ITC 2/54

Information Technology and Control
Vol. 54 / No. 2/ 2025
pp. 735-750
DOI 10.5755/j01.itc.54.2.41005

Towards Real-World Power Grid Scenarios: Video Action Detection with Cross-scale Selective Context Aggregation

Received 2025/03/27 | Accepted after revision 2025/05/19

HOW TO CITE: Meng, L., Yu, S., Luo, S., Li, A. (2025). Towards Real-World Power Grid Scenarios: Video Action Detection with Cross-scale Selective Context Aggregation. *Information Technology and Control,* 54(2), 735-750. https://doi.org/10.5755/j01.itc.54.2.41005

Towards Real-World Power Grid Scenarios: Video Action Detection with Cross-scale Selective Context Aggregation

Lingwen Meng*, Siwu Yu, Shasha Luo, Anjun Li

Electric Power Research Institute of Guizhou Power Grid Co. Ltd, Guiyang 450046, China

Corresponding author: mengyao@ncwu.edu.cn

In this study, we propose a single-stage model for video action detection and a real-world action detection dataset POWER collected from real power operation scenarios. While previous studies have made significant progress in overall classification and localization performance, they often struggle with the actions that have short duration, hindering the application of these approaches. To address this, we introduce the Cross-scale Selective Context Aggregation Network (CSCAN), which focuses on improving the detection of short actions. This network integrates three key components: 1) a cross-scale feature conduction structure combined with a tailored alignment mechanism; 2) a selective context aggregation module based on gating mechanism; and 3) an effective scale-invariant consistency training strategy to enable the model to learn scale-invariant action representation. We evaluated our method on the self-collected dataset POWER and on the most widely used action detection benchmarks THUMOS14 and ActivityNet v1.3. The extensive results show that our model outperforms other approaches, especially in detecting real-world short actions, demonstrating the effectiveness of our approach.

KEYWORDS: Action Detection, Deep Learning, Video Understanding.

1. Introduction

Image-based Object Detection (OD) has been widely applied in diverse scenarios [11], [13]. However, in power grid operation scenes, significant environmental interference and complex relationships among multiple targets make traditional OD techniques are no longer sufficient to support the compliance audits of complex grid operations. Therefore, the importance of Temporal Action Detection (TAD) based on consecutive video frames is growing increasingly significant. TAD aims to identify and localize human actions in untrimmed videos by assigning semantic labels and precise temporal boundaries (start and end points) to each action instance.

In our collected POWER dataset, action durations are highly dispersed, as an example showing in Figure 1. Similar issues also exist in public datasets. Experiments show that detecting short actions is often more challenging even with the powerful TAD model ActionFormer [44]. When we experimented with various baselines, we found that models which capture context information more completely tend to perform better in detecting short actions. Take several models with similar structures as examples: TemporalMaxer [30], ActionFormer [44], and ActionMamba [4]. Their key backbone modules use max-pooling, local attention, and selective state space model (DB-Mamba [6], a bi-directional recurrent structure) for context modeling, respectively. Clearly, in terms of the completeness of context information, ActionMamba [4] performs the best, followed by ActionFormer [44], and TemporalMaxer [30] performs the worst, as shown in the Figure 2. The experimental results correspond to their respective performances.

This phenomenon has inspired us to further enrich the modeling of context. We attempted to conduct context modeling from a broader perspective, where shallower network layers can also obtain processed contextual information from deeper layers, as shown in the right of Figure 2. This is reflected in the back propagation of features from deeper to shallower network layers. Based on this idea, we proposed the Cross-scale Selective Context Aggregation Network (CSCAN), which incorporates three key components:

- First, a cross-scale feature conduction structure named Top-Down Pathway (TDP) combined with a tailored alignment mechanism in section 3.3. Overall, TDP is a macroscopic structure, where features from upper layers are gradually integrated into lower layers. This involves an alignment mechanism to align features of different temporal sizes.
- Second, a selective context fusion module based on gating mechanism, Cross-scale Selective Fusion (CSF), in section 3.4. Other reference fusion methods often employ direct addition [17], [20] or layer-level weights [43] on cross-scale features, whereas we go a step further by using a point-to-point level gating mechanism to control feature fusion across different scales.
- Third, we use scale consistency training to promote model learning scale-invariant representation in section 3.6. This part introduces controlled scale perturbations to the input data during training through Gaussian sampling of scaling factors. This forces the model to learn scale-invariant features, reducing its bias towards detecting actions of specific durations. The scaling is applied to both the input features and corresponding labels, ensuring consistency throughout the model.

In addition, we introduce the POWER dataset, which comprises 430 hours of video footage captured from

Figure 1

An example from dataset POWER. In this video, four types of actions occur, as illustrated in the legend. Among these four actions, the longest one is ”Put Grounding Rod” (indicated by the green segment), while the shortest is ”Verification” (indicated by the blue segment).

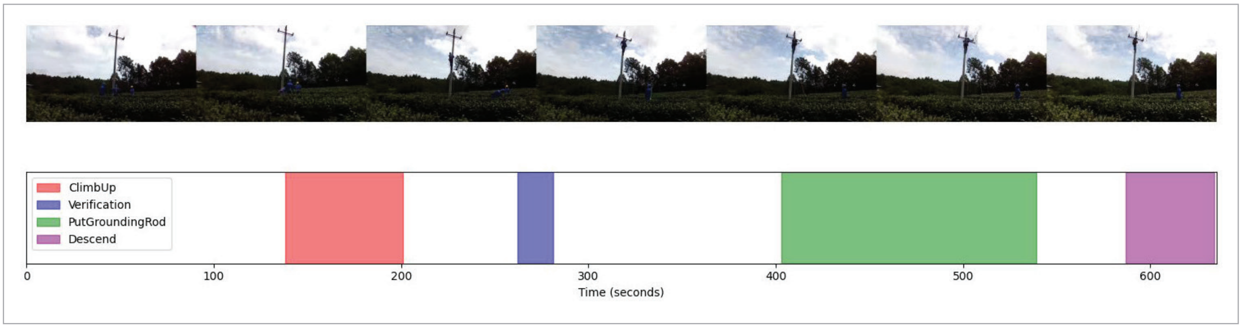
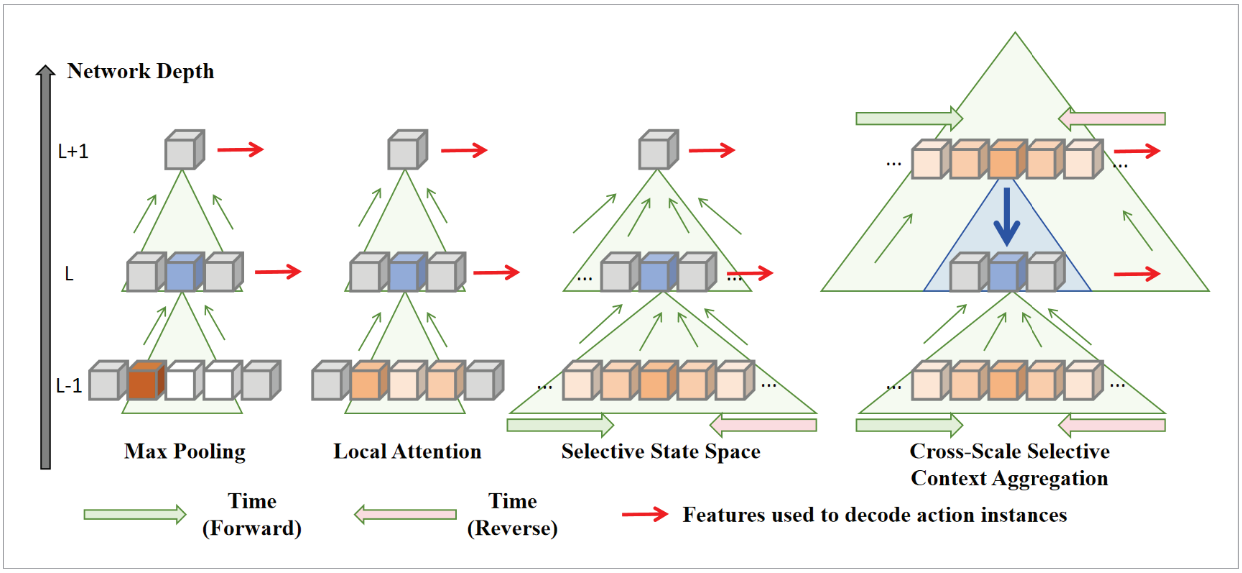


Figure 2

The difference of context modeling between our method and other TAD methods. The blue squares represent the features we are currently focusing on, while the orange squares indicate the sources of information for the blue squares. In "Max Pooling," the colors are only white and dark brown, which means that the information only includes the most salient part of the features. In "Local Attention," it is different; the information being passed is a weighted sum of features within a limited time interval from the lower layer. In "Selective State Space," the information being passed includes all possible information from the past and future, which has been causally modeled. In our method, we extend this modeling process to the upper-level features.



real-world scenarios of power grid operations. All 2011 videos are meticulously annotated. Tests of our model on this dataset demonstrate its capability to effectively address the issue of insufficient detection performance for short actions in real scenarios.

Our model has also undergone extensive testing on the public datasets THUMOS14 [14] and ActivityNet v1.3 [12], where it outperformed previous state-of-the-art models under various experimental settings.

2. Related Work

2.1 Temporal Action Detection

Temporal Action Detection (TAD) methods, as outlined in [33], are typically categorized into two-stage and single-stage approaches. Two-stage methods, such as those described in [2][18][41], initially generate candidate action proposals and subsequently refine these proposals through boundary regression and action classification. For instance, BMN [18] introduces a boundary-matching mechanism to produce flexible and variable-length proposals with reasonable confidence scores. TCA-Net [3] enhances proposal quality by aggregating "local and global" temporal contexts and refining boundaries progressively. ContextLoc [49] models temporal features from local, global, and inter-proposal perspectives, while VSGN [46] leverages graph neural networks to capture cross-scale relationships by training videos and their rescaled counterparts. Proposals can be generated using fixed anchors of single or multiple sizes [39], [40] or through direct boundary regression [37], [27]. Despite their elaborate fusion mechanisms, two-stage frameworks often suffer from complex designs and error propagation in proposal generation. In contrast, single-stage methods [2], [9] directly predict temporal boundaries and action categories for each instance. Innovations in this category include AFSD [16], which introduces boundary pooling to highlight salient features, and TadTR [23] and TallFormer [5], which incorporate transformers for improved performance. Anchor-free models like ActionFormer [44] and TriDet [29] have demonstrated surprising performance in TAD tasks.

Notably, ActionMamba [4] replaces the transformer block in ActionFormer [44] with a decomposed bidirectional Mamba block, achieving high mAP scores on TAD benchmark datasets.

2.2 Feature Aggregation in Image Task

Feature aggregation involves consolidating feature maps produced by the backbone network into a unified feature vector that represents a target. Basic techniques include max pooling and mean pooling. R-MAC [31] extracts feature vectors by focusing on regions with maximum activation, identifying the most salient areas within an image. FPN [17] constructs a top-down architecture with lateral connections to capture high level semantic features across all scales. PANet [20] enhances this approach by incorporating a bottom-up path aggregation mechanism, facilitating the transmission of low-level information to higher levels. NASFPN [10] employs neural architecture search to identify an efficient FPN structure. FPT [45] integrates transformers [32] to aggregate information across different layers and transform features within the current layer. CARAFE [34] serves as a lightweight and efficient upsampling operator capable of aggregating information from large receptive fields. DRFPN [25] uses attention mechanisms to adaptively merge features both channel-wise and level-wise. SCPNet [7] proposes a method for aggregating local features through feature splicing and constructing a corresponding loss function for learning. AFPN [43] introduces an asymptotic feature aggregation approach for image segmentation. Despite these advancements in image tasks, few studies have explored the application of feature aggregation techniques in TAD tasks. Therefore, our research proposes a novel method that combines feature aggregation with TAD techniques.

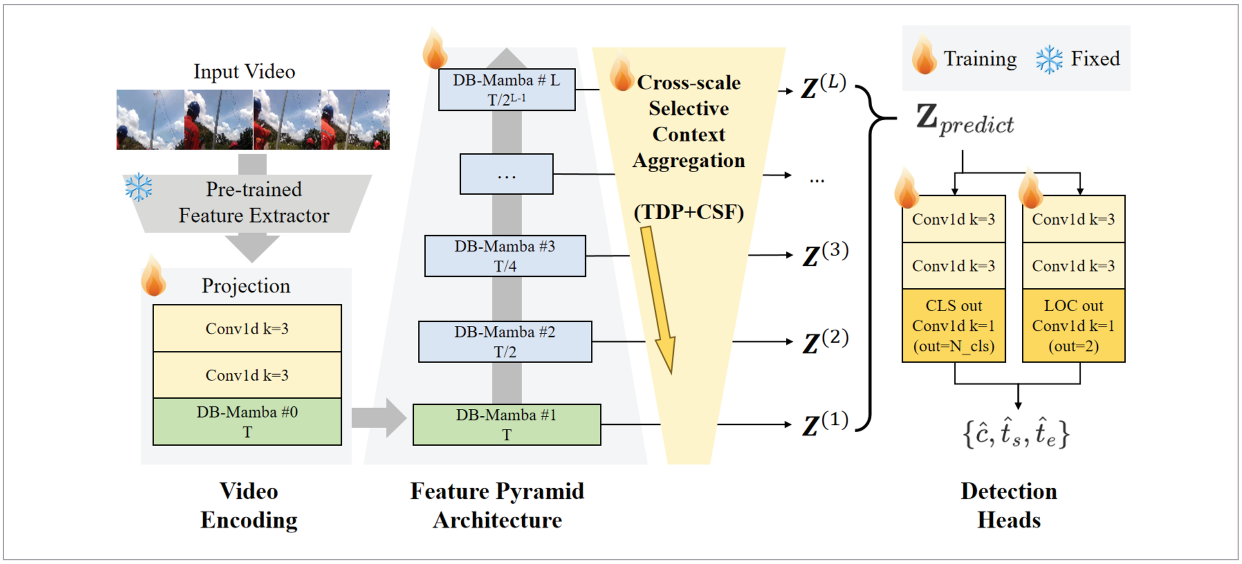
3. Model

3.1 Preliminary

When provided with an untrimmed video, the primary objective of TAD is to predict a set of action instances. Each action can be denoted as $\Psi = \{c, t_s, t_e\}$, denoting the action category, start time, and end time, respectively.

Figure 3

The pipeline of CSCAN. First, video frames are extracted by a pre-trained model to obtain raw features. Then, after conversion by the feature projection module, the features are transformed into a low-dimensional subspace suitable for TAD tasks. Subsequently, after a feature pyramid structure, we obtain a group of feature maps at different scales. After applying our proposed structures, Top-Down Pathway (TDP) and Cross-scale Selective Fusion (CSF) to these feature maps, each layer of features will integrate information from other layers. By collecting all the features and feeding them into the classification and localization modules, we obtain the final predictions.



Training end-to-end with video data incurs computational costs that are thousands of times higher than those of corresponding image tasks. Therefore, for computational efficiency and fair comparison with other baselines [44], [4], [29], [15], we use I3D [1] and InternVideo [38] features pre-extracted by other researchers for public datasets THUMOS14 [14] and ActivityNet v1.3 [12]. For our collected dataset POWER, we use VideoMAEv2-b [35] for feature extraction. The extracted feature corresponding to each video is $\mathbf{x} \in \mathbf{R}^{C_E \times T}$, where C_E denotes the number of output channels of video encoder and T represents the temporal dimension and is proportional to the video length.

Following ActionMamba [4], we construct a basic feature pyramid consisting of L layers of DB-Mamba block [4], where pooling or other downsampling methods are applied. DB-Mamba is an advanced module designed to enhance the performance and efficiency of data processing and feature extraction, combining the strengths of bidirectional processing with a unique dual-layer architecture to capture both short-term and long-term dependencies effectively.

3.2 Overview

The entire pipeline is briefly illustrated in Figure 3 and its caption.

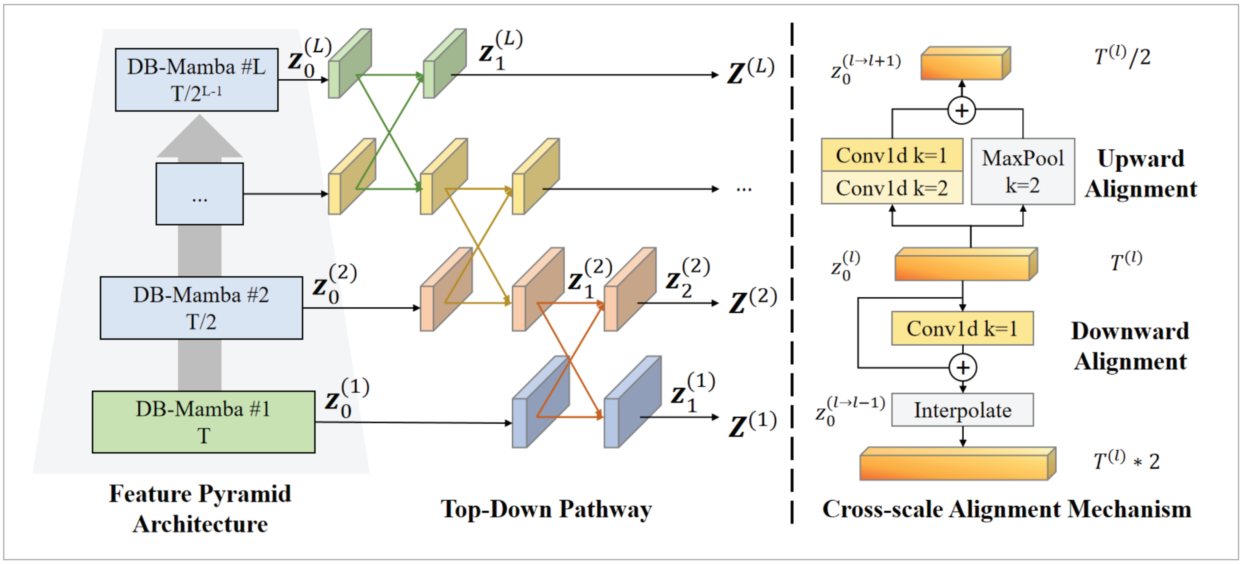
3.3 Top-Down Path

To address multi-scale action challenges, various solutions have been proposed. Models like ActionFormer [44], TemporalMaxer [30], and TriDet [29] use feature pyramids with layerwise downsampling to expand high-level feature receptive fields. Yet, this design only aggregates context from small to large scales, and the contextual information of the lowerlevel features is limited. To address this issue, we propose a feature conduction structure named Top-Down Pathway (TDP) which leverages coarse-grained features' rich contextual information to complement fine-grained features, as shown in Figure 4. Unlike methods in object detection [17], [20], our approach to contextual fusion involves bidirectional information exchange each time and we design the new alignment mechanism.

The input for the TDP is the output of the feature pyramid mentioned above, *i.e.*, $\mathbf{Z}_{\text{fpn}}$. We will gradually incorporate the output information from each upper layer into the input of the next lower layer. Initially, there are two primary challenges: the first is how to align features of different sizes, and the second is how to fuse these aligned features. This subsection tackles the first challenge, and next subsection address the second challenge. Alignment

Figure 4

The structure of TDP and implementation of the alignment mechanism. First, the outputs of all layers are fused from top to bottom until the lowest layer. Each fusion occurs between adjacent layers. The fusion process is completed by the CSF module shown in Figure 5.



can be categorized into upward alignment and downward alignment.

Upward alignment involves reducing the size of the feature map. We achieve this by using MaxPooling to capture the main information from the lower layer, combined with residual information obtained from a stacked two-layer convolutional network. Suppose we have feature $\mathbf{z}_0^{(l)}$ from layer l with length $T^{(l)}$, and we need to align the feature map $\mathbf{z}_0^{(l)}$ with its counterpart from higher layer with length $T^{(l)}/2$. The subscript denotes the $\mathbf{z}_0^{(l+1)}$ number of times of fusion. The feature transformation can be described as follows:

$$\mathbf{z}_0^{(l\rightarrow l+1)} = \text{MAXPooling}(\mathbf{z}_0^{(l)}) + \varphi_2(\varphi_1(\mathbf{z}_0^{(l)})), \quad (1)$$

where the kernel size of MaxPooling(·) is 3, with a stride of 2. $\varphi_1(\cdot)$ denotes using depth-wise convolution to conduct channel transformation, and $\varphi_2(\cdot)$ represents the use of a kernel size of 3 and a step size of 2 for 1d-convolution to perform temporal transformation.

Aligning down means to enlarge the feature map, which requires aligning from the feature map $\mathbf{z}_0^{(l)}$ to $\mathbf{z}_0^{(l-1)}$. Also, use a layer of depth-wise Conv, $\varphi_1(\cdot)$, to obtain the residufl transformation, add it to $\mathbf{z}_0^{(l)}$, and then use nearest neighbor interpolation to resize the feature map:

$$\mathbf{z}_0^{(l\rightarrow l-1)} = \text{Interpolate}(\mathbf{z}_0^{(l)} + \phi(\mathbf{z}_0^{(l-1)})), \quad (2)$$

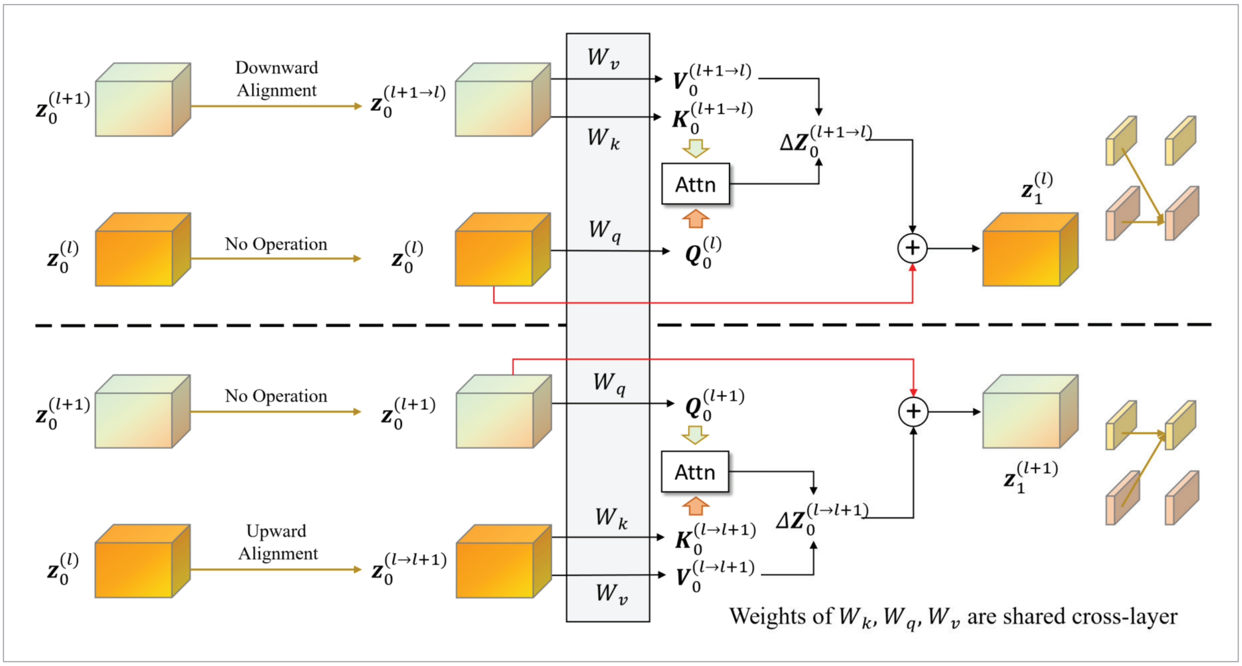
where the source and target temporal size of the interpolation operation are $T^{(l)}$, $T^{(l-1)}$, respectively.

Subsequently, as shown in Figure 4, the contextually aggregated features obtained from the $\mathbf{z}_0^{(l)}$ and $\mathbf{z}_0^{(l-1)}$are denoted as $\mathbf{z}_1^{(l)}$ and $\mathbf{z}_1^{(l-1)}$, where the subscripts represent the number of times the feature has been aggregated. Then, the aggregation of $\mathbf{z}_1^{(l-1)}$ and $\mathbf{z}_0^{(l-2)}$ results in $\mathbf{z}_2^{(l-1)}$ and $\mathbf{z}_1^{(l-2)}$. The concrete implementation of cross-scale features is introduced in the next subsection.

The primary innovation in our TDP lies in the feature fusion process, where we have introduced a cross-layer attention mechanism which is introduced in the following subsection. This mechanism allows the network to dynamically weigh the importance of features from different layers, thereby enhancing the representation capability and adapt-

Figure 5

Illustration of CSF. CSF involves aggregating and fusing features from different scales, utilizing an cross attention mechanism with a residual structure to effectively integrate information from various levels while preserving the original information.


$z_0^{(l+1)}$
Downward Alignment
$z_0^{(l+1\rightarrow l)}$
W_v
W_k
$V_0^{(l+1\rightarrow l)}$
$K_0^{(l+1\rightarrow l)}$
$\Delta Z_0^{(l+1\rightarrow l)}$
Attn
$z_1^{(l)}$
$z_0^{(l)}$
No Operation
$z_0^{(l)}$
W_q
$Q_0^{(l)}$
$z_0^{(l+1)}$
No Operation
$z_0^{(l+1)}$
W_q
$Q_0^{(l+1)}$
Attn
$z_1^{(l+1)}$
W_k
$\Delta Z_0^{(l\rightarrow l+1)}$
$z_0^{(l)}$
Upward Alignment
$z_0^{(l\rightarrow l+1)}$
$K_0^{(l\rightarrow l+1)}$
$V_0^{(l\rightarrow l+1)}$
W_v
Weights of W_k, W_q, W_v are shared cross-layer

ability of the model. By incorporating cross-layer attention, TDP can better capture both local and global contextual information, leading to improved performance in tasks such as object detection and segmentation.

3.4 Cross-Scale Selective Fusion

For feature fusion, simple methods like stitching or pointwise addition have drawbacks: stitching increases computational cost greatly, while pointwise addition introduces noise. ASFF [39] proposed adaptive fusion method to reduce channels to 8-16 to express hierarchical importance, but still retains noise. We propose that a residual-based attention structure is better suited, as residuals preserve original features and make model easy to converge, while attention module selectively filters effective information. Thus, we designed the Cross-scale Selective Fusion (CSF) for multi-scale context aggregation.

The input of CSF is a couple of scale-aligned features $\{\mathbf{z}_0^{(l+1)}, \mathbf{z}_0^{(l)}\}$ from adjacent levels, for example, where the subscript denotes the number of fusion experiences. See the example in Figure 5, to obtain $\mathbf{z}_1^{(l+1)}$ and $\mathbf{z}_1^{(l)}$, we first need to convert $\mathbf{z}_0^{(l+1)}$ and $\mathbf{z}_0^{(l)}$ into queries:

$$\mathbf{Q}_0^{(l)} = \boldsymbol{W}_q \mathbf{z}_0^{(l)}, \mathbf{Q}_0^{(l+1)} = \boldsymbol{W}_q \mathbf{z}_0^{(l+1)}. \tag{3}$$

Then, we need to obtain $\mathbf{z}_0^{(l\to l+1)}$ and $\mathbf{z}_0^{(l+1\to l)}$ using Equations (1)-(2), respectively to get their corresponding keys and values:

$$\begin{aligned} \mathbf{K}_0^{(l\to l+1)} &= \boldsymbol{W}_k \mathbf{z}_0^{(l\to l+1)}, \mathbf{V}_0^{(l\to l+1)} = \boldsymbol{W}_v \mathbf{z}_0^{(l\to l+1)}, \\ \mathbf{K}_0^{(l+1\to l)} &= \boldsymbol{W}_k \mathbf{z}_0^{(l+1\to l)}, \mathbf{V}_0^{(l+1\to l)} = \boldsymbol{W}_v \mathbf{z}_0^{(l+1\to l)}, \end{aligned} \tag{4}$$

where $\boldsymbol{W}_q$, $\boldsymbol{W}_k$, $\boldsymbol{W}_v$ are trainable weights shared across layers. Then, we use these tensors to calculate the attention to further obtain residual information which is denoted as $\Delta\mathbf{z}$:

$$\begin{aligned} \Delta\, \mathbf{z}_0^{(l\to l+1)} &= \mathrm{softmax}(\mathbf{Q}_0^{(l)} (\mathbf{K}_0^{(l\to l+1)})^T / \sqrt{d})\mathbf{V}_0^{(l\to l+1)}, \\ \Delta\, \mathbf{z}_0^{(l+1\to l)} &= \mathrm{softmax}(\mathbf{Q}_0^{(l)} (\mathbf{K}_0^{(l+1\to l)})^T / \sqrt{d})\mathbf{V}_0^{(l+1\to l)}, \end{aligned} \tag{5}$$

where d is the number of channels of these attention components. We add these outputs to $\mathbf{z}_0^{(l)}$ and $\mathbf{z}_0^{(l+1)}$ to get $\mathbf{z}_1^{(l)}$ and $\mathbf{z}_{1+1}^{(l)}$ that integrates information from another layers:

$$\mathbf{z}_1^{(l)} = \mathbf{z}_0^{(l)} + \Delta\, \mathbf{z}_0^{(l+1\to l)}, \mathbf{z}_1^{(l+1)} = \mathbf{z}_0^{(l+1)} + \Delta\, \mathbf{z}_0^{(l\to l+1)}. \tag{6}$$

By repeating this operation from the top layer down to the bottom layer, we ultimately obtain a set of features $\{\mathbf{z}_1^{(1)}, \mathbf{z}_2^{(2)}, \mathbf{z}_2^{(3)}, \ldots, \mathbf{z}_2^{(L-1)}, \mathbf{z}_1^{(L)}\}$, using capital letter Z to abbreviate it as $\mathbf{z}_{\mathrm{fpn}} = \{\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \mathbf{Z}^{(3)}, \ldots, \mathbf{Z}^{(L)}\}$.

3.5 Detection Heads and Postprocess

The detection head consists of a classification head and a localization head. The first part of the classification head and localization head is composed of stacked one-dimensional convolutions, where we denote the operation of one convolution layer as $f(\cdot)$:

$$f(\mathbf{z}) = \mathrm{ReLU}(\mathrm{LayerNorm}(\mathrm{dwConv}(\mathbf{z}))), \tag{7}$$

where the input of the detection head is $\mathbf{z}_{\mathrm{fpn}}$ with time dimensions of $\{T, T/2, \ldots, T/2^{L-1}\}$, respectively. By concatenating them in time series, we obtain $\mathbf{z}_{\mathrm{predict}} \in \mathrm{R}^{C\times T_{sum}}$, where $T_{sum} = \sum_{l=0}^{L} T/2^l$ and C is the channel number. According to the settings of the feature extraction process, after simple calculations, we can obtain the time scale of the feature at each position in the video $\mathbf{P} \in \mathrm{R}^{1\times T_{sum}}$ and the size of the receptive field corresponding to the current position $\mathbf{S} \in \mathrm{R}^{2\times T_{sum}}$.

For classification, we use two layers of 1d-conv to obtain the classification score:

$$\hat{c} = \mathrm{dwConv}_{\mathrm{C}\to\mathrm{K}}(\mathrm{f}_{c2}(\mathrm{f}_{c1}(\mathbf{z}_{\mathrm{predict}}))), \tag{8}$$

where $\hat{c} = \mathrm{R}^{K\times T_{sum}}$, and K is the number of action classes to be classified.

For localization, we use a similar operation, but ultimately obtain the relative position within the segment with a dimension of 2:

$$\hat{r} = \mathrm{dwConv}_{\mathrm{C}\to 2}(\mathrm{f}_{r2}(\mathrm{f}_{r1}(\mathbf{z}_{\mathrm{predict}}))), \tag{9}$$

where $\hat{r} = \mathrm{R}^{K\times T_{sum}}$. The first row of elements represents the relative start position of the action within the segment, and the second row of elements represents the relative end position of the action within the segment. Organize $\hat{r}$, $\mathbf{P}$ and $\mathbf{S}$, we can obtain T_{sum} absolute time coordinates $\hat{b} = \mathbf{P} + \mathbf{S}\hat{r}$. These boundary predictions with their classification scores are postprocessed using SoftNMS to obtain the final N prediction results.

3.6 Scale Consistency Training

In order to better capture the scale-invariant features in actions, we designed a temporal scaling augmentation strategy based on Gaussian sampling to assist our training. Specifically, we first perform random Gaussian sampling to obtain scaling factors around 0. Subsequently, we map these factors to an exponential space to obtain scaling ratios. This process ensures that the scaling ratios are symmetrically distributed in the logarithmic space and within a reasonable range. By doing so, controlled yet diverse scale perturbations are introduced to the input data:

$$\gamma = 2^{\xi}, \xi \sim N(0, \delta), \tag{10}$$

where γ denotes the scaling ratio, ξ refers to a random number that follows a normal distribution and δ is a parameter set to 1 by default.

Before projection, we scale the temporal dimension of $\mathbf{x} \in \mathrm{R}^{C_E \times T}$ with probability p, which ensures that we can learn the information of the original samples.

$$\mathbf{x}' = \mathrm{Interpolate}_{T \to \gamma T}(\mathbf{x}). \tag{11}$$

At the same time, we also perform time-series scaling on the training labels, and the corresponding labels are changed from $\{c, t_s, t_e\}$ to $\{c, \gamma t_s, \gamma t_e\}$. Through time-series scaling, the time-series dimensions of all subsequent features will change.

Our overall loss consists of classification loss, localization loss, and scale loss. The classification loss L_{cls} uses focal loss [40], the localization loss L_{reg} uses dioU [41], and the design of the single prediction for the scale loss is as follows:

$$L_{scale,i} = \frac{1}{2}(\log(\widehat{t}_{e,i} - \widehat{t}_{s,i}) - \log(t_{e,i} - t_{s,i})), \tag{12}$$

where N denotes the number of predictions. Finally, the overall training loss is calculated by:

$$L = \lambda_{cls} L_{cls} + \lambda_{reg} L_{reg} + L_{scale}, \tag{13}$$

among which λ_{cls} and λ_{reg} are the hyper-parameters, both of them are set to 1 by default.

4. Experiments

4.1 Datasets

In order to assess the efficacy of our method, akin to the majority of prior studies, we carried out comprehensive experiments on two prominent TAD benchmarks: THUMOS14 and ActivityNet-1.3. In addition, we conducted extensive experiments on our self-collected dataset, POWER, for validation. We employed the mean average precision (mAP) metric, calculated across various temporal intersection over union (tIoU) thresholds, as it is a universally accepted standard for evaluating the performance of TAD models, thereby serving as the cornerstone for evaluating our proposed approach.

THUMOS14 contains 200 untrimmed videos labeled for training and 213 for testing, spanning 20 sports categories. The actions within these videos are notably shorter and more densely concentrated in terms of their quantity per video, presenting a formidable obstacle for TAD. With an average video duration of 4.4 minutes and action instances averaging just 5 seconds, THUMOS14 also features background segments occupying an average of 71% of each video's length. To evaluate our method's performance on this dataset, consistent with the majority of baseline models, we utilize mAP at tIoU thresholds ranging from 0.3 to 0.7 with an interval of 0.1, along with the overall average mAP.

ActivityNet v1.3 comprises 10,024 untrimmed videos from 200 daily activity categories designated for training and 4,926 videos for testing. In comparison to THUMOS14, the action instances featured in ActivityNet-1.3 are notably longer and distributed more sparsely within each video in terms of quantity. The average duration of each video is 2 minutes, while the average length of an action instance stands at 48 seconds. On average, each video encompasses 1.41 activities. In accordance with established conventions, mAP@[0.50:0.05:0.95] is employed as the evaluation metric for ActivityNet-1.3, and we report both the overall average mAP as well as mAPs specifically at the thresholds of 0.5, 0.75, and 0.95.

EPIC-KITCHENS 100 is an extension of the original EPIC-KITCHENS dataset. It contains over 3,500 hours of first-person video footage captured in natural kitchen settings, involving a wide range of everyday activities such as cooking, cleaning, and food preparation. The dataset is unique in its scale and the richness of the annotations provided.

POWER dataset mainly involves workers performing tasks on utility poles during power outages. The dataset contains 2,011 videos captured under various lighting and filming conditions, documenting power outage operations. It includes five types of actions: climbing, voltage testing, grounding wire installation, grounding wire removal, and descending from the pole, totaling 4,696 action instances. The average length of each instance is 66 seconds, and the total duration of all videos is 430 hours. The metric of POWER is the same as that of THUMOS14. The specific information of this dataset is presented in Table 1.

4.2 Implementation Details

For THUMOS14 [14] and ActivityNet v1.3 [12], consistent with most TAD models [44], [22], we use In-

Table 1

Detailed information about the POWER dataset.

Item	Training	Test	All	
Videos	1694	311	2011	
Instances	3863	833	4696	
Duration	359.63	70.84	430.48	
Action	**Training**	**Test**	**All**	**Duration**
ClimbUp	1360	344	1704	67.3
Verification	360	94	454	27.9
Put Grounding Rod	451	111	562	116
Descending	1362	223	1585	52.8
Take Grounding Rod	330	61	391	83.2
All	3863	833	4696	66

Table 2

Performance comparison with other TAD methods on THUMOS14 and ActivityNet v1.3.

Method	Feature	THUMOS14 [14]						ActivityNet v1.3 [12]			
		0.3	0.4	0.5	0.6	0.7	Avg	0.5	0.75	0.95	Avg
BMN [18]	TSN [36]	56.0	47.4	38.8	29.7	20.5	38.5	50.1	34.8	8.3	33.9
G-TAD [42]	TSN [36]	56.0	47.4	38.8	29.7	20.5	38.5	50.1	34.8	8.3	33.9
TCA-Net [3]	TSN [36]	60.6	53.2	44.6	36.8	26.7	44.3	52.3	36.7	6.9	35.5
VSGN [46]	TSN [36]	66.7	60.4	52.4	41.0	30.4	50.2	52.4	36.0	8.4	35.1
ContextLoc [49]	I3D [1]	68.3	63.8	54.3	41.8	26.2	50.9	56.0	35.2	3.6	34.2
RCL [37]	I3D [1]	70.1	62.3	52.9	42.7	30.7	51.0	51.7	35.3	8.0	34.4
AFSD [16]	I3D [1]	67.3	62.4	55.5	43.7	31.1	52.0	52.4	35.3	6.5	34.4
TAGS [26]	I3D [1]	68.6	63.8	57.0	46.3	31.8	52.8	56.3	36.8	9.6	36.5
MUSES [24]	I3D [1]	68.9	64.0	56.9	46.3	31.0	53.4	50.0	35.0	6.6	34.0
TALLFormer [5]	I3D [1]	68.4	-	57.6	-	30.8	53.9	41.3	27.3	6.3	27.2
TadTR [23]	I3D [1]	74.8	69.1	60.1	46.6	32.8	56.7	52.8	37.1	10.8	36.1
ActionFormer [44]	I3D [1]	82.1	77.8	71.0	59.4	43.9	66.8	53.5	36.2	8.2	35.6
ASL [28]	I3D [1]	83.1	79.0	71.7	59.7	45.8	67.9	54.1	37.4	8.0	36.2
TriDet [29]	I3D [1]	83.6	80.1	72.9	62.4	47.4	69.3	-	-	-	-
CSCAN(Ours)	**I3D [1]**	**84.3**	**80.7**	**73.2**	**62.5**	**47.2**	**69.5**	**56.0**	**38.2**	**8.3**	**36.7**
ActionFormer [44]	InternVideo-6B [38]	82.3	81.9	75.1	65.8	50.3	71.9	61.5	44.6	12.7	41.2
ActionMamba [4]	InternVideo-6B [38]	86.9	83.1	76.9	65.9	50.8	72.7	62.4	43.5	10.2	42.0
CSCAN(Ours)	**InternVideo-6B [38]**	**88.3**	**84.4**	**78.6**	**67.0**	**51.8**	**74.0**	**63.9**	**45.2**	**12.8**	**42.9**

Table 3
Performance comparison with other TAD methods on EPIC-KITCHEN.

Dataset	Method	Feature	POWER					
			0.1	0.2	0.3	0.4	0.5	Avg.
EPIC-KITCHEN Verb	BMN[18]	SlowFast [8]	10.8	8.8	8.4	7.1	5.6	8.4
	ActionFormer [44]	SlowFast [8]	26.6	25.4	24.2	22.3	19.1	23.5
	Tridet[29]	SlowFast [8]	28.6	27.4	26.1	24.2	20.8	25.4
	CSCAN(Ours)	**SlowFast [8]**	**29.9**	**28.6**	**26.2**	**25.6**	**21.4**	**26.3**
EPIC-KITCHEN Noun	BMN[18]	SlowFast [8]	10.3	8.3	6.2	4.5	3.4	6.5
	ActionFormer [44]	SlowFast [8]	25.2	24.1	22.7	20.5	17.0	21.9
	Tridet[29]	SlowFast [8]	27.4	26.3	24.6	22.2	18.3	23.8
	CSCAN(Ours)	**SlowFast [8]**	**28.3**	**27.1**	**25.9**	**23.0**	**19.1**	**24.7**

Table 4
Performance comparison with other TAD methods on POWER.

Method	Feature	POWER					
		0.3	0.4	0.5	0.6	0.7	Avg.
TemporalMaxer [30]	VideoMAEv2-b [35]	68.2	64.2	54.7	41.3	26.2	50.9
TriDet [29]	VideoMAEv2-b [35]	70.9	66.2	58.7	46.7	32.2	54.9
ActionFormer [44]	VideoMAEv2-b [35]	71.6	66.7	60.4	48.5	31.6	55.7
ActionMamba [4]	VideoMAEv2-b [35]	75.8	71.0	62.7	50.8	34.1	58.9
CSCAN(Ours)	**VideoMAEv2-b [35]**	**78.2**	**73.7**	**65.9**	**53.2**	**36.5**	**61.6**

ternVideo [38] and I3D pre-trained on Kinetics [1] as inputs. For THUMOS14, 16-frame clips at 30 fps with a stride of 4 are used, and feature sequences are standardized to 2048 via cropping or padding. For ActivityNet-1.3, videos are resampled to 30 fps, split into 16-frame clips with a stride of 16, and sequences are set to 256. All frames are center-cropped and resized to 224×224. For dataset POWER, we apply VideoMAEv2-b [35] as video encoder, using clip size of 16 and clip stride of 4. For EPIC-KITCHEN, we adopt SlowFast as backbone feature.

For THUMOS14, the model was trained for 35 epochs with a learning rate of 0.0001, including a 5-epoch warm-up and 30epoch cosine annealing. A mini-batch size of 2 and AdamW optimizer (weight decay 0.01) were used. For ActivityNet v1.3, training lasted 20 epochs with a learning rate of 1e-3, a 5-epoch warm-up, and 15-epoch cosine annealing, with a mini-batch size of 16. Other settings matched THUMOS14. A 6-layer pyramid structure was employed, with each layer's length scaled by 2 and feature dimension fixed at 512. Feature temporal scaling was applied with p = 0.5. For POWER, training lasted 35 epochs with a learning rate of 0.0002, The other settings are the same as those used for training on THUMOS14. For EPIC-KITCHEN, learning rate is set to 0.0002 and training epoch is set to 23 and 19 for "verb" and "noun", respectively.

4.3 Comparison to Other TAD Methods

In the context of THUMOS14, as demonstrated in Table 2 (left), by leveraging features derived from I3D, our approach achieves an exceptional average mAP of 69.5%, marking a +2.7% improvement over the previously most frequently used baseline, ActionFormer [44], which employs the same features. Furthermore, it surpasses TriDet [29], which utilizes well-designed Triend heads for post-processing, by a slight margin of 0.2% mAP. Specifically, at a temporal intersection over union (tIoU) of 0.5, our method attains an mAP of 73.2%, surpassing previous state-of-the-art meth ods such as ActionFormer by +2.2% and TriDet by +0.3%. When utilizing InternVideo features, our proposed CSCAN on THUMOS14 performs significantly better than with I3D features, owing to its superior temporal-spatial modeling capability gained from masked auto-encoding training. CSCAN achieves a 74.0% mAP, which outperforms the last two methods that have used InternVideo features in all evaluation metrics.

Regarding ActivityNet v1.3, as shown in Table 2(right), our method also exhibits strong performance on the larger dataset, both for widely used I3D features and newly used InternVideo features. This further corroborates the effectiveness and superiority of our method. Our method achieves the best performance using InternVideo features, consistent with the results on THUMOS14. Specifically, with InternVideo features, our method achieves an average mAP of 42.9%, which is 0.9% higher than the state-of-the-art method, ActionMamba [4].

In our evaluation on the EPIC-KITCHEN dataset, we primarily compare our approach with TriDet [29], which has previously shown strong performance. The results are summarized in Table 3. Our method achieves substantial enhancements in both subtasks, namely verb and noun recognition, attaining average mAP scores of 26.3% and 24.7%, respectively.

On the POWER dataset, CSCAN demonstrates a significant advantage over other TAD models, as shown in Table 4. This advantage is more pronounced compared to its performance on THUMOS14 and ActivityNet v1.3.

In summary, the model CSCAN demonstrates improved positioning accuracy due to multi-scale feature fusion. Addi tional ablation experiments have confirmed the effectiveness of each module.

5. Ablation Study and Discussion

For brevity, we will refer to THUMOS14 as THUMOS and ActivityNet v1.3 as ANET in the following text. We use I3D features for THUMOS and ANET and use VideoMAEv2-b features for POWER.

We design the ablation experiments to test the effectiveness of each module in our proposed CSCAN on THUMOS, ANET and POWER. The ablation experiments for each module are presented in Table 5. It can be seen that the addition of the TDP and CSF modules can bring an average mAP improvement of 2.7/0.9/2.2% for three datasets, respectively. Further adding scale consistency loss L_{scale}, the contribution of all modules of our CSCAN can reach a comprehensive 3.0/1.3/2.9% mAP. For specific experiments of each module, please refer to the following context.

Table 5

Ablation study of components of CSCAN.

#	TDP	CSF	L_{scale}	THUMOS	ANET	POWER
1				66.5	35.4	58.9
2	√	√		69.2	36.3	61.1
3			√	67.4	35.8	60.1
4	√	√	√	69.5	36.7	61.6

5.1 Impact of Feature Conduction Path

Our TDP merges adjacent-level feature maps top-down. Table 6 evaluates merge strategies for CSCAN: (1) no merge, (2) FPN-style aggregation, and (3) layer-wise dense connection like AFPN [9]. On THUMOS, FPN-style improves average mAP by 2.4% over no merge (ID 2 vs. ID 1), while AFPNstyle improves it by 1.5% (ID 3 vs. ID 1). On ANET, FPNstyle and AFPN-style boost average mAP by 0.7% and 0.5%, respectively, compared to no merge. TDP, combining both strategies, enhances average mAP by 2.7% and 0.9% on the two datasets. On the POWER dataset, the design of TDP is reliable as well.

Table 6

Ablation study on effectiveness of merging path

#	Config	THUMOS	ANET	POWER
1		66.5	35.4	58.9
2	FPN [17]	68.9	36.1	59.9
3	AFPN [43]	68.0	35.9	60.1
4	TDP	69.2	36.3	61.1

5.2 Impact of Cross-Scale Selective Fusion

Our CSF module fuses multi-layer information into a single layer. We compare it with three baselines:

average weight, adaptive weight [21], and self-attention (replacing crosslayer queries with self-queries). Average weight introduces unnecessary noise, while self-attention fails to identify useful features from other layers. Adaptive weight [21] in this context refers to the use of a linear layer's output to compute the weights between different layers dynamically. As shown in Table 7, CSF uses current-layer features as queries to filter useful features from other layers. On Three datasets, CSF outperforms average weight by 2.6/1.3/5.2% mAP, adaptive weight by 1.0/0.6/3.5% mAP, and self-attention by 1.2/0.6/3.3% mAP, respectively.

Table 7

Ablation study on effectiveness of fusion module.

#	Config	THUMOS	ANET	POWER
1	Average Weight	66.9	35.4	56.4
2	Adaptive Weight [21]	68.5	36.1	58.1
3	Self-Attention	68.3	36.1	58.3
4	**CSF**	**69.5**	**36.7**	**61.6**

5.3 Ablation Study of Scaling Strategy in Scale Consistency Training

For the scale consistency training, we tried two scaling strategies. One is linear interpolation, and the other is nearest interpolation. We found that the performance of the two is similar, and the experimental results are shown in Table 8. We tried different σ and p under a fixed random seed, and found that performance is best when σ is 1 and p equal to 0.5, but this pair is not significantly better than other pairs. However, it is certain that the effect of using temporal scaling is significantly better than that of not using temporal scaling (others v.s. ID 1).

5.4 Discussion on Training Convergence

In Figure 6, we visualize the curve of training losses of three models (CSCAN, ActionFormer, and ActionMamba). It is observed that at the beginning of training, due to the increased overall complexity introduced by our module, the overall loss is relatively high. However, as the experiment progresses, the loss of CSCAN drops rapidly and remains superior to the other two models throughout the subsequent process.

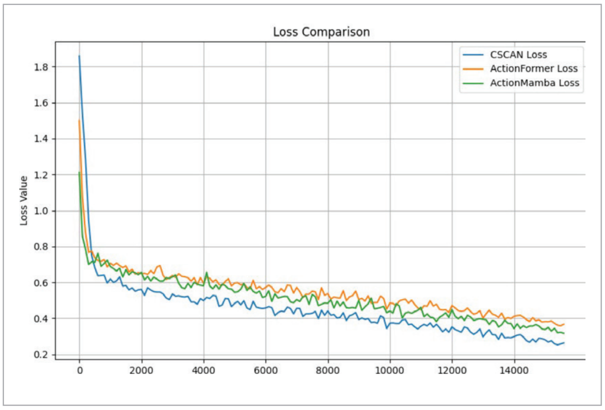
Table 8

Ablation study on effectiveness of fusion module.

#	Setting	THUMOS	ANET	POWER
1	W/OL_{scale} ($p = 0$)	69.2	36.3	61.1
2	Linear ($p = 0.5, \sigma = 1$)	69.4	36.5	61.3
3	Linear ($p = 0.5, \sigma = 2$)	69.5	36.5	61.8
4	Linear ($p = 1, \sigma = 1$)	69.6	36.6	61.5
5	Nearest ($p = 0.5, \sigma = 1$)	69.5	36.7	61.6
6	Nearest ($p = 0.5, \sigma = 2$)	69.4	36.6	61.7
7	Nearest ($p = 1, \sigma = 1$)	69.2	36.4	61.3

Figure 6

Training loss of three models on POWER.

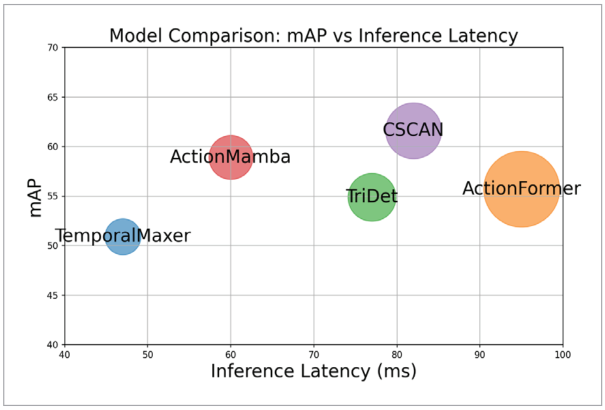


5.5 Efficiency Analysis

Although there have been advancements in network technology [48], current application scenarios still place high demands on the inference performance of models. Therefore, it is necessary to compare the efficiency differences among various models to guide future improvements. We have evaluated the performance of different models on the POWER dataset. Figure 7 illustrates the relationship between inference latency and performance. The size of the circle for each model reflects its computational cost. It can

Figure 7

Efficiency analysis of different models on POWER.



be observed that CSAN outperforms the baseline ActionFormer in all aspects.

5.6 Statistical Analysis: Improvement and Potential Limitation

We conduct a statistical analysis of the prediction results of ActionMamba [4] and CSCAN in Figure 8 and Figure 9, respectively. It is evident that shorter actions are more difficult to detect, and the recognition rate for short actions of CSCAN is significantly higher than that of ActionMamba. For actions with a duration of less than 30 seconds (size S), the proportion of False-Negative predictions decreased substantially from 76% to 56%. This improvement is reflected in the per-class action mAP shown in Table 9, where the mAP for voltage testing, which has an average duration of 28 seconds, increased from 35.55% to 46.08%.

It is important to note that among the five actions in POWER, ClimbUp and Descending are the simplest, so the advantage of CSCAN is not as pronounced for these two actions. In contrast, Verification, PutGroundingRod, and TakeGroundingRod are relatively more complex, and CSCAN shows more significant improvements, particularly for short action Verification with average duration of 27.9 seconds.

Figure 8

False-Negative evaluation results of ActionMamba [4] trained on POWER. Coverage represents the proportion of action to video duration. Length means the duration of action instance.

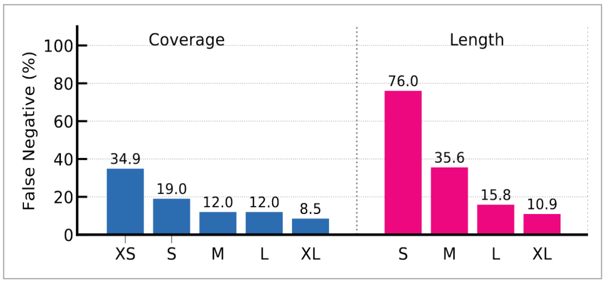


Figure 9

False-Negative evaluation results of CSCAN trained on POWER. Coverage represents the proportion of action to video duration. Length means the duration of action instance.

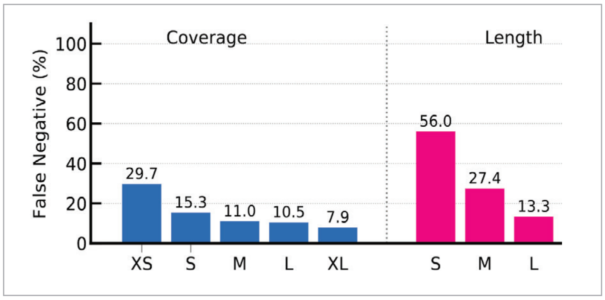
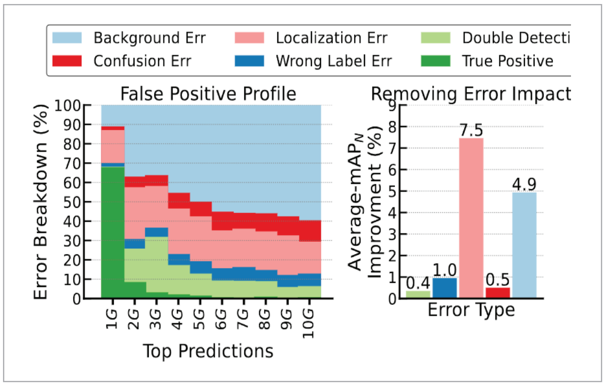


Figure10

False-Positive Analysis of CSCAN trained on POWER.



As shown in Figure 10, the primary errors in the predictions are localization errors and background error (False Positive). This highlights two key directions for future research: first, improving the recall rate of diverse action categories, where there is still a 10% improvement potential. Second, designing more reliable localization mechanisms or modules to enhance the overall localization accuracy of the model, which offers approximately an 18% improvement potential.

Table 9

Per-Class mAP on POWER. U: ClimbUp, V: Verification, P: PutGroundingRod, D: Descending, T: TakeGroundingRod.

Method	Avg.	U	V	P	D	T
TemporalMaxer [4]	50.9	64.2	27.9	46.7	72.8	42.9
TriDet [27]	54.9	68.6	32.9	56.1	73.9	43.2
ActionFormer [3]	55.8	65.2	35.2	58.4	74.7	45.3
ActionMamba [5]	58.9	67.8	39.2	63.1	75.8	48.7
CSCAN(Ours)	**61.6**	**68.5**	**46.1**	**63.7**	**75.1**	**54.4**

6. Conclusion

In this paper, we introduce a Cross-scale Selective Context Aggregation Network (CSCAN) for video temporal action detection. CSCAN models cross-layer information in fea- ture pyramid architecture. CSCAN mainly consists of three components. The first is Top-Down Pathway (TDP), an ag- gregation path at macro view, accompanied by a cross-scale feature alignment mechanism. while the second is Cross-scale Selective Fusion (CSF), a well-designed block for merge information at micro view, featured in its cross-layer query. Third is the proposed scale consistency loss for training, boosting the performance of this model on THUMOS14, ActivityNet v1.3 and POWER. We hope that CSCAN can make some contributions to the community of TAD, and help other scholars in their research.

Acknowledgement

This research was funded by Guizhou Power Grid Co. Ltd, grant number GZKJXM20222320.

Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, author-ship, and/or publication of this article.

Data Sharing Agreement

The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

References

1. Carreira, J., Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In CVPR, 2017, 6299-6308. https://doi.org/10.1109/CVPR.2017.502
2. Chao, Y. W., Vijayanarasimhan, S., Seybold, B., Ross, D. A., Deng, J., Sukthankar, R. Rethinking the Faster R-CNN Architecture for Temporal Action Localization. In 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, 1130-1139. https://doi.org/10.1109/CVPR.2018.00124
3. Chen, D., Zha, Z. J., Liu, J., Xie, H., Zhang, E. R., Yongdong, W., Cheng, W. H., Yamasaki, T., Wang, M., Ngo, C. W. Temporal-Contextual Attention Network for Video-Based Person Re-Identification. In Advances in Multimedia Information Processing-PCM 2018. Cham: Springer International Publishing, 2018, 146-157. https://doi.org/10.1007/978-3-030-00776-8_14
4. Chen, G., Huang, Y., Xu, J., Pei, B., Chen, Z., Li, Z., Wang, J., Li, K., Lu, T., Wang, L. Video Mamba Suite: State Space Model as a Versatile Alternative for Video Understanding. arXiv preprint arXiv:2403.09626.
5. Cheng, F., Bertasius, G. TallFormer: Temporal Action Localization with a Long-Memory Transformer. In ECCV, 2022, 503-521. https://doi.org/10.1007/978-3-031-19830-4_29
6. Dao, T., Gu, A. Transformers Are SSMs: Generalized Models and Efficient Algorithms Through Structured State Space Duality. 2024. [Online]. Available: https://arxiv.org/abs/2405.21060
7. Ding, Z., Wang, A., Chen, H., Zhang, Q., Liu, P., Bao, Y., Yan, W., Han, J. Exploring Structured Semantic Prior for Multi Label Recognition with Incomplete Labels. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 3398-3407. https://doi.org/10.1109/CVPR52729.2023.00331
8. Feichtenhofer, C., Fan, H., Malik, J., He, K. SlowFast Networks for Video Recognition. In Feichtenhofer, C., Haoq, F., Jitendra, M., and Kaiming, H. (Eds.), SlowFast Networks for Video Recognition. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2018, 6201-6210. https://doi.org/10.1109/ICCV.2019.00630
9. Gao, J., Yang, Z., Sun, C., Chen, K., Nevatia, R. TURN TAP: Temporal Unit Regression Network for Temporal Action Proposals. In 2017 IEEE International Conference on Computer Vision (ICCV), 2017, 3648-3656. https://doi.org/10.1109/ICCV.2017.392

10. Ghiasi, G., Lin, T. Y., Le, Q. V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. In 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, 7029-7038. https://doi.org/10.1109/CVPR.2019.00720

11. He, L., Ge, X., Hao, C., Zhang, L., Chang, S. Identification of Dress Code of Workers in Substation Based on YOLO V5. Power Systems and Big Data, 2021, 24(10), 1-8.

12. Heilbron, F. C., Escorcia, V., Ghanem, B., Niebles, J. C. ActivityNet: A Large-Scale Video Benchmark for Human Activity Understanding. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, 961-970. https://doi.org/10.1109/CVPR.2015.7298698

13. Jiang, C., Du, J., Nan, Z., Song, M. Fault Identification System for Wind Turbine Tower Systems Based on Improved YOLOv7 Algorithm. Power Systems and Big Data, 2023, 26(10), 17-25.

14. Jiang, Y., Liu, J., Zamir, A. R., Toderici, G., Laptev, I., Shah, M., Sukthankar, R. THUMOS Challenge: Action Recognition with a Large Number of Classes. 2014.

15. Jin, X., Zhang, T. MTSN: Multiscale Temporal Similarity Network for Temporal Action Localization. Proceedings of the 31st ACM International Conference on Multimedia, 2023. https://doi.org/10.1145/3581783.3612455

16. Lin, C., Xu, C., Luo, D., Wang, Y., Tai, Y., Wang, C., Li, J., Huang, F., Fu, Y. Learning Salient Boundary Feature for Anchor-Free Temporal Action Localization. In 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, 3319-3328. https://doi.org/10.1109/CVPR46437.2021.00333

17. Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S. Feature Pyramid Networks for Object Detection. In 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, 936-944. https://doi.org/10.1109/CVPR.2017.106

18. Lin, T., Liu, X., Li, X., Ding, E., Wen, S. BMN: Boundary-Matching Network for Temporal Action Proposal Generation. In 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, 3888-3897. https://doi.org/10.1109/ICCV.2019.00399

19. Lin, T. Y., Goyal, P., Girshick, R., He, K., Dollár, P. Focal Loss for Dense Object Detection. In 2017 IEEE International Conference on Computer Vision (ICCV), 2017, 2999-3007. https://doi.org/10.1109/ICCV.2017.324

20. Liu, S., Qi, L., Qin, H., Shi, J., Jia, J. Path Aggregation Network for Instance Segmentation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018. https://doi.org/10.1109/CVPR.2018.00913

21. Liu, S., Huang, D., Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. arXiv, vol. abs/1911.09516, 2019.

22. Liu, S., Zhang, C. L., Zhao, C., Ghanem, B. End-to-End Temporal Action Detection With 1B Parameters Across 1000 Frames. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2024, 18591-18601. https://doi.org/10.1109/CVPR52733.2024.01759

23. Liu, X., Wang, Q., Hu, Y., Tang, X., Zhang, S., Bai, X. End-to-End Temporal Action Detection with Transformer. IEEE Transactions on Image Processing (TIP), 2022. https://doi.org/10.1109/TIP.2022.3195321

24. Liu, X., Hu, Y., Bai, S., Ding, F., Bai, X., Torr, P. H. Multi-Shot Temporal Event Localization: A Benchmark. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021, 12596-12606. https://doi.org/10.1109/CVPR46437.2021.01241

25. Ma, J., Chen, B. Dual Refinement Feature Pyramid Networks for Object Detection. 2020. arXiv preprint arXiv:2012.01733

26. Nag, S., Zhu, X., Song, Y. Z., Xiang, T. Proposal-Free Temporal Action Detection via Global Segmentation Mask Learning. In Computer Vision - ECCV 2022, Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T. (Eds.), Cham: Springer Nature Switzerland, 2022, 645-662. https://doi.org/10.1007/978-3-031-20062-5_37

27. Ning, R., Zhang, C., Zou, Y. SRF-Net: Selective Receptive Field Network for Anchor-Free Temporal Action Detection. In ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, 2460-2464. https://doi.org/10.1109/ICASSP39728.2021.9414253

28. Shao, J., Wang, X., Quan, R., Zheng, J., Yang, J., Yang, Y. Action Sensitivity Learning for Temporal Action Localization. 2023 IEEE/CVF International Conference on Computer Vision (ICCV), 13411-13423, 2023. https://doi.org/10.1109/ICCV51070.2023.01238

29. Shi, D., Zhong, Y., Cao, Q., Ma, L., Li, J., Tao, D. TriDet: Temporal Action Detection with Relative Boundary Modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, 18857-18866. https://doi.org/10.1109/CVPR52729.2023.01808

30. Tang, T. N., Kim, K., Sohn, K. TemporalMaxer: Maximize Temporal Context with Only Max Pooling for Temporal Action Localization. arXiv preprint arXiv:2303.09055, 2023.

31. Tolias, G., Sicre, R., Jégou, H. Particular Object Retrieval with Integral Max-Pooling of CNN Activations. CoRR, vol. abs/1511.05879, 2015.

32. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. Attention Is All You Need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, 6000-6010.

33. Wang, B., Zhao, Y., Yang, L., Long, T., Li, X. Temporal Action Localization in the Deep Learning Era: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(1), 2171-2190. https://doi.org/10.1109/TPAMI.2023.3330794

34. Wang, J., Chen, K., Xu, R., Liu, Z., Loy, C. C., Lin, D. CARAFE: Content-Aware Reassembly of Features. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, 3007-3016. https://doi.org/10.1109/ICCV.2019.00310

35. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.

36. VideoMAE V2: Scaling Video Masked Autoencoders with Dual Masking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, 14549-14560. https://doi.org/10.1109/CVPR52729.2023.01398

37. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L. V.

38. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In ECCV, 2016.

39. Wang, Q., Zhang, Y., Zheng, Y., Pan, P. RCL: Recurrent Continuous Localization for Temporal Action Detection. In 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2022, 13556-13565. https://doi.org/10.1109/CVPR52688.2022.01320

40. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., Xing, S., Chen, G., Pan, J., Yu, J., Wang, Y., Qiao, Y.

41. InternVideo: General Video Foundation Models via Generative and Discriminative Learning. arXiv preprint arXiv:2212.03191, 2022.

42. Xu, H., Das, A., Saenko, K. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In Proceedings of the International Conference on Computer Vision (ICCV), 2017. https://doi.org/10.1109/ICCV.2017.617

43. Xu, H., Das, A., Saenko, K. Two-Stream Region Convolutional 3D Network for Temporal Activity Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019,41(10), 2319-2332. https://doi.org/10.1109/TPAMI.2019.2921539

44. Xu, L., Wang, X., Liu, W., Feng, B. Cascaded Boundary Network for High-Quality Temporal Action Proposal Generation. IEEE Transactions on Circuits and Systems for Video Technology, 2020, 30(10), 3702-3713. https://doi.org/10.1109/TCSVT.2019.2944430

45. Xu, M., Zhao, C., Rojas, D. S., Thabet, A., Ghanem, B. G-TAD: Sub-Graph Localization for Temporal Action Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020. https://doi.org/10.1109/CVPR42600.2020.01017

46. Yang, G., Lei, J., Zhu, Z., Cheng, S., Feng, Z., Liang, R. AFPN: Asymptotic Feature Pyramid Network for Object Detection. arXiv preprint arXiv:2306.15988, 2023. https://doi.org/10.1109/SMC53992.2023.10394415

47. Zhang, C. L., Wu, J., Li, Y. ActionFormer: Localizing Moments of Actions with Transformers. In European Conference on Computer Vision, ser. LNCS, vol. 13664, 2022, 492-510. https://doi.org/10.1007/978-3-031-19772-7_29

48. Zhang, D., Zhang, H., Tang, J., Wang, M., Hua, X., Sun, Q. Feature Pyramid Transformer. Berlin, Heidelberg: Springer-Verlag, 2020, 323-339. https://doi.org/10.1007/978-3-030-58604-1_20

49. Zhao, C., Thabet, A. K., Ghanem, B. Video Self-Stitching Graph Network for Temporal Action Localization. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 13658-13667. https://doi.org/10.1109/ICCV48922.2021.01340

50. Zheng, W., Liu, L., Ye, R. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(07), 12993-13000. https://doi.org/10.1609/aaai.v34i07.6999

51. Zhou, Z., Shojafar, M., Abawajy, J., Bashir, A. K. IADE: An Improved Differential Evolution Algorithm to Preserve Sustainability in a 6G Network. IEEE Transactions on Green Communications and Networking, 2021, 5(4), 1747-1760. https://doi.org/10.1109/TGCN.2021.3111909

52. Zhu, Z., Tang, W., Wang, L., Zheng, N., Hua, G. Enriching Local and Global Contexts for Temporal Action Localization. 2021 IEEE/CVF International Conference on Computer Vision (ICCV), 2021, 13496-13505. https://doi.org/10.1109/ICCV48922.2021.0132