BOAT: Bilateral Local Attention Vision Transformer (Supplementary Material)

Tan Yu[1]
tan.yu1503@gmail.com
Gangming Zhao[2]
gangmingzhao@gmail.com
Ping Li[1]
pingli98@gmail.com
Yizhou Yu[2]
yizhouy@acm.org

[1] Cognitive Computing Lab
Baidu Research
[2] Department of Computer Science
The University of Hong Kong

1 Network Architecture and Complexity

The proposed BOAT model is trained from end to end. Clustering only partitions patches into groups, and does not involve any learnable parameters.

1.1 BOAT-Swin

BOAT-Swin is built by replacing a subset of the blocks in Swin with the proposed Bilateral Local Attention (BLA) Block. Similar to Swin, our BOAT-Swin consists of 4 stages. In each of the first three stages, we replace half of the Swin blocks with our BLA blocks. In all different versions, our BOAT-Swin has the same number of blocks as the corresponding version of Swin. To be specific, our BOAT-Swin-Tiny has $[2,2,6,2]$ blocks in four stages, our BOAT-Swin-Small has $[2,2,18,2]$ blocks in four stages, and our BOAT-Swin-Base has $[2,2,18,2]$ blocks in four stages. In stage i, we set the number of clusters in our feature-space local attention (FSLA) module of the BLA block to 2^{8-2i}. That is, the number of levels in our balanced hierarchical clustering is $K = 8-2i$. The detailed specifications of our BOAT-Swin are given in Table 1.

1.2 BOAT-CSWin

BOAT-CSWin also has 4 stages. In all different versions, our BOAT-CSWin has the same number of blocks as the corresponding version of CSWin. Except for the 4-th stage of BOAT-CSwin-Tiny, BOAT-CSWin replaces half of the blocks in each stage of CSwin with BLA blocks. Our BOAT-CSWin-Tiny has $[1,2,21,1]$ blocks in four stages, our BOAT-CSWin-Small has $[2,4,32,2]$ blocks in four stages, and our BOAT-CSWin-Base has $[2,4,32,2]$

	BOAT-Swin-T	BOAT-Swin-S	BOAT-Swin-B
	concat 4×4	concat 4×4	concat 4×4
stage 1	[win. sz. 7×7, dim 96, head 3 Swin-Block win. sz. 7×7, dim 96, head 3 BLA-Block] $\times1$	[win. sz. 7×7, dim 96, head 3 Swin-Block win. sz. 7×7, dim 96, head 3 BLA-Block] $\times1$	[win. sz. 7×7, dim 128, head 4 Swin-Block win. sz. 7×7, dim 128, head 4 BLA-Block] $\times1$
	concat 2×2	concat 2×2	concat 2×2
stage 2	[win. sz. 7×7, dim 192, head 6 Swin-Block win. sz. 7×7, dim 192, head 6 BLA-Block] $\times1$	[win. sz. 7×7, dim 192, head 6 Swin-Block win. sz. 7×7, dim 192, head 6 BLA-Block] $\times1$	[win. sz. 7×7, dim 256, head 8 Swin-Block win. sz. 7×7, dim 256, head 8 BLA-Block] $\times1$
	concat 2×2	concat 2×2	concat 2×2
stage 3	[win. sz. 7×7, dim 384, head 12 Swin-Block win. sz. 7×7, dim 384, head 12 BLA-Block] $\times3$	[win. sz. 7×7, dim 384, head 12 Swin-Block win. sz. 7×7, dim 384, head 12 BLA-Block] $\times9$	[win. sz. 7×7, dim 512, head 16 Swin-Block win. sz. 7×7, dim 512, head 16 BLA-Block] $\times9$
	concat 2×2	concat 2×2	concat 2×2
stage 4	[win. sz. 7×7, dim 768, head 24 Swin-Block] $\times2$	[win. sz. 7×7, dim 768, head 24 Swin-Block] $\times2$	[win. sz. 7×7, dim 1024, head 32 Swin-Block] $\times2$

Table 1: Detailed network architecture of BOAT-Swin.

blocks in four stages. In stage i, we set the number of clusters in our feature-space local attention (FSLA) module of the BLA block to 2^{8-2i}. That is, the number of levels in our balanced hierarchical clustering is $K = 8-2i$. The detailed specifications of our BOAT-CSWin are given in Table 2.

1.3 Time Complexity

Let us denote the number of tokens in the whole image by N, the number of levels in hierarchical clustering by K, and the feature dimension by C. The complexity of hierarchical binary clustering with 2 iterations is only $\mathcal{O}(4KNC)$. As $K = \log_2(N/n)$ where n is the number of tokens in each cluster, $K \ll N$. Thus our hierarchical binary clustering has a much lower complexity than global self-attention, whose complexity is $\mathcal{O}(N^2C)$.

	BOAT-CSWin-T	BOAT-CSWin-S	BOAT-CSWin-B
	conv	conv	conv
stage 1	[win. sz. 7×7, dim 64, head 2 BLA-Block] × 1	[win. sz. 7×7, dim 64, head 2 BLA-Block win. sz. 7×7, dim 64, head 2 CSWin-Block] × 1	[win. sz. 7×7, dim 96, head 4 BLA-Block win. sz. 7×7, dim 96, head 4 CSWin-Block] × 1
	conv	conv	conv
stage 2	[win. sz. 7×7, dim 128, head 4 BLA-Block win. sz. 7×7, dim 128, head 4 CSwin-Block] × 1	[win. sz. 7×7, dim 128, head 4 BLA-Block win. sz. 7×7, dim 128, head 4 CSWin-Block] × 2	[win. sz. 7×7, dim 192, head 8 BLA-Block win. sz. 7×7, dim 192, head 8 CSWin-Block] × 2
	conv	conv	conv
stage 3	[win. sz. 7×7, dim 256, head 8 BLA-Block win. sz. 7×7, dim 256, head 8 CSWin-Block] × 10 [win. sz. 7×7, dim 768, head 8 BLA-Block] × 1	[win. sz. 7×7, dim 256, head 8 BLA-Block win. sz. 7×7, dim 256, head 8 CSWin-Block] × 16	[win. sz. 7×7, dim 384, head 16 BLA-Block win. sz. 7×7, dim 384, head 16 CSWin-Block] × 16
	conv	conv	conv
stage 4	[win. sz. 7×7, dim 512, head 16 BLA-Block] × 1	[win. sz. 7×7, dim 512, head 16 BLA-Block win. sz. 7×7, dim 512, head 16 CSWin-Block] × 1	[win. sz. 7×7, dim 768, head 32 BLA-Block win. sz. 7×7, dim 768, head 32 CSWin-Block] × 1

Table 2: Detailed network architecture of BOAT-CSWin.

2 Ablation Studies

The impact of the number of iterations. As shown in Algorithm 1 of the main text, our balanced binary clustering is an iterative algorithm. We evaluate the impact of the number of iterations on performance. As shown in Table 3, using a single iteration, the BOAT-Swin-Tiny model achieves 82.1% top-1 accuracy. When the number of iterations is increased to 2 and 3, the top-1 accuracy reaches 82.3. By default, we use two iterations in all experiments.

Note that the proposed balanced binary clustering is not required to converge. As shown in Table 3, good performance can already be achieved using a single iteration (clustering is far from converging). Clustering only needs to divide patches into groups with roughly similar features. Convergence is not a critical concern.

# of iter.	1	2	3
Top-1 Accuracy	82.1	82.3	82.3

Table 3: The impact of the number of iterations in balanced binary clustering on the image classification accuracy of the BOAT-Swin-Tiny model.

The impact of BOAT on latency. We test the latency per batch using a single NVIDIA A100 GPU with a batch size of 256. On 224×224 images, the latency of Swin-Tiny per batch is 182ms and the latency of our BOAT-Swin-Tiny per batch is 254ms.

The impact of overlapping clustering on latency. We test the latency per batch using a single NVIDIA A100 GPU with a batch size of 256. On 224×224 images, the per-batch latency of BOAT-Swin-Tiny with overlapping clustering is 254ms while that of BOAT-Swin-Tiny without overlapping clustering is 245ms.