Efficient Human Vision Inspired Action Recognition Using Adaptive Spatiotemporal Sampling

Khoi-Nguyen C. Mac, Minh N. Do, *Fellow, IEEE*, and Minh P. Vo

***Abstract*— Adaptive sampling that exploits the spatiotemporal redundancy in videos is critical for always-on action recognition on wearable devices with limited computing and battery resources. The commonly used fixed sampling strategy is not context-aware and may under-sample the visual content, and thus adversely impacts both computation efficiency and accuracy. Inspired by the concepts of foveal vision and pre-attentive processing from the human visual perception mechanism, we introduce a novel adaptive spatiotemporal sampling scheme for efficient action recognition. Our system pre-scans the global scene context at low-resolution and decides to skip or request high-resolution features at salient regions for further processing. We validate the system on EPIC-KITCHENS and UCF-101 (split-1) datasets for action recognition, and show that our proposed approach can greatly speed up inference with a tolerable loss of accuracy compared with those from state-of-the-art baselines. Source code is available in https://github.com/knmac/adaptive_spatiotemporal.**

***Index Terms*— Adaptive sampling, spatiotemporal, action recognition.**

I. Introduction

OUR visual world is highly predictive, making it highly inefficient to process each individual piece of data with the same amount of effort. To cope with it, human perceptual system subconsciously pre-scans the scene to determine important events before actual processing. This mechanism is known as *pre-attentive* processing [2], [3], [4]. The pre-capturing images, although appear to be less clear, constructs the global perception of the scene [5], [6]. Furthermore, the human brain also focuses on certain regions within our *foveal* visual field [5], [7], [8], [9]. These two behaviors are strikingly similar to the objective of our temporal and spatial sampling, respectively.

Sampling has also been one of the most studied problems in various areas of video analysis, such as action recognition and video summarization [10], [11], [12], [13], [14], [15], due to the redundancy between consecutive frames. With the increase in model complexity, it gets progressively expensive to process a single frame. This is even more crucial for resource-limited devices such as AR/VR headsets, like Google Glasses, HoloLens, Ray-Ban Stories, *etc.* [16], [17], [18]. However, picking a fixed sampling scheme does not guarantee the performance as important information may be under-sampled. Temporally, it is evident that the number of frames required to represent a video vary, depending on the action categories [19]. Therefore, an adaptive sampling rate is preferred as over-sampling results in more computational cost while under-sampling can make performance suffer. Similarly, spatial sampling is also necessary in general computer vision tasks, which is applicable on individual frames of a video sequence. It is preferred to have an adaptive sampling scheme as having a fixed one also leads to similar problems as in time domain.

Inspired by the mechanism of human visual perception, we propose a novel adaptive spatiotemporal sampling framework to imitate the human vision. Fig. 1 shows an overview

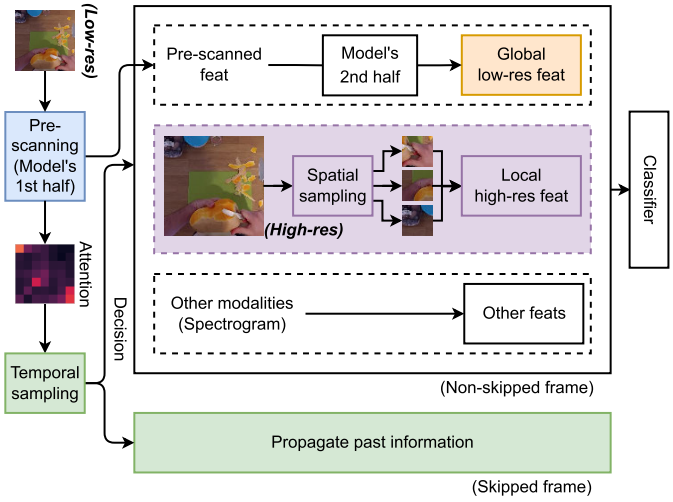


Fig. 1. Our proposed system has two major components: temporal and spatial sampling. Based on a pre-scanned features, the temporal sampler decides whether to process the frame fully (Full model), or skip to the frame and propagate past information (bottom block). The spatial sampler in turns select RoIs from high-res input to augment the features with low-res inputs. We also include features from other available modalities if a frame is fully processed. We color-code the spatial sampling as purple and temporal sampling as green. We further illustrate details of the two routines in Fig. 7 and Fig. 10.

Manuscript received 26 September 2022; revised 30 May 2023 and 11 August 2023; accepted 18 August 2023. Date of publication 31 August 2023; date of current version 21 September 2023. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Clinton Fookes. *(Corresponding author: Khoi-Nguyen C. Mac.)*

Khoi-Nguyen C. Mac was with the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Champaign, IL 61801 USA. He is now with Amazon, Seattle, WA 98121 USA (e-mail: knmac@amazon.com).

Minh N. Do is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana–Champaign, Champaign, IL 61801 USA, and also with the VinUni-Illinois Smart Health Center, VinUniversity, Hanoi, Vietnam (e-mail: minhdo@illinois.edu).

Minh P. Vo is with Spree3D, Alameda, CA 94502 USA (e-mail: phuocminhvo@gmail.com).

Digital Object Identifier 10.1109/TIP.2023.3310661

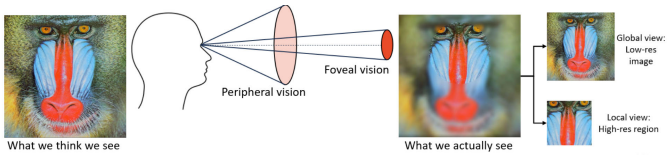


Fig. 2. The foveal vision in human corresponds to sharp and centered vision to obtain fine local details, while the peripheral vision corresponds to low visual acuity to get coarser global information. This is similar to the local-view from high-res region and global-view from low-res image in our spatial sampler.

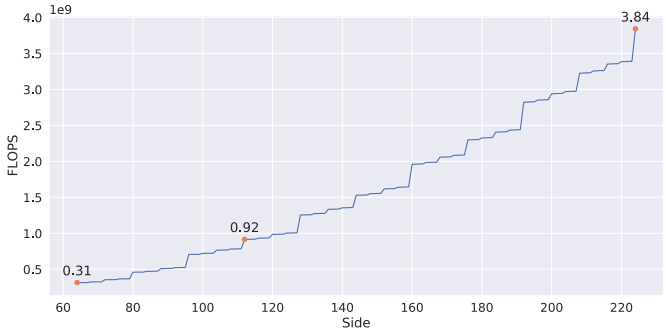


Fig. 3. Complexity of SAN19 with different input dimensions. The horizontal axis shows the side N of an input with dimension $3 \times N \times N$. We highlight the values where $N = 224, 112, 64$, corresponding to our choices for the size of high-res, low-res, and cropped high-res images.

of the entire system, with two main components that are built upon the self-attention mechanism: (1) spatial sampler uses the observed attention to sample regions of interest and (2) temporal sampler hallucinates attention in the next frame to model future expectation.

The *spatial sampler* is motivated by human *foveal vision* (Fig. 2). The idea is to only focus on specific regions rather than the whole scene to save computation. It can be seen that lower input sizes significantly reduce the computational complexity. However, it also compromises the performance. To overcome this, we use input at two different resolutions: low-res whole image with size of 112×112 and high-res image crops with size of 64×64. The low-res images are processed as a whole for pre-scanning process of temporal sampler and global feature extraction. For the high-res input, we retrieve regions corresponding to the most "important" locations based on the extracted attention and use them to augment the low-res image for the visual recognition task. In our system, we use the low-res image of size 112×112 and top-k regions of size 64×64. Fig. 3 analyzes the computational complexity in GFLOPS with respect to the spatial dimension of RGB images. We highlight the GFLOPS with size 224×224 (high-res baseline), compared with size 112×112 (low-res input) and size 64×64 (cropped high-res regions).

The *temporal sampler* follows the concept of *pre-attentive processing* (Fig. 4) such that it extracts attention by briefly pre-scanning a low-res input and decides whether to further process the frame if something interesting happens. Since it is possible to predict what would happen in the future [20], [21], we consider an event "interesting" if it is drastically different from what is expected. The idea of using another network for pre-scanning has been discussed in other work [12], [22], however in our proposed approach, we split a backbone network into two halves and use the first one to pre-scan

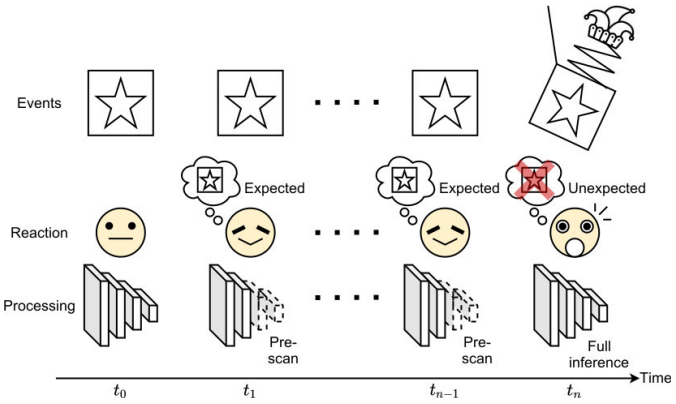


Fig. 4. Pre-attentive processing is a subconscious accumulation of info from the environment, *ie.*, all available information is pre-attentively processed, then our brains will choose the important event to dive deep in. Since, motion is also a pre-attentive feature, this mechanism inspires our temporal sampling scheme. As a by-product, predictable events are more likely to be ignored. This is similar to pre-scanning new frames to see if they are similar to the hallucinated prediction in our temporal sampler.

instead of introducing an additional one. We observe that two consecutive frames produce similar attention at some certain intermediate layers, making it possible to pre-scan by forwarding up to such layers. To model future expectation, we hallucinate future attention and compare it with the observed one. When the hallucination matches the actual attention, there is no unexpected event and the model simply uses the previous classification results. Otherwise, the remaining processing routine is carried out to compute new classification.

We demonstrate the effectiveness of our system on the action recognition task on EPIC-KITCHENS [23] and UCF-101 (split-1) [24] datasets. Our system reduces computational complexity with a tolerable loss of accuracy compared to the baseline counterparts. We also provide qualitative results to reason the sampling results.

To the best of our knowledge, we are the first ones to model the spatiotemporal sampling based on such visual perception mechanisms in human. Although there have been several attempts to address the problems of adaptive sampling in video analysis, our approach can flexibly take advantage of more sophisticated backbone networks, as long as they rely on the self-attention techniques. We trust that our work can benefit the community by encouraging more studies that connect the vision between computers and humans.

Contribution: (1) We introduce a novel adaptive spatiotemporal sampling scheme inspired by human vision, where the temporal sampler can pre-scan (mid-way inference) the low-res input to decide whether to skip processing by comparing the observed and hallucinated attention. (2) As a part of the sampling routine, our spatial sampler selects small high-res RoIs induced by the attention map in pre-scanning process. (3) We showcase the system on egocentric and generic videos where our model reduces the computational power with a small loss of accuracy.

II. Related Work

A. Bio-Inspired Action Recognition

Action recognition models inspired by human biology has been introduced even before the explosion of deep learning.

Tracing back to Giese and Poggio in 2003 [25], their work analyzes the dorsal and ventral streams in the brains, which corresponds to how humans perceive spatial and visual features respectively. Following this work, researchers model action recognition by studying human brains' activity: [26] uses a neurally plausible memory-trace learning rule; [27] applies neuro-biological model of motion processing and proposes motion-sensitive units; [28] proposes a model that communicated through discrete spikes; [29] focuses on MT cells, which are sensitive to motion contrasts; [30] created human action templates for human object recognition based on neuro-biological model. Such research often focuses more on the biology aspect and adopts classical machine learning solutions for modeling. More recently, [31] introduces a photonic hardware approach and implements a simple RNN model; [32] adopts LSTM and Spatial pyramid pooling to extract robust features of each frame's tracked area. Such work typically explores more advanced deep learning techniques but the proposed models do not focus much on the interpretability. Furthermore, bio-inspired action recognition work generally conducts experiments on generic action datasets, where the bodies are fully observable and the number of action classes is limited, such as KTH [33], Weizmann [34], UCSD [35]. Our work is not completely based on neuro-biological models. Instead, we observe there is a similarity between human vision and the trending attention mechanism, and leverage this to build the spatio-temporal sampling in our system. Our model also provides interpretability via the generated hallucination. Furthermore, the model is trained on more recent datasets and can work with both generic and ego-centric action recognition.

B. Action Recognition

The task of action recognition has evolved from the traditional two-stream networks [36] to more advanced models, *eg.*, C3D, I3D, ResNet3D, R(2+1)D, TBN, TSN, and LSTA [37], [38], [39], [40], [41], [42], [43]. Such standard techniques often demand expensive computation, leading to the challenge of high power consumption [44], which is crucial for action recognition using always-on wearable devices, such as AR/VR glasses. Our adaptive sampling scheme aims to address this problem. Another branch of action recognition focuses on modeling the spatial and temporal structure of human bodies, via the skeletal joints in each person or how different people interact as a whole group [45], [46], [47], [48], [49]. In the scope of this paper, we do not address such structures and only focus on the sampling aspect.

C. Adaptive Inference and Sampling

Techniques to reduce the complexity of deep networks can be divided into three sub-categories: ignoring layers in deep models, removing input regions, and skipping frames. [50] introduces a stochastic method to drop layers during the training phase. SkipNet [51] and BlockDrop [52] later propose to use reinforcement learning to dynamically drop layers for both training and validation. In spatial domain, RS-Net [53] can decide which resolution to switch to by sharing parameters among different image scales. PatchDrop [54], on the other hand, removes unimportant regions of input images via reinforcement learning. For applications in the area of general video analysis, it is more desirable to rely on time sampling. It has been shown that temporal redundancy results in wasted computation, as some videos only require a single frame to represent [19]. There have been attempts to process videos at multiple frame rates as different actions can happen at different paces [55], [56]. Recently, SC-Sampler [12] and ARNet [22] tackle temporal sampling by using additional simple networks for pre-scanning the features. [57] focuses on spatial redundancy and frame-skipping is a special case in their formulation. In contrast, [58] focuses on the time domain and do not consider partial spatial information. We explicitly model spatiotemporal sampling using self-attention, as inspired by human vision, and only use a sub-collection of layers to pre-scan. Our model is also more interpretable with visualizable attention and hallucination.

D. Self-Attention

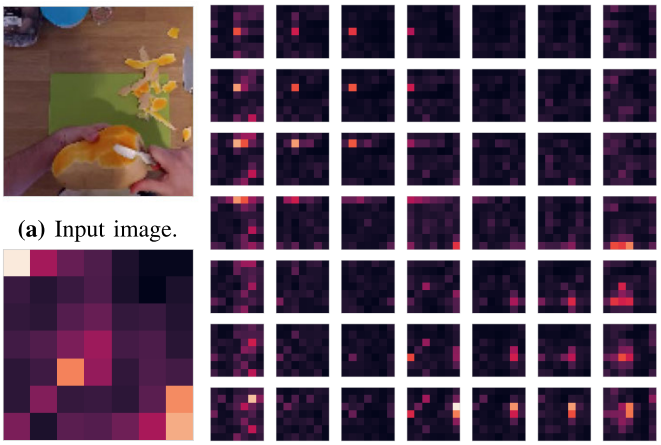
In computer vision, gradient-based methods are usually used to generate saliency maps, which can determine the regions where a trained model considers "relevant" to the output [59], [60], [61], [62]. More recently, self-attention is introduced in natural language processing community as a way to direct the focus of deep nets [63]. Since the attention allows a model to focus more on important regions, such self-attention mechanism has been attracting great interest from the computer vision community [64], [65], [66]. Our approach uses such attention as a driving mechanism to find the important regions and frames, allowing spatiotemporal sampling adaptively. More recently, vision transformer has been introduced as another attention-based approach and adopted in several work [67], [68], [69], [70], [71], [72]. Since our method operates on top of attention, the backbone models can be flexibly interchanged with any attention-based CNNs. Here, we choose to use the SAN-19 backbone [64] and focus on improving efficiency of the baseline models.

III. Approach

Consider a video dataset $\mathcal{D} = \{(\mathbf{v}_n, \mathbf{y}_n)\}_{n=1}^{N}$, where $\mathbf{v}_n$ is a video sequence and $\mathbf{y}_n$ is the corresponding groundtruth label. We assume that the video sequences have the same length of T frames, *ie.*, $\mathbf{v}_n = [\mathbf{x}_n^{(1)}, \mathbf{x}_n^{(2)}, \ldots, \mathbf{x}_n^{(T)}]$, where each frame is $\mathbf{x}_n^{(t)} \in \mathbb{R}^{3\times H\times W}, \forall t \in \{1, .., T\}$. Suppose that we have a video classifier $F(\mathbf{v}_n) = \hat{\mathbf{y}}_n$, with some complexity $\mathcal{O}_F$. The goal is to construct another classifier $\tilde{F}$ with less complexity while retaining the accuracy. We address this by introducing the temporal sampler $\mathcal{T}$ and spatial sampler $\mathcal{S}$, *ie.*, $\hat{\mathbf{y}}_n = \tilde{F}(\mathbf{x}_n; \mathcal{T}, \mathcal{S})$, such that $\mathcal{O}_{\tilde{F}} < \mathcal{O}_F$. At a high level, the spatial sampler chooses the top-k regions based on the most activated areas in the attention map. The temporal sampler decides whether to skip frames, whose attentions are similar to the model's future prediction.

A. Cumulative Global Attention

Our cumulative global attention is built upon the pairwise attention formulation of Zhao et al. [64]. We rewrite this



(a) Input image. **(b)** Cumulative global attention $\mathbf{A}$. **(c)** Local attentions $\mathbf{a}_i$ from different footprint $\mathcal{R}_i$.

Fig. 5. Local attention and cumulative global attention from the same input image and at the same layer. Hotter color indicates more salient regions. We average across the channel dimension for visualization.

pairwise attention as

$$\mathbf{z}_i = \sum_{j \in \mathcal{R}(i)} \alpha\left(Q(\mathbf{x}_i), K(\mathbf{x}_j)\right) \odot V(\mathbf{x}_j), \tag{1}$$

where $i, j \in \mathbb{R}^2$ are the spatial indices, $Q(\mathbf{x}_i)$, $K(\mathbf{x}_j)$, and $V(\mathbf{x}_j)$ are the query, key, and value encodings, and α is the compatibility function, usually defined as a softmax. Such compatibility function is locally defined over the footprint $\mathcal{R}(i)$. We then denote the *local attention* at i as

$$\mathbf{a}_i = \left[\alpha\left(Q(\mathbf{x}_i), K(\mathbf{x}_j)\right)\right], \quad j \in \mathcal{R}(i). \tag{2}$$

Learning to generate such attentions is difficult because we also need to model the underlying relationship of neighboring footprints, *ie.*, a_i and a_{i+1} have overlapping footprint. However, it is simpler to generate a global attention map where the footprints are already encoded. Thus, we use the *cumulative global attention*, defined as

$$\mathbf{A} = \sum_i \mathbf{a}_i \otimes \mathbb{1}\{\mathcal{R}(i)\}, \tag{3}$$

where $\mathbb{1}\{\mathcal{R}(i)\}$ is the indicator function that removes locations outside of footprint $\mathcal{R}(i)$ and $\otimes$ is the multiplication of $\mathbf{a}_i$ with the corresponding footprint. Notice that $\mathbf{a}_i$ has the same spatial dimension as $\mathcal{R}(i)$, while $\mathbf{A}$ has the same spatial dimension as the input feature map $\phi(\mathbf{x})$. For simplicity, unless stated otherwise we use "attention" to denote the cumulative global attention in the remaining sections.

The locations of neighboring footprints here are similar to the concept of sliding the kernel window in convolution. If the stride is less than the kernel size, it will result in overlapping regions. Fig. 6 shows an example with kernel size of 3 and stride of 1, causing overlapping of 2. Since the overlapping regions carry replicated information, it is not efficient if we represent attention as a collection of $\mathbf{a}_i$. Therefore, we aggregate them as $\mathbf{A}$ and learn to generate its future version using the hallucinator. We have included this clarification in the updated manuscript.

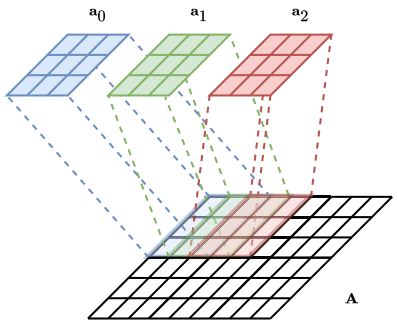


Fig. 6. Cumulative global attention addresses the sliding effects. The blue ($\mathbf{a}_0$), green ($\mathbf{a}_1$), and red squares ($\mathbf{a}_2$) are local attentions with neighboring footprints. The bottom square is the aggregated global attention $\mathbf{A}$.

Fig. 5 shows an example of the cumulative global attention (Fig. 5b), aggregated from the local attentions across multiple footprints (Fig. 5c), given the same input (Fig. 5a). We see that the neighboring $\mathbf{a}_i$ are overlapping (because of stride of 1 in this example), similar to convolution. By using $\mathbf{A}$, we avoid having to encode such overlapping conditions, making it easier to learn. Specifically, the activation of $\mathbf{A}$ reflects the "important regions" in the input images, being the hands and the bowl (top-left corner). It motivates to use such global attention maps to find ROIs in our spatial sampler.

B. Spatial Sampler

The goal of the spatial sampler is to provide the high-resolution inputs at locations where it matters, which is similar to foveal vision in human. Formally, given input $\mathbf{x} \in \mathbb{R}^{3\times H\times W}$, we compute the corresponding low and high-res inputs $\mathbf{x}_l \in \mathbb{R}^{3\times \frac{H}{d}\times \frac{W}{d}}$ and $\mathbf{x}_{h,k} \in \mathbb{R}^{3\times H'\times W'}$, respectively obtained by rescaling (with the down-sampling factor d) and cropping $\mathbf{x}$ ($H' < H, W' < W$) at k different locations. While d is defined as our hyper-parameter, the cropping regions for $\mathbf{x}_{h,k}$ are computed by the spatial sampler $\mathcal{S}$. Giving attention $\mathbf{A}$, we find all connected regions and pick the k regions with highest summation. We then linearly project those regions back to pixel space, based on the scaling of spatial dimension between the input image $\mathbf{x}$ and the attention $\mathbf{A}$.

Fig. 7 shows the details of the spatial sampler. We extract the attention from the low-res image $\mathbf{x}_l$ and use it to sample the top-k regions in the original image $\mathbf{x}$. This results in $\mathbf{x}_{h,k}$ with lower spatial dimension, while retaining the original resolution of $\mathbf{x}$. As we use the same backbone network to process images of different resolution, we add a global average pooling layer at the end of the feature extractor to remove the spatial dimension. The features are then fed to the three-head GRU classifier. The heads correspond to low-res features, high-res features, and their concatenation are used to encourage strong learning feature at each resolution. We constrain the scaling factor d and the bounding box size H', W' such that the complexity of using $\mathbf{x}_l$ and $\mathbf{x}_{h,k}$'s is less than that of $\mathbf{x}$. We choose $d = 2$ and $H' = W' = 64$, based on our complexity analysis in Fig. 3.

In Fig. 8, we illustrate some example results of our spatial sampler, extracting the top 3 regions in a few frames of a video sequence. The colors here denote the order of the bounding

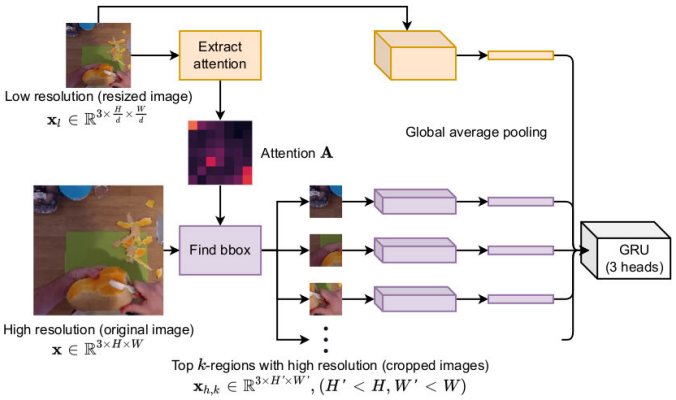


Fig. 7. Spatial sampler uses attention from low-res image to sample the top-k regions from the (original) high-res input. $\mathbf{x}_l$ gives a global view, while $\mathbf{x}_{h,k}$ provides local views at important regions of the original image $\mathbf{x}$. The final global average pooling removes spatial dimension of the features, which are combined and fed to the three-head GRU classifier. The heads correspond to low-res features, high-res features, and their concatenation, and are used to encourage strong learning feature at each resolution.

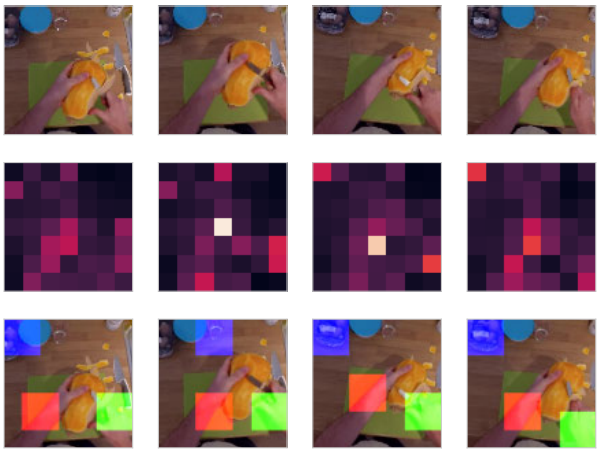


Fig. 8. Sampled regions from the top-3 spatial sampler. From top to bottom: (1) input frames, (2) attention, and (3) bounding boxes in pixel space. Red, green, and blue colors denote the top 1, 2, and 3 accordingly. The trajectories of the boxes reflect to their activated regions. Such behavior allows prediction of future attentions.

boxes, based on the most activated regions in the attention. It is observed that the sampled regions are not varying rapidly when the activation are similar. This usually happens when the actions are occurring slowly, suggesting that we can predict future attentions in such cases.

C. Hallucinator

The objective of our hallucinator is to predict future information. If our prediction is similar to the actual observation in the future, then such event is well-expected and there is no need to run further inference. Instead of generating the whole RGB frame in pixel level, our hallucinator only generates the attention map. Intuitively, such attention maps can locate important regions of the inputs and can be simpler to generate than RGB frames. Assuming the temporal consistency, *ie.*, the attentions of consecutive frames $\mathbf{A}^{(t)}$ and $\mathbf{A}^{(t+1)}$ are similar if the action is slow enough, it is possible to hallucinate the future attentions from the current frame. We formalize our hallucinator $\mathcal{H}$ as:

$$\tilde{\mathbf{A}}^{(t+1)} = \mathcal{H}(\mathbf{A}^{(t)}) \quad \text{s.t. } \tilde{\mathbf{A}}^{(t+1)} \approx \mathbf{A}^{(t+1)}, \tag{4}$$

where $\tilde{\mathbf{A}}^{(t+1)}$ is the hallucination (predicted future attention). To quantify the similarity between $\tilde{\mathbf{A}}^{(t+1)}$ and $\mathbf{A}^{(t+1)}$, we use the structural similarity index measure (SSIM) [73] as this metrics can compare the structure of input tensors. We train the hallucinator by minimizing our belief loss:

$$\mathcal{L}_b = -\frac{1}{T-1}\sum_{t=2}^{T} \text{SSIM}(\mathcal{H}(\mathbf{A}^{(t-1)}), \mathbf{A}^{(t)}), \tag{5}$$

where the function SSIM() computes the structural similarity between the hallucination $\mathcal{H}(\mathbf{A}^{(t-1)})$ and the attention $\mathbf{A}^{(t)}$. We minimize the negative SSIM score since the default SSIM ranges from 0 to 1, where larger SSIM indicates higher similarity.

We build the hallucinator as a convolutional LSTM [74] with encoder-decoder layers and apply teacher forcing technique [75] for the training routine. The idea is to gradually increase the number of hallucinated frames as input. We rewrite the hallucinator during training as:

$$\tilde{\mathbf{A}}^{(t+1)} = \begin{cases} \mathcal{H}(\mathbf{A}^{(t)}), & p \le F_r, \\ \mathcal{H}(\tilde{\mathbf{A}}^{(t)}), & p > F_r, \end{cases} \tag{6}$$

where p is a uniformly randomized and $F_r \in [0, 1]$ is the teacher forcing ratio. Such ratio is initialized as 1 and gradually decayed every epoch to increase the chance of hallucinating from $\tilde{\mathbf{A}}^{(t)}$. Note that F_r is not available for evaluation, and $\tilde{\mathbf{A}}^{(t+1)} = \mathcal{H}(\tilde{\mathbf{A}}^{(t)})$. We also add in a warm-up phase to initialize hidden memory of the hallucinator, *ie.*, $\tilde{\mathbf{A}}^{(t+1)} = \mathcal{H}(\mathbf{A}^{(t)}), \forall t \le t_{warm}$ for both training and evaluation phases.

Fig. 9 shows an example of the hallucination from a video sequence, where the first row is the input video sequence, the second row is the attention extracted from a layer, and the last row is the hallucination, generated by our hallucinator. There is a missing hallucination at the first frame because we are generating future attention. It is observed that the most activated regions of the attention here are located around the two hands. As the hands move in time, these regions also move with a similar manner, in both the attention and hallucination. It suggests that our hallucinator can predict where the important regions would be in the future. We also provide the negative SSIM scores at the bottom to compare the structural similarity between the attention and hallucination. Note that our objective here is not to generate a perfect hallucination, but only to use it as a guideline for the temporal sampler.

D. Temporal Sampler

Given a video sequence $\mathbf{v} = [\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(T)}]$, the objective of the temporal sampler is to adaptively select a subset of "important" frames that can still represent $\mathbf{v}$, similar to human's pre-attentive processing mechanism. A frame $\mathbf{x}^{(t)}$ is considered as unimportant if we can reasonably predict its attention. From Section III-A, we can retrieve the attention

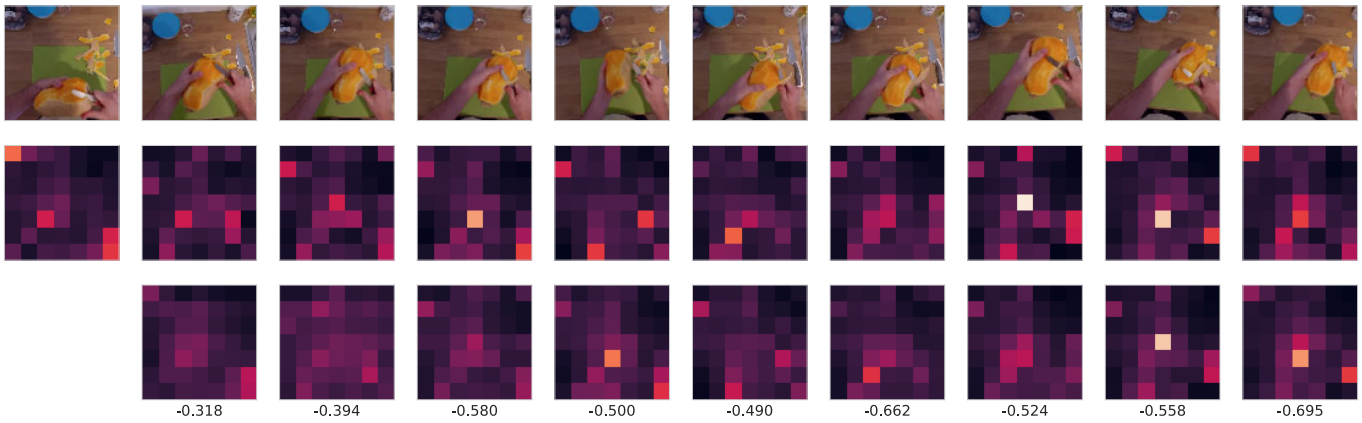


Fig. 9. Attention and corresponding hallucination of a video sequence. From top to bottom: (1) input frames, (2) attention, and (3) hallucination. Negative SSIM scores between the attention and hallucination are included at the bottom (0 means most different and -1 means most similar). Activated regions of the attentions and hallucinations match the movements of the hands across frames, showing the temporal consistency property.

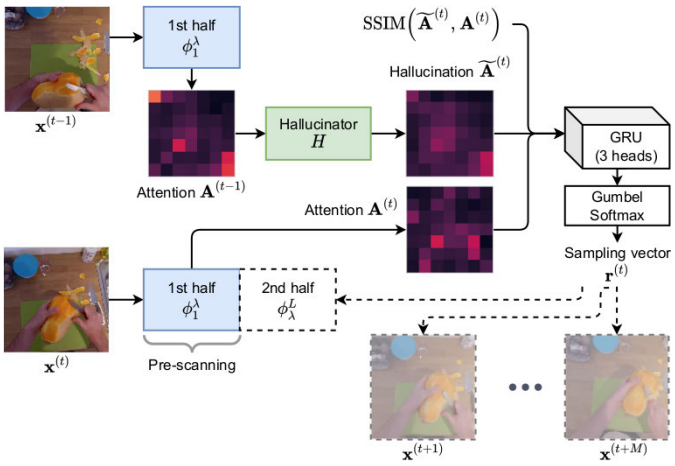


Fig. 10. Temporal sampler with inputs at $t-1$ and t. Attention from the model's first half at time t, hallucination computed at time $t-1$, and their SSIM score are fed to a GRU to compute the sampling vector $\mathbf{r}^{(t)}$, deciding how many frames to skip (including the second half of the current frame). Model weights are shared across frames.

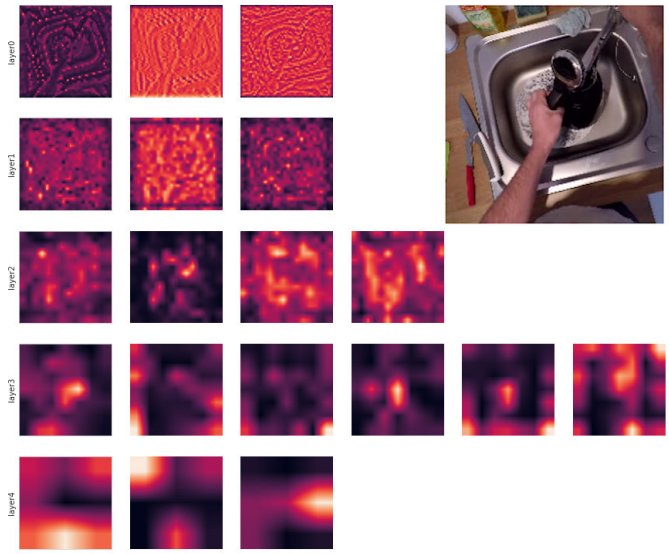
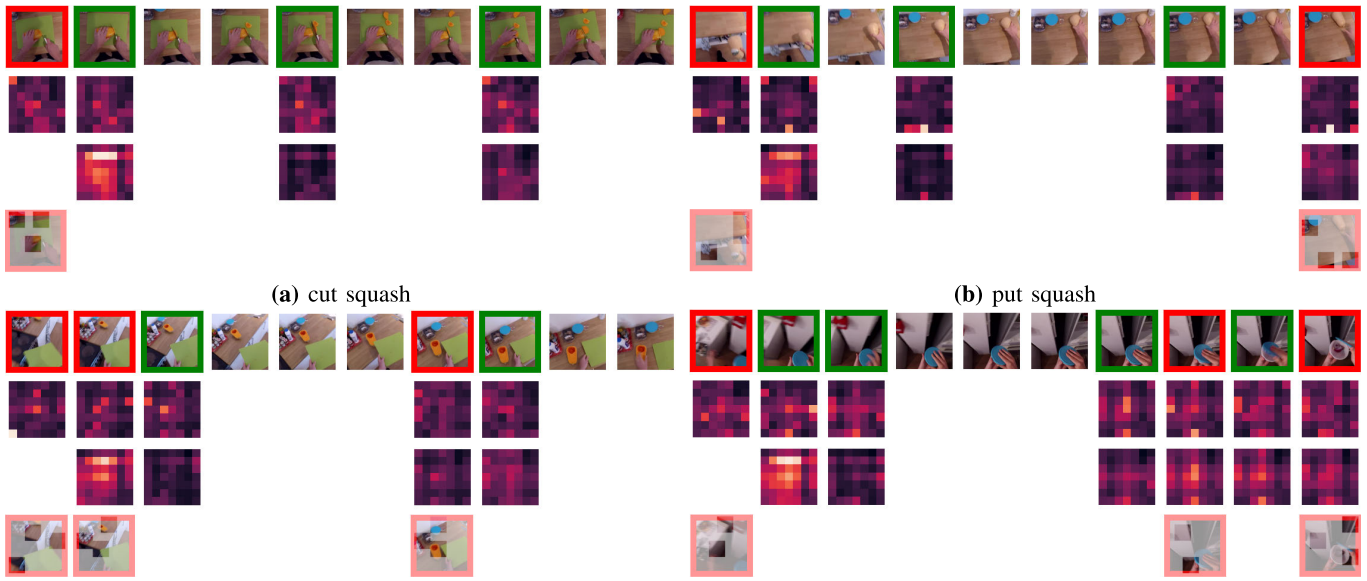


Fig. 11. Attention extracted from all bottleneck layers of SAN19. The input image is showed in the top-right corner. Earlier layers show more fragmented regions while latter ones provide more concise attentions. We visualize the attention maps with bilinear interpolation for better *visibility* across different layers. The color mapping of the visualization is not normalized because of different range of values across layers.

and hallucination at any arbitrary layer from a model of L layers. Suppose that the attention is extracted at layer $\lambda < L$, it is wasteful to compute the last $L-\lambda$ layers if the temporal sampler decides to skip this frame. In other words, we can forward a frame up to layer λ and choose to run the rest of the model adaptively.

Formally, consider the feature extractor of a deep network of L layers as a composite function, we can split it into two halves at layer $\lambda \in \{1, \ldots, L\}$, *ie.*, $\phi_1^L(\mathbf{x}) = \phi_\lambda^L(\phi_1^\lambda(\mathbf{x}))$. The first half ϕ_1^λ is used for pre-scanning while the second half ϕ_λ^L can also be augmented with information from other modalities for the classification task later. The temporal sampler $\mathcal{T}$ determines the sampling routine by computing a sampling vector $\mathbf{r} = [\mathbf{r}^{(1)}, \ldots, \mathbf{r}^{(T)}]$, *ie.*, $\mathcal{T}(\mathbf{v}) = [\mathbf{x}^{(t)} \times \mathbf{r}^{(t)}]_{t=1}^T$, with $\mathbf{r}^{(t)} \in \{0,1\}^{M+1}$, where $\mathbf{r}^{(t)}[m] = 1$ means we can skip m frames, $m \in \{0, \ldots, M\}$. Fig. 10 shows the details of the temporal sampler. At time t, the attention $\mathbf{A}^{(t)}$ is extracted using the first half of the feature extractor ϕ_1^λ. To generate the sampling vector $\mathbf{r}^{(t)}$, the flattened feature is concatenated with the hallucination and the corresponding SSIM score, and then fed to a GRU. The output features are fed to a Gumbel Softmax [76], which makes the sampling vector differentiable.

Given the sampling vector, we now describe our frame-skipping routine. Denoting such number of skipping frames as $m^* = \text{argmax}_m \mathbf{r}^{(t)}[m]$, there are two possible scenarios: $m^* = 0$ and $m^* \in [1, M]$. In the *first* case, we do not skip anything and continue to run the remaining part of the network, thus the complexity is that of the full pipeline $\mathcal{O}_{full}$. In the *second* case, we only pre-scan the current frame, which has already been done, and skip computation on the next $m^* - 1$ frames. The classification results and memory from recurrent models are propagated accordingly. The complexity for these m^* frames is $\mathcal{O}_{pre} = \mathcal{O}_{\phi_1^\lambda} + \mathcal{O}_H + \mathcal{O}_\mathcal{T}$, being the model's first half, hallucinator, and temporal sampler. Note



(a) cut squash **(b)** put squash **(c)** put board **(d)** open container

Fig. 12. Qualitative results of spatiotemporal sampling on video sequences. From top to bottom are: the input video sequence, attention, hallucination generated from past attention, and the non-max suppressed top salient regions from the spatial sampler. The frames with red boundary are the non-skipped ones (full inference) while the frames with green boundary denote pre-scanning (running the first half). The frames without any boundary are the skipped ones. We choose to use the first frame to initiate the samplers and always associate it with the full inference.

that $\mathcal{O}_{full} = \mathcal{O}_{pre} + \mathcal{O}_{rest}$, where $\mathcal{O}_{rest}$ is the complexity of running the rest of the pipeline, including spatial sampler, other modalities, and classifier. Under such policy, we train the temporal sampler by minimizing the weighted sum of classification loss $\mathcal{L}_{class}$ (only using full-inference frames) and efficiency loss $\mathcal{L}_e$, which is defined as:

$$\mathcal{L}_e = n_{full} \cdot \mathcal{O}_{full} + n_{pre} \cdot \mathcal{O}_{pre}, \tag{7}$$

where n_{full} and n_{pre} are respectively the number of frames with full inference and only pre-scanning. Without any constraints, it is possible that no frame would be fed to the second half of the pipeline, *ie.*, $\text{argmax}_m \mathbf{r}^{(t)}[m] \neq 0, \forall t$. To avoid such scenario, we include a warm-up step, where the full pipeline is run at the first frame. It also helps initialize memory for recurrent models and ensures that we have classification result for at least one frame.

IV. Experiments

We evaluate our system on EPIC-KITCHENS 2018 [23], following the training and validation splits of [41], and split-1 of UCF-101 [24]. EPIC-KITCHENS contains 55 hours of full-HD, 60fps egocentric videos, each of which is associated with a verb (125 classes) and a noun (331 classes) label. The action is thus defined as a pair of the corresponding verb and noun labels, *eg.*, [cut, squash] and [open, container]. UCF-101 is a dataset of generic actions, consisting of 27 hours of 25fps videos, divided into 101 action classes. For UCF-101, we report results on split-1 (9537 training and 3783 validation videos) to ensure that our experiments are comparable.

For EPIC-KITCHENS, we use two input modalities, namely *RGB* and *Spectrogram*, corresponding to the vision and audio domains. Although the optical flow is provided as a part of the dataset, we avoid using it because such data are computationally expensive in real-life scenarios. We treat RGB inputs as the guiding modality of the system because our hallucinator and samplers rely on the attention from vision data. The spectrogram inputs act as the additional modality and are only used when a frame is not skipped by the temporal sampler. For UCF-101, we only use the RGB modality for better comparison with our baseline.

We benchmark our system with top-1 and top-5 accuracy for both datasets. For EPIC-KITCHENS, we include the accuracy of three domains: action, verb, and noun. To assess the system's efficiency, we further report the models' FLOPS per frame, which is proportional to inference time and power consumption. Since the model complexity is time-variant for the experiments with temporal sampler, we instead provide the accumulated FLOPS over the whole validation sets and the average FLOPS per frame. We also report the trade-off factor, defined as GFLOPS per the top-1 accuracy, to compare efficiency of different models (lower means better). This metric indicate how much of computation is required for one percent of accuracy on average.

A. Implementation Details

Our system uses SAN19 with pairwise self-attention (equivalent to ResNet50 [64]) as the backbone network to extract features. For EPIC-KITCHENS, the additional Spectrogram (256×256) is constructed from the audio channels using the same processing procedure as in [41]. Fig. 11 shows attention maps from all bottleneck layers of SAN19, where the input with size of 3×112x112 is provided in the top-right corner of the figure. We see that latter layers result in more concise and smaller attention maps, which suggests easier hallucination of the future. However, using more layers also requires more complexity, as being shown in Fig. 13, where the horizontal

TABLE I

RESULTS OF BASELINES AND SPATIAL SAMPLER $\mathcal{S}$ ON EPIC-KITCHENS *Val-Set*. THE AVERAGE GFLOPS PER-FRAME IS INCLUDED TO INDICATE THE MODEL COMPLEXITY. WE SHOWCASE THE PERFORMANCE AS TOP-1 AND TOP-5 PER-VIDEO ACCURACY FOR ACTION, VERB, AND NOUN DOMAINS. WE ALSO INCLUDE THE EFFICIENCY TRADE-OFF (LOWER MEANS BETTER), COMPUTED AS THE GFLOPS OVER TOP-1 ACCURACY, TO SHOW HOW MUCH GFLOPS IS NEEDED WITH EACH ACCURACY PERCENT ON AVERAGE. THIS METRICS IS ALSO USEFUL TO COMPARE MODELS WITH DIFFERENT BACKBONES. WE RETRAIN TBN [41] WITH DIFFERENT INPUT MODALITIES ON VALIDATION SET TO COMPARE WITH OUR BASELINE SAN19. USING THE SAME INPUT MODALITIES, OUR BASELINE ACHIEVES COMPARABLE RESULTS WITH TBN. OUR MODELS WITH SPATIAL SAMPLERS ARE DENOTED AS $\mathcal{S}_k$, WHERE k IS THE NUMBER OF ROIS EXTRACTED. $\mathcal{S}_0$ MEANS NO SPATIAL SAMPLING. $\mathcal{S}_3$ PROVIDES THE BEST ACCURACY AMONG SPATIAL SAMPLERS AND STILL WITH A LOWER COMPLEXITY THAN THE BASELINE

Model	Backbone	RGB,Flow, Audio	Size RGB	Avg GFLOPS	Top-1	Top-5	Verb Top-1	Verb Top-5	Noun Top-1	Noun Top-5	Trade -off
SlowFast [55], [56]	Res101,8x8	R	224	13.25	21.9	39.7	55.8	83.1	27.4	52.1	0.605
AVSlowFast [56]	Res101,8x8	R+A	224	16.13	24.2	43.6	58.7	83.6	31.7	58.4	0.667
TBN [41](ret)	Inception	R+A+F	224	6.95	34.83	54.09	63.31	88.29	46.00	68.31	0.200
TBN [41](ret)	Inception	R+A	224	4.62	28.32	60.30	56.96	86.61	41.03	65.35	0.163
SAN19-base	Res50	R+A	224	8.64	27.52	57.55	55.84	86.24	39.83	62.84	0.314
$\mathcal{S}_0$	SAN19	R+A	112	**5.80**	24.56	53.34	53.50	83.95	34.57	58.84	**0.236**
$\mathcal{S}_1$	SAN19	R+A	112	6.16	25.23	54.05	55.17	**84.49**	35.49	59.68	0.244
$\mathcal{S}_2$	SAN19	R+A	112	6.48	24.94	53.55	55.21	84.15	35.20	59.17	0.260
$\mathcal{S}_3$	SAN19	R+A	112	6.80	**25.77**	**54.42**	**55.71**	84.15	**35.78**	**59.84**	0.264

axis shows the layer names and the vertical axis indicates the accumulated FLOPS up to that layer. Therefore, we choose to extract attention at layer3-0 of the backbone network because it shows good trade-off between complexity and performance in our experiments. This gives us the attention feature map with the dimensionality of $32 \times 7 \times 7$.

The hallucinator is a Conv-LSTM with 1 layer and 32 hidden dimensions. It is equipped with a encoder and a decoder, each is a 2D Conv layer with kernel of size 3×3 and 32 channels. The action classifier used with our spatial and temporal sampler is a three-head GRU, corresponding to the global features (low-res RGB and Spectrogram), local features (cropped high-res RGB and Spectrogram), and their concatenation to the primary GRU head. The goal of the multi-head architecture is to ensure the network extract prominent features from the cropped regions rather than relying solely on the low-res image. Each head of the GRU classifier and our GRU temporal sampler share the same architecture of 2 layers and 1024 hidden dimension.

The whole system is trained in multiple phases. We first train the two feature extraction modules with FC classifier, corresponding to the low-res and high-res inputs. We train the models with the standard cross-entropy loss for 100 epochs, using SGD with momentum of 0.9 [77], with decaying at epochs 30, 60, and 90. The weights of feature extraction modules are frozen and used for other models. The hallucinator is then trained using the belief loss $\mathcal{L}_b$ in Eq. (5) with teacher forcing routine [75]. In our experiments, we choose to set the decay factor of F_r as 0.95 and the warm-up t_{warm} as 5 frames. For the spatial sampler with three-head classifier, the predictions on all heads are averaged and the model is trained using the loss $\mathcal{L}_{class} = \sum_{h=1}^{3} \theta_h \mathcal{L}_h$, where $\mathcal{L}_h$ is the cross-entropy loss of a head and θ_h is the corresponding scaling. The temporal sampler is jointly trained with the pretrained three-head classifier and the fixed spatial sampler, using the total loss $\mathcal{L}_{class} + \theta_e \mathcal{L}_e$, where $\mathcal{L}_e$ is the efficiency loss described in Eq. (7) with the corresponding scaling θ_e. We train each

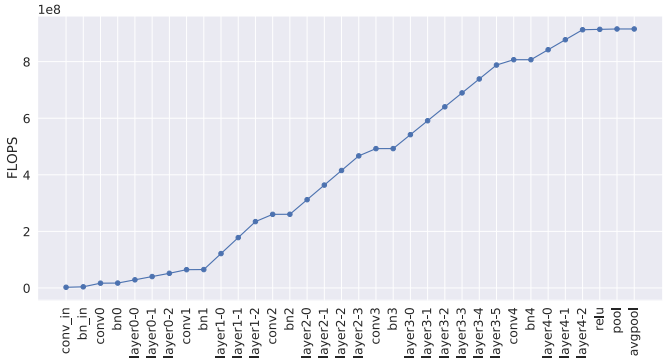


Fig. 13. Accumulated complexity of different layers from SAN19 model, where the input size is fixed as $3 \times 112 \times 112$.

sampling model for 50 epochs using Adam optimizer [78]. For feature extraction modules, we only sample three frames since we only aim to extract spatial features instead of temporal ones in this phase. As for the sampling modules, we use 10 frames for EPIC-KITCHENS and 16 frames for UCF-101 for better comparability with other frameworks.

B. Qualitative Results

We demonstrate our qualitative results from EPIC-KITCHENS dataset in Fig. 12 to show the outputs of both spatial and temporal samplers. The frames are uniformly sampled from the validation videos. We use "red" and "green' color to highlight full inference or simply pre-scanning. Unmarked frames are skipped without any computation. The spatial sampler only runs on "red" frames to enrich data, therefore the cropped regions are only available for these frames. We choose to use $\mathcal{S}_3$ and $\mathcal{T}_4$ in qualitative experiments since they provide more observable sampling results for visualization, regardless of their effect on the qualitative performance.

Overall, the temporal sampler can adaptively sample the frames. The number of "red" frames is fewer than the original

TABLE II

RESULTS OF SPATIAL SAMPLER $\mathcal{S}$ AND TEMPORAL SAMPLERS $\mathcal{T}$ ON EPIC-KITCHENS *Val-Set*. EACH BLOCK SHOW THE RESULTS CORRESPONDING TO THE TEMPORAL SAMPLER $\mathcal{T}_M$, WHERE M IS THE MAXIMUM NUMBER OF FRAMES THAT $\mathcal{T}$ ALLOWS TO SKIP. THE TABLE INCLUDES THE PERCENTAGE OF FRAMES BEING SKIPPED (*Skip %*), PRE-SCANNED (*Prescan (%)*), AND FULLY PROCESSED (*Full (%)*), THE ACCUMULATED TERA-FLOPS OVER THE WHOLE VALIDATION SET, AND THE AVERAGE COMPUTATION SAVING COMPARED TO ITS SPATIAL SAMPLER COUNTERPART. ALL MODELS HAVE TEMPORAL SAMPLING, EXCEPT FOR THE FIRST ROW, WHICH IS COPIED FROM TABLE I FOR COMPARISON. ALL TEMPORAL SAMPLERS SAVE THE COMPUTE COMPARED TO $\mathcal{S}_0$ WITH TOLERABLE LOSS OF ACCURACY

Model	Skip (%)	Prescan (%)	Full (%)	Total TFLOPS	Avg GFLOPS	Top-1	Top-5	Verb Top-1	Verb Top-5	Noun Top-1	Noun Top-5	Trade -off	Speed up
$\mathcal{S}_0$	0.00	0.00	100.00	139.14	5.80	24.56	53.34	53.50	83.95	34.57	58.84	0.236	-
$\mathcal{S}_0, \mathcal{T}_1$	0.00	41.96	58.04	86.59	3.61	22.81	52.29	52.00	83.07	32.90	**57.92**	0.158	1.60x
$\mathcal{S}_1, \mathcal{T}_1$	0.00	49.17	50.83	**81.27**	**3.39**	22.98	51.63	53.25	83.07	33.15	57.30	**0.147**	1.71x
$\mathcal{S}_2, \mathcal{T}_1$	0.00	49.96	50.04	83.96	3.50	23.52	52.09	53.34	82.32	32.90	**57.92**	0.149	1.76x
$\mathcal{S}_3, \mathcal{T}_1$	0.00	**50.00**	**50.00**	87.47	3.66	**24.06**	**52.88**	**54.17**	**83.70**	**33.74**	**57.92**	0.152	**1.77x**
$\mathcal{S}_0, \mathcal{T}_2$	14.05	52.34	33.61	**53.82**	**2.24**	**22.52**	50.75	51.59	82.61	32.15	56.84	**0.100**	2.58x
$\mathcal{S}_1, \mathcal{T}_2$	14.35	51.58	34.07	56.91	2.37	21.94	51.50	51.50	**83.32**	**33.24**	56.88	0.108	2.44x
$\mathcal{S}_2, \mathcal{T}_2$	12.74	52.18	35.08	61.11	2.55	22.23	**51.92**	**51.92**	83.28	32.03	**57.59**	0.115	2.42x
$\mathcal{S}_3, \mathcal{T}_2$	**15.02**	**52.70**	**32.28**	59.27	2.47	21.23	51.50	48.92	83.28	32.36	56.88	0.116	**2.62x**
$\mathcal{S}_0, \mathcal{T}_3$	**25.99**	48.36	25.65	**42.19**	**1.76**	20.64	49.37	49.21	82.40	30.28	55.00	**0.085**	**3.29x**
$\mathcal{S}_1, \mathcal{T}_3$	25.46	48.54	25.99	44.63	1.86	21.06	50.29	**50.00**	82.99	**31.86**	56.09	0.088	3.11x
$\mathcal{S}_2, \mathcal{T}_3$	25.75	48.60	**25.64**	46.04	1.92	**21.23**	**51.71**	49.57	**83.28**	31.23	**57.51**	0.090	3.21x
$\mathcal{S}_3, \mathcal{T}_3$	25.18	**48.92**	25.89	48.41	2.02	20.73	50.67	49.99	83.20	31.19	56.24	0.097	3.21x
$\mathcal{S}_0, \mathcal{T}_4$	34.80	44.65	**20.55**	**34.58**	**1.44**	18.85	48.83	47.16	81.78	29.11	54.67	**0.077**	**4.01x**
$\mathcal{S}_1, \mathcal{T}_4$	32.06	**46.16**	21.78	38.12	1.59	18.89	49.12	**48.79**	81.90	30.15	55.34	0.084	3.64x
$\mathcal{S}_2, \mathcal{T}_4$	**35.53**	40.03	24.44	43.05	1.80	**19.35**	**49.42**	48.29	**82.03**	**30.61**	**55.76**	0.093	3.43x
$\mathcal{S}_3, \mathcal{T}_4$	34.63	44.52	20.85	39.66	1.65	18.56	47.50	47.00	81.61	29.36	54.34	0.089	3.92x

TABLE III

ACCURACY AND SPEED-UP FACTORS ACROSS DIFFERENT SPATIAL SAMPLERS $\mathcal{S}$ AND TEMPORAL SAMPLERS $\mathcal{T}$ ON EPIC-KITCHENS VAL-SET. EACH COLUMN INDICATES NUMBER OF REGIONS k FOR SPATIAL SAMPLER $\mathcal{S}_k$ WHILE EACH ROW DESCRIBES THE SAMPLING RANGE M FOR TEMPORAL SAMPLER $\mathcal{T}_M$. EACH CELL IS A PAIR OF TOP-1 ACCURACY AND SPEED-UP TIME CORRESPONDING TO A SPATIOTEMPORAL SETTING. USING SPATIAL SAMPLER IMPROVES THE ACCURACY, BUT REQUIRES MORE COMPLEXITY. HIGHER TEMPORAL SAMPLING RANGE M CORRESPONDS TO MORE SPEED-UP, BUT ALSO SACRIFICES MORE ACCURACY

	$\mathcal{S}_0$	$\mathcal{S}_1$	$\mathcal{S}_2$	$\mathcal{S}_3$
$\mathcal{T}_1$	**22.81**, 1.60x	**22.98**, 1.71x	**23.52**, 1.76x	***24.06***, 1.77x
$\mathcal{T}_2$	22.52, 2.58x	21.94, 2.44x	22.23, 2.42x	21.23, 2.62x
$\mathcal{T}_3$	20.64, 3.29x	21.06, 3.11x	21.23, 3.21x	20.73, 3.21x
$\mathcal{T}_4$	18.85, ***4.01x***	18.89, **3.64x**	19.35, **3.43x**	18.56, **3.92x**

video length and can compactly describe the complete action. In Fig. 12a, the action cutting squash is a simple example since it can be easily represented using a single frame. We see that aside from the warming up first frame, the temporal sampler here only pre-scans three frames and skip the rest of them. The action of putting down a squash in Fig. 12b is another interesting example, where the first and last frame are selected, corresponding to when the actor is holding the squash in hand and placing it on the chopping board. These two sampled frames concisely represent the action putting down is reality. A similar example is illustrated in Fig. 12c, where the actor is putting down a chopping board. Fig. 12d depicts a more challenging video sequence of opening a container, as the background is not informative and the container are not opened until the final frame, resulting in more consecutive pre-scanned frames.

C. Quantitative Results

Table I shows the quantitative results of our spatial sampler on EPIC-Kitchens. The first two rows are TBN [41] and SAN19-baseline with FC classifier, both use high-res RGB (224×224) and Spectrogram (256×256). Since TBN relies on Inception backbone, its model complexity is not directly comparable with our experiments, with use SAN19 backbone. However, our baseline model provides comparable accuracy. Since the main objective of the paper is to increase efficiency of a given model, we focus on comparing performance and complexity with the baseline SAN19.

The rest of Table I shows our results of the spatial sampler with GRU classifier, using the low-res RGB (112×112), cropped high-res RGB (64×64), and the same Spectrogram inputs. We denote $\mathcal{S}_k$ as our spatial sampler with top-k RoIs, where $\mathcal{S}_0$ means no spatial sampling involved. It can be seen that by simply decreasing the image resolution, $\mathcal{S}_0$ can reduce the complexity by 2.84 GFLOPS with a small loss of 2.96% in accuracy. By adding the sampled regions from the spatial sampler $\mathcal{S}_1$, $\mathcal{S}_2$, and $\mathcal{S}_3$, the accuracy is consistently improved. We achieve the best performance with $\mathcal{S}_3$ across our spatial sampling experiments. This gets close to the performance of the baseline, with 1.75% accuracy different but still can save 1.84 GFLOPS of computation. Furthermore, the efficiency trade-off of $\mathcal{S}_3$ is lower, showing that the model with spatial sampler is more efficient in terms of average GFLOPS per accuracy.

Table II shows our results with both spatial and temporal samplers. For convenient comparison, we include the results of

TABLE IV

RESULTS OF BASELINES AND SPATIAL SAMPLERS $\mathcal{S}$ ON UCF-101 (*Split-1*) FOR RGB INPUTS. WE EVALUATE OUR SYSTEMS USING TOP-1 AND TOP-5 ACCURACY, TOGETHER WITH AVERAGE GFLOPS PER-FRAME AND TRADE-OFF FACTORS, SIMILAR TO TABLE I. WE RETRAIN TSN [43] AND ACHIEVE COMPARABLE ACCURACY WITH OUR BASELINE SAN19. OTHER INCLUDED MODELS SHOULD BE COMPARED THROUGH THE TRADE-OFF FACTOR BECAUSE OF DIFFERENT BACKBONES. OVERALL, $\mathcal{S}_2$ HAS THE BEST ACCURACY AMONG ALL SPATIAL SAMPLERS WITH SIGNIFICANTLY LESS GFLOPS THAN THE BASELINES

Model	Backbone	Size	Avg GFLOPS	Top-1	Top-5	Trade-off
Res3D [79]	Res3D-18	224	19.3	85.8	-	0.225
DSN [80]	Res2D-18+ Res3D-18	224	20.36	86.5	-	0.235
SMART [81]	Res152	224	14.12	75.5	-	0.187
TSN [43](ret)	Res50	224	4.12	80.94	95.66	0.051
SAN19-base	Res50	224	3.75	80.57	94.08	0.047
$\mathcal{S}_0$	SAN19	112	**0.90**	69.81	90.11	**0.013**
$\mathcal{S}_1$	SAN19	112	1.23	72.14	90.59	0.017
$\mathcal{S}_2$	SAN19	112	1.55	**72.19**	90.62	0.021
$\mathcal{S}_3$	SAN19	112	1.87	72.03	**91.15**	0.026

$\mathcal{S}_0$ from Table I with its accumulated TFLOPS over the whole validation set. We observe no skipping frames in $\mathcal{T}_1$ because this model only allows either pre-scanning or full-inference. Each block in the table shows the results for a different temporal sampler $\mathcal{T}_M$, where M determines the sampling range, *ie.*, the maximum number of frames to skip. Table III summarizes the performance across different choices of spatial and temporal samplers. Each pair of items in the table represent top-1 accuracy and speed-up time corresponding to a set of spatiotemporal sampling parameters. Compared to the spatial-sampler-only counterparts, $\mathcal{T}_4$ can reduce the complexity up to 4.01 times by sacrificing more accuracy. On the other end of the spectrum, $\mathcal{T}_1$ can approximate the original accuracy, and still with speed-up time up to 1.77x.

Notice that we do not compare against uniform temporal sampling routines in our experiments. Since the frames are already uniformly sampled (to 10 frames in EPIC-KITCHENS), the closest comparison between a naive sampling and our sampling routine is to compare $\mathcal{S}_0$ with $(\mathcal{S}_0, \mathcal{T}_1)$ in Table II, where $\mathcal{S}_t$ means no spatial sampling and $\mathcal{T}_1$ means whether to skip one (more) frame. This results in a slight drop in accuracy and reduction in compute. We acknowledge this is not the ideal comparison because these two models still have different sampling rates, but the performance gap here is not significant. Furthermore, such drop in accuracy is expected as the design of our system is not to run the temporal sampler alone. In fact, our temporal sampler is meant to run in conjunction with the spatial sampler (temporal sampler is trained on top of spatial sampler). It is because temporal sampler aims to reduce compute and spatial sampler aims to compensate for the accuracy drop, as seen in $(\mathcal{S}_3, \mathcal{T}_1)$ in Table I.

We report the results of UCF-101 in Table IV and Table V, following similar convention as in Table I and Table II. We reproduce the results of TSN [43] on split-1 of the dataset using our hardware for more comparable results. Behaviors similar to EPIC-Kitchens are observed in this dataset, *ie.*, the both spatial and temporal samplers can reduce complexity while maintaining comparable accuracy. $\mathcal{S}_3, \mathcal{T}_1$ has the highest speed-up with only 0.82 loss of top-1 accuracy and $\mathcal{S}_1, \mathcal{T}_1$ has the best accuracy with 1.37x speed-up.

TABLE V

RESULTS OF BASELINES AND SPATIAL AND TEMPORAL SAMPLERS ON UCF-101 (*Split-1*). WE ACHIEVE THE BEST RESULTS BY COMBINING $\mathcal{S}_3$ WITH $\mathcal{T}_1$. THE SPATIOTEMPORAL SAMPLING ROUTINE PROVIDES GOOD SPEED UP COMPARED TO THEIR SPATIAL-SAMPLER-ONLY COUNTERPARTS, WHILE STILL RETAINING COMPARABLE ACCURACY

Model	Total TFLOPS	Avg GFLOPS	Top-1	Top-5	Trade-off	Speed-up
$\mathcal{S}_0, \mathcal{T}_1$	**44.56**	**0.74**	70.21	89.96	**0.010**	1.23x
$\mathcal{S}_1, \mathcal{T}_1$	54.61	0.90	**71.43**	90.48	0.013	1.37x
$\mathcal{S}_2, \mathcal{T}_1$	64.48	1.07	71.19	90.72	0.015	1.46x
$\mathcal{S}_3, \mathcal{T}_1$	74.50	1.23	71.21	**90.75**	0.017	**1.52x**

V. CONCLUSION AND FUTURE WORK

We introduce an attention-based spatiotemporal sampling scheme to adaptively sample videos for efficient action recognition. Spatial sampler provides a global view at low-res and local salient views at high-res. Temporal sampler pre-scans and decides the sampling strategy by comparing the current attention with the past hallucination. In the scope of this paper, we aim to verify the feasibility of our sampling routine on top of a fixed backbone that supports self-attention mechanism. However, simpler approaches using different backbones may produce competitive performance and complexity, *eg.*, TBN model shows potential for further compute reduction and improved accuracy, as seen in Table I. Aside from backbones, different mechanisms to generate and use attention maps could also further improve the overall performance and efficiency. We leave the investigation into such moving parts as some of the potential directions for future exploration.

ACKNOWLEDGMENT

This work is included as Chapter 5 in Khoi-Nguyen C. Mac's Doctoral Dissertation: "Learning efficient temporal information in deep networks: From the viewpoints of applications and modeling" [1]. The work of Khoi-Nguyen C. Mac was done prior to joining Amazon.

REFERENCES

[1] K.-N. C. Mac, "Learning efficient temporal information in deep networks: From the viewpoints of applications and modeling," Ph.D. dissertation, Dept. Elect. Comput. Eng., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, 2021.

[2] L. G. Appelbaum and A. M. Norcia, "Attentive and pre-attentive aspects of figural processing," *J. Vis.*, vol. 9, no. 11, p. 18, Oct. 2009.

[3] M. Atienza, J. L. Cantero, and C. Escera, "Auditory information processing during human sleep as revealed by event-related brain potentials," *Clin. Neurophysiol.*, vol. 112, no. 11, pp. 2031–2045, Nov. 2001.

[4] X. Meng and Z. Wang, "A pre-attentive model of biological vision," in *Proc. IEEE Int. Conf. Intell. Comput. Intell. Syst.*, vol. 3, Nov. 2009, pp. 154–158.

[5] L. Barghout-Stein, *On Differences Between Peripheral and Foveal Pattern Masking*. Berkeley, CA, USA: Univ. California, 1999.

[6] S. A. Klein, T. Carney, L. Barghout-Stein, and C. W. Tyler, "Seven models of masking," *Proc. SPIE*, vol. 3016, pp. 13–24, Jun. 1997.

[7] M. Iwasaki and H. Inomata, "Relation between superficial capillaries and foveal structures in the human retina," *Investigative Ophthalmol. Vis. Sci.*, vol. 12, pp. 1698–1705, Dec. 1986.

[8] H. Kolb, "Simple anatomy of the retina," in *Webvision: The Organization of the Retina and Visual System*. Salt Lake City, UT, USA: Univ. of Utah Health Sciences Center, 2011. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK11533/

[9] Z. Kourtzi and N. Kanwisher, "Cortical regions involved in perceiving object shape," *J. Neurosci.*, vol. 20, no. 9, pp. 3310–3318, May 2000.

[10] B. Gong, W.-L. Chao, K. Grauman, and F. Sha, "Diverse sequential subset selection for supervised video summarization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 2069–2077.

[11] M. Gygli, H. Grabner, and L. Van Gool, "Video summarization by learning submodular mixtures of objectives," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3090–3098.

[12] B. Korbar, D. Tran, and L. Torresani, "SCSampler: Sampling salient clips from video for efficient action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6231–6241.

[13] B. Mahasseni, M. Lam, and S. Todorovic, "Unsupervised video summarization with adversarial LSTM networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2982–2991.

[14] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2718–2726.

[15] K. Zhang, W.-L. Chao, F. Sha, and K. Grauman, "Video summarization with long short-term memory," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 766–782.

[16] (2021). *Google Glass*. Accessed Nov. 1, 2021. [Online]. Available: https://www.google.com/glass/start/

[17] (2021). *Microsoft Hololens*. Accessed Nov. 1, 2021. [Online]. Available: https://www.microsoft.com/en-us/hololens/

[18] Rayban. (2021). *Ray-Ban*. Accessed Nov. 1, 2021. [Online]. Available: https://www.ray-ban.com/

[19] D.-A. Huang et al., "What makes a video a video: Analyzing temporal information in video understanding models and datasets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7366–7375.

[20] R. Gao, B. Xiong, and K. Grauman, "Im2Flow: Motion hallucination from static images for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5937–5947.

[21] J. Guan, Y. Yuan, K. M. Kitani, and N. Rhinehart, "Generative hybrid representations for activity forecasting with no-regret learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 173–182.

[22] Y. Meng et al., "AR-Net: Adaptive frame resolution for efficient action recognition," in *Proc. Eur. Conf. Comput. Vis.*, Nov. 2020, pp. 86–104.

[23] D. Damen et al., "Scaling egocentric vision: The EPIC-KITCHENS dataset," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2018, pp. 720–736.

[24] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.

[25] M. A. Giese and T. Poggio, "Neural mechanisms for the recognition of biological movements," *Nature Rev. Neurosci.*, vol. 4, no. 3, pp. 179–192, Mar. 2003.

[26] R. Sigala, T. Serre, T. Poggio, and M. Giese, "Learning features of intermediate complexity for the recognition of biological motion," in *Proc. 15th Int. Conf. Artif. Neural Netw., Biol. Inspirations (ICANN)*. Warsaw, Poland: Springer, Sep. 2005, pp. 241–246.

[27] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A biologically inspired system for action recognition," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[28] M.-J. Escobar, G. S. Masson, T. Vieville, and P. Kornprobst, "Action recognition using a bio-inspired feedforward spiking network," *Int. J. Comput. Vis.*, vol. 82, no. 3, pp. 284–301, May 2009.

[29] M.-J. Escobar and P. Kornprobst, "Action recognition via bio-inspired features: The richness of center–surround interaction," *Comput. Vis. Image Understand.*, vol. 116, no. 5, pp. 593–605, May 2012.

[30] B. Yousefi, C. K. Loo, and A. Memariani, "Biological inspired human action recognition," in *Proc. IEEE Workshop Robotic Intell. Informationally Structured Space (RiiSS)*, Apr. 2013, pp. 58–65.

[31] P. Antonik, N. Marsal, D. Brunner, and D. Rontani, "Human action recognition with a large-scale brain-inspired photonic computer," *Nature Mach. Intell.*, vol. 1, no. 11, pp. 530–537, Nov. 2019.

[32] J. Chen, R. D. J. Samuel, and P. Poovendran, "LSTM with bio inspired algorithm for action recognition in sports videos," *Image Vis. Comput.*, vol. 112, Aug. 2021, Art. no. 104214.

[33] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *Proc. 17th Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2004, pp. 32–36.

[34] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2247–2253, Dec. 2007.

[35] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. IEEE Int. Workshop Vis. Surveill. Perform. Eval. Tracking Surveill.*, 2005, pp. 65–72.

[36] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inform. Process. Syst.*, 2014, pp. 1–9.

[37] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[38] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733.

[39] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.

[40] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.

[41] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "EPIC-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5491–5500.

[42] S. Sudhakaran, S. Escalera, and O. Lanz, "LSTA: Long short-term attention for egocentric action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 9946–9955.

[43] L. Wang et al., "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 20–36.

[44] R. Possas, S. P. Caceres, and F. Ramos, "Egocentric activity recognition on a budget," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5967–5976.

[45] B. Xu, X. Shu, J. Zhang, G. Dai, and Y. Song, "Spatiotemporal decouple-and-squeeze contrastive learning for semisupervised skeleton-based action recognition," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 10, 2023, doi: 10.1109/TNNLS.2023.3247103.

[46] B. Xu and X. Shu, "Pyramid self-attention polymerization learning for semi-supervised skeleton-based action recognition," 2023, *arXiv:2302.02327*.

[47] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence constrained graph LSTM for group activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 636–647, Feb. 2022.

[48] X. Shu, L. Zhang, G.-J. Qi, W. Liu, and J. Tang, "Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 6, pp. 3300–3315, Jun. 2022.

[49] X. Shu, J. Tang, G.-J. Qi, W. Liu, and J. Yang, "Hierarchical long short-term concurrent memory for human interaction recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 1110–1118, Mar. 2021.

[50] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 646–661.

[51] X. Wang, F. Yu, Z.-Y. Dou, T. Darrell, and J. E. Gonzalez, "SkipNet: Learning dynamic routing in convolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 409–424.

[52] Z. Wu et al., "BlockDrop: Dynamic inference paths in residual networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8817–8826.

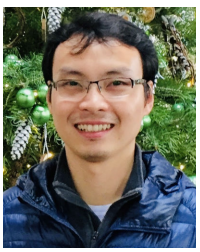
[53] Y. Wang, F. Sun, D. Li, and A. Yao, "Resolution switchable networks for runtime efficient image recognition," 2020, *arXiv:2007.09558*.

[54] B. Uzkent and S. Ermon, "Learning when and where to zoom with deep reinforcement learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12342–12351.

[55] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6201–6210.
[56] F. Xiao, Y. Jae Lee, K. Grauman, J. Malik, and C. Feichtenhofer, "Audiovisual SlowFast networks for video recognition," 2020, *arXiv:2001.08740*.
[57] Y. Wang, Z. Chen, H. Jiang, S. Song, Y. Han, and G. Huang, "Adaptive focus for efficient video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16229–16238.
[58] H. Kim, M. Jain, J.-T. Lee, S. Yun, and F. Porikli, "Efficient action recognition via dynamic knowledge propagation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13699–13708.
[59] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 31, 2018, pp. 1–11.
[60] M. T. Ribeiro, S. Singh, and C. Guestrin, "'Why should I trust you?' Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.
[61] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[62] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 818–833.
[63] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inform. Process. Syst.*, 2017, pp. 5998–6008.
[64] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10073–10082.
[65] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in *Proc. Adv. Neural Inform. Process. Syst.*, vol. 32, 2019, pp. 68–80.
[66] G. Ren, T. Dai, P. Barmpoutis, and T. Stathaki, "Salient object detection combining a self-attention module and a feature pyramid network," *Electronics*, vol. 9, no. 10, p. 1702, Oct. 2020.
[67] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–21.
[68] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6816–6826.
[69] H. Yin, A. Vahdat, J. M. Alvarez, A. Mallya, J. Kautz, and P. Molchanov, "A-ViT: Adaptive tokens for efficient vision transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 10799–10808.
[70] L. Meng et al., "AdaViT: Adaptive vision transformers for efficient image recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12299–12308.
[71] C.-Y. Wu et al., "MeMViT: Memory-augmented multiscale vision transformer for efficient long-term video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 13577–13587.
[72] Y. Li et al., "MViTv2: Improved multiscale vision transformers for classification and detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 4794–4804.
[73] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, 2003, pp. 1398–1402.
[74] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 1–9.
[75] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.
[76] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with Gumbel–Softmax," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–13.
[77] N. Qian, "On the momentum term in gradient descent learning algorithms," *Neural Netw.*, vol. 12, no. 1, pp. 145–151, Jan. 1999.
[78] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
[79] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri, "ConvNet architecture search for spatiotemporal feature learning," 2017, *arXiv:1708.05038*.
[80] Y.-D. Zheng, Z. Liu, T. Lu, and L. Wang, "Dynamic sampling networks for efficient action recognition in videos," *IEEE Trans. Image Process.*, vol. 29, pp. 7970–7983, 2020.
[81] S. N. Gowda, M. Rohrbach, and L. Sevilla-Lara, "Smart frame selection for action recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 2, pp. 1451–1459.

Khoi-Nguyen C. Mac received the B.S. degree from the University of Science, Vietnam National University, in 2011, the M.E. degree in multimedia communication systems from Eurecom, Télécom ParisTech, in 2014, and the Ph.D. degree from the Department of Electrical and Computer Engineering (ECE), University of Illinois at Urbana–Champaign (UIUC), in 2021. He is currently a Research Scientist with Amazon. His research interests include improving the efficiency of temporal information modeling in deep networks across different domains, including vision and audio. Specifically, he is interested in applications in the area of video analysis, such as action recognition, detection, and sampling.

Minh N. Do (Fellow, IEEE) is currently the Thomas and Margaret Huang Endowed Professor with the Department of Electrical and Computer Engineering and holds affiliate appointments with the Coordinated Science Laboratory, Beckman Institute for Advanced Science and Technology, Department of Bioengineering, Department of Computer Science, and College of Medicine, University of Illinois at Urbana–Champaign, where he has been a Faculty Member since 2001. From 2020 to 2021, he was the Vice Provost and then continues as an Honorary Vice Provost with VinUniversity, the first private not-for-profit Vietnamese university established based on international standards. He has contributed to several tech-transfer efforts, including as a Co-Founder and CTO of Personify andChief Scientist of Misfit. His current research interests include signal processing, computational imaging, geometric vision, data science, and smart health. More information can be found at: minhdo.ece.illinois.edu.



Minh P. Vo received the Ph.D. degree from The Robotics Institute, Carnegie Mellon University. He is currently the Head of Machine Learning with Spree3D, a high-tech fashion eCommerce startup. His research interests include developing large-scale dynamic human scene understanding systems in order to create virtual environments that are perceptually indistinguishable from reality. He was a recipient of the Qualcomm Innovation Fellowship.