LEENet: Learned Early Exit Network

Learning Optimal Early Exit Policy for Efficiency Improvements in DNNs

Austin Chemelli, Anthony Wong, Blake Sanie, Dylan Mace, Dylan Small, Nic Zacharis

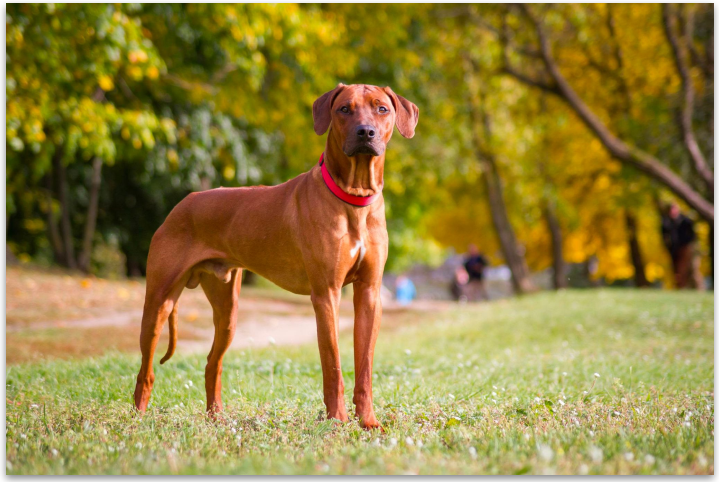
Georgia Tech

Project Motivation & Background

Georgia Tech

Introduction

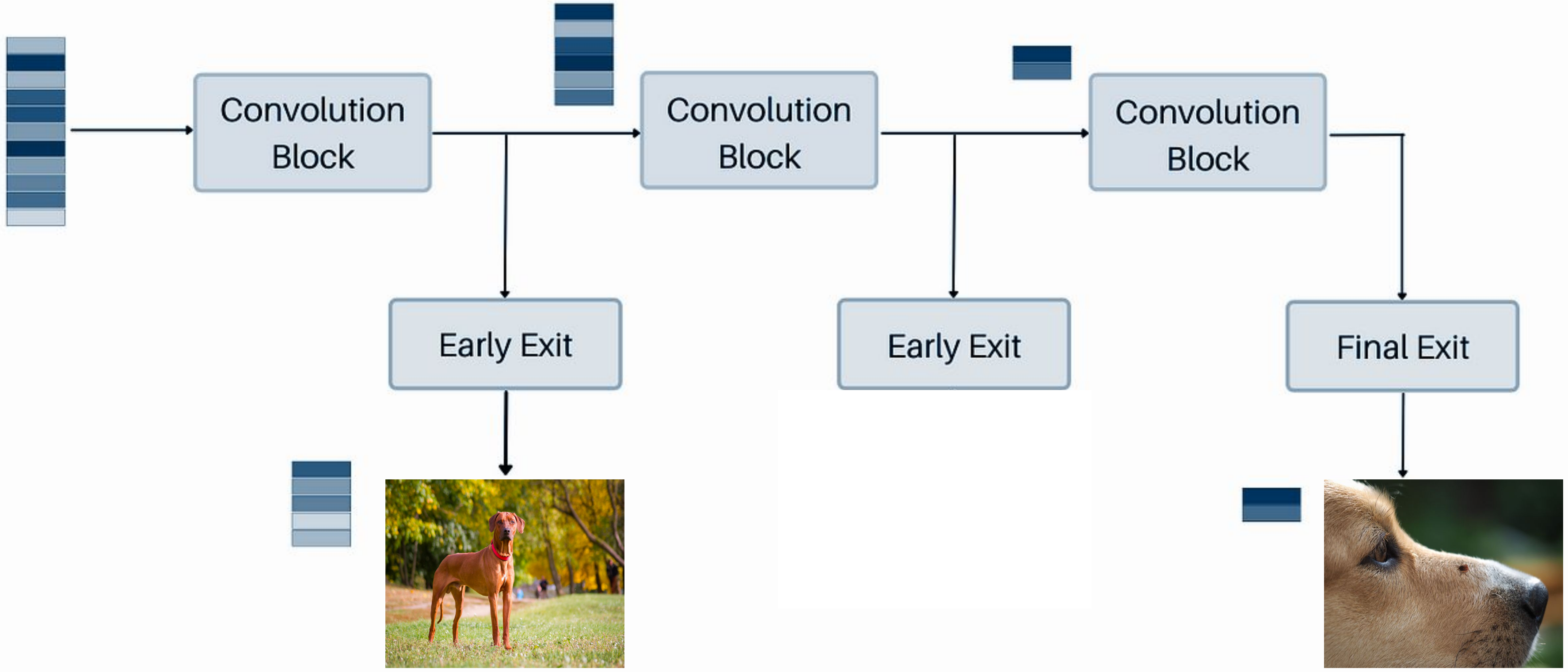
- Suppose we have two images that we want to classify:



- The right image is harder to classify
- Current ML models use the same parameters/computations for all images

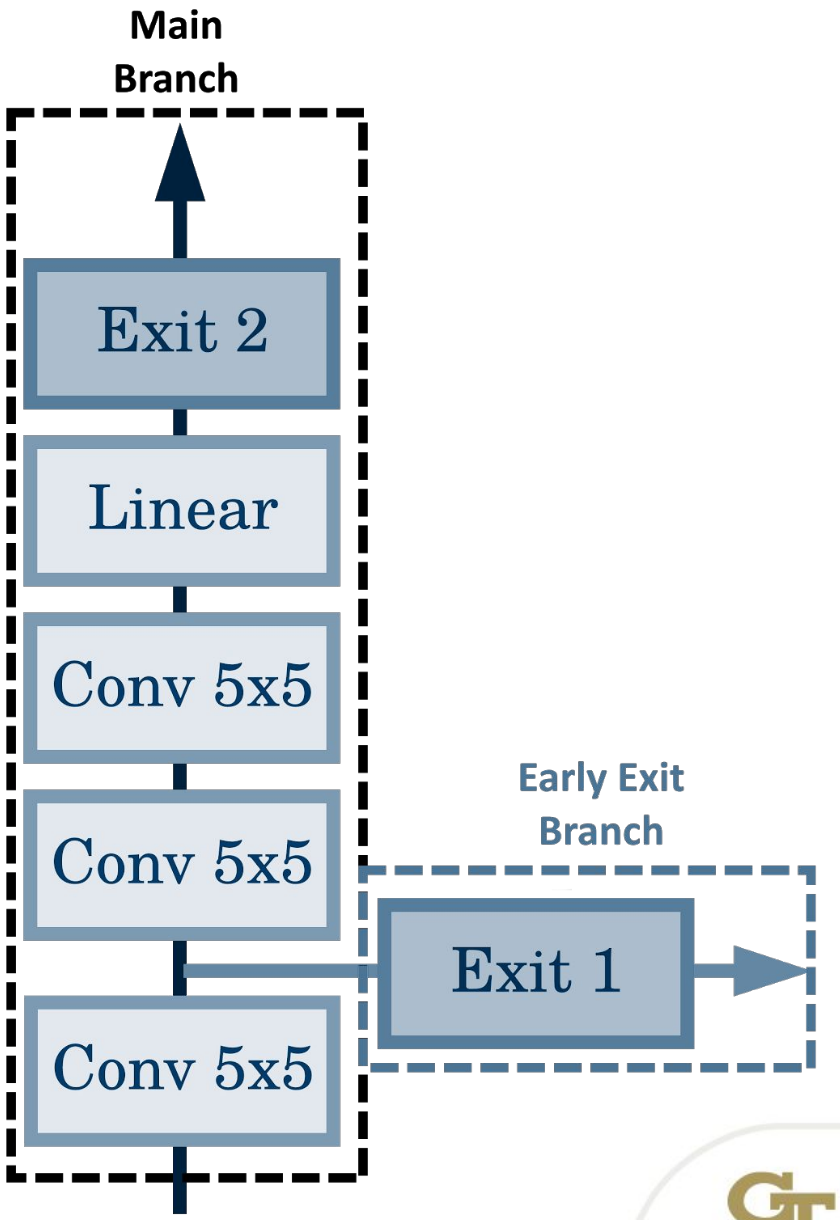
What is Early Exit?

- Place classifiers at **multiple** locations throughout the model
- At each potential exit, a **confidence value** dictates whether to use the exit
  - If threshold met, classify image



Problem Domain

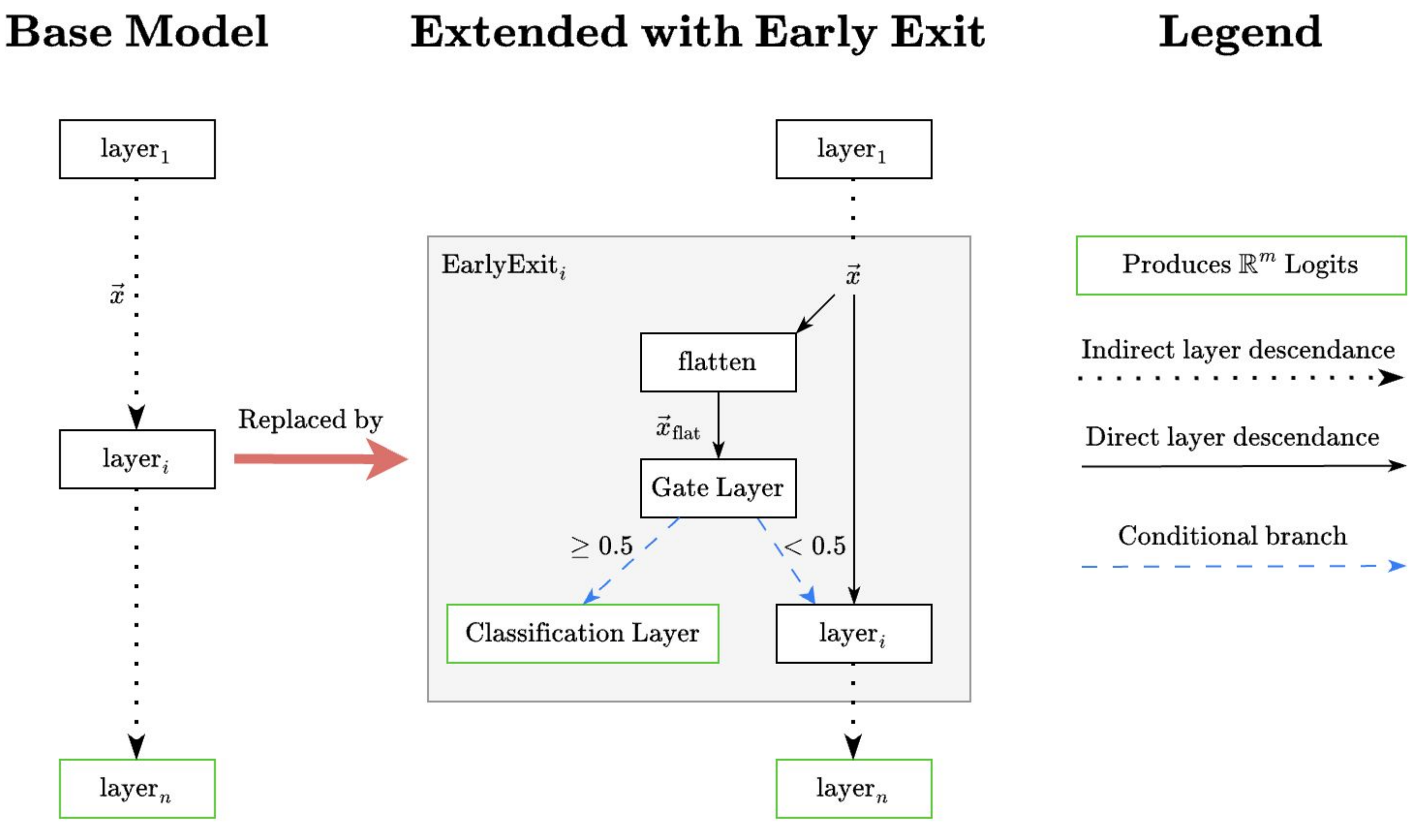
- How many exit layers to insert?
- Where to insert possible new exits?
- **How to teach a model when to use each exit?**



Project Information

Our Approach

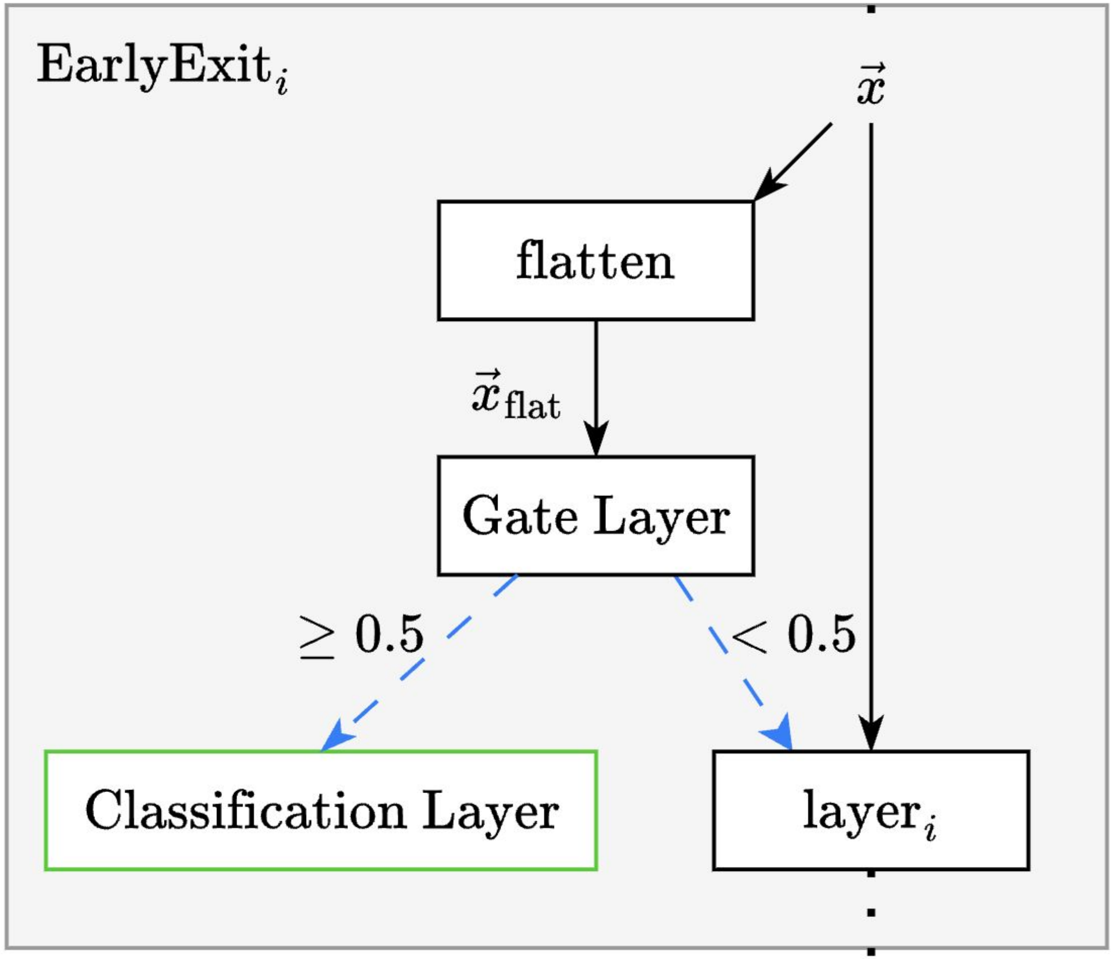
- Remove all manual exit confidences
- Insert “gate layer” to decide whether to exit
- Learn gate layer parameters to optimize accuracy/cost tradeoff
- Insert hyperparameters for tuning tradeoff



Georgia Tech

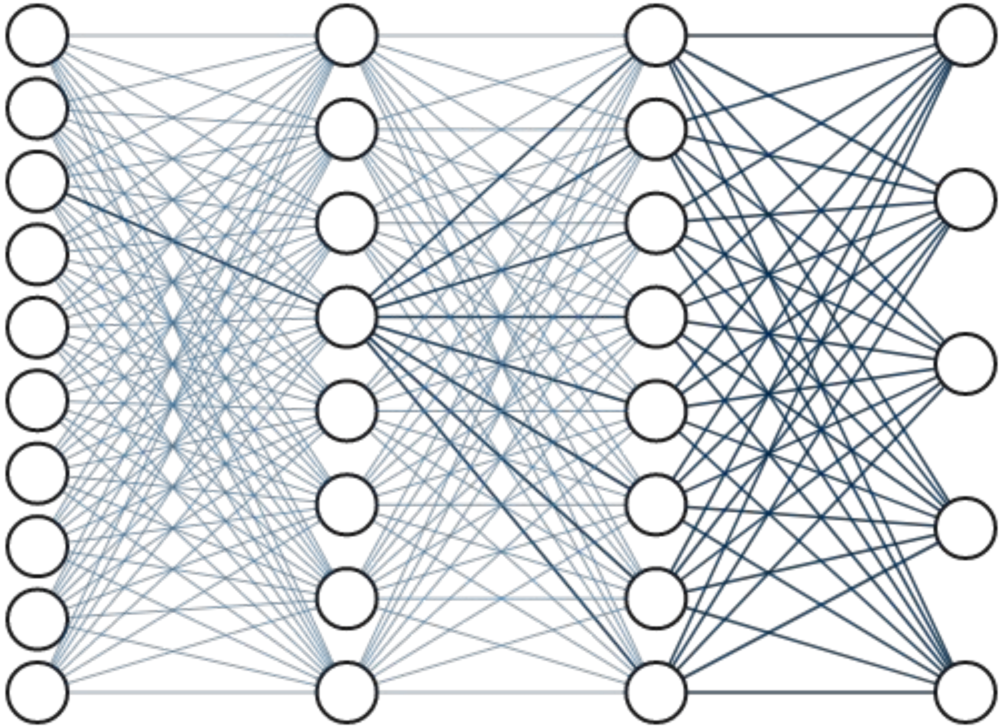
Our Approach (cont.)

- Input gets flattened
  - Output Size: (n,)
- Dot product with gate layer parameters
  - Output Size: (1,)
- Decide whether to exit
  - Gate layer output above threshold
- If exit, classify n input to logits
  - Output Size: (n_classes,)
- If not exit, feed through original network



Training Process

- Train each early exit classifier
  - These can be trained individually since they are unrelated
  - **Loss**: Categorical Cross Entropy
- Train final classification layer
  - This is just transfer learning onto your dataset
  - **Loss:** Categorical Cross Entropy
- Train gate layers
  - These have to be trained at the same time
  - **Loss**: Custom Loss Function (next slides)



Georgia Tech

Custom Gate Loss Function

$$\text{loss} = \frac{1-\alpha}{n}\left[\sum_{(X,y,\hat{y},g)}\left(\sum_{i=0}^{|g|}\left(\text{CE}(y,\hat{y}_i)\cdot g_i\prod_{j=0}^{i-1}\bar{g}_j\right)\right)\right] + \left[\sum_{(X,y,\hat{y},g)}\left(\sum_{i=0}^{|g|}\left(\text{costs}_i\cdot g_i\prod_{j=0}^{i-1}\bar{g}_j\right)+\text{costs}_{|g|}\prod_{j=0}^{|g|}\bar{g}_j\right)\right]\frac{\alpha}{n}$$

Minimize CE (Maximize Accuracy)

Minimize Computational Cost (Maximize Efficiency)

Control Weighting of Accuracy + Cost

Maximizing Accuracy

$$\left[\sum_{(X,y,\hat{y},g)} \left(\sum_{i=0}^{|g|} \left(\mathrm{CE}(y,\hat{y}_i) \cdot g_i \prod_{j=0}^{i-1} \bar{g}_j\right)\right)\right]$$

- Iterate over all images in the batch
- Calculate product summed over all gates:
  - Cross Entropy loss of:
    - **y** : true classification
    - $\hat{\mathbf{y}}_i$: predicted classification logits from gate i
  - Probability of exiting at gate i:
    - $\mathbf{g}_i$: exit confidence for gate i (larger means more confident)
    - $\bar{\mathbf{g}}_j$: forward confidence for gate j (equal to $1-g_i$)

Minimize this term for higher net model accuracy

Maximizing Efficiency

$$\left[\sum_{(X,y,\hat{y},g)}\left(\sum_{i=0}^{|g|}\left(\text{costs}_i \cdot g_i \prod_{j=0}^{i-1}\bar{g}_j\right) + \text{costs}_{|g|}\prod_{j=0}^{|g|}\bar{g}_j\right)\right]$$

- Iterate over all images in the batch
- Calculate product summed over all gates:
  - Cost of exiting at gate i:
    - Derived as **% parameters utilized**
    - Value ranges from **(0, 1]**
  - Probability of exiting at gate i:
    - $\mathbf{g_i}$: exit confidence for gate i (larger means more confident)
    - $\mathbf{\bar{g}_j}$: forward confidence for gate j (equal to $1\text{-}g_i$)

Minimize this term for lower net inference cost

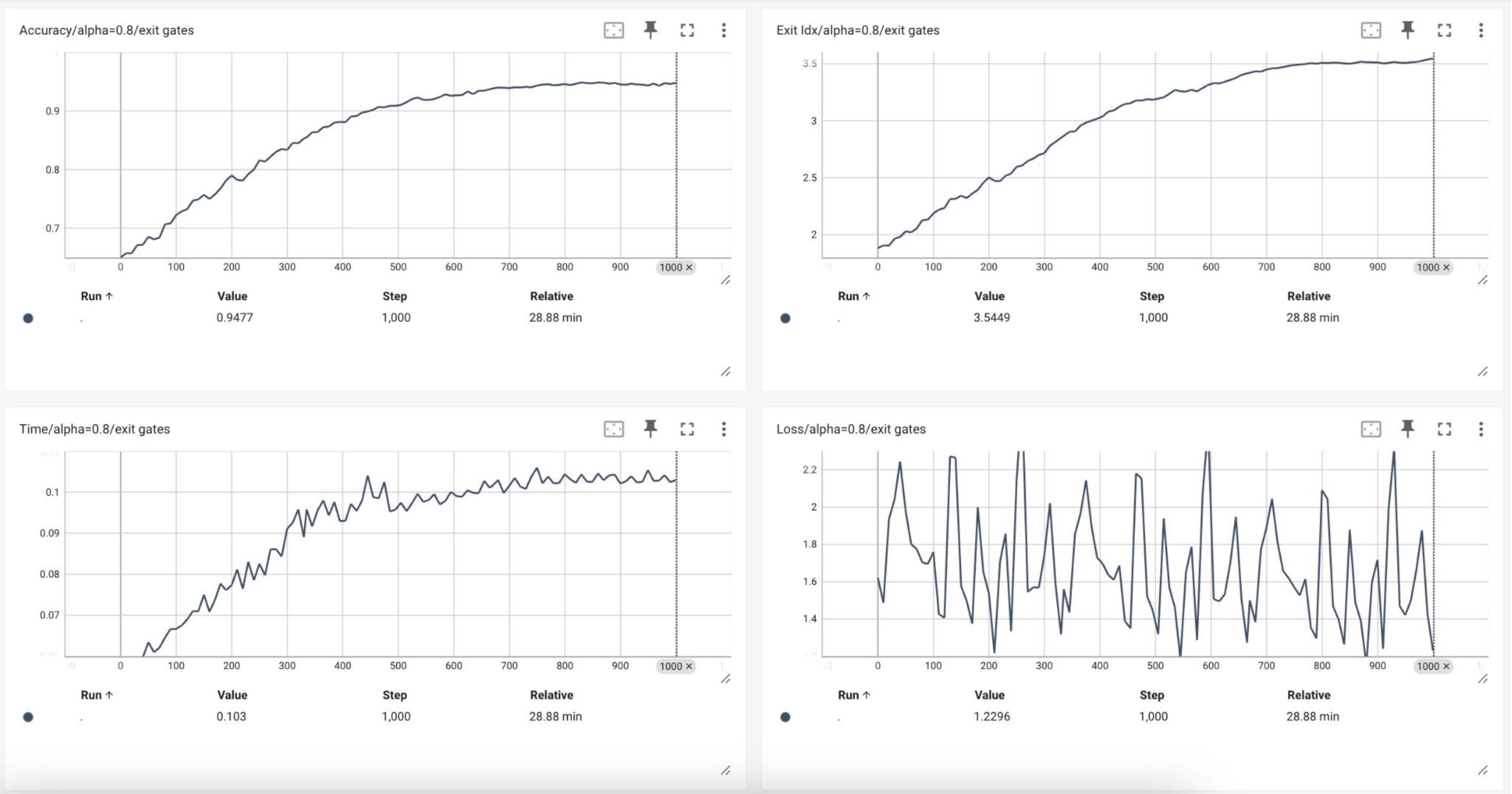
Initial Results
+
Remaining Work

Initial Results (ResNet50/ImageNetTE, α=0.8)

25% Time Savings

50% Cost Reduction

5% Accuracy Drop



Georgia Tech

Correct Examples: Wolf (VGG11/CIFAR100, α=0.6)



Exit 2 of 5

Consistent Face Scale and Position



Exit 3 of 5

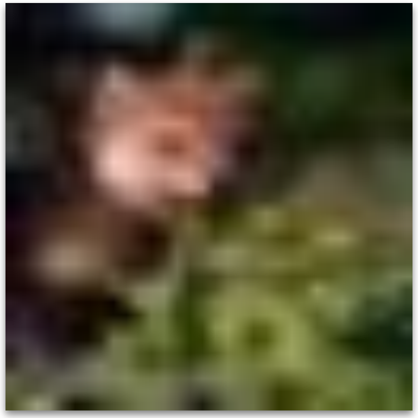
Discoloration, scale and position variation

Georgia Tech

Incorrect Examples: Wolf (VGG11/CIFAR100, α=0.6)

Fox

Seal



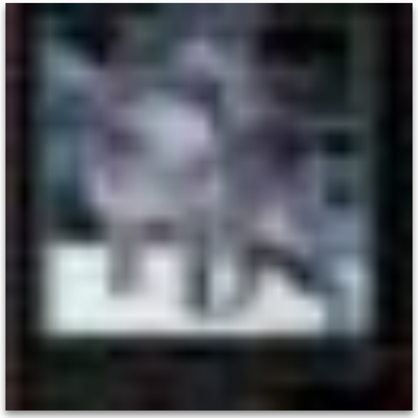
Fox

Exit 2 of 5

Subject/background interference

Rabbit

Flatfish



Turtle

Exit 3 of 5

Misleading subject representation

Next Steps

- Code cleaning + implementation improvements
- Pushing Time Savings
  - Multithreading gate and classification layers
  - Changing gate and classifier architectures
- Image Augmentations
- Configure training on more powerful computers (GPUs, Pace, etc.)
- Calculate trendlines when varying α
  - Average model accuracy
  - Average exit location
  - Average computation time
  - Average computation cost
- Train on a large set of different datasets/ML models
  - AlexNet, MSDNet, ...
- Calculate scaling scores based upon dataset size
- Continue to do comparison studies with previous work