Efficient Video Action Detection with Token Dropout and Context Refinement
Supplementary Material

Lei Chen[1] Zhan Tong[2] Yibing Song[3] Gangshan Wu[1] Limin Wang[1,4*]
[1]State Key Laboratory for Novel Software Technology, Nanjing University
[2]Ant Group [3]AI[3] Institute, Fudan University [4]Shanghai AI Lab
leichen1997@outlook.com zhantong.2023@gmail.com yibingsong.cv@gmail.com
gswu@nju.edu.cn lmwang@nju.edu.cn

In this supplementary material, we provide more details of EVAD from the following aspects:

- The detailed architecture illustration is in § A.
- The implementation details are in § B.
- Additional experimental results are in § C.
- Results analysis and visualization are in § D.

A. Architectures

We present the architectural details of EVAD based on 16-frame vanilla ViT-Base and ViT-Large backbones used in the experiments and the corresponding output sizes of each stage, as shown in Table 1. The *token pruning* with keep rate ρ is executed three times in total, following each stage. The intermediate keyframe features from backbone stages 1-4 are used for actor localization via EVAD localization branch, and the spatiotemporal features from stage 4 are used for actor feature refinement and final action classification prediction via EVAD classification branch. The computational costs of both models are reduced by 40% (ViT-B) and 42% (ViT-L) at ρ=0.7, respectively.

B. Implementation Details

Query-based actor localization head. The localization head initializes n learnable proposal boxes and corresponding proposal features, which can be optimized together in the network. Then, the head utilizes RoIAlign operations to extract RoI features for each box. Next, a sequentially-stacked Dynamic Instance Interactive head [9, 1] is conducted on each RoI feature to generate the final predictions conditioned on proposal features. Finally, two task-specific prediction layers are used to produce the prediction boxes and corresponding actor confidence scores.

Configurations. By default, the token pruning module is incorporated into the 4^{th}, 7^{th}, and 10^{th} layer of ViT-B (with 12 layers in total) and incorporated into the 7^{th}, 13^{th}, and 19^{th} layer of ViT-L (with 24 layers in total). We specify the number of query n as 100 and the dimension of query as 256, and we use 6 dynamic instance interactive modules in actor localization branch, same as in [1]. For action classification branch, the dimension of context refinement decoder is 384 (ViT-B) and 512 (ViT-L), and the depth of decoders for these two backbones is 6 and 12. The backbone is initialized with Kinetics-pretrained weights from *VideoMAE* and other newly added layers are initialized with Xavier [4].

Losses and optimizers. For all experiments, we simply follow those in the original paper of WOO. Specifically, the loss function includes the set prediction loss and the action classification loss, where the set prediction loss consists of the cross-entropy loss over two classes (person and background), L1 loss and GIoU loss on the box. The action classification loss is denoted by the binary cross-entropy loss. We set the loss weight as λ_{ce}=2, λ_{L1}=5, λ_{GIoU}=2, λ_{bce}=12. We use AdamW [6] with weight decay 1×10^{-4} as the optimizer and apply intermediate supervision on the output of each layer in localization and classification branches.

Training and inference recipes. Following the $1\times$ training schedule in [9, 1], we train all models for 12 epochs with an initial learning rate of 2.5×10^{-5} and reduce the learning rate by $10\times$ at epoch 5 and 8. We apply a linear warm-up from 2.5×10^{-6} to 2.5×10^{-5} at the first two epochs. The mini-batch consists of 16 video clips and all models are trained with 8 GPUs (2 clips per device), and for the model with ViT-L backbone, both the mini-batch and the learning rate are reduced by 1/2 of the original. For ablation studies on AVA, we set the shortest side of each video frame to 224 for efficient exploration, and for comparisons to the state-of-the-art methods, we set the shortest side to 288 unless otherwise specified. For the experiments on JHMDB, we perform random scaling to each video frame input and

[*] Corresponding author.

Stage	Vision Transformer (Base)	Vision Transformer (Large)	Output Sizes
data	stride 4×1×1		3×16×224×224
cube	2×16×16, 768 stride 2×16×16	2×16×16, 1024 stride 2×16×16	C×8×14×14
stage1	$\begin{bmatrix}\text{MHSA}(768)\\ \text{FFN}(3072)\end{bmatrix}\times 3$	$\begin{bmatrix}\text{MHSA}(1024)\\ \text{FFN}(4096)\end{bmatrix}\times 6$	C×[8×14×14]
token pruning	$\begin{bmatrix}\text{MHSA}(768)\\ \text{keep rate}=\rho\\ \text{FFN}(3072)\end{bmatrix}\times 1$	$\begin{bmatrix}\text{MHSA}(1024)\\ \text{keep rate}=\rho\\ \text{FFN}(4096)\end{bmatrix}\times 1$	C×[8×14×14×ρ]
stage2	$\begin{bmatrix}\text{MHSA}(768)\\ \text{FFN}(3072)\end{bmatrix}\times 2$	$\begin{bmatrix}\text{MHSA}(1024)\\ \text{FFN}(4096)\end{bmatrix}\times 5$	C×[8×14×14×ρ]
token pruning	$\begin{bmatrix}\text{MHSA}(768)\\ \text{keep rate}=\rho\\ \text{FFN}(3072)\end{bmatrix}\times 1$	$\begin{bmatrix}\text{MHSA}(1024)\\ \text{keep rate}=\rho\\ \text{FFN}(4096)\end{bmatrix}\times 1$	C×[8×14×14×ρ^2]
stage3	$\begin{bmatrix}\text{MHSA}(768)\\ \text{FFN}(3072)\end{bmatrix}\times 2$	$\begin{bmatrix}\text{MHSA}(1024)\\ \text{FFN}(4096)\end{bmatrix}\times 5$	C×[8×14×14×ρ^2]
token pruning	$\begin{bmatrix}\text{MHSA}(768)\\ \text{keep rate}=\rho\\ \text{FFN}(3072)\end{bmatrix}\times 1$	$\begin{bmatrix}\text{MHSA}(1024)\\ \text{keep rate}=\rho\\ \text{FFN}(4096)\end{bmatrix}\times 1$	C×[8×14×14×ρ^3]
stage4	$\begin{bmatrix}\text{MHSA}(768)\\ \text{FFN}(3072)\end{bmatrix}\times 2$	$\begin{bmatrix}\text{MHSA}(1024)\\ \text{FFN}(4096)\end{bmatrix}\times 5$	C×[8×14×14×ρ^3]
norm	LayerNorm(768)	LayerNorm(1024)	C×[8×14×14×ρ^3]
GFLOPs, ρ=0.7/1.0	134.2 / 223.8	409.4 / 707.9	-

Table 1: **Architecture details of EVAD backbone.** The *token pruning* denotes a transformer layer with keyframe-centric token pruning and is the same as the blocks in each stage when keep rate ρ=1.0. The output sizes are denoted by $\{C{\times}T{\times}S{\times}\rho\}$ for channel, temporal, spatial and keep rate sizes.

set its shortest side to range from 256 to 320 pixels. For the experiments on UCF101-24, we set the shortest side of each video frame to 224 for training and 256 for inference. We perform the same training recipe as in the baseline for EVAD models under different keep rates without additional modifications, which indicates that our method can be simply incorporated into existing models and work well.

For inference, given an input video clip, EVAD directly predicts 100 proposal boxes and the corresponding person detection and action classification scores. The prediction boxes with a detection score larger than 0.7 are taken as the final results.

C. Additional Results

We compare our EVAD with the state-of-the-art methods on AVA v2.1 in Table 2. With fewer input frames, EVAD with ViT-B backbone outperforms most two-stage and end-to-end models and has comparable performance to AIA with mAP of 31.1 vs. 31.2. When we apply a larger backbone ViT-L and use the same pre-trained dataset as AIA, the performance can surpass AIA by a large margin. Also, it outperforms the newly end-to-end methods TubeR and STMixer, the latter two equipped with long-term feature banks.

model	e2e	$T \times \tau$	backbone	pre-train	mAP
AVA [5]*	✗	40 × 1	I3D-VGG	K400	15.8
LFB [11]	✗	32 × 2	I3D-R101-NL	K400	27.7
CA-RCNN [12]	✗	32 × 2	R50-NL	K400	28.0
SlowFast [2]	✗	32 × 2	SF-R101-NL	K600	28.2
ACAR-Net [7]	✗	32 × 2	SF-R101-NL	K400	30.0
AIA [10]	✗	32 × 2	SF-R101	K700	31.2
ACRN [8]*	✓	20 × 1	S3D-G	K400	17.4
VAT [3]	✓	64 × 1	I3D-VGG	K400	25.0
WOO [1]	✓	32 × 2	SF-R101-NL	K600	28.0
TubeR [15]	✓	32 × 2	CSN-152	IG+K400	31.7
STMixer [13]	✓	32 × 2	CSN-152	IG+K400	34.4
EVAD, ρ=0.7	✓	16 × 4	ViT-B	K400	31.1
EVAD, ρ=0.7	✓	16 × 4	ViT-L	K700	**38.7**

Table 2: **Comparison with the state-of-the-art on AVA v2.1.** ✓denotes an end-to-end approach using a unified backbone, and ✗denotes a two-stage approach using two separated backbones. $T \times \tau$ refers to the frame number and corresponding sample rate. Methods marked with * leverage optical flow input.

Next, we provide the video-mAP results on UCF101-24 and JHMDB. As shown in Table 3, we compare our EVAD with the state-of-the-art frame-level detector WOO and tubelet-level detector TubeR. Our method follows the pipeline of WOO and is also a frame-level detector. We achieve better performance than WOO under various set-

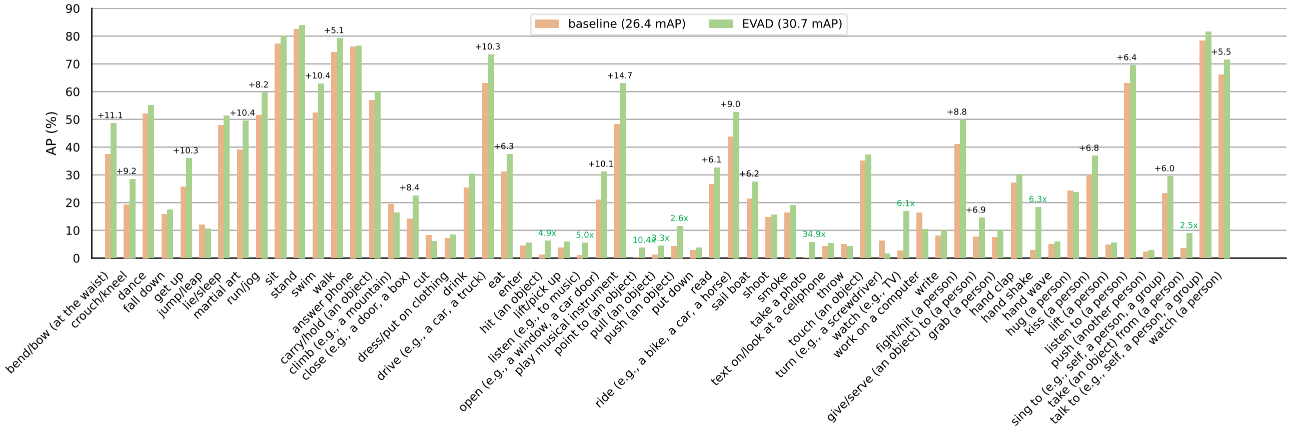


Figure 1: **Per-category AP for ViT baseline (26.4 mAP) and EVAD (30.7 mAP) on AVA.** Categories that increase in absolute value by more than 5% are marked in **black** and those more than twice the AP of the baseline are marked in **green**.

model	UCF24				JHMDB	
	f-mAP	0.20	0.50	0.50:0.95	0.20	0.50
WOO‡ [1]	76.7	74.4	55.8	26.0	70.0	69.5
TubeR* [15] (I3D)	81.3	**85.3**	**60.2**	29.7	81.8	80.7
TubeR [15] (CSN-152)	83.2	83.3	58.4	28.9	**87.4**	**82.3**
EVAD, ρ=1.0	84.9	76.6	60.1	**30.0**	78.2	77.1
EVAD, ρ=0.6	**85.1**	76.4	58.8	29.1	79.0	77.8

Table 3: **Comparison on UCF24 and JHMDB with video-mAP.** ‡ indicates our implementation. Methods marked with * leverage optical flow input.

tings of the video-mAP. Without using tubelet annotations for training, our method has the lower performance than TubeR at multiple IoU thresholds. However, we observe that our EVAD achieves on-par or even better performance when increasing IoU thresholds on UCF101-24. This indicates that our EVAD can generate high-quality action tubes.

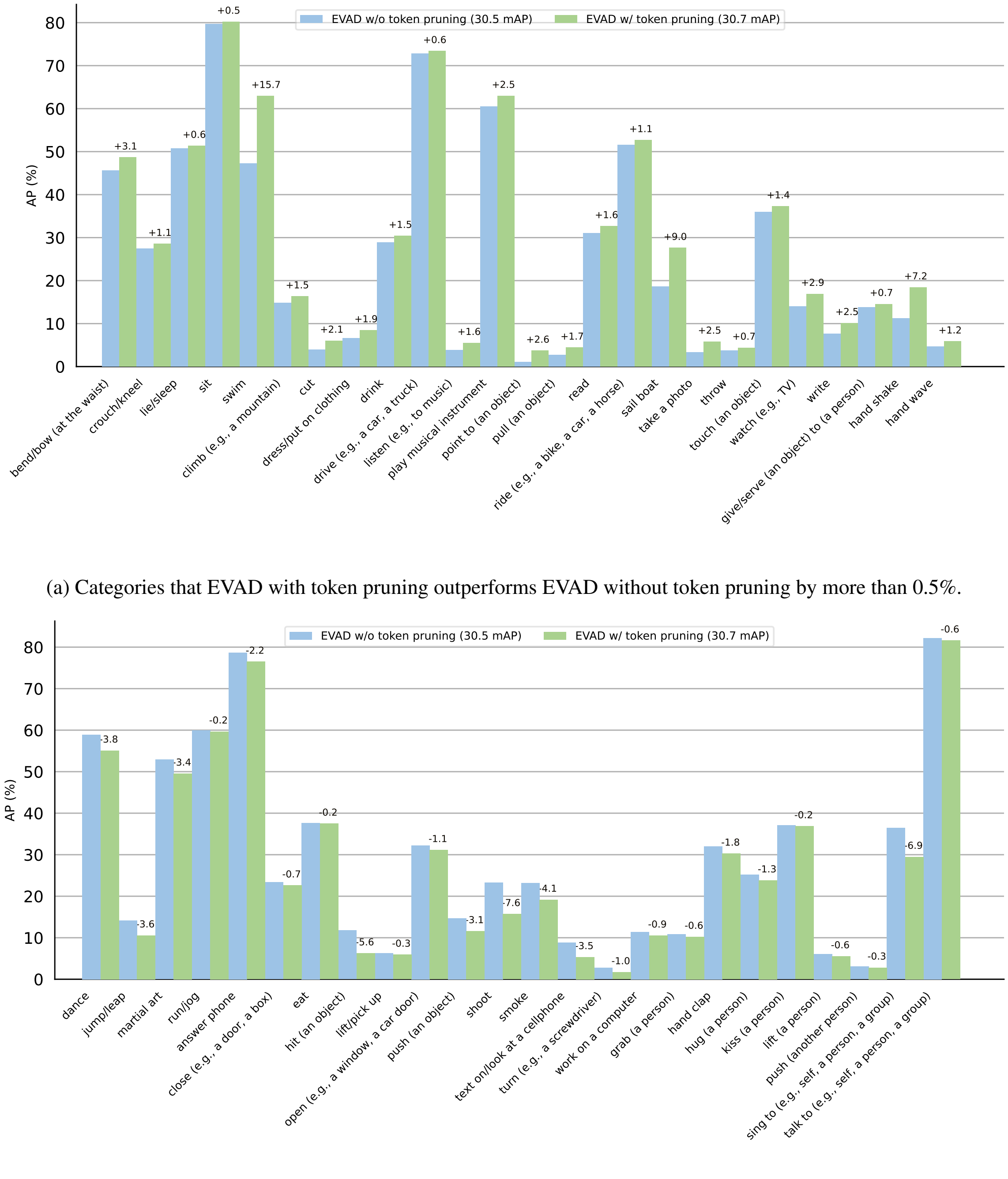
D. Result Analysis and Visualization

Firstly, we compare the per-category performance of a ViT baseline and our EVAD on AVA, as shown in Fig. 1. Our method improves in 53 out of 60 categories, with significant improvements for categories with fast movement (e.g., *bend/bow (at the waist)* (+11.1%) and *martial art* (+10.4%)) and categories with scene interaction (e.g., *drive (e.g., a car, a truck)* (+10.3%) and *play musical instrument* (+14.7%)). This illustrates the effectiveness of two core designs of our EVAD, i.e., the proposed keyframe-centric token pruning can preserve tokens that contain sufficient action semantics, and these preserved tokens can enrich each actor spatiotemporal and scene feature through the proposed context refinement decoder.

As seen in the experimental results, our keyframe-centric token pruning enables EVAD to achieve comparable performance to its counterpart without pruning, and we further compare the performance of these two models on each category of AVA, as shown in Fig. 2. We observe that although the overall performance of two models is comparable, the performance trend of them is inconsistent on each category. Concretely, EVAD with token pruning performance increases a lot in *swim* (+15.7%), *sail boat* (+9.0%), and *hand shake* (+7.2%) categories, and decreases a lot in *hit* (-5.6%), *shoot* (-7.6%), and *sing to* (-6.9%). We consider that token pruning drops a high percentage of tokens (66%), resulting in poor performance on categories with small motion or interaction with small objects, and good performance on categories with opposite characteristics.

To show the effectiveness of our token pruning method for retaining semantic cues, we collect more visualizations of token pruning as a supplementary of Figure 4 in our paper, as shown in Fig. 3. EVAD is able to preserve important tokens in non-keyframes, e.g., for the person putting on clothing in *example 2*, it can preserve the sleeve with a large movement deformation. Moreover, we observe that those frames further away from the keyframe retain a greater number of tokens in most examples. Due to the slowness of video semantics varying in the temporal dimension [14], frames adjacent to the keyframe have higher semantic redundancy. When we perform keyframe-centric tokens pruning, more tokens from adjacent frames are discarded.

Finally, to illustrate that using preserved video tokens for context refinement can maintain the same performance as using the whole video tokens, we visualize the attention maps of our context refinement decoder at different token keep rates, as shown in Fig. 4, where the attention result is the average of the attention maps between n actors of interest and M video tokens. We observe that those regions with



(a) Categories that EVAD with token pruning outperforms EVAD without token pruning by more than 0.5%.

(b) Categories that EVAD without token pruning performs better.

Figure 2: **Per-category AP for EVAD w/o token pruning (30.5mAP) and EVAD w/ token pruning (30.7 mAP) on AVA.**

high attentive values of the decoder without token pruning can be preserved at various keep rates, e.g., the hat and the wearing hand in *example 1*. This further demonstrates that our token pruning can retain semantic information for action classification and the proposed context refinement decoder can enrich actor features via remaining context, which maintain the detection accuracy.

References

[1] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *IEEE/CVF International Conference on Computer Vision*, 2021.

[2] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and

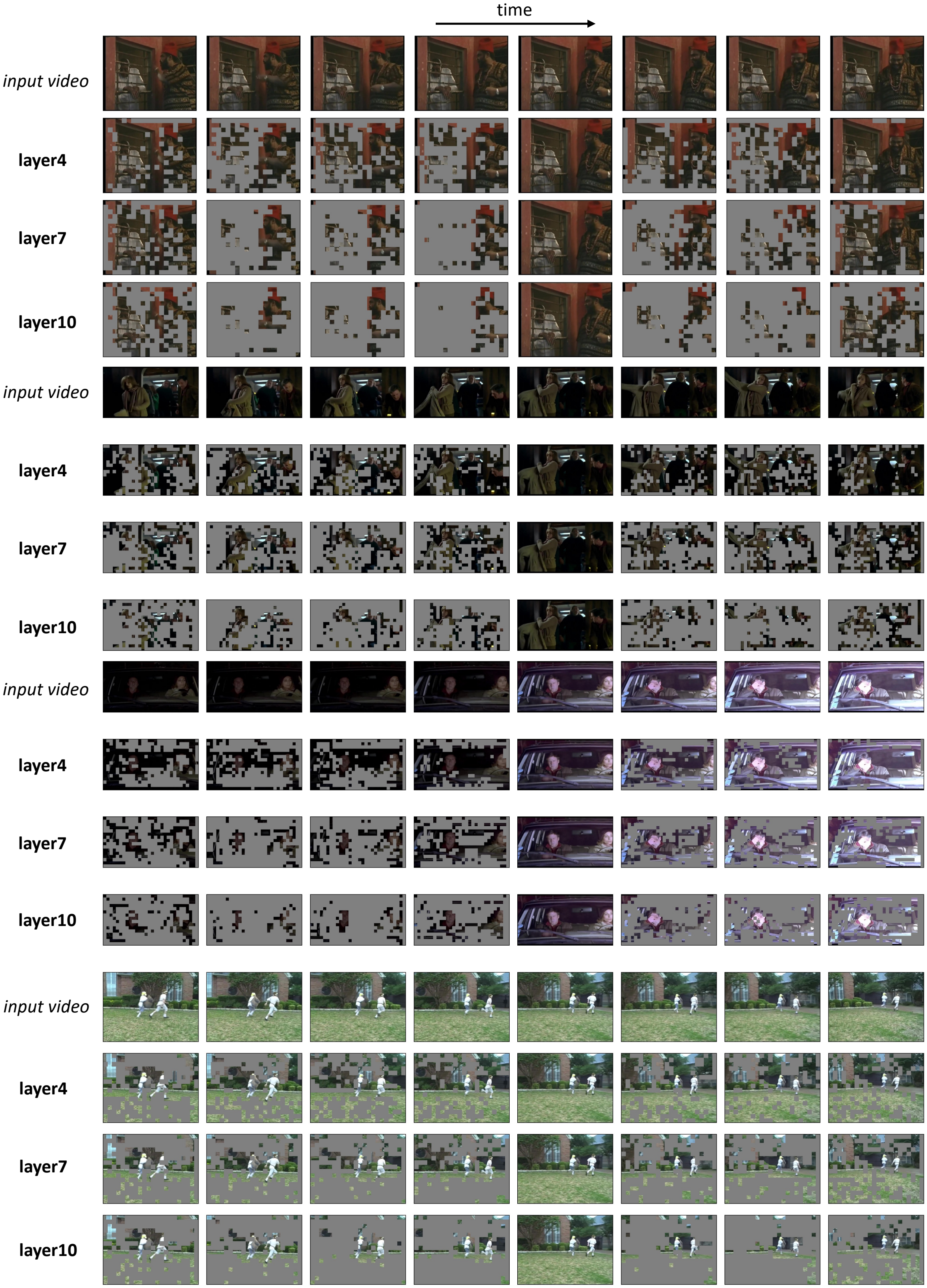


Figure 3: **More visualization of preserved tokens by encoder layers with keyframe-centric token pruning on AVA.** Given an input of 8-frame intermediate tokens, the keyframe tokens are all retained on the fifth column, and redundant tokens in other frames are progressively removed.

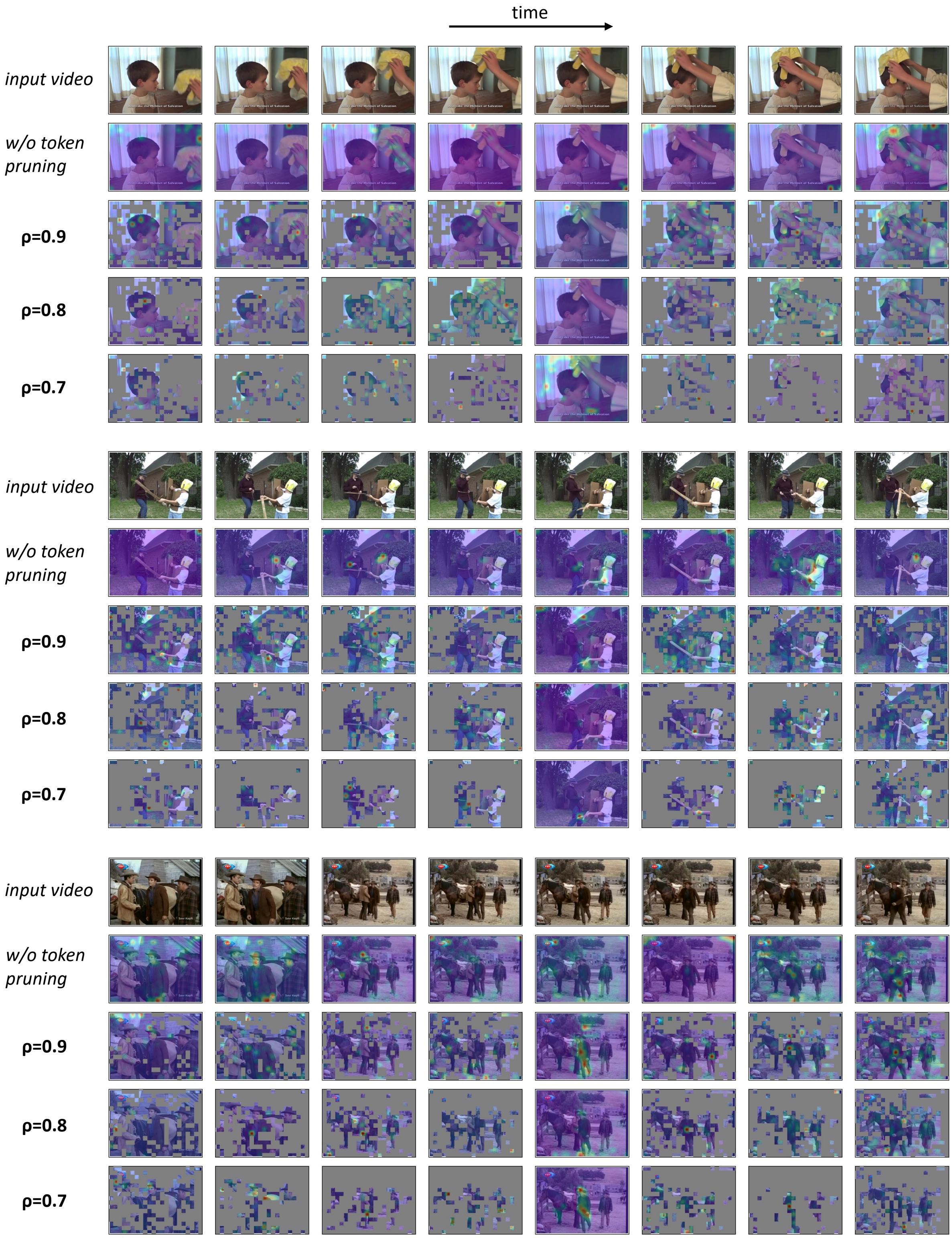


Figure 4: **Visualization of context refinement decoder attention maps on AVA.** For each example, 1st row: RGB frames of the input clip with a stride of 8 for viewing better, where the fifth image represents the keyframe. 2nd row: attention maps of the last decoder layer without token pruning between actors of interest and the whole video tokens. 3rd to 5th rows: attention maps of the last layer at different token keep rates (ρ=0.9/0.8/0.7) between actors of interest and the preserved tokens.

Kaiming He. Slowfast networks for video recognition. In *IEEE/CVF International Conference on Computer Vision*, 2019.

[3] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[4] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010.

[5] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[7] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[8] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *European Conference on Computer Vision*, 2018.

[9] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[10] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision*, 2020.

[11] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross B. Girshick. Long-term feature banks for detailed video understanding. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[12] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu. Context-aware rcnn: A baseline for action detection in videos. In *European Conference on Computer Vision*, 2020.

[13] Tao Wu, Mengqi Cao, Ziteng Gao, Gangshan Wu, and Limin Wang. Stmixer: A one-stage sparse action detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[14] Zhang Zhang and Dacheng Tao. Slow feature analysis for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 2012.

[15] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al. Tuber: Tubelet transformer for video action detection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.