南京大学120周年校庆
120th ANNIVERSARY
NANJING UNIVERSITY
1902 - 2022

ResT V2: Simpler, Faster and Stronger

Presenter：Qing-Long Zhang

2022/10/16

Efficient ViTs：

- Reintroducing “Sliding Window " -> Complicated Structure
  - ① Patch input into non-overlapping windows: W-MSA
  - ② Window information communication： Swin, Shuffle Transformer, Twins, etc.
- Reducing Spatial Dimension of MSA->Lower Accuracy
  - ① Downsampling K、V： PVTv1 & v2, ResTv1
  - ② Downsampling Q、K、V： MViTv1 & v2

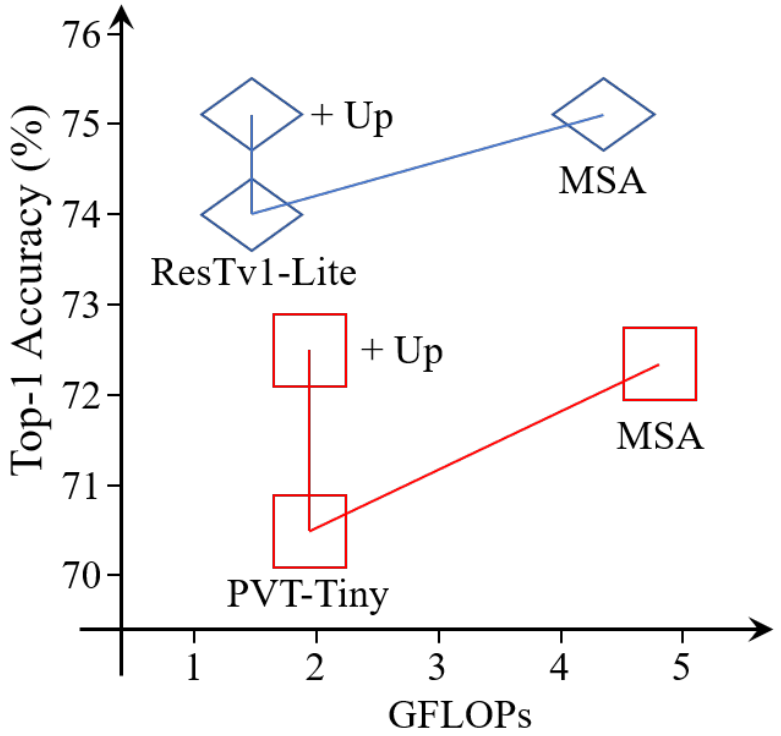


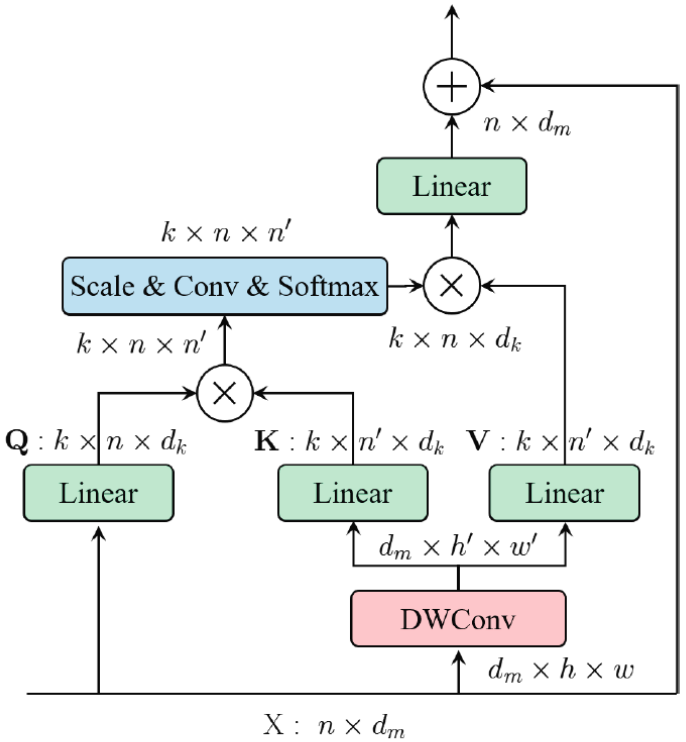
Figure 1: Top-1 Accuracy of ResT-Lite [47] and PVT-Tiny [35] under 100 epochs training settings. Results show that downsampling operation will impair the performance while adding an upsampling operation can address this issue. Detailed comparisons are shown in Appendix A.

Motivation：

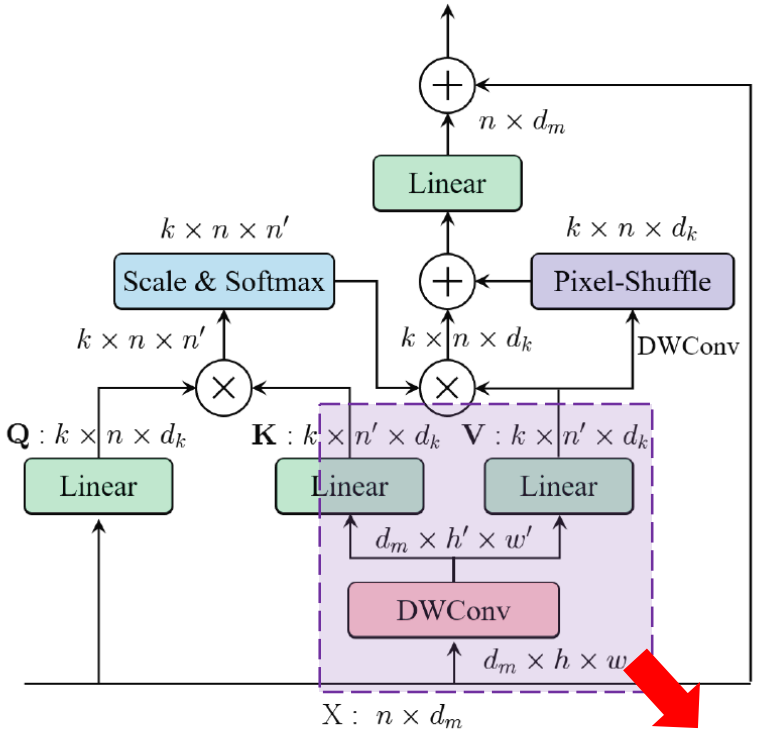
- Down sampling in MSA impair global dependency modeling ability
- Proposed to utilizing up-sample module to reconstruct the lost information

ResTv2：Simpler, Faster and Stronger

$$\mathrm{EMSAv2}(Q,K,V)=\mathrm{Softmax}(\frac{QK^{\mathrm{T}}}{\sqrt{d_k}})V+\mathrm{Up(V)} \quad (1)$$



(a) EMSA module



(b) EMSAv2 module **Shared Parts**

Figure 2: Comparison of EMSA in ResTv1 and EMSAv2 in ResTv2. To simplify, all normalization operators in EMSA and EMSAv2 are not displayed.

Efficiently Combine SA and ConvNet

- Sharing down-sample and linear projection

1. SA Branch： EMSA

- Eliminating multi-head interaction module

2. ConvNet Branch： Hourglass

down sample, with LN

$x' = \mathrm{DWConv}_1(x)$

reshape, linear, reshape

$V = \mathrm{Linear}(x')$

reshape, up sample, with LN

$\mathrm{out} = \mathrm{nn.PixelShuffle}(\mathrm{DWConv}_2(V))$

Model configurations

Table 9: Detailed architecture specifications

Module	Output	ResTv2-T	ResTv2-S	ResTv2-B	ResTv2-L
stem	56 × 56	Patch Embedding: Conv-3_C/2_2, Conv-3_C_2, Conv-1_C_1, PA			
stage1	56 × 56	[Ev2_1_8, MLP_96] ×1	[Ev2_1_8, MLP_96] ×1	[Ev2_1_8, MLP_96] ×1	[Ev2_2_8, MLP_128] ×2
stage2	28 × 28	Patch Embedding: Conv-3_2C_2, PA			
		[Ev2_2_4, MLP_192] ×2	[Ev2_2_4, MLP_192] ×2	[Ev2_2_4, MLP_192] ×3	[Ev2_4_4, MLP_256] ×3
stage3	14 × 14	Patch Embedding: Conv-3_4C_2, PA			
		[Ev2_4_2, MLP_384] ×6	[Ev2_4_2, MLP_384] ×12	[Ev2_4_2, MLP_384] ×16	[Ev2_8_2, MLP_512] ×16
stage4	7 × 7	Patch Embedding: Conv-3_8C_2, PA			
		[Ev2_8_1, MLP_768] ×2	[Ev2_8_1, MLP_768] ×2	[Ev2_8_1, MLP_768] ×3	[Ev2_16_1, MLP_1024] ×2
Classifier		Average Pooling, 1000D Fully-Connected Layer			
FLOPs		4.0G	5.8G	7.7G	13.5G

Block number in the first stage is set to 1 to save computation cost.

EMSAv2:heads=2, reduction ratio=8

MLP: in dim=512, hidden dim=512×4

ResTv2：Simpler, Faster and Stronger

Table 1: Classification accuracy on ImageNet-1k.

Model	Image Size	Params	FLOPs	Throughput	Top-1 (%)	Top-5 (%)
RegNetY-4G [30]	224^2	21M	4.0G	**1156**	79.4	94.7
ConvNeXt-T [24]	224^2	29M	4.5G	775	82.1	**95.9**
Swin-T [23]	224^2	28M	4.5G	755	81.3	95.5
Focal-T [41]	224^2	29M	4.9G	319	82.2	95.9
ResTv1-B [47]	224^2	30M	4.3G	673	81.6	95.7
ResTv2-T	224^2	30M	4.1G	826	**82.3**	95.5
ResTv2-T	384^2	30M	12.7G	319	**83.7**	**96.6**
RegNetY-8G [30]	224^2	39M	8.0G	591	79.9	94.9
ResTv2-S	224^2	41M	6.0G	**687**	**83.2**	**96.1**
ResTv2-S	384^2	41M	18.4G	256	**84.5**	**96.7**
ConvNeXt-S [24]	224^2	50M	8.7G	447	83.1	**96.4**
Swin-S [23]	224^2	50M	8.7G	437	83.2	96.2
Focal-S [41]	224^2	51M	9.4G	192	83.6	96.2
ResTv1-L [47]	224^2	52M	7.9G	429	83.6	96.3
ResTv2-B	224^2	56M	7.9G	**582**	**83.7**	96.3
ResTv2-B	384^2	56M	24.3G	210	**85.1**	**97.2**
RegNetY-16G [30]	224^2	84M	15.9G	334	80.4	95.1
ConvNeXt-B [24]	224^2	89M	15.4G	292	83.8	**96.7**
Swin-B [23]	224^2	88M	15.4G	278	83.5	96.5
Focal-B [41]	224^2	90M	16.4G	138	84.0	96.5
ResTv2-L	224^2	87M	13.8G	**415**	**84.2**	96.5
ConvNeXt-B [24]	384^2	89M	45.0G	96	85.1	**97.3**
Swin-B [23]	384^2	88M	47.1G	85	84.5	97.0
ResTv2-L	384^2	87M	42.4G	**141**	**85.4**	97.1

Conclusion：

- ResTv2 outperforms the Focal counterparts, averaging ×1.8 times inference speed acceleration.
- ResTv2-T outperforms Swin-T with +1.0% Top-1 accuracy and +9.4% throughput.
- Input resolution scaling from 224 to 384, average Top-1 accuracy increased by +1.4%.

Ablation Study

Table 2: Ablation experiments with ResTv2-T on ImageNet-1k (100 Epochs)

output of down-sample operation

(a) **Upsampling Targets.** Upsampling **V** works the best.

Targets	Top-1 (%)	Top-5 (%)
w/o	79.04	94.61
x'	79.64 **+0.6**	94.90
K	80.03 **+1.0**	94.95
V	**80.33** **+1.3**	**95.06**

(b) **Upsampling Strategies.** Pixel-Shuffle achieves better speed-accuracy trade-off.

Upsample	Params	FLOPs	Top-1 (%)	
w/o	**30.26M**	**4.08G**	79.04	
nearest	**30.26M**	**4.08G**	79.16	**+0.12**
bilinear	**30.26M**	**4.08G**	79.28	**+0.24**
pixel-shuffle	30.43M	4.10G	**80.33**	**+1.29**

With more blocks:
Cv2: 2->3->6->2
Cv3: 2->3->6->3

(c) **ConvNet or EMSA?** Both of them can boost the performance.

Branches	Params	FLOPs	Top-1 (%)
EMSA	30.26M	4.08G	79.04
ConvNet	**26.11M**	**3.56G**	77.18
ConvNetv2	26.67M	4.09G	77.91
EMSAv2	30.43M	4.10G	**80.33**
ConvNetv3	30.43M	4.54G	78.63

(d) **Positional Embedding.** Both RPE and PA work well, but PA is more flexible.

PE	Params	Top-1 (%)	
w/o	**30.42M**	79.94	
APE [11]	30.98M	79.99	
RPE [31]	30.48M	80.32	
PA [47]	30.43M	**80.33**	**+0.4**
PEG	30.43M	80.17	

Visualization

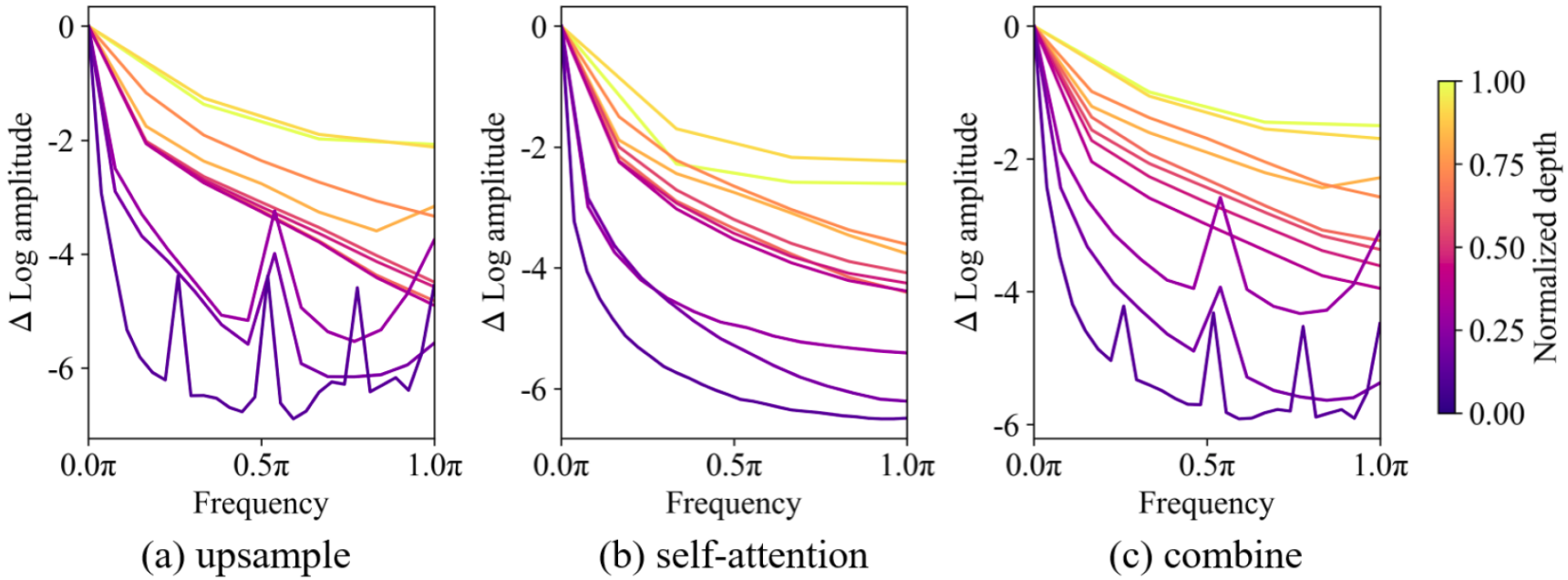


Figure 3: **Relative log amplitudes of Fourier transformed feature maps.** Δ Log amplitude is the difference between the log amplitude at normalized frequency 0.0π (center) and 1.0π (boundary).

- 11 different colored polylines represent 11 blocks in ResTv2-T.
- Only using half-diagonal components of shift Fourier results. 0.0π, 0.5π, and 1.0π can represent low-, medium-, and high-frequency, respectively.

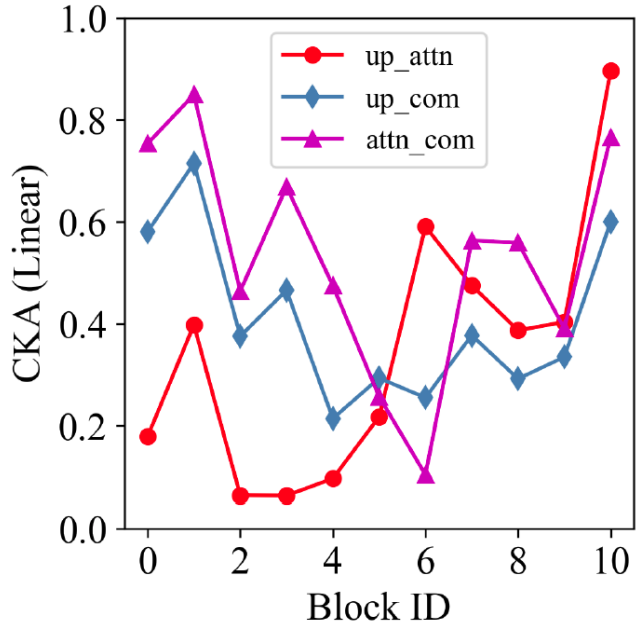


Figure 4: **Linear CKA Similarity** between EMSA, Upsample and EMSAv2 with ResTv2-T.

- "up" , "attn" and "com" are short of Upsample, SA and combine (i.e., EMSAv2) , respectively.

Downstream Experiments

Table 3: **Object detection results of fine-tuning styles on COCO val2017 with ResTv2-T using Mask RCNN.** Inference "ms/iter" is measured on a V100 GPU, and FLOPs are calculated with 1k validation images.

(a) Object detection results.

Style	Params.	FLOPs	ms/iter	AP^{box}	AP^{mask}
Win	49.94M	205.2G	149.6	43.95	40.42
CWin	49.96M	212.5G	150.7	44.07	40.44
HWin	49.94M	218.9G	135.9	45.02	41.56
Global	49.94M	229.7G	**79.9**	**46.13**	**42.03**

(b) Detailed GFLOPs Analysis

Style	Conv	Linear	Matmul	Others
Win	119.09	82.00	3.69	0.47
CWin	126.29	82.00	3.69	0.47
HWin	118.57	79.71	20.17	0.45
Global	116.95	75.70	36.66	0.42

Window attention greatly reduces theoretical matrix multiply FLOPs with the cost of decreasing the computational density, resulting in lower actual speed.

- Win：Window Attention, constraining all EMSAv2 modules into fixed windows: [64, 32, 16, 8].
- CWin, HWin：After the last block in each stage, CWin adds DWConv-7×7, and HWin replaces Win with global attention to enable information to communicate across windows.
- Global：Global Attention, directly adopts ViTs into downstream tasks.

Downstream Experiments

Table 4: **COCO object detection and segmentation results using Mask-RCNN.** We measure FPS on one V100 GPU. FLOPs are calculated with image size (1280, 800).

Backbones	AP^{box}	AP^{mask}	Params.	FLOPs	FPS
ResNet-50 [15]	41.0	37.1	44.2M	260G	24.1
ConvNeXt-T [24]	46.2	41.7	48.1M	262G	23.4
Swin-T [23]	46.0	41.6	47.8M	264G	21.8
ResTv2-T	**47.6**	**43.2**	49.9M	253G	**25.0**
ResNet-101 [15]	42.8	38.5	63.2M	336G	13.5
Swin-S [23]	48.5	43.3	69.1M	354G	17.4
ResTv2-S	**48.1**	**43.3**	60.7M	290G	**21.3**
ResTv2-B	**48.7**	**43.9**	75.5M	328G	**18.3**

Table 5: **ADE20K validation results using UperNet.** Following Swin, we report mIoU results with multiscale testing. FLOPs are based on input sizes of (2048, 512).

Backbones	input crop.	mIoU	Params.	FLOPs	FPS
ResNet-50 [15]	512^2	42.8	66.5M	952G	23.4
ConvNeXt-T [24]	512^2	46.7	60.2M	939G	19.9
Swin-T [23]	512^2	45.8	59.9 M	941G	21.1
ResTv2-T	512^2	**47.3**	62.1M	977G	**22.4**
ResNet-101 [15]	512^2	44.9	85.5M	1029G	20.3
ConvNeXt-S [24]	512^2	49.0	81.9M	1027G	15.3
Swin-S [23]	512^2	49.2	81.3M	1038G	14.7
ResTv2-S	512^2	**49.2**	72.9M	1035G	**20.0**
ResTv2-B	512^2	**49.6**	87.6M	1095G	**19.2**

- Experimental results on COCO detection, and ADE20K semantic segmentation show that the proposed ResTv2 can outperform the recently state-of-the-art backbones by a large margin.

Conclusion

- We proposed ResTv2, a simpler, faster, and stronger multi-scale vision Transformer for image recognition.
- ResTv2 adopts an up-sample module in EMSAv2 to reconstruct the lost information due to the down-sample operation.
- Different techniques for better applying ResTv2 to downstream tasks. Results show that the theoretical FLOPs are not a good reflection of actual speed, particularly running on GPUs.

Thank You

敬请各位老师同学批评指正！

诚耀百廿 雄创一流