Swin Transformer

Swin Transformer: Hierarchical Vision Transformer using Shifted Windows

Ze Liu[†*] Yutong Lin[†*] Yue Cao[*] Han Hu[*‡] Yixuan Wei[†]
Zheng Zhang Stephen Lin Baining Guo
Microsoft Research Asia
{v-zeliu1,v-yutlin,yuecao,hanhu,v-yixwe,zhez,stevelin,bainguo}@microsoft.com

2021 ICCV

ICCV 2021 best paper

val). Its performance surpasses the previous state-of-the-art by a large margin of +2.7 box AP and +2.6 mask AP on COCO, and +3.2 mIoU on ADE20K, demonstrating the potential of Transformer-based models as vision backbones. The hierarchical design and the shifted window approach also prove beneficial for all-MLP architectures. The code and models are publicly available at https://github.com/microsoft/Swin-Transformer.

Swin Transformer

- State of the Art | Object Detection on COCO test-dev (using additional training data)
- State of the Art | Instance Segmentation on COCO test-dev (using additional training data)
- State of the Art | Object Detection on COCO minival (using additional training data)
- State of the Art | Instance Segmentation on COCO minival (using additional training data)
- Ranked #8 | Semantic Segmentation on ADE20K (using additional training data)
- Ranked #9 | Semantic Segmentation on ADE20K val
- State of the Art | Action Recognition on Something-Something V2 (using additional training data)
- Ranked #2 | Action Classification on Kinetics-400 (using additional training data)
- Ranked #2 | Action Classification on Kinetics-600 (using additional training data)

论文地址：https://arxiv.org/abs/2103.14030
源码地址：https://github.com/microsoft/Swin-Transformer
博文地址：https://blog.csdn.net/qq_37541097/article/details/121119988

Swin Transformer

目录

- 1 网络整体框架
- 2 Patch Merging详解
- 3 W-MSA详解
  - MSA模块计算量
  - W-MSA模块计算量
- 4 SW-MSA详解
- 5 Relative Position Bias详解
- 6 模型详细配置参数

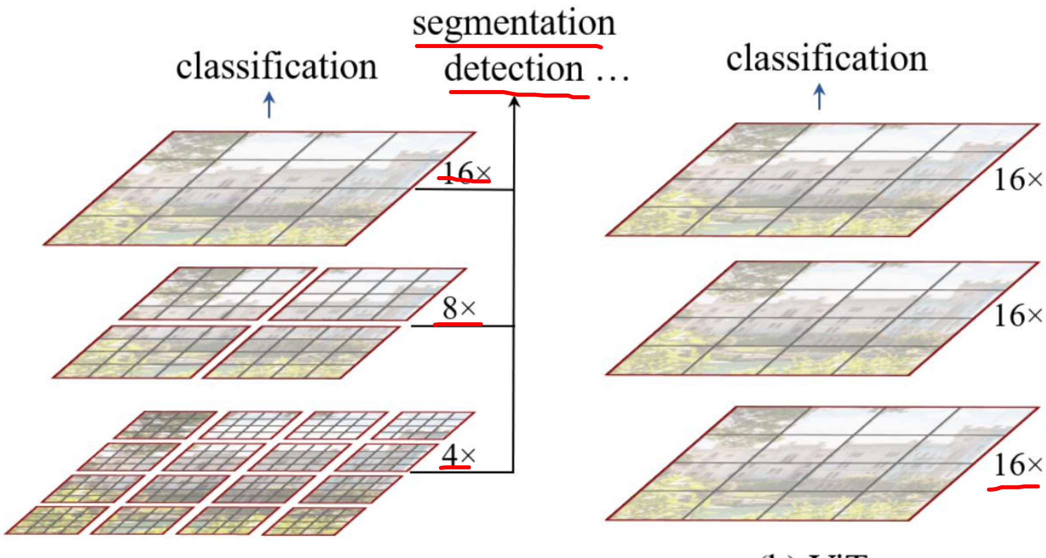
Swin Transformer

State of the Art | Object Detection on COCO test-dev (using additional training data)
State of the Art | Instance Segmentation on COCO test-dev (using additional training data)
State of the Art | Object Detection on COCO minival (using additional training data)
State of the Art | Instance Segmentation on COCO minival (using additional training data)
Ranked #8 | Semantic Segmentation on ADE20K (using additional training data)
Ranked #9 | Semantic Segmentation on ADE20K val
State of the Art | Action Recognition on Something-Something V2 (using additional training data)
Ranked #2 | Action Classification on Kinetics-400 (using additional training data)
Ranked #2 | Action Classification on Kinetics-600 (using additional training data)

Swin Transformer

网络整体框架

Swin Transformer与Vision Transformer对比



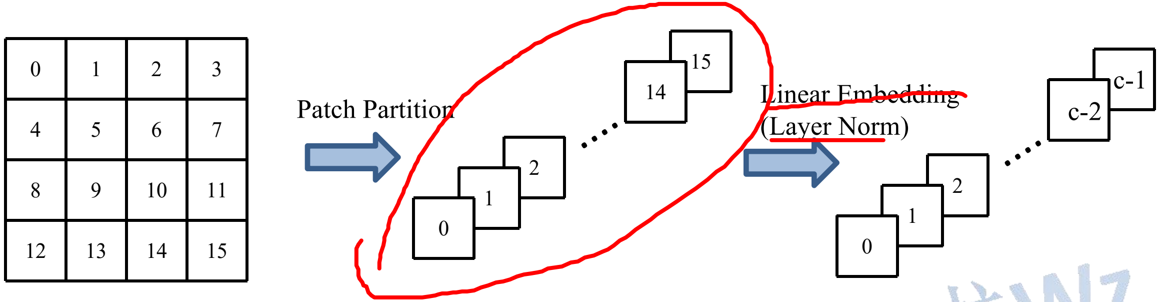
(a) Swin Transformer (ours) (b) ViT

method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
(a) Regular ImageNet-1K trained models					
RegNetY-4G [48]	224^2	21M	4.0G	1156.7	80.0
RegNetY-8G [48]	224^2	39M	8.0G	591.6	81.7
RegNetY-16G [48]	224^2	84M	16.0G	334.7	82.9
EffNet-B3 [58]	300^2	12M	1.8G	732.1	81.6
EffNet-B4 [58]	380^2	19M	4.2G	349.4	82.9
EffNet-B5 [58]	456^2	30M	9.9G	169.1	83.6
EffNet-B6 [58]	528^2	43M	19.0G	96.9	84.0
EffNet-B7 [58]	600^2	66M	37.0G	55.1	84.3
ViT-B/16 [20]	384^2	86M	55.4G	85.9	77.9
ViT-L/16 [20]	384^2	307M	190.7G	27.3	76.5
DeiT-S [63]	224^2	22M	4.6G	940.4	79.8
DeiT-B [63]	224^2	86M	17.5G	292.3	81.8
DeiT-B [63]	384^2	86M	55.4G	85.9	83.1
Swin-T	224^2	29M	4.5G	755.2	81.3
Swin-S	224^2	50M	8.7G	436.9	83.0
Swin-B	224^2	88M	15.4G	278.1	83.5
Swin-B	384^2	88M	47.0G	84.7	84.5
(b) ImageNet-22K pre-trained models					
method	image size	#param.	FLOPs	throughput (image / s)	ImageNet top-1 acc.
R-101x3 [38]	384^2	388M	204.6G	-	84.4
R-152x4 [38]	480^2	937M	840.5G	-	85.4
ViT-B/16 [20]	384^2	86M	55.4G	85.9	84.0
ViT-L/16 [20]	384^2	307M	190.7G	27.3	85.2
Swin-B	224^2	88M	15.4G	278.1	85.2
Swin-B	384^2	88M	47.0G	84.7	86.4
Swin-L	384^2	197M	103.9G	42.1	87.3

Table 1. Comparison of different backbones on ImageNet-1K classification. Throughput is measured using the GitHub repository of [68] and a V100 GPU, following [63].

Swin Transformer

网络整体框架



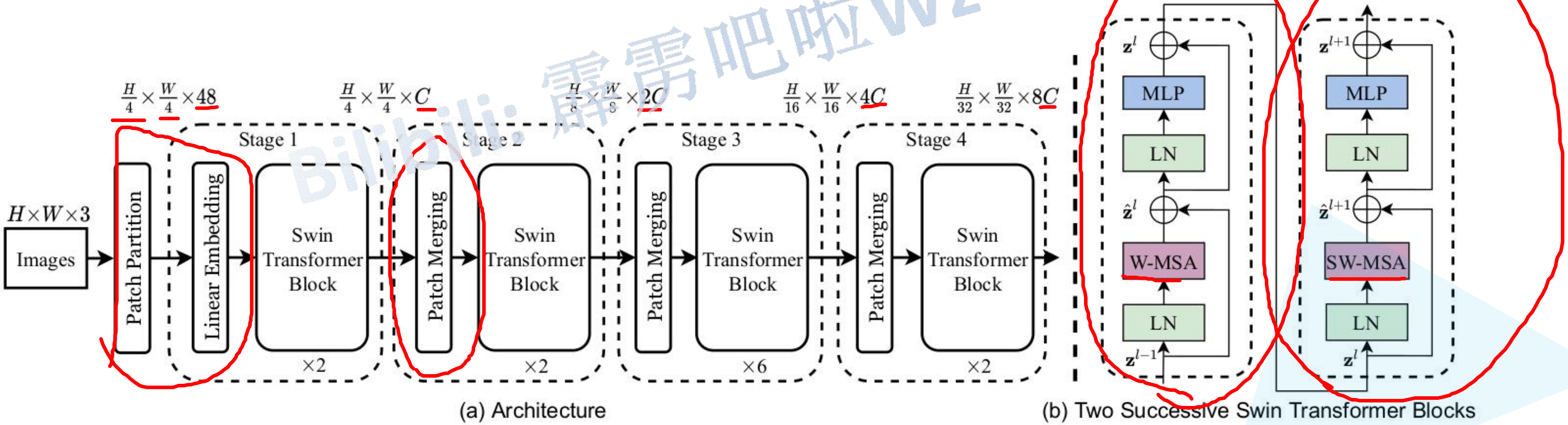
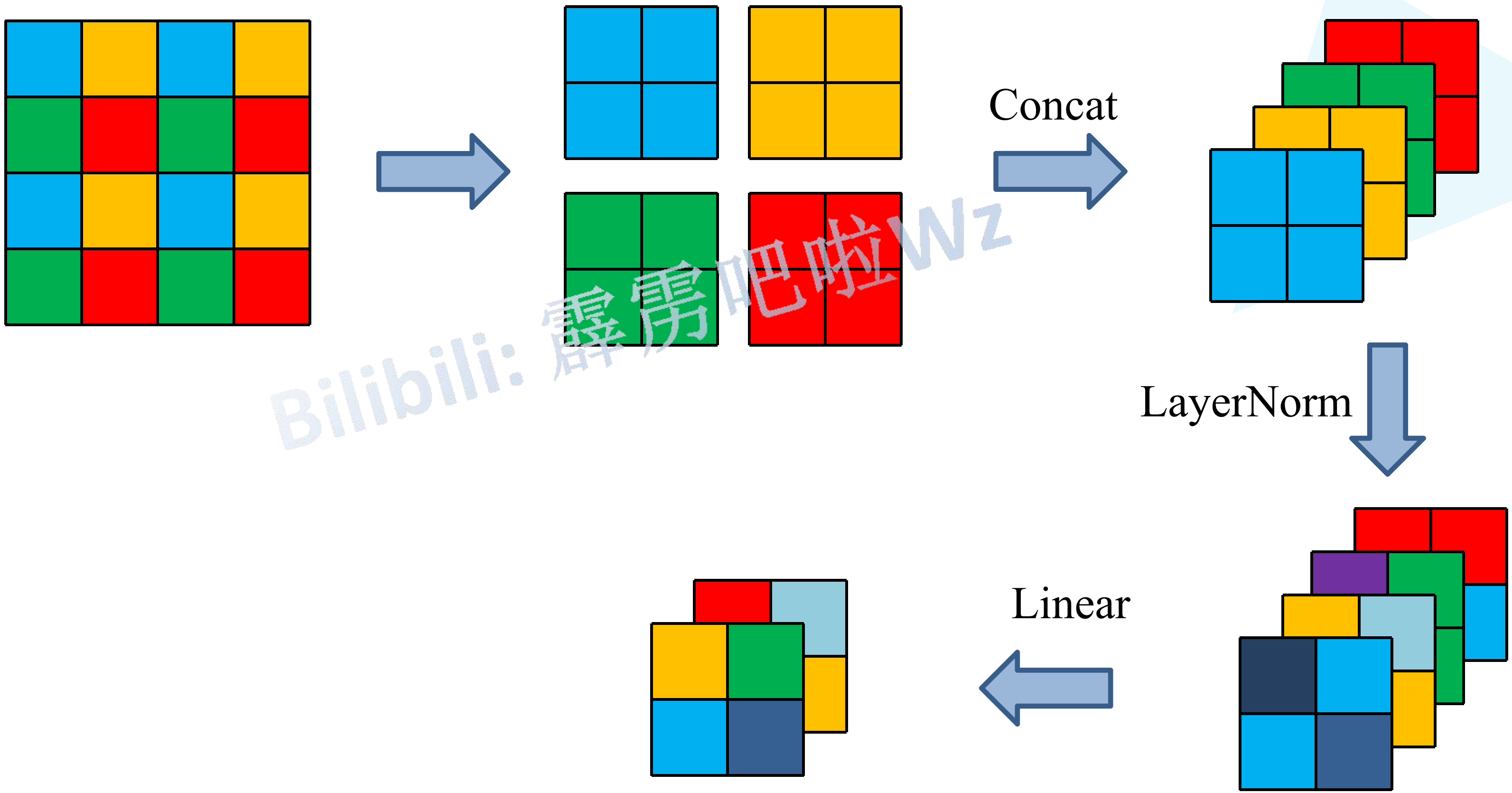


Figure 3. (a) The architecture of a Swin Transformer (Swin-T); (b) two successive Swin Transformer Blocks (notation presented with Eq. (3)). W-MSA and SW-MSA are multi-head self attention modules with regular and shifted windowing configurations, respectively.

Swin Transformer

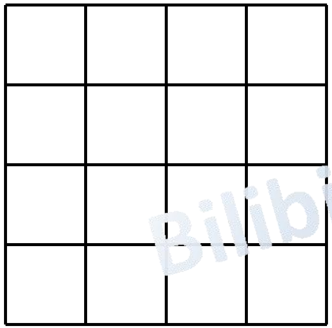
Patch Merging



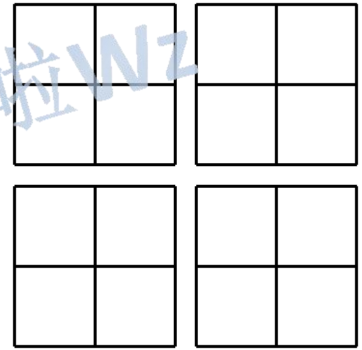
Swin Transformer

W-MSA

目的：减少计算量
缺点：窗口之间无法进行信息交互



Multi-head Self-Attention



Windows Multi-head Self-Attention

Swin Transformer

W-MSA

$$\Omega(\text{MSA}) = 4hwC^2 + 2(hw)^2C, \tag{1}$$

$$\Omega(\text{W-MSA}) = 4hwC^2 + 2M^2hwC, \tag{2}$$

- h代表feature map的高度
- w代表feature map的宽度
- C代表feature map的深度
- M代表每个窗口（Windows）的大小

h=w=112
M=7
C=128

节省：40124743680 FLOPs

W-MSA

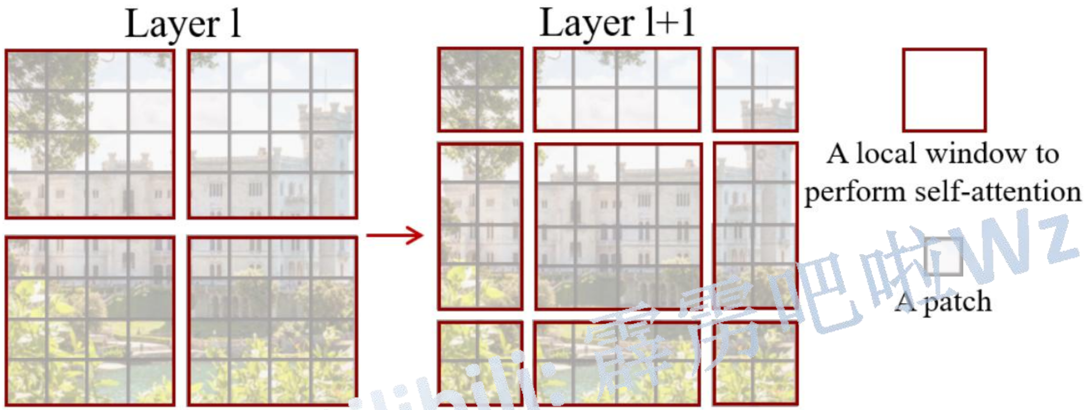
$$A^{a \times b} \cdot B^{b \times c}$$

FLOPs: $a \times b \times c$

Swin Transformer

Shifted Window

目的：实现不同Window之间的信息交互



Shifted Windows Multi-Head Self-Attention（SW-MSA）

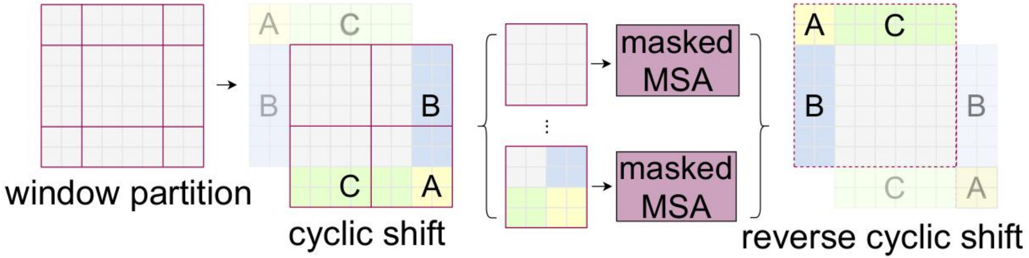
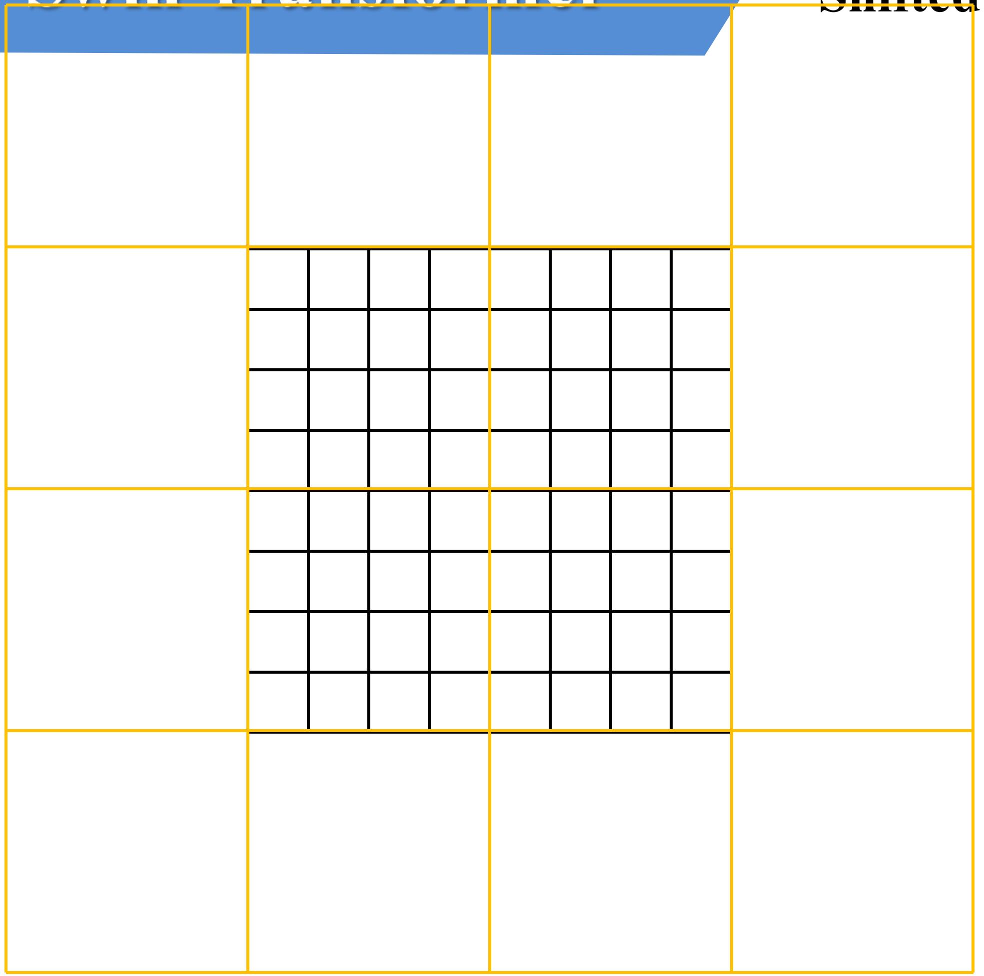


Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

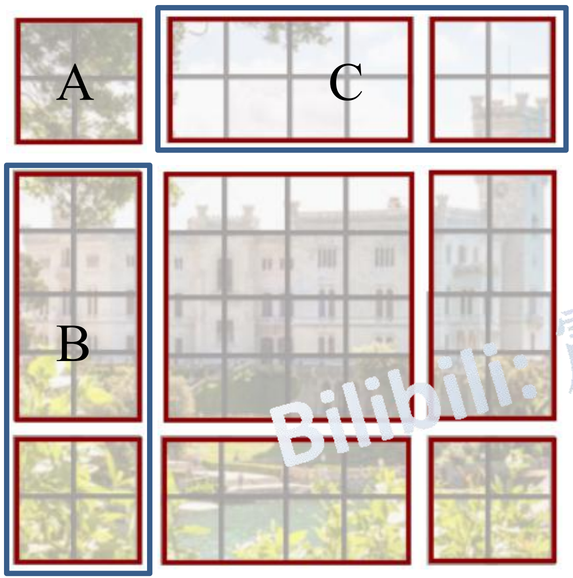
Swin Transformer

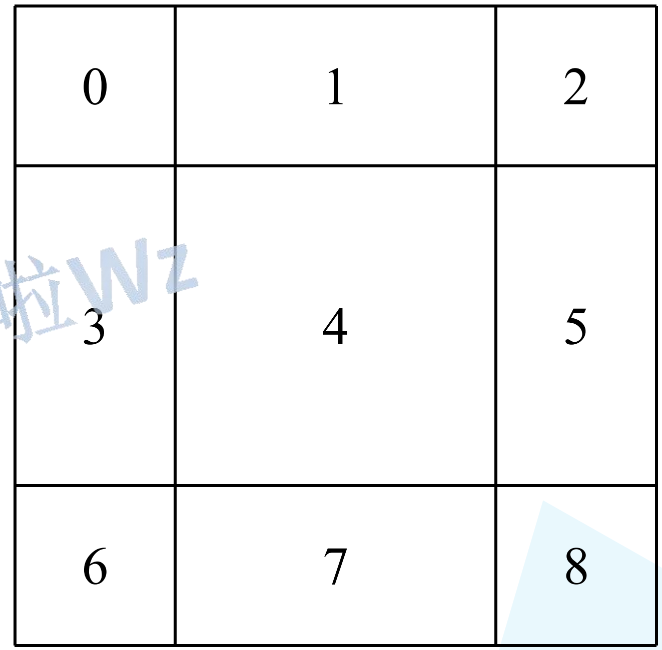
Shifted Window



Swin Transformer

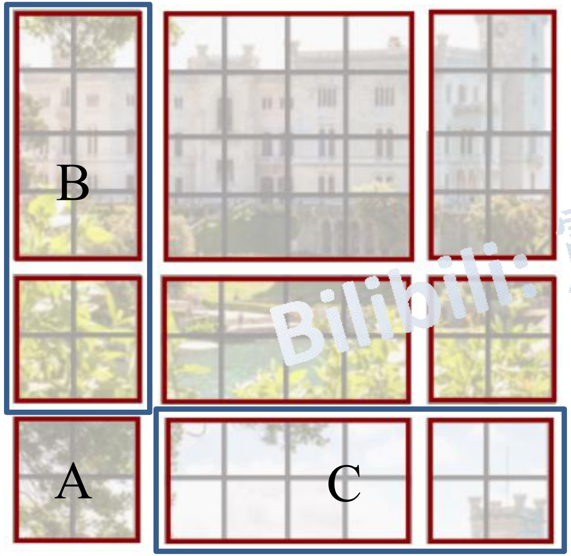
Shifted Window





Swin Transformer

Shifted Window

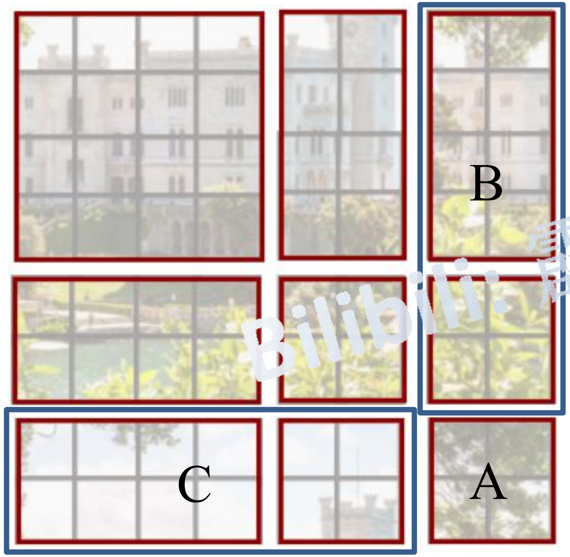


3	4	5
6	7	8
0	1	2

Swin Transformer

Shifted Window

$$\mathrm{Attention}(Q, K, V) = \mathrm{SoftMax}(QK^T/\sqrt{d} + B)V, \quad (4)$$

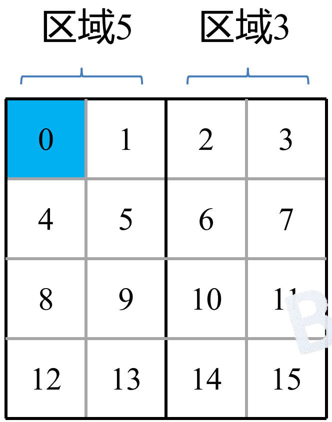


4	5	3
7	8	6
1	2	0

Swin Transformer

Shifted Window

$$\text{Attention}(Q,K,V) = \text{SoftMax}(QK^T/\sqrt{d}+B)V, \quad (4)$$



$$\alpha_{0,0}\ \alpha_{0,1}\ \underbrace{\boxed{\alpha_{0,2}\ \alpha_{0,3}}}_{-100}\ \alpha_{0,4}\ \alpha_{0,5}\ \underbrace{\boxed{\alpha_{0,6}\ \alpha_{0,7}}}_{-100}\ \alpha_{0,8}\ \alpha_{0,9}\ \underbrace{\boxed{\alpha_{0,10}\ \alpha_{0,11}}}_{-100}\ \alpha_{0,12}\ \alpha_{0,13}\ \underbrace{\boxed{\alpha_{0,14}\ \alpha_{0,15}}}_{-100}$$

$$\Big\downarrow \ \boxed{\text{Softmax}}$$

$$\hat{\alpha}_{0,0}\ \hat{\alpha}_{0,1}\ \boxed{0\ \ 0}\ \hat{\alpha}_{0,4}\ \hat{\alpha}_{0,5}\ \boxed{0\ \ 0}\ \hat{\alpha}_{0,8}\ \hat{\alpha}_{0,9}\ \boxed{0\ \ 0}\ \hat{\alpha}_{0,12}\ \hat{\alpha}_{0,13}\ \boxed{0\ \ 0}$$

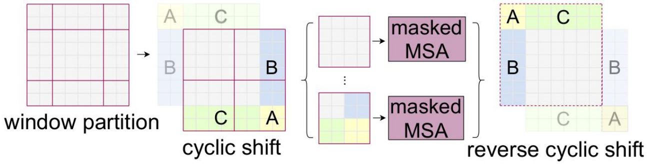


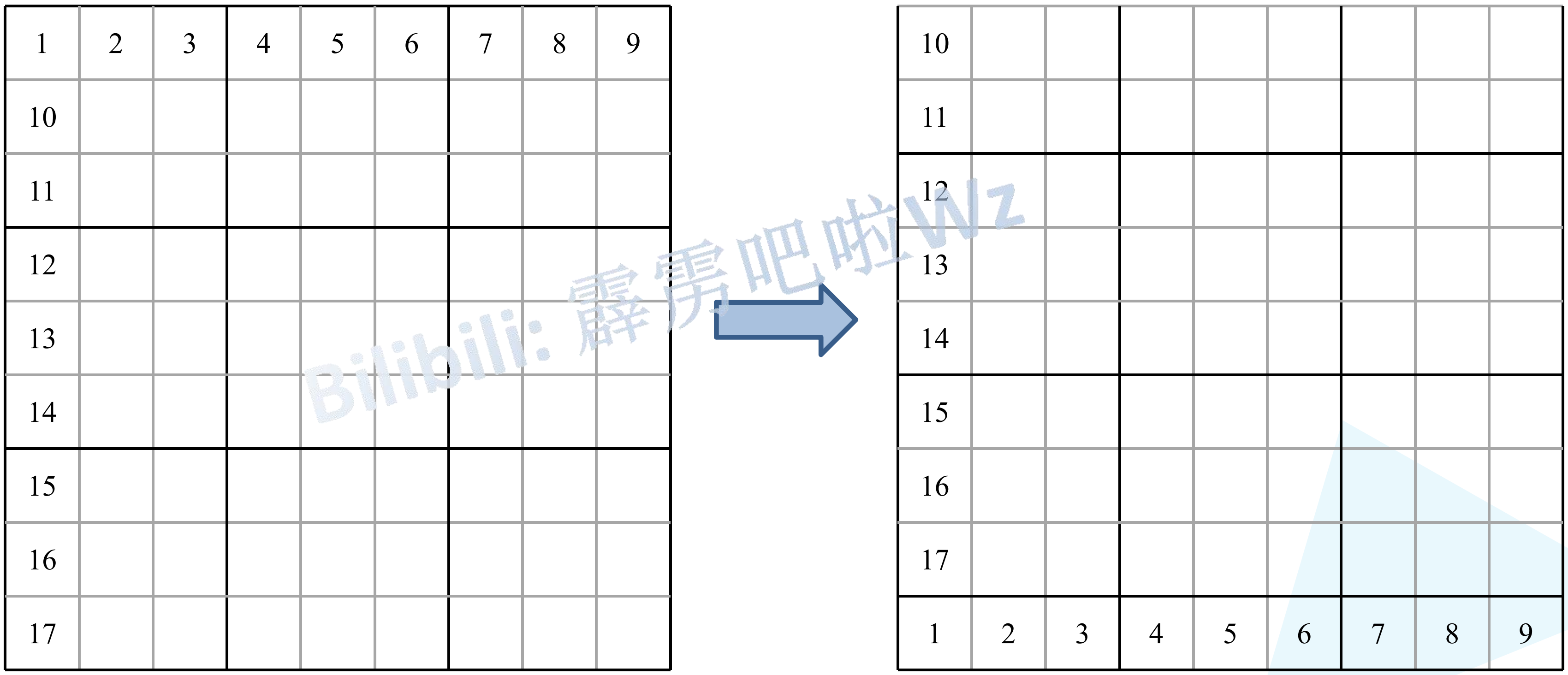
Figure 4. Illustration of an efficient batch computation approach for self-attention in shifted window partitioning.

注意，全部计算完后需要将数据挪回到原来的位置上

Swin Transformer

Shifted Window

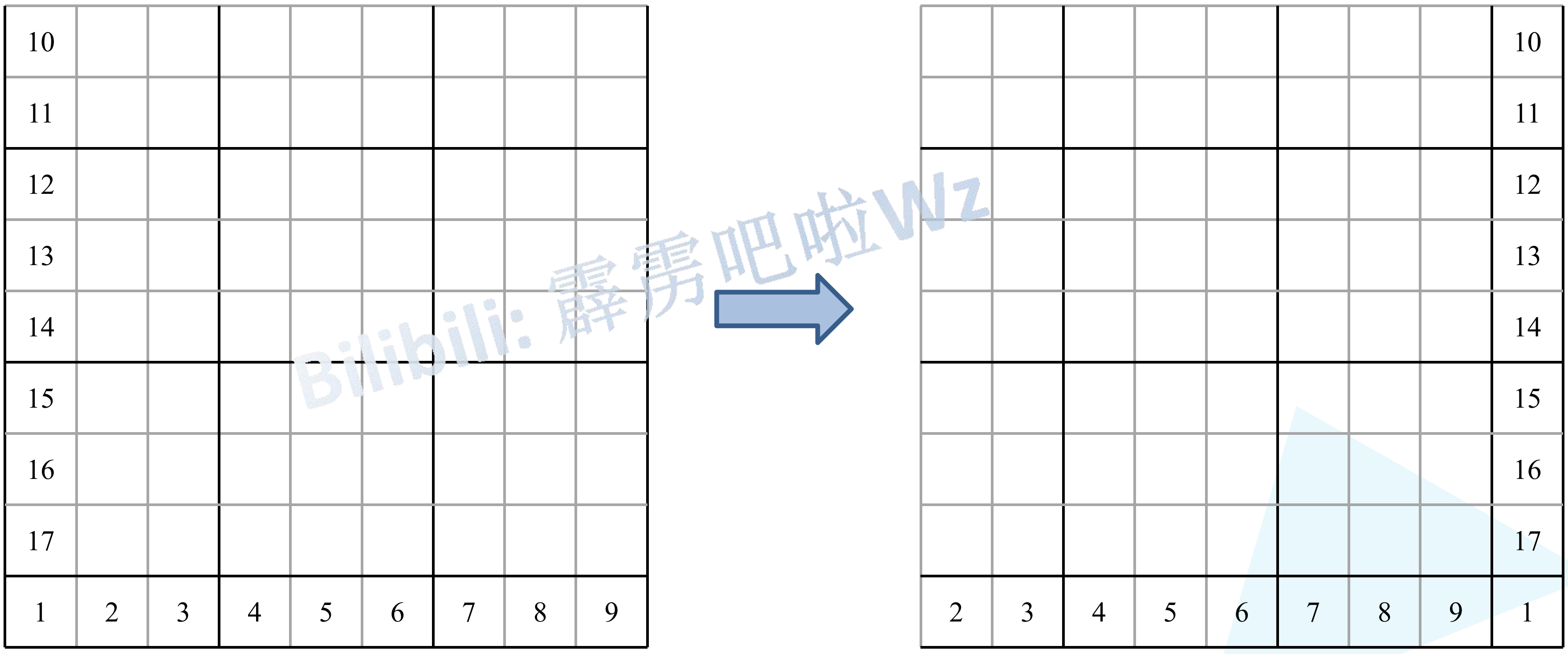
$$\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{M}{2} \right\rfloor$$



Swin Transformer

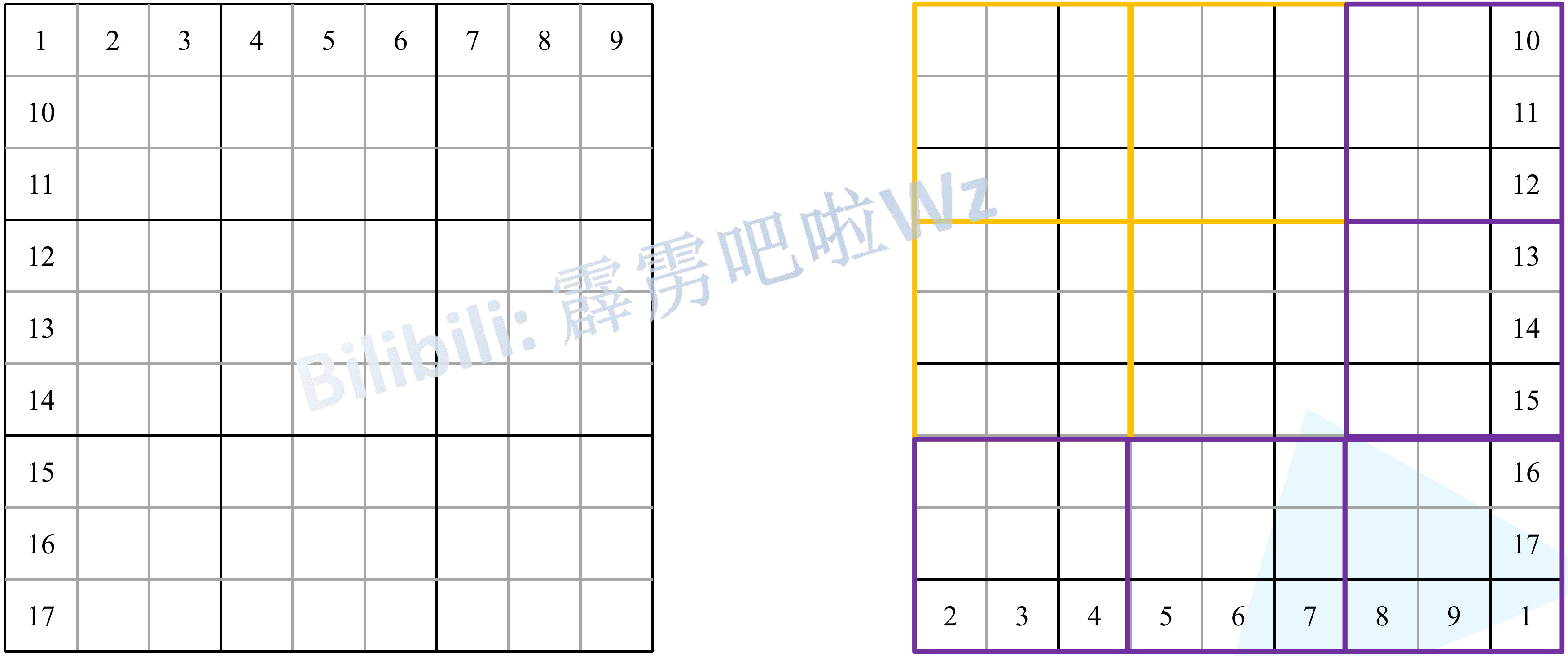
Shifted Window

$$\left\lfloor \frac{M}{2} \right\rfloor, \left\lfloor \frac{M}{2} \right\rfloor$$



Swin Transformer

Shifted Window



Swin Transformer

Relative position bias

$$\text{Attention}(Q, K, V) = \text{SoftMax}(QK^T/\sqrt{d} + B)V, \quad (4)$$

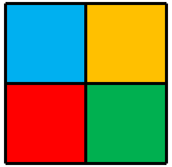
	ImageNet		COCO		ADE20k
	top-1	top-5	AP^{box}	AP^{mask}	mIoU
w/o shifting	80.2	95.1	47.7	41.5	43.3
shifted windows	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**
no pos.	80.1	94.9	49.2	42.6	43.8
abs. pos.	80.5	95.2	49.0	42.4	43.2
abs.+rel. pos.	81.3	95.6	50.2	43.4	44.0
rel. pos. w/o app.	79.3	94.7	48.2	41.9	44.1
rel. pos.	**81.3**	**95.6**	**50.5**	**43.7**	**46.1**

Table 4. Ablation study on the *shifted windows* approach and different position embedding methods on three benchmarks, using the Swin-T architecture. w/o shifting: all self-attention modules adopt regular window partitioning, without *shifting*; abs. pos.: absolute position embedding term of ViT; rel. pos.: the default settings with an additional relative position bias term (see Eq. (4)); app.: the first scaled dot-product term in Eq. (4).

Swin Transformer

Relative position bias

feature map



0, 0	0, 1
1, 0	1, 1

绝对位置索引

第一个数字代表行
第二个数字代表列

蓝色q和所有k
匹配时**相对**位置索引

0, 0	0, -1
-1, 0	-1,-1

橙色q和所有k
匹配时**相对**位置索引

0, 1	0, 0
-1, 1	-1, 0

红色q和所有k
匹配时**相对**位置索引

1, 0	1, -1
0, 0	0, -1

绿色q和所有k
匹配时**相对**位置索引

1, 1	1, 0
0, 1	0, 0

0, 0	0, -1	-1, 0	-1,-1
0, 1	0, 0	-1, 1	-1, 0
1, 0	1, -1	0, 0	0, -1
1, 1	1, 0	0, 1	0, 0

Swin Transformer

Relative position bias

0, 0	0, -1	-1, 0	-1,-1
0, 1	0, 0	-1, 1	-1, 0
1, 0	1, -1	0, 0	0, -1
1, 1	1, 0	0, 1	0, 0

偏移从0开始，
行、列标加上M-1

1, 1	1, 0	0, 1	0, 0
1, 2	1, 1	0, 2	0, 1
2, 1	2, 0	1, 1	1, 0
2, 2	2, 1	1, 2	1, 1

Swin Transformer

Relative position bias

1, 1	1, 0	0, 1	0, 0
1, 2	1, 1	0, 2	0, 1
2, 1	2, 0	1, 1	1, 0
2, 2	2, 1	1, 2	1, 1

行标乘上2M-1 →

3, 1	3, 0	0, 1	0, 0
3, 2	3, 1	0, 2	0, 1
6, 1	6, 0	3, 1	3, 0
6, 2	6, 1	3, 2	3, 1

Swin Transformer

Relative position bias

3, 1	3, 0	0, 1	0, 0
3, 2	3, 1	0, 2	0, 1
6, 1	6, 0	3, 1	3, 0
6, 2	6, 1	3, 2	3, 1

行、列标相加

4	3	1	0
5	4	2	1
7	6	4	3
8	7	5	4

Relative position bias

relative position bias table

(2M-1) X (2M-1)

0.1	0.2	0.3	0.8	0.1	0.6	0.4	0.4	0.7
0	1	2	3	4	5	6	7	8

$$\mathrm{Attention}(Q, K, V) = \mathrm{SoftMax}(QK^T/\sqrt{d} + B)V,$$

4	3	1	0
5	4	2	1
7	6	4	3
8	7	5	4

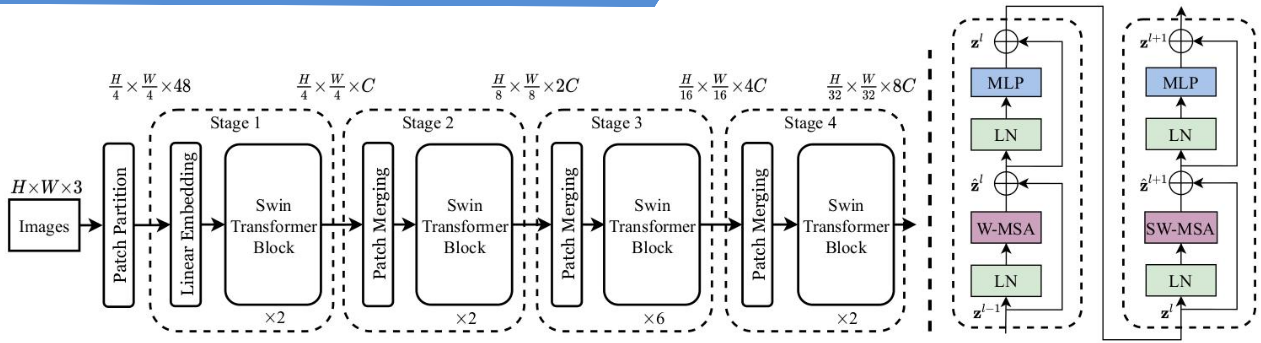
relative position index

0.1	0.8	0.2	0.1
0.6	0.1	0.3	0.2
0.4	0.4	0.1	0.8
0.7	0.4	0.6	0.1

relative position bias

Swin Transformer

模型详细配置参数



(a) Architecture (b) Two Successive Swin Transformer Blocks

	downsp. rate (output size)	Swin-T	Swin-S	Swin-B	Swin-L
stage 1	4× (56×56)	concat 4×4, 96-d, LN	concat 4×4, 96-d, LN	concat 4×4, 128-d, LN	concat 4×4, 192-d, LN
		[win. sz. 7×7, dim 96, head 3] × 2	[win. sz. 7×7, dim 96, head 3] × 2	[win. sz. 7×7, dim 128, head 4] × 2	[win. sz. 7×7, dim 192, head 6] × 2
stage 2	8× (28×28)	concat 2×2, 192-d , LN	concat 2×2, 192-d , LN	concat 2×2, 256-d , LN	concat 2×2, 384-d , LN
		[win. sz. 7×7, dim 192, head 6] × 2	[win. sz. 7×7, dim 192, head 6] × 2	[win. sz. 7×7, dim 256, head 8] × 2	[win. sz. 7×7, dim 384, head 12] × 2
stage 3	16× (14×14)	concat 2×2, 384-d , LN	concat 2×2, 384-d , LN	concat 2×2, 512-d , LN	concat 2×2, 768-d , LN
		[win. sz. 7×7, dim 384, head 12] × 6	[win. sz. 7×7, dim 384, head 12] × 18	[win. sz. 7×7, dim 512, head 16] × 18	[win. sz. 7×7, dim 768, head 24] × 18
stage 4	32× (7×7)	concat 2×2, 768-d , LN	concat 2×2, 768-d , LN	concat 2×2, 1024-d , LN	concat 2×2, 1536-d , LN
		[win. sz. 7×7, dim 768, head 24] × 2	[win. sz. 7×7, dim 768, head 24] × 2	[win. sz. 7×7, dim 1024, head 32] × 2	[win. sz. 7×7, dim 1536, head 48] × 2

Table 7. Detailed architecture specifications.

沟通方式

1.github

https://github.com/WZMIAOMIAO/deep-learning-for-image-processing

2.bilibili

https://space.bilibili.com/18161609/channel/index

3.CSDN

https://blog.csdn.net/qq_37541097/article/details/103482003