# Supplementary Document:
# 3D CNNs with Adaptive Temporal Feature Resolution

Mohsen Fayyaz[1,*], Emad Bahrami[1,*],
Ali Diba[2], Mehdi Noroozi[3], Ehsan Adeli[4], Luc Van Gool[2,5], Juergen Gall[1]
[1]University of Bonn, [2]KU Leuven, [3]Bosch Center for Artificial Intelligence,
[4]Standford University, [5]ETH Zürich
{lastname}@iai.uni-bonn.de, emadbahramirad@gmail.com,
{firstname.lastname}@kuleuven.be, mehdi.noroozi@de.bosch.de, eadeli@stanford.edu

## Appendix

This document provides supplementary material as mentioned in the main paper.

## A. Implementation Details

**Modified 3DResNet-18** The architecture of our modified 3DResNet-18 is shown in Table A.1. In case of 3DResNet-18+ATFR, we place SGS after the *ResBlock 2*.

| stage | layer | output size |
|---|---|---|
| raw | - | 32 × 244 × 224 |
| conv$_1$ | 5 × 7 × 7, 8, stride 1, 2, 2 | 32 × 112 × 112 |
| pool$_1$ | $1 \times 3 \times 3$, max, stride $1, 2, 2$ | $32 \times 56 \times 56$ |
| res$_2$ | $\begin{bmatrix} 3 \times 1 \times 1, 8 \\ 1 \times 3 \times 3, 8 \\ 1 \times 1 \times 1, 32 \end{bmatrix} \times 2$ | $32 \times 56 \times 56$ |
| res$_3$ | $\begin{bmatrix} 1 \times 1 \times 1, 64 \\ 1 \times 3 \times 3, 64 \\ 1 \times 1 \times 1, 256 \end{bmatrix} \times 2$ | $32 \times 28 \times 28$ |
| res$_4$ | $\begin{bmatrix} 1 \times 1 \times 1, 128 \\ 1 \times 3 \times 3, 128 \\ 1 \times 1 \times 1, 512 \end{bmatrix} \times 2$ | $32 \times 14 \times 14$ |
| res$_5$ | $\begin{bmatrix} 3 \times 1 \times 1, 256 \\ 1 \times 3 \times 3, 256 \\ 1 \times 1 \times 1, 1024 \end{bmatrix} \times 2$ | $32 \times 7 \times 7$ |
|  | global average pool, fc | $1 \times 1 \times 1$ |

Table A.1: Modified 3DResNet-18

| Model | Train | Inference (fps) |
|---|---|---|
| X3D-S | 131h | 2834 |
| X3D-S+ATFR | **121h** | **4295** |

Table A.2: Runtime on Kinetics-400.

**SlowFast-8x8-R50+ATFR** Following [9] for training on the Something-Something-V2 dataset, the input temporal length to the SlowFast-8x8-R50+ATFR is set to 64. Due to the intensive size of the temporal domain, we limit the the temporal domain size of the SGS for each path-way. For the fast path-way we set the temporal domain size to 8. In other words, SGS is applied over temporal blocks with temporal length of 8 and temporal stride of 8. For the slow path-way we set the temporal domain size to 2. Since we drop zero bins in SGS, this may cause size mismatch for fusion in lateral connections. We therefore zero pad the smaller size tensors to the bigger ones.

## B. Runtime

To evaluate the runtime, we use X3D-S as the base model and report the runtimes for training and inference. As shown in Table A.2, SGS reduces the training time on Kinetics by 10h. The ATFR equipped model processes almost 51% more frames per second (fps) during inference. Our approach also requires less memory and we can use a larger batch size (BS), namely 256 instead of 208. This shows that the proposed approach substantially reduces GFLOPs, training and inference time, and memory usage.

## C. Different number of bins

The number of the sampling bins $B$ controls the maximum number of possible output feature maps of

[*] Mohsen Fayyaz and Emad Bahrami equally contributed to this work. Emad Bahrami contributed to this project while he was a visiting researcher at the Computer Vision Group of the University of Bonn.

| B | 4 | 8 | 16 | 32 |
|---|---|---|---|---|
| top1 | 61.4 | 64.7 | 64.7 | 69.6 |
| top5 | 86.3 | 85.8 | 86.2 | 88.8 |
| GFLOPs | 3.5 | 4.2 | 5.5 | 14.0 |

Table A.3: Ablations on the effect of changing the numbers of bins $B$ for 3DResNet-18+ATFR on Mini-Kinetics. The model is trained and validated for different number of bins. We show top-1 and top-5 classification accuracy (%).

| B | 4 | 8 | 16 | 32 |
|---|---|---|---|---|
| top1 | 51.1 | 61.4 | 64.4 | 69.6 |
| top5 | 75.1 | 83.5 | 85.5 | 88.8 |
| GFLOPs | 3.5 | 5.0 | 8.0 | 14.0 |

Table A.4: Ablations on the effect of changing the numbers of bins $B$ only during inference for 3DResNet-18+ATFR on Mini-Kinetics. The model is trained with 32 bins, but inference is performed with a different number of bins. We show top-1 and top-5 classification accuracy (%).

the SGS module. By setting $B = T$, the SGS module can keep all feature maps in case it is needed. To study the effect of changing $B$, we have evaluated the model performance by changing $B$ during training and inference. The base model is the 3DResNet-18 (Fig. A.1) trained on Mini-Kinetics. As it can be seen in Table A.3, reducing $B$ decreases the accuracy, but also the GFLOPS. This is expected since SGS is forced to discard information for each video if $B < T$ even if there is no redundancy among the feature maps.

As a second experiment, we change the number of bins only for inference while we train the model with $B = 32$. This setting is interesting since it shows how flexible the approach is and if GFLOPS can be reduced at inference time without the need to retrain the model. The results are shown in Table A.4. If we compare the results with Table A.3, we observe that the accuracy for training with $B = 32$ and testing with $B = 16$ is only slightly lower than training and testing with $B = 16$. This shows that the GFLOPS can be reduced on the fly if it is required. However, if the difference between the number of bins during training and during inference is getting larger, the accuracy drops.

## D. Cartesian/Spherical Coordinates

As mentioned in in Sec. 5.2.1, we use the magnitude of the embedding vectors as the similarity measurement to create the similarity bins. Instead of magnitudes, we can use other measures. While the results are reported in Table 1 of the paper, we describe how the approach works with spherical coordinates.

To use the spherical coordinates of the vectors for creating the similarity bins, we use multi-dimensional bins and sampling kernels. In an $L$ dimensional spherical coordinate system, we can use different subsets of coordinates for $\Delta_t^k$ with varying number of elements $K$ to create similarity bins, *e.g.*, $K = L$ when using all of the coordinates, $K = L - 1$ when using angular coordinates, or $K = 1$ when using the radial coordinate only. Therefore, similar to Eq. (2) and (3) of the paper, we can estimate $\beta_b^k$ for every $\Delta^k$.

By using a sampling kernel $\Psi(\Delta_t^k, \beta_b^k)$ as in Eq. (4) of the paper but for each $k$, a differentiable multi-dimensional sampling operation can be defined by

$$\mathcal{O}_b = \sum_{t=1}^{T} \mathcal{I}_t \prod_{k=1}^{K} \Psi(\Delta_t^k, \beta_b^k). \qquad (1)$$

## E. Similarity Guided Sampling Visualization

The SGS layer aggregates similar input temporal feature maps into the same output feature map. To better understand such aggregation operation, we have visualized the input and output feature maps of the SGS layer in Figure A.1. We have used a 3DResNet-50+ATFR trained on the Mini-Kinetics dataset. The sampling kernel used in this experiment is the linear kernel and the number of bins is set to 32. As it can be seen in Figure A.1, the input temporal feature maps are aggregated to 4 output feature maps. The aggregated feature maps contain both the spatial and temporal information. In this example, the $4^{th}$ channel of the aggregated feature maps capture some motion flow that can be seen in the visualization.

## F. UCF101 and HMDB51 Results

**UCF-101** [6] contains 13K videos with 101 action classes. It is split into 3 splits with around 9.5K videos in each. For this dataset, we report the average accuracy over three splits.

**HMDB-51** [5] has about 7000 videos with 51 action classes. It contains 3 splits for training and validation. Similar to UCF-101, we report the average accuracy over three splits. Table A.5 shows the results on UCF-101 and HMDB51. The GFLOPs of our 3DResNet-R50+ATFR on UCF-101 and HMDB-51 are 22.2 and 23.1, respectively. As it can be seen, 3DResNet+ATFR gets comparable results compared to other 3D CNNs while having less GFLOPs as discussed in the paper.

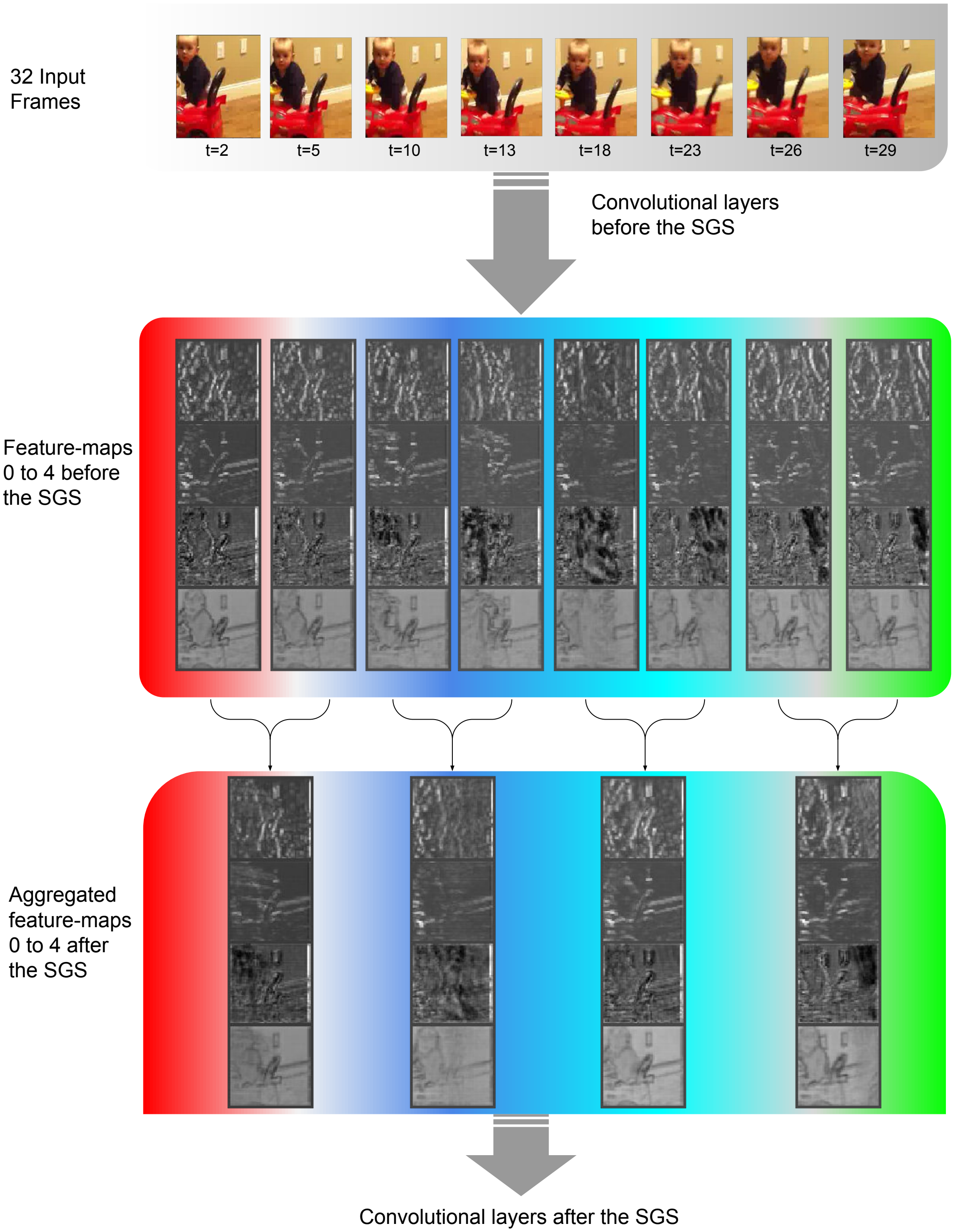
32 Input Frames
t=2
t=5
t=10
t=13
t=18
t=23
t=26
t=29
Convolutional layers before the SGS
Feature-maps 0 to 4 before the SGS
Aggregated feature-maps 0 to 4 after the SGS
Convolutional layers after the SGS

Figure A.1: Visualization of the feature maps of 3DResNet-50+ATFR with linear kernel. In the first row, 8 frames out of 32 input frames are shown. The corresponding temporal feature maps of *ResBlock 2* are depicted in the second row. The third row shows the aggregated feature maps after the SGS. Note that we only show the first 4 channels of the feature maps for better visualization.

| model | backbone | UCF101 | HMDB51 |
|---|---|---|---|
| C3D [7] | RenNet18 | 89.8 | 62.1 |
| RGB-I3D [1] | Inception V1 | 95.6 | 74.8 |
| R(2+1)D [8] | ResNet50 | 96.8 | 74.5 |
| DynamoNet [4] | ResNet101 | 96.6 | 74.9 |
| HATNet [3] | ResNet50 | 97.8 | 76.5 |
| 3DResNet+ATFR | ResNet50 | 97.9 | 76.7 |

Table A.5: Comparison with other methods on UCF101 and HMDB51.

| Model | top1 | GFLOPs |
|---|---|---|
| X3D-S | 77.9 | 1.9 |
| X3D-S+ATFR | 78.0 | **1.1** |
| X3D-S+Temporal Attention | 78.3 | 1.9 |

Table A.6: Comparison with attention modules. The models are trained and tested on the Mini-Kinetics dataset.

## G. Comparison to Attention/Gating Mechanisms

To better analyze the effect of our similarity guided sampling mechanism, we add attention modules to the base model and compare the final accuracy and GFLOPs to the base model and the ATFR model. To this end, we use a temporal attention mechanism following [2]. Similar to our SGS module, we add this attention module on top of the ResBlock2. As it can be seen in Table A.6, the model equipped with the attention module achieves similar accuracy while requiring higher GFLOPs compared to the model equipped with SGS. The reason for such a great difference in GFLOPs is that attention modules perform a weighting of the features, while our approach clusters and reduces features. If all features are the same, the attention module should weight them equally while our approach reduces them to one feature.

## References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017.

[2] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ 2-nets: Double attention networks. In *Advances in neural information processing systems*, pages 352–361, 2018.

[3] Ali Diba, Mohsen Fayyaz, Vivek Sharma, Manohar Paluri, Jürgen Gall, Rainer Stiefelhagen, and Luc Van Gool. Large scale holistic video understanding. In *ECCV*, 2020.

[4] Ali Diba, Vivek Sharma, Luc Van Gool, and Rainer Stiefelhagen. Dynamonet: Dynamic action and motion network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6192–6201, 2019.

[5] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[6] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[7] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[8] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018.

[9] Chao-Yuan Wu, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl. A multigrid method for efficiently training video models. In *CVPR*, 2020.