Learning to Defer with Early-Exit Neural Networks

Mark Coates

Department of Electrical and Computer Engineering

McGill University

McGill Networks Research Laboratory

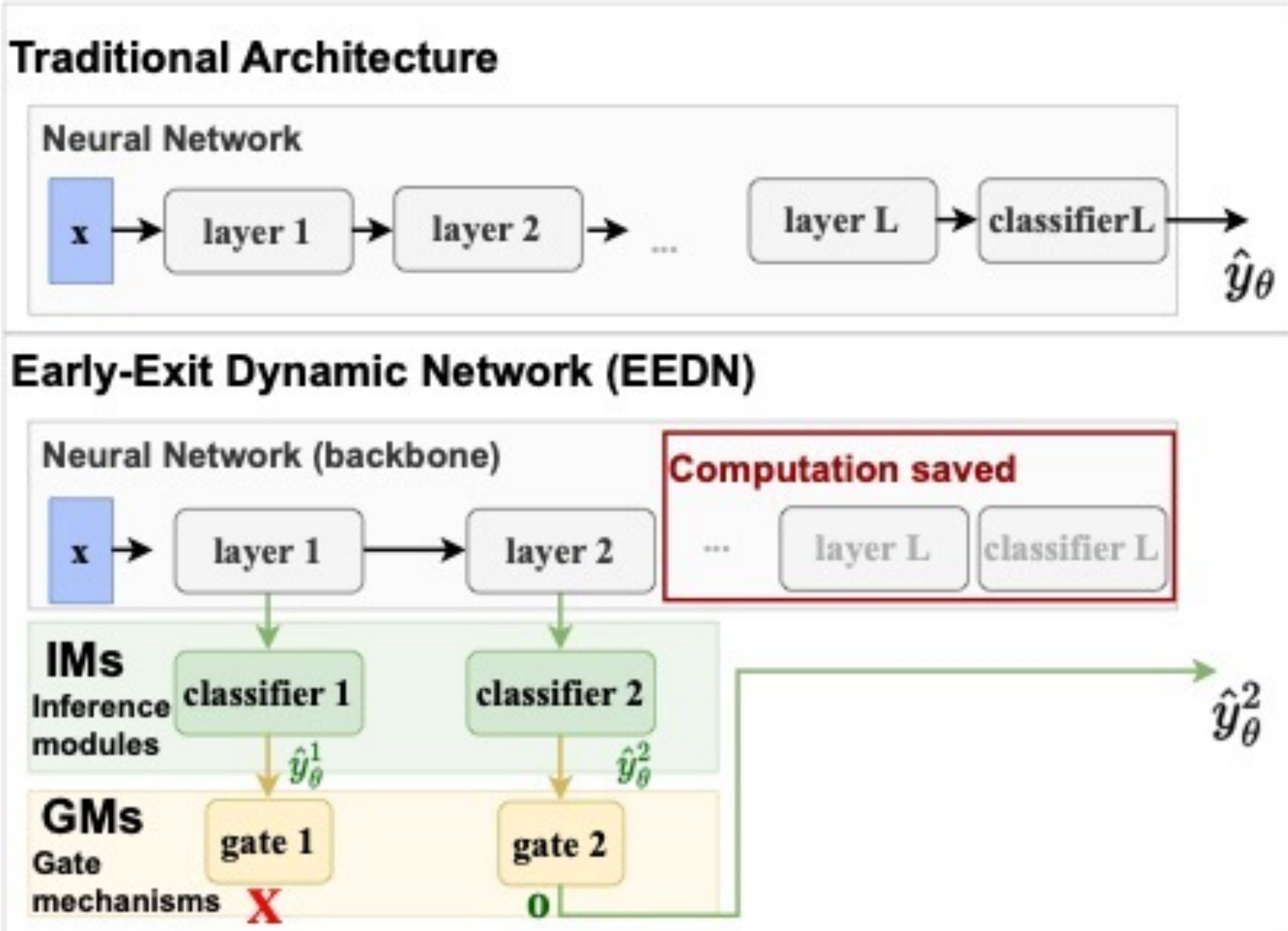
McGill

ILLS International Laboratory on Learning Systems

Mila

Joint work with Florence Regol (Ph.D. student) and Joud Chataoui (M.Sc. Student)

F. Regol, J. Chataoui, and M. Coates, "Jointly Learning to Exit and Infer in Dynamic Neural Networks: JEI-DNN," to appear, ICLR 2024.

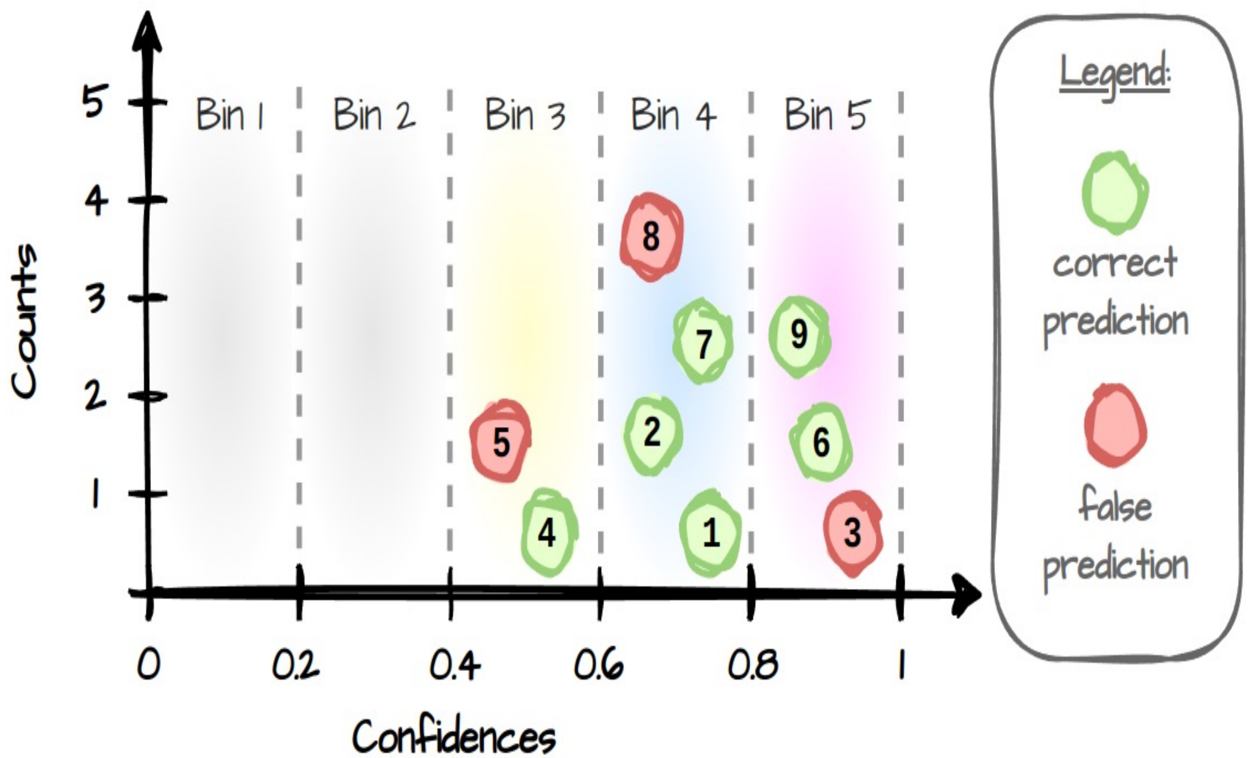

Traditional Architecture
Neural Network
x
layer 1
layer 2
...
layer L
classifierL
$\hat{y}_\theta$
Early-Exit Dynamic Network (EEDN)
Neural Network (backbone)
Computation saved
x
layer 1
layer 2
...
layer L
classifier L
IMs
Inference modules
classifier 1
classifier 2
$\hat{y}_\theta^1$
$\hat{y}_\theta^2$
$\hat{y}_\theta^2$
GMs
Gate mechanisms
gate 1
gate 2
X
o

Previous Approaches

- Threshold-based Gate Mechanisms
  - Confidence score thresholds to decide whether to exit
  - BranchyNet[1], MSDNet[2], RANet[3], CF-ViT[4], Dynamic Perceiver[5]
- Frozen Backbones and IMs + Learnable Gate Mechanisms
  - EPNet[6], PTEENet[7], EENet[8]
- Adaptive IM training
  - BoostedNet[9], L2W[10]

1: Teerapittayanon et al., ICPR 2016
2: Huang et al., ICLR 2018
3: Yang et al., CVPR 2020
4: Chen et al., AAAI 2023
5: Han et al., ICCV 2023
6: Dai et al., ICPR 2020
7: Lahiany et al., ICLR 2018
8: Ilhan et al., CVPR 2023
9: Yu et al., ICPR 2023
10: Han et al., ICLR 2022

Expected Calibration Error: $$ECE = \sum_{m=1}^{M} \frac{|B_m|}{n} |acc(B_m) - conf(B_m)|$$



Bins B_m

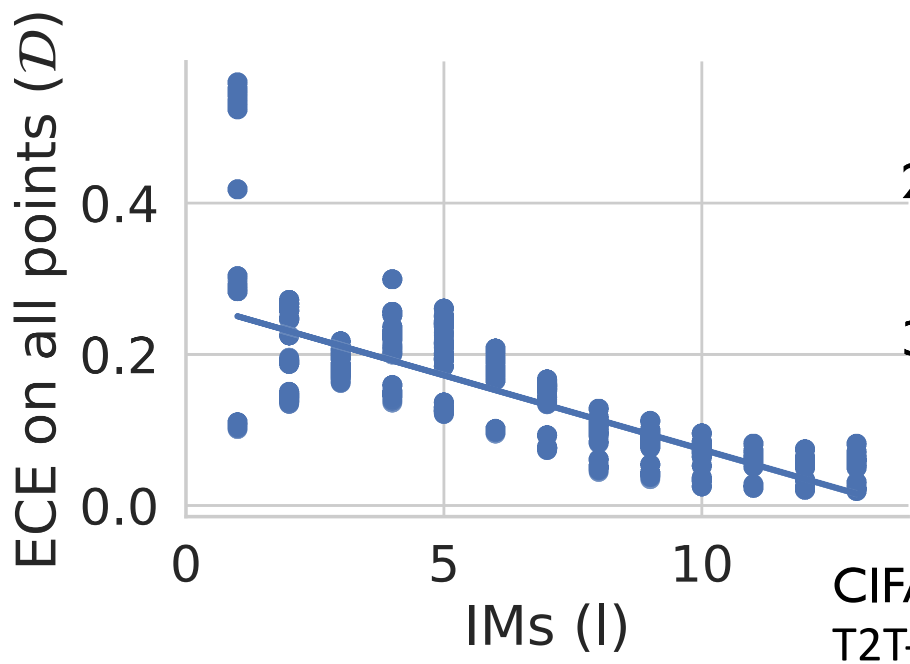
$conf(B_m)$: avg. pred. prob in bin m

$acc(B_m)$: accuracy in bin m

n: total number of points

Maja Pavlovic, Expected Calibration Error (ECE): A Step-by-Step Visual Explanation, Towards Data Science, Jul. 2023.

$$\text{gate } l = \begin{cases} earlyexit & \text{if } \max_k \hat{\mathbf{p}}_k^l > \eta \\ continue & \text{o.w.} \end{cases}, \hat{\mathbf{p}}^l = \text{predicted prob. IM } l$$



Inference Module (layer)

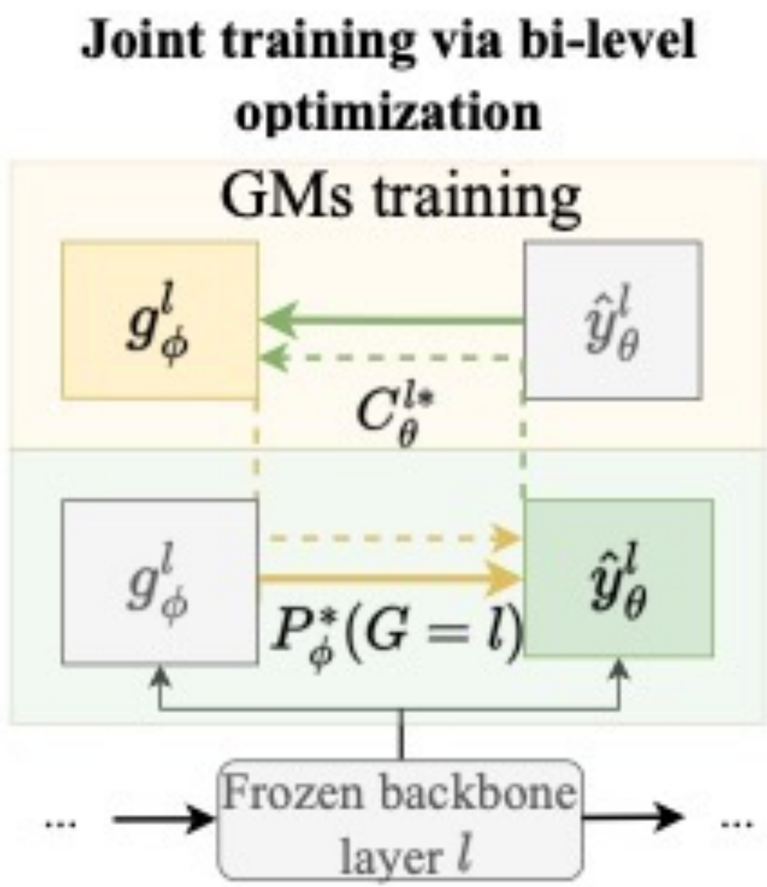
Problems:

1. IMs and GMs disconnected during training → training-test mismatch
2. Uncertainty information is consumed by thresholding → harder to calibrate
3. Relies on well-calibrated classifiers

CIFAR100

T2T-ViT-14 (Yuan et al., ICCV 2021)

Our Solution: JEI-DNN



- Many lightweight gates and IMs
- Jointly learn the gate mechanisms and IMs
- Model the gate output $P_\phi(G = l|\mathbf{x}_i)$

$$
\begin{aligned}
C_\theta^{*l} &= \text{Cost of choosing IM } l, \\
P_\phi^*(G = l) &= \text{Prob. of choosing IM } l, \\
g_\phi^l &= \text{Parameter of } P_\phi(G), \\
\hat{y}_\theta^l &= \text{Prediction of IM } l.
\end{aligned}
$$

Modelling the gate variables

- Only need to evaluate $P_\phi(G = l|\mathbf{x}_i)$ if dynamic evaluation reaches layer l
- Can use any intermediate values calculated by gates, IMs, and base architecture
- Denote aggregate information $\mathbf{c}_i^{\leq l}$
- Construct an additive model:

$$P_\phi(G = 1|\mathbf{x}_i) = g_\phi^1(\mathbf{c}_i^{\leq 1}),$$

$$P_\phi(G = l|\mathbf{x}_i) = \min(g_\phi^l(\mathbf{c}_i^{\leq l}), 1 - \sum_{j=1}^{l-1} g_\phi^j(\mathbf{c}_i^{\leq j})) \quad \text{for} \quad l = 2, \ldots, L\,.$$

Loss of JEI-DNN

- Combination of a cross entropy loss (accuracy of prediction) + inference cost (fixed cost per layer)

$$\mathcal{L} = \mathbb{E}_{Y,X}\mathbb{E}_{G|X}[\mathcal{L}^{CE}(Y, \hat{\mathbf{p}}_\theta^{G|X}(X)) + \lambda I C^{G|X}].$$

- Approximated loss

Prob. Exiting at Gate l

$$\mathcal{L} \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L} \boxed{(\mathcal{L}^{CE}(y_i, \hat{\mathbf{p}}_\theta^{l}(\mathbf{x}_i)) + \lambda I C^{l}_{\text{norm}})} \boxed{\min\left(g_\phi^{l}(\mathbf{c}_i^{\leq l}), 1 - \sum_{j=1}^{l-1} g_\phi^{j}(\mathbf{c}_i^{\leq j})\right)}.$$

Cost of Exiting at Gate l: Inference + Accuracy

Optimization

- Optimization is challenging because of the min operator
- Bi-level optimization
- Let $C^l_{\theta(\phi)} = \mathcal{L}^{CE}(y_i, \hat{\mathbf{p}}^l_\theta(\mathbf{x}_i)) + \lambda IC^l_{\text{norm}}$

$$\phi^* = \arg\min_\phi \mathcal{L}^{out} \triangleq \arg\min_\phi \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L} C^l_{\theta^*(\phi)} P_\phi(G = l|\mathbf{x}_i),$$

Outer: gate parameters

$$s.t. \quad \theta^*(\phi) = \arg\min_\theta \mathcal{L}^{in} \triangleq \arg\min_\theta \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}(\mathcal{L}^{CE}(y_i, \hat{\mathbf{p}}^l_\theta(\mathbf{x}_i)) + \lambda IC^l_{\text{norm}})P_\phi(G = l|\mathbf{x}_i).$$

Inner: IM parameters

Optimizing the IMs

- The gate probabilities take $\mathbf{c}_i^{\leq l}$ as input.
- These can depend on θ. Make the dependence explicit by writing

$$P_\phi(G = l|\mathbf{x}_i) = G_\phi(m(\theta, \mathbf{x}_i)).$$

- Then:

$$\frac{\partial \mathcal{L}^{in}}{\partial \theta} = \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}\frac{\partial \mathcal{L}^{CE}(y_i, \hat{\mathbf{p}}_\theta^l(\mathbf{x}_i))}{\partial \theta} P_\phi(G = l|\mathbf{x}_i) + \frac{\partial G_\phi(m(\theta, \mathbf{x}_i))}{\partial \theta} C_{\theta(\phi)}^l \,,$$

Optimizing the IMs

$$\frac{\partial \mathcal{L}^{in}}{\partial \theta} = \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}\frac{\partial \mathcal{L}^{CE}(y_i, \hat{\mathbf{p}}_\theta^l(\mathbf{x}_i))}{\partial \theta} P_\phi(G = l|\mathbf{x}_i) + \frac{\partial G_\phi(m(\theta, \mathbf{x}_i))}{\partial \theta} C_{\theta(\phi)}^l \,,$$

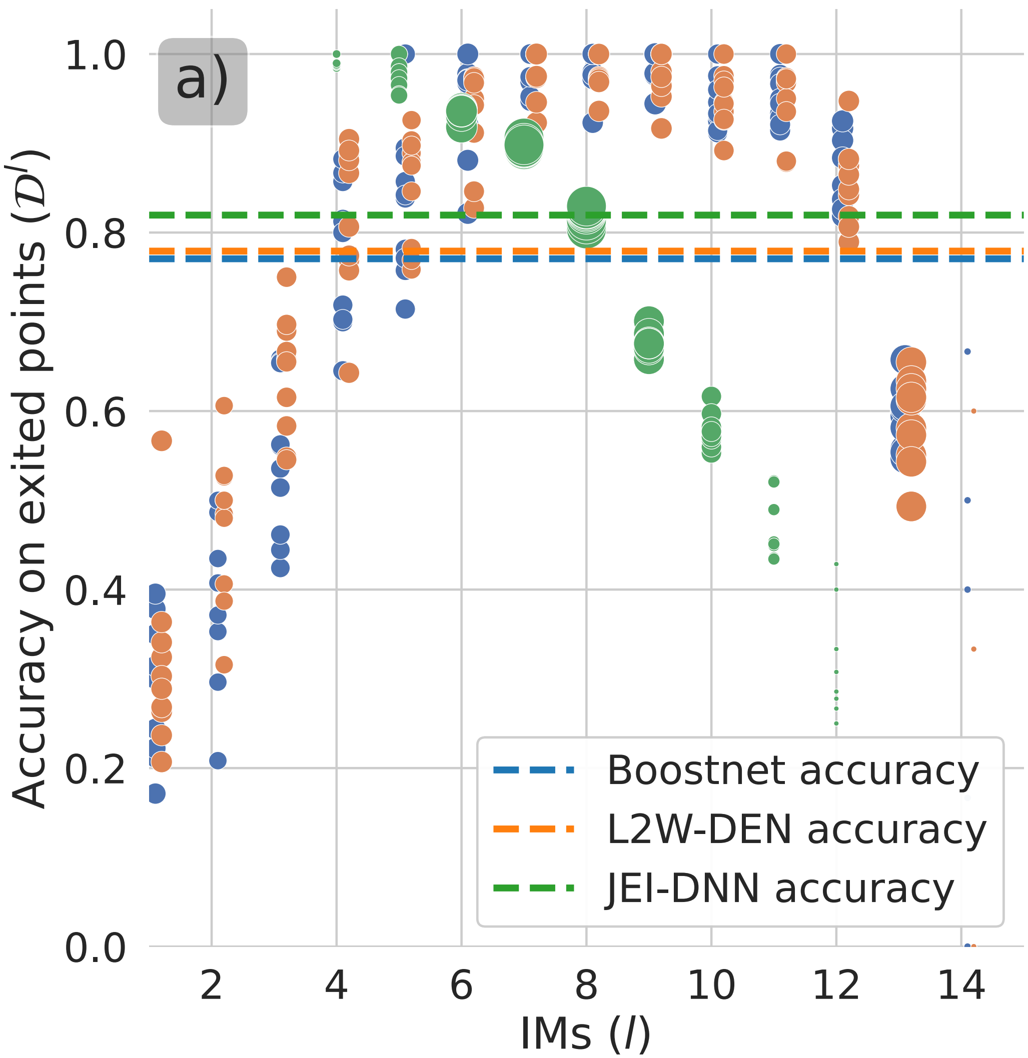
- First term is the gradient corresponding to a weighted cross-entropy loss
- Weights emerge directly from the gating mechanism
- Second term is driven by impact of θ on the gates
- Practical approximation: ignore the second term (represents a secondary effect)

Optimizing the Gates

Hard!

$$\frac{\partial \mathcal{L}^{out}}{\partial \phi} = \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}\frac{\partial P_\phi(G=l|\mathbf{x}_i)}{\partial \phi} C^l_{\theta^*(\phi)} = \frac{1}{N}\sum_{i=1}^{N}\sum_{l=1}^{L}\frac{\partial \min\left(g^l_\phi(\mathbf{c}_i^{\leq l}), 1-\sum_{j=1}^{l-1} g^j_\phi(\mathbf{c}_i^{\leq j})\right)}{\partial \phi} C^l_{\theta^*(\phi)}\,.$$

- Alternative approach: define a surrogate binary classification problem
- Construct binary targets for $g^1_\phi(\mathbf{c}_i^{\leq 1}), \ldots, g^{L-1}_\phi(\mathbf{c}_i^{\leq L-1})$
- Evaluate the cost of each gate $C^l_{\theta^*(\phi)}$ and determine lowest cost $l^* = \arg\min_l C^l_{\theta^*(\phi)}$
- Targets for binary tasks $t^1_i, \ldots, t^L_i$ are $t^j_i = 0 \text{ for } j < l^*$ and $t^j_i = 1 \text{ for } j \geq l^*$
- The surrogate tasks and initial objective share same solution if there exists ϕ such that the gating mechanisms can always select lowest-cost gates.



Exit Behaviour

- JEI-DNN versus state-of-the-art baselines BoostedNet and L2W-DEN

- Higher accuracy on earlier exits
- Lower accuracy on later exits

- Higher average accuracy

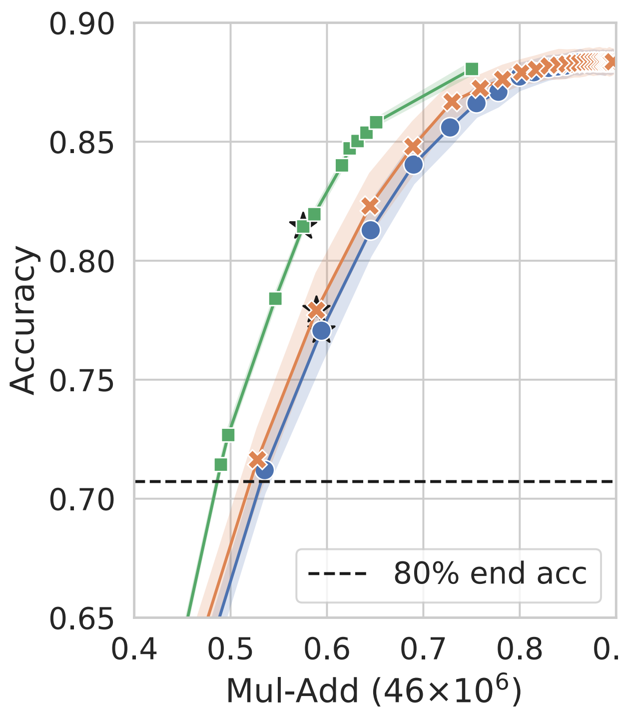
- Concentrates on relatively few IMs

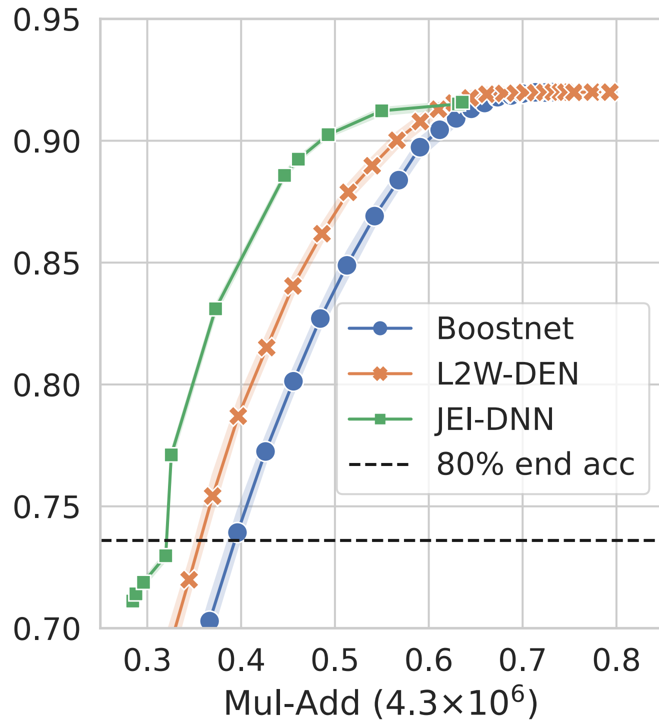
- Only starts to exit at layers 4-5.

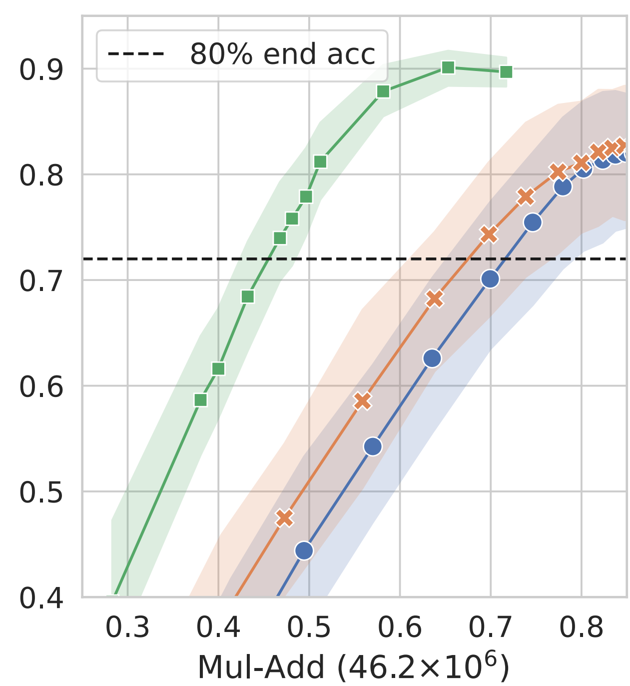
Experiments

- Image classification using vision transformer: T2T-ViT-14 (Yuan et al., ICCV 2021)
- Single layer neural network gates and IMs at all layers
- Gates operate on 4 statistics based on IM output
  - (i) Max. pred. prob, entropy of predictions, entropy of scaled predictions, difference between two most confident predictions

- CIFAR10, CIFAR100: 60,000 32x32 color images with 10/100 classes
- Cropped digit SVHN: 99,289 32x32 color images of house numbers
- CIFAR100-LT: imbalanced classes (100x most common vs least common)
- ImageNet: 1.2 million 244 x 244 images; 1000 classes.

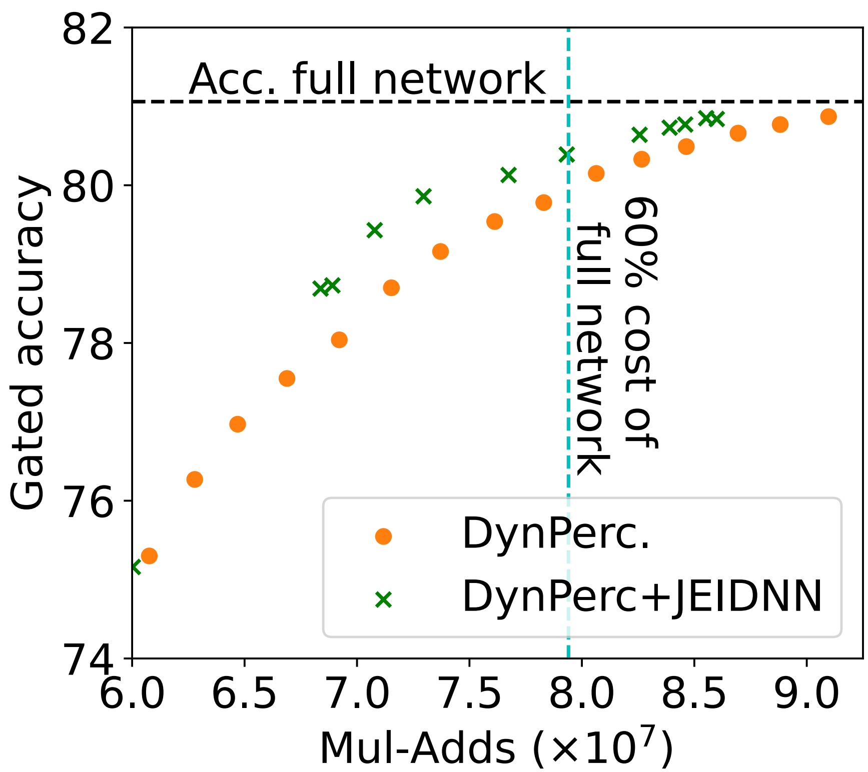




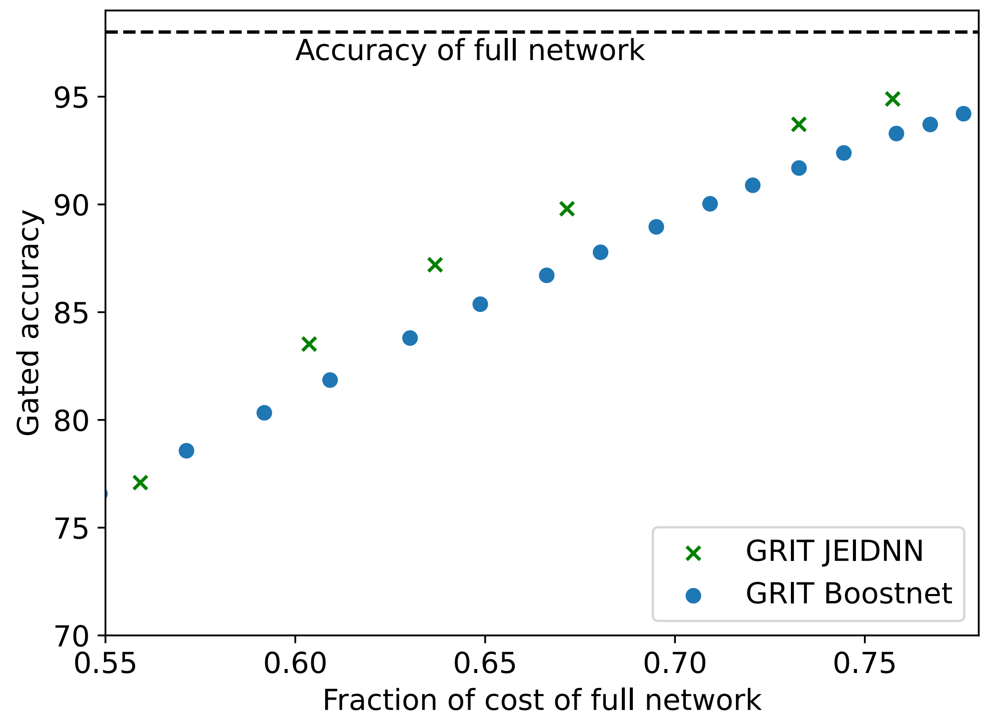


- CIFAR100 with T2T-ViT-14
- SVHN with T2T-ViT-7
- CIFAR100-LT with T2T-ViT-14

Other Architectures



- ImageNet with Dynamic Perceiver



- Super-resolution MNIST with Graph Transformer