Lightweight Structure-Aware Attention for Visual Understanding

Heeseung Kwon, Francisco M. Castro, Manuel J. Marin-Jimenez, Nicolas Guil, Karteek Alahari

Lightweight Structure-Aware Attention for Visual Understanding

Heeseung Kwon[1] Francisco M. Castro[2] Manuel J. Marin-Jimenez[3] Nicolas Guil[2] Karteek Alahari[1]
[1]Univ. Grenoble Alpes, Inria, CNRS, Grenoble INP, LJK
[2]Department of Computer Architecture, University of Málaga
[3]Department of Computing and Numerical Analysis, University of Córdoba

Abstract

Vision Transformers (ViTs) have become a dominant paradigm for visual representation learning with self-attention operators. Although these operators provide flexibility to the model with their adjustable attention kernels, they suffer from inherent limitations: (1) the attention kernel is not discriminative enough, resulting in high redundancy of the ViT layers, and (2) the complexity in computation and memory is quadratic in the sequence length. In this paper, we propose a novel attention operator, called lightweight structure-aware attention (LiSA), which has a better representation power with log-linear complexity. Our operator learns structural patterns by using a set of relative position embeddings (RPEs). To achieve log-linear complexity, the RPEs are approximated with fast Fourier transforms. Our experiments and ablation studies demonstrate that ViTs based on the proposed operator outperform self-attention and other existing operators, achieving state-of-the-art results on ImageNet, and competitive results on other visual understanding benchmarks such as COCO and Something-Something-V2. The source code of our approach will be released online.

1. Introduction

Since the emergence of the vision transformer (ViT) [12], transformers have become the dominant neural architecture for visual understanding, outperforming convolutional neural networks (CNNs). Self-attention, a core operator of ViT, has relative merits compared to convolution because of the adjustable attention kernel and its ability to capture long-range dependencies. However, self-attention has inherent limitations for visual recognition. First, the attention kernel has difficulty learning discriminative features due to the lack of desirable inductive biases, resulting in high redundancy of the ViT layers [27, 56]. Thus, it usually requires a large amount of data [12] and aggressive augmentations [45] to obtain good performance. Second, the complexity of self-attention is quadratic in the length of its input sequence, making the operator impractical for high-resolution images and difficult to be adopted for hierarchical models.

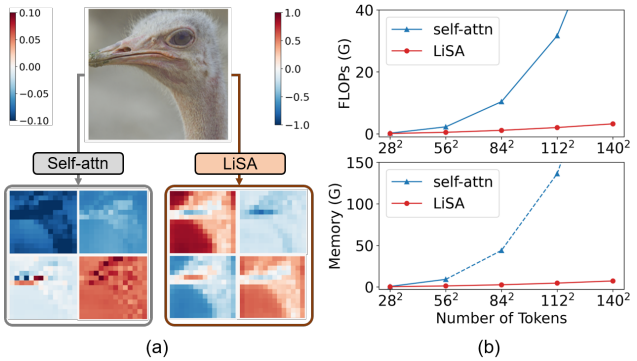


Figure 1. **Self-attention vs. LiSA.** (a) Feature visualization of self-attention & LiSA: compared to self-attention, LiSA learns better discriminative features by capturing geometric structural patterns. (b) Computation (FLOPs) & memory cost: LiSA is significantly more efficient than self-attention when the sequence length increases, due to its log-linear complexity.

Recent approaches have proposed new types of operators to address the limitations of self-attention. Some of them have attempted to learn better discriminative features with self-attention by including relative position embeddings (RPEs) [2, 10, 30, 37] or capturing geometric structures (*e.g.*, image gradients, video motion) [23, 61]. However, these operators have high computational complexity, and thus it is hard to capture long-range dependencies on high resolutions [2, 10, 23, 30, 61]. Some other methods have proposed efficient attention operators to handle the complexity of self-attention [5, 7, 31, 33, 35, 50]. Although these operators have linear complexity with factorized softmax attention kernel, they often underperform on visual understanding, compared to the original attention-based models [33, 35].

In this paper, we propose an effective yet efficient operator, *lightweight structure-aware attention (LiSA)*, which learns effective discriminative features while requiring only log-linear complexity. To improve the expressivity, we exploit a set of RPEs for learning both convolutional inductive biases (*e.g.*, translation invariance) and geometric structural

patterns in the query-key correlation. Fig. 1a illustrates a few sample feature activations from the early layers of the same-scale models (DeiT-T [45] vs. LiSA-I-T). LiSA effectively captures geometric structures in the image, while self-attention feature activations are relatively weak and uninformative in the early layers due to the lack of desirable inductive biases. Although the use of the RPEs is effective for LiSA, it is infeasible to cover long-range dependencies due to their computation cost. Thus, we leverage them efficiently with fast Fourier transforms (FFTs), achieving log-linear complexity. We demonstrate this efficiency in Fig. 1b. While the complexity of self-attention increases exponentially with respect to the number of tokens, that of LiSA increases gracefully due to log-linear complexity.

Our main contributions are as follows: (1) to overcome the limitations of self-attention, we propose a new attention operator called LiSA, which learns both convolutional priors and structural patterns with log-linear complexity, and, (2) LiSANets, the models based on our LiSA operator, outperform other counterparts, achieving competitive results on visual understanding benchmarks, ImageNet [11], COCO [29], and Something-Something-V2 [15], respectively.

2. Related Work

ViTs for visual understanding. After the success of ViT [12], transformer architectures have been widely adopted in a variety of visual understanding tasks [1, 3, 43, 46, 54, 56]. Several approaches have proposed improvements to the original ViT [12], *e.g.*, using a teacher-student scheme [45], a better tokenization scheme [56], or using small splits of the tokens to obtain richer local information [16]. Recently, several approaches try to combine transformers and CNNs to leverage the best of each world. Some of these incorporate convolutions into the attention block [13, 14, 27, 53] to increase the expressivity of self-attention. Others employ the hierarchical structure of CNNs to learn better discriminative features [10, 14, 27, 28, 30, 51, 53, 60]. To handle the complexity of hierarchical ViTs, these approaches use convolutions instead of self-attention operators in the early stages [10, 27], adopt local attention [30, 60], or downsampled attention [14, 28, 51, 53]. While these methods heavily depend on handcrafted designs, our proposed model purely based on LiSA achieves notable performance without such designs due to the high expressivity and efficiency of LiSA.

Highly-expressive operators. The new types of operators proposed recently, increase the representation power by developing self-attention [2, 10, 23, 30, 37, 61] or convolution [6, 21, 26, 34]. Attention-based operators have achieving this by adding convolutional priors [10, 30, 37] or capturing relational structures [2, 23, 61]. Convolution-based operators have dynamically adapted convolution kernels based on the input features [6, 21, 26, 34]. However, these highly-expressive operators come with high computational complexity and are typically limited to local interactions [2,23,26,61]. One example is the relational self-attention (RSA) [23], which is related to our work. RSA is one of the most expressive operators that captures structural patterns with relational components, but it is also limited to local interactions due to its high computational complexity. In contrast, our proposed LiSA shows the highest level of expressivity by capturing global structural patterns with log-linear complexity.

Lightweight operators. Some of the existing lightweight attention operators factorize the softmax attention kernel [7, 22, 41, 50]. While they have linear complexity, they usually perform worse than the original attention in terms of accuracy [33, 35]. Other approaches have attempted to linearize RPE-added attention operators [5, 31, 33], but they still underperform on visual recognition. Recently, a few approaches adopt FFTs for efficiently covering global receptive fields [25, 33, 39]. The Global Filter (GF) layer [39] is one such operator, which implements an efficient global circular convolution with FFTs. However, the expressivity of the GF layer is constrained due to static convolution kernels. Our LiSA also adopts this technique for efficiency but focuses on learning structural patterns with its dynamic attention kernels, leading to better performance.

3. Background

Self-attention. The self-attention operator [47] is a core component in transformer architectures that generates the query-key attention for updating the value. Let N denote the sequence length (the number of tokens) and C the number of input channels. Given an input feature $\boldsymbol{X} \in \mathbb{R}^{N\times C}$, query, key, value, $\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V} \in \mathbb{R}^{N\times C}$, are firstly produced by independent linear projections, and each element of the output $\boldsymbol{Y} \in \mathbb{R}^{N\times C}$ of self-attention is expressed as

$$Y_{i,k} = \sum_{j}^{N} \sigma(A_{i,j}) V_{j,k}, \quad A_{i,j} = \frac{1}{\sqrt{C}} \sum_{k}^{C} Q_{i,k} K_{j,k}. \quad (1)$$

Note that σ is the softmax function along j-axis. Two main characteristics of self-attention are that: (1) the operator represents a global interaction where the size of the attention kernel for each query is equal to N, and (2) the attention kernel dynamically changes according to the input feature. However, it is unable to encode the relative order of tokens due to the lack of convolutional inductive biases, resulting in performance degradation on visual recognition.

Relative position embedding (RPE). One of the popular schemes to handle the lack of convolutional inductive biases is adopting an RPE for the self-attention operator [10,30,37]. A common RPE has the shape of a Toeplitz matrix, and it

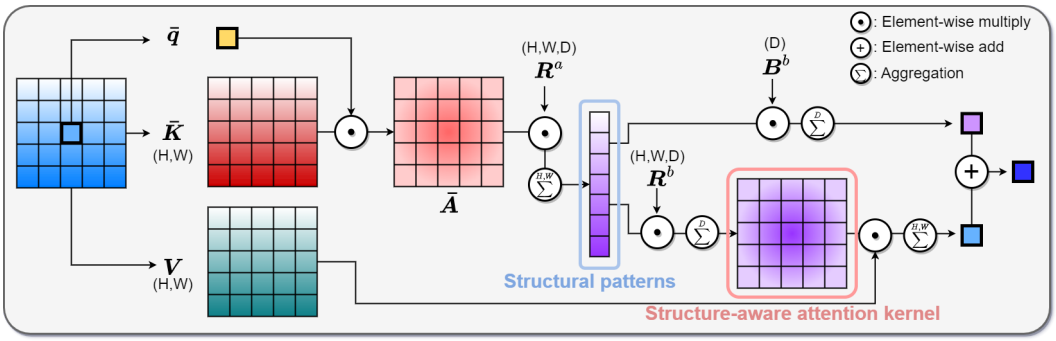


Figure 2. **Computational graph of Structure-aware Attention (SA) for each query.** After obtaining the query-key dot-product correlation ($\bar{\boldsymbol{A}}$), structural patterns in $\bar{\boldsymbol{A}}$ are encoded by $\boldsymbol{R}^a$, and utilized in two ways: 1) the patterns are used for generating a structure-aware attention kernel with $\boldsymbol{R}^b$, and 2) directly projected as a structural feature with $\boldsymbol{B}^b$. Note that $N = H \times W, C = 1$ in this figure.

consists of learnable weights which can be expressed as

$$\mathcal{T}(\boldsymbol{e}) = \begin{pmatrix} e_N & e_{N+1} & e_{N+2} & \cdots & e_{2N-1} \\ e_{N-1} & e_N & e_{N+1} & \cdots & e_{2N-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ e_1 & e_2 & e_3 & \cdots & e_N \end{pmatrix}, \tag{2}$$

where $\boldsymbol{e} = \{e_1, e_2, \cdots e_{2N-1}\}$. When RPE is added, the attention operator is formulated as

$$Y_{i,k} = \sum_j^N \sigma(A_{i,j} + R_{i,j}) V_{j,k}, \ \boldsymbol{R} = \mathcal{T}(\boldsymbol{e}) \in \mathbb{R}^{N \times N}. \tag{3}$$

By introducing relative positional information into attention, the self-attention operator obtains the ability to learn convolutional inductive biases.

Limitations of self-attention with RPE. Despite several approaches showing the effectiveness of RPE, the self-attention operator has inherent limitations. First, the expressivity of the operator is still insufficient; it is difficult to capture geometric structures (*e.g.*, image gradients, video motion) since the softmax attention kernel may not be effective for encoding gradient information due to its non-negativity [23, 38]. Additionally, although the query-key correlation $\boldsymbol{A}$ suppresses photometric variations and reveals geometric structures [24, 40], the kernel is aggregated with the value $\boldsymbol{V}$ without considering structural patterns in $\boldsymbol{A}$. Second, the operator suffers from quadratic complexity ($\mathcal{O}(N^2)$) since the non-linear softmax function and RPE are hard to be linearized. Although a few approaches [7, 33, 50] have attempted to approximate the softmax function with kernelized methods to make the operator more efficient, they do not improve over the original transformer in accuracy due to its training instability [33] or approximation errors [35].

4. Structure-aware Attention

4.1. Basic Form of Structure-aware Attention (SA)

Learning convolutional inductive biases. To handle the limitations of self-attention, we devise a new attention operator that leverages the advantages of convolution. Unlike the conventional usage of an RPE (Eq. 3), we employ it as multiplicative weights for learning convolutional priors as follows:

$$Y_{i,k} = \sum_j^N \bar{A}_{i,j} R_{i,j} V_{j,k}, \quad \bar{A}_{i,j} = \sum_k^C \bar{Q}_{i,k} \bar{K}_{j,k}. \tag{4}$$

Note that $\bar{\boldsymbol{Q}}, \bar{\boldsymbol{K}}$ are L2-normalized query and key, respectively. In Eq. 4, the RPE $\boldsymbol{R}$ not only learns relative token orders, but also actively adjusts the dot-product attention values with its learnable weights. We remove the softmax function to allow the attention kernel to include negative values, which may be effective for encoding structural information. The query and key are L2-normalized to obtain the normalization effect of a softmax, which is helpful for stabilizing the training procedure [33]. Since matrix multiplication between the Toeplitz matrix $\boldsymbol{R}$ and the value $\boldsymbol{V}$ is equivalent to a global convolution [42] that applies the convolution kernel $\boldsymbol{e} \in \mathbb{R}^{2N-1}$ for the value $\boldsymbol{V}$, the operator can also be considered as a global dynamic convolution where the dynamic component of the convolution kernel is based on the query-key correlation. Thus, the proposed operator merges the characteristics of self-attention and convolution.

Learning structural patterns. Nevertheless, the above operator is (Eq. 4) still limited for capturing rich structural patterns in the dot-product attention $\bar{\boldsymbol{A}}$ since the N-size attention kernel for each query is aggregated with the value $\boldsymbol{V}$ before the patterns are encoded. To effectively capture the structural patterns, we aggregate and recompose the attention values with multiple RPEs. The updated operator is

formulated as:

$$Y_{i,k} = \sum_{j}^{N}\sum_{d}^{D}\sum_{n}^{N}(\bar{A}_{i,n}R^{a}_{i,n,d})(R^{b}_{i,j,d}V_{j,k} + B^{b}_{k,d}) \\ = \sum_{c}^{C}\bar{Q}_{i,c}\sum_{d}^{D}\sum_{j,n}^{N}(\bar{K}_{n,c}R^{a}_{i,n,d})(R^{b}_{i,j,d}V_{j,k} + B^{b}_{k,d}), \tag{5}$$

where $\boldsymbol{R}^a = \{\mathcal{T}(\boldsymbol{e}^a_1), \mathcal{T}(\boldsymbol{e}^a_2), \cdots, \mathcal{T}(\boldsymbol{e}^a_D)\}, \boldsymbol{R}^b = \{\mathcal{T}(\boldsymbol{e}^b_1), \mathcal{T}(\boldsymbol{e}^b_2), \cdots, \mathcal{T}(\boldsymbol{e}^b_D)\} \in \mathbb{R}^{N\times N\times D}$ are RPE tensors composed of sets of Toeplitz matrices and $\boldsymbol{B}^b \in \mathbb{R}^{C\times D}$ is a learnable projection matrix, respectively. Note that D is the number of hidden channels. The computational graph of Eq. 5 is illustrated in Fig. 2. For each query, the learnable RPE tensor $\boldsymbol{R}^a$ captures structural patterns by encoding attention values as a D-size vector. We utilize this vector in two ways: first, for generating a new structure-aware attention kernel along the j-axis with the RPE tensor $\boldsymbol{R}^b$, and then also directly projected as a feature representation with the learnable matrix $\boldsymbol{B}^b$. The generated attention kernel with $\boldsymbol{R}^b$ updates $\boldsymbol{V}$ in a structure-aware manner like the relational kernel of RSA [23]. The feature representation with $\boldsymbol{B}^b$ directly encodes structural patterns, which is related to correlation-based representations [24, 48]. While these prior methods have focused on capturing local structures by convolutions, our operator captures global geometric structures through RPEs.

4.2. Improving the Expressivity of SA

We can further improve its expressivity by exploiting semantic information of the input channels. Here we describe the advanced form of structure-aware attention.

Capturing channel-wise structural patterns. To exploit the semantics of the input channels, we employ a different type of query-key correlation, the Hadamard-product correlation. A few approaches [23, 61] have demonstrated that Hadamard-product correlation is more effective than the dot-product one due to the use of richer query-key semantics. Considering the Hadamard correlation is a 3-dimensional tensor $\bar{A}_{i,n,c} = \bar{Q}_{i,c}\bar{K}_{n,c} \in \mathbb{R}^{N\times N\times C}$, we expand the RPE tensor $\boldsymbol{R}^a$ by C channels for encoding the Hadamard correlation. The modified operator is formulated as follows:

$$Y_{i,k} = \sum_{c}^{C}\bar{Q}_{i,c}\sum_{d}^{D}\sum_{j,n}^{N}(\bar{K}_{n,c}\tilde{R}^{a}_{i,n,c,d})(R^{b}_{i,j,d}V_{j,k} + B^{b}_{k,d}), \tag{6}$$

where $\tilde{\boldsymbol{R}}^a \in \mathbb{R}^{N\times N\times C\times D}$ is the expanded RPE tensor, and learnable weights in the tensor increase from $\boldsymbol{E}^a = \{\boldsymbol{e}^a_1, \boldsymbol{e}^a_2, \cdots, \boldsymbol{e}^a_D\} \in \mathbb{R}^{(2N-1)\times D}$ to $\tilde{\boldsymbol{E}}^a = \{\boldsymbol{e}^a_1, \boldsymbol{e}^a_2, \cdots, \boldsymbol{e}^a_{CD}\} \in \mathbb{R}^{(2N-1)\times C\times D}$. In Eq. 6, for each query, the expanded tensor $\tilde{\boldsymbol{R}}^a$ captures channel-wise structural patterns by encoding a $(N \times C)$ Hadamard correlation matrix as a D-size vector. Since this process does not require additional computation, the operator can exploit rich semantics through the Hadamard correlation by only increasing the number of parameters.

operator	computation	memory
self-attention [47]	$\mathcal{O}(N^2C)$	$\mathcal{O}(N^2 + NC)$
structure-aware attention (Eq. 4)	$\mathcal{O}(N^2C)$	$\mathcal{O}(N^2 + NC)$
structure-aware attention (Eq. 5)	$\mathcal{O}(N^2CD)$	$\mathcal{O}(NCD)$
LiSA (FFT approximation)	$\mathcal{O}(NCD\log_2 N)$	$\mathcal{O}(NCD)$

Table 1. **Comparison of complexity of the operators.** N, C, D denote the sequence length, the number of channels, and the number of latent channels respectively. Our operator has log-linear and linear complexity in computation and memory respectively.

Combining features based on input semantics. While the current form relies on query-key similarities for extracting features, we can also produce a different type of features that are fully based on input semantics, through a simple bias term. When we add a bias term $\boldsymbol{B}^a \in \mathbb{R}^{C\times D}$ to Eq 6,

$$Y_{i,k} = \sum_{c}^{C}\bar{Q}_{i,c}\sum_{d}^{D}\sum_{j,n}^{N}(\bar{K}_{n,c}\tilde{R}^{a}_{i,n,c,d} + B^{a}_{c,d})(R^{b}_{i,j,d}V_{j,k} + B^{b}_{k,d}) \tag{7}$$

$$= \sum_{c}^{C}\bar{Q}_{i,c}\sum_{d}^{D}\sum_{j,n}^{N}(\bar{K}_{n,c}\tilde{R}^{a}_{i,n,c,d}R^{b}_{i,j,d}V_{j,k} + B^{a}_{c,d}R^{b}_{i,j,d}V_{j,k} + \bar{K}_{n,c}\tilde{R}^{a}_{i,n,c,d}B^{b}_{k,d} + B^{a}_{c,d}B^{b}_{k,d}). \tag{8}$$

In Eq. 8, we obtain two additional terms. While the last term $(\sum_{c,d}^{C,D}\bar{Q}_{i,c}B^a_{c,d}B^b_{k,d})$ is a common linear projection of the query feature, the second term $(\sum_{c,j,d}^{C,N,D}\bar{Q}_{i,c}B^a_{c,d}R^b_{i,j,d}V_{j,k})$ produces a different type of features. Considering the multiplication between Toeplitz matrices $\boldsymbol{R}^b$ and the value $\boldsymbol{V}$ is equivalent to a global convolution, the second term can be regarded as a global dynamic convolution, where the dynamic component is based on the query semantics. This is roughly equivalent to Lambda convolution or Involution operators [2, 26], which have shown the generated features are effective for learning spatial structures. Similarly, our operator gets the benefit of combining features based on input semantics through the additive bias term.

4.3. Lightweight Structure-aware Attention (LiSA)

Although our proposed operator (SA) is highly-expressive, it is impractical for neural architectures due to its huge computational complexity. Here we describe our final form, LiSA, which significantly reduces the complexity by efficiently processing the heavy RPE tensors through FFTs.

Approximating RPEs with FFTs. In Eq. 7, RPE tensor

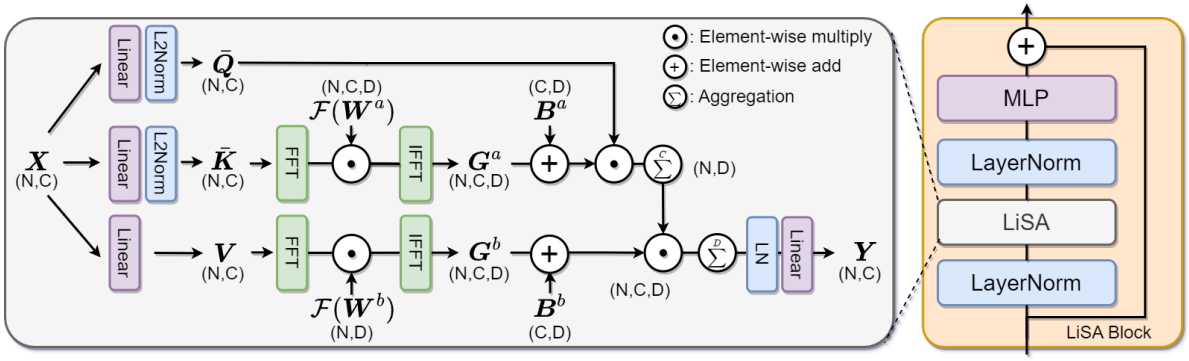


Figure 3. **Computational graph of LiSA and its block configuration.** See text for details.

multiplications can be considered as global convolutions:

$$Y_{i,k} = \sum_{c}^{C} \bar{Q}_{i,c} \sum_{d}^{D} (G^a_{i,c,d} + B^a_{c,d})(G^b_{i,k,d} + B^b_{k,d}), \quad (9)$$
$$\text{where } \boldsymbol{G}^a = \bar{\boldsymbol{K}} * \tilde{\boldsymbol{E}}^a, \boldsymbol{G}^b = \boldsymbol{V} * \boldsymbol{E}^b \in \mathbb{R}^{N\times C\times D}.$$

Note that $\boldsymbol{E}^b = \{\boldsymbol{e}^b_1, \boldsymbol{e}^b_2, \cdots, \boldsymbol{e}^b_D\} \in \mathbb{R}^{(2N-1)\times D}$ are learnable weights of the RPE tensor $\boldsymbol{R}^b$, and $*$ denotes convolution. $\boldsymbol{G}^b$ is a global convolution that applies the global kernels $\boldsymbol{E}^b \in \mathbb{R}^{(2N-1)\times D}$ for value $\boldsymbol{V} \in \mathbb{R}^{N\times C}$ by sharing the kernels across C channels, and $\boldsymbol{G}^a$ is a depth-wise global convolution that applies the global kernels $\tilde{\boldsymbol{E}}^a \in \mathbb{R}^{(2N-1)\times C\times D}$ for the key $\bar{\boldsymbol{K}} \in \mathbb{R}^{N\times C}$. To reduce the complexity of these global convolutions, we approximate them as *global circular convolutions*, indicating that the RPEs are altered by circular position embeddings consisting of circulant matrices. These circular convolutions can be efficiently computed by FFTs [39] via the convolution theorem of Fourier transform: *multiplication in the frequency domain is equal to circular convolution in the time domain.* Thus, we can rewrite the convolutions as:

$$\boldsymbol{G}^a \approx \bar{\boldsymbol{K}} \circledast \boldsymbol{W}^a = \mathcal{F}^{-1}(\mathcal{F}(\bar{\boldsymbol{K}}) \odot \mathcal{F}(\boldsymbol{W}^a)),$$
$$\boldsymbol{G}^b \approx \boldsymbol{V} \circledast \boldsymbol{W}^b = \mathcal{F}^{-1}(\mathcal{F}(\boldsymbol{V}) \odot \mathcal{F}(\boldsymbol{W}^b)), \quad (10)$$

Note that $\boldsymbol{W}^a = \{\boldsymbol{w}^a_1, \cdots, \boldsymbol{w}^a_{CD}\} \in \mathbb{R}^{N\times C\times D}$, $\boldsymbol{W}^b = \{\boldsymbol{w}^b_1, \cdots, \boldsymbol{w}^b_D\} \in \mathbb{R}^{N\times D}$ are learnable weights of the circular convolutions and $\circledast, \odot, \mathcal{F}, \mathcal{F}^{-1}$ denotes circular convolution, element-wise multiplication, FFT, and IFFT, respectively. The circular convolutions have the half size of parameters since the kernel size reduces from $(2N-1)$ to N. As shown in Tab. 1, we reduce the complexity in computation and memory on a linear scale by approximating heavy global interactions with FFTs. Moreover, since the input features and weights are real-valued, we can additionally reduce the complexity by half via RFFT and inverse RFFT. The computational graph of LiSA is illustrated in Fig. 3.

Model	#Blocks	#Channels (#heads)
LiSANet-I-T	12	192 (12)
LiSANet-H-S	[1, 2, 11, 2]	[64 (4), 128 (8), 256 (16), 512 (32)]
LiSANet-H-B	[2, 3, 16, 3]	[96 (6), 192 (12), 384 (24), 768 (48)]

Table 2. Detailed configurations of different variants of LiSANet. For hierarchical models, we provide the number of channels and blocks in 4 stages. FLOPs are calculated with a 224×224 input.

5. Experiments

We first describe the implementation details and then present extensive results. This includes a set of comprehensive ablation studies and a state-of-the-art comparison on ImageNet-1K [11]. Finally, we also verify the effectiveness of LiSA on object detection with COCO [29], and video action recognition with Something-Something-V2 (SSV2) [15].

5.1. Implementation details

LiSA block. Our proposed block follows the traditional transformers sequence of layers [1, 12, 45]: layer normalization (LN), attention operator, LN and MLP. Instead of using a traditional attention operator, we use LiSA. The latent channel size D is set to 16 as default. The overall block configuration is shown in Fig. 3.

LiSANet. To demonstrate the effectiveness of LiSA, we define two different style transformer architectures as shown in Tab. 2. The first style is *an isotropic model* (LiSANet-I-T) which has no downsampling layers and fixes the number of tokens (14×14) at all depths. The second style is *hierarchical models* (LiSANet-H-S, LiSANet-H-B) following the recent hierarchical ViT architectures [27, 30, 39]. Hierarchical models (LiSANet-H-S, LiSANet-H-B) are composed of 4 stages with a different number of blocks, and the number of tokens is downsampled in each stage. All the details of our variants are summarized in the Supplementary Material.

Setup. For ImageNet-1K, our isotropic model (LiSANet-I-T) is trained for 150 epochs, and hierarchical models

operator	FLOPs	#params	top-1	top-5
Self-attn [47]	1.25 G	5.72 M	71.0	90.0
Self-attn w/ RPE [37]	1.25 G	5.72 M	72.2	90.9
Self-attn w/ RPE ($C\uparrow$) [37]	1.40 G	6.44 M	73.4	91.6
Depthwise conv (7×7) [19]	0.84 G	4.49 M	69.0	89.2
GF layer [39]	0.82 G	4.90 M	69.5	89.4
GF layer ($C\uparrow$) [39]	1.27 G	7.37 M	72.4	91.0
Lambda convolution [2]	2.41 G	5.41 M	72.6	91.0
RSA [23]	5.34 G	8.23 M	74.5	92.2
LiSA (ours)	1.21 G	6.36 M	**74.9**	**92.4**

(a) **Comparison with other basic operators**.

operator	FLOPs	#params	top-1	top-5
Self-attn	1.25 G	5.72 M	71.0	90.0
Self-attn w/ RPE	1.25 G	5.72 M	72.2	90.9
Self-attn w/ h RPEs	1.25 G	5.81 M	72.7	91.3
SA (Eq. 4)	1.25 G	5.72 M	72.4	91.2
SA (Eq. 4 w/ h RPEs)	1.25 G	5.81 M	73.6	91.7
SA (Eq. 5)	3.92 G	5.99 M	73.9	92.0
+ bias term $\boldsymbol{B}^a$	3.92 G	5.99 M	74.2	92.1
+ Hadamard corr	3.92 G	8.09 M	74.9	92.3
LiSA (ours)	1.21 G	6.36 M	**74.9**	**92.4**

(b) **Effectiveness of LiSA components**.

kernel size	FLOPs	#params	top-1	top-5
Local - 3×3	1.22 G	5.75 M	71.9	90.7
Local - 5×5	1.45 G	5.80 M	73.8	91.7
Local - 7×7	1.80 G	5.88 M	73.8	91.8
Global (ours)	1.21 G	6.36 M	**74.9**	**92.4**

(c) **Impact of global interactions**.

D	FLOPs	#params	top-1	top-5
1	1.11 G	5.76 M	71.3	90.5
4	1.13 G	5.88 M	73.6	91.6
8	1.17 G	6.04 M	74.4	92.2
16 (ours)	1.21 G	6.36 M	**74.9**	**92.4**

(d) **Influence of the number of latent channels** D.

Table 3. **Ablation studies on ImageNet**. Top-1, top-5 accuracy (%), FLOPs (G) and the number of paramaters (M) are shown.

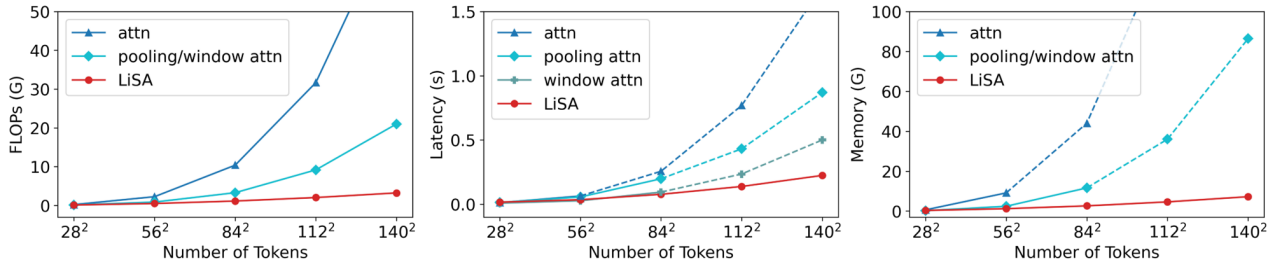


Figure 4. **Comparisons among LiSA & efficient attention blocks in (a) FLOPs, (b) latency, and (c) memory.** Dotted lines denote estimated values due to the limited GPU memory. The latency and memory is measured by an RTX A5000 GPU with batch size 16.

(LiSANet-H-S, LiSANet-H-B) are trained for 300 epochs. We follow the rest of the training recipes suggested in [30,45] for a fair comparison. For COCO, we adopt standard Mask R-CNN [17] detection frameworks, which employ ImageNet-1K pre-trained weights for fine-tuning. We use a $1\times$ schedule (12 epochs), and follow the same recipe as in [30]. For SS-V2, the isotropic model (LiSANet-I-T) is trained for 60 epochs from scratch, and we adopt the uniform sampling strategy [49] for training and a single crop inference for testing. Other recipes are in the Supplementary Material.

5.2. Ablation studies

We use the isotropic model (LiSANet-I-T) for ablations since the operators with quadratic complexity are hard to be adopted for hierarchical models due to their extreme memory consumption shown in Fig. 4c. Unless specified otherwise, we use 224×224 resolution for input.

Comparison with other operators. In Tab. 3a, we compare our LiSA operator with several others, including self-attention [37,47], convolution [19,39], and highly-expressive operators [2,23]. For a fair comparison, we only replace our operator with others in the LiSA blocks, and all the receptive fields are set as global, except for the 7×7 depthwise convolution [19]. As expected, self-attention with RPE [37] outperforms vanilla self-attention [47] in accuracy, showing the impact of convolutional inductive biases. GF layer [39], an FFT-based global convolution, outperforms local depthwise convolution [19], and the accuracy becomes comparable to the self-attention when increasing FLOPs (6^{th} row). However, the accuracy of the GF layer is lower than Lambda convolution or RSA due to the limited expressivity. RSA [23] shows the effectiveness of learning structural patterns with high accuracy, but it requires a huge computation budget. LiSA shows the best trade-off between accuracy and FLOPs, achieving the best accuracy among the operators with lower FLOPs. The accuracy of LiSA is even 1.5% higher than the self-attention with increased parameters (3^{rd} row), indicating that the accuracy gain of LiSA does not come from the increased parameters.

Effectiveness of LiSA components. In Tab. 3b, we demonstrate the effectiveness of each component of LiSA. We first compare the way of learning convolutional inductive biases between self-attention and ours (Eq. 4). With the same FLOPs and the number of parameters, our approach that uses

an RPE as multiplicative weights ("SA (Eq. 4)", 4th row in the table) is better than standard self-attention with RPE (2nd row) in accuracy. The accuracy gap between self-attention and ours becomes more clear when we use an independent RPE for each head (3rd row vs. 5th row), indicating our RPEs defined in Eq. 4 are more beneficial than those of standard self-attention. This indicates that our attention containing negative values is potentially more effective for learning spatial features such as gradient information compared to softmax attention. Next, we demonstrate our structure-aware attention variants in the third part of the table. Comparing Eq. 4 with Eq. 5 (4th row vs. 6th row), we validate the effectiveness of learning structural patterns, which improves by 1.5% in top-1 accuracy. The next two rows in the table show the impact of exploiting semantic information of input channels. With the bias term $\boldsymbol{B}^a$ and the Hadamard correlation, our operator improves the top-1 accuracy by 1.0% with negligible additional cost. Lastly, LiSA with FFT approximation dramatically reduces the computational cost ($3\times$) without compromising accuracy.

Effect of global interactions. Tab. 3c studies the influence of local and global interactions. Focusing on the local kernels (first three rows in the table), the larger the kernel, the larger the number of parameters, FLOPs, and accuracies. This shows that a big kernel with more trainable parameters produces better results but at an increased computational cost. However, our global version (last row) allows a larger number of parameters with smaller FLOPs due to FFT, resulting in the best performance among all the variants.

Effect of the number of latent channels D. Tab. 3d summarizes this study using four different values of D. Since D represents the size of the encoded vector that learns structural patterns from the query-key correlation, it is expected that larger values result in better performances. As shown in Tab. 3d, the accuracy is indeed improved with larger values of D, being 16 the best one. We use this value in our attention kernel. Note that the cases over $D = 16$ are not reported since the accuracy becomes saturated.

Efficiency of the LiSA block. In Fig. 4, we demonstrate the efficiency of LiSA in terms of FLOPs, latency, and GPU memory consumption. We measure the performance of a single block ($C = 96$) by varying the number of tokens. For a fair comparison, we replace our LiSA block with other attention blocks [14, 30]. Window attention block [30] efficiently computes local attention with shifted windows, and pooling attention block [14] downsamples the key and value before computing attention for reducing complexity. We set the window size as $N/4$ and the pooling stride as 2×2. Window and pooling attention are more efficient than the standard one in all cases, but their values increase exponentially since their computational complexities are still quadratic in the number of tokens. In comparison, the values of LiSA increase gracefully due to the log-linear complexity, and thus LiSA achieves the best efficiency in all cases. Additional details are presented in the Supplementary Material.

model	attn+conv	FLOPs	#params	top-1
ResNet-50 [18]		4.1 G	26 M	76.1
PVT-S [51]		3.8 G	25 M	79.8
Deit-S [45]		4.6 G	22 M	79.9
RegNetY-4.0GF [36]		4.0 G	21 M	80.0
Swin-Ti [30]		4.5 G	29 M	81.2
T2T-ViT-14 [56]		4.8 G	22 M	81.5
GFNet-H-S [39]		4.6 G	32 M	81.5
CvT-13 [53]	✓	4.5 G	20 M	81.6
CoAtNet-0 [10]	✓	4.2 G	25 M	81.6
MViTv2-T [28]	✓	4.7 G	24 M	82.3
LiSANet-H-S (ours)		2.9 G	19 M	**82.5**
ResNet-101 [18]		7.9 G	45 M	77.4
RegNetY-8.0GF [36]		8.0 G	39 M	81.7
PVT-L [51]		9.8 G	61 M	81.7
Deit-B [45]		17.5 G	86 M	81.8
T2T-ViT-24 [56]		13.8 G	64 M	82.3
CvT-21 [53]	✓	7.1 G	32 M	82.5
GFNet-H-B [39]		8.6 G	54 M	82.9
Swin-S [30]		8.7 G	50 M	83.1
Swin-B [30]		15.4 G	88 M	83.4
CoAtNet-1 [10]	✓	8.4 G	42 M	83.3
CoAtNet-2 [10]	✓	15.7 G	75 M	84.1
MViTv2-B [28]	✓	10.2 G	52 M	**84.4**
LiSANet-H-B (ours)		9.6 G	59 M	**84.4**

Table 4. **Comparison to the state-of-the-art models on ImageNet.** We compare our models with several state-of-the-art architectures. FLOPs (G), the number of parameters (M), top-1, top-5 accuracy (%) on ImageNet validation set are shown. All the models use 224×224 resolution images for training and testing.

model	FLOPs	#params	Mask R-CNN $1\times$ schedule					
			AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
Res-50 [18]	260 G	44 M	38.0	58.6	41.4	34.4	55.1	36.7
PVT-S [51]	245 G	44 M	42.9	65.8	47.1	40.0	62.7	42.9
Twins-S [8]	238 G	44 M	24.0	50.0	41.4	34.4	55.1	36.7
Swin-T [30]	264 G	48 M	42.2	64.6	46.2	39.1	61.6	42.0
ViL-S [59]	218 G	45 M	44.9	67.1	49.3	41.0	64.2	44.1
Focal-T [55]	291 G	49 M	44.8	67.7	49.2	41.0	64.7	44.2
LiSANet-H-S	232 G	38 M	**46.1**	**67.9**	**50.5**	**41.6**	**64.8**	**44.9**

Table 5. **Comparison with other models on COCO object detection**. FLOPs (G), the number of parameters (M), box mAP (AP^b) and mask mAP (AP^m) are shown. Note that FLOPs are measured at resolution 800×1280.

5.3. State-of-the-art Results on ImageNet-1K

In Tab. 4, we compare our two hierarchical models with state-of-the-art approaches including CNNs [18, 36], ViTs [30, 39, 45, 51, 56], and the models containing both convolution and self-attention [10, 28, 53] (marked with ✓ in 'attn+conv' column). In the top half of the table, we present the results of small models that have comparable FLOPs

operator	FLOPs	#params	top-1	top-5
Self-attention [47]	7.36 G	5.82 M	18.0	40.9
Self-attention w/ RPE [37]	7.36 G	5.87 M	24.0	50.0
Depthwise conv ($3 \times 7 \times 7$) [19]	3.75 G	4.82 M	33.0	60.9
GF layer [39]	3.53 G	6.54 M	28.7	40.0
Lambda convolution [2]	26.87 G	6.34 M	34.5	63.1
RSA [23]	72.58 G	23.45 M	34.1	62.7
LiSA (ours)	5.16 G	8.37 M	**38.1**	**67.1**

Table 6. **Comparison with other basic operators on SS-V2**. Top-1, top-5 accuracy (%), FLOPs (G) and the number of parameters (M) are shown.

and the number of parameters. Models that incorporate convolutions into ViTs [10, 53] perform better than the others on the accuracy measure. Our proposed model, LiSANet-H-S, clearly outperforms all the other models in terms of both accuracy and FLOPs. In the case of larger models, grouped in the bottom half of the table, many of the methods show comparable performance. Swin-S & -B [30] shows the effectiveness of hierarchical ViT based on its efficient window attention technique. CoAtNet-1 & -2 [10] obtains an effective result by adopting depthwise convolutions in the early stages and self-attentions in the latter stages. MViTv2-B [28], which shows the highest accuracy, adopts depthwise convolutions and additional residual connections inside the attention blocks for the performance boost. While these models rely on elaborated architecture designs, our proposed model, LiSANet-H-B, is a pure transformer architecture without any complicated modification, achieving state-of-the-art accuracy with less computation.

5.4. Object Detection & Video Action Recognition

Object detection on COCO. To show the generalization ability of LiSA, we conduct object detection experiments on the COCO dataset using the standard Mask R-CNN [17] detection framework with ImageNet-1K pre-trained weights and following the experimental setup in [30] ($1\times$ schedule of 12 epochs). In Tab. 5, we compare the results of small models that have comparable FLOPs and parameters. As expected, LiSANet-H-S shows the best performance among CNN [18] and ViT [30, 51, 55, 59] backbones in AP^b and AP^m, while maintaining its efficiency. Since object detection is a high-resolution computer vision task (*e.g.*, 800×1280), these results demonstrate that LiSA is a better fit for processing a large number of tokens compared to other attention methods [30, 51, 55].

Video action recognition on SS-V2. We perform experiments on video data to show the wide applicability of LiSA. We compare LiSA with other attention operators trained from scratch on SS-V2 [15] in Tab. 6. We sample 8 frames per video, and the rest of the details are the same as in Sec. 5.2. Since structural patterns of videos, *i.e.*, motion patterns, are important cues for recognizing video actions,

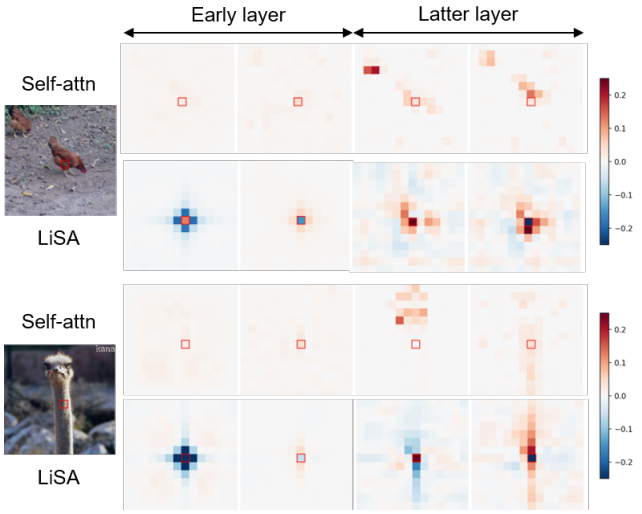


Figure 5. **Attention kernels of self-attention & LiSA.** Attention kernels from different layers and heads are visualized. For each sample, the top row is self-attention and the bottom is LiSA. Note that the red box in the center of each subfigure is the query pixel.

the operators that learn convolutional priors [2, 19, 39, 61] or geometric structures [23] are more effective than their self-attention counterparts [37, 47]. LiSA shows even better performance on video than image, in terms of both accuracy and complexity. While FLOPs of the attention [47, 61] and highly-expressive [2, 23] operators significantly grow due to the increased number of tokens ($T \times H \times W$), LiSA remains efficient due to its log-linear complexity.

5.5. Visualization

In Fig. 5, we visualize both self-attention and LiSA kernels of different layers and heads from isotropic models. As expected, LiSA kernels contain much more diverse patterns compared to self-attention kernels. Self-attention kernels in the early layers often fail to capture relevant context, and those in the latter layers are effective but they usually capture redundant information. Unlike self-attention, LiSA kernels in the early layers focus on encoding local features. Some of these look similar to Sobel filters or Laplacian filters, which are beneficial for learning local structural information. Considering that modern hybrid models [10, 27, 54], which replace self-attention with convolution in early layers obtain an extra accuracy gain, the behavior of LiSA kernels in early layers seems reasonable. Meanwhile, LiSA kernels in the latter layers concentrate on the context relevant to the target object like self-attention does. LiSA, however, generates more diverse shapes of kernels than self-attention, which implies that they aggregate the relevant context and further consider structural patterns inside the context at the aggregation. Therefore, this visualization demonstrates that our structure-aware attention kernel can be more expressive and flexible compared to the self-attention kernel.

6. Conclusion

In this paper, we have presented LiSA, a novel expressive, yet efficient, attention operator that learns rich structural patterns with log-linear complexity. Our comprehensive ablation studies have shown that the ViTs based on LiSA, LiSANets, outperform their counterparts in accuracy and computational complexity. Our LiSANet, which is purely based on LiSA for spatial modeling, has achieved competitive performance on various kinds of visual understanding tasks. We believe that LiSA can be further extended for diverse understanding tasks, including machine translation.

Acknowledgments

This work has been supported in part by the ANR grant AVENUE (ANR-18-CE23-0011), the Junta de Andalucía of Spain (P18-FR-3130 and P20_00430, including European Union funds) and the Ministry of Education of Spain (PID2019-105396RB-I00). We also thank the EuroHPC JU for the GPU computing hours.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 5, 12

[2] Irwan Bello. Lambdanetworks: Modeling long-range interactions without attention. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 4, 6, 8

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 12

[5] Peng Chen. Permuteformer: Efficient relative position encoding for long sequences. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021. 1, 2

[6] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[7] Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, et al. Rethinking attention with performers. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 3

[8] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 7

[9] Ekin Dogus Cubuk, Barret Zoph, Jon Shlens, and Quoc Le. Randaugment: Practical automated data augmentation with a reduced search space. *Proc. Neural Information Processing Systems (NeurIPS)*, 33:18613–18624, 2020. 12

[10] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 7, 8

[11] 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. 2, 5

[12] 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. In *Proc. International Conference on Learning Representations (ICLR)*, 2021. 1, 2, 5

[13] Stéphane d'Ascoli, Hugo Touvron, Matthew L Leavitt, Ari S Morcos, Giulio Biroli, and Levent Sagun. Convit: Improving vision transformers with soft convolutional inductive biases. In *Proc. International Conference on Machine Learning (ICML)*, 2021. 2

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 7

[15] 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. 2, 5, 8

[16] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Proc. Neural Information Processing Systems (NeurIPS)*, 34, 2021. 2

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2017. 6, 8, 12

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 7, 8

[19] 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. 6, 8

[20] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Proc. European Conference on Computer Vision (ECCV)*, 2016. 12

[21] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc Van Gool. Dynamic filter networks. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2016. 2

[22] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive

transformers with linear attention. In *Proc. International Conference on Learning Representations (ICLR)*, 2020. 2

[23] Manjin Kim, Heeseung Kwon, Chunyu Wang, Suha Kwak, and Minsu Cho. Relational self-attention: What's missing in attention for video understanding. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 3, 4, 6, 8

[24] Heeseung Kwon, Manjin Kim, Suha Kwak, and Minsu Cho. Learning self-similarity in space and time as generalized motion for video action recognition. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 3, 4

[25] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021. 2

[26] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2, 4

[27] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning. In *Proc. International Conference on Learning Representations (ICLR)*, 2022. 1, 2, 5, 8, 12, 13

[28] 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 *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2, 7, 8, 12, 13

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. European Conference on Computer Vision (ECCV)*, 2014. 2, 5

[30] 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 *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 5, 6, 7, 8, 12

[31] Antoine Liutkus, Ondrej Cifka, Shih-Lun Wu, Umut Simsekli, Yi-Hsuan Yang, and Gael Richard. Relative positional encoding for transformers with linear complexity. In *Proc. International Conference on Machine Learning (ICML)*, 2021. 1, 2

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 12

[33] Shengjie Luo, Shanda Li, Tianle Cai, Di He, Dinglan Peng, Shuxin Zheng, Guolin Ke, Liwei Wang, and Tie-Yan Liu. Stable, fast and accurate: Kernelized attention with relative positional encoding. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 1, 2, 3

[34] Ningning Ma, Xiangyu Zhang, Jiawei Huang, and Jian Sun. Weightnet: Revisiting the design space of weight networks. In *Proc. European Conference on Computer Vision (ECCV)*, 2020. 2

[35] Zhen Qin, Weixuan Sun, Hui Deng, Dongxu Li, Yunshen Wei, Baohong Lv, Junjie Yan, Lingpeng Kong, and Yiran Zhong. cosformer: Rethinking softmax in attention. *arXiv preprint arXiv:2202.08791*, 2022. 1, 2, 3

[36] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 7

[37] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*, 2019. 1, 2, 6, 8

[38] 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. 3

[39] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 2, 5, 6, 7, 8, 12, 14

[40] 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. 3

[41] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proc. Winter Conference on Applications of Computer Vision (WACV)*, 2021. 2

[42] Gilbert Strang. A proposal for toeplitz matrix calculations. *Studies in Applied Mathematics*, 74(2):171–176, 1986. 3

[43] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[44] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. IEEE conference on computer vision and pattern recognition*, 2016. 12

[45] 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 (ICML)*, 2021. 1, 2, 5, 6, 7, 12

[46] Ashish Vaswani, Prajit Ramachandran, Aravind Srinivas, Niki Parmar, Blake Hechtman, and Jonathon Shlens. Scaling local self-attention for parameter efficient visual backbones. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[47] 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. 2, 4, 6, 8

[48] 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. 4

[49] 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. 6, 12

[50] Sinong Wang, Belinda Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 1, 2, 3

[51] 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 *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 7, 8

[52] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 12

[53] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 2, 7, 8

[54] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 2, 8

[55] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *Proc. Neural Information Processing Systems (NeurIPS)*, 2021. 7, 8

[56] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 1, 2, 7

[57] 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 *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019. 12

[58] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proc. International Conference on Learning Representations (ICLR)*, 2018. 12

[59] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2021. 7, 8

[60] Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, and Tomas Pfister. Aggregating nested transformers. *arXiv preprint arXiv:2105.12723*, 2021. 2

[61] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 4, 8

[62] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proc. AAAI Conference on Artificial Intelligence (AAAI)*, 2020. 12

A. Implementation details

A.1. Architecture Details

LiSA. In addition to the details included in the main paper, we provide pseudo-code of structure-aware attention (Eq. 4 & Eq. 5 in the main paper) and LiSA in Fig. 6 and Fig. 7, respectively. The notation of multi-head is omitted for clarity and simplicity. As shown in the pseudo-code, we effectively reduce the computational complexity with the FFT approximation. In actual implementation, we employ 2D RFFT/IRFFT for images and 3D variant for videos.

LiSANet. A detailed overview of our proposed architectures is shown in Tab. 7. LiSANet-I-T is composed of a single stage following the traditional ViT guidelines [1, 45]. The number of tokens is constant in this model: 14×14. Focusing on the blocks, we have an initial patch embedding layer with a stride of 16 pixels and then, 12 LiSA blocks with an embedding size of 192. On the other hand, our hierarchical models follow the guidelines proposed in [27, 30, 39]. Both models (LiSANet-H-S and LiSANet-H-B) are composed of the same four stages with different token sizes and numbers of tokens. In the earlier stages, the token size is larger producing a smaller number of tokens. We adopt the overlapping patch embedding strategy [52] for hierarchical models. The number of LiSA blocks differs in both models, with LiSANet-H-B the larger model employing a larger number of blocks (16 vs. 24 blocks).

A.2. Experimental Setup

Image classification. Our models are trained with AdamW [32] with a weight decay of 0.05 and a learning rate of $\frac{0.0005}{512} \cdot batch\_size$ with a cosine decay scheduler and 20 warm-up epochs. Our isotropic model (LiSANet-I-T) is trained for 150 epochs and hierarchical models are trained for 300 epochs. Following the training recipe proposed in [30], we apply several regularization techniques such as Mixup [58], Cutmix [57], label smoothing [44] and stochastic depth [20]. The stochastic depth strategy is applied only to the hierarchical models with a probability of 0.1 and 0.4 for the LiSANet-H-S and LiSANet-H-B models, respectively. In addition, we also apply several data augmentation techniques like Rand-Augment [9], random erasing [62], and repeated augmentation. Note that all these hyperparameter values and data-augmentation techniques are selected following the training recipes of the previous works [30, 45].

Object detection. We adopt standard Mask R-CNN [17] detection frameworks, and ImageNet-1K pre-trained model (LiSANet-H-S) are utilized as backbones. weights for fine-tuning. We use a $1\times$ schedule (12 epochs) with total batch size 16, and follow the same recipe as in [30]. The code is mainly based on mmdetection [4]. For training, the shorter side of the image is resized to 800 pixels while keeping the longer side no more than 1333 pixels. AdamW [32] with a weight decay of 0.05 is adopted as an optimizer, and the initial learning rate is set as 0.0001. Stochastic depth rate is set as 0.1.

```
# B: batches, N: tokens, C: channels, D: latent_channels
def structure_aware_attn_Eq4(input, e):
# input shape: [B,N,C], e shape: [2N-1]
    qkv = linear_proj(input, channels=3C) # shape: [B,N,3C]
    query,key,value = split(qkv, [C,C,C], dim=-1)
    # query,key,value shape: [B,N,C]
    query = L2norm(query) # shape: [B,N,C]
    key = L2norm(key) # shape: [B,N,C]

    R = Toeplitz(e) # shape: [N,N]
    attn = einsum(query,key, 'BNC,BMC->BNM') # shape: [B,N,N
        ]
    attn_R = attn * R # shape: [B,N,N]

    out = einsum(attn_R,value,'BNM,BMC->BNC') # shape: [B,N,
        C]
    out = linear_proj(out, channels=C) # shape: [B,N,C]
    return out

def structure_aware_attn_Eq5(input, Ea, Eb, Bb):
# input shape: [B,N,C], Ea,Eb shape: [2N-1,D], Bb shape: [C,
     D]
    qkv = linear_proj(input, channels=3C) # shape: [B,N,3C]
    query,key,value = split(qkv, [C,C,C], dim=-1)
    # query,key,value shape: [B,N,C]
    query = L2norm(query) # shape: [B,N,C]
    key = L2norm(key) # shape: [B,N,C]

    Ra = Toeplitz(Ea) # shape: [N,N,D]
    Rb = Toeplitz(Eb) # shape: [N,N,D]
    K_Ra = einsum(key,Ra,'BMC,NMD->BNCD') # shape: [B,N,C,D]
    Rb_V = einsum(Rb,value, 'NMD,BMV->BNVD') # shape: [B,N,C
        ,D]
    Rb_V_Bb = Rb_V + Bb # shape: [B,N,C,D]

    out = einsum(query,K_Ra,Rb_v_Bb,'BNC,BNCD,BNVD->BNV') #
         shape: [B,N,C]
    out = linear_proj(out, channels=C) # shape: [B,N,C]
    return out
```

Figure 6. **Pseudo-code for structure-aware attention.** We describe the way of learning convolutional inductive biases in structure-aware attention (Eq. 4) and its basic form (Eq. 5) presented in Sec. 4.1 of the main paper.

Video action recognition. Our video models are trained with AdamW [32] with a weight decay of 0.05 and $\frac{0.0002}{32} \cdot batch\_size$ with a cosine decay scheduler and 5 warm-up epochs. Total epoch is set as 60 epochs, and we apply several regularization techniques following [1, 27, 28]. We adopt the uniform sampling strategy [49] for training and a single crop inference for testing.

Code. Our source code is included for reproducibility.

B. Additional Ablations

Comparison between LiSANet & (conv+attn) models. In this ablation study, we provide a new comparison between the state-of-the-art (conv+attn) models [27, 28] and LiSANet with hierarchical architecture. Both models [27, 28] incorporate self-attention with convolution operators in each block to increase expressivity. Since self-attention requires a large amount of memory in higher reso-

	Output Size	LiSANet-I-T	LiSANet-H-S	LiSANet-H-B
Stage1	$\frac{H}{4} \times \frac{W}{4}$	-	Overlap Patch Embed↓4 LiSA Block (64) × 1	Overlap Patch Embed↓4 LiSA Block (96) × 2
Stage2	$\frac{H}{8} \times \frac{W}{8}$	-	Overlap Patch Embed↓2 LiSA Block (128) × 2	Overlap Patch Embed↓2 LiSA Block (192) × 3
Stage3	$\frac{H}{16} \times \frac{W}{16}$	Patch Embed↓16 LiSA Block (192) × 12	Overlap Patch Embed↓2 LiSA Block (256) × 11	Overlap Patch Embed↓2 LiSA Block (384) × 16
Stage4	$\frac{H}{32} \times \frac{W}{32}$	-	Overlap Patch Embed↓2 LiSA Block (512) × 2	Overlap Patch Embed↓2 LiSA Block (768) × 3
Classifier		Global Average Pooling, Linear		

Table 7. **Details of LiSANet variants.** Patch Embed↓n denotes a patch embedding layer that downsamples features with a stride n.

```
# B: batches, N: tokens, C: channels, D: latent_channels
def LiSA(input, Wa, Wb, Ba, Bb):
# input shape: [B,N,C], Wa shape: [N,C,D], Wb shape: [N,D],
      Ba,Bb shape: [C,D]
    qkv = linear_proj(input, channels=3C) # shape: [B,N,3C]
    query,key,value = split(qkv, [C,C,C], dim=-1)
    # query,key,value shape: [B,N,C]
    query = L2norm(query) # shape: [B,N,C]
    key = L2norm(key) # shape: [B,N,C]

    K_fft = rfft(key, dim=1) # shape: [B,N//2+1,C]
    Wa_fft = rfft(Wa, dim=0) # shape: [N//2+1,C,D]
    K_Wa = einsum(K_rfft, Wa_rfft, 'BMK,MKD->BMKD') # shape:
          [B,N//2+1,C,D]
    K_Wa = irfft(K_Wa, dim=1) # shape: [B,N,C,D]
    K_Wa_Ba = K_Wa + Ba # shape: [B,N,C,D]

    V_fft = rfft(value, dim=1) # shape: [B,N//2+1,C]
    Wb_fft = rfft(Wb, dim=0) # shape: [N//2+1,D]
    V_Wb = einsum(V_rfft, Wb_rfft, 'BMV,MD->BMVD') # shape:
         [B,N//2+1,C,D]
    V_Wb = irfft(V_Wb, dim=1) # shape: [B,N,C,D]
    V_Wb_Bb = V_Wb + Bb # shape: [B,N,C,D]

    out = einsum(query,K_Wa_Ba,V_Wb_Bb,'BNK,BNKD,BNVD->BNV')
          # shape: [B,N,C]
    out = linear_proj(out, channels=C) # shape: [B,N,C]
    return out
```

Figure 7. **Pseudo-code for LiSA.** We describe the final form of LiSA described in Sec. 4.3 of the main paper.

lutions, Uniformer [27] only uses depthwise convolutions in earlier layers and MViTv2 [28] downsamples features with convolutions before applying attention. For a fair comparison, we use the same configuration (shown in Tab 8a) for all the models. Tab. 8b summarizes the results on ImageNet, which show that the accuracy of LiSANet-H-T is higher than the state-of-the-art (conv+attn) models with comparable FLOPs and params. This is impressive since we do not apply any modification for our block while other models rely on elaborated architecture designs and handcrafted schemes.

Model	#Blocks	#Channels (#heads)
MViTv2-H-T	[1, 2, 5, 2]	[64 (4), 128 (8), 256 (16), 512 (32)]
Uniformer-H-T	[1, 2, 5, 2]	[64 (4), 128 (8), 256 (16), 512 (32)]
LiSANet-H-T	[1, 2, 5, 2]	[64 (4), 128 (8), 256 (16), 512 (32)]

(a) **Model configuration of hierarchical-tiny models.**

model	FLOPs	#params	top-1
MViTv2-H-T	2.1 G	12.0 M	79.3
Uniformer-H-T	1.7 G	11.9 M	79.7
LiSANet-H-T (D=8)	1.8 G	12.7 M	80.4
LiSANet-H-T (D=8) w/ conv	1.8 G	12.7 M	80.9

(b) **Performance comparison among hierarchical-tiny models.** Note that D is the number of latent channels of LiSA. Top-1, accuracy (%), FLOPs (G), and the number of parameters (M) are shown.

Table 8. **Comparison between (conv+attn) models & ours.**

For example, LiSANet-H-T obtains an additional gain by using the depthwise convolution scheme of [27]. These results show that the models based on LiSA outperform the (conv+attn) models with hierarchical architecture. We use the official source code of Uniformer [27] and MViTv2 [28] for implementing their blocks. For Uniformer, we only apply depthwise convolutions for the early 2 stages, following their paper. For MViTv2, we downsample features with stride 2 for the early 2 stages, and do not downsample for the other stages.

Fine-tuning on higher resolutions. We have verified that fine-tuning LiSANet on higher resolutions can boost image recognition accuracy. Tab. 9 summarizes the results of LiSANet-I-T on ImageNet. LiSANet obtains a 2.5% gain when we use 384×384 resolution. While MLP-mixer models are hard to adapt to higher resolutions since they process a fixed number of tokens, LiSANet can be easily

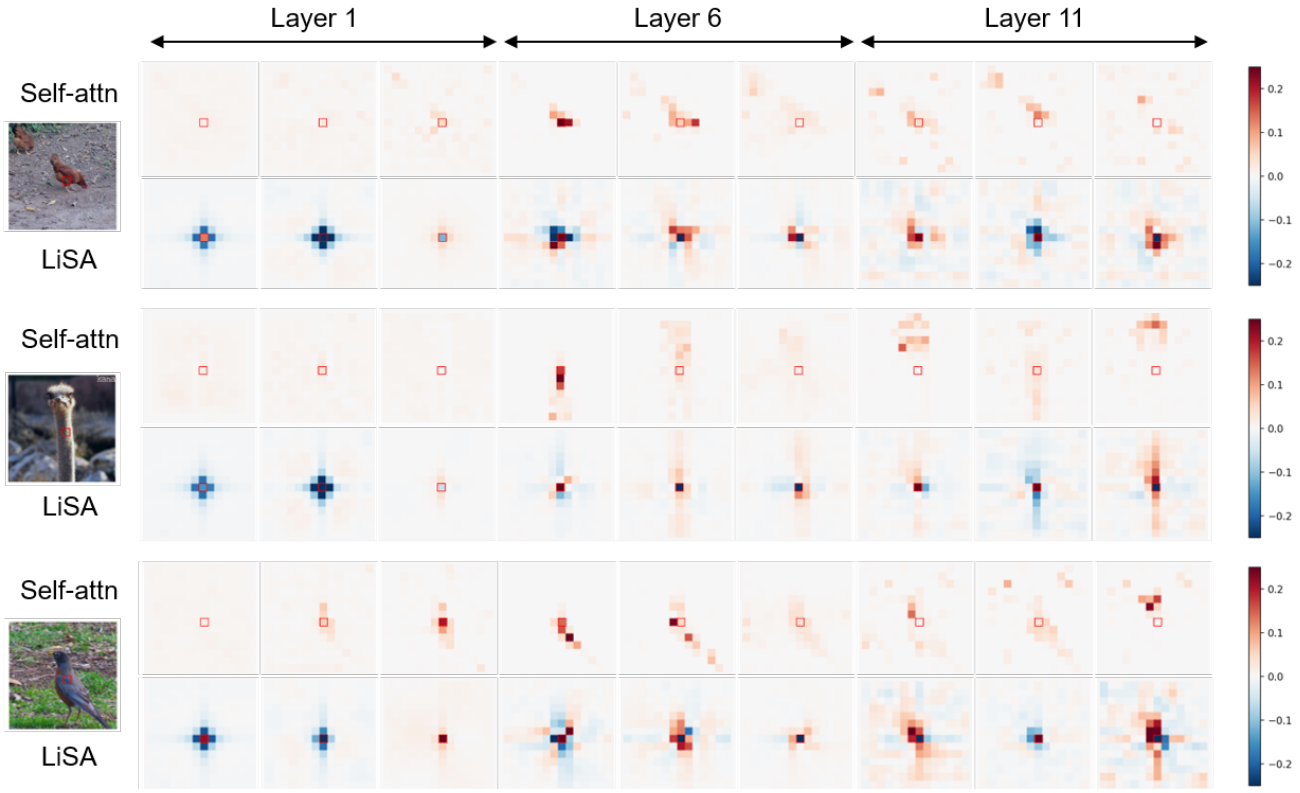
model	Image size	FLOPs	#params	top-1
LiSANet-I-T	224×224	1.21 G	6.36 M	74.9
LiSANet-I-T	384×384	3.62 G	7.82 M	77.4

Table 9. **Fine-tuning to higher resolutions on ImageNet**. Image size, Top-1, accuracy (%), FLOPs (G) and the number of paramaters (M) are shown.

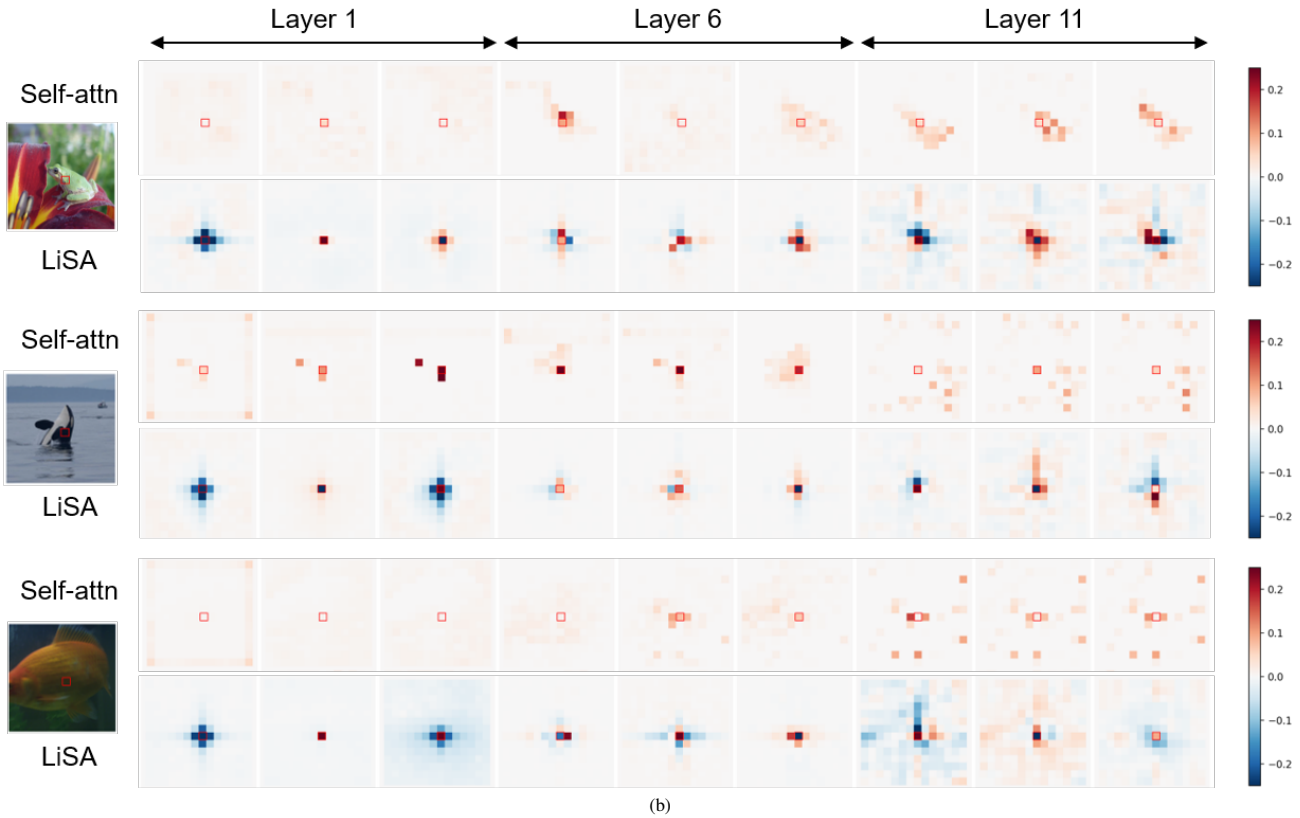
interpolated to higher resolutions due to the property of Discrete Fourier transform, where each element of the time (*i.e.* spatial) domain is a sampling of a continuous spectrum in the frequency domain. Since the circular embeddings $\boldsymbol{W}^a$, $\boldsymbol{W}^b$ can be considered as samplings of continuous spectrums, changing the resolution is equal to changing the sampling interval of spectrums [39]. Thus, LiSA can be adapted to higher resolutions by simple interpolation.

C. Visualization

In Fig. 8, we additionally visualize both self-attention and LiSA kernels of different layers and heads from isotropic models.



(a)



(b)

Figure 8. **Attention kernels of self-attention & LiSA.** Attention kernels from different layers and heads are visualized. For each sample, the top row is self-attention and the bottom is LiSA. Note that the red box in the center of each subfigure is the query pixel.