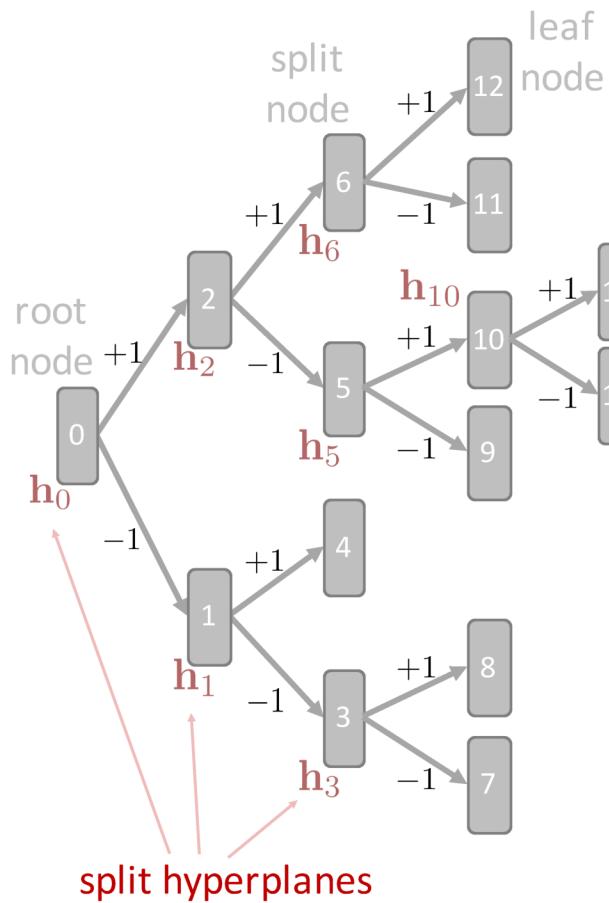
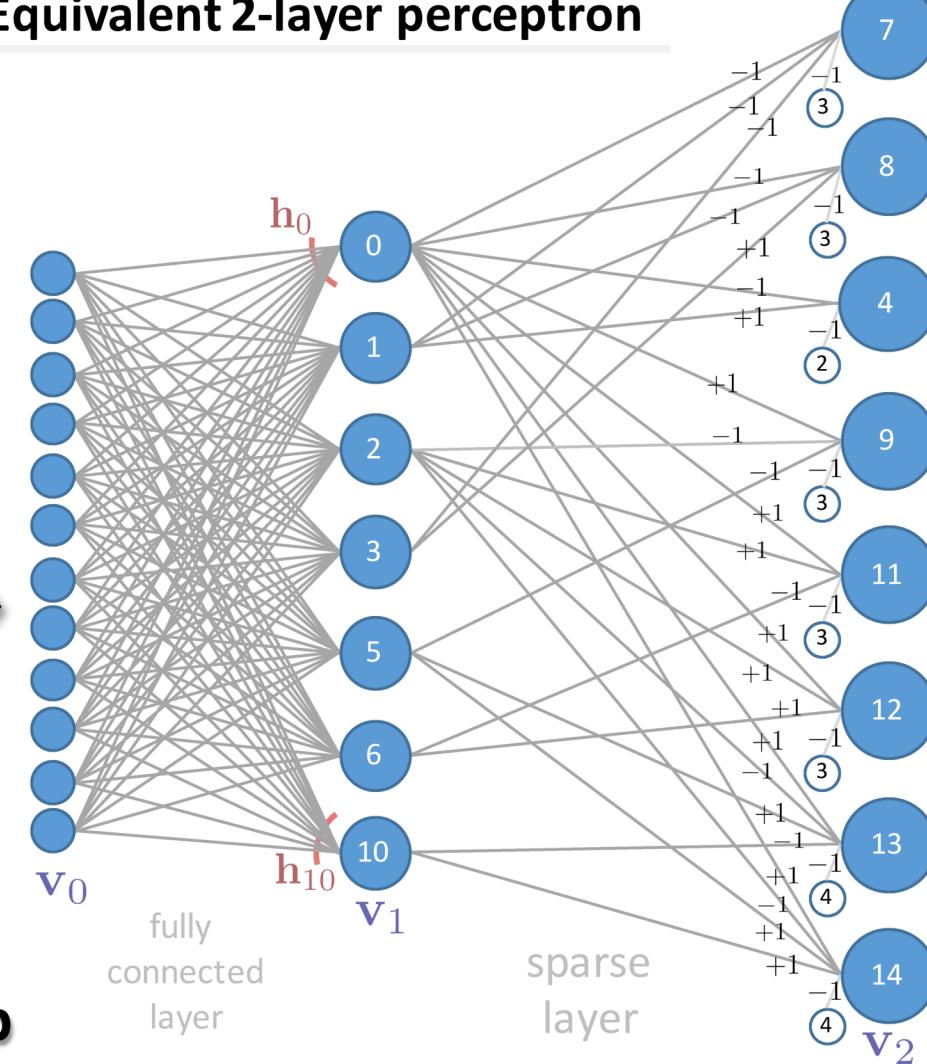


## Decision tree

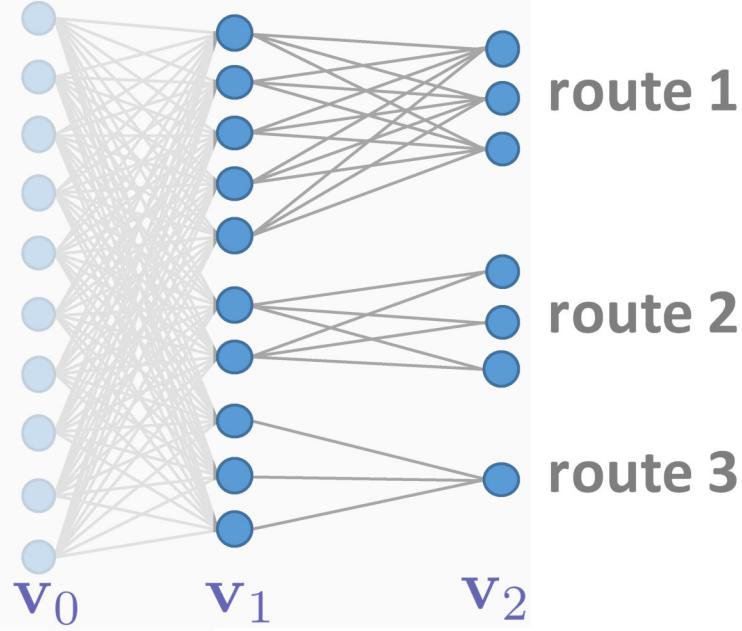
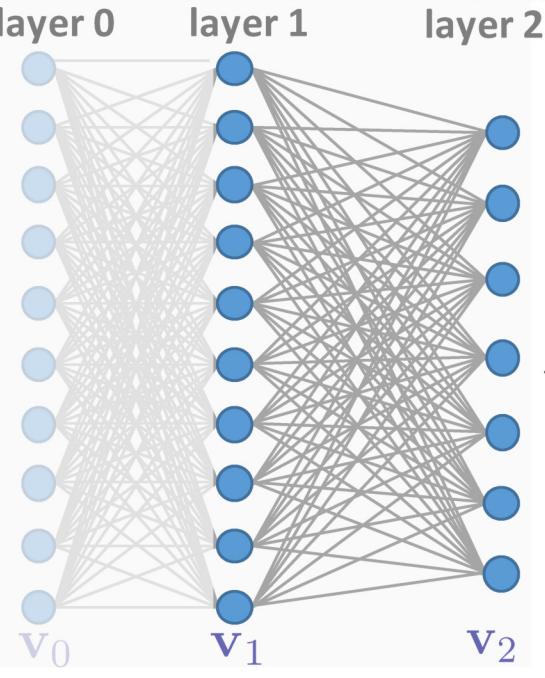


## Equivalent 2-layer perceptron



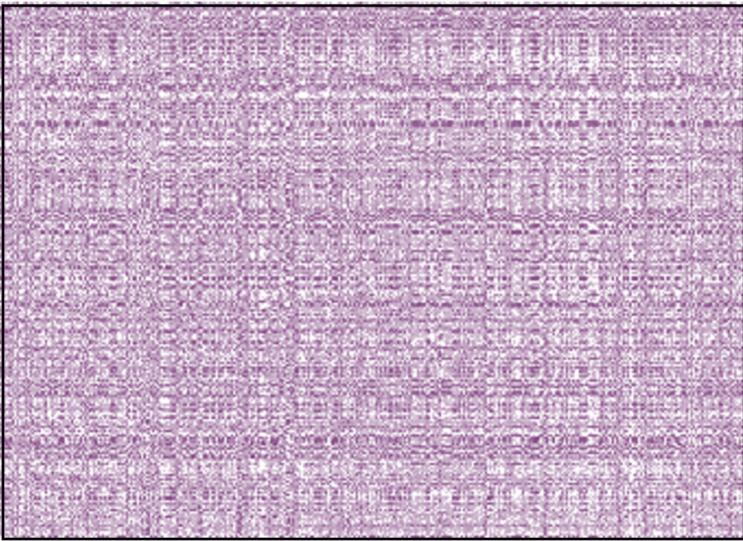
**a**

**b**

**d Routed perceptron****a Two-layer perceptron**

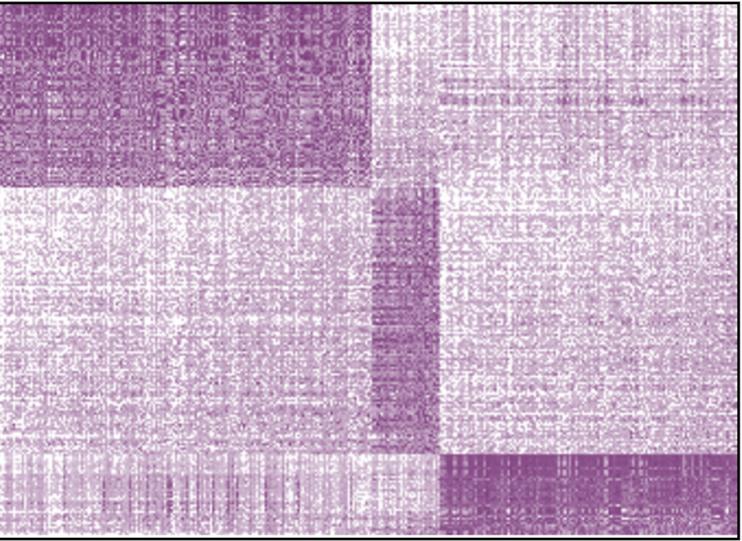
activations in layer 2

$$\Lambda_{12} = \text{acts. layer 1}$$

**b**

activations in layer 2

$$\Lambda'_{12} = \text{acts. layer 1}$$

**c**

$$\Lambda_{12}^{routed}$$

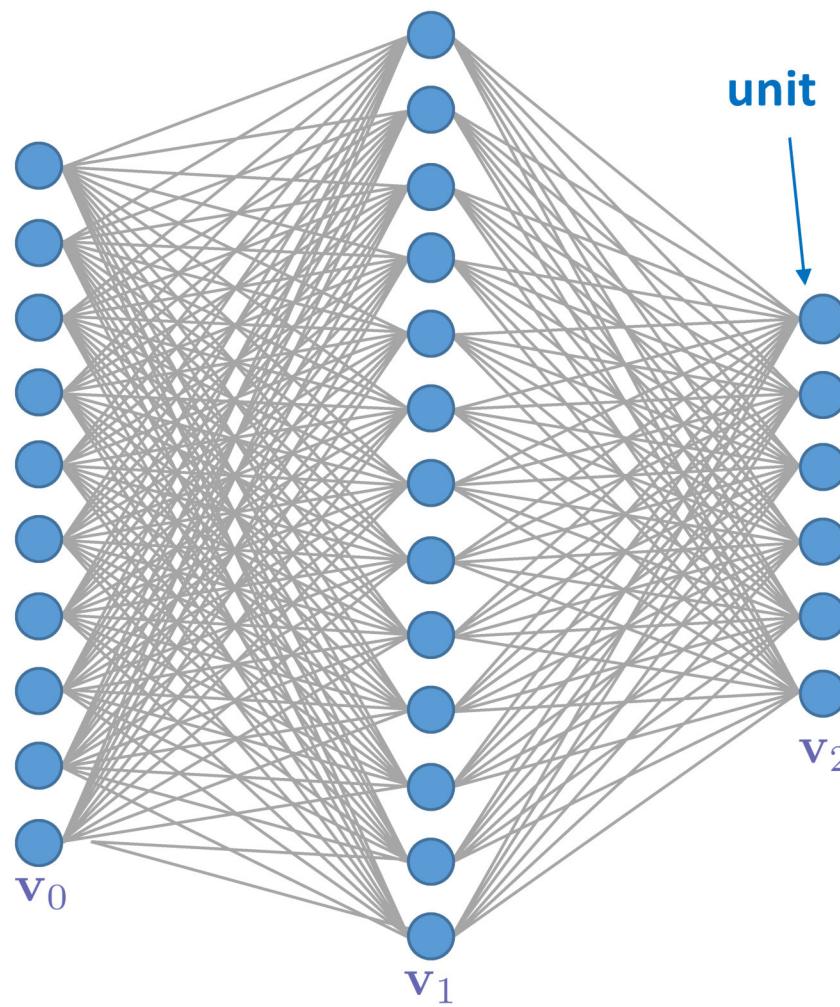
**e**

activations in layer 2

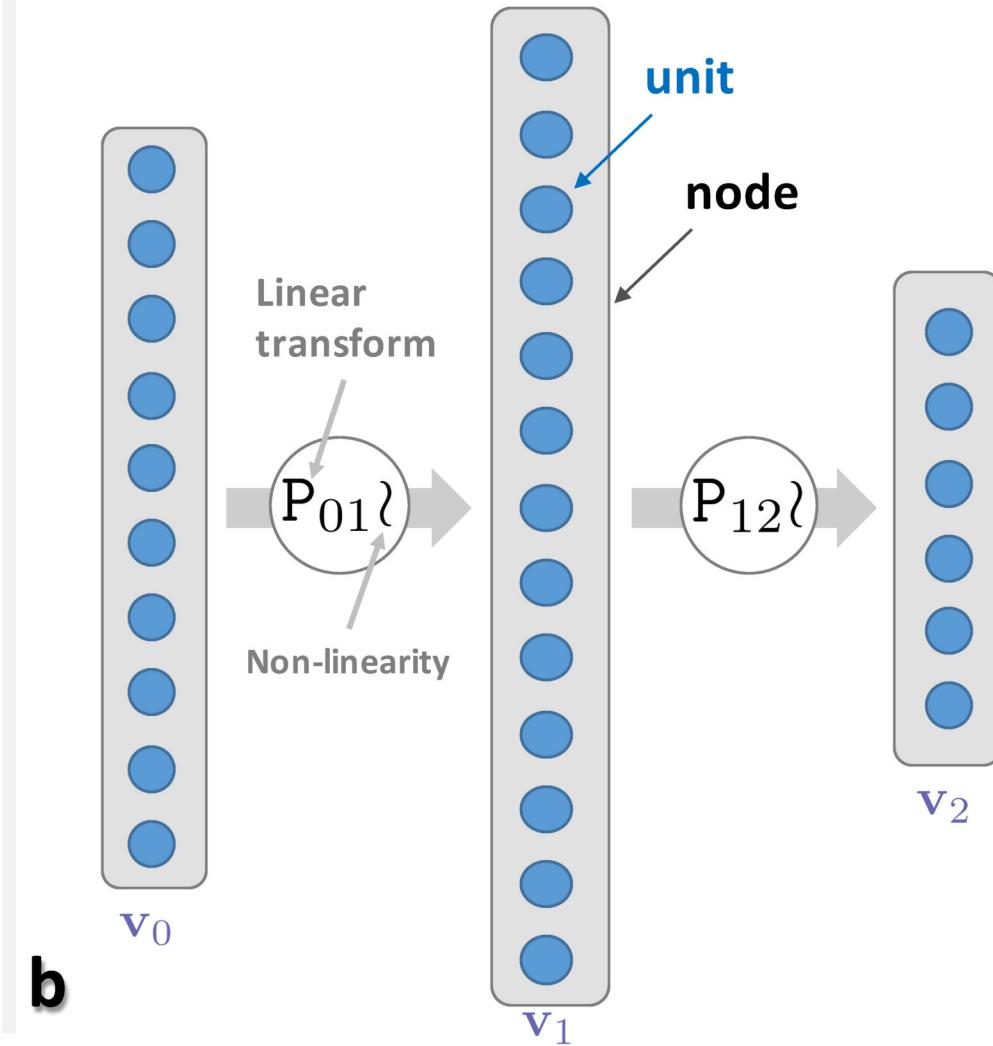
$$= \text{acts. layer 1}$$



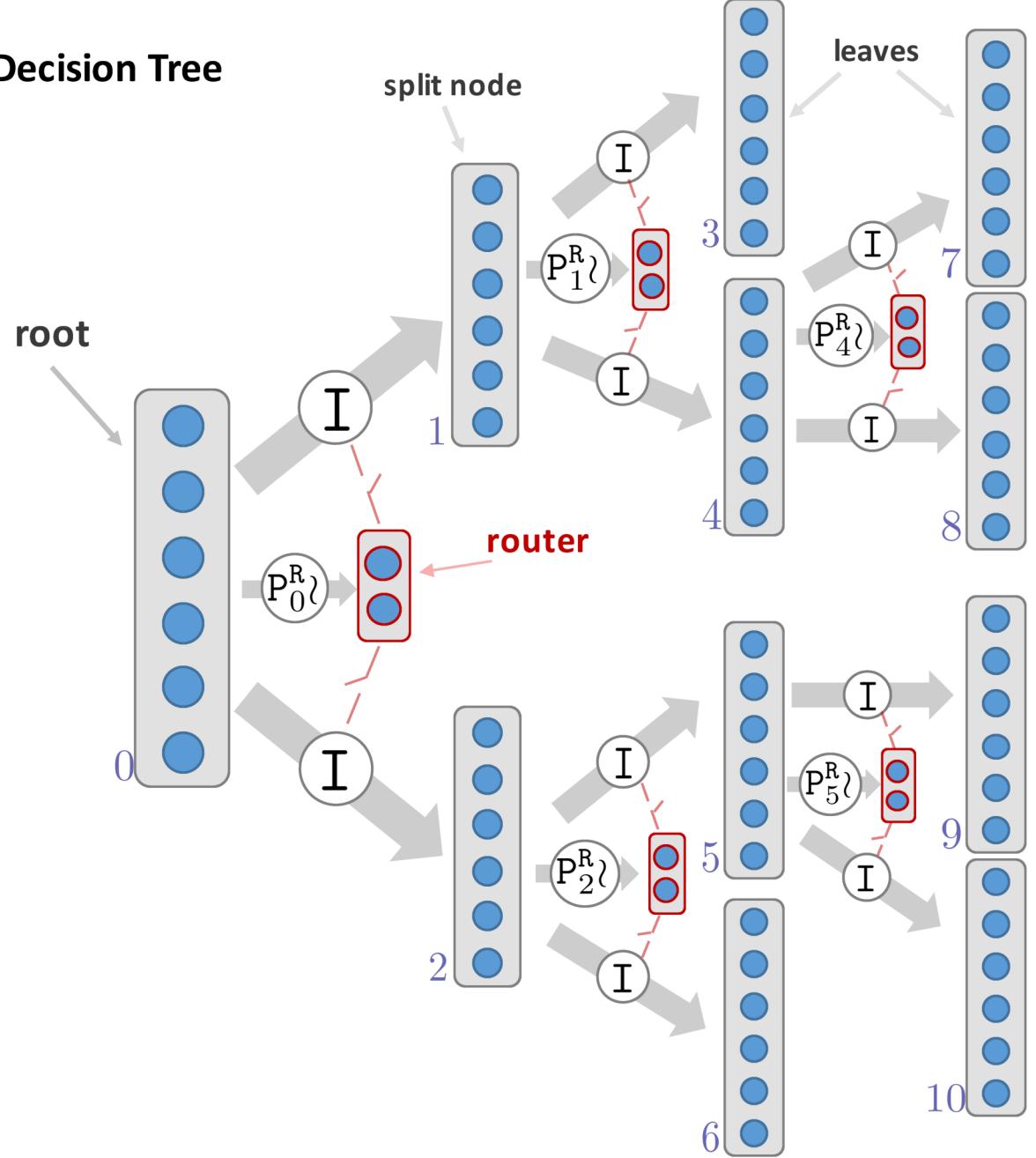
## Old notation for a 2-layer perceptron

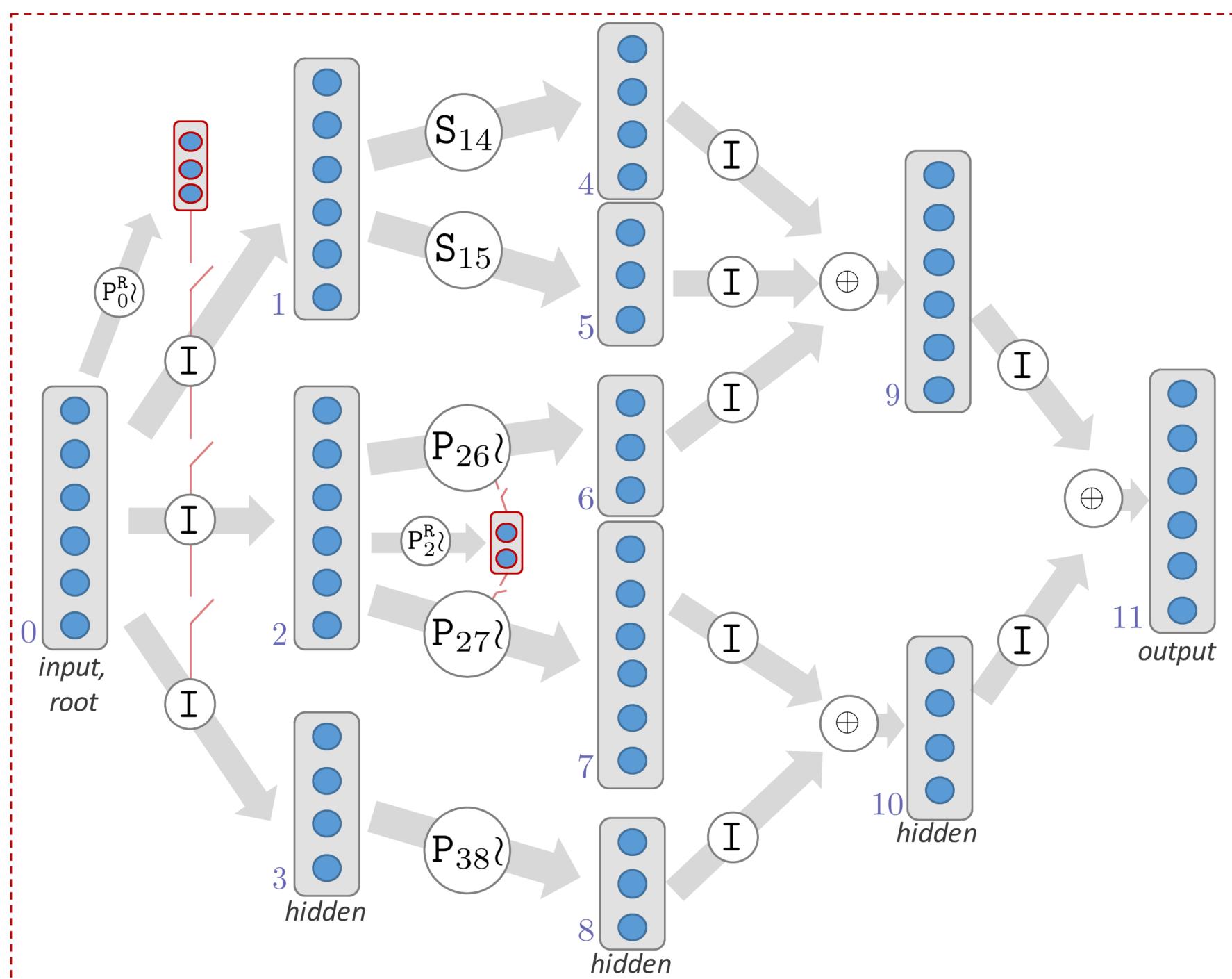


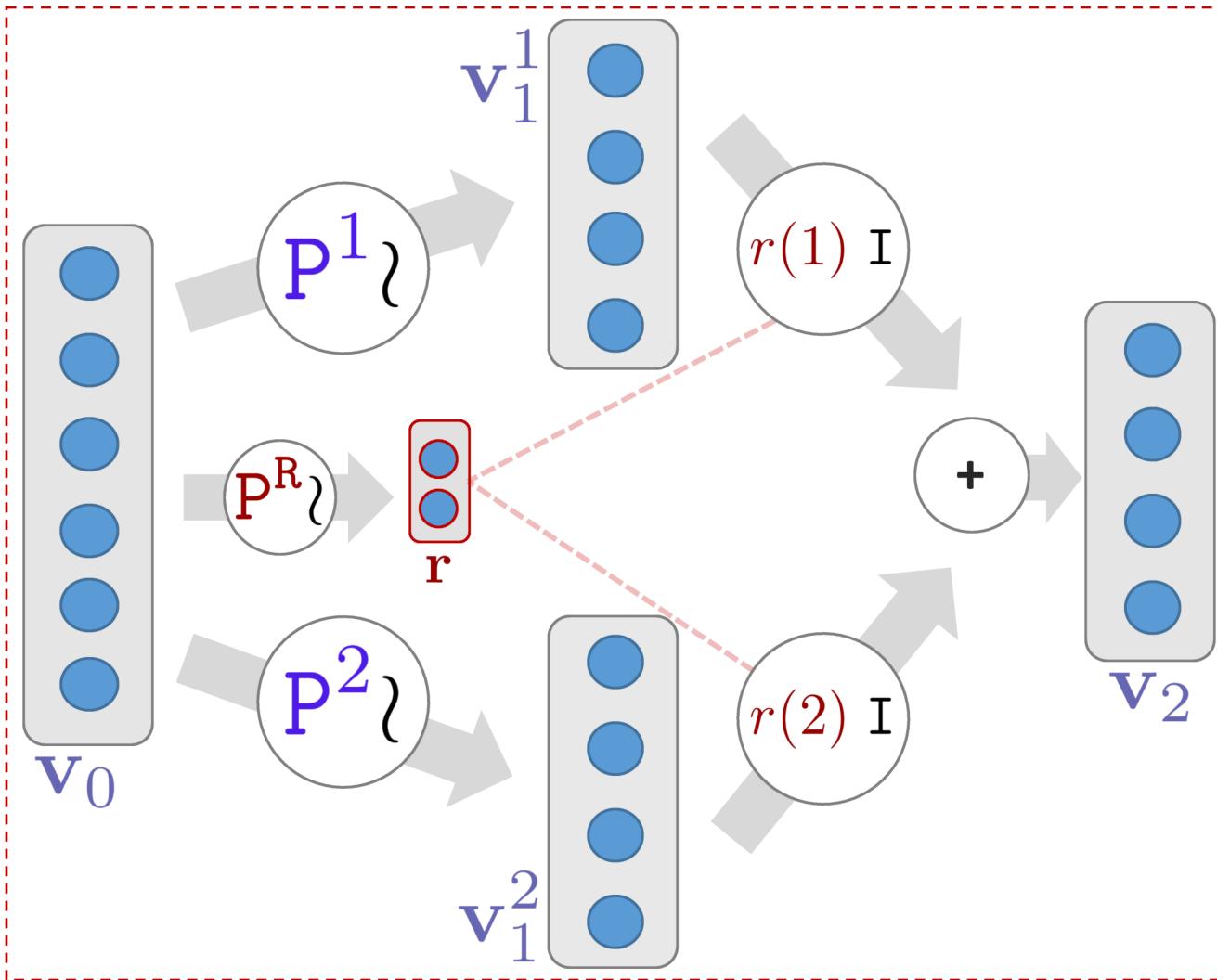
## New notation



## Decision Tree







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

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

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

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

$$\mathbf{r} = \sigma(\mathbf{R} \mathbf{x}) \quad \text{Soft routing weights}$$

$$\Delta \theta_{t+1} := -\rho \frac{\partial E}{\partial \theta} \Big|_t \quad \text{The parameter update rule}$$

$$\theta := \{\mathbf{R}, \{\mathbf{P}^j\}\} \quad \text{The parameters to be optimized}$$

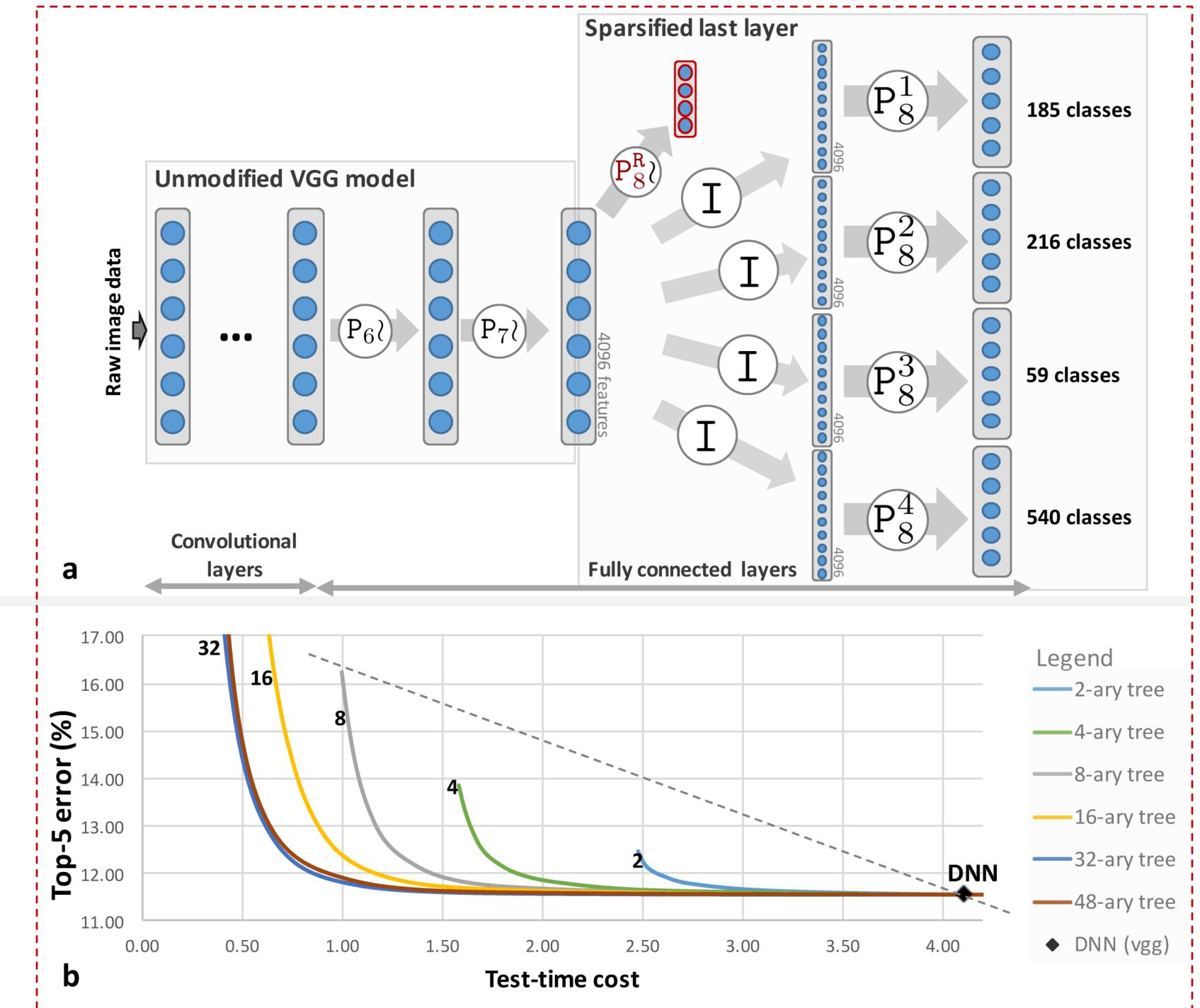
$$\mathcal{T} = \{(\mathbf{x}_1, \mathbf{y}_1^*), \dots, (\mathbf{x}_i, \mathbf{y}_i^*), \dots, (\mathbf{x}_N, \mathbf{y}_N^*)\} \quad \text{The labelled training set}$$

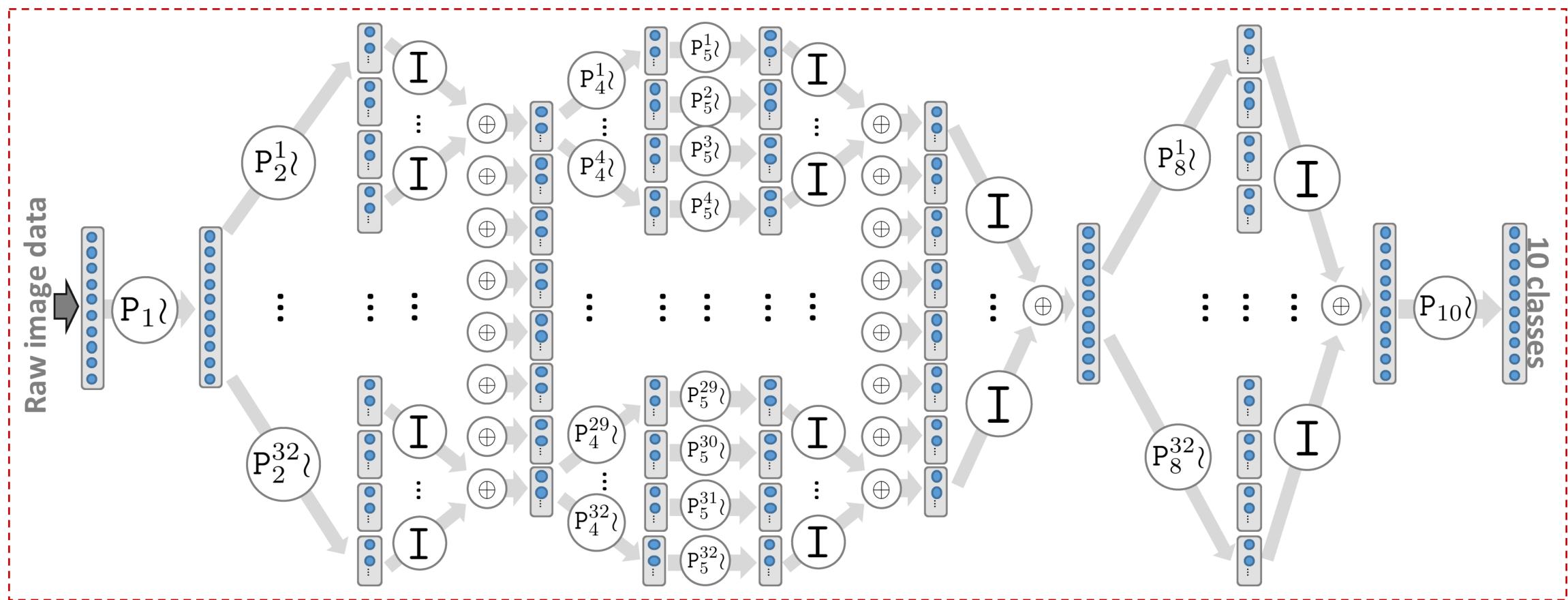
Chain rule to compute partial derivatives for gradient descent

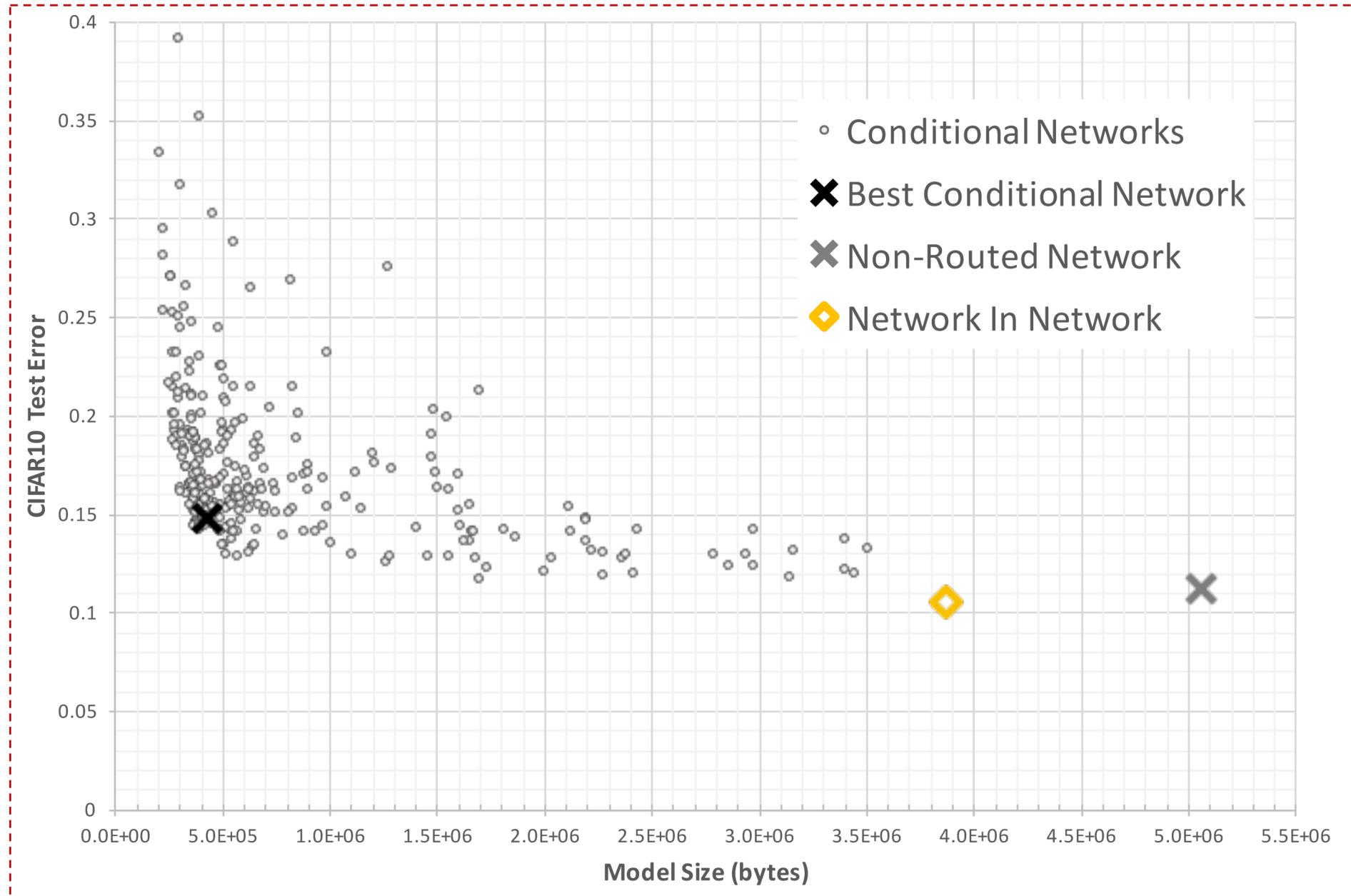
$$\phi^j := \mathbf{P}^j \mathbf{x} \quad \mathbf{y}^j = \sigma(\phi^j)$$

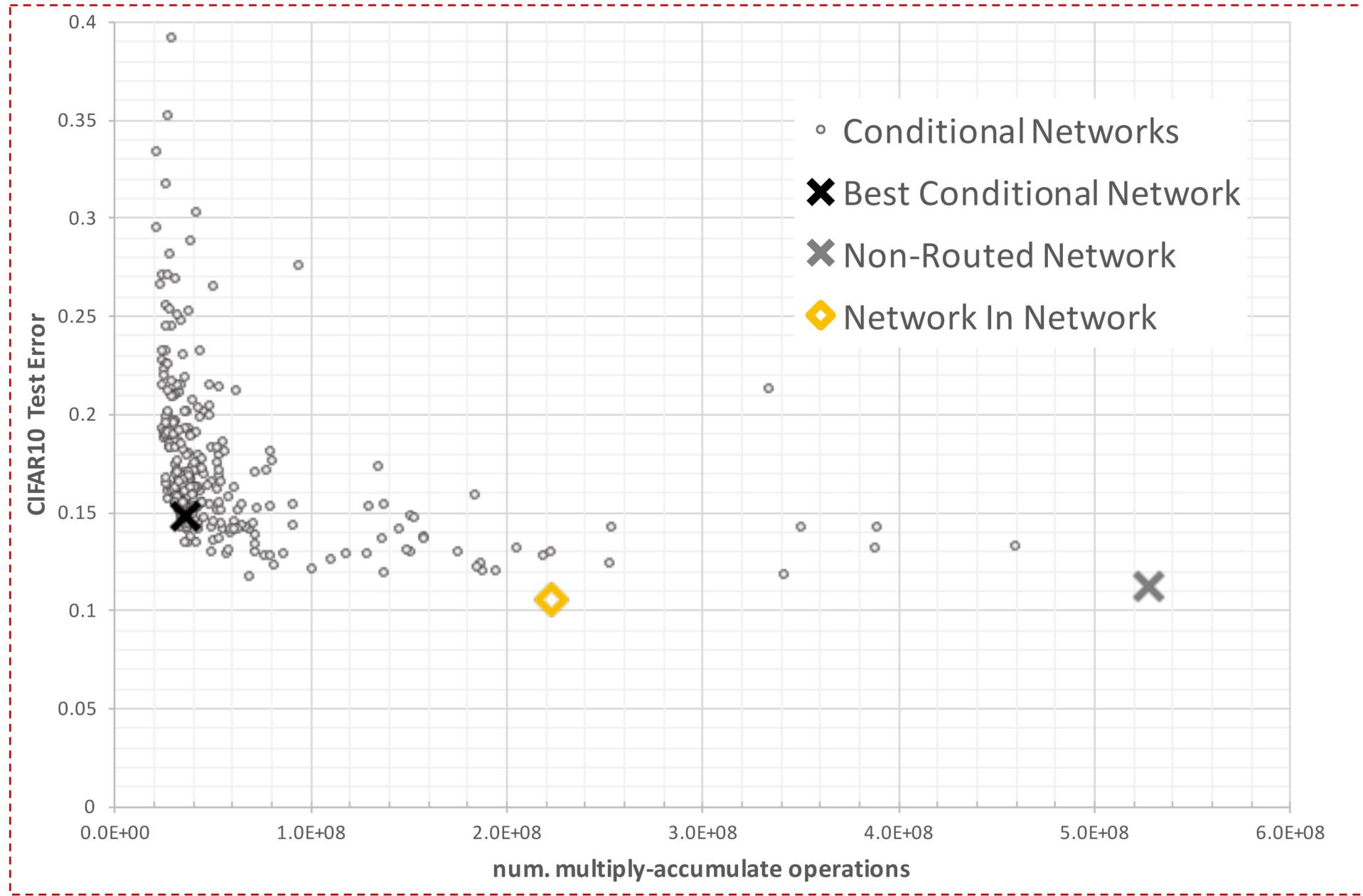
$$\frac{\partial E}{\partial \theta} = \frac{\partial E}{\partial \mathbf{y}} \frac{\partial \mathbf{y}}{\partial \theta} = \frac{\partial E}{\partial \mathbf{y}} \left( \frac{\partial \mathbf{r}}{\partial \mathbf{R}} \mathbf{Y} + \sum_j \mathbf{r}(j) \frac{\partial \mathbf{y}^j}{\partial \phi^j} \frac{\partial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

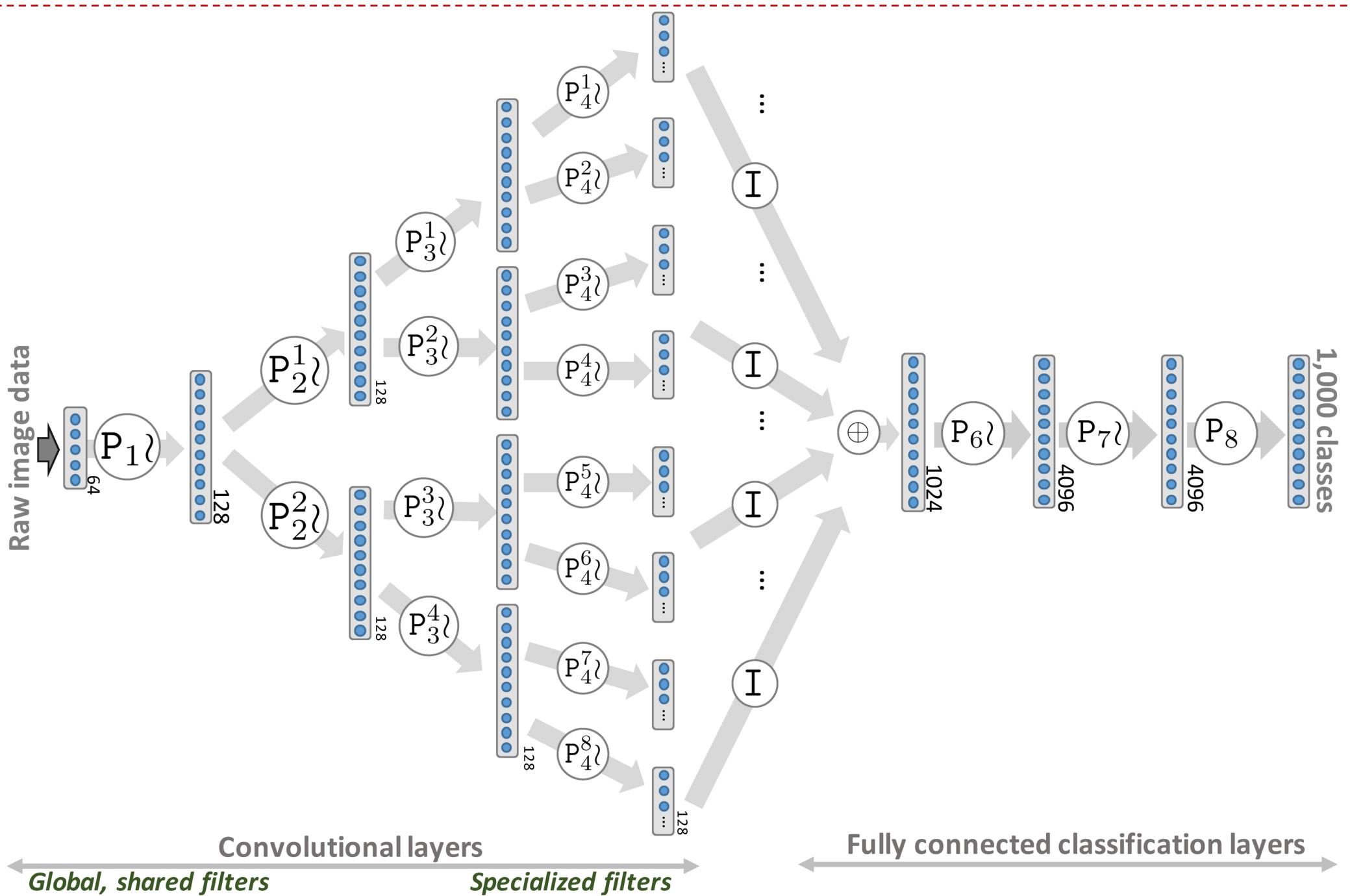
Routing weights influence the back-propagating errors differently for different routes



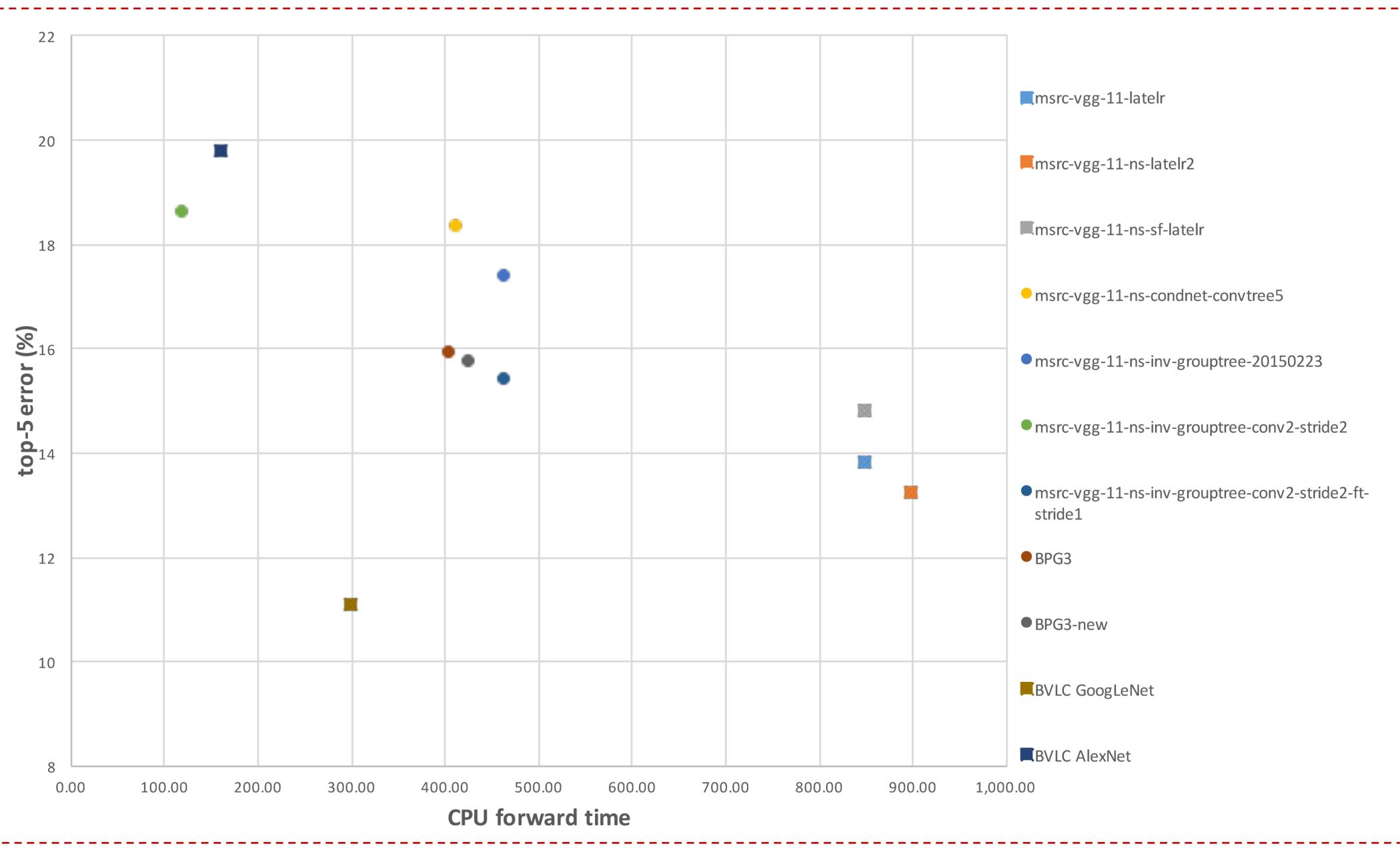




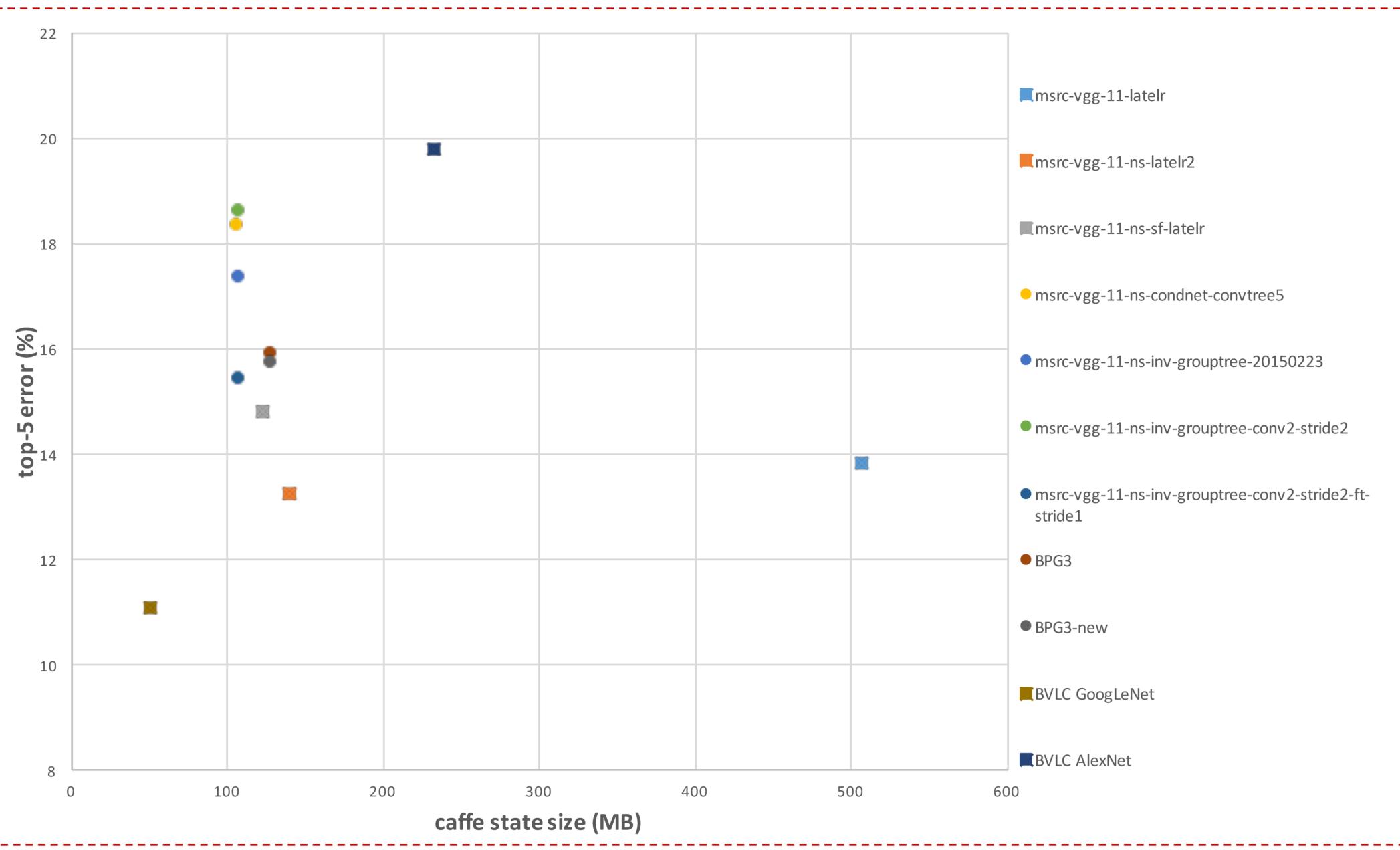


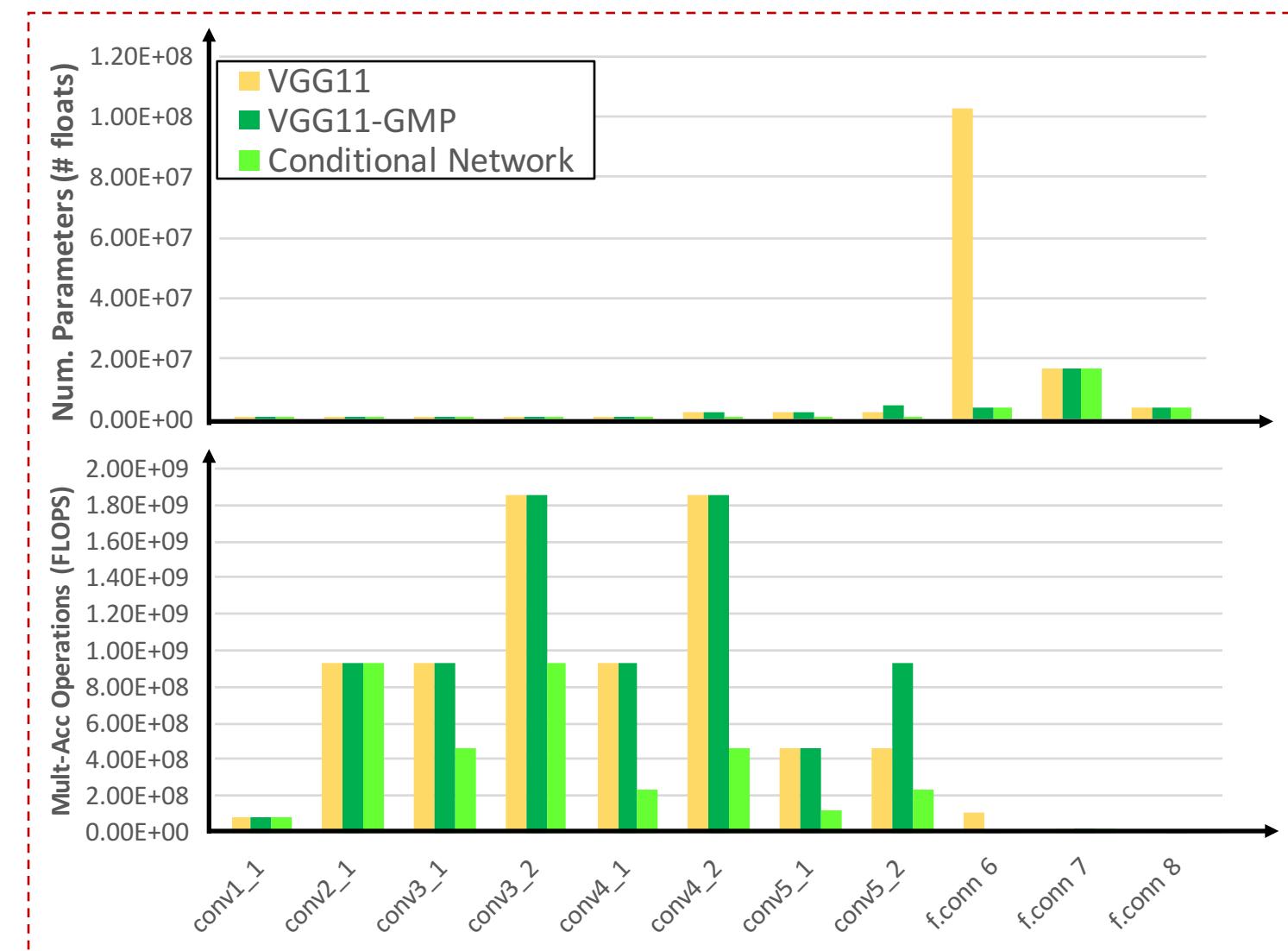


# ImageNet

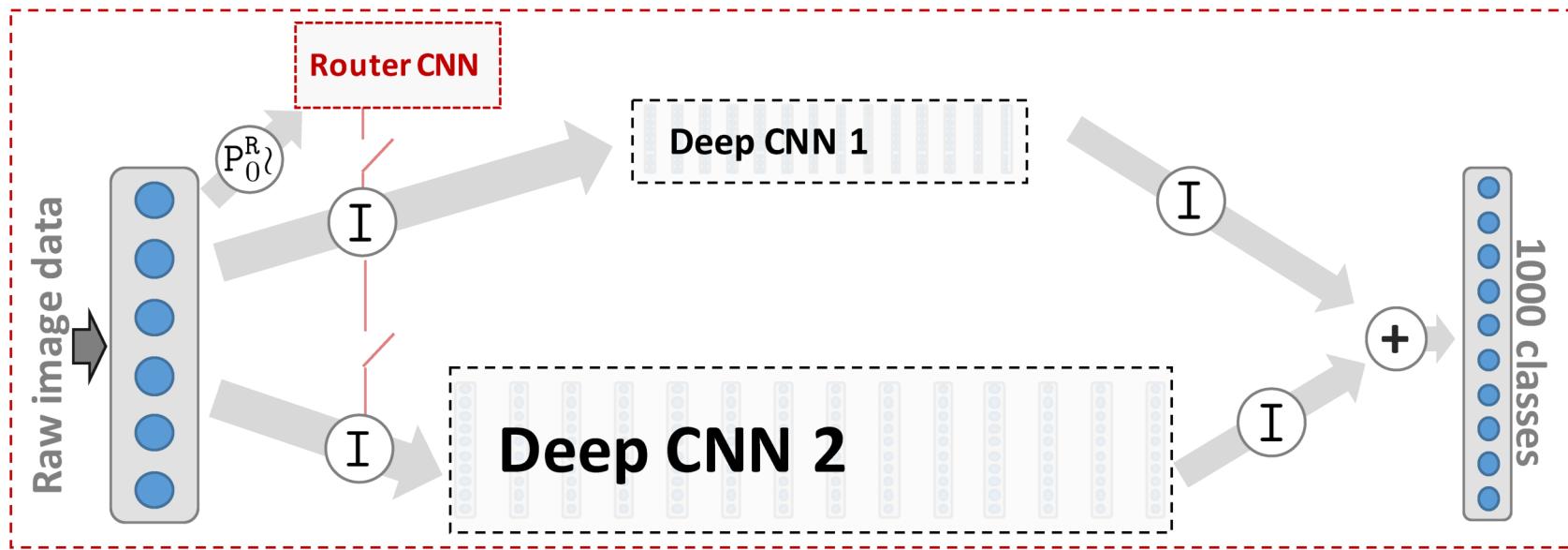


# ImageNet

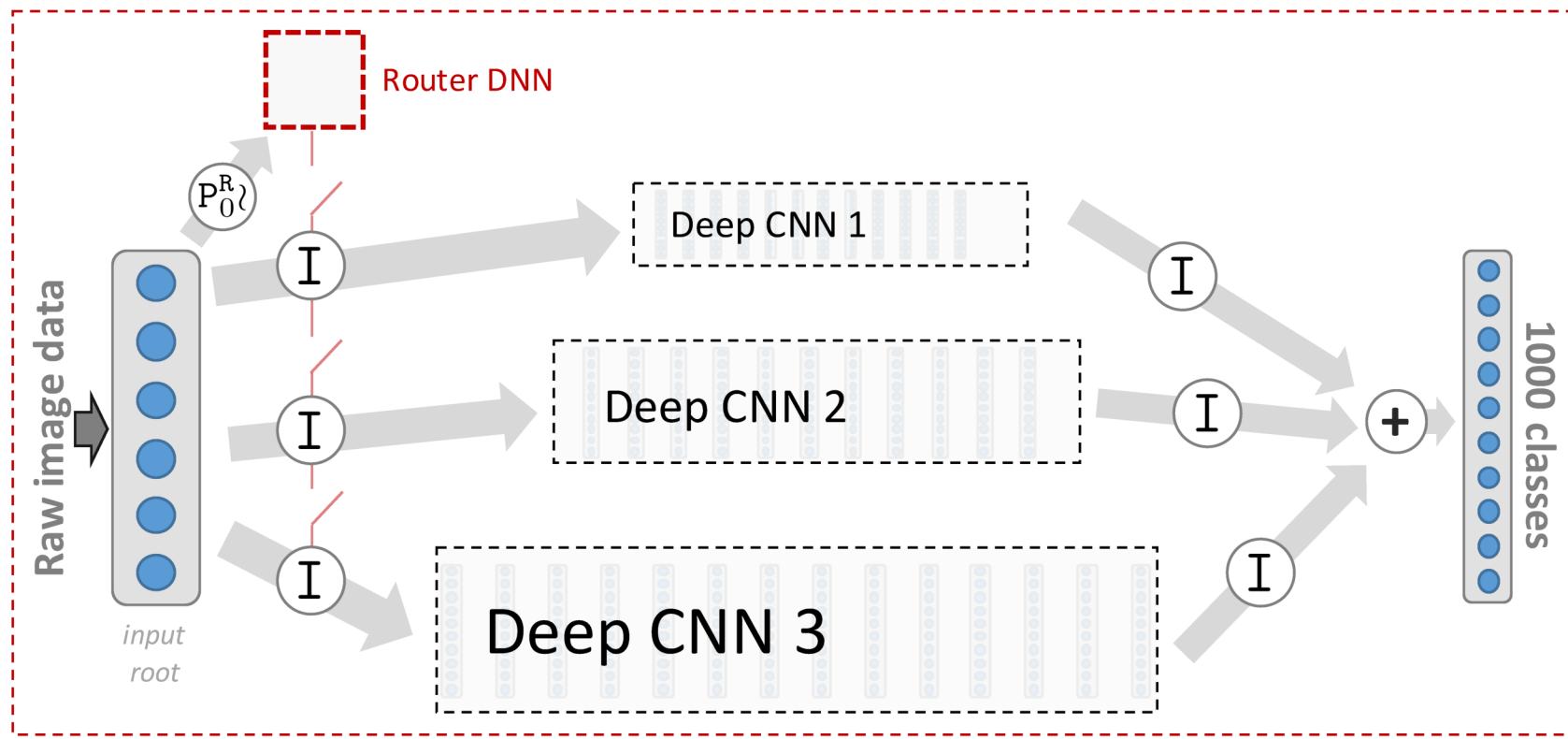




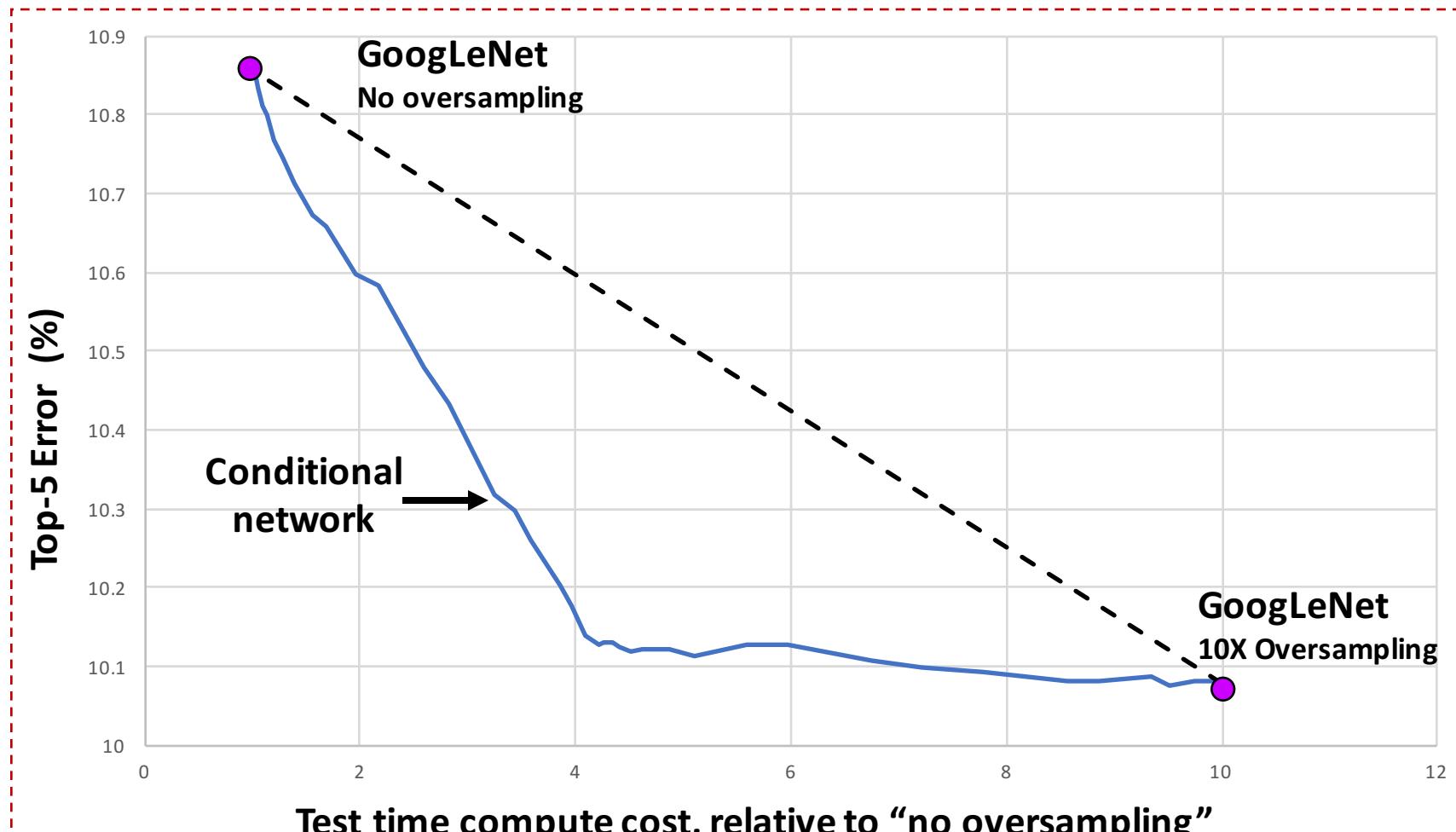
# Mixing



# Mixing



# Mixing



OLD unused

