Supplemental Material to “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

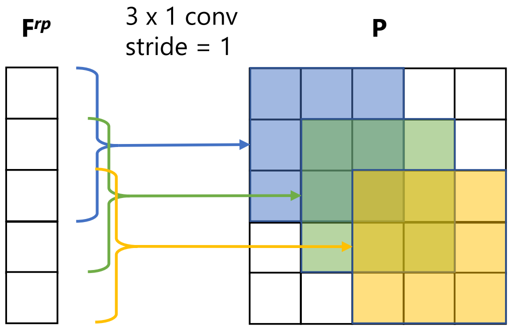

Figure 1: **Example of sliding windows for predicting $\mathbf{P}$ in BAT-Block.** Here the shape of $\mathbf{F}^{rp}$ is 4×1 and the kernel size of convolution is 3×1. The values in the white cell of $\mathbf{P}$ are set to 0. And the values in the overlaping regions are averaged.

1. Video Classification Architecture

We conduct experiments on video classification based on ResNet-50 C2D and ResNet-50 I3D proposed in [6], the details of architectures (with 8-frame clips as input) are shown in Table 1.

2. Sliding Windows for BAT-Block

For video classification, we perform spatially fully-convolutional inference on videos whose shorter spatial side is rescaled to 256 and take 3 crops of 256×256 to cover the spatial dimensions, and for the temporal domain, 10 clips are evenly sampled from a full-length video, and most recent works adopt this inference method [6, 1, 2].

The main challenge of fully-convolutional inference for our methods is how to process multi-resolution for *transformation predictor* since the size of $\mathbf{P}$ and $\mathbf{Q}$ are fixed after initialization. Inspired by sliding windows in convolutional layer, a specially designed sliding window is proposed to predict multi-resolution matrices $\mathbf{P}$ and $\mathbf{Q}$. For example, as depicted in Figure 1, the shape of $\mathbf{F}^{rp}$ is 3×1 during training and increases to 4×1 due to the shape of input $\mathbf{X}$ increases to 4×4 during inference. We slide the convolutional windows across $\mathbf{F}^{rp}$ with stride 1 to generate three 3×3 matrices. We place these matrices along the diagonal with a stride of 1, and for the overlapping parts the values are averaged while the the values of uncovered regions are set to 0. Finally a 4×4 matrix is generated and so the transformation can be applied on $\mathbf{X}$.

Besides fully-convolutional inference, another popular inference method in video classification is using 25 clips and ten crops for each video [5, 3, 4]. In Table 2 we compare these two inference methods and show the effectiveness of the specially designed sliding windows for BAT-Block. It can be seen that fully-convolutional inference can achieve a better accuracy with a tiny margin for most models. The main advantage of full-convolutional inference is efficiency.

3. More Visualization Results

Let $\mathbf{X} \in \mathbb{R}^{H\times W}$ be the input of a BAT-Block with spatial resolution $H\times W$. Its channel dimension is hereafter omitted for ease of statement. $\mathbf{P} \in \mathbb{R}^{H\times H}, \mathbf{Q} \in \mathbb{R}^{W\times W}$ are the transformation matrices, namely $\mathbf{Y} = \mathbf{PXQ}$. Since the learned transforms are often complex compositions of different elementary transforms, it is difficult to understand the learned transforms by directly observing $\mathbf{Y}$. We may have two ways for circumventing this issue:

1) The first method has been introduced in our main draft. Each value in **P / Q** is regard as the attention weight and re-projected to the input by formula (8). And as analyzed in Figure 4, different groups focus on different parts. This division and cooperation mechanism can simplify the complex recognition problems in most cases. And more attention maps can be found in Figure 3.

2) Next let us introduce a second visualization method in details. For any pixel $\mathbf{Y}_{i,j}$ of output $\mathbf{Y}$, we can get the contribution of $\mathbf{X}_{m,n}$ by the following formula:

$$\mathbf{C}^{i,j}_{m,n} = \mathbf{P}_{i,m}\mathbf{Q}_{n,j}, i, m \in [0, H), j, n \in [0, W), \qquad (1)$$

where subscripts of a variable represent its spatial location. $\mathbf{C}^{i,j} \in \mathbb{R}^{H\times W}$ is the contribution distribution over $\mathbf{X}$ for $\mathbf{Y}_{i,j}$, termed as contribution map. For a specific $\mathbf{Y}$,

Stage	ResNet-50 C2D	ResNet-50 I3D	Output Size
conv_1	$1\times7\times7$, 64, stride (1,2,2)	$3\times7\times7$, 64, stride (1,2,2)	$8 \times 112 \times 112$
pool_1	$1\times3\times3$ max, stride (1,2,2)		$8 \times 56 \times 56$
res_2	$\begin{bmatrix} 1\times1\times1, & 64 \\ 1\times3\times3, & 64 \\ 1\times1\times1, & 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times1\times1, & 64 \\ 1\times3\times3, & 64 \\ 1\times1\times1, & 256 \end{bmatrix} \times 3$	$8 \times 56 \times 56$
pool_2	$2\times1\times1$ max, stride (2,1,1)		$4 \times 56 \times 56$
res_3	$\begin{bmatrix} 1\times1\times1, & 128 \\ 1\times3\times3, & 128 \\ 1\times1\times1, & 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3\times1\times1, & 128 \\ 1\times3\times3, & 128 \\ 1\times1\times1, & 512 \end{bmatrix} \times 4$	$4 \times 28 \times 28$
res_4	$\begin{bmatrix} 1\times1\times1, & 256 \\ 1\times3\times3, & 256 \\ 1\times1\times1, & 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3\times1\times1, & 256 \\ 1\times3\times3, & 256 \\ 1\times1\times1, & 1024 \end{bmatrix} \times 6$	$4 \times 14 \times 14$
res_5	$\begin{bmatrix} 1\times1\times1, & 512 \\ 1\times3\times3, & 512 \\ 1\times1\times1, & 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3\times1\times1, & 512 \\ 1\times3\times3, & 512 \\ 1\times1\times1, & 2048 \end{bmatrix} \times 3$	$4 \times 7 \times 7$
global average pool, fc			$1 \times 1 \times 1$

Table 1: **C2D and I3D models based on ResNet-50 backbone for video classification.** The kernel size and output maps are represented in the format of $T \times H \times W$, typically with the number of channels following. The input size is $8 \times 224 \times 224$ with 8 stacked frames.

Backbone	#Frames	Method	250 views	30 views
ResNet-50	8	C2D	71.9	72.0 (+0.1)
		I3D	72.6	72.7 (+0.1)
		C2D + NL	73.8	73.8
		I3D + NL	73.5	73.5
		C2D + BAT	74.5	74.6 (+0.1)
		I3D + BAT	74.8	75.1 (+0.3)
		C2D + 3D-BAT	75.2	75.5 (+0.3)
		C2D + 3D-BAT†	75.9	75.8 (-0.1)
ResNet-50	16	C2D + 3D-BAT	**76.5**	76.9 (+0.4)
ResNet-50	64	C2D + 3D-BAT	-	**77.7**
ResNet-101	8	C2D + 3D-BAT	76.2	76.2
ResNet-101	16	C2D + 3D-BAT	-	77.4

Table 2: **Results on Kinetics-400 with different inference methods.** The method with 10 crops and 25 segments is noted as 250 views and the fully-convolutional inference method is noted as 30 views here. In the brackets are the gaps to 250 views.

there exists $H \times W$ contribution maps. Examples are randomly selected and depicted in Figure 2, and three important properties can be seen: Firstly, (b) / (d) shows that receptive fields are varying over different groups. Some groups can achieve the global receptive field. Secondly, receptive fields are also conditioned on the input. For example, almost every region of (a) is informative, therefore the receptive fields shown in (b) are larger than that in (d). Thirdly, the contribution distribution is related to (i, j). In the case of (e), when (i, j) locates nearby the foreground, the contribution weight is larger, which can enhance the important information.

References

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

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

[3] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[4] Shuyang Sun, Zhanghui Kuang, Lu Sheng, Wanli Ouyang, and Wei Zhang. Optical flow guided feature: A fast and robust motion representation for video action recognition. In *CVPR*, pages 1390–1399, 2018.

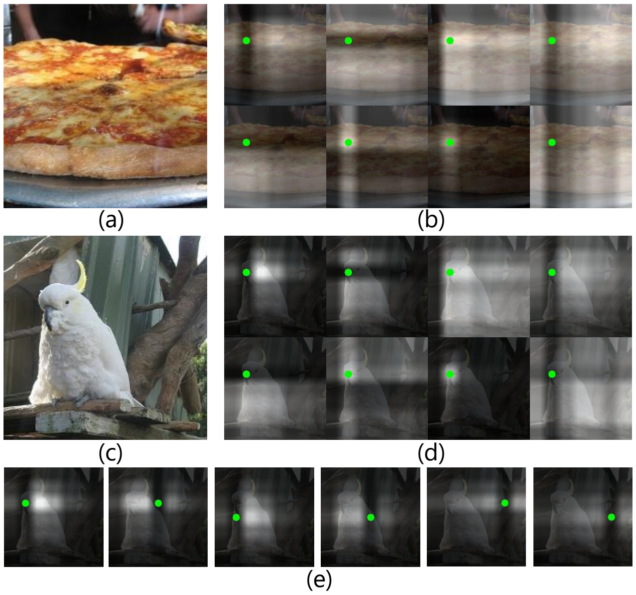

Figure 2: **Some contribution maps of the last BAT block with 8 attention groups.** (a) and (c) are two input images, others are contribution maps and the green circle marks the location (i, j) of $\mathbf{Y}_{i,j}$. The brighter region means more contribution, and the larger bright area means larger receptive field. (b) / (d) explores the difference between 8 groups under the same (i, j) (the order of groups keeps the same as that of Figure 4 in the main text). (e) is a set of contribution maps of image (c) under the same group (the first group) but different (i, j).

[5] 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.

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

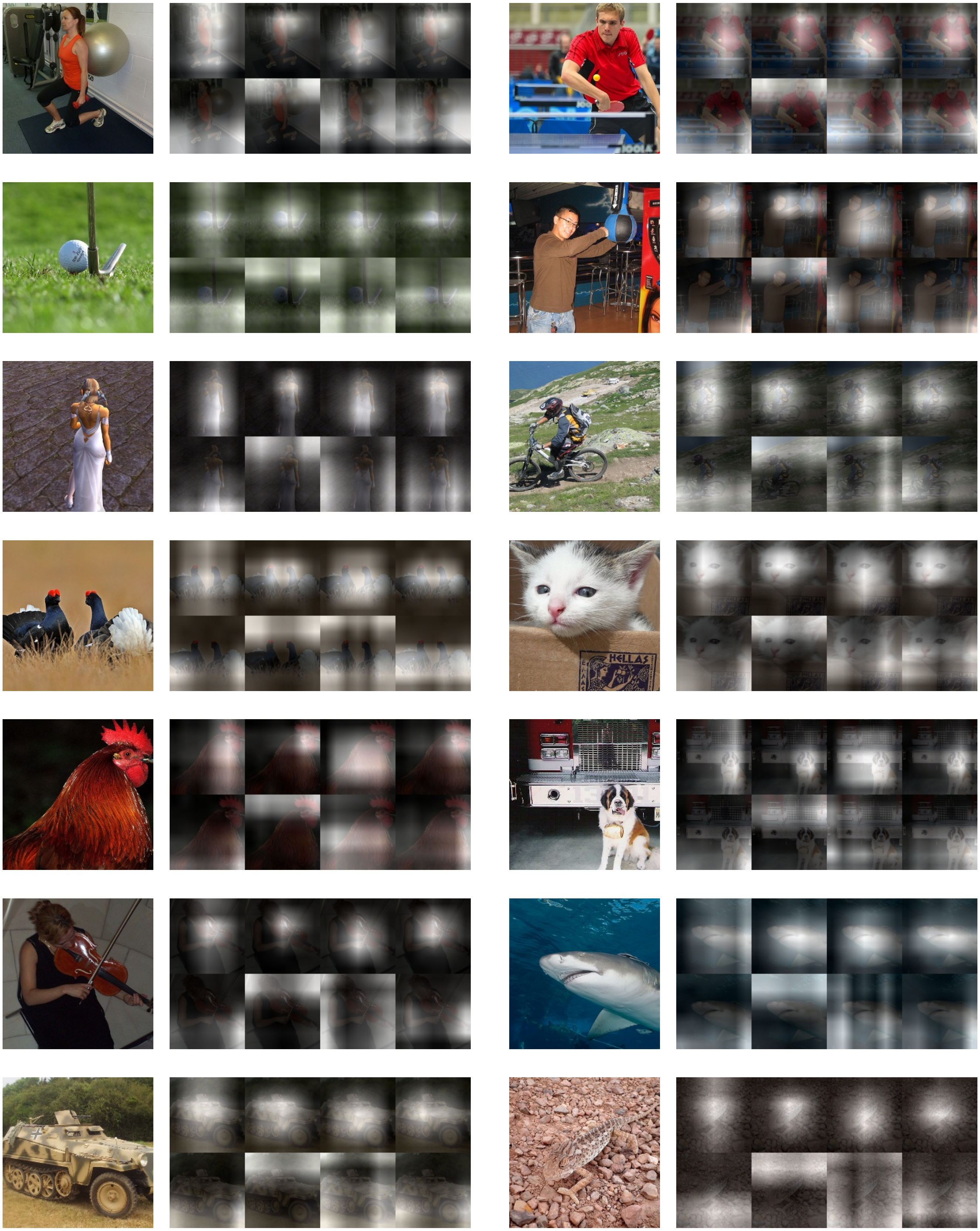

Figure 3: **More examples of attention weight.**