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...
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 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 Rols 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 neuro-biological model of motion processing and implements a simple RNN. More recently, [31] introduces a photonic hardware approach and implements a simple RNN 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 stochastic method to drop layers during the training phase. SkipNet [51] and BlockDrop [52] later propose 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 $D = \{(v_n, y_n)\}_{n=1}^N$, where $v_n$ is a video sequence and $y_n$ is the corresponding groundtruth label. We assume that the video sequences have the same length of $T$ frames, i.e., $v_n = [x_n^{(1)}, x_n^{(2)}, \ldots, x_n^{(T)}]$, where each frame is $x_n^{(t)} \in \mathbb{R}^{2 \times H \times W}$, $\forall t \in \{1, \ldots, T\}$. Suppose that we have a video classifier $F(v_n) = \hat{y}_n$, with some complexity $O_F$. The goal is to construct another classifier $\hat{F}$ with less complexity while retaining the accuracy. We address this by introducing the temporal sampler $T$ and spatial sampler $S$, i.e., $\hat{y}_n = F(x_n; T, S)$, such that $O_{\hat{F}} < O_F$. 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 attention 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...
We average across the channel dimension for visualization. Hotter color indicates more salient regions.

where \( R \) compatibility function is locally defined over the footprint \( \mathcal{R}(i) \). Such compatibility function, usually defined as a softmax. Such clarification in the updated manuscript.

Therefore, we aggregate them as \( A = \{ a_i \} \), where \( a_i \) 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

\[
A = \sum_i a_i \otimes \mathbb{1}[\mathcal{R}(i)],
\]

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 \( a_i \) with the corresponding footprint. Notice that \( a_i \) has the same spatial dimension as \( \mathcal{R}(i) \), while \( A \) has the same spatial dimension as the input feature map \( \phi(x) \). 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 \( a_i \). Therefore, we aggregate them as \( A \) and learn to generate its future version using the hallucinator. We have included this clarification in the updated manuscript.

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 \( a_i \) are overlapping (because of stride of 1 in this example), similar to convolution. By using \( A \), we avoid having to encode such overlapping conditions, making it easier to learn. Specifically, the activation of \( A \) 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 \( x \in \mathbb{R}^{3 \times H \times W} \), we compute the corresponding low and high-res inputs \( x_l \in \mathbb{R}^{3 \times \frac{H}{4} \times \frac{W}{4}} \) and \( x_h \in \mathbb{R}^{3 \times H' \times W'} \), respectively obtained by rescaling (with the down-sampling factor \( d \)) and cropping \( x \) \((H' < H, W' < W)\) at \( k \) different locations. While \( d \) is defined as our hyper-parameter, the cropping regions for \( x_{h,k} \) are computed by the spatial sampler \( S \). Giving attention \( A \), 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 \( x \) and the attention \( A \).

Fig. 7 shows the details of the spatial sampler. We extract the attention from the low-res image \( x_l \) and use it to sample the top-\( k \) regions in the original image \( x \). This results in \( x_{h,k} \) with lower spatial dimension, while retaining the original resolution of \( x \). 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 \( x_l \) and \( x_{h,k} \)’s is less than that of \( x \). 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
than RGB frames. Assuming the temporal consistency, important regions of the inputs and can be simpler to generate need to run further inference. Instead of generating the whole information. If our prediction is similar to the actual observation

C. Hallucinator

actions are occurring slowly, suggesting that we can predict future attentions in such cases.

Fig. 7. Spatial sampler uses attention from low-res image to sample the top-\(k\) regions from the (original) high-res input. \(x_h\) gives a global view, while \(x_{h,k}\) provides local views at important regions of the original image \(x\). 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.

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.

D. Temporal Sampler

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

Hallucinator \(\mathcal{H}\) as:

\[
\tilde{A}^{(t+1)} = \mathcal{H}(A^{(t)}) \quad \text{s.t.} \quad \tilde{A}^{(t+1)} \approx A^{(t+1)},
\]

where \(\tilde{A}^{(t+1)}\) is the hallucination (predicted future attention). To quantify the similarity between \(\tilde{A}^{(t+1)}\) and \(A^{(t+1)}\), 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:

\[
\mathcal{L}_b = - \frac{1}{T-1} \sum_{t=2}^{T} \text{SSIM}(\mathcal{H}(\tilde{A}^{(t-1)}), A^{(t)}),
\]

where the function \(\text{SSIM}(\cdot)\) computes the structural similarity between the hallucination \(\mathcal{H}(A^{(t-1)})\) and the attention \(A^{(t)}\).

Fig. 8 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.
and hallucination at any arbitrary layer from a model of $L$ layers. Suppose that the attention is extracted at layer $\lambda \leq L$, 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 $\lambda$ 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 $\lambda \in \{1, \ldots, L\}$, i.e., $\phi_{\lambda}^L(x) = \phi_{\lambda}^L(\phi_{\lambda}^L(x))$. The first half $\phi_{\lambda}^L$ is used for pre-scanning while the second half $\phi_{\lambda}^L$ can also be augmented with information from other modalities for the classification task later. The temporal sampler $T$ determines the sampling routine by computing a sampling vector $r(t)$, $\forall t \in \mathbb{N}$, with $r(t) \in \{0, 1\}^{M+1}$, where $r(t)[m] = 1$ 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 $A(t)$ is extracted using the first half of the feature extractor $\phi_{\lambda}^L$. To generate the sampling vector $r(t)$, 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 $m^* = \arg\max_m r(t)[m]$, 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 $O_{\text{full}}$. 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 $O_{\text{pre}} = O_{\phi_{\lambda}^L} + O_H + O_T$, being the model’s first half, hallucinator, and temporal sampler. Note
that $O_{\text{full}} = O_{\text{pre}} + O_{\text{rest}}$, where $O_{\text{rest}}$ 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 $L_{\text{class}}$ (only using full-inference frames) and efficiency loss $L_e$, which is defined as:

$$L_e = n_{\text{full}} \cdot O_{\text{full}} + n_{\text{pre}} \cdot O_{\text{pre}},$$

(7)

where $n_{\text{full}}$ and $n_{\text{pre}}$ 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, i.e., $\arg\max_r r^{(t)}[m] \neq 0, \forall t$. 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, e.g., [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 $3 \times 112 \times 112$ 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
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 × 7 × 7. The hallucinator is a Conv-LSTM with 1 layer and 32 hidden dimensions. It is equipped with an 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 \( t_{\text{warm}} \) 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 \( \mathcal{L}_{\text{class}} = \sum_{h=1}^{3} \theta_h \mathcal{L}_b \), where \( \mathcal{L}_b \) is the cross-entropy loss of a head and \( \theta_h \) 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 \( \mathcal{L}_{\text{class}} + \theta_e \mathcal{L}_e \), where \( \mathcal{L}_e \) is the efficiency loss described in Eq. (7) with the corresponding scaling \( \theta_e \). We train each 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 \( S_3 \) and \( T_3 \) 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 I

| 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  | Res101,8x8    | K               | 224      | 13.23      | 21.9  | 39.7  | 53.8       | 83.1       | 27.4       | 52.1       | 0.605     |
| AVSlowFast| Res101,8x8    | R+A             | 224      | 16.13      | 24.2  | 43.6  | 58.7       | 83.6       | 31.7       | 58.4       | 0.667     |
| TBN [41]  | Inception     | R+A+F           | 224      | 6.95       | 34.83 | 54.09 | 63.31      | 88.29      | 46.00      | 68.31      | 0.200     |
| TBN [41]  | Inception     | R+A             | 224      | 4.62       | 28.32 | 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     |

---

---

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
A challenging video sequence of opening a container, as the actor is putting down a chopping board. Fig. 12d depicts a more interesting example, where the actor is holding the squash in hand and placing it on the chopping board. These two sampled frames correspond to when the actor is holding the squash in hand 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 $S_k$ as our spatial sampler with top-$k$ RoIs, where $S_0$ means no spatial sampling involved. It can be seen that by simply decreasing the image resolution, $S_0$ 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 $S_1$, $S_2$, and $S_3$, the accuracy is consistently improved. We achieve the best performance with $S_3$ across our spatial sampling experiments. This gets close to the performance of the baseline, with 1.75% accuracy difference but still can save 1.84 GFLOPS of computation. Furthermore, the efficiency trade-off of $S_3$ 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 top-3 RoIs, the accuracy is consistently improved. 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...
TABLE IV
RESULTS OF BASELINES AND SPATIAL SAMPLERS \( S \) 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, \( S_0 \) 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  | Res3D-18 | 224      | 19.3         | 85.8  | -         | 0.225 |
| DSN    | Res2D-18+ | 224  | 20.36      | 86.5  | -         | 0.235 |
| SMART  | Res152   | 224      | 14.12       | 75.5  | -         | 0.187 |
| TSN    | Res50    | 224      | 4.12        | 80.94 | 95.66     | 0.051 |
| SAN19-base | Res50  | 224 | 3.75  | 80.57 | 94.08     | 0.047 |
| \( S_0 \) | SAN19    | 112      | 0.90        | 69.81 | 90.11     | 0.013 |
| \( S_1 \) | SAN19    | 112      | 1.23        | 72.14 | 90.59     | 0.017 |
| \( S_2 \) | SAN19    | 112      | 1.55        | 72.19 | 90.62     | 0.021 |
| \( S_3 \) | SAN19    | 112      | 1.87        | 72.03 | 91.15     | 0.026 |

\( S_0 \) from Table I with its accumulated TFLOPS over the whole validation set. We observe no skipping frames in \( T_1 \) because this model only allows either pre-scanning or full-inference. Each block in the table shows the results for a different temporal sampler \( T_M \), where \( M \) determines the sampling range, i.e., 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, \( T_4 \) can reduce the complexity up to 4.01 times by sacrificing more accuracy. On the other hand of the spectrum, \( T_1 \) 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 \( S_0 \) with \( (S_0, T_1) \) in Table II, where \( S_0 \) means no spatial sampling and \( T_1 \) means whether to skip one (more) frame. This results in a slight drop in accuracy and complexity. 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 \( (S_3, T_1) \) 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, i.e., the both spatial and temporal samplers can reduce complexity while maintaining comparable accuracy. \( S_3, T_1 \) has the best speed-up with only 0.82 loss of top-1 accuracy and \( S_1, T_1 \) has the best accuracy with 1.37x speed-up.

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 prescans 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, e.g., 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.
