



**re-parameterization**



# CONTENTS

1

**Motivation**

2

**ACNet**  
**RepVGG**  
**DBB**

3

**RefConv**



# Motivation





# ● 重参化

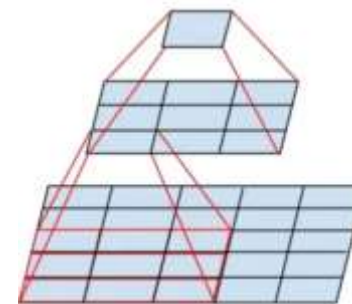
## 重参化：

1. **结构重参数化(SR, 包括 ACB、RepVGG 和 DBB)**
2. **权重重参数化(WR, 包括 WN、CWN 和 OWN)**

结构重参数化 (Structural Re-parameterization) 在训练中向模型添加额外的分支以提高性能，然后等效地将训练结构简化为与原始模型相同的推理

权重重参数化 (Weight Re-parameterization) 指改变神经网络模型的权重参数，以提高模型的训练效果或泛化性能。

# VGG式模型



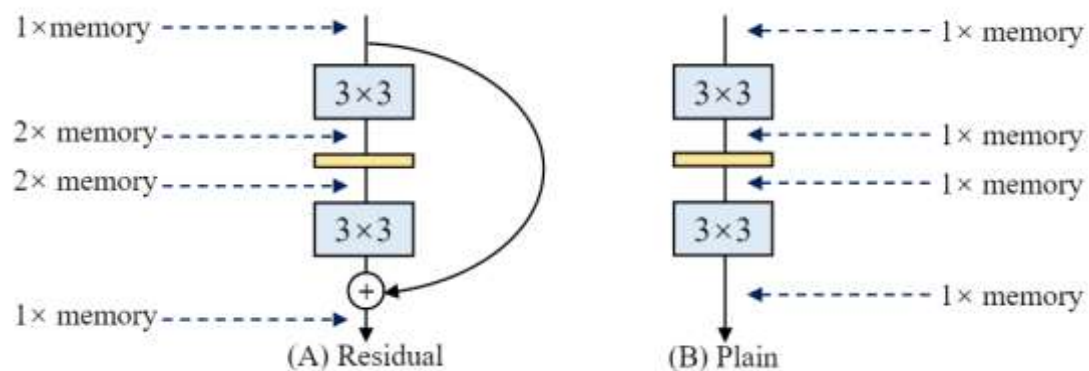
Kernel size	Theoretical FLOPs (B)	Time usage (ms)	Theoretical TFLOPS
$1 \times 1$	420.9	84.5	9.96
$3 \times 3$	3788.1	198.8	<b>38.10</b>
$5 \times 5$	10522.6	2092.5	10.57
$7 \times 7$	20624.4	4394.3	9.38

优:

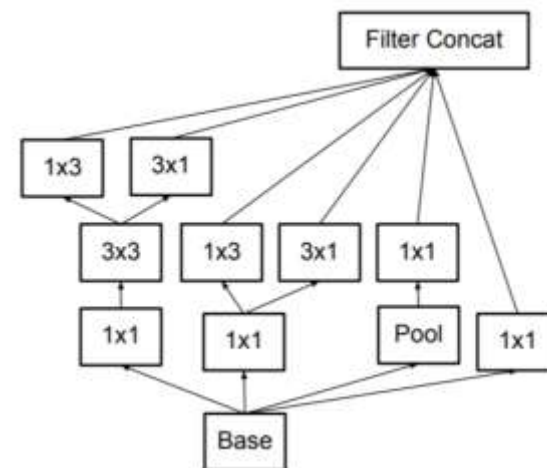
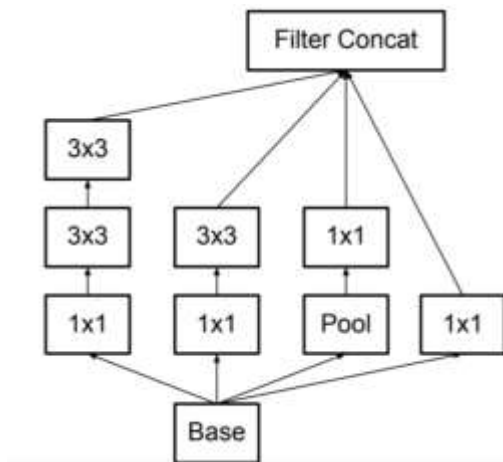
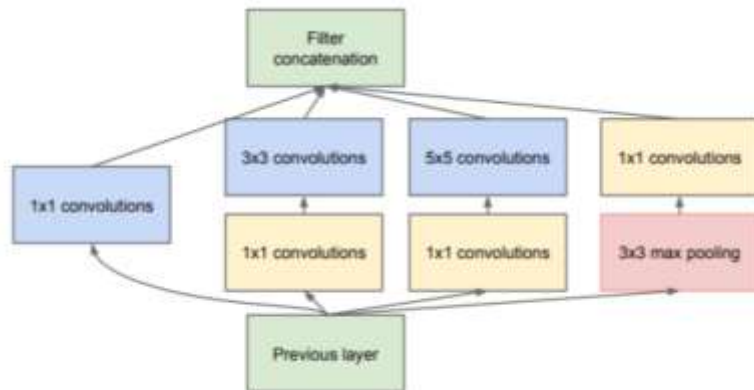
1. 并行度高, 速度快
2. 省显存
3. 对定制硬件特别友好

缺:

性能不足



# Inception



## Inception:

优：不同尺度和复杂性的各种路径的多分支拓扑和组合可以丰富特征空间，提高性能。

缺：复杂的结构通常会减慢推理的速度

# re-parameterization

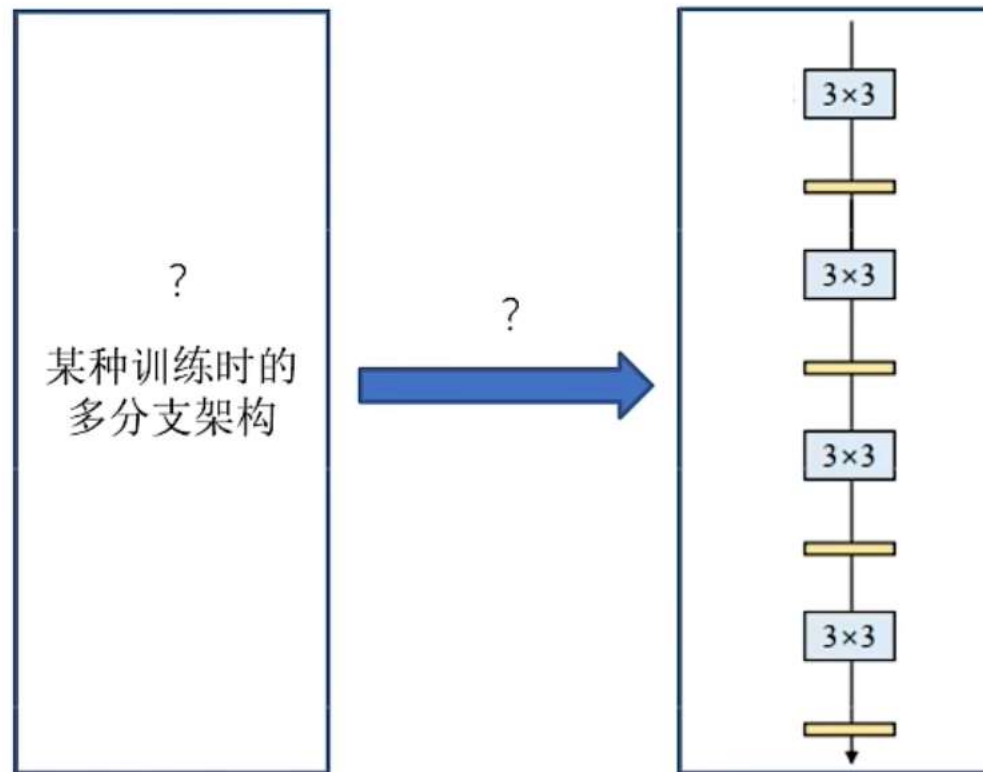
**多分支**模型训练出的精度高  
推理时效率低

**单路**模型训练出的精度低  
推理时效率高

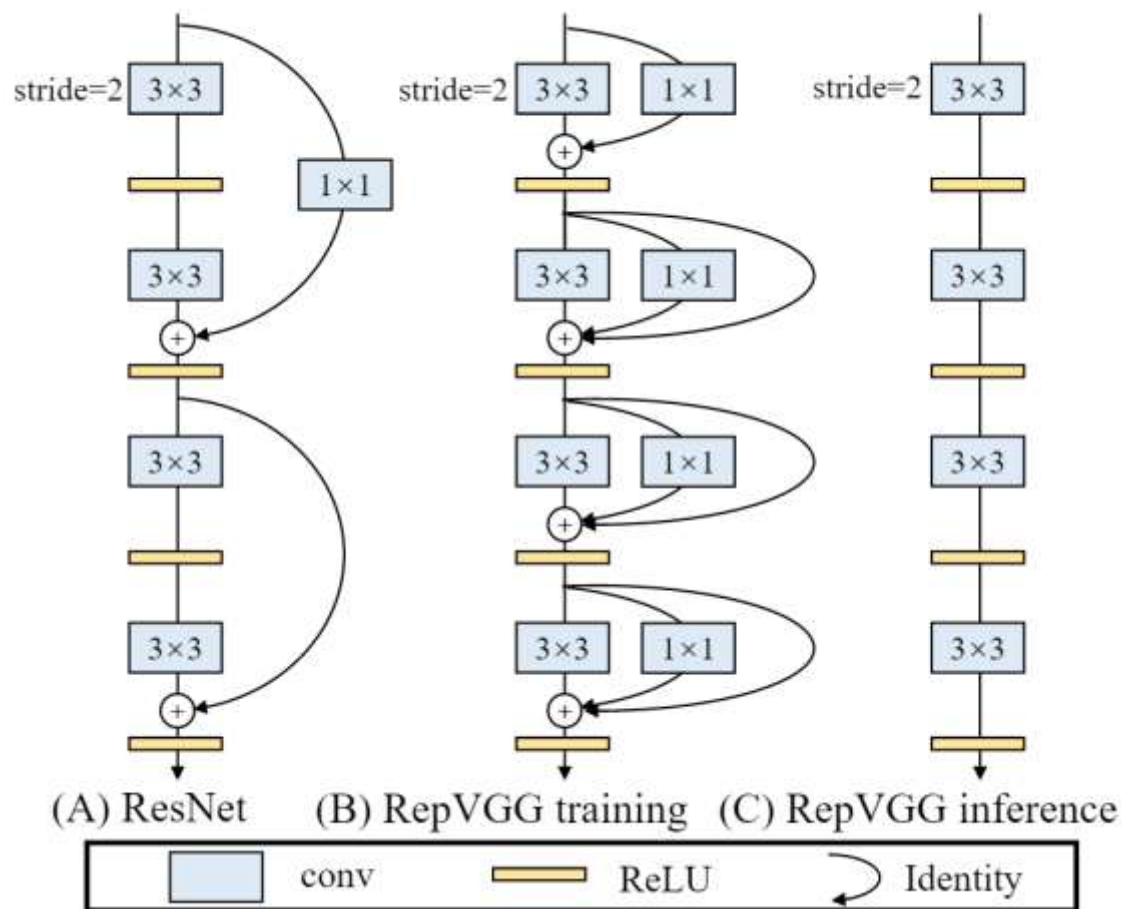
问题：

模型训练时的架构一定要与推理时的架构相同吗？

是否可能实现两个不同架构之间的等价转换？



# re-parameterization



## 结构重参数化的实质:

- 结构与参数一一对应
- 通过参数的等价转化实现结构的等价转换



02

**ACNet**

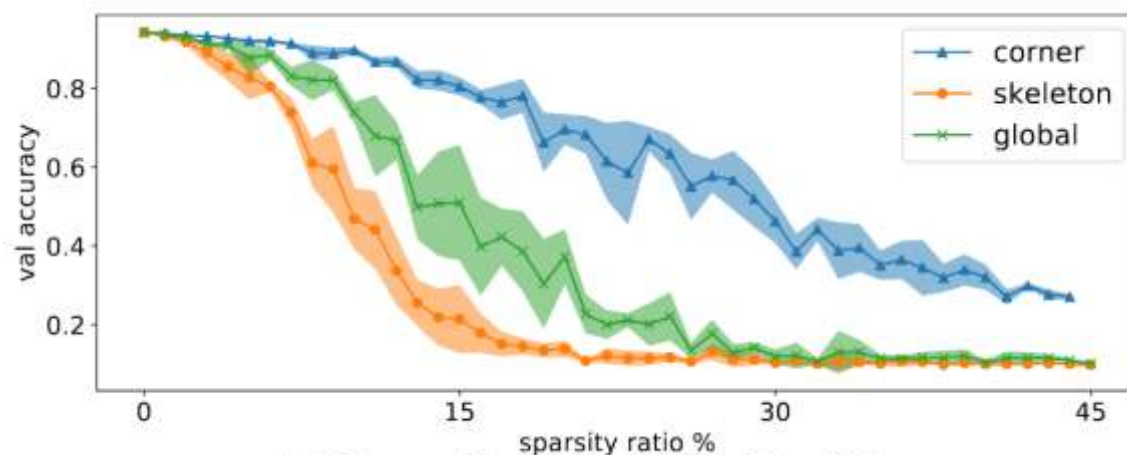
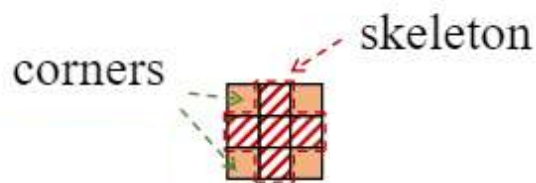


# ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks

Xiaohan Ding<sup>1</sup> Yuchen Guo<sup>2</sup> Guiguang Ding<sup>1</sup> Jungong Han<sup>3</sup>

<sup>1</sup> Beijing National Research Center for Information Science and Technology (BNRist);

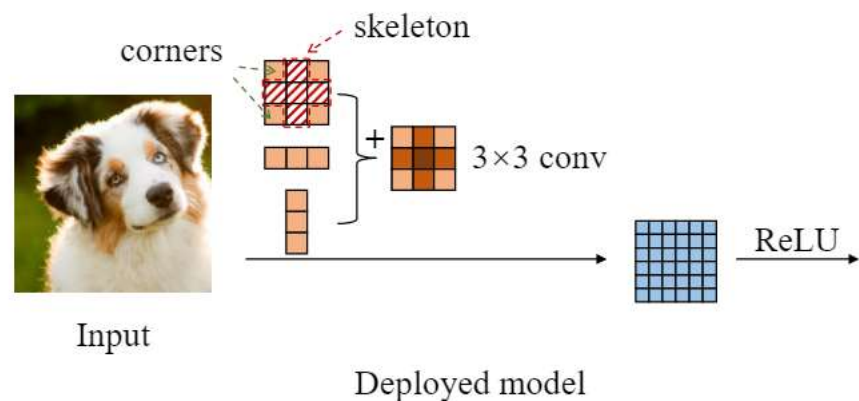
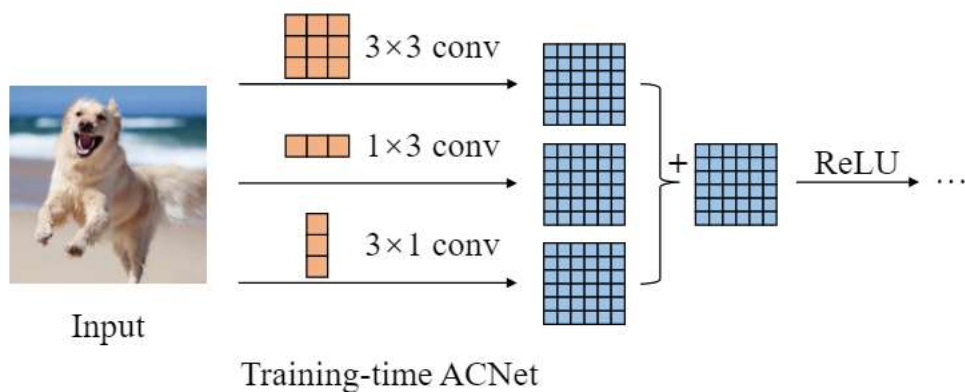
方形卷积核分布不均匀，在中心交叉位置（称为内核的“骨架”）的权重通常在幅度上更大



(a) Normally trained ResNet-56.

试图通过挖掘**正交**方面来加强标准卷积层：权重与其内核中的空间位置之间的关系

# Asymmetric Convolution Block (ACB)



0.826	0.860	0.833
0.850	0.935	0.867
0.821	0.854	0.827

(a) Normal.

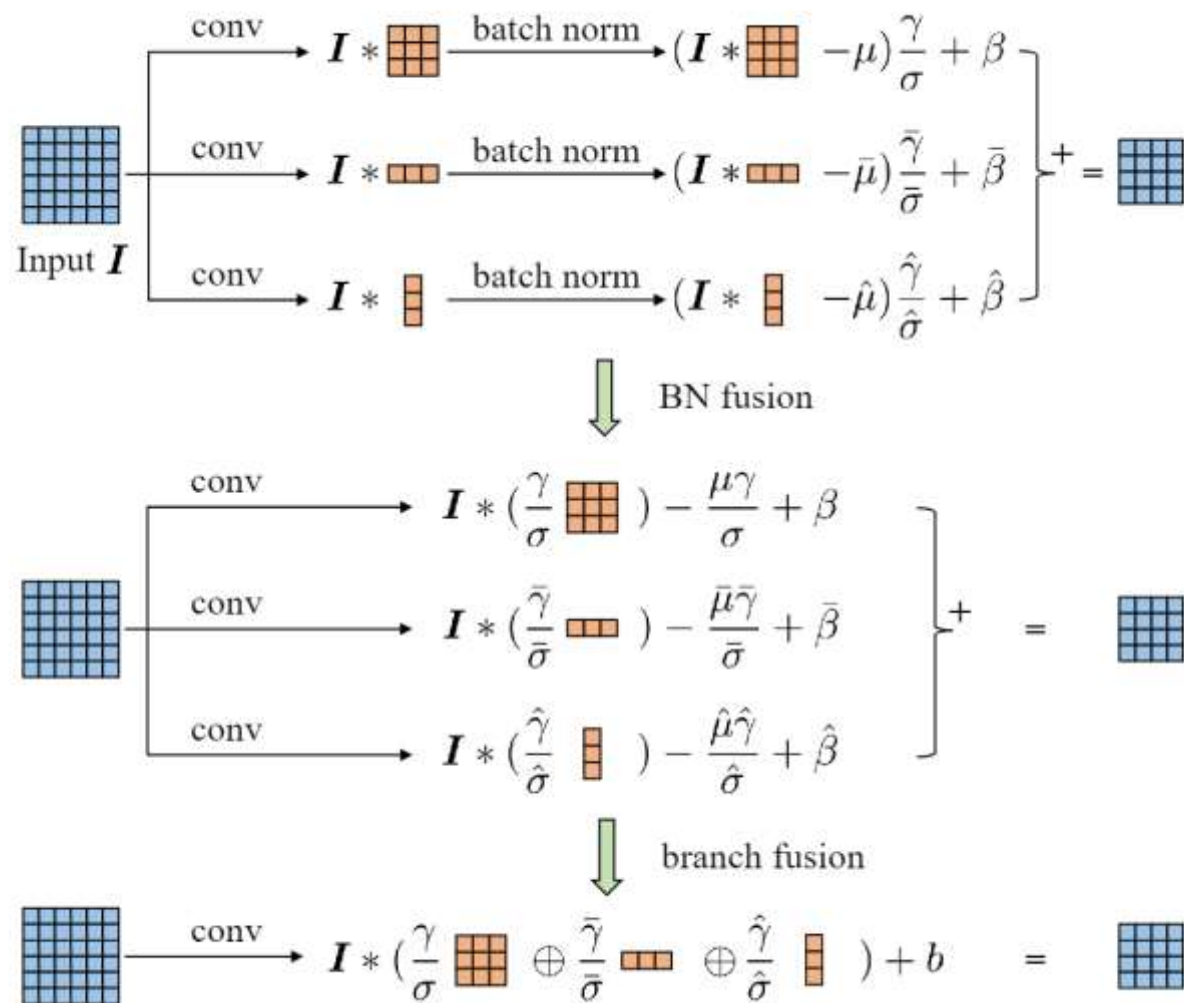
0.399	0.771	0.399
0.674	1.000	0.666
0.396	0.753	0.393

(b) ACNet, skeleton.

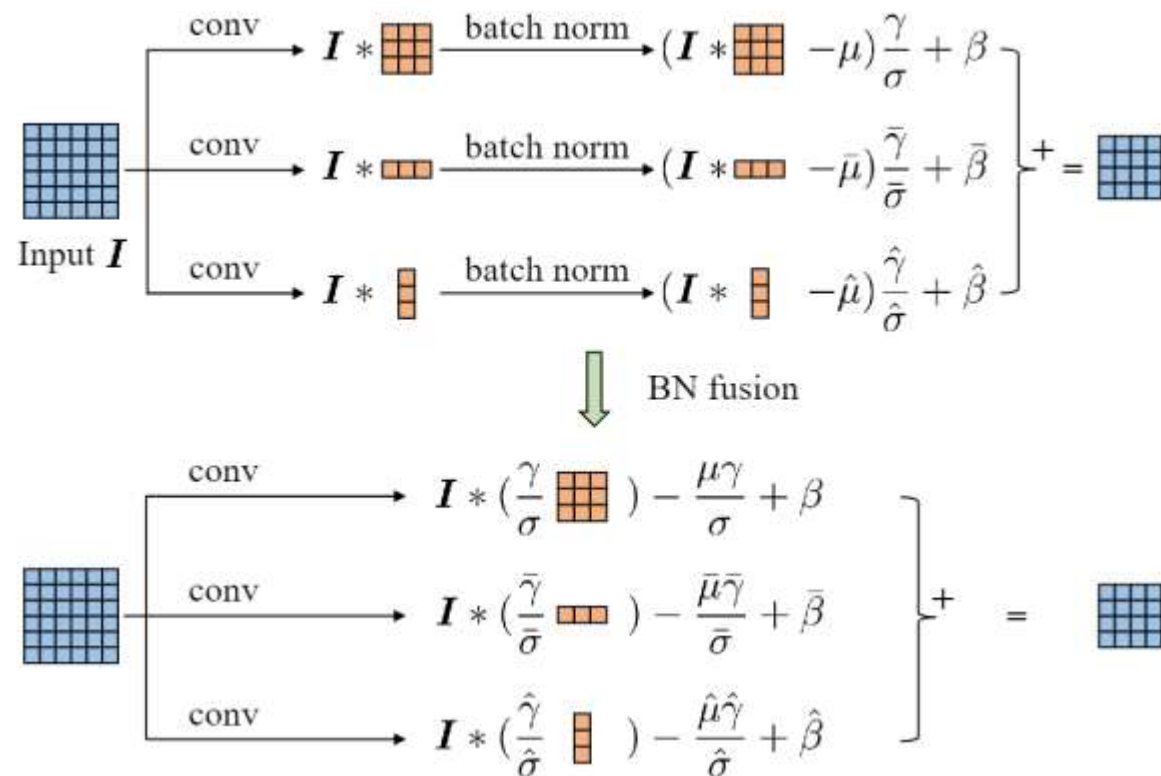
使用非对称卷积来显式地增强标准方核层的表征能力

提出了 ACB，包括三个并行层，分别为  $d \times d$ 、 $1 \times d$  和  $d \times 1$  核，其中输出相加以丰富特征空间

# Asymmetric Convolution Block (ACB)



# ACB for free inference-time improvements



将每个ACB转换为产生相同输出的标准卷积层

BN fusion:

$$O_{::,j} = \sum_{k=1}^C M_{::,k} * F_{::,k}^{(j)}$$

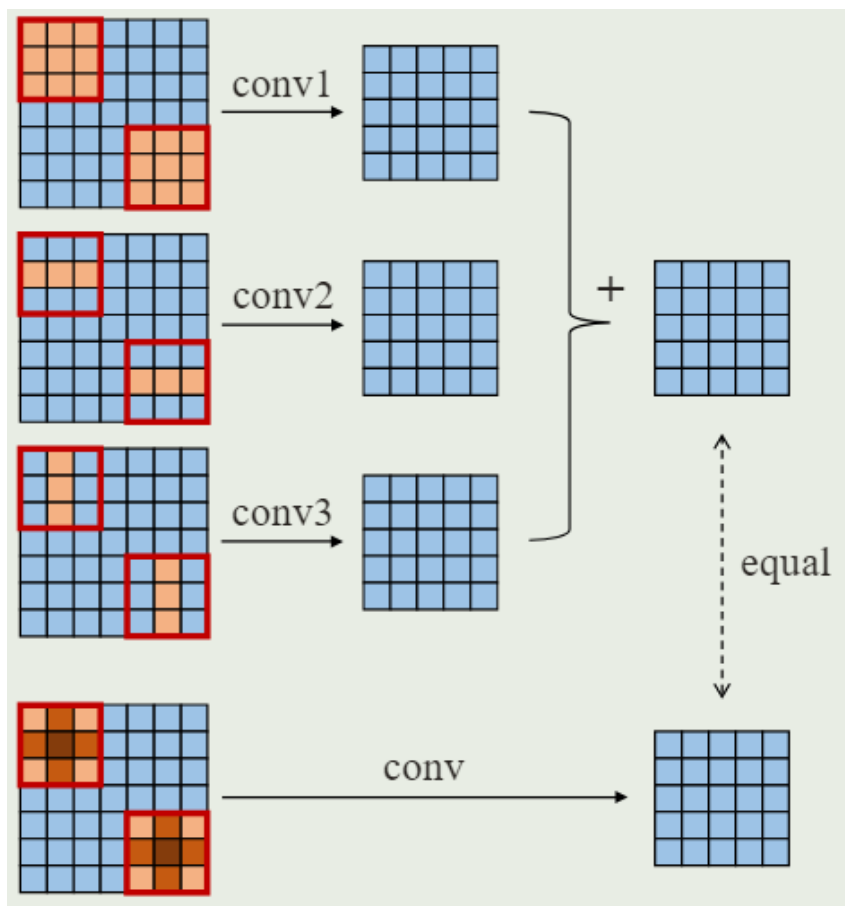
$$O_{::,j} = \left( \sum_{k=1}^C M_{::,k} * F_{::,k}^{(j)} - \mu_j \right) \frac{\gamma_j}{\sigma_j} + \beta_j$$

其中  $\mu$  和  $\sigma$  分别是批量归一化的通道均值和标准差的值,  $\gamma$  和  $\beta$  分别是学习的比例因子和偏置项

将一个新内核构造为  $\frac{\gamma_j}{\sigma_j} F^{(j)}$

以及添加的偏差项  $-\frac{\mu_j \gamma_j}{\sigma_j} + \beta_j$

# Exploiting the additivity of convolution



可加性:

$$I * K^{(1)} + I * K^{(2)} = I * (K^{(1)} \oplus K^{(2)})$$

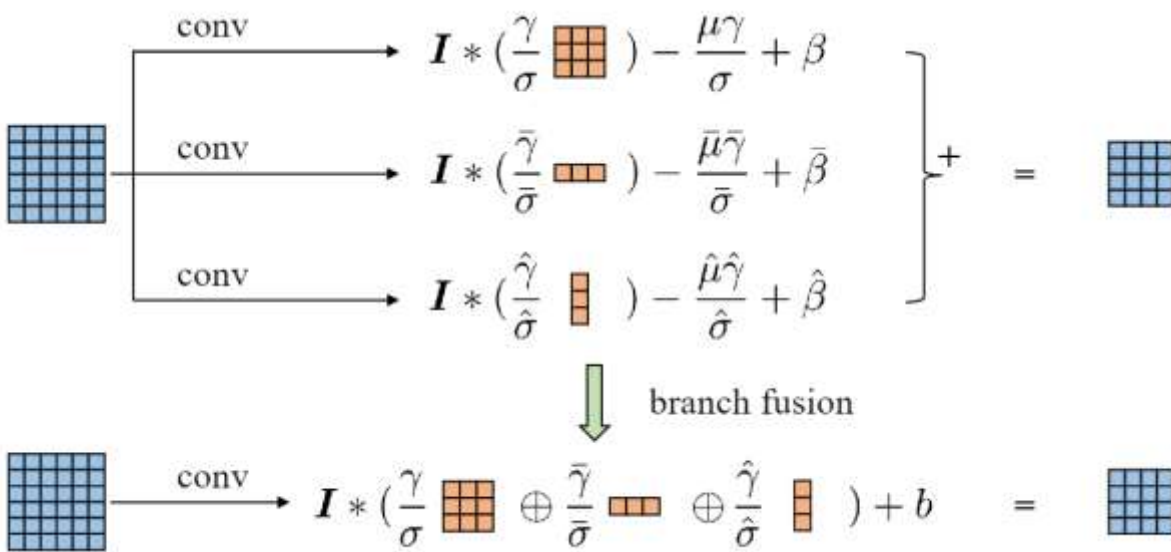
对于具有  $F^{(j)}$  的某个卷积核, 输出通道  $o$  上的某个点  $O_{:,j}$  由

$$y = \sum_{c=1}^C \sum_{h=1}^H \sum_{w=1}^W F_{h,w,c}^{(j)} X_{h,w,c}$$

$x$  是输入图上对应的滑动窗口

保持可加性的关键是共享相同的滑动窗口

# ACB for free inference-time improvements



Branch fusion:

$$F'^{(j)} = \frac{\gamma_j}{\sigma_j} F^{(j)} \oplus \frac{\bar{\gamma}_j}{\bar{\sigma}_j} \bar{F}^{(j)} \oplus \frac{\hat{\gamma}_j}{\hat{\sigma}_j} \hat{F}^{(j)},$$

$$b_j = -\frac{\mu_j \gamma_j}{\sigma_j} - \frac{\bar{\mu}_j \bar{\gamma}_j}{\bar{\sigma}_j} - \frac{\hat{\mu}_j \hat{\gamma}_j}{\hat{\sigma}_j} + \beta_j + \bar{\beta}_j + \hat{\beta}_j.$$

对任意滤波器 j :

$$O_{::,j} + \bar{O}_{::,j} + \hat{O}_{::,j} = \sum_{k=1}^c M_{::,k} * F'^{(j)}_{::,k} + b_j$$

其中  $O_{::,j}$ ,  $\bar{O}_{::,j}$  和  $\hat{O}_{::,j}$  分别是原始  $3 \times 3$ 、 $1 \times 3$  和  $3 \times 1$  分支的输出。



两个重要卷积核的性质：

**齐次性：**

$$I \circledast (pF) = p(I \circledast F), \forall p \in \mathbb{R}$$

**可加性：**

$$I \otimes F^{(1)} + I \otimes F^{(2)} = I \otimes (F^{(1)} + F^{(2)})$$

请注意，只有当两个卷积具有相同的配置（例如，通道数、内核大小、步幅、填充等）时，可加性才成立，以便它们共享相同的滑动窗口对应 X



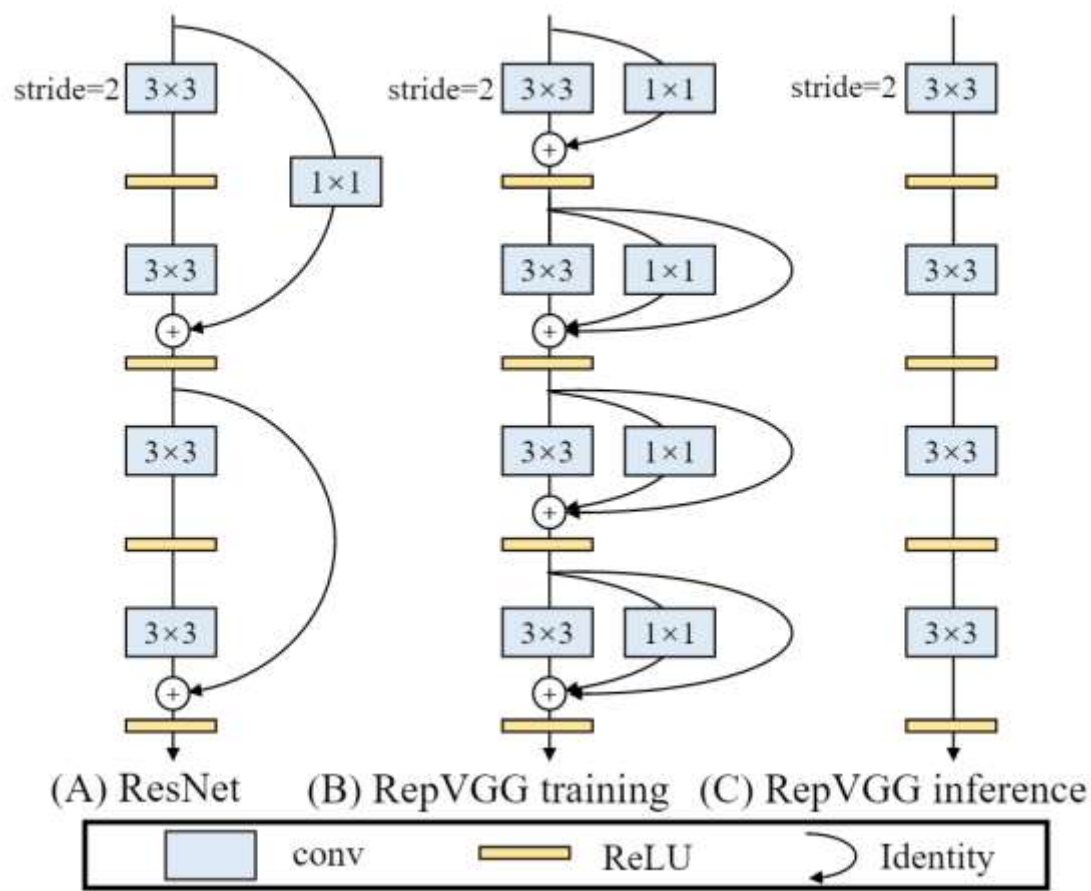
02

# RepVGG

Making VGG-style ConvNets Great Again



# Training-time Multi-branch Architecture



Identity branch	$1 \times 1$ branch	Accuracy	Inference speed w/o re-param
		72.39	1810
✓		74.79	1569
	✓	73.15	1230
✓	✓	<b>75.14</b>	1061

多分支在**推理**方面存在缺陷，但分支对**训练**有益

受 ResNet 启发

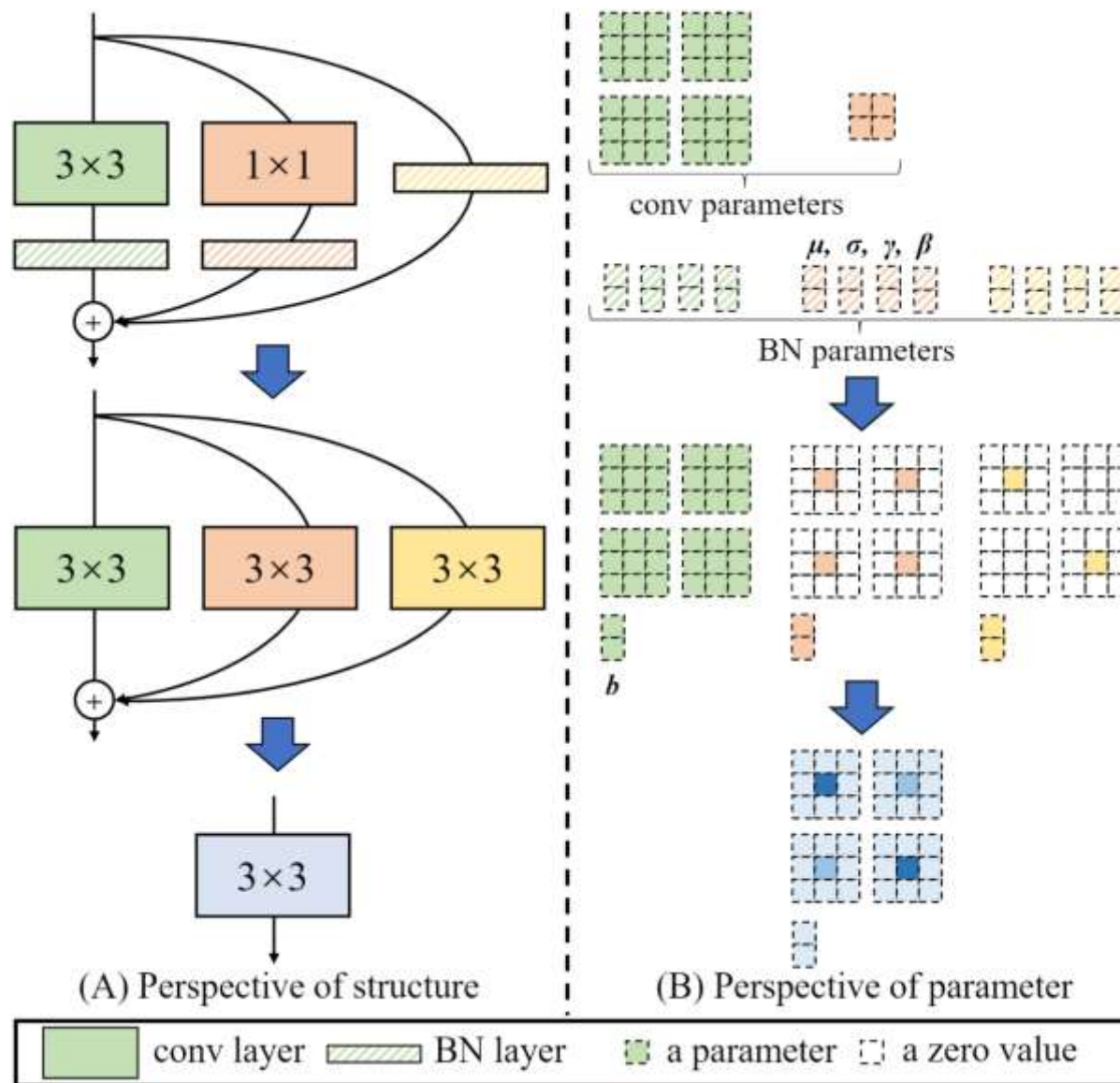
building block:

$$y = x + g(x) + f(x)$$

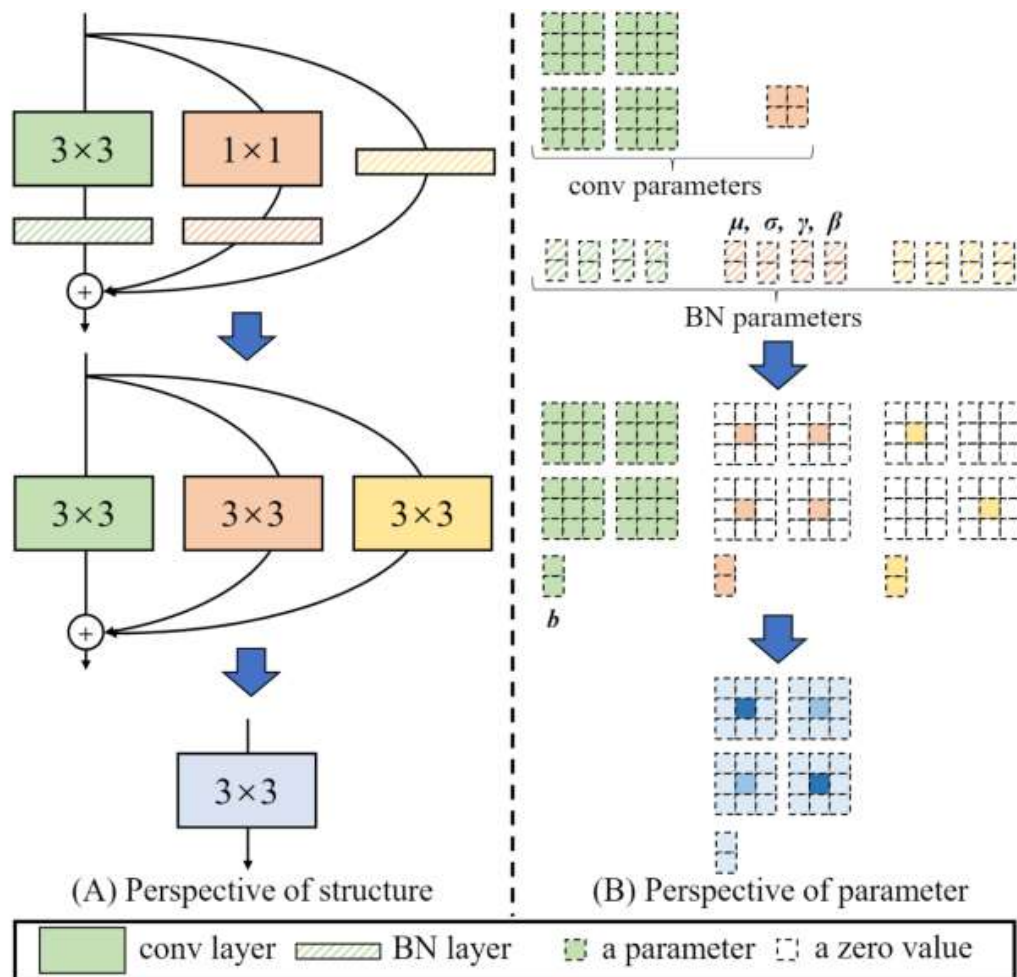
# Re-param for Plain Inference-time Model

in\_channels=2

out\_channels=2



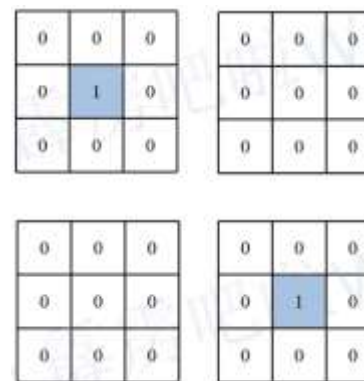
# Re-param for Plain Inference-time Model



合并卷积核:

the identity branch: 将 BN 层转换成  $3 \times 3$  的卷积核

将 BN 层看作内核为 1 的  $1 \times 1$  的卷积核并将其 padding 成  $3 \times 3$  的卷积核



02

**DBB**



## Diverse Branch Block: Building a Convolution as an Inception-like Unit

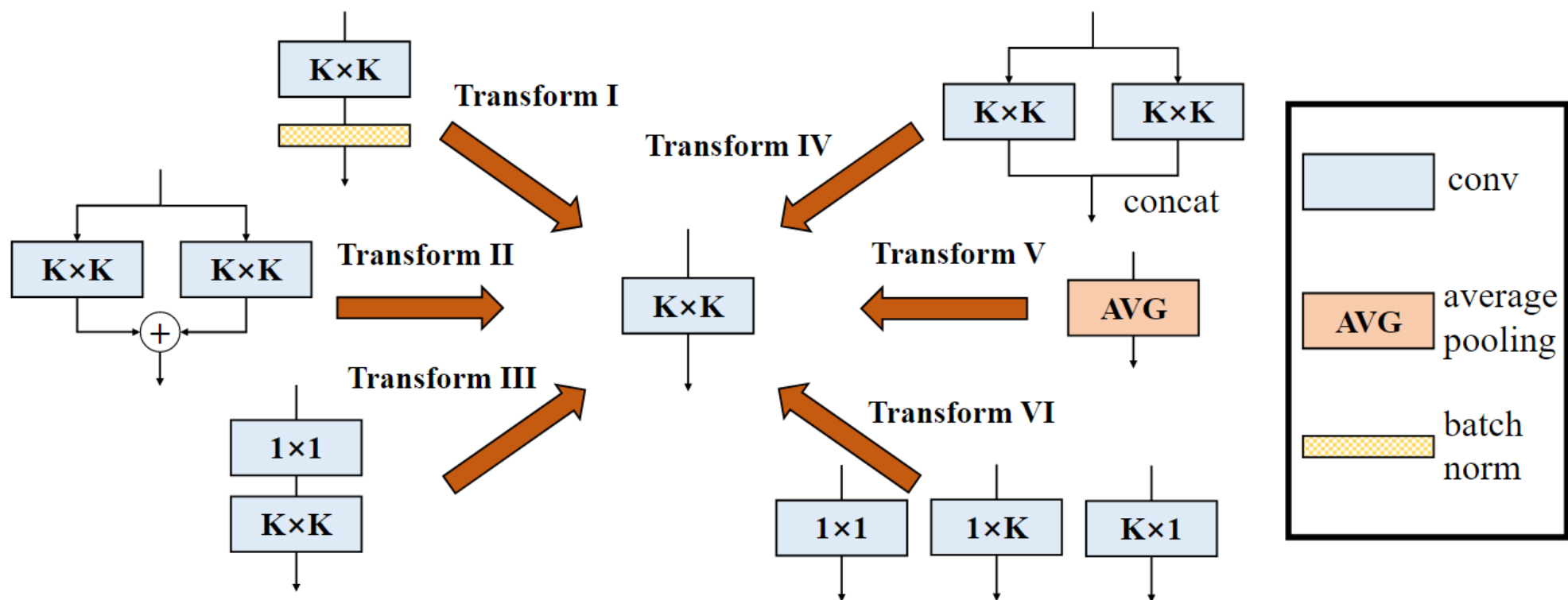
Xiaohan Ding <sup>1\*</sup> Xiangyu Zhang <sup>2</sup> Jungong Han <sup>3</sup> Guiguang Ding <sup>1†</sup>

<sup>1</sup> Beijing National Research Center for Information Science and Technology (BNRist);

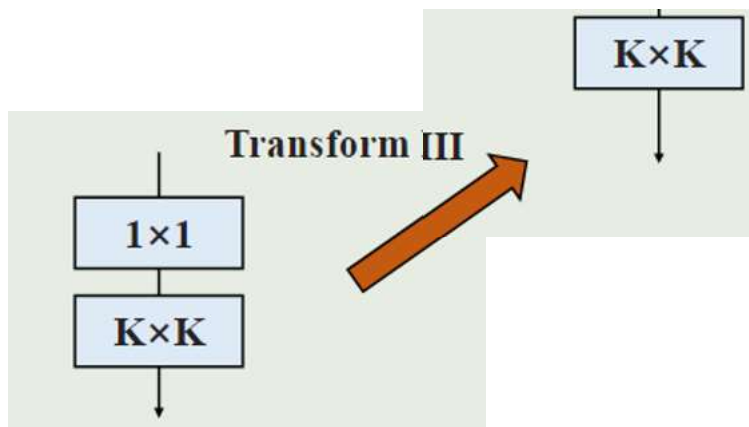
**多样化分支块(DBB): 一个卷积神经网络 (ConvNet) 的通用构建块, 以在没有任何推理时间成本的情况下提高性能。**

- 通过组合不同尺度和复杂性的不同分支来丰富特征空间来增强单个卷积的表示能力
- 训练后, DBB 可以**等效地转换为单个卷积层进行部署**

# A Convolution for Diverse Branches



# Transform III: a conv for sequential convolutions



以将  $1 \times 1$  conv - BN -  $K \times K$  conv - BN 序列合并为一个  $K \times K$  conv

$$O' = (I \otimes F^{(1)} + \text{REP}(b^{(1)})) \otimes F^{(2)} + \text{REP}(b^{(2)})$$

将 conv 的**可加性**应用于上式:

$$O' = I \otimes F^{(1)} \otimes F^{(2)} + \text{REP}(b^{(1)}) \otimes F^{(2)} + \text{REP}(b^{(2)})$$

希望最终得到的是一个卷积:

$$O' = I \circledast F' + \text{REP}(b')$$

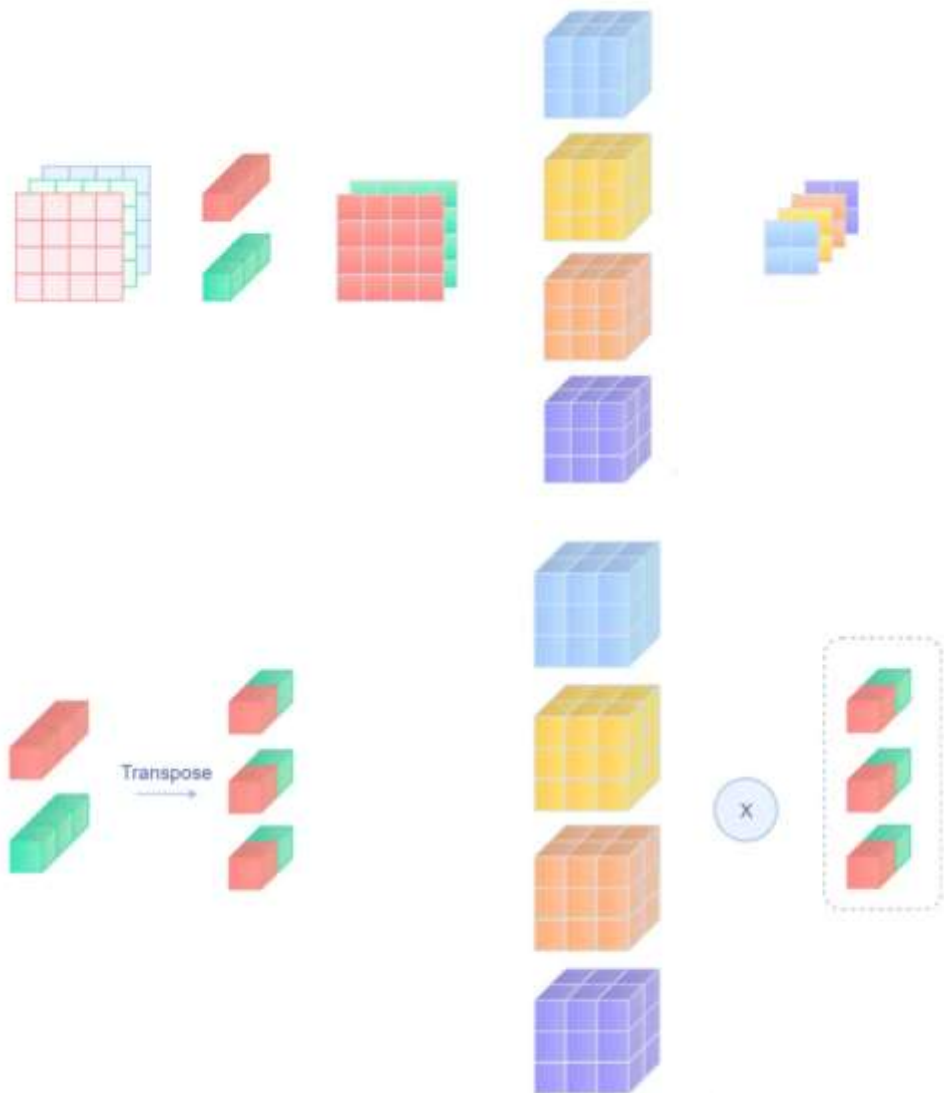
$1 \times 1$  卷积, 只进行通道间的线性组合操作而没有空间上的相加操作, 因此可以通过线性重组  $K \times K$  卷积核中的参数将其合并到  $K \times K$  卷积核中

$$F' \leftarrow F^{(2)} \circledast \text{TRANS}(F^{(1)})$$

TRANS: 是  $F^{(1)}$  前两个维度的转置



# Transform III: a conv for sequential convolutions



假设输入是一个三通道的图片，1x1卷积的输出通道为2，3x3卷积的输出通道为4

$$\begin{matrix} 2 & 3 \\ F_1(D, C, 1, 1) \\ F_2(E, D, K, K) \\ 4 & 2 \end{matrix}$$

1x1卷积核的第零维和第一维互相调换位置:

$$\text{Transpose}(F_1) : F_1(D, C, 1, 1) \rightarrow F_1(C, D, 1, 1)$$

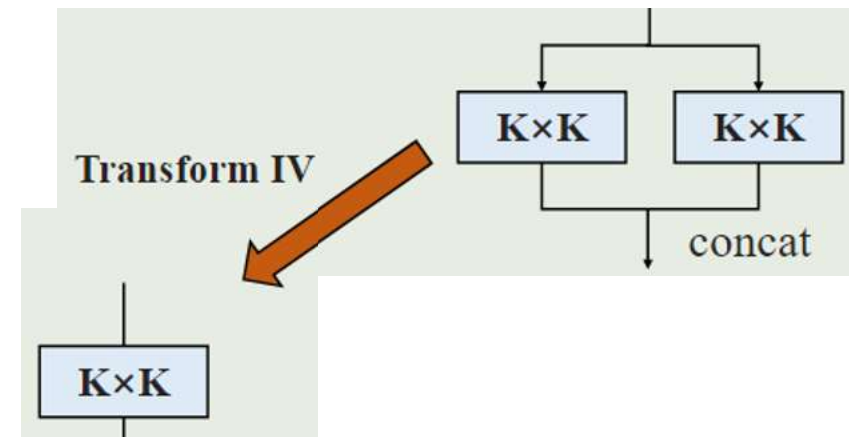
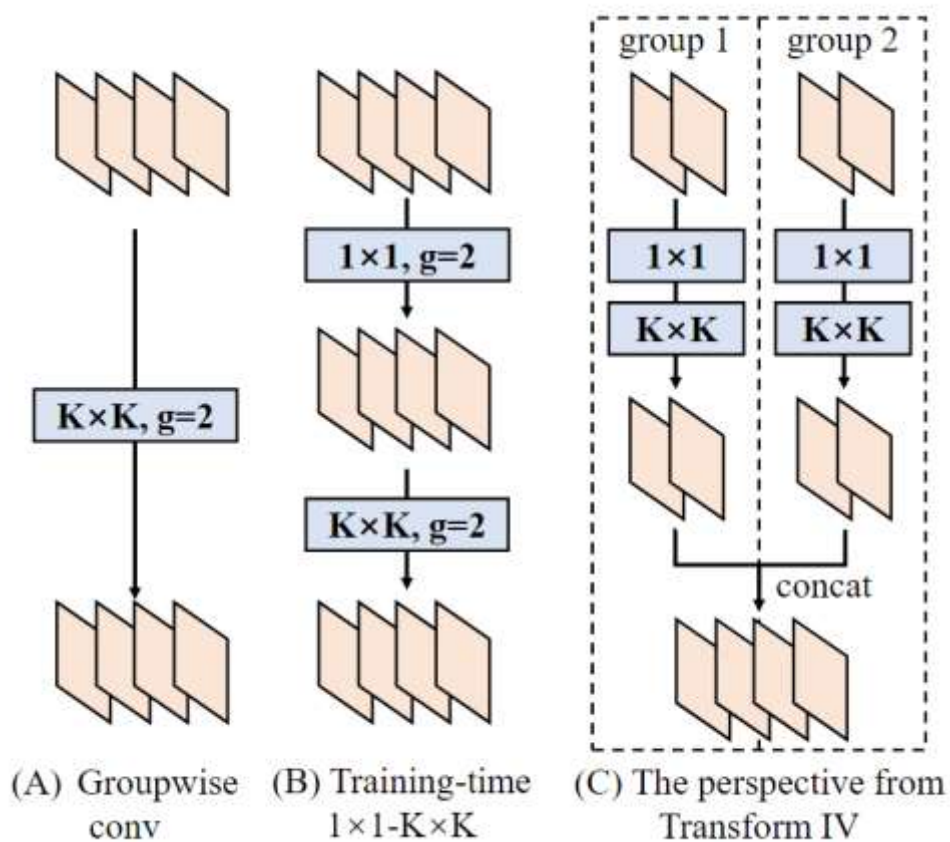
然后3x3卷积核权重与转置后的"1x1卷积核"进行卷积操作

$$F_2 \otimes \text{Transpose}(F_1) \text{形状为} (E, C, K, K)$$

最后输入与其做卷积操作，整个流程可以写为

$$\text{Input} \otimes F_2 \otimes (\text{Transpose}(F_1))$$

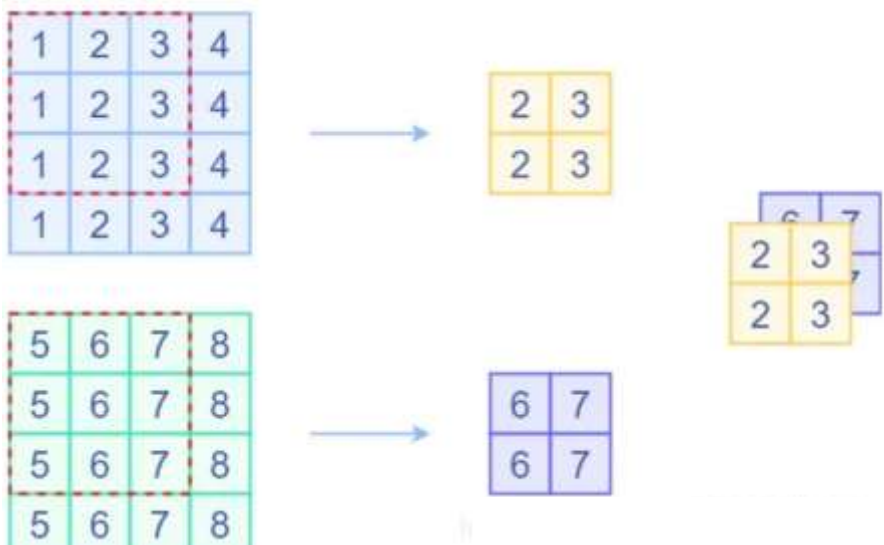
# Transform IV: a conv for depth concatenation



对于  $F^{(1)} \in \mathbb{R}^{D_1 \times C \times K \times K}, b^{(1)} \in \mathbb{R}^{D_1}, F^{(2)} \in \mathbb{R}^{D_2 \times C \times K \times K}, b^{(2)} \in \mathbb{R}^{D_2}$   
 沿输出通道维度拼接得到:  $F' \in \mathbb{R}^{(D_1+D_2) \times C \times K \times K}, b' \in \mathbb{R}^{D_1+D_2}$

$$\text{CONCAT}(I \otimes F^{(1)} + \text{REP}(b^{(1)}), I \otimes F^{(2)} + \text{REP}(b^{(2)})) \\ = I \otimes F' + \text{REP}(b').$$

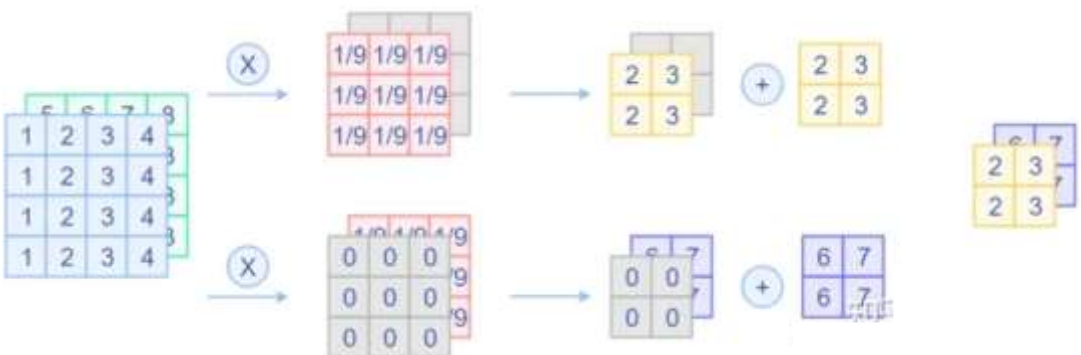
# ● Transform V: a conv for average pooling



一个  $C$  个通道的内核大小  $K$  和步幅  $s$  的平均池化层等价于具有相同  $K$  和  $s$  的 conv

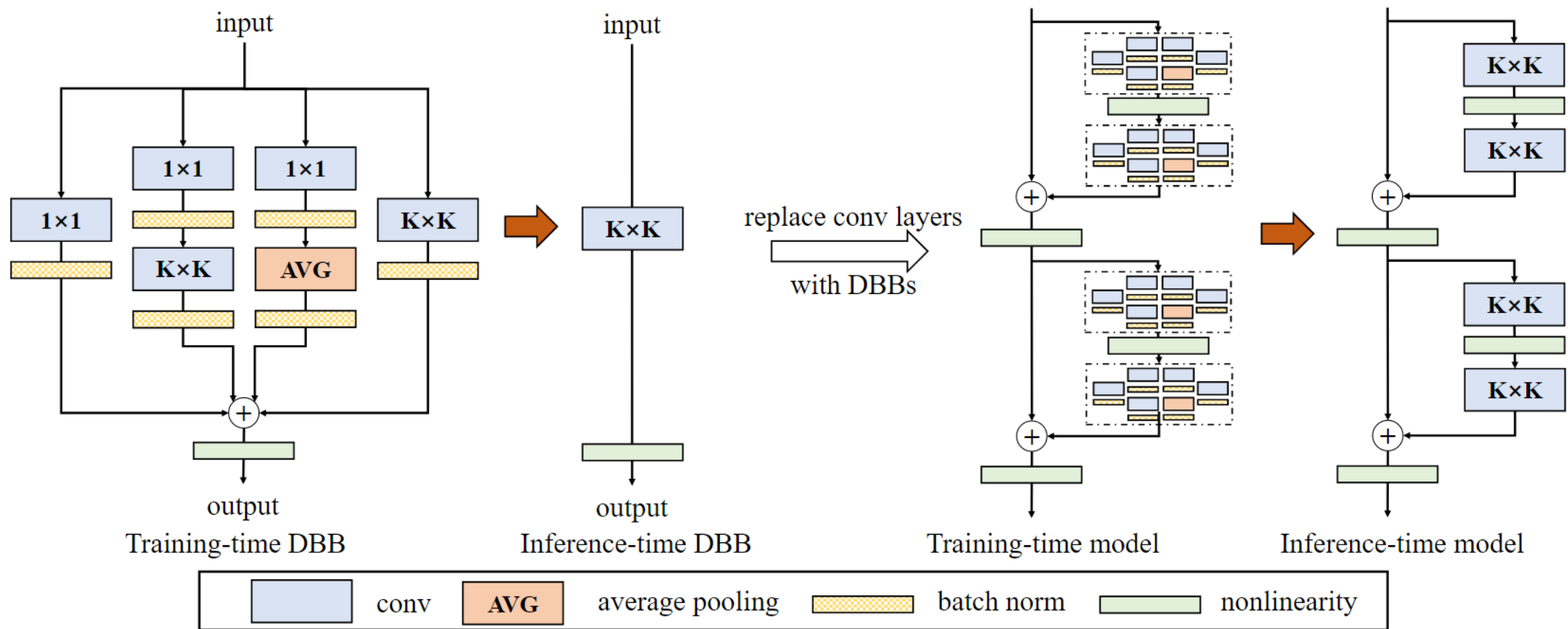
$$F' \in \mathbb{R}^{C \times C \times K \times K}$$

池化层是针对各个输入通道的(如 Depthwise 卷积), 而卷积层会将所有输入通道的结果相加`



$$F'_{d,c,::} = \begin{cases} \frac{1}{K^2} & \text{if } d = c, \\ 0 & \text{elsewise.} \end{cases}$$

# An Inception-like DBB Instance

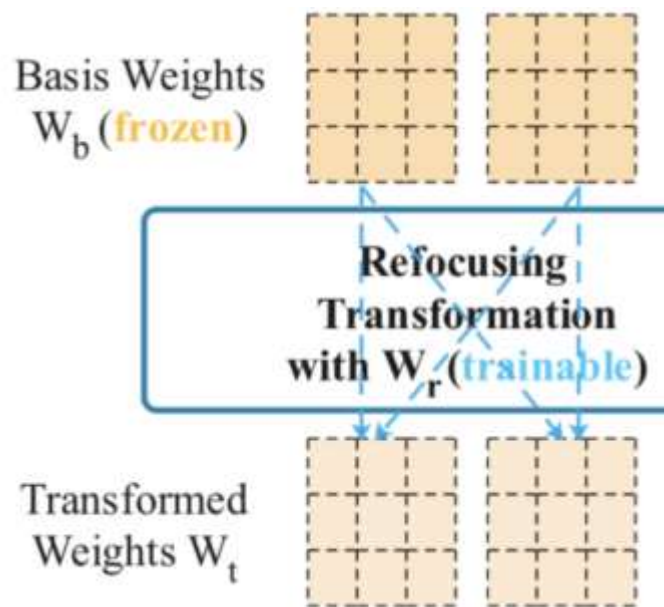


03

**RefConv**



# RefConv

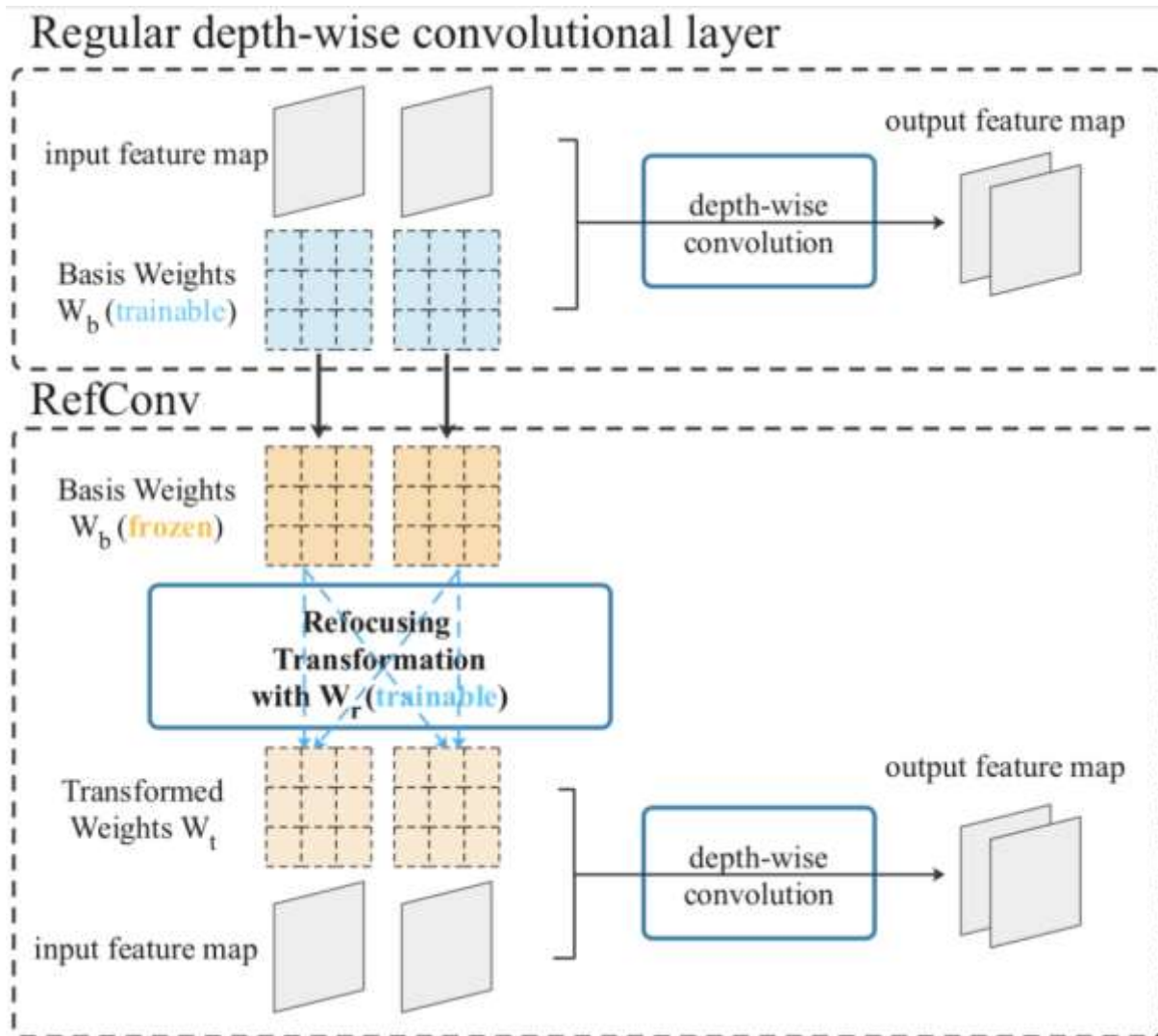


RefConv: 一种基于重参数化操作的重聚焦卷积方法

提出一种名为**重参数化重聚焦卷积 (Re-parameterized Refocusing Convolution)**的技术，**建立了现有结构参数之间的联系**

**动机：在不改变模型定义或不引入任何推理成本的情况下为模型添加更多的先验**

# DEPTH-WISE REFCONV



预训练好的 DW Conv 的基权重  $W_b$  的变换

$$W_t = T(W_b, W_r)$$

$W_t$ : 变换权重  $W_t$   
 $W_r$ : 变换函数  $T()$  的参数

在本文中, 使用密集卷积作为  $T()$ , 以便重新聚焦变换

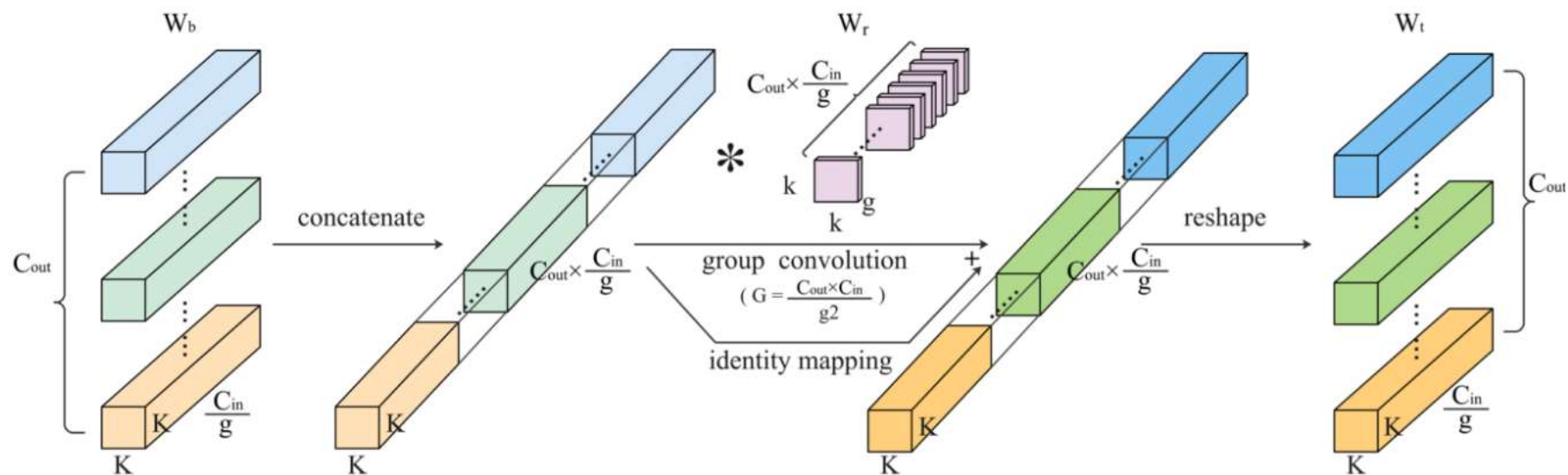
$$W_t = W_b * W_r + W_b$$

受残差学习的启发, 重聚焦变换在基权值上学习增量, 而不是原始映射



# GENERAL REFCONV

$$\mathbf{W}_t = \mathbf{W}_b * \mathbf{W}_r + \mathbf{W}_b$$



特殊情况:

1. DW Conv: 此时  $g = C_{out} = C_{in}$ , 则  $G = \frac{C_{out}C_{in}}{g^2} = 1$ , 变换权重  $\mathbf{W}_r \in \mathbb{R}^{C_{out} \times C_{in} \times K \times K}$   
(此时的重参化的重聚焦卷积像是 Dense Conv 进行建立了现有结构参数之间的联系)
2. Dense Conv: 此时  $g=1$ , 则  $G = C_{out}C_{in}$ , 变换权重  $\mathbf{W}_r \in \mathbb{R}^{(C_{out}C_{in}) \times 1 \times k \times k}$   
(此时的重参化的重聚焦卷积像是 DW Conv, 不执行跨通道重组)



# RefConv

优:

- 不会改变模型结构
- 高效且节省内存
- 与数据无关

Index	Top-1 Accuracy		Params (M)		FLOPs (G)		Memory (G)	
Model	Base.	RefC.	Base.	RefC.	Base.	RefC.	Base.	RefC.
MobileNetv1	72.18%	72.96% (+0.82%)	3.22	28.29	150.76	150.96	19.83	20.21
MobileNetv2	71.68%	72.35% (+0.67%)	3.56	44.11	90.37	90.72	24.21	24.98
MobileNetv3-S	61.95%	63.42% (+1.47%)	2.94	11.15	17.12	17.20	14.49	14.68
MobileNetv3-L	71.73%	72.91% (+1.18%)	5.48	34.06	61.15	61.41	24.33	24.85
MobileNeXt	71.57%	72.81% (+1.24%)	3.31	109.35	79.37	80.92	30.29	32.21
HBONet	71.61%	72.59% (+0.98%)	4.56	44.49	83.71	84.10	25.26	25.66
EfficientNet-B0	75.78%	76.74% (+0.96%)	4.98	72.67	103.53	104.20	31.02	31.78
ShuffleNetv1	63.17%	64.30% (+1.13%)	1.81	4.56	35.52	35.55	14.61	14.82
ShuffleNetv2	67.66%	68.92% (+1.26%)	2.28	5.94	39.65	39.69	13.17	13.30
ResNet-18	70.69%	71.63% (+0.94%)	11.72	22.74	472.08	472.20	15.52	15.76
ResNet-50	76.16%	76.96% (+0.80%)	25.61	36.97	1063.35	1063.55	32.14	32.54
ResNet-101	77.14%	77.72% (+0.68%)	44.63	66.01	2018.72	2018.96	42.97	43.92
DenseNet-169	76.17%	76.90% (+0.73%)	14.18	17.24	884.81	884.95	49.10	49.96
FasterNet-S	78.76%	79.91% (+1.15%)	28.44	34.65	1091.28	1091.85	24.24	24.62
ConvNeXt-T	80.82%	81.68% (+0.96%)	28.59	57.37	1139.10	1139.97	38.53	38.62

Model	ResNet-18			MobileNetv2		
Method	Top1-Accuracy	FLOPs (G)	Memory (G)	Top1-Accuracy	FLOPs (G)	Memory (G)
Baseline	70.69% (+0.00%)	472.08	15.52	71.68% (+0.00%)	90.37	24.21
ACB	71.47% (+0.78%)	761.25	18.76	71.99% (+0.31%)	98.62	32.02
RepVGGB	71.21% (+0.52%)	522.85	17.02	72.11% (+0.33%)	94.39	28.38
DBB	71.25% (+0.56%)	1097.40	26.18	72.25% (+0.48%)	125.61	43.52
WN	70.81% (+0.12%)	472.10	15.53	71.76% (+0.08%)	90.37	24.21
CWN	70.85% (+0.16%)	472.10	15.53	71.78% (+0.10%)	90.37	24.21
OWN	70.83% (+0.14%)	472.10	15.53	71.75% (+0.07%)	90.37	24.21
<b>RefConv</b>	<b>71.63% (+0.94%)</b>	472.20	15.76	<b>72.35% (+0.67%)</b>	90.72	24.98

# RefConv

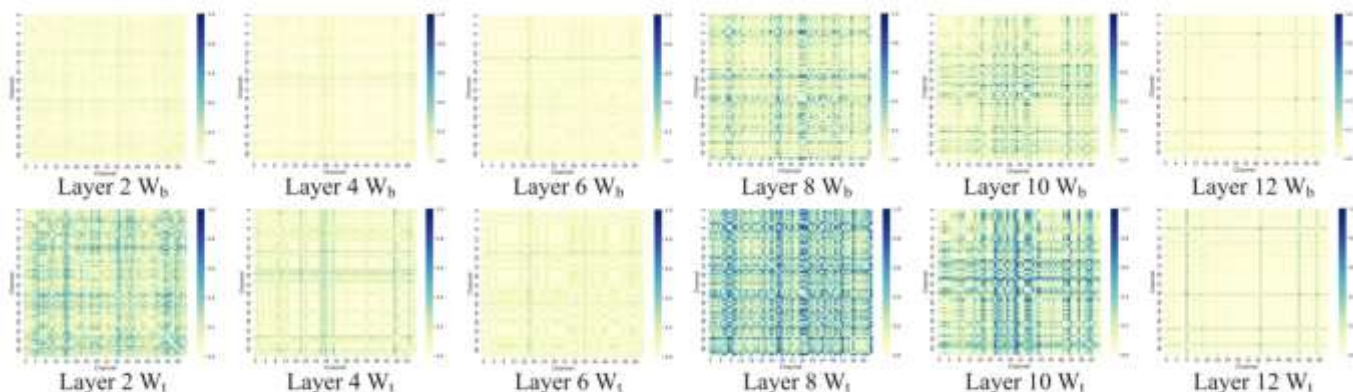


Figure 4: The similarity matrix of the first 64 channels of  $W_b$  and  $W_t$  in different layers. The backbone model is MobileNetV1 trained on ImageNet. To improve the readability, the original value of KL divergence is added with 1 and then taken 10-base logarithm. A point with a darker color represents a larger value, hence a lower similarity.

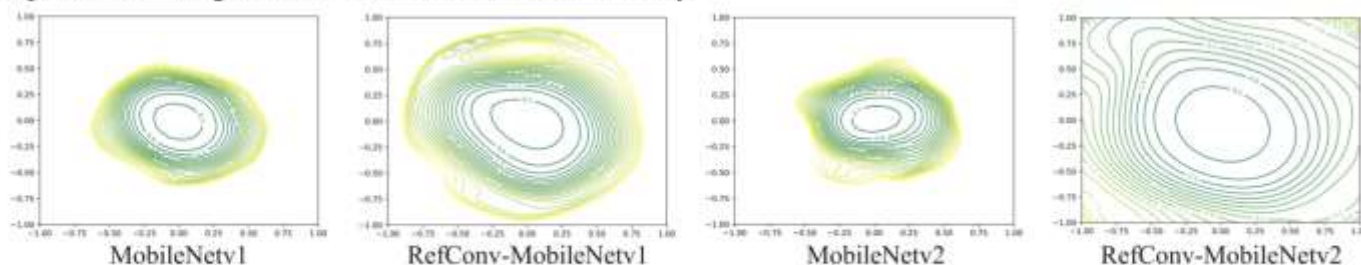


Figure 5: Visualization of the loss landscapes.

利用 Kullback-Leibler (KL)散度来衡量不同通道对之间的相似性

RefConv减少了信道冗余，增强了表示多样性

RefConv 的损失景观具有更宽、更稀疏的轮廓，这表明 RefConv 的损失曲率更平坦，具有更好的泛化能力



THANKS