Principles of Visual Tokens for Efficient Video Understanding

Supplementary Material

1. Additional Experimental Results

Looking at the simplicity of LITE, one could think that learning to predict the oracle is particularly easy and straightforward. After all, if an MLP can learn this, potentially a more sophisticated model should outperform it. This section describes the variants evaluated, giving additional insights and showcasing that token-selection is far from trivial.

Cleaning the oracle training data. Visualization suggests that the oracle is somewhat noisy. Could we could clean the artifacts to improve the performance of the oracle? We explore several cleaning strategies:

- Edges. Grad-CAM often produces noisy high activations at image boundaries. We decrease the values on edges to reduce boundary artifacts.
- Isolated peaks. The raw value of the oracle often has isolated peaks, which can misrepresent the true important areas. We remove small areas below a certain threshold.
- Sharpening the distribution. The original distribution of the oracle is often smooth. We transform oracle values to amplify the differences between high and low values, making the distribution more concentrated around 0 or 1.

Including global information. The LITE model assesses the value of a token based solely on the information of one token. This is a simplification of the way tokens are used, and overlooks qualities such as diversity of information, relationship, etc. We experimented with adding a global branch to LITE , where instead of simply using the value of a token in isolation, we use several 3D convolutions to acquire the nearby context information. Then, a self-attention operation is employed to capture long-range dependencies and model interactions between tokens, allowing the network to integrate global information effectively.

Adding complexity to the selector. A 3-layer MLP is a fairly simple model, and potentially we should be able to do better with a more sophisticated model. Adding layers or using other architectures can potentially lead to better results, even at the cost of an increased amount of computation. We experimented with several aspects. We used more layers (up to 5) to add capacity to the MLP. We also tried other architectures and replace the MLP with single or multiple transformer blocks.

It is remarkable to see in Tab. 1 that none of these variants significantly outperform the 3-layer MLP architecture. This points to a striking conclusion: the oracle is extremely hard to predict, and the MLP achieves a balance between accuracy and avoiding overfitting.

Model	Top-1	Top-5
MLP selector	65.03	88.52
Edges	64.98	88.92
Isolated peaks	64.83	88.67
Sharpen distribution	64.80	87.98
Global branch	65.10	88.55

Table 1. Results of impact of different variants of data cleaning strategies and integration of global branch. Results are tested with 4K samples of the SS-V2 test set.

Adaptive budget. We test the adaptive budget results on the Kinetics-400 dataset, as detailed in Tab. 2. These results confirm consistency with our previous tests on the SS-V2 dataset. The LITE++ model is promising, enabling us to save nearly 30% of GFLOPs compared to the LITE model, while maintaining the accuracy drop within 0.4.

Model	GFLOPs × views	Top-1
VideoMAE-LITE$_{70}$	118x2x3	81.14
VideoMAE-LITE++$_{70}$	84x2x3$_{\downarrow 29\%}$	80.75$_{\downarrow 0.4}$
VideoMAE-LITE$_{50}$	80x2x3	80.36
VideoMAE-LITE++$_{50}$	62x2x3$_{\downarrow 23\%}$	80.02$_{\downarrow 0.3}$

Table 2. Result comparison between LITE and LITE++ for Kinetics-400 dataset. The blue numbers indicate the reduced percentage of GFLOPs and accuracy.

Adding complexity to the selector. We test more sophisticated models as selectors for token selection, as shown in Tab. 3. The experimental results indicate that using more complex architectures does not significantly improve accuracy but does substantially increase GFLOPs usage. Therefore, using simple 3-layer MLP is the optimal choice in balancing accuracy and GFLOPs .

Model	GFLOPs (Selector / Total)	Top-1	Top-5
3-layer MLP	0.5 / 80	69.91	91.99
5-layer MLP	0.6 / 80	69.79	91.97
1-layer Transformer Block	14.9 / 95	69.96	92.07

Table 3. Results of using different selectors for token selection on the SS-V2 dataset, with a P-Ratio maintained at 0.5. Each video in this specific set of experiments was evaluated using (2 temporal clip x 3 spatial crops) views.

Select tokens from different layers. Table 4 displays the results of our test involving token selection at various posi-

tions within the network. For instance, the experiment with block number of 0 indicates that token selection was performed before the first transformer block. Different block numbers correspond to token selection occurring before various transformer blocks. The experimental results reveal that token selection at the beginning of the network yields the best outcomes, whereas selection in the middle produces the worst results. Additionally, initiating token selection early in the process helps reduce GFLOPs significantly. Therefore, in the LITE model, we position the selector before the first transformer block and conduct token selection at the outset.

Model	Block No.	GFLOPs	Top-1	Top-5
VideoMAE	–	181	70.16	92.12
VideoMAE-LITE$_{50}$	0	80	69.43	90.92
VideoMAE-LITE$_{50}$	2	97	67.19	89.94
VideoMAE-LITE$_{50}$	5	122	66.88	90.05
VideoMAE-LITE$_{50}$	8	147	68.29	90.96

Table 4. Results of token selection before different transformer blocks using the LITE model on the SS-V2 dataset, with a P-Ratio maintained at 0.5. For a quick test, each video in this specific set of experiments was evaluated using (1 temporal clip x 3 spatial crops) views.

Hyperparameters for training the selector. For training the LITE selector, we use the AdamW optimizer with a learning rate of 1e-4, a batch size of 4, and train for 20 epochs.

Composition of GFLOPs. The GFLOPs reported in Tables 1, 2, 3, 5 and 6 in the main paper already included the additional computation from LITE / LITE++. We have included a breakdown in the table below. Both LITE and LITE++ modules require negligible compute, needing only 1% and 3% of the total computation respectively (with P-Ratio of 0.5). We will include this analysis in the revision for further clarity.

	Backbone	LITE	MoviNet	Total
LITE	79.84	0.50	–	80.34
LITE++	63.00	0.50	1.49	64.99

Table 5. Composition of GFLOPs in LITE.

Backbone generalization. The table below shows experiments of another backbone on Video Swin Transformer with SS-V2 dataset. The reported GFLOPs have already accounted for the additional computation introduced by LITE. Results demonstrate that our model achieves strong generalizability across different backbones with different pretraining methods.

Model	GFLOPs × views	Top-1	Top-5
VideoSwin	321x1x3	69.6	92.7
VideoSwin-LITE$_{90}$	288x1x3	69.4	92.6
VideoSwin-LITE$_{80}$	255x1x3	69.0	92.4
VideoSwin-LITE$_{70}$	226x1x3	68.2	92.0

Table 6. Results of LITE Model with Video Swin Transformer on the SS-V2 dataset.

Correlation between MoviNet confidence and video difficulty. The figure below shows a clear correlation between class-wise MoviNet confidence and VideoMAE model accuracy, with a Spearman correlation coefficient of 0.78. Classes with higher accuracy typically exhibit higher MoviNet confidence. Therefore, based on our Principle 5 and the trend illustrated below, it is reasonable to use MoviNet confidence as an indicator of video difficulty.

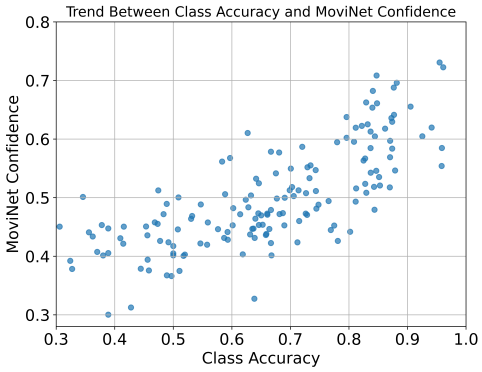

Figure 1. Correlation between class-wise MoviNet confidence and VideoMAE model accuracy on SS-V2 dataset.

2. Additional Visualizations

We present additional visualizations of token selection by our selector from the SS-V2 and Kinetics-400 datasets. These include the original RGB frames, along with the top 50%, top 30%, and top 10% of tokens selected by our selector. The non-white areas indicate the tokens that have been selected. Figures 2 to 5 show the visualization of the Kinetics-400 dataset, each labeled with its class. Figures 6 to 9 show the visualization of the SS-V2 dataset, each labeled with its class.

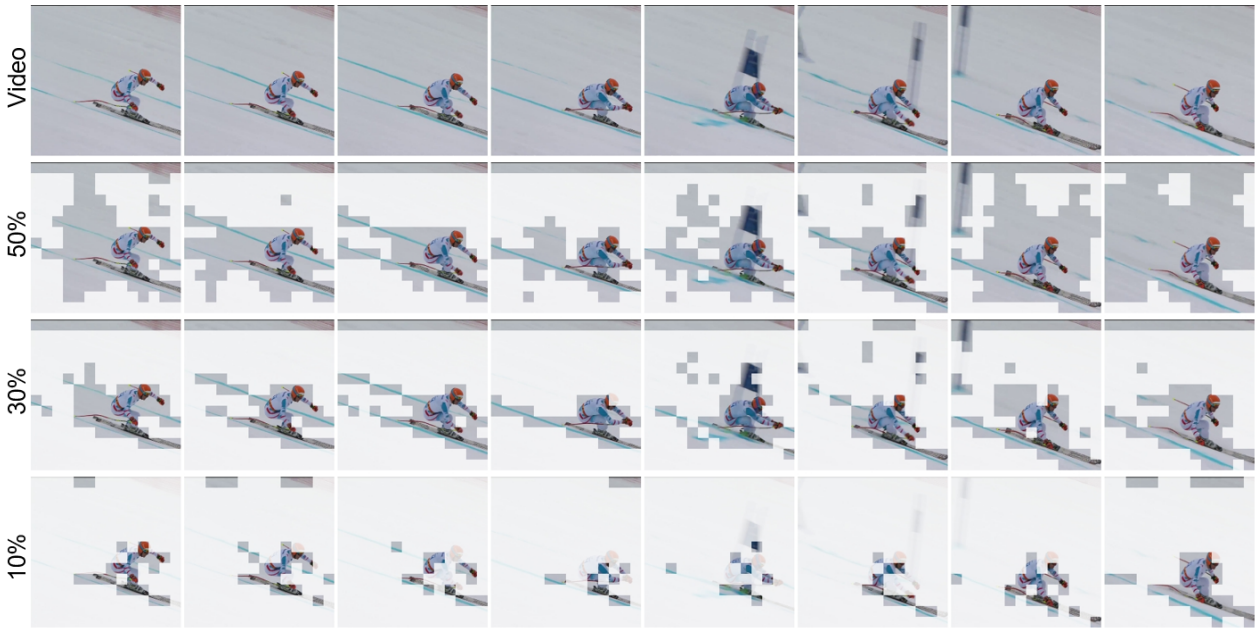

Figure 2. Visualization of token selection by LITE in the Kinetics-400 dataset. Class label: “Skiing slalom”.

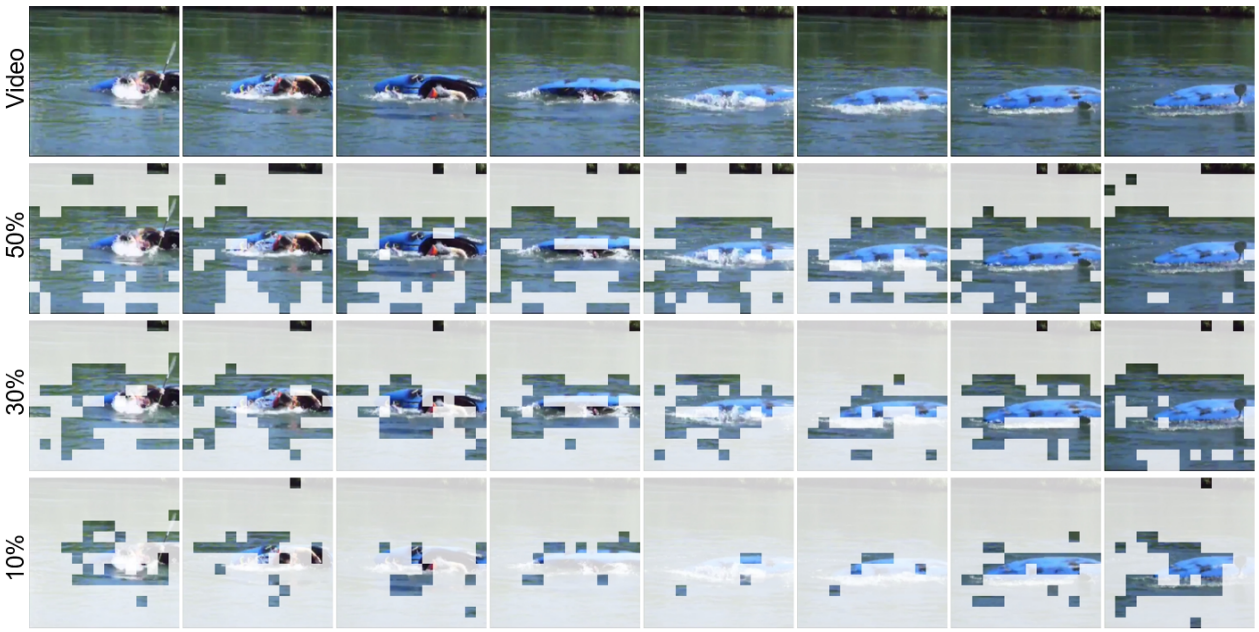

Figure 3. Visualization of token selection by LITE in the Kinetics-400 dataset. Class label: “Canoeing or kayaking”.

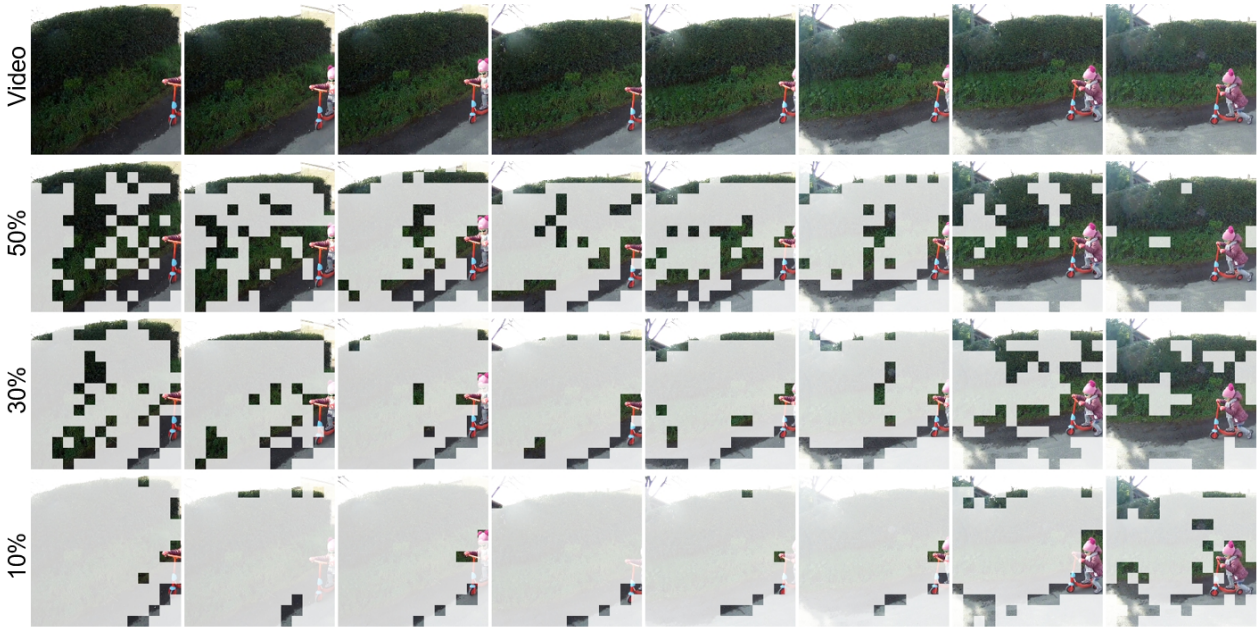

Figure 4. Visualization of token selection by LITE in the Kinetics-400 dataset. Class label: “Riding scooter”.

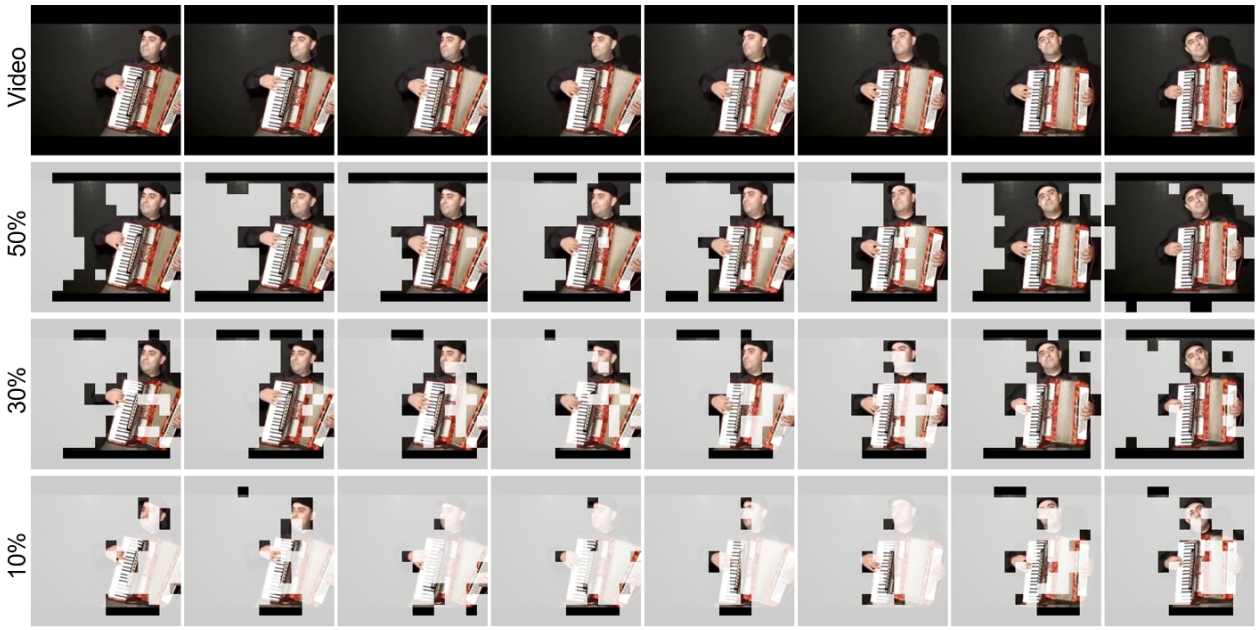

Figure 5. Visualization of token selection by LITE in the Kinetics-400 dataset. Class label: “Playing accordion”.

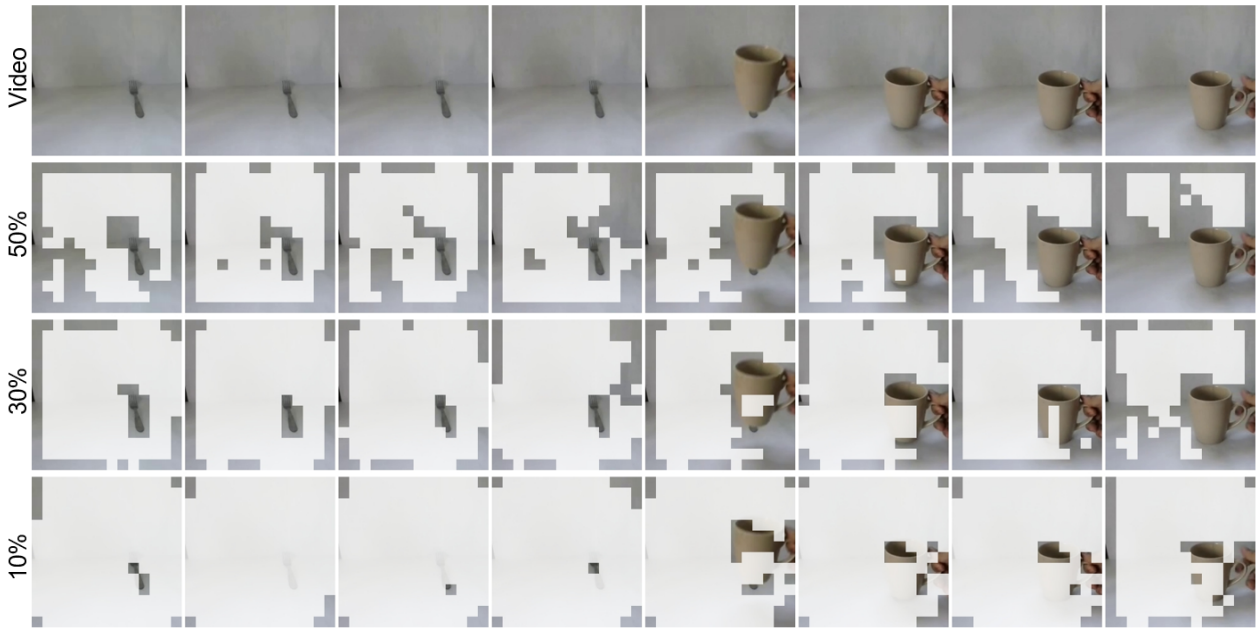

Figure 6. Visualization of token selection by LITE in the SS-V2 dataset. Class label: “Putting something in front of something”.

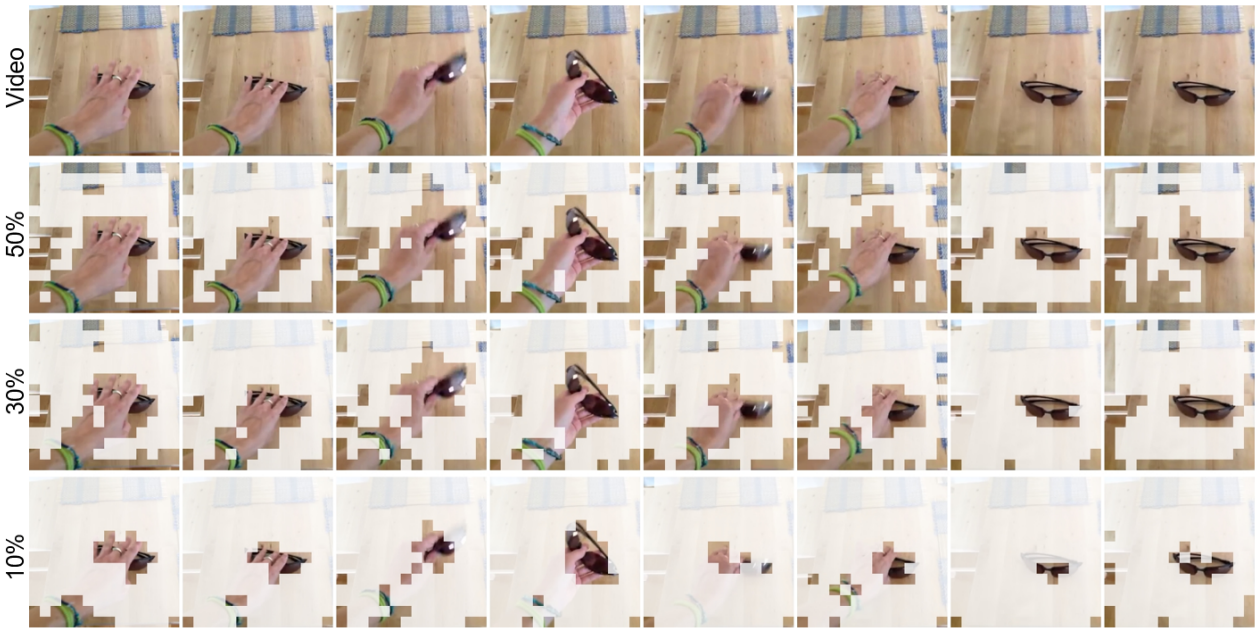

Figure 7. Visualization of token selection by LITE in the SS-V2 dataset. Class label: “Pretending to turn something upside down”.

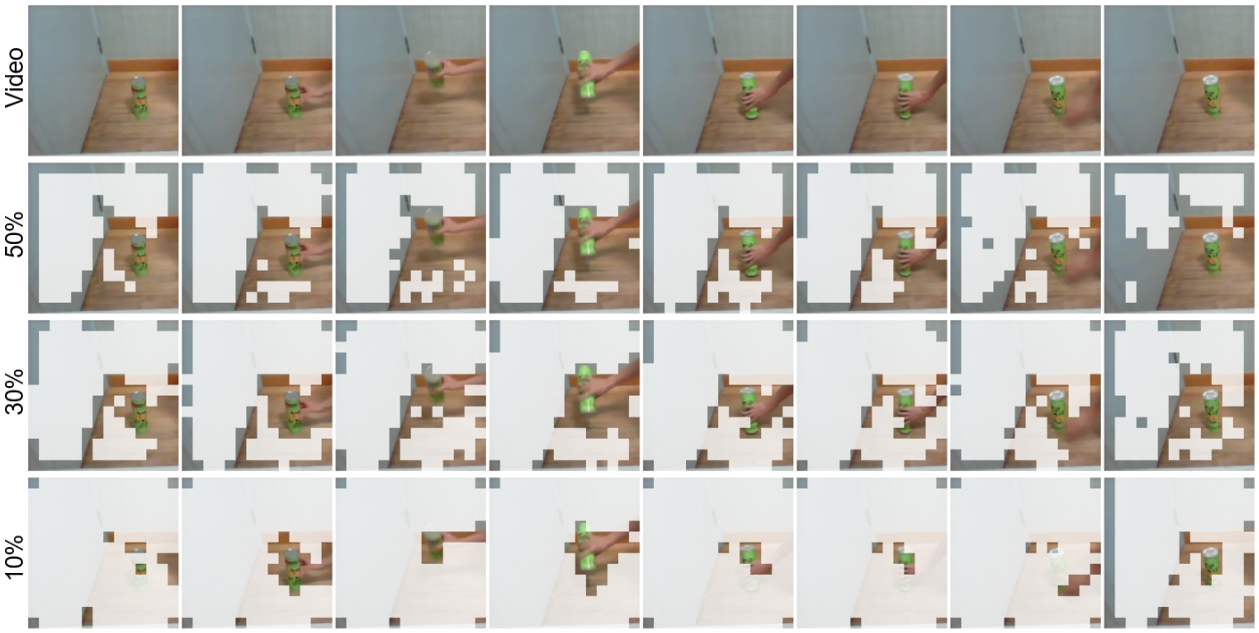

Figure 8. Visualization of token selection by LITE in the SS-V2 dataset. Class label: “Turning something upside down”.

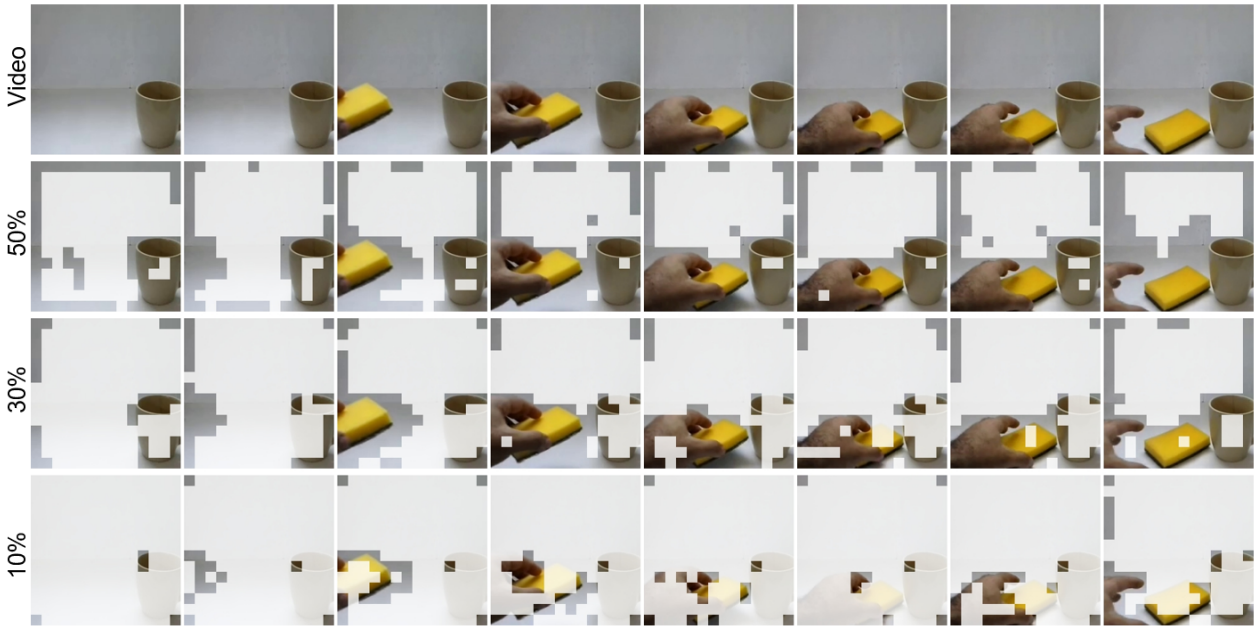

Figure 9. Visualization of token selection by LITE in the SS-V2 dataset. Class label: “Putting something next to something”.