CCF-CV走进高校系列报告会
西安电子科技大学, 2016.12.28

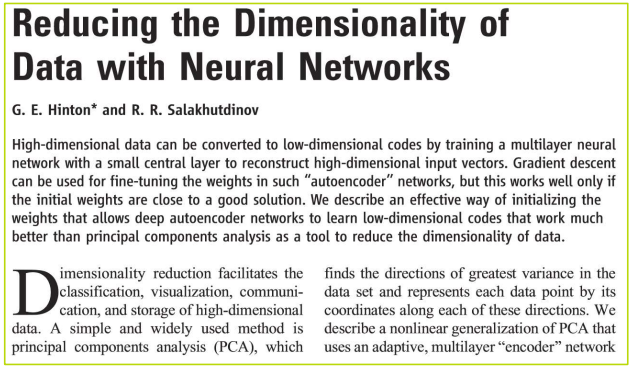
Deep Fusion:
A Neural Network Architecture Design Pattern

Jingdong Wang
Lead Researcher
Microsoft Research, Beijing, China

Deep learning in the past 10 years

- Reducing the dimensionality of data with neural networks, Science, 2006
  - Fast learning algorithms for Restricted Boltzmann machine

Do not work as expected



Reducing the Dimensionality of Data with Neural Networks

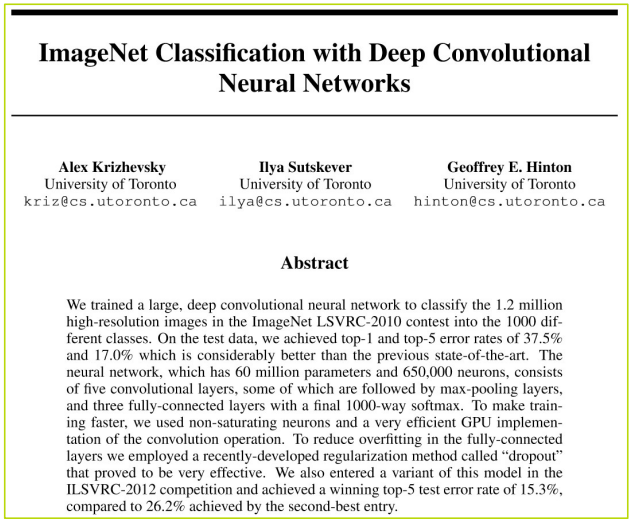
G. E. Hinton* and R. R. Salakhutdinov

High-dimensional data can be converted to low-dimensional codes by training a multilayer neural network with a small central layer to reconstruct high-dimensional input vectors. Gradient descent can be used for fine-tuning the weights in such "autoencoder" networks, but this works well only if the initial weights are close to a good solution. We describe an effective way of initializing the weights that allows deep autoencoder networks to learn low-dimensional codes that work much better than principal components analysis as a tool to reduce the dimensionality of data.

Dimensionality reduction facilitates the classification, visualization, communication, and storage of high-dimensional data. A simple and widely used method is principal components analysis (PCA), which finds the directions of greatest variance in the data set and represents each data point by its coordinates along each of these directions. We describe a nonlinear generalization of PCA that uses an adaptive, multilayer "encoder" network

- ImageNet Classification with deep convolutional neural networks, NIPS, 2012
  - Dramatic performance improvement
  - ImageNet, GPU

Win almost in all the applications



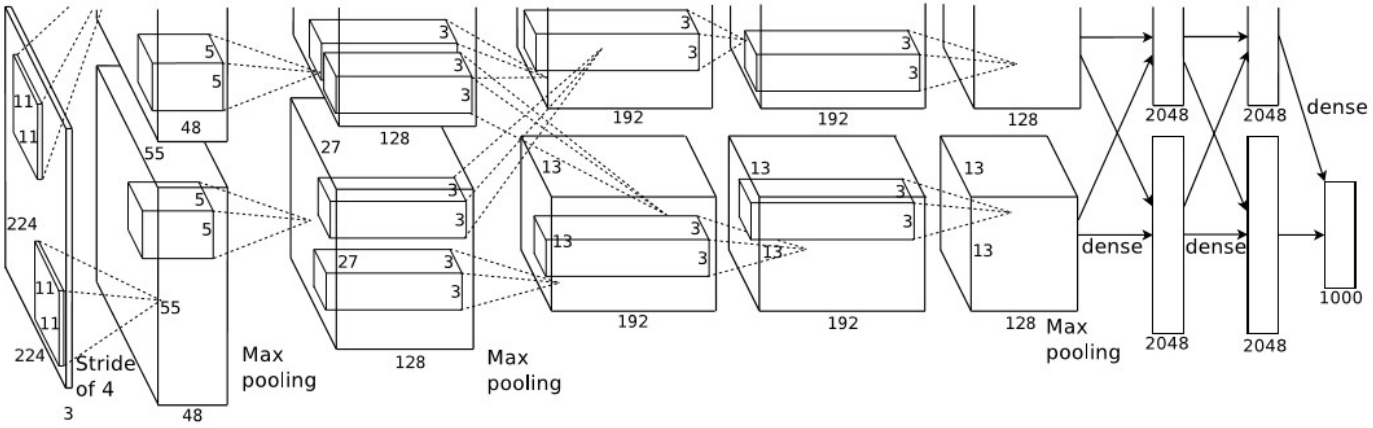
ImageNet Classification with Deep Convolutional Neural Networks

Alex Krizhevsky University of Toronto kriz@cs.utoronto.ca
Ilya Sutskever University of Toronto ilya@cs.utoronto.ca
Geoffrey E. Hinton University of Toronto hinton@cs.utoronto.ca

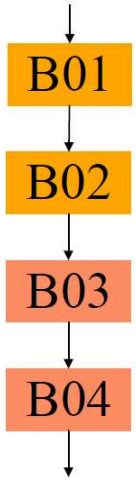
Abstract

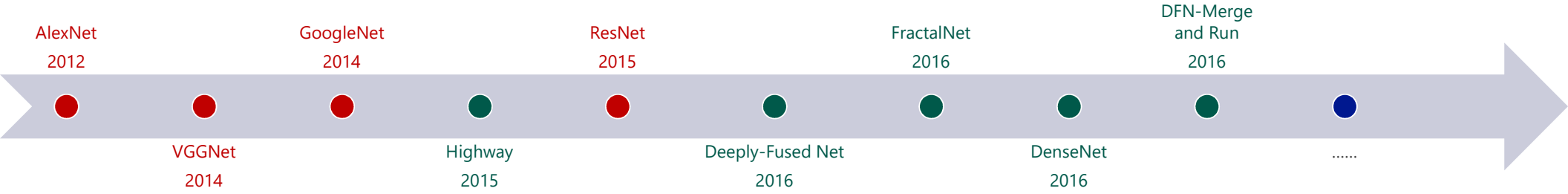
We trained a large, deep convolutional neural network to classify the 1.2 million high-resolution images in the ImageNet LSVRC-2010 contest into the 1000 different classes. On the test data, we achieved top-1 and top-5 error rates of 37.5% and 17.0% which is considerably better than the previous state-of-the-art. The neural network, which has 60 million parameters and 650,000 neurons, consists of five convolutional layers, some of which are followed by max-pooling layers, and three fully-connected layers with a final 1000-way softmax. To make training faster, we used non-saturating neurons and a very efficient GPU implementation of the convolution operation. To reduce overfitting in the fully-connected layers we employed a recently-developed regularization method called "dropout" that proved to be very effective. We also entered a variant of this model in the ILSVRC-2012 competition and achieved a winning top-5 test error rate of 15.3%, compared to 26.2% achieved by the second-best entry.

AlexNet, 2012



7 layers





Dropout: A Simple Way to Prevent Neural Networks from Overfitting

Nitish Srivastava NITISH@CS.TORONTO.EDU
Geoffrey Hinton HINTON@CS.TORONTO.EDU
Alex Krizhevsky KRIZ@CS.TORONTO.EDU
Ilya Sutskever ILYA@CS.TORONTO.EDU
Ruslan Salakhutdinov RSALAKHU@CS.TORONTO.EDU
Department of Computer Science
University of Toronto
10 Kings College Road, Rm 3302
Toronto, Ontario, M5S 3G4, Canada.

Dropout ≈ Ensembling

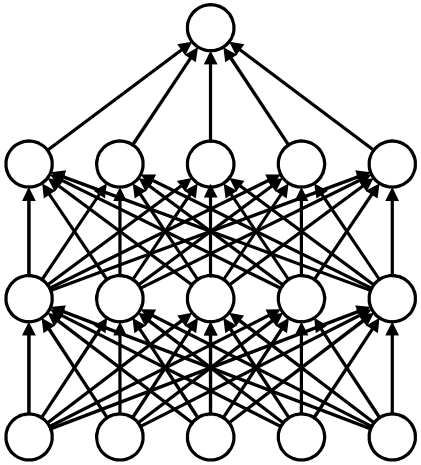
Editor: Yoshua Bengio

Abstract

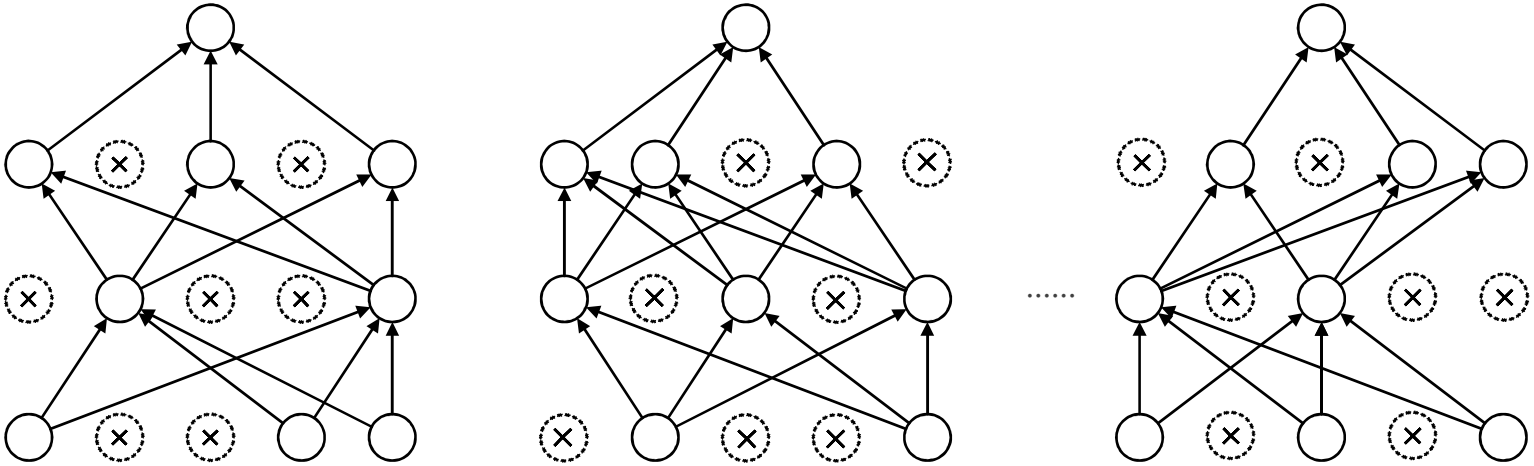
Deep neural nets with a large number of parameters are very powerful machine learning systems. However, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. During training, dropout samples from an exponential number of different "thinned" networks. At test time, it is easy to approximate the effect of averaging the predictions of all these thinned networks by simply using a single unthinned network that has smaller weights. This significantly reduces overfitting and gives major improvements over other regularization methods. We show that dropout improves the performance of neural networks on supervised learning tasks in vision, speech recognition, document classification and computational biology, obtaining state-of-the-art results on many benchmark data sets.

Keywords: neural networks, regularization, model combination, deep learning

Dropout



Predict with
an approximate ensemble

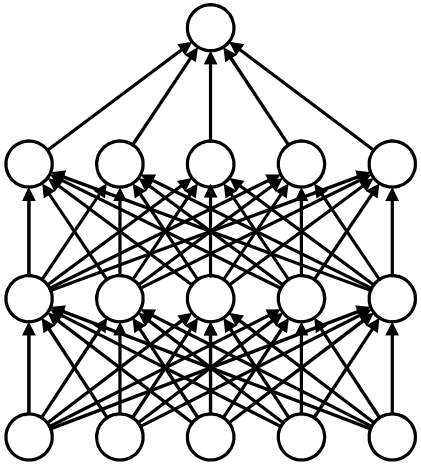


Train thinned nets generated from dropout: Each iteration trains a thinned net

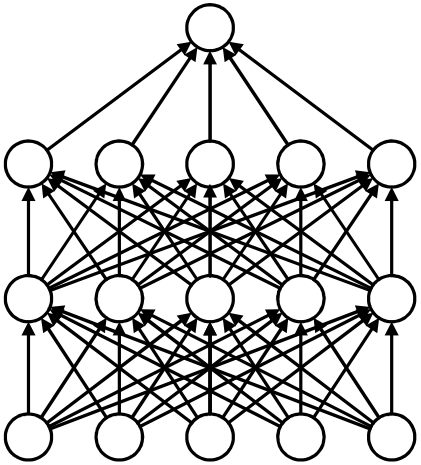
Train exponential number of thinned networks with the same training data

Predict from one network like ensembling an exponential number of thinned networks

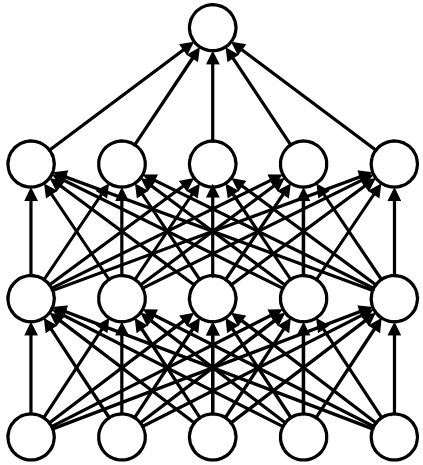
DisturbLabel



Predict with
an approximate ensemble

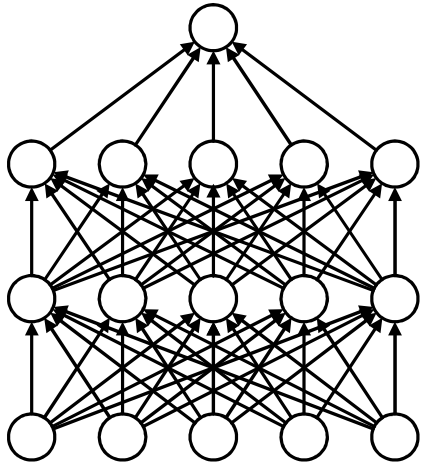


$\{(\mathbf{x}_n, \tilde{\mathbf{y}}_n^1)\}$



$\{(\mathbf{x}_n, \tilde{\mathbf{y}}_n^2)\}$

.....



.....

$\{(\mathbf{x}_n, \tilde{\mathbf{y}}_n^t)\}$

Train the network: with a different training set in each iteration

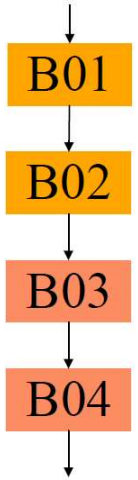
Train the network of same structure with exponential number of different training data

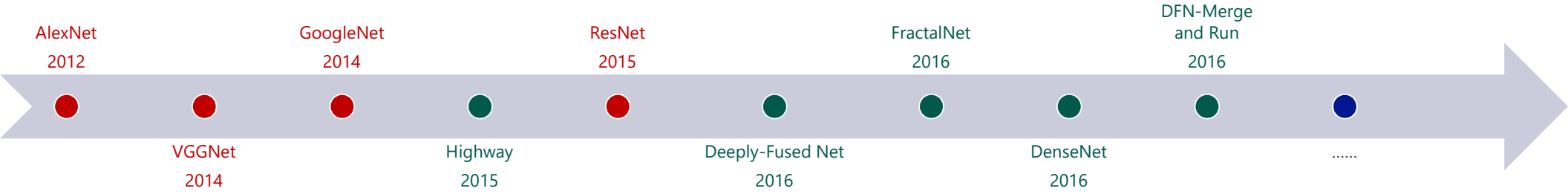
Predict from one network like ensembling an exponential number of networks of same structure

Lingxi Xie, Jingdong Wang, Zhen Wei, Meng Wang, Qi Tian: DisturbLabel: Regularizing CNN on the Loss Layer. CVPR (2016)

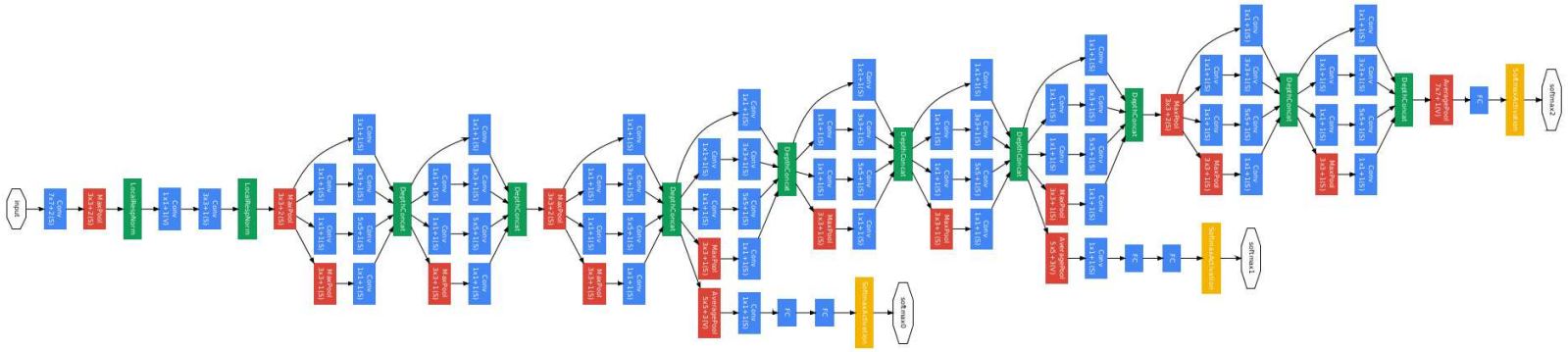
VGGNet, 2014

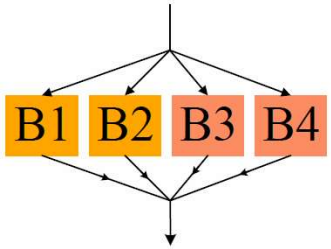
19 layers



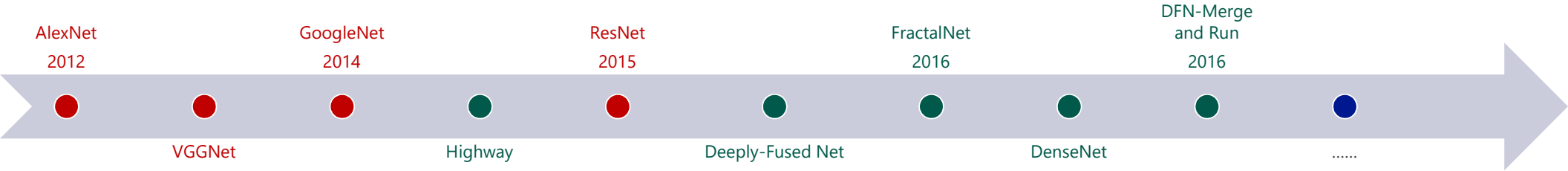


GoogLeNet, 2014

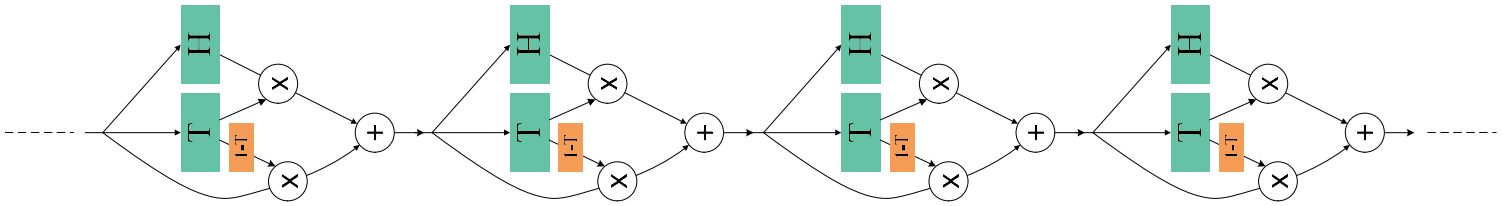


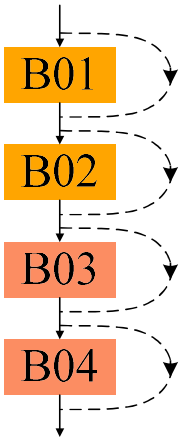


22 layers, Inception: multi-branch, concatenation fusion

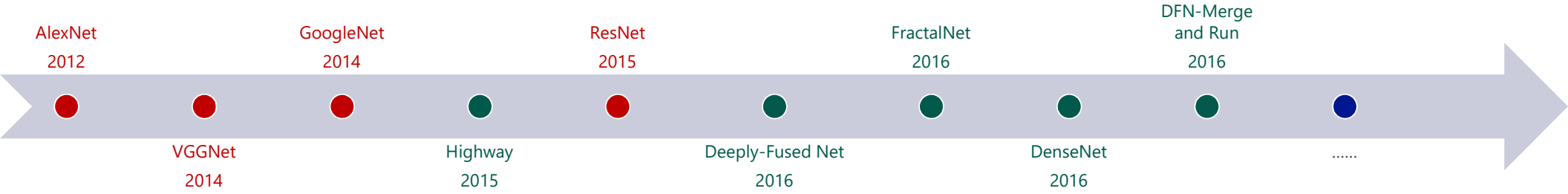


Highway, 2015

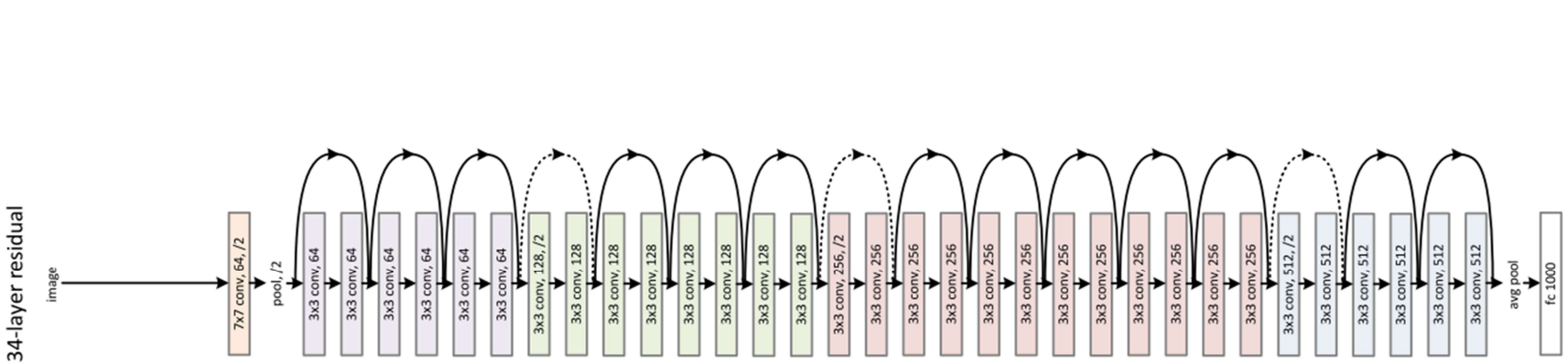


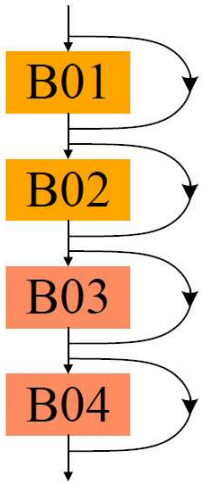


100+ layers, skip connection **eases** the **training** of a very deep network

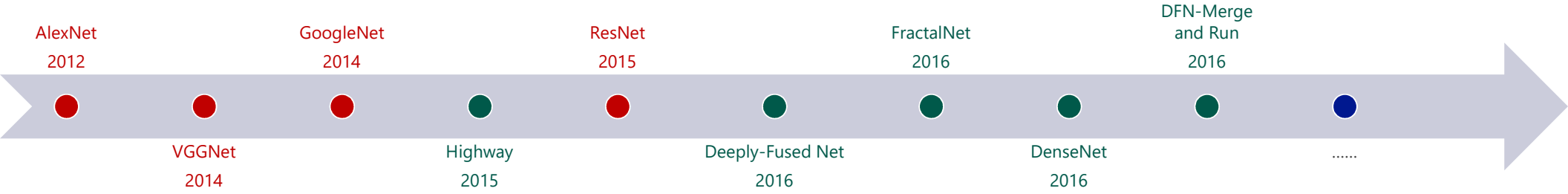


ResNet, 2015

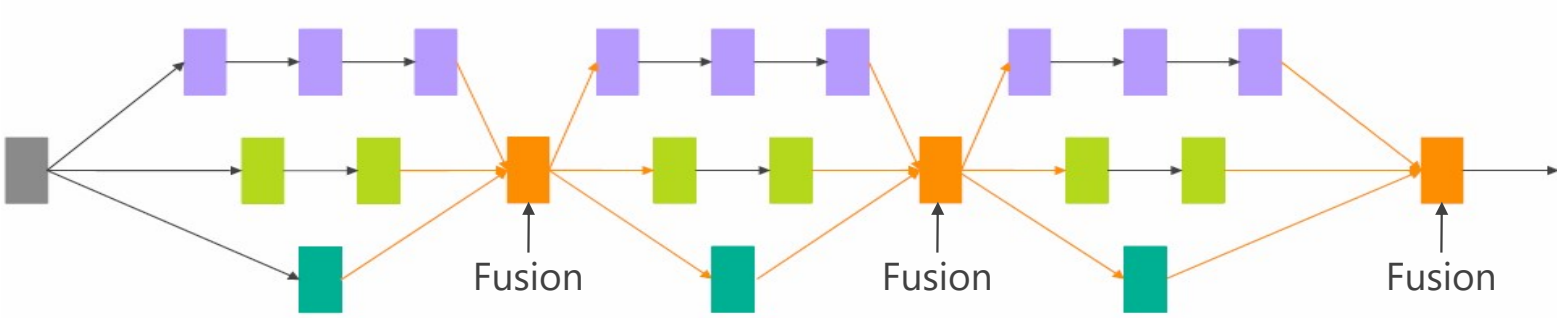


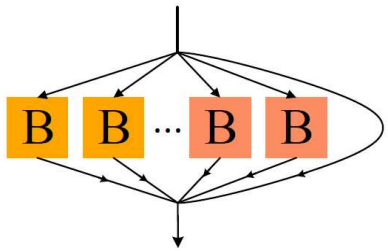


1000+ layers: deeper is better

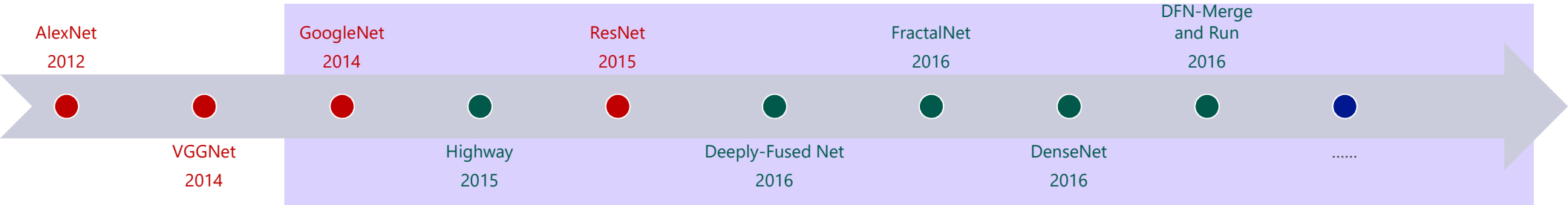


Deeply-fused net, 2016

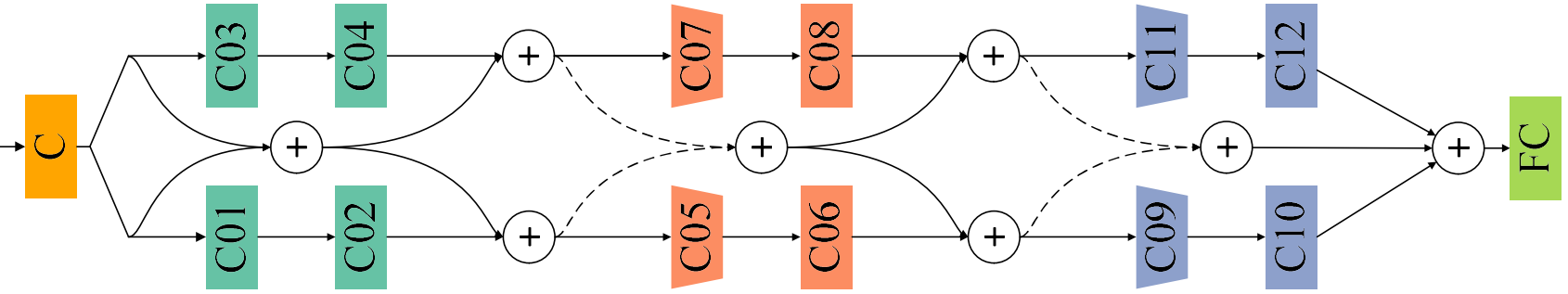


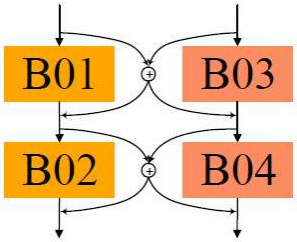


Multiple weight-shared paths: **Unifying** GoogLeNet, Highway, ResNet

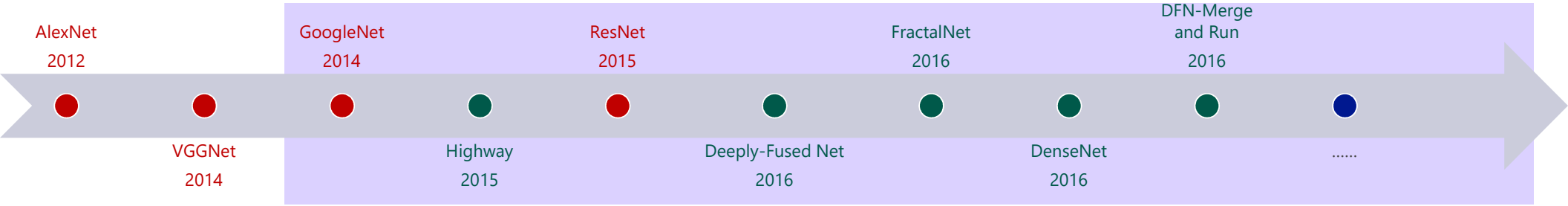


Merge and run, 2016





Deep fusion by merge and run: **shallower could be better**

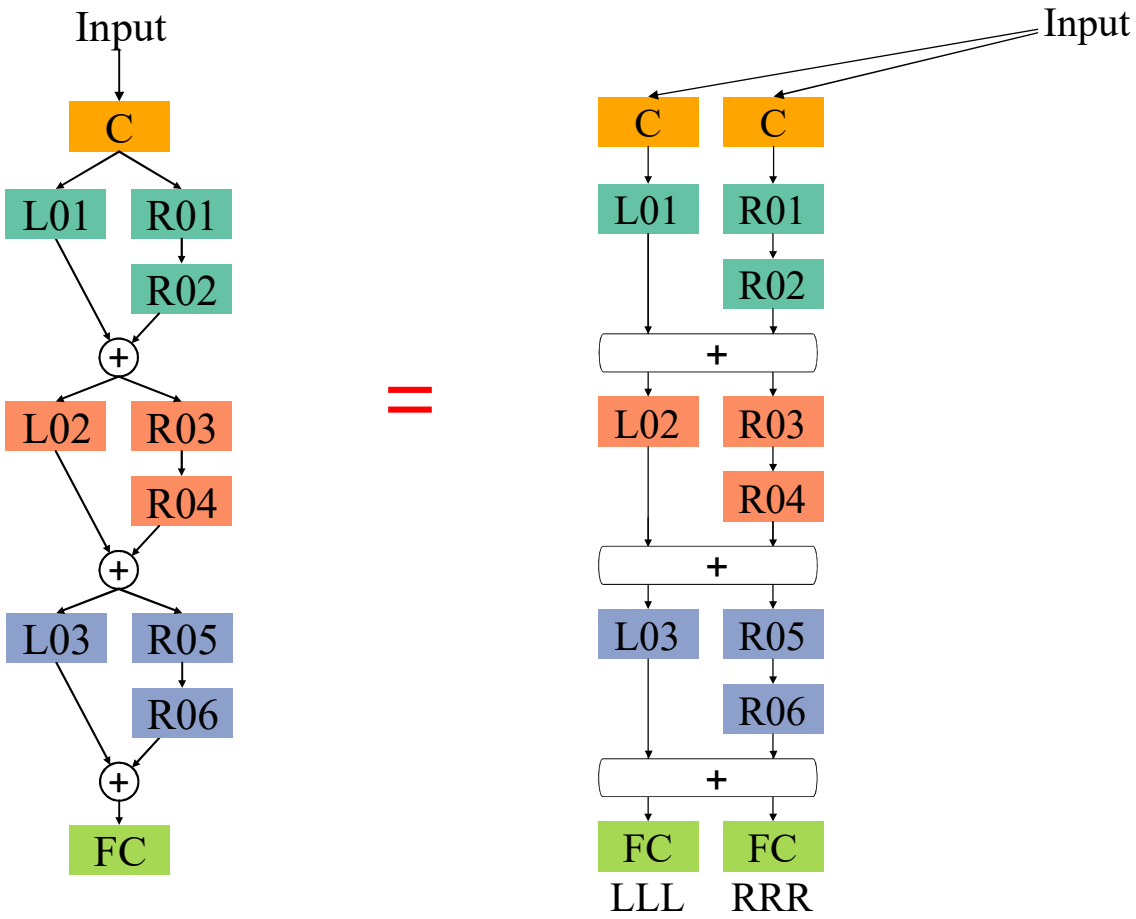


Deeply-Fused Nets

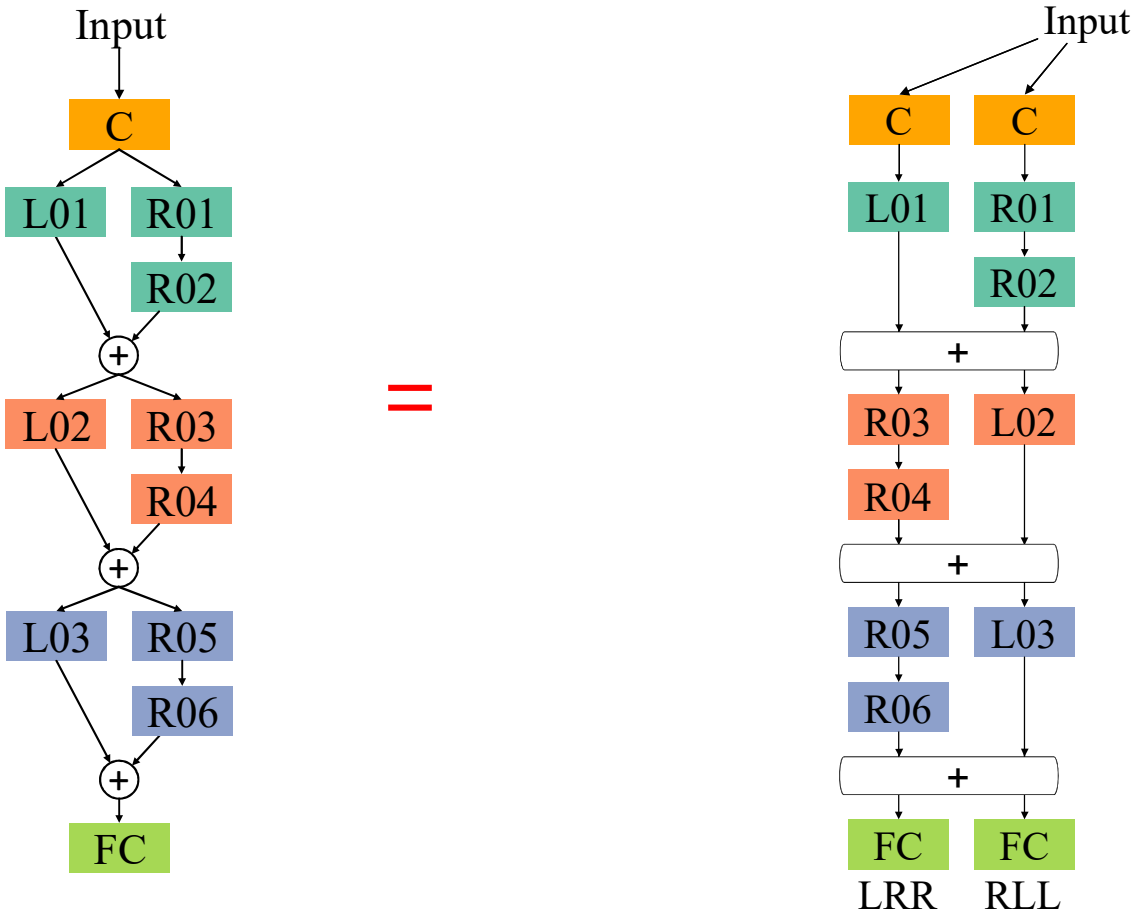
Jingdong Wang, Zhen Wei, Ting Zhang, Wenjun Zeng: Deeply-Fused Nets. CoRR abs/1605.07716 (2016)

A deeply-fused net can be formed from **many different** base networks

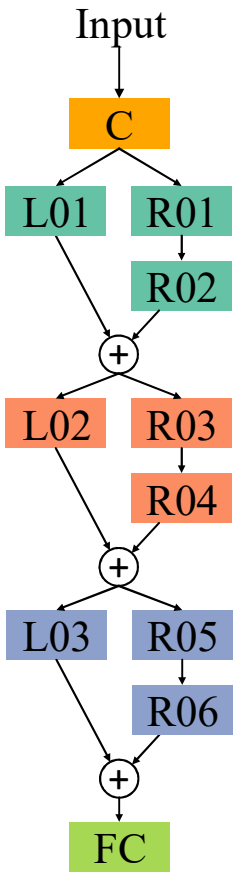
A deeply-fused net can be formed from **many different** base networks



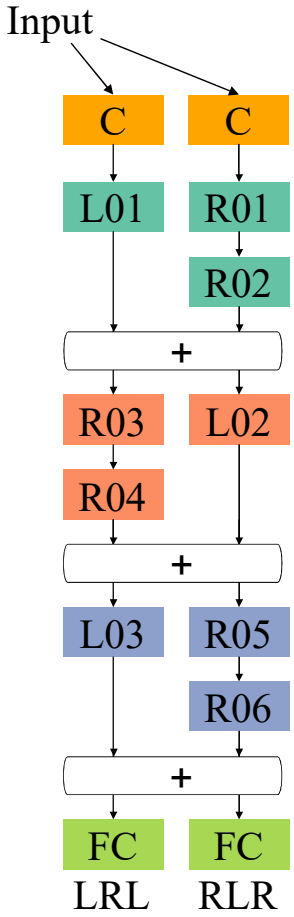
A deeply-fused net can be formed from **many different** base networks



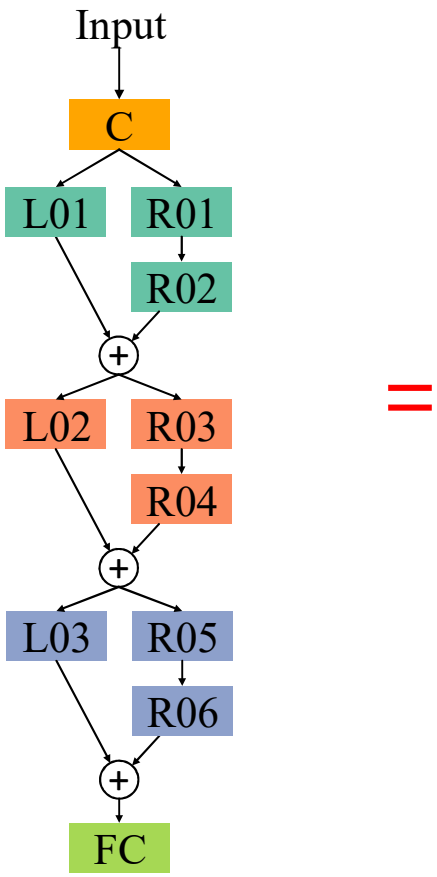
A deeply-fused net can be formed from **many different** base networks



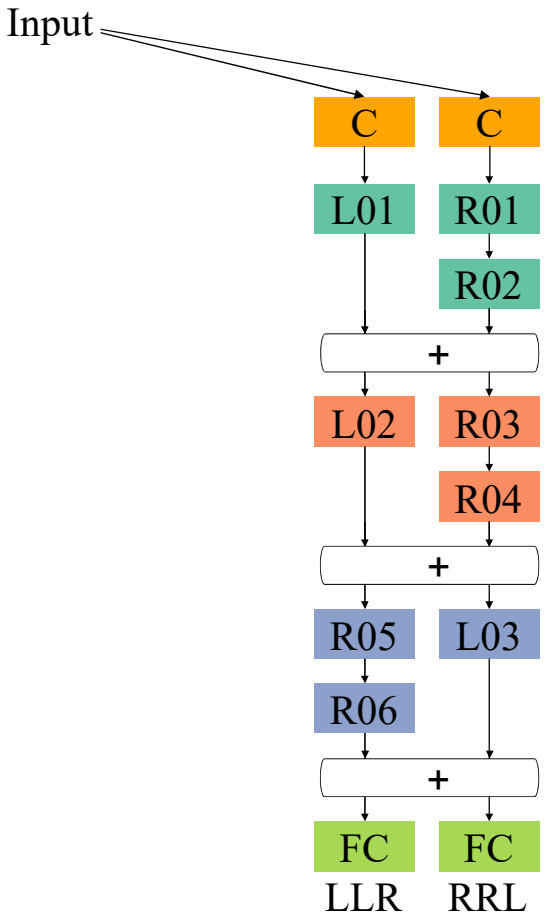
=



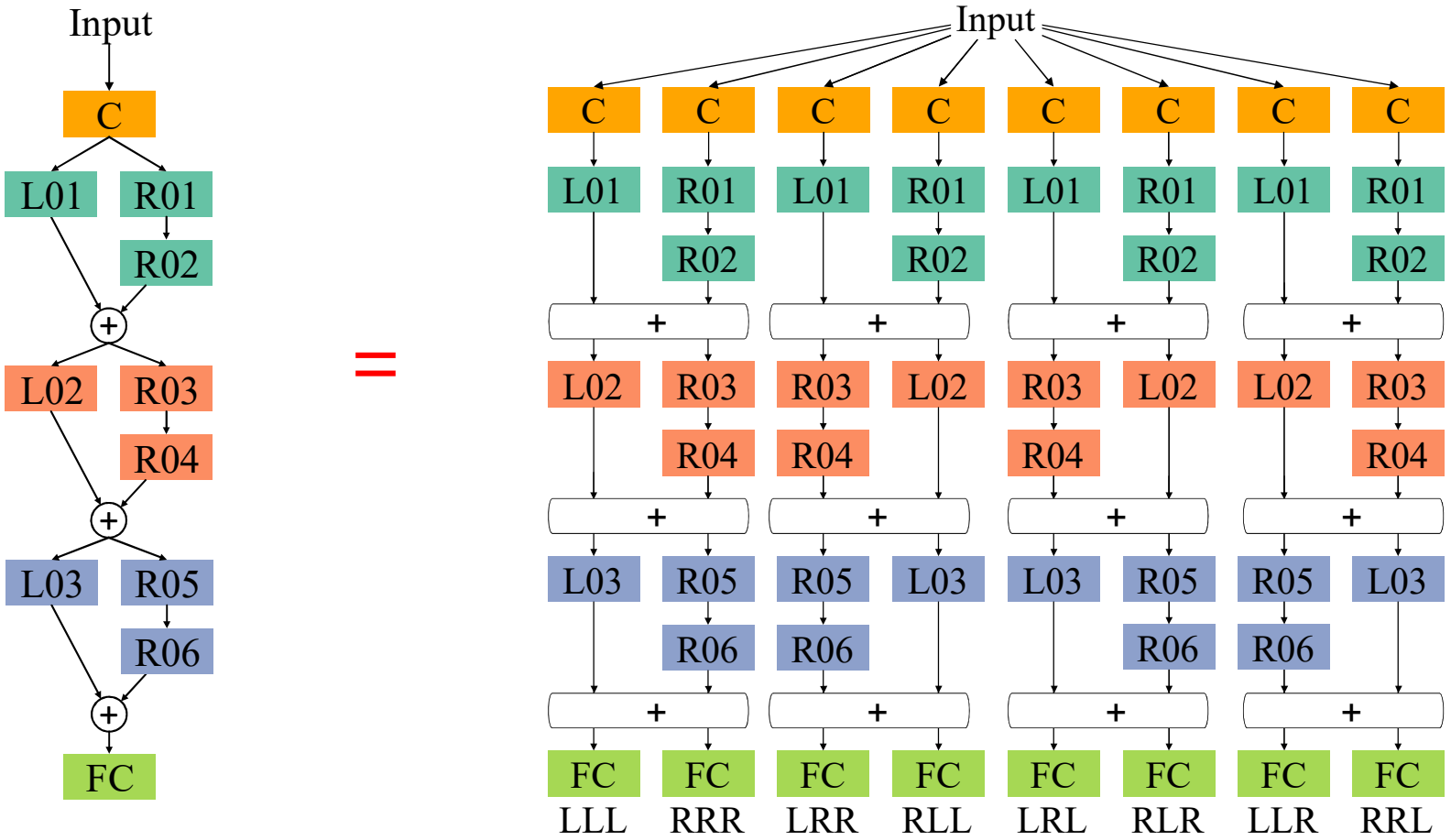
A deeply-fused net can be formed from **many different** base networks



=

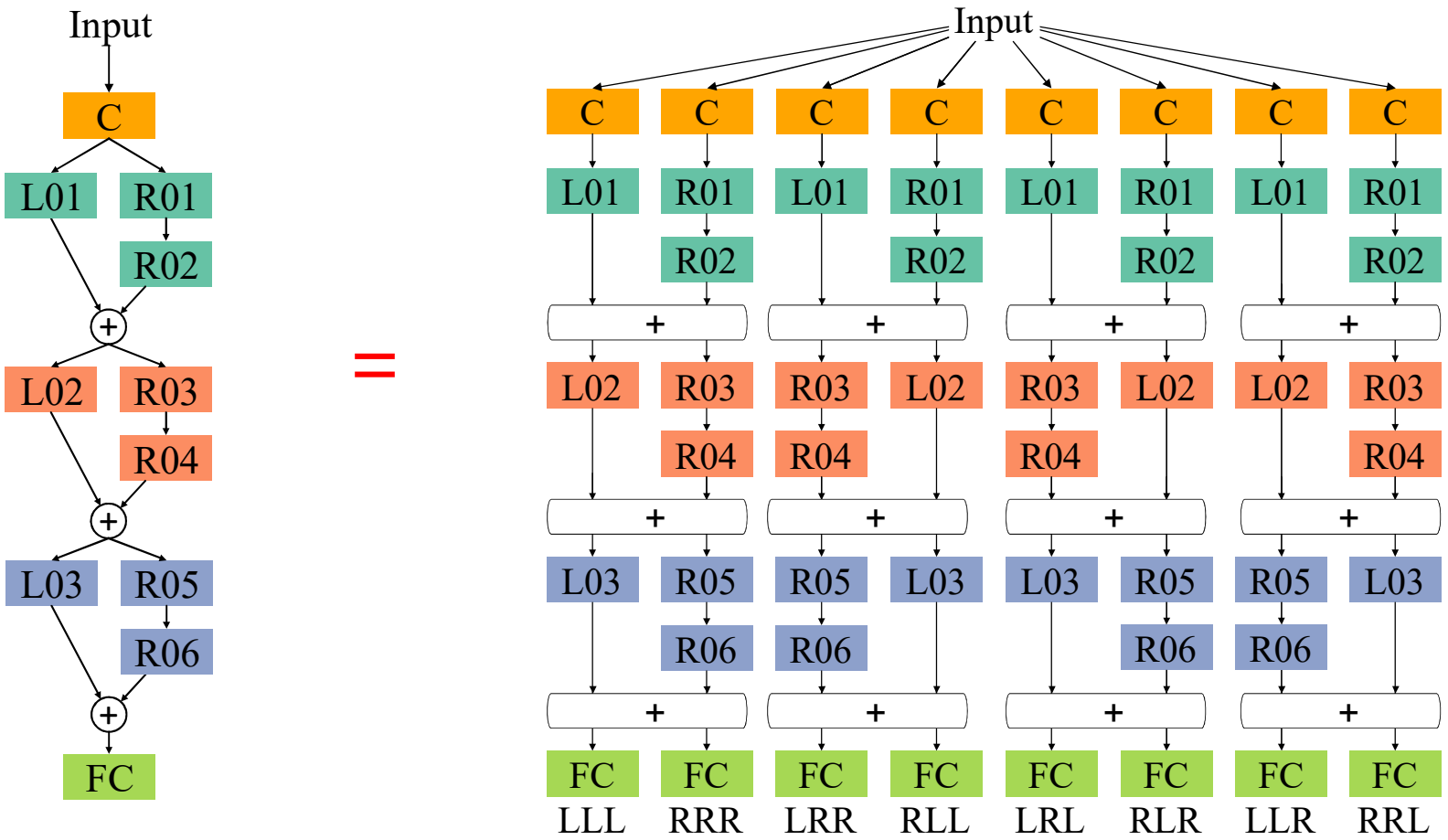


A deeply-fused net can be formed from **many different** base networks

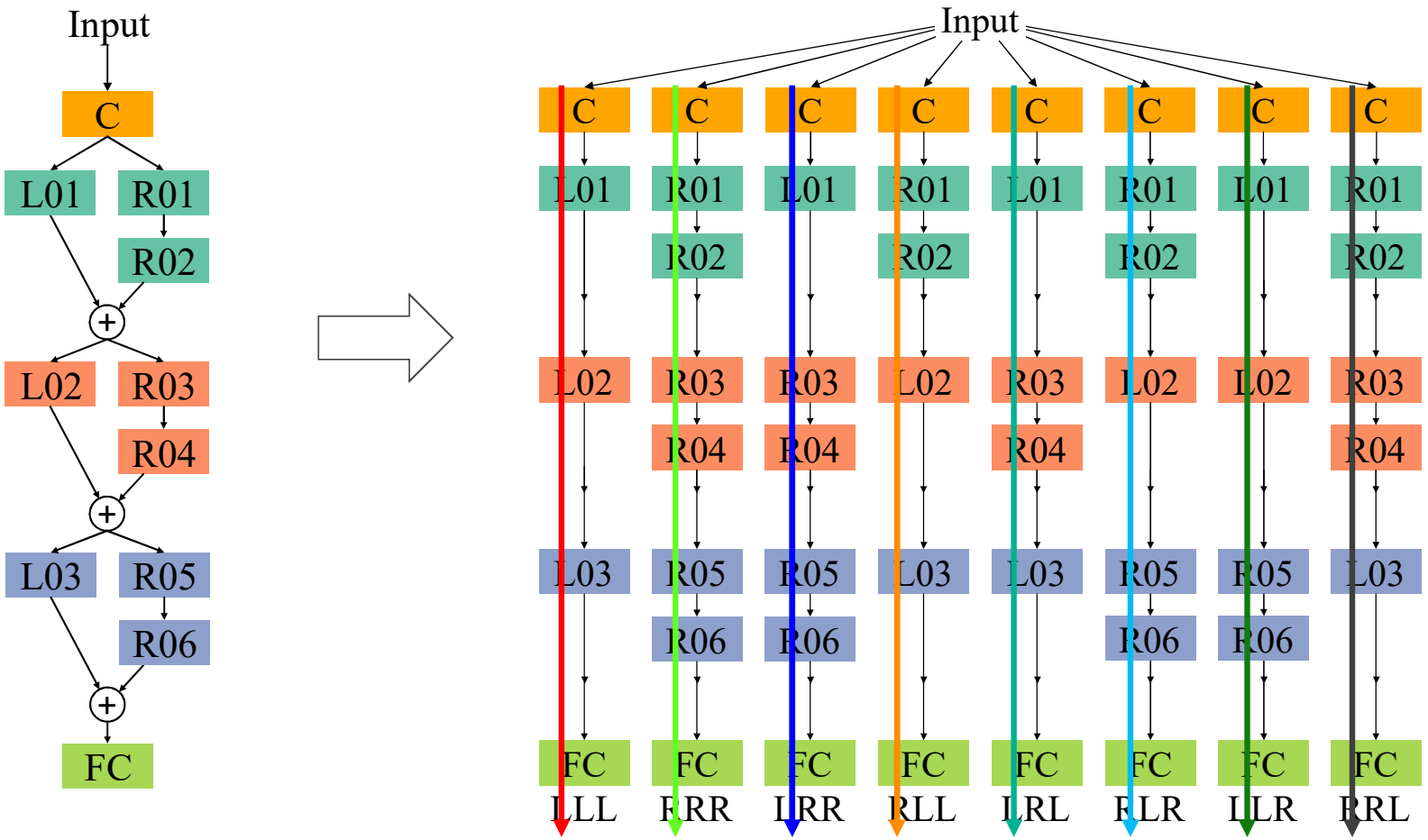


A deeply-fused net is a **multi-path multi-scale** network

A deeply-fused net is a **multi-path multi-scale** network

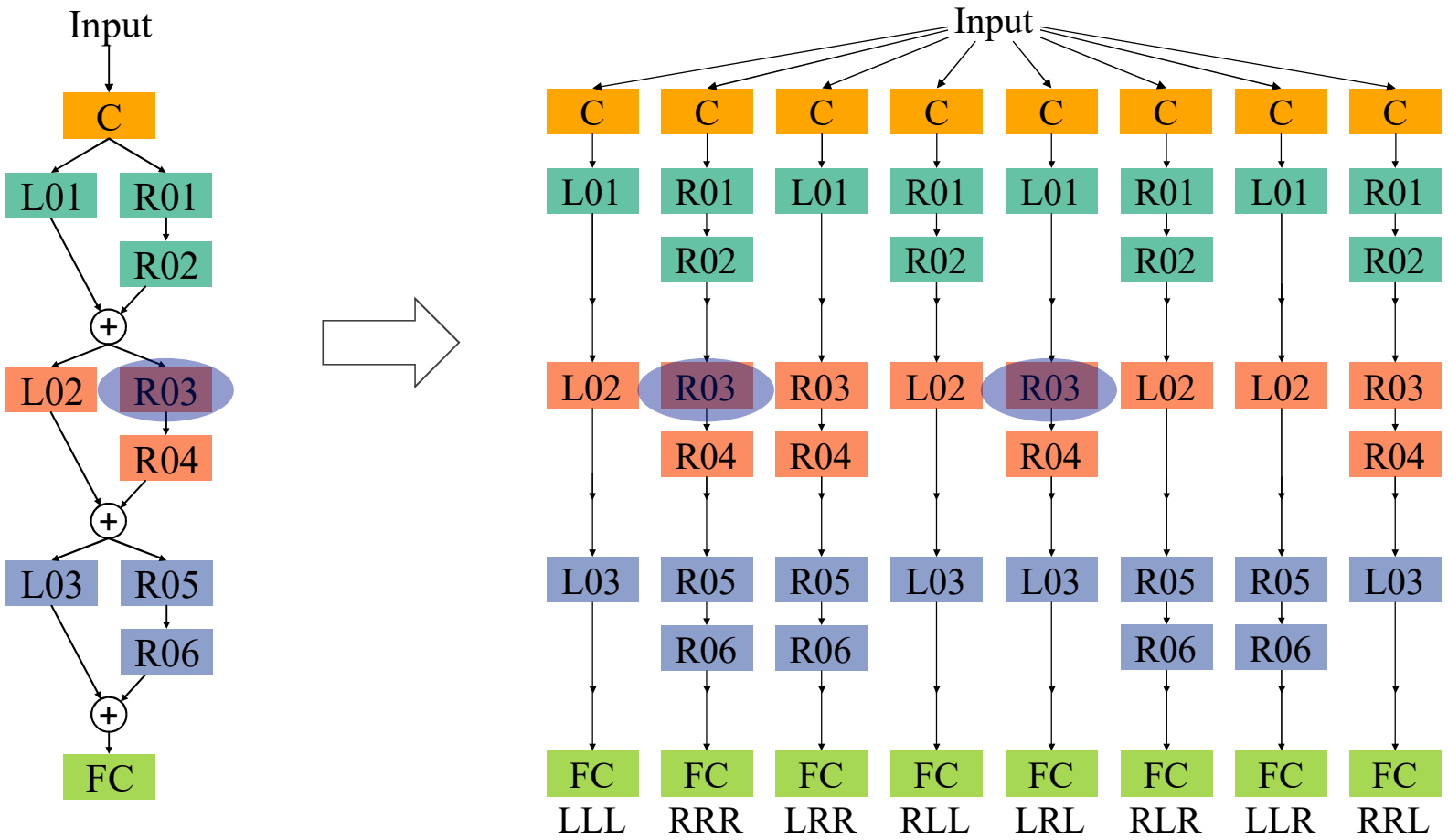


A deeply-fused net is a **multi-path multi-scale** network

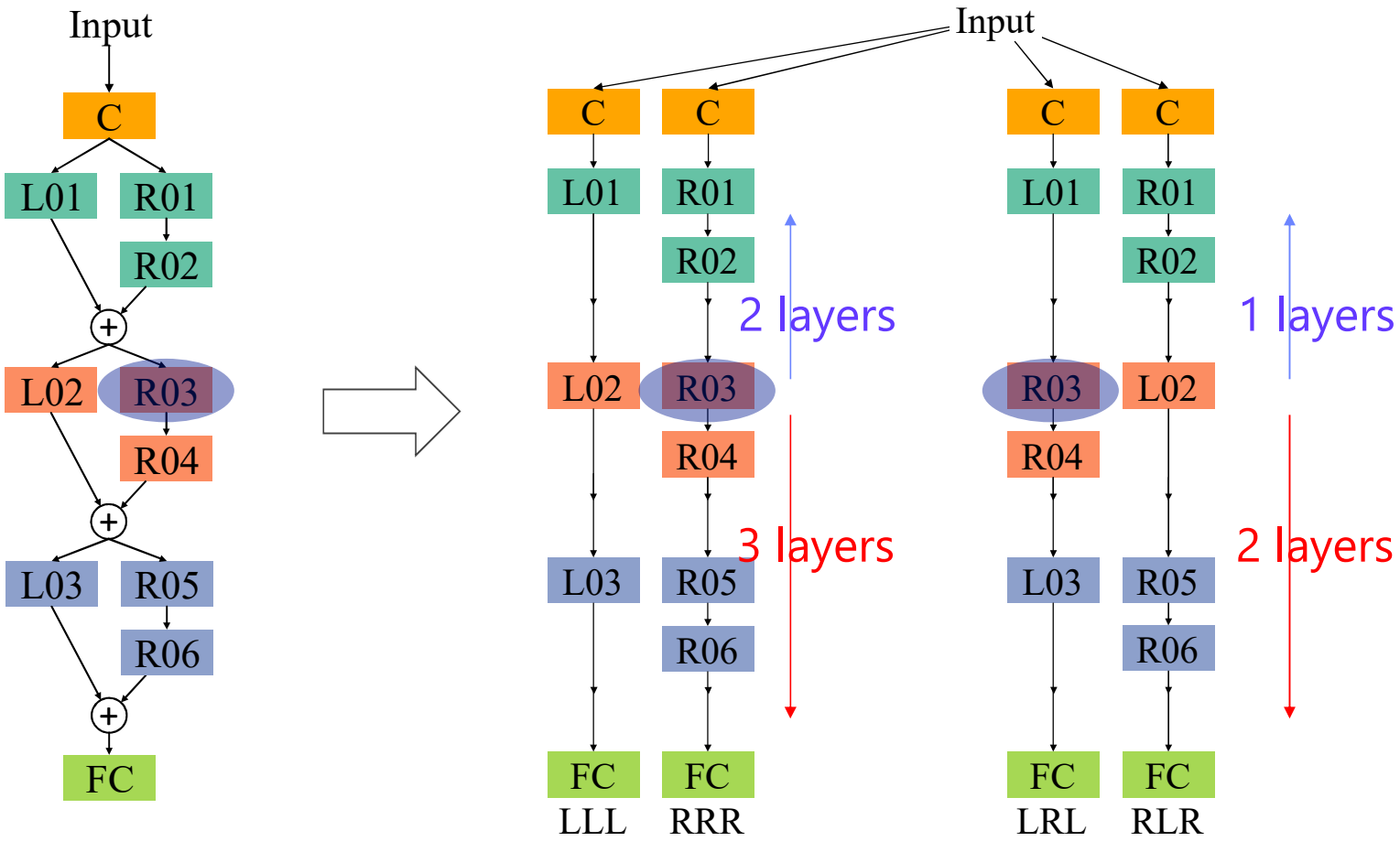


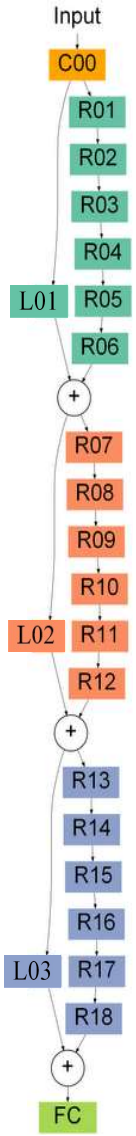
Each layer has an **express way** to input and output

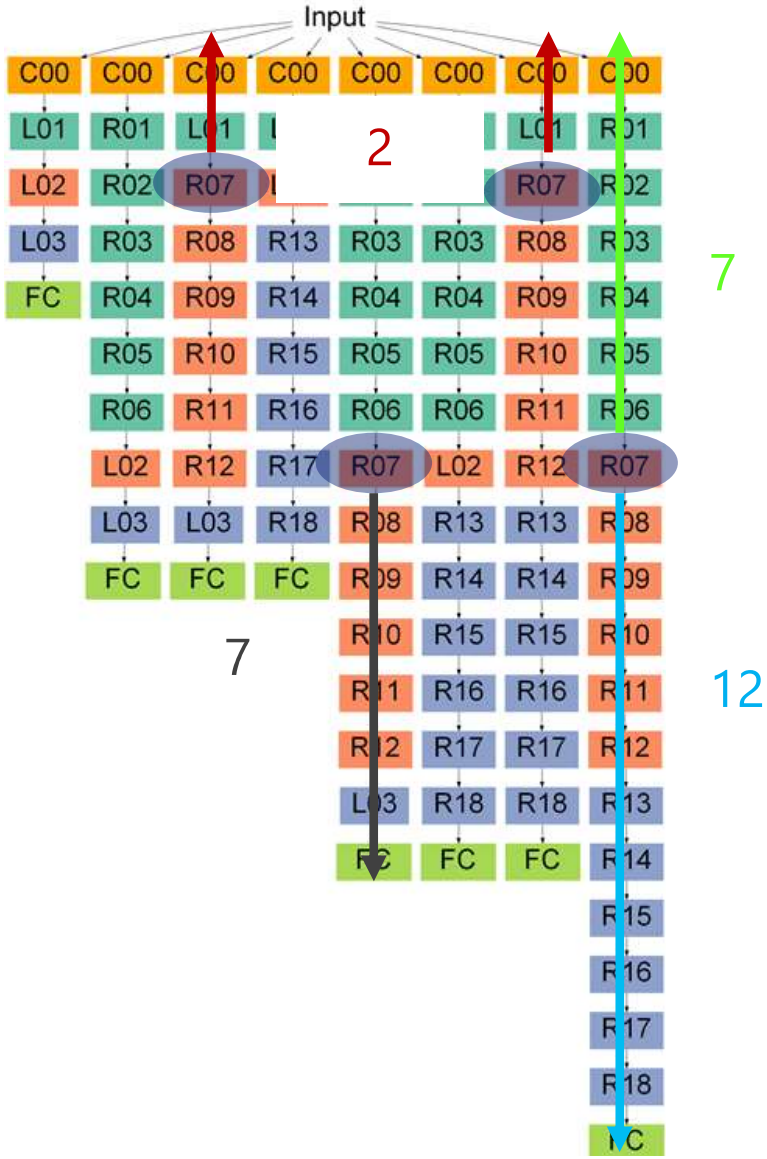
Each layer has an **express way** to input and output



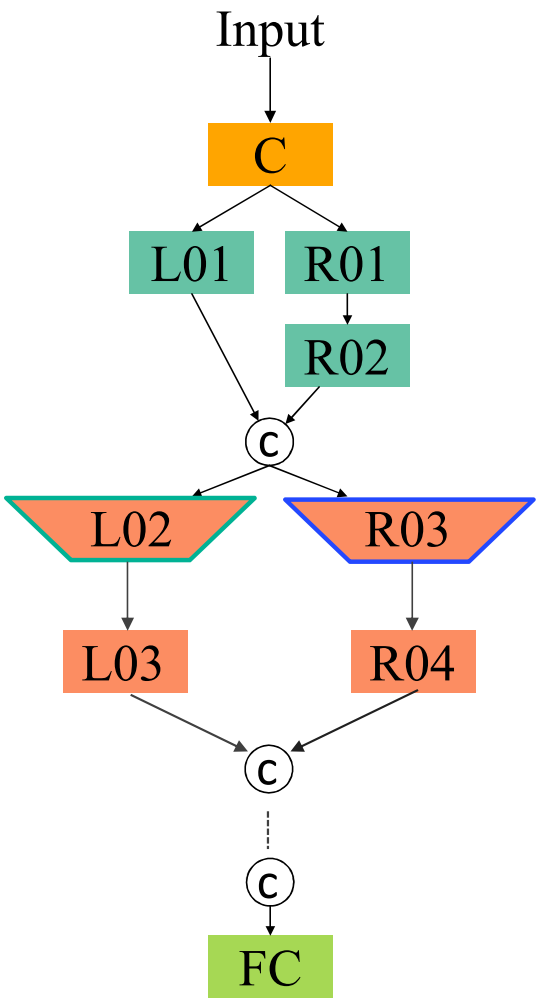
Each layer has an **express way** to input and output



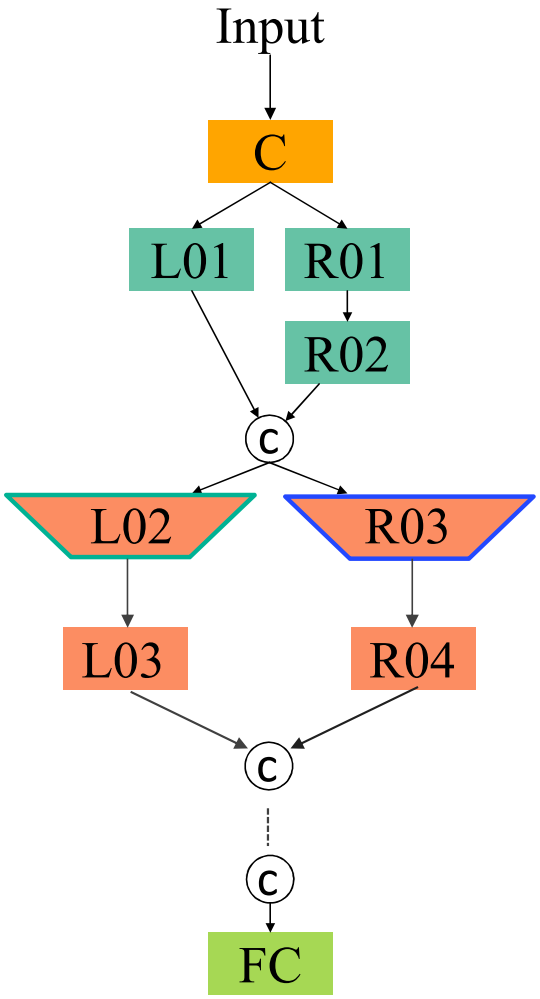

Input
C00
R01
R02
R03
R04
L01
R05
R06
+
R07
R08
R09
R10
L02
R11
R12
+
R13
R14
R15
R16
L03
R17
R18
+
FC

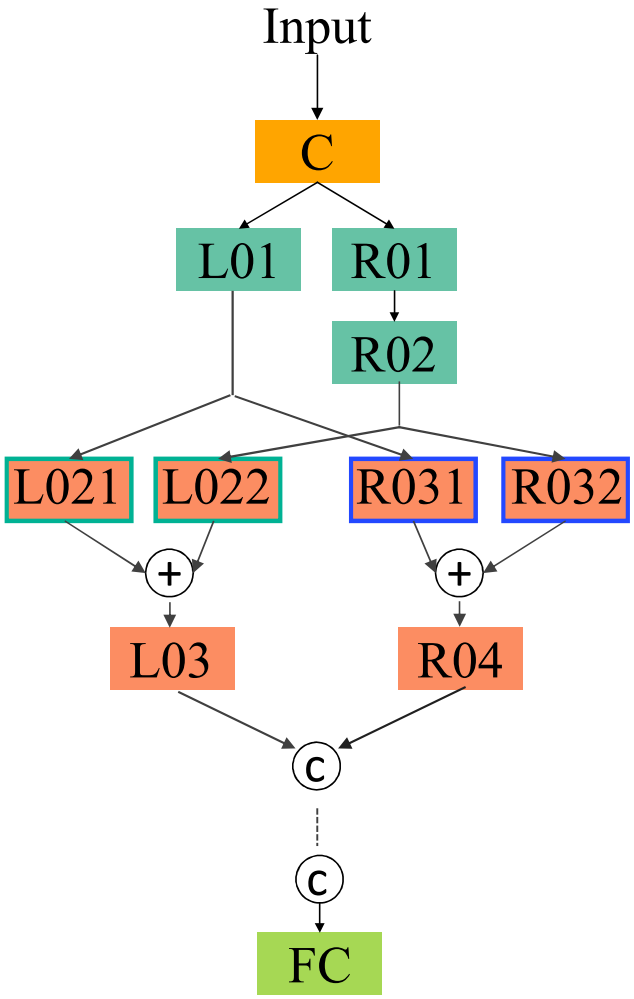

Input
2
7
7
12

Concatenation in GoogLeNet inception is a sum fusion



Concatenation in GoogLeNet inception is a sum fusion



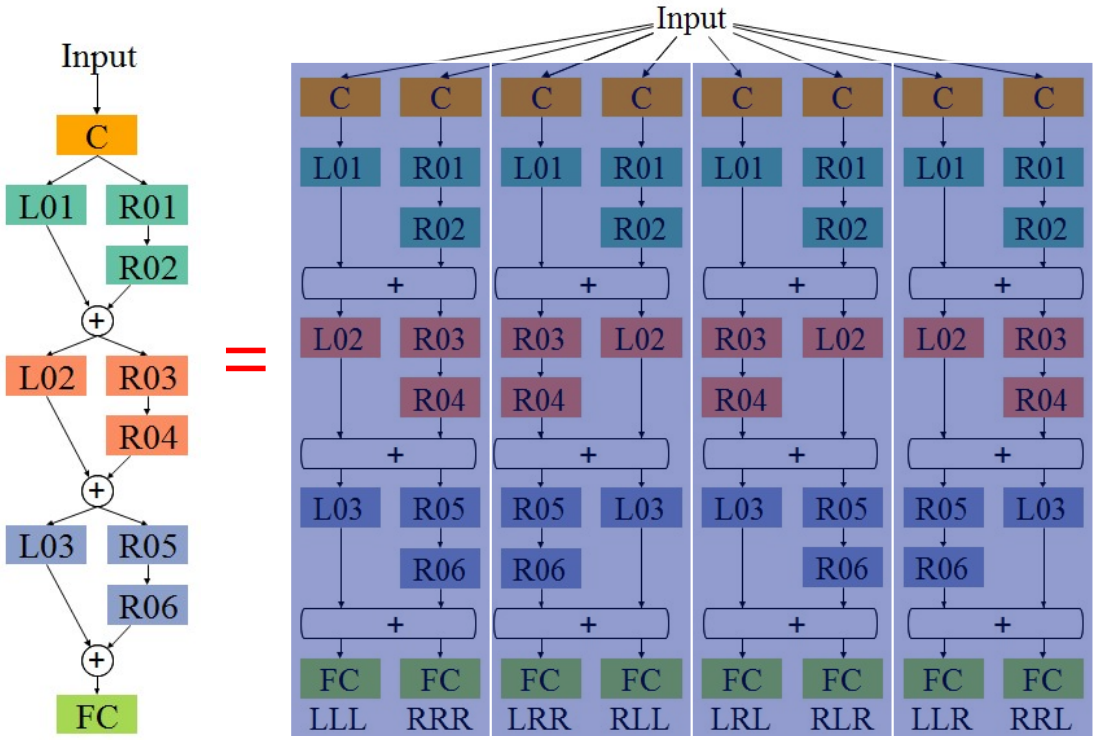


Deep fusion behaves like ensembling

Liming Zhao, Jingdong Wang, Xi Li, Zhuowen Tu, Wenjun Zeng: On the Connection of Deep Fusion to Ensembling. CoRR abs/1611.07718 (2016)

The **architecture** of a deeply-fused net **resembles** an ensemble

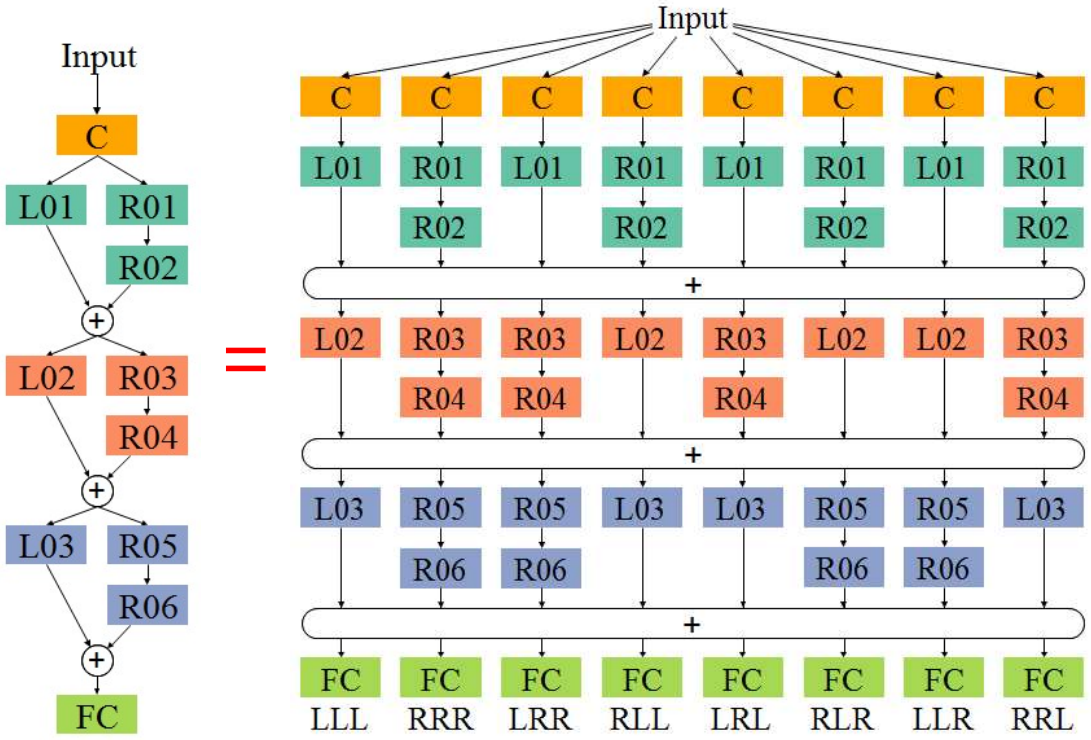
The **architecture** of a deeply-fused net **resembles** an ensemble



Deep fusion

Expanded view

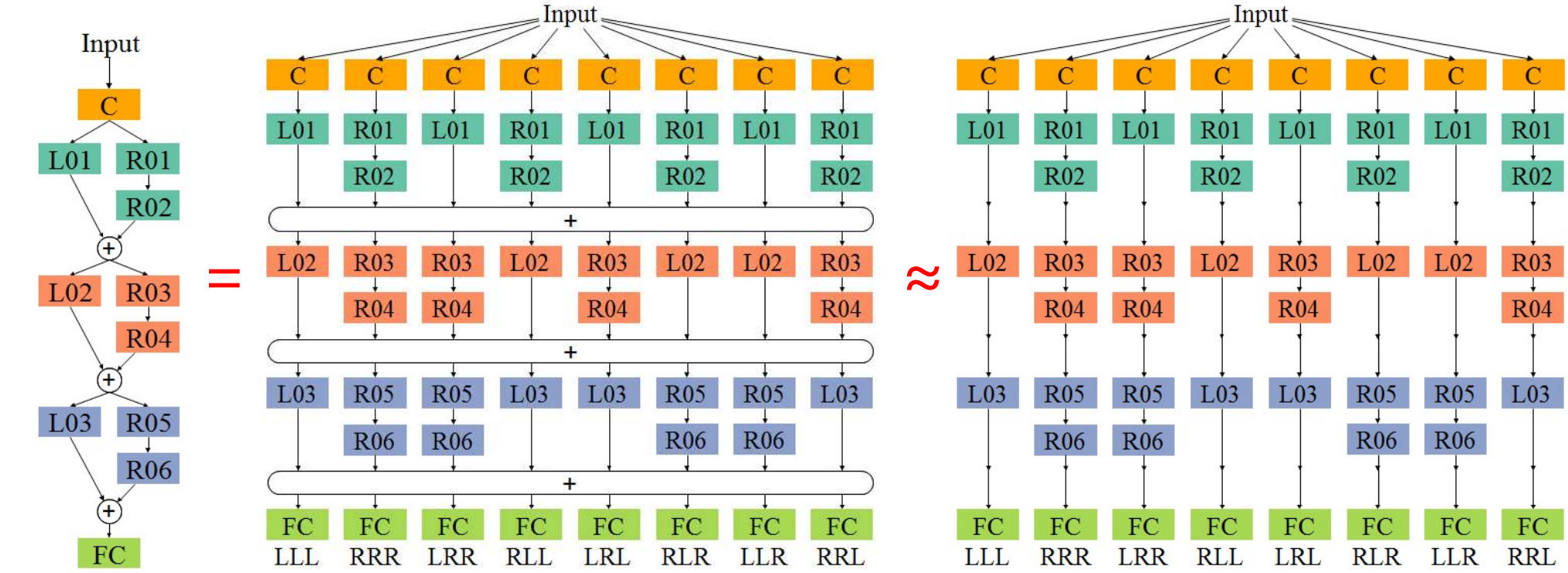
The **architecture** of a deeply-fused net **resembles** an ensemble



Deep fusion

Expanded view

The **architecture** of a deeply-fused net **resembles** an ensemble

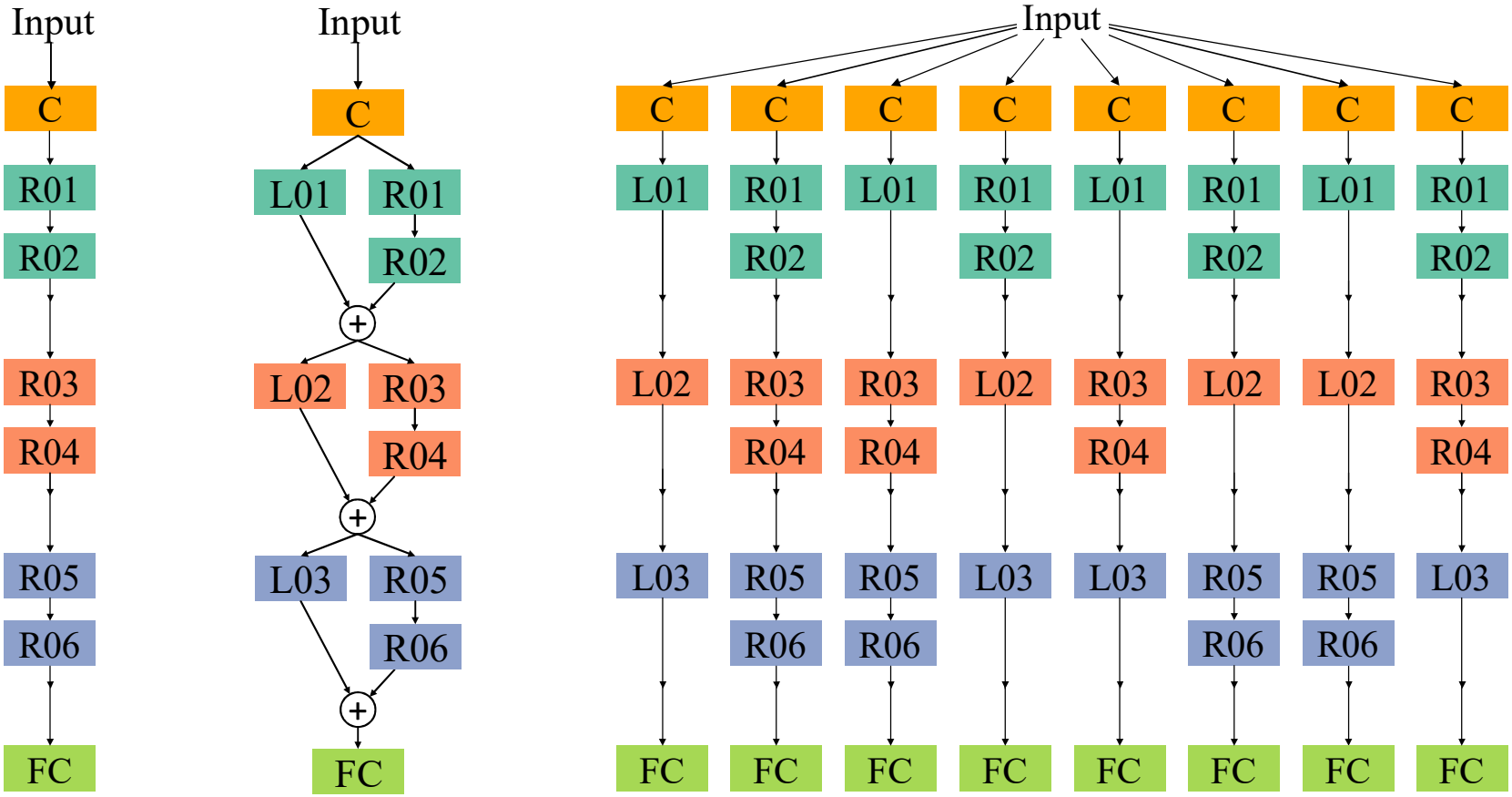


Deep fusion

Expanded view

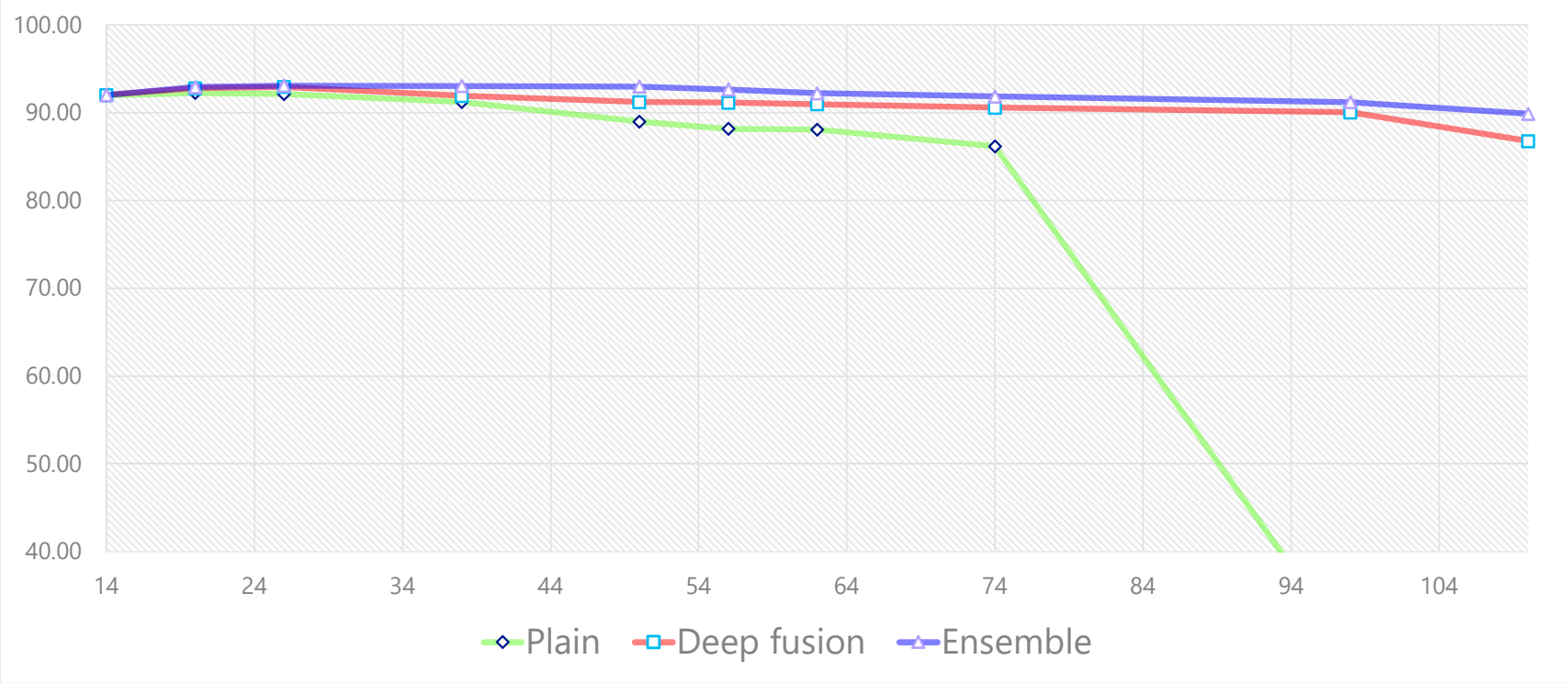
An ensemble

The **performances** of deeply-fused nets **resemble** ensembles

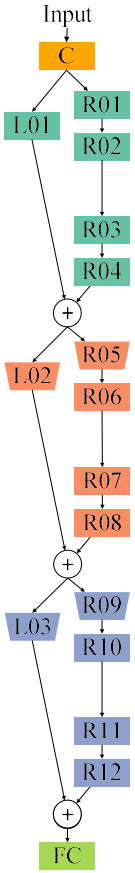


Plain Deep fusion Ensemble

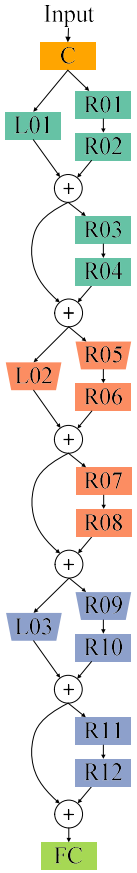
The **performances** of deeply-fused nets **resemble** ensembles



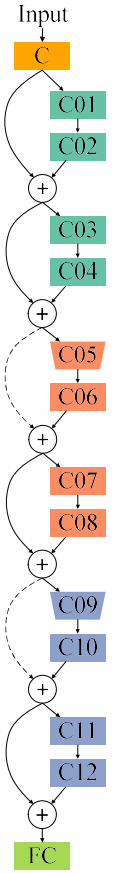
Larger ensemble sizes leads to **better performances**



3 fusions

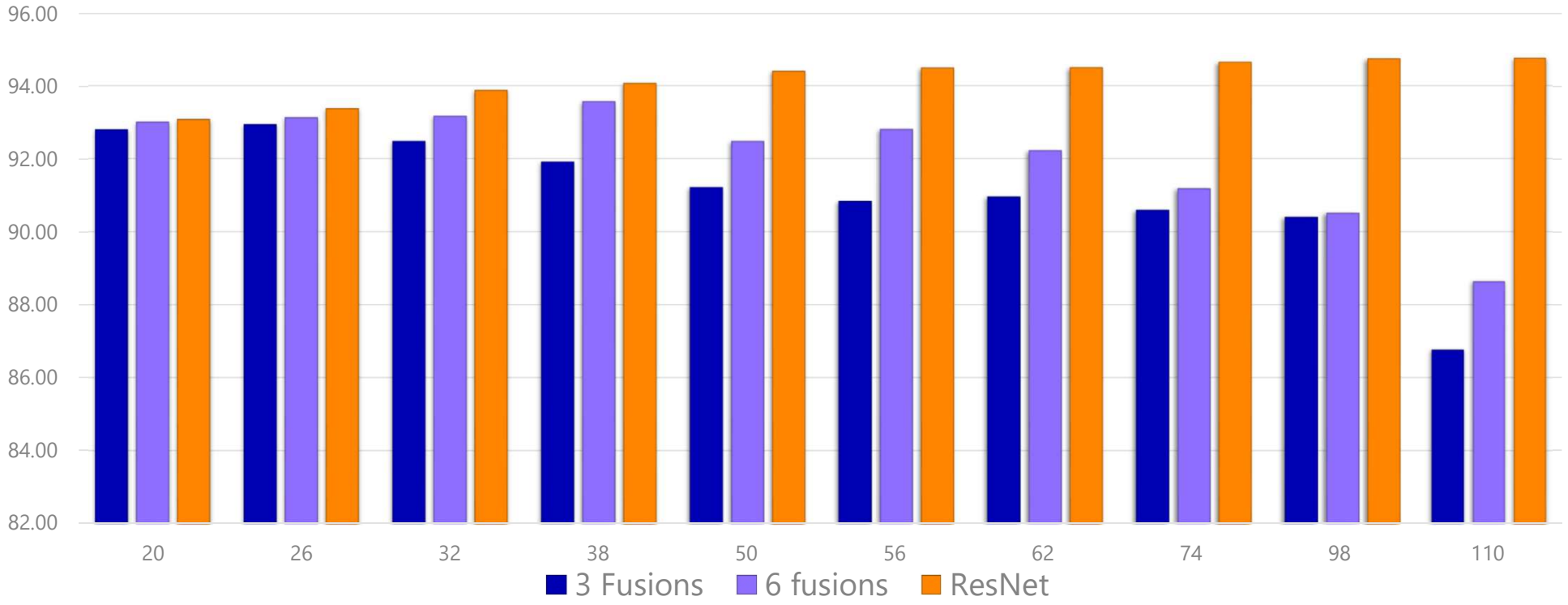


6 fusions

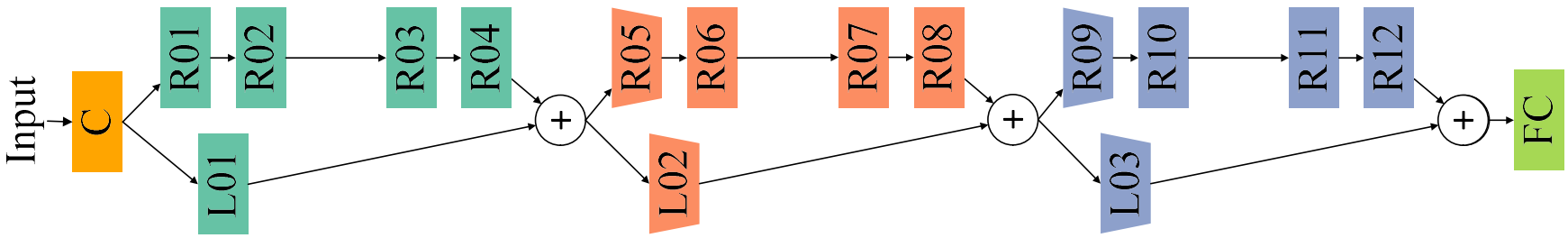


ResNet

Larger ensemble sizes leads to **better performances**

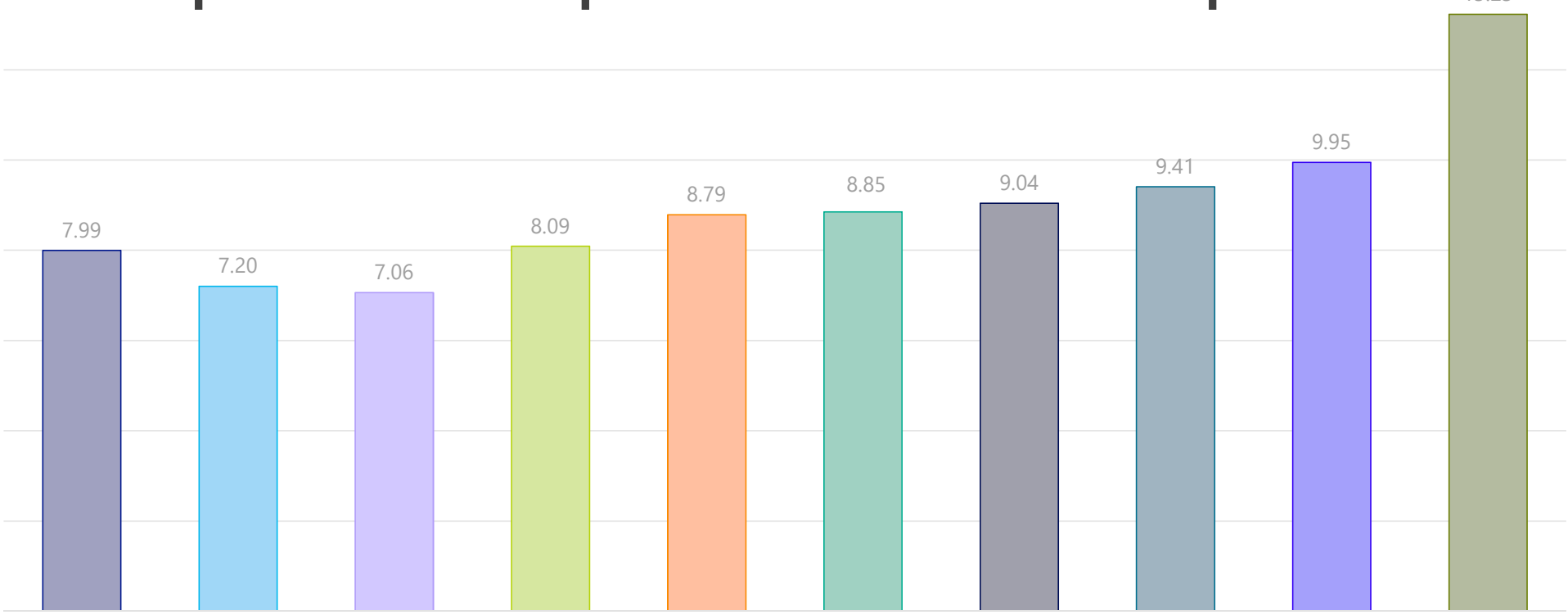


The **capabilities of component networks** affect the **performances**



Deeply-fused nets: Different depths, 3 fusions

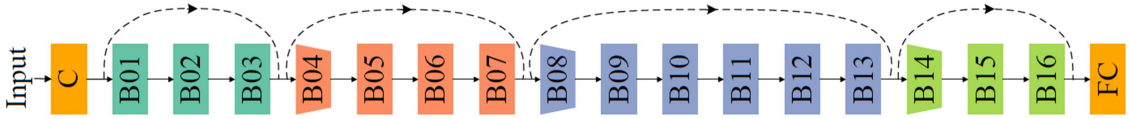
The **capabilities of component networks** affect the **performances**



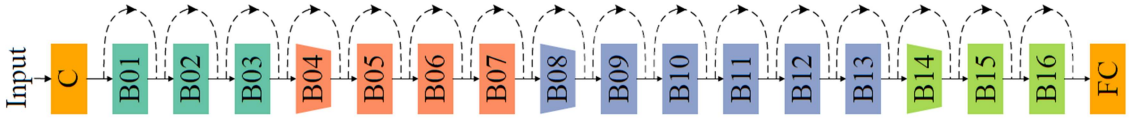
14	20	26	38	50	56	62	74	98	110
5	5	5	5	5	5	5	5	5	5
8	10	12	16	20	22	24	28	36	50
11	15	19	27	35	39	43	51	67	75
14	20	26	38	50	56	62	74	98	110

Empirical study on ImageNet

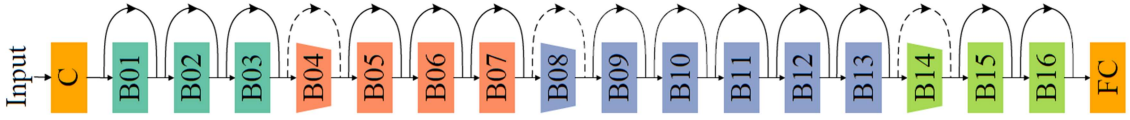
	ResNet	DFN1	DFN2
Top1 validation error	**24.94**	**25.10**	**27.17**
Top5 validation error	**7.46**	**7.85**	**8.98**



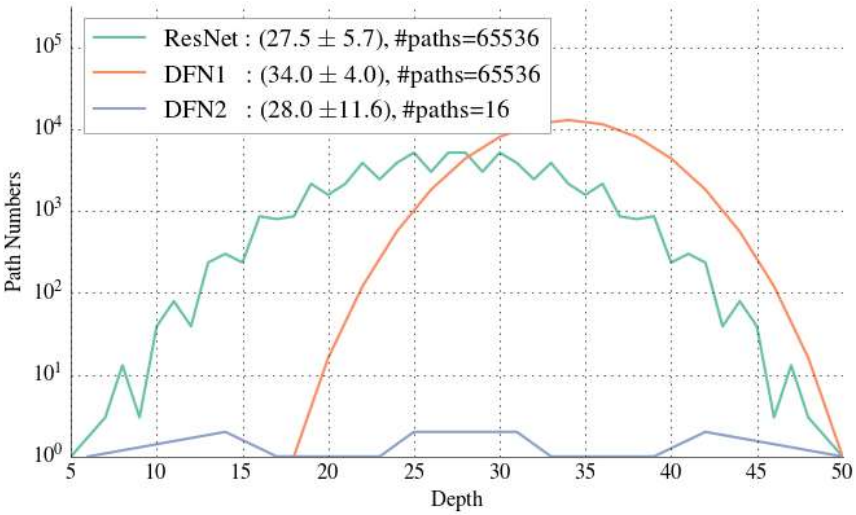
(a) DFN2



(b) DFN1

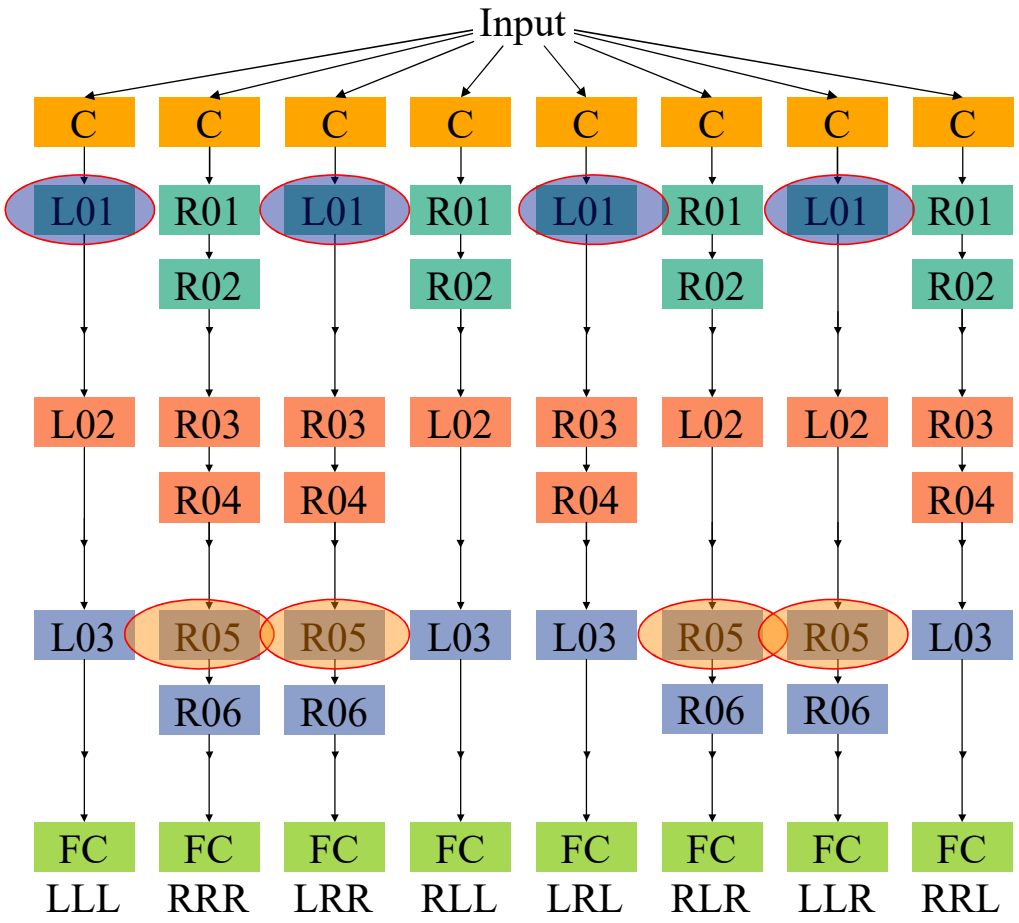


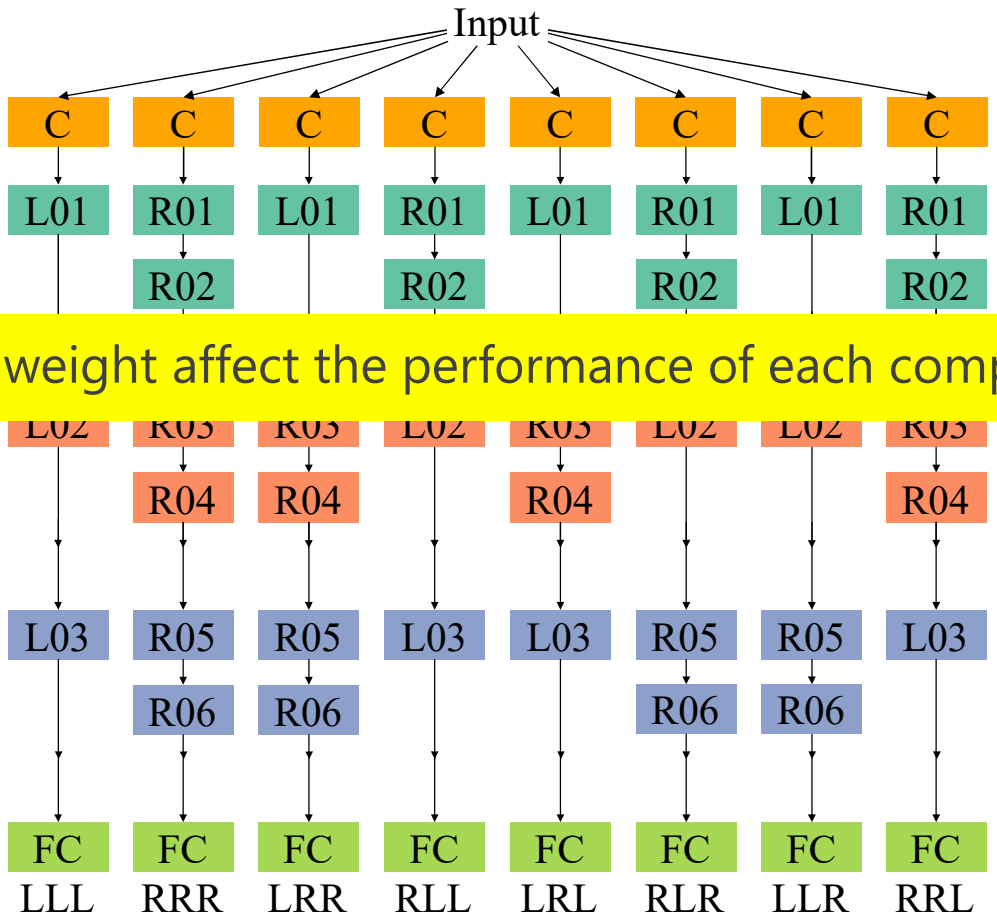
(c) ResNet



What the depth brings?

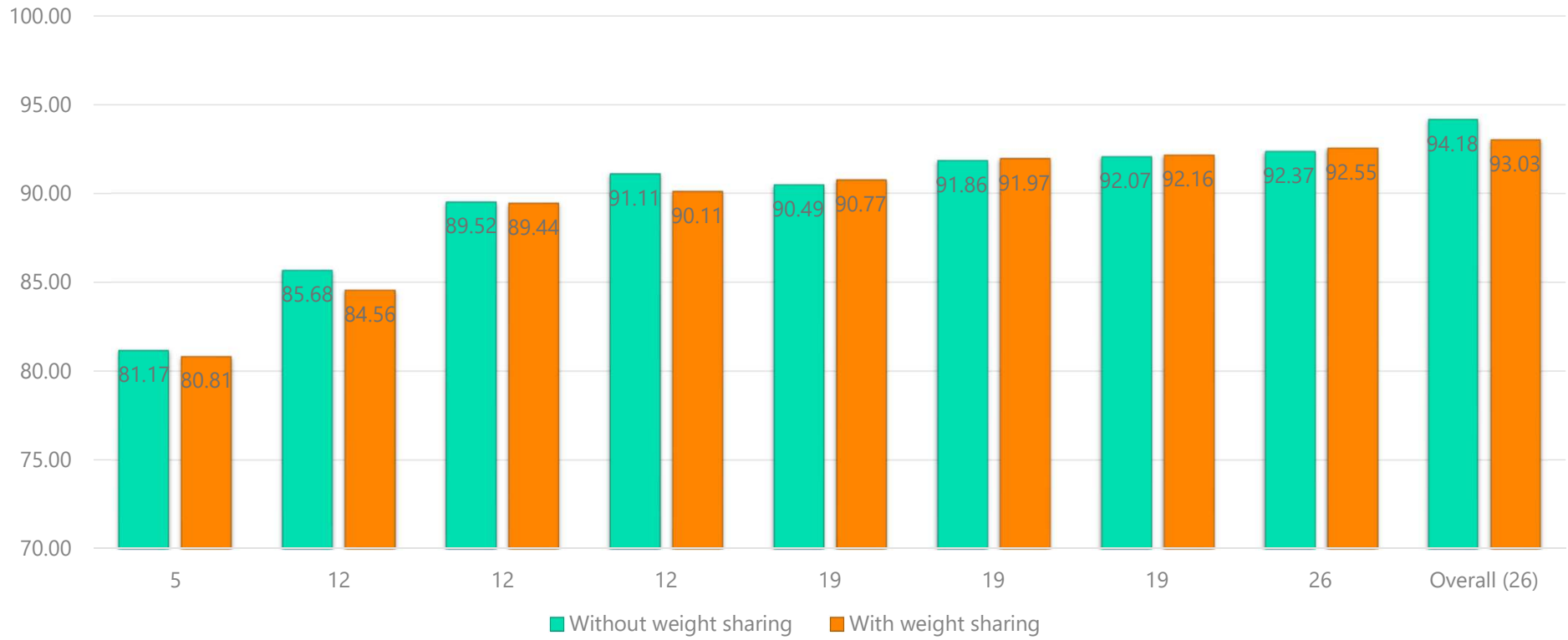
Liming Zhao, Jingdong Wang, Xi Li, Zhuowen Tu, Wenjun Zeng: On the Connection of Deep Fusion to Ensembling. CoRR abs/1611.07718 (2016)


Input
C
L01
R01
R02
L02
R03
R04
L03
R05
R06
FC
LLL
RRR
LRR
RLL
LRL
RLR
LLR
RRL

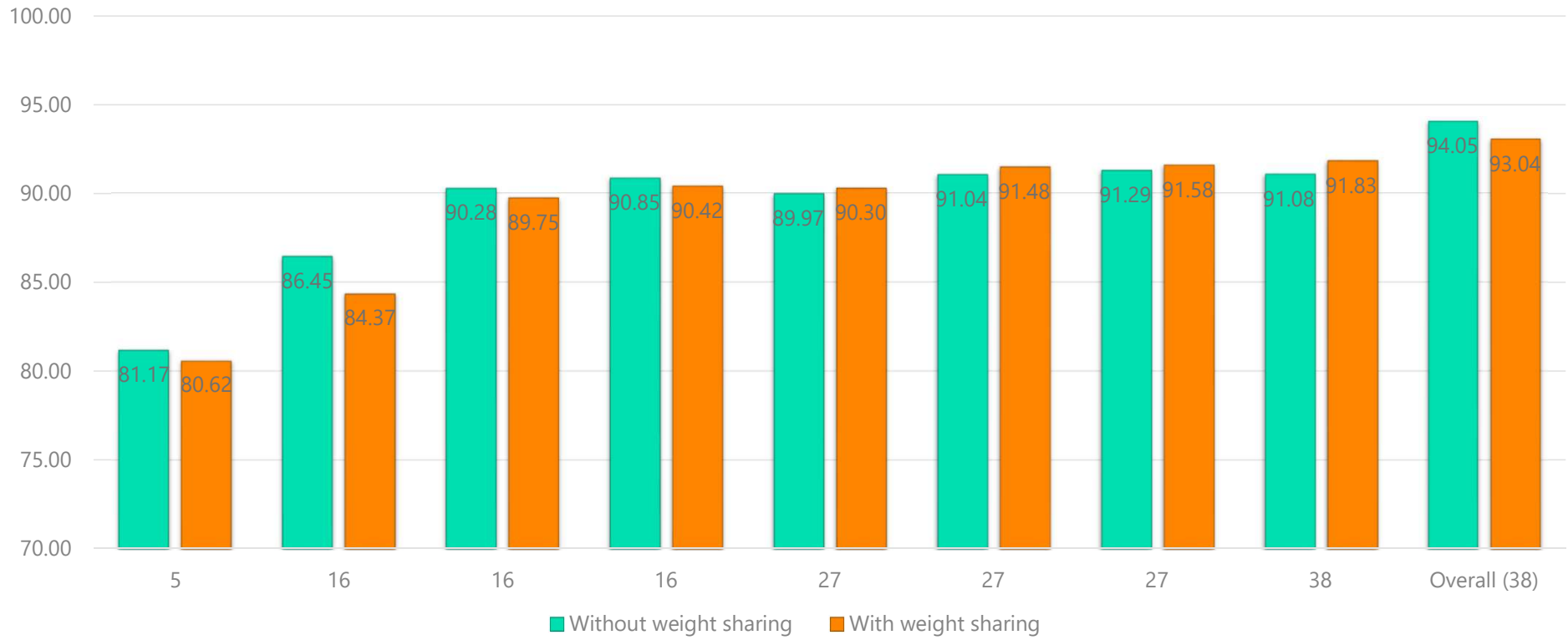


How sharing weight affect the performance of each component?

How sharing weight affect the performance of each component?

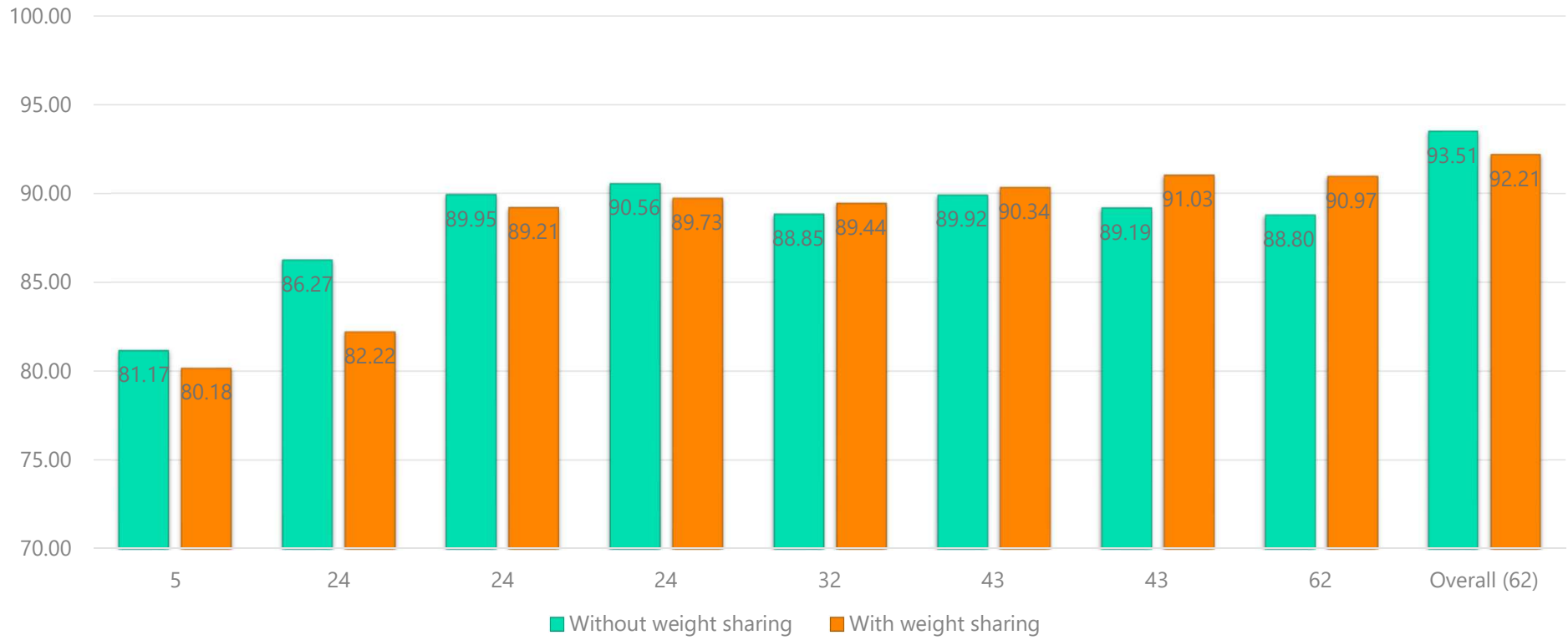


How sharing weight affect the performance of each component?

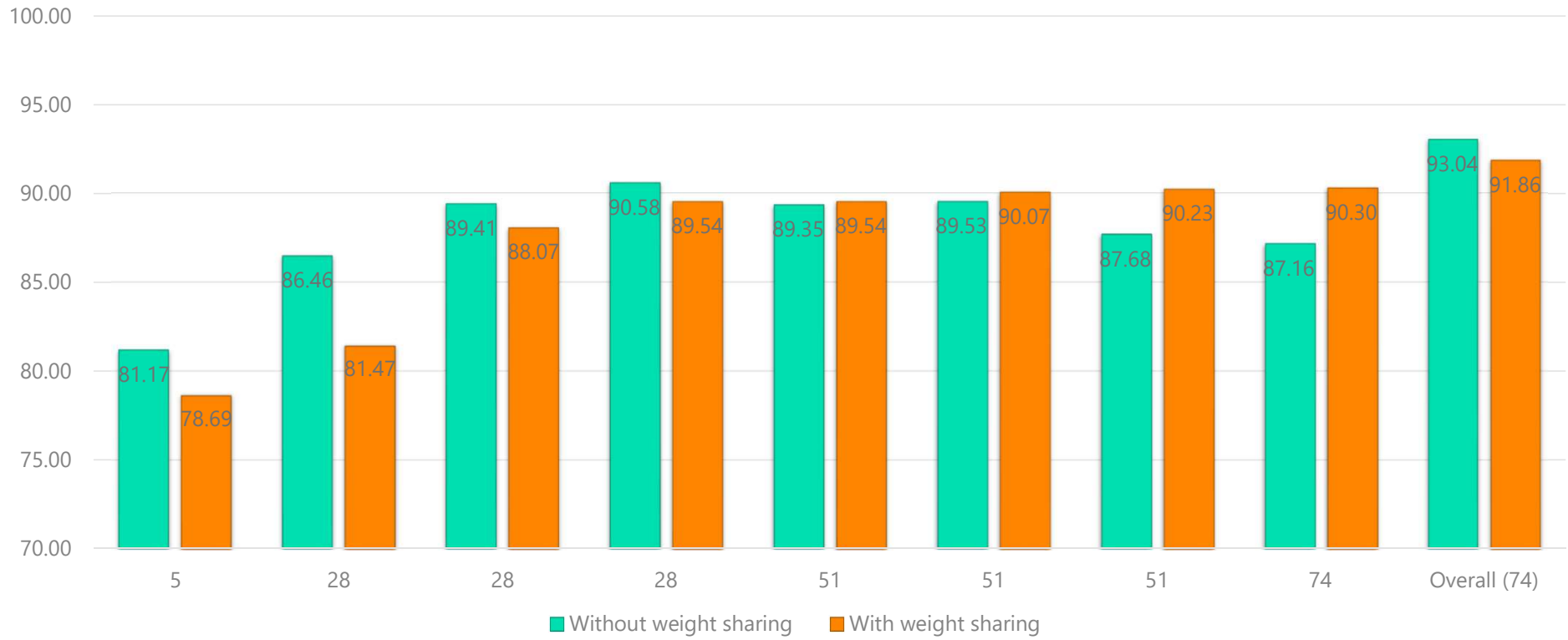


How sharing weight affect the performance of each component?

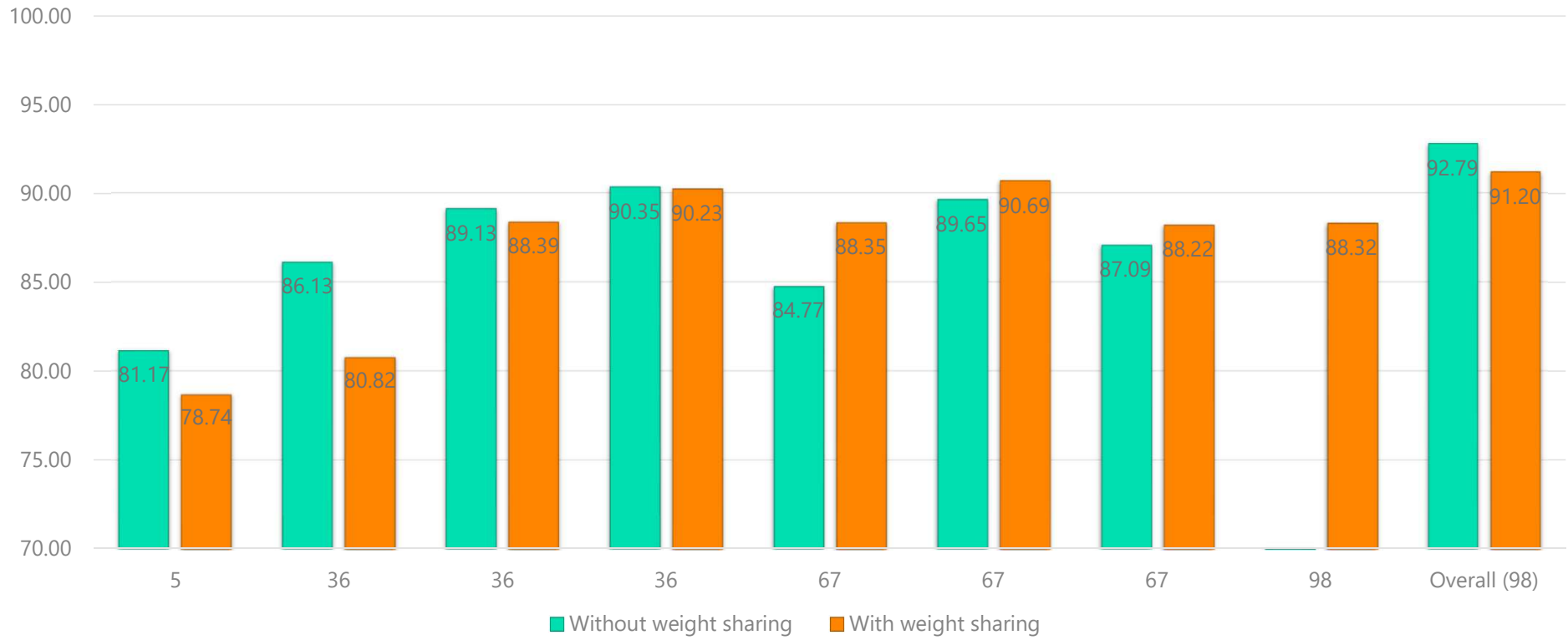
How sharing weight affect the performance of each component?



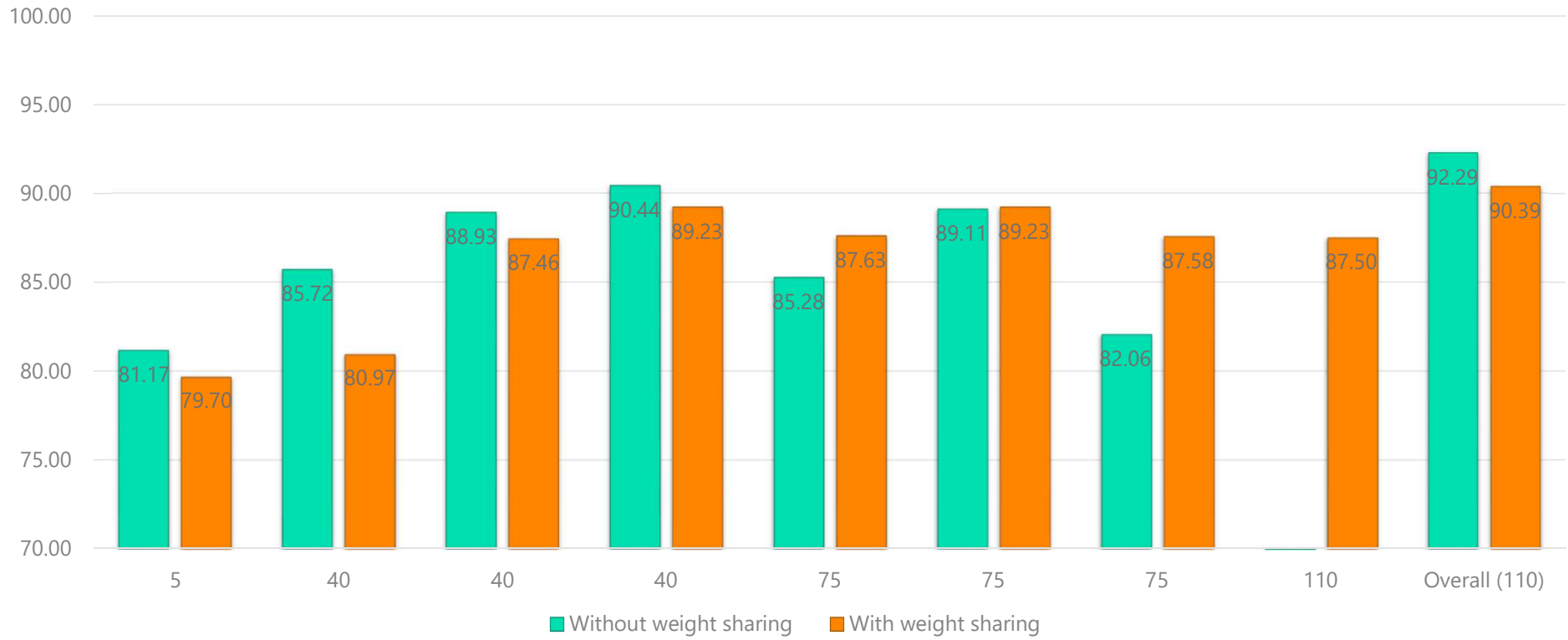
How sharing weight affect the performance of each component?

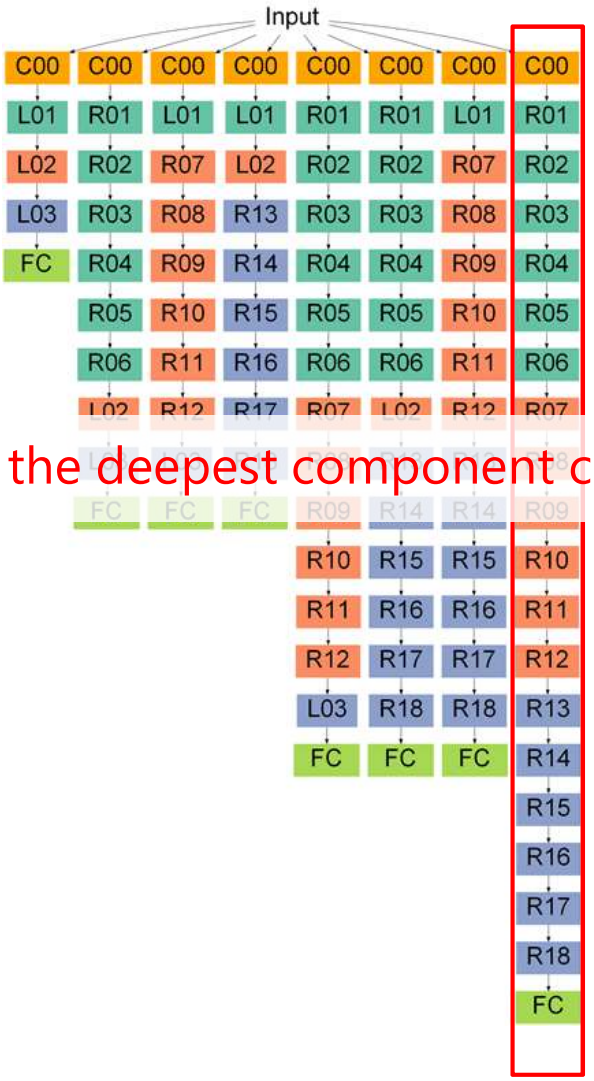


How sharing weight affect the performance of each component?

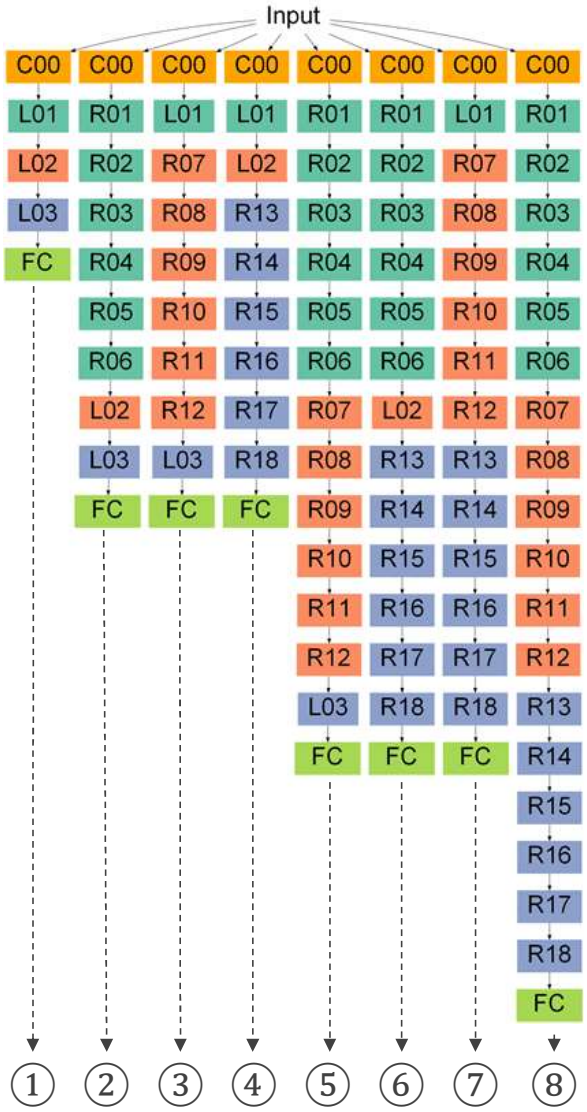


How sharing weight affect the performance of each component?

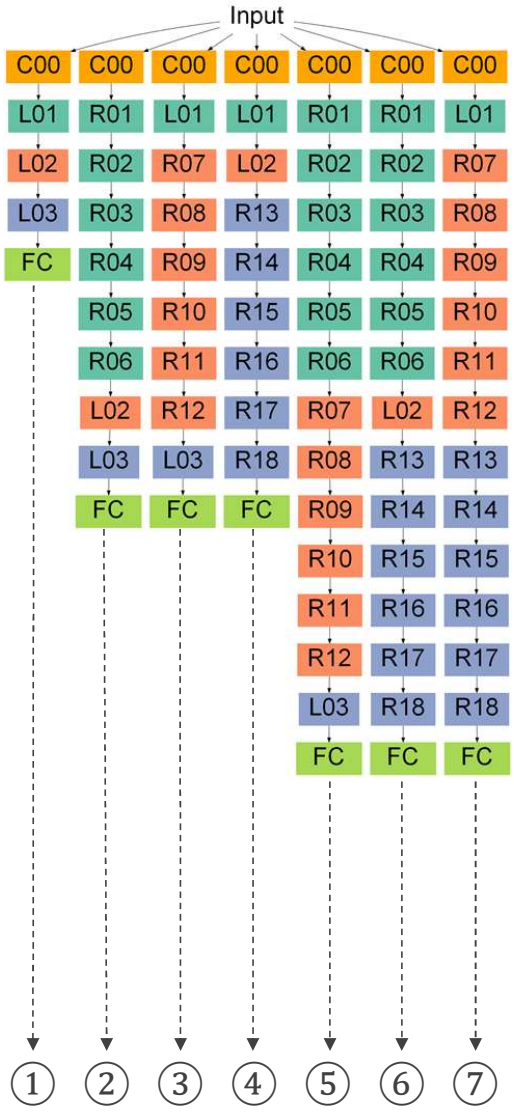




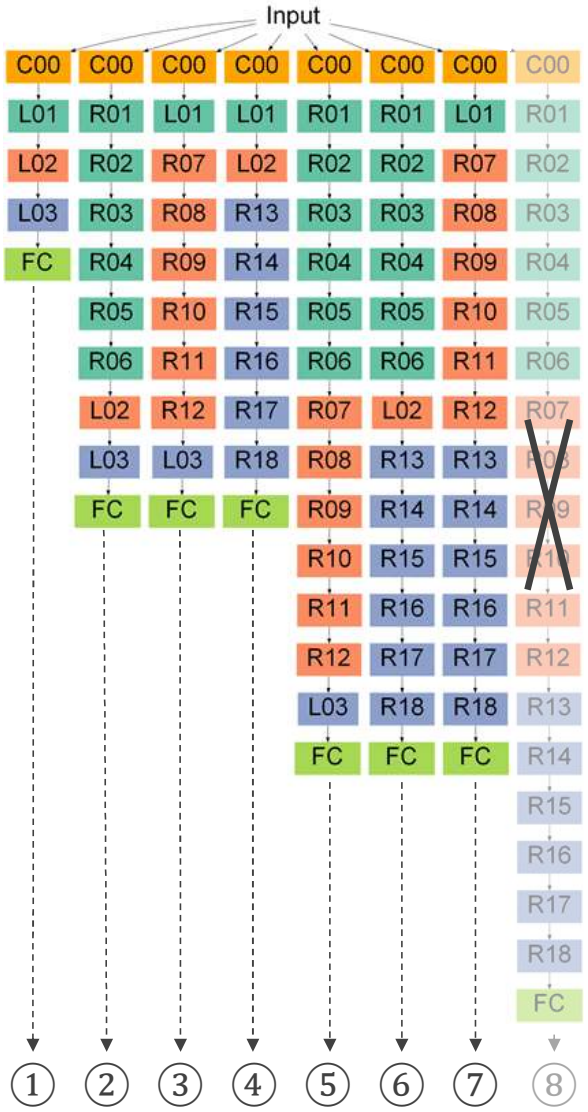
How the deepest component contributes?



Ensemble-8

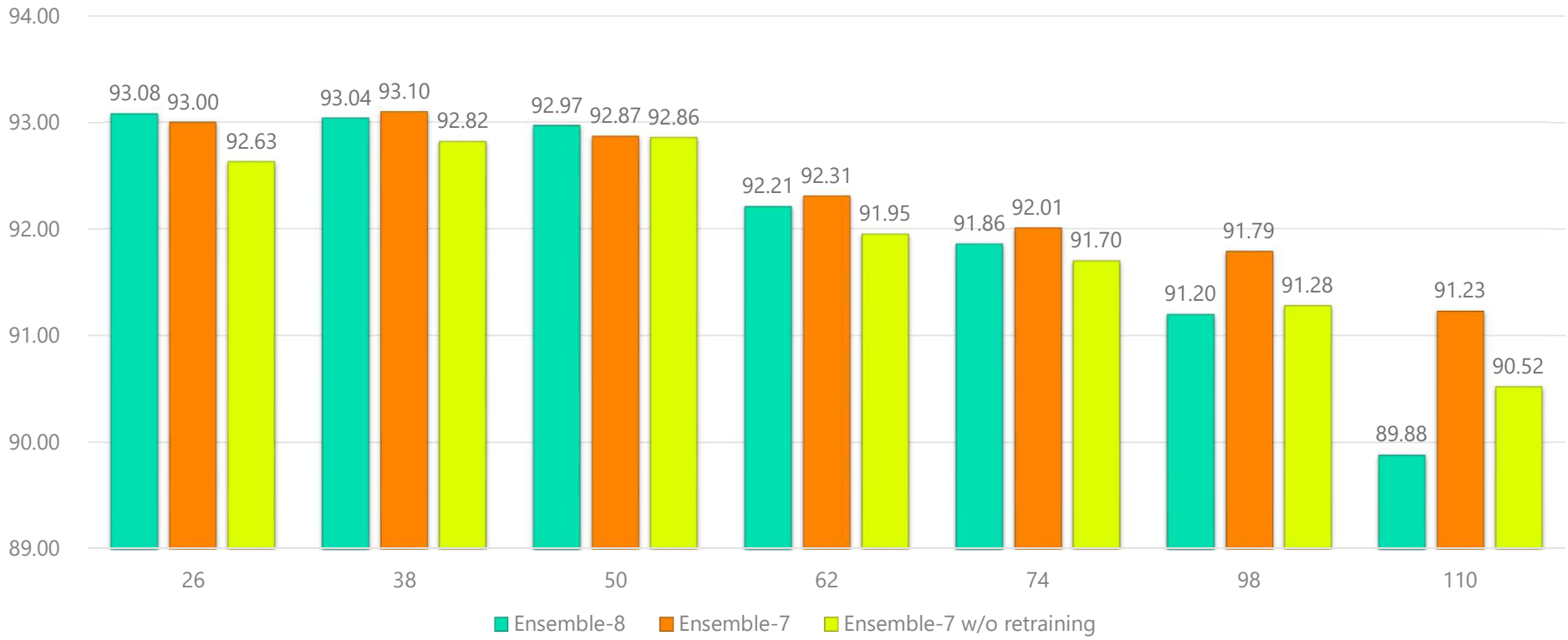


Ensemble-7

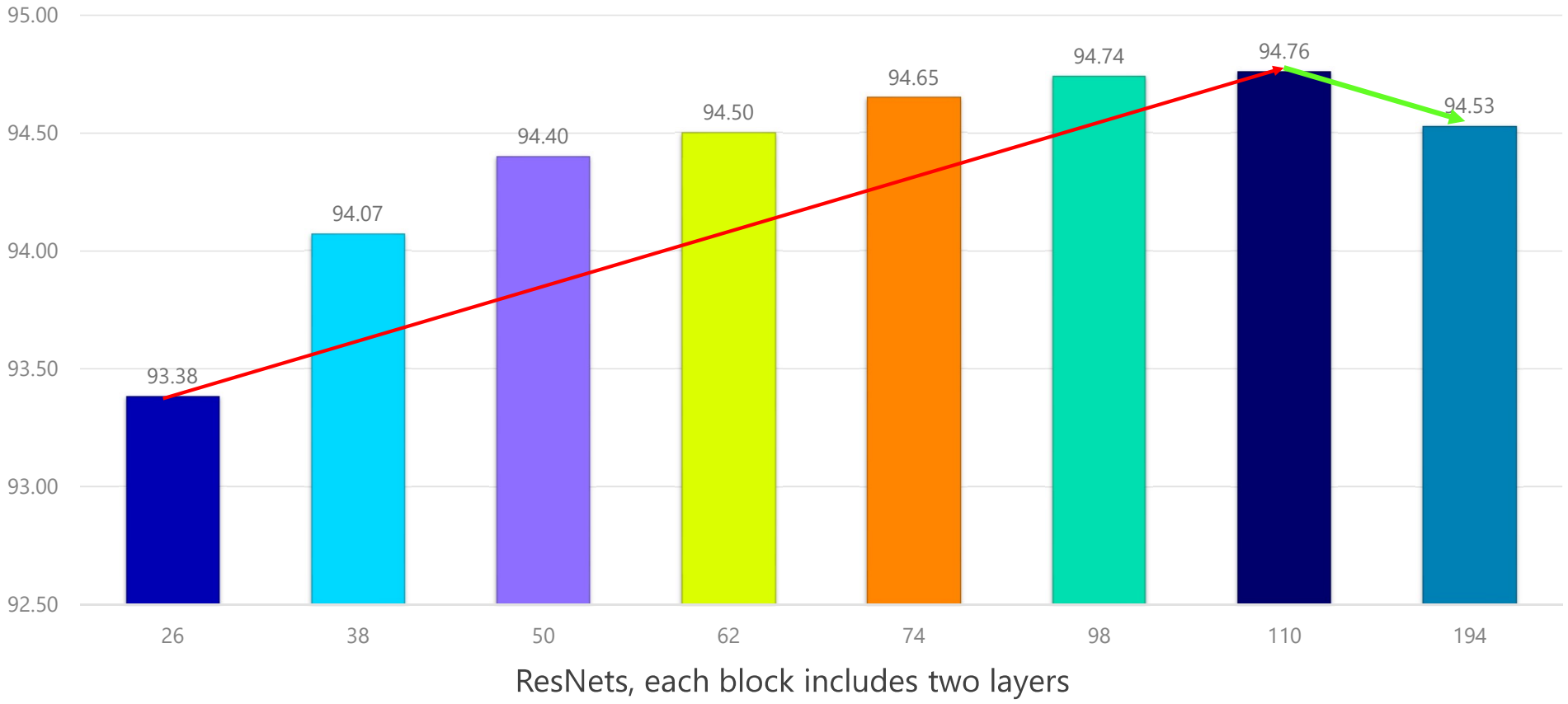


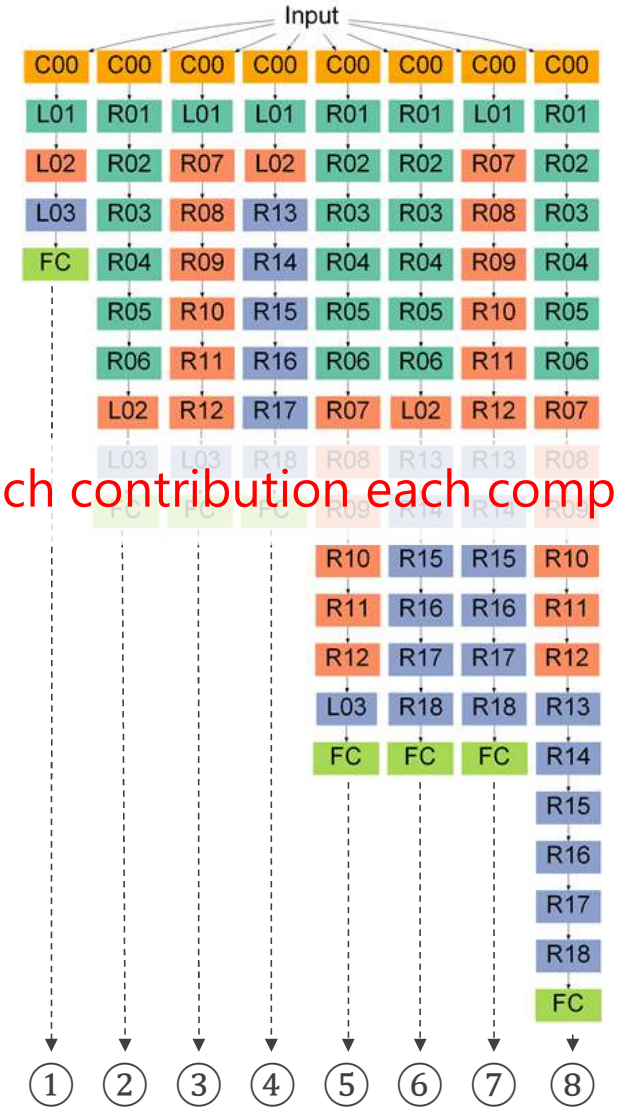
Ensemble-7 w/o retraining

The deepest component **harms** the training

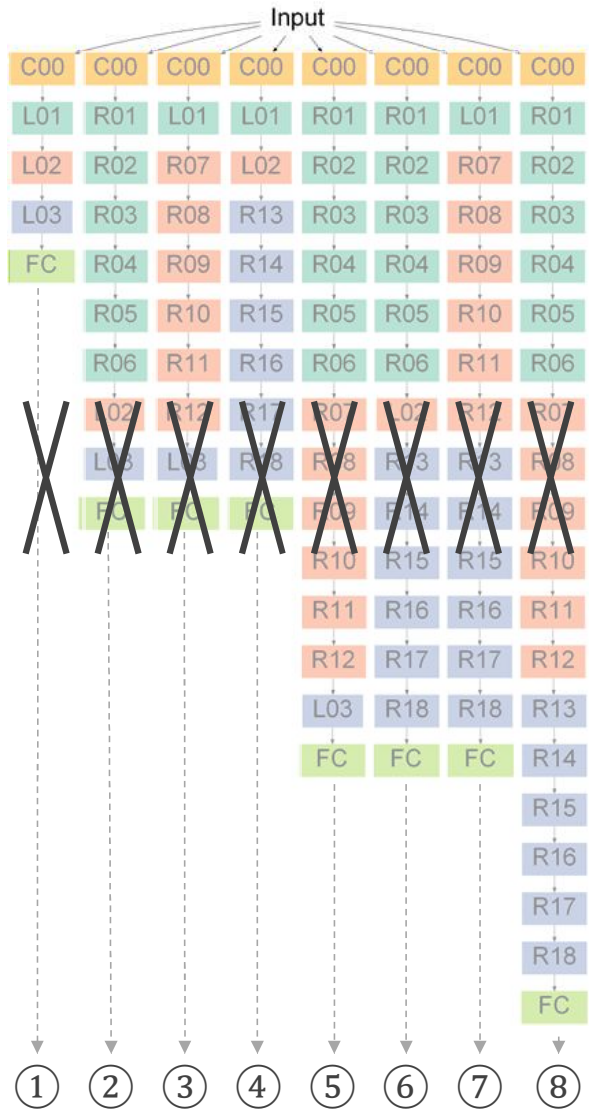


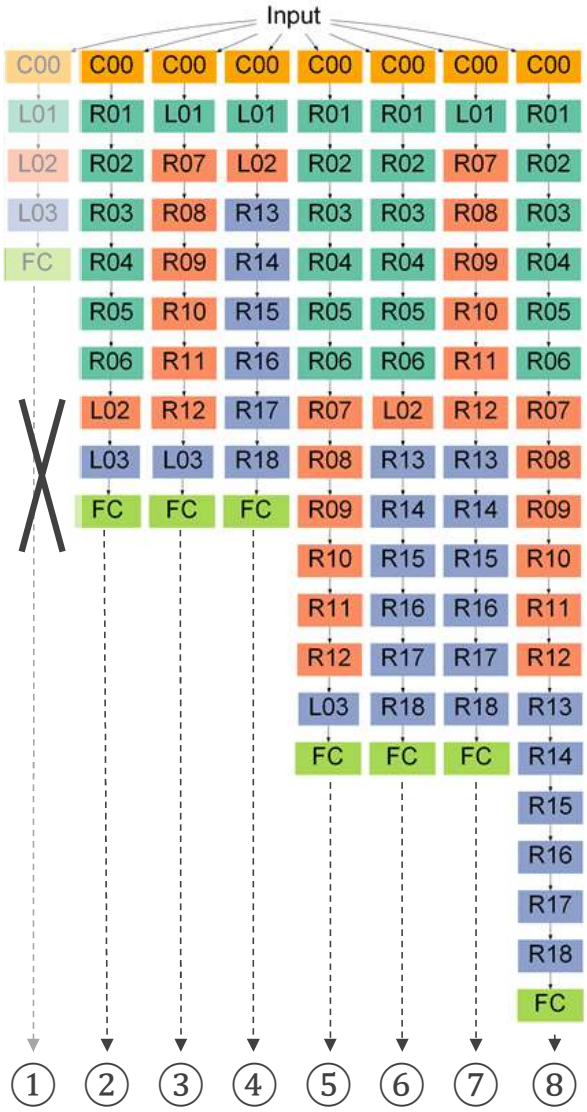
Greater depths leads to **larger** ensemble sizes, but **weaker** components

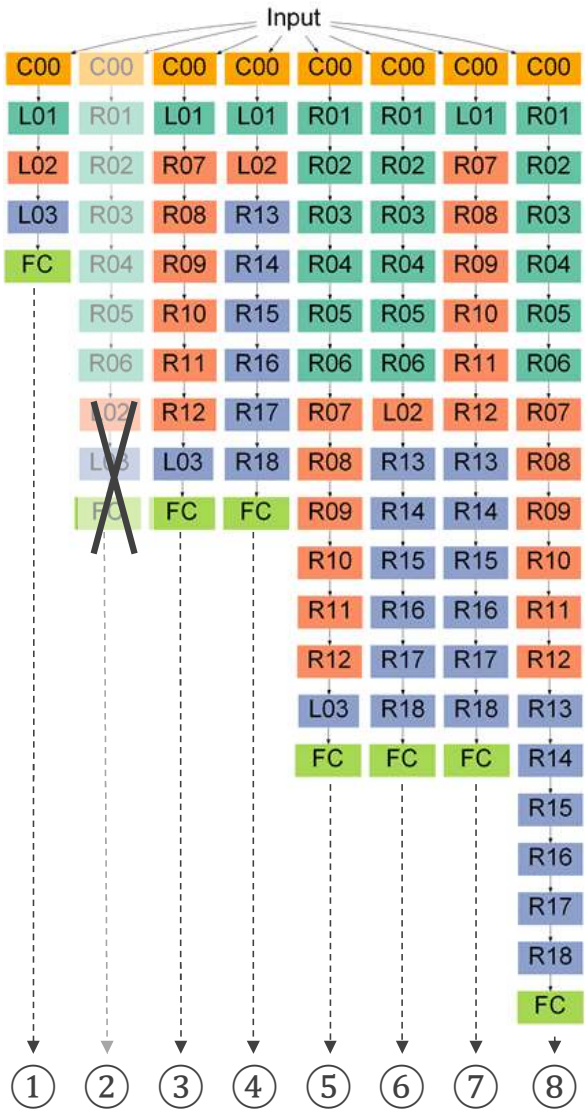


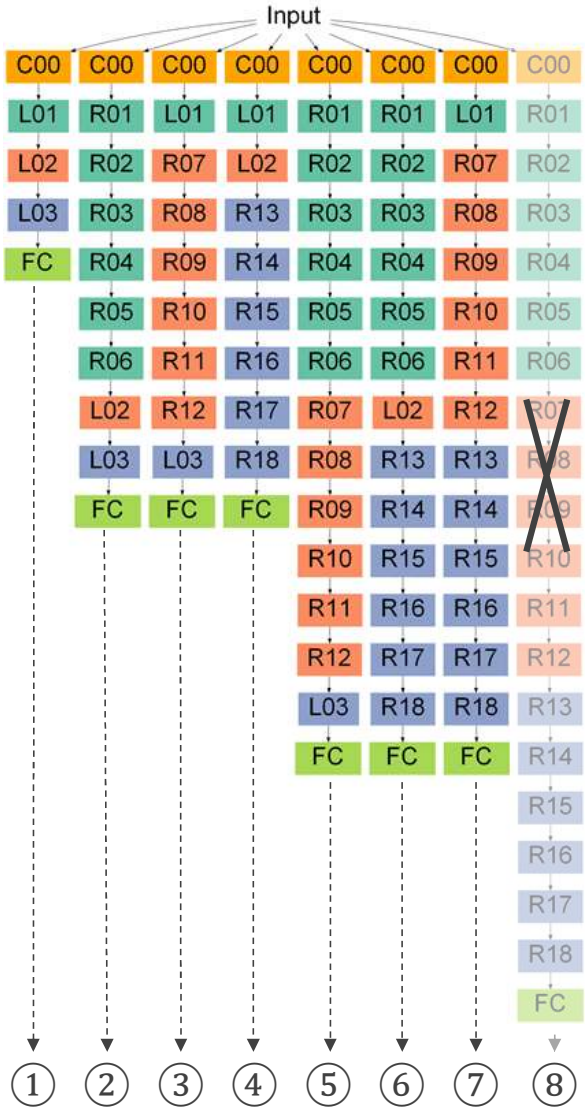


How much contribution each component makes?

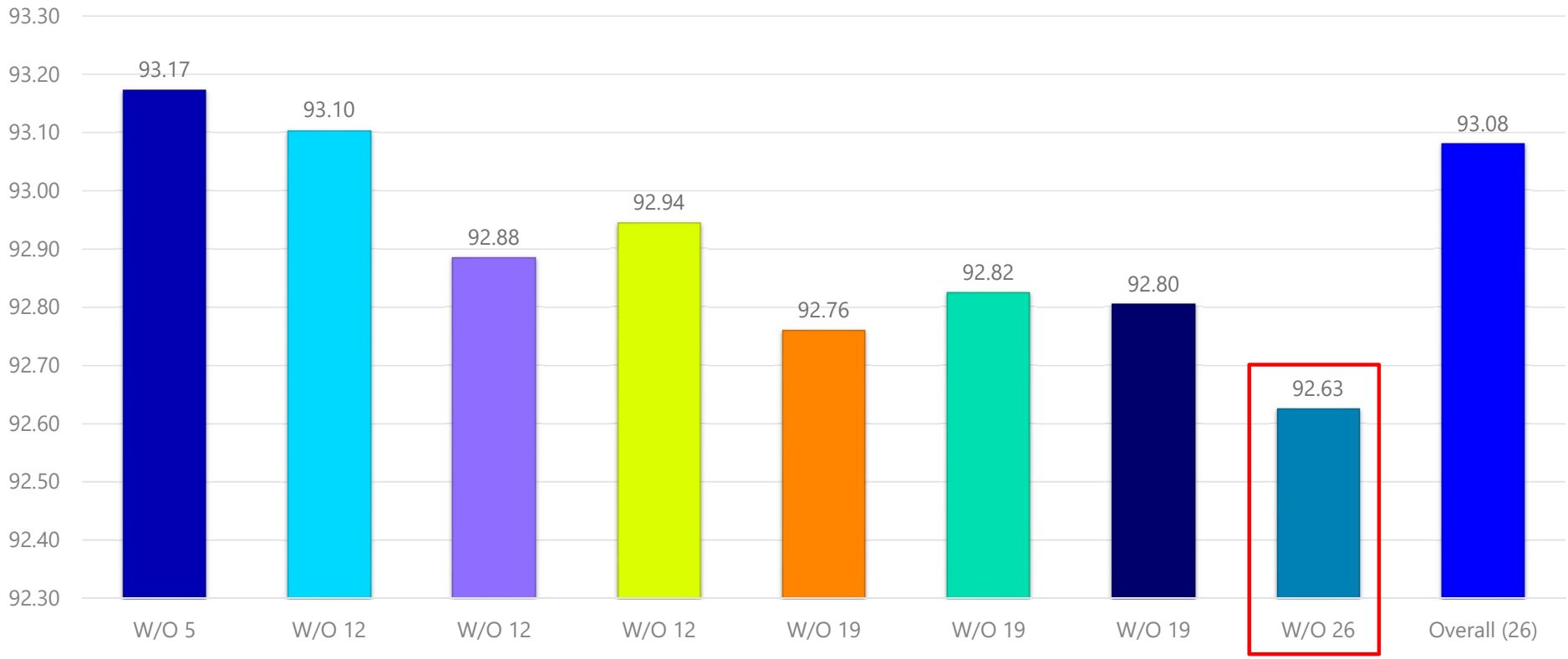

Input
C00
L01
L02
L03
FC
R01
R02
R03
R04
R05
R06
R07
R08
R09
R10
R11
R12
R13
R14
R15
R16
R17
R18
1
2
3
4
5
6
7
8


Input
C00 L01 L02 L03 FC
C00 R01 R02 R03 R04 R05 R06 L02 L03 FC
C00 L01 R07 R08 R09 R10 R11 R12 L03 FC
C00 L01 L02 R13 R14 R15 R16 R17 R18 FC
C00 R01 R02 R03 R04 R05 R06 R07 R08 R09 R10 R11 R12 L03 FC
C00 R01 R02 R03 R04 R05 R06 L02 R13 R14 R15 R16 R17 R18 FC
C00 L01 R07 R08 R09 R10 R11 R12 R13 R14 R15 R16 R17 R18 FC
C00 R01 R02 R03 R04 R05 R06 R07 R08 R09 R10 R11 R12 R13 R14 R15 R16 R17 R18 FC
1 2 3 4 5 6 7 8

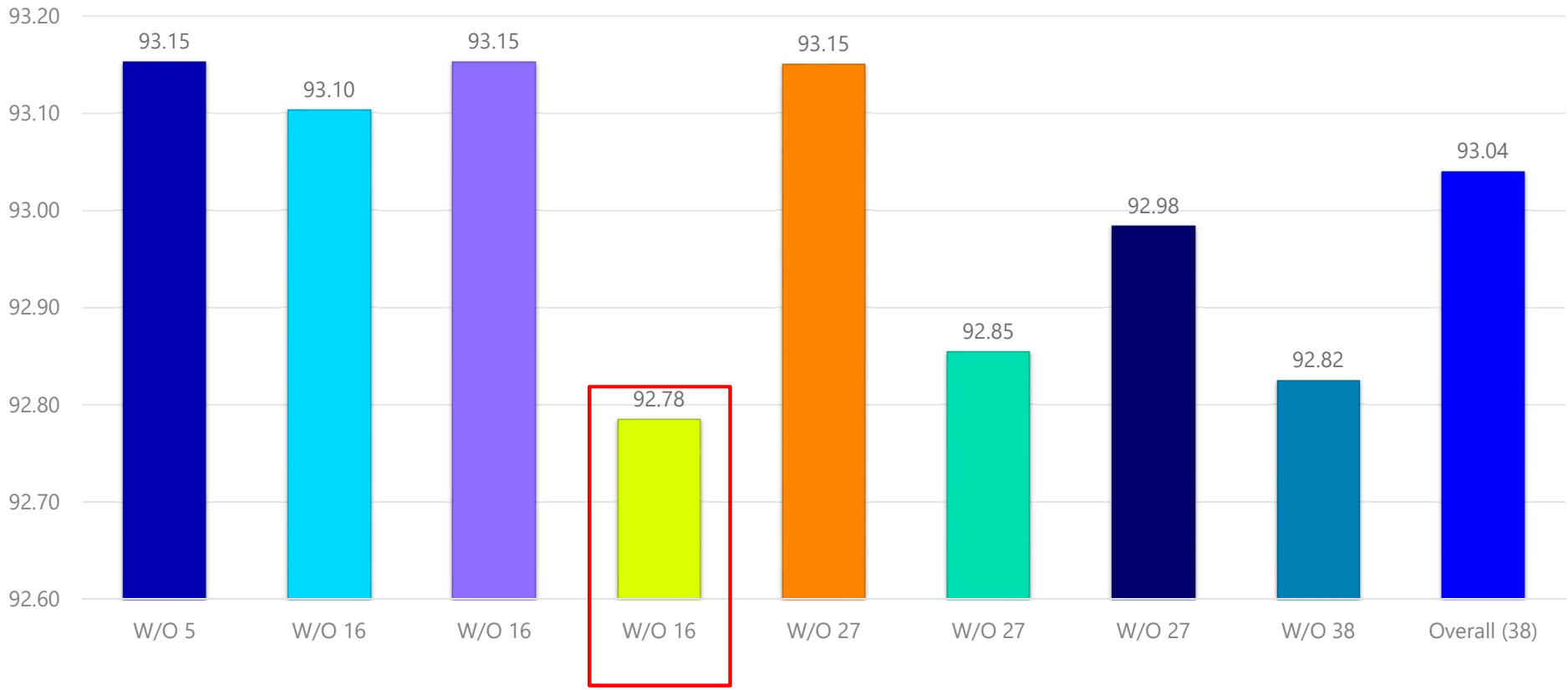

Input
C00 L01 L02 L03 FC
C00 R01 R02 R03 R04 R05 R06
C00 L01 R07 R08 R09 R10 R11 R12 L03 FC
C00 L01 L02 R13 R14 R15 R16 R17 R18 FC
C00 R01 R02 R03 R04 R05 R06 R07 R08 R09 R10 R11 R12 L03 FC
C00 R01 R02 R03 R04 R05 R06 L02 R13 R14 R15 R16 R17 R18 FC
C00 L01 R07 R08 R09 R10 R11 R12 R13 R14 R15 R16 R17 R18 FC
C00 R01 R02 R03 R04 R05 R06 R07 R08 R09 R10 R11 R12 R13 R14 R15 R16 R17 R18 FC
1
2
3
4
5
6
7
8


Input
C00 C00 C00 C00 C00 C00 C00 C00
L01 R01 L01 L01 R01 R01 L01 R01
L02 R02 R07 L02 R02 R02 R07 R02
L03 R03 R08 R13 R03 R03 R08 R03
FC R04 R09 R14 R04 R04 R09 R04
R05 R10 R15 R05 R05 R10 R05
R06 R11 R16 R06 R06 R11 R06
L02 R12 R17 R07 L02 R12
L03 L03 R18 R08 R13 R13
FC FC FC R09 R14 R14
R10 R15 R15
R11 R16 R16
R12 R17 R17
L03 R18 R18
FC FC FC
R10
R11
R12
R13
R14
R15
R16
R17
R18
FC
1 2 3 4 5 6 7 8

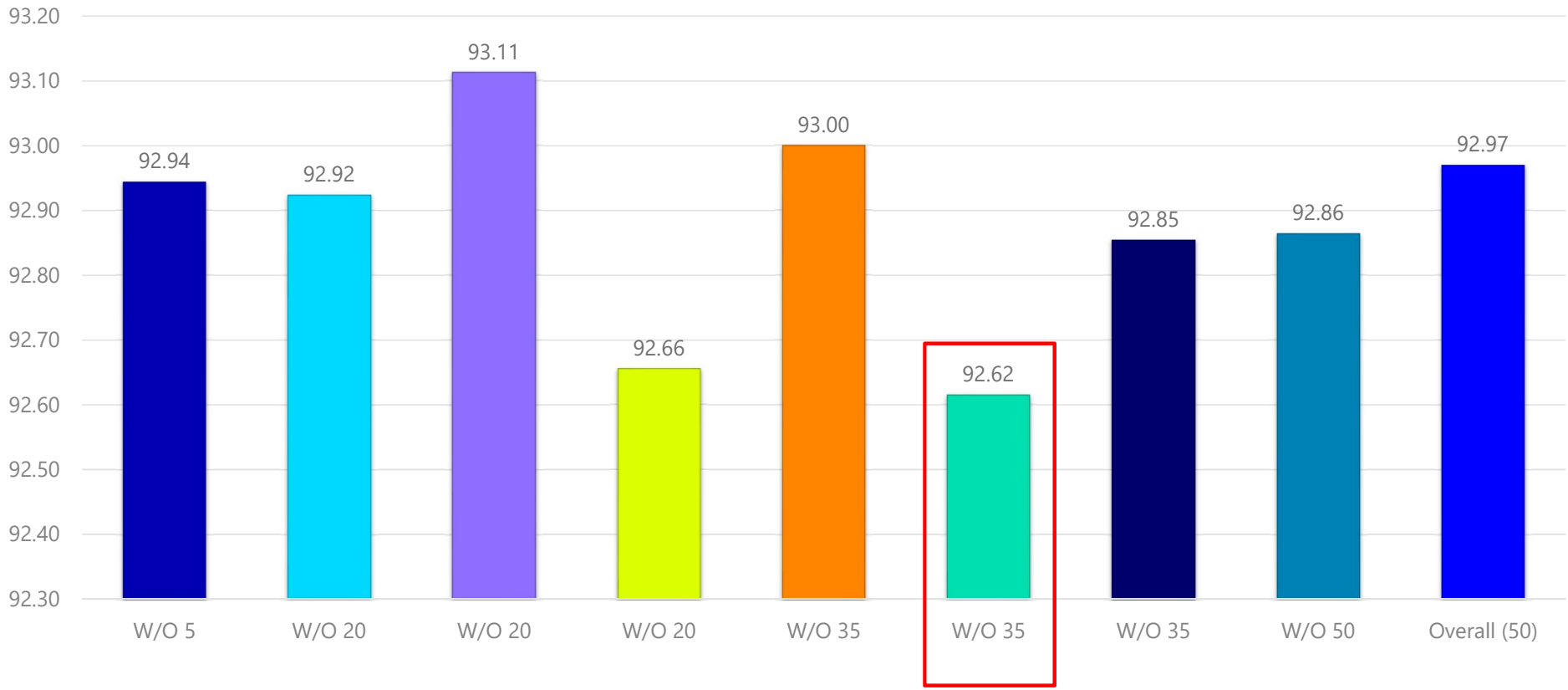
How much contribution each component makes?



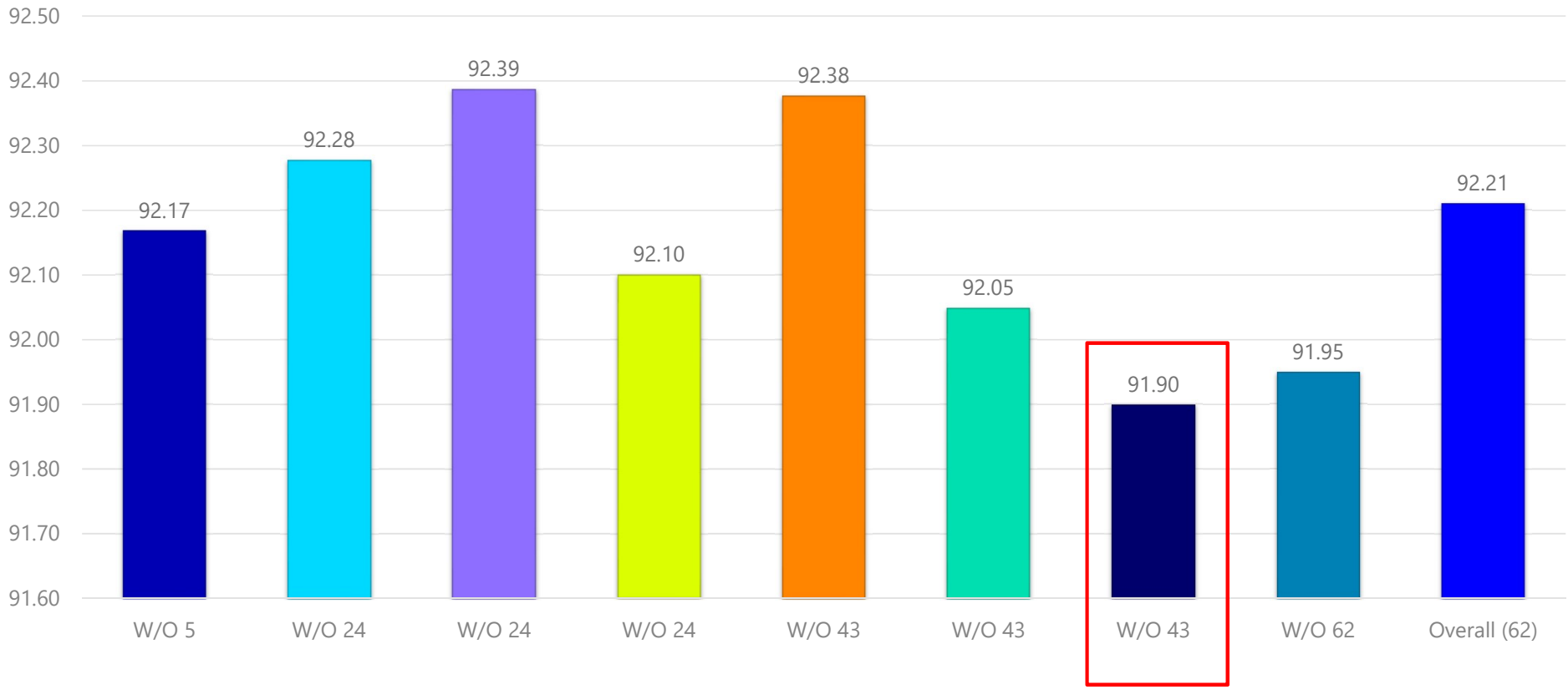
How much contribution each component makes?



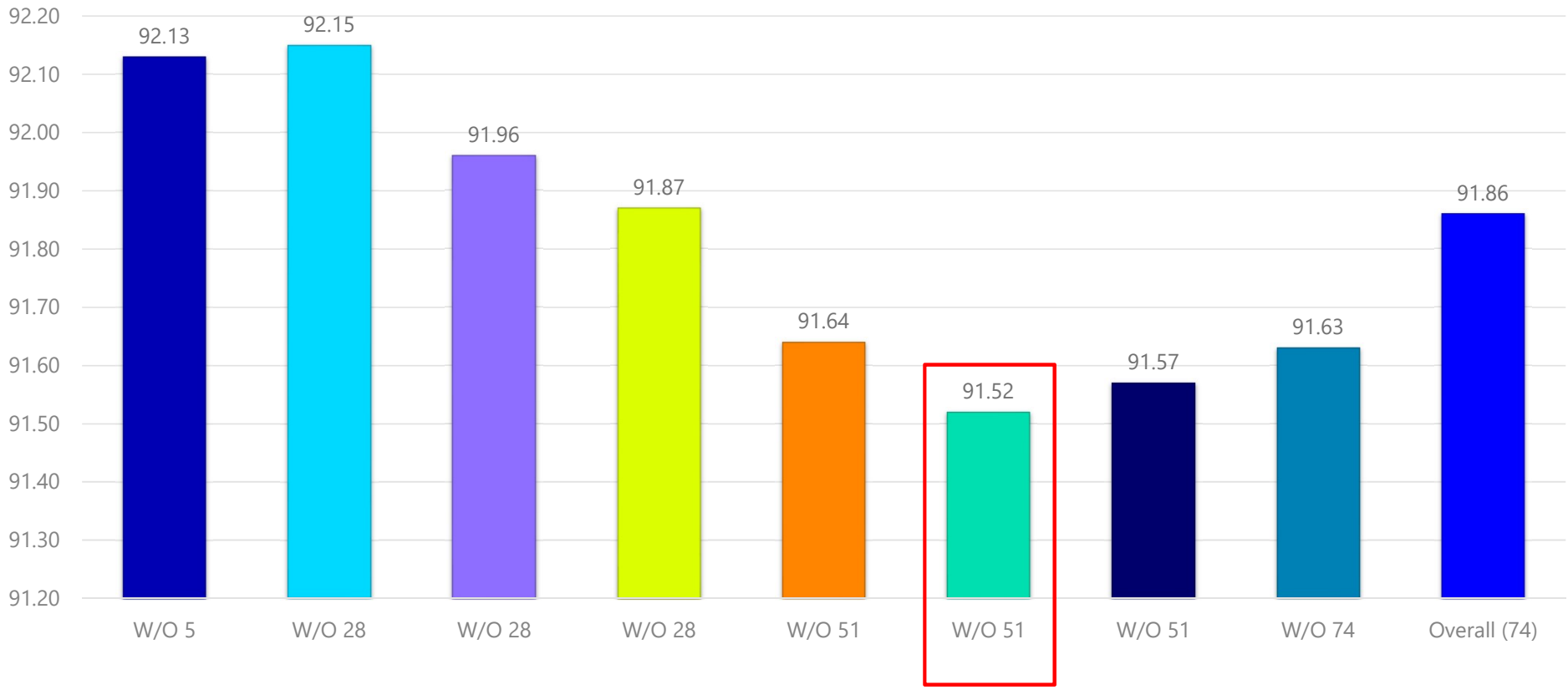
How much contribution each component makes?



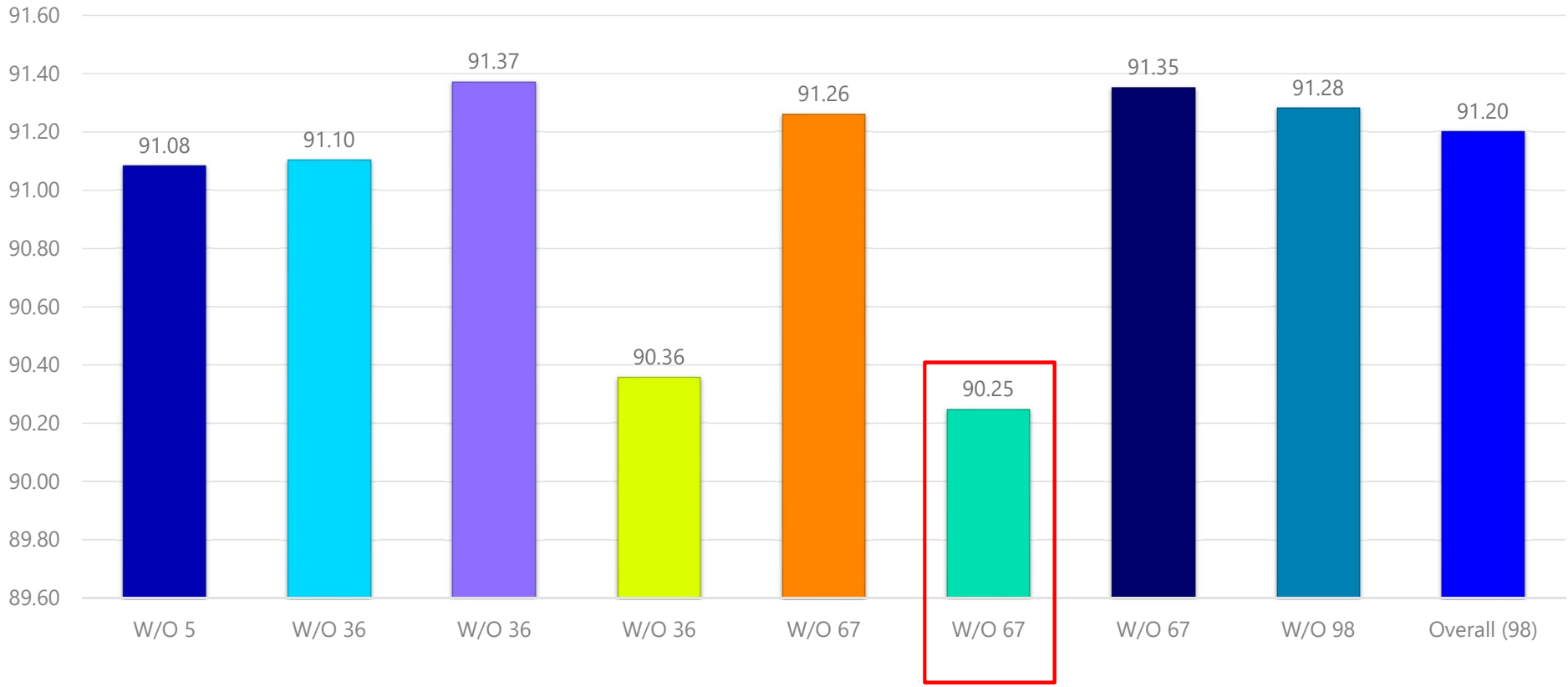
How much contribution each component makes?



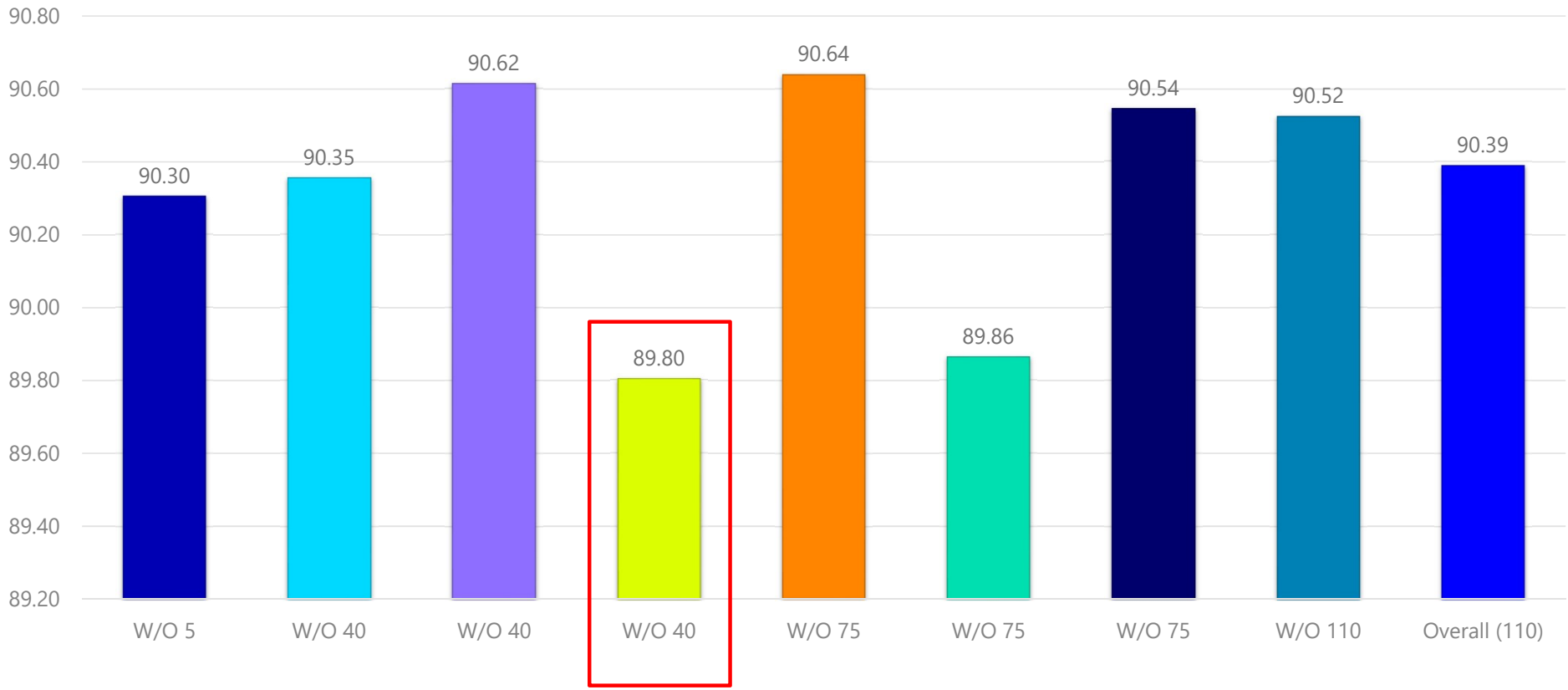
How much contribution each component makes?



How much contribution each component makes?



How much contribution each component makes?

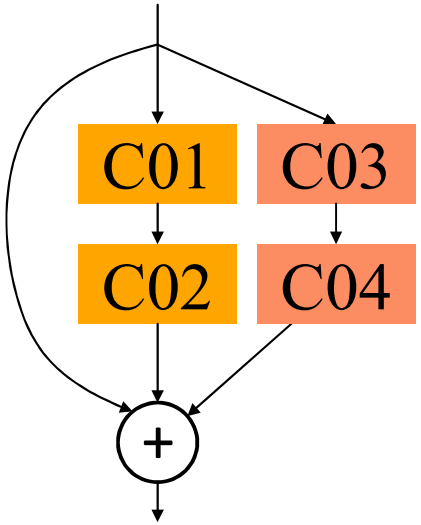


Our approach: Merge-and-Run

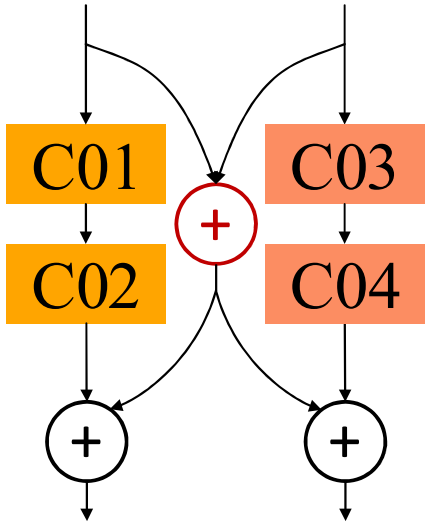
Liming Zhao, Jingdong Wang, Xi Li, Zhuowen Tu, Wenjun Zeng: On the Connection of Deep Fusion to Ensembling. CoRR abs/1611.07718 (2016)

Design rules:
(1) Large ensemble size; (2) Avoid too deep networks

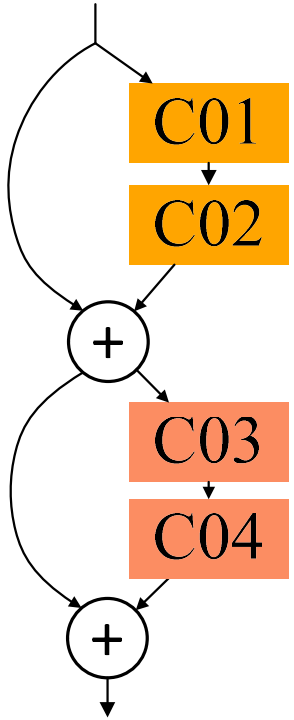
Design rules:
(1) Large ensemble size; (2) Avoid too deep networks



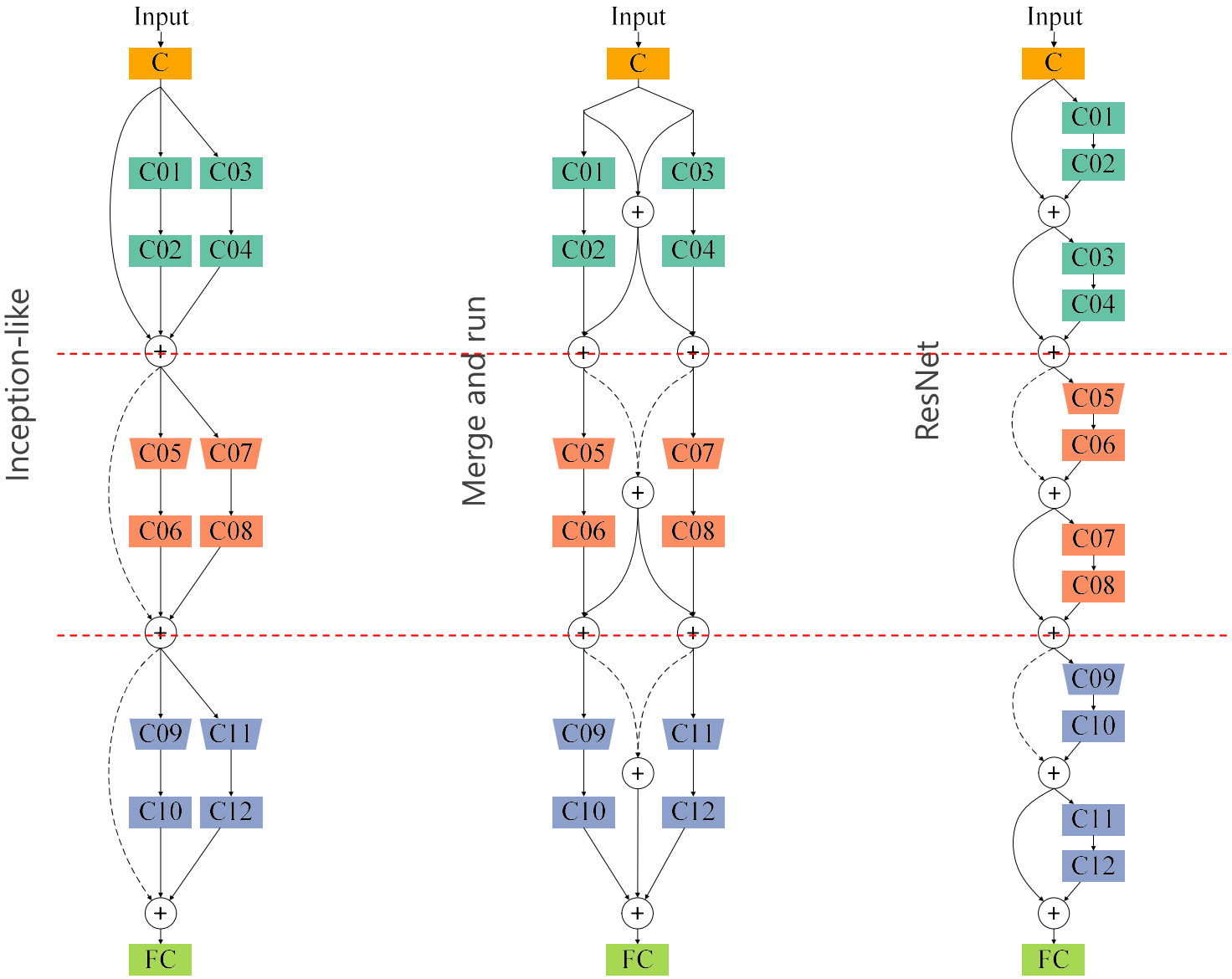
Inception-like
3 paths

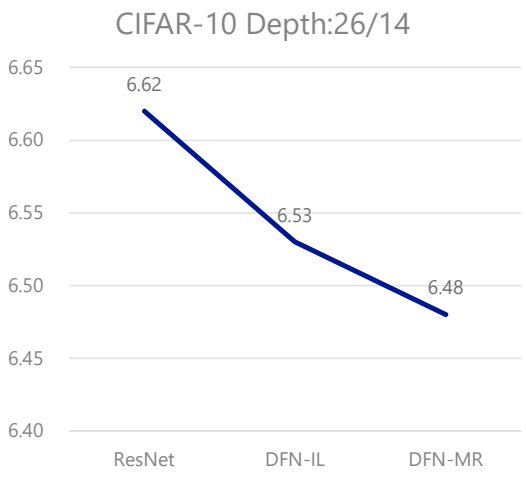


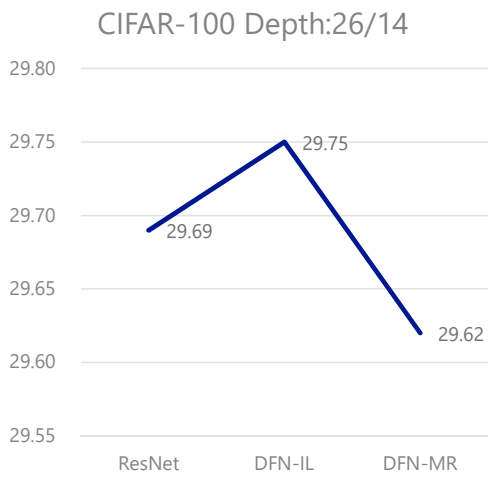
Merge and run
6 paths

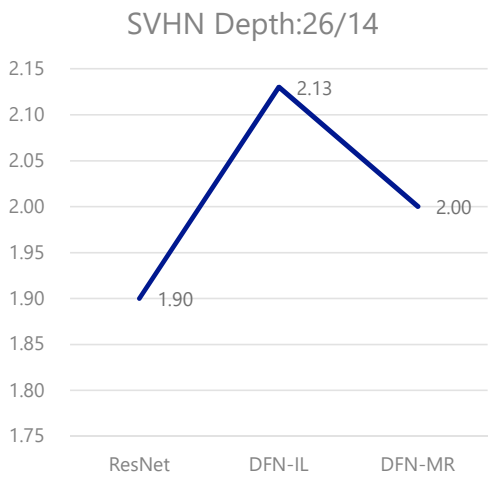


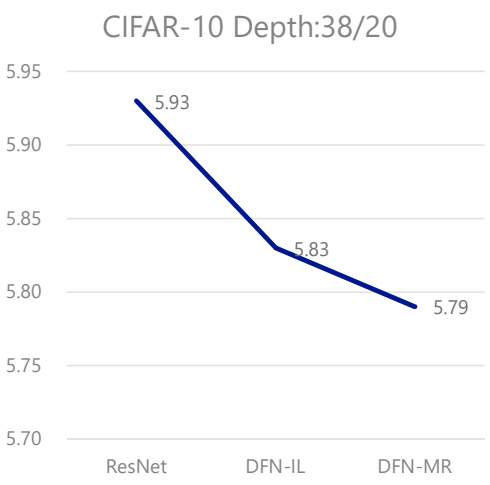
ResNet
4 paths

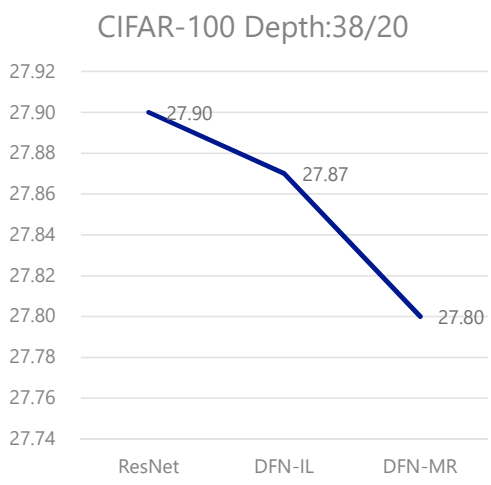

Inception-like
Merge and run
ResNet
Input
C
C01
C02
C03
C04
C05
C06
C07
C08
C09
C10
C11
C12
FC

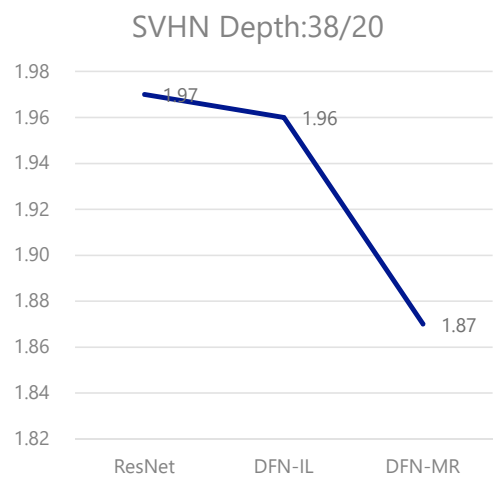

CIFAR-10 Depth:26/14
6.65
6.60
6.55
6.50
6.45
6.40
6.62
6.53
6.48
ResNet
DFN-IL
DFN-MR

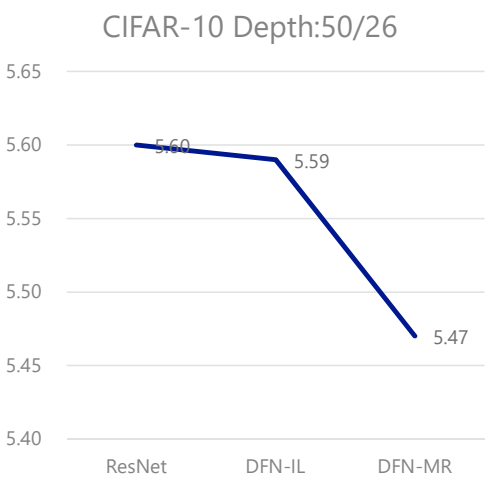

CIFAR-100 Depth:26/14
29.80
29.75
29.70
29.65
29.60
29.55
29.69
29.75
29.62
ResNet
DFN-IL
DFN-MR

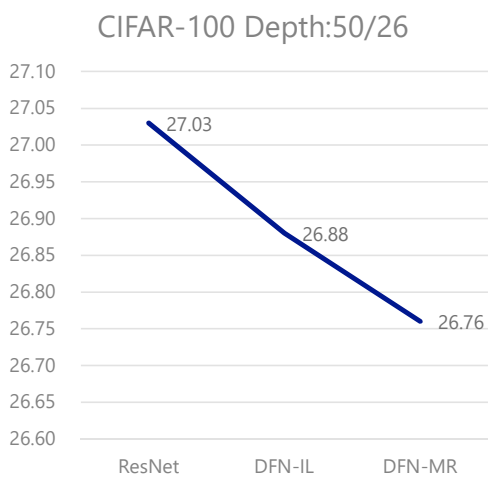

SVHN Depth:26/14
2.15
2.10
2.05
2.00
1.95
1.90
1.85
1.80
1.75
1.90
2.13
2.00
ResNet
DFN-IL
DFN-MR

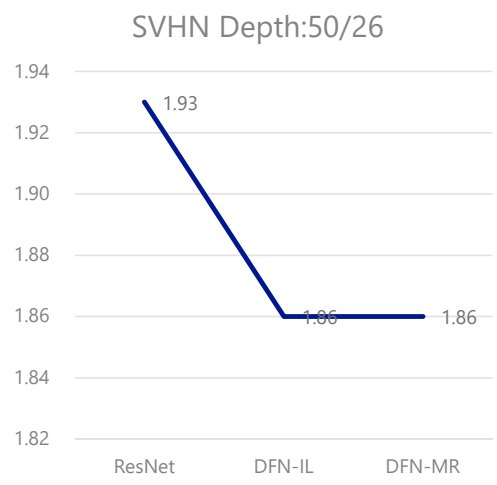

CIFAR-10 Depth:38/20
5.95
5.90
5.85
5.80
5.75
5.70
5.93
5.83
5.79
ResNet
DFN-IL
DFN-MR

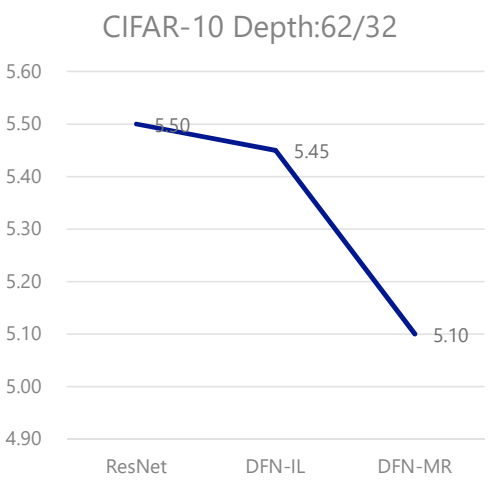

CIFAR-100 Depth:38/20
27.92
27.90
27.88
27.86
27.84
27.82
27.80
27.78
27.76
27.74
27.90
27.87
27.80
ResNet
DFN-IL
DFN-MR

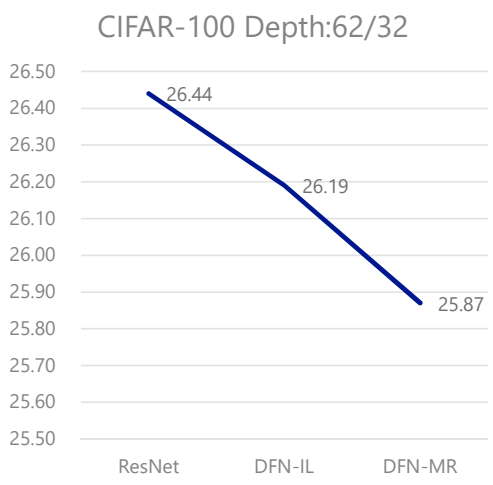

SVHN Depth:38/20
1.98
1.96
1.94
1.92
1.90
1.88
1.86
1.84
1.82
1.97
1.96
1.87
ResNet
DFN-IL
DFN-MR

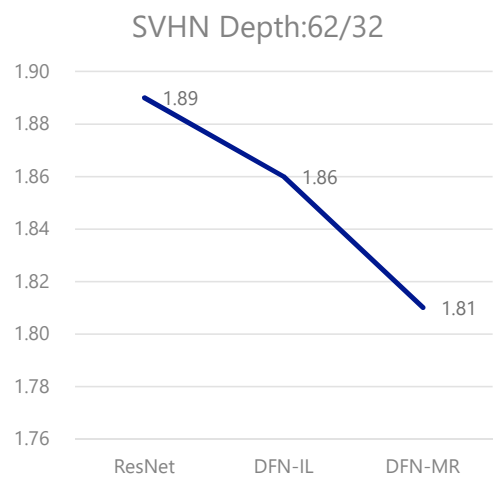

CIFAR-10 Depth:50/26
5.65
5.60
5.55
5.50
5.45
5.40
5.60
5.59
5.47
ResNet
DFN-IL
DFN-MR

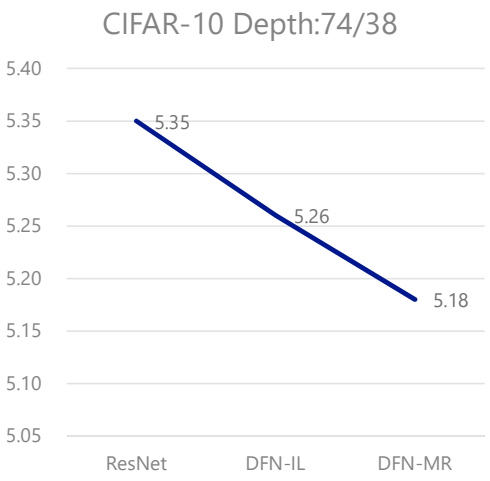

CIFAR-100 Depth:50/26
27.10
27.05
27.00
26.95
26.90
26.85
26.80
26.75
26.70
26.65
26.60
27.03
26.88
26.76
ResNet
DFN-IL
DFN-MR

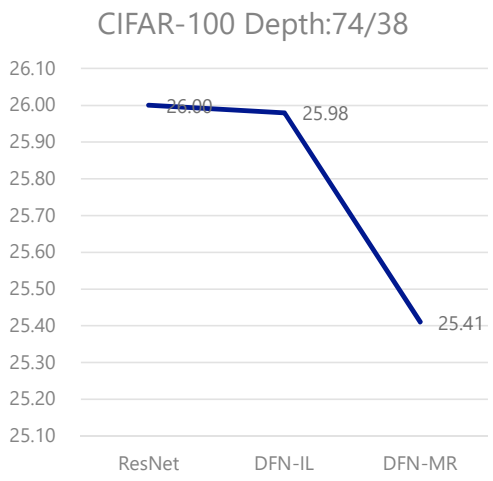

SVHN Depth:50/26
1.94
1.92
1.90
1.88
1.86
1.84
1.82
1.93
1.86
1.86
ResNet
DFN-IL
DFN-MR

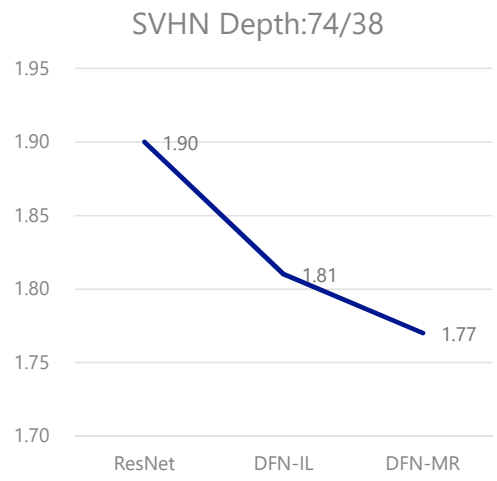

CIFAR-10 Depth:62/32
5.60
5.50
5.40
5.30
5.20
5.10
5.00
4.90
5.50
5.45
5.10
ResNet
DFN-IL
DFN-MR

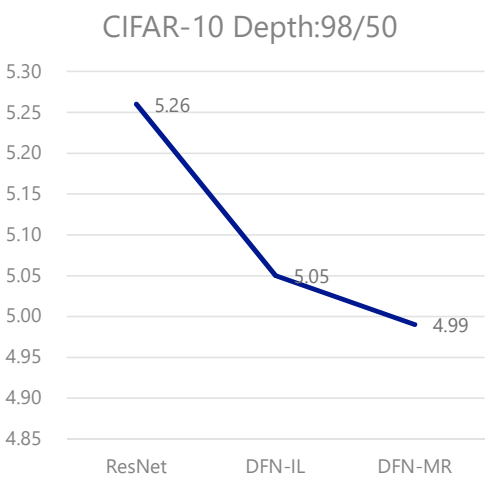

CIFAR-100 Depth:62/32
26.50
26.40
26.30
26.20
26.10
26.00
25.90
25.80
25.70
25.60
25.50
26.44
26.19
25.87
ResNet
DFN-IL
DFN-MR

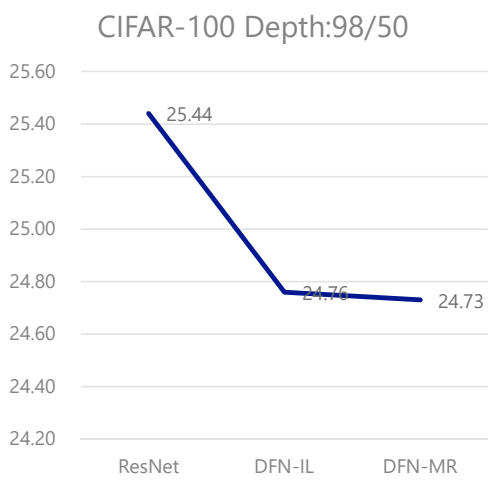

SVHN Depth:62/32
1.90
1.88
1.86
1.84
1.82
1.80
1.78
1.76
1.89
1.86
1.81
ResNet
DFN-IL
DFN-MR

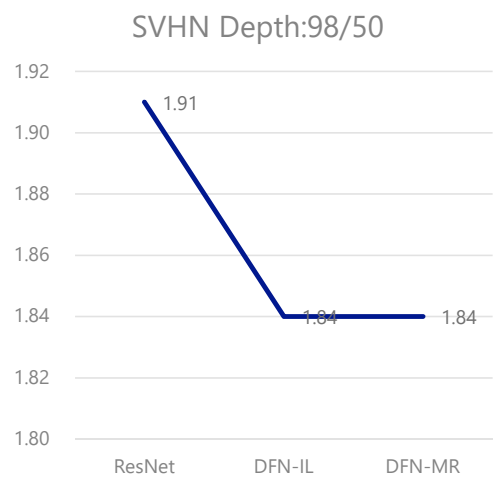

CIFAR-10 Depth:74/38
5.40
5.35
5.30
5.25
5.20
5.15
5.10
5.05
5.35
5.26
5.18
ResNet
DFN-IL
DFN-MR

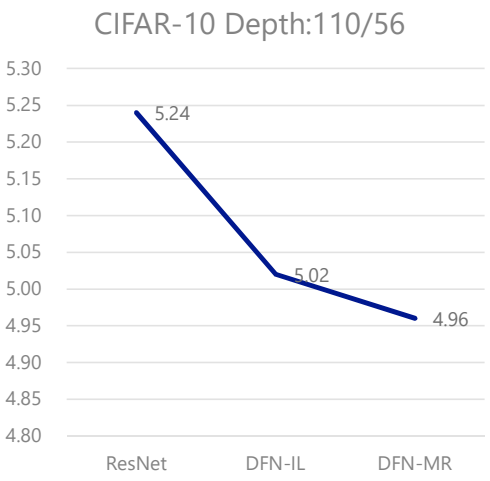

CIFAR-100 Depth:74/38
26.10
26.00
25.90
25.80
25.70
25.60
25.50
25.40
25.30
25.20
25.10
26.00
25.98
25.41
ResNet
DFN-IL
DFN-MR

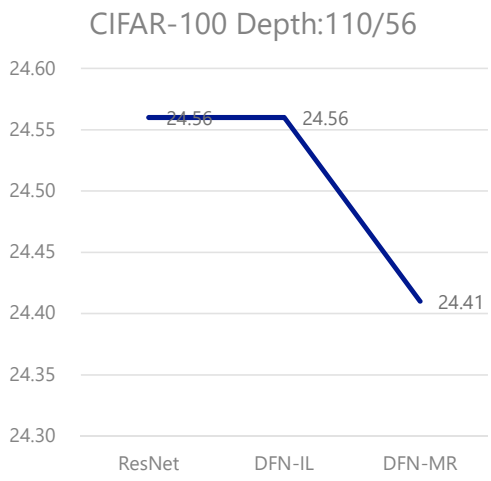

SVHN Depth:74/38
1.95
1.90
1.85
1.80
1.75
1.70
1.90
1.81
1.77
ResNet
DFN-IL
DFN-MR

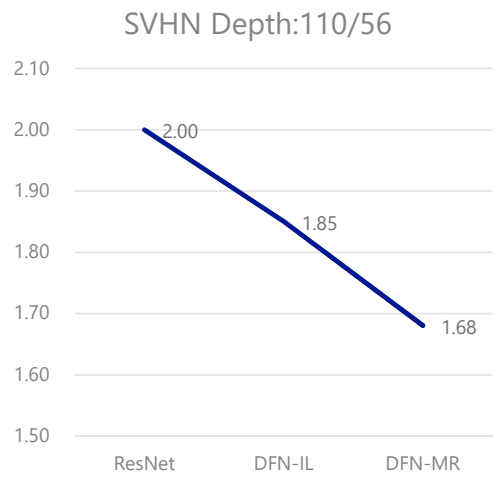

CIFAR-10 Depth:98/50
5.30
5.25
5.20
5.15
5.10
5.05
5.00
4.95
4.90
4.85
5.26
5.05
4.99
ResNet
DFN-IL
DFN-MR

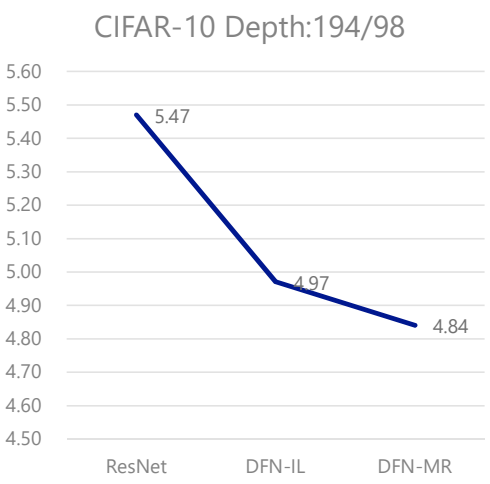

CIFAR-100 Depth:98/50
25.60
25.40
25.20
25.00
24.80
24.60
24.40
24.20
25.44
24.76
24.73
ResNet
DFN-IL
DFN-MR

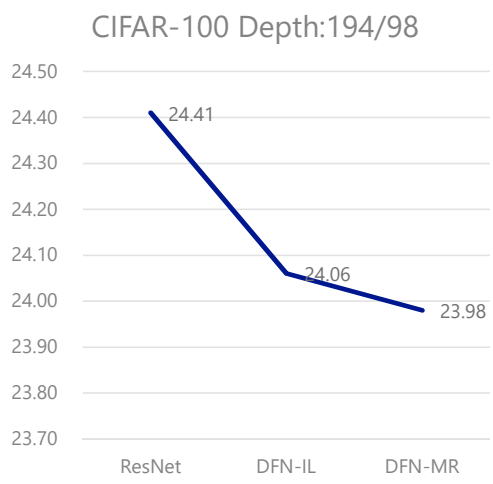

SVHN Depth:98/50
1.92
1.90
1.88
1.86
1.84
1.82
1.80
1.91
1.84
1.84
ResNet
DFN-IL
DFN-MR

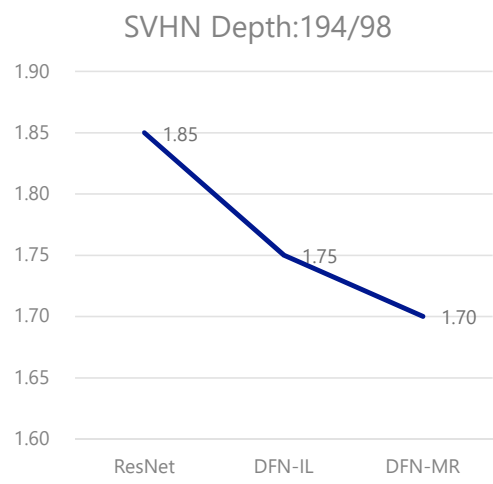

CIFAR-10 Depth:110/56
5.30
5.25
5.20
5.15
5.10
5.05
5.00
4.95
4.90
4.85
4.80
5.24
5.02
4.96
ResNet
DFN-IL
DFN-MR


CIFAR-100 Depth:110/56
24.60
24.55
24.50
24.45
24.40
24.35
24.30
24.56
24.56
24.41
ResNet
DFN-IL
DFN-MR


SVHN Depth:110/56
2.10
2.00
1.90
1.80
1.70
1.60
1.50
2.00
1.85
1.68
ResNet
DFN-IL
DFN-MR


CIFAR-10 Depth:194/98
5.60
5.50
5.40
5.30
5.20
5.10
5.00
4.90
4.80
4.70
4.60
4.50
5.47
4.97
4.84
ResNet
DFN-IL
DFN-MR

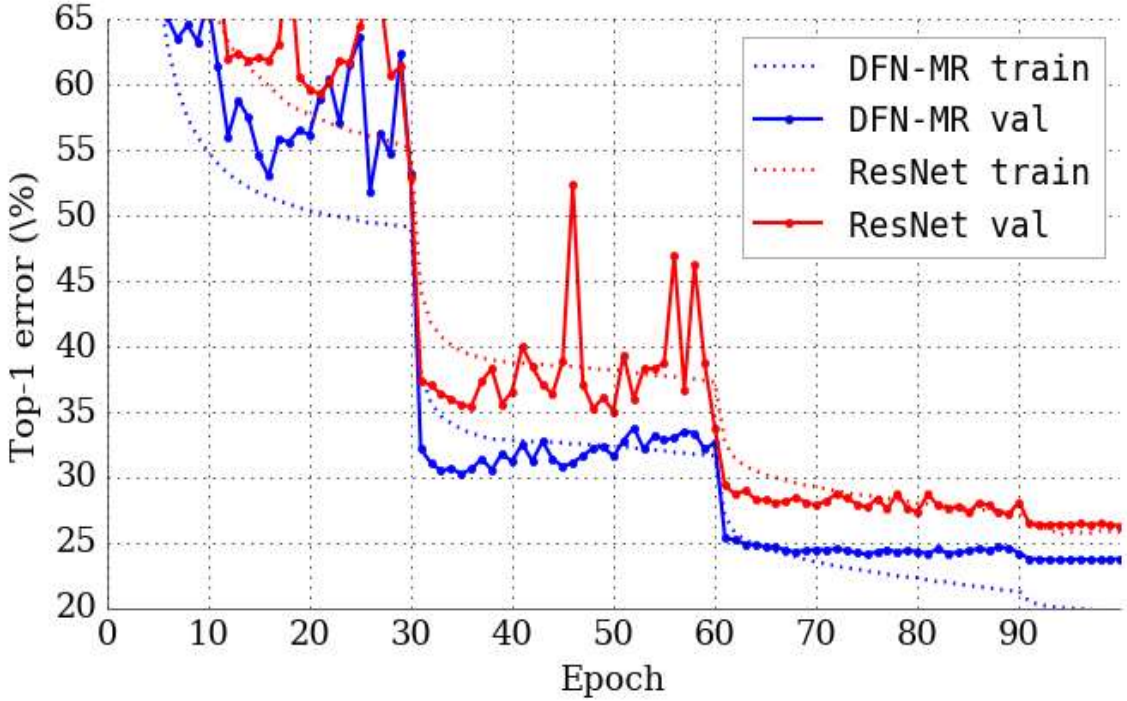

CIFAR-100 Depth:194/98
24.50
24.40
24.30
24.20
24.10
24.00
23.90
23.80
23.70
24.41
24.06
23.98
ResNet
DFN-IL
DFN-MR


SVHN Depth:194/98
1.90
1.85
1.80
1.75
1.70
1.65
1.60
1.85
1.75
1.70
ResNet
DFN-IL
DFN-MR

Comparison with state-of-the-arts

Method	Depth	#Params.	Cifar-10	Cifar-100	SVHN
DSN	-	-	7.97	34.57	1.92
FractalNet with DO/DP	21	38.6M	5.22	23.30	2.01
	21	38.6M	4.60	23.73	1.87
ResNet	110	1.7M	6.41	27.22	2.01
Multi ResNet	200	10.2M	4.35	20.42	-
Wide ResNet	16	11.0M	4.81	22.07	-
	28	36.5M	4.17	20.50	-
DenseNet	40	1.0M	5.24	24.42	1.79
	100	27.2M	3.74	19.25	1.59
DFN (ours)	56	1.7M	4.94	24.46	1.66
DFN (ours)	32	14.9M	3.94	19.25	1.51
DFN (ours)	50	24.8M	3.57	19.00	1.55

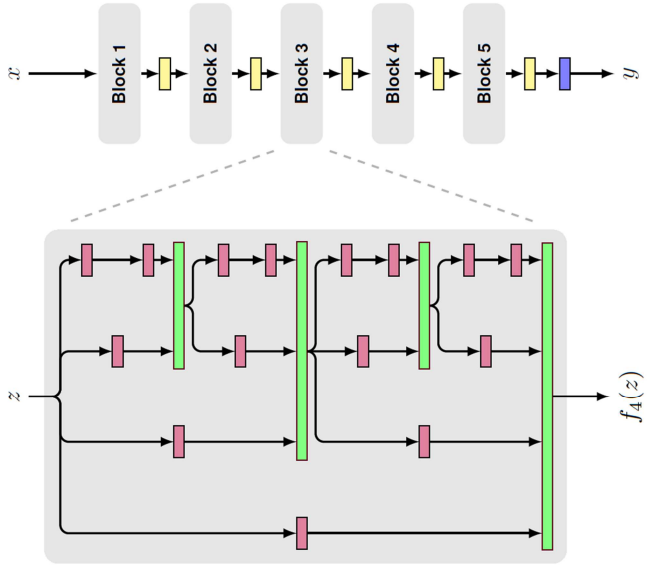
ImageNet Results

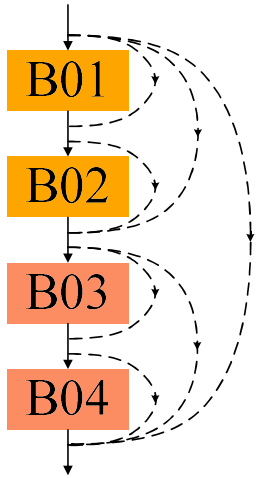


	ResNet-101 (He's)	ResNet-101	DFN
	44.5M		43.3M
Top1 validation error	23.60	26.41	23.66
Top5 validation error	7.10	8.50	**6.81**
Top-1 training error	17.00	25.75	19.72
Top-5 training error	-	8.12	6.59

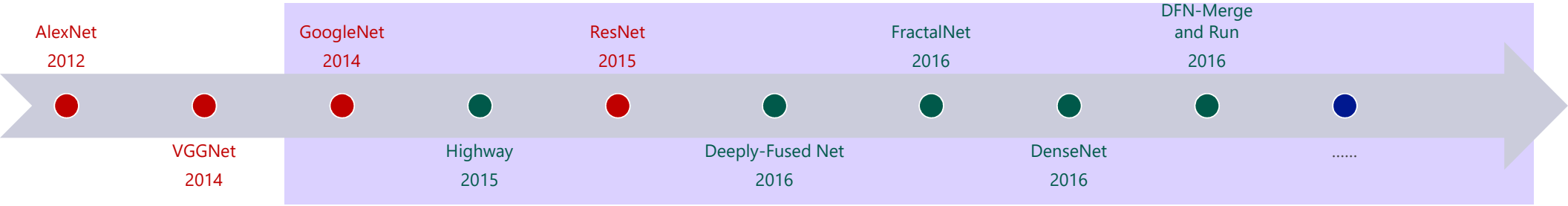
Other network structures

FractalNet, 2016

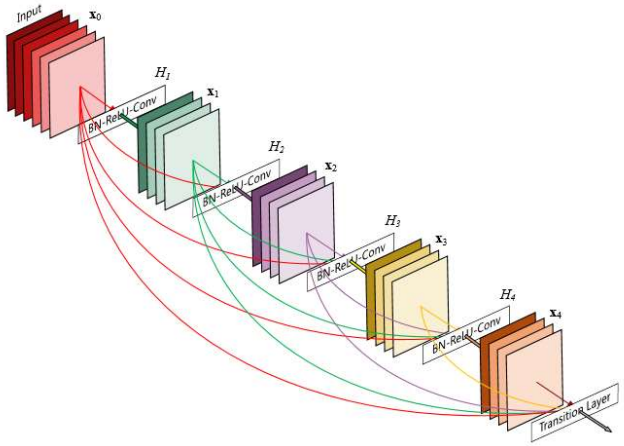


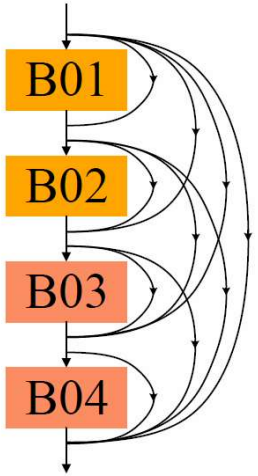


Multi-path, without identify connection, **deep fusion in deep fusion**



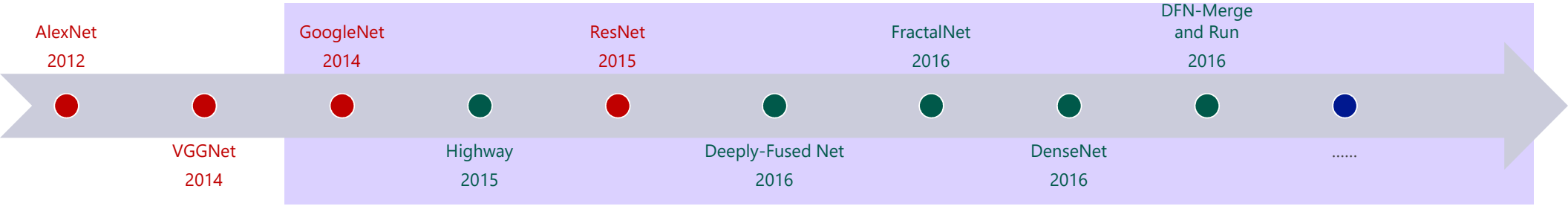
DenseNet, 2016





Dense connection: Express way to **all layers**, deeper is better

Deep graph fusion



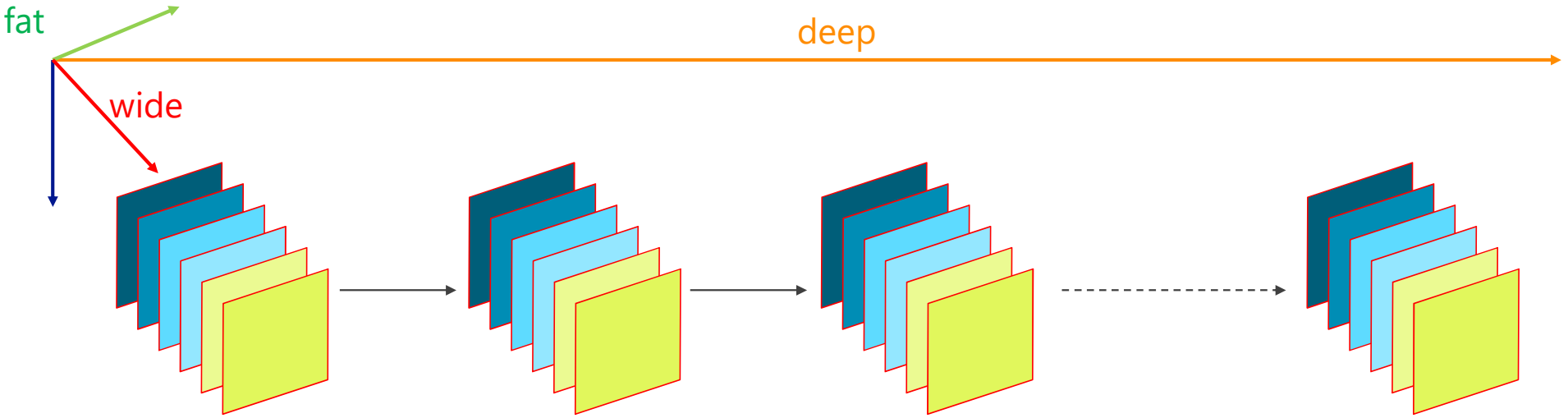
Summary

Deep fusion

- Multiple paths
- Long and short
- Express way between layers
- Weight sharing

Ultra-deep is misleading!

What's next? Wider? Fatter?



Thanks!
Q&A