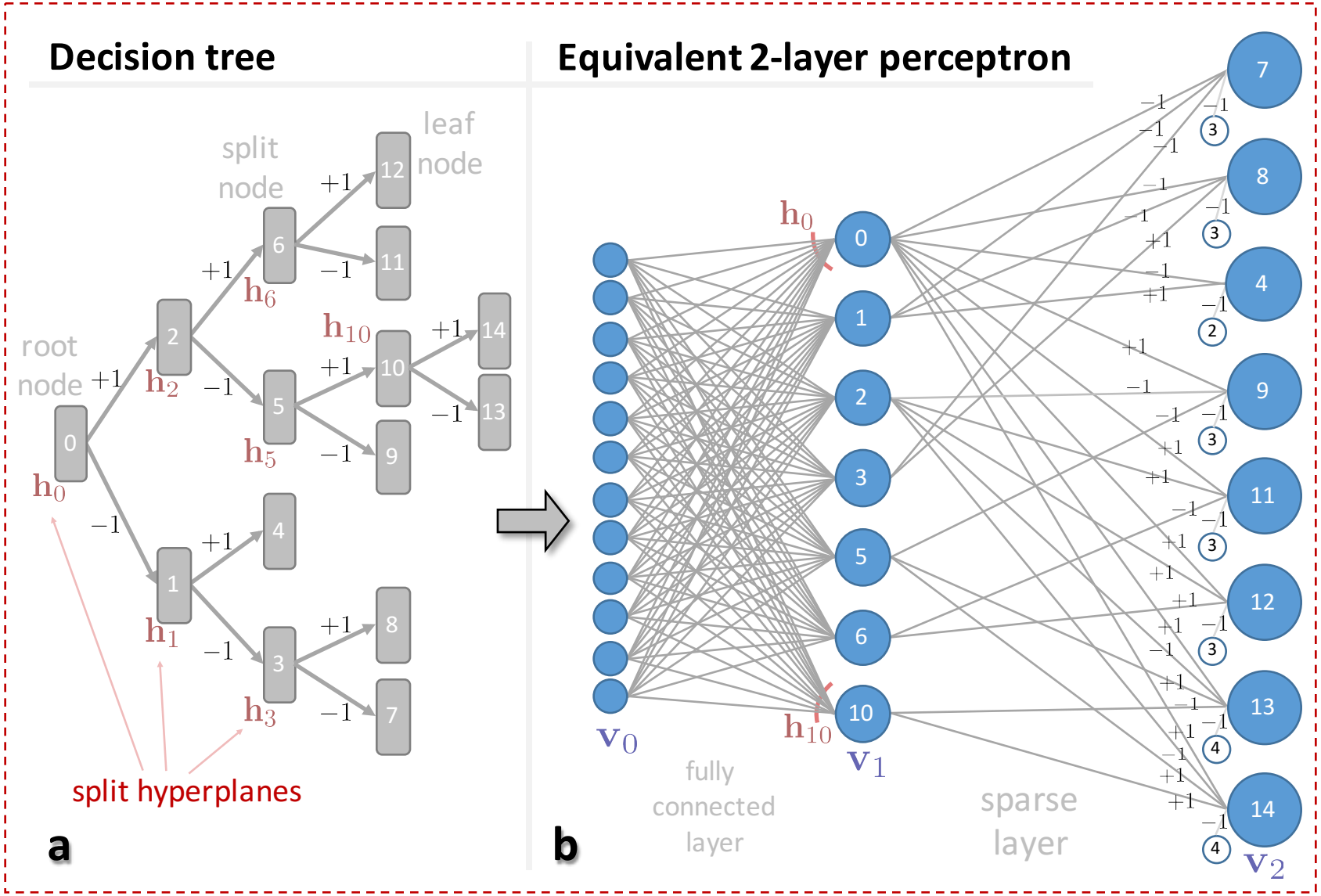
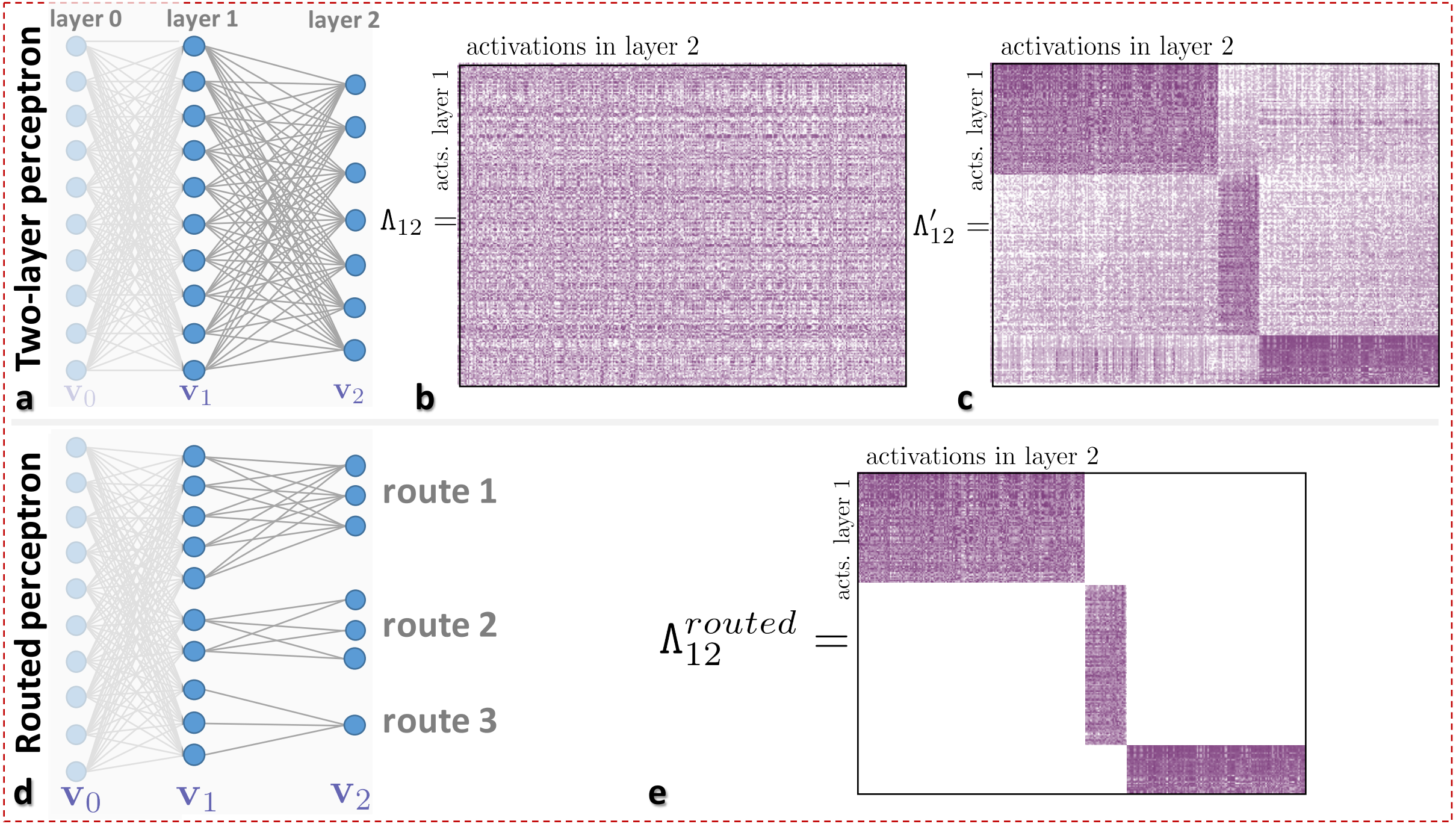
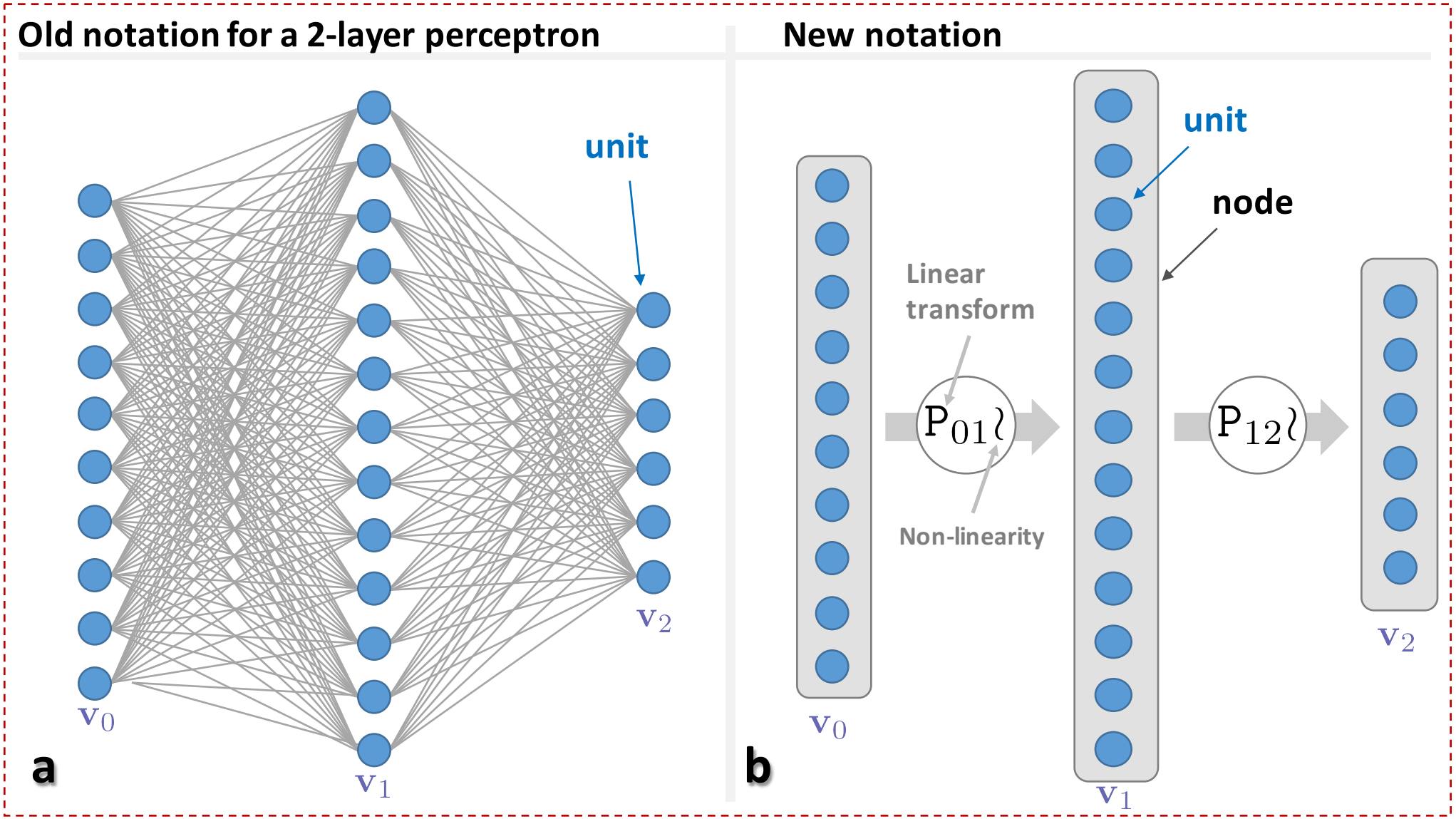
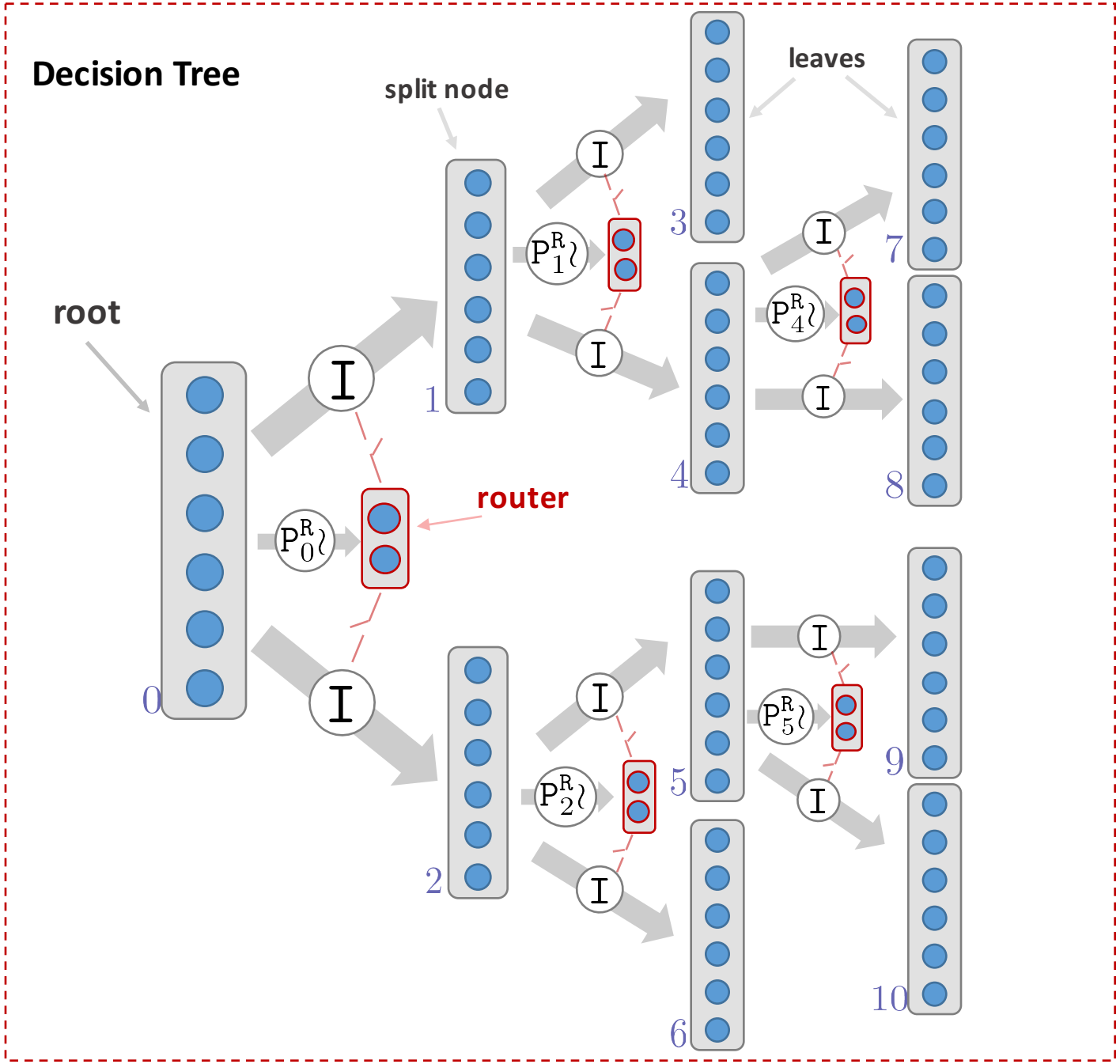
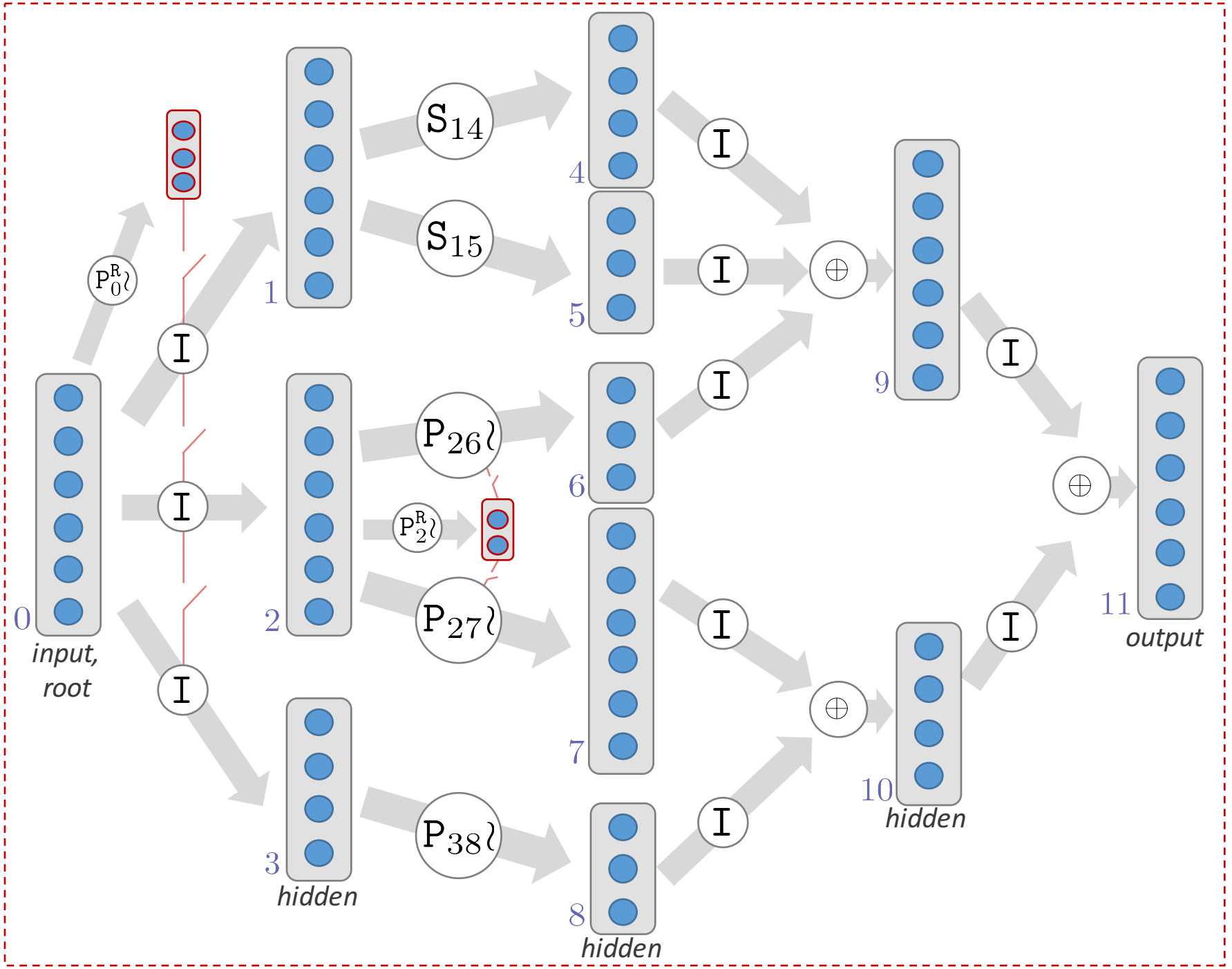
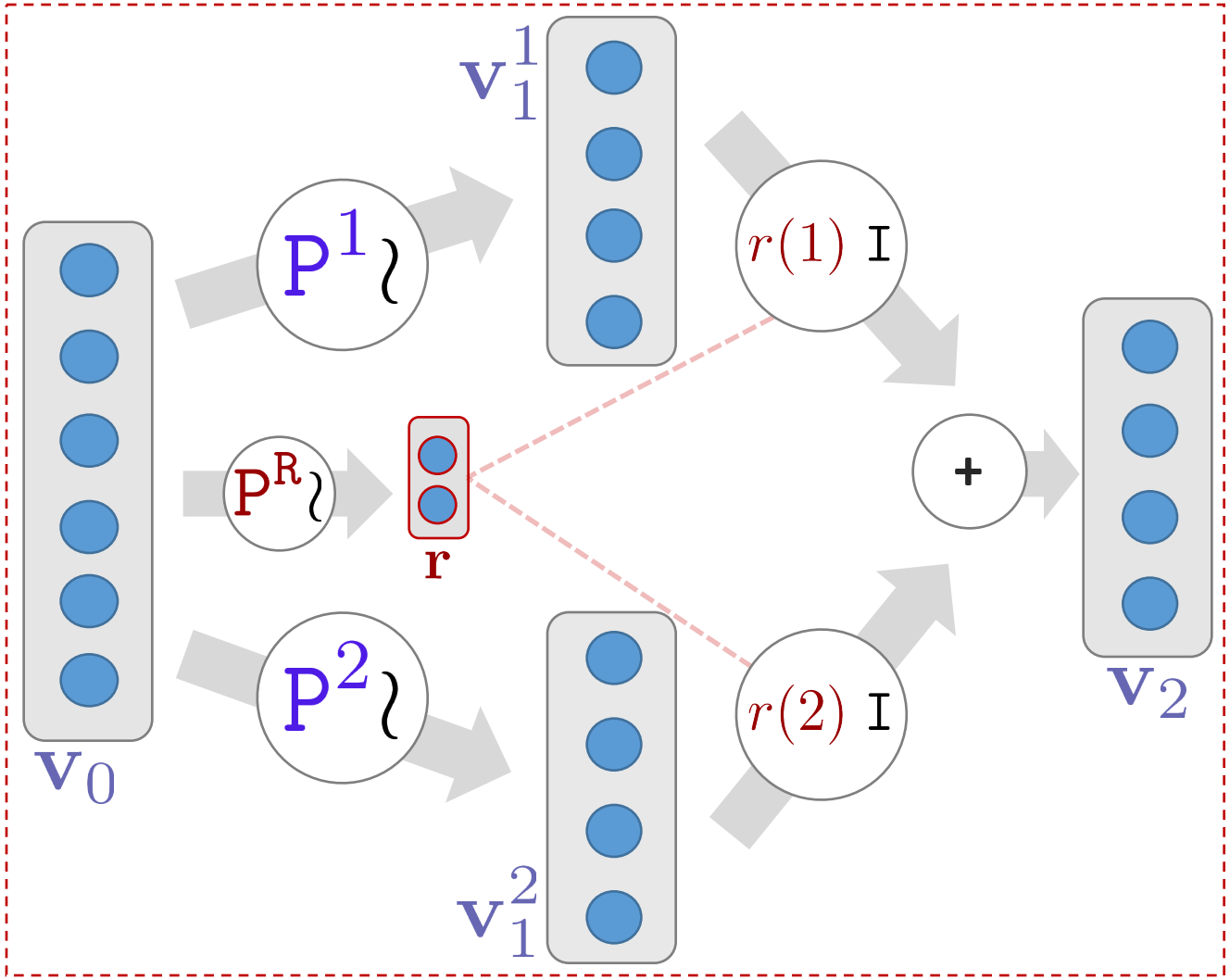

Decision tree
Equivalent 2-layer perceptron
root node
split node
leaf node
split hyperplanes
a
b
fully connected layer
sparse layer
$\mathbf{v}_0$
$\mathbf{v}_1$
$\mathbf{v}_2$
$\mathbf{h}_0$
$\mathbf{h}_1$
$\mathbf{h}_2$
$\mathbf{h}_3$
$\mathbf{h}_5$
$\mathbf{h}_6$
$\mathbf{h}_{10}$


a Two-layer perceptron
layer 0
layer 1
layer 2
v_0
v_1
v_2
b
activations in layer 2
acts. layer 1
Λ_{12} =
c
activations in layer 2
acts. layer 1
Λ'_{12} =
d Routed perceptron
route 1
route 2
route 3
v_0
v_1
v_2
e
activations in layer 2
acts. layer 1
Λ_{12}^{routed} =


Old notation for a 2-layer perceptron
New notation
unit
v_0
v_1
v_2
a
b
unit
node
Linear transform
P_{01}
P_{12}
Non-linearity
v_0
v_1
v_2


Decision Tree
split node
leaves
root
router
0
1
2
3
4
5
6
7
8
9
10
I
$P_0^R(\cdot)$
$P_1^R(\cdot)$
$P_2^R(\cdot)$
$P_4^R(\cdot)$
$P_5^R(\cdot)$


0
input,
root
1
2
3
hidden
4
5
6
7
8
hidden
9
10
hidden
11
output
S_{14}
S_{15}
P_{26}
P_{27}
P_{38}
P^R_0
P^R_2
I
⊕



$$E(\boldsymbol{\theta}) = \frac{1}{2}\sum_{i}^{N} ||\mathbf{y}_i^* - \mathbf{y}_i(\boldsymbol{\theta})||^2$$ The energy to be minimized

$$\mathbf{y}(\boldsymbol{\theta}) = \mathbf{r}(\boldsymbol{\theta})\ \mathrm{Y}(\boldsymbol{\theta})$$ Network's forward mapping

$$\mathrm{Y} = \begin{bmatrix} \vdots & \vdots & \vdots \\ - & \mathbf{y}^j & - \\ \vdots & \vdots & \vdots \end{bmatrix}$$ Matrix of outputs for all routes

$$\mathbf{y}^j = \sigma\left(\mathrm{P}^j\mathbf{x}\right)$$ Intermediate output for j-th route

$$\mathbf{r} = \sigma\left(\mathrm{R}\mathbf{x}\right)$$ Soft routing weights

$$\Delta\boldsymbol{\theta}_{t+1} := -\rho \left.\frac{\partial E}{\partial \boldsymbol{\theta}}\right|_t$$ The parameter update rule

$$\boldsymbol{\theta} := \{\mathrm{R}, \{\mathrm{P}^j\}\}$$ The parameters to be optimized

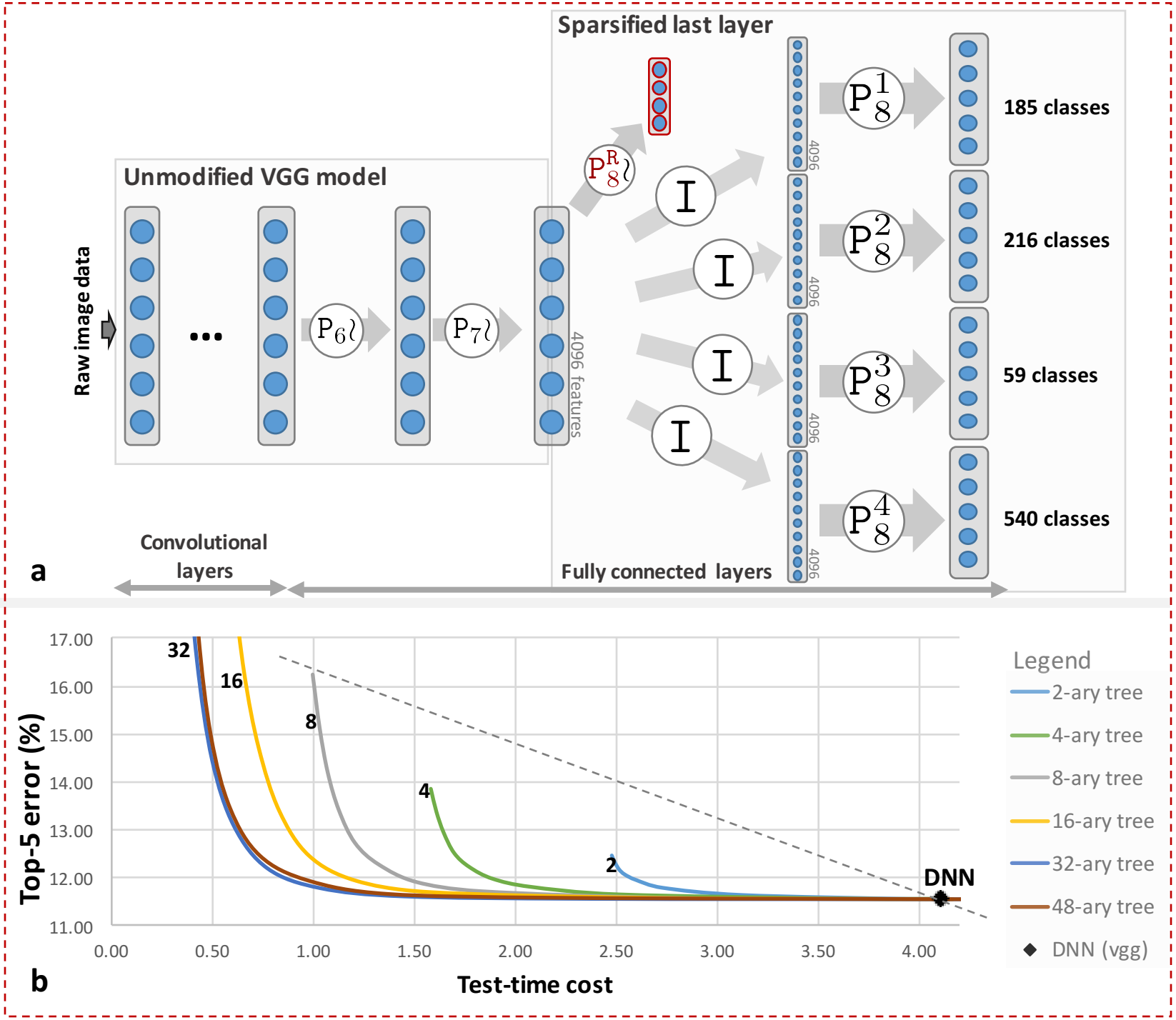
$$\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1^*), \ldots, (\mathbf{x}_i, \mathbf{y}_i^*), \ldots, (\mathbf{x}_N, \mathbf{y}_N^*)\}$$ The labelled training set

Chain rule to compute partial derivatives for gradient descent

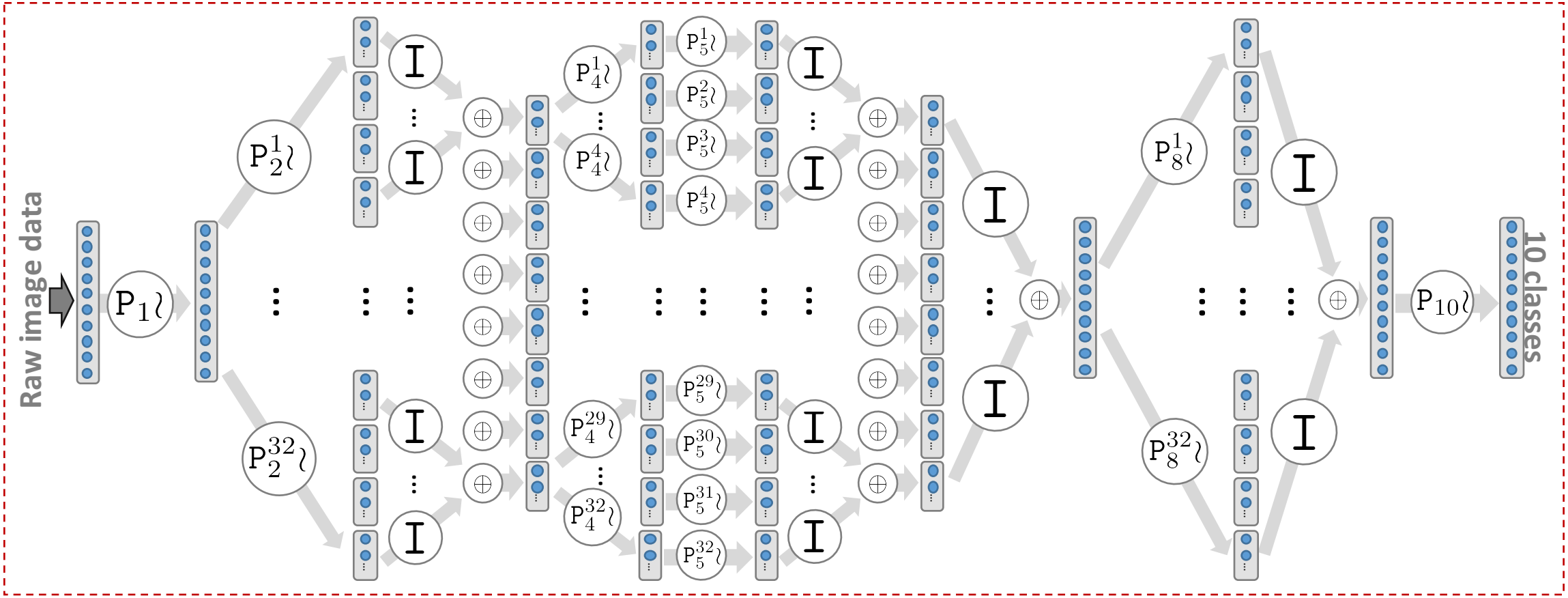
$$\boldsymbol{\phi}^j := \mathrm{P}^j\mathbf{x} \qquad \mathbf{y}^j = \sigma(\boldsymbol{\phi}^j)$$

$$\frac{\partial E}{\partial \boldsymbol{\theta}} = \frac{\partial E}{\partial \mathbf{y}}\frac{\partial \mathbf{y}}{\partial \boldsymbol{\theta}} = \frac{\partial E}{\partial \mathbf{y}}\left(\frac{\partial \mathbf{r}}{\partial \mathrm{R}}\mathrm{Y} + \sum_j r(j)\frac{\partial \mathbf{y}^j}{\partial \boldsymbol{\phi}^j}\frac{\partial \boldsymbol{\phi}^j}{\partial \mathrm{P}^j}\right)$$

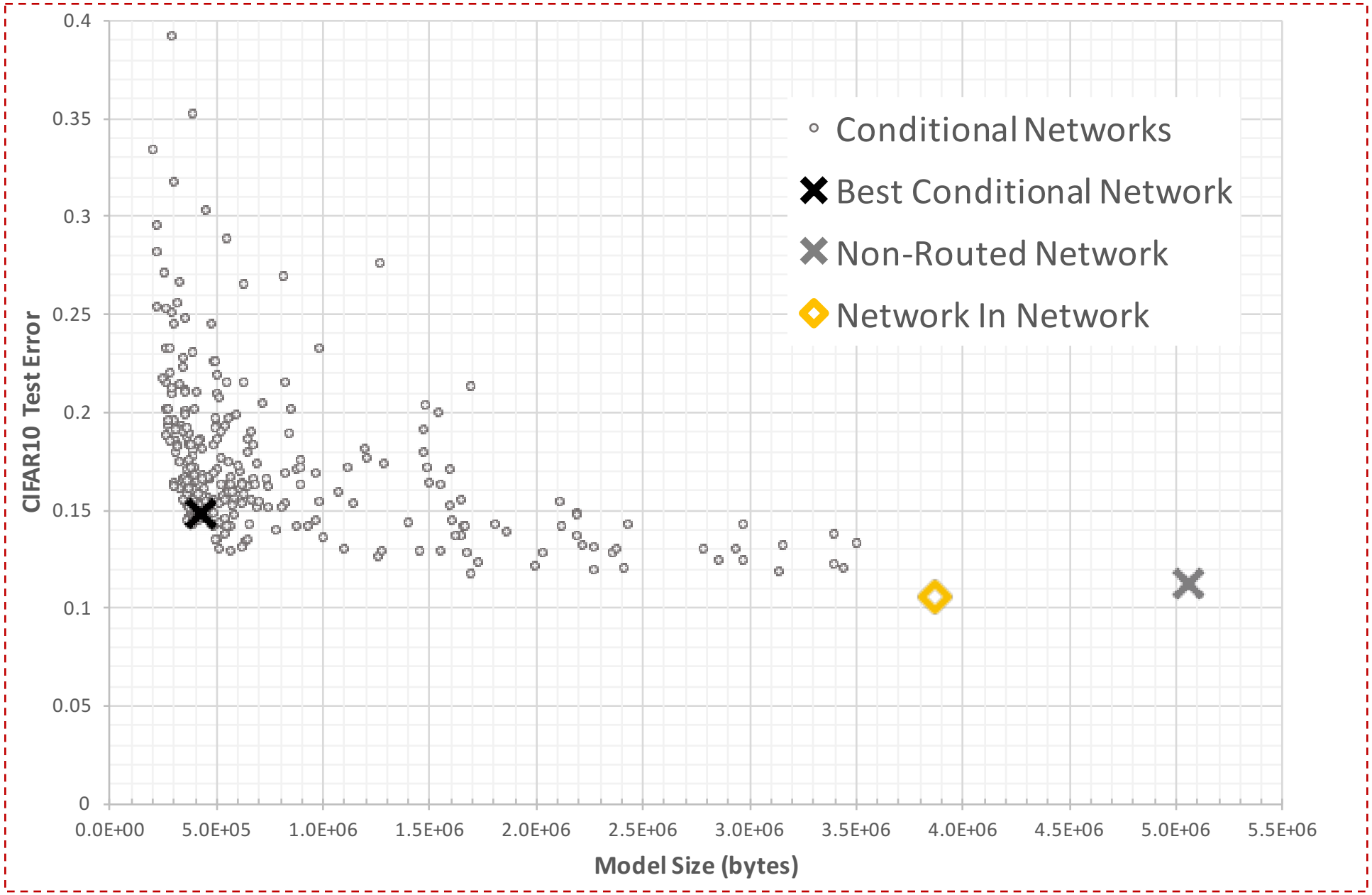
Routing weights influence the back-propagating errors differently for different routes


Sparsified last layer
Unmodified VGG model
Raw image data
$P_{6}\wr$
$P_{7}\wr$
4096 features
$P_{8}^{R}\wr$
I
I
I
I
4096
4096
4096
4096
P_{8}^{1}
P_{8}^{2}
P_{8}^{3}
P_{8}^{4}
185 classes
216 classes
59 classes
540 classes
Convolutional layers
Fully connected layers
a
17.00
16.00
15.00
14.00
13.00
12.00
11.00
Top-5 error (%)
0.00
0.50
1.00
1.50
2.00
2.50
3.00
3.50
4.00
Test-time cost
32
16
8
4
2
DNN
Legend
2-ary tree
4-ary tree
8-ary tree
16-ary tree
32-ary tree
48-ary tree
DNN (vgg)
b

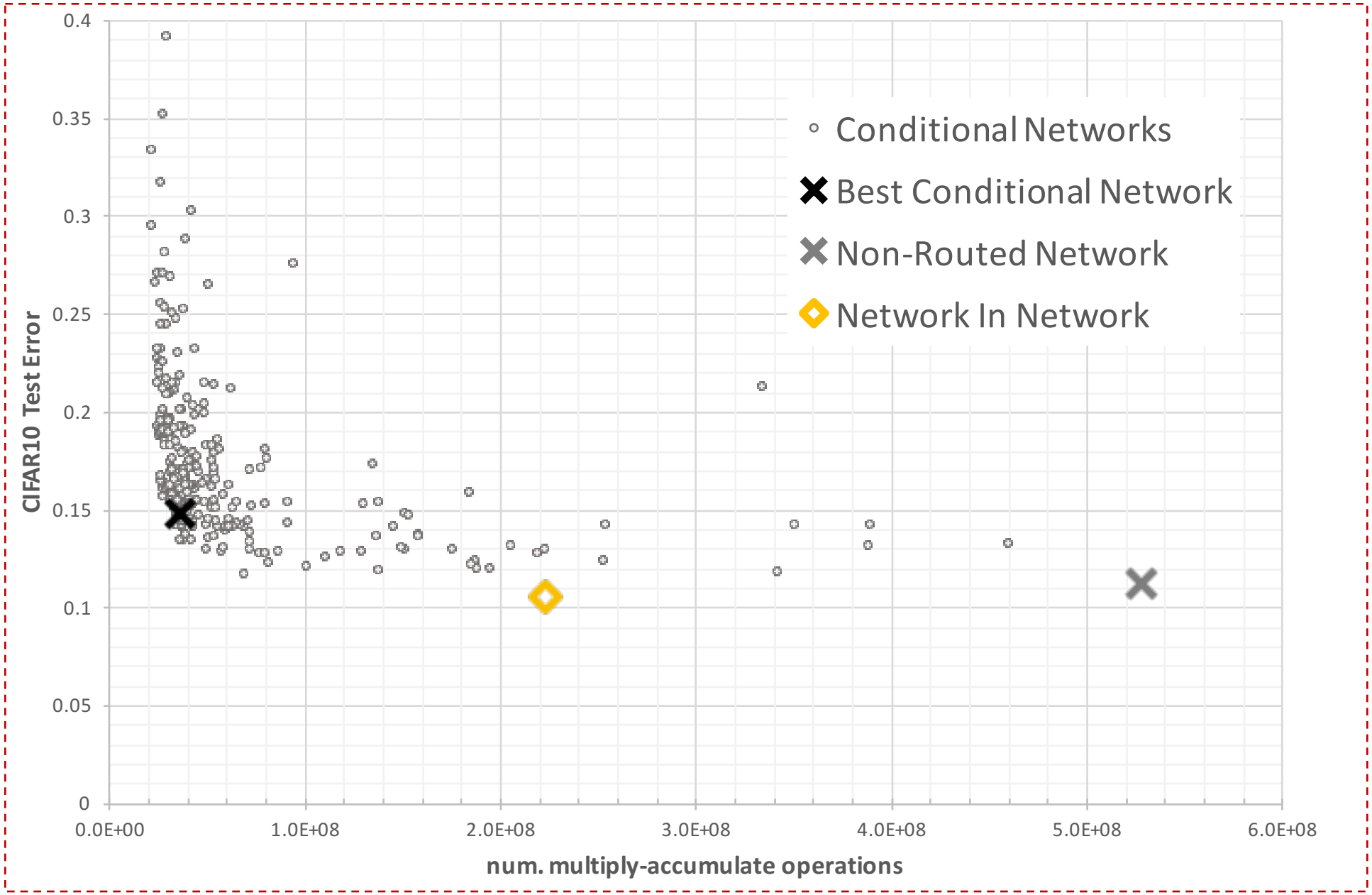
CIFAR


Raw image data
P_1
P_2^1
P_2^{32}
P_4^1
P_4^4
P_4^{29}
P_4^{32}
P_5^1
P_5^2
P_5^3
P_5^4
P_5^{29}
P_5^{30}
P_5^{31}
P_5^{32}
P_8^1
P_8^{32}
P_{10}
10 classes

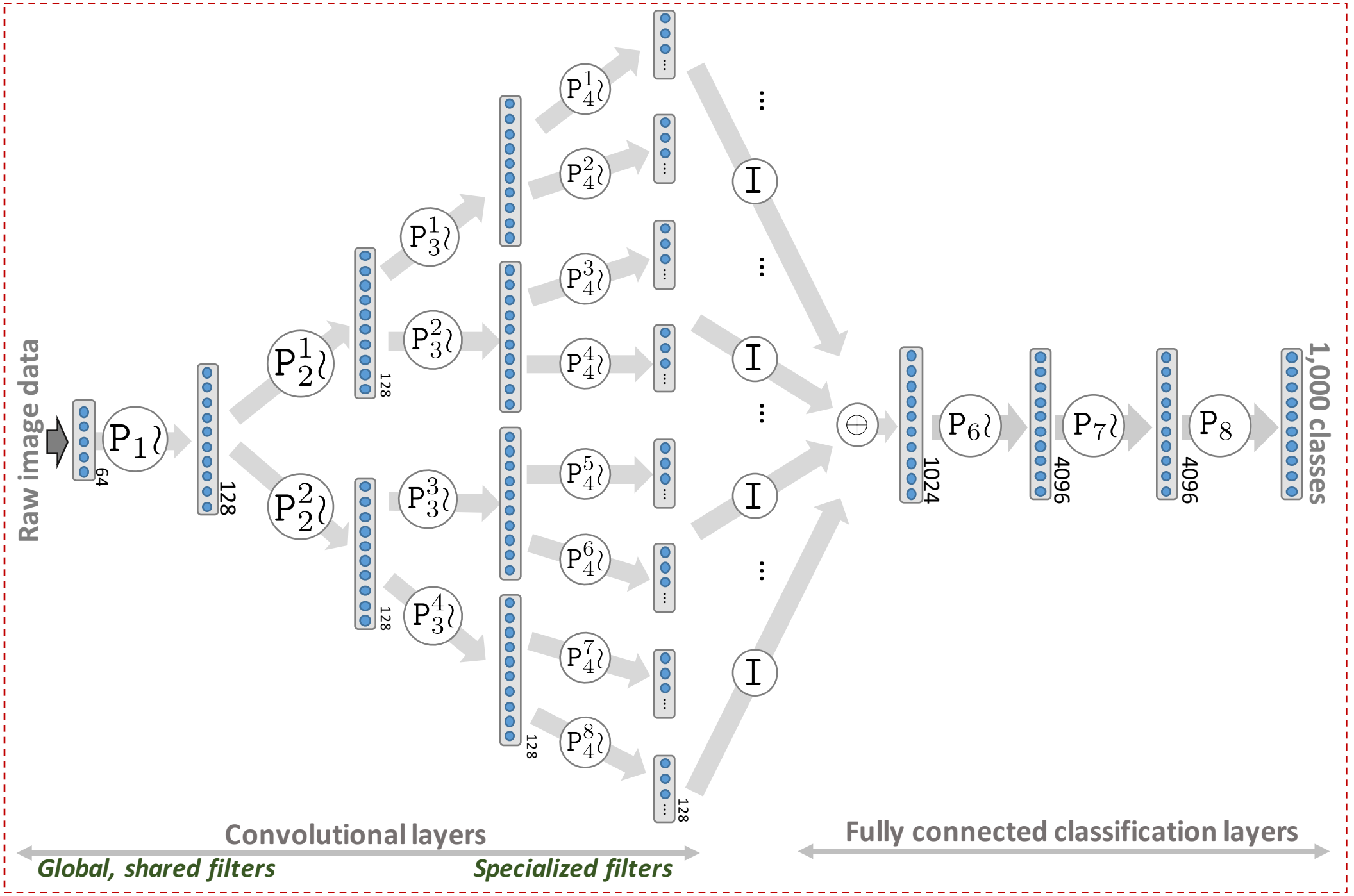
CIFAR


Conditional Networks
Best Conditional Network
Non-Routed Network
Network In Network
CIFAR10 Test Error
Model Size (bytes)
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.0E+00
5.0E+05
1.0E+06
1.5E+06
2.0E+06
2.5E+06
3.0E+06
3.5E+06
4.0E+06
4.5E+06
5.0E+06
5.5E+06

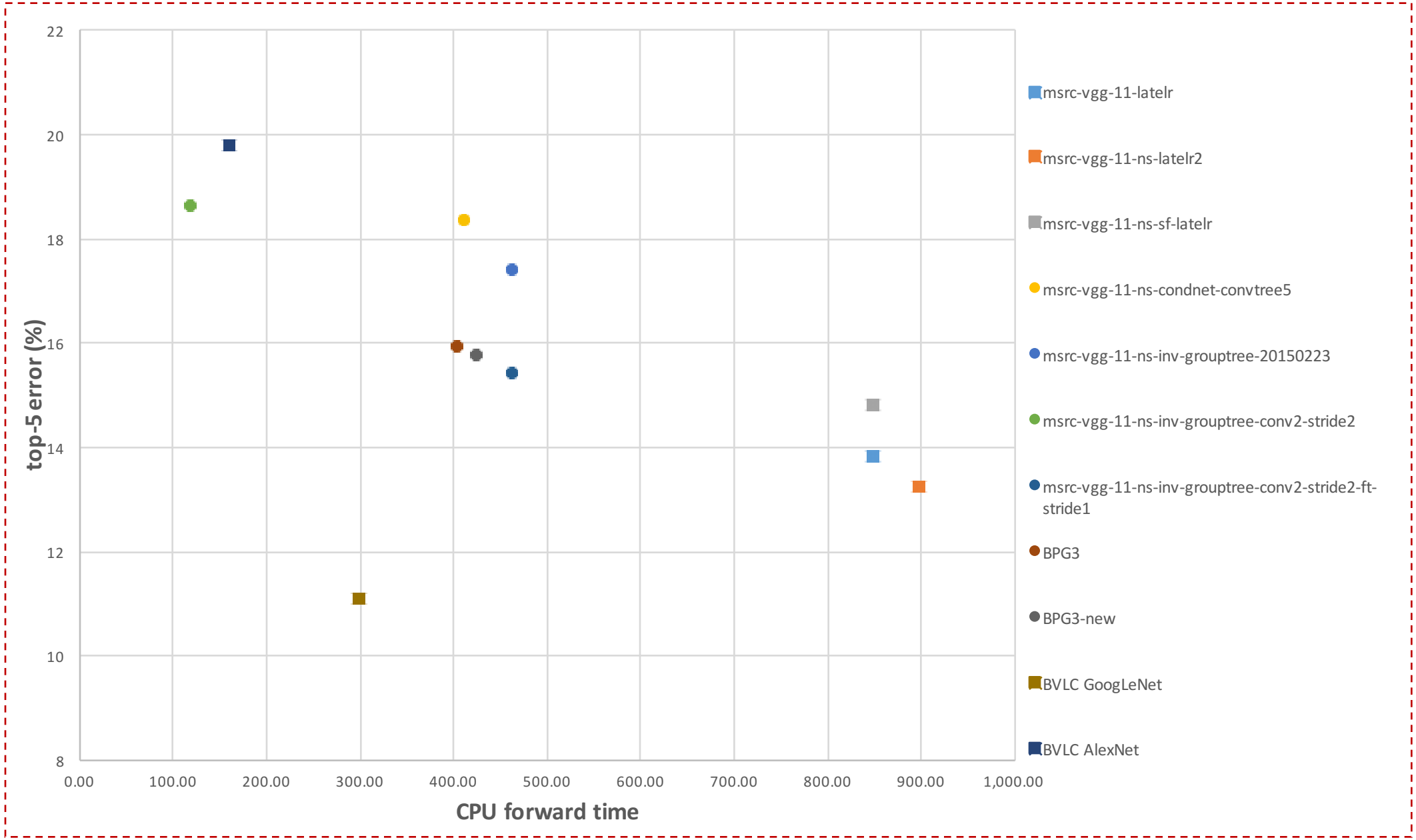
CIFAR


0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
CIFAR10 Test Error
0.0E+00
1.0E+08
2.0E+08
3.0E+08
4.0E+08
5.0E+08
6.0E+08
num. multiply-accumulate operations
Conditional Networks
Best Conditional Network
Non-Routed Network
Network In Network

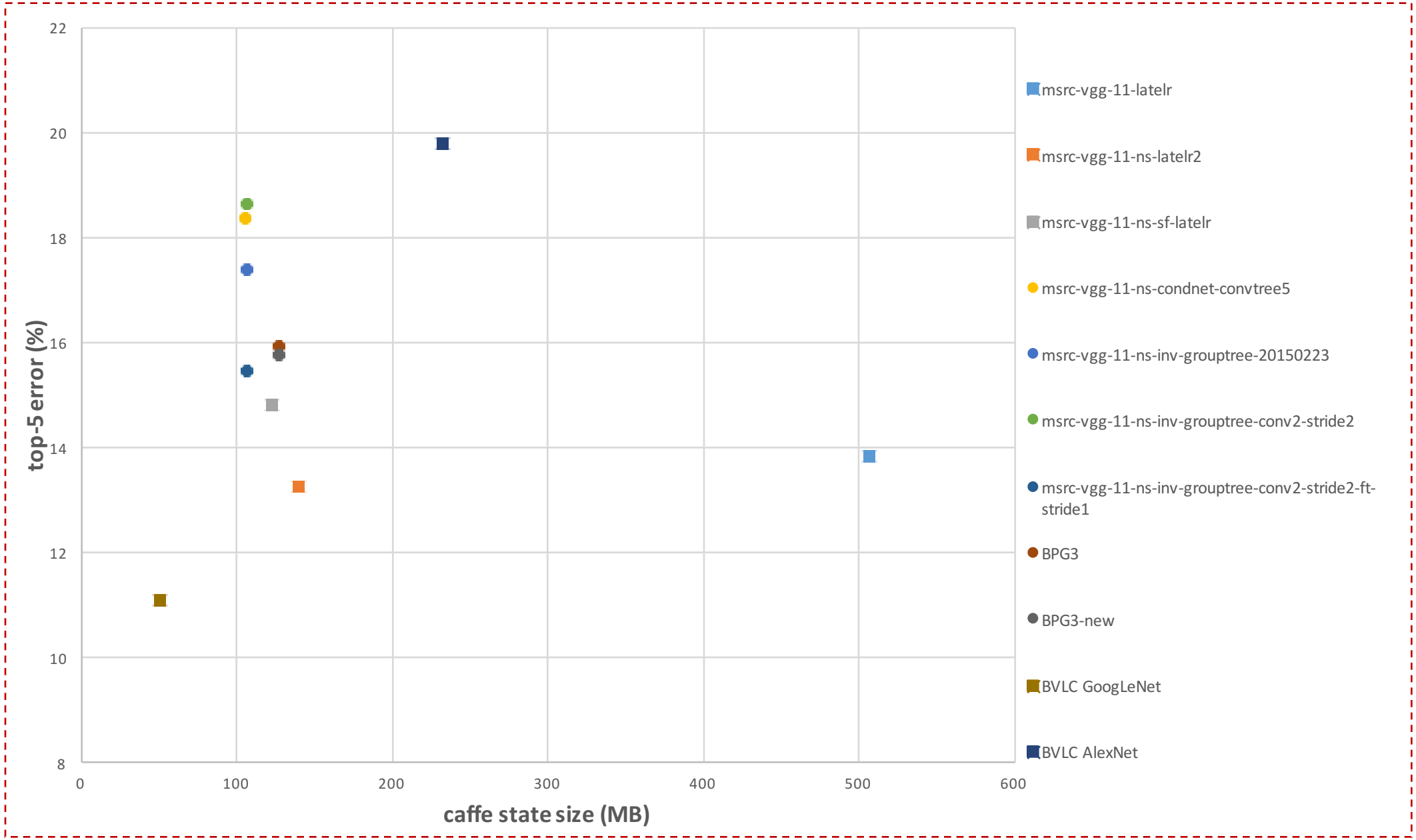
ImageNet


Raw image data
64
128
128
128
128
128
1024
4096
4096
1,000 classes
Convolutional layers
Global, shared filters
Specialized filters
Fully connected classification layers

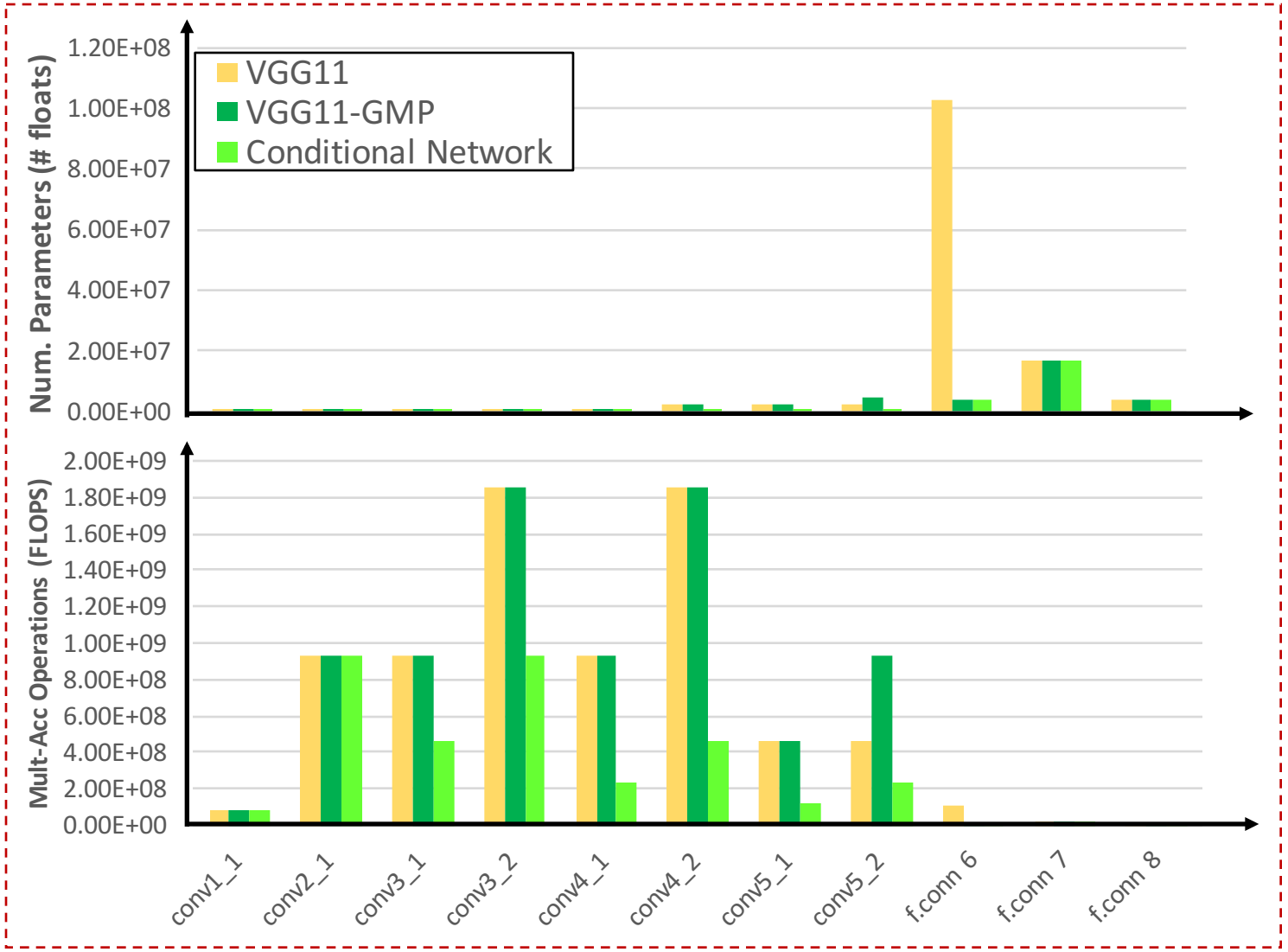
ImageNet


22
20
18
16
14
12
10
8
top-5 error (%)
0.00
100.00
200.00
300.00
400.00
500.00
600.00
700.00
800.00
900.00
1,000.00
CPU forward time
msrc-vgg-11-latelr
msrc-vgg-11-ns-latelr2
msrc-vgg-11-ns-sf-latelr
msrc-vgg-11-ns-condnet-convtree5
msrc-vgg-11-ns-inv-grouptree-20150223
msrc-vgg-11-ns-inv-grouptree-conv2-stride2
msrc-vgg-11-ns-inv-grouptree-conv2-stride2-ft-stride1
BPG3
BPG3-new
BVLC GoogLeNet
BVLC AlexNet

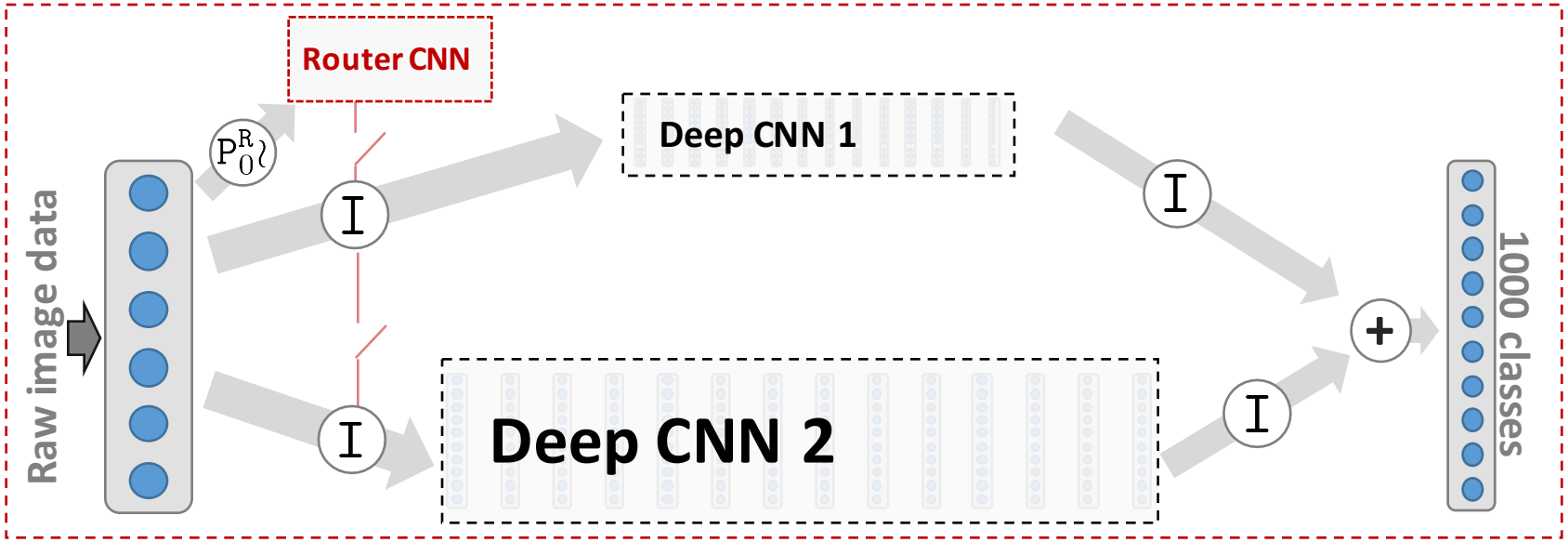
ImageNet


22
20
18
16
14
12
10
8
top-5 error (%)
0
100
200
300
400
500
600
caffe state size (MB)
msrc-vgg-11-latelr
msrc-vgg-11-ns-latelr2
msrc-vgg-11-ns-sf-latelr
msrc-vgg-11-ns-condnet-convtree5
msrc-vgg-11-ns-inv-grouptree-20150223
msrc-vgg-11-ns-inv-grouptree-conv2-stride2
msrc-vgg-11-ns-inv-grouptree-conv2-stride2-ft-stride1
BPG3
BPG3-new
BVLC GoogLeNet
BVLC AlexNet

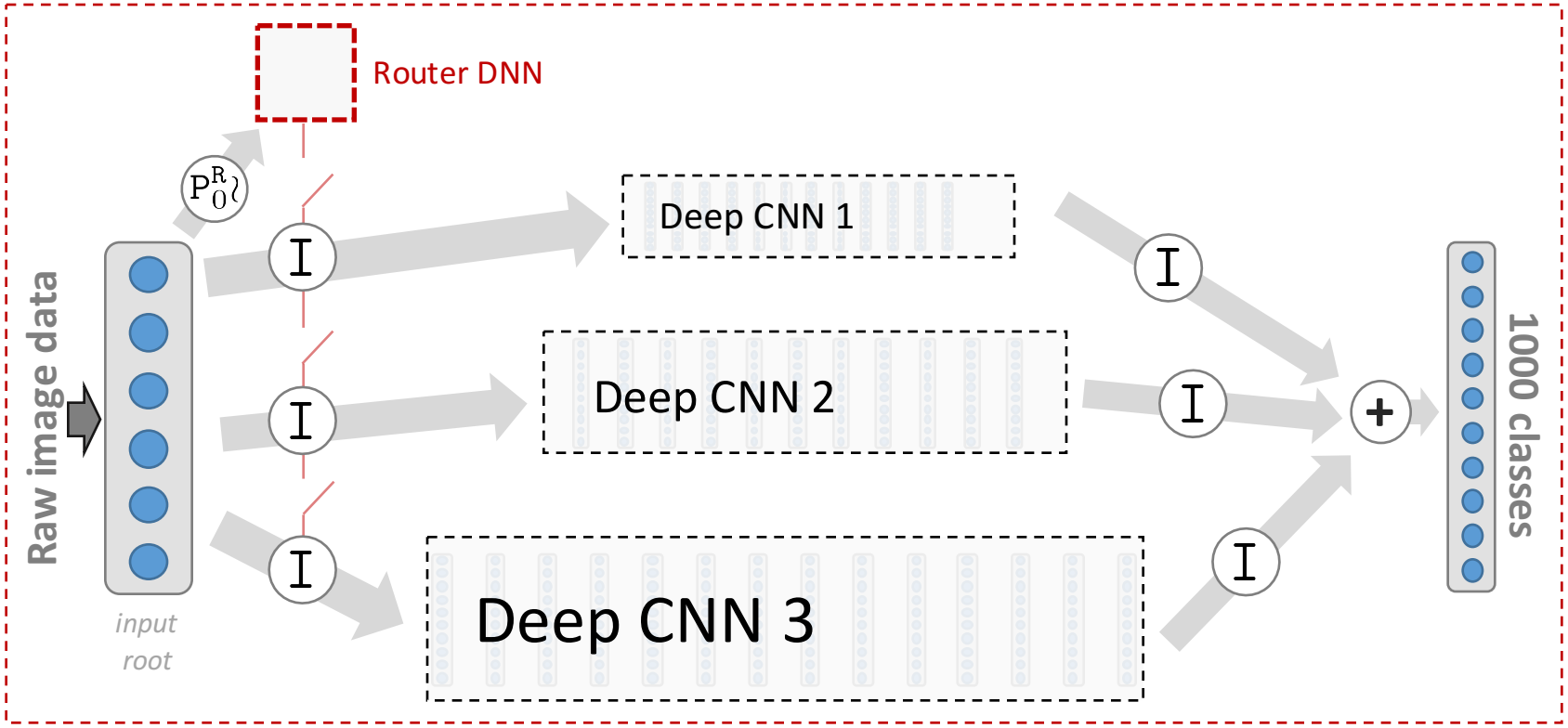
ImageNet


Num. Parameters (# floats)
1.20E+08
1.00E+08
8.00E+07
6.00E+07
4.00E+07
2.00E+07
0.00E+00
VGG11
VGG11-GMP
Conditional Network
Mult-Acc Operations (FLOPS)
2.00E+09
1.80E+09
1.60E+09
1.40E+09
1.20E+09
1.00E+09
8.00E+08
6.00E+08
4.00E+08
2.00E+08
0.00E+00
conv1_1
conv2_1
conv3_1
conv3_2
conv4_1
conv4_2
conv5_1
conv5_2
f.conn 6
f.conn 7
f.conn 8

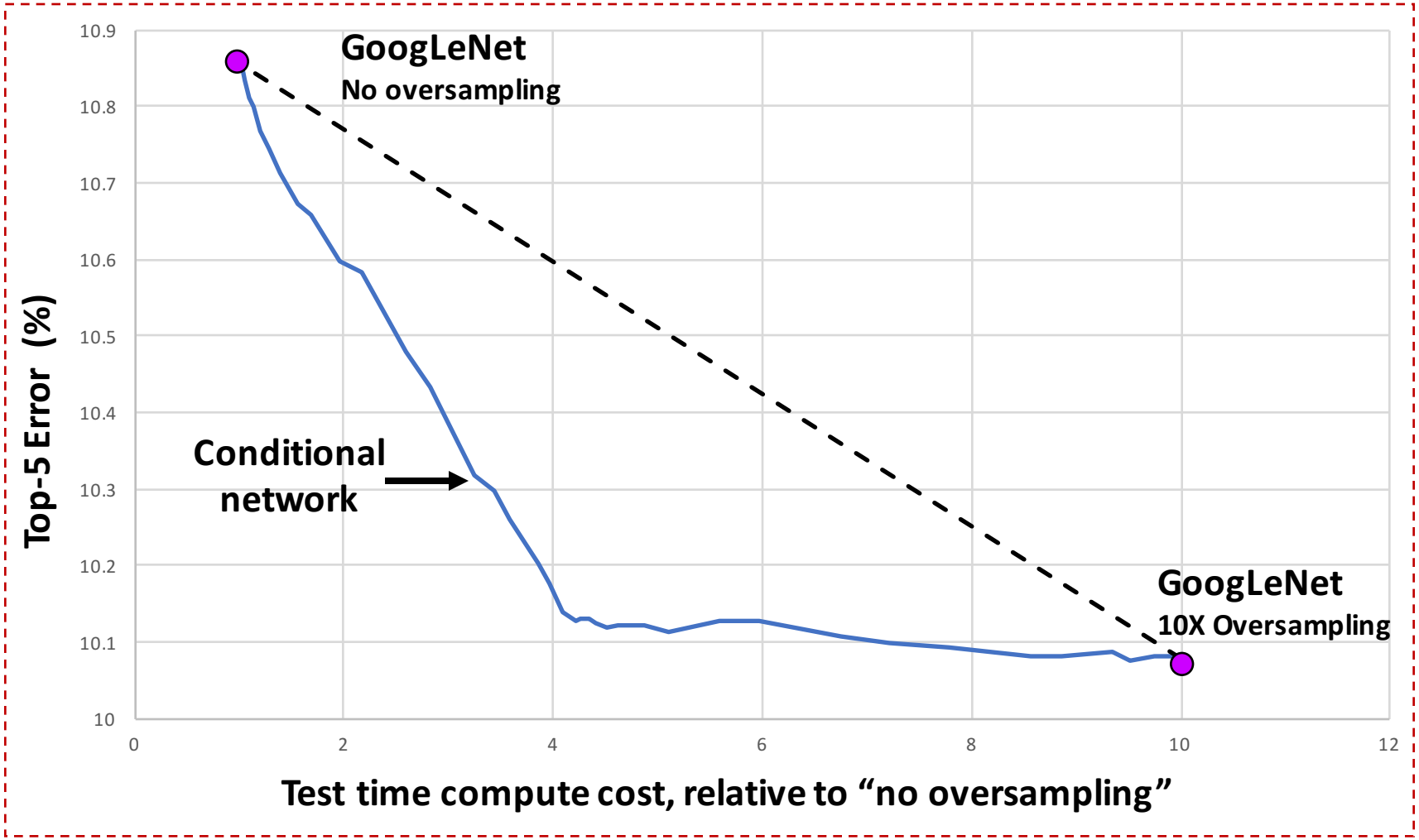
Mixing



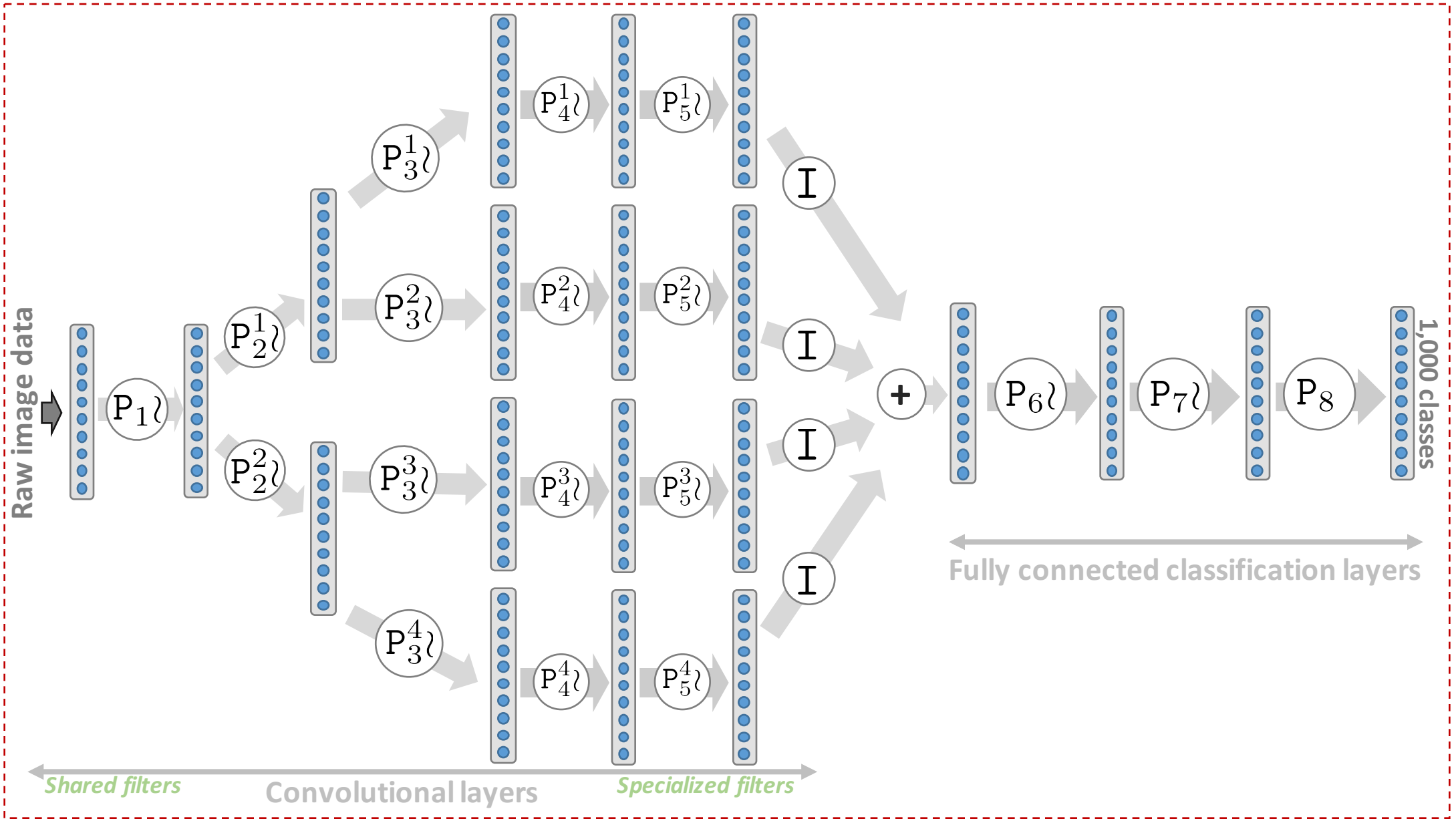
Mixing


Router DNN
Raw image data
input
root
Deep CNN 1
Deep CNN 2
Deep CNN 3
1000 classes

Mixing


GoogLeNet
No oversampling
Conditional network
GoogLeNet
10X Oversampling
Top-5 Error (%)
10.9
10.8
10.7
10.6
10.5
10.4
10.3
10.2
10.1
10
0
2
4
6
8
10
12
Test time compute cost, relative to “no oversampling”

OLD unused


Raw image data
P_1
P_2^1
P_2^2
P_3^1
P_3^2
P_3^3
P_3^4
P_4^1
P_4^2
P_4^3
P_4^4
P_5^1
P_5^2
P_5^3
P_5^4
I
I
I
I
+
P_6
P_7
P_8
1,000 classes
Fully connected classification layers
Shared filters
Convolutional layers
Specialized filters