Prune Spatio-temporal Tokens by Semantic-aware Temporal Accumulation

Shuangrui Ding[1,3]* Peisen Zhao[2] Xiaopeng Zhang[2] Rui Qian[3] Hongkai Xiong[1] Qi Tian[2]†
[1]Shanghai Jiao Tong University [2]Huawei Cloud [3]The Chinese University of Hong Kong
{dsr1212, xionghongkai}@sjtu.edu.cn qr021@ie.cuhk.edu.hk
{pszhao93, zxphistory}@gmail.com tian.qi1@huawei.com

Abstract

*Transformers have become the primary backbone of the computer vision community due to their impressive performance. However, the unfriendly computation cost impedes their potential in the video recognition domain. To optimize the speed-accuracy trade-off, we propose **S**emantic-aware **T**emporal **A**ccumulation score (STA) to prune spatio-temporal tokens integrally. STA score considers two critical factors: temporal redundancy and semantic importance. The former depicts a specific region based on whether it is a new occurrence or a seen entity by aggregating token-to-token similarity in consecutive frames while the latter evaluates each token based on its contribution to the overall prediction. As a result, tokens with higher scores of STA carry more temporal redundancy as well as lower semantics thus being pruned. Based on the STA score, we are able to progressively prune the tokens without introducing any additional parameters or requiring further re-training. We directly apply the STA module to off-the-shelf ViT and VideoSwin backbones, and the empirical results on Kinetics-400 and Something-Something V2 achieve over* 30% *computation reduction with a negligible* $\sim 0.2\%$ *accuracy drop. The code is released at* https://github.com/Mark12Ding/STA.

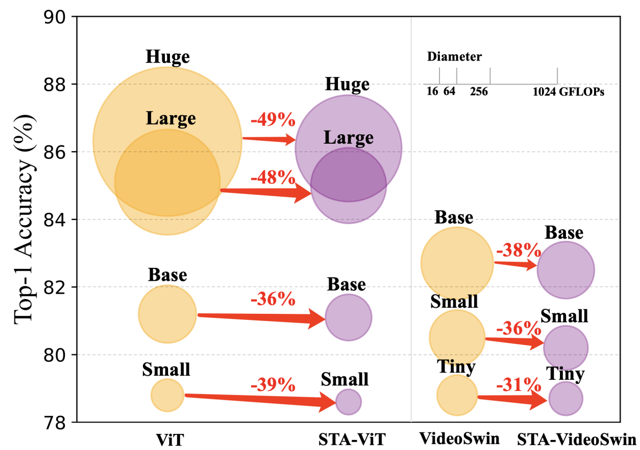


Figure 1: Kinectics-400 result for ViT and VideoSwin. The bubble's area is proportional to FLOPs of a variant in a model family. ViT/VideoSwin here takes $16/32 \times 224^2$ video. The proposed STA saves over 30% FLOPs for all model variants with a negligible drop in performance.

1. Introduction

Recently, there has been an unstoppable shift in the general backbone design from Convolutional Neural Networks (ConvNets) to Transformers, which are originally employed in natural language processing, and has shown promising potential for various vision tasks [8, 48, 47, 22, 1, 32, 5]. The key component of Transformers is the self-attention mechanism, which is apt to capture long-range dependencies and empowers ViT to perceive the global reception field. The seminal work, Vision Transformer (ViT) [8] closely follows the original Transformer architecture [38]. Equipped with a large-scale model and dataset, ViT outperforms ConvNets in image classification by a considerable margin. Inspired by this superior scaling behavior, Transformers have gained popularity as a backbone choice and are widely adopted for image recognition [22, 37], action recognition [1, 23], semantic segmentation [49, 6], action detection [43, 12], temporal perception [33, 34], *etc.*

Despite the promising potential of Transformers in spatio-temporal vision tasks, such as action recognition, the quadratic increase in complexity caused by the temporal dimension makes the video Transformers computationally unfriendly compared to images. For instance, earlier work TimeSformer [2], which applies Transformer backbone for video, required 7.14 Tera FLOPs to achieve 80.7% accuracy on the Kinetics-400 action recognition benchmark. The excessive computational cost makes it impractical for deployment in real-world scenarios. Therefore, there is an urgent need to explore ways to profit from the performance gains

*Work done during an internship at Huawei Cloud.
†Corresponding author. Email: tian.qi1@huawei.com.

of Transformers while maintaining an affordable computation burden.

Fortunately, Transformers can handle a flexible number of tokens as input. Recent attempts [31, 30, 42], that dynamically prune tokens for images, have remarkably reduced computation overhead. These pruning approaches have inspired us to explore token pruning in the video domain, so as to balance accuracy and computation costs accordingly. However, performing frame-wise pruning alone seems to be not optimal since it ignores temporal context and disrupts the dynamic structure of the video. To address this issue, recent work STTS [40] decouples token pruning into temporal and spatial dimensions. Specifically, STTS first drops meaningless frames and then filters out detail-rich regions from the remained frames. However, this spatio-temporal decoupling strategy lacks contextual modeling of continuous temporal information, leading to limited performance.

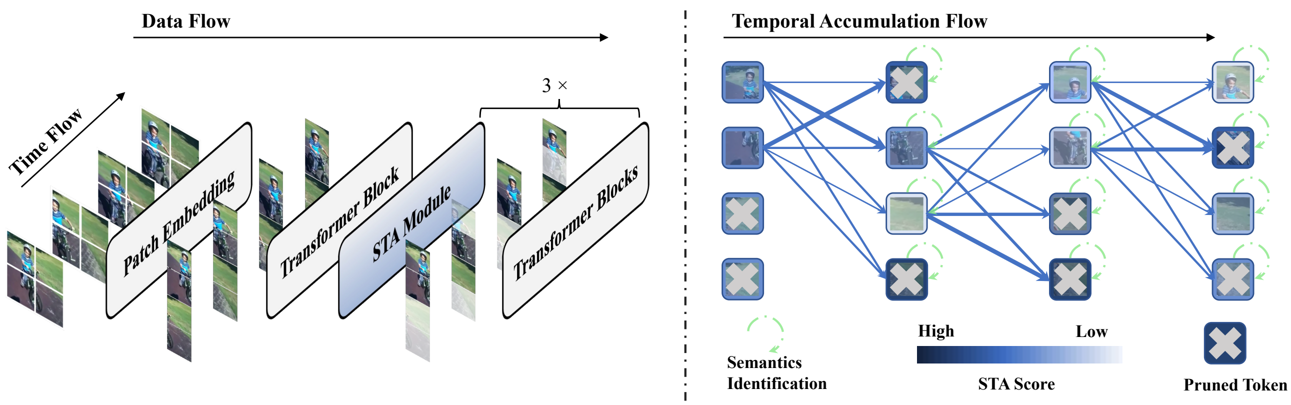
In this paper, we argue that pruning spatio-temporal tokens integrally can lead to further computation reduction at an acceptable cost of accuracy degradation. To this end, we propose the **S**emantic-aware **T**emporal **A**ccumulation (STA) score to depict the importance of each token. We take two factors into consideration, **temporal redundancy** and **semantic importance**. Our motivation is to discard tokens that have similar counterparts appearing earlier in the sequence while retaining only semantically significant tokens. As an example, static backgrounds across all timestamps contain highly repetitive information that is unnecessary to be included. Therefore, keeping only a few representative background patches is sufficient for reasoning. Specifically, we evaluate the temporal redundancy of a region by determining whether it is a novel or previously observed entity. In practice, we aggregate repetitive information on a per-frame basis and assign each token a score of temporal repetition degree. Nevertheless, there are cases where a repetitive region reveals a crucial action and should be retained. For example, if the sequence of tokens describes human-body motion, it is necessary to keep all the tokens for better understanding, even if there are only slight differences over time. Thus, we also take semantic importance into account during the pruning procedure. To depict each token's semantic contribution to video recognition, we take the summation of the feature activation map and then integrate this summation with the score of temporal aggregation to enhance the awareness of semantics. Based on the STA score, we progressively prune the tokens of the video Transformers three times. The whole pruning process does not introduce any tuning parameter and directly accelerates the off-the-shelf video Transformers without the need to retrain.

We apply our pruning strategy to two mainstream video Transformers, ViT [8] and VideoSwin [23], and evaluate 10 off-the-shelf backbones on two action recognition benchmarks, Kinetics-400 [14] and Something-Something V2 [11], to demonstrate the effectiveness of our method. As shown in Figure 1, we achieve significant computation reduction with a negligible accuracy drop on Kinetics-400. For instance, using ViT-H as the backbone, by hierarchically pruning 57% of the input tokens, STA reduces 49% FLOPs while the accuracy drop is only 0.2%. Besides, with STA, FLOPs of VideoSwin-B are decreased by 38% while maintaining 82.5% accuracy with only a minimal drop of 0.2%. A similar trend can also be observed in the Something-Something V2 dataset. Notably, we surpass STTS [40] by a 0.4% accuracy gain with 40% fewer FLOPs on Kinetics-400 and by a 0.5% accuracy increase with 20% fewer FLOPs on Something-Something V2 when using the same backbone.

2. Related Work

Video Transformers. Designing Transformer-based architectures for vision tasks has emerged as a general trend in the computer vision community, as evidenced by several recent works [8, 22, 50, 28, 49, 24, 39]. With an unprecedented number of parameters and millions of training data, Transformers significantly outperform prior arts spanning a variety of tasks, not only in image but also in video understanding tasks. Various variants of self-attention have been introduced in prior works [2, 26, 47, 1, 4, 28, 9, 23, 46, 19] to capture the spatiotemporal relationship. However, using pure patch-based Transformers incurs prohibitive costs on memory and computation when extracting global-range features from the whole video. To deal with it, Motionformer [28] introduces trajectory attention that focuses on implicitly determined motion paths and optimizes the quadratic calculation via efficient decomposition. MeMViT [43] proposes caching 'memory' of past frames and attending to the summarized prior context in an online manner. In this paper, we propose an orthogonal approach to make Transformers lighter by pruning the spatio-temporal tokens with high temporal redundancy.

Token pruning for Transformers. Several works [31, 30, 42, 20, 25, 3, 17] have focused on reducing the number of tokens involved in the calculation to accelerate image Transformer models. In specific, DynamicViT [30] trains a lightweight decision module to rate the importance of each token and prune low-score tokens progressively. EViT [20] preserves the attentive tokens guided by the class token attention and fuses inattentive ones without the help of any extra parameter. ToMe [3] combines similar tokens to directly expedite off-the-shelf ViT without needing to train. While similar to ToMe [3], our method functions as a simple plug-in to enhance off-the-shelf video Transformers without



(a) Conceptual pipeline of video Transformer with STA module. **(b) Illustration of the proposed STA module.**

Figure 2: An overview of our STA token-pruning algorithm for video Transformers. (a) STA module is a simple plug-in and can be inserted at the beginning or end of the Transformer block. In practice, we conduct a three-stage progressive strategy to prune the token with STA. (b) Our semantic-aware temporal accumulation algorithm. The wider arrows connecting two adjacent frames represent higher weight to transport the STA score.

requiring additional training. However, ToMe is an image-based pruning technique that does not model any temporal relations. In contrast, our method specifically devises a temporal aggregation mechanism tailored for video data. Although STTS [40] trains a score network to choose predefined anchors from filtered frames within a video, the empirical speed-accuracy trade-off remains limited. This is because decoupled anchor-level selection still retains unwanted spatial redundancy. On the contrary, we prune the video at the token level and eliminate the meaningless content by a large margin.

Efficient video recognition. Due to the nature of the extra dimension, video understanding is computationally intensive. Thus, there have been attempts [51, 10, 36, 44, 15, 41, 7, 18, 16, 21, 13, 29] to design lightweight modules that enjoy both high efficiency and high accuracy. ECO [51] introduces a network architecture to sample a small subset of frames and learns the temporal context between these frames. Besides, AdaFocus [41] improves the computational efficiency by adopting a light-weighted ConvNet to localize the most salient spatiotemporal regions. While previous works have mainly concentrated on accelerating ConvNet models, efforts toward accelerating video Transformers have been relatively sparse and open to exploration. A recent approach [27] devises a novel token-based sampling using k-centered search before feeding tokens into video Transformers. Although we also select semantically meaningful tokens for video Transformers, our dynamic model processes whole tokens at the early stage and prunes them based on model-dependent features.

3. Approach

The goal of this paper is to develop a principal token-pruning algorithm for video Transformers that achieves an optimal balance between cost, speed, and performance without requiring model re-training. We start by analyzing the video Transformers at hand and observe two interesting phenomena, detailed in Sec. 3.3. *First*, we find the high temporal redundancy when comparing the inter-frame similarity. *Second*, the area that contributes to the final prediction usually takes up a small portion. Motivated by these two findings, we carefully develop two principles to prune the tokens with high temporal redundancy and retain the meaningful tokens. The overall framework is shown in Figure 2(a). Our method is mainly built on the standard columnar Transformer [8], which we briefly go through in Sec. 3.1. Later, in Sec. 3.2, we elaborate on the proposed metric to help prune unnecessary tokens. Finally, Sec. 3.3 discusses why STA works in video Transformer.

3.1. Revisit of Video Transformer

Video Transformers generally process video data as a 1D sequence of tokens and directly model the relationship between them. Initially, video Transformer linearly projects 3D data tubes into high-dimensional embeddings. Assuming the dimensions of video clips as $\{T, H, W\}$ and the size of 3D tubes as $\{t, h, w\}$, then the number of token embeddings is $n = n_t \times n_s = |\frac{T}{t}| \times |\frac{HW}{hw}|$. Additionally, positional embeddings are added to each token to break the permutation invariance. After the patch embedding layer, an n-token sequence $\mathbf{X} \in \mathbb{R}^{n \times d}$ is passed into the self-attention layer, which computes a weighted sum of the values based on the affinity of other tokens. Mathematically, the self-attention is formulated as:

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{1}$$

where $\mathbf{Q},\mathbf{K},\mathbf{V} = f_Q(\mathbf{X}), f_K(\mathbf{X}), f_V(\mathbf{X}) \in \mathbb{R}^d$ are typically linear transformations of $\mathbf{X}$. After spatial-temporal interaction, the tokens are sent into a feedforward network f_{FFN}, which consists of a three-layer MLP to exchange inter-channel information.

3.2. Semantic-aware Temporal Accumulative Score

Our intuitive criterion is to drop a token if similar tokens exist before it while reserving semantically meaningful tokens. To this end, our approach considers two factors when determining whether to retain or discard a token in the Transformer. The first is its similarity to the other tokens along the temporal axis, and the second is its contribution to the class attribute. We discuss these two principles in order.

Temporal redundancy. Intuitively, a token should be removed if similar tokens have already existed in previous frames. Therefore, we remove tokens frame-by-frame by comparing whether similar tokens have been retained. For simplicity, we reduce a constant number of tokens in each frame to ensure parallel computing. We introduce the accumulative temporal score $\mathbf{A} \in [0,1]^{n_t \times n_s}$ to model the probability of dropping a token conditioned on the specific frame t. Specifically, we define:

$$\mathbf{A}_{t,s} := \mathbb{P}_{\text{drop}}(\mathbf{X}_{t,s}) \in [0,1] \quad \text{s.t.} \sum_{s=1}^{n_s} \mathbf{A}_{t,s} = 1, \tag{2}$$

where tokens with higher temporal accumulative scores are more likely to be pruned because they carry a high degree of temporal redundancy. Next, we eliminate r tokens with the highest scores from $\mathbf{A}$ at t-th frame and transfer the remaining probability distribution $\mathbf{A}_t^{'} \in \mathbb{R}^{(n_s-r)\times 1}$ to the next frame via the transition probability $\mathbb{P}_{\text{drop}}(\mathbf{X}_t|\mathbf{X}_{t-1}^{'}) \in \mathbb{R}^{n_s \times (n_s-r)}$. By excluding the dropped tokens of the last frame, we effectively restart the repetition aggregation. This prevents the high scores from being concentrated on specific tokens and allows for the global identification of redundancy. Mathematically,

$$\begin{aligned}
&\mathbf{A}_{t+1} := \mathbb{P}_{\text{drop}}(\mathbf{X}_{t+1}|\mathbf{X}_t^{'})\mathbf{A}_t^{'}, \\
&\mathbb{P}_{\text{drop}}(\mathbf{X}_{t+1}|\mathbf{X}_t^{'}) := \text{softmax}(f(\mathbf{X}_{t+1})f(\mathbf{X}_t^{'})^T),
\end{aligned} \tag{3}$$

where f is the projection head to measure the similarity, and we construct transition probability by softmax-based affinity matrix. Note that we do not need to train a new projection head f because the self-attention provides the necessary functionality, and the key function f_K extracts most relevant knowledge for affinity estimation, as shown in Table 6c. This formulation allows us to connect all temporally distinct tokens through a simple Markov chain and aggregate potential redundancy from the first frame to every subsequent frame.

Algorithm 1 Pseudocode of **STA** in a PyTorch-like style.

```
# x: token embedding, n_t x n_s x d
# I: image-based token pruning method
# r: drop number per frame
# sim: token-to-token affinity function

# min-max norm, Eqn.(4)
aam = norm(x.abs().sum(-1)) # size: (n_t, n_s, 1)

# token removal at 1-st frame
x_0 = I(x[0]) # size: (n_s-r, d)

# initialization
x_list, x_old = [x_0], x_0
for t in range(1, n_t):
   # token-to-token affinity matrix
   A_t = sim(x[t], x_old) # size: (n_s, n_s-r)
   # accumulative temporal score
   s_acc = mm(A_t, s_acc) # size: (n_s, 1)
   # class-aware accumulative temporal score, Eqn.(5)
   s = s_acc * (1-aam[t])
   s = s.squeeze(dim=-1) # size: (n_s)

   # keep tokens with the minimal score
   i_t = s.topk(k=N-r, largest=False) # size: (n_s-r)
   x_old = x[t, i_t] # size: (n_s-r, d)
   x_list = x_list.append(x_old)

   # cut off the dropped tokens' score
   s_acc = s_acc[i_t] # size: (n_s-r, 1)
   # first-order norm
   s_acc = s_acc / s_acc.sum()

return stack(x_list, dim=0) # size: (n_t, n_s-r, d)
```

mm: matrix multiplication.

Semantic importance. Up until this point, our approach has focused on capturing temporally repetitive information. However, we have neglected the influence of semantic attributes. In other words, we have treated each token equally, regardless of its contribution to the semantics of the class. To integrate semantics importance with our approach, we use the activation-based attention map $\mathcal{F}$ [45], which takes the feature matrix $\mathbf{X} \in \mathbb{R}^{n_t \times n_s \times d}$ as input and produces a score for each token in the matrix. Specifically, we define the semantic score for token $\mathbf{X}_{t,s}$ as:

$$\mathcal{F}(\mathbf{X}_{t,s}) = \sum_{i=1}^{d} |\mathbf{X}_{t,s,i}| \in \mathbb{R}^+. \tag{4}$$

Intuitively, through the summation of absolute activation values over channel dimension, a high absolute activation suggests a significant contribution to next layers. Moreover, we apply STA on off-the-shelf Transformers supervised by semantic labels, where high activation areas tend to represent discriminative category information. Thus, activation-based attention maps could effectively capture the importance or relevance of the token to the overall semantics. We then use this score to re-weigh the temporal accumulative scores $\mathbf{A}$, giving tokens with high semantic contributions less weight in the pruning process. This ensures that tokens with high semantics are more likely to be retained, even if they have a high degree of temporal redundancy.

Finally, we compute the semantic-aware temporal accumulative score $\widetilde{\mathbf{A}}_{t,s}$ by integrating the semantic score $\mathcal{F}(\mathbf{X}_{t,s})$ with the accumulative temporal score $\mathbf{A}$, *i.e.,*

$$\widetilde{\mathbf{A}}_{t,s} = (1 - \mathcal{F}(\mathbf{X}_{t,s}))\mathbf{A}_{t,s}, \tag{5}$$

where $\mathcal{F}(\mathbf{X}_{t,s})$ is min-max normalized to the range [0,1]. We utilize the semantic-aware accumulative temporal scores to guide token removal for all subsequent frames, except for the first frame. Thus, we adopt an image-based token pruning method on the first frame to kick off our algorithm. Once tokens are discarded through our strategy, they are never employed in subsequent layers, thus accelerating the inference of the Transformer.

We summarize the pseudocode of STA in Alg. 1. The algorithm takes token embedding $\mathbf{X} \in \mathbb{R}^{n_t \times n_s \times d}$, an image-based token pruning method I, the number of tokens to drop per frame r, and a token-to-token affinity function as inputs. The algorithm calculates the STA score for each frame, selects the tokens with the minimal STA score, and retains them for the next frame. This process is repeated for all frames and returns the resulting token embedding matrix with the retained tokens $\mathbf{X}' \in \mathbb{R}^{n_t \times (n_s - r) \times d}$.

Summary of the superiority of STA. Compared to the previous token-pruning methods, our approach, STA, offers three significant merits when applied to video data:

- STA fully considers the potential repetition of tokens along the temporal axis and eliminates the genuine redundancy with insignificant semantics. The temporal aggregation design makes the scoring mechanism more motion-aware and suitable for video data;
- STA works as a plug-in module without the introduction of additional parameters and it does not require the retraining of the video Transformer;
- STA achieves a complexity of $O\left(n_t n_s (n_s - r)\right)$, resulting in negligible additional FLOPs that only take up a small percentage of the total forward pass. Moreover, our algorithm allows for the bulk of computation to be done in parallel, making it friendly to modern GPU devices.

Overall, our approach is efficient and easily deployable, making it an ideal solution for pruning video Transformers.

3.3. Discussion

In this section, we present a two-part practical analysis to shed the light on the intuition behind STA.

Model	Small	Base	Large	Huge
ViT	5.10	5.38	5.07	5.55
Rand-ViT	5.00	5.32	4.95	5.46
STA-ViT	4.43	4.74	4.28	4.78

Table 1: Trajectory sum for ViT family on the Kinetics-400 validation set. Compared to random pruning, STA-ViT decreases the temporal redundancy significantly.

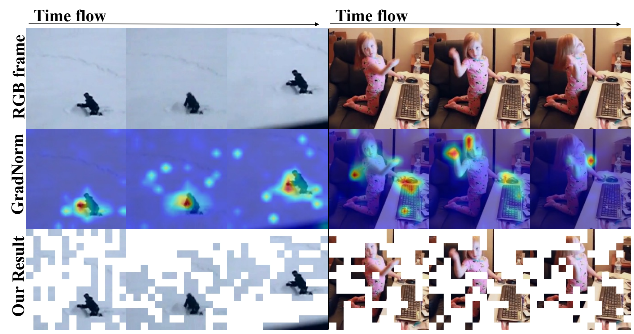


Figure 3: Gradient visualization for ViT-Large on the Kinetics-400 validation set. We stack original RGB frames, gradient norm heatmap, and our pruning result from top to bottom. Our pruning algorithm preserves the area of rich semantics well.

Does STA effectively reduce the temporal redundancy? To answer this question, we first define temporal redundancy as the frequency with which similar tokens appear at different timestamps. We then assess this phenomenon in a video by probing the last frame, denoted as $\mathbf{X_{-1}} \in \mathbb{R}^{n_s \times d}$, and aggregating the cosine similarity between each token and the most similar tokens in previous frames. We term this aggregation as trajectory sum $S \in \mathbb{R}$. Mathematically,

$$S = \frac{1}{n_s} \sum_{i=1}^{n_s} \sum_{t=1}^{n_t - 1} \max_{j \in \{1, \cdots, n_s\}} \text{cos-sim}(\mathbf{X}_{-1}, \mathbf{X}_t)_{ij}, \tag{6}$$

where $\text{cos-sim}(\mathbf{X}, \mathbf{Y})_{ij} = \frac{\mathbf{X}_i \cdot \mathbf{Y}_j}{|\mathbf{X}_i||\mathbf{Y}_j|}$. A higher trajectory sum indicates greater temporal redundancy and lower diversity along the temporal axis. When all the frames are the same, the score would reach its theoretical maximum, which is $N_t - 1$. Then, we compare our method with the standard Transformer and random-prune counterparts in terms of the proposed trajectory sum. Table 1 shows that the video Transformer exhibits heavy temporal redundancy and random pruning fails to alleviate them considerably. In contrast, STA achieves a far lower trajectory sum, indicating that it effectively eliminates temporal redundancy.

Does STA retain semantics-rich tokens? To manifest it, we calculate the gradient norm for each token when backpropagating the training loss $\mathcal{L}$. We then aggregated the first-order gradient norm of $\mathbf{X}^l$ at each layer l to obtain the GradNorm, which reflects the contribution of each token to the final prediction. Mathematically,

$$\text{GradNorm}(\mathbf{X}) = \sum_{l=1}^{L}\sum_{i=1}^{d}\left|\frac{\partial \mathcal{L}}{\partial \mathbf{X}^l_{:,:,i}}\right| \in \mathbb{R}^{n_t \times n_s}, \quad (7)$$

where L is the total number of the Transformer block, *e.g.*, $L = 24$ for ViT-Large. Figure 3 shows the GradNorm distribution in the form of a heatmap. The heatmap reveals sparse patterns across the board, indicating that most tokens do not contribute significantly to the final prediction. Instead, the key regions responsible for the final prediction are usually motion-centric entities. This agrees with our intuition of highlighting semantically meaningful regions. With the help of activation-based attention map $\mathcal{F}$ in Eqn. (4), STA retains almost all areas of high-activation GradNorm as is evidenced in the last row of Figure 3.

4. Experiments

4.1. Experimental Setup

Datasets and backbones. We evaluate our algorithm on two standard video action recognition datasets. Kinetics-400 [14] (K400) and Something-Something V2 [11] (SSV2). Kinetics-400 is a large-scale action recognition dataset sourced from YouTube, which consists of around 10-second clips with 400 human action classes. The training and validation set has approximately 240K and 20K videos, respectively. Something-Something V2 is a motion-heavy benchmark with 174 labels, where the object and background are shared across the different action categories. Around 170K and 25K videos exist in the training and validation set of SSV2, respectively. We implement our STA strategy with two mainstream Transformer-based backbones, namely ViT [8] and VideoSwin [23]. For ViT, we sample 16 frames with 224^2 pixels as input and the size of 3D tube is $\{2, 16, 16\}$. Therefore, the number of tokens for all layers is $n = \frac{16}{2} \times \frac{224^2}{16^2} = 8 \times 14^2$. We load the open-sourced checkpoints from VideoMAE [35] due to its superior performance. For VideoSwin, the input resolution is 32×224^2 and the size of 3D tube is $\{2, 4, 4\}$. The number of tokens at four hierarchical stages would be $\{16 \times 56^2, 16 \times 28^2, 16 \times 14^2, 16 \times 7^2\}$. Though our algorithm is not intended for window-based Transformers that require structural integrity, we still find a simple solution. We compensate for the discarded token with the nearest kept tokens when computing the window attention and speed up the rest of the process, such as the linear projection. In total, we test 10 off-the-shelf backbones with different pre-trained weights on two benchmarks.

Implementation details. To progressively remove inattentive tokens, we apply our STA module three times. For ViT, we insert our STA module at the end of the 1^{st}, $(1 + L/3)^{th}$, $(1 + 2L/3)^{th}$ block, where L is the total number of Transformer blocks. For VideoSwin, there are four hierarchical stages with varied resolutions and the STA module is located at the end of the first three stages. We denote different variants of STA as STA^{r_1}-Model, where r_1 is the number of spatial tokens dropped per frame at the first stage. We adopt a decreasing schedule, which reduces the dropped number by half at each stage. For instance, STA^{64}-ViT-L indicates pruning $\{8 \times 64, 8 \times 32, 8 \times 16\}$ tokens on ViT-Large. We weigh the computation via two metrics, FLOPs (floating-point operations) and throughput. FLOPs are reported with the help of fvcore library[1] and throughput (clips/s) is measured at a batch size of 32 on a single 32G Tesla V100. Besides, we closely follow the inference metric for ViT and VideoSwin in [35, 23].

- ViT [35]: To evaluate on K400, we sample 5 × 3 views by uniformly selecting 5 16-frame clips from a full-length video in the temporal dimension with the frame stride of 4. For each clip, we resize the shorter spatial side to 224 pixels and extract 3 crops of 224 × 224 resolution that cover the longer spatial axis. The final score is the average score computed over all views. For SSV2, we follow a similar procedure by sampling 2 clips × 3 crops and the frames stride is 2.
- VideoSwin [23]: For K400, we extract 4 32-frame clips from each full-length video using a temporal stride of 2 and a spatial size of 224 × 224. Similarly, for SSV2, we extract 1 set of clips using a spatial size of 224 × 224 and 3 spatial crops, with a frame stride of 2. Besides, we prune VideoSwin three times as $\{r_1, 1.5r_1/4, 2r_1/16\}$. For instance, STA^{256}-VideoSwin-S indicates pruning $\{16 \times 256, 16 \times 96, 16 \times 32\}$ on VideoSwin-S.

4.2. Main Results

We conduct a thorough investigation of two off-the-shelf model families, ViT [8] and VideoSwin [23], on the Kinetics-400 and Something-Something V2 datasets. The results presented in Table 2 demonstrate that our proposed method can significantly reduce the computational costs of ViT models by $25\% \sim 49\%$, with negligible impacts on performance (-0.2% $\sim$ -1.0%). It is worth noting our method shows a favorable trade-off between complexity and performance for larger models. For instance, our method reduces the FLOPs of ViT-Huge by half to just 611 GFLOPs, with only a 0.2% drop in accuracy.

To demonstrate the potential of our method to generalize well on various transformer backbones, we are conducting

[1] https://github.com/facebookresearch/fvcore

Base Model	Metrics	Drop Number r_1			
		0	32	48	64
ViT-S	K400 Acc. (%)	78.8	78.8 (-0.0)	78.6 (-0.2)	78.1 (-0.7)
	SSV2 Acc. (%)	66.8	66.6 (-0.2)	66.4 (-0.4)	65.8 (-1.0)
	GFLOPs	57	42 (-26%)	35 (-39%)	29 (-49%)
ViT-B	K400 Acc. (%)	81.2	81.2 (-0.0)	81.1 (-0.1)	80.8 (-0.4)
	SSV2 Acc. (%)	70.6	70.4 (-0.2)	70.3 (-0.3)	69.9 (-0.7)
	GFLOPs	180	136 (-24%)	116 (-36%)	96 (-47%)
ViT-L	K400 Acc. (%)	85.1	85.2 (+0.1)	85.1 (-0.0)	85.0 (-0.1)
	GFLOPs	597	446 (-25%)	376 (-37%)	308 (-48%)
ViT-H	K400 Acc. (%)	86.3	86.3 (-0.0)	86.2 (-0.1)	86.1 (-0.2)
	GFLOPs	1192	890 (-25%)	748 (-37%)	611 (-49%)

Table 2: Main Results for STA-ViT family on Kinetics-400 [14] (K400) and Something-Something V2 [11] (SSV2). All input resolution is 16×224^2.

Base Model	Metrics	Drop Number r_1			
		0	192	256	320
VideoSwin-T	K400 Acc. (%)	78.8	78.7 (-0.1)	78.7 (-0.1)	78.6 (-0.2)
	GFLOPs	88	68 (-23%)	61 (-31%)	54 (-39%)
VideoSwin-S	K400 Acc. (%)	80.5	80.3 (-0.2)	80.2 (-0.3)	80.1 (-0.4)
	GFLOPs	166	121 (-27%)	106 (-36%)	91 (-45%)
VideoSwin-B	K400 Acc. (%)	82.7	82.5 (-0.2)	82.5 (-0.2)	82.3 (-0.4)
	K400 GFLOPs	282	202 (-28%)	176 (-38%)	149 (-47%)
	SSV2 Acc. (%)	69.6	69.6 (-0.0)	69.5 (-0.1)	69.2 (-0.4)
	SSV2 GFLOPs	321	241 (-25%)	215 (-33%)	188 (-41%)

Table 3: Main Results for STA-VideoSwin family on Kinetics-400 [14] (K400) and Something-Something V2 [11] (SSV2). All input resolution is 32×224^2. VideoSwin-B employs varying window sizes for K400 and SSV2, leading to a discrepancy in FLOPs.

further experiments on VideoSwin [23], a modern architecture that uses a window shuffling operation to interchange information. As VideoSwin is naturally unsuitable for unstructured tokens, we are filling the dropped locations during the window attention operation with the nearest tokens and then discarding the replicated tokens after the attention operation. Empirical results in Table 3 indicate that the performance of VideoSwin holds until FLOPs fall by roughly 40%. This observation verifies that both columnar ViT and hierarchical VideoSwin have heavy and unnecessary computations that can be significantly optimized.

Comparison with the state of the art. Firstly, we tabulate a comparison of our proposed method on K400 in Table 4. Our model performs favorably in terms of both accuracy and computation cost. For example, ViT-L equipped with our STA achieves the same accuracy as MViTv2-L [19] but with less than a quarter of the computational cost. Moreover, STTS [40] proposes a scorer network to conduct dynamic token selection separately in space and time, requiring to be trained in an end-to-end fashion. Our result surpasses STTS by 0.4% accuracy using the same backbone VideoSwin-B but with only 60% GFLOPs. This result verifies that leveraging the model itself to weigh the redundancy of tokens is sufficient to reduce complexity. We also report the result on SSV2 in Table 5. The superior performance of our proposed method verifies that STA prunes the inconsequential tokens via temporal cues, as it is known that understanding SSV2 mainly relies on temporal information. Specifically, ViT-B equipped with our STA surpasses most of the prior arts with a considerably minor complexity of 116 GFLOPs. For VideoSwin, our strategy outperforms STTS-VideoSwin by 0.5% accuracy with 80% of the computation cost.

Visualization of STA. Figure 4 shows image patches corresponding to kept tokens after three stages. The results align with our objective of resisting temporal redundancy and retaining informative tokens. In a tennis sequence, STA preserves the most meaningful patches, including a human

Model	GFLOPs×views	Top-1
TimeSformer-L [2]	$8353 \times 1 \times 3$	80.7
Motionformer-L [28]	$1185 \times 10 \times 3$	80.2
ViViT [1]	$3981 \times 4 \times 3$	84.9
Swin-L [23]	$2107 \times 10 \times 5$	84.9
MViTv2-L [19]	$2828 \times 5 \times 3$	86.1
ViT-H [35]	$1192 \times 5 \times 3$	86.3
STTS-VideoSwin-B [40]	$253 \times 4 \times 3$	81.9
ToMe-ViT-L [3]	$281 \times 10 \times 1$	84.5
STA^{320}-VideoSwin-B (ours)	$149 \times 4 \times 3$	82.3
STA^{64}-ViT-L (ours)	$308 \times 5 \times 3$	85.0
STA^{64}-ViT-H (ours)	$611 \times 5 \times 3$	86.1

Table 4: Comparisons with the-state-of-the-arts method on Kinetics-400. We report the computational cost with a single view (temporal clip with spatial crop) × the number of views (FLOPs× view$_{\text{time}}$ × view$_{\text{space}}$). Gray represents that this method leverages the dynamic token pooling to optimize existing backbones.

Model	GFLOPs×views	Top-1
TimeSformer-L [2]	$5549 \times 1 \times 3$	62.4
Motionformer-L [28]	$1185 \times 1 \times 3$	68.1
MViTv2-B [19]	$225 \times 1 \times 3$	70.5
VideoSwin-B [23]	$321 \times 1 \times 3$	69.6
ViT-B [35]	$180 \times 2 \times 3$	70.6
STTS-VideoSwin-B [40]	$237 \times 1 \times 3$	68.7
STA^{320}-VideoSwin-B (ours)	$188 \times 1 \times 3$	69.2
STA^{48}-ViT-B (ours)	$116 \times 2 \times 3$	70.3

Table 5: Comparisons with the-state-of-the-arts method on Something-Something V2. We report the computational cost with a single view (temporal clip with spatial crop) × the number of views (FLOPs× view$_{\text{time}}$ × view$_{\text{space}}$). Gray represents that this method leverages the dynamic token pooling to optimize existing backbones.

at the far end of the court, and filters out dull backgrounds like the blue ground. The temporal aggregation design ensures that the kept tokens are not just the most salient ones but also a variety of regions, preserving diversity within videos for better reasoning.

4.3. Ablation Study

To find the optimal strategy, we conduct a series of ablation studies. We evaluate off-the-shelf ViT-Large on K400 by default and report accuracy, FLOPs, and throughput for reference unless otherwise stated.

Token removal at the first frame. To investigate how the first-frame removal affects performance, we conduct experiments on three candidates. (1) **Random Prune**: we randomly select r tokens to discard. (2) **Grid Prune**: we split the first frame into $\sqrt{r} \times \sqrt{r}$ grids spatially and drop one random token per grid. (3) **ToMe Prune**: inspired by recent image pruning method ToMe [3], we get rid of the most similar tokens by the simple bipartite soft matching. The difference here is that we just drop tokens rather than 'merge' tokens. Note that all three pruning ways are negligible in the terms of computation and lead to similar speedup. It actually echoes the main insight of STA, leveraging temporal aggregation to reduce spatio-temporal redundancy. Even given a random initialization state, the sequential STA strategy could still decrease the total temporal redundancy and optimize the video Transformers effectively.

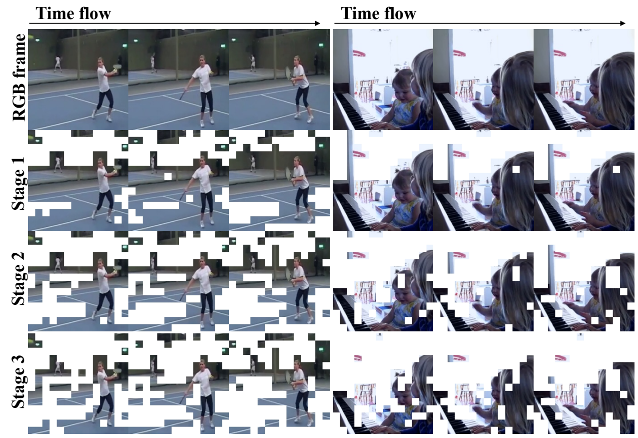


Figure 4: Visualization of the proposed STA strategy. We masked out the discarded tokens with white boxes. STA not only retains informative tokens but also ensures diverse regions for improved video reasoning.

Pruning schedule. We explore how to assign the number of dropping tokens among three stages. When maintaining a similar throughput, we devise three types of schedules:

$$\text{Constant Schedule:}\{8 \times 48, 8 \times 48, 8 \times 48\};$$
$$\text{Decreasing Schedule:}\{8 \times 64, 8 \times 32, 8 \times 16\};$$
$$\text{Increasing Schedule:}\{8 \times 27, 8 \times 54, 8 \times 108\}.$$

As shown in Table 6d, the decreasing schedule owns the least accuracy reduction with similar throughput. It verifies that standard video Transformers process a great number of uninformative tokens that can be dropped at the beginning.

Temporal accumulation order. Besides starting the accumulation flow from the beginning of the input video, we could also kick off at the ending frames. In Table 6b, we empirically find alternating order at different dropping stages outperforms the consistent counterparts. We speculate that the same accumulation direction would amplify

Method	Top-1	Top-5
Random	84.78	96.46
Grid	84.85	96.50
ToMe	84.96	96.50

(a) Ablation on token removal methods at the first frame.

Order	Top-1	Top-5
F-B-F	84.96	96.50
B-F-B	84.97	96.43
F-F-F	84.26	96.41
B-B-B	84.35	96.37

(b) Ablation on temporal accumulation order.

Similarity	Top-1	Top-5
f_Q	84.83	96.52
f_K	84.96	96.50
f_V	84.84	96.50
f_{FFN}	84.94	96.50

(c) Ablation on different similarity function f.

Schedule	Top-1	Top-5	clips/s
constant	84.68	96.36	47
decreasing	84.96	96.50	47
increasing	77.68	93.72	44

(d) Ablation on dropping schedule among three stages.

Score	ViT-S	ViT-B	ViT-L
$1 - \mathcal{F}(\mathbf{X}_{t,s})$	77.33	80.52	84.87
$\mathbf{A}_{t,s}$	77.78	80.43	84.69
$(1 - \mathcal{F}(\mathbf{X}_{t,s}))\mathbf{A}_{t,s}$	78.12	80.82	84.96

(e) Ablation on scoring mechanism. Top-1 is reported.

Table 6: Results of STA ablation experiments. F and B in (b) mean forward and backward order, respectively. The baseline ViT-L without STA obtains 85.05% Top-1 and 96.55% Top-5 accuracy on K400 at 19.5 clips/s. Gray is our default setting.

intrinsic propagation error but alternating the order counteracts it, leading to more reasonable pruning.

Similarity function choice. We ablate four similarity project heads $\{f_Q, f_V, f_K, f_{\text{FFN}}\}$. Table 6c shows that key function f_k captures the most correct affinity with minimal noise. The observation coincides with previous work [3].

Scoring mechanism. To explore how accumulation score and semantic identification boost each other, we conduct the experiment with different scoring formulas. Table 6e demonstrates that considering both temporal redundancy and semantics helps in discovering informative tokens. The results on ViT-S show that temporal aggregation modeled by the Markov Chain plays an important role in the pruning process, while semantic importance functions effectively for ViT-B and ViT-L.

Performance vs. prune number r. To seek the sweet spot of our algorithm, we vary the prune number r at the first stage ranging from [16, 96] and evaluate the Top-1 accuracy. In addition, we compare our STA with the Random pruning baseline. As displayed in Figure. 5, STA behaves fairly robust to the token reduction and consistently surpasses the result of the random pruning. Specifically, $r = 64$ doubles the throughput but just drops 0.1% accuracy. This confirms that our algorithm retains the semantics-rich tokens with the lowest redundancy.

5. Conclusion

In conclusion, we propose a new token pruning strategy, Semantic-aware Temporal Accumulation (STA), for

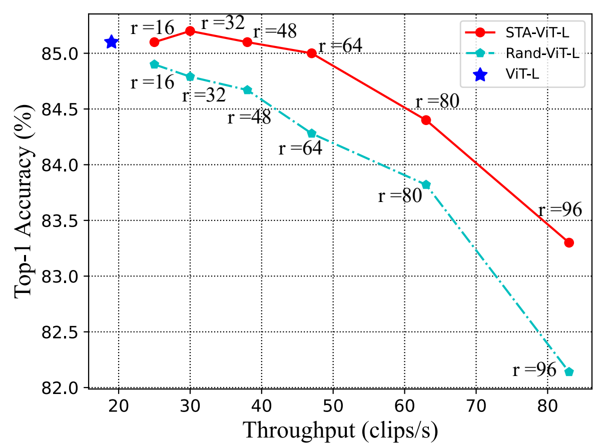


Figure 5: Top-1 accuracy and throughput under two pruning methods with various prune numbers r.

video Transformers that can significantly reduce computation overhead with a subtle accuracy drop. Specifically, we consider temporal redundancy and semantic importance when deciding to keep or drop the token. Our approach does not introduce any parameter and can directly accelerate the off-the-shelf video Transformers without training. The extensive experiments demonstrate that our method empowers video Transformers to obtain a competitive speed-accuracy trade-off compared to the prior arts.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China under Grant 62250055, Grant 61932022, Grant 62120106007, and in part by the Program of Shanghai Science and Technology Innovation Project under Grant 20511100100.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6836–6846, October 2021. 1, 2, 8

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *ICML*, volume 2, page 4, 2021. 1, 2, 8

[3] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023. 2, 8, 9

[4] Adrian Bulat, Juan Manuel Perez Rua, Swathikiran Sudhakaran, Brais Martinez, and Georgios Tzimiropoulos. Space-time mixing attention for video transformer. *Advances in Neural Information Processing Systems*, 34:19594–19607, 2021. 2

[5] Jiaqi Chen, Jiachen Lu, Xiatian Zhu, and Li Zhang. Generative semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7111–7120, 2023. 1

[6] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 1

[7] Shuangrui Ding, Maomao Li, Tianyu Yang, Rui Qian, Haohang Xu, Qingyi Chen, Jue Wang, and Hongkai Xiong. Motion-aware contrastive video representation learning via foreground-background merging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9716–9726, 2022. 3

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. 1, 2, 3, 6

[9] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6824–6835, 2021. 2

[10] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213, 2020. 3

[11] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, pages 5842–5850, 2017. 2, 6, 7

[12] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3148–3159, June 2022. 1

[13] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2000–2009, 2019. 3

[14] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 2, 6, 7

[15] Bruno Korbar, Du Tran, and Lorenzo Torresani. Scsampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6232–6242, 2019. 3

[16] Myunggi Lee, Seungeui Lee, Sungjoon Son, Gyutae Park, and Nojun Kwak. Motion feature network: Fixed motion filter for action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 387–403, 2018. 3

[17] Jin Li, Yaoming Wang, Xiaopeng Zhang, Yabo Chen, Dongsheng Jiang, Wenrui Dai, Chenglin Li, Hongkai Xiong, and Qi Tian. Progressively compressed auto-encoder for self-supervised representation learning. In *The Eleventh International Conference on Learning Representations*, 2023. 2

[18] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 3

[19] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022. 2, 7, 8

[20] Youwei Liang, Chongjian GE, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. EVit: Expediting vision transformers via token reorganizations. In *International Conference on Learning Representations*, 2022. 2

[21] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7083–7093, 2019. 3

[22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1, 2

[23] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 1, 2, 6, 7, 8

[24] Jiachen Lu, Jinghan Yao, Junge Zhang, Xiatian Zhu, Hang Xu, Weiguo Gao, Chunjing Xu, Tao Xiang, and Li Zhang. Soft: Softmax-free transformer with linear complexity. *Advances in Neural Information Processing Systems*, 34:21297–21309, 2021. 2

[25] Lingchen Meng, Hengduo Li, Bor-Chun Chen, Shiyi Lan, Zuxuan Wu, Yu-Gang Jiang, and Ser-Nam Lim. Adavit: Adaptive vision transformers for efficient image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12309–12318, 2022. 2

[26] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3163–3172, 2021. 2

[27] Seong Hyeon Park, Jihoon Tack, Byeongho Heo, Jung-Woo Ha, and Jinwoo Shin. K-centered patch sampling for efficient video recognition. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 160–176. Springer, 2022. 3

[28] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, 34:12493–12506, 2021. 2, 8

[29] Rui Qian, Shuangrui Ding, Xian Liu, and Dahua Lin. Static and dynamic concepts for self-supervised video representation learning. In *European Conference on Computer Vision*, pages 145–164. Springer, 2022. 3

[30] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021. 2

[31] Michael Ryoo, AJ Piergiovanni, Anurag Arnab, Mostafa Dehghani, and Anelia Angelova. Tokenlearner: Adaptive space-time tokenization for videos. *Advances in Neural Information Processing Systems*, 34:12786–12797, 2021. 2

[32] Jing Tan, Jiaqi Tang, Limin Wang, and Gangshan Wu. Relaxed transformer decoders for direct action proposal generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13526–13535, 2021. 1

[33] Jing Tan, Yuhong Wang, Gangshan Wu, and Limin Wang. Temporal perceiver: A general architecture for arbitrary boundary detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[34] Jing Tan, Xiaotong Zhao, Xintian Shi, Bin Kang, and Limin Wang. Pointtad: Multi-label temporal action detection with learnable query points. *Advances in Neural Information Processing Systems*, 35:15268–15280, 2022. 1

[35] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022. 6, 8

[36] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 3

[37] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV*, pages 459–479. Springer, 2022. 1

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1

[39] Qiang Wan, Zilong Huang, Jiachen Lu, Gang Yu, and Li Zhang. Seaformer: Squeeze-enhanced axial transformer for mobile semantic segmentation. *arXiv preprint arXiv:2301.13156*, 2023. 2

[40] Junke Wang, Xitong Yang, Hengduo Li, Li Liu, Zuxuan Wu, and Yu-Gang Jiang. Efficient video transformers with spatial-temporal token selection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXV*, pages 69–86. Springer, 2022. 2, 3, 7, 8

[41] Yulin Wang, Zhaoxi Chen, Haojun Jiang, Shiji Song, Yizeng Han, and Gao Huang. Adaptive focus for efficient video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16249–16258, 2021. 3

[42] Yulin Wang, Rui Huang, Shiji Song, Zeyi Huang, and Gao Huang. Not all images are worth 16x16 words: Dynamic transformers for efficient image recognition. *Advances in Neural Information Processing Systems*, 34:11960–11973, 2021. 2

[43] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022. 1, 2

[44] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S Davis. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1278–1287, 2019. 3

[45] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations*, 2017. 4

[46] Xuefan Zha, Wentao Zhu, Lv Xun, Sen Yang, and Ji Liu. Shifted chunk transformer for spatio-temporal representational learning. *Advances in Neural Information Processing Systems*, 34:11384–11396, 2021. 2

[47] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. Vidtr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13577–13587, 2021. 1, 2

[48] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 1

[49] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 1, 2

[50] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 2

[51] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 695–712, 2018. 3