How Much Temporal Long-Term Context is Needed for Action Segmentation? Supplemental Material

Emad Bahrami[1] Gianpiero Francesca[2] Juergen Gall[1,3]
[1]University of Bonn, Germany [2]Toyota Motor Europe, Belgium
[3]Lamarr Institute for Machine Learning and Artificial Intelligence, Germany

We provide additional experiments and implementation details.

1. Implementation Details

As mentioned in the paper, we use 9 layers and 4 stages for all datasets. We use $W = G = 64$ for Assembly101 and 50Salads, and $W = 64$ and $G = 8$ for the Breakfast dataset since the videos are shorter than Assembly101 and 50Salads. We use Adam [2] optimizer and cosine learning rate decay [4]. The starting learning rate for Breakfast and Assembly101 is 0.00025 and the decay to 0.00005 starts after 15 epochs. We train Breakfast for 150 epochs and Assembly101 for 120 epochs. The model for 50Salads is trained for 200 epochs with a fixed learning rate of 0.00065.

2. Impact of Temporal Downsampling

Fig. 1 shows the impact of temporally downsampling the input. In this experiment, the model has access to the full context of a video but in a lower temporal resolution since the input is temporally downsampled. The performance of the model degrades compared to no downsampling.

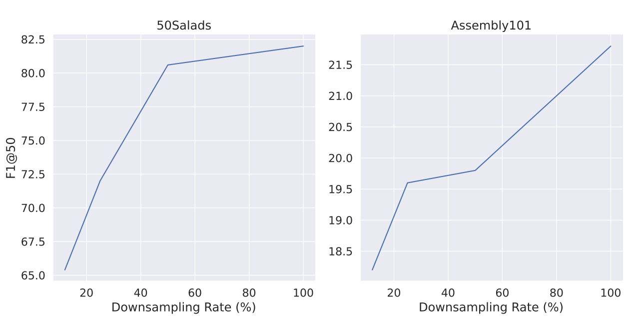

Figure 1: Impact of different downsampling rates on the 50Salads dataset (left) and the Assembly101 dataset (right).

Features	F1@{10, 25, 50}			Edit	Acc
CLIP	65.8	57.6	44.2	64.2	62.4
I3D	89.4	87.7	82.0	83.2	87.7

Table 1: Results are on 50Salads.

Attention	F1@{10, 25, 50}			Edit	Acc
FlashAttention	55.2	53.0	48.7	42.6	84.6
RandomAttention	49.0	45.7	41.8	37.2	85.6
Ours	89.4	87.7	82.0	83.2	87.7

Table 2: Results are on 50Salads.

3. Other Features

In order to evaluate the impact of using vision-language models, we extract features using CLIP [5] from 50Salads and report the result of action segmentation in Table 1. Without additional fine-tuning, the features do not perform well.

4. Alternative Efficient Attentions

We compare in Table 2 our approach with RandomAttention [6] from XFormer [3] and FlashAttention [1]. These types of attention focus on sparseness and result in fragmented segments, which is indicated by high accuracy, but very low F1 and Edit scores.

References

[1] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[3] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, and Daniel Haziza. xformers: A modular and hackable transformer modelling library. `https://github.com/facebookresearch/xformers`, 2022.

[4] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *International Conference on Learning Representations (ICLR)*, 2017.

[5] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

[6] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems (NeurIPS)*, 33, 2020.