Representation Shift: Unifying Token Compression with FlashAttention

Joonmyung Choi[1*] Sanghyeok Lee[1*] Byungoh Ko[1]
Eunseo Kim[1] Jihyung Kil[2] Hyunwoo J. Kim[3†]
[1]Korea University [2]Adobe Research [3]KAIST
{pizard, cat0626, ko990128, pingdoll3110}@korea.ac.kr
jkil@adobe.com hyunwoojkim@kaist.ac.kr

Abstract

Transformers have demonstrated remarkable success across vision, language, and video. Yet, increasing task complexity has led to larger models and more tokens, raising the quadratic cost of self-attention and the overhead of GPU memory access. To reduce the computation cost of self-attention, prior work has proposed token compression techniques that drop redundant or less informative tokens. Meanwhile, fused attention kernels such as FlashAttention have been developed to alleviate memory overhead by avoiding attention map construction and its associated I/O to HBM. This, however, makes it incompatible with most training-free token compression methods, which rely on attention maps to determine token importance. Here, we propose Representation Shift, a training-free, model-agnostic metric that measures the degree of change in each token's representation. This seamlessly integrates token compression with FlashAttention, without attention maps or retraining. Our method further generalizes beyond Transformers to CNNs and state space models. Extensive experiments show that Representation Shift enables effective token compression compatible with FlashAttention, yielding significant speedups of up to 5.5× and 4.4× in video-text retrieval and video QA, respectively. Code is available at https://github.com/mlvlab/Representation-Shift.

1. Introduction

Transformers, initially proposed for natural language processing (NLP) [55], have become a prominent architecture in the vision domain. Following the pioneering work ViTs [19], numerous subsequent studies have extended Transformers to various vision tasks, *e.g.*, image classification [15, 19, 37, 52, 53, 59], object detection [8, 23, 58, 63, 75, 78], segmentation [12, 49, 76], and video understanding [26–28, 32, 44, 51, 61, 62, 64]. While these works have proven to be effective, the quadratic complexity of the self-attention mechanism remains a critical bottleneck, limiting the scalability of Transformer based architectures.

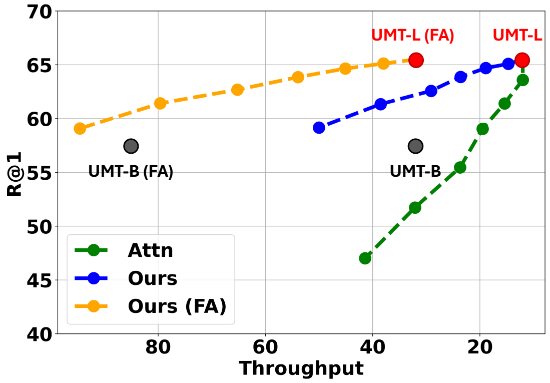


Figure 1. Comparison of importance metrics for token pruning (average over 7 video-text retrieval benchmarks in Table 2). Pruning with a conventional attention-based score (Attn) yields poor speed-accuracy trade-offs on UMT-L and is incompatible with FlashAttention (FA). In contrast, our proposed representation shift accelerates both vanilla UMT-L and UMT-L with FlashAttention, achieving superior trade-offs compared to downscaling to UMT-B and attention-based scores.

To address this problem, a wide range of approaches have been proposed to accelerate Transformers across various domains, such as vision and natural language processing (NLP). Early works tackled the computational burden by proposing sparse attention mechanisms [3, 25, 57, 66] and architectural modifications [15, 37, 48, 54, 59, 73] to approximate self-attention, such as low-rank approximations and sparse attention patterns. However, these methods often introduce structural deviations from the original Transformer architecture, making them incompatible with widely adopted pretrained models. As a result, vanilla Transformers [19] remain the dominant choice in practice, supported by the widespread availability of pre-trained models across a variety of domains. Here, one promising approach to accelerate pre-trained vanilla Transformers is FlashAtten-

[*] Equal contribution.
[†] Corresponding author.

tion [16], which optimizes GPU memory access of self-attention while maintaining the original formulation. While FlashAttention preliminarily focused on the long sequences of LLM, it also demonstrates substantial acceleration with Vision Transformers as in recent works [1, 11, 42, 60, 64]. Another line of work in accelerating Vision Transformers is token compression [4, 13, 24, 29, 33, 39, 41, 43, 46, 56, 69, 71], which reduces computational cost by pruning or merging tokens. Since determining which tokens to retain is crucial, previous works incorporate token importance measurement as a fundamental step. Some approaches [41, 46, 69] introduce additional learnable networks to predict token importance, and other works [13, 20, 29, 39, 56] employ attention-based heuristics as a surrogate for token importance. Although these works have shown promising acceleration on Vision Transformers, methods that employ learnable networks necessitate extra training, making them infeasible in a training-free manner. Also, attention-based scoring methods limit their use when the attention map is unavailable (e.g., FlashAttention, CNN). While FlashAttention alone provides substantial acceleration, achieving a 1.5× speedup on DeiT-S and 2.7× speedup on UMT-B, existing token pruning methods fail to further improve efficiency in a training-free setting due to their reliance on learnable modules or attention maps.

To address this, we propose a token importance criterion that is training-free and model-agnostic, based on representation shift, which quantifies the change in token embeddings before and after the layer(Figure 2). This simple but effective approach successfully captures the amount of information amplified by any operation, *e.g.*, FFN, Attention, and Convolutions. By leveraging representation shift as an importance metric, our method effectively identifies and removes redundant tokens. Since our method is not dependent on attention mechanisms, it generalizes beyond Transformers to architectures like CNNs [21, 38, 65] and SSMs [30, 36, 77], while seamlessly integrating with fused kernel operations such as FlashAttention for efficient inference. Experimental results show that our method outperforms existing attention-based token importance methods in both accuracy and efficiency on vanilla Transformers. Specifically, we achieve impressive throughput improvements of about 5.5× speedup with UMT [32] on multiple video-text retrieval benchmarks. Moreover, unlike prior attention-dependent methods, our approach additionally generalizes to previously unsupported architectures such as CNNs and state space models. In sum, our key contributions are as follows:

- We propose a novel approach for estimating token importance, called **representation shift**, which directly captures the amount of information amplified by each operation. This model-agnostic importance score can be computed in a training-free manner with negligible overhead.

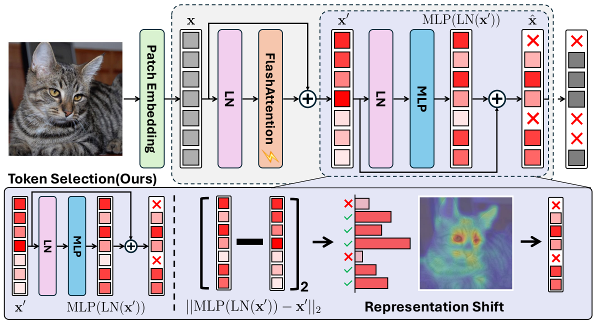


Figure 2. Illustration of representation shift for token importance. We compute the L2 distance between token representations before and after the MLP layer to quantify how much each token is emphasized by the transformation.

- To the best of our knowledge, this is the first token reduction method applicable to both FlashAttention and CNNs.
- Through extensive experiments on video and image understanding tasks, we demonstrate that combining FlashAttention with our representation shift–based token pruning yields notable inference speedups.

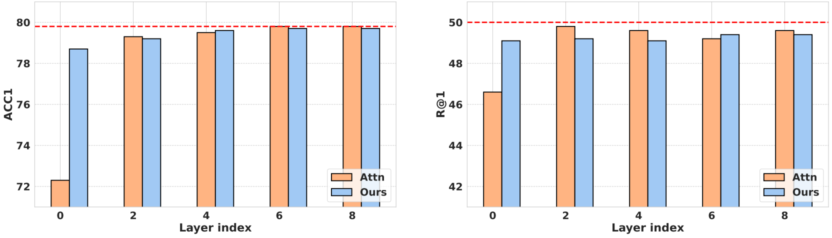
2. Related works

Efficient Vision Transformers. Built with ViTs [19], self-attention [55] are introduced to handle various vision tasks. Following works [35, 70], such as DeiT [52], further improve data efficiency of Vision Transformers. However, despite the competitive performance, the quadratic cost of self-attention with respect to the number of tokens remains the major bottleneck. To address this issue, earlier works [14, 22, 25, 45, 57, 66] have tried to find an efficient approximation of self-attention. For instance, Reformer [25] achieves the $O(N \log N)$ complexity with a hashing function, and Linformer [57] approximates the self-attention via a low-rank matrix, resulting in the linear cost of $O(N)$. Nystromformer [66] and performer [14] also present the linear approximation of the self-attention. In parallel, several works [3, 9, 48, 73] have focused on sparsifying the attention map to lessen complexity. Similarly, recent vision transformers [15, 37, 54, 58, 59] reduce the number of key and value tokens. PVT [58, 59] introduce spatial-reduction attention that downsamples the key and value tokens before attention, and Swin [37], Twins [15], and MaxViT [54] also apply local attention to reduce the reference tokens. Also, for the deployment in edge-devices, a line of work [7, 20, 34, 40, 72] has been proposed. More recently, with the aim to reduce the latency by memory-bound operation, FlashAttention [16] conducts attention calculation within fast SRAM minimizing the memory access to slow HBM. In this work, we aim to further boost the FlashAttention with token compression.

Token Compression. Since the cost heavily relies on the

Method	DeiT-S [52]		UMT-B [32]	
	Acc-1	Thr	R@1	Thr
Self-Attention	79.8	2308	50.0	32
FlashAttention	79.8	**3552**	50.0	**85**

Table 1. Comparison of FlashAttention [16] with standard self-attention. Throughputs are measured with NVIDIA RTX A6000. ImageNet [18] and MSRVTT [68] are used for image and video understanding, respectively.



(a) DeiT-S

(b) UMT-B

Figure 3. Comparison of representation shift and attention-based scores as importance for token pruning. For DeiT [52] and UMT [32], 40 and 1100 tokens are pruned at each layer. The red line indicates baseline performance without token compression.

number of tokens, recent works [4, 13, 24, 33, 39, 41, 43, 46, 56, 69, 71] explicitly focus on compressing the token. To preserve the core information of an image after compressing tokens, they generally prune or merge unimportant tokens. Importance estimation typically follows two major approaches. First is the additional learnable network to predict the importance. For instance, AdaViT [41] and DynamicViT [46] introduce additional learnable decision networks to select the tokens to be compressed, and A-ViT [69] also needs to train additional parameters for calculating the importance of the tokens. Second one is to utilize intermediate attention scores as a surrogate function for measuring the importance. Specifically, EViT [33] and BAT [39] approximate the importance of the tokens using the attention score for the class tokens, which indicate the influence of each token on the final prediction. Zero-TPrune [56] measures the informativeness of tokens via a ranking method with attention maps inspired by Page Rank [5]. In the video domain, vid-TLDR [13] captures the salient regions based on the entropy of the attention scores. While the aforementioned works have proven to be effective in compressing tokens with the affordable speed-accuracy trade-offs, they require either additional training or attention maps. Note that FlashAttention does not provide intermediate attention scores to minimize memory access on HBM. As a result, despite the much faster speed of FlashAttention over standard self-attention, it is not straightforward to apply previous token compression methods in a training-free manner.

3. Method

3.1. Preliminaries

In Vision Transformers [19, 52, 53], the input image is first partitioned into a set of image patches $\mathbf{x} \in \mathbb{R}^{N \times C}$, called tokens, where $N = \frac{H}{P} \times \frac{W}{P}$ is the number of tokens, $H \times W$ is the resolution of the image, and P is the patch size. This set of tokens is then processed via self-attention defined as:

$$\mathrm{SA}(\mathbf{x}) = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{C}}\right)V, \tag{1}$$

where $[Q, K, V] = \mathbf{x}W$, $W \in \mathbb{R}^{C \times 3C}$ is a learnable projection matrix. This process incurs the quadratic cost of $O(N^2C + NC^2)$. To mitigate this cost, recent works [13, 33, 41, 46, 69] explicitly prune less informative tokens, resulting in a reduced token set $\tilde{\mathbf{x}} \in \mathbb{R}^{(N-r) \times C}$, where r is the number of pruned tokens.

The importance of tokens, $s \in \mathbb{R}^L$, is typically estimated using the attention map, $A = \mathrm{Softmax}\left(\frac{QK^\top}{\sqrt{C}}\right)$, which is the byproduct of self-attention. For example, the importance of the tokens can be defined as the attention scores relative to the class token $q_{\text{cls}} \in \mathbb{R}^{1 \times C}$:

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

or as a summarized attentiveness across all query vectors:

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

where $A_i = \mathrm{Softmax}\left(\frac{q_i K^\top}{\sqrt{C}}\right)$. While these attention-based scores have proven effective as surrogate measures for the informativeness of the tokens, they are not applicable when the attention map is unavailable, as in the case of FlashAttention [16]. In our preliminary experiments (Table 1), FlashAttention also brings substantial speedup over standard attention in Vision and Video Transformers, *e.g.*, DeiT [52] and UMT [32]. Despite the promising results, we cannot further boost it with previous attention-based token compressions. Here, we aim to develop a simple yet effective model-agnostic method for quantifying token importance in a training-free manner.

3.2. Representation shift for token importance

In our preliminary experiments, we observed that the representation shifts of the tokens through a network layer reflect their contribution to the prediction of the model. Here, we first define the representation shift, and then provide the quantitative and qualitative results to validate it. Given in-

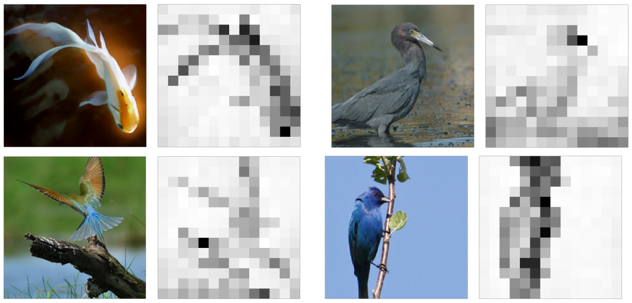


Figure 4. Visualization of representation shift. Given the image (left), we visualize (right) the representation shift ($\Delta\mathbf{x}$) of each token before and after the attention layer.

put tokens $\mathbf{x} \in \mathbb{R}^{L\times C}$, the representation shift for importance score s is defined as

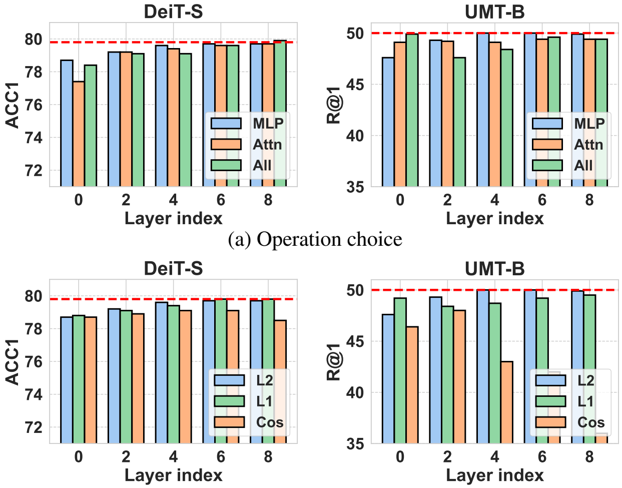
$$s = \Delta\mathbf{x} = \mathcal{D}(F(\mathbf{x}), \mathbf{x}), \tag{4}$$

where $F(\cdot)$ indicates the transformation of the layer (*e.g.*, Attention and MLP) and $\mathcal{D}$ is the distance metric like L2 distance, *i.e.*, $\mathcal{D}(F(\mathbf{x}), \mathbf{x}) = \|F(\mathbf{x}) - \mathbf{x}\|_2$. In other words, the representation shift reflects the extent to which each token is emphasized by the function. F.

Our central hypothesis is that critical tokens tend to have a higher representation shift, as the network encourages them to emphasize the core information or suppresses redundant signals. Conversely, the tokens with minimal representation shift are likely to be irrelevant to target tasks.

To validate this hypothesis, we conduct toy experiments with DeiT-S [52] on image classification (ImageNet1K [18]) and UMT-B [32] on video-text retrieval (MSRVTT [68]), and summarize the results in Figure 3. For comparison, we first evaluated token importance with attention-based metrics and our representation shift, respectively, and then dropped the tokens having the lowest k scores at each layer ([0,2,4,6,8]). We use $k = 40$ for DeiT and $k = 1100$ for UMT. For attention-based scoring, we opt Equation (2) for DeiT used in [33, 39], and Equation (3) for UMT since the class token is generally absent in video transformers. Also, for representation shift, we compute the L2 distance between the representation of the tokens before and after the attention layer as $\Delta\mathbf{x} = \|\text{SA}(\text{LN}(\mathbf{x})) - \mathbf{x}\|_2 \in \mathbb{R}^L$. As summarized in Figure 3, pruning based on representation shift achieves competitive or better performance compared to pruning with prevalent attention-based scores. We demonstrate that the representation shift is a sufficient approximation of the token importance as well as conventional attention-based scores. Notably, our method introduces no additional learnable parameters and remains applicable even when intermediate attention maps are inaccessible, as in the case of FlashAttention.

We also conduct qualitative analysis of the representation shift (Figure 4) in the middle of DeiT. Interestingly, it captures the foreground object, which aligns with the concept of saliency detection. In other words, we can suppress the noise from the tokens irrelevant to the main content by compressing them based on the proposed scores. Based on quantitative and qualitative analysis, we underscore the effectiveness of the representation shift for token importance. In the following subsection, we will provide a thorough investigation of the representation shift.



(a) Operation choice

(b) Distance metric

Figure 5. Analysis on (a) operation choice and (b) distance metric for representation shift. In our experiments, we evaluate the impact of operation choice by pruning tokens based on the representation shift computed using the L2 norm for each candidate operation. Similarly, for the analysis of distance metric selection, we prune tokens using each distance metric with the MLP layer.

3.3. Exploration on representation shift

Operation choice. Given $\mathbf{x} \in \mathbb{R}^{L\times C}$, the attention blocks of Vision Transformers are typically computed as

$$\mathbf{x}' = \text{SA}(\text{LN}(\mathbf{x})) + \mathbf{x}, \tag{5}$$

$$\hat{\mathbf{x}} = \text{MLP}(\text{LN}(\mathbf{x}')) + \mathbf{x}', \tag{6}$$

where LN is Layer Normalization. We investigate the impact of the operation choice for representation shift, especially for three cases: representation shift through (i) attention as $\Delta\mathbf{x} = \mathcal{D}(\text{SA}(\text{LN}(\mathbf{x})), \mathbf{x})$, (ii) MLP as $\Delta\mathbf{x} = \mathcal{D}(\text{MLP}(\text{LN}(\mathbf{x}')), \mathbf{x}')$, and (iii) entire attention block including Equations (5) and (6) as $\Delta\mathbf{x} = \mathcal{D}(\hat{\mathbf{x}}, \mathbf{x})$. We conducted ablation experiments to evaluate the efficacy of each metric as alternatives for token importance. Under the same settings of the previous section, we prune a fixed number of tokens per layer based on the computed L2 distance scores and evaluate the impact on overall model performance. Figure 5a reveals that token pruning guided by the representation shift through sole MLP generally outperforms other metrics across the layer and models. Since the attention layer inherently facilitates information exchange across tokens, its transformation may be more diffuse. In contrast,

Dataset	Metric	UMT-B [32]			UMT-L [32]		
		Base	Attn	Ours	Base	Attn	Ours
	Throughput ↑	32	57 (×1.78)	**175** (×5.47)	12	23 (×1.91)	**66** (×5.50)
	GFlops ↓	303.3	156.4	156.4	984.6	478.5	478.5
MSRVTT [68]	R@1 ↑	50.0	47.6	**48.0**	58.7	50.2	**56.5**
	R@5 ↑	76.8	74.1	**74.4**	81.3	72.7	**79.6**
	R@10 ↑	83.9	81.7	**82.0**	86.8	80.3	**86.0**
MSVD [10]	R@1 ↑	62.1	**60.3**	57.7	70.3	64.0	**67.9**
	R@5 ↑	89.3	**83.7**	80.5	89.3	84.4	**87.5**
	R@10 ↑	93.2	**89.0**	86.4	93.2	89.7	**92.2**
ActivityNet [6]	R@1 ↑	57.2	**54.2**	50.3	65.6	53.2	**62.9**
	R@5 ↑	83.7	**81.1**	78.5	89.1	80.3	**87.3**
	R@10 ↑	91.6	**89.6**	88.1	94.9	88.8	**93.8**
DiDeMo [2]	R@1 ↑	62.1	**57.7**	56.9	70.8	58.2	**67.3**
	R@5 ↑	86.8	82.7	**83.3**	90.6	83.8	**89.1**
	R@10 ↑	92.1	88.6	**89.2**	94.5	89.9	**93.1**
LSMDC [47]	R@1 ↑	32.7	29.0	**30.0**	42.2	34.4	**39.8**
	R@5 ↑	54.1	50.1	**51.1**	64.9	56.6	**62.9**
	R@10 ↑	63.3	59.2	**59.7**	72.3	64.1	**70.0**
SSV2-label [31]	R@1 ↑	64.0	58.0	**59.1**	72.4	60.6	**69.3**
	R@5 ↑	88.3	83.9	**84.4**	93.4	85.7	**91.0**
	R@10 ↑	92.9	**90.8**	90.7	96.7	91.1	**94.9**
SSV2-Template [31]	R@1 ↑	74.6	65.4	**69.0**	78.4	67.5	**74.8**
	R@5 ↑	93.9	91.3	**92.4**	95.9	91.9	**95.0**
	R@10 ↑	96.8	95.0	**95.3**	97.8	94.9	**97.5**

Table 2. Video-text retrieval on MSRVTT [68], MSVD [10], ActivityNet [6], DiDeMo [2], LSMDC [47], SSV2-Label/Template [31].

the MLP operates on each token independently, leading to a more discriminative representation shift that captures token-specific contributions. Based on these findings, we adopt the representation shift at MLP as our primary measure for token importance.

Distance metrics. We further explore which distance metric $\mathcal{D}$ is most appropriate for estimating the representation shift. A straightforward approach is the (i) **L2** norm as $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_2$, which computes the Euclidean distance between input and output representations, capturing the absolute magnitude of the transformation. We also study the efficacy of (ii) **L1** Norm as $\mathcal{D}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|_1$, which is more robust to the outliers. Additionally, (iii) cosine distance (**Cos**), *i.e.*, $(\mathcal{D}(\mathbf{x}, \mathbf{y}))_i = 1 - \frac{\mathbf{x}_i \cdot \mathbf{y}_i}{\|\mathbf{x}_i\| \|\mathbf{y}_i\|}$, computes angular difference between vectors, emphasizing directional change rather than magnitude. For comparison of distance metrics, we compute the representation shift before and after the MLP layer as $\mathcal{D}(\mathrm{MLP}(\mathrm{LN}(\mathbf{x}')), \mathbf{x}')$, and drop the tokens. As shown in Figure 5b, the L2 distance consistently produces more robust results as a token importance compared to other distance metrics. Our analysis indicates that cosine similarity is suboptimal for assessing token importance in the deeper layers of Transformers. Also, although the L1 distance performs favorably at the first layer, it consistently underperforms relative to the L2 distance in subsequent layers. Therefore, we will use L2 distance for representation shift as the default distance metric.

4. Experiments

In this section, we will present the results of video understanding tasks in Section 4.1, image classification in Section 4.2, and analysis of the proposed method in Section 4.3.

4.1. Video Understandings

Settings. To validate the efficacy of the representation shift, we first conducted token pruning based on representation shift with several video tasks, where the large number of tokens across frames imposes significant computational costs. We use the UMT [32], a Video Transformer built with vanilla attention, as our baseline for video-text retrieval [2, 6, 10, 31, 47, 68], and video question-answering [67]. For comparison with attention-based scores, we also use the averaged attention scores as in Equation (3), since the class token is not available at Video Transformer. We progressively reduce the number of tokens by 20% and 10% in each of the first three layers of UMT for video-text retrieval and video question-answering, respectively, by applying pruning based on both metrics. FlashAttention is used in the case of representation shift, as the attention-based score is not compatible with it. All experiments are conducted in a training-free manner.

Video-text retrieval. In video-text retrieval, the model retrieves the most related text given a video (video-to-text retrieval, V2T) or finds the most relevant video for a text query (text-to-video retrieval, T2V). We report the

Dataset	Metric	UMT-B [32]			UMT-L [32]		
		Base	vid-TLDR	+Ours	Base	vid-TLDR	+Ours
MSRVTT [68]	Throughput ↑	32.0	43.6 (×1.36)	**131.4** (×4.10)	12.0	19.1 (×1.59)	**41.0** (×3.42)
	R@1 ↑	50.0	**50.8**	**50.8**	**58.7**	58.5	58.6
MSVD [10]	Throughput ↑	32.0	42.2 (×1.32)	**131.1** (×4.10)	12.0	19.9 (×1.66)	**40.0** (×3.33)
	R@1 ↑	62.1	62.7	**62.8**	70.3	70.4	**70.6**
ActivityNet [6]	Throughput ↑	32.0	34.2 (×1.07)	**114.7** (×3.58)	12.0	18.1 (×1.51)	**40.4** (×3.34)
	R@1 ↑	**57.2**	56.6	56.8	65.6	65.2	**66.0**
DiDeMo [2]	Throughput ↑	32.0	33.9 (×1.06)	**129.3** (×4.05)	12.0	18.3 (×1.53)	**51.5** (×4.29)
	R@1 ↑	62.1	**62.4**	62.0	70.8	70.4	**70.9**
LSMDC [47]	Throughput ↑	32.0	36.5 (×1.14)	**110.7** (×3.46)	12.0	16.6 (×1.38)	**50.8** (×4.23)
	R@1 ↑	**32.7**	32.4	32.4	**42.2**	41.9	41.9
SSV2-label [31]	Throughput ↑	32.0	34.1 (×1.07)	**114.4** (×3.58)	12.0	16.4 (×1.37)	**39.9** (×3.33)
	R@1 ↑	**64.0**	63.8	63.5	**72.4**	72.1	71.8
SSV2-Template [31]	Throughput ↑	32.0	38.1 (×1.19)	**106.6** (×3.33)	12.0	16.0 (×1.33)	**45.2** (×3.77)
	R@1 ↑	**74.6**	74.0	73.9	78.4	78.1	**78.5**

Table 3. Extensibility of representation shift with other token compression, vid-TLDR [13]. +Ours indicates the results of vid-TLDR [13] after replacing the importance metric with representation shift and adopting FlashAttention.

Method	**GFlops**	**Throughput**	**MSR-QA**	**MSVD-QA**
UMT-B [32]	303.3	32	44.9	48.1
UMT-B-Attn	217.7	39(x1.22)	44.8	46.5
UMT-B-Ours	217.7	128(x4.00)	44.6	47.0
UMT-L [32]	984.6	12	49.5	55.2
UMT-L-Attn	690.5	15(x1.25)	49.5	54.2
UMT-L-Ours	690.5	46(x3.83)	49.0	54.9

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

harmonic mean of results of V2T and T2V on seven benchmarks: MSRVTT [68], MSVD [10], ActivityNet [6], DiDeMo [2], LSMDC [47], SSV2-Label/Template [31]. For comparing the efficiency, we also measure and provide both FLOPs (G) and throughput (vid/s) using a single NVIDIA RTX A6000 with a batch size of 20, given the video consisting of 12 frames with 224^2 resolutions. Given the baseline model without token pruning (**Base**), we apply token pruning with attention-based scores (**Att**) and representation shift (**Ours**), respectively. The results are presented in Table 2. Since our representation shift enables the token pruning to work with FlashAttention, it brings a promising $5.47\times$ and $5.5\times$ speed-up in UMT-B and UMT-L, respectively. Our approach nearly doubles the throughput compared to token pruning methods based on traditional attention scores with standard attention. Further, despite the faster inference, our approach has shown competitive or even better performance, achieving up to 9.7% R@1 gain, especially with UMT-L on ActivityNet compared to attention-based pruning. On average, we observe a $+7.2\%$ improvement in R@1 with UMT-L. It is worth noting that applying token pruning with representation shift offers a more favorable speed-accuracy trade-off than simply downscaling the model, as UMT-L with representation shift (66 vid/s) achieves approximately $2\times$ higher throughput than base UMT-B (32 vid/s), while consistently surpassing it.

We further explore the applicability of representation shift with other token compression work by replacing the importance metric of vid-TLDR [13], a token merging method for efficient video transformer. Following the original configuration of vid-TLDR, including the reduction ratio and layer choice, we report the results on video-text retrieval. In Table 3, we demonstrate the solid advantage of representation shift with other token compression. Originally, vid-TLDR employed an attention-based metric to detect salient regions of the image, which was thus incompatible with FlashAttention. However, by substituting the importance metric with our representation shift, we can harness the efficiency of FlashAttention along with vid-TLDR. Specifically, under the same reduction ratio, our representation shift achieves an average speed-up of $3.74\times$ and $3.67\times$ in UMT-B and UMT-L with the minimal performance drop.

Video question-answering. We also demonstrate the efficiency of the proposed approach in video question-answering (video QA) tasks. In video QA, the model generates responses to questions related to a given video. To evaluate this, we assess each method on MSRVTT-QA, MSVD-QA benchmarks [67], summarizing the results in Table 4. Similar to video-text retrieval, we compare the three cases: the baseline model without pruning (**Base**), the model with attention-based token pruning (**Att**), and (**Ours**). Compared to the Base model, we demonstrate a promising improvement, achieving approximately $4\times/3.83\times$ higher throughput in UMT-B/L. Further, despite being faster than conventional attention-based pruning, our approach achieves com-

Method	Metric	Base	Attn	Ours
Deit-T	Acc	72.1	65.5	68.3
	Throughput	6725	10949	13296
	GFLOPs	1.3	0.8	0.8
Deit-S	Acc	79.8	72.1	77.8
	Throughput	3002	4844	5948
	GFLOPs	4.6	3.0	3.0
Deit-B	Acc	81.8	76.9	79.6
	Throughput	1037	2065	2428
	GFLOPs	17.6	11.5	11.5

Table 5. ImageNet1K [18] classification results with DeiT [52].

parable or even better performance. Notably, in the UMT-L, we observe significant improvements of 0.5% and 0.7% on MSRVTT and MSVD, respectively.

4.2. Image Classification

Vision Transformers. We experiment on image classification with ImageNet1K [18]. For vision transformers, we use DeiT [52] without additional training, and report the top-1 accuracy and throughput with a batch size of 512. For comparison, we use attention scores for class token (Equation (2)) used in EViT [33], and BAT [39]. For the representation shift, we use the same settings (L2, MLP) as video understandings, along with FlashAttention. After quantifying the importance of the tokens in the [1,4,7] layers of DeiT, we pruned the 20% tokens at each layer. As shown in Table 5, although the same proportion of tokens is pruned, our method consistently outperforms the attention-based scores. Specifically, combined with FlashAttention, the representation shift achieves 1.2× higher throughput with the gain of +2.8%, +5.7%, and +2.7% accuracy gain in DeiT-T/S/B compared to attention-based scoring. We believe that representation shift provides more robust importance scores than traditional attention scores, resulting in a significant performance gap.

CNN and SSM. Since representation shift is a model-agnostic approach to estimate the token importance, it naturally extends to other architectures, which have been underexplored in previous token compressions. For this, we first conduct experiments with ResNet [21] on ImageNet1K. In CNNs, we measure the representation shift before and after each stage, as ResNet does not contain MLPs. Since the convolutional operation in ResNet only works with a 2D grid structure, token pruning in CNNs cannot be performed in a straightforward manner. So, we consider two variants of token pruning: i) removing the least important tokens from each row and column (Token-wise, **T-W**), and ii) averaging the representation shift across each row and column and then pruning tokens line by line from those rows and columns with the lowest average values (Line-wise, **L-W**), akin to [50]. Specifically, by each approach, we remove 8 columns and 8 rows after the first stage, and 4 columns and 4 rows after the second stage. As the resolutions are changed after token compression in CNNs, we finetune the model for 100 epochs, including 10 cooldown epochs to refine this change. Table 6 reveals that both pruning approaches with representation shift bring substantial throughput improvements in ResNet. We observe at least 18% speed up in both pruning approaches. Especially, line-wise pruning shows very competitive performance with the base ResNet without pruning, achieving the higher throughput of 7112/3553 (img/s) compared to the original throughput of 5811/2927 (img/s) in ResNet-34/50.

Method	Metric	Base	L-W	T-W
ResNet-34	Acc	73.2	72.8	72.2
	Throughput	5811	7112	6867
	GFLOPs	3.7	2.5	2.5
ResNet-50	Acc	76.1	76.4	75.9
	Throughput	2927	3553	3489
	GFLOPs	4.1	2.7	2.7

Table 6. ImageNet1K [18] classification results with ResNet [21]. L-W: Line-wise pruning, T-W: Token-wise pruning

Method	Metric	Base	ToP-ViM [74]	Ours
ViM-T	Acc	76.1	75.1	75.5
	Throughput	1603	1758	1754
	GFLOPs	1.5	1.3	1.3

Table 7. ImageNet1K [18] classification results with ViM [77].

We also validate representation shift with State Space Model (SSM) using Vision Mamba (ViM) [77] in Table 7. Overall, we largely follow the settings of ToP-ViM [74], which is designed for accelerating SSM by pruning tokens based on the activated values. We observe the improvements of +0.4% on ViM-T under a similar throughput of Top-ViM. These results suggest that representation shift is a generalizable approach for various architectures.

4.3. Analyses

Qualitative Results. For a deeper understanding of the behavior of representation shift, we provide a qualitative analysis with a visualization. In Figure 6, given the image sample (left), we qualitatively compare the attention-based scores of Equation (2) used in [33, 39], and our proposed representation shift using the DeiT-B [52] consisting of 12 attention layers. To investigate the behavior of each method across early, middle, and deeper layers, we evaluate them at the 1st, 5th, and 9th layers of the model. First, in the early stage (L=1), the attention map generally shows low reliability as discussed in prior works [13, 33], which is not a desirable option for token importance. On the other hand, our representation shift successfully detects the foreground object even in the first layer. In the middle layer (L=5), representation shift still captures the main content better than attention scores. Lastly, in Vision Transformers,

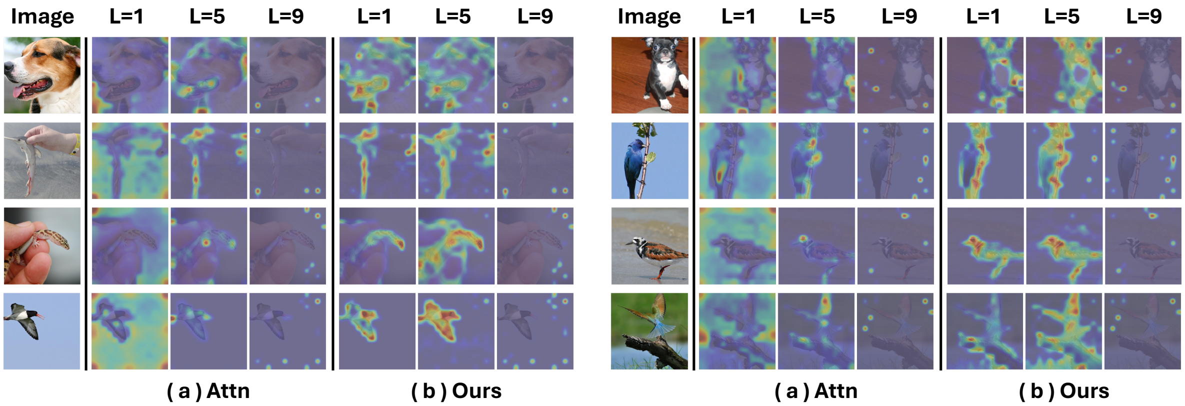


Figure 6. Qualitative comparison between attention scores (Attn) and representation shift (Ours). Given each sample, we visualize (a) the attention scores with respect to the class token and (b) representation shift in the [1,5,9] layers of the DeiT-B [52].

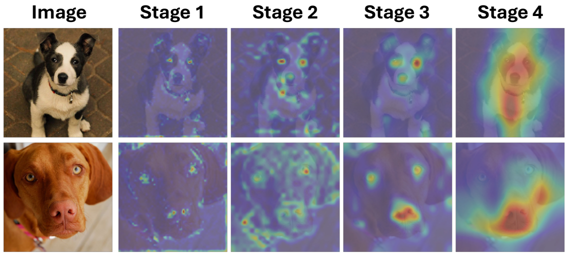


Figure 7. Visualization of representation shift in ResNet-50 [21].

it is well-known that global information is gathered in a few specific tokens as the layer passes [17], having higher attention scores. In this respect, it would be better to mimic the attention map in the latter layer (L=9) to avoid information loss for retaining the informative tokens. To summarize, the representation shift mitigates the low reliance of attention scores in the early layer and finds the salient region till the middle layer, helping the model to capture fine-grained patterns. Further, it enables capturing the token having high-level semantics in the latter layer.

Additionally, we visualize the representation shift through each stage of ResNet-50 [21] in Figure 7. The results reveal that the embedding of the foreground tokens tends to have a more drastic shift than background tokens in every stage. In other words, the network updates the foreground tokens more aggressively, while background tokens, being less critical, undergo only subtle updates. Consequently, the representation shift inherently serves as informativeness of the token to the task, allowing for token pruning without compromising overall performance as shown in Table 6.

Reliability analysis. To assess the reliability of representation shift as an importance metric, we conduct an extreme pruning experiment using DeiT-S [52] on ImageNet-1K [18], where we retain either the top or the bottom 50% of tokens ranked by their representation shift scores. As shown in Table 8, across all transformer layers (L1–L11), retaining the top 50% consistently yields significantly higher accuracy than keeping the bottom 50%, demonstrating the robustness of the importance signal. On average, the top 50% selection achieves 78.0% accuracy, whereas the bottom 50% only reaches 51.7%, resulting in a substantial performance gap of 26.3%. This consistent gap across layers validates that representation shift effectively identifies informative tokens, supporting its reliability.

Token Selection	L1	L3	L5	L7	L9	L11	Avg
Top 50%	76.3	76.1	78.5	79.4	78.5	78.9	78.0
Bottom 50%	51.4	51.9	47.0	49.6	56.1	54.1	51.7

Table 8. Accuracy when retaining top/bottom-50% tokens

5. Conclusion

In this paper, we propose a novel training-free, model-agnostic token importance criterion based on representation shift, which effectively quantifies the information contribution of each operation. Unlike conventional methods, our approach operates independently of attention maps, allowing seamless integration with FlashAttention while achieving competitive accuracy and substantial inference speed improvements. Moreover, its applicability extends beyond Transformers to CNNs, making it a versatile approach for enhancing the efficiency of various vision models while preserving performance. Additionally, we qualitatively demonstrate that our approach successfully detects foreground objects in early and middle layers more effectively than existing methods and informative tokens in latter layers, highlighting its potential as an improved token importance criterion for efficient token compression.

Acknowledgements.

This work was partly supported by IITP grant funded by MSIP & MSIT (No. RS-2024-00443251, No. RS-2024-00457882), NRF grant funded by MSIT (NRF-2023R1A2C2005373), and IITP-ITRC grant funded by MSIT (IITP-2025-RS-2024-00436857).

References

[1] Mujadded Al Rabbani Alif and Muhammad Hussain. Yolov12: A breakdown of the key architectural features. *arXiv:2502.14740*, 2025. 2

[2] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *ICCV*, pages 5803–5812, 2017. 5, 6

[3] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv:2004.05150*, 2020. 1, 2

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. *ICLR*, 2023. 2, 3

[5] Sergey Brin. The pagerank citation ranking: bringing order to the web. *ASIS*, 1998. 3

[6] 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

[7] Han Cai, Junyan Li, Muyan Hu, Chuang Gan, and Song Han. Efficientvit: Multi-scale linear attention for high-resolution dense prediction. *ICCV*, 2023. 2

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

[9] Beidi Chen, Tri Dao, Eric Winsor, Zhao Song, Atri Rudra, and Christopher Ré. Scatterbrain: Unifying sparse and low-rank attention. *NeurIPS*, 2021. 2

[10] David Chen and William B 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. 5, 6

[11] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *CVPR*, 2024. 2

[12] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *NeurIPS*, 2021. 1

[13] Joonmyung Choi, Sanghyeok Lee, Jaewon Chu, Minhyuk Choi, and Hyunwoo J Kim. vid-tldr: Training free token merging for light-weight video transformer. In *CVPR*, 2024. 2, 3, 6, 7

[14] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. *ICLR*, 2021. 2

[15] 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. 1, 2

[16] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 2022. 2, 3

[17] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *ICLR*, 2024. 8

[18] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 3, 4, 7, 8

[19] 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*, 2021. 1, 2, 3

[20] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. In *ICCV*, 2021. 2

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2, 7, 8

[22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*, 2020. 2

[23] Jongha Kim, Jihwan Park, Jinyoung Park, Jinyoung Kim, Sehyung Kim, and Hyunwoo J Kim. Groupwise query specialization and quality-aware multi-assignment for transformer-based visual relationship detection. In *CVPR*, 2024. 1

[24] Sehoon Kim, Sheng Shen, David Thorsley, Amir Gholami, Woosuk Kwon, Joseph Hassoun, and Kurt Keutzer. Learned token pruning for transformers. In *KDD*, 2022. 2, 3

[25] Nikita Kitaev, Łukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. *ICLR*, 2020. 1, 2

[26] Dohwan Ko, Joonmyung Choi, Juyeon Ko, Shinyeong Noh, Kyoung-Woon On, Eun-Sol Kim, and Hyunwoo J Kim. Video-text representation learning via differentiable weak temporal alignment. In *CVPR*, 2022. 1

[27] Dohwan Ko, Joonmyung Choi, Hyeong Kyu Choi, Kyoung-Woon On, Byungseok Roh, and Hyunwoo J Kim. Meltr: Meta loss transformer for learning to fine-tune video foundation models. In *CVPR*, 2023.

[28] Ji Soo Lee, Jongha Kim, Jeehye Na, Jinyoung Park, and Hyunwoo J Kim. Vidchain: Chain-of-tasks with metric-based direct preference optimization for dense video captioning. In *AAAI*, 2025. 1

[29] Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J Kim. Multi-criteria token fusion with one-step-ahead attention for efficient vision transformers. In *CVPR*, 2024. 2

[30] Sanghyeok Lee, Joonmyung Choi, and Hyunwoo J Kim. Efficientvim: Efficient vision mamba with hidden state mixer based state space duality. In *CVPR*, 2025. 2

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

[32] Kunchang Li, Yali Wang, Yizhou Li, Yi Wang, Yinan He, Limin Wang, and Yu Qiao. Unmasked teacher: Towards training-efficient video foundation models. In *ICCV*, 2023. 1, 2, 3, 4, 5, 6

[33] 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, 3, 4, 7

[34] Xinyu Liu, Houwen Peng, Ningxin Zheng, Yuqing Yang, Han Hu, and Yixuan Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In *CVPR*, 2023. 2

[35] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *NeurIPS*, 2021. 2

[36] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, Jianbin Jiao, and Yunfan Liu. Vmamba: Visual state space model. *NeurIPS*, 2024. 2

[37] 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. 1, 2

[38] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 2

[39] Sifan Long, Zhen Zhao, Jimin Pi, Shengsheng Wang, and Jingdong Wang. Beyond attentive tokens: Incorporating token importance and diversity for efficient vision transformers. In *CVPR*, 2023. 2, 3, 4, 7

[40] Sachin Mehta and Mohammad Rastegari. Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer. *ICLR*, 2022. 2

[41] 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. 2, 3

[42] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *TMLR*, 2023. 2

[43] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. IA-RED^2: Interpretability-aware redundancy reduction for vision transformers. *NeurIPS*, 2021. 2, 3

[44] Jinyoung Park, Jeehye Na, Jinyoung Kim, and Hyunwoo J Kim. Deepvideo-r1: Video reinforcement fine-tuning via difficulty-aware regressive grpo. *arXiv preprint*, 2025. 1

[45] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *ICLR*, 2022. 2

[46] 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

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

[48] Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. Efficient content-based sparse attention with routing transformers. *TACL*, 2021. 1, 2

[49] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 1

[50] Diwei Su, Cheng Fei, and Jianxu Luo. Removing rows and columns of tokens in vision transformer enables faster dense prediction without retraining. In *ECCV*, 2024. 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

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2, 3, 4, 7, 8

[53] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021. 1, 3

[54] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 1, 2

[55] 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. 1, 2

[56] Hongjie Wang, Bhishma Dedhia, and Niraj K Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *CVPR*, 2024. 2, 3

[57] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv:2006.04768*, 2020. 1, 2

[58] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 1, 2

[59] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational visual media*, 2022. 1, 2

[60] Xiang Wang, Hangjie Yuan, Shiwei Zhang, Dayou Chen, Jiuniu Wang, Yingya Zhang, Yujun Shen, Deli Zhao, and Jingren Zhou. Videocomposer: Compositional video synthesis with motion controllability. *NeurIPS*, 2023. 2

[61] 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

[62] 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

[63] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based object detection. *AAAI*, 2022. 1

[64] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, et al. Internvideo2: Scaling foundation models for multimodal video understanding. In *ECCV*, 2024. 1, 2

[65] Sanghyun Woo, Shoubhik Debnath, Ronghang Hu, Xinlei Chen, Zhuang Liu, In So Kweon, and Saining Xie. Convnext v2: Co-designing and scaling convnets with masked autoencoders. 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. 1, 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 *ACMMM*, 2017. 5, 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. 3, 4, 5, 6

[69] Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, 2022. 2, 3

[70] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *ICCV*, 2021. 2

[71] Xin Yuan, Hongliang Fei, and Jinoo Baek. Efficient transformer adaptation with soft token merging. In *CVPR*, 2024. 2, 3

[72] Seokju Yun and Youngmin Ro. Shvit: Single-head vision transformer with memory efficient macro design. In *CVPR*, 2024. 2

[73] 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*, 2020. 1, 2

[74] Zheng Zhan, Zhenglun Kong, Yifan Gong, Yushu Wu, Zichong Meng, Hangyu Zheng, Xuan Shen, Stratis Ioannidis, Wei Niu, Pu Zhao, et al. Exploring token pruning in vision state space models. *NeurIPS*, 2024. 7

[75] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *ICLR*, 2023. 1

[76] 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 *CVPR*, 2021. 1

[77] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *ICML*, 2024. 2, 7

[78] 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