A. SUPPLEMENTARY MATERIAL

A.1. Attention Rollout Implementation.

We provide additional implementation details for our method. Transformer blocks in ViT may apply multiple attention matrices through parallel heads, as well as residual connections. In these cases, we represent the attention matrix A_l as follows:

$$\begin{aligned} A'_l &= \frac{1}{h}\sum_h A_l^h + I \\ A_l(m,n) &= \frac{A'_l(m,n)}{\sum_k A'_l(m,k)} \end{aligned} \tag{1}$$

The attention rollout $\tilde{A}$ is given by Eq. (**??**). In practice, instead of computing $\tilde{A}$ and follow Eq. (**??**), as $\tilde{A}^T = (A_L \times A_{L-1} \times ... \times A_1)^T = A_1^T \times \cdots \times A_{L-1}^T \times A_L^T$, we compute the importance vector s_0 recursively as follows, reducing computations by a factor of N:

$$s_l = \begin{cases} A_{l+1}^T s_{l+1} & \text{if } l < L \\ s_L & \text{if } l = L \end{cases} \tag{2}$$

A.2. Additional Implementation Details.

(1) OD (2) action recognition (temporal network training?) (3)...

A.3. Tracking Window Size.

Results of an ablation study on the size of the window used for token tracking are given in Table 1. Our experiments indicate that token tracking with a window size $r = 1$ improves accuracy compared to the no-tracking baseline ($r = 0$). This becomes more apparent with larger motion in the videos, as shown in the results for 1/4 Sampling. We also note that further increase of the window size does not lead to improvements.

Table 1: Effect of tracking window size r on ImageNet VID .

Window Size	GFLOPs	mAP50 (%)	
		1/1 Sampling	1/4 Sampling
$r = 0$	46.77	82.0	79.3
$r = 1$	46.78	82.1	80.0
$r = 2$	46.80	82.0	79.9
$r = 3$	46.82	82.0	79.6

A.4. Additional Qualitative Results.

We provide additional qualitative results for ImageNet VID in Figure XXX, EPIC-Kitchens in Figure XXX and Kinetics-400 in Figure XXX.

A.5. Temporal Token Tracking.

Our method propagates importance scores between successive frames as defined in Eq XXX. Intuitively, it can be regarded as a template matching in the token space. While more advanced tracking methods can be used, our motivation lies in applying them on tokens rather than on frame pixels, due to the following traits; first, as we want to relate tokens between successive frames, we seek to avoid additional propagation between tokens and pixels which might impact such relation. Second, tracking methods may require less computations when applied on tokens rather than pixels. We suggest that input tokens, derived using a linear projection on non-overlapping image patches, retain locality such that translation in the pixel space would be similarly depicted in the token space. In addition, the aforementioned projection in ViTs used in this work is $\mathbb{R}^{16\times16\times3} \rightarrow \mathbb{R}^{768}$, potentially preserving the amount of information. We note, however, that the above does not hold for subsequent token transformations. In particular, attention layers exchange information between tokens and hence do not preserve locality. Furthermore, tokens in successive frames undergo different transformations, making their latent representations incommensurate.

A.6. Token Pruning at Intermediate Layers.

Our method estimates the importance of input tokens at frame t, z_0^t, with respect to predictions in frame $t-1$. At subsequent layers, attention rollout can also be used to propagate importance scores from s_0^t to s_l^t, $1 < l < L$. Following Eq (**??**), we get:

$$\begin{aligned} s_0^t &= A_{(l,1)}^T s_l^t \\ s_l^t &= (A_{(l,1)}^T)^{-1} s_0^t \end{aligned} \tag{3}$$

With Eq. (3), token pruning can be applied gradually across transformer blocks. In our experiments, we observed marginal improvement in complexity-accuracy tradeoffs, along with a runtime increase due to matrix inversion. We note that while such pruning in intermediate layers enables gradual loss of information, pruning at s_l^t is as informed as at s_0^t, as they both stem from the importance of s_L^{t-1}.