



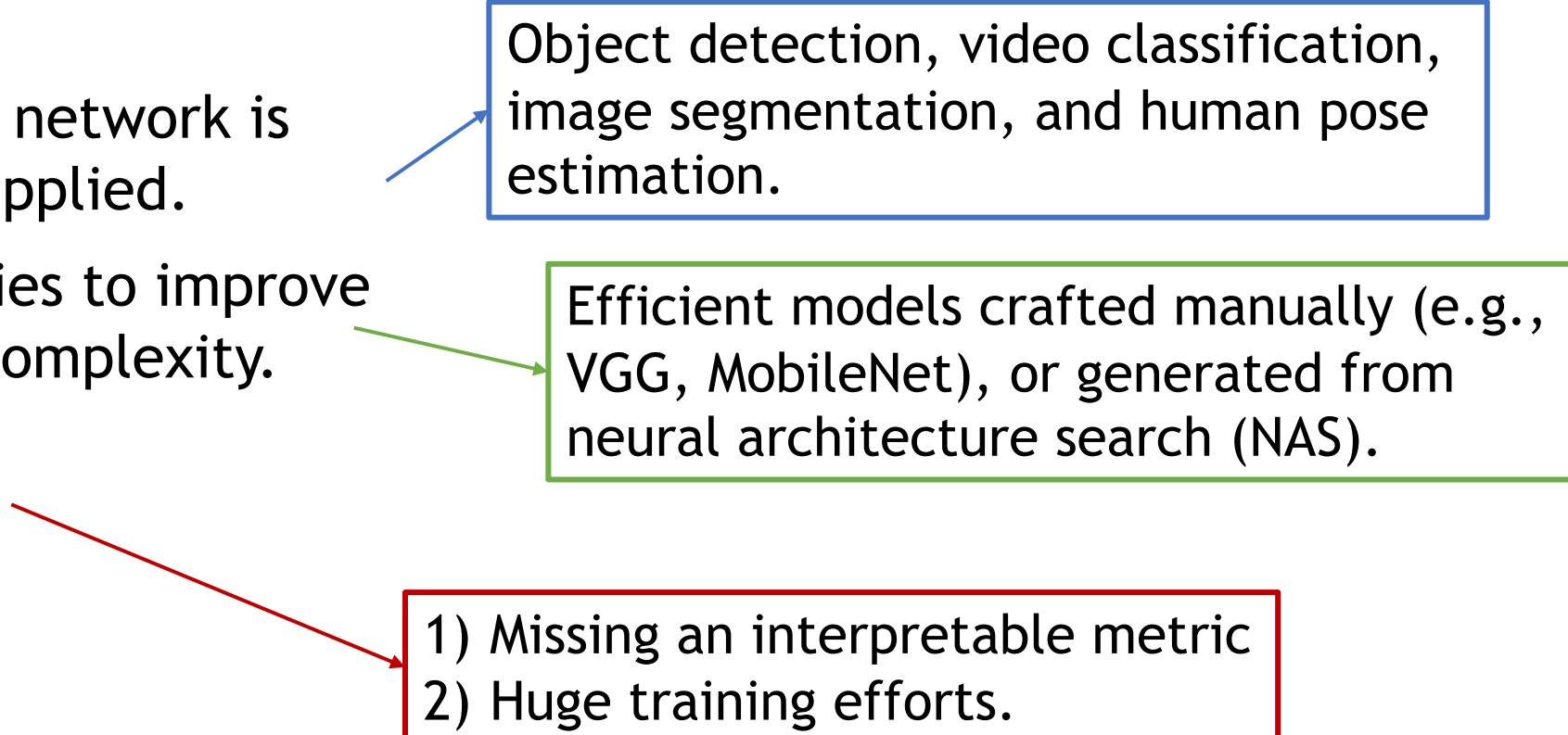
An Efficient Quantitative Approach for Optimizing Convolutional Neural Networks

Yuke Wang, Boyuan Feng, *Xueqiao Peng, Yufei Ding

*The Ohio State University
UC at Santa Barbara

Background

- Convolutional neural network is popular and widely applied.
- Existing CNN work tries to improve accuracy or reduce complexity.
- Two major issues.



Object detection, video classification, image segmentation, and human pose estimation.

A blue arrow points from this box to the first bullet point. A green arrow points from the second bullet point to the box below. A red arrow points from the third bullet point to the box below.

Efficient models crafted manually (e.g., VGG, MobileNet), or generated from neural architecture search (NAS).

- 1) Missing an interpretable metric
- 2) Huge training efforts.

Existing Convolutions

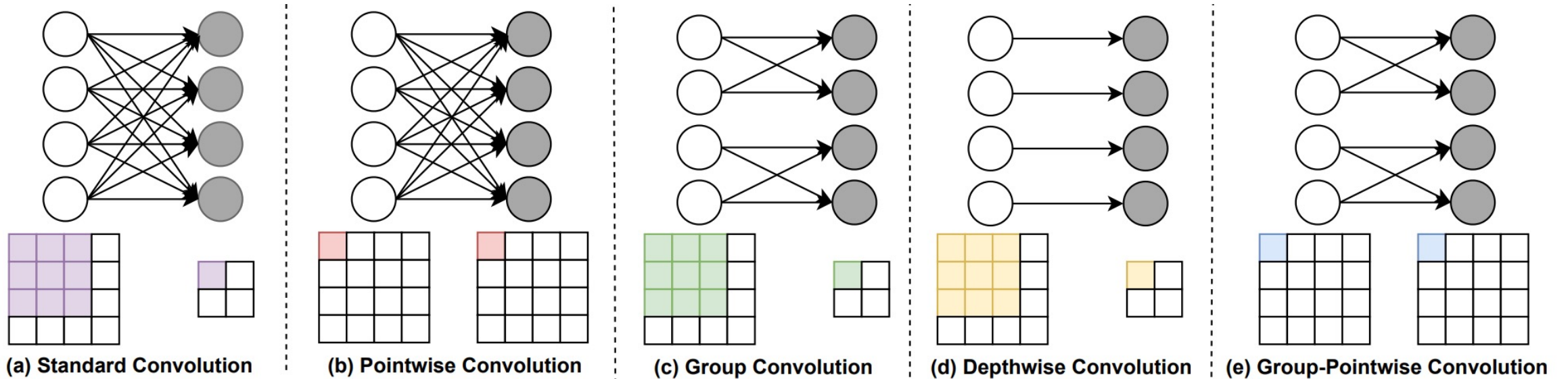





Figure 1: Channel mapping (top) and Spatial mapping (bottom) of the standard convolution and factorized convolution kernel.

Receptive Field

- Quantifies the local representation ability in a single traditional convolution layer.  A larger receptive field leads to higher accuracy.
- Fails to quantify the global representation ability across layers.  Modern CNNs have a large number of convolution layers with diverse receptive fields stacked in a CNN stage.
- Fails to consider the channel number.  Channel information is critical in modern convolution layers (e.g., Depthwise convolution and Channel-wise convolution).

Contributions

- 3D-Receptive Field (3DRF), an interpretable metric.
- CNN model stage-level design.
- CNN model kernel-level design.



Decide the number of convolution kernels at different stages.

Decide the type of the convolution kernel to use (standard convolution kernels or efficient factorized kernels).

3D-RECEPTIVE FIELD

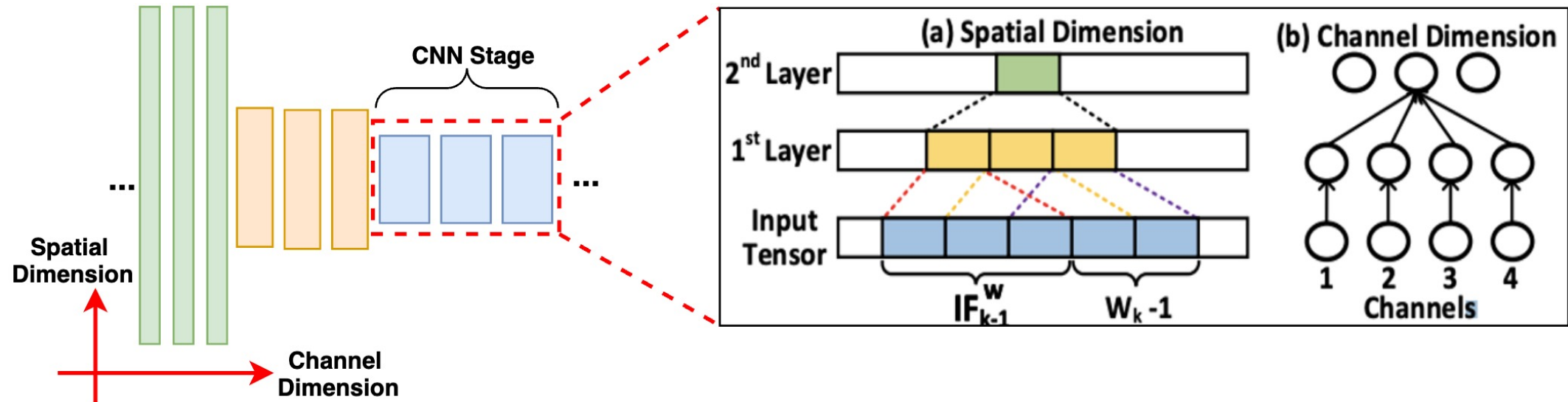
- 3D-Receptive Field (3DRF).

$$3DRF_k = (3DRF_k^w)^d * 3DRF_k^c$$

$$3DRF_k^w = \min(3DRF_{k-1}^w + w_k - 1, w_0)$$

$$3DRF_k^c = \min(g(3DRF_{k-1}^c, T_k), c_0)$$

Measuring the representation ability of each neuron in a convolution layer

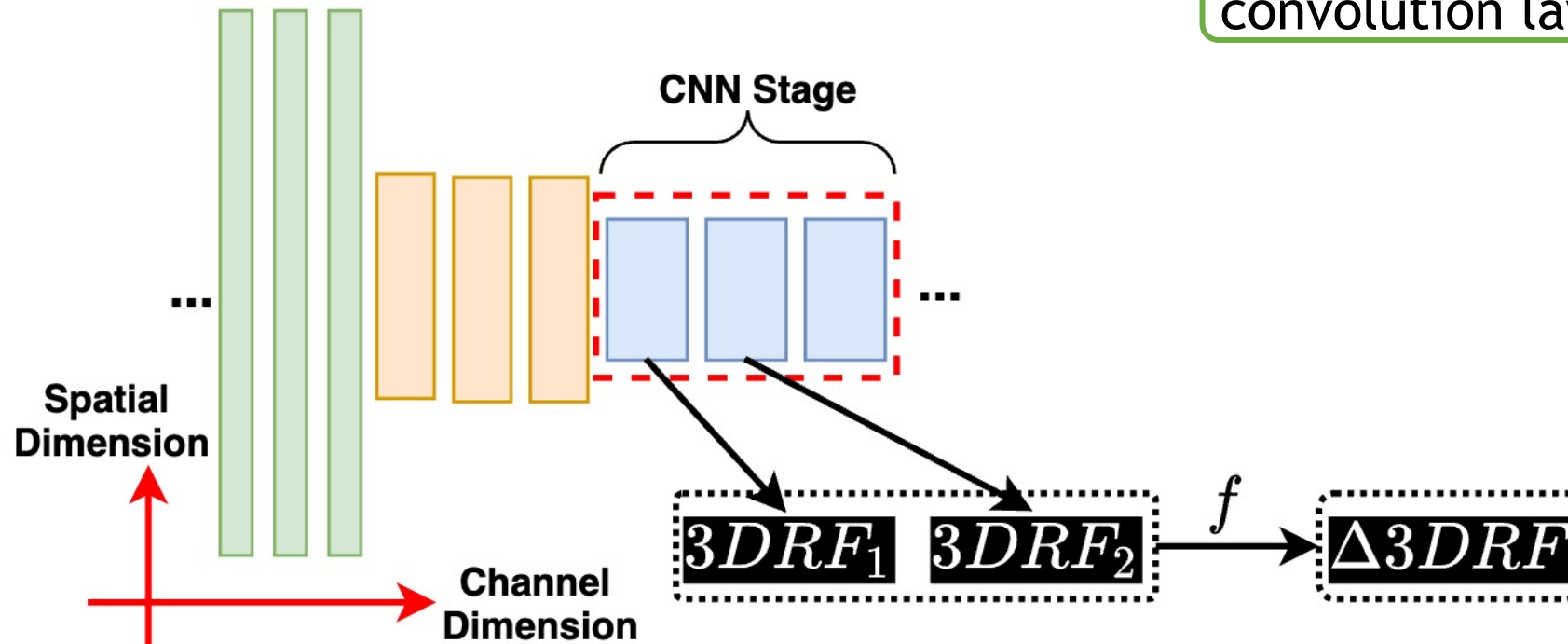


3D-RECEPTIVE FIELD (Cont'd)

- 3DRF Gain.

$$\Delta 3DRF_k = \frac{3DRF_k - 3DRF_{k-1}}{3DRF_{k-1}} * e^{-\alpha * \frac{3DRF_{k-1}}{V_0}}$$

Quantifying the representation ability change between two consecutive convolution layers.



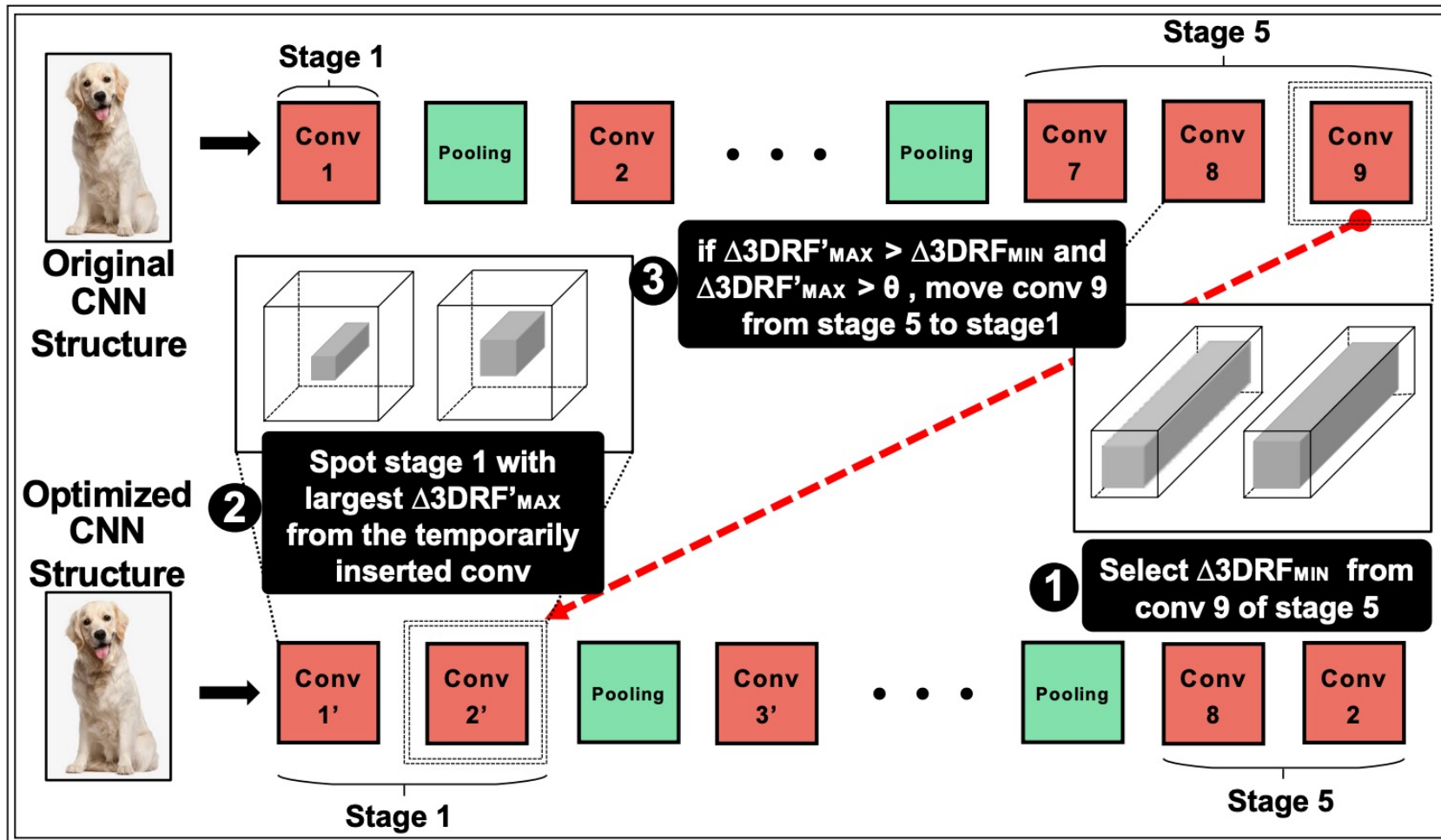
Case Study: Accuracy Impact of 3DRF Gain

- VGG11 as the baseline structure and run it on CIFAR-10 dataset.
- Five VGG-variants by inserting a single standard convolution before each max pooling.

Table 2: Impact of 3DRF Gain ($\Delta 3DRF$) over Accuracy.

Network	$\Delta 3DRF$	Accuracy (%)	Δ Accuracy (%)
VGG-11	0	92.68	0
Variant-1	1.73	93.56	0.88
Variant-2	1.60	93.46	0.78
Variant-3	0.29	92.75	0.07
Variant-4	0.0	92.58	-0.10
Variant-5	0.0	92.41	-0.27

Stage-level Organizer



Kernel-Level Decomposer

- Reduces the computational cost of a CNN architecture design.

- Rule of Kernel Replacement

- 1) Quality Condition: $3DRF(N) = 3DRF(S)$ for the same input tensor;
- 2) Compact Condition: $3DRF(N - x) < 3DRF(S)$ if we remove a factorized kernel x from N

- Unify the previous construction of the convolution block and build a new convolution blocks and one efficient factorized kernel.

Substituting its standard convolution kernels with less computational expensive convolution blocks.

Ensures the effectiveness of N with regards to its learning capacity,

Guarantees its optimality in terms of computation efficiency.

Kernel-Level Decomposer (cont'd)

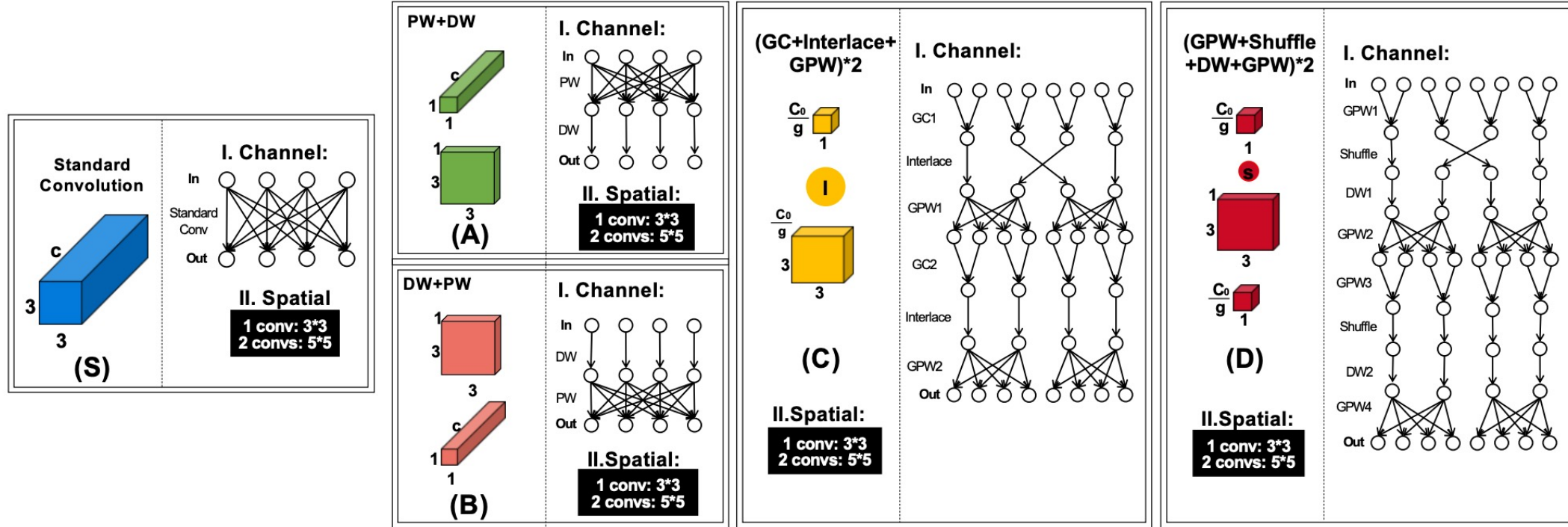


Figure 4: Illustration of the 3DRF, both in the channel (I) and spatial (II) dimension, for the standard kernels (S) and previous convolution blocks (A-D). g is the number of groups for GC and GPW. The arrow denotes the flow from inputs to outputs in the channel dimension, and the number of input channels that could flow into an output neuron would be the channel dimension of 3DRF for that block. We omit the process of computing the spatial size of 3DRF, while only giving the computed result based on Equation 4 in the figure.

Evaluation

- The state-of-the-art CNN models (VGG16 and VGG19, MobileNet and ResNet50).
- We use CIFAR-10 (CIFAR-100) and ImageNet dataset.

Evaluation (Cont'd)

Stage-level Optimizer

Table 3: Performance comparison (CIFAR-10) between original CNNs and reorganized structures.

Network	MFLOPs	Param.	Acc. (%)	$\Delta 3DRF$
VGG16	310	14.73M	92.64	-
VGG16-opt	370	5.10M	92.95	2.30
VGG19	400	20.04M	91.91	-
VGG19-opt	490	8.09M	92.89	3.13
MobileNet	50	3.22M	90.67	-
MobileNet-opt	50	1.13M	92.05	3.94
ResNet50	1,300	23.52M	93.75	-
ResNet50-opt	1,310	17.24M	95.79	0.76

Stage-level Optimizer

Table 4: Performance comparison (CIFAR-100) between original CNNs and reorganized structures.

Network	MFLOPs	Param.	Acc. (%)	$\Delta 3DRF$
VGG16	330	34.02M	72.93	-
VGG16-opt	390	24.39M	74.64	2.30
VGG19	420	39.33M	72.23	-
VGG19-opt	500	27.38M	74.00	3.13
MobileNet	50	3.32M	65.98	-
MobileNet-opt	50	1.23M	71.45	3.94
ResNet50	1,310	23.71M	77.39	-
ResNet50-opt	1,380	21.89M	78.25	0.76

Evaluation (cont'd)

Stage-level Optimizer

Table 5: Performance comparison (ImageNet) between original CNNs and reorganized structures.

Network	MFLOPs	Param.	Acc. (%)	$\Delta 3DRF$
VGG16	15,500	138.36M	71.59	-
VGG16-opt	16,900	133.82M	72.17	0.39
VGG19	19,670	143.67M	72.38	-
VGG19-opt	21,060	141.34M	72.61	1.09
MobileNet	580	4.23M	70.60	-
MobileNet-opt	570	3.52M	71.05	2.59
ResNet50	4,120	25.56M	76.15	-
ResNet50-opt	4,130	23.67M	76.56	0.47

Kernel-level Optimizer

Table 6: Kernel-level design (CIFAR-10) on VGG16-opt.

Network	MFLOPs	Param.	Acc.(%)
Baseline	370	9.64M	92.95
DW+PW	50	1.11M	92.12
DW+GPW-g2	30	0.67M	92.35
DW+GPW-g4	20	0.36M	88.05
DW+GPW-g8	10	0.20M	86.41
DW+RPW-g2-o33%	30	0.66M	92.52
DW+RPW-g2-o50%	30	0.66M	92.70
DW+RPW-g4-o33%	20	0.36M	91.61
DW+RPW-g4-o50%	20	0.36M	91.59
DW+RPW-g8-o33%	10	0.20M	89.86
DW+RPW-g8-o50%	10	0.20M	90.19

Thank you