Supplementary: E2E-LOAD: End-to-End Long-form Online Action Detection

Shuqiang Cao[1*] Weixin Luo[2*] Bairui Wang[2] Wei Zhang[1†] Lin Ma[2†]
[1]School of Control Science and Engineering, Shandong University
[2]Meituan
sqiangcao@mail.sdu.edu.cn, luowx@shanghaitech.edu.cn, davidzhang@sdu.edu.cn
{bairuiwong, forest.linma}@gmail.com

The document first presents the details of streaming reasoning so that readers can better comprehend the OAD task and the design of the proposed E2E-LOAD. We then demonstrate additional qualitative and quantitative analyses of the proposed E2E-LOAD. Additionally, we introduce details of the datasets and implementations excluded in the main paper because of limited space.

1. Preliminaries

To help readers better understand the critical challenges of OAD and the proposed model, we provide a comprehensive analysis of the streaming reasoning and explain the underlying design principles of our proposed solution.

Streaming Reasoning. Unlike offline video tasks that usually involve videos of fixed duration and can access all frames at once, streaming video requires processing a sequence of frames that arrive one at a time and whose overall length is unknown. As time goes on, the length of the sequence gradually increases, making it challenging to model the long-term interactions efficiently. In addition, all frames are the same at adjacent moments in a streaming video except for the latest one. Limiting repetitive modeling of these frames is critical as the model in online scenarios necessitates a real-time response. To summarize, traditional offline video approaches do not address these pain points well, i.e., long video understanding and efficient inference, which inspired us to design a streaming video processing framework for OAD and make end-to-end training feasible.

2. Online Inference

In this section, we present Algorithm 1, which outlines the online inference algorithm of E2E-LOAD to aid the reader in understanding how E2E-LOAD achieves efficiency. To simplify matters, we exclude the long-term history, which undergoes a similar inference process as SM. We highlight the Stream Buffer $\mathbf{M}$ and Efficient Inference (EI) technique, which are essential for efficient inference of the model.

Algorithm 1: Online Inference of E2E-LOAD.

Input: Set of video frames $\mathcal{V}$.
Output: Per-frame predictions $\mathcal{L}$.

Initialize Stream Buffer $\mathbf{M} = \mathbf{0}^{T_S \times D}$;
for f *in* $\mathcal{V}$ **do**
  $\mathbf{X}^0_{T_S} = \mathrm{Attn_S}\left(f\right)$;
  $\mathbf{M} = \mathrm{Cat}(\mathbf{M}_{1:-1}, \mathbf{X}^0_{T_S})$;
  $\mathbf{X}^0_{[1:T_S]} = \mathbf{M}$;
  for l *in* SM **do**
    if *EI* **then**
      $\mathbf{X}^{l+1}_{T_S} = \mathrm{Attn_{ST}}\left(\mathbf{X}^l_{T_S}, \mathbf{X}^l_{[1:T_S]})\right)$;
      $\mathbf{X}^{l+1}_{[1:T_S]} = \mathrm{Cat}\left(\mathbf{X}^{l+1}_{[2:T_S]}, \mathbf{X}^{l+1}_{T_S}\right)$;
    else
      $\mathbf{X}^{l+1}_{[1:T_S]} = \mathrm{Attn_{ST}}\left(\mathbf{X}^l_{[1:T_S]}, \mathbf{X}^l_{[1:T_S]}\right)$;
  $P_f = \mathrm{Classifier}(\mathbf{X}^{-1}_{T_S})$;
  Append P_f into $\mathcal{L}$;
return $\mathcal{L}$

3. Dataset

THUMOS'14 [4] contains 413 untrimmed videos about sports. Following the previous work [9, 2], we implement training on 200 videos from the validation set and testing on 213 videos from the test set. These videos cover 21 classes, including background and 20 action classes. Each video includes 15.7 actions on average, and 71% of the video frames are background.

TVSeries [3] annotated 30 realistic actions (e.g., run and smoke) on 27 episodes from 6 popular TV series, with a total of 16 hours. The model is trained on 20 videos and tested on the rest seven videos. This dataset is challenging

*Authors contributed equally.
†Corresponding author.

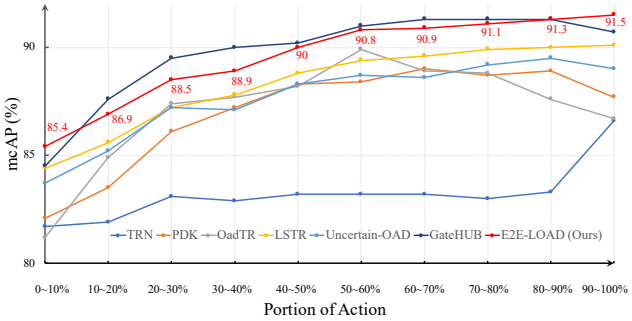


Figure 1: **Online Action Detection Performance on Different Portions of Actions in mcAP (%).** The model can make more reliable predictions as more and more content of the ongoing action is observed. E2E-LOAD (red) outperforms all the approaches at the starting and ending stages and achieves comparable results in the middle stage with GateHUB.

due to the diversity of viewpoints, actions, and background frames.

HDD [7] collected 137 human driving sessions by an instrumented vehicle. This dataset describes 11 kinds of driving actions. Each session contains RGB images and non-visual sensor data. Following the previous settings [8], we take the RGB data as inputs and use 100 videos for training and 37 videos for testing.

4. Implementations Details

Hyperparameter Settings. For THUMOS'14 and TVSeries, we follow the previous works [9] by extracting video frames at 24 FPS and down-sampling the video into a series of chunks at 4 FPS as inputs. Each chunk contains $\tau = 6$ consecutive frames. Following TRN [8] for HDD, we extracted video frames at 3 FPS, and each chunk contains one frame. The E2E-LOAD takes $T_S = 32$ current chunks as inputs and $T_L = 32$ (training) or 128 (testing) long-term historical chunks as input. The chunks for long-term history are sampled at 1 FPS, which covers longer than the current window. As for the detailed structure of E2E-LOAD, we set the patch size $t \times h \times w$ to $3 \times 4 \times 4$ for Chunk Embedding (CE). For the Stream Buffer (SB), we configure the depth L_{SB} to 5 and set the buffer size T to $T_S + T_L$. We employ 2D convolutional layers with a kernel size of 3×3, and the stride of the convolution is set to 1, except the 1^{st} and 3^{rd} attention layers, where the stride is 2. For the Short-term Modeling (SM), we configure the depth L_{SM} to 11. We employ 3D convolutional layers (Convs) with a kernel size of $1 \times 3 \times 3$, and the spatial stride of the convolution is set to 1, except after the 8^{th} attention layer, where it becomes 2. All the temporal strides are consistently assigned to 1. For Long-term Compression (LC), we stack

Benchmark	Pretraining	mAP/mcAP (%)
THUMOS'14	✔	72.4
THUMOS'14	✘	17.4
TVSeries	✔	90.3
TVSeries	✘	65.0

Table 1: **Ablation Study for Pre-training on Two Benchmark Datasets.** The model with pretraining can easily achieve great performance, while the model without pretraining suffers from convergence.

$L_{LC} = 4$ spatial-temporal layers with a temporal stride of $2, 2, 1, 1$, respectively. For Long-Short-term Fusion (LSF), we adopt cross-attention and perform long-short-term fusion at the 5^{th} layer of the SM module.

Training. Following the previous settings [9, 2], we pretrain the proposed E2E-LOAD on Kinetics-400 [1]. We follow the augmentation techniques of MViTv2 [5] and take 224×224 crops as input. Also, we use AdamW [6] with a weight decay of 5e-4 and a learning rate of 1e-4. The model is trained with a warm-up learning rate of 1e-6 for 15 epochs and decays according to the cosine function. We train the E2E-LOAD with a batch size of 16 for 50 epochs.

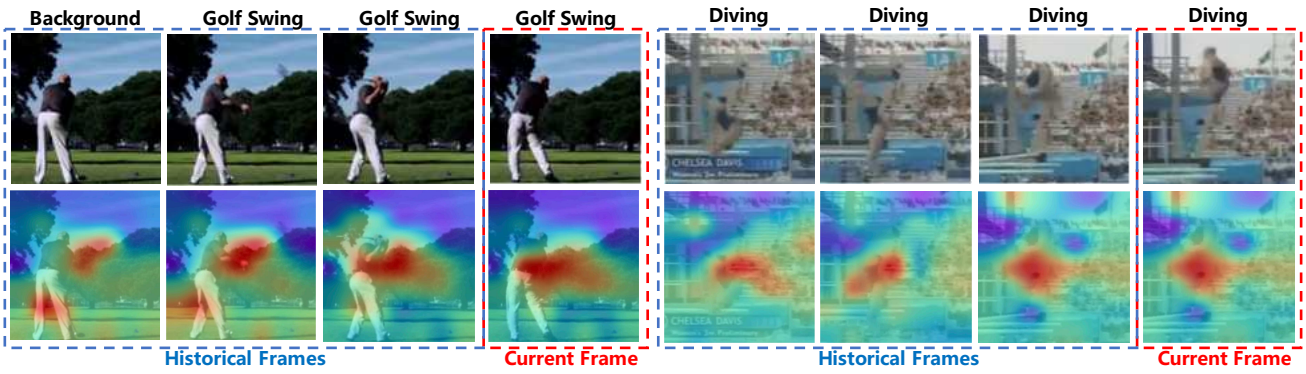
Evaluation. Following the previous works [9, 8, 2], we adopt mean Average Precision (mAP) to evaluate THUMOS'14 and HDD and mean calibrate average precision (mcAP) [3] to evaluate TVSeries. We evaluate the methods on THUMOS'14 and TVSeries at 4 FPS and HDD at 3 FPS. During inference, we scale the shorter side of the frame to 256 and take the 224×224 center crop as inputs, while the previous works [9, 2] adopts multi-crops inference, which will bring extra computational costs.

5. Additional Quantitative Analysis

In this section, we first evaluate the E2E-LOAD's performance on different action portions on TVSeries. Then we experiment with the effect of pretraining parameters on model performance.

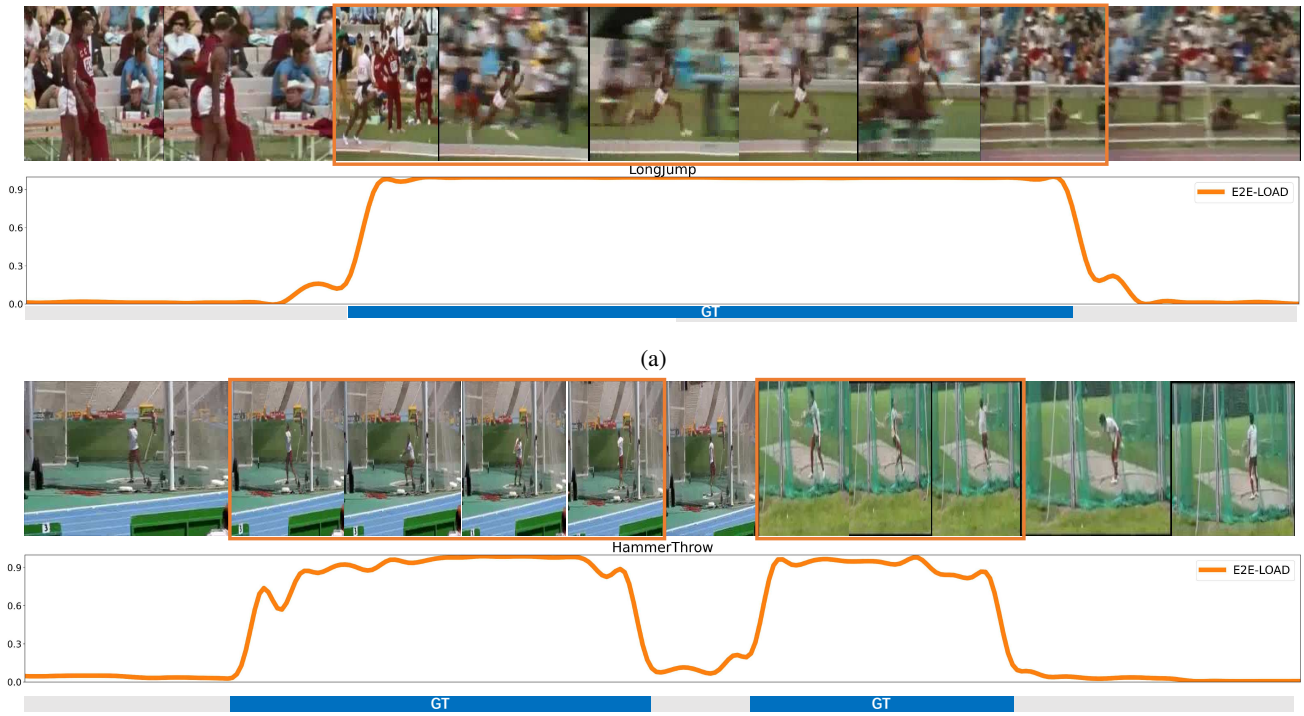
5.1. Evaluation for Different Portions of Action

Following the previous works [8, 9, 2], we report performance on portions of actions on TVSeries, which means each action process is equally divided into 10 parts and the evaluation is conducted on each part (*e.g.* $0\% - 10\%$) separately. This evaluation metric reflects the recognition performance of the model at different action stages, *i.e.* starting, middle, and ending stages. It can be observed from Table 1, the proposed E2E-LOAD outperforms all the approaches at the starting ($0\% - 10\%$) and ending ($90\% - 100\%$) stages and achieve comparable results on the middle stages with the state-of-art methods. Based on this observation, we can conclude that our E2E-LOAD



(a) (b)

Figure 2: **Spatial-temporal Attention Visualizations on THUMOS'14.** The highlight degree of a region represents the value of the corresponding attention score. We illustrate the attention distributions of the current frame (red dotted box) on the historical frames (blue dotted box).



(a)

(b)

Figure 3: **Action Scores Visualization.** We visualize the action scores of two cases (*i.e.* (a) Long Jump and (b) Hammer Throw) on THUMOS'14. The orange curves represent the confidence of the current frames at each moment, and the blue bar represents the period of the ground truth.

can better perceive action boundaries and gain better recognition performance. The detection of the beginning of the action is critical, especially in the actual scene, as it helps to detect the relevant actions in time. On the other hand, our approach can give more reliable predictions as more content of the ongoing action is observed, while the other approaches will lead to a slight performance drop at the late stage of the action. The spatial-temporal attention can well explore the action clues from the observed history. In summary, we claim the superiority of the E2E-LOAD based on

the two observations.

5.2. Train from Scratch

We pre-trained our method on Kinetics-400 [1] for a fair comparison with previous work. We ablate the impact of the pre-training based on *Baseline+LC+LSF+EI*. We can observe from Table 1 the performance drastically drops without pre-training. We believe the video transformer with huge parameters cannot converge well on small datasets, *i.e*. THUMOS'14 and TVSeries.

6. Additional Qualitative Analysis

In this section, we conduct a qualitative analysis of E2E-LOAD by visualizing the spatial-temporal attention distributions and the classification scores on THUMOS'14 to validate the effectiveness of the proposed E2E-LOAD.

6.1. Visualization of Spatial-temporal Attention

We visualize the distributions of the spatial-temporal attention in Figure 2. Concretely, the incoming frame attends all the spatial-temporal regions and assigns weights for more discriminative aggregation. Since the [CLS] token related to the current frame is adopted for classification, we plot its attention on each spatial-temporal region. As shown in Figure 2a and Figure 2b, the highlight degree of the region reflects the value of its attention score at the current moment. We can see that the subjects are correlated at different moments, and the noisy regions are suppressed, which results in more reliable reasoning as expected.

6.2. Visualization of the Action Scores

In addition to the visualization of the spatial-temporal attention distribution, we also visualize the action scores of the current frames in Figure 3. We can see from Figure 3a the action score curve is steep at the action boundaries, which means that our method can detect the start and the end of actions very quickly in real scenarios. This property is consistent with the discussion on the performance of different action portions in section 5.1. We believe spatial-temporal attention can effectively distinguish between the action and the complicated background. Besides, from Figure 3b, we can observe that our model is still able to give reliable predictions when short actions occur continuously and rapidly. These examples demonstrate the effectiveness of our approach.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[2] Junwen Chen, Gaurav Mittal, Ye Yu, Yu Kong, and Mei Chen. Gatehub: Gated history unit with background suppression for online action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19925–19934, 2022.

[3] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *European Conference on Computer Vision*, pages 269–284. Springer, 2016.

[4] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[5] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.

[6] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018.

[7] Vasili Ramanishka, Yi-Ting Chen, Teruhisa Misu, and Kate Saenko. Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[8] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S Davis, and David J Crandall. Temporal recurrent networks for online action detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5532–5541, 2019.

[9] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, 34, 2021.