vid-TLDR: Training Free Token merging for Light-weight Video Transformer

Joonmyung Choi* Sanghyeok Lee* Jaewon Chu Minhyuk Choi Hyunwoo J. Kim†

Department of Computer Science and Engineering, Korea University
{pizard, cat0626, allonsy07, sodlqnf123, hyunwoojkim}@korea.ac.kr

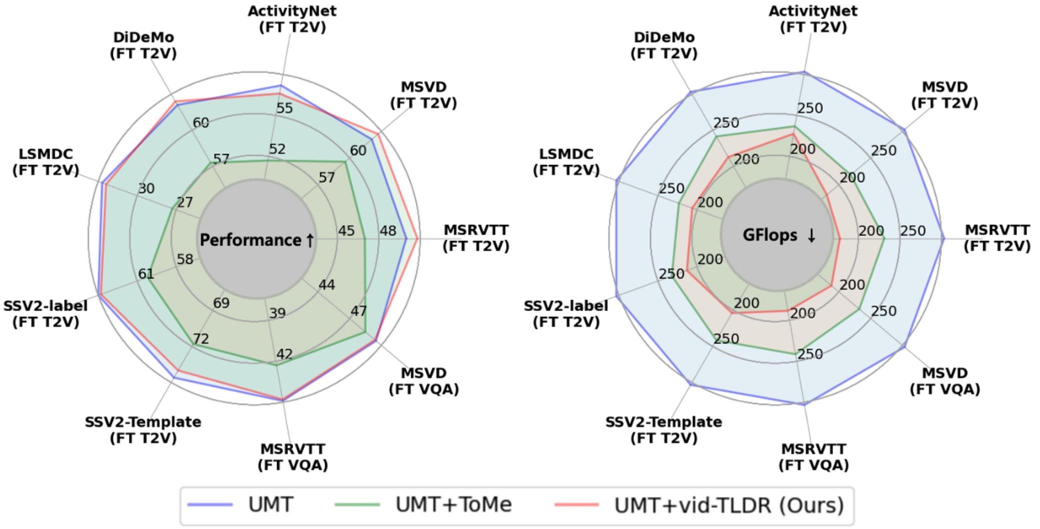


Figure 1. Comparison of vid-TLDR (Ours) with UMT [33]. Without any additional training, vid-TLDR obtains comparable or even better performance than the base model UMT (left) while reducing the considerable computational cost (right). UMT-B (87M) is used.

Abstract

Video Transformers have become the prevalent solution for various video downstream tasks with superior expressive power and flexibility. However, these video transformers suffer from heavy computational costs induced by the massive number of tokens across the entire video frames, which has been the major barrier to train and deploy the model. Further, the patches irrelevant to the main contents, e.g.*, backgrounds, degrade the generalization performance of models. To tackle these issues, we propose training-free token merging for lightweight video Transformer (vid-TLDR) that aims to enhance the efficiency of video Transformers by merging the background tokens without additional training. For vid-TLDR, we introduce a novel approach to capture the salient regions in videos only with the attention map. Further, we introduce the saliency-aware token merging strategy by dropping the background tokens and sharpening the object scores. Our experiments show that vid-TLDR significantly mitigates the computational complexity of video Transformers while achieving competitive performance compared to the base model without vid-TLDR. Code is available at https://github.com/mlvlab/vid-TLDR.*

1. Introduction

With the success of Transformers in computer vision, *e.g.*, classification [14, 52], object detection [10, 32, 43, 61, 75, 77], segmentation [59, 64], a line of works [16, 33, 51, 57, 60, 76] have proposed video Transformers to comprehend the video for various downstream tasks. The attention mechanism in Transformers shows the desirable characteristics for video understanding such as the ability to capture the spatial and temporal dependencies at the same time. Consequently,

*: Equal contribution, †: Corresponding author.

these video Transformers have been the primary backbones for the various downstream tasks in the video domain, including action recognition [65, 73], video-text retrieval [17, 38], video question-answering [18, 63], etc. Meanwhile, the self-attention mechanism entails the dot-product calculation between tokens, which brings the quadratic cost in the number of tokens. This poses a challenge for existing video Transformers like UMT [33] that tokenize the whole video into a large number of tokens.

In the image domain, several works have tried to mitigate the heavy computation of attention by refining the attention itself [6, 25, 58, 66], or limiting the range of attention by a pre-defined window [13, 39]. Yet, these works are not a favorable solution for video Transformers since the methods entail architectural changes, requiring re-training video models with large datasets. As an alternative, in the image domain several works have proposed 'training-free' token reduction methods using the flexibility of Transformers in handling a variable number of input tokens. For instance, prior works [35, 47] simply prune or merge the uninformative tokens to reduce the computational cost based on attentiveness. However, we observed that the existing training-free token reduction methods for images are suboptimal for video transformers. First, previous attention-based informative scores are not accurate enough to use in early layers as discussed in [35]. So, the token reduction cannot be performed at earlier layers. Further, the attention scores of video Transformers contain a temporal bias, which makes it difficult to directly adopt them as the informativeness of the tokens, see Figure 2.

Based on these observations, we propose vid-TLDR, **T**raining-free token merging for **L**ight-weight vi**D**eo TransformeR, to effectively merge the tokens through two steps. First, we conduct *saliency detection via attention sharpness*. We observe that our proposed metric understands the salient region, which is more informative than backgrounds, even with the attention map in the first layer of Transformers. We also introduce the *saliency-aware token merging*, a training-free plug-in module to suppress the tokens irrelevant to the target tasks. Through saliency-aware token merging, we effectively drop the information of tokens in backgrounds and further contrast the informativeness of the foreground objects. Based on these components, we minimize the hindrance by irrelevant tokens from the early layers of video Transformers. Through experiments, we show that, without any additional training, the adoption of vid-TLDR brings performance improvements of (+0.8%, +0.5%, +1.1%) with at least 39.5% lower FLOPs in UMT-B [33] on MSRVTT [68], MSVD [11], DiDeMo [2], respectively. To summarize, the contributions of vid-TLDR are presented as follows:

- We propose the novel token merging method vid-TLDR, which reduces the tokens irrelevant to target tasks from the early layers of the video Transformer.
- We detect the salient region of videos based on the sharpness of the attention scores even from the first layer.
- Based on the saliency scores, we also propose saliency-aware token merging with the masked saliency scores for adaptively adjusting the informativeness of the tokens.
- vid-TLDR shows the competitive performance with the baselines across four benchmarks in video-text retrieval and two benchmarks in video question-answering. It is worth noting that vid-TLDR even shows superior performance while reducing the computational complexity.

2. Preliminaries

In this section, we briefly review the video Transformers and token reduction approaches, then introduce techniques to measure the informativeness of tokens using the attention map of the video Transformers.

Video Transformer. In Transformers [53], the self-attention mechanism is defined as

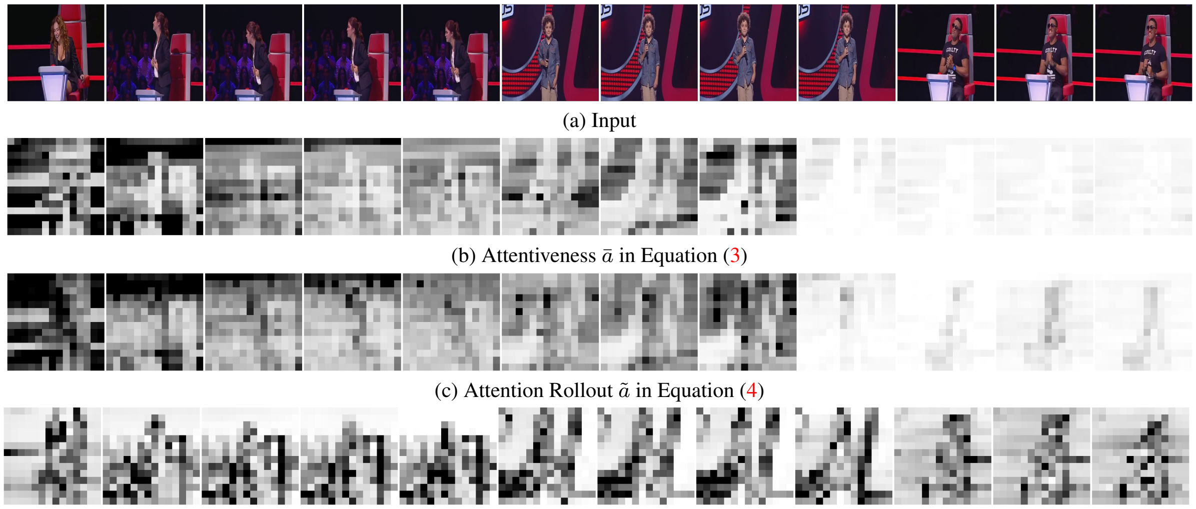
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{C}}\right)V, \tag{1}$$

where $Q, K, V \in \mathbb{R}^{N\times C}$ are the projection of the tokens $X \in \mathbb{R}^{N\times C}$ by learnable matrices $W_Q, W_K, W_V \in \mathbb{R}^{C\times C}$. Given a video clip composed of T frames in the resolution of $H \times W$, video Transformers first generate the tokens $X \in \mathbb{R}^{N\times C}$ by considering the clip as the set of tubes, where $N = \frac{T}{t} \times \frac{H}{P} \times \frac{W}{P}$, and $(t \times P \times P)$ is the size of each tube. All tokens in the video Transformers interact with others across the frames by spatio-temporal attention. Despite the advantage of capturing both spatial and temporal dependencies, it also demands enormous computational resources to handle the large number of tokens. Compared to one image, the computational cost for a clip is increased by $(\frac{T}{t})^2$ times since the cost of attention is quadratic in the number of tokens. This cost further increases as the number of frames per tube t decreases.

Token Reduction Methods. Based on the flexibility of Transformers in the number of tokens, token reduction approaches [8, 15, 26, 36, 47] reduce the intermediate tokens by pruning or merging them, leading to the lower computational cost $O((N')^2C + N'C^2)$, where $N' < N$. To minimize information loss after reduction, they mainly prune/merge the tokens based on the attentiveness defined as

$$a_{\text{cls}} = \text{softmax}\left(\frac{q_{\text{cls}}K^\top}{\sqrt{C}}\right), \tag{2}$$

where q_{cls} is the query vector of the class token. Prior works achieve a competitive performance with the original model through additional training. In parallel, ToMe [8] has demonstrated the possibility of training-free token merging in the image domain using the similarity of the tokens. This simple



(a) Input

(b) Attentiveness $\bar{a}$ in Equation (3)

(c) Attention Rollout $\tilde{a}$ in Equation (4)

(d) Ours s in Equation (5)

Figure 2. Visualization of the attention map of each method in the first layer. Both attentiveness $\bar{a}$ and attention Rollout $\tilde{a}$ confused with foreground objects and the background, also have a temporal bias, resulting in overall low attention in the later frames. These problems are mitigated in our method, focusing on the object across all frames.

plug-in approach is favorable for video Transformers considering its huge complexity, yet the capability to capture the informativeness of the tokens is absent.

Informativness of tokens. Previous works [35, 47], due to the low reliability of the attention map in the early stage, could not reduce the tokens in the earlier layers. However, we believe that the early pruning/merging is desirable for video Transformers from two perspectives: 1) it prevents the interaction between the tokens irrelevant to the main contents by only retaining the salient tokens, and 2) it largely alleviates the complexity of the entire layers with fewer tokens from the beginning of Transformer. To validate this, we explore whether the attention map is a sufficient approximation of the informativeness estimator even in the first layer. Note that, due to the absence of the class tokens in recent video Transformers, we modify Equation (2) by summarizing the whole query vector as

$$\bar{a} = \frac{1}{N}\sum_{i}^{N} A_i, \tag{3}$$

where A_i is the i-th row vector of the attention matrix $A = \text{softmax}\left(\frac{QK^\top}{\sqrt{C}}\right)$. Further, we visualize the attention rollout [1], which is the well-known saliency detector by quantifying the flow of attention from the tokens in the l-th layer to output, defined as

$$\tilde{A}^l = \overset{\curvearrowleft}{\prod}{}_{i=l}^{L} A^i \text{ and } \tilde{a}^l = \frac{1}{N}\sum_{i}^{N} \tilde{A}^l_i, \tag{4}$$

where L is the number of layer in Transformers, and $A^i \in \mathbb{R}^{N\times N}$ is the attention map in i-th layer. As shown in Figure 2, the attentiveness $\bar{a}$ failed to capture salient tokens due to the low reliability of the attention map in the early stage, and attention rollout $\tilde{a}$ also largely confused the foreground objects. Further, we have observed the temporal biases of video Transformers, *e.g.*, the later frames exhibit lower activation regardless of the importance of the frames.

3. Method

We introduce vid-TLDR, **T**raining free token merging for **L**ight-weight vi**D**eo transforme**R**. The goal of vid-TLDR is to effectively merge the tokens from the early stage by two steps: 1) Saliency detection via attention sharpness (Section 3.1), 2) Saliency-aware token merging (Section 3.2).

3.1. Saliency detection via attention sharpness

As discussed in Section 2, existing works do not reduce the tokens in the first few layers because of the low reliability. However, we believe that tokens irrelevant to the target tasks should be reduced as early as possible to minimize their adverse influence. To this end, we analyze the attention scores $A_i = \text{softmax}(\frac{q_i K^\top}{\sqrt{C}})$ in the first self-attention layer concerning the foreground and background tokens. Figure 4 reveals that the background tokens are quite equally affected by neighboring tokens, whereas the tokens of foreground objects gather the information from more specific tokens showing sharper attention scores compared to backgrounds. Based on this observation, we devise a *sharpness function* S to capture the saliency of tokens with entropy. Specifically, using the negative entropy given as $H_i = \sum_j^N A_{ij} \log A_{ij}$, we define the sharpness function S as

$$s_i = S(H_i) = \frac{H_i - \min(H)}{\max(H) - \min(H)}, \tag{5}$$

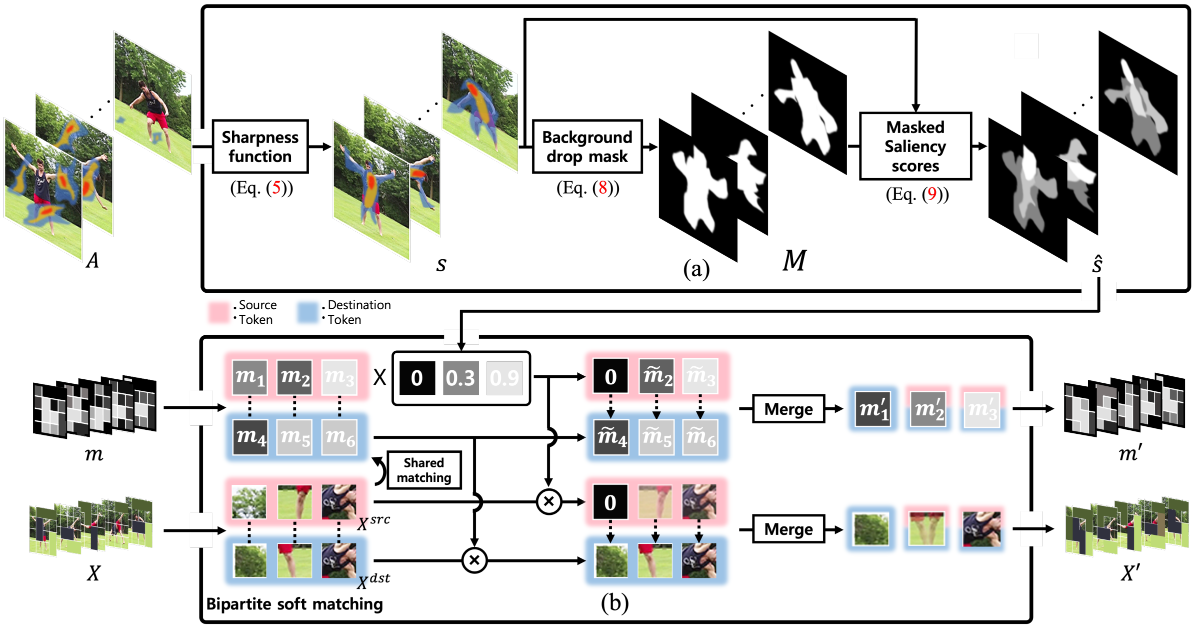


Figure 3. Pipeline of **vid-TLDR**. (a) Given the attention map A, the saliency score s is approximated by the *sharpness function* S (Eq. (5)). After that, we generate *background drop* mask M (Eq. (8)) to minimize the disturbance of background tokens. With s and M, we generate *masked saliency scores* $\hat{s}$ (Eq. (9)). (b) Given the tokens X and their corresponding mass m, we conduct the matching to group the input tokens. Following that, we update the mass m to $\tilde{m}$ with $\hat{s}$ (Eq. (10)) to highlight important foreground tokens and minimize the hindrance of background tokens. With updated mass $\tilde{m}$, the grouped tokens are merged into a token m' and X' (Eq. (11))

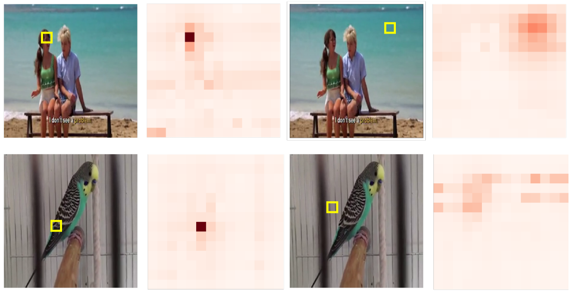


Figure 4. Visualization of attention scores in the first layer of UMT-B [33]. Given the query q_i denoted as a yellow square, we visualize the attention score a_i. Tokens in the foreground objects show a sharper attention map compared to the background tokens.

Layer	GFLOPs	T2V	V2T	Mean
-	303.3	**51.0**	49.0	50.0
1	**237.6**	49.6	**51.1**	**50.4**
2	243.1	49.6	50.8	50.2
3	248.6	49.9	50.5	50.2
4	254.0	49.6	50.7	50.2
5	259.5	49.7	50.6	50.2
6	265.0	49.6	50.7	50.2
7	270.5	48.8	50.1	49.5

Table 1. The comparative study of where to reduce the tokens.

where $H = [H_1, \ldots, H_N]$. Since an informative foreground token has low entropy, its saliency score s_i is usually high.

Remarks. To validate the saliency scores s, we conduct the experiments by pruning the 400 tokens in each layer of UMT [33] on video-text-retrieval task with MSRVTT [68]. In Table 1, we demonstrate that the pruning by the proposed score accelerates the base model (first row) with the competitive performance across layers. Interestingly, the token reduction in the earlier layers is more effective. To be specific, the token reduction in the first layer shows the best results (50.4%) with the lowest FLOPs (237.6 (G)). Based on this observation, we mainly applied our token reduction to earlier layers. Qualitative results in Figure 2d show that the saliency scores in the first layer successfully detect the salient region.

3.2. Saliency-aware token merging

Given the saliency scores $s = [s_1, ..., s_N]$ of the tokens, our goal is to merge tokens while suppressing the influence of the irrelevant tokens with low saliency scores. We start with a brief review of a training-free token reduction method, ToMe [8]. ToMe splits tokens X into two sets X^{src}, $X^{\text{dst}} \subset X$ and performs the bipartite soft matching between the two sets to form token groups. For each group $\mathcal{G}_i$, the features are aggregated as

$$x'_i = \sum\nolimits_{j\in\mathcal{G}_i} \frac{m_j x_j}{\sum_{j'\in\mathcal{G}_i} m_{j'}}, \tag{6}$$

where $\{x_i\}_{i\in\mathcal{G}_i} \subset X, \forall_{\mathcal{G}_i,\mathcal{G}_j} \mathcal{G}_i \cap \mathcal{G}_j = \emptyset$, and m_i is the mass of token x_i. Then, the attention is also refined with m as

$$A'_{ij} = \text{softmax}\left(A_{ij} + \log m_j\right), \tag{7}$$

Dataset	Metric	UMT-B			UMT-L		
		Base	ToMe	Ours	Base	ToMe	Ours
MSRVTT	GFLOPs ↓	303.3	231.4	**178.0**	984.6	**529.7**	563.1
	R@1 ↑	50.0	47.0 (−3.0)	**50.8** (+0.8)	58.7	55.8 (−2.9)	**58.5** (−0.2)
	R@5 ↑	76.8	73.2 (−3.6)	**75.7** (−1.1)	81.3	79.6 (−1.7)	**81.3** (±0.0)
	R@10 ↑	83.9	82.1 (−1.8)	**83.8** (−0.1)	86.8	86.1 (−0.7)	**86.9** (+0.1)
MSVD	GFLOPs ↓	303.3	218.7	**181.3**	984.6	574.5	**563.1**
	R@1 ↑	62.1	59.6 (−2.5)	**62.7** (+0.6)	70.3	69.5 (−0.8)	**70.4** (+0.1)
	R@5 ↑	84.7	83.8 (−0.9)	**84.8** (+0.1)	89.3	88.7 (−0.6)	**90.5** (+1.2)
	R@10 ↑	90.0	89.0 (−1.0)	**89.8** (−0.2)	93.2	92.7 (−0.5)	**94.0** (+0.8)
ActivityNet	GFLOPs ↓	303.3	236.8	**227.6**	984.6	574.5	**572.9**
	R@1 ↑	57.2	51.7 (−5.5)	**56.6** (−0.6)	65.6	62.5 (−3.1)	**65.2** (−0.4)
	R@5 ↑	83.7	80.7 (−3.0)	**83.4** (−0.3)	89.1	86.9 (−2.2)	**88.7** (−0.4)
	R@10 ↑	91.6	89.6 (−2.0)	**91.3** (−0.3)	94.9	93.6 (−1.3)	**94.5** (−0.4)
DiDeMo	GFLOPs ↓	303.3	241.4	**212.8**	984.6	574.5	**559.0**
	R@1 ↑	62.1	57.3 (−4.8)	**62.4** (+0.3)	70.8	68.0 (−2.8)	**70.4** (−0.4)
	R@5 ↑	86.8	82.6 (−4.2)	**86.2** (−0.6)	90.6	89.4 (−1.2)	**90.5** (−0.1)
	R@10 ↑	92.1	89.3 (−2.8)	**91.6** (−0.5)	94.5	93.8 (−0.7)	**94.0** (−0.5)
LSMDC	GFLOPs ↓	303.3	223.2	**206.2**	984.6	574.5	**583.7**
	R@1 ↑	32.7	27.3 (−5.4)	**32.4** (−0.3)	42.2	39.2 (−3.0)	**41.9** (−0.3)
	R@5 ↑	54.1	49.1 (−5.0)	**53.3** (−0.8)	64.9	61.6 (−3.3)	**64.1** (−0.8)
	R@10 ↑	63.3	57.3 (−6.0)	**63.2** (−0.1)	72.3	68.7 (−3.6)	**70.8** (−1.5)
SSV2-label	GFLOPs ↓	303.3	232.2	**212.9**	984.6	627.2	**610.9**
	R@1 ↑	64.0	60.2 (−3.8)	**63.8** (−0.2)	72.4	69.9 (−2.5)	**72.1** (−0.3)
	R@5 ↑	88.3	86.1 (−2.2)	**87.7** (−0.6)	93.4	92.2 (−1.2)	**93.0** (−0.4)
	R@10 ↑	92.9	91.6 (−1.3)	**92.7** (−0.2)	96.7	95.8 (−0.9)	**96.5** (−0.2)
SSV2-Template	GFLOPs ↓	303.3	241.4	**203.7**	984.6	627.2	**572.9**
	R@1 ↑	74.6	71.8 (−2.8)	**74.0** (−0.6)	78.4	77.0 (−1.4)	**78.1** (−0.3)
	R@5 ↑	93.9	**93.9** (±0.0)	93.4 (−0.5)	95.9	95.1 (−0.8)	**95.8** (−0.1)
	R@10 ↑	96.8	**96.6** (−0.2)	96.3 (−0.5)	97.8	97.7 (−0.1)	**97.9** (+0.1)

Table 2. Video-text retrieval on MSRVTT [68], MSVD [11], ActivityNet [9], DiDeMo [2], LSMDC [49], SSV2-Label/Template [30]. Underlined results indicate the number reported with the official repository of UMT [33] that corrects the misconfiguration of it.

where A_{ij} is the element in i-th row and j-th column. For the mass m of the tokens, ToMe uses the number of constituent tokens. Although it has been proven effective in alleviating redundancies, it cannot adaptively adjust the influence of merged tokens considering its importance. We here propose a **saliency-aware token merging** that estimates the mass reflecting the saliency of the tokens and then merges the tokens with their corresponding mass to minimize the hindrance induced by the uninformative tokens. First, we introduce the *background drop* mask to update the mass of foreground and background tokens selectively. Given the saliency scores, we define the mask as

$$M_i = \mathbf{1}_{\{s_i > \bar{s}\}}, \tag{8}$$

where $\mathbf{1}$ is the indicator function and $\bar{s} = \frac{1}{N}\sum_i^N s_i$. Using the mask above, our framework sets the saliency scores to 0 if a token has a saliency score lower than the average saliency score. With M, we define *masked saliency scores* $\hat{s}$ to focus more on informative tokens among the foreground objects by masking and rescaling the saliency scores as

$$\hat{s}_i = \frac{M_i(s_i - \bar{s})}{\max(M_i(s_i - \bar{s}))}. \tag{9}$$

Yet, we have one more issue while adopting $\hat{s}$ as the mass. If all tokens in the group $\mathcal{G}_i$ have the saliency scores lower than $\bar{s}$, feature aggregation may result in a zero vector losing the entire information. So, to prevent this, we compute the mass with $\hat{s}$ as

$$\tilde{m}_i = \begin{cases} \hat{s}_i m_i, & \text{if } x_i \in X^{\text{src}} \\ m_i & \text{if } x_i \in X^{\text{dst}} \end{cases}. \tag{10}$$

Then, feature merging of Equation (6) is modified as

$$x'_i = \sum\nolimits_{j\in\mathcal{G}_i} \frac{\tilde{m}_j x_j}{\sum_{j'\in\mathcal{G}_i}\tilde{m}_{j'}} \text{ and } m'_i = \sum\nolimits_{j\in\mathcal{G}_i}\tilde{m}_j, \tag{11}$$

where m'_i is the mass of i-th fused token. It is worth noting that this saliency-aware token merging can be viewed as feature aggregation only with foreground tokens since the scores of the background tokens are set to 0. In other

Method	#Pairs	MSR.	MSVD	Act.	DiDe.
ClipBERT [29]	5.4M	22.0	-	21.3	20.4
Frozen [5]	5M	31.0	33.7	-	34.6
VIOLET [16]	138M	34.5	-	-	32.6
All-in-one [56]	138M	37.9	-	22.4	32.7
LAVENDER [34]	30M	40.7	50.1	-	53.4
Singularity [30]	17M	42.7	-	48.9	53.1
OmniVL [55]	17M	47.8	-	-	52.4
VINDLU [12]	25M	46.5	-	55.0	61.2
CLIP4Clip [40]	400M	44.5	46.2	40.5	42.8
CLIP-ViP [69]	500M	54.2	-	53.4	50.5
InternVideo [60]	646M	55.2	**58.4**	62.2	57.9
UMT-B [33]	25M	51.0	50.8	58.3	63.7
UMT-L [33]	25M	**58.8**	58.2	**66.8**	**72.5**
UMT-B-Ours	25M	50.9	50.5	57.8	64.1
UMT-L-Ours	25M	58.1	57.9	66.7	72.3

Table 3. Text-to-video retrieval on MSRVTT (MSR.) [68], DiDeMo (DiDe.) [2], ActivityNet (Act.) [9], MSVD [11]. "#Pairs" denotes the number of pre-training pairs. We use the models of each dataset in Table 2 to report ours.

Method	#Pairs	GFLOPs	MSR-QA	MSVD-QA
ALPRO [31]	5M	392.5	42.1	45.9
JustAsk [71]	69M	340.7	41.5	47.5
All-in-one [56]	138M	1017.0	44.3	47.9
MERLOT [76]	180M	-	43.1	-
VIOLET [16]	138M	282.0	43.9	47.9
Singularity [30]	17M	211.0	43.9	-
OmniVL [55]	17M	-	44.1	51.0
VINDLU [12]	25M	278.5	44.6	-
FrozenBiLM [72]	400M	340.7	47.0	54.8
InternVideo [60]	646M	666.2	**47.1**	55.5
VideoCoCa [70]	4.8B	29820	46.0	**56.9**
UMT-B [33]	25M	303.3	44.9	49.5
UMT-L [33]	25M	984.6	**47.1**	55.2
UMT-B-Ours	25M	**188.5**	44.8	49.4
UMT-L-Ours	25M	569.8	47.0	54.9

Table 4. Video question-answering on MSRVTT-QA [67] & MSVD-QA [67].

words, it is a combination of token merging and token pruning. Although the same number of tokens are merged in each video, we dynamically adjust the influence of uninformative tokens by suppressing the mess, leading to promising improvements by the proposed saliency-aware token merging, see Section 4.2.

4. Experiments

Baselines. To show that vid-TLDR effectively boosts video Transformer, we opt for UMT [33] as the baseline, which achieves state-of-the-art performance on various video tasks. Since vid-TLDR is the training-free plug-in module, we simply apply it right after the self-attention in the early layer of UMT and evaluate it without any additional training. We conduct vid-TLDR in the first four layers. The detailed reduced number of tokens for each dataset is provided in the supplement. In multi-modal tasks, we adopt vid-TLDR on the vision encoder of UMT. Except for the reduced number of tokens, we conduct whole experiments under the same evaluation settings of UMT. We also report the results with the previous merging method, ToMe [8], which can be added to the pre-trained video Transformer. For the settings of ToMe, we respect the default setups, where the same number of tokens are merged based on the similarity in every layer. For a fair comparison, we try to maintain similar FLOPs.

4.1. Experimental results.

Video-text retrieval First, we summarize the results of video-text retrieval with MSRVTT [68], MSVD [11], ActivityNet [9], and DiDeMo [2], LSMDC [49], Something-Something [30]. Video-text retrieval contains two subtasks: video-to-text retrieval, and text-to-video retrieval. Video-to-text retrieval is to find the most relevant text concerning the given video, while text-to-video retrieval is conducted in the opposite direction. We report the average of the results in Table 2. As summarized, our proposed method consistently shows competitive performances compared to base UMT [33] and outperforms ToMe[7] in every backbone and dataset. Compared to ToMe, vid-TLDR shows a performance gap of (+4.0%, +2.1%) on average R@1 in UMT-B, UMT-L even with the lower FLOPs. Further, vid-TLDR with UMT-B even surpasses the base model with the improvements of (+0.8%, +0.6%, +0.3%) R@1 while reducing FLOPs by (41.3%, 40.2%, 29.8%) in MSRVTT, MSVD, DiDeMo, respectively. And, we also observe that vid-TLDR achieves competitive performance with base UMT-L despite the much lower FLOPs. We further provide the comparison with other video backbones in text-to-video retrieval (Table 3). For reporting the table, we experiment with the model used in Table 2. Although we largely reduce the computational cost of UMT-B, and UMT-L by 34.1%, 42.7% on average, they still show superior performance compared to other video backbones in MSRVTT, DiDeMo, and ActivityNet.

Video question answering. We experiment with video question answering with MSR-QA [67] and MSVD-QA [67], summarizing the results in Table 4. In MSR-QA, the results of UMT-B and UMT-L are 44.9% and 47.1%, respectively. After adopting vid-TLDR on each model, we achieved the competitive performance of 44.8% and 47.0% while reducing FLOPs by 37.9% in UMT-B, and 42.1% in UMT-L. Similarly, we could lessen the computational cost in MSVD-QA with the small performance degradation despite the much lower FLOPs. Specifically, the performance drop is only 0.1% and 0.3% in UMT-B and UMT-L with much lower FLOPs compared to the original model. To summarize, vid-TLDR boosts the model with a minor accuracy drop in video

Method	UCF101		SSV2	
	GFLOPs	Acc	GFLOPs	Acc
VideoMAE [51]	180.5	91.3	180.5	70.8
+ToMe [8]	58.4	75.7	58.4	58.5
+vid-TLDR	56.1	90.1	56.6	69.6

Table 5. Action recognition with VideoMAE on UCF101 [50] & Something Something V2 [19]

Method	**FLOPs**	**Base**	**Novel**	**HM**
ViFi-CLIP [48]	563	92.9	67.7	78.3
+ToMe [8]	279	72.0	44.8	55.3
+vid-TLDR	279	91.3	63.2	74.7

Table 6. Base-to-novel generalization on UCF101 [50]

question answering as well as video-text retrieval.

vid-TLDR with other video Transformers and tasks. To demonstrate the generalizability of vid-TLDR, we apply vid-TLDR to other video Transformers: VideoMAE [51], and ViFi-CLIP [48]. The results of action recognition with VideoMAE are presented in Table 5. We use Something Something V2 (**SSV2**) [19] and UCF101 [50]. Experimental results are promising that vid-TLDR lessens the complexity of VideoMAE by almost 70% (180.5 (G) $\rightarrow$ 56.1 (G)) with a minimal performance degradation compared to ToMe (*e.g.*, 14.4% (75.7 $\rightarrow$ 90.1) improvement over ToMe). Further, we adopt vid-TLDR on ViFi-CLIP in the base-to-novel generalization tasks (*i.e.*, training only with the base classes, then evaluating on both seen (base). As shown in Table 6 with UCF-101, vid-TLDR halving the FLOPs (563 $\rightarrow$ 279) of ViFi-CLIP surpassing ToMe with a 19.4% (55.3% $\rightarrow$ 74.7%) gain in the harmonic mean (**HM**).

4.2. Ablation studies

In this section, we provide the ablations studies of vid-TLDR. We studied the effectiveness of each component with UMT-B [33] and MSRVTT [68] (Table 7). The first row of the table indicates the base model without any token reduction. First, regarding the metric for informativeness, we have compared our proposed saliency scores s in Equation (9) with the attentiveness $\bar{a}$ in Equation (3), and attention rollout $\tilde{a}$ in Equation (4). For comparison, based on each metric, we simply prune the token having the low scores, equal to the number used for reporting Table 2. As shown in the table, since the tokens are dropped in the earlier layers, the attentiveness and attention rollout may drop the salient tokens resulting in a substantial performance drop in both text-to-video retrieval and video-to-text retrieval. Specifically, the performance degradation on average is 1.0%, 0.7% when using $\bar{a}$ and $\tilde{a}$. Further, although attention rollout shows slightly better performance than simple attentiveness, it shows worse FLOPS due to the repeated forward process for estimating attention flows. On the other hand, by dropping the tokens based on our saliency scores s estimated by the sharpness of attention, we could lessen the FLOPs with superior performance compared to base. Finally, introducing the saliency-aware token merging, we have achieved the +0.8% (50.0% $\rightarrow$ 50.8%) gain despite the 58.7% FLOPs compared to the original UMT.

$\bar{a}$	$\tilde{a}$	s	S.A. ToMe	GFLOPS	T2V	V2T	Mean
-	-	-	-	303.3	**51.0**	49.0	50.0
✓				**175.9**	50.8	47.2	49.0
	✓			479.2	50.2	48.3	49.3
		✓		**175.9**	50.6	50.0	50.3
		✓	✓	178	50.9	**50.7**	**50.8**

Table 7. Ablations studies on vid-TLDR. The first row indicates the base UMT-B [33]. S.A. ToMe denotes the saliency-aware token merging.

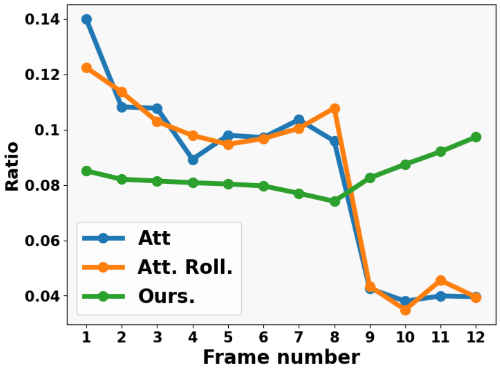


Figure 5. The ratio of the sum of the scores in each frame. Given the informativeness scores in the first layer of the video Transformer, we extract the sum of scores in each frame and then calculate the ratio to the total scores across all the frames.

4.3. Analysis

Temporal bias. As we discussed in in Section 2, the video Transformers contains the temporal bias that neglects the later frames of a video. For a better understanding, we here quantitatively compare three metrics, attentiveness $\bar{a}$ (Att.), attention rollout $\tilde{a}$ (ATT. Roll.), and our saliency scores $\hat{s}$. We measure each score with UMT-B [33] in MSRVTT [68]. After normalizing the score across all frames, we calculate the sum of the scores for each frame. In other words, it represents the ratio of scores per each frame concerning the total score. (see Figure 5) In both attentiveness and attention rollout, the earlier frames show higher scores compared to later frames, specifically, the score of the first frame is almost $\times 3$ times higher than the last frame. As a result, if we rely on these metrics, it is prone to merge the tokens in the later frame without considering their informativeness of them. Whereas, as shown in the figure, our proposed $\hat{s}$ shows the more robust ratio across the frame, capturing the informative tokens even in the last frame.

Visualization of vid-TLDR results. In Figure 6, we show

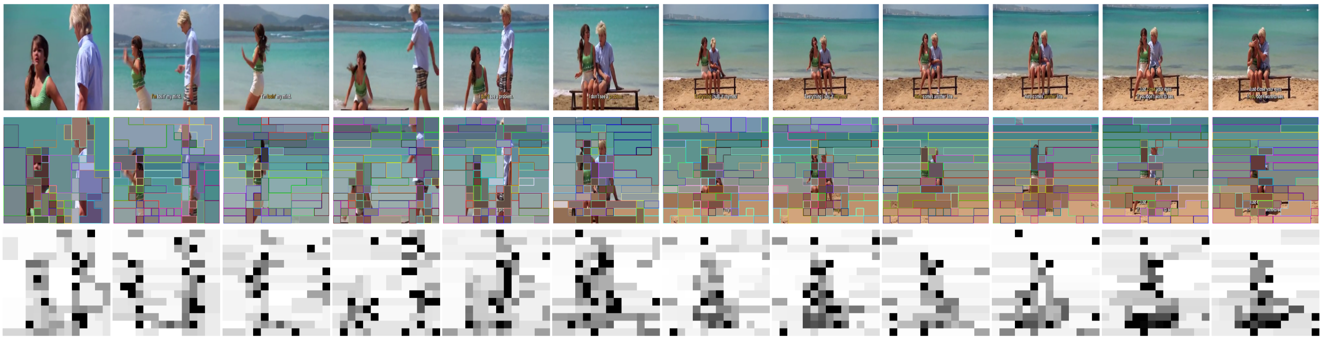


Figure 6. Qualitative results of vid-TLDR. Given video clips, we visualize the merged tokens and their corresponding mass.

the qualitative results to understand the behavior of vid-TLDR. Given video clips of MSRVTT, we visualize the merged token. We further provide the heatmaps for analyzing the mass of each token. More precisely, we divide the mass of merged tokens by the number of constituent tokens to represent the mass concerning each input token. As shown in the figure, through the saliency-aware token merging, the foreground object shows a much higher mass than the background tokens. In short, through Equation (7), our vid-TLDR minimizes the hindrance from the background tokens during the self-attention layer.

5. Related works

Vision Transformer. ViT [14] has become one of the most popular and basic components in computer vision along with CNNs. With the surge of large-scale datasets, its lower inductive bias compared to CNNs endows it with robust generalization, leading to another successive adoption on several downstream tasks in computer vision, including classification [14, 52], detection [10, 32, 43, 61, 75, 77] segmentation [59, 64], image encoding [3, 21, 62] for generation [28]. Although Transformer shows promising results on computer vision the attention mechanism [4] incurs quadratic complexity and restrains its scalability. So, there have been many attempts to mitigate this problem. These attempts can be categorized as ameliorating model architecture itself, leveraging pre-trained models, or reducing the input tokens. For example, [23, 25, 54, 58, 74] focused on the attention mechanism itself to approximate the complexity to linear. More recent works enabled acceleration even without model modification by pruning [24, 27, 44] or merging [7, 8, 42, 45] input tokens, with minimal performance degradation.

Video understanding. Video understanding is not the same as the image, since frames of video are not independent images, but highly related to each other in the temporal axis. Prior works have leveraged transformers for understanding video, including retrieval [12, 16, 30, 33, 40, 55, 56, 70], question answering [12, 30, 31, 55, 56, 60, 70, 71, 76], captioning [20, 37] and representation learning [22, 51, 57]. Especially, along with the success of image foundation models, some works [41, 48] leverage CLIP [46], which is pretrained ViTs with the large-scale image-text pairs, for understanding video. Also, recent studies [33, 51, 57, 60, 76] focus on scaling the Transformers for video foundation models to utilize the flexibility for multi-modal tasks. Yet, despite the intensive computational cost caused by the massive number of tokens, efficient video Transformers are less explored.

6. Conclusion

In this paper, we propose vid-TLDR, training free token merging for light-weight video Transformer. We demonstrate the necessity of performing early token merging in video Transformers and delineate the associated challenges. To address these challenges, we design the new token-merging mechanism as follows. First, we devise a better saliency detector using attention sharpness which can localize salient regions even in the early layers of the Transformer and mitigate the temporal biases of video Transformers. In addition, we revise the mass of token merging so that the influence of uninformative tokens is suppressed and the importance of tokens within the foreground objects is taken into account. The experiments show that our method consistently outperforms previous merging methods in every backbone and dataset even with lower computation, verifying both the efficacy and efficiency of our method in video Transformers.

Acknowledgments

This work was partly supported by ICT Creative Consilience Program through the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT)(IITP-2024-2020-0-01819), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(NRF-2023R1A2C2005373), and Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government (24ZB1200, Fundamental Technology Research for Human-Centric Autonomous Intelligent Systems).

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers. *arXiv:2005.00928*, 2020. 3

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, 2017. 2, 5, 6

[3] Roman Bachmann, David Mizrahi, Andrei Atanov, and Amir Zamir. Multimae: Multi-modal multi-task masked autoencoders. In *ECCV*, 2022. 8

[4] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014. 8

[5] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *ICCV*, 2021. 6

[6] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020. 2

[7] Daniel Bolya and Judy Hoffman. Token merging for fast stable diffusion. *CVPR Workshop on Efficient Deep Learning for Computer Vision*, 2023. 6, 8

[8] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *ICLR*, 2022. 2, 4, 6, 7, 8

[9] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015. 5, 6

[10] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 8

[11] David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011. 2, 5, 6

[12] Feng Cheng, Xizi Wang, Jie Lei, David Crandall, Mohit Bansal, and Gedas Bertasius. Vindlu: A recipe for effective video-and-language pretraining. In *CVPR*, 2023. 6, 8

[13] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *NeurIPS*, 2021. 2

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020. 1, 8

[15] Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *ECCV*, 2022. 2

[16] Tsu-Jui Fu, Linjie Li, Zhe Gan, Kevin Lin, William Yang Wang, Lijuan Wang, and Zicheng Liu. VIOLET: End-to-End Video-Language Transformers with Masked Visual-token Modeling. In *arXiv:2111.1268*, 2021. 1, 6, 8

[17] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 2

[18] Difei Gao, Luowei Zhou, Lei Ji, Linchao Zhu, Yi Yang, and Mike Zheng Shou. Mist: Multi-modal iterative spatial-temporal transformer for long-form video question answering. In *CVPR*, 2023. 2

[19] 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 *ICCV*, 2017. 7

[20] Xin Gu, Guang Chen, Yufei Wang, Libo Zhang, Tiejian Luo, and Longyin Wen. Text with knowledge graph augmented transformer for video captioning. In *CVPR*, 2023. 8

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 8

[22] Bingkun Huang, Zhiyu Zhao, Guozhen Zhang, Yu Qiao, and Limin Wang. Mgmae: Motion guided masking for video masked autoencoding. In *ICCV*, 2023. 8

[23] A. Katharopoulos, A. Vyas, N. Pappas, and F. Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 8

[24] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *KDD*, 2022. 8

[25] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *ICLR*, 2020. 2, 8

[26] Zhenglun Kong, Peiyan Dong, Xiaolong Ma, Xin Meng, Wei Niu, Mengshu Sun, Xuan Shen, Geng Yuan, Bin Ren, Hao Tang, et al. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *ECCV*, 2022. 2

[27] Heejun Lee, Minki Kang, Youngwan Lee, and Sung Ju Hwang. Sparse token transformer with attention back tracking. In *ICLR*, 2022. 8

[28] Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv:2107.04589*, 2021. 8

[29] Jie Lei, Linjie Li, Luowei Zhou, Zhe Gan, Tamara L. Berg, Mohit Bansal, and Jingjing Liu. Less is more: Clipbert for video-and-language learningvia sparse sampling. In *CVPR*, 2021. 6

[30] Jie Lei, Tamara L. Berg, and Mohit Bansal. Revealing single frame bias for video-and-language learning. In *ACL*, 2023. 5, 6, 8

[31] Dongxu Li, Junnan Li, Hongdong Li, Juan Carlos Niebles, and Steven CH Hoi. Align and prompt: Video-and-language pre-training with entity prompts. In *CVPR*, 2022. 6, 8

[32] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *CVPR*, 2022. 1, 8

[33] Kunchang Li, Yali Wang, Yizhuo Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. *ICCV*, 2023. 1, 2, 4, 5, 6, 7, 8

[34] Linjie Li, Zhe Gan, Kevin Lin, Chung-Ching Lin, Ce Liu, Zicheng Liu, and Lijuan Wang. Lavender: Unifying video-language understanding as masked language modeling. In *CVPR*, 2023. 6

[35] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *ICLR*, 2022. 2, 3

[36] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. *ICLR*, 2022. 2

[37] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *CVPR*, 2022. 8

[38] Song Liu, Haoqi Fan, Shengsheng Qian, Yiru Chen, Wenkui Ding, and Zhongyuan Wang. Hit: Hierarchical transformer with momentum contrast for video-text retrieval. In *ICCV*, 2021. 2

[39] 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 *ICCV*, 2021. 2

[40] Huaishao Luo, Lei Ji, Ming Zhong, Yang Chen, Wen Lei, Nan Duan, and Tianrui Li. Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. *Neurocomputing*, 2022. 6, 8

[41] Yiwei Ma, Guohai Xu, Xiaoshuai Sun, Ming Yan, Ji Zhang, and Rongrong Ji. X-clip: End-to-end multi-grained contrastive learning for video-text retrieval. In *ACM Multimedia*, 2022. 8

[42] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. Token pooling in vision transformers for image classification. In *WACV*, 2023. 8

[43] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, 2021. 1, 8

[44] 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 *CVPR*, 2022. 8

[45] Zizheng Pan, Bohan Zhuang, Haoyu He, Jing Liu, and Jianfei Cai. Less is more: Pay less attention in vision transformers. In *AAAI*, 2022. 8

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 8

[47] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, 2021. 2, 3

[48] Hanoona Rasheed, Muhammad Uzair Khattak, Muhammad Maaz, Salman Khan, and Fahad Shahbaz Khan. Fine-tuned clip models are efficient video learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6545–6554, 2023. 7, 8

[49] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description. *IJCV*, 2017. 5, 6

[50] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv:1212.0402*, 2012. 7

[51] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *NeurIPS*, 2022. 1, 7, 8

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*. PMLR, 2021. 1, 8

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017. 2

[54] A. Vyas, A. Katharopoulos, and F. Fleuret. Fast transformers with clustered attention. *NeurIPS*, 2020. 8

[55] Junke Wang, Dongdong Chen, Zuxuan Wu, Chong Luo, Luowei Zhou, Yucheng Zhao, Yujia Xie, Ce Liu, Yu-Gang Jiang, and Lu Yuan. Omnivl: One foundation model for image-language and video-language tasks. *NeurIPS*, 2022. 6, 8

[56] Jinpeng Wang, Yixiao Ge, Rui Yan, Yuying Ge, Kevin Qinghong Lin, Satoshi Tsutsui, Xudong Lin, Guanyu Cai, Jianping Wu, Ying Shan, et al. All in one: Exploring unified video-language pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6598–6608, 2023. 6, 8

[57] Limin Wang, Bingkun Huang, Zhiyu Zhao, Zhan Tong, Yinan He, Yi Wang, Yali Wang, and Yu Qiao. Videomae v2: Scaling video masked autoencoders with dual masking. In *CVPR*, 2023. 1, 8

[58] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv:2006.04768*, 2020. 2, 8

[59] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, and Huaxia Xia. End-to-end video instance segmentation with transformers. In *CVPR*, 2021. 1, 8

[60] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv:2212.03191*, 2022. 1, 6, 8

[61] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022. 1, 8

[62] QuanLin Wu, Hang Ye, Yuntian Gu, Huishuai Zhang, Liwei Wang, and Di He. Denoising masked autoencoders help robust classification. In *ICLR*, 2023. 8

[63] Junbin Xiao, Pan Zhou, Tat-Seng Chua, and Shuicheng Yan. Video graph transformer for video question answering. In *ECCV*, 2022. 2

[64] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and

efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 1, 8

[65] Zhen Xing, Qi Dai, Han Hu, Jingjing Chen, Zuxuan Wu, and Yu-Gang Jiang. Svformer: Semi-supervised video transformer for action recognition. In *CVPR*, 2023. 2

[66] Yunyang Xiong, Zhanpeng Zeng, Rudrasis Chakraborty, Mingxing Tan, Glenn Fung, Yin Li, and Vikas Singh. Nyströmformer: A nyström-based algorithm for approximating self-attention. In *AAAI*, 2021. 2

[67] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *MM*, 2017. 6

[68] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*, 2016. 2, 4, 5, 6, 7

[69] Hongwei Xue, Yuchong Sun, Bei Liu, Jianlong Fu, Ruihua Song, Houqiang Li, and Jiebo Luo. Clip-vip: Adapting pre-trained image-text model to video-language representation alignment. *arXiv:2209.06430*, 2022. 6

[70] Shen Yan, Tao Zhu, Zirui Wang, Yuan Cao, Mi Zhang, Soham Ghosh, Yonghui Wu, and Jiahui Yu. Video-text modeling with zero-shot transfer from contrastive captioners. *arXiv:2212.04979*, 2022. 6, 8

[71] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Just ask: Learning to answer questions from millions of narrated videos. In *ICCV*, 2021. 6, 8

[72] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. *NeurIPS*, 35, 2022. 6

[73] Jiewen Yang, Xingbo Dong, Liujun Liu, Chao Zhang, Jiajun Shen, and Dahai Yu. Recurring the transformer for video action recognition. In *CVPR*, 2022. 2

[74] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *NeurIPS*, 33, 2020. 8

[75] Yuhang Zang, Wei Li, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Open-vocabulary detr with conditional matching. In *ECCV*, 2022. 1, 8

[76] Rowan Zellers, Ximing Lu, Jack Hessel, Youngjae Yu, Jae Sung Park, Jize Cao, Ali Farhadi, and Yejin Choi. Merlot: Multimodal neural script knowledge models. *NeurIPS*, 2021. 1, 6, 8

[77] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2021. 1, 8