TadML: A fast temporal action detection with Mechanics-MLP

Bowen Deng[1,2], Shuangliang Zhao[2], and Dongchang Liu[1]

[1] Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China
dongchang.liu@ia.ac.cn
[2] Guangxi University, Nanning, 69121, China
szhao@gxu.edu.cn

Abstract. Temporal Action Detection (TAD) involves identifying action categories and their respective start and end frames in lengthy untrimmed videos, with current models utilizing both RGB and optical flow streams that require manual intervention, add computational complexity, and consume time. Moreover, two-stage approaches prioritizing proposal generation in the ini-tial stage result in a substantial reduction in inference speed. To address this, we propose a single-stage anchor-free method that solely utilizes the RGB stream and incorporates a novel Newtonian Mechanics-MLP architec-ture. Our model achieves comparable accuracy to existing state-of-the-art models but with significantly faster inference speeds, clocking in at an av-erage of 4.44 videos per second on THUMOS14. Our approach showcases the potential of MLP in downstream tasks like TAD. The source code is available at https://github.com/BonedDeng/TadML.

Keywords: Temporal action dcetection · MLP-like · RGB and optical flow · Real time · Anchor free.

1 Introduction

As videos become ubiquitous in the wake of advances in mobile communication and the internet, video understanding has become increasingly important in both academia and industry. In particular, temporal action detection, detecting categories, start and end timestamps of human actions in untrimmed footage, has diverse applications in areas such as human-computer interaction, video surveillance, and intelligent security[1]. In the past, numerous TAD frameworks employed complex pipelines. Some earlier methods even utilized manually crafted features, including color and texture features of each frame, for video action classification and detection. Currently, research on temporal action detection (TAD) has shifted towards utilizing deep models that combine raw RGB streams and optical flow, emerging as the mainstream and potential approach. RGB frames contain vital shape and spatial information of videos, which are necessary for Temporal Action Detection (TAD). Most TAD research also use optical flow, a two-dimensional velocity field that captures action information and three-dimensional scene structure of the observed object. However, fusing RGB and optical flow data requires time-consuming conversion, computations, and resources. TAD has two objectives: predicting action categories based on available information and their corresponding timestamps in the video. While TAD shares similarities with object detection, it is

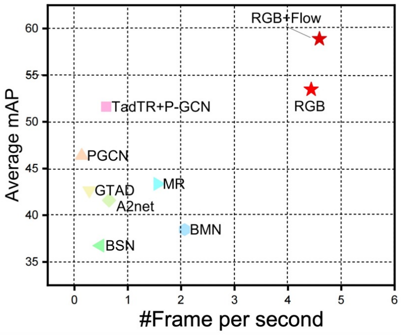

Fig. 1. Comparing to the performance (average chart) and speed of the latest time action detection model on THUMOS 14. Our method shows advanced performance and very fast speed when using RGB stream.

focused on detecting actions in the time domain, unlike object detection that identifies positions in the spatial domain, leading TAD methods to draw inspiration from object detection research. TAD models can be categorized into one-stage and two-stage models based on their network structure. Two-stage frameworks generate proposals with high recall, which are then classified to predict corresponding labels, but their inference speed is slower and incurs higher computing costs than one-stage frameworks. One-stage frameworks simultaneously generate the start and end frames of each action and their corresponding labels in a single step, making them more efficient for real-time applications.

Based on the anchor structure, previous research can be categorized into three groups: (a) action-guided methods, such as BSN[2], (b) anchor-based methods, such as BMN[3], and (c) anchor-free methods, such as AFSD[4]. Methods that employ anchors not only exhibit high time complexities, namely (T^2) and $(c*t)$, but also require numerous hyperparameters to be fine-tuned, including the scale and quantity of anchors, as well as the computational cost of IOU thresholds[5]. Drawing inspiration from anchor-free models in target detection research, anchor-free methods have emerged as the mainstream approach and demonstrated significant potential in TAD. In this paper, we present

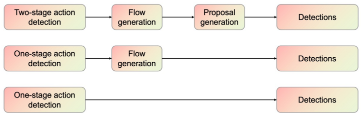

Fig. 2. The image showcases three mainstream methods. the traditional two stream method, the two stream one stage method and the RGB only one stage method.

a novel TAD model called Mechanics-MLP, which employs a one-stage anchor-free framework and considers each token as a force. The Newtonian Mechanics-MLP unit inspired by MLP's success in computer vision backbones, achieved promising results using the $\beta - \text{GIoU}$ loss for TAD, with our Tad-ML model achieving a maximum average precision (mAP) of 69.73% (RGB and optical flow streams at tIoU=0.4) on THUMOS14 while exhibiting rapid processing speeds and superior accuracy compared to other methods, as demonstrated in Figure 1a. As shown in figure 2, illustrating that B represents a two-stage method, C represents a one-stage method, and D represents an end-to-end one-stage method. Our pro-posed method eliminates the need for the optical flow conversion pipeline, leading to faster processing. In conclusion, our Mechanics-MLP TAD model demonstrates promising results and offers a fresh perspective on designing one-stage anchor-free frameworks for TAD tasks. In summary, our paper has the following contributions:

- To the best of our knowledge, TadML achieves state-of-the-art or highly competitive performance on benchmark data while significantly surpassing previous methods in terms of inference speed, achieving an impressive 4.44 videos per second inference speed on THUMOS14.
- TadML demonstrates that optical flow data is not necessary for TAD tasks, thus improving the model's inference speed and improves performance in both RGB stream and two-stream by optimizing neck layers, while also finding $\beta - Giou$ to be a more appropriate metric for TAD.
- Our Newtonian mechanics-based MLP model confirms the applicability of MLP for TAD and achieves highly competitive results using both RGB and optical flow data.

2 Related Works

This section provides a comprehensive review of previous studies related to Action Recognition, Temporal Action Detection, and MLP.

Temporal Action Recognition. Action recognition in video clips, which involves identifying human actions in 2D frame sequences, has traditionally relied on manual feature extraction and classification through methods such as HOG, HOF, Dense Trajectories, SVM, and RF, but deep learning methods like Lrcns, R(2+1), and I3D have since become dominant in the field. These models use CNNs to extract spatial features, while RNNs like LSTM[7] and 3D convolutions enable temporal feature extraction and improved re-mote loss and long-distance time modeling[8]. The goal of sequential action detection is to identify action instances, time boundaries, and categories in videos, with two-stage methods dividing videos into proposals before assigning them to specific categories and one-stage models directly localizing and classifying actions for improved efficiency. The BMN model simplifies the BSN process and improves efficiency through a boundary matching mechanism, while PGCN[9] employs graph convolutional networks to facilitate context and background information exchange. In contrast to two-stage models, one-stage models directly localize and classify actions, resulting in improved efficiency. For instance, SSAD simultaneously performs category prediction, time series offset correction, and IOU prediction, bypassing the requirement of initially predicting candidate time intervals. AFSD[4] maximizes the utilization of boundary characteristics and extracts essential boundary features through boundary pooling.

MLP. Various new computer vision architectures, including transformers and MLP, have recently demonstrated superior performance compared to CNN in several upstream tasks. Visual MLP-Like methods exhibit simplistic stacked MLP architectures, as exemplified by MLP-Mixer, which applies MLP independently and across image patches and has achieved comparable performance to SOTA models on the ImageNet dataset. The MorphMLP[10] architecture emphasizes local information in the low-level layer and gradually transitions to a long-term model in the high-level layer. The WaveMLP[11] introduce quantum mechanics into MLP. Both of these architectures have demonstrat-ed competitive performance in image classification tasks. Motivated by these findings, our objective is to investigate the potential of MLP-Like architectures in TAD, showcasing their applicability in visual downstream tasks as well.

3 Method

3.1 Overview

TadML architecture features three elements, including a backbone module for feature extraction and down-sampling in time, a time fusion pyramid network (TFPN) as the neck, and action and time prediction branches acting as the head. Representing the untrimmed video datasets as $D = \{D_{train}, D_{test}\}$, each video in set as $V \in \mathbb{R}^{T \times C \times H \times W}$, where T,C,H,W represent time step, channel, height and width respectively. In most TAD tasks, V will be converted into (V_{rgb}, V_{opt}) first, where V_{rgb} contains RGB streams, and V_{opt} contains optical flow. This conversion takes a lot of time and calculation resources. Our model only takes RGB data as input. We obtain the output via the module, where output $Y = (d_{i,s}, d_{i,e}, c_i)$. Here, $d_{i,s}$, $d_{i,e}$ are the distances between the current time step and the start and end of this action. A moment is either part of one action category or part of the background category, denoted with c_i.

3.2 Architecture

Mechanics token mixing block. Compared with the time complexity of the multi-head attention mechanism in Transformer, we aim to develop a similar approach that is relatively simple for rapid application in video understanding tasks. The MLP-like model is a neural architecture that is primarily made up of fully-connected layers and non-linear activation functions. We enhance token aggregation by dynamically adjusting the relationship between tokens and fixed weights in MLP through the application of Newtonian mechanics principles. In this work, given $D^0 \in \{X_1, X_2 ..., X_n\}$ with n time steps. z^0 is projected by F_a and F_b with FC layer, where the angle between F_a and F_B is θ and W^i and W^j are the weight with learnable parameters. According to the laws of mechanics, their resultant force is A_f , which is calculated by summing the vectors of F_a and F_b.

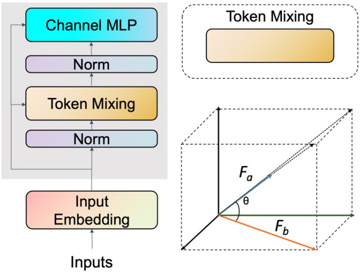

Fig. 3. Left the diagram of a block in the Mechanics-MLP architecture, right is token mixing.

$$
\begin{aligned}
&Token_{mechanics} = \{X_1, X_2 ..., X_n\} \\
&F_a = FC(Token_{mechanics}, W^i) \\
&F_b = FC(Token_{mechanics}, W^j) \\
&A_f = \sqrt{F_a + F_b + 2F_aF_b \cos\theta}
\end{aligned}
\tag{1}
$$

The inputs (embedding) in a basic mechanics unit undergo sequential processing through a mechanics token mixing block and a channel mixing block. Both two mixing block operation capture spatial information by blending features from multiple tokens. Furthermore, a layer normalization step is performed before each mixing operation, as depicted in Figure 3. The mechanics token mixing MLP consists of one MTM module, which aggregates various tokens by considering both F_1 and F_2, and applies the ReLU activation function. The channel mixing MLP extracts features for each token.

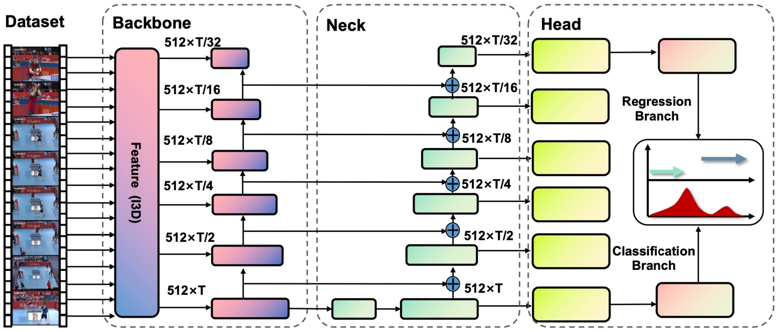

Fig. 4. The architecture consists of three main parts: a backbone module for feature extraction and downsampling in time, a time fusion pyramid network (TFPN) serving as the neck, and action and time prediction branches operating as the head.

$$\begin{aligned}
X &= Norm(X) \\
Z &= A_f(X) \\
Z &= Norm(Z) \\
Z &= Channel - FC(Z)
\end{aligned} \tag{2}$$

Backbone. To begin with, we generate video clips with a constant time window from the input video and reshape each clip to (T,C,H,W). Feature extraction converts the video into a sequence of feature vectors corresponding to the RGB visual modality. Current TAD methods struggle to achieve fast detection due to their reliance on optical flow, which consumes computing resources and introduces time-consuming conversion processes. The conversion process is bypassed in our model's backbone module, eliminating the need for cumbersome operations. The backbone network utilizes a pre-trained I3D model on the Kinetics datasets to extract 3D features. The videos are segmented into short, overlapping 8-frame chunks. Finally, we obtain $Y_{I3D} = (Y_{rgb}, Y_{opt})$, where both of them are 1024-dim features. Different from previous, our method only needs Y_{rgb}. We also tried Y_{I3D} as our input, and it also played a good effect, indicating the superiority of our method. The feature is flattened along the last four dimensions to form a two-dimensional feature sequence encompassing both temporal and spatial information from the entire video. The feature sequence is then passed through a multi-layer semantic module (MSM) consisting of six down-sample layers. Each layer, composed of mechanics units, has an output dimension of 512. The outputs of the layers are (512, t/2), (512, t/4), (512, t/8), (512, t/16), (512, t/32), and (512, t/64), as shown in Figure 4. This module produces multi-coarse texture basic features as its output. Temporal Feature Pyramid Network. the TAD architecture incorporates the concept of object detection, making the utilization of complex adaptive attention modules in AFFPN and the repeated superposition method in BIFPN impractical. The neck is designed to establish connections between semantically strong, low-temporal-resolution features and semantically weak, high-temporal-resolution features. This is achieved through a six-layer pathway comprising ((T,512), (T/2,512), (T/4,512), (T/8,512), (T/16,512), (T/32,512)). The six-layer design proves highly effective in extracting temporal content. The high-resolution features extracted by the backbone are sequentially up-sampled to each of the six layers. The up-sampling operation involves bi-linear interpolation and simultaneous combination with the high-resolution features. This modified approach simplifies the model while capturing features from more detailed examples. Temporal Action Detection Heads. In TadML, the Temporal Action Detection Heads (TADH) simultaneously predict action categories and temporal boundaries through two branches: the classification branch, which estimates the probability of each class c_i and the regression branch that forecasts the starting and ending time distance $(t_{i,s}, t_{i,e})$. Each time step t further decodes an action instance, including: action start time and action end time denoted by $(T_s = T - T^e$ and $T_e = T + T^s)$ and an action confidence score denoted with c_i. Both branches are constructed with three MLP layers and two LayerNorm layers. An additional activation ReLU layer is included in the classification branch to predict the category ID.

Loss Construction. $\mathcal{L}_{cls}$ is Focal loss that employed as the classification loss, effectively mitigating class imbalance by adjusting the weights of positive and negative samples based on their classification difficulty levels, thereby enhancing overall detection accuracy. $\mathcal{L}_{reg}$ is the regression loss that defined to differentiate the regression of instance time boundaries. For the regression loss, we define it to distinguish the instance time boundary regression. We propose an improvement of GIoU, called $\beta - GIoU$, to constructed the regression loss. In $\beta - GIoU$, the hyper-parameter β is defined to detect shapes sensitively in the error term of the predicted value and position, A is candidate object, the B represents the object ground-truth and C is minimum bounding box area. To accomplish this, the total loss is defined by two super parameters λ_{cls} and λ_{reg} are set up for classification loss and regression loss separately. T_{at} is an indicator function indicating to determine whether there is an action in the time step. while T_{+} represents the total number of positive samples.

$$\begin{aligned}
&\mathcal{L}_{reg} = \mathcal{L}_{\beta-\text{ GloU}} = 1 - IoU + \left(\frac{|C \backslash (A \cup B^{gt})|}{|C|}\right)^{\beta} \\
&\mathcal{L}_{cls}(\{\hat{y}_i\}) = \frac{1}{N}\sum_{i} \ell_{\text{focal}}(\hat{y}_i, y_i) \\
&\mathcal{L} = \sum_{k=1}^{N}(\frac{\mathcal{L}_{cls}}{T}\ell_{cls} + \frac{\lambda_{reg}}{T_{+}}\mathbb{T}_{at}\mathcal{L}_{reg})
\end{aligned} \tag{3}$$

Train and Inference. We implement our network using the pytorch framework. All experiments were run on a workstation equipped a single Tesla P100 GPU, and Intel(R) Xeon(R) CPU (E5-2690 v4 @ 2.90GHz). The models were trained for 80 epochs using Adam with warm-up for training, which is crucial for achieving model convergence and optimal performance. The base learning rate is set to 10^{-5} and the batch size was set to 4. The weights of the loss terms, λ_{cls} and λ_{reg} were both set to 1. The parameter β, in $\beta - Giou$, was set to 3. Input sequences were cropped or padded to the maximum length 2304. During the inference process, only the action predictions from the last lightweight MLP layer are considered, while the full sequences are fed into the model. Our model takes the input video X and outputs $(t_{i,s}, t_{i,e}, c_i)$ for each time step T across all neck levels. The final TAD results are generated by processing action candidates with Non-maximum Suppression (Soft-NMS) to remove highly overlapping instances.

4 Experiments and Results

We evaluated the effectiveness of our proposed approach through benchmark evaluations on THUMOS14[12] and ActivityNet1.3[13], as well as extensive ablation stud-ies to analyze model performance.

4.1 Evaluation

For all datasets, we report the standard mean average precision (mAP) at different temporal intersection over union (tIoU) thresholds, which is widely used to evaluate TAD models. For the THUMOS14, the tIoU threshold is selected from 0.3, 0.4, 0.5, 0.6, 0.7. For ActivityNe-t1.3, the tIoU threshold is 0.5, 0.75, 0.95. We also report the average graph with fine-scale tIoU threshold ([0.5, 0.95] step is 0.05). THUMOS14 is comprised of 413 untrimmed videos spanning 20 action categories. Each video contains 15 action

Table 1. Performance comparison with methods on THUMOS14, measured by mAP at different IoU thresholds, and average mAP in [0.3 : 0.1 : 0.7] on THUMOS14.

Type	Method	RGB stream	0.3	0.4	0.5	0.6	0.7	Avg
Two-stage	CDC[14]	×	40.10	29.40	23.30	13.10	7.90	20.76
	TCN[15]	×	-	33.30	25.60	15.90	9.00	-
	TURN-TAP[16]	×	44.10	34.90	25.60	-	-	-
	R-C3D[17]	×	44.80	35.60	28.90	-	-	-
	MGG[18]	×	53.9	46.8	37.4	29.5	21.3	37.78
	BMN[3]	×	56	47.4	38.8	29.7	20.5	38.48
	DBG[19]	×	57.8	49.4	39.8	30.2	21.7	39.78
	BSN++[20]	×	59.90	45.90	41.30	31.90	22.80	40.36
	GCN[9]	×	63.6	57.8	49.1	-	-	-
	TAL-Net[21]	×	53.2	48.5	42.8	33.8	20.8	39.8
	G-TAD[22]	×	58.7	52.7	44.9	33.6	23.8	42.7
	MR[23]	×	53.9	50.7	45.4	38.0	28.5	43.3
	ContextLoc[24]	×	68.3	63.8	54.3	41.8	26.2	50.88
One-stage	PBRNet[25]	×	58.5	54.6	51.3	41.8	29.5	-
	A2Net[26]	×	58.6	54.1	45.5	32.5	17.2	41.6
	A2Net	×	58.6	54.1	45.5	32.5	17.2	41.6
	G-TAD[27]	✓	57.8	47.2	38.8	-	-	-
	TadTR[1]	×	62.4	57.4	49.2	37.8	26.3	46.6
	TadML	✓	68.78	64.66	56.61	45.40	31.88	53.46
	TadML	×	73.29	69.73	62.53	53.36	39.60	59.70

instances on average, each instance has an average of 8% overlapping with others. The datasets are divided into two subsets: a verification set and a test set. The verification set contains 200 videos and the test set contains 213 videos. Following the standard setup, we use validation sets for training and the testing videos for evaluation. The experimental results on THUMOS14 are shown in table 1. The results are presented in Table 1. Without optical flow input, our method achieves an average mAP of 53.46% ([0.3: 0.1: 0.7]), with a mAP of 56.61% at tIoU=0.5 and a mAP of 31.88% at tIoU=0.7. This result exceeds those of most methods of TAD, even including models with additional optical flow input. This suggests that our model is not only achieves comparable in accuracy but also faster than most methods in practice. This is because our model skips the step of conversion (from raw RGB to optical flow), which is especially time-consuming. In order to further prove the superiority of our model, we also conducted experiments using optical flow input. The results show that our model achieves an average mAP of 59.7% ([0.3 : 0.1 : 0.7]), with a mAP of 62.53% at tIoU=0.5 and a mAP of 39.6% at tIoU=0.7. ActivityNet1.3 is a large-scale action dataset that comprises 200 action classes and ap-

Table 2. Performance comparison with methods on ActivityNetv1.3, measured by mAP at different IoU thresholds, and average mAP in [0.5 : 0.75 : 0.95] on ActivityNetv1.3.

Method	Single-stage	0.5	0.75	0.95	Avg
R-C3D[17]	×	26.80	–	–	–
TAL-Net[21]	×	38.23	18.30	1.30	20.22
BSN[2]	×	56.45	29.96	8.02	30.03
BMN [3]	×	50.07	34.78	8.29	33.85
P-GCN[9]	×	42.90	28.14	2.47	26.99
Contextloc[24]	×	51.24	31.40	2.83	30.59
TadTR+BMN[1]	×	50.51	35.35	8.18	34.55
A2Net[26]	✓	43.55	28.69	3.70	27.75
SSN[28]	✓	43.26	28.70	5.63	28.28
TadTR[1]	✓	49.08	32.58	8.49	32.27
G-TAD[27]	✓	50.36	34.60	9.02	34.09
AFSD[4]	✓	52.4	35.3	6.5	34.4
Ours	✓	53.15	35.75	7.47	34.94

proximately 2k untrimmed videos. And it's total video length exceeds 600 hours. The dataset has been split into three subsets, with 10,024 videos for training, 4,926 videos for validation, and 5,044 videos for testing. In line with the established meth-odology, we trained TadML on the training set and test, the performance on the validation set. The experiment results on ActivityNet v1.3 are presents in table 2. The results are presented in Table 2. Using I3D features, our method achieves an average mAP of 34.94% ([0.5 : 0.05 : 0.95]), this outperforms all previous methods that use the same features by at least 0.6%. This improvement is significant as it is averaged across multiple tIoU thresholds, including those tight ones e.g. 0.95. Furthermore, by employing the pre-training method from TSP, we slightly improve our results, achieving an 36.0% average mAP. Our model thus outperforms the best method with the same features by a small margin. Again, our method largely outperforms TadTR. Our results are only inferior to TCANet—a latest two-stage method using stronger SlowFast features. We conjecture that our method will also benefit from better features. Nevertheless, our simple model clearly demonstrates state-of-the-art results on this challenging dataset.

4.2 Ablation Study

In this section, we conducted several ablation studies on THUMOS14 to further verify the efficacy of our model. Our experiments examined the efficacy of key components and recommended hyper-parameters settings. For all experiments, we kept the evaluation settings constant and only made changes to the corresponding components. Neck layers play a crucial role in temporal action detection, as evidenced by our comparison of different neck layers and RGB streams presented in Table 3. The number of neck layers ranges from 1 to 7, and as the number of neck layers increases, the average mAP also increased from 36.03% to 53.46%, and reach its peak at 6th neck layer. At this point, when the neck layer added again, the performance stars to decline. Furthermore,

Table 3. Study of different number of frozen stages of backbone on THUMOS14 in terms of mAP(%)@tIoU.

Neck Stages	RGB	0.3	0.4	0.5	0.6	0.7
1	×	56.09	49.11	38.35	24.92	12.03
2	×	61.74	55.6	44.81	29.39	14.29
3	×	65.74	60.16	49.91	36.09	19.78
4	×	66.82	62.32	53.82	43.26	28.96
5	×	66.98	62.77	55.42	44.81	31.82
6	×	68.7	64.66	56.61	45.40	31.88
7	×	68.7	64.66	56.61	45.40	-
1	✓	62.7	57.06	46.64	30.73	14.52
2	✓	68.04	62.6	52.37	35.65	18.52
3	✓	70.78	66.1	57.6	44.36	27.41
4	✓	68.7	64.66	56.61	45.40	-
5	✓	73.32	68.91	62.28	52.81	39.04
6	✓	72.79	69.49	62.72	52.29	38.94
7	✓	73.59	69.69	62.79	53.13	40.22

our method also exhibits great performance in two-streams. with the number of layers are set to 7, achieving an average mAP is 59.7%. We have also conducted comparisons using different MLP-Like blocks. While keeping other parameter settings keep unchanged. The results are presented in Table 4, when only RGB streams are used, the average MAP achieved with Waveblock is 51.64%, with MorphMLP it is 52.09%, and with TadML it is 53.46%. To further evaluate our model, we have also included optical flow for comparison. Our model achieved an average MAP as high as 59.70%. These results are shown in Table 5. When the weight of classification and regression loss is set at 1, the best performance is achieved by β, which achieves a MAP of 53.46% when the value of β is 3.

Table 4. Study of three different backbone (WaveMLP, MorphMLP, Mechaincs-MLP) on THUMOS14 in terms of mAP(%)@tIoU.

Neck Stages	RGB	0.3	0.4	0.5	0.6	0.7	Avg
WaveMLP[11]	×	66.87	62.46	54.33	44.00	30.53	51.64
WaveMLP[11]	✓	72.01	68.02	61.51	52.01	38.28	58.36
MorphMLP[10]	×	66.91	62.83	54.93	44.57	31.20	52.09
MorphMLP[10]	✓	72.21	69.12	62.87	52.55	38.47	59.04
Ours	×	68.78	64.66	56.61	45.40	31.88	53.46
Ours	✓	73.29	69.73	62.53	53.36	39.60	59.70

Table 5. Study of different β ($\beta - Giou$) on THUMOS14 in terms of mAP(%)@tIoU.

β	0.3	0.4	0.5	0.6	0.7	Avg
1	66.68	63.60	56.55	43.77	31.24	52.57
2	67.95	63.55	56.68	45.12	31.97	53.05
3	68.78	64.66	56.61	45.40	31.88	53.46
4	67.75	64.03	56.47	43.74	31.34	52.67
5	67.64	63.82	56.43	44.06	31.17	52.63
10	67.44	63.80	56.42	44.15	30.81	52.52

5 Conclusion

In this work, we introduce TadML, an anchor-free one-stage MLP method designed for TAD using RGB stream input. Our method simplifies the traditional TAD pipe-line by eliminating the need for manual conversion of optical flow data. Additionally, we propose $\beta - GIoU$ for the framework. To the best of our knowledge, TadML is the first MLP-like model suitable for TAD. We leverage Newtonian Mechanics to address the token mixing problem. TadML showcases the potential of MLP-like methods in downstream visual tasks, surpassing many recent methods (including those using optical flow input) and offering twice the inference speed of BMN. Due to its independence from optical flow conversion, our method holds promise for practical applications in the field of TAD. Moreover, it achieves impressive performance when both RGB and flow data are utilized. Our goal is to promote the development of efficient models for temporal action detection and facilitate their adoption in industrial set-tings. The source code is available at https://github.com/BonedDeng/TadML.

Acknowledgements Thanks for the support of supervisor Zhao in Guangxi University and the internship opportunities provided by supervisor Liu in the Institute of Automation, Chinese Academy of Sciences.

References

1. Liu, X., Bai, S., Bai, X. An Empirical Study of End-to-End Temporal Action Detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 658–666. IEEE, Long Beach, CA, USA (2019).
2. Lin, T., Zhao, X., Su, H., Wang, C., Yang, M. BSN: Boundary Sensitive Network for Temporal Action Proposal Generation. In: Ferrari, V., Hebert, M., Sminchisescu, C., and Weiss, Y. (eds.) Computer Vision – ECCV 2018. pp. 3–21. Springer International Publishing, Cham (2018).
3. 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), pp. 3888–3897. IEEE, Seoul, Korea (South) (2019).
4. 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), pp. 3319–3328. IEEE, Nashville, TN, USA (2021).
5. Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 658–666. IEEE, Long Beach, CA, USA (2019).
6. Uijlings, J.R.R., Duta, I.C., Rostamzadeh, N., Sebe, N. Realtime Video Classification using Dense HOF/HOG. In: Proceedings of International Conference on Multimedia Retrieval, pp. 145–152. Association for Computing Machinery, New York, NY, USA (2014).
7. Graves, A. Long Short-Term Memory. In: Graves, A. (ed.) Supervised Sequence Labelling with Recurrent Neural Networks, pp. 37–45. Springer, Berlin, Heidelberg (2012).
8. Dai, C., Wei, Y., Xu, Z., Chen, M., Liu, Y., Fan, J. An Investigation into Performance Factors of Two-Stream I3D Networks. In: 2021 26th International Conference on Automation and Computing (ICAC), pp. 1–6 (2021).
9. Zeng, R., Huang, W., Tan, M., Rong, Y., Zhao, P., Huang, J., Gan, C. Graph Convolutional Networks for Temporal Action Localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 4560–4570. IEEE, Seoul, Korea (South) (2019).
10. Zhang, D.J., Li, K., Wang, Y., Chen, Y., Chandra, S., Qiao, Y., Liu, L., Shou, M.Z. MorphMLP: An Efficient MLP-Like Backbone for Spatial-Temporal Representation Learning. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., and Hassner, T. (eds.) Computer Vision – ECCV 2022, pp. 230–248. Springer Nature Switzerland, Cham (2022).
11. Tang, Y., Han, K., Guo, J., Xu, C., Li, Y., Xu, C., Wang, Y. An Image Patch is a Wave: Phase-Aware Vision MLP. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10925–10934. IEEE, New Orleans, LA, USA (2022).
12. Idrees, H., Zamir, A.R., Jiang, Y.-G., Gorban, A., Laptev, I., Sukthankar, R., Shah, M. The THUMOS challenge on action recognition for videos “in the wild.” Computer Vision and Image Understanding. 155, 1–23 (2017).

13. 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), pp. 961–970 (2015).
14. Yang, K., Qiao, P., Li, D., Lv, S., Dou, Y. Exploring temporal preservation networks for precise temporal action localization. In: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, pp. 7477–7484. AAAI Press, New Orleans, Louisiana, USA (2018).
15. Dai, X., Singh, B., Zhang, G., Davis, L.S., Chen, Y.Q. Temporal Context Network for Activity Localization in Videos. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5727–5736. IEEE, Venice (2017).
16. 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), pp. 3648–3656. IEEE, Venice (2017).
17. Xu, H., Das, A., Saenko, K. R-C3D: Region Convolutional 3D Network for Temporal Activity Detection. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 5794–5803. IEEE, Venice (2017).
18. Liu, Y., Ma, L., Zhang, Y., Liu, W., Chang, S.-F. Multi-Granularity Generator for Temporal Action Proposal. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3599–3608. IEEE, Long Beach, CA, USA (2019).
19. Song, Q., Zhou, Y., Hu, M., Liu, C. Faster learning of temporal action proposal via sparse multilevel boundary generator. Multimed Tools Appl. (2023).
20. Sooksatra, S., Watcharapinchai, S. A Comprehensive Review on Temporal-Action Proposal Generation. J Imaging. 8, 207 (2022).
21. 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, pp. 1130–1139. IEEE, Salt Lake City, UT (2018).
22. Xu, M., Zhao, C., Rojas, D.S., Thabet, A., Ghanem, B. G-TAD: Sub-Graph Localization for Temporal Action Detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10153–10162. IEEE, Seattle, WA, USA (2020).
23. Zhao, P., Xie, L., Ju, C., Zhang, Y., Wang, Y., Tian, Q. Bottom-Up Temporal Action Localization with Mutual Regularization. In: Vedaldi, A., Bischof, H., Brox, T., and Frahm, J.-M. (eds.) Computer Vision – ECCV 2020. pp. 539–555. Springer International Publishing, Cham (2020).
24. Zhu, Z., Tang, W., Wang, L., Zheng, N., Hua, G. Enriching Local and Global Contexts for Temporal Action Localization. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 13496–13505. IEEE, Montreal, QC, Canada (2021).
25. Dai, P., Li, Z., Zhang, Y., Liu, S., Zeng, B. PBR-Net: Imitating Physically Based Rendering Using Deep Neural Network. IEEE Transactions on Image Processing. 29, 5980–5992 (2020).
26. Yang, L., Peng, H., Zhang, D., Fu, J., Han, J. Revisiting Anchor Mechanisms for Temporal Action Localization. IEEE Transactions on Image Processing. 29, 8535–8548 (2020).
27. Long, F., Yao, T., Qiu, Z., Tian, X., Luo, J., Mei, T. Gaussian Temporal Awareness Networks for Action Localization. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 344–353 (2019).
28. Zhao, Y., Xiong, Y., Wang, L., Wu, Z., Tang, X., Lin, D. Temporal Action Detection with Structured Segment Networks. Int J Comput Vis. 128, 74–95 (2020).