Non-Local Neural Networks with Grouped Bilinear Attentional Transforms

Lu Chi[1,2], Zehuan Yuan[2], Yadong Mu[1*], Changhu Wang[2]
[1]Peking University, Beijing, China, [2]ByteDance AI Lab, Beijing, China
{chilu,myd}@pku.edu.cn, {yuanzehuan,wangchanghu}@bytedance.com

Abstract

Modeling spatial or temporal long-range dependency plays a key role in deep neural networks. Conventional dominant solutions include recurrent operations on sequential data or deeply stacking convolutional layers with small kernel size. Recently, a number of non-local operators (such as self-attention based [57]) have been devised. They are typically generic and can be plugged into many existing network pipelines for globally computing among any two neurons in a feature map. This work proposes a novel non-local operator. It is inspired by the attention mechanism of human visual system, which can quickly attend to important local parts in sight and suppress other less-relevant information. The core of our method is learnable and data-adaptive bilinear attentional transform (BA-Transform), whose merits are three-folds: first, BA-Transform is versatile to model a wide spectrum of local or global attentional operations, such as emphasizing specific local regions. Each BA-Transform is learned in a data-adaptive way; Secondly, to address the discrepancy among features, we further design grouped BA-Transforms, which essentially apply different attentional operations to different groups of feature channels; Thirdly, many existing non-local operators are computation-intensive. The proposed BA-Transform is implemented by simple matrix multiplication and admits better efficacy. For empirical evaluation, we perform comprehensive experiments on two large-scale benchmarks, ImageNet and Kinetics, for image / video classification respectively. The achieved accuracies and various ablation experiments consistently demonstrate significant improvement by large margins.

1. Introduction

This era has witnessed the vigorous development of deep neural networks, with significant empirical success in a plethora of important real-life vision tasks [28, 36, 45, 56]. The neural architectures of convolutional networks are still undergoing rapid evolution. Much of recent endeavor has been devoted to designing deeper [48, 17] or wider [61, 14] network architectures, or more effective atomic convolutional operators [6, 20]. The main interest of this work is modeling long-range spatial [57] or temporal [56] dependencies in deep convolutional networks. To this end, classic neural networks, such as VGG-Net [48] or ResNet [17], mostly adopt a scheme of deeply stacking many convolutional layers with small receptive fields (*e.g.*, 3×3 kernels in ResNet [17] and $3 \times 3 \times 3$ spatio-temporal kernels in C3D [52]).

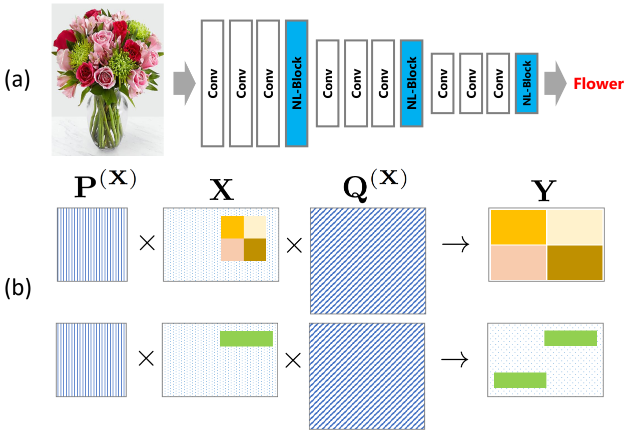

Figure 1: (a) Typical architecture of neural networks with non-local operators, where non-local neural blocks (highlighted in blue) are sparsely added into original network pipeline to instantaneously achieve large receptive fields. (b) Illustration of our proposed *bilinear attentional transform* (BA-Transform). With properly-learned matrices $\mathbf{P}^{(\mathbf{X})}, \mathbf{Q}^{(\mathbf{X})}$ in the transformation formula $\mathbf{Y} = \mathbf{P}^{(\mathbf{X})}\mathbf{X}\mathbf{Q}^{(\mathbf{X})}$, BA-Transform can conduct a variety of operations (selective zooming and dispersing to distant positions as shown in this sub-figure) on attended features. The super-scripts in $\mathbf{P}, \mathbf{Q}$ emphasize their dependence on $\mathbf{X}$.

One of current research fronts regarding effectively enlarging neural receptive fields is to sparsely insert non-local operators into an existing network pipeline. An illustration of such a architecture is shown in Figure 1(a). The main challenge for sparse insertion of non-local operators is their

[*] Corresponding author.

high time complexity. For example, in [57], Wang et al. proposed a seminal non-local neural operator based on self-attention, which requires $\mathcal{O}(N^2)$ (N counts all locations in the feature map) vector multiplication. Arguably, the scheme in Figure 1(a) can strike a good tradeoff between complexity and performance.

Our method is inspired by human visual perception. For the optical signal received at the retina, human eyes are believed to conduct both a bottom-up procedure for obtaining low-level abstraction, and top-down attentional operations that quickly locate most interesting parts from the entire field of visual scene. The eyes will focus on the attended regions for further inspection. Such attention mechanism is clearly more effective for visual understanding in comparison with blind processing. This has inspired substantial efforts on devising various powerful attentional neural networks [41, 16, 25] used for visual analysis and generation. Figure 1(b) illustrates our proposed *bilinear attentional transform* (BA-Transform). It processes an input feature map $\mathbf{X}$ to obtain a new $\mathbf{Y}$ via the formula $\mathbf{Y} \leftarrow \mathbf{P}^{(\mathbf{X})}\mathbf{X}\mathbf{Q}^{(\mathbf{X})}$, where all variables are matrices and their sizes can be inferred from context. Motivating our advocate of the BA-Transform we consider two desiderata:

Firstly, human are remarkably capable of capturing complex attention patterns, which can be accomplished even at a single glimpse. The attended parts in visual field can be spatially or temporally disjoint (*e.g.*, in a video of boxing action, two boxers shall both be paid attention to, even if they might be distant to one another), or highly complex. It is thus crucial to enforce that neural attentional units have sufficiently powerful modeling capability. Our proposed BA-Transform supports a large variety of operations on the attended image or video parts albeit its simplicity, including numerous affine transformation (selective scaling, shift, rotation, cropping etc.), suppressing / strengthening local structure or even global reasoning, as partially illustrated in Figure 1(b).

Secondly, bilinear matrix multiplication is amenable to efficient differential calculation. In practice, we can add neural blocks that implement BA-Transform into an existing network, and jointly train all neural layers in an end-to-end fashion. Top-down supervision can be gradually back-propagated to shallow layers and enforce the consistency between learned attentions and top-down supervision. When plugged in neural architecture with skip connections, BA-Transforms tend to learn complementary attentions at different insertion, corroborated by empirical studies in our experimental section.

The proposed BA-Transform naturally inherits almost all advantages of its precedent works [57, 24]: capturing long-range interactions via directly connecting all locations, seamlessly combined with many existing neural networks, and elevating performance even being inserted only very few times. In addition to all above, we also propose a channel-grouping scheme. This enforces that a same attention pattern is shared within a group. This explicitly enables that multiple heterogeneous attention patterns can be simultaneously learned for the same feature map in the neural networks.

The rest of this paper is organized as following: We first review related work in Section 2 and detail the proposed block design in Sections 3 and 4. Section 5 showcases the effectiveness of this global operator by conducting experiments in two tasks, including image recognition and video classification.

2. Related Work

Neural Attention. Human visual system is known to have high resolution at the fovea and low resolution in the periphery [46]. Attention mechanism bridges this gap, and inspires much of recent development in the computer vision domain. Successful applications of attention in vision tasks include image classification [55, 59], image generation [16], segmentation [4, 12], action recognition [57, 37] etc. In neural networks, attention of each pixel can be softly estimated (*i.e.*, soft attention) and hardly classified into 0 or 1 (*i.e.*, hard attention). A popular treatment to obtain hard attention is learning to crop image regions using pre-trained detector [58, 34] or policies trained via reinforcement learning [41]. Our proposed method falls into the category of soft attention [25, 13, 54, 57, 9], where the attention units are typically differentiable and trained by gradient back-propagation. Some classic methods treat attention as bottom-up saliency [55, 9]. Recent advances have increasingly emphasized the drive of top-down supervision. Another taxonomy is based on whether attention is learned locally or globally. Deformable convolution [10, 64] seeks for local interesting pixels. self-attention based attention methods [57, 24] globally connect all locations. Among globally-learned attention models, some accomplishes in one shot (*e.g.*, STN [25] and our proposed method), and others recurrently reinforce the model [41, 16, 27].

Network Architecture. Recent years have observed a blossom of novel neural networks. Classic networks (*e.g.*, VGG [48]) favor convolutions with small kernels. The global interaction among all locations are obtained by deeply stacking many convolutional layers and utilizing skip connections (*e.g.*, ResNet [17] and DenseNet [23]). More complex kernels and networks can be automatically found via neural architecture search (NAS) [1, 35, 5]. It has also been intensively explored to use mixed or large receptive fields. For instance, Inception [50, 51] and SKNet [31] use an ensemble of differently-sized kernels. Most relevant works to ours are non-local neural networks [57, 7, 24] and GloRe unit [8], which can globally disperse any local message. In our proposed BA-Transform, group-wise

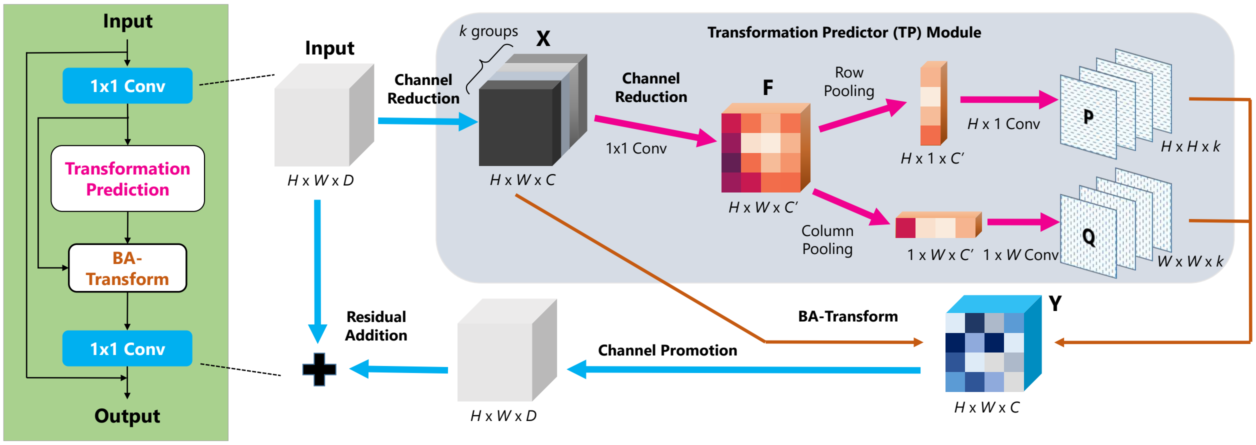

Figure 2: Design of our proposed BAT-Block. The left panel draws the computational pipeline of our proposed BAT-Block. The right panel shows more details, particularly the *transformation predictor*. To save space, batch normalization, ReLU and our proposed *row / column normalization* are not shown here. More explanation is found in Section 3.2.

attention is adopted, similar to group or depth convolutions used in MobileNet [19, 47, 18], ShuffleNet [63, 39] and IGCNet [49].

Image / Video Classification. A majority of neural networks [17, 55, 23, 10, 64, 6] are developed for tackling image recognition. Early development of deep network based video classification directly borrows pre-trained image models. Features are first extracted from frame-based video snippets, and fused either by recurrent aggregation [60] or naive pooling [56]. Karpathy et al. in [26] first introduced 3D convolutional operation to this task. The follow-up work of I3D [2] proposed a better network initialization by inflating pre-trained 2D filters to 3D. To expedite computing spatio-temporal convolutions, some recent methods decoupled 3D convolution and sequentially execute along the spatial and temporal dimensions [43, 53, 62]. TSM [32] presented an efficient method to model temporal information by shifting in timescale.

3. The Proposed Approach

We first introduce a general definition of *bilinear attentional transform* (BA-Transform) in Section 3.1. In practice, BA-Transform is wrapped into a neural block which can be dropped into any arbitrary CNN architectures. The details, including various engineering considerations, can be found in Section 3.2.

3.1. Formulation

Let $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ be a feature map with C channel. H, W denote the sizes along two spatial dimensions respectively. Our goal is to design an operator which transforms an input $\mathbf{X}$ into a same-sized output $\mathbf{Y} \in \mathbb{R}^{H \times W \times C}$. Critically, each element of $\mathbf{Y}$ is expected to relate to multiple features in $\mathbf{X}$ (*i.e.*, the non-local property of capturing long-range dependence), and contain all key information of $\mathbf{X}$ (*i.e.*, $\mathbf{Y}$ learns the *attention* of $\mathbf{X}$). Different from existing self-attention [54] based non-local operators [57] and its approximate accelerated variants [7, 24], we here utilize bilinear matrix product as below,

$$\mathbf{Y}_c = \mathbf{P}^{(\mathbf{X})} \cdot \mathbf{X}_c \cdot \mathbf{Q}^{(\mathbf{X})}, \tag{1}$$

where the sub-script c denote an $H \times W$ slice of $\mathbf{X}$ or $\mathbf{Y}$ along the c-th channel. $\mathbf{P}^{(\mathbf{X})}$ and $\mathbf{Q}^{(\mathbf{X})}$ are transformation matrices to be learned, with the size of $H \times H$ and $W \times W$ respectively. Their super-scripts imply that both are dependent on the input data $\mathbf{X}$, thus data-adaptive. For brevity, hereafter the super-scripts will be omitted.

Let us give some intuitive explanation for Eqn. (1). Once properly learned, according to the theory of elementary matrix [40], the left-multiplier $\mathbf{P}$ can be represented as the product of three kinds of elementary matrices that interchanging rows, multiplying row by a scalar, or adding a multiple of row to another row, respectively. Likewise, the learned right-multiplier $\mathbf{Q}$ defines a series of elementary column operations to $\mathbf{X}$. The joint function of $\mathbf{P}, \mathbf{Q}$ enables a large spectrum of transformation of $\mathbf{X}$, including selective zooming, suppressing / enhancing specific sub-matrices of $\mathbf{X}$ etc. Two special cases are found in Figure 1(b).

The previous work STN [25] aims to learn invariance to translation, scale, rotation and more generic warping by affine transformation. Our work differently pay varying attention weights on the feature map. STN is also limited by the number of spatial transformers, while ours can capsule several attentional operations in a pair of $\mathbf{P}, \mathbf{Q}$. Additionally, our proposed BA-Transform does not suffer from the black border problem [25]. When compared with self-attention based operators [57], our method tends to exhibit superior performance, supposedly owing to effectively modeling complex attentional patterns.

3.2. Basic 2D BAT-Block for Image Tasks

We term a neural block that wraps and implements an instance of BA-Transform as BAT-Block. The architecture of BAT-Block is shown in Figure 2. Following by the common practice of residual block [17, 55], we add two 1 × 1 convolutions into the BAT-Block. Any input feature maps will first go through channel reduction at the beginning and channel promotion at exit. Residual connection is also adopted. There are two key procedures in the BAT-Block, namely *transformation predictor* and BA-Transform respectively. The former reads $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ (the output of the first 1×1 convolution) and predicts two parametric matrices $\mathbf{P} \in \mathbb{R}^{H \times H}$ and $\mathbf{Q} \in \mathbb{R}^{W \times W}$ conditioned on $\mathbf{X}$. The latter is previously described in Eqn. 1.

Empirically, it is observed that the matrix norms of $\mathbf{P}, \mathbf{Q}$ tend to explode after a few epoches of gradient back-propagation. For the consideration of numerical stability, we enforce all elements in $\mathbf{P}, \mathbf{Q}$ are non-negative and normalize them in an L_1 sense by rows or columns respectively, as below:

$$\mathbf{P}_{i,j} \leftarrow \frac{\mathbf{P}_{i,j}}{\sum_{k=1}^{H} \mathbf{P}_{i,k}}, \quad \mathbf{Q}_{i,j} \leftarrow \frac{\mathbf{Q}_{i,j}}{\sum_{k=1}^{W} \mathbf{Q}_{k,j}}, \tag{2}$$

where i, j, k collaboratively compose valid indices for accessing an individual element in $\mathbf{P}$ and $\mathbf{Q}$.

Now we elaborate on two core operations in transformation predictor:

1. Feature compression via channel reduction and row / column pooling. The reduced feature map $\mathbf{X}$ (from some D channels to C) is often sill too large for computing over the global receptive field. To further reduce the time complexity, we further reduce the number of channels in $\mathbf{X}$ via a 1×1 convolutional layer, followed by a batch normalization layer and ReLU. The obtained representation is denoted as $\mathbf{F} \in \mathbb{R}^{H \times W \times C'}$, where $C' \ll C$.

We expect that each individual element in $\mathbf{P}, \mathbf{Q}$ is estimated globally conditioned on $\mathbf{F}$. To this end, it is necessary to extract some global, compact representation from $\mathbf{F}$, particularly when $\mathbf{F}$ still has high spatial resolution.

Inspired by the recently proposed *corner pooling* [29], a seemingly effective solution is to squeeze $\mathbf{F}$ in either row-wise or column-wise manner. To be specific, we devise *row / column pooling* on $\mathbf{F}$. Let $\mathbf{F}^{rp} \in \mathbb{R}^{H \times 1 \times C'}, \mathbf{F}^{cp} \in \mathbb{R}^{1 \times W \times C'}$ be the output of row pooling or column pooling, respectively. These operations are defined as below (channel-wise index is omitted for brevity):

$$\mathbf{F}_i^{rp} = \max \left\{ \mathbf{F}_{i,j} \mid 1 \leqslant j \leqslant W \right\}, \tag{3}$$

$$\mathbf{F}_j^{cp} = \max \left\{ \mathbf{F}_{i,j} \mid 1 \leqslant i \leqslant H \right\}. \tag{4}$$

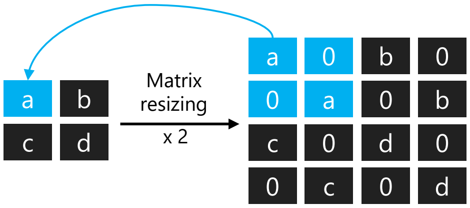

Figure 3: Illustration of matrix resizing. In this example $H/s_h = W/s_w = 2$.

The procedure is intuitively illustrated in the right panel of Figure 2.

2. Estimate $\mathbf{P}, \mathbf{Q}$ with full convolution. The next step estimates $\mathbf{P}$ and $\mathbf{Q}$ from the compact pooled features $\mathbf{F}^{rp}, \mathbf{F}^{cp}$, respectively. To this end, we simply adopt learnable convolutional kernels. Importantly, to ensure global information is utilized in learning $\mathbf{P}, \mathbf{Q}$, kernels with global receptive fields that access all pooled features are used. For example, $H \times 1$ kernels for obtaining $\mathbf{P}$ and $1 \times W$ kernels for $\mathbf{Q}$. Different from Squeeze-and-Excitation (SE) operation in [22], our method can largely preserve spatial information which is crucial for predicting $\mathbf{P}$ or $\mathbf{Q}$. SE does not take spatiality into account.

3.3. Improving 2D BAT-Block

Our practical investigation also reveals the particular effectiveness of two techniques, in the sense of either acceleration or accuracy elevation.

Block based matrix estimation. Predicting full resolution of $\mathbf{P}, \mathbf{Q}$ from the pooled features still imply tremendous parameters. As shown later in our experiments, over-parameterized BA-Transform can adversely affects the generalization performance. Inspired by the super-pixel idea widely used in image analysis, we implement a variant of BA-Transform that harnesses a block-based matrix form. Specifically, the feature map $\mathbf{F}$ is uniformly divided into $s_h \times s_w$ blocks along its two spatial dimensions, where s_h, s_w are some integers (*e.g.*, 7) typically divisible by H, W, respectively. Row or column pooling is conducted by the following updated formula:

$$\mathbf{F}_i^{rp} = \max\{\mathbf{F}_{k,j} | i \leqslant k < i \times \frac{H}{s_h}, 1 \leqslant j \leqslant W\}, \tag{5}$$

$$\mathbf{F}_j^{cp} = \max\{\mathbf{F}_{i,k} | j \leqslant k < j \times \frac{W}{s_w}, 1 \leqslant i \leqslant H\}. \tag{6}$$

This leads to much smaller pooled features $\mathbf{F}^{cp} \in \mathbb{R}^{s_h \times 1 \times C'}$ and $\mathbf{F}^{rp} \in \mathbb{R}^{1 \times s_w \times C'}$. Correspondingly, the sizes of $\mathbf{P}, \mathbf{Q}$ are shrunk to $s_h \times s_h$ or $s_w \times s_w$, respectively. This requires significantly fewer parameters. A normal routine as described in Section 3.2 is called to learn the BA-Transform. Afterwards, we adopt a simple strategy to restore the full-resolution of $\mathbf{P}, \mathbf{Q}$. As depicted in Figure 3, when the element-block correspondence is determined between two matrices, each element in the low-resolution matrix is coped to the respective diagonal locations in the high-resolution matrix. Other cases that $H/s_h = W/s_w \neq 2$ can be likewise derived.

Channel-grouping multi-head attentions. Following [54], we adopt the implementation of multi-head attention. Specifically, $\mathbf{X}$ is uniformly split into $k > 1$ groups along the channel dimension. The idea is illustrated in Figure 2. For each group, a unique pair of $(\mathbf{P}, \mathbf{Q})$ will be learned and utilized in Eqn. (1), k pairs of $(\mathbf{P}, \mathbf{Q})$ in total. This arguably enhances the ability of tackling complex attention patterns. Our experiments demonstrate this simple idea of channel grouping.

3.4. Spatio-Temporal 3D Block for Video Tasks

BAT-Block can be trivially extended to high dimensions. In video tasks, a popular treatment is to stack features from consecutive frames. The input variables are thus 4D tensors, *e.g.*, $\mathbf{X} \in \mathbb{R}^{T \times H \times W \times C}$ with T being the number of stacked frames. Let $\mathbf{X}_{<t,*,*,*>} \in \mathbb{R}^{1 \times H \times W \times C}$ be a time-indexed slice of $\mathbf{X}$. To extend the 2D BAT-Block, we first separately process each time slice $\mathbf{X}_{<t,*,*,*>}$ using the learned $\mathbf{P}, \mathbf{Q}$ according to Eqn. 1, obtaining the corresponding output $\mathbf{Y}_{<t,*,*,*>}$.

Next, information from different time slices are fused along the time dimension. Let $\mathbf{T} \in \mathbb{R}^{T \times T}$ be the learnable transform matrix in the time scale, and $\mathbf{Z} \in \mathbb{R}^{T \times H \times W \times C}$ be the final result. 3D BAT-Block has the following extra computation:

$$\mathbf{Z}_c \leftarrow \mathbf{T} \otimes \left[\mathbf{Y}_{<1,*,*,c>}, \mathbf{Y}_{<2,*,*,c>}, \ldots, \mathbf{Y}_{<T,*,*,c>}\right], \tag{7}$$

where $\mathbf{Y}_{<t,*,*,c>} \in \mathbb{R}^{1 \times H \times W \times 1}$ is a slice indexed by time and feature channel. $\otimes$ denotes ordinal convolution. $\mathbf{Z}_c \in \mathbb{R}^{T \times H \times W \times 1}$ is c-th channel of $\mathbf{Z}$. Essentially, $\mathbf{T}$ defines a 1×1 timescale convolution that operates on the concatenated T slices from $\mathbf{Y}$.

In practice, $\mathbf{T}$ can be simultaneously learned with $\mathbf{P}$ and $\mathbf{Q}$ in *transformation predictor*. Similar to row or column pooling, we design an average pooling along time, which is also adopted for frame feature representation in most video understanding tasks [33, 3]. To better capture temporal dynamics, we also implement multi-head attention at temporal dimension.

3.5. Complexity Analysis

Table 1 compares the number of parameters and FLOPs between the standard NL block [57] and our proposed BAT-block. Here we only take 2D block as an example and 3D block should reach the same conclusion. Since $C' \ll C$, $s_h \ll H$ and $s_w \ll W$, the complexity of terms with these symbols is negligible. It can be found that BAT-Block is more light-weight than NL block and the advantage of computation cost is much more obvious especially when the input resolution increase.

	NL block [57]	BAT-Block
#Params	$2C^2$	$\frac{5}{4}C^2 + \frac{1}{2}CC' + 2C'ks^3$
FLOPs	$2C^2HW + CH^2W^2$	$\frac{5}{4}C^2HW$ $+\frac{1}{2}CHW(H+W)$ $+\frac{1}{2}CC'HW + 2C'ks^3$

Table 1: **Complexity analysis.** For brevity, here we set $s_h = s_w = s$ and $C = D/2$ for BAT-Block, which is also consistent with experiments in Section 4.

4. Experiments

To validate the effectiveness and efficiency of the proposed block, we conduct comprehensive experiments on two standard tasks: image classification and video classification, where the large-scale ImageNet [28] and Kinetics [2] benchmark datasets are used respectively. Besides the percentage accuracy, we also report GFLOPs and the amount of parameters (Params#M) for each network variant. Unless explicitly stated, 5 BAT-Blocks are evenly inserted to a specific model on Res3 and Res4 with $C = D/2$, $C' = k = 4$, and $s_h = s_w = 7$. We adopt 3D BAT-Blocks for video classification where the block-based matrix trick is not used along time dimension since the number of input frames is not that large.

4.1. Experimental Setups

Image Classification. All network variants are trained on 4 GPUs for 90 epochs with the same strategy using PyTorch [42]. The learning rate starts from 0.1 and decreases by a factor of 0.1 after 30, 60 and 80 epochs. The batch size is set to 256. We adopt the Stochastic Gradient Descent (SGD) optimizer during training. The validation accuracy are obtained in the same way as [17, 22, 61] based on 224×224 single center crop.

Video Classification. We conduct experiments on Kinetics-400 [2] for human action recognition. Kinetics is a large-scale trimmed video dataset that contains more than 300K video clips in total. To conduct ablation studies, following [62] we create a smaller dataset named as Mini-Kinetics-200, which contains 200 categories. For each category, we randomly sample 400 examples for the training set and 25 examples for the validation set.

We choose ResNet-50 C2D and ResNet-50 I3D [57] as our backbone. The models are initialized from the pre-trained weights on ImageNet and finetuned on 4 GPUs with a mini-batch of 64 clips. The standard cross-entropy loss is used to guild video classification. All the models use 8-frame input clips with a stride of 8 frames (so covering 64 frames in the raw view). The spatial size of input is

Method	GFLOPs	#Params	Top-1
baseline	4.14	25.56	76.3
full resolution	5.50	31.86	77.6
downsampling	5.22	30.23	78.1
block-based	5.44	30.23	**78.3**

Table 2: **Ablation studies of block-based matrix.**

fixed as 224 × 224. For Kinetics-400, all the models are trained for 100 epochs with a learning rate starting from 0.01 and decreasing by a factor of 10 after 40 and 80 epochs. For Mini-Kinetics-200, the total epochs are 50 and a linear warm-up strategy [15] is used in the first 2 epochs. In addition, a cosine schedule [38] is adopted to perform learning rate decay. To reduce over-fitting, we also utilize dropout with a ratio of 0.5 after the global average pooling layer. Meanwhile, the weight decay is set to 0.0001.

We adopt the same data augmentation as [56], i.e., random horizontal flipping, random cropping and scale jittering. We report the clip Top-1 accuracy by selecting the center clip with center crop, and the video Top-1 accuracy by using 10-clip in time dimension, 3-crop spatially fully-convolutional inference [57, 11, 32]. More details can be found in *Supplementary Materials*.

4.2. Results on ImageNet

We conduct ablation studies on ImageNet using the standard ResNet-50 [17] by default.

Ablation Study of block-based matrix. In order to reduce the computation cost and the amount of parameters especially for the input of high resolution, we introduce the block-based matrix in Section 3.3. Here we also explore an alternative method termed as *downsampling* in Table 2. Instead of resizing the predicted matrix $\mathbf{P}$ / $\mathbf{Q}$ to the full resolution, this method downsamples the input feature map $\mathbf{X}$ to $\mathbf{X}^{ds} \in \mathbb{R}^{s_h \times s_w \times C}$ firstly and then operate BA-Transform on $\mathbf{X}^{ds}$ to get $\mathbf{Y}^{ds}$ with the spatial size $s_h \times s_w$. At last, we upsample Y^{ds} to the full resolution $\mathbf{Y}$ by bilinear interpolation. We also conduct experiments that predict $\mathbf{P}$ / $\mathbf{Q}$ with full resolution directly. As shown in Table 2, both the *downsampling* and block-based method can reduce GFLOPs and the number of parameters compared with *full resolution* while the block-based one obtains a higher performance. We analyse that more details are kept in our block, which are probably crucial for classification. Results on ResNet-50 show that the block-based method could also reduce over-fitting.

Res3	Res4	Res5	Top-1	Top-5
			76.3	92.9
+1			77.1	93.5
	+1		77.2	93.7
		+1	76.7	93.0
+1	+1		77.7	93.9
+2	+3		**78.3**	**94.0**

Table 3: **Performance gain by varying the inserting positions and counts of BAT-Blocks.**

k	GFLOPs	#Params	Top-1
baseline	4.14	25.56	76.3
0	5.43	30.17	76.5
1	5.44	30.18	77.9
2	5.44	30.19	78.0
4	5.44	30.23	78.3
16	5.45	31.09	**78.4**

Table 4: **Ablation studies of multi-head attention.** k represents the number of attention groups. $k = 0$ means that the $\mathbf{P}$ / $\mathbf{Q}$ is fixed as identity matrix.

Backbone	Method	GFLOPs(Δ)	#Params(Δ)	Top-1
ResNet-18	baseline	-	-	70.2
	+NL	0.23	0.17	70.9
	+BAT	0.03	0.13	**71.3**
ResNet-50	baseline	-	-	76.3
	+NL	3.55	7.36	77.5
	+BAT	1.30	4.67	**78.3**

Table 5: **Comparisons with NL block on ImageNet.**

Different numbers and stages. Table 3 explores different numbers of BAT-Blocks inserted to different locations of a model. We find that even one BAT-Block inserted at Res3 or Res4 can bring a significant improvement, and the improvement of a BAT-Block on Res5 is minor, which may be caused by the small spatial size (7 × 7) that can not provide precise spatial information. More BAT-Blocks continue to improve the performance.

Multi-head attention. We explore the effectiveness brought by multi-head attention with different k in Table 4. In order to confirm whether the improvement mainly benefits from the extra parameters, we design a new baseline by setting $\mathbf{P}$ / $\mathbf{Q}$ to an identity matrix and not conditioned on the input, noted as $k = 0$ in Table 4. As can be seen, there is tiny improvement by simply introducing extra parameters. Even one group of attention could bring a noticeable improvement (+1.4%) with negligible parameters introduced compared with the new baseline. This phenomenon shows that the *transformer predictor* is the key to improve performance and it's very light-weight. And more groups of attention can further improve the performance, but the gain diminishes quickly.

Comparisons with NL block. NL block [57] has been proved to significantly improve performance in several tasks by modeling long range dependencies [57, 7, 24]. We

Method	GFLOPs	#Params	Top-1
SE-ResNet-50 [22]	4.2	28.1	76.9
GE-ResNet-50 [21]	-	31.1	76.8
SRM-ResNet-50 [30]	-	25.6	77.1
A^2-Net [7]	-	-	77.0
DenseNet-201 [23]	4.4	20.0	77.4
ResNeXt-50 (32 × 4d) [61]	4.3	25.0	77.8
Res2Net-50 (14w×8s) [14]	4.2	-	78.1
Oct-ResNet-50 [6]	2.4	25.6	77.3
ResNet-101 [17]	7.9	44.6	77.4
ResNet-152 [17]	11.6	60.2	78.3
SE-ResNet-152 [22]	11.7	67.2	78.4
ResNeXt-101 (32 × 4d) [61]	16.5	88.8	78.8
AttentionNeXt-56 [55]	6.3	31.9	78.8
ResNet-50 + BAT	5.4	30.2	78.3
SE-ResNet-50 + BAT	5.5	33.1	78.4
ResNext-50 (32×4d) + BAT	5.6	29.7	78.6
ResNet-101 + BAT	9.2	49.2	79.1
ResNet-152 + BAT	12.9	64.9	**79.4**

Table 6: **Comparisons with state-of-the-art on ImageNet.**

compare these two blocks to show the superiority of ours. We insert NL blocks at the same locations as those of BAT-Block and the results are shown in Table 5. Obviously, our proposed method is more light-weighted and effective compared with NL block, with only $13\% \sim 31\%$ GFLOPs and fewer parameters to achieve higher accuracy.

Comparisons with state-of-the-art. In order to verify the generality of BAT-Blocks, we also conduct experiments on some other popular networks and go deeper with BAT-Blocks. As shown in Table 6, consistent performance gain could be obtained by inserting BAT-Blocks even for a very deep model, ResNet-152. Additionally, adding BAT blocks to shallower models can outperform several deeper neural networks. For example, ResNet-50 with BAT blocks achieves the same accuracy as the original ResNet-152 while using only half GFLOPs and parameters around.

4.3. Results on Kinetics

Comparisons with NL block. Table 7 shows the results of video classification on Mini-Kinetics-200. We found that it was easy to overfit in video classification for models adding BAT-Blocks, and initializing BAT-Blocks with parameters pre-trained on ImageNet could alleviate this problem in large measure. For a fair comparison, we also conduct the experiment with pre-trained NL blocks. We can find that BAT blocks with only spatial attention could achieve a better accuracy than NL blocks with less computation cost and fewer parameters whether NL network is pre-trained on ImageNet or not.

Method	Pre-trained	GFLOPs	#Params	Val	Train
baseline	-	19.55	23.92	66.4	71.6
NL	No	30.69	31.28	67.7	72.6
NL	Yes	30.69	31.28	68.8	74.6
BAT (4, 0)	Yes	24.76	28.60	69.5	76.0
BAT (4, 1)	Yes	24.77	28.60	70.3	77.1
BAT (4, 2)	Yes	24.77	28.60	70.5	**77.6**
BAT (4, 4)	Yes	24.77	28.60	**70.6**	77.5

Table 7: **Results on Mini-Kinetics-200.** *Pre-trained* means whether the newly added blocks are pre-trained on ImageNet. BAT (k_s, k_t) represents BAT block with k_s groups of spatial attention and k_t groups of temporal attention. All models adopt ResNet-50 C2D as backbone. Clip Top-1 accuracy is reported here.

Method	3D-Conv	GFLOPs	#Params	Top-1
A^2-Net [7]	Yes	40.8	-	74.6
Oct-I3D [6]	Yes	25.6	-	74.6
TSM [32]	No	32.8	24.3	74.1
GloRe [8]	Yes	28.9	-	75.1
C2D	No	19.6	24.3	72.0
I3D	Yes	28.4	28.4	72.7
C2D + NL	No	30.7	31.7	73.8
I3D + NL	Yes	39.5	35.4	73.5
C2D + BAT	No	24.8	29.2	74.6
I3D + BAT	Yes	33.6	32.9	75.1
C2D + 3D-BAT	No	24.8	29.2	75.5
C2D + 3D-BAT†	No	24.8	29.2	**75.8**

Table 8: **Results on Kinetics-400.** The first set is recent state-of-the-art, the second set is our re-implemented models, and the last set is our methods. The group number of spatial attention is 8 and that of the temporal attention is set to 4. All the models use ResNet-50 as backbone and 8 frames as input. "†" represents finetuning with TSN framework [56].

Attention on temporal dimension. We also examine if the temporal attention works well for video classification and the results are reported in Table 7. The improvement is significant (+0.8%) by adding a even single group of spatial attention with negligible extra parameters. More groups of temporal attention could further improve the performance.

Results on Kinetics-400. Here we compare our methods with state-of-the-art on the full dataset to demonstrate the effectiveness and efficiency of our BAT-Block. It has been widely proven that the performance of video classification is closely related with the number of input frames and backbone architectures [57, 11, 44], and as a result, for fair comparisons, we only focus on comparing with models using 8-frame clips as input and ResNet-50 as backbone.

Firstly, we re-implemented the C2D, I3D baselines and NL networks under the same settings as ours. All the results can be found in Table 8, which indicate that the proposed 2D BAT-Block consistently improves the performance over both C2D and I3D baselines, and the benefit of temporal at-

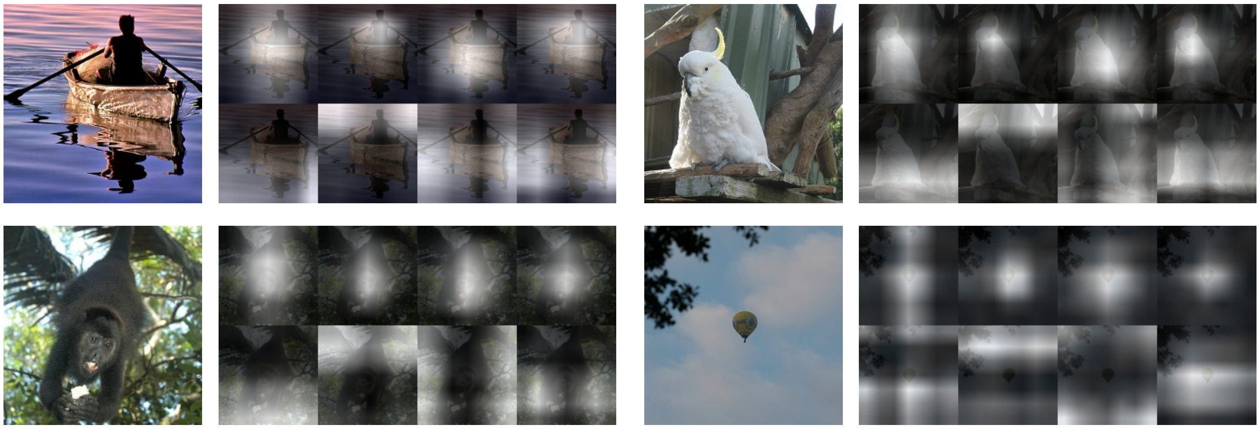

Figure 4: **Examples of attention weight.** To investigate where BAT-blocks focus for each group of attention, we visualize the attention weights of the last BAT block with 8 attention groups as the last block is most related to the final classification. These samples are taken from the validation set of ImageNet randomly. From left to right, each group contains an RGB image and its corresponding eight attention maps. For clear clarification, we note the attention weight images of each group as *a-d* (the top 4 images, from left to right) and *e-h* (the bottom 4 images, from left to right). We find that, for all examples, *e* pays more attention to the bottom of images while *f* focuses on the top regions, and these two attention collaborate with each other to classify an image by splitting the full image into two sub-regions. Additionally, *g* tends to observe the background, and *h* prefers to focus on the regions nearby the foreground. Therefore, they may help the network model rich context information. It can be found that all *a-d* focus on the foreground, but there are still some differences each corresponding various discriminative details. More examples can be found in *supplementary materials*.

Backbone	8-frame	16-frame	64-frame
ResNet-50	75.5	76.9	**77.7**
ResNet-101	76.2	77.4	-

Table 9: **Results on Kinetics-400 with different length of sequences or backbones.** All the models adopt C2D + 3D-BAT. For models with sequences longer than 8 frames, the block-based matrix trick is adopted along time dimension to reduce parameters, and the number of blocks are set to 8×8.

tention is noticeable. Additionally, our method outperforms NL networks by a large margin. Introducing 3D convolution improves C2D + BAT by 0.5% while temporal attention improves 0.9% with almost zero growth of both GFLOPs and parameter numbers, which shows the powerful ability of BAT-blocks to model 3D information.

Comparing with other state-of-the-art, we can see that simply adding 2D BAT-blocks on a basic network can achieve a comparable results with other recent methods, and models with 3D BAT-block outperform most competitive models. Specifically, to our best knowledge, after finetuning with TSN framework [56], we can achieve a new state-of-the-art among models under the similar complexity.

Longer sequences and deeper networks. Finally, we investigate the generality of our methods on longer input videos or deeper networks. And the results can be found in Table 9. For comparison, based on ResNet-101 and 128-frame clips, the accuracy of C2D baseline is 75.3% and I3D + NL is 77.7% [57], which shows that our methods work well on longer sequences or deeper networks.

4.4. Visualization

The above experiments have shown the effectiveness of BAT-Block on both 2D and 3D tasks, here we visualize several attention weight maps to investigate how BAT-Block works. In order to visualize where a block pays attention over input images, we adopt the following formula to re-project the attention weights to the input feature maps:

$$\mathbf{W} = \mathbf{P}^{\intercal}\mathbf{A}\mathbf{Q}^{\intercal}, \tag{8}$$

where $\mathbf{A} \in \mathbb{R}^{H \times W}$ is an all-ones matrix. $\mathbf{W}$ is the re-projected attention weight with shape $H \times W$. The results are normalized between 0 and 255 for visualization. Some examples are depicted and analyzed in Figure 4.

5. Conclusion

We propose *BA-Transform*, a novel method which can model various attentional operations by matrix multiplication. The core operation is to learn data-adaptive grouped bilinear attentional transforms. We wrap this operation to a BAT-Block and carefully design *transformation predictor*. It can be dropped into most existing networks and optimized easily. Extensive experiments on image classification and video action recognition verify the superiority of our method both in accuracy and efficiency.

Acknowledgement: This work is partially supported by Beijing Natural Science Foundation (Z190001), Beijing Municipal Commission of Science and Technology (Z181100008918005) and National Natural Science Foundation of China (61772037).

References

[1] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. In *ICLR*, 2017.

[2] João Carreira and Andrew Zisserman. Quo vadis, action recognition? A new model and the kinetics dataset. In *CVPR*, pages 4724–4733, 2017.

[3] Jingwen Chen, Yingwei Pan, Yehao Li, Ting Yao, Hongyang Chao, and Tao Mei. Temporal deformable convolutional encoder-decoder networks for video captioning. In *AAAI*, pages 8167–8174, 2019.

[4] Liang-Chieh Chen, Yi Yang, Jiang Wang, Wei Xu, and Alan L. Yuille. Attention to scale: Scale-aware semantic image segmentation. In *CVPR*, pages 3640–3649, 2016.

[5] Xin Chen, Lingxi Xie, Jun Wu, and Qi Tian. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation. In *ICCV*, 2019.

[6] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. *ICCV*, 2019.

[7] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Aˆ2-nets: Double attention networks. In *NIPS*, pages 350–359, 2018.

[8] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Shuicheng Yan, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *CVPR*, pages 433–442, 2019.

[9] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, pages 2219–2228, 2019.

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *ICCV*, pages 764–773, 2017.

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *ICCV*, pages 6202–6211, 2019.

[12] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *CVPR*, pages 3146–3154, 2019.

[13] Jianlong Fu, Heliang Zheng, and Tao Mei. Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition. In *CVPR*, pages 4476–4484, 2017.

[14] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. *IEEE TPAMI*, 2020.

[15] Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677, 2017.

[16] Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra. DRAW: A recurrent neural network for image generation. In *ICML*, pages 1462–1471, 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[18] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. In *ICCV*, 2019.

[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. *CoRR*, abs/1704.04861, 2017.

[20] Han Hu, Zheng Zhang, Zhenda Xie, and Stephen Lin. Local relation networks for image recognition. *CoRR*, abs/1904.11491, 2019.

[21] Jie Hu, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. Gather-excite: Exploiting feature context in convolutional neural networks. In *NeurIPS*.

[22] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018.

[23] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 2261–2269, 2017.

[24] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. Ccnet: Criss-cross attention for semantic segmentation. In *ICCV*, pages 603–612, 2019.

[25] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NIPS*, pages 2017–2025, 2015.

[26] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Fei-Fei Li. Large-scale video classification with convolutional neural networks. In *CVPR*, pages 1725–1732, 2014.

[27] Adam R. Kosiorek, Alex Bewley, and Ingmar Posner. Hierarchical attentive recurrent tracking. In *NIPS*, pages 3053–3061, 2017.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[29] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *ECCV*, pages 765–781, 2018.

[30] HyunJae Lee, Hyo-Eun Kim, and Hyeonseob Nam. SRM: A style-based recalibration module for convolutional neural networks. In *ICCV*, pages 1854–1862, 2019.

[31] Xiang Li, Wenhai Wang, Xiaolin Hu, and Jian Yang. Selective kernel networks. In *CVPR*, pages 510–519, 2019.

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

[33] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. BSN: boundary sensitive network for temporal action proposal generation. In *ECCV*, pages 3–21, 2018.

[34] Chenchen Liu, Xinyu Weng, and Yadong Mu. Recurrent attentive zooming for joint crowd counting and precise localization. In *CVPR*, pages 1217–1226, 2019.

[35] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In *ICLR*, 2019.

[36] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *CoRR*, abs/1411.4038, 2014.

[37] Xiang Long, Chuang Gan, Gerard de Melo, Jiajun Wu, Xiao Liu, and Shilei Wen. Attention clusters: Purely attention based local feature integration for video classification. In *CVPR*, pages 7834–7843, 2018.

[38] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[39] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In *ECCV*, pages 122–138, 2018.

[40] Carl D. Meyer, editor. *Matrix Analysis and Applied Linear Algebra*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

[41] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *CoRR*, abs/1406.6247, 2014.

[42] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[43] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *ICCV*, pages 5534–5542, 2017.

[44] Zhaofan Qiu, Ting Yao, Chong-Wah Ngo, Xinmei Tian, and Tao Mei. Learning spatio-temporal representation with local and global diffusion. In *CVPR*, pages 12056–12065, 2019.

[45] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015.

[46] Laura Walker Renninger, James M. Coughlan, Preeti Verghese, and Jitendra Malik. An information maximization model of eye movements. In *NIPS*, pages 1121–1128, 2004.

[47] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[49] Ke Sun, Mingjie Li, Dong Liu, and Jingdong Wang. IGCV3: interleaved low-rank group convolutions for efficient deep neural networks. In *BMVC*, page 101, 2018.

[50] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[51] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.

[52] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[53] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, pages 6450–6459, 2018.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017.

[55] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *CVPR*, pages 6450–6458, 2017.

[56] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, pages 20–36, 2016.

[57] Xiaolong Wang, Ross B. Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, pages 7794–7803, 2018.

[58] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *ECCV*, pages 399–417, 2018.

[59] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: convolutional block attention module. In *ECCV*, pages 3–19, 2018.

[60] Zuxuan Wu, Xi Wang, Yu-Gang Jiang, Hao Ye, and Xiangyang Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *ACM Multimedia*, pages 461–470, 2015.

[61] Saining Xie, Ross B. Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017.

[62] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *CoRR*, abs/1712.04851, 2017.

[63] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018.

[64] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets V2: more deformable, better results. In *CVPR*, pages 9308–9316, 2019.