Supplementary: A Dual Filtering Auxiliary Transformer for Efficient Online Action Detection in Streaming Videos

Shicheng Jing[1,2] and Liping Xie[1,2(✉)]

[1] School of Automation, Southeast University, Nanjing, 210096, China
[2] Key Laboratory of Measurement and Control of Complex Systems of Engineering, Ministry of Education, Nanjing, 210096, China
scjing10@gmail.com, lpxie@seu.edu.cn

In this supplementary material, §Supp.1 contains more details of the structural design and code implementation, §Supp.2 reports analysis and discussions on the additional results, and §Supp.3 includes more thorough and detailed ablation experiments for DFAformer. Finally, §Supp.4 provides more intuitive and expressive charts to enable readers to understand our proposed DFAformer qualitatively.

1 Implementation Details

In this section, we elaborate on the structural design aspects of DFAformer not covered in the main manuscript due to space restrictions. Further, we elucidate additional details regarding feature encoding and hyperparameter configurations.

1.1 Structural design

Our proposed Equalized Gated Unit (EGU), primarily inspired by GateHUB [2], aims to filter element-level redundant information among video frames. EGU adopts an ingenious symmetric structure called EqualizedGate, to achieve a balance of suppressed and enhanced video frames. To make DFAformer reproduction easier, here we provide a diagram of the internal structure of EGU, as shown in Fig. 1.

1.2 Feature Encoding.

For THUMOS14 and TVSeries, following [2,14], we make decisions at the 6-length-size chunk level, that is, performance is measured every 0.25 seconds. As for feature extractor, following [2,14], we use the TSN [11] models implemented in an out-of-the-box toolbox [3]. Specifically, the features are extracted by one visual model with the ResNet-50 [8] architecture from the central frame of each chunk and one motion model with the same architecture from the stacked optical flow fields between 6 consecutive frames. The visual and motion features are concatenated along the channel dimension as the final feature $\mathbf{f}$.

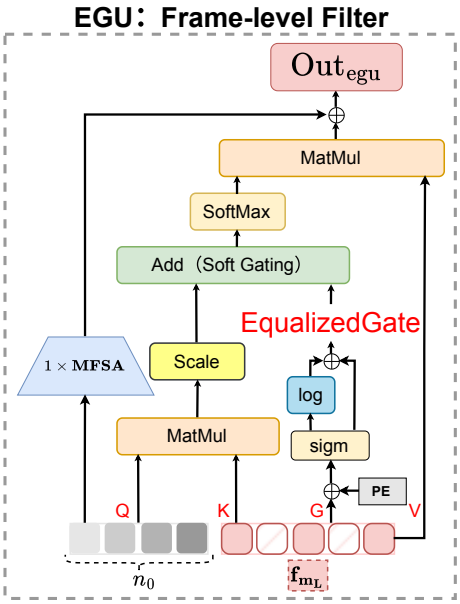


Fig. 1. Internal structure of Equalized Gated Unit (EGU).

1.3 Hyperparameter configurations

In addition to the settings mentioned in the paper, we set $n_0 = n_1 = m_S \times$ FPS, where FPS denotes the frame rate of the chunk-level videos. Due to the power of the two-stage filtering mechanism, we only use the EGU and RFU once rather than stacking them multiple times. Moreover, we set the number of decoder layers to 2 and the number of JSU decoder layers to 1.

2 Additional Results

2.1 Additional comparison on ActivityNet pretrained features

As shown in Table 1, we further compare DFAformer with other state-of-the-art (SOTA) methods on THUMOS14 using two-stream features from TSN pretrained on ActivityNet [1]. From the table, we observe that DFAformer achieves much higher performance, outperforming GateHUB [2] by about 0.3% in mAP on THUMOS14. Table 2 shows the consistent results on TVSeries as THUMOS14. Once again, DFAformer surpasses all existing methods. We note that the performance improvement on features pretrained on ActivityNet is less considerable than on features pretrained on Kinetics. We consider this because features pretrained on Kinetics are extracted from a model trained on a larger dataset, resulting in more discriminative features. Nevertheless, DFAformer outperforms previous SOTA methods by a large margin on both features, which proves the superiority and effectiveness of DFAformer.

Table 1. Comparison between DFAformer and other SOTA methods on the THUMOS14 dataset [9] in terms of mAP (%), using ActivityNet pretrained features.

Method	Reference	Architecture	mAP(%)
RED [7]	BMVC'17	RNN	45.3
TRN [13]	ICCV'19		47.2
IDN [5]	CVPR'20		50.0
FATS [10]	PR'21		51.6
TFN [6]	PR'21		55.7
OadTR [12]	ICCV'21	Transformer	58.3
Colar [15]	CVPR'22		59.4
LSTR [14]	NeurIPS'21		65.3
GateHUB [2]	CVPR'22		69.1
DFAformer	-	Transformer	**69.4**

Table 2. Comparison between DFAformer and other SOTA methods on the TVSeries dataset [4] in terms of cmAP (%), using ActivityNet pretrained features.

Method	Reference	Architecture	mAP(%)
RED [7]	BMVC'17	RNN	79.2
FATS [10]	PR'21		81.7
TRN [13]	ICCV'19		83.7
IDN [5]	CVPR'20		84.7
TFN [6]	PR'21		85.0
OadTR [12]	ICCV'21	Transformer	85.4
Colar [15]	CVPR'22		86.0
LSTR [14]	NeurIPS'21		88.1
GateHUB [2]	CVPR'22		88.4
DFAformer	-	Transformer	**88.6**

2.2 Top 10 action categories where OAD performance improves the most

Fig. 2 shows the top 10 action categories where OAD performance improves the most by DFAformer. As can be seen, DFAformer has improved performance by a large margin in several action categories. And it is interesting to note that using the proposed DFAformer, detection performance improves both for relatively simple actions such as *ThrowDiscus* and for relatively complex actions such as *Billiards*. *ThrowDiscus* has a large range of movement and long duration. Conversely, *Billiards* has a small range of movement and short duration. This suggests that the benefits of the proposed DFAformer are applicable to various types of actions in the task of OAD.

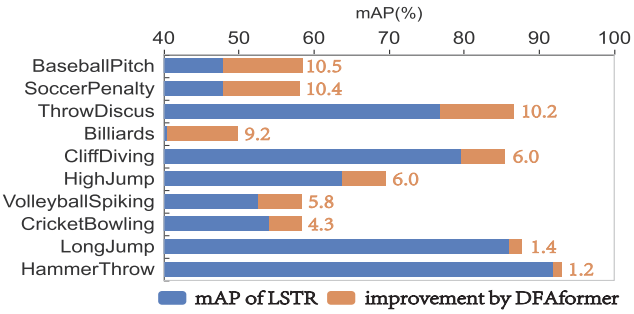


Fig. 2. Top 10 action categories where OAD performance improves the most by DFAformer. The blue line indicates the mAP of the LSTR [14] and the orange line indicates the increment of our proposed DFAformer on the LSTR. The orange numbers in the graph denote the exact size of the increment in percentage. (Best viewed in color.)

3 Further Ablations

For a comprehensive analysis of the DFAformer, we report more thorough and detailed ablation results in this section.

Ablations about channel number of DFAformer. We carry out ablations on the channel number *d_model* of DFAformer. As shown in Fig. 3a, we observe that 768 is a proper choice for channel number. When the *d_model* is relatively small(*e.g.*, 512), the model capacity is limited and the performance is relatively poor, resulting in underfitting. As the *d_model* increases, the model capacity becomes larger, and the performance becomes stronger. Moreover, further increasing the *d_model* may lead to overfitting after a specific limit. As to why our channel number is relatively small compared to other models, it is mainly because our auxiliary task assume part of the model capacity.

Ablations about head number. Multi-head attention mechanism allows the network to learn different patterns. We conduct experiments to study how the number of heads affect OAD performance. In Fig. 3b, we empirically set the number of heads to 8, which is sufficient to capture enough patterns for OAD.

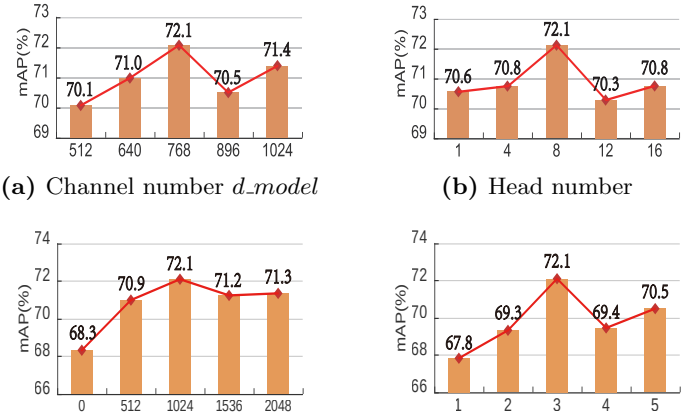
Ablations about long-term memory length. We analyze the effect of different lengths of long-term m_L memory. Specifically, when there is no long-term memory (m_L=0), the performance drops sharply. As the length of long-term memory increases, performance increases then decreases, probably due to the introduction of too many irrelevant video frames. From Fig. 3c, we find that m_L=1024 is optimal for OAD.

Ablations about short-term memory length. As before, we conduct experiments to explore the impact of different lengths of short-term m_S memory. As illustrated in Fig. 3d, we observe that m_S=3 achieves the best results. When the length of short-term memory is too short (*e.g.*, m_S=1), there are fewer supervised signals available for model training and the model is underfitting, resulting in low performance. When the length of short-term memory is too long (*e.g.*,

m_S=5), the model can be trained with more supervised signals and the model is overfitting, also resulting in low performance.

Ablations about future memory length. To further investigate the effect of the lengths of future memory in JSU module, we test $m_F \in \{0.5, 1, 1.5, 2, 2.5\}$. In Fig. 3e, the future memory changes in length does not cause much fluctuation in terms of performance, reaching an optimum point at m_F=2. It is a good indication of the effectiveness of the JSU module on OAD performance improvement. An excessively long m_F introduces noise leading to performance degradation.

Ablations about query number. As in the previous work, we use n_0 learnable tokens to compress the long-term memory with a length of m_L into a compressed memory with a length of n_0. For the choice of n_0, we intuitively consider that it should match the number of video frames in the short-term memory, which allows the model to consider the corresponding number of feature representations in the short-term memory in advance at the stage of compressing the long-term memory. In addition, the results in Fig. 3f demonstrate our conjecture well, achieving the best results at $n_0 = m_S \times$ FPS.



(a) Channel number *d_model* **(b)** Head number

(c) Long-term memory length m_L **(d)** Short-term memory length m_S

(e) Future memory length m_F **(f)** Query number n_0

Fig. 3. More thorough and detailed ablation experiments for DFAformer, measured by mAP(%) on THUMOS14 dataset.

4 Qualitative Observations

This section presents a collection of visual examples to enhance the discussion in the main manuscript. The objective is twofold: to demonstrate the capability of DFAformer in accurately categorizing ongoing actions and to provide readers with a quick and intuitive understanding of the online action detection domain from a qualitative perspective.

In particular, the x-axis signifies the ongoing frame in the streaming video, while the y-axis represents the confidence level associated with the prediction of the correct action. We juxtapose DFAformer with the SOTA method, LSTR [14]. In Fig. 4, the solid red line denotes DFAformer's confidence in accurate class prediction, the dashed green line signifies the equivalent measure for LSTR, and the solid black line depicts DFAformer's confidence in predicting the background class. The confidence associated with the background class reflects DFAformer's capability to resist misclassification of the ongoing action.

As depicted in Fig. 4, to facilitate a more intuitive visual interpretation, we have incorporated sample video frames in the first row. Here, the grey and blue boxes represent the background and action classes, respectively. We can observe that DFAformer detects actions at an earlier stage of the action compared to LSTR [14], possibly thanks to our design of the JSU module. As the JSU module correlates the past with the future, the model learns subtle action discrimination information at an early stage of the action. Another noteworthy point is that the DFAformer can detect actions with a higher confidence level and maintain that high level for the duration of the action. It is a solid testament to the reliability and effectiveness of the structural design of DFAformer.

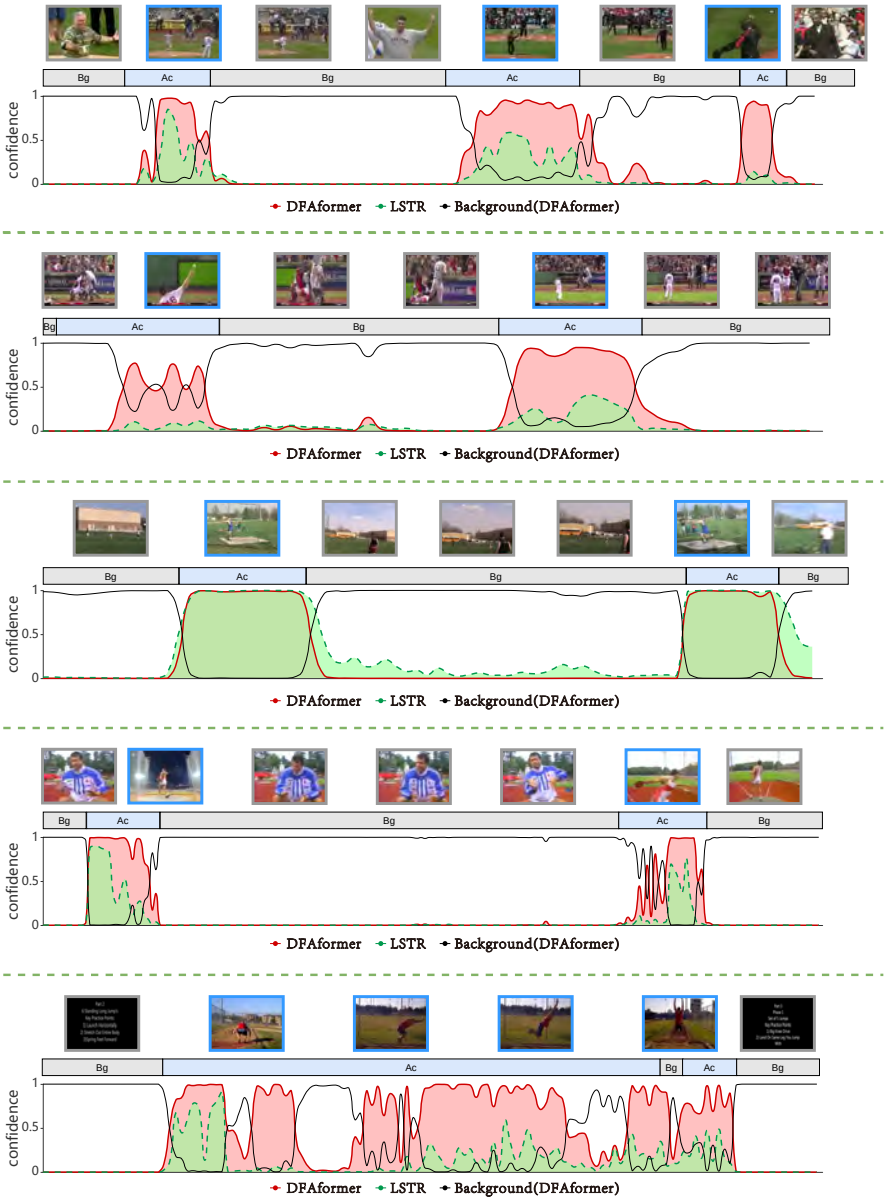


Fig. 4. Visualization of online action detection results for streaming video sequences. The x-axis represents the ongoing frame in the streaming video, and the y-axis represents the confidence level of predicting the correct action. The solid red line represents the confidence that DFAformer predicts the correct class, the dashed green line represents the confidence that LSTR predicts the correct class, and the solid black line represents the confidence that DFAformer predicts the background class. 'Bg' and 'Ac' denote background and action, respectively.

References

1. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: a large-scale video benchmark for human activity understanding. In: CVPR. pp. 961–970 (2015)
2. Chen, J., Mittal, G., Yu, Y., Kong, Y., Chen, M.: Gatehub: gated history unit with background suppression for online action detection. In: CVPR. pp. 19925–19934 (2022)
3. Contributors, M.: Openmmlab's next generation video understanding toolbox and benchmark. https://github.com/open-mmlab/mmaction2 (2020)
4. De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., Tuytelaars, T.: Online action detection. In: ECCV. pp. 269–284 (2016)
5. Eun, H., Moon, J., Park, J., Jung, C., Kim, C.: Learning to discriminate information for online action detection. In: CVPR. pp. 809–818 (2020)
6. Eun, H., Moon, J., Park, J., Jung, C., Kim, C.: Temporal filtering networks for online action detection. PR **111**, pp. 107695 (2021)
7. Gao, J., Yang, Z., Nevatia, R.: Red: Reinforced encoder-decoder networks for action anticipation. In: BMVC (2017)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
9. Jiang, Y.G., Liu, J., Roshan Zamir, A., Toderici, G., Laptev, I., Shah, M., Sukthankar, R.: THUMOS challenge: action recognition with a large number of classes. http://crcv.ucf.edu/THUMOS14/ (2014)
10. Kim, Y.H., Nam, S., Kim, S.J.: Temporally smooth online action detection using cycle-consistent future anticipation. PR **116**, pp. 107954 (2021)
11. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Van Gool, L.: Temporal segment networks: towards good practices for deep action recognition. In: ECCV. pp. 20–36 (2016)
12. Wang, X., Zhang, S., Qing, Z., Shao, Y., Zuo, Z., Gao, C., Sang, N.: Oadtr: Online action detection with transformers. In: ICCV. pp. 7565–7575 (2021)
13. Xu, M., Gao, M., Chen, Y.T., Davis, L.S., Crandall, D.J.: Temporal recurrent networks for online action detection. In: ICCV. pp. 5532–5541 (2019)
14. Xu, M., Xiong, Y., Chen, H., Li, X., Xia, W., Tu, Z., Soatto, S.: Long short-term transformer for online action detection. In: NeurIPS. pp. 1086–1099 (2021)
15. Yang, L., Han, J., Zhang, D.: Colar: Effective and efficient online action detection by consulting exemplars. In: CVPR. pp. 3160–3169 (2022)