Learning Correlation Structures for Vision Transformers

Manjin Kim[1] Paul Hongsuck Seo[2*] Cordelia Schmid[3] Minsu Cho[1*]
[1]POSTECH [2]Korea University [3]Google Research
http://cvlab.postech.ac.kr/research/StructViT/

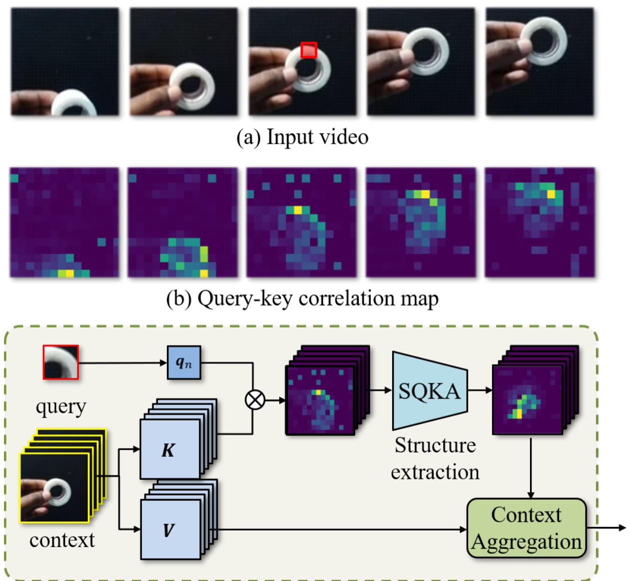
Abstract

We introduce a new attention mechanism, dubbed structural self-attention (StructSA), that leverages rich correlation patterns naturally emerging in key-query interactions of attention. StructSA generates attention maps by recognizing space-time structures of key-query correlations via convolution and uses them to dynamically aggregate local contexts of value features. This effectively leverages rich structural patterns in images and videos such as scene layouts, object motion, and inter-object relations. Using StructSA as a main building block, we develop the structural vision transformer (StructViT) and evaluate its effectiveness on both image and video classification tasks, achieving state-of-the-art results on ImageNet-1K, Kinetics-400, Something-Something V1 & V2, Diving-48, and FineGym.

1. Introduction

How visual elements interact with each other in space and time is a crucial cue for visual understanding, *e.g.*, recognizing actions in a video or analyzing scene layout patterns in an image. In computer vision, such relational patterns are effectively captured by the structure of correlations or similarities across visual elements in different positions [3, 57]; a correlation structure of an image reveals spatial layouts of similar patterns [30, 33] and that of a video provides bi-directional motion likelihoods [32, 36]. The ability to recognize those structural patterns may allow to better perform visual reasoning and generalize against challenging appearance variations and domain shifts [21, 65].

In this work, we introduce a novel self-attention mechanism, named *structural self-attention* (StructSA), that effectively leverages diverse structural patterns for visual representation learning. While the standard self-attention mechanism uses raw query-key correlations individually and ignores their geometric structures, the proposed StructSA recognizes diverse structural patterns from the correlations between the query and local chunks of keys via convolution and uses them to dynamically aggregate local contexts of value features, effectively capturing rich structural patterns such as scene layouts, object motion, and inter-object relations in images and video. As illustrated in Fig. 1 and detailed in Sec. 3, this is mainly achieved by empowering the standard self-attention mechanism with long-range convolutional interactions and dynamic contextual feature aggregation. To investigate the effect of StructSA, we also provide an in-depth analysis on the relationship to recent self-attention variants with convolutional projection [18, 41, 42, 70, 72], showing their potential and limitation in leveraging structural patterns.



(a) Input video
(b) Query-key correlation map
(c) Structural self-attention (StructSA)

Figure 1. **Structural Self-Attention.** Given an input video and a query indicated by the red box in (a), the query-key correlation maps in (b) clearly reveal the structures of spatial layout and motion with respect to the query. The proposed attention mechanism in (c) is designed to leverage these rich structural patterns for computing attention scores in the self-attention process.

To validate the effect of StructSA, we develop the structural vision transformer that adopts it as a main building

*Co-corresponding authors.

block, and perform an extensive set of experiments on both image and video classification tasks, showing the effectiveness of learning structural patterns for visual representations. Our main contributions are as follows:

- We introduce structural self-attention (StructSA) that learns correlation structures for visual representations with the vision transformer.
- We provide an in-depth analysis on the relationship between StructSA and self-attention variants with convolutional projections.
- Our new transformer network achieves state-of-the-art results on ImageNet-1K, Kinetics-400, Something-Something V1&V2, Diving-48, and FineGym.

2. Related Work

Transformer Networks in Vision. Since transformer networks [66] showed remarkable success in natural language processing [5, 14], they have widely been adopted in various computer vision tasks as an alternative to CNNs [1, 7, 17, 58, 59]. Despite of their success, the pure transformer networks require a large amount of training data compared to CNNs where convolution operations introduce desirable inductive biases such as locality and translation invariance allowing more efficient training [17, 53]. This incentivized several methods to inherit the convolutional inductive biases via knowledge distillation [63], local self-attention [27, 45, 54], and architectural fusion [9, 12, 23, 38, 70, 72]. While recent methods using a convolutional projection [18, 41, 70, 72] achieve remarkable improvements, we show that self-attention with a convolutional projection can be derived as a special form of our proposed method.

Correlation Structure Modeling. Geometric structure of correlations between visual features, *i.e.*, patterns of how they are similar to each other, allows us to understand relational patterns in visual data for various computer vision tasks. Spatial self-correlation in images is used for suppressing photometric variations and revealing geometric layout of objects in the image [30, 33, 57]. Spatial cross-correlation between different images is often used for establishing semantic correspondences capturing structural similarities [24, 50, 55]. In the video domain, several methods exploit the structure of spatial cross-correlations between consecutive frames to estimate optical flow [16, 74] or to learn motion features for action recognition [35, 67]. Kwon *et al.* [36] propose spatio-temporal self-correlations for learning bi-directional motion features and Kim *et al.* [32] introduce relational self-attention that generates attention weights dynamically from the structure of the spatio-temporal self-correlations. However, these two methods use self-correlations between the query and its local spatio-temporal neighborhoods only, thus, are limited in learning global relational patterns between distant features. Inspired by this, we introduce structural self-attention that capturing not only the spatio-temporal local self-correlation but also cross-correlations between features in the distance, utilizing both motion and global spatio-temporal inter-feature relations for learning motion-centric video representations.

3. Our Approach

We propose a novel self-attention mechanism, named *structural self-attention* (StructSA), that is designed to leverage rich correlation structures naturally emerging in key-query interactions of attention. We start by revisiting the vanilla self-attention and its limitation and then describe the details of StructSA. We also provide an in-depth analysis of recent self-attention variants with convolutional projections from the perspective of learning structural patterns.

3.1. Background: Self-Attention

Self-attention (SA) [66] is a primitive operation for modern transformer networks [1, 17, 63]. Given N input features $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N] \in \mathbb{R}^{N \times C}$, SA first projects the input $\boldsymbol{X}$ linearly into queries, keys, and values, and transforms each C-dimensional input feature $\boldsymbol{x}_i$ into a contextualized output feature $\boldsymbol{y}_i$ by

$$\boldsymbol{y}_i = \sigma\left(\boldsymbol{q}_i \boldsymbol{K}^\top\right) \boldsymbol{V} = \sum_j^N \sigma_j \left(\boldsymbol{q}_i {\boldsymbol{k}_j}^\top\right) \boldsymbol{v}_j \in \mathbb{R}^{1 \times C}, \quad (1)$$
$$\boldsymbol{q}_i = \boldsymbol{x}_i \boldsymbol{W}^{\mathrm{Q}}, \quad \boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^{\mathrm{K}}, \quad \boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}^{\mathrm{V}}$$

where σ is a softmax function and $\boldsymbol{W}^Q, \boldsymbol{W}^K, \boldsymbol{W}^V \in \mathbb{R}^{C \times C}$ are projection matrices for query $\boldsymbol{q}_i \in \mathbb{R}^{1 \times C}$, keys $\boldsymbol{K} \in \mathbb{R}^{N \times C}$, and values $\boldsymbol{V} \in \mathbb{R}^{N \times C}$, respectively. Here we use a 1-dimensional sequence of input features for notational simplicity, and the operation can be extended to a larger dimensionality. After computing a correlation map $\boldsymbol{q}_i \boldsymbol{K}^\top$, the vanilla SA uses individual correlation values independently, *i.e.*, $\boldsymbol{q}_i {\boldsymbol{k}_j}^\top$, for value aggregation while ignoring the *structure* of the map, which leads to the same output regardless of the order of features. This permutation invariance prevents SA from capturing spatial layouts [27, 57] or motions [32, 36, 44] of objects in images or videos. Positional encoding for SA helps spatial awareness, but the *structures* of correlations are still not recognized in value aggregation [10, 11, 32].

3.2. Structural Self-Attention

We introduce a novel self-attention mechanism, named *structural self-attention* (StructSA), that effectively incorporates rich structural patterns of query-key correlation into contextual feature aggregation. The StructSA mechanism consists of two steps: (i) structural query-key attention and (ii) contextual value aggregation. Unlike the vanilla query-key attention where individual correlation values themselves are used as attention scores, the *structural query-key* *attention* takes the correlation map as a whole and detect structural patterns from it in attention scoring. The subsequent *contextual value aggregation* then combines the attention scores together to compute diverse sets of kernel weights that are used for dynamically collecting local contexts of value features.

Structural Query-Key Attention. To transform the vanilla query-key attention into structure-aware one, the structural query-key attention (SQKA) deploy convolutions on top of query-key correlation $\boldsymbol{q}_i \boldsymbol{K}^{\mathsf{T}}$:

$$\boldsymbol{A}_i = \sigma\left(\mathtt{conv}\left(\boldsymbol{q}_i \boldsymbol{K}, \boldsymbol{U}^{\mathrm{K}}\right)\right) \in \mathbb{R}^{N \times D}, \tag{2}$$

where $\boldsymbol{U}^{\mathrm{K}} \in \mathbb{R}^{M \times D}$ is D convolutional kernels with size M. Note that σ is a softmax function taken over all ND entries in the input matrix; we observe that it is empirically more stable compared to D individual softmax operations over N entries. Each element of $\boldsymbol{A}_i$ is computed as

$$\begin{aligned} \boldsymbol{a}_{i,j} &= \sigma_j\left(\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}} \boldsymbol{U}^{\mathrm{K}}\right) \in \mathbb{R}^{1 \times D}, \\ \boldsymbol{K}_j &= \boldsymbol{X}_{(j)} \boldsymbol{W}^{\mathrm{K}} \in \mathbb{R}^{M \times C}, \end{aligned} \tag{3}$$

where σ_j returns a D-dimensional softmax-ed output for jth location and $\boldsymbol{X}_{(j)} \in \mathbb{R}^{M \times C}$ is local context features whose context window is centered at j.

Unlike the vanilla query-key attention, which is agnostic to its neighborhood structure, SQKA is empowered by convolution to recognize a local correlation structure of $\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times M}$ and transform it into a D-dimensional vector; the convolution kernels $\boldsymbol{U}^{\mathrm{K}}$ act as *correlation pattern detectors*. In particular, when $i = j$, the correlation map reduces to local self-similarity [57] that is known to be effective for capturing spatial layout patterns [57] or spatio-temporal motion [32, 36], meaning that SQKA recognizes diverse correlation patterns including self-similarity [57] via long-range interaction between query i and context j.

Contextual Value Aggregation. Given SQKA recognizing structural patterns of correlation, StructSA combines SQKA entries into a weight $\kappa_{i,j}^{\text{struct}}$ to aggregate value $\boldsymbol{v}_j$:

$$\boldsymbol{y}_i = \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}} \boldsymbol{U}^{\mathrm{K}}\right) \boldsymbol{u}^{\mathrm{V}\mathsf{T}} \boldsymbol{v}_j = \sum_{j=1}^{N} \kappa_{i,j}^{\text{struct}} \boldsymbol{v}_j, \tag{4}$$

where $\boldsymbol{u}^{\mathrm{V}} \in \mathbb{R}^{1 \times D}$ is a vector that linearly combines D pattern scores to the final attention weight.

We further extend Eq. (4) to generate a spatial kernel not a single scalar for each position j. We call this method contextual aggregation and it has a following form:

$$\begin{aligned} \boldsymbol{y}_i = \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}} \boldsymbol{U}^{\mathrm{K}}\right) \boldsymbol{U}^{\mathrm{V}\mathsf{T}} \boldsymbol{V}_j &= \sum_{j=1}^{N} \boldsymbol{\kappa}_{i,j}^{\text{struct}} \boldsymbol{V}_j, \\ \boldsymbol{V}_j = \boldsymbol{X}_{(j)} \boldsymbol{W}^{\mathrm{V}} &\in \mathbb{R}^{M \times C}. \end{aligned} \tag{5}$$

Compared to $\boldsymbol{u}^{\mathrm{V}}$ that produces a scalar weighting a single element $\boldsymbol{v}_j$, $\boldsymbol{U}^{\mathrm{V}} \in \mathbb{R}^{M \times D}$ generates spatial kernel dynamically aggregating a local context of value $\boldsymbol{V}_j$ for every position j; each column of $\boldsymbol{U}^{\mathrm{V}}$, *i.e.*, $\boldsymbol{U}_{:,d}^{\mathrm{V}} \in \mathbb{R}^{M \times 1}$, plays a role as a *context aggregator* that performs a weighted pooling of local context $\boldsymbol{V}_j$ and thus different combinations of these context aggregators result in diverse dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$ for different locations j.

3.3. Relationship to Convolutional Self-Attention

Recent vision transformers [18, 23, 41, 70, 72] often adopt convolutional inductive biases in the form of self-attention with convolutional projections (ConvSA). In this section, we analyze ConvSA with a lens of StructSA and show its potential and limitation for learning structures from query-key correlations. Different from SA, ConvSA computes project keys and values $\boldsymbol{K}^{\text{conv}}, \boldsymbol{V}^{\text{conv}} \in \mathbb{R}^{N \times C}$ using a convolution operation over the input feature map $\boldsymbol{X}$:

$$\boldsymbol{K}^{\text{conv}} = [\boldsymbol{k}_1^{\text{conv}}, \cdots, \boldsymbol{k}_N^{\text{conv}}] = \mathtt{conv}(\boldsymbol{X}, \mathbf{W}^{\mathrm{K}}), \tag{6}$$

$$\boldsymbol{V}^{\text{conv}} = [\boldsymbol{v}_1^{\text{conv}}, \cdots, \boldsymbol{v}_N^{\text{conv}}] = \mathtt{conv}(\boldsymbol{X}, \mathbf{W}^{\mathrm{V}}), \tag{7}$$

where $\mathtt{conv}$ is a convolution operation, and $\mathbf{W}^{\mathrm{K}}, \mathbf{W}^{\mathrm{V}} \in \mathbb{R}^{M \times C \times C}$ are kernel weights with a kernel size M for key and value projections, respectively.

In most previous methods, ConvSA is implemented with a channel-wise separable convolution [26], which consists of two factorized convolution operations, *i.e.*, point-wise and channel-wise convolutions [18, 23, 41, 70, 72]. In this case, each key $\boldsymbol{k}_i^{\text{conv}}$ and value $\boldsymbol{v}_i^{\text{conv}}$ is computed from a local context $\boldsymbol{X}_i$ by

$$\boldsymbol{k}_i^{\text{conv}} = \boldsymbol{u}^{\mathrm{K}\mathsf{T}} \boldsymbol{X}_i \boldsymbol{W}^{\mathrm{K}} = \boldsymbol{u}^{\mathrm{K}\mathsf{T}} \boldsymbol{K}_i \in \mathbb{R}^{1 \times C}, \tag{8}$$

$$\boldsymbol{v}_i^{\text{conv}} = \boldsymbol{u}^{\mathrm{V}\mathsf{T}} \boldsymbol{X}_i \boldsymbol{W}^{\mathrm{V}} = \boldsymbol{u}^{\mathrm{V}\mathsf{T}} \boldsymbol{V}_i \in \mathbb{R}^{1 \times C}, \tag{9}$$

where $\boldsymbol{W}^{\mathrm{K}}, \boldsymbol{W}^{\mathrm{V}} \in \mathbb{R}^{C \times C}$ are weights for the linear projection that are equivalent to point-wise convolution, and $\boldsymbol{u}^{\mathrm{K}}, \boldsymbol{u}^{\mathrm{V}} \in \mathbb{R}^{M \times 1}$ are channel-wise convolution weights that are used to spatially aggregate linearly projected context $\boldsymbol{K}_i$ and $\boldsymbol{V}_i$, respectively. Note that here we assume the channel-wise convolution weights are shared across channels for simplicity without loss of generality and the full derivation is available in our supplementary materal A.

From Eq. (1) combined with Eq. (8) and (9), a transformed output $\boldsymbol{y}_i$ in ConvSA is obtained by

$$\begin{aligned} \boldsymbol{y}_i &= \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i \boldsymbol{k}_j^{\text{conv}\mathsf{T}}\right) \boldsymbol{v}_j^{\text{conv}} \\ &= \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}} \boldsymbol{u}^{\mathrm{K}}\right) \boldsymbol{u}^{\mathrm{V}\mathsf{T}} \boldsymbol{V}_j = \sum_{j=1}^{N} \boldsymbol{\kappa}_{i,j}^{\text{conv}} \boldsymbol{V}_j, \end{aligned} \tag{10}$$

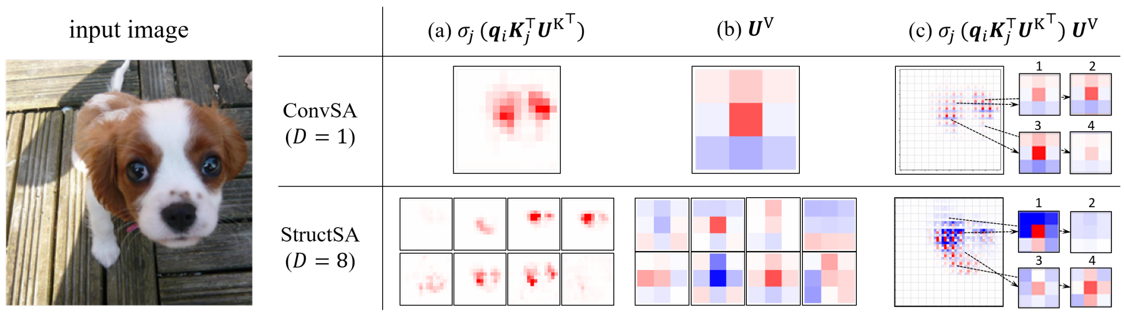


Figure 2. **Visualization of ConvSA and StructSA on ImageNet-1K.** The query location i is set to the center of the image and the kernel size $M = 3 \times 3$. Given the left input image, we compare ConvSA ($D = 1$) and StructSA ($D = 8$) in terms of (a) D attention maps $\sigma_{jD}(\boldsymbol{q}_i\boldsymbol{K}_j^{\intercal}\boldsymbol{U}^{\mathrm{K}^{\intercal}})$, (b) local feature aggregation patterns learned in $\boldsymbol{U}^{V}$, and (c) the combinations of (a) and (b). Note that in (c), each location j has an aggregation map of the kernel size $M = 3\times3$ and thus we also show enlarged images for four different sample locations j.

where σ_j is jth entry of the softmax over N tokens. This reveals that an attention score $\sigma_j(\boldsymbol{q}_i\boldsymbol{k}_j^{\mathrm{conv}\intercal})$ is computed by projecting a local correlation map $\boldsymbol{q}_i\boldsymbol{K}_j^{\intercal} \in \mathbb{R}^{1\times M}$ by $\boldsymbol{u}^{\mathrm{K}}$, and a dynamic kernel $\boldsymbol{\kappa}_{i,j}^{\mathrm{conv}}$ for the final feature aggregation of $\boldsymbol{V}_j$ is obtained by weighting the aggregation pattern presented in $\boldsymbol{u}^{\mathrm{V}}$ using the computed attention map. Given that correlation map $\boldsymbol{q}_i\boldsymbol{K}_j^{\intercal}$ represents a structural pattern, we can interpret that $\boldsymbol{u}^{\mathrm{K}}$ acts as a *pattern detector* that extracts a specific correlation pattern from $\boldsymbol{q}_i\boldsymbol{K}_j^{\intercal}$, whereas $\boldsymbol{u}^{\mathrm{V}}$ plays a role as a *context aggregator* that performs a weighted pooling of local context $\boldsymbol{V}_j$. Due to the presence of this pattern detector $\boldsymbol{u}^{\mathrm{K}}$ and its corresponding context aggregator $\boldsymbol{u}^{\mathrm{V}}$, ConvSA can leverage a structural pattern of input for context aggregation.

Limitation. Although ConvSA can learn, unlike SA, a structural pattern over correlation maps by $\boldsymbol{u}^{\mathrm{K}}$, it only learns a single pattern and encodes various shapes in correlation maps into a scalar value representing the similarity against the learned pattern; as the result, the final dynamic kernel $\boldsymbol{\kappa}_{i,j}^{\mathrm{conv}}$ for every j reduces to the identical pattern of $\boldsymbol{u}^{\mathrm{V}}$ with different weighting only. This lack of expressiveness in $\boldsymbol{u}^{\mathrm{K}}$ and $\boldsymbol{u}^{\mathrm{V}}$ prevents ConvSA from capturing diverse structural patterns and generating diverse dynamic kernels. In constrast, StructSA learns D different pattern extractors in $\boldsymbol{U}^{\mathrm{K}}$ and represents various local correlation shapes by a set of D similarity scores. These scores are then combined with the D context aggregators in $\boldsymbol{U}^{\mathrm{V}}$; different combinations of these context aggregators result in diverse dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\mathrm{struct}}$ for different locations j.

Visualization. The aforementioned difference between ConvSA and StructSA, as well as their effects, can be better understood by visualizing the kernel computation process. Figure 2 provides such a visual comparison of how structural patterns are used in ConvSA and StructSA given an example image from ImageNet-1K [13]. From a query-key correlation map, ConvSA generates a single attention map (Fig. 2a, top). These scores are then combined with the context aggregator $\boldsymbol{u}^{\mathrm{V}}$ (Fig. 2b, top), which conveys only a single aggregation pattern. This causes local features to be aggregated with the identical pattern in $\boldsymbol{u}^{\mathrm{V}}$ for all locations, and the only difference remains in their scales (Fig. 2c, top). In contrast, StructSA generates diverse attention maps using D pattern detectors, each capturing different structures in the query-key correlation maps (Fig. 2a, bottom), and combines them with different context aggregators (Fig. 2b, bottom) resulting in rich aggregation patterns for different locations j (Fig. 2c, bottom). For more in-depth comparison, please refer to our supplementary material D.

4. Experiments

To validate the effectiveness of the proposed method on visual representation learning, we conduct extensive experiments on image and video classification benchmarks.

4.1. Experimental Setup

Datasets. **ImageNet-1K** [13] is a large-scale dataset with 1.2M images labeled by 1000 object classes. **Kinetics-400** [31] is one of the most popular large-scale video datasets with 400 action classes. We use 241k action clips available online. **Something-Something-V1 & V2** [22, 49] are both large-scale action recognition benchmarks, including 108k and 220k action clips, respectively. Both datasets share the same motion-centric action classes, e.g., 'pushing something from left to right,' so thus capturing fine-grained motion is crucial to achieving the better performance. **Diving-48** [39] is a fine-grained action benchmark that is heavily dependent on temporal modeling [3], containing 18k videos with 48 diving classes. **FineGym** [56] is a motion-centric benchmark that includes gymnastics action classes with severe deformations.

Training & Testing Protocols. For image classification, we follow the training strategy of DeiT [63] adopting random clipping, random horizontal flipping, mixup [80], cutmix [78], random erasing [81] and label-smoothing [51] to augment the input images for training. We train all models from scratch for 300 epochs using AdamW optimizer [48] with a cosine learning rate schedule including 5 warm-up epochs. The batch size, learning rate, and weight decay are set to 1024, 1e-3, and 0.05, respectively. For comparison on stronger experiment setup [29, 38, 41, 77], we also train our models using Token Labeling [29] and larger resolution images, *i.e.*, 384 × 384, following the protocols in [38].

For video classification, we follow training protocols and data augmentation recipes in MViT [18]. For Kinetics-400, we sample 16 or 32 frames using the dense sampling strategy [71]. We temporally inflate the model weights pretrained on ImageNet-1K and finetune it for 110 epochs including 10 warm-up epochs. We use AdamW [48] optimizer with the cosine learning rate schedule. We set the batch size, learning rate, weight decay, and stochastic depth rate to 64, 2e-4, 0.05, and 0.1, respectively. For Something-Something, Diving-48, and FineGym, we utilize the segment-based sampling strategy [68] and do not use the random horizontal flip for data augmentation. We initialize the model with the weights pretrained on Kinetics-400 and finetune the model for 60 epochs including 5 warm-up epochs. Other training hyperparameters are the same as those for Kinetics-400. For testing, we sample multiple clips at different temporal indices for each clip or cropping different spatial regions and then obtain the final score by computing an average over the scores for each clip. We train all models once using 8 to 16 NVIDIA A100 GPUs.

Metrics. We measure top-1 and top-5 accuracy as performance metrics, except for FineGym, we compute averaged per-class top-1 accuracy. As efficiency metrics, we measure the number of parameters and FLOPs.

4.2. Analysis of StructSA

StructSA can be readily integrated into any existing ViTs to enhance visual representations by capturing correlation structures. In this subsection, we experimentally validate and analyze the impact of StructSA. Here, for a direct comparison with SA, we choose to use DeiT-S [63] as the baseline backbone; DeiT is a pure SA-based vision transformer and thus adequate for validating the effect of StructSA, avoiding any intervention of additional components. In this analysis, we replace all the SA layers in DeiT with StructSA layers. The evaluations are done on ImageNet-1K [13] and Something-Something-V1 [22] benchmarks while varying the structure dimension D, the kernel size M, and context aggregation methods. We follow the training and testing protocols in Sec. 4.1, except that we directly finetune the ImageNet-1K-pretrained model on Something-Something-V1 using random cropping only for data augmentation.

Structure Dimension D. Table 1a shows the effect of the structure dimension D. Compared to the baseline with the vanilla SA ($D = 0$), applying ConvSA ($D = 1$) improves the performance as shown in [9, 72]. As we increase D from 1 to 8, we obtain gradual improvements up to 0.5%p and 0.9%p at top-1 accuracy on ImageNet-1K and Something-Something-V1. This confirms the limitation of ConvSA and the effectiveness of StructSA.

D	ImageNet-1K		Something V1	
	top-1	top-5	top-1	top-5
0	80.5	95.0	48.3	76.6
1	80.8	95.2	49.7	77.6
2	80.9	95.2	50.1	78.0
4	81.1	95.4	50.4	78.2
8	**81.3**	**95.5**	**50.6**	**78.5**

(a) **Structure dimension** D.

M	ImageNet-1K		Something V1	
	top-1	top-5	top-1	top-5
-	80.5	95.0	48.3	76.6
$1 \times 1\,(\times 1)$	80.6	95.0	48.5	76.9
$3 \times 3\,(\times 3)$	**81.1**	**95.4**	50.4	**78.2**
$5 \times 5\,(\times 5)$	**81.1**	**95.4**	**50.5**	**78.2**
$7 \times 7\,(\times 7)$	81.0	95.2	**50.5**	78.1

(b) **Kernel size** M.

aggregation	ImageNet-1K		Something V1	
	top-1	top-5	top-1	top-5
-	80.5	95.0	48.3	76.6
element	80.9	95.2	49.6	77.5
context	**81.1**	**95.4**	**50.4**	**78.2**

(c) **Context aggregation method.**

Table 1. **Ablation studies on ImageNet-1K and Something-Something V1**. Top-1 and top-5 accuracies (%) are shown. Otherwise specified, we use 16 frames as input and set $D = 4$, $M = 3 \times 3 \times 3$, and patch-wise context aggregation as default.

Kernel Size M. In Table 1b, we also investigate different kernel sizes M. Compared to the baseline, the model with the kernel size $M = 1 \times 1 \times 1$ performs similar accuracies on both datasets whereas that with $M = 3 \times 3 \times 3$ improves the performance dramatically; it validates the effectiveness of learning geometric structures. The performance saturates as the kernel size gets larger than $5 \times 5 \times 5$.

Context Aggregation Method. In Table 1c, we also compare different context aggregation method. As a result, patch-wise aggregation performs 0.2%p and 0.7%p at top-1 accuracy on ImageNet-1K and Something-Something-V1. For more ablation experiments, please refer to our supplementary material B.

4.3. Comparison to State of the Art

4.3.1 Structural Vision Transformer (StructViT)

To build an advanced vision transformer considering the recent development of multiscale representation learning [18, 38, 41, 70, 76], we integrate StructSA into UniFormer [38]. The transformer network, dubbed the Structural Vision

model	type	# blocks	# channels (# heads)
StructViT-S	[C,C,S,S]	[3, 4, 8, 3]	[64, 128, 320 (5), 512 (8)]
StructViT-B	[C,C,S,S]	[5, 8, 20, 7]	[64, 128, 320 (5), 512 (8)]
StructViT-L	[C,C,S,S]	[5, 10, 24, 7]	[128, 192, 448 (7), 640 (10)]

Table 2. **Configurations of StructViT variants**. "C" and "S" denote a convolution and StructSA block, respectively.

Transformer (StructViT) is constructed by replacing all vanilla self-attention in UniFormer with StructSA as a main building block. It takes as input either a video clip or an image $\tilde{\boldsymbol{X}} \in \mathbb{R}^{T \times H \times W \times 3}$, where T, H, and W denotes the temporal length, height, and width of the input, respectively. For images as input, we set the temporal length $T = 1$. Before being fed into our networks, the input video is tokenized into overlapping 3D tublets of size $3 \times 4 \times 4 \times 3$ with a stride of $2 \times 4 \times 4$ while the input image is into non-overlapping 2D patches of size $4 \times 4 \times 3$.

Our networks comprise four stages, each of which has multiple neural blocks, and leverage a hierarchical design with decreasing resolutions and increasing number of channels from early to late stages following [38]. For the first two stages, each block consists of a conditional positional encoding layer [9], a convolutional layer, and an MLP, whereas the convolutional layer is replaced by a StructSA layer in the blocks in the last two stages. Note that when we employ StructSA, we use multi-head configurations and do not share weights across channels for channel-wise convolutions. We build three different StructViT architectures as shown in Table 2. Our models are comparable to UniFormers [38] in the same sizes as our model configurations are based on UniFormer's; adding the structure dimension D in StructSA introduces only few additional parameters.

In practice, StructSA introduces additional FLOPs for processing instances compared to the vanilla SA. One way of building an efficient StructSA is to adopt a larger stride in the key/value projections, which effectively reduces the number of keys and values [18, 41]. We test a few variants with a larger stride to see the performances of StructViT with matching FLOPs with their corresponding UniFormer architectures. We denote each model with StructViT-X-D-S where X, D, and S represent the architecture size, the structure dimension, and the stride, respectively. For training, we use stochastic depth [28] with the probability of 0.1/0.3/0.4 for StructViT-S/B/L, respectively. We use random cropping only for StructViT-B on Something-Something, Diving-48, and FineGym. We use 8 NVIDIA A100 GPUs for training StructViT-S/B and 16 GPUs for StructViT-L. We follow the protocols in Sec. 4.1 for the rest.

4.3.2 Image Classification

In Table 3, we compare StructViT with other state-of-the-art CNNs, ViTs, and their hybrid models. The results show that StructViT outperforms other methods in all sizes. Compared to EfficientNets [61, 62] that are obtained by extensive architecture search, our models show comparable or even better performances in both base and large configura-

method	#params (M)	FLOPs (G)	IN1K top-1
EffcientNet-B5 [61]	30	9.9	83.6
ConvNext-T [46]	29	4.5	83.1
DeiT-S [63]	22	4.6	79.9
PVT-S [70]	25	3.8	79.8
Swin-T [45]	29	4.5	81.3
Focal-T [75]	29	4.9	82.2
CSwin-T [15]	23	4.3	82.7
CvT-13 [72]	20	4.5	81.6
CoAtNet-0 [12]	25	4.2	81.6
LV-ViT-S [29]	26	6.6	83.3
UniFormer-S [38]	22	3.6	82.9
UniFormer-S* ↑384 [38]	22	11.9	84.6
MViTv2-S [41]	24	4.7	82.3
StructViT-S-4-2	23	3.6	82.9
StructViT-S-4-1	23	4.3	83.2
StructViT-S-8-1	24	5.4	83.3
StructViT-S-4-1*	23	4.3	84.0
StructViT-S-4-1* ↑384	23	17.3	**85.2**
EffcientNet-B7 [61]	66	39.2	84.3
ConvNext-B [46]	89	15.4	83.8
ConvNext-B ↑384 [46]	89	45.0	85.1
PVT-L [70]	61	9.8	81.7
Swin-S [45]	50	8.7	83.0
Focal-S [75]	51	9.1	83.5
CSwin-S [15]	35	6.9	83.6
CvT-21 [72]	32	7.1	82.5
CoAtNet-1 [12]	42	8.4	83.3
LV-ViT-M [29]	56	16.0	84.1
UniFormer-B [38]	50	8.3	83.8
UniFormer-B* ↑384 [38]	50	27.2	86.0
MViTv2-S [41]	35	7.0	83.6
MViTv2-B [41]	52	10.2	84.4
MViTv2-B ↑384 [41]	52	36.7	85.2
StructViT-B-4-2	51	8.3	84.0
StructViT-B-4-1	51	9.9	84.2
StructViT-B-8-1	52	12.0	84.3
StructViT-B-4-1*	51	9.9	85.4
StructViT-B-4-1* ↑384	51	40.7	**86.5**
EfficientNetV2-L [62]	121	52	85.7
ConvNext-L [46]	198	34.4	84.3
ConvNext-L ↑384 [46]	198	101.0	85.5
Swin-B [45]	88	15.4	83.3
Focal-B [75]	90	16.0	83.8
CSwin-B [15]	78	15.0	84.2
CoAtNet-3 [12]	168	34.7	84.5
LV-ViT-L ↑288 [29]	150	59.0	85.3
UniFormer-L* [38]	100	12.6	85.6
UniFormer-L* ↑384 [38]	100	39.2	86.0
MViTv2-L [41]	218	42.1	85.3
MViTv2-L ↑384 [41]	218	140.2	86.0
StructViT-L-4-1*	103	15.4	86.0
StructViT-L-4-1* ↑384	103	85.2	**86.7**

Table 3. **Comparisons to the state-of-the-art methods on ImageNet-1K.** *Trained with Token Labeling [29].

tions, requiring much less amount of computational cost. Compared to our baseline, UniFormers, StructViTs consistently improve top-1 accuracy regardless of their sizes, demonstrating the benefits of learning geometric structures in image understanding. We further evaluate our models on stronger setup using Token Labeling [29] and 384×384 images, and observe consistent improvements over the baselines. While StructSA introduces some additional FLOPs, we also test variants whose stride for key/value convolutions is set to 2 (S-4-2 and B-4-2) matching its FLOPs to that of the baseline; we still observe some gain with the base model (B-4-2) without additional FLOPs while the small model (S-4-2) shows comparable results. For more comprehensive analysis, we provide experimental results on dense prediction tasks, *i.e.*, object detection, semantic segmentation, and instance segmentation. Please refer to our supplementary material C for the detail.

4.3.3 Video Classification

Kinetics-400. Table 4 compares our method with previous state-of-the-art methods on Kinetics-400. Each block in the table groups methods based on their network structures: CNNs, ViTs, and hybrid methods. We first observe that our best model (B-4-1) achieves state-of-the-art performance. Our method outperforms CNN-based approaches even with less computational cost (S-4-2) in most cases. Compared to MoViNets [34] which are the most advanced CNNs obtained by an extensive NAS, our method shows comparable scores with fewer FLOPs (S-4-1).

When we compare our model to the ViT-based ones, our model outperforms them by large margins while using significantly fewer compute. For instance, StructViT-B-4-1 with single crop (second last row in Table 4) shows top-1 accuracy gain by 1.6%p while using only 55% of computes compared to MTV-B, the best performing ViT-based model. Note also that our model is pretrained on ImageNet-1K, which is much smaller than ImageNet-21K on which the ViT-based models are pretrained.

Finally, our best models (S-8-1 & B-4-1) show 0.5%p to 0.8%p accuracy gains over the baseline UniFormer models in different size configurations. When we use larger strides (S-4-2 and B-4-2) to match the FLOPs of the baselines, we still observe accuracy gains ranging from 0.2%p to 0.3%p.

Something-Something, Diving-48 and FineGym. Table 5a summarizes the results on Something-Something-V1&V2. We observe the same trends as on Kinetics-400. StructViT-S-4-2 outperforms UniFormer-S on Something-Something-V2 by 0.6%p in top-1 accuracy, and StructViT-S-8-1 enlarges the gap to 1.3%p, leveraging correlation structures more effectively. StructViT-B-4-1 achieves new state-of-the-art performances on both V1 and V2 without the strong data augmentation methods used in [38, 41].

method	pretrain	frame× crop×clip	FLOPs (G)	K400 top-1	K400 top-5
SlowFast+NL [20]	-	16×3×10	7020	79.8	93.9
ip-CSN [64]	Sports1M	32×3×10	3270	79.2	93.8
X3D-XL [19]	-	16×3×10	1452	79.1	93.9
MoViNet-A5 [34]	-	120×1×1	281	80.9	94.9
MoViNet-A6 [34]	-	120×1×1	386	81.5	95.3
TimeSformer-HR [4]	IN-21K	16×3×1	5109	79.7	94.4
TimeSformer-L [4]	IN-21K	96×3×1	7140	80.7	94.7
X-ViT [6]	IN-21K	16×3×1	850	80.2	94.7
Mformer-HR [52]	IN-21K	16×3×10	28764	81.1	95.2
ViViT-L [1]	IN-21K	16×3×4	17352	80.6	94.7
Swin-B [47]	IN-1K	32×3×4	3384	80.6	94.6
Swin-B [47]	IN-21K	32×3×4	3384	82.7	95.5
MTV-B [73]	IN-21K	32×3×4	4790	81.8	95.0
MViT-B,16×4 [18]	-	16×1×5	353	78.4	93.5
MViT-B,32×3 [18]	-	32×1×5	850	80.2	94.4
Dualformer-S [42]	IN-1K	32×1×4	636	80.6	94.9
Dualformer-B [42]	IN-1K	32×1×4	1072	81.1	95.0
UniFormer-S [38]	IN-1K	16×1×4	167	80.8	94.7
UniFormer-B [38]	IN-1K	32×1×4	1036	82.9	95.4
MViTv2-B,32×3 [41]	-	32×1×5	1125	82.9	95.7
StructViT-S-4-2	IN-1K	16×1×4	169	81.1	95.5
StructViT-S-4-1	IN-1K	16×1×4	327	81.4	95.7
StructViT-S-8-1	IN-1K	16×1×4	541	81.6	95.8
StructViT-B-4-2	IN-1K	32×1×4	1045	83.1	95.5
StructViT-B-4-1	IN-1K	32×1×4	2658	83.3	95.6
StructViT-B-4-1	IN-1K	32×3×4	7974	**83.4**	**95.8**

Table 4. **Comparisons to the state-of-the-art methods on Kinetics-400.**

Table 5b and Table 5c show the results on Diving-48 [39] and FineGym [56]. Our models outperform the baseline, UniFormer-B, obtaining significant accuracy gains by 0.9%p and 0.7%p on Diving-48 and FineGym, respectively. This indicates that learning spatio-temporal correlation structures play a crucial role in capturing fine-grained motion patterns. Our model sets new state-of-the-art performances with large margins (4.1%p on Diving-48; 3.3%p and 3.1%p on FineGym) over the previous methods without additional box annotations on both datasets. Note that ORViT [25] uses additional object bounding box annotations to train an object detector.

4.4. Visualizations of StructSA

Figure 3 visualizes example dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$ computed from self-similarity map ($i = j$) on Something-Something-V1 to illustrate how StructSA encodes motion features from the spatiotemporal correlation structure. We observe that StructSA builds kernels for spatiotemporal gradient filters similar to those that are already known to be effective for capturing different types of motions [60], *e.g.*, Sobel filters (first) or Laplacian filters (second and third), over local contexts similarly to [32].

method	pretrain	frame× crop×clip	FLOPs (G)	Something V1 top-1	Something V1 top-5	Something V2 top-1	Something V2 top-5
TEA [40]	IN-1K	16×1×1	70	51.9	80.3	-	-
MSNet [35]	IN-1K	16×1×1	101	52.1	82.3	64.7	89.4
CT-Net [37]	IN-1K	16×1×1	75	52.5	80.9	64.5	89.3
TDN [69]	IN-1K	16×1×1	72	53.9	82.1	65.3	89.5
SELFYNet [36]	IN-1K	16×1×1	77	54.3	82.9	65.7	89.8
RSANet [32]	IN-1K	16×1×1	72	54.0	81.1	66.0	89.9
TimeSformer-HR [4]	IN-21K	16×3×1	5109	-	-	62.5	-
TimeSformer-L [4]	IN-21K	96×3×1	7140	-	-	62.3	-
ViViT-L [1]	K400	16×3×4	11892	-	-	65.4	89.8
X-ViT [6]	IN-21K	32×3×1	1270	-	-	65.4	90.7
Mformer-HR [52]	K400	16×3×1	2876	-	-	67.1	90.6
Mformer-L [52]	K400	32×3×1	3555	-	-	68.1	91.2
Swin-B [47]	K400	32×3×1	963	-	-	69.6	92.7
MViT-B,64×3 [18]	K400	64×1×3	1365	-	-	67.7	90.9
MViT-B-24,32×3 [18]	K600	32×1×3	708	-	-	68.7	91.5
UniFormer-S [38]	K400	16×3×1	125	57.2	84.9	67.7	91.4
UniFormer-B [38]	K400	32×3×1	777	60.9	87.3	71.2	92.8
MViTv2-B [41]	K400	32×3×1	675	-	-	70.5	92.7
StructViT-S-4-2	K400	16×3×1	126	57.6	85.3	68.3	91.3
StructViT-S-4-1	K400	16×3×1	246	58.0	85.5	68.8	91.9
StructViT-S-8-1	K400	16×3×1	405	58.0	85.7	69.0	92.1
StructViT-B-4-2	K400	32×3×1	784	61.1	87.7	71.1	92.7
StructViT-B-4-1	K400	32×3×1	1963	**61.3**	**87.8**	**71.5**	**93.1**

(a) **Something-Something V1 & V2**

model	top-1
SlowFast-R101 [20]	77.6
TimeSformer [4]	75.0
TimeSformer-HR [4]	78.0
SViT-DD [2]	79.8
TimeSformer-L [4]	81.0
TQN [79]	81.8
RSANet-R50 [32]	84.2
UniFormer-B* [38]	87.4
ORViT† [25]	88.0
StructViT-B-4-2	87.8
StructViT-B-4-1	**88.3**

(b) **Diving-48**

model	Gym288	Gym99
TRN [82]	33.1	68.7
I3D [8]	27.9	63.2
TSM [43]	34.8	70.6
$\text{TSM}_{\text{Two-stream}}$ [43]	46.5	81.2
RSANet-R50 [32]	50.9	86.4
UniFormer-B* [38]	53.5	88.9
StructViT-B-4-2	53.8	89.3
StructViT-B-4-1	**54.2**	**89.5**

(c) **FineGym**

Table 5. **Comparisons to the state-of-the-art methods on three motion-centric video classification benchmarks.** Our StructViT achieves new state-of-the-art on all the benchmarks. For FineGym, we measure averaged per-class accuracy while top-k accuracy is measured for Something-Something and Diving-48. *Reproduced by our experimental setup. †Trained with additional bbox annotations.

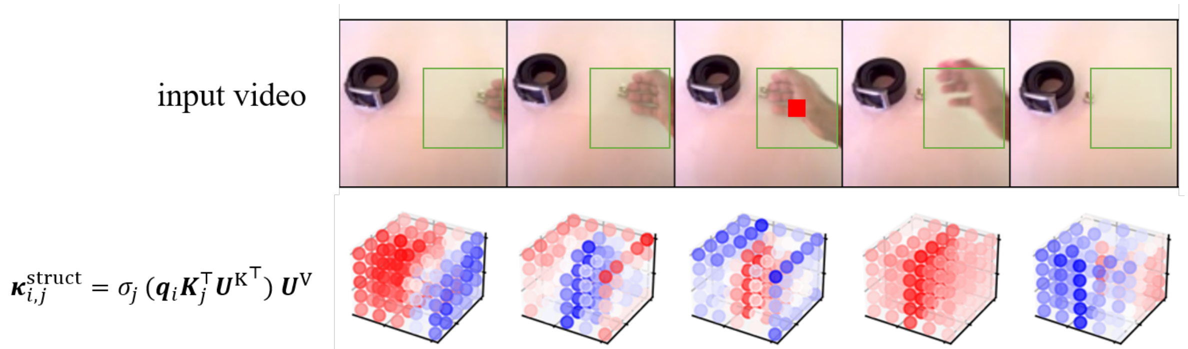


Figure 3. **Visualization of dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$ in StructSA on Something-Something-V1.** The top row shows the input frames that contain the input spatiotemporal local context (indicated by green boxes) used in the dynamic kernel computation. The bottom row presents the resulting dynamic kernels $\boldsymbol{\kappa}_{i,j}^{\text{struct}}$ for a StructSA head when $i = j$. Note that the computed dynamic kernels are computed with self-similarity map ($i = j$) to illustrate its effectiveness in capturing motions in videos. We use StructViT-S-4-1 with $M = 5 \times 5 \times 5$.

5. Conclusion

We have introduced a novel self-attention mechanism, StructSA, that exploits rich structural patterns of query-key correlation for visual representation learning. StructSA leverages spatial (and temporal) structures of local correlations and aggregates chunks of local features globally across entire locations. Structural Vision Transformer (StructViT) using StructSA as the main attention module achieves state-of-the-art results on both image and video classification benchmarks. We believe leveraging structural patterns of correlation in attention will also benefit other tasks in computer vision and natural language processing. We leave this for future work.

Acknowledgements This work was supported by the IITP grants (No. 2022-0-00290: Visual Intelligence for space-time understanding and generation (45%), No. 2022-0-00264: Comprehensive video understanding and generation with knowledge-based deep logic (50%), No. 2019-0-01906: AI graduate school program at POSTECH (5%)) funded by Ministry of Science and ICT, Korea.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 2, 7, 8

[2] Elad Ben-Avraham, Roei Herzig, Karttikeya Mangalam, Amir Bar, Anna Rohrbach, Leonid Karlinsky, Trevor Darrell, and Amir Globerson. Bringing image scene structure to video via frame-clip consistency of object tokens. *arXiv preprint arXiv:2206.06346*, 2022. 8

[3] Chiraz BenAbdelkader, Ross G Cutler, and Larry S Davis. Gait recognition using image self-similarity. *EURASIP Journal on Advances in Signal Processing*, 2004(4):1–14, 2004. 1

[4] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 7, 8

[5] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *NeurIPS*, 33:1877–1901, 2020. 2

[6] Adrian Bulat, Juan Manuel Perez Rua, Swathikiran Sudhakaran, Brais Martinez, and Georgios Tzimiropoulos. Space-time mixing attention for video transformer. *NeurIPS*, 34:19594–19607, 2021. 7, 8

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *CVPR*, 2017. 8

[9] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 2, 5, 6

[10] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019. 2

[11] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 2

[12] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *NeurIPS*, 34:3965–3977, 2021. 2, 6

[13] 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. 4, 5

[14] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2

[15] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, pages 12124–12134, 2022. 6

[16] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *ICCV*, 2015. 2

[17] 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. *arXiv preprint arXiv:2010.11929*, 2020. 2

[18] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021. 1, 2, 3, 5, 6, 7, 8

[19] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *CVPR*, 2020. 7

[20] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, 2019. 7, 8

[21] Robert Geirhos, Kantharaju Narayanappa, Benjamin Mitzkus, Tizian Thieringer, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Partial success in closing the gap between human and machine vision. In *NeurIPS*, 2021. 1

[22] 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. 4, 5

[23] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, pages 12175–12185, 2022. 2, 3

[24] Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *ICCV*, 2017. 2

[25] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *CVPR*, pages 3148–3159, 2022. 7, 8

[26] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 3

[27] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3464–3473, 2019. 2

[28] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016. 6

[29] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *NeurIPS*, 34:18590–18602, 2021. 5, 6, 7

[30] Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *ICCV*, pages 8822–8833, 2021. 1, 2

[31] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 4

[32] Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. Relational self-attention: What's missing in attention for video understanding. *NeurIPS*, 34:8046–8059, 2021. 1, 2, 3, 7, 8

[33] Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *CVPR*, 2017. 1, 2

[34] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *CVPR*, pages 16020–16030, 2021. 7

[35] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Motionsqueeze: Neural motion feature learning for video understanding. *arXiv preprint arXiv:2007.09933*, 2020. 2, 8

[36] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for action recognition. *arXiv preprint arXiv:2102.07092*, 2021. 1, 2, 3, 8

[37] Kunchang Li, Xianhang Li, Yali Wang, Jun Wang, and Yu Qiao. {CT}-net: Channel tensorization network for video classification. In *ICPR*, 2021. 8

[38] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *arXiv preprint arXiv:2201.09450*, 2022. 2, 5, 6, 7, 8

[39] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *ECCV*, 2018. 4, 7

[40] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *CVPR*, 2020. 8

[41] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, pages 4804–4814, 2022. 1, 2, 3, 5, 6, 7, 8

[42] Yuxuan Liang, Pan Zhou, Roger Zimmermann, and Shuicheng Yan. Dualformer: Local-global stratified transformer for efficient video recognition. *arXiv preprint arXiv:2112.04674*, 2021. 1, 7

[43] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *ICCV*, 2019. 8

[44] Xingyu Liu, Joon-Young Lee, and Hailin Jin. Learning video representations from correspondence proposals. In *CVPR*, 2019. 2

[45] 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*, pages 10012–10022, 2021. 2, 6

[46] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 6

[47] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *CVPR*, pages 3202–3211, 2022. 7, 8

[48] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[49] Farzaneh Mahdisoltani, Guillaume Berger, Waseem Gharbieh, David Fleet, and Roland Memisevic. On the effectiveness of task granularity for transfer learning. *arXiv preprint arXiv:1804.09235*, 2018. 4

[50] Juhong Min and Minsu Cho. Convolutional hough matching networks. In *CVPR*, pages 2940–2950, 2021. 2

[51] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *NeurIPS*, 32, 2019. 4

[52] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. *NeurIPS*, 34:12493–12506, 2021. 7, 8

[53] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *NeurIPS*, 34:12116–12128, 2021. 2

[54] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *NeurIPS*, 2019. 2

[55] Paul Hongsuck Seo, Jongmin Lee, Deunsol Jung, Bohyung Han, and Minsu Cho. Attentive semantic alignment with offset-aware correlation kernels. In *ECCV*, 2018. 2

[56] Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *CVPR*, 2020. 4, 7

[57] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *CVPR*, 2007. 1, 2, 3

[58] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, pages 7262–7272, 2021. 2

[59] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, pages 7464–7473, 2019. 2

[60] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010. 7

[61] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. International Conference on Machine Learning*, 2019. 6

[62] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *Proc. International Conference on Machine Learning*, pages 10096–10106. PMLR, 2021. 6

[63] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proc. International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 2, 4, 5, 6
[64] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *ICCV*, 2019. 7
[65] Shikhar Tuli, Ishita Dasgupta, Erin Grant, and Thomas L Griffiths. Are convolutional neural networks or transformers more like human vision? *arXiv preprint arXiv:2105.07197*, 2021. 1
[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2
[67] Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *CVPR*, 2020. 2
[68] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 5
[69] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *CVPR*, pages 1895–1904, 2021. 8
[70] 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*, pages 568–578, 2021. 1, 2, 3, 5, 6
[71] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018. 5
[72] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, pages 22–31, 2021. 1, 2, 3, 5, 6
[73] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *CVPR*, pages 3333–3343, 2022. 7
[74] Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *NeurIPS*, 2019. 2
[75] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021. 6
[76] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 5
[77] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE TPAMI*, 2022. 5
[78] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 4
[79] Chuhan Zhang, Ankush Gupta, and Andrew Zisserman. Temporal query networks for fine-grained video understanding. In *CVPR*, pages 4486–4496, 2021. 8
[80] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4
[81] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, pages 13001–13008, 2020. 4
[82] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 8