# STRUCTVIT: LEARNING CORRELATION STRUCTURES FOR VISION TRANSFORMERS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We introduce the structural self-attention (StructSA) mechanism that leverages structural patterns of query-key correlation for visual representation learning. StructSA generates attention by recognizing space-time structures of correlations and performs long-range interactions across entire locations, effectively capturing structural patterns, *e.g.*, spatial layouts, motion, or inter-object relations in images and videos. 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 others in space and time is a crucial cue for visual understanding, *e.g.*, recognizing object layouts in an image or human interactions in a video. In computer vision, such meta-patterns are effectively captured by the structure of correlations or similarities across visual elements in different positions (BenAbdelkader et al., 2004; Shechtman & Irani, 2007). A correlation structure of an image reveals spatial layouts of similar patterns (Kim et al., 2017; Kang et al., 2021) and that of a video provides bi-directional motion likelihoods (Kwon et al., 2021; Kim et al., 2021). The ability to recognize those structural patterns allows to better generalize against challenging appearance variations and domain shifts (Geirhos et al., 2021; Tuli et al., 2021).

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. We first show that vision transformer networks with self-attention and its convolutional variant (Wu et al., 2021; Wang et al., 2021b; Fan et al., 2021; Liang et al., 2021) are both limited in leveraging structural patterns. The standard self-attention mechanism uses raw query-key correlations individually and ignores their structure, whereas its variant with convolutional projection turns out to have only limited access to the structure of query-key correlations. In contrast, the proposed StructSA recognizes diverse structural patterns from the correlations maps between the query and local chunks of keys. This is achieved by extending the convolution projections used with self-attention under our new interpretation. We add a new structure dimension to the convolution projection allowing to capture multiple patterns from a single correlation map. We then develop the structural vision transformer (StructViT) that adopts StructSA as a main neural block, and perform extensive sets of experiments on both image and video classification tasks, showing the effectiveness of learning structural patterns for visual representations. Our main contributions are summarized as follows:

- We provide a new interpretation on the self-attention with convolutional projections and show its potential to learn structural information of the correlations.
- We introduce structural self-attention (StructSA) that learns correlation structures for visual representations with the Vision Transformer (StructViT).
- The proposed StructViT achieves new state-of-the-art results on ImageNet-1K, Kinetics-400, Something-Something V1&V2, Diving-48, and FineGym.

# 2 RELATED WORK

## 2.1 TRANSFORMER NETWORKS IN VISION

Since transformer networks (Vaswani et al., 2017) showed remarkable success in natural language processing (Devlin et al., 2018; Brown et al., 2020), they have widely been adopted in various computer vision tasks as an alternative to CNNs (Sun et al., 2019; Dosovitskiy et al., 2020; Arnab et al., 2021; Carion et al., 2020; Strudel et al., 2021). 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 (Dosovitskiy et al., 2020; Raghu et al., 2021). This incentivized several methods to inherit the convolutional inductive biases via knowledge distillation (Touvron et al., 2021), local self-attention (Hu et al., 2019; Ramachandran et al., 2019; Liu et al., 2021), and architectural fusion (Dai et al., 2021; Li et al., 2022a; Guo et al., 2022; Chu et al., 2021; Wang et al., 2021b; Wu et al., 2021; Fan et al., 2021). Methods using a convolutional projection instead of a linear projection (Wang et al., 2021b; Wu et al., 2021; Fan et al., 2021) show that the convolutional projection effectively encodes position information achieving remarkable results for both image and video representations. In this work, we provide a new interpretation for the self-attention with the convolutional projections and show its missing capability to learn structural patterns in the correlation. We then introduce a novel self-attention mechanism that leverages such structural information for visual representation learning.

## 2.2 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 (Shechtman & Irani, 2007; Kim et al., 2017; Kang et al., 2021). Spatial cross-correlation between different images is often used for establishing semantic correspondences capturing structural similarities (Han et al., 2017; Seo et al., 2018; Min & Cho, 2021). In the video domain, several methods exploit the structure of spatial cross-correlations between consecutive frames to estimate optical flow (Dosovitskiy et al., 2015; Yang & Ramanan, 2019) or to learn motion features for action recognition (Wang et al., 2020; Kwon et al., 2020). Kwon et al. (2021) propose spatio-temporal self-correlations for learning bi-directional motion features and Kim et al. (2021) 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

The query-key correlations of self-attention Vaswani et al. (2017) capture geometric structures that can represent spatial layouts (Shechtman & Irani, 2007) or motions (Kim et al., 2021) of objects in images or videos. However, in vanilla self-attention, individual correlation values are directly used as weights for feature aggregation disregarding the structure within the *correlation map*. In this work, we aim to develop a novel self-attention process that utilizes these geometric structures. To this end, we first provide a novel interpretation of the self-attention with convolutional projections (ConvSA) where ConvSA generates the attention scores (Kim et al., 2021) using local geometric structures in a correlation map. We then extend such a process to encode the local geometric structures into a vector representation instead of a scalar, and call this extended attention process *structural self-attention*. Finally, we propose our model architectures called Structural vision transformers (StructViT), which use our structural self-attention as a basic building block.

### 3.1 PRELIMINARY: SELF-ATTENTION AND CONVOLUTIONAL PROJECTION

Self-attention (SA) Vaswani et al. (2017) is the neural block for building modern transformer networks (Dosovitskiy et al., 2020; Touvron et al., 2021; Arnab et al., 2021). 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

$$\boldsymbol{y}_i = \sigma\left(\boldsymbol{q}_i \boldsymbol{K}^{\mathsf{T}}\right) \boldsymbol{V} \in \mathbb{R}^{1 \times C}, \tag{1}$$

where $\sigma$ is a softmax function and $\boldsymbol{q}_i = \boldsymbol{x}_i \boldsymbol{W}^{\mathrm{Q}}$, $\boldsymbol{K} = \boldsymbol{X}\boldsymbol{W}^{\mathrm{K}}$, $\boldsymbol{V} = \boldsymbol{X}\boldsymbol{W}^{\mathrm{V}}$ are query, keys and values linearly projected from the inputs by projection matrices $\boldsymbol{W}^{Q}, \boldsymbol{W}^{K}, \boldsymbol{W}^{V} \in \mathbb{R}^{C \times C}$, respectively. While computing correlations $\boldsymbol{q}_i \boldsymbol{K}^{\mathsf{T}}$, this self-attention uses them individually and ignores their structure; this is easily seen by its invariance with respect to permutation of input $\boldsymbol{X}$.

Despite its success in various tasks (Carion et al., 2020; Dosovitskiy et al., 2020; Arnab et al., 2021; Strudel et al., 2021), pure self-attention transformers are known to suffer from data-hungry and unstable training (Raghu et al., 2021; Chen et al., 2021). To tackle this issue, recent research (Wu et al., 2021; Wang et al., 2021b; Fan et al., 2021; Li et al., 2022b; Guo et al., 2022) introduces inductive biases to transformers by adopting self-attention with convolutional projections (ConvSA). Different from SA, ConvSA projects keys and values using a convolution operation over the input feature map $\boldsymbol{X}$:

$$\boldsymbol{K}^{\mathrm{conv}} = [\boldsymbol{k}_1^{\mathrm{conv}}, \cdots, \boldsymbol{k}_N^{\mathrm{conv}}] = \texttt{conv}(\boldsymbol{X}, \mathbf{W}^{\mathrm{K}}) \in \mathbb{R}^{N \times C}, \tag{2}$$

$$\boldsymbol{V}^{\mathrm{conv}} = [\boldsymbol{v}_1^{\mathrm{conv}}, \cdots, \boldsymbol{v}_N^{\mathrm{conv}}] = \texttt{conv}(\boldsymbol{X}, \mathbf{W}^{\mathrm{V}}) \in \mathbb{R}^{N \times C}, \tag{3}$$

where $\texttt{conv}$ 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. Here, we use 1-dimensional convolution for notational simplicity but the operation can be simply extended to convolutions with a larger dimensionality.

In most previous methods, ConvSA is implemented with a channel-wise separable convolution (Howard et al., 2017), which consists of two factorized convolution operations, *i.e.*, point-wise and channel-wise convolutions (Wu et al., 2021; Wang et al., 2021b; Fan et al., 2021; Li et al., 2022b; Guo et al., 2022). In this case, each key $\boldsymbol{k}_i^{\mathrm{conv}}$ and value $\boldsymbol{v}_i^{\mathrm{conv}}$ is computed from a local context $\boldsymbol{X}_i = \boldsymbol{X}_{i-\lfloor \frac{M}{2} \rfloor : i+\lfloor \frac{M}{2} \rfloor} \in \mathbb{R}^{M \times C}$ by

$$\boldsymbol{k}_i^{\mathrm{conv}} = \boldsymbol{u}^{\mathrm{K}} \boldsymbol{X}_i \boldsymbol{W}^{\mathrm{K}} = \boldsymbol{u}^{\mathrm{K}} \boldsymbol{K}_i \in \mathbb{R}^{1 \times C}, \tag{4}$$

$$\boldsymbol{v}_i^{\mathrm{conv}} = \boldsymbol{u}^{\mathrm{V}} \boldsymbol{X}_i \boldsymbol{W}^{\mathrm{V}} = \boldsymbol{u}^{\mathrm{V}} \boldsymbol{V}_i \in \mathbb{R}^{1 \times C}, \tag{5}$$

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 $\boldsymbol{u}^{\mathrm{K}}, \boldsymbol{u}^{\mathrm{V}} \in \mathbb{R}^{1 \times M}$ 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 Appendix A.

### 3.2 ANALYSIS OF CONVSA

In this section, we provide a novel interpretation of ConvSA with a lens of a dynamic kernel (Kim et al., 2021) and show its potential for learning structures from query-key correlations. From Eq. (1) combined with Eq. (4) and (5), a transformed output $\boldsymbol{y}_i$ in ConvSA is obtained by

$$\boldsymbol{y}_i = \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i \boldsymbol{k}_j^{\mathrm{conv}\mathsf{T}}\right) \boldsymbol{v}_j^{\mathrm{conv}} = \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}} \boldsymbol{u}^{\mathrm{K}\mathsf{T}}\right) \boldsymbol{u}^{\mathrm{V}} \boldsymbol{V}_j = \sum_{j=1}^{N} \boldsymbol{\kappa}_{i,j}^{\mathrm{conv}} \boldsymbol{V}_j, \tag{6}$$

where $\sigma_j$ is $j$th entry of the softmax over $N$ tokens. This reveals that an attention score $\sigma_j(\boldsymbol{q}_i \boldsymbol{k}_j^{\mathrm{conv}\mathsf{T}})$ is computed by projecting a local correlation map $\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}} \in \mathbb{R}^{1 \times M}$ 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 $\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}}$ represents a structural pattern, we can interpret that $\boldsymbol{u}^{\mathrm{K}}$ acts as a *pattern detector* that extracts a specific correlation pattern from $\boldsymbol{q}_i \boldsymbol{K}_j^{\mathsf{T}}$, 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.

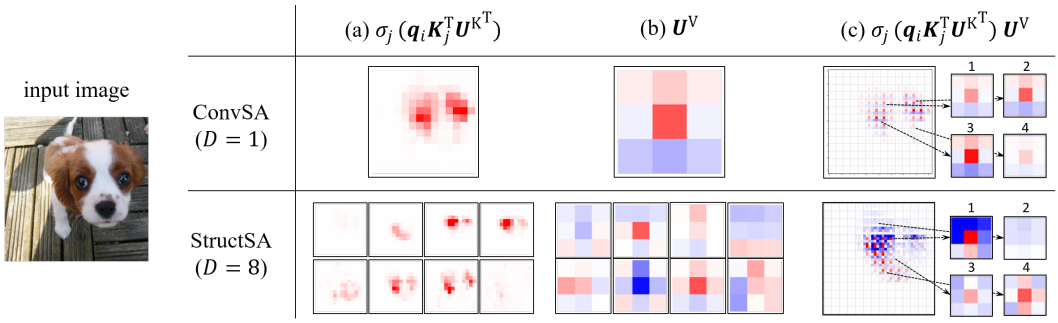


Figure 1: **Visualization of ConvSA and StructSA on ImageNet-1K.** The query location $i$ and the kernel size $M$ are set to the center location and $3 \times 3$. Given the left image as input, we compare ConvSA ($D = 1$) and StructSA ($D = 8$) in terms of (a) $D$ attention maps $\sigma_{jD}(\boldsymbol{q}_i\boldsymbol{K}_j^{\mathsf{T}}\boldsymbol{U}^{\mathrm{K}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

**Limitation of ConvSA.** 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.

### 3.3 Structural Self-Attention

We propose a novel self-attention mechanism, named *structural self-attention* (StructSA). The core idea of StructSA is to encode correlations $\boldsymbol{q}_i\boldsymbol{K}_j^{\mathsf{T}}$ into a $D$-dimensional vector, rather than a scalar, that recognizes richer structural patterns to produce a dynamic kernel. Note that we call this new vector dimension with $D$ a structure dimension. To compute StructSA, we extend the pattern detector $\boldsymbol{u}^{\mathrm{K}}$ and the context aggregator $\boldsymbol{u}^{\mathrm{V}}$ to matrices $\boldsymbol{U}^{\mathrm{K}}, \boldsymbol{U}^{\mathrm{V}} \in \mathbb{R}^{D\times M}$ resulting in

$$\boldsymbol{K}_i^{\mathrm{struct}} = \boldsymbol{U}^{\mathrm{K}}\boldsymbol{K}_i \in \mathbb{R}^{D\times C}, \tag{7}$$

$$\boldsymbol{V}_i^{\mathrm{struct}} = \boldsymbol{U}^{\mathrm{V}}\boldsymbol{V}_i \in \mathbb{R}^{D\times C}. \tag{8}$$

Plugging Eq. (7) and (8) into $\boldsymbol{k}_j^{\mathrm{conv}}$ and $\boldsymbol{v}_j^{\mathrm{conv}}$ of Eq. (6), the output $y_i$ of StructSA is computed by

$$\boldsymbol{y}_i = \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i\boldsymbol{K}_j^{\mathrm{struct}^{\mathsf{T}}}\right)\boldsymbol{V}_j^{\mathrm{struct}} = \sum_{j=1}^{N} \sigma_j\left(\boldsymbol{q}_i\boldsymbol{K}_j^{\mathsf{T}}\boldsymbol{U}^{\mathrm{K}^{\mathsf{T}}}\right)\boldsymbol{U}^{\mathrm{V}}\boldsymbol{V}_j = \sum_{j=1}^{N} \boldsymbol{\kappa}_{i,j}^{\mathrm{struct}}\boldsymbol{V}_j \tag{9}$$

where the softmax function $\sigma_j$ returns a $D$-dimensional output for $j$th location. Note that this softmax is taken over all $ND$ entries in the input matrix as we observe it is empirically more stable compared to $D$ individual softmax operations over $N$ entries.

Compared to ConvSA where only a single structural pattern is learned by the pattern extractor $\boldsymbol{u}^{\mathrm{K}}$, 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 $j$. In the case of $i = j$, the correlation map corresponds to the local self-similarity (Shechtman & Irani, 2007) that is known to capture geometric structures such as spatial layout (Shechtman & Irani, 2007) or spatio-temporal motion (Kwon et al., 2021), meaning that when $i = j$, the pattern detector $\boldsymbol{U}^{\mathrm{K}}$ reduces to capturing a self-similarity pattern as in (Kim et al., 2021).

We illustrate this dynamic kernel computation process with an example input image from ImageNet-1K (Deng et al., 2009) in Figure 1. We compare ConvSA and StructSA ($D = 8$) show how structural patterns are used in these methods. Given a query-key correlation map, ConvSA generates a single attention map shown in column (a). Then these scores are combined with the context aggregator $\boldsymbol{u}^{\mathrm{V}}$

(column (b)), that represents a single aggregation pattern. As the result, local features are aggregated with the identical pattern in $\boldsymbol{u}^{\mathrm{V}}$ for every location and the only differences are their scales as shown in column (c). In contrast, StructSA generates multiple attention maps using $D$ pattern detectors each capturing different structures in the query-key correlation maps (column (a)), and combines them with different context aggregators shown in column (b) resulting in diverse aggregation patterns for different locations $j$ as illustrated in column (c).

### 3.4 Structural Vision Transformer (StructViT)

Finally, we introduce Structural Vision Transformers (StructViT) that use our StructSA as the basic building block; our StructViT has unified network configurations for both image and video classification tasks. For image input, we simply set the temporal sizes in both input shape and operation parameters (*e.g.*, convolution kernel and stride) are set to 1. We extend the state-of-the-art method Uniformer (Li et al., 2022a) by replacing all vanilla self-attention with StructSA. Note that when we employ StructSA, we use multi-head configurations and do not share weights across channels for the channel-wise convolutions. Our network takes a video clip or an image $\tilde{\boldsymbol{X}} \in \mathbb{R}^{T\times\ H\times W\times 3}$ as input, where $T$, $H$, and $W$ represent spatiotemporal resolutions of the video clip ($T = 1$ for an image input). We tokenize an input video clip into overlapping 3D tublets of size $3 \times 4 \times 4 \times 3$ with a stride of $2 \times 4 \times 4$ (non-overlapping 2D patches of size $4 \times 4 \times 3$ for image) to feed them into our network. Our network comprises four stages, each of which has multiple neural blocks, and it leverages a hierarchical design to produce feature maps with decreasing resolutions and increasing number of channels from early to late stages following (Li et al., 2022a). For the first two stages, each block consists of a conditional positional encoding layer, a convolutional layer, and an MLP, whereas the convolutional layer is replaced with a StructSA layer in the blocks in the last two stages. For the detailed network design for each block at different stages, please refer to (Li et al., 2022a). We build three different StructViT architectures where the number of channels and blocks for each stage are defined as follows:

- StructViT-S: # channels =$\{$64,128,320,512$\}$, # blocks = $\{$3,4,8,3$\}$
- StructViT-B: # channels =$\{$64,128,320,512$\}$, # blocks = $\{$5,8,20,7$\}$
- StructViT-L: # channels =$\{$128,192,448,640$\}$, # blocks = $\{$5,10,24,7$\}$.

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. 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.

## 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 Image classification

#### 4.1.1 Experimental setup

We conduct image classification experiments on ImageNet-1K (Deng et al., 2009). We follow the training strategy of DeiT (Touvron et al., 2021) adopting random clipping, random horizontal flipping, mixup (Zhang et al., 2017), cutmix (Yun et al., 2019), random erasing (Zhong et al., 2020) and label-smoothing (Müller et al., 2019) to augment the input images for training. We train all models from scratch for 300 epochs using AdamW optimizer (Loshchilov & Hutter, 2017) 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. We also use stochastic depth (Huang et al., 2016) with the probability of 0.1/0.3/0.4 for StructViT-S/B/L, respectively. We use 8 NVIDIA A100 GPUs for training StructViT-S/B and 16 GPUs for StructViT-L. Our model should be comparable to

Uniformer Li et al. (2022a) in the same sizes as our model configurations are based on Uniformer's; adding the structure dimension $D$ in StructSA introduces few additional parameters.

#### 4.1.2 RESULTS

In Table 1, we compare StructViT with other state-of-the-art CNNs, ViTs, and hybrid models. The results show that StructViT outperforms other methods in all sizes. Compared to EfficientNets (Tan & Le, 2019; 2021) that are obtained by extensive architecture search, our models show comparable or even better performances in both base and large configurations, requiring much less amount of computational cost. Compared to our baseline, Uniformers, StructViTs consistently bring gains in top-1 accuracy regardless of its size, demonstrating the benefits of learning geometric structures in image understanding. 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) to match its FLOPs to that of the baselines; We still observe some gain with the base model (B-4-2) without additional FLOPs while the small model (S-4-2) shows comparable performance.

Table 1: **Comparisons to the state-of-the-art methods on ImageNet-1K.** *Trained with token labeling (Jiang et al., 2021).

| method | #param (M) | FLOPs (G) | IN1K Top-1 |
|---|---|---|---|
| RegNetY-4G (Radosavovic et al., 2020) | 21 | 4.0 | 80.0 |
| EffcientNet-B4 (Tan & Le, 2019) | 19 | 4.2 | 82.9 |
| EffcientNet-B5 (Tan & Le, 2019) | 30 | 9.9 | 83.6 |
| DeiT-S (Touvron et al., 2021) | 22 | 4.6 | 79.9 |
| PVT-S (Wang et al., 2021b) | 25 | 3.8 | 79.8 |
| T2T-14 (Yuan et al., 2021) | 22 | 5.2 | 80.7 |
| Swin-T (Liu et al., 2021) | 29 | 4.5 | 81.3 |
| Focal-T (Yang et al., 2021) | 29 | 4.9 | 82.2 |
| CSwin-T (Dong et al., 2022) | 23 | 4.3 | 82.7 |
| CvT-13 (Wu et al., 2021) | 20 | 4.5 | 81.6 |
| CoAtNet-0 (Dai et al., 2021) | 25 | 4.2 | 81.6 |
| LV-ViT-S Jiang et al. (2021) | 26 | 6.6 | 83.3 |
| Uniformer-S (Li et al., 2022a) | 22 | 3.6 | 82.9 |
| StructViT-S-4-2 (ours) | 23 | 3.6 | 82.9 |
| StructViT-S-4-1 (ours) | 23 | 4.3 | 83.2 |
| StructViT-S-8-1 (ours) | 24 | 5.4 | **83.3** |
| RegNetY-8G (Radosavovic et al., 2020) | 39 | 8.0 | 81.7 |
| EffcientNet-B7 (Tan & Le, 2019) | 66 | 39.2 | 84.3 |
| PVT-L (Wang et al., 2021b) | 61 | 9.8 | 81.7 |
| T2T-24 (Yuan et al., 2021) | 64 | 13.2 | 82.2 |
| Swin-S (Liu et al., 2021) | 50 | 8.7 | 83.0 |
| Focal-S (Yang et al., 2021) | 51 | 9.1 | 83.5 |
| CSwin-S (Dong et al., 2022) | 35 | 6.9 | 83.6 |
| CvT-21 (Wu et al., 2021) | 32 | 7.1 | 82.5 |
| Container (Gao et al., 2021) | 22 | 8.1 | 82.7 |
| CoAtNet-1 (Dai et al., 2021) | 42 | 8.4 | 83.3 |
| LV-ViT-M (Jiang et al., 2021) | 56 | 16.0 | 84.1 |
| Uniformer-B (Li et al., 2022a) | 50 | 8.3 | 83.8 |
| StructViT-B-4-2 (ours) | 51 | 8.3 | 84.0 |
| StructViT-B-4-1 (ours) | 51 | 9.9 | 84.2 |
| StructViT-B-8-1 (ours) | 52 | 12.0 | **84.3** |
| RegNetY-16G (Radosavovic et al., 2020) | 84 | 16.0 | 82.9 |
| EfficientNetV2-L (Tan & Le, 2021) | 121 | 52 | 85.7 |
| Swin-B (Liu et al., 2021) | 88 | 15.4 | 83.3 |
| Focal-B (Yang et al., 2021) | 90 | 16.0 | 83.8 |
| CSwin-B (Dong et al., 2022) | 78 | 15.0 | 84.2 |
| CoAtNet-3 (Dai et al., 2021) | 168 | 34.7 | 84.5 |
| LV-ViT-L ↑288 (Jiang et al., 2021) | 150 | 59.0 | 85.3 |
| VOLO-D3 (Yuan et al., 2022) | 86 | 20.6 | 85.4 |
| Uniformer-L* (Li et al., 2022a) | 100 | 12.6 | 85.6 |
| StructViT-L-4-1* (ours) | 103 | 15.4 | **86.0** |

### 4.2 VIDEO CLASSIFICATION

#### 4.2.1 EXPERIMENTAL SETUP

We conduct experiments for video classification on Kinetics-400 (Kay et al., 2017), Something-Something V1&V2 (Goyal et al., 2017; Mahdisoltani et al., 2018), Diving48 (Li et al., 2018), and FineGym (Shao et al., 2020). For training, we follow the strategy in MViT (Fan et al., 2021). For Kinetics, we sample 16 or 32 frames using the dense sampling strategy (Wang et al., 2018). We use random cropping and horizontal flipping for data augmentation. We temporally inflate the model weights pretrained on ImageNet-1K and finetune it for 110 epochs including 10 warm-up epochs. We use AdamW (Loshchilov & Hutter, 2017) optimizer with cosine learning rate schedule. We set the total batch size, learning rate, weight decay, and stochastic depth rate to 64, 2e-4, 0.05, and 0.1, respectively. For Something-Something V1&V2, Diving48, and FineGym, we utilize the segment-based sampling strategy (Wang et al., 2016). We only use random cropping for data augmentation. We initialize the model with the weights pretrained on Kinetics-400 and finetune the model for 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 by sampling different temporal indices for each

Table 2: **Comparisons to the state-of-the-art methods on Kinetics-400.**

| method | pretrain | #frame×#crop×#clip | FLOPs (G) | K400 top-1 | K400 top-5 |
|---|---|---|---|---|---|
| TDN$_{EN}$(Wang et al., 2021a) | IN-1K | (8+16)×3×10 | 5940 | 79.4 | 94.4 |
| SlowFast(Feichtenhofer et al., 2019) | - | 8×3×10 | 3180 | 77.9 | 93.2 |
| SlowFast+NL(Feichtenhofer et al., 2019) | - | 16×3×10 | 7020 | 79.8 | 93.9 |
| ip-CSN(Tran et al., 2019) | Sports1M | 32×3×10 | 3270 | 79.2 | 93.8 |
| CorrNet(Wang et al., 2020) | Sports1M | 32×3×10 | 6720 | 81.0 | - |
| X3D-M(Feichtenhofer, 2020) | - | 16×3×10 | 186 | 76.0 | 92.3 |
| X3D-XL(Feichtenhofer, 2020) | - | 16×3×10 | 1452 | 79.1 | 93.9 |
| MoViNet-A5(Kondratyuk et al., 2021) | - | 120×1×1 | 281 | 80.9 | 94.9 |
| MoViNet-A6(Kondratyuk et al., 2021) | - | 120×1×1 | 386 | 81.5 | 95.3 |
| ViT-B-VTN (Neimark et al., 2021) | IN-21K | 250×1×1 | 3992 | 78.6 | 93.7 |
| TimeSformer-HR(Bertasius et al., 2021) | IN-21K | 16×3×1 | 5109 | 79.7 | 94.4 |
| TimeSformer-L(Bertasius et al., 2021) | IN-21K | 96×3×1 | 7140 | 80.7 | 94.7 |
| X-ViT(Bulat et al., 2021) | IN-21K | 16×3×1 | 850 | 80.2 | 94.7 |
| Mformer-HR(Patrick et al., 2021) | IN-21K | 16×3×10 | 28764 | 81.1 | 95.2 |
| ViViT-L(Arnab et al., 2021) | IN-21K | 16×3×4 | 17352 | 80.6 | 94.7 |
| Swin-B(Liu et al., 2022) | IN-1K | 32×3×4 | 3384 | 80.6 | 94.6 |
| MTV-B (Yan et al., 2022) | IN-21K | 32×3×4 | 4790 | 81.8 | 95.0 |
| MViT-B,16×4(Fan et al., 2021) | - | 16×1×5 | 353 | 78.4 | 93.5 |
| MViT-B,32×3(Fan et al., 2021) | - | 32×1×5 | 850 | 80.2 | 94.4 |
| Dualformer-S (Liang et al., 2021) | IN-1K | 32×1×4 | 636 | 80.6 | 94.9 |
| Dualformer-B (Liang et al., 2021) | IN-1K | 32×1×4 | 1072 | 81.1 | 95.0 |
| Uniformer-S (Li et al., 2022a) | IN-1K | 16×1×4 | 167 | 80.8 | 94.7 |
| Uniformer-B (Li et al., 2022a) | IN-1K | 32×1×4 | 1036 | 82.9 | 95.4 |
| StructViT-S-4-2 (ours) | IN-1K | 16×1×4 | 169 | 81.1 | 95.5 |
| StructViT-S-4-1 (ours) | IN-1K | 16×1×4 | 327 | 81.4 | 95.7 |
| StructViT-S-8-1 (ours) | IN-1K | 16×1×4 | 541 | 81.6 | 95.8 |
| StructViT-B-4-2 (ours) | IN-1K | 32×1×4 | 1045 | 83.1 | 95.5 |
| StructViT-B-4-1 (ours) | IN-1K | 32×1×4 | 2658 | 83.3 | 95.6 |
| StructViT-B-4-1 (ours) | IN-1K | 32×3×4 | 7974 | **83.4** | **95.8** |

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.

#### 4.2.2 Results on Kinetics-400

Table 2 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 the 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 (Kondratyuk et al., 2021) that 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 2) shows 1.6% absolute accuracy gain 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 and B-4-1) show 0.5% to 0.8% absolute 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 some absolute gains ranging from 0.2% to 0.3%.

#### 4.2.3 Results on Something-Something, Diving-48 and FineGym

Table 3a summarizes the results on Something-Something V1&V2. We observe the same trends as on Kinetics-400. Our full model sets a new state-of-the-art performances on both V1 and V2 while the gains are slightly smaller when tested with matching FLOPs compared to the Uniformer models.

Table 3b and Table 3c show the results on Diving-48 (Li et al., 2018) and FineGym (Shao et al., 2020). Our model sets new state-of-the-art performances with large margins (4.1% on Diving-48;

Table 3: **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. *Trained with additional bounding box annotations.

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

| method | pretrain | #frame× #crop×#clip | FLOPs (G) | Something V1 top-1 | Something V1 top-5 | Something V2 top-1 | Something V2 top-5 |
|---|---|---|---|---|---|---|---|
| TSN(Wang et al., 2016) | IN-1K | 16×1×1 | 66 | 19.9 | 47.3 | 30.0 | 60.5 |
| TSM(Lin et al., 2019) | IN-1K | 16×1×1 | 66 | 47.2 | 77.1 | - | - |
| GST(Luo & Yuille, 2019) | IN-1K | 16×1×1 | 59 | 48.6 | 77.9 | 62.6 | 87.9 |
| TEA(Li et al., 2020) | IN-1K | 16×1×1 | 70 | 51.9 | 80.3 | - | - |
| MSNet(Kwon et al., 2020) | IN-1K | 16×1×1 | 101 | 52.1 | 82.3 | 64.7 | 89.4 |
| CT-Net(Li et al., 2021) | IN-1K | 16×1×1 | 75 | 52.5 | 80.9 | 64.5 | 89.3 |
| TDN(Wang et al., 2021a) | IN-1K | 16×1×1 | 72 | 53.9 | 82.1 | 65.3 | 89.5 |
| SELFYNet (Kwon et al., 2021) | IN-1K | 16×1×1 | 77 | 54.3 | 82.9 | 65.7 | 89.8 |
| RSANet (Kim et al., 2021) | IN-1K | 16×1×1 | 72 | 54.0 | 81.1 | 66.0 | 89.9 |
| TimeSformer-HR(Bertasius et al., 2021) | IN-21K | 16×3×1 | 5109 | - | - | 62.5 | - |
| TimeSformer-L(Bertasius et al., 2021) | IN-21K | 96×3×1 | 7140 | - | - | 62.3 | - |
| ViViT-L(Arnab et al., 2021) | K400 | 16×3×4 | 11892 | - | - | 65.4 | 89.8 |
| X-ViT(Bulat et al., 2021) | IN-21K | 16×3×1 | 850 | - | - | 65.2 | 90.6 |
| X-ViT(Bulat et al., 2021) | IN-21K | 32×3×1 | 1270 | - | - | 65.4 | 90.7 |
| Mformer-HR(Patrick et al., 2021) | K400 | 16×3×1 | 2876 | - | - | 67.1 | 90.6 |
| Mformer-L(Patrick et al., 2021) | K400 | 32×3×1 | 3555 | - | - | 68.1 | 91.2 |
| Swin-B(Liu et al., 2022) | K400 | 32×3×1 | 963 | - | - | 69.6 | 92.7 |
| MViT-B,64×3(Fan et al., 2021) | K400 | 64×1×3 | 1365 | - | - | 67.7 | 90.9 |
| MViT-B-24,32×3(Fan et al., 2021) | K600 | 32×1×3 | 708 | - | - | 68.7 | 91.5 |
| Uniformer-S (Li et al., 2022a) | K400 | 16×3×1 | 125 | 57.2 | 84.9 | 67.7 | 91.4 |
| Uniformer-B (Li et al., 2022a) | K400 | 32×3×1 | 777 | 60.9 | 87.3 | 71.2 | 92.8 |
| StructViT-S-4-2 (ours) | K400 | 16×3×1 | 126 | 57.2 | 85.0 | 67.9 | 91.3 |
| StructViT-S-4-1 (ours) | K400 | 16×3×1 | 246 | 57.5 | 85.3 | 68.2 | 91.8 |
| StructViT-S-8-1 (ours) | K400 | 16×3×1 | 405 | 57.6 | 85.5 | 68.4 | 92.0 |
| StructViT-B-4-2 (ours) | K400 | 32×3×1 | 784 | 61.1 | 87.7 | 71.1 | 92.7 |
| StructViT-B-4-1 (ours) | K400 | 32×3×1 | 1963 | **61.3** | **87.8** | **71.5** | **93.1** |

(b) **Diving-48**

| model | top-1 |
|---|---|
| SlowFast-R101 (Feichtenhofer et al., 2019) | 77.6 |
| TimeSformer (Bertasius et al., 2021) | 75.0 |
| TimeSformer-HR (Bertasius et al., 2021) | 78.0 |
| TimeSformer-L (Bertasius et al., 2021) | 81.0 |
| RSANet-R50 (Kim et al., 2021) | 84.2 |
| ORViT* (Herzig et al., 2022) | 88.0 |
| StructViT-B-4-1 | **88.3** |

(c) **FineGym**

| model | Gym288 | Gym99 |
|---|---|---|
| TRN (Zhou et al., 2018) | 33.1 | 68.7 |
| I3D (Carreira & Zisserman, 2017) | 27.9 | 63.2 |
| TSM (Lin et al., 2019) | 34.8 | 70.6 |
| $\text{TSM}_{\text{Two-stream}}$ (Lin et al., 2019) | 46.5 | 81.2 |
| RSANet-R50 (Kim et al., 2021) | 50.9 | 86.4 |
| StructViT-B-4-1 | **54.2** | **89.5** |

3.3% and 3.1% on FineGym) over the previous methods without additional box annotations on both datasets. Note that ORViT Herzig et al. (2022) uses additional object bounding box annotations to train an object detector.

### 4.3 Ablation Studies

We conduct ablation studies to investigate the impact of different parameters of StructSA. We test StructViT-S on ImageNet-1K and Something-Something V1 while varying the structure dimension $D$ and the kernel size $M$. For ImageNet-1K, we train our model from scratch whereas we initialize the model with weights pretrained on ImageNet-1K when testing on Something-Something V1. We use 16 frames as input for video experiments.

Table 4a shows the effect of the structure dimension $D$. When simply applying ConvSA ($D = 1$) to our baseline with SA ($D = 0$), both methods show similar performances on both datasets whereas StructSA ($D > 1$) clearly brings large improvements. This confirms the limitation of ConvSA and the effectiveness of StructSA. As we increase $D$, the model shows larger improvements on both

Table 4: **Ablation studies on ImageNet-1K and Something-Something V1**. Top-1 and top-5 accuracies (%) are shown. For (b), we fix the structure dimension $D$ to 4.

(a) **Structure dimension** $D$.

| $D$ | ImageNet-1K | | Something V1 | |
|---|---|---|---|---|
| | top-1 | top-5 | top-1 | top-5 |
| 0 | 82.9 | 96.2 | 52.0 | 80.2 |
| 1 | 82.9 | 96.3 | 52.1 | 80.2 |
| 2 | 83.1 | 96.4 | 52.5 | 80.9 |
| 4 | 83.2 | 96.6 | 52.7 | 81.2 |
| 8 | **83.3** | **96.6** | **52.9** | **81.3** |

(b) **Kernel size** $M$**.**

| $M$ | ImageNet-1K | | Something V1 | |
|---|---|---|---|---|
| | top-1 | top-5 | top-1 | top-5 |
| $1 \times 1$ ($\times 1$) | 83.0 | 96.2 | 52.2 | 80.4 |
| $3 \times 3$ ($\times 3$) | **83.2** | **96.6** | 52.7 | **81.2** |
| $5 \times 5$ ($\times 5$) | 83.1 | 96.5 | **52.8** | **81.2** |
| $7 \times 7$ ($\times 7$) | 83.1 | 96.5 | 52.6 | 81.0 |

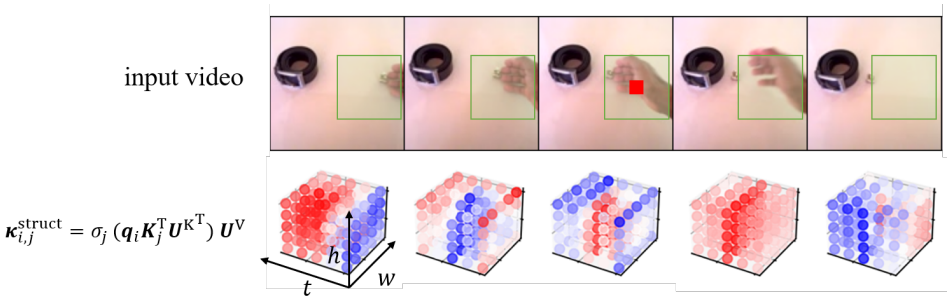


Figure 2: **Visualization of dynamic kernels $\kappa_{i,j}^{\text{struct}}$ in StructSA on Something-Something V1.** The top row shows the input frames that contain the input spatiotemporal local context (indicated by green boxes) used in the dynamic kernel computation. The bottom row presents the resulting dynamic kernels $\kappa_{i,j}^{\text{struct}}$ 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 $M = 5 \times 5 \times 5$.

datasets. In Table 4b, we also investigate different kernel sizes $M$. Compared to the baseline, enlarging kernel size to $M = 3 \times 3 \times 3$ improves the accuracy on both datasets; this validates the effectiveness of learning geometric structures. The performance saturates as the kernel size gets larger than $5 \times 5 \times 5$.

### 4.4 Visualizations of StructSA

Figure 2 visualize example dynamic kernels $\kappa_{i,j}^{\text{struct}}$ computed from self-similarity map ($i = j$) on Something-Something V1. We observe that StructSA builds kernels for spatiotemporal gradient filters that are similar to those that are already known to be effective for capturing different types of motions (Szeliski, 2010), *e.g.*, Sobel filters (first example) or Laplacian filters (second and third), over local contexts similarly to Kim et al. (2021).

## 5 Conclusion

We introduce a novel self-attention mechanism, named structural self-attention (StructSA), that exploits structural patterns of the pixel-wise correlations for visual representation learning. Instead of using a correlation individual to aggregate each feature element, StructSA leverages spatial (and temporal) structures of local correlations and aggregates chunks of local features globally across entire locations, effectively capturing relational information, *e.g.*, spatial layouts, motion, or inter-object relations in images and videos. Based on StructSA, we present a new architecture, named Structural Vision Transformer (StructViT), and demonstrate 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.

## REFERENCES


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

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.

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

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. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Adrian Bulat, Juan Manuel Perez Rua, Swathikiran Sudhakaran, Brais Martinez, and Georgios Tzimiropoulos. Space-time mixing attention for video transformer. *Advances in Neural Information Processing Systems*, 34:19594–19607, 2021.

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*, pp. 213–229. Springer, 2020.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. *arXiv preprint arXiv:2106.01548*, 2021.

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.

Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

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.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12124–12134, 2022.

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 *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2015.

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.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.

Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

Peng Gao, Jiasen Lu, Hongsheng Li, Roozbeh Mottaghi, and Aniruddha Kembhavi. Container: Context aggregation network. *arXiv preprint arXiv:2106.01401*, 2021.

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 *Advances in Neural Information Processing Systems 34*, 2021.

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 *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12175–12185, 2022.

Kai Han, Rafael S Rezende, Bumsub Ham, Kwan-Yee K Wong, Minsu Cho, Cordelia Schmid, and Jean Ponce. Scnet: Learning semantic correspondence. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017.

Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3148–3159, 2022.

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.

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*, pp. 3464–3473, 2019.

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

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. *Advances in Neural Information Processing Systems*, 34:18590–18602, 2021.

Dahyun Kang, Heeseung Kwon, Juhong Min, and Minsu Cho. Relational embedding for few-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8822–8833, 2021.

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.

Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. Relational self-attention: What's missing in attention for video understanding. *Advances in Neural Information Processing Systems*, 34:8046–8059, 2021.

Seungryong Kim, Dongbo Min, Bumsub Ham, Sangryul Jeon, Stephen Lin, and Kwanghoon Sohn. Fcss: Fully convolutional self-similarity for dense semantic correspondence. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16020–16030, 2021.

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

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.

Kunchang Li, Xianhang Li, Yali Wang, Jun Wang, and Yu Qiao. {CT}-net: Channel tensorization network for video classification. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=UoaQUQREMOs.

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*, 2022a.

Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4804–4814, 2022b.

Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In *Proc. European Conference on Computer Vision (ECCV)*, 2018.

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

Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3202–3211, 2022.

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

Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

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.

Juhong Min and Minsu Cho. Convolutional hough matching networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2940–2950, 2021.

Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021.

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. *Advances in neural information processing systems*, 34:12493–12506, 2021.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10428–10436, 2020.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Stand-alone self-attention in vision models. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2019.

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

Dian Shao, Yue Zhao, Bo Dai, and Dahua Lin. Finegym: A hierarchical video dataset for fine-grained action understanding. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7262–7272, 2021.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7464–7473, 2019.

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

Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proc. International Conference on Machine Learning (ICML)*, 2019.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International Conference on Machine Learning*, pp. 10096–10106. PMLR, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2019.

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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2017.

Heng Wang, Du Tran, Lorenzo Torresani, and Matt Feiszli. Video modeling with correlation networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

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 *Proc. European Conference on Computer Vision (ECCV)*, 2016.

Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1895–1904, 2021a.

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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 568–578, 2021b.

Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22–31, 2021.

Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3333–3343, 2022.

Gengshan Yang and Deva Ramanan. Volumetric correspondence networks for optical flow. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2019.

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.

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.

Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

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 *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 13001–13008, 2020.

Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *Proc. European Conference on Computer Vision (ECCV)*, 2018.

# APPENDIX A FULL DERIVATION OF CONVSA AND STRUCTSA

## APPENDIX A.1 CONVSA

In this section, we provide a full derivation of ConvSA with a conventional channel-wise convolution, of which weights are not shared across channels. Given the channel-wise convolution weights $\boldsymbol{H}^{\mathrm{K}}, \boldsymbol{H}^{\mathrm{V}} \in \mathbb{R}^{C \times M}$, $c$-th channel of each key $k_i^{\mathrm{conv}}$ and value $v_i^{\mathrm{conv}}$ is computed as,

$$\boldsymbol{k}_{i,c}^{\mathrm{conv}} = \boldsymbol{H}_c^{\mathrm{K}} \boldsymbol{K}_{i,:,c} \in \mathbb{R}, \tag{10}$$

$$\boldsymbol{v}_{i,c}^{\mathrm{conv}} = \boldsymbol{H}_c^{\mathrm{V}} \boldsymbol{V}_{i,:,c} \in \mathbb{R}, \tag{11}$$

where $\boldsymbol{K}_{i,:,c}$ and $\boldsymbol{V}_{i,:,c}$ indicates $M$ elements of $c$th dfdf. Plugging Eqs. 10 and 11 into Eq. 6, each channel of ConvSA output is computed as,

$$\boldsymbol{y}_{i,c} = \sum_{j=1}^{N} \sigma_j \left( \boldsymbol{q}_i \boldsymbol{k}_j^{\mathrm{conv}\mathsf{T}} \right) \boldsymbol{v}_j^{\mathrm{conv}} \tag{12}$$

$$= \sum_{j=1}^{N} \sigma_j \left( \sum_{c=1}^{C} \boldsymbol{q}_{i,c} \boldsymbol{H}_c^{\mathrm{K}} \boldsymbol{K}_{j,:,c} \right) \boldsymbol{v}_j^{\mathrm{conv}} \tag{13}$$

$$= \sum_{j=1}^{N} \sigma_j \left( \sum_{m=1}^{M} \sum_{c=1}^{C} \boldsymbol{q}_{i,c} \boldsymbol{H}_{c,m}^{\mathrm{K}} \boldsymbol{K}_{i,m,c} \right) \boldsymbol{v}_j^{\mathrm{conv}} \tag{14}$$

$$= \sum_{j=1}^{N} \sigma_j \left( \mathrm{vec}\left( \mathbb{1} \boldsymbol{q}_i \odot \boldsymbol{K}_j \right) \mathrm{vec}\left( \boldsymbol{H}^{\mathrm{K}} \right) \right) \boldsymbol{H}_c^{\mathrm{V}} \boldsymbol{V}_{j,:,c}, \tag{15}$$

where $\mathbb{1} \in \mathbb{R}^{M \times 1}$ is column-wise one vector.

## APPENDIX A.2 STRUCTSA

$\boldsymbol{H}^{\mathrm{K}}, \boldsymbol{H}^{\mathrm{V}} \in \mathbb{R}^{C \times M}$ To compute StructSA, we extend the pattern detector $\boldsymbol{H}^{\mathrm{K}}$ and the context aggregator $\boldsymbol{H}^{\mathrm{V}}$ to matrices $\mathbf{H}^{\mathrm{K}}$, $\mathbf{H}^{\mathrm{V}}$ and compute keys $\boldsymbol{K}_{i,c}^{\mathrm{struct}}$ and values $\boldsymbol{V}_{i,c}^{\mathrm{struct}}$, as,

$$\mathbf{H}^{\mathrm{K}} = [\boldsymbol{H}_1^{\mathrm{K}}, \cdots, \boldsymbol{H}_D^{\mathrm{K}}] \in \mathbb{R}^{D \times C \times M} \tag{16}$$

$$\mathbf{H}^{\mathrm{V}} = [\boldsymbol{H}_1^{\mathrm{V}}, \cdots, \boldsymbol{H}_D^{\mathrm{V}}] \in \mathbb{R}^{D \times C \times M} \tag{17}$$

$$\boldsymbol{K}_{i,c}^{\mathrm{struct}} = \mathbf{H}_{:,c}^{\mathrm{K}} \boldsymbol{K}_{i,:,c} \in \mathbb{R}^{D}, \tag{18}$$

$$\boldsymbol{V}_{i,c}^{\mathrm{struct}} = \mathbf{H}_{:,c}^{\mathrm{V}} \boldsymbol{V}_{i,:,c} \in \mathbb{R}^{D}, \tag{19}$$

From Eq. 18, 19 and 9, each channel of StructSA output can be formulated as,

$$\boldsymbol{y}_{i,c} = \sum_{j=1}^{N} \sigma_j \left( \boldsymbol{q}_i \boldsymbol{K}_j^{\mathrm{struct}\mathsf{T}} \right) \boldsymbol{V}_j^{\mathrm{struct}} \tag{20}$$

$$= \sum_{j=1}^{N} \sigma_j \left( \sum_{c=1}^{C} \boldsymbol{q}_{i,c} \mathbf{H}_{:,c}^{\mathrm{K}} \boldsymbol{K}_{j,:,c} \right) \boldsymbol{V}_j^{\mathrm{struct}} \tag{21}$$

$$= \sum_{j=1}^{N} \sigma_j \left( \sum_{m=1}^{M} \sum_{c=1}^{C} \boldsymbol{q}_{i,c} \mathbf{H}_{:,c,m}^{\mathrm{K}} \boldsymbol{K}_{i,m,c} \right) \boldsymbol{V}_j^{\mathrm{struct}} \tag{22}$$

$$= \sum_{j=1}^{N} \sigma_j \left( \mathrm{vec}\left( \mathbb{1} \boldsymbol{q}_i \odot \boldsymbol{K}_j \right) f\left( \mathbf{H}^{\mathrm{K}} \right)^{\mathsf{T}} \right) \boldsymbol{H}_c^{\mathrm{V}} \boldsymbol{V}_{j,:,c}, \tag{23}$$

$$\text{where } f\left( \mathbf{H}^{\mathrm{K}} \right)^{\mathsf{T}} \in \mathbb{R}^{D \times MC}. \tag{24}$$