The First International Conference on
Machine Learning and Knowledge Discovery (MLKD 2024)
Amirkabir University of Technology, December 18-19, 2024

Enhancing Temporal Action Localization with 3D Input and Attention Mechanisms

Mozhgan Mokari* Khosrow Haj Sadeghi†

Abstract

Temporal action localization (TAL) in untrimmed videos poses a significant challenge due to the accurate determination of action timing and type within noisy or irrelevant content. In this paper, we introduce SeqAttNet, an innovative end-to-end network that aims to advance TAL performance through novel approaches. Our model combines attention mechanisms with a compact two-dimensional sequential network and utilizes 3D input aggregation to optimize accuracy and computational efficiency. SeqAttNet outperforms existing methods, achieving over 87% greater efficiency on the ActivityNet dataset while being over 70 times smaller compared to baseline. Despite its compact nature, SeqAttNet maintains competitive accuracy, surpassing larger models such as TriDet in overall efficiency and achieving more than twice the efficiency on the ActivityNet dataset. Our findings demonstrate that SeqAttNet effectively balances performance with computational cost, delivering high accuracy while notably reducing network complexity. This makes it a valuable tool for practical TAL applications, where both precision and efficiency are essential.

Keywords: Temporal action localization, efficient attention, and Action recognition

1 Introduction

Human action recognition is essential for applications like surveillance, behavior analysis, and video retrieval. While significant progress has been made, detecting actions in untrimmed videos remains a major challenge due to the need for accurate temporal localization within noisy or irrelevant content. Temporal action localization (TAL) requires both the identification of action boundaries and the action type, which is crucial for tasks like video retrieval. Existing TAL methods struggle with balancing performance and computational complexity, making them less practical for real-world applications.

*Department of Electrical Engineering, Sharif University of Technology, mozhgan.mokari@ee.sharif.edu
†Department of Electrical Engineering, Sharif University of Technology, ksadeghi@sharif.edu

Recent advances in attention mechanisms have improved accuracy by allowing models to focus on relevant segments of data. However, these gains often come at the cost of increased model complexity and higher computational demands.
In this paper, we propose SeqAttNet, a compact and efficient end-to-end network for TAL that integrates attention mechanisms and utilizes 3D input aggregation. Our model combines a two-dimensional sequential network with attention-based modules to maintain competitive accuracy while significantly reducing network size. Key contributions include:

- Development of an efficient, end-to-end TAL network
- Integration of attention mechanisms for enhanced accuracy
- Introduction of 3D input aggregation to maximize efficiency
- Introduction of a two-dimensional sequential network for computationally efficient processing
- Significant reduction in network complexity

The remainder of this paper is structured as follows: Section 2 reviews related work. Section 3 details the overall method and specific techniques used. The evaluation and results are presented in Section 4. Finally, the paper concludes in Section 5, where we summarize our findings and discuss their implications.

2 Related Work

Temporal action localization (TAL) has seen numerous advancements aimed at improving the precision of action boundary detection. Kong et al. [15] introduced the Boundary Likelihood Pinpointing (BLP) network, which enhances boundary accuracy through probabilistic approaches. However, the reliance on predefined anchors limits its flexibility, particularly for actions with extreme durations. To address this, Yang et al. [8] proposed A2Net, combining anchor-based and anchor-free mechanisms to detect actions of varying lengths, offering greater adaptability without anchor constraints.
Wang et al. [13] presented the Multi-Level Temporal

Pyramid Network (MLTPN), which improves feature discrimination through a pyramid architecture but at the cost of increased computational demands, potentially limiting its use in real-time applications. Graph-based methods have also contributed to TAL. Zeng et al. [11] introduced the Proposal-based Graph Convolutional Network (P-GCN), which uses graph convolutional networks to model relationships between action proposals, improving classification and localization. Similarly, Wang et al. [7] and Xu et al. [10] actively employ graph-based architectures to model temporal context and improve action localization performance. However, despite the improvements in feature representation and detection accuracy, the complexity of these graph structures results in significantly higher computational demands, making them less efficient for real-time or resource-constrained environments.

Liu and Wang [12] introduced the Progressive Boundary Refinement Network (PBRNet), which addresses vague action boundaries through a three-step cascaded detection process. PBRNet achieves state-of-the-art performance but involves a multi-stage refinement process. Liu et al. [17] explored the impact of end-to-end learning on temporal action detection, finding that end-to-end training improves performance. However, the simultaneous optimization of both the video encoder and detection head increases the model's overall computational burden.

Attention-based models have gained prominence since the introduction of "Attention Is All You Need" by Vaswani et al. [14]. Vaudaux-Ruth et al. [5] introduced SALAD, which incorporates a self-assessment learning approach that functions as an attention mechanism. SALAD improves detection accuracy by refining the focus on relevant frames, though it also requires additional computational resources due to the dual-task learning process. Zhao et al. [1] introduced SegTAD, reformulating TAL as a semantic segmentation task, achieving high accuracy with fine-grained supervision. Transformer-based models like TadTR [16] and RTD-Net [6] have further simplified the TAL pipeline but still demand significant computational power due to their reliance on attention mechanisms.

In [9], Mokari et al. developed an End-to-End network for parallel error estimation and classification, enhancing boundary precision. Building on these efforts, we introduces the SeqAttNet, which incorporates attention mechanisms into a simplified and efficient architecture, optimizing TAL performance while reducing computational load. We use [9] as a baseline for comparison and further improve accuracy and efficiency through innovative design choices.

3 Approach

3.1 Problem Definition

The task of Temporal Action Localization (TAL) involves detecting actions in untrimmed videos by determining the time intervals of action occurrences. Given a sequence of RGB frames I, the objective is to detect the actions within these frames, represented as action proposals $\Psi = \{\varphi_n = (t_{ns}, t_{ne}, C_n, S_n)\}$, where t_{ns} and t_{ne} indicate the start and end times of an action. C_n classifies the action type, and S_n provides a confidence score for each detection.

Our approach focuses on generating action proposals, estimating timing boundary errors, and classifying actions within a unified end-to-end framework. The key components of our method, including 3D input aggregation, temporal attention network and Two-Dimensional Sequential network, are designed to enhance both accuracy and efficiency.

3.2 3D Input Amassment

In our approach, video frames are first transformed into feature sequences using feature extraction techniques, representing the content within each frame. The feature vector $x_i \in \mathbb{R}^F$ at time step i captures the information from the corresponding frame, where F is the dimension of the feature vector.

Given the high dimensionality of these feature vectors and the large number of frames in each video, this results in highly complex input data. To avoid the need for overly large networks, we propose a technique called 3D Input Amassment, which reduces network size and computational requirements by converting the input into three-dimensional form and processing it with two-dimensional network structures.

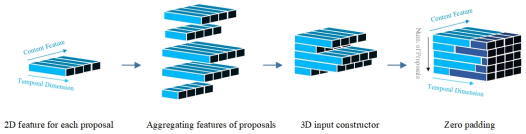

Figure 1: Preparing 3D Input.

In this approach, instead of concatenating feature vectors for each candidate proposal, we first create a three-dimensional feature space, as illustrated in Figure 1. For each candidate proposal, covering a specific time interval of the video, we align the feature vectors x_i corresponding to the frames within that interval along the time axis, forming a two-dimensional feature map. These maps are then aggregated along the third dimension to create a 3D input for the subsequent stages.

Since candidate proposals vary in length, the resulting three-dimensional input is zero-padded along the time

axis to ensure uniform dimensions across proposals. The final 3D input X, which contains the 2D feature maps of all proposals, is defined as $X \in \mathbb{R}^{N \times F \times T}$, where N represents the number of proposals, F is the feature vector dimension, and T is the temporal length.
As shown in Figure 2, this 3D input allows us to apply the same temporal attention network across different content dimensions, enabling efficient and accurate analysis of the action proposals.

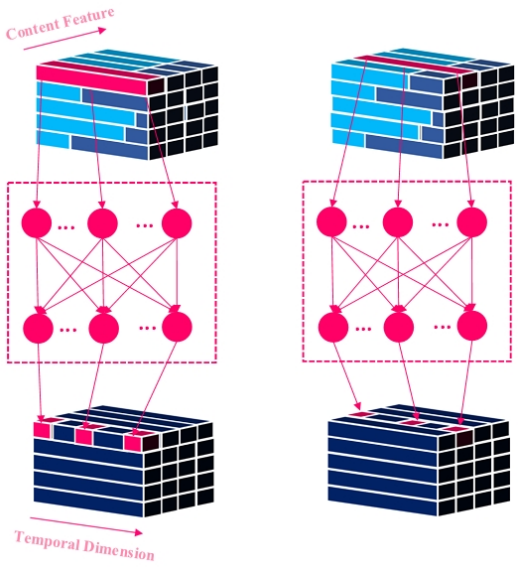

Figure 2: Applying Same Temporal Attention Network into different content dimension of 3D input.

3.3 Temporal Attention Network

To enhance efficiency while maintaining simplicity, many recent architectures leverage attention-based networks to prioritize crucial features within the input data. In our proposed method, we also incorporate attention mechanisms to control the influence of different features in the input, highlighting the more significant ones. Given that the input data spans multiple frames, representing various ways actions unfold over time, it is logical to assume that not all time intervals contribute equally to determining the action type and its boundaries.
In our approach, we employ the temporal attention network that operates along the time axis, focusing on the temporal relationships between frames. Since our input is organized in a three-dimensional structure—separated by time and content dimensions—the attention mechanism is applied exclusively in the temporal direction. Instead of assigning an individual attention network for each content dimension, we use a unified attention network across the entire time axis for each content, reducing the overall complexity of the network. This temporal attention network is a linear structure with dimensions $T \times T$, where T is the number of time steps. The attention network computes the importance of each time step in relation to the others, ensuring that the influence of content across different times is captured uniformly. As illustrated in Figure 2, the same attention network is applied to all content dimensions at each time step.
This temporal attention mechanism is utilized separately for classification and temporal error estimation, as the importance of different values within the 3D input varies based on the application. The temporal attention improves the accuracy of action detection by selectively focusing on the most relevant time intervals for each task.

3.4 Two-Dimensional Sequential Network

In this section, we present an additional technique designed to optimize the overall structure: the Two-Dimensional Sequential Network. Rather than employing a complex network that simultaneously processes the interactions between both content and time dimensions, we introduce a sequential structure that handles these dimensions in two distinct stages. This two-stage approach considers the time dimension and content dimension separately, which simplifies the computational process and reduces the complexity of the model. Since the two parts of the sequential network handle the input from different directions, we refer to it as a Two-Dimensional Sequential Network.

The technique is applied independently for both

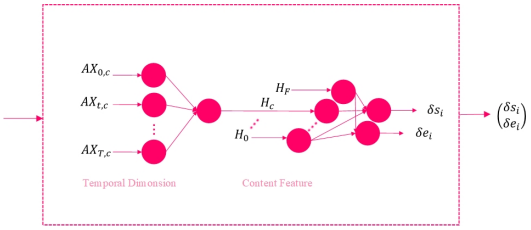

Figure 3: Two-Dimensional Temporal Error Estimation Network.

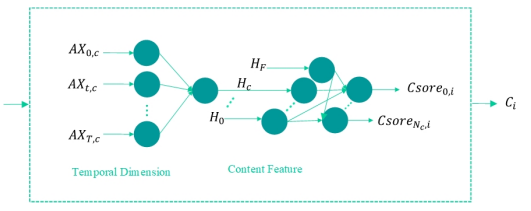

Figure 4: Two-Dimensional Classification Network.

classification and temporal error estimation, resulting

in two distinct network structures, as shown in Figures 3 and 4. In the first stage, the network processes the time dimension, utilizing a linear layer to compute an intermediate variable H_i for the intermediate layer. According to Eq. (1), H_i is a weighted linear combination of the values of the i-th content across different time steps, followed by the ReLU activation function. The input to this linear temporal layer is the original input, modified by the attention weights, denoted as AX.

$$H_i = \text{ReLU}\left(\sum_{j=1}^{T} w_j \cdot AX_{ji}\right) \tag{1}$$

In this stage, identical weights are used across all content dimensions, meaning that the effect of content values at different time steps is uniformly dependent on the temporal relationships. In the second stage of the network, the content dimension is processed, generating the final output variables through a linear layer based on the intermediate content values from the previous stage.
This sequential processing approach simplifies the interactions between time and content dimensions, reducing the overall model complexity. By decoupling the temporal and content relationships into separate stages, we achieve a more efficient network structure while maintaining the accuracy and performance of the model.

3.5 SeqAttNet Overall Architecture

SeqAttNet is designed to efficiently handle Temporal Action Localization (TAL) using a combination of specialized modules. Each component is carefully crafted to balance accuracy with computational efficiency, ensuring an end-to-end structure capable of generating action proposals, estimating temporal boundaries, and classifying actions in untrimmed videos. The overall architecture of SeqAttNet is shown in Figure 5, where the different blocks are color-coded to distinguish their functions.
The key stages of SeqAttNet are as follows:

- Proposal Generation: Produces candidate action proposals.
- Temporal Error Estimation: Refines proposal timing with error correction.
- Action Classification:Assigns action labels to proposals.
- Inference and Post-processing: Filters and refines final action proposals.

Detailed descriptions of each module follow.

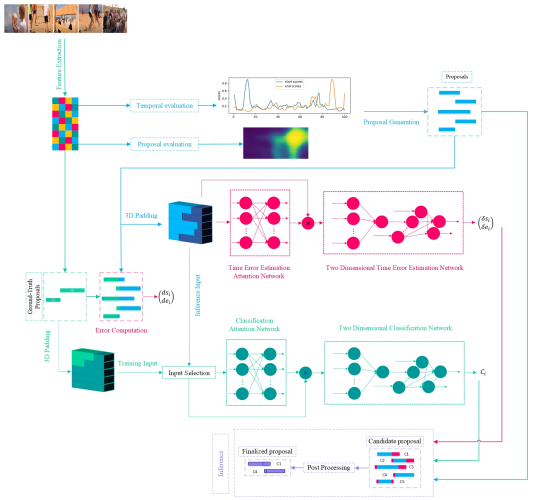

Figure 5: The overall architecture of proposed method, SeqAttNet.

3.5.1 Proposal Generation

The Proposal Generation module is tasked with identifying candidate time intervals where actions occur. Inspired by the framework in [9], the proposal generation process evaluates the temporal structure of the input video from two perspectives: temporal evaluation and proposal evaluation.
In temporal evaluation, the likelihood of an action starting or ending at each time point is estimated, while proposal evaluation generates a Confidence matrix that predicts the probability of an action starting at a given time and lasting for a specific duration. This matrix helps identify candidate proposals, which are then used in subsequent stages. The structure of the evaluation module is detailed in Table 1.

Table 1: Details of evaluation module architecture.

Layer	Kernel	Stride	Dimension	Activation function	Output
Base Module					
Conv1D	3	1	256	Relu	$256 \times T$
Conv1D	3	1	256	Relu	$256 \times T$
Temporal Evaluation Module					
Conv1D	3	1	256	Relu	$256 \times T$
Conv1D	1	1	1	Sigmoid	$1 \times T$
Proposal Evaluation Module					
Conv1D	3	1	256	Relu	$256 \times T$
Boundary Matching			N-32		$256 \times 32 \times D \times T$
Conv3D	32,1,1	32,0,0	512	Relu	$512 \times 1 \times D \times T$
Squeeze					$512 \times D \times T$
Conv2D	1,1	0,0	128	Relu	$128 \times D \times T$
Conv2D	3,3	1,1	128	Relu	$128 \times D \times T$
Conv2D	3,3	1,1	128	Relu	$128 \times D \times T$
Conv2D	1,1	0,0	2	Sigmoid	$2 \times D \times T$

3.5.2 Error Estimation

In the temporal error estimation step, the temporal errors δs_i and δe_i for each candidate proposal's start and

end times are calculated. This is done using a two-dimensional sequential network combined with a temporal attention mechanism. The error estimates allow the model to refine the start and end points of the detected action
The attention network first calculates attention weights for each time step, which are applied to the input features to generate weighted inputs. The weighted inputs then pass through a sequential network to estimate the temporal errors.
Simultaneously, using the ground-truth data, the exact temporal errors ds_i and de_i are calculated as:

$$ds_i = \frac{t_{gs} - t_{is}}{T}, \quad de_i = \frac{t_{ge} - t_{ie}}{T}$$

where t_{gs} and t_{ge} are the ground-truth start and end times, and t_{is} and t_{ie} are the predicted start and end times of the proposal.

Table 2: Details of Time Error Estimation architecture.

Layer	Dimension	Activation function	Output
Attention Network			
Linear	$T \times T$	-	Attention Map
Production	$T \times T$	-	$AX_{T \times F}$
Two-Dimensional Sequential Network			
Linear	$T \times 1$	Relu	H_F
Linear	$F \times 2$	tanh	δs_i and δe_i

3.5.3 Classification

The Action Classification module is responsible for assigning action labels to the proposals generated in the earlier stages. Like the error estimation module, the classification network applies an attention map to the 3D input features, highlighting the most important content for classification. A two-dimensional sequential network then generates the classification score $Cscore_k$, which indicates the likelihood of an action belonging to class k.
The architecture of the classification module is presented in Table 3. During the training phase, ground-truth proposals are used to prevent the propagation of errors from the proposal generation stage, while in the inference phase, the proposals generated by the model are classified.

3.6 Inference

In the inference stage, the final set of action proposals is generated and scored. Each proposal is scored based on both the proposal confidence P and the classification score $Cscore_{C_i}$ using the following equation:

$$S_i = P \times Cscore_{C_i}$$

Table 3: Details of Classification architecture.

Layer	Dimension	Activation function	Output
Attention Network			
Linear	$T \times T$	-	Attention Map
Production	$T \times T$	-	$AX_{T \times F}$
Two-Dimensional Sequential Network			
Linear	$T \times 1$	Relu	H_F
Linear	$F \times N_c$	Logsoftmax	$Cscore_{N_c}$

where S_i represents the overall confidence for proposal i. The proposals with the highest scores are selected as the final set of action detections.

3.6.1 Post Processing

Since only one action occurs at any time, temporal overlaps between action proposals are irrelevant. Therefore, to reduce false positives, Soft Non-Maximum Suppression (SNMS) [4] is applied to the overlapping proposals, reducing the likelihood of selecting less probable ones. SNMS adjusts the confidence scores of proposals with temporal overlaps, ensuring that the most likely proposals are chosen.

3.7 Training

The details related to the training of the proposed structure are examined in this section. These details include the preparation of the data required for training the model and the loss function used.

3.7.1 Training Data

The training data for the network comprises annotated intervals of action instances with corresponding class labels per video. Sequence labels G_S and $G_E \in \mathbb{R}^T$, generated for the temporal boundaries of proposals, are employed in temporal evaluation training.
For each action instance in the ground truth $\varphi_g = (t_s, t_e)$, where the duration $d_g = t_e - t_s$, the start and end regions are delineated as:

$$r_S = [t_s - \frac{d_g}{10}, t_s + \frac{d_g}{10}] \tag{2}$$

$$r_E = [t_e - \frac{d_g}{10}, t_e + \frac{d_g}{10}] \tag{3}$$

For every time step t_n, the temporal region is defined as:

$$r_{t_n} = [t_n - \frac{d_f}{10}, t_n + \frac{d_f}{10}] \tag{4}$$

where $d_f = t_n - t_{n-1}$. We compute the maximum intersection over region ratio for each temporal region with

start and end regions independently. These values are represented as $g^s_{t_n}$ and $g^e_{t_n}$ for the start and end ground-truth probabilities of t_n. Thus,

$$G_S = \{g^s_{t_n}\}^T_{n=1} \tag{5}$$

$$G_E = \{g^e_{t_n}\}^T_{n=1} \tag{6}$$

To establish the label map $G_C \in \mathbb{R}^{D\times T}$ for proposal evaluation, serving as the ground-truth for the confidence matrix, we compute the Intersection over Union (IoU) between each hypothetical proposal and the ground-truth instances. The maximum IoU values are then utilized to populate the G_C matrix. We consider all conceivable proposals, starting from every temporal point and spanning diverse durations.

3.7.2 Loss function

The loss function definitions for all modules are integrated to enable an end-to-end network training process. This combined loss is represented by Eq. (7):

$$L = L_{ev} + \lambda_{cl} \cdot L_{cl} + \lambda_{er} \cdot L_{er} + \lambda_2 \cdot L_2(\Theta) \tag{7}$$

In this equation, L_{ev} is the evaluation loss, L_{cl} represents the classification loss, and L_{er} indicates the mean squared error for the temporal error estimation outputs. The regularization term L_2 applies norm-2 regularization to the parameters of the overall loss function.
The individual loss functions are defined as follows. The Cross Entropy loss, used for L_{cl}, is given in Eq. (8). Here, B denotes the batch size, $y_{i,c} \in \{0, 1\}$ is the binary ground truth label, and $P_{i,c} \in [0, 1]$ is the predicted action probability for the i-th proposal in the c-th action class:

$$L_{cl} = \frac{1}{B}\sum_{i=1}^{B}\sum_{c=1}^{N_c} y_{i,c}\log(P_{i,c}) \tag{8}$$

The evaluation loss L_{ev} comprises L_{te}, the temporal evaluation loss, and L_{pe}, the proposal evaluation loss for confidence matrices. L_{te} is computed as shown in Eq. (9), where P_S and P_E represent the predicted start and end probability sequences:

$$L_{te} = L_{bl}(P_S, G_S) + L_{bl}(P_E, G_E) \tag{9}$$

Here, $L_{bl}(P, G)$ refers to the weighted binary logistic regression loss, detailed in Eq. (10):

$$\frac{1}{l_w}\sum_{i=1}^{l_w}(\alpha^+ \cdot b_i \cdot \log(p_i) + \alpha^- \cdot (1 - b_i) \cdot \log(1 - p_i)) \tag{10}$$

In this formula, b_i is the binary equivalent of g_i after applying a threshold $\theta = 0.5$. The weights α^+ and α^- are computed using Eq. (11) and Eq. (12), with l_w representing the temporal length of the video:

$$\alpha^+ = \frac{l_w}{l^+} \tag{11}$$

$$\alpha^- = \frac{l_w}{l^-} \tag{12}$$

where

$$l^+ = \sum b_i \tag{13}$$

$$l^- = l_w - l^+ \tag{14}$$

The Proposal Evaluation module employs two different loss functions to produce two distinct confidence matrices: Regression Map M_{CR} and Binary Classification Map M_{CC}. These confidence matrices are used to assess proposals as previously explained:

$$L_{pe} = L_C(M_{CC}, G_C) + \lambda_R \cdot L_R(M_{CR}, G_C) \tag{15}$$

Here, L_C is L_{bl}, and L_R is the norm-2 regularization. The evaluation loss function L_{ev} is defined in Eq. (16):

$$L_{ev} = L_{pe} + \lambda_{te} \cdot L_{te} \tag{16}$$

Finally, the total loss function is defined by combining all individual losses, as specified in Eq. (17). The proposed end-to-end network is trained by optimizing L:

$$L = L_{pe} + \lambda_{te} \cdot L_{te} + \lambda_{cl} \cdot L_{cl} + \lambda_{er} \cdot L_{er} + \lambda_2 \cdot L_2(\Theta) \tag{17}$$

4 Experiment

In this section, we evaluate the performance of our proposed end-to-end network, comparing it with state-of-the-art methods, and provide a detailed analysis of the results.

4.1 ActivityNet Dataset

The ActivityNet dataset [3] is a crucial collection of videos that aids in comprehending human actions through an extensive array of untrimmed footage. This dataset covers 648 hours and includes 200 varied action classes, with around 100 videos per class, amounting to approximately 23,000 activity instances for analysis. These classes encompass a wide range of everyday activities. ActivityNet is our dataset due to its depiction of various daily activities—most untrimmed videos feature multiple action instances, presenting the dual challenges of undefined action boundaries and numerous action instances for our method to address.

4.2 Evaluation Metric

We use the Average Precision (AP) metric to measure performance, calculated for each action class C using Eq. (18). Mean Average Precision (mAP) is derived by averaging AP values across all classes for different Temporal Intersection over Union (tIoU) thresholds, as shown in Eq. (19).

$$AP(c) = \frac{\sum_{k=1}^{n}(prec(k) \times rel(k))}{\sum_{k=1}^{n} rel(k)} \tag{18}$$

$$mAP = \frac{1}{N_c}\sum_{c=1}^{N_c} AP(c) \tag{19}$$

4.3 Setup

The model's hyperparameters were fine-tuned through limited validation evaluations, with final values listed in Table 4. These values were used for the final performance evaluations.

Table 4: Hyper parameters setting.

Loss Function		Training	
λ_R	10	Batch size(Nb)	16
λ_{te}	1	Learning rate	0.001
λ_{cl}	1	epoch	5
λ_{er}	1	-	-
λ_2	0.001	-	-

4.4 Performance

We assessed our method on the ActivityNet dataset and compared it to recent approaches. As shown in Tables 5, the efficiency of the proposed structure is significantly higher than all existing approaches. For instance, although the TriDet [2] method offers higher accuracy than all other methods, it utilizes a network more than three times larger than the proposed structure, resulting in lower overall efficiency.

The comparison with the baseline [9] reveals that incorporating attention mechanisms alongside the proposed novelties not only preserved the accuracy of the structure but also led to increased accuracy. This, combined with a substantial reduction in network size—over 70 times smaller—resulted in an approximately 87% improvement in overall efficiency.

The results indicate that the proposed structure achieves a highly optimized design, maintaining network performance while significantly reducing the network's size.

Table 5: Temporal action localization results on validation and testing set of ActivityNet v1.3 where threshold for mAP is changed from 0.5 to 0.95 with 0.05 step (mAP@tIoU).

Methods	Param.	@0.95	@0.75	@0.5	Average	Ave/Param
PBRNet [12]	89M	8.98	34.97	53.96	35.01	0.39
RTD-Net [6]	32M	8.61	30.68	47.21	30.83	0.96
E2E [17]	36M	7.84	33.26	48.97	32.65	0.9
TriDet [2]	16M	**8.4**	**38.0**	**54.7**	**36.8**	2.3
Baseline [9]	367M	5.18	28.39	37.88	25.28	0.06
SeqAttNet (Ours)	**5M**	6.17	29.85	39.6	26.38	**5.27**

To further evaluate the proposed method, several predicted samples are shown in Figure 6. As observed, the action class and the overall duration have been correctly identified, with some boundaries being highly precise. However, in some instances, despite a good overlap with the ground truth data, the boundaries differ by a few seconds from the exact action boundaries. While these discrepancies may be expected given the video quality, improving the accuracy of boundary detection could significantly enhance the overall performance of the method.

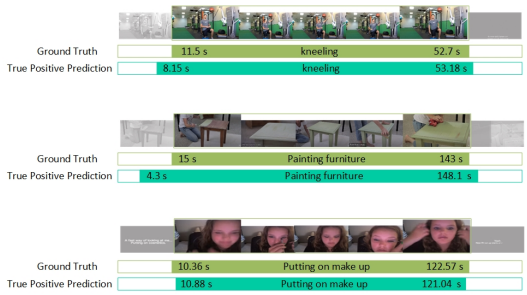

Figure 6: True-Positive samples of predictions.

5 Conclusion

In this paper, we introduced SeqAttNet, a novel and efficient network designed for Temporal Action Localization (TAL), which integrates attention mechanisms with 3D input amassment to form a compact Two-Dimensional Sequential Network. Our proposed architecture significantly reduces the computational complexity and network size while maintaining strong performance in action classification and temporal error estimation. Compared to state-of-the-art methods, SeqAttNet achieves a notable improvement in the accuracy-to-parameters ratio, as demonstrated by experiments on the ActivityNet dataset. The results show that SeqAttNet outperforms most current methods in terms of efficiency, achieving competitive accuracy while using a fraction of the parameters. Specifically, SeqAttNet reduces the number of network parameters by more than

70 times compared to the baseline, leading to an 87% improvement in efficiency on the ActivityNet dataset. While the network delivers precise action class identification and overall duration estimation, some minor discrepancies remain in boundary predictions. These could be further addressed by refining boundary detection techniques, potentially enhancing the method's overall performance. Nevertheless, SeqAttNet represents a highly optimized solution for TAL tasks, offering a balance of accuracy and computational efficiency, making it suitable for real-world applications where both performance and resources are critical considerations.

References

[1] Ch. Zhao, M. Ramazanova, M. Xu, and B. Ghanem. SegTAD: Precise Temporal Action Detection via Semantic Segmentation. *European Conference on Computer Vision*, 576–593, 2022.

[2] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li and D. Tao. Tridet: Temporal Action Detection with Relative Boundary Modeling. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18857–18866, 2023.

[3] F. Caba Heilbron, V. Escorcia, B. Ghanem and J. Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. *Proceedings of the ieee conference on computer vision and pattern recognition*, 961–970, 2015.

[4] F. Caba Heilbron, V. Escorcia, B. Ghanem and J. Carlos Niebles. Soft-NMS–improving object detection with one line of code. *Proceedings of the IEEE international conference on computer vision*, 5561–5569, 2017.

[5] G. Vaudaux-Ruth, A. Chan-Hon-Tong and C. Achard. SALAD: Self-assessment learning for action detection. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, jan, 2021.

[6] J. Tan, J. Tang, L. Wang and G. Wu. Relaxed Transformer Decoders for Direct Action Proposal Generation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 13526–13535, 2021.

[7] L. Wang, Ch. Zhai, Q. Zhang, W. Tang, N. Zheng and G. Hua. Graph-based temporal action co-localization from an untrimmed video. *Neurocomputing*, 434:211–223, 2021.

[8] L. Yang, H. Peng, D. Zhang, J. Fu and J. Han Revisiting anchor mechanisms for temporal action localization. *IEEE Transactions on Image Processing*, 29:8535–8548, 2020.

[9] M. Mokari and Kh. Haj Sadeghi Enhancing temporal action localization in an end-to-end network through estimation error incorporation. *Image And Vision Computing*, 145:104994, 2024.

[10] M. Xu, Ch. Zhao, D. Rojas, A. Thabet and B. Ghanem. G-TAD: Sub-Graph Localization for Temporal Action Detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10156–10165, 2020.

[11] Q. Liu and Z. Wang. Graph Convolutional Networks for Temporal Action Localization. *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 34:11612–11619, 2020.

[12] Q. Liu and Z. Wang. Progressive boundary refinement network for temporal action detection. *Proceedings of the AAAI conference on artificial intelligence*, 34:11612–11619, 2020.

[13] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang and Ch. Gan Multi-level Temporal Pyramid Network for Action Detection. *Chinese Conference on Pattern Recognition and Computer Vision (PRCV)*, 7094–7103, 2019.

[14] V. Ashish. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[15] W. Kong, N. Li, Sh.Liu, Th. Li and G. Li BLP-boundary likelihood pinpointing networks for accurate temporal action localization. *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1647–1651, 2019.

[16] X. Liu, Q. Wang, Y. Hu, X. Tang, Sh. Zhang, S. Bai, and X. Bai. End-to-End Temporal Action Detection with Transformer. *IEEE Transactions on Image Processing*, 31:5427–5441, 2022.

[17] X. Liu, S. Bai and X. Bai. An empirical study of end-to-end temporal action detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20010–20019, 2022.