$ for dimensionality expansion, it undergoes activation with the *GELU* activation function, and then is multiplied with the weight parameters $W_{down}\in\mathbb{R}^{d1\times d}$ to reduce it back to its original dimensionality. $X_o\in\mathbb{R}^{N\times d}$ represents the output of the Adapter module. Here, N denotes the number of tokens in the Transformer of the TubeRAPT network, and d represents the feature dimension of tokens, which is 256 in this paper.



(a) Adapter structure (b) Prefix structure

Figure 3: The architecture of APT. (a) Adapter structure, (b) Prefix structure

As shown in Fig. 3b, this represents the portion of the Prefix structure in this paper, formalized as shown in Eq. (7).

$$P_o = Tanh\left(P_e \cdot \dot{W}_{Up}\right) \cdot \dot{W}_{Down}, e, o \in \{K, V\} \tag{7}$$

where $P_e \in \mathbb{R}^{n \times d}$ represents the parameters of the Embedding, $\dot{W}_{up} \in \mathbb{R}^{d \times d2}$ denotes the weight parameters providing dimensionality expansion, and $\dot{W}_{Down} \in \mathbb{R}^{d2 \times d}$ represents the weight parameters for dimensionality reduction. During this process, it undergoes activation with the Tanh activation function. P_o denotes the prefix vector. 'n' represents the number of keys and values provided by the prefix vector. The tokens of the key-value pairs output by the Prefix FFN are concatenated with the original key-value pairs and finally inputted into the multi-head attention module.

When prefix vectors are added to the input sequence, they undergo the same linear mapping and self-attention computation process described above. This means that the prefix vectors not only provide additional contextual information but also influence the way the entire sequence is processed through the workflow of the self-attention mechanism. In this way, prefix vectors can manipulate attention distributions internally in the model, thereby guiding the model to generate more accurate outputs tailored to specific tasks.

Fig. 2 depicts the structure of the encoder and decoder of the TubeRAPT network, including the position and addition method of the APT module. Similar to TubeR, TubeRAPT has L layers of encoder and L layers of decoder. The boxes labeled with "Adapter" in Fig. 2 indicate the positions where the Adapter module is added, while the blue "P" represents the positions where prefix vectors are added. Assuming the size of the input tokens to the Transformer is (1, d), the total input of the Transformer is (N, d), where N is the number of tokens and d is the dimensionality of tokens. To maintain token dimensions unchanged during computation, the input dimension of the Adapter is d, the hidden dimension is $d1$, and the output dimension remains d. The Prefix FFN outputs n vectors of size (1, d), which are then concatenated with the key-value pairs of the attention layer.

(1) Encoder: In the encoder of TubeRAPT, the Adapter of the APT module is parallel to the FFN layer, while the prefix vectors P_K^1 and P_V^1 of the APT module are concatenated with the keys and values of the ACA layer, and the APT module is added to all L layers of the encoder.

(2) Decoder: In the decoder of TubeRAPT, the Adapter of the APT module remains parallel to the FFN layer. The APT module has four prefix vectors in the decoder. The first part of the decoder performs self-attention operations on the encoder queries, then conducts cross-attention with the output memory matrix of the last layer of the encoder. The prefix vectors P_K^2 and P_V^2 of the APT module are concatenated with the keys and values of the self-attention layer, while P_K^3 and P_V^3 are concatenated with the keys and values calculated by the memory matrix of the cross-attention layer.

3.2 The Training Method of TubeRAPT

As shown in Fig. 4, similar to other methods that utilize parameter-efficient fine-tuning, the TubeRAPT network proposed in this paper also uses the TubeR model, which was pre-trained on a pre-training dataset. It then freezes all parameters of the TubeR model and adds the Adapter and Prefix Training (APT) module, training only the parameters of the APT module on the target dataset. Compared to traditional full fine-tuning methods, this parameter-efficient fine-tuning approach effectively leverages the knowledge learned from the pre-training dataset, thereby achieving better performance on the target task. The TubeRAPT network is characterized by its use of the rich representations learned by the TubeR model on a large-scale pre-training dataset. These representations are preserved through the freezing of parameters, and the model is then fine-tuned on a specific target dataset, allowing it to better adapt to new tasks.

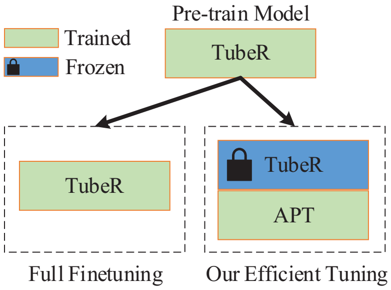


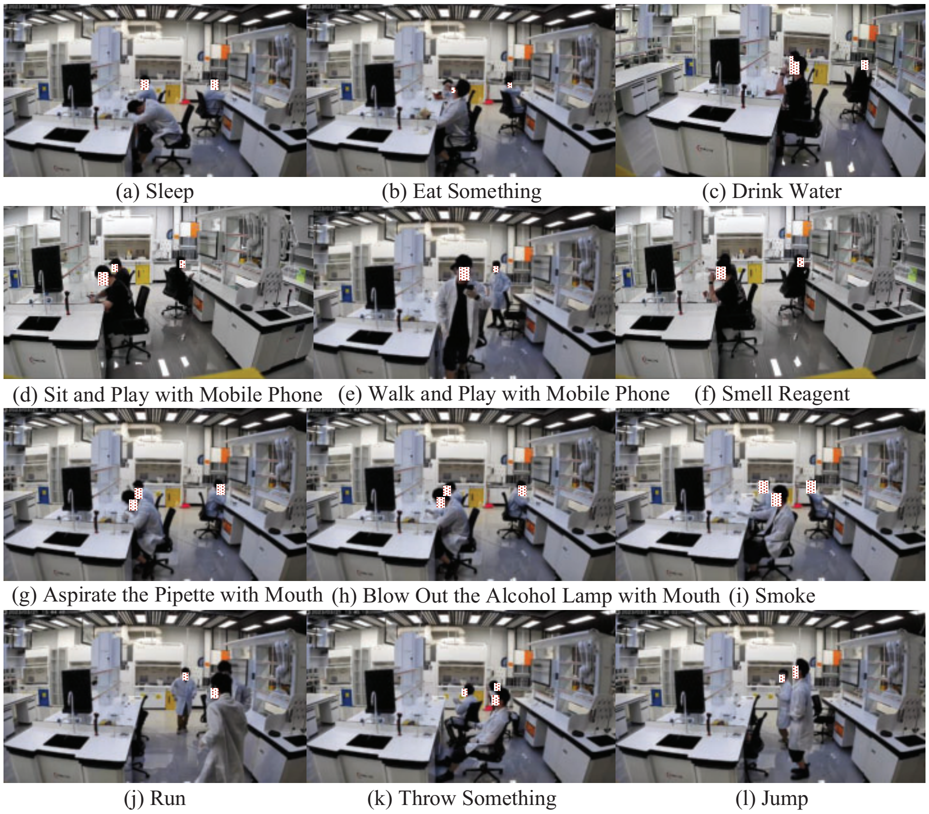
Figure 4: Comparison between full finetuing and our efficient tuning

4 Experiments

Section 4.1 introduces the CLA dataset, Section 4.2 describes the experimental setup, and Section 4.3 presents the experimental results and analysis. The experimental environment used in this study consisted of an Intel(R) Xeon(R) Gold 6330 and RTX 3090.

4.1 CLA Dataset

This paper uses the Crazy Lab Activity dataset, abbreviated as CLA, for relevant experiments. The CLA dataset is annotated in the format of the AVA dataset [13]. It is a dataset designed for studying and detecting abnormal actions in laboratory environments. The data for this dataset were collected using cameras inside a laboratory, aiming to record and analyze video data of different actions under laboratory conditions. The dataset includes 46 laboratory video segments, comprising 29 training videos and 17 test videos, with a training to testing ratio of 1.7:1. Each video segment is 15 min long, with a resolution of 1920 × 1080, and both the training and test sets include footage from three different angles. The CLA dataset encompasses 12 types of abnormal actions observed in the lab, including: Sleep, Eat Something, Drink Water, Sit and Play with Mobile Phone, Walk and Play with Mobile Phone, Smell Reagent, Aspirate the Pipette with Mouth, Blow Out the Alcohol Lamp with Mouth, Smoke, Run, Throw Something, and Jump, corresponding to parts (a) to (l) in Fig. 5.



(a) Sleep (b) Eat Something (c) Drink Water

(d) Sit and Play with Mobile Phone (e) Walk and Play with Mobile Phone (f) Smell Reagent

(g) Aspirate the Pipette with Mouth (h) Blow Out the Alcohol Lamp with Mouth (i) Smoke

(j) Run (k) Throw Something (l) Jump

Figure 5: Twelve types of abnormal actions in the CLA dataset. (a) Sleep, (b) Eat Something, (c) Drink Water, (d) Sit and Play with Mobile Phone, (e) Walk and Play with Mobile Phone, (f) Smell Reagent, (g) Aspirate the Pipette with Mouth, (h) Blow Out the Alcohol Lamp with Mouth, (i) Smoke, (j) Run, (k) Throw Something, (l) Jump

The CLA dataset was filmed entirely within the laboratory of Kerric (Guangdong) Laboratory Equipment Research and Manufacture Co., Ltd. (Foshan, China). The laboratory covers an area of 108 square meters and contains nine tables, each with a width of 75 centimeters and a length of 360 centimeters, as shown in Fig. 6. The cameras are represented in Fig. 6 as two overlapping squares, marked with a ‘C’. This paper features two scenes within the laboratory, referred to as Scene A and Scene B. Each scene has three positions, with at most one person conducting experiments at each position. Scene A is captured by three cameras from three different angles, while Scene B is captured by two cameras from the left and right sides, with each camera placed as high as possible. This design aims to maximize coverage of the laboratory scene and capture as many angles and perspectives as possible.

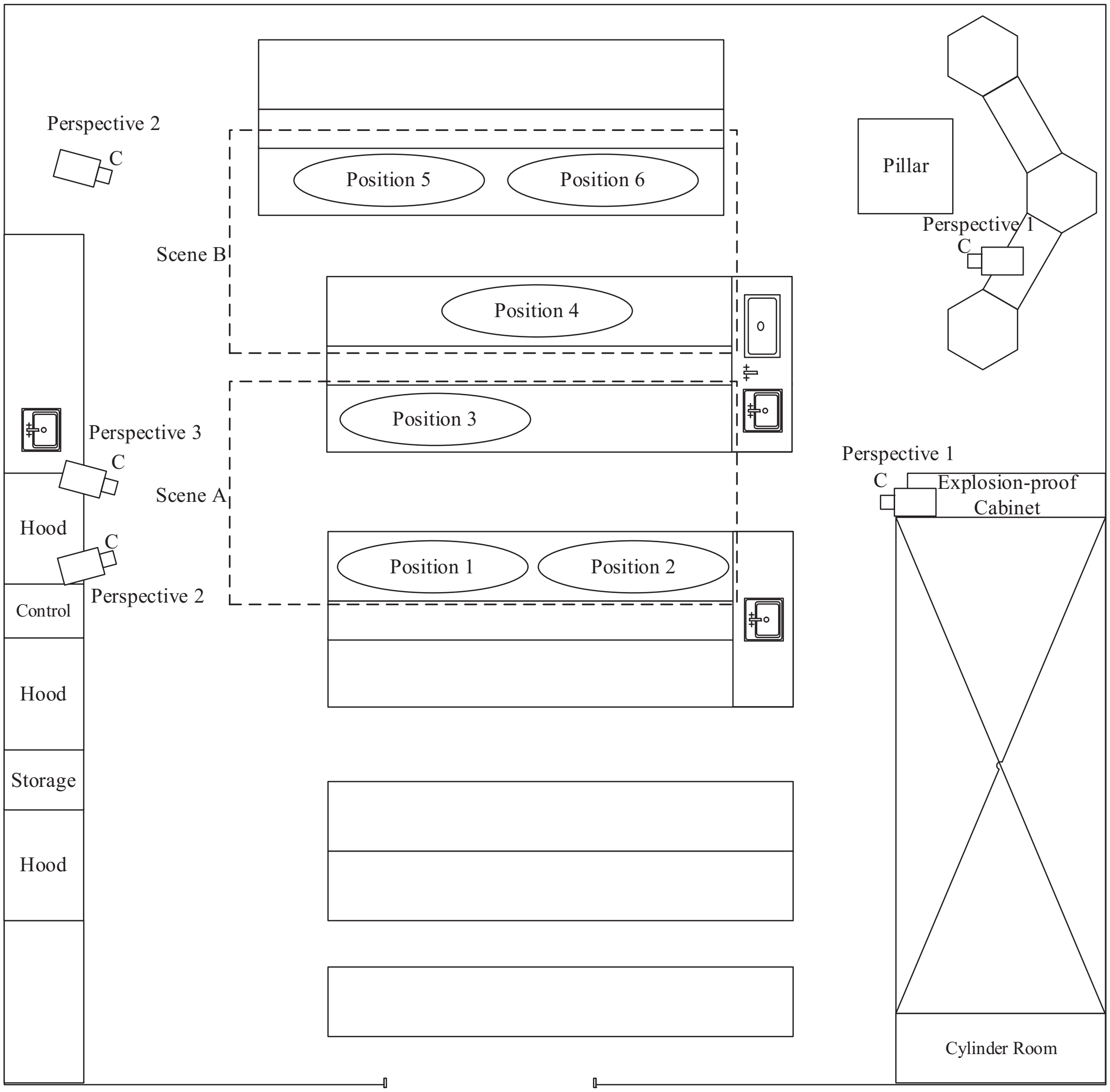


Figure 6: Laboratory floor plan

The dataset discussed in this paper involves two laboratory scenes, each containing at least two people, ensuring the complexity of the dataset. The dataset involves four different individuals, dressed in two types of clothing: regular attire and personal protective equipment (PPE). The PPE attire includes a white coat, plastic gloves, and protective eyewear. Scenes A and B, along with corresponding views and different attire, are illustrated in Fig. 7. In Scene A, View 3 and in Scene B, View 1 feature regular attire, while Views 1 and 2 in Scene A, and View 2 in Scene B feature PPE attire.

The distribution of training and testing sets, along with the label distribution across various categories in the dataset, is illustrated in Fig. 8. Overall, the data is relatively balanced, with no significant long-tail effect. The least frequent action observed is blowing out an alcohol lamp, while the most frequent action is sitting and using a cellphone.

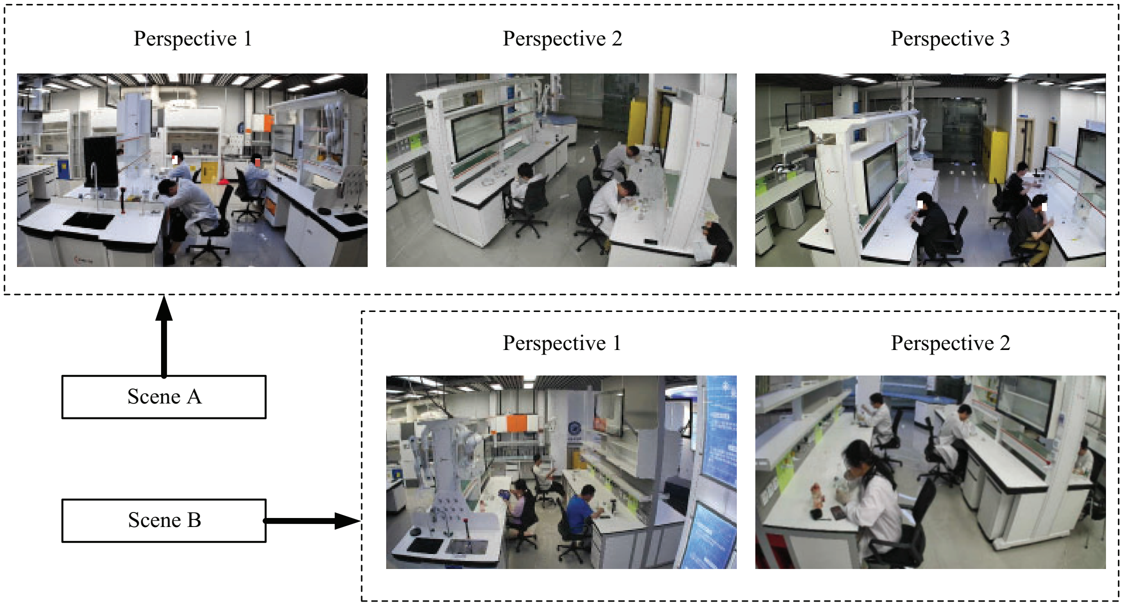


Figure 7: Schematic diagram of perspectives corresponding to different scenes

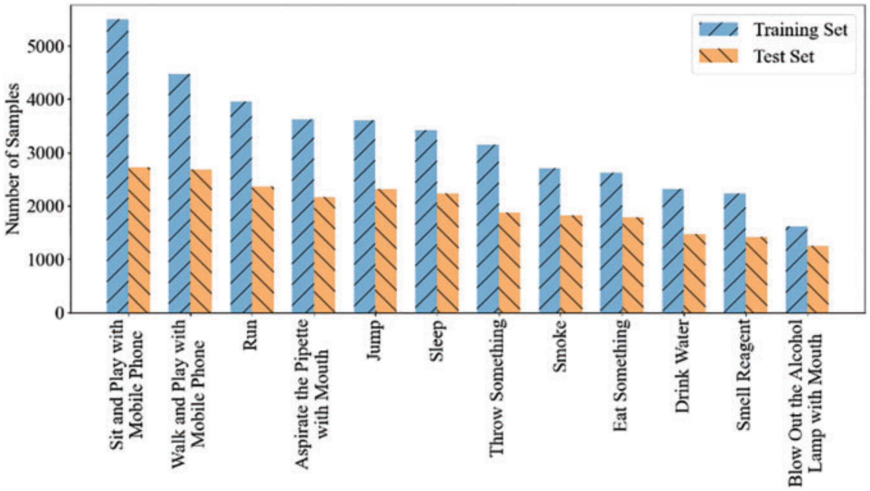


Figure 8: CLA dataset distribution

As shown in Fig. 9, the widths of the bounding boxes for human targets in this dataset mostly range from 100 to 300, while the heights mostly range from 200 to 600, indicating that most bounding boxes are vertically oriented rectangles. The areas of the bounding boxes are primarily concentrated between 20,000 and 180,000, corresponding to dimensions roughly between 100 × 200 to 300 × 600. The largest bounding box measures 766 × 926.

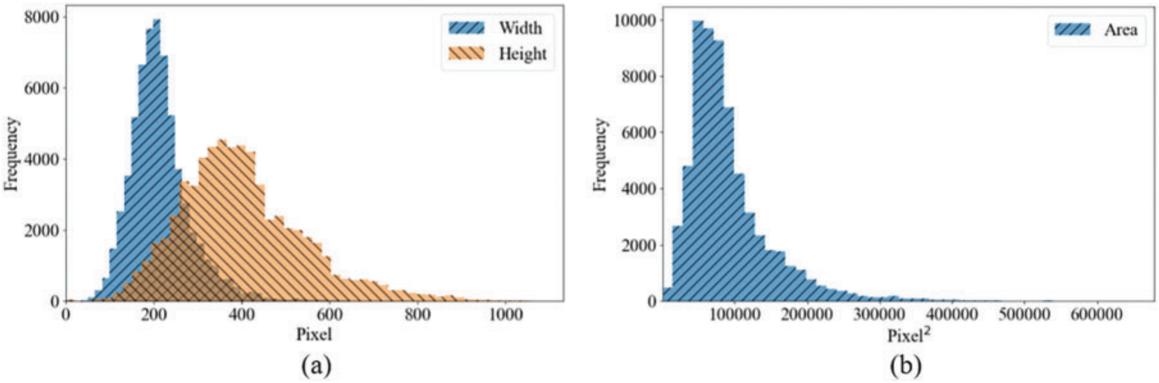


Figure 9: Distribution chart of bounding box width, height, size, and area frequency. (a) Frequency distribution of width and height, (b) Frequency distribution of area

4.2 Experiment Settings

The experiments in this paper utilized the AdamW optimizer for parameter optimization, with an initial learning rate set at 0.0001 and a weight decay coefficient also set at 0.0001. The batch size used was 6, and the number of epochs was 20. During the training phase, data augmentation strategies were employed, including random horizontal flipping, random cropping, and color jittering. Specifically, the video frame size after random cropping was 256 × 455. In the testing phase, the class confidence threshold was set at 0.8, with the video frame's shorter side always scaled to 256 pixels. This paper reports frame-mAP@IoU = 0.5 following [13] using a single, center-crop inference protocol. Additionally, TubeRAPT was initialized using the pre-trained weights of TubeR [20] and further trained the APT module. The number of layers L in the encoder and decoder was set to 6, the intermediate dimension $d1$ of the Adapter was set to 512, the number of prefix vectors n was 30, and the intermediate dimension $d2$ of the Prefix FFN was set to 800. After the training of the APT module was completed, the adaptive clustering attention was then added to the network.

4.3 Results and Analysis

In this section, the paper will discuss the experimental results and analysis of TubeRAPT on the CLA dataset.

4.3.1 Ablation Study

To further analyze the specific contributions of the APT (Adapter and Prefix Training) module and ACA (Adaptive Clustering Attention) to the performance of the TubeRAPT network, this study designed a series of ablation experiments. In these experiments, the paper independently and jointly trained the Adapter and Prefix parts and assessed their individual and combined impacts on the performance of the TubeRAPT network. Subsequently, the ACA was integrated directly without training to evaluate its impact on the performance of the TubeRAPT network. As shown in Table 1, even though the proportion of trained parameters in the Adapter is relatively small, its contribution to the overall mAP was still very significant, achieving an increase of 0.28%. This indicates that in the TubeRAPT network, the Adapter plays an important role in enhancing model performance, despite having relatively fewer training parameters. This may be because the Adapter module can better capture the correlations between data, thereby enhancing the network's generalization and representational capabilities. Furthermore, the paper found that combining the Adapter and Prefix into the APT module results in better overall performance than training the two components separately. This suggests a synergistic effect between Adapter and Prefix, where the combined APT module can more effectively boost network performance and yield better results.

Table 1: Table of ablation study results for the APT module

Model	Pre-train	Parameter	Train ratio	mAP
TubeR+Adapter	AVA v2.2	93.61 M	3.48%	51.28
TubeR+Prefix	AVA v2.2	105.27 M	16.37%	51.00
TubeRAPT	AVA v2.2	108.42 M	19.83%	51.33

As shown in Table 2, integrating ACA into the TubeR method resulted in a slight decrease of 0.02% in its mAP metric, whereas combining ACA with TubeRAPT improved the performance slightly by 0.01% over the original TubeRAPT network. This result could be attributed to the enhancement provided by the APT module, allowing ACA to more effectively focus on key queries, thus enabling the model's encoder to better concentrate on capturing important information.

Table 2: Table of ablation study results for adaptive clustering attention

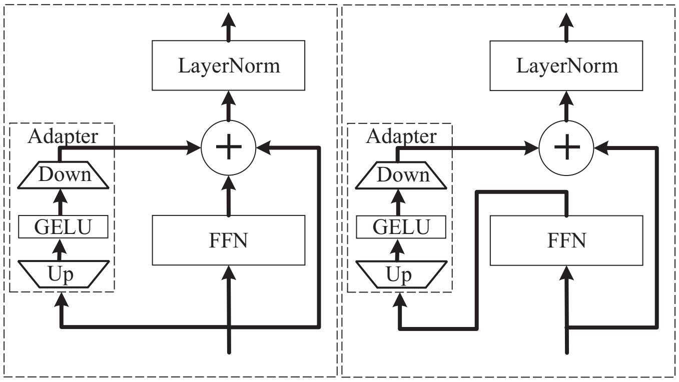
Model	Pre-train	ACA	mAP
TubeR	AVA v2.2	×	66.91
TubeR	AVA v2.2	√	66.89
TubeRAPT	CLA	×	68.43
TubeRAPT	CLA	√	68.44

Based on the above experimental results, we can conclude that in the TubeRAPT network, the Adapter component contributes the most to performance, with a certain synergy existing between it and the Prefix component. The combined APT module plays a crucial role in improving network performance. Meanwhile, the impact of ACA on the performance of the TubeRAPT network is relatively limited. This study provides important guidance and insights for understanding the structure of the TubeRAPT network.

4.3.2 Determine the Position of the Adapter

As shown in Fig. 10, this paper experimented with two different methods of adding adapters.

In the parallel structure, as shown in Fig. 10a, adapters are added to the network in parallel. This means that the data is separately inputted into both the adapter and the FFN network, and then the outputs of both are combined for the next step of computation. In contrast, in the serial structure, as shown in Fig. 10b, the output of the FFN serves as the input to the adapter, and then a series of subsequent computations are performed together in the original network.



(a) Parallel structure (b) Serial structure

Figure 10: Schematic diagram for parallel and serial structures. (a) Parallel structure, (b) Serial structure

It is worth noting that, as shown in Table 3, there is no significant performance gap between the parallel and serial structures. This could be because in the serial structure, both the Adapter module and the FFN adopt a residual connection design. This means that the output of the Adapter is added to the input of the FFN through a residual connection, which helps to avoid information loss and error accumulation, resulting in comparable performance between the serial and parallel structures for the Adapter.

Table 3: Table of results for different structures

Structure	mAP
Parallel structure	51.33
Serial structure	51.33

Therefore, whether in parallel or serial structures, the Adapter module can effectively improve network performance, and there is no significant difference between them. This finding suggests that when choosing how to add adapters, the specific situation and requirements can dictate which method to use. Additionally, the use of serial addition can enhance performance through residual connections, providing important guidance for further adjustments and improvements in Adapter design. Given the convenience of implementation in code and with no significant performance differences, this study opts for the parallel structure as the method for adding adapters to the APT module.

4.3.3 Setting ACA Iterations

As shown in Table 4, the computation time of ACA increases with the number of rounds, with a significant rapid growth observed between 32 and 40 rounds.

The rounds represent a hyperparameter used to adjust the adaptive clustering attention. Specifically, rounds refer to the number of independent hash tables. Each hash table has a unique set of hash functions that map input vectors to a fixed number of hash values, thereby partitioning the space into multiple cells. The same hash value implies that vectors fall into the same cell, thus the value of rounds affects the number of generated clustering clusters and the precision of space partitioning.

Table 4: Comparison table of computation time between adaptive clustering attention and self-attention

Types of attention	Iterations	Time
Self-attention	—	0.10137367248535156
Adaptive clustering attention	8	0.06808781623840332
Adaptive clustering attention	16	0.0687859058380127
Adaptive clustering attention	32	0.06969904899597168
Adaptive clustering attention	40	0.07232546806335449

As shown in Table 5, the mAP values demonstrate an increasing trend when the number of rounds for ACA ranges from 8 to 32, peaking at 68.44. However, when the number of rounds increases to 40, the mAP drops to 68.35, falling short of the performance of TubeRAPT without integrated ACA. This may be due to the higher number of rounds causing query over-concentration, leading to partial information loss.

Table 5: Results table for adaptive clustering attention *vs.* self-attention

Model	Iterations	ACA	mAP
TubeRAPT	—	×	68.43
TubeRAPT	8	√	66.05
TubeRAPT	16	√	68.36
TubeRAPT	32	√	68.44
TubeRAPT	40	√	68.35

In this study, setting the number of rounds to 32 yielded the best results. This configuration not only improved the time efficiency of self-attention computation by 31.25% but also had minimal impact on performance. Under the condition of 32 rounds, the TubeRAPT network achieved a good balance between performance and speed. These results highlight the crucial role of ACA in the TubeRAPT network.

4.3.4 Model Comparison

As shown in Table 6, the TubeRAPT network, using the same pre-trained weights, achieves 76.7% performance of the TubeR network with only 19.83% of the parameters trained. Furthermore, when utilizing TubeR weights pre-trained with CLA, the performance of the TubeRAPT network without ACA even surpasses the original TubeR network by 1.52%. This result underscores the remarkable performance of the TubeRAPT network in terms of effectiveness and superiority, especially in the application of pre-trained weights. The inference speed of the TubeR method is 364.23 FPS, while the inference speed of TubeRAPT is relatively lower at 269.34 FPS. However, TubeRAPT still meets the requirements for real-time performance.

Table 6: Horizontal comparison table of model performance

Model	Pre-train	Parameter	Train ratio	ACA	mAP
TubeR	AVA v2.2	90.46 M	100%	×	66.91
TubeRAPT	AVA v2.2	108.42 M	19.83%	×	51.33
TubeRAPT	CLA	108.42 M	19.83%	×	68.43
TubeRAPT	CLA	108.42 M	19.83%	√	68.44

More notably, when the APT module is combined with ACA, not only does it enhance processing speed, but it also achieves a slight performance increase, reaching the highest performance of 68.44% in our comparative experiments, as shown in Table 6. This finding highlights the significant potential of combining the APT module with ACA, indicating that performance improvement is not limited to faster speeds but also includes subtle performance enhancements. This is particularly crucial in practical applications, especially when dealing with large-scale datasets, where even slight performance improvements could lead to significant overall benefits.

Table 7 illustrates the performance comparison of our method with recent outstanding action detection models such as ACRN [42], Slow-only [14], SlowFast [14], VideoMAE [43], and TubeR [20]. It can be observed from Table 7 that our method achieves the highest mAP, and TubeRAPT also exhibits a significant advantage in terms of parameter count compared to other networks.

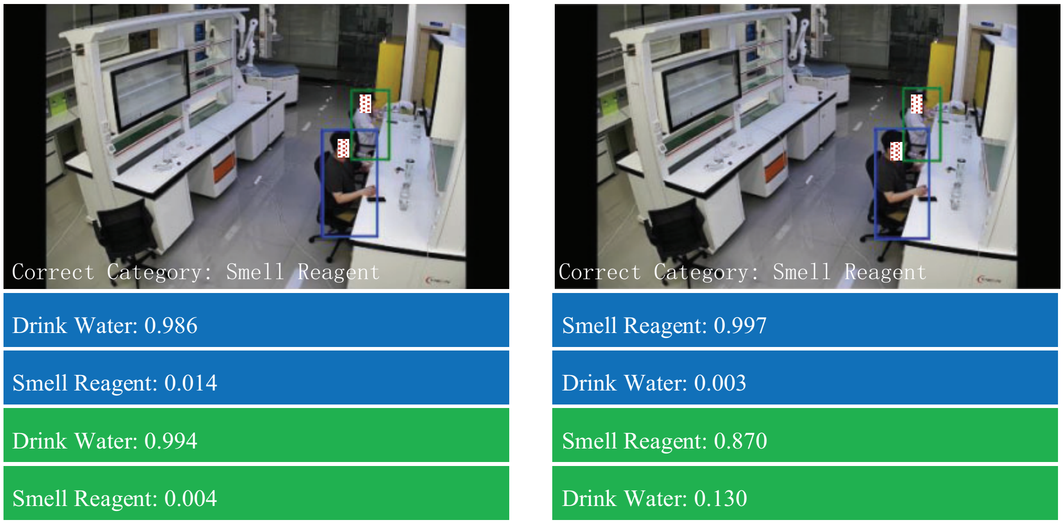
Table 7: Vertical comparison table of model performance

Model	Backbone	Pre-train	Parameter	mAP
ACRN	Res-50	Kinetics-400	92.08 M	31.18
Slow-only	Res-50	Kinetics-400	31.66 M	34.72
SlowFast	Res-50	Kinetics-400	33.67 M	37.39
VideoMAE	ViT-Base	Kinetics-400	86.24 M	38.13
TubeR	CSN-50	Kinetics-400	74.45 M	63.55
TubeR	CSN-152	AVA v2.2	90.46 M	66.91
TubeRAPT	CSN-152	CLA	108.42 M	68.44

4.3.5 Visualization

As shown in Fig. 11, the current abnormal action captured in the video frame is sniffing chemicals. Both TubeR and TubeRAPT accurately identify the human body positions, but TubeR shows a deviation in class prediction for the abnormal action, incorrectly classifying it as drinking water action, which bears similarity to the sniffing chemicals action in appearance, as shown in Fig. 11a.

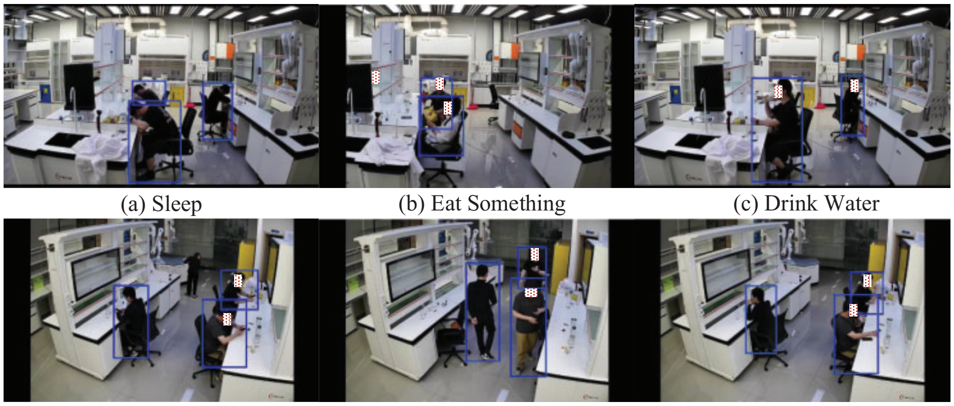
In contrast, as depicted in Fig. 11b, TubeRAPT correctly predicts the abnormal actions of the two individuals. For the target selected in the green box, its action closely resembles drinking water, thus TubeRAPT assigns a high confidence of 0.13, which is significantly higher than the blue-boxed target. This difference indicates the introduction of the APT module, enabling TubeRAPT to not only accurately learn the correct classification of abnormal actions but also understand subtle similarities between different actions.



(a) Detection using TubeR (b) Detection using TubeRAPT

Figure 11: Detection results for TubeR and TubeRAPT. (a) Detection using TubeR, (b) Detection using TubeRAPT

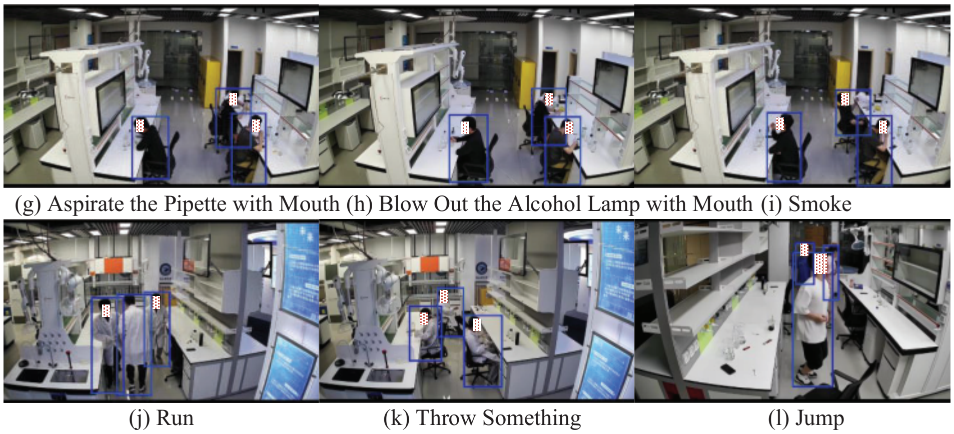
As shown in Fig. 12, the TubeRAPT network successfully detects 12 classes of abnormal actions. Panels (a) to (c) display the detection results from viewpoint 1 of Scene A in the CLA dataset, while (d) to (f) present the results from viewpoint 2 of the same scene, and (g) to (i) depict the results from viewpoint 3 of Scene A. Panels (j) and (k) show the results from viewpoint 1 of Scene B, while (l) illustrates the results from viewpoint 2 of Scene B. It is noteworthy that the action categories and target box predictions in each subfigure are accurate. Fig. 12 fully demonstrates the outstanding generalization ability of the TubeRAPT network, showcasing its strong adaptability to different scenes and viewpoints.



(a) Sleep (b) Eat Something (c) Drink Water

(d) Sit and Play with Mobile Phone (e) Walk and Play with Mobile Phone (f) Smell Reagent

Figure 12: (Continued)

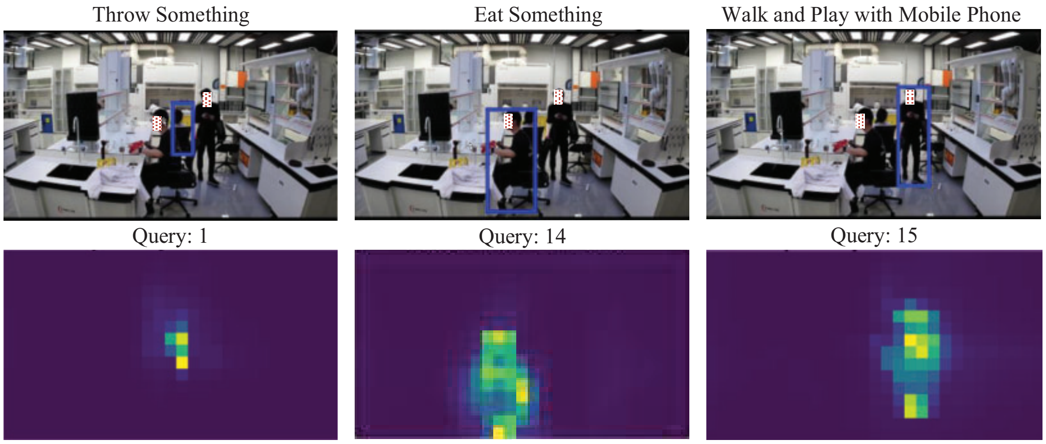


(g) Aspirate the Pipette with Mouth (h) Blow Out the Alcohol Lamp with Mouth (i) Smoke

(j) Run (k) Throw Something (l) Jump

Figure 12: TubeRAPT prediction results display for the 12 types of abnormal actions. (a) Sleep, (b) Eat Something, (c) Drink Water, (d) Sit and Play with Mobile Phone, (e) Walk and Play with Mobile Phone, (f) Smell Reagent, (g) Aspirate the Pipette with Mouth, (h) Blow Out the Alcohol Lamp with Mouth, (i) Smoke, (j) Run, (k) Throw Something, (l) Jump

As illustrated in Fig. 13, each query in the TubeRAPT network corresponds to a response in the final layer decoder attention matrix, reflecting the focus of attention while processing input data. It is evident from the decoder's focus during the input data processing that it successfully identifies the position of each person and accurately classifies their action. Specifically, the first query identifies the action of throwing objects, the fourteenth query identifies the action of eating, and the fifteenth query identifies the action of walking while using a mobile phone. This result fully demonstrates the TubeRAPT network's capability to accurately learn and comprehend video content. Further analysis of the attention matrix reveals that the TubeRAPT network can predict accurate action categories and target localization through its classification and regression heads. This not only further validates the effectiveness and accuracy of the TubeRAPT network in detecting abnormal action tasks but also demonstrates the network's deep understanding of input data, enabling it to extract key information and accurately classify and localize it.



Throw Something Eat Something Walk and Play with Mobile Phone

Query: 1 Query: 14 Query: 15

Figure 13: Heatmap of TubeRAPT query results

These visualized results not only highlight the robust performance of the TubeRAPT network but also enhance understanding of the network's interpretability.

5 Conclusion

This paper proposes an innovative TubeRAPT network based on the TubeR method framework and elaborates on the training strategies for the APT module and TubeRAPT network. The method proposed in this paper not only detects abnormal acntion but also classifies it into 12 different types of anomalies. Through ablation experiments, this study reveals the Adapter's significantly greater contribution compared to the prefix within the APT module and finds that their combination is superior to their individual application. Additionally, minimal impact of ACA on the mAP performance of the TubeRAPT network is observed. Furthermore, experimental explorations into different architectures of the Adapter indicate that both serial and parallel architectures within the Adapter employ residual connections, resulting in insignificant performance differences.

To enhance the processing speed of the TubeRAPT network, the ACA module is introduced. Experimental findings suggest that when the hash round is set to 32, the TubeRAPT network achieves the optimal balance between speed and performance. Under this configuration, TubeR's performance slightly decreases, while TubeRAPT's performance marginally improves. Further horizontal comparative experiments demonstrate a 1.53% performance improvement of the TubeRAPT network over the TubeR network.

By introducing the APT module and ACA, the TubeRAPT network achieves significant improvements in both performance and speed. These achievements not only provide new perspectives and methods for anomaly detection in laboratory scenarios but also contribute valuable insights for further improvements and optimizations of deep learning models.

Acknowledgement: We would like to thank College of Mathematics and Informatics at South China Agricultural University, and Kerric (Guangdong) Laboratory Equipment Research and Manufacture Co., Ltd. for supporting this study.

Funding Statement: This work was supported by the Philosophy and Social Sciences Planning Project of Guangdong Province of China (GD23XGL099), the Guangdong General Universities Young Innovative Talents Project (2023KQNCX247), and the Research Project of Shanwei Institute of Technology (SWKT22-019).

Author Contributions: Study conception and design: Changyu Liu and Guogang Huang; data processing and analysis: Hao Huang and Yingqi Liang; draft manuscript preparation: Hao Huang and Changyu Liu; grammar revision: Guogang Huang and Chunyin Wu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The Crazy Lab Activity (CLA) dataset used in this study contains sensitive personal information and holds commercial value. Due to privacy concerns and proprietary considerations, this dataset cannot be made publicly available. The CLA dataset is maintained as a trade secret to protect both individual privacy and the company's competitive advantage. Researchers interested in accessing this data for verification purposes may contact the corresponding author to discuss potential collaborative opportunities, subject to strict confidentiality agreements and ethical approval. We have strived to present our methodologies and findings in detail within these constraints.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] N. Houlsby *et al.*, "Parameter-efficient transfer learning for NLP," in *Proc. 36th Int. Conf. Mach. Learn.*, Long Beach, CA, USA, 2019, pp. 4944–4953.

[2] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, pp. 107–123, 2005. doi: 10.1007/s11263-005-1838-7.

[3] A. Prest, V. Ferrari, and C. Schmid, "Explicit modeling of human-object interactions in realistic videos," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 835–848, 2012. doi: 10.1109/TPAMI.2012.175.

[4] Y. Su, H. Zhu, Y. Tan, S. An, and M. Xing, "Prime: Privacy-preserving video anomaly detection via motion exemplar guidance," *Knowl.-Based Syst.*, vol. 278, 2023, Art. no. 110872. doi: 10.1016/j.knosys.2023.110872.

[5] A. Aboah, U. Bagci, A. R. Mussah, N. J. Owor, and Y. Adu-Gyamfi, "DeepSegmenter: Temporal action localization for detecting anomalies in untrimmed naturalistic driving videos," in *Proc. 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Workshops*, Vancouver, BC, Canada, 2023, pp. 5359–5365.

[6] F. Sato, R. Hachiuma, and T. Sekii, "Prompt-guided zero-shot anomaly action recognition using pretrained deep skeleton features," in *Proc. 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 6471–6480.

[7] V. Le and Y. Kim, "Attention-based residual autoencoder for video anomaly detection," *Appl. Intell.*, vol. 53, no. 3, pp. 3240–3254, 2023. doi: 10.1007/s10489-022-03613-1.

[8] G. Gkioxari and J. Malik, "Finding action tubes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 759–768.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, 2014, pp. 580–587.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 29th Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, 2015, pp. 91–99.

[11] S. Saha, G. Singh, M. Sapienza, P. H. S. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," in *Proc. 27th British Mach. Vis. Conf.*, York, UK, 2016, pp. 58.1–58.13.

[12] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (T-CNN) for action detection in videos," in *Proc. 16th IEEE Int. Conf. Comput. Vis.*, Venice, Italy, 2017, pp. 5823–5832.

[13] C. Gu *et al.*, "AVA: A video dataset of spatio-temporally localized atomic visual actions," in *Proc. 31st Meet. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 6047–6056.

[14] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. 17th IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Republic of Korea, 2019, pp. 6201–6210.

[15] J. Pan, S. Chen, M. Z. Shou, Y. Liu, J. Shao, and H. Li, "Actor-context-actor relation network for spatio-temporal action localization," in *Proc. 2021 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 464–474.

[16] L. Ping, J. Cao, and X. Ye, "Prototype contrastive learning for point-supervised temporal action detection," *Expert. Syst. Appl.*, vol. 213, 2023, Art. no. 118965. doi: 10.1016/j.eswa.2022.118965.

[17] A. Vaswani *et al.*, "Attention is all you need," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, 2017, pp. 5999–6009.

[18] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent.*, 2021.

[19] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. 16th European Conf. Comput. Vis.*, Glasgow, UK, 2020, pp. 213–229.

[20] J. Zhao *et al.*, "TubeR: Tubelet transformer for video action detection," in *Proc. 2022 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, New Orleans, LA, USA, 2022, pp. 13588–13597.

[21] Y. Xin *et al.*, "Parameter-efficient fine-tuning for pre-trained vision models: A survey," arXiv preprint arXiv:2402.02242, 2024.

[22] S. Chen *et al.*, "AdaptFormer: Adapting vision transformers for scalable visual recognition," in *Proc. 36th Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2022.

[23] T. Yang, Y. Zhu, Y. Xie, A. Zhang, C. Chen and M. Li, "AIM: Adapting image models for efficient video action recognition," in *Proc. 11th Int. Conf. Learn. Represent.*, Kigali, Rwanda, 2023.

[24] D. Yin, L. H. B. Li, and Y. Zhang, "Adapter is all you need for tuning visual tasks," arXiv preprint arXiv:2311.15010, 2023.

[25] S. Jie and Z. Deng, "Convolutional bypasses are better vision transformer adapters," arXiv preprint arXiv:2207.07039, 2022.

[26] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. Joint Conf. 59th Annu. Meet. Assoc. Comput. Linguist. 11th Int. Joint Conf. Natural Lang. Process.*, Bangkok, Thailand, 2021.

[27] C. Xu, S. Yang, Y. Wang, Z. Wang, Y. Fu and X. Xue, "Exploring efficient few-shot adaptation for vision transformers," arXiv preprint arXiv:2301.02419, 2023.

[28] Q. Gao *et al.*, "A unified continual learning framework with general parameter-efficient tuning," in *Proc. 2023 IEEE/CVF Int. Conf. Comput. Vis.*, Paris, France, 2023, pp. 11449–11459.

[29] B. Lester, R. Al-Rfou, and N. Constant, "The power of scale for parameter-efficient prompt tuning," in *Proc. 2021 Conf. Empir. Methods in Natural Lang. Process.*, Punta Cana, Dominican Republic, 2021.

[30] S. Kornblith, J. Shlens, and Q. V. Le, "Do better ImageNet models transfer better?" in *Proc. 32nd IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, 2019, pp. 2656–2666.

[31] E. Hu *et al.*, "Low-rank adaptation of large language models," in *Proc. 10th Int. Conf. Learn. Represent.*, 2022.

[32] K. Chang *et al.*, "Prompting and adapter tuning for self-supervised encoder-decoder speech model," in *Proc. 2023 IEEE Automatic Speech Recognit., Understanding Workshop*, Taipei, Taiwan, 2023.

[33] R. K. Mahabadi, J. Henderson, and S. Ruder, "Compacter: Efficient low-rank hypercomplex adapter layers," in *Proc. 35th Conf. Neural Inf. Process. Syst.*, 2021, vol. 2, pp. 1022–1035.

[34] B. X. Yu, J. Chang, L. Liu, Q. Tian, and C. W. Chen, "Towards a unified view on visual parameter-efficient transfer learning," arXiv preprint arXiv:2210.00788, 2022.

[35] C. Tu, Z. Mai, and W. Chao, "Visual query tuning: Towards effective usage of intermediate representations for parameter and memory efficient transfer learning," in *Proc. 2023 IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Vancouver, BC, Canada, 2023, pp. 7725–7735.

[36] D. Chen, "Aggregate, decompose, and fine-tune: A simple yet effective factor-tuning method for vision transformer," arXiv preprint arXiv:2311.06749, 2023.

[37] M. Jia *et al.*, "Visual prompt tuning," in *Proc. 17th European Conf. Comput. Vis.*, Tel Aviv, Israel, 2022, pp. 709–727.

[38] S. An *et al.*, "Input-tuning: Adapting unfamiliar inputs to frozen pretrained models," arXiv preprint arXiv:2203.03131, 2022.

[39] D. Tran, H. Wang, M. Feiszli, and L. Torresani, "Video classification with channel-separated convolutional networks," in *Proc. 17th IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Republic of Korea, 2019, pp. 5551–5560.

[40] M. Zheng, P. Gao, R. Zhang, X. Wang, H. Li, and H. Dong, "End-to-end object detection with adaptive clustering transformer," in *Proc. 32nd British Mach. Vis. Conf.*, 2021.

[41] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," in *Proc. 10th Int. Conf. Learn. Represent.*, 2022.

[42] C. Sun, A. Shrivastava, C. Vondrick, K. Murphy, R. Sukthankar, and C. Schmid, “Actor-centric relation network,” in *Proc. 15th European Conf. Comput. Vis.*, Munich, Germany, 2018, pp. 335–351.

[43] Z. Tong, Y. Song, J. Wang, and L. Wang, “VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training,” in *Proc. 36th Conf. Neural Inf. Process. Syst.*, New Orleans, LA, USA, 2022.