Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.072964

ARTICLE

DyLoRA-TAD: Dynamic Low-Rank Adapter for End-to-End Temporal Action Detection

Jixin Wu[1,2], Mingtao Zhou[2,3], Di Wu[2,3], Wenqi Ren[4], Jiatian Mei[2,3] and Shu Zhang[1,*]

[1]School of Information Science and Technology, Yunnan Normal University, Kunming, 650500, China
[2]Yunnan Key Laboratory of Smart Education, Yunnan Normal University, Kunming, 650500, China
[3]Key Laboratory of Education Informatization for Nationalities, Ministry of Education, Yunnan Normal University, Kunming, 650500, China
[4]School of Cyber Science and Technology, Sun Yat-sen University, Guangzhou, 510275, China

*Corresponding Author: Shu Zhang. Email: zhangshu@ynnu.edu.cn

Received: 08 September 2025; Accepted: 12 November 2025; Published: 12 January 2026

ABSTRACT: End-to-end Temporal Action Detection (TAD) has achieved remarkable progress in recent years, driven by innovations in model architectures and the emergence of Video Foundation Models (VFMs). However, existing TAD methods that perform full fine-tuning of pretrained video models often incur substantial computational costs, which become particularly pronounced when processing long video sequences. Moreover, the need for precise temporal boundary annotations makes data labeling extremely expensive. In low-resource settings where annotated samples are scarce, direct fine-tuning tends to cause overfitting. To address these challenges, we introduce Dynamic Low-Rank Adapter (DyLoRA), a lightweight fine-tuning framework tailored specifically for the TAD task. Built upon the Low-Rank Adaptation (LoRA) architecture, DyLoRA adapts only the key layers of the pretrained model via low-rank decomposition, reducing the number of trainable parameters to less than 5% of full fine-tuning methods. This significantly lowers memory consumption and mitigates overfitting in low-resource settings. Notably, DyLoRA enhances the temporal modeling capability of pretrained models by optimizing temporal dimension weights, thereby alleviating the representation misalignment of temporal features. Experimental results demonstrate that DyLoRA-TAD achieves impressive performance, with 73.9% mAP on THUMOS14, 39.52% on ActivityNet-1.3, and 28.2% on Charades, substantially surpassing the best traditional feature-based methods.

KEYWORDS: Temporal action detection; end-to-end training; dynamic low-rank adapter; parameter-efficient fine-tuning; video understanding

1 Introduction

Temporal Action Detection (TAD) [1–3] is a fundamental yet highly challenging task in the field of video understanding [4–6]. Its core objective is to identify specific action categories and localize their start and end times within untrimmed long videos. This task holds significant practical value in various critical domains, such as surveillance and security [7], educational video analysis [8], healthcare monitoring, and autonomous driving.

In recent years, TAD models have made significant progress in architectural design. However, recent studies have highlighted two emerging trends: end-to-end training [9–11] and fine-tuning of vision foundation models [12,13]. End-to-end TAD methods [9,10,14,15] jointly train video encoders and action detectors, enabling simultaneous modeling of local spatiotemporal features and global temporal dependencies, while offering notable advantages over feature-based methods [16–18] by facilitating knowledge transfer from pretrained models. Fine-tuning pretrained vision models, either fully [19,20] or partially, has shown strong performance across various downstream vision tasks. In TAD, recent end-to-end methods incorporate such fine-tuning to further improve performance. For example, Liu et al. [14] proposed the Temporal-Informative Adapter (TIA), which partially fine-tunes models like VideoMAE-S [21] and achieves notable improvements over traditional feature-based detection methods.

It is evident that integrating the strengths of end-to-end training with the fine-tuning of large vision models can significantly enhance the generalization ability and computational efficiency of TAD models, thereby facilitating progress toward higher accuracy and real-time performance. However, current end-to-end approaches in TAD predominantly rely on computationally intensive full fine-tuning, which often leads to issues such as catastrophic forgetting and overfitting during transfer learning—particularly when applied to downstream datasets with limited annotations, a common scenario in the TAD domain.

In this work, we aim to overcome the aforementioned limitations by enhancing TAD performance through the integration of vision foundation model fine-tuning and end-to-end training. In recent years, vision foundation models such as ViT [22] and VideoMAE [21] have demonstrated remarkable transferability across various video understanding tasks. This has motivated researchers to explore more efficient fine-tuning strategies that can fully leverage the pretrained knowledge while minimizing interference with the model's original parameters. Low-Rank Adaptation (LoRA) [23] introduces learnable low-rank matrices into pretrained models, allowing efficient adaptation to new tasks with minimal parameter overhead while mitigating overfitting.

We designed the Dynamic Low-Rank Adapter (DyLoRA) and applied it to the TAD task, forming the DyLoRA-TAD method. This approach significantly enhances the end-to-end training efficiency of TAD models through a lightweight architecture design. As shown in Fig. 1, the proposed method (DyLoRA-TAD) achieves remarkable results in the TAD task by cleverly combining the strengths of end-to-end training and fine-tuning strategies.

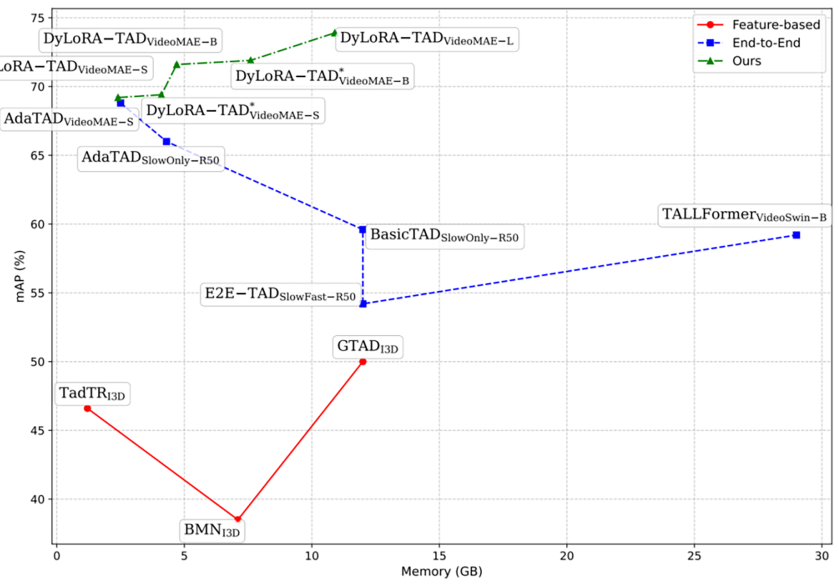


Figure 1: Comparison of different TAD models in terms of mean Average Precision (mAP, %, y-axis) and GPU memory usage during training (GB, x-axis). In the figure, the blue line represents the performance of a series of end-to-end trained TAD models, the red line represents the performance of feature-based TAD models, and the green line corresponds to the proposed DyLoRA-TAD methods

To achieve these goals, we first constructed an end-to-end baseline based on the efficient TAD framework Actionformer [18]. Experiments revealed that full fine-tuning introduces significant computational redundancy. To address this, we propose an adaptive rank adjustment mechanism that dynamically controls the update rank of adaptation parameters, balancing parameter size and performance. Based on this, we designed DyLoRA, a novel adapter deployed as the only learnable module between backbone layers. DyLoRA is optimized for TAD tasks and incorporates a temporal dimension weight optimization, effectively mitigating the loss of temporal feature representation caused by partial fine-tuning. Validated on multiple TAD datasets, DyLoRA-TAD achieves 73.9% mAP on THUMOS14, demonstrating the effectiveness of combining end-to-end training with parameter-efficient fine-tuning in TAD.

Our main contributions are summarized as follows:

1. We propose a lightweight Dynamic Low-Rank Adapter. Unlike traditional low-rank adaptation methods with static rank configuration, DyLoRA dynamically adjusts the update rank of the adaptation parameters, adaptively balancing computational resource consumption and model performance. Additionally, DyLoRA enables progressive optimization of the temporal dimension weights while freezing the pretrained spatial feature extractor, thereby enhancing the model's ability to model complex temporal dependencies in long video sequences.
2. Based on DyLoRA, we design an efficient end-to-end TAD framework named DyLoRA-TAD. By fine-tuning only a small subset of parameters in the pretrained model, the framework requires less than 5% of the parameters compared to full fine-tuning methods, yet achieves superior performance. This parameter-efficient fine-tuning strategy not only consistently boosts performance but also highlights the significance of vision foundation model fine-tuning in TAD. Furthermore, by adopting partial fine-tuning under different application scenarios, it effectively mitigates the computational overhead of full fine-tuning and the overfitting issue in low-resource settings.
3. This paper conducts in-depth research based on the DyLoRA-TAD framework, exploring the potential value of multi-scale feature aggregation in enhancing model performance. We constructed the Temporal DyLoRA Fuser module, which fine-tunes the weight allocation between components to achieve effective integration of multi-source feature information. This strategy not only strengthens the expressive capability of the original framework but also significantly improves overall performance, leading to the evolution of a more advanced DyLoRA-TAD*.

2 Related Work

2.1 Temporal Action Detection

Existing TAD methods can be mainly divided into three categories: one-stage methods [6,16,18,24], two-stage methods [22,25–28], and DETR-based methods [29,30]. One-stage methods refer to approaches that directly perform action localization and classification within a multi-scale feature pyramid framework. For example, TriDet [16] improves the pyramid structure to enhance model performance. One-stage methods are efficient for real-time processing but may lack precise localization of action boundaries in complex scenarios. Two-stage methods first use one stage for feature extraction and optimization, followed by a second stage to refine these features for improved accuracy. For instance, BMN [17] jointly optimizes boundary prediction and boundary matching processes, allowing for more precise localization of action segments and improving detection accuracy. Two-stage methods are suitable for handling complex action patterns but tend to have higher computational costs. DETR-based methods use the Transformer [31,32] architecture for end-to-end action detection. For example, TadTR [29] leverages self-attention mechanisms to capture long-range dependencies, making it particularly well-suited for multi-scale actions and complex background interference, demonstrating strong representational capabilities. MD-TAPN [16] introduces a truncated attention-aware proposal network with multi-scale dilation to improve action boundary localization and detection accuracy, especially for complex temporal patterns.

From the perspective of feature learning, TAD methods can be further divided into two categories: feature-based methods and end-to-end training methods. Feature-based methods [9,33–35] typically separate feature extraction and action detection into two distinct stages, relying on pre-extracted features such as RGB or optical flow. This results in a disconnection between feature representation and detection objectives. In contrast, end-to-end methods [9,11,14,29,36] jointly optimize feature extraction and detection tasks through deep learning models. For example, PBRNet [36] reduces redundant computation by predicting action boundaries and categories in parallel. PCL [37] proposes a prototype contrastive learning framework for point-supervised TAD, enhancing feature representation and action localization under weak supervision. ActionFormer [18] uses Transformers to encode global temporal context and directly outputs action segments; TadTR [29] introduces learnable query vectors to locate actions, simplifying the detection process. Despite the significant progress made by end-to-end methods, most still rely on full fine-tuning paradigms, which face the challenge of high computational costs, requiring further optimization of lightweight training strategies.

2.2 Parameter-Efficient Fine-Tuning

In recent years, Parameter-Efficient Fine-Tuning (PEFT) [38–41] has emerged as a research hotspot in deep learning. Compared to full fine-tuning methods, PEFT introduces a small number of trainable parameters into the pretrained model, significantly reducing computational costs and memory usage while preserving the integrity of the pretrained knowledge. In TAD tasks, the high-dimensional spatio-temporal features of video data often lead to a sharp increase in model parameters, making PEFT particularly important for reducing computational resource consumption.

Existing studies have explored a variety of PEFT strategies. For example, the early Adapter Tuning [42] inserts lightweight adaptation modules between Transformer layers to enable parameter-efficient updates. Subsequent methods like Prompt Tuning [43] and Prefix Tuning [44] introduce learnable prompt tokens to adapt models to specific scenarios, though their effectiveness is limited in long-sequence tasks. LoRA perturbs pretrained weights using low-rank decomposition of weight increments, enabling zero inference latency while preserving the original computational graph structure, offering a new perspective for joint visual-temporal modeling.

Although PEFT significantly reduces the number of trainable parameters, tasks such as video understanding, which are both data-intensive and computationally demanding, still require more efficient fine-tuning solutions. Our work builds on PEFT methods, particularly drawing inspiration from the LoRA mechanism, to explore its potential in TAD tasks for the first time, and adopts a more efficient design framework to enhance performance.

3 Methodology

In this section, we will progressively introduce DyLoRA-TAD. First, we will define some symbols and explore end-to-end TAD methods to establish an end-to-end baseline model. Next, we will introduce DyLoRA, a dynamic low-rank adapter designed specifically for efficient TAD. Finally, we propose the Temporal DyLoRA Fuser module, aimed at enhancing DyLoRA's ability to model temporal features across different dimensions, resulting in the formation of DyLoRA-TAD*.

3.1 Notations

The task definition of TAD is as follows: $X \in \mathbb{R}^{3\times h\times w\times t}$ represents an untrimmed video sequence, where h and w denote the height and width of each frame, respectively, and t is the number of frames. The action category set is denoted as $A = \{a_1, a_2, \cdots, a_N\}$, which includes all possible action types, with N representing the total number of action categories. The temporal action annotations are denoted as $G = \{g_i = (s_i, e_i, c_i)\}_{i=1}^{M}$, where each annotation g_i represents an action instance, including the start time, end time, and action category, and M is the total number of action instances. The predicted candidate proposals are denoted as $P = \{\hat{p}_j = (\hat{s}_j, \hat{e}_j, \hat{c}_j, \hat{\sigma}_j)\}_{j=1}^{K}$, where each proposal $\hat{p}_j$ includes the predicted start time $\hat{s}_j$, end time $\hat{e}_j$, action category $\hat{c}_j$, and confidence score $\hat{\sigma}_j$, with K representing the number of predicted action instances. The goal of TAD is to make P as close as possible to covering G.

3.2 End-to-End Baseline Model

The proposed end-to-end TAD architecture consists of two main components: feature extraction and action detection. To enable the model to better understand and distinguish action transitions in videos, we adopt ActionFormer as the action detection module.

In the feature extraction stage, Snippet Representation and Frame Representation are two common approaches for encoding raw video frames. The snippet representation method divides a video into multiple short segments (e.g., 16 frames each), with each snippet processed by a backbone network to extract segment-level features. In contrast, the frame representation method treats the entire video as a single snippet and feeds it directly into the network, extracting only spatially pooled features. This approach is particularly suitable for attention-based models, as it effectively reduces the complexity of temporal attention computation. According to previous studies [14], frame representation not only outperforms snippet representation in terms of performance but also consumes less memory. Therefore, we adopt frame representation to encode videos in our experiments.

In Table 1, when extending from the smaller video backbone network VideoMAE-S [21] to a larger-scale model, the simple end-to-end baseline we constructed experiences a significant surge in computational resource consumption during training. More importantly, this baseline employs the full fine-tuning approach, resulting in insufficient transfer learning capability. Furthermore, in low-resource settings, full fine-tuning is prone to overfitting or catastrophic forgetting. If the downstream task's dataset lacks sufficient diversity, it can even undermine the powerful feature representations learned from the pre-trained model.

Table 1: Full fine-tuning experiments of the end-to-end baseline on the THUMOS14 dataset. The input uses frame representation with dimensions of $1 \times 3 \times 768 \times 160 \times 160$

Setting	Backbone	0.3	0.4	0.5	0.6	0.7	Avg	Mem (GB)
End to End	VideoMAE-S	82.60	77.48	70.46	60.17	45.80	67.30	3.0
	VideoMAE-B	85.42	80.77	73.28	62.65	48.18	70.06	8.4
	VideoMAE-L	87.15	83.83	77.05	67.79	51.52	73.47	18.6

3.3 Dynamic Low-Rank Adapter

To address the issues of the end-to-end baseline, this paper adopts a parameter-efficient fine-tuning mechanism and proposes a fine-tuning module called DyLoRA to achieve efficient transfer learning for end-to-end TAD.

First, let's review the structure of the LoRA, which was proposed in research [23]. For a standard pre-trained Transformer model, its linear transformation is typically represented as:

$$Y = X * W \tag{1}$$

here, $X \in \mathbb{R}^{b\times d}$ represents the input data (where b is the batch size and d is the feature dimension), $W \in \mathbb{R}^{d\times d}$ is the original weight matrix, and $Y \in \mathbb{R}^{b\times d}$ is the output.

To avoid directly updating the original weight matrix W, LoRA introduces a low-rank adjustment term in a parallel branch. This adjustment term consists of two low-rank matrices, A and B, which are used to learn weight updates. Specifically, the adjustment term ΔW is formulated as:

$$\Delta W = B * A \tag{2}$$

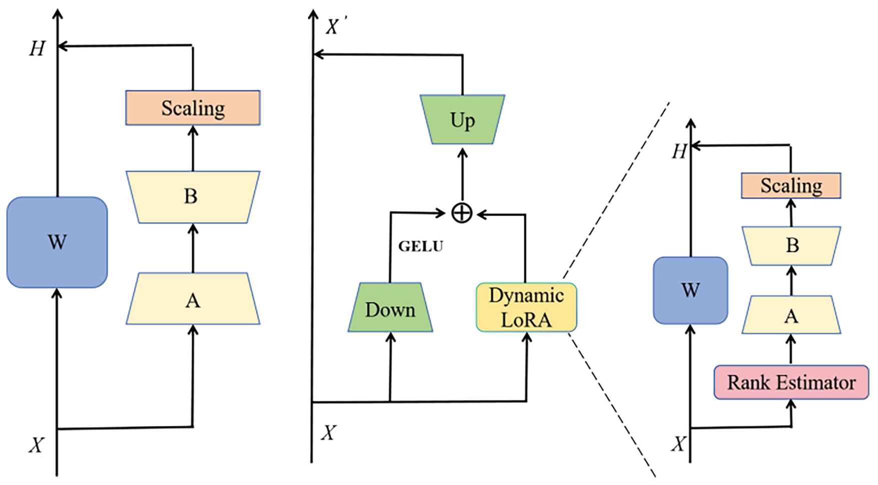
here, matrix $A \in \mathbb{R}^{d\times r}$ and matrix $B \in R^{r\times d}$ are low-rank matrices with rank r $(r << d)$.

Finally, the low-rank adjustment term ΔW is added to W, enabling fine-tuning of the original weight matrix. The linear transformation with LoRA is updated as:

$$Y = X * (W + \Delta W) = X * (W + B * A) \tag{3}$$

In LoRA, a scaling factor α/r is used to adjust the magnitude of the low-rank updates. Here, α is a hyperparameter. Its main purpose is to control the impact of the newly introduced parameters during the low-rank adaptation process, thereby preventing negative effects on the model caused by overly large parameter updates.

While LoRA enhances efficiency by fine-tuning low-rank decomposed weights, its fixed-rank constraint hinders the modeling of task-specific complexities, thus limiting adaptability to diverse scenarios. To overcome this limitation, we propose the DyLoRA, as shown in Fig. 2b.



(**a**) Low-Rank Adapter (**b**) Dynamic Low-Rank Adapter

Figure 2: Architecture of (**a**) LoRA Adapter and our (**b**) Dynamic Low-Rank Adapter (DyLoRA). We design a Rank Estimator to enable dynamic rank adjustment

DyLoRA retains the residual design of LoRA, where the input data $X \in \mathbb{R}^{d\times h\times w\times t}$ is a four-dimensional tensor. The specific structure is as follows: First, the input data is dimensionally reduced via a downsampling layer, and the Gaussian Error Linear Unit (GELU) activation function is inserted. This branch captures spatial features to provide auxiliary information for the DyLoRA layer. The formula is as follows:

$$X_{down} = \mathrm{GELU}(W_{down} \cdot X) \tag{4}$$

where $W_{down} \in \mathbb{R}^{d\times\frac{d}{\beta}}$ is a weight matrix, β is the reduction factor for the intermediate dimension, and $\beta > 1$.

Then, the DyLoRA layer is used for model parameter updates and low-rank adaptation. In the DyLoRA layer, we design a rank estimator. The rank estimator consists of three linear layers and ReLU activation functions, with the final layer using the Sigmoid activation function σ to ensure the output value is between 0 and 1. The rank estimator estimates a score based on the dimensionality of the input features (representing the proportion of the maximum rank), and then converts this score into the actual rank value, ensuring that the rank is between 1 and a preset maximum rank $r_{max} = 32$. The computation flow of the rank estimator is as follows:

$$X_{pool} = \frac{1}{h\cdot w}\sum_{i=1}^{h}\sum_{j=1}^{w} X_{d,i,j,t};\, X_{pool} \in \mathbb{R}^{d\times t} \tag{5}$$

$$\hat{r} = \sigma\left(W_3 \cdot \mathrm{ReLU}\left(W_2 \cdot \mathrm{ReLU}\left(W_1 \cdot X_{pool} + b_1\right) + b_2\right) + b_3\right) \tag{6}$$

$$r_e = \lfloor \hat{r} \cdot r_{max} \rfloor \tag{7}$$

here, X_{pool} is the two-dimensional vector obtained through average pooling, W_1, W_2, W_3, b_1, b_2 and b_3 are the weights and biases of the three linear layers, respectively. $\hat{r}$ is the probability score predicted by the rank estimator, and r_e is the actual rank computed by the rank estimator.

Following this, in DyLoRA, the low-rank matrix operation continues, where the rank r_e determined by the rank estimator is used for the initialization of matrices A_r and B_r. Based on the two low-rank matrices A_r and B_r, the pre-trained model parameters are fine-tuned, and the calculation formula is as follows:

$$A_r = A\left[:, : r_e\right]; B_r = B\left[: r_e, :\right] \tag{8}$$

$$X_{DyLoRA} = X\left(B_r \cdot A_r + W\right) \tag{9}$$

here, X_{DyLoRA} represents the output of the DyLoRA layer, W is the original weight matrix, with a dimension of $d \times d$.

Finally, the two branches are fused to obtain $\hat{X}$, and an upsampling layer is applied to restore the original dimensions. The computation is as follows:

$$\hat{X} = X_{down} + X_{DyLoRA} \tag{10}$$

$$X' = \gamma \cdot \hat{X} \cdot W_{up} + X \tag{11}$$

here, X' denotes the final output of the Dynamic Low-Rank Adapter, and γ is a learnable hyperparameter initialized to 1.

DyLoRA introduces a dynamic rank adjustment mechanism that adaptively tunes the rank of low-rank matrices and other key hyperparameters based on the characteristics of the input data. This allows the model to effectively address varying task demands and complexity. By leveraging the benefits of adaptive parameter modulation, DyLoRA flexibly optimizes the model during training to fully unleash the potential of large-scale pre-trained models, while overcoming the performance bottlenecks encountered by traditional LoRA in handling highly complex tasks. In particular, DyLoRA mitigates performance degradation caused by increasing rank through optimized update scaling, ensuring efficient and stable fine-tuning across diverse application scenarios. As a result, DyLoRA is especially well-suited for applications that seek optimal performance in resource-constrained environments, enhancing both model efficiency and expressiveness.

The overall architecture of our end-to-end training model is shown in Fig. 3. The architecture mainly consists of four parts: First, the feature embedding layer, which converts the input video into feature vectors;

Next, the frozen Transformer layers, which use pre-trained parameters and remain fixed during training, aimed at leveraging their powerful feature extraction capabilities to provide stable foundational feature representations; Then, the Dynamic Low-Rank Adapter module, which fine-tunes and adapts to different application scenarios; Finally, the detection head, which detects and locates action instances in the video from the refined feature representations.

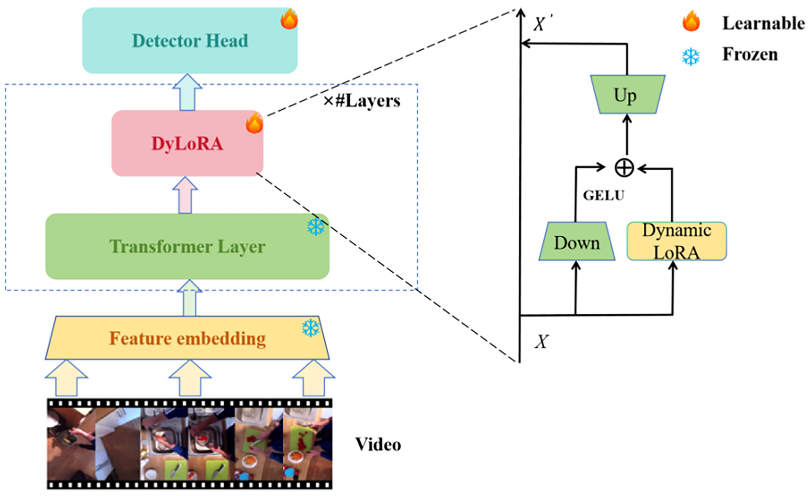


Figure 3: The overall architecture of DyLoRA-TAD for temporal action detection, including the feature embedding layer, frozen Transformer layers, learnable DyLoRA layers, and detection head

3.4 DyLoRA-TAD*

To further investigate the impact of multi-scale temporal feature fusion [45] on the optimization of the DyLoRA-TAD, we designed a simple yet effective module named Temporal DyLoRA Fuser. As illustrated in Fig. 4, this module is primarily integrated between each DyLoRA layer and the detection head, serving to enhance temporal feature representation across different scales.

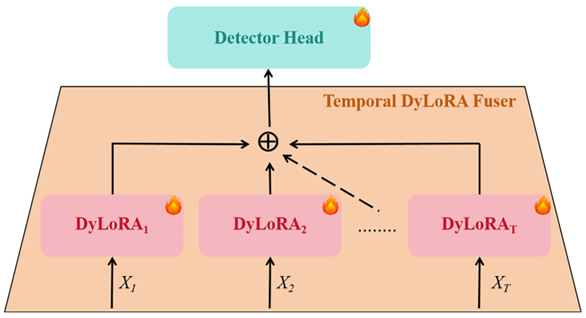


Figure 4: The Temporal DyLoRA Fuser. This module processes multi-scale temporal features through multiple DyLoRA layers and enhances the DyLoRA-TAD model's capability to understand and handle complex temporal sequences by fusing the outputs of these layers

First, we initialize a weight vector matrix $W = [w_1, w_2, \cdots, w_T]$ based on the number of adapters T. Then, we compute the Softmax values of the weight vector W, and expand them to match the shape of the DyLoRA layer outputs X'. The core implementation is as follows:

$$W' = \mathrm{Softmax}(W) \tag{12}$$

$$W'' = \text{Expand}(W') \tag{13}$$

Finally, a weighted summation is applied to fuse the temporal features extracted from different adapters, resulting in the fused representation X_{Fuser}, which is then fed into the detection head for action category classification. The detailed implementation is as follows:

$$X_{\text{Fuser}} = \sum_{i} X' \cdot W'' \tag{14}$$

$$\overline{X} = \text{Detector}(X_{\text{Fuser}}) \tag{15}$$

In this process, we introduce the concept of multi-scale temporal feature fusion, which captures dynamic changes in video sequences across different temporal scales. This approach not only enables the model to better understand the video content but also improves the accuracy of complex action recognition. Moreover, by applying the Softmax function to the weight vector, the relative importance of each adapter is effectively quantified, thus guiding the subsequent feature fusion process more precisely. Ultimately, the fused features generated through this method provide the detection head with richer and more representative information, contributing to the overall performance improvement of the model.

4 Experiments

4.1 Datasets and Evaluation Metrics

To validate the effectiveness of our proposed method, we conduct experiments on three widely used action detection datasets: THUMOS14 [46], ActivityNet-1.3 [47], and Charades [48]. THUMOS14 consists of 20 action categories and 413 untrimmed third-person videos, and it is widely adopted in action detection tasks. ActivityNet-1.3 covers a diverse range of daily activity categories, containing over 10,000 YouTube videos, making it a large-scale benchmark for video understanding. Charades focuses on multi-label action detection in everyday indoor scenarios and is particularly suitable for studying complex video understanding and action interaction.

Based on common evaluation standards, we use mean average precision (mAP) at different Temporal Intersection over Union (tIoU) thresholds and the average mAP as performance evaluation metrics. Specifically, for the ActivityNet-1.3 dataset, the tIoU thresholds are $\{0.5, 0.75, 0.95\}$; for the THUMOS14 dataset, the tIoU thresholds are $\{0.3, 0.4, 0.5, 0.6, 0.7\}$; and for the Charades dataset, we primarily evaluate the average mAP at different tIoU thresholds.

4.2 Implementation Details

The experiments were conducted using PyTorch 2.0 [49] and the MMAction2 [50] framework, with training performed on two NVIDIA GeForce RTX 4090 GPUs. The model's detection head utilizes the Actionformer architecture, and only the adapter parameters in the backbone network are trained, while the other parameters are frozen. The hyperparameter configurations for different datasets are detailed in Table 2. For data preprocessing, we followed previous research [14]. Considering the variation in action durations in the ActivityNet-1.3 dataset, we resized the videos to a fixed length of 768 frames. For the THUMOS14 and Charades datasets, as the videos are generally longer, we randomly cropped fixed-length segments for processing. To demonstrate the statistical reliability of the experimental results, we report the average and standard deviation (mean ± std) over three independent runs under the same settings.

Table 2: Experimental settings on different TAD datasets

Datasets	THUMOS14	ActivityNet-1.3	Charades
Preprocessing	Sliding window	Resize	Resize
Frame number	768	192 × 4	512
Adapter learning rate	2e−4	1e−4	4e−4
Frame resolution		160 × 160	
Warmup epoch	5	5	5
Total epoch	65	15	14

4.3 *Comparison and Analysis with State-of-the-Art Methods*

Table 3 shows a performance comparison of our proposed DyLoRA-TAD with other state-of-the-art methods on the THUMOS14 dataset. In the preliminary experiments, we used the VideoMAE-S version as the backbone network, which was previously applied in [14]. In contrast, our DyLoRA adapter outperforms its design based on deep convolution TIA adapters, achieving better performance with lower memory consumption. This comparison further highlights the advantages of LoRA fine-tuning and end-to-end training. The experimental results show that after applying DyLoRA for low-rank adaptation, the DyLoRA-TAD model achieves a 73.9% performance improvement on the THUMOS14 dataset.

Table 3: Results on the THUMOS14 and ActivityNet-1.3 datasets under different tIoU thresholds. "E2E" indicates end-to-end training, and "Mem" refers to the memory usage (in GB) during training. The best results are highlighted in bold, and the previous best results are underlined. To ensure a fair comparison, the baseline experiments from [14] in this table use the same backbone network as in our experiments

Method	Backbone	E2E	Flow	Mem (GB)	THUMOS14				ActivityNet-1.3			
					0.3	0.5	0.7	Avg.	0.5	0.75	0.95	Avg.
BMN [17]	TSN	✗	✓	–	56.0	38.8	20.5	38.5	50.1	34.8	8.3	33.9
TadTR [29]	I3D	✗	✓	–	62.5	49.2	26.3	46.6	49.1	32.6	8.5	32.3
ActionFormer [18]	SlowFast-R50	✗	✗	–	78.7	65.2	39.7	62.3	54.3	37.0	8.1	36.0
ActionFormer [18]	I3D	✗	✓	–	82.1	71.0	43.9	66.8	53.5	36.2	8.2	35.6
ASL [51]	I3D	✗	✓	–	83.1	71.7	45.8	67.9	54.1	37.4	8.0	36.2
DyFaDet [52]	I3D	✗	✓	–	84.0	72.7	47.9	69.2	–	–	–	–
DyFaDet [52]	R(2+1)D	✗	✓	–	–	–	–	–	58.1	39.6	8.4	38.5
TriDet [16]	I3D	✗	✓	–	83.6	72.9	47.4	69.3	54.7	38.0	8.4	36.8
FDDet [53]	I3D	✗	✓	–	83.5	72.8	48.2	69.4	–	–	–	–
FDDet [53]	R(2+1)D	✗	✓	–	–	–	–	–	57.8	39.3	8.7	38.2
AFSD [33]	I3D	✓	✓	12	67.3	55.5	31.1	52.0	52.4	35.3	6.5	34.4
E2E-TAD [35]	SlowFast-R50	✓	✗	12	69.4	56.0	34.9	54.2	50.5	36.0	10.3	35.1
BasicTAD [15]	SlowOnly-R50	✓	✗	12	75.5	63.5	37.4	59.6	51.2	33.4	7.6	33.1
TALLFormer [10]	VideoSwin-B	✓	✗	29	76.0	63.2	34.5	59.2	54.1	36.2	7.9	35.6
Re2TAL [11]	Re2VideoSwin-T	✓	✗	24	77.0	62.4	36.3	59.4	54.8	37.8	9.0	36.8
AdaTAD [14]	SlowFast-R50	✓	✗	4.3	81.0	69.4	44.5	66.0	55.3	38.1	8.9	37.1
AdaTAD [14]	VideoMAE-S	✓	✗	2.5	84.5	71.6	46.9	68.8	56.2	39.0	9.1	37.9
AdaTAD [14]	VideoMAE-B	✓	✗	4.9	87.0	75.3	49.2	71.5	56.8	39.4	9.7	38.4
AdaTAD [14]	VideoMAE-L	✓	✗	11	87.7	76.7	52.4	73.5	57.7	40.6	10.1	39.2
DyLoRA-TAD	VideoMAE-S	✓	✗	2.4	84.7	72.5	46.7	69.3 ± 0.3	56.4	38.8	9.3	37.9 ± 0.1
DyLoRA-TAD*	VideoMAE-S	✓	✗	4.1	84.2	72.4	48.4	69.5 ± 0.1	56.4	39.1	9.4	38.2 ± 0.2
DyLoRA-TAD	VideoMAE-B	✓	✗	4.7	86.1	73.7	50.7	71.7 ± 0.2	57.0	39.4	9.6	38.6 ± 0.2
DyLoRA-TAD*	VideoMAE-B	✓	✗	7.6	85.8	74.6	51.2	71.9 ± 0.2	57.3	39.7	9.7	38.9 ± 0.1
DyLoRA-TAD	VideoMAE-L	✓	✗	**10.9**	**88.5**	**77.3**	**51.8**	**73.9 ± 0.1**	**57.9**	**40.8**	**10.3**	**39.5 ± 0.2**

In addition, we conducted exploratory experiments with the DyLoRA-TAD* variant on the THUMOS14 dataset. The results show that, whether using VideoMAE-S [21] or VideoMAE-B [21] as the backbone network, DyLoRA-TAD* consistently outperforms the original DyLoRA-TAD model. However, when training the DyLoRA-TAD* model with the VideoMAE-L [21] backbone in an end-to-end manner, we encountered GPU memory overflow due to the large model size and limited GPU resources. To mitigate memory limitations, future work could adopt gradient checkpointing, mixed-precision training, or smaller backbone variants. Experiments with smaller VideoMAE backbones show that DyLoRA significantly reduces memory usage and remains scalable for long videos under standard hardware. Nevertheless, based on the completed experimental results, we can confirm that the Temporal DyLoRA Fuser module provides a clear performance boost to the DyLoRA-TAD.

Meanwhile, Table 3 presents the mAP performance of different methods on the ActivityNet-1.3 dataset under various tIoU thresholds. The experimental results demonstrate that our proposed DyLoRA-TAD method significantly outperforms other approaches across all evaluation metrics. As the scale of the backbone network increases, DyLoRA-TAD shows a clear upward trend in performance. Notably, DyLoRA-TAD excels under higher tIoU thresholds, highlighting its strong capability for precise temporal localization. Moreover, compared to state-of-the-art methods such as AdaTAD [14] and TriDet [16], DyLoRA-TAD achieves an average mAP of 39.52% across the entire tIoU range, surpassing all baselines and further emphasizing its superior performance.

We also present the performance of DyLoRA-TAD on the Charades dataset (see Table 4). When using VideoMAE-L [21] as the backbone network, the model achieved an average mAP of 27.8%, demonstrating its efficient performance in action detection. Additionally, we conducted experiments with the DyLoRA-TAD* variant on the Charades dataset. The results showed that the model achieved an average mAP of 28.4%, further confirming that the Temporal DyLoRA Fuser effectively optimizes the DyLoRA-TAD model.

Table 4: Average mAP (%) results of DyLoRA-TAD on the Charades dataset. This table primarily discusses the average performance of various models under different tIoU thresholds, based on findings from previous research

Method	Backbone	Charades (Avg.)
MLAD [3]	I3D	18.4
SuperEvent [54]	I3D	18.6
TGM [55]	I3D	20.6
PointTAD [2]	I3D	22.1
DualDETR [1]	I3D	23.2
PDAN [56]	I3D	23.7
CoarseFine [57]	X3D	25.1
MS-Temba [4]	I3D	25.3
CTRN [11]	I3D	25.3
MS-TCT [6]	I3D	25.4
DyLoRA-TAD	VideoMAE-S	20.2 ± 0.2
DyLoRA-TAD	VideoMAE-B	23.9 ± 0.2
DyLoRA-TAD	VideoMAE-L	27.8 ± 0.1
DyLoRA-TAD*	VideoMAE-S	21.4 ± 0.2
DyLoRA-TAD*	VideoMAE-B	24.6 ± 0.3
DyLoRA-TAD*	VideoMAE-L	**28.4 ± 0.1**

Overall, as model complexity and video processing improve, average mAP increases, highlighting the importance of choosing suitable architectures and backbones for TAD. While DyLoRA-TAD performs strongly, it still has limitations: on THUMOS14, short-duration actions are hard to localize precisely; on ActivityNet-1.3, distinguishing long, semantically similar actions is challenging, causing drops at high tIoU; and on Charades, overlapping actions and complex scenes lead to category confusion and missed detections. Future work should enhance temporal modeling and context awareness to improve robustness in complex scenarios.

4.4 Ablation Study

In this section, we further evaluate the proposed DyLoRA-TAD method through ablation studies and validate the advantages of LoRA fine-tuning in improving TAD performance and reducing memory consumption. All of our ablation experiments are conducted on the THUMOS14 dataset.

The ablation study on adapter design (see Table 5) compares different architecture designs. The baseline model with a frozen backbone achieved a mAP of 58.4%, while our DyLoRA-TAD method reached 69.2%, representing a 10.8% improvement. Compared to TIA [14], DyLoRA achieves higher mAP with similar memory consumption, highlighting its significance in TAD tasks. Additionally, RaRA [58] and the Wavelet transform [59] achieve performance close to DyLoRA but with significantly higher memory usage. Removing the rank estimator in DyLoRA results in a 0.3% performance drop and increased memory consumption, indicating the critical role of the rank estimator in maintaining both performance and efficiency. Regarding the computational overhead of dynamic rank prediction, our quantitative experiments show that the extra cost is negligible: memory usage slightly decreases (2.8 G → 2.5 G) while mAP improves (68.9 → 69.2). This indicates that dynamic rank prediction does not offset DyLoRA's lightweight advantage, and its initialization and update strategies further optimize both performance and memory efficiency.

Table 5: Ablation study showing different adapter architecture designs, with the backbone network using the VideoMAE-S version. When the rank estimator is removed, we set the rank to 8 for the experiment

Setting	Mem.	mAP	Gains
Frozen	2.0 G	58.4	
+Full Fine-Tuning	3.0 G	67.3	+8.9
+TIA [14]	2.5 G	68.6	+10.2
+RoRA [58]	2.9 G	69.0	+10.6
+Wavelet transform [59]	4.1 G	69.1	+10.7
+DyLoRA (w/o Rank Estimator)	2.8 G	68.9	+10.5
+DyLoRA	**2.5 G**	**69.2**	**+10.8**

4.5 Experiments on Parameters

Based on the ablation study of the Rank Estimator in Section 4.4, we further investigate the impact of setting a fixed rank value on model performance. Detailed experimental data can be found in Table 6, where the rank value is set in the range of 8 to 32 with a step size of 4. The results show that when using VideoMAE-B as the backbone network, setting the rank to 28 yields the best model performance with an accuracy of 71.9%. When the backbone network is extended from the VideoMAE-B version to the VideoMAE-L version, setting the rank to 20 achieves performance comparable to that of the original DyLoRA-TAD.

Table 6: The experimental results for the static rank parameter are presented, where the DyLoRA-TAD' model refers to the version with the Rank Estimator removed, characterized by fixing the rank to a specific value

Method	Backbone	Rank	THUMOS14					
			0.3	**0.4**	**0.5**	**0.6**	**0.7**	**Avg.**
DyLoRA-TAD$'$	VideoMAE-B	8	84.8	80.2	72.8	62.4	48.9	69.8
		12	85.0	80.5	73.1	63.0	49.6	70.2
		16	84.7	80.1	72.9	62.2	47.0	69.4
		20	85.3	81.3	73.4	63.5	48.3	70.4
		24	84.7	80.2	73.9	63.0	48.9	70.1
		28	**85.9**	**82.9**	**74.7**	**64.8**	**51.0**	**71.9**
		32	85.5	80.3	73.5	63.6	48.7	70.1
DyLoRA-TAD$'$	VideoMAE-L	8	87.0	82.9	76.0	66.2	51.4	72.7
		12	86.7	83.1	76.3	66.6	52.2	73.0
		16	86.4	82.8	75.8	66.1	51.3	72.5
		20	**88.2**	**84.5**	**77.0**	**66.6**	**51.5**	**73.9**
		24	87.0	82.9	76.0	66.2	51.4	72.7
		28	86.7	83.7	76.7	66.7	52.0	73.2
		32	87.0	83.0	76.7	66.5	51.1	72.8

Although setting a fixed rank can achieve performance close to that of dynamic rank adaptation, we observe that this approach significantly increases memory overhead. For example, the ablation study in Table 5 shows that using the Rank Estimator reduces GPU memory usage from 2.8 GB to 2.5 GB, significantly alleviating the resource consumption issue. This also demonstrates from another perspective that DyLoRA not only effectively improves model performance but also makes more efficient use of resources.

5 Conclusion

This study proposes the DyLoRA-TAD framework to address the high computational cost and small-sample overfitting issues faced during fine-tuning large-scale pre-trained video models for TAD. We design a dynamic low-rank adaptive adapter, DyLoRA, which introduces low-rank matrix decomposition techniques to efficiently fine-tune the critical layers of pre-trained models. This approach significantly reduces training parameters and memory consumption while effectively alleviating overfitting. Specifically, DyLoRA enhances the model's ability to capture temporal dependencies, addressing the inadequacies of traditional pre-trained models in temporal feature representation. Furthermore, we explore the performance improvement brought by the multi-scale temporal feature fusion module, Temporal DyLoRA Fuser. DyLoRA-TAD, through end-to-end training and task-oriented efficient fine-tuning, enhances the model's generalization ability and stability, achieving breakthroughs in detection accuracy and providing a lightweight, efficient, and practical solution for real-world applications such as video anomaly detection. We believe that DyLoRA, as a lightweight and efficient fine-tuning solution, demonstrates significant potential in the field of temporal action detection and offers important insights and references for the future development of TAD methods.

Acknowledgement: The authors would like to acknowledge the support from the National Natural Science Foundation of China, the Major Science and Technology Project of Yunnan Province, and the Scientific Research Fund of the Yunnan Provincial Department of Education.

Funding Statement: This work was partially supported by the National Natural Science Foundation of China (Grant No. 62266054), the Major Science and Technology Project of Yunnan Province (Grant No. 202402AD080002), and the Scientific Research Fund of the Yunnan Provincial Department of Education (Grant No. 2025Y0302).

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, Jixin Wu and Mingtao Zhou; methodology, Jixin Wu and Mingtao Zhou; software, Jixin Wu; validation, Jixin Wu, Mingtao Zhou and Di Wu; formal analysis, Jixin Wu; investigation, Jixin Wu and Mingtao Zhou; resources, Wenqi Ren and Jiatian Mei; data curation, Jixin Wu; writing—original draft preparation, Jixin Wu; writing—review and editing, Jixin Wu, Mingtao Zhou and Shu Zhang; visualization, Jixin Wu; supervision, Shu Zhang; project administration, Shu Zhang; funding acquisition, Shu Zhang. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data that support the findings of this study are openly available in the OpenTAD GitHub repository at https://github.com/sming256/OpenTAD (accessed on 01 September 2025).

Ethics Approval: Not applicable.

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

References

1. Zhu Y, Zhang G, Tan J, Wu G, Wang L. Dual DETRs for multi-label temporal action detection. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 18559–69. doi:10.1109/CVPR52733.2024.01756.
2. Tan J, Zhao X, Shi X, Kang S, Wang L, Sun Y, et al. PointTAD: multi-label temporal action detection with learnable query points. Adv Neural Inf Process Syst. 2022;35:15268–80.
3. Tirupattur P, Duarte K, Rawat YS, Shah M. Modeling multi-label action dependencies for temporal action localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 1460–70.
4. Sinha A, Raj MS, Wang P, Helmy A, Das S. MS-temba: multi-scale temporal mamba for efficient temporal action detection. arXiv:2501.06138. 2025.
5. Dai R, Das S, Bremond F. CTRN: class-temporal relational network for action detection. arXiv:2110.13473. 2021.
6. Dai R, Das S, Kahatapitiya K, Ryoo MS, Brémond F. MS-TCT: multi-scale temporal ConvTransformer for action detection. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–4; New Orleans, LA, USA. p. 20009–19. doi:10.1109/CVPR52688.2022.01941.
7. Lysova T. Intersecting perspectives: video surveillance in urban spaces through surveillance society and security state frameworks. Cities. 2025;156(1):105544. doi:10.1016/j.cities.2024.105544.
8. Wu J, Mou S, Zhou M, Wu D, Zou S, Zhang S. The student classroom behavior recognition based on feature enhancement. In: Proceedings of the 2024 IEEE 8th International Conference on Vision, Image and Signal Processing (ICVISP); 2024 Dec 27–29; Kunming, China. p. 1–7. doi:10.1109/ICVISP64524.2024.10959424.
9. Yang M, Chen G, Zheng YD, Lu T, Wang L. BasicTAD: an astounding RGB-only baseline for temporal action detection. Comput Vis Image Underst. 2023;232(1):103692. doi:10.1016/j.cviu.2023.103692.
10. Cheng F, Bertasius G. TallFormer: temporal action localization with a long-memory transformer. In: Proceedings of the Computer Vision—ECCV 2022; 2022 Oct 23–27; Tel Aviv, Israel. p. 503–21. doi:10.1007/978-3-031-19830-4_29.
11. Zhao C, Liu S, Mangalam K, Ghanem B. Re2TAL: rewiring pretrained video backbones for reversible temporal action localization. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. p. 10637–47. doi:10.1109/CVPR52729.2023.01025.
12. Xin Y, Yang J, Luo S, Du Y, Qin Q, Cen K, et al. Parameter-efficient fine-tuning for pre-trained vision models: a survey and benchmark. arXiv:2402.02242. 2024.

13. Bai H, LeCun Y, Levine S, Lin Z, Ma Y, Pan J, et al. Fine-tuning large vision-language models as decision-making agents via reinforcement learning. In: Proceedings of the Advances in Neural Information Processing Systems 37; 2024 Dec 10–15; Vancouver, BC, Canada. p. 110935–71. doi:10.52202/079017-3522.
14. Liu S, Zhang CL, Zhao C, Ghanem B. End-to-end temporal action detection with 1B parameters across 1000 frames. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 18591–601. doi:10.1109/CVPR52733.2024.01759.
15. Piergiovanni AJ, Ryoo M. Temporal gaussian mixture layer for videos. In: Proceedings of the 36th International Conference on Machine Learning (ICML); 2019 Jun 9–15; Long Beach, CA, USA. p. 5152–61.
16. Shi D, Zhong Y, Cao Q, Ma L, Lit J, Tao D. TriDet: temporal action detection with relative boundary modeling. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2023 Jun 17–24; Vancouver, BC, Canada. p. 18857–66. doi:10.1109/CVPR52729.2023.01808.
17. Lin T, Liu X, Li X, Ding E, Wen S. BMN: boundary-matching network for temporal action proposal generation. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 2019 Oct 27–Nov 2; Seoul, Republic of Korea. p. 3888–97. doi:10.1109/iccv.2019.00399.
18. Zhang CL, Wu J, Li Y. ActionFormer: localizing moments of actions with transformers. In: Proceedings of the Computer Vision—ECCV 2022; 2022 Oct 23–27; Tel Aviv, Israel. p. 492–510. doi:10.1007/978-3-031-19772-7_29.
19. Radford A, Kim JW, Hallacy C, Ramesh A, Goh G, Agarwal S, et al. Learning transferable visual models from natural language supervision. In: Proceedings of the 38th International Conference on Machine Learning (ICML); 2021 Jul 18–24; Online. p. 8748–63.
20. Yang T, Zhu Y, Xie Y, Zhang A, Chen C, Li M. AIM: adapting image models for efficient video action recognition. arXiv:2302.03024. 2023.
21. Tong Z, Song Y, Wang J, Wang L. VideoMAE: masked autoencoders are data-efficient learners for self-supervised video pre-training. In: Proceedings of the Advances in Neural Information Processing Systems 35 (NeurIPS 2022); 2022 Nov 28–Dec 9; New Orleans, LA, USA. p. 10078–93.
22. Patro BN, Namboodiri VP, Agneeswaran VS. SpectFormer: frequency and attention is what you need in a vision transformer. In: Proceedings of the 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2025 Feb 26–Mar 6; Tucson, AZ, USA. p. 9543–54. doi:10.1109/WACV61041.2025.00924.
23. Hu EJ, Shen Y, Wallis P, Allen-Zhu Z, Li Y, Wang S, et al. LoRA: low-rank adaptation of large language models. arXiv:2106.09685. 2021.
24. Li W, Yao D, Gong C, Chu X, Jing Q, Zhou X, et al. CausalTAD: causal implicit generative model for debiased online trajectory anomaly detection. In: Proceedings of the 2024 IEEE 40th International Conference on Data Engineering (ICDE); 2024 May 13–16; Utrecht, The Netherlands. p. 4477–90. doi:10.1109/ICDE60146.2024.00341.
25. Xu M, Zhao C, Rojas DS, Thabet A, Ghanem B. G-TAD: sub-graph localization for temporal action detection. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 10153–62. doi:10.1109/cvpr42600.2020.01017.
26. Xia K, Wang L, Zhou S, Zheng N, Tang W. Learning to refactor action and co-occurrence features for temporal action localization. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 13874–83. doi:10.1109/CVPR52688.2022.01351.
27. Qing Z, Su H, Gan W, Wang D, Wu W, Wang X, et al. Temporal context aggregation network for temporal action proposal refinement. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 485–94. doi:10.1109/CVPR46437.2021.00055.
28. Zhao Z, Wang D, Zhao X. Movement enhancement toward multi-scale video feature representation for temporal action detection. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Oct 1–6; Paris, France. p. 13509–18. doi:10.1109/ICCV51070.2023.01247.
29. Liu X, Wang Q, Hu Y, Tang X, Zhang S, Bai S, et al. End-to-end temporal action detection with transformer. IEEE Trans Image Process. 2022;31:5427–41. doi:10.1109/TIP.2022.3195321.
30. Shi D, Zhong Y, Cao Q, Zhang J, Ma L, Li J, et al. ReAct: temporal action detection with relational queries. In: Proceedings of the Computer Vision—ECCV 2022; 2022 Oct 23–27; Tel Aviv, Israel. p. 105–21. doi:10.1007/978-3-031-20080-9_7.

31. Han K, Xiao A, Wu E, Guo J, Xu C, Wang Y. Transformer in transformer. In: Proceedings of the Advances in Neural Information Processing Systems 34 (NeurIPS 2021); 2021 Dec 6–14; Online. p. 15908–19.
32. Han K, Wang Y, Chen H, Chen X, Guo J, Liu Z, et al. A survey on vision transformer. IEEE Trans Pattern Anal Mach Intell. 2023;45(1):87–110. doi:10.1109/TPAMI.2022.3152247.
33. Lin C, Xu C, Luo D, Wang Y, Tai Y, Wang C, et al. Learning salient boundary feature for anchor-free temporal action localization. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 3319–28. doi:10.1109/cvpr46437.2021.00333.
34. Wang C, Cai H, Zou Y, Xiong Y. RGB stream is enough for temporal action detection. arXiv:2107.04362. 2021.
35. Liu X, Bai S, Bai X. An empirical study of end-to-end temporal action detection. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2022 Jun 18–24; New Orleans, LA, USA. p. 19978–87. doi:10.1109/CVPR52688.2022.01938.
36. Dai P, Li Z, Zhang Y, Liu S, Zeng B. PBR-net: imitating physically based rendering using deep neural network. IEEE Trans Image Process. 2020;29:5980–92. doi:10.1109/TIP.2020.2987169.
37. Li P, Cao J, Ye X. Prototype contrastive learning for point-supervised temporal action detection. Expert Syst Appl. 2023;213:118965. doi:10.1016/j.eswa.2022.118965.
38. Ding N, Qin Y, Yang G, Wei F, Yang Z, Su Y, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. Nat Mach Intell. 2023;5(3):220–35. doi:10.1038/s42256-023-00626-4.
39. Fu Z, Yang H, So AM, Lam W, Bing L, Collier N. On the effectiveness of parameter-efficient fine-tuning. Proc AAAI Conf Artif Intell. 2023;37(11):12799–807. doi:10.1609/aaai.v37i11.26505.
40. Han Z, Gao C, Liu J, Zhang J, Zhang SQ. Parameter-efficient fine-tuning for large models: a comprehensive survey. arXiv:2403.14608. 2024.
41. Gu J, Yuan J, Cai J, Zhou X, Fan L. La-LoRA: parameter-efficient fine-tuning with layer-wise adaptive low-rank adaptation. Neural Netw. 2025;194(10):108095. doi:10.1016/j.neunet.2025.108095.
42. Zhang R, Han J, Liu C, Gao P, Zhou A, Hu X, et al. LLaMA-adapter: efficient fine-tuning of language models with zero-init attention. arXiv:2303.16199. 2023.
43. Jia M, Tang L, Chen BC, Cardie C, Belongie S, Hariharan B, et al. Visual prompt tuning. In: Proceedings of the Computer Vision—ECCV 2022; 2022 Oct 23–27; Tel Aviv, Israel. p. 709–27. doi:10.1007/978-3-031-19827-4_41.
44. Li XL, Liang P. Prefix-tuning: optimizing continuous prompts for generation. arXiv:2101.00190. 2021.
45. Zhang Y, Zhang T, Wu C, Tao R. Multi-scale spatiotemporal feature fusion network for video saliency prediction. IEEE Trans Multimed. 2024;26:4183–93. doi:10.1109/TMM.2023.3321394.
46. Jiang YG, Liu J, Zamir AR, Laptev I. THUMOS Challenge 2015 [Internet]. 2015 [cited 2025 Sep 1]. Available from: http://www.thumos.info/.
47. Heilbron FC, Escorcia V, Ghanem B, Niebles JC. ActivityNet: a large-scale video benchmark for human activity understanding. In: Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2015 Jun 7–12; Boston, MA, USA. p. 961–70. doi:10.1109/CVPR.2015.7298698.
48. Sigurdsson GA, Varol G, Wang X, Farhadi A, Laptev I, Gupta A. Hollywood in homes: crowdsourcing data collection for activity understanding. In: Proceedings of the Computer Vision—ECCV 2016; 2016 Oct 11–14; Amsterdam, The Netherlands. p. 510–26. doi:10.1007/978-3-319-46448-0_31.
49. Melo MA. Accelerate model training with PyTorch 2.X: build more accurate models by boosting the model training process. Birmingham, UK: Packt Publishing Limited; 2024. doi:10.0000/9781805121916.
50. MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark [Internet]. 2020 [cited 2025 Sep 1]. Available from: https://github.com/open-mmlab/mmaction2.
51. Shao J, Wang X, Quan R, Zheng J, Yang J, Yang Y. Action sensitivity learning for temporal action localization. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Oct 1–6; Paris, France. p. 13411–23. doi:10.1109/iccv51070.2023.01238.
52. Yang L, Zheng Z, Han Y, Cheng H, Song S, Huang G, et al. DyFADet: dynamic feature aggregation for temporal action detection. In: Proceedings of the Computer Vision—ECCV 2024; 2024 Sep 29–Oct 4; Milan, Italy. p. 305–22. doi:10.1007/978-3-031-72952-2_18.

53. Zhu X, Zhu Y, Chen T, Wu W, Dang Y. FDDet: frequency-decoupling for boundary refinement in temporal action detection. arXiv:2504.00647. 2025.
54. Piergiovanni A, Ryoo MS. Learning latent super-events to detect multiple activities in videos. In: Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2018 Jun 18–23; Salt Lake City, UT, USA. p. 5304–13. doi:10.1109/CVPR.2018.00556.
55. Liu S, Xu M, Zhao C, Zhao X, Ghanem B. ETAD: training action detection end to end on a laptop. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW); 2023 Jun 17–24; Vancouver, BC, Canada. p. 4525–34. doi:10.1109/cvprw59228.2023.00476.
56. Dai R, Das S, Minciullo L, Garattoni L, Francesca G, Bremond F. PDAN: pyramid dilated attention network for action detection. In: Proceedings of the 2021 IEEE Winter Conference on Applications of Computer Vision (WACV); 2021 Jan 3–8; Waikoloa, HI, USA. p. 2969–78. doi:10.1109/wacv48630.2021.00301.
57. Kahatapitiya K, Ryoo MS. Coarse-fine networks for temporal activity detection in videos. In: Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2021 Jun 20–25; Nashville, TN, USA. p. 8381–90. doi:10.1109/CVPR46437.2021.00828.
58. Liu J, Kong Z, Dong P, Yang C, Shen X, Zhao P, et al. RoRA: efficient fine-tuning of LLM with reliability optimization for rank adaptation. arXiv:2501.04315. 2025.
59. Zhang D. Wavelet transform. In: Fundamentals of image data mining. Cham, Switzerland: Springer; 2019. p. 35–44. doi:10.1007/978-3-030-17989-2_3.