Efficient Neural Network Inference and Training Using Early Exit Strategies

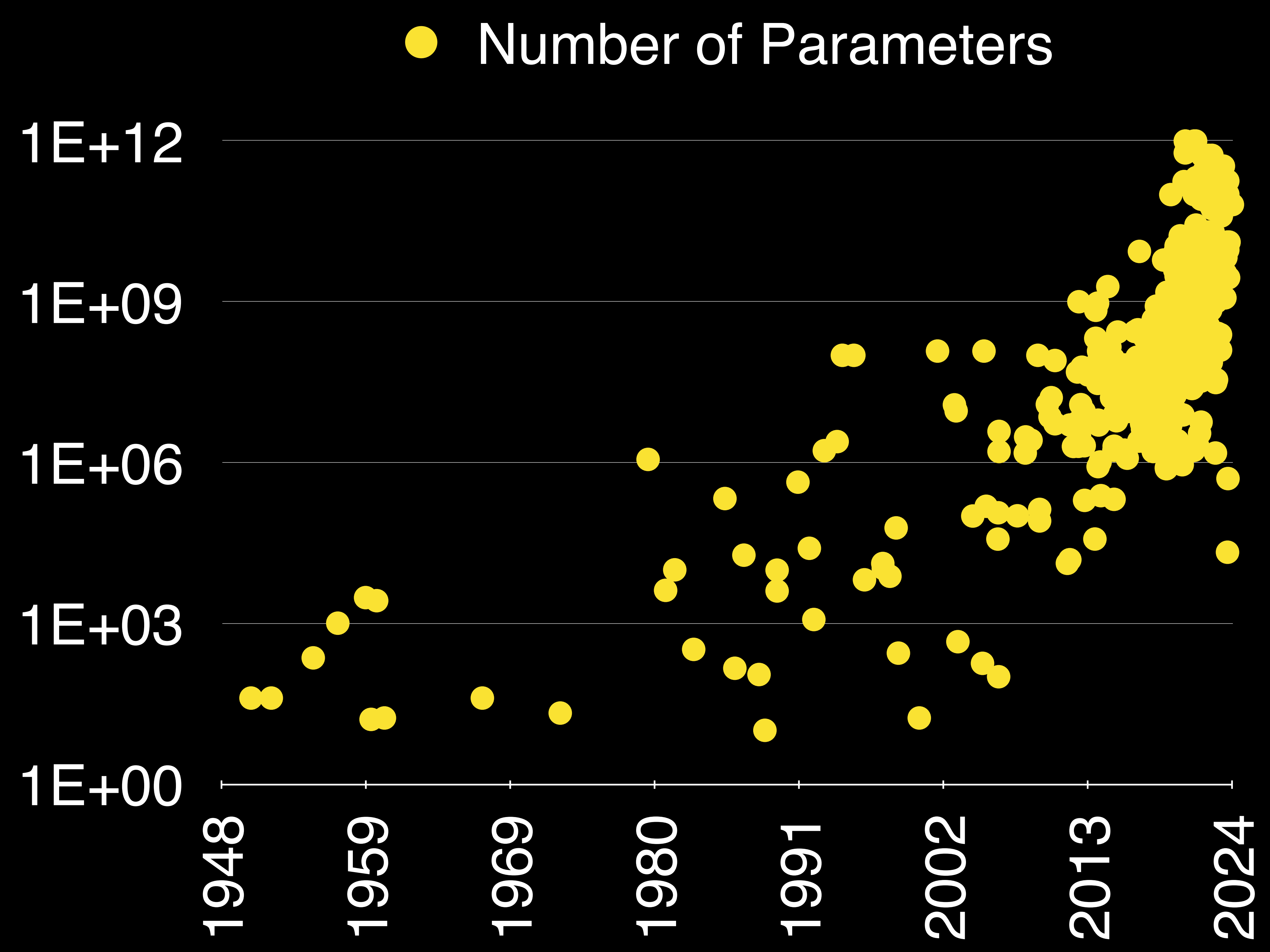
Alperen Görmez
Department of Electrical and Computer Engineering
University of Illinois Chicago
Dissertation Defense
October 23, 2024

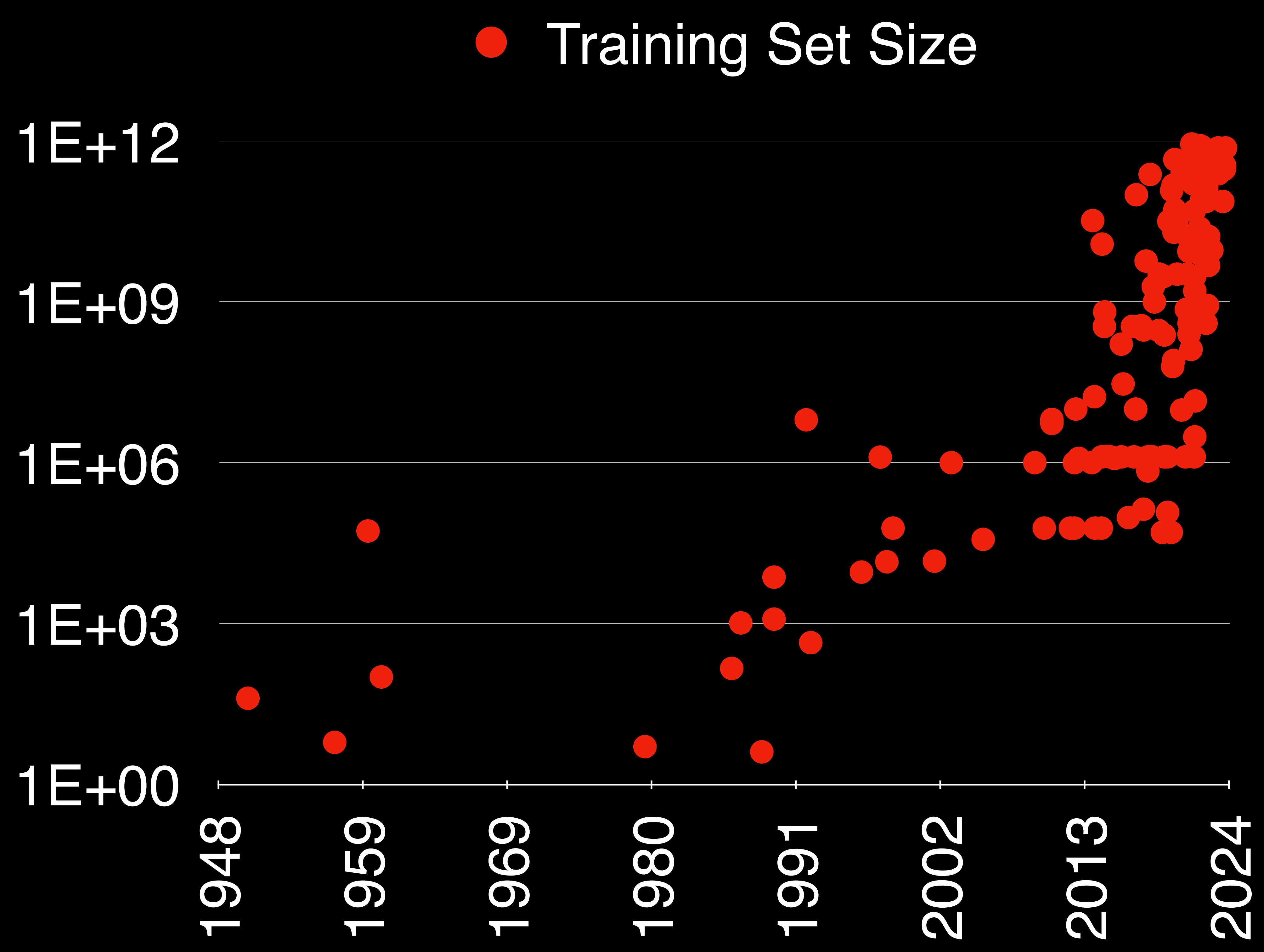
Defense Committee:
Erdem Koyuncu, Chair and Advisor
Abolfazl Asudeh
Natasha Devroye
Mesrob Ohannessian
Besma Smida

Agenda

1. Problem
2. Background
3. E^2CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning
4. Pruning Early Exit Networks
5. Class Based Thresholding in Early Exit Semantic Segmentation Networks
6. Dataset Pruning Using Early Exit Networks
7. Class-aware Initialization of Early Exits for Pre-training Large Language Models
8. Future Work
9. Conclusion

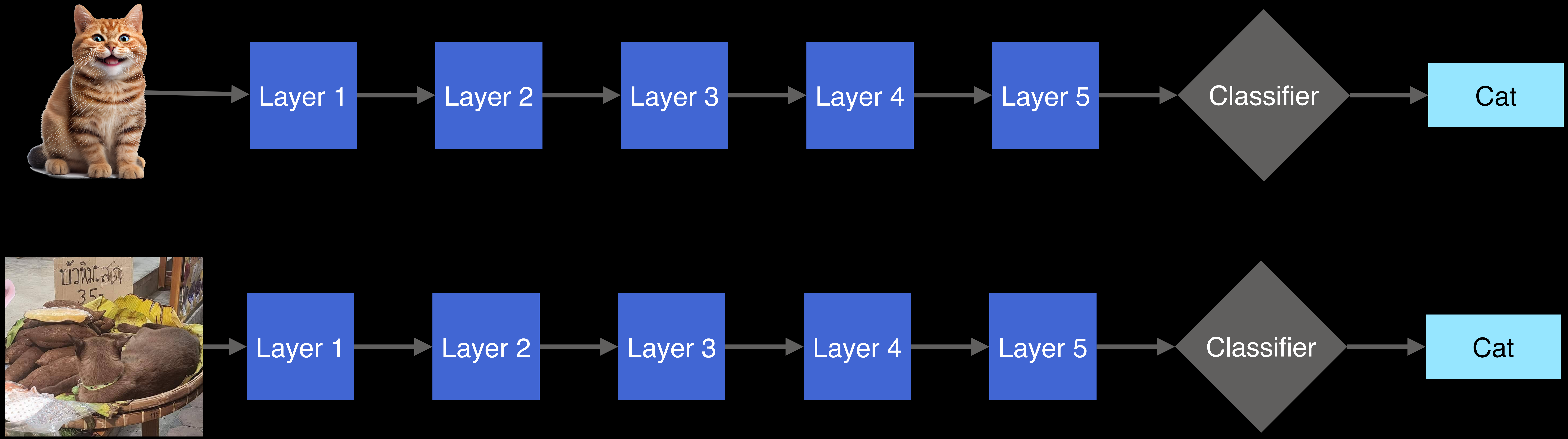
Problem





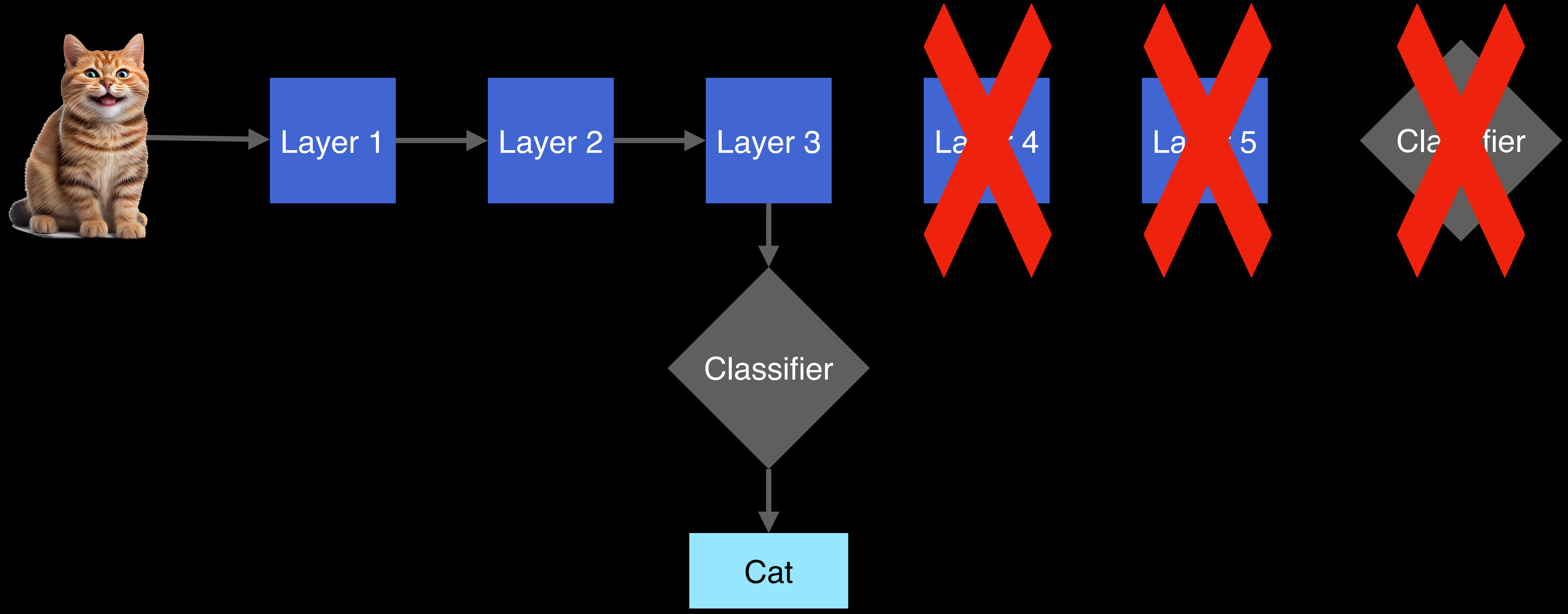
Inference and training costs rise.
How can we reduce the costs?

Background



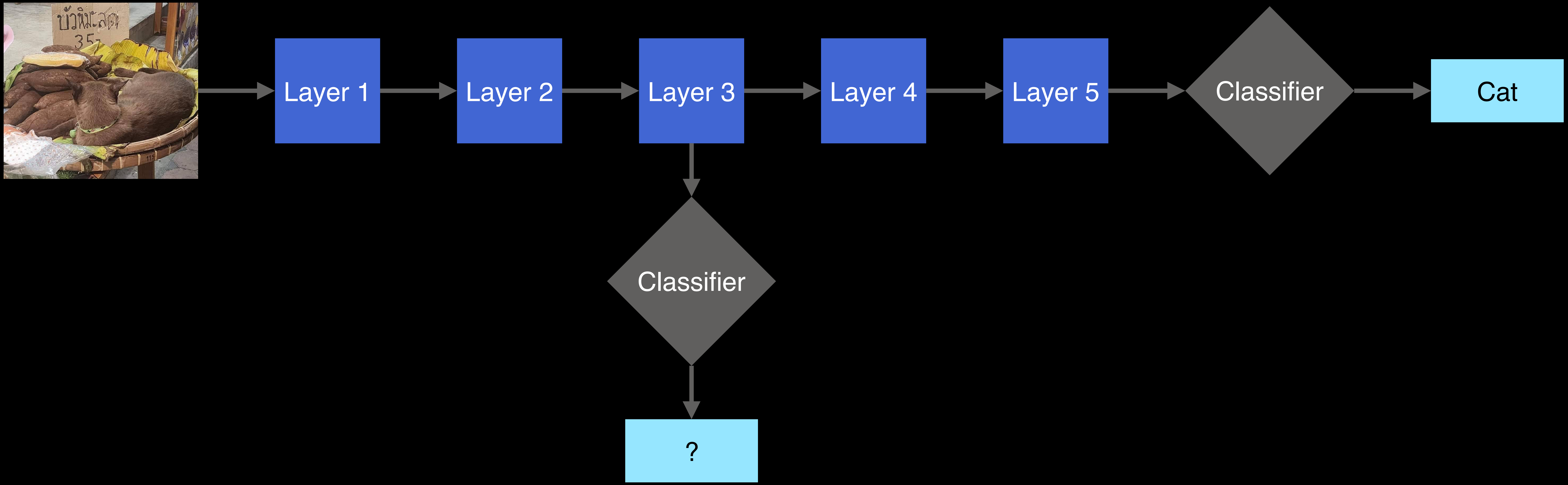
Real world data is heterogeneous.

Background



Easy data should exit early.

Background

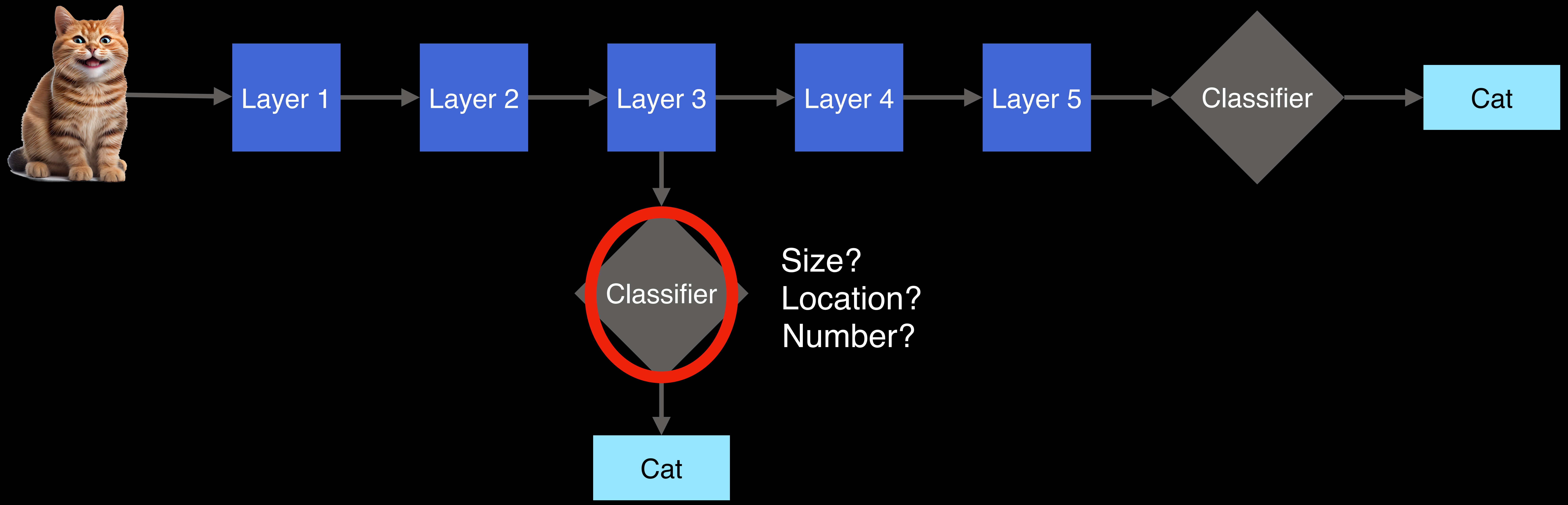


Difficult data should utilize full computation.

E^2CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning

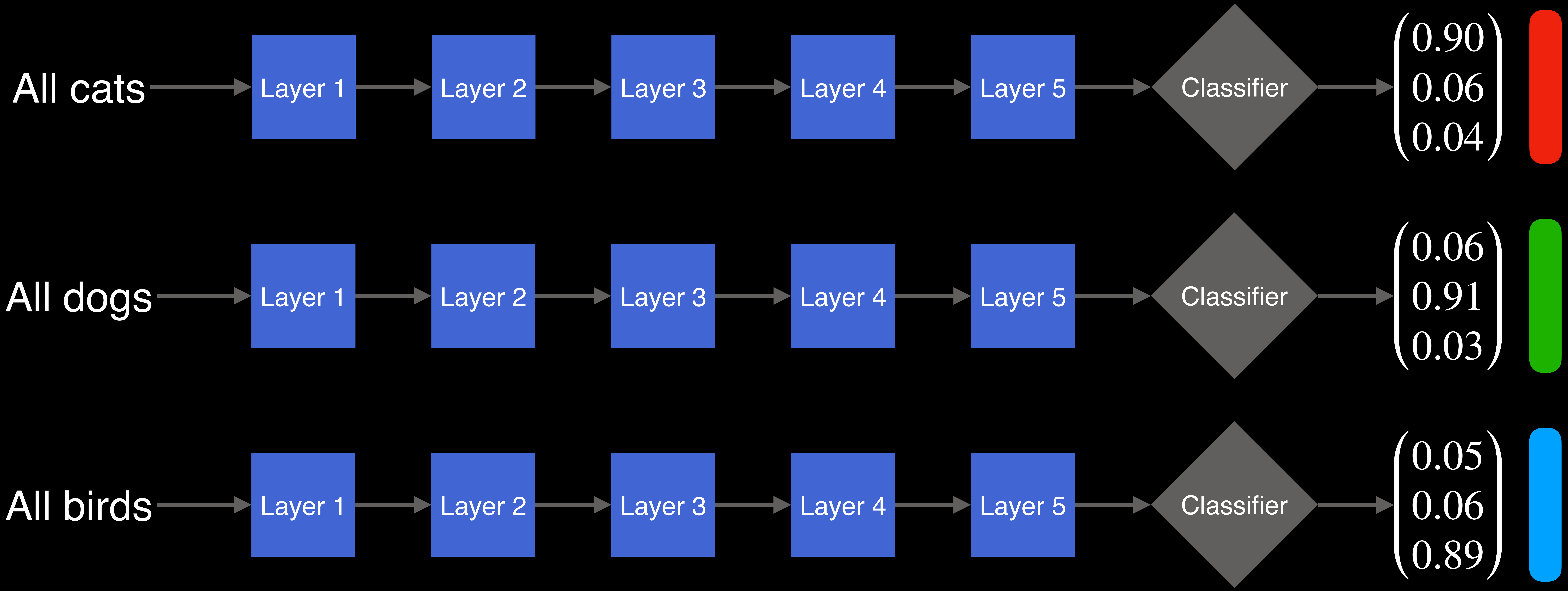
A. Görmez, V. R. Dasari and E. Koyuncu, "E2CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning," *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022, pp. 1-8, doi: 10.1109/IJCNN55064.2022.9891952. Also presented at Eastern European Machine Learning Summer School, July 2022 (Top-voted poster of the summer school).

E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning

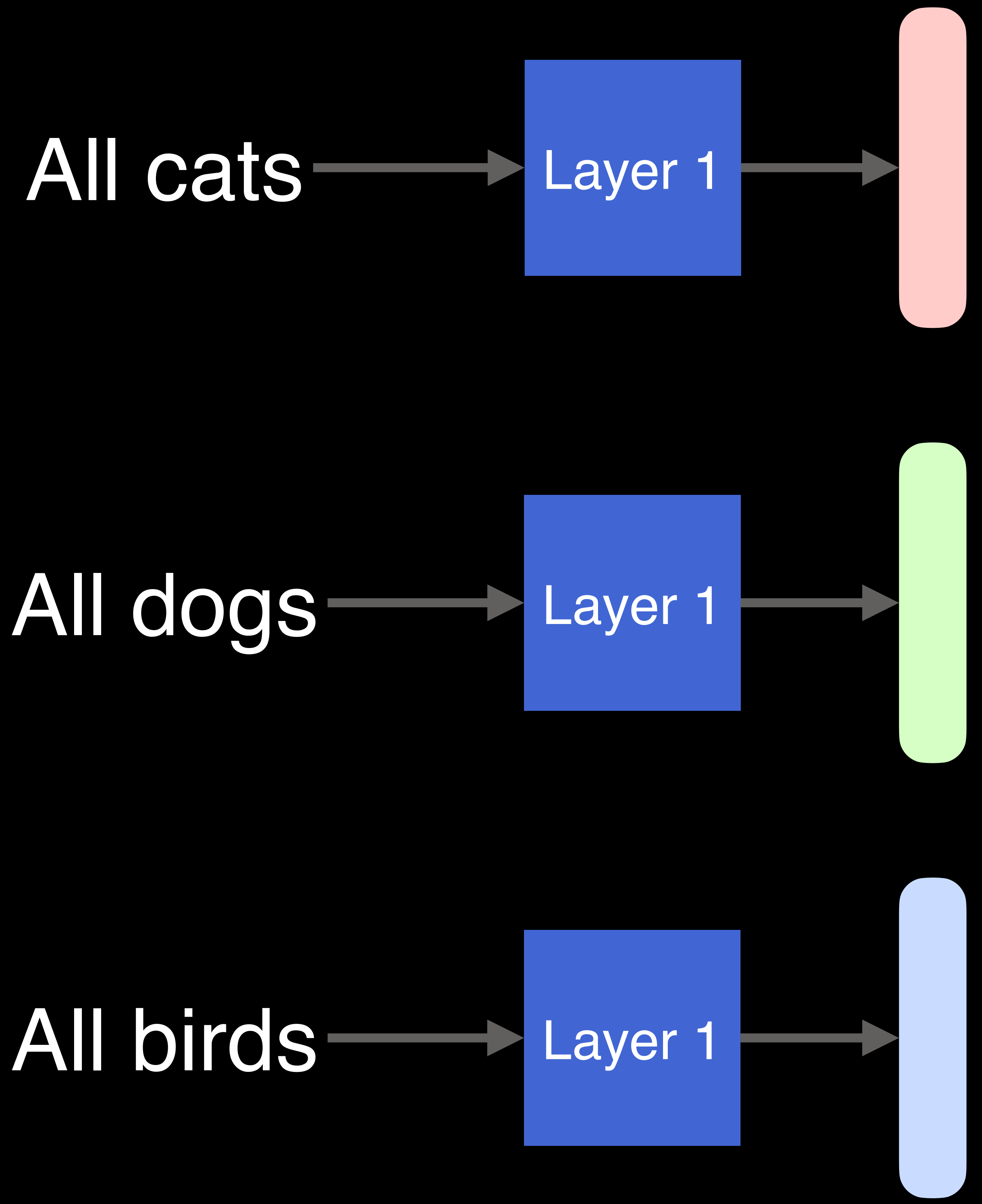


Modification, further training, and hyper parameter tuning required.

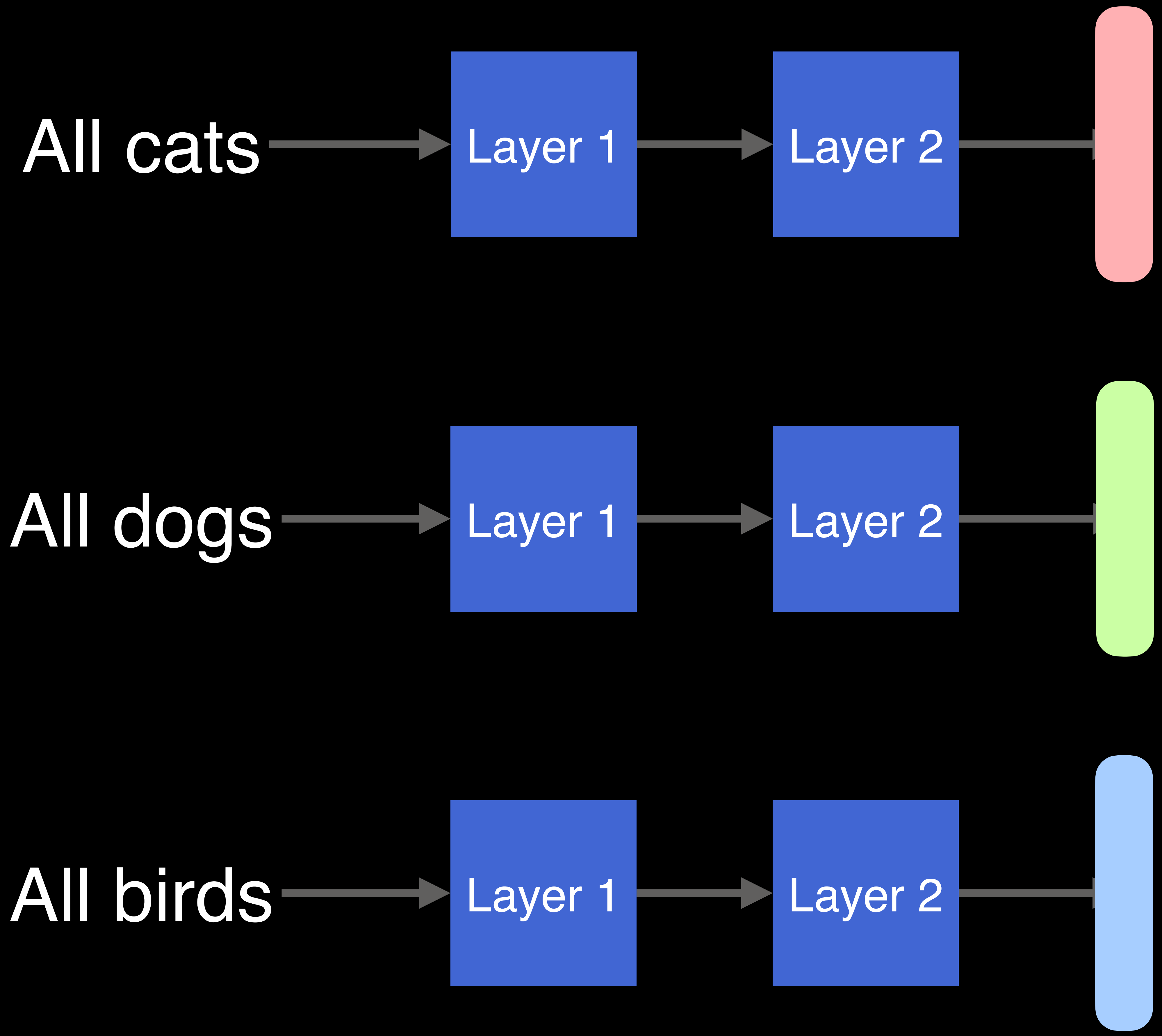
E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



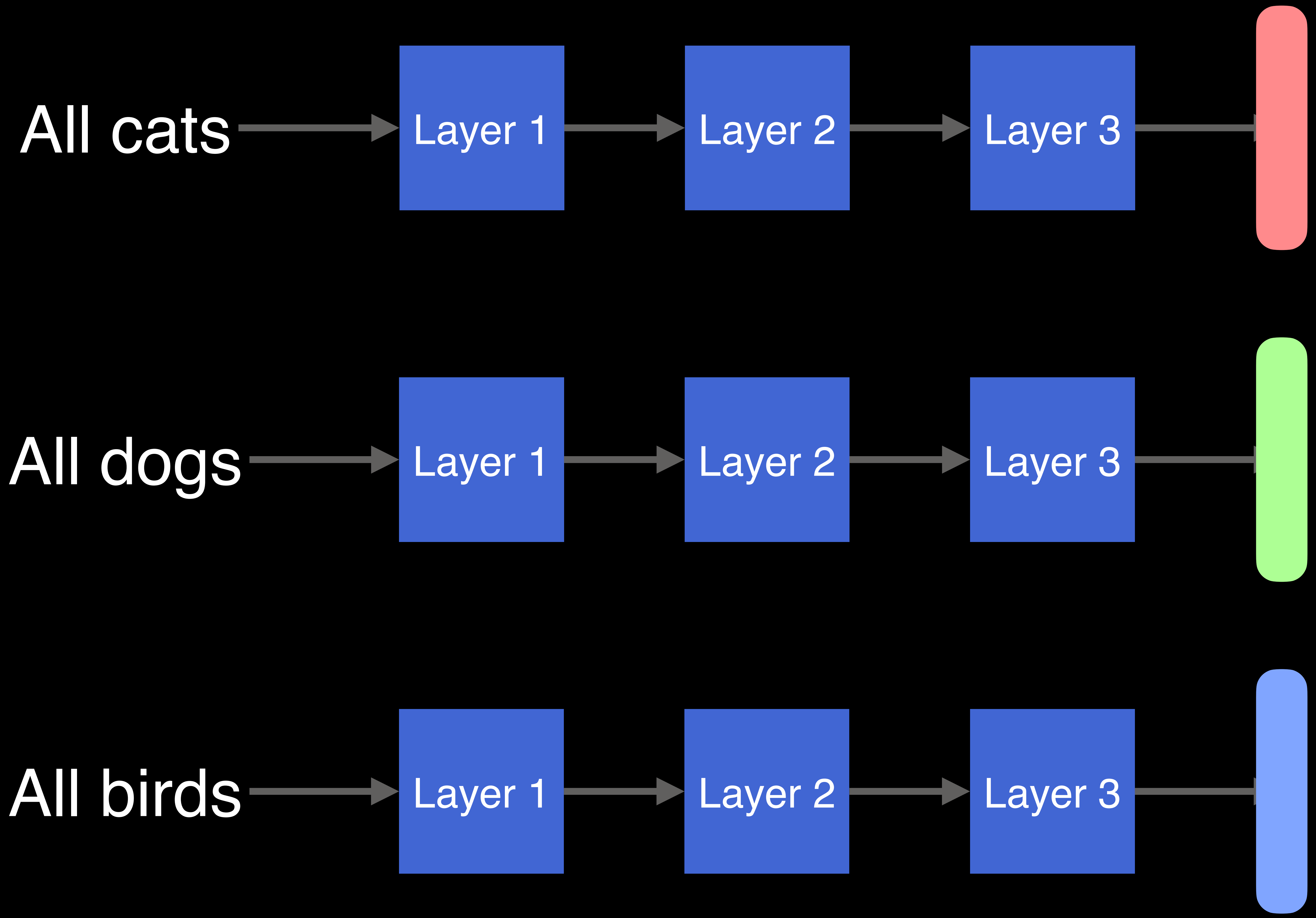
E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



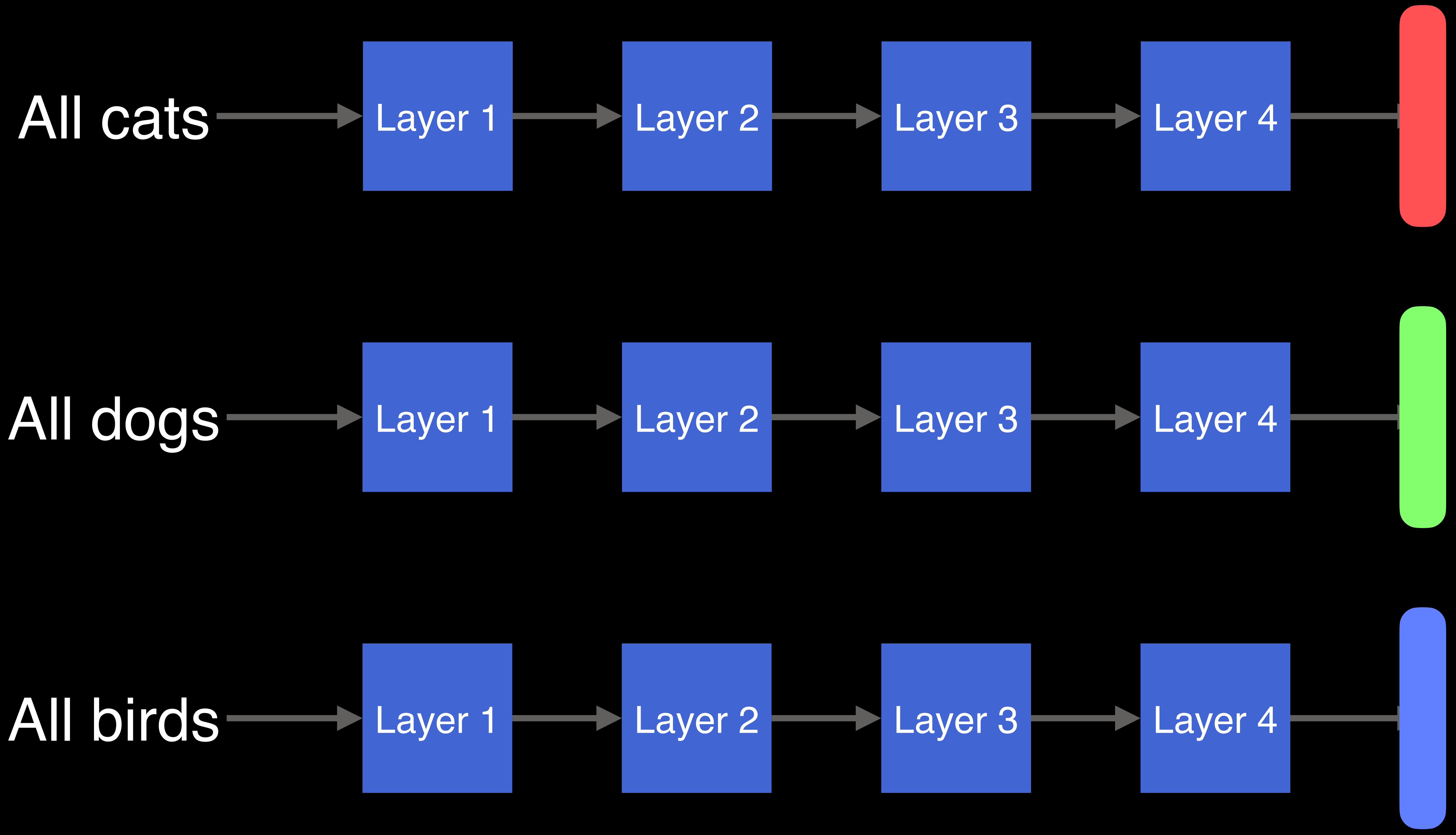
E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



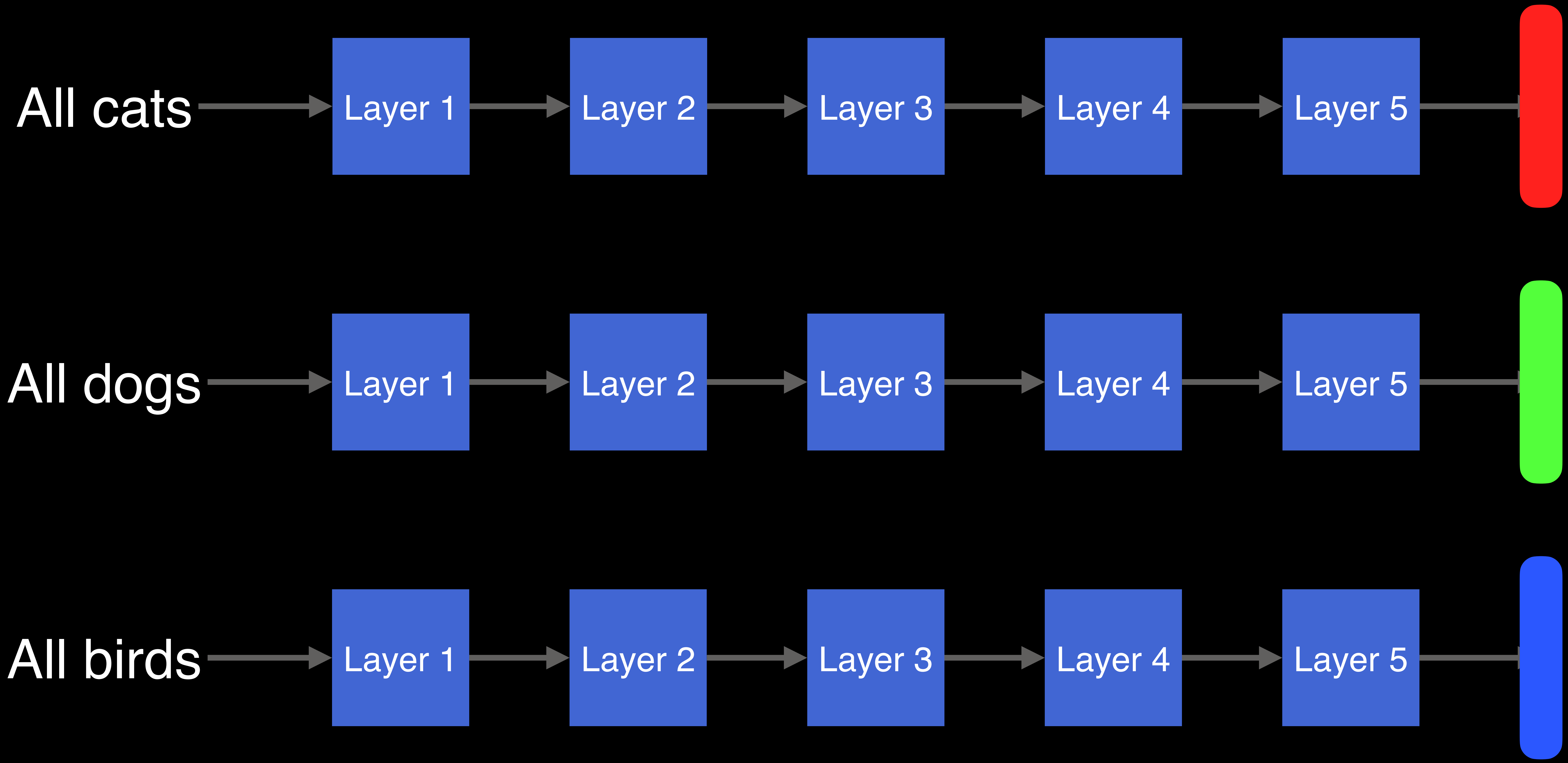
E^2CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



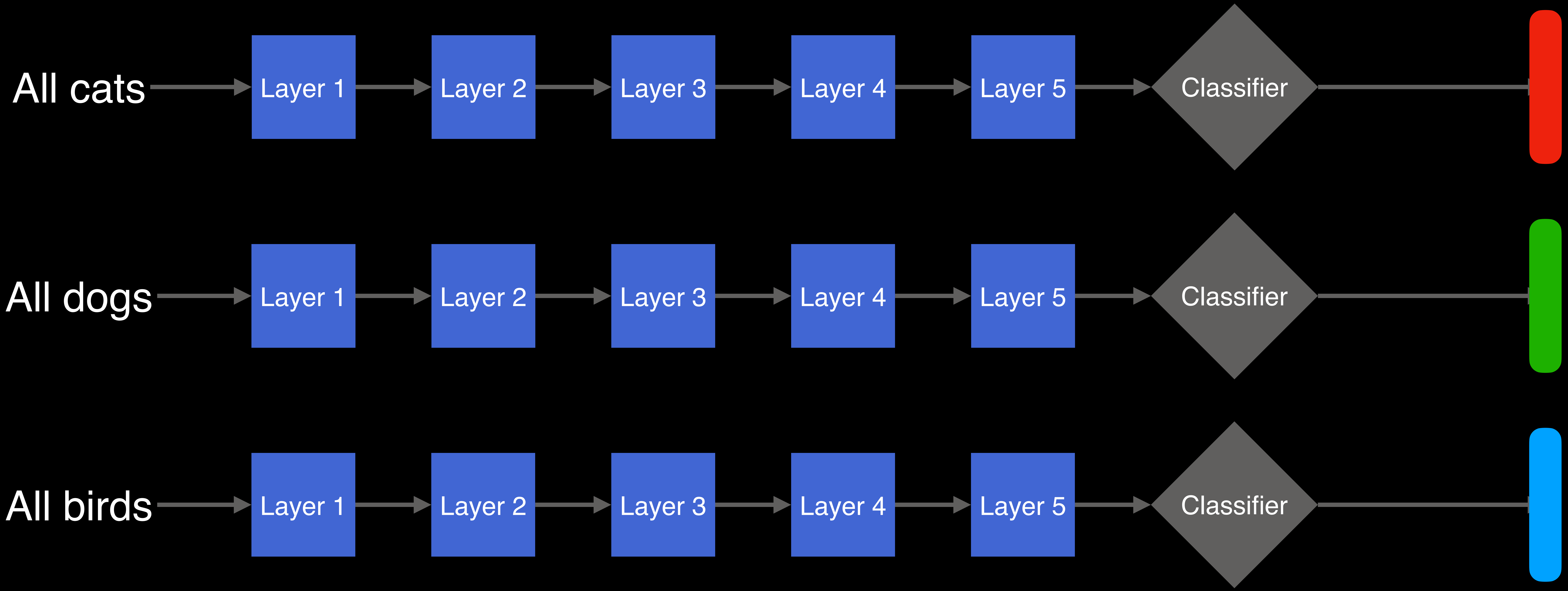
E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



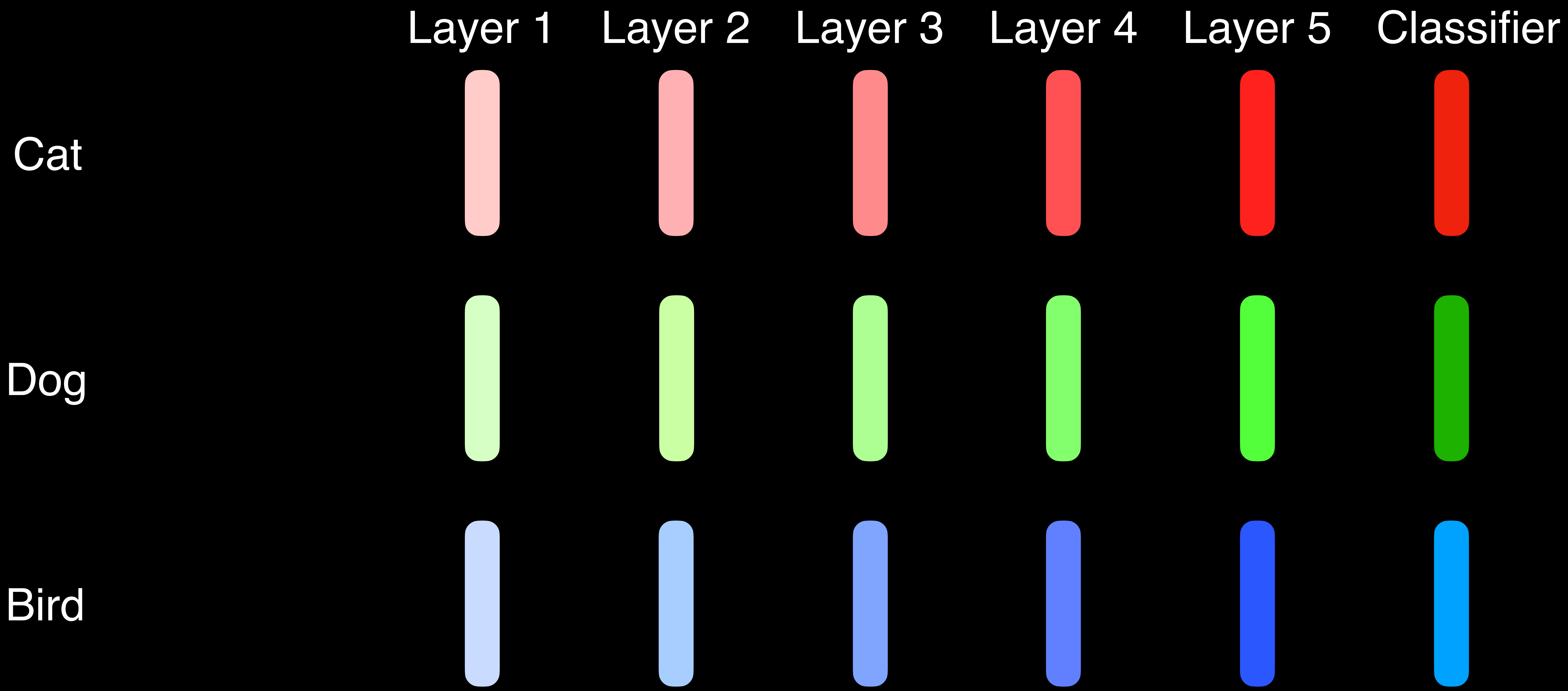
E^2CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



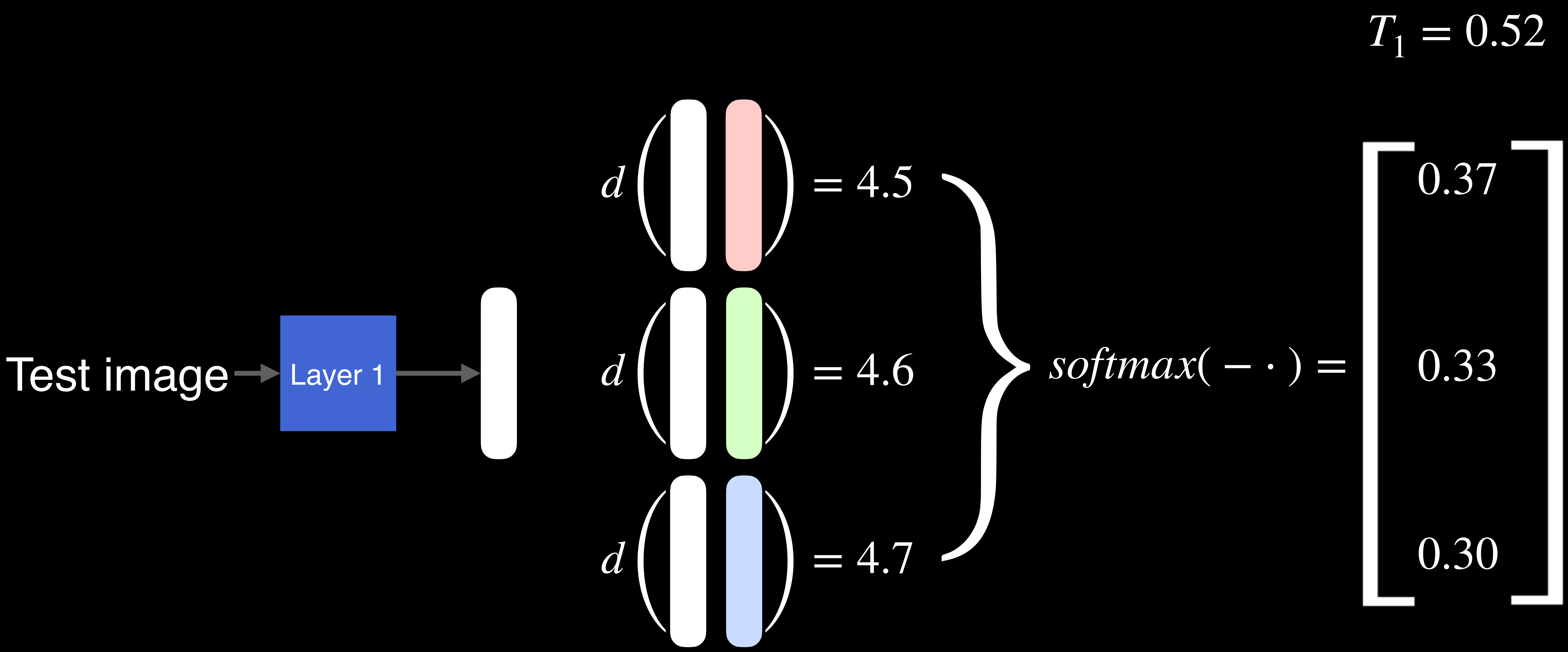
E^{2}CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



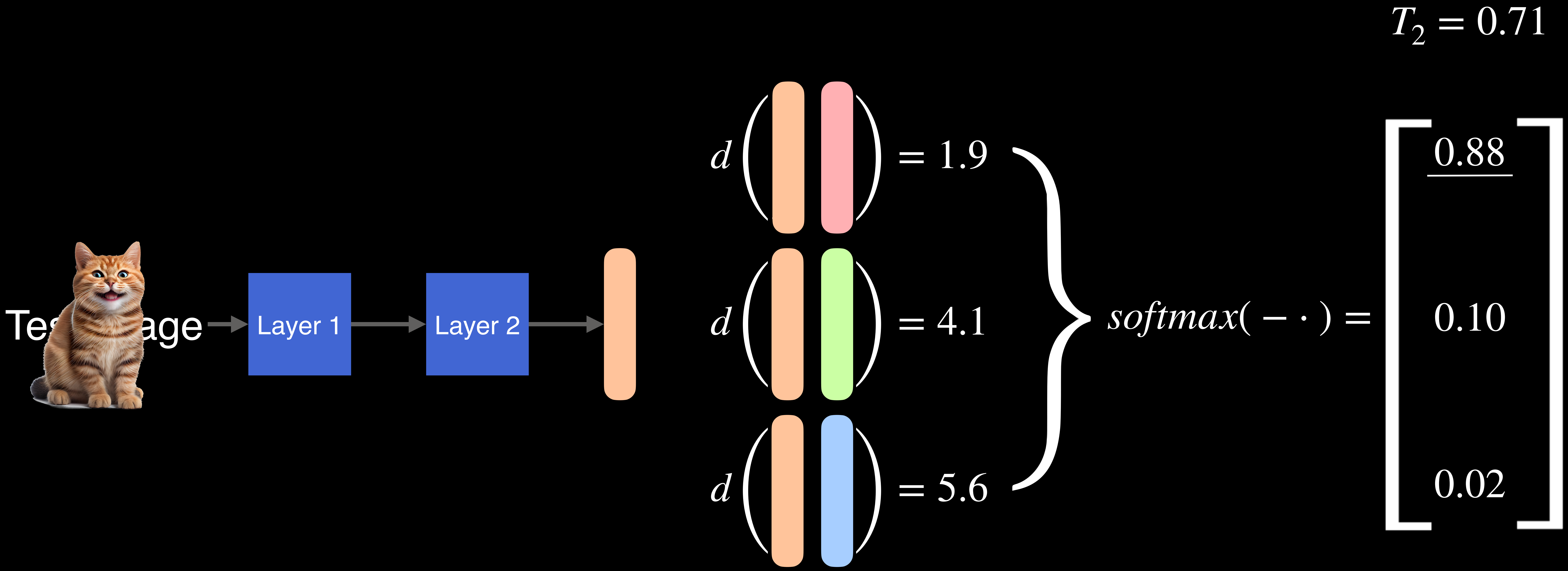
E^{2}CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning

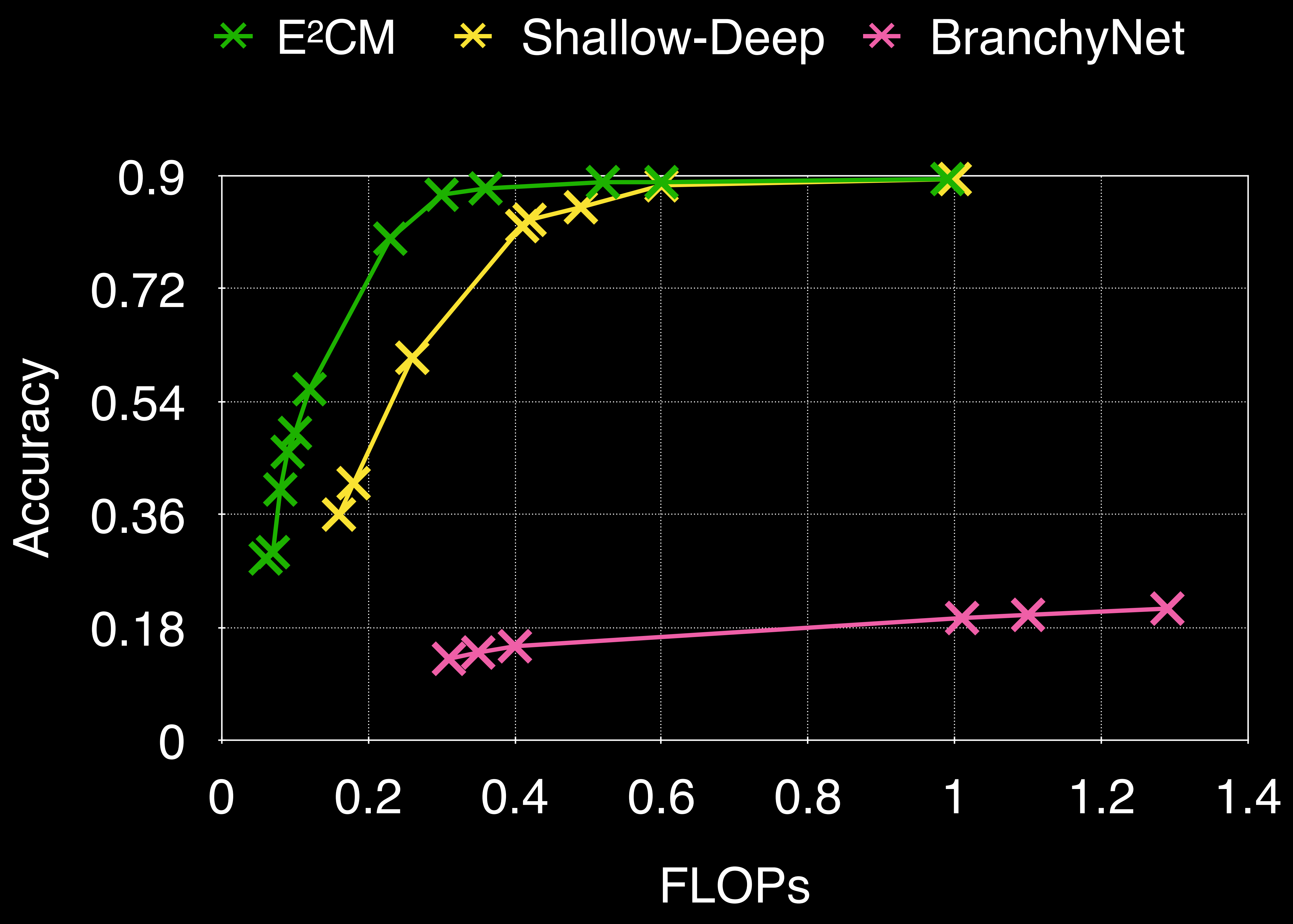


E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning

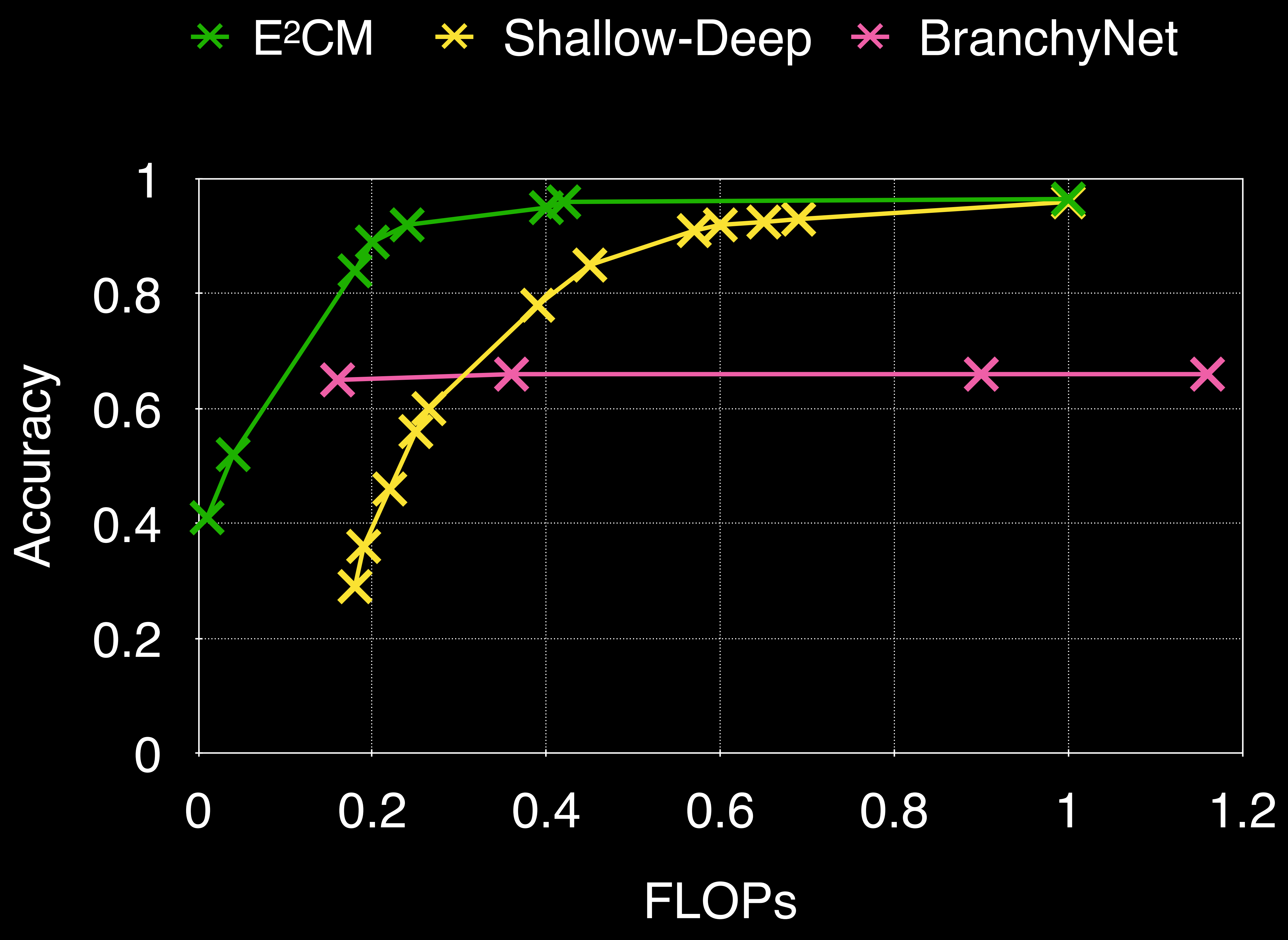


No modification, no further training, and no hyper parameter tuning required.

E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



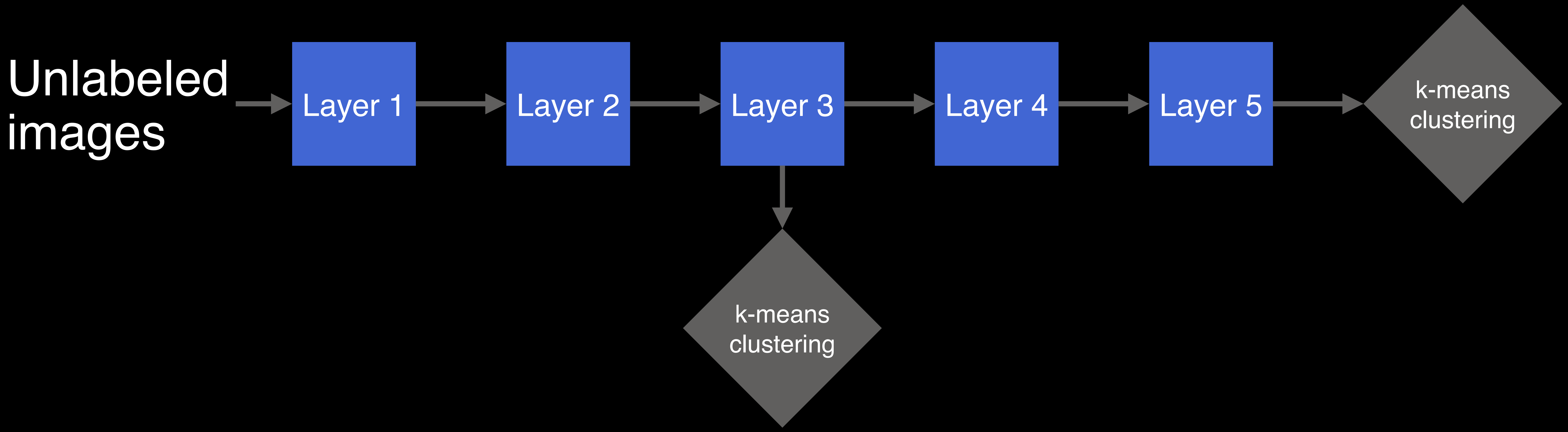
ResNet-152 & CIFAR-10



WideResNet-101 & KMNIST

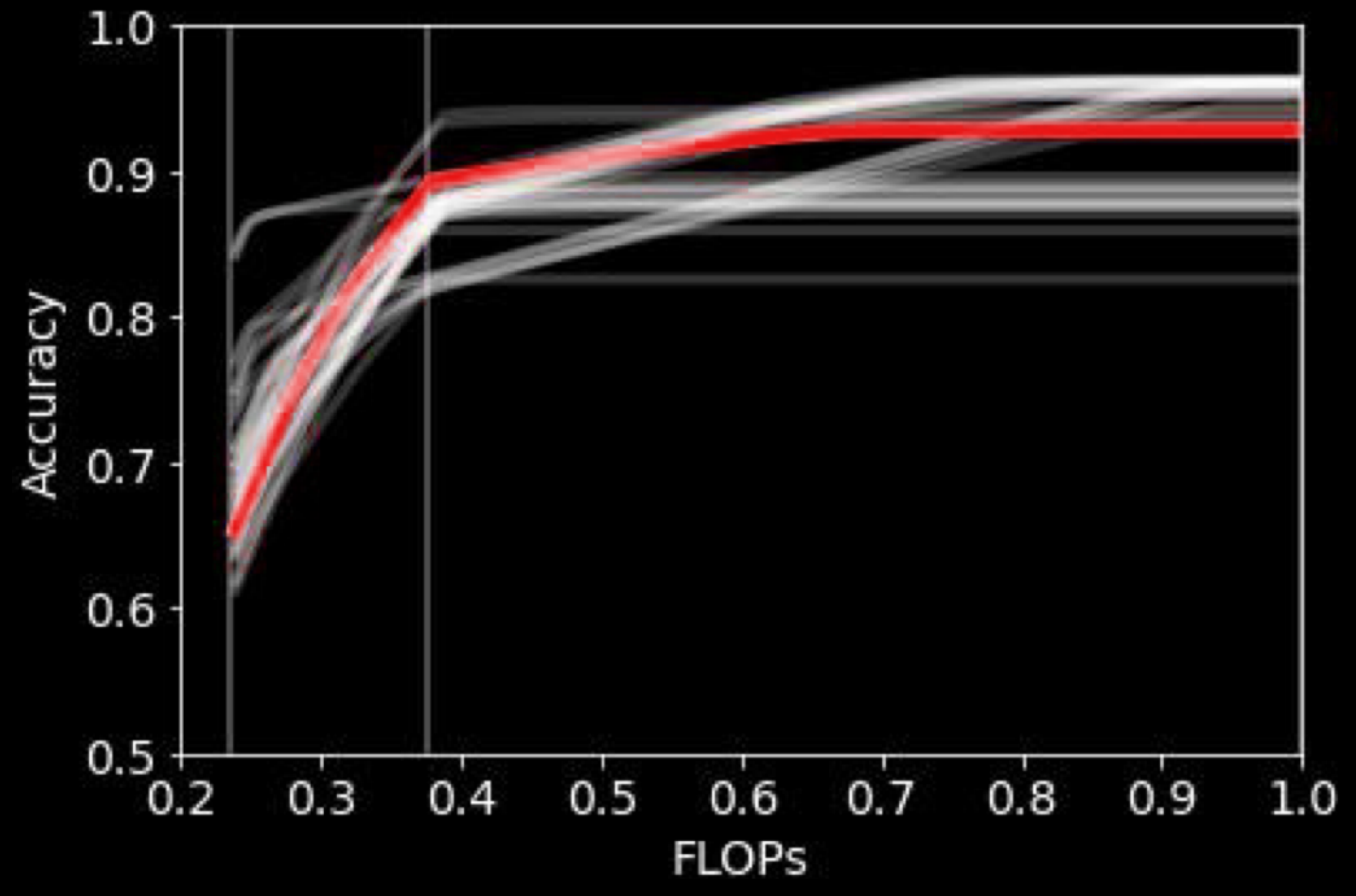
E²CM performs better under a fixed training time budget of one epoch.

E^{2}CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning

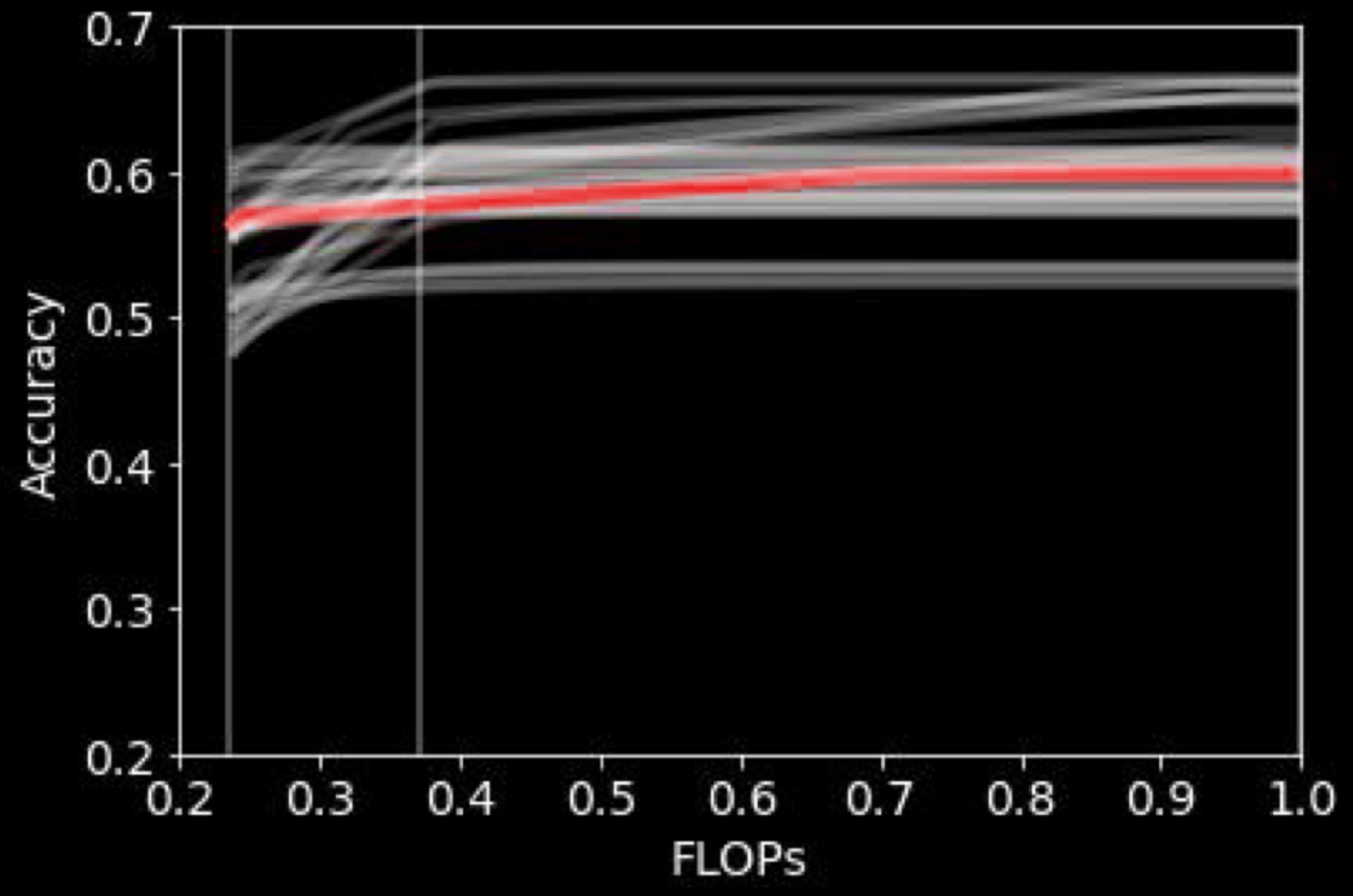


E^{2}CM can be applied to unsupervised learning too.

E²CM: Early Exit via Class Means for Efficient Supervised and Unsupervised Learning



MNIST



Fashion-MNIST

Thesis Contributions

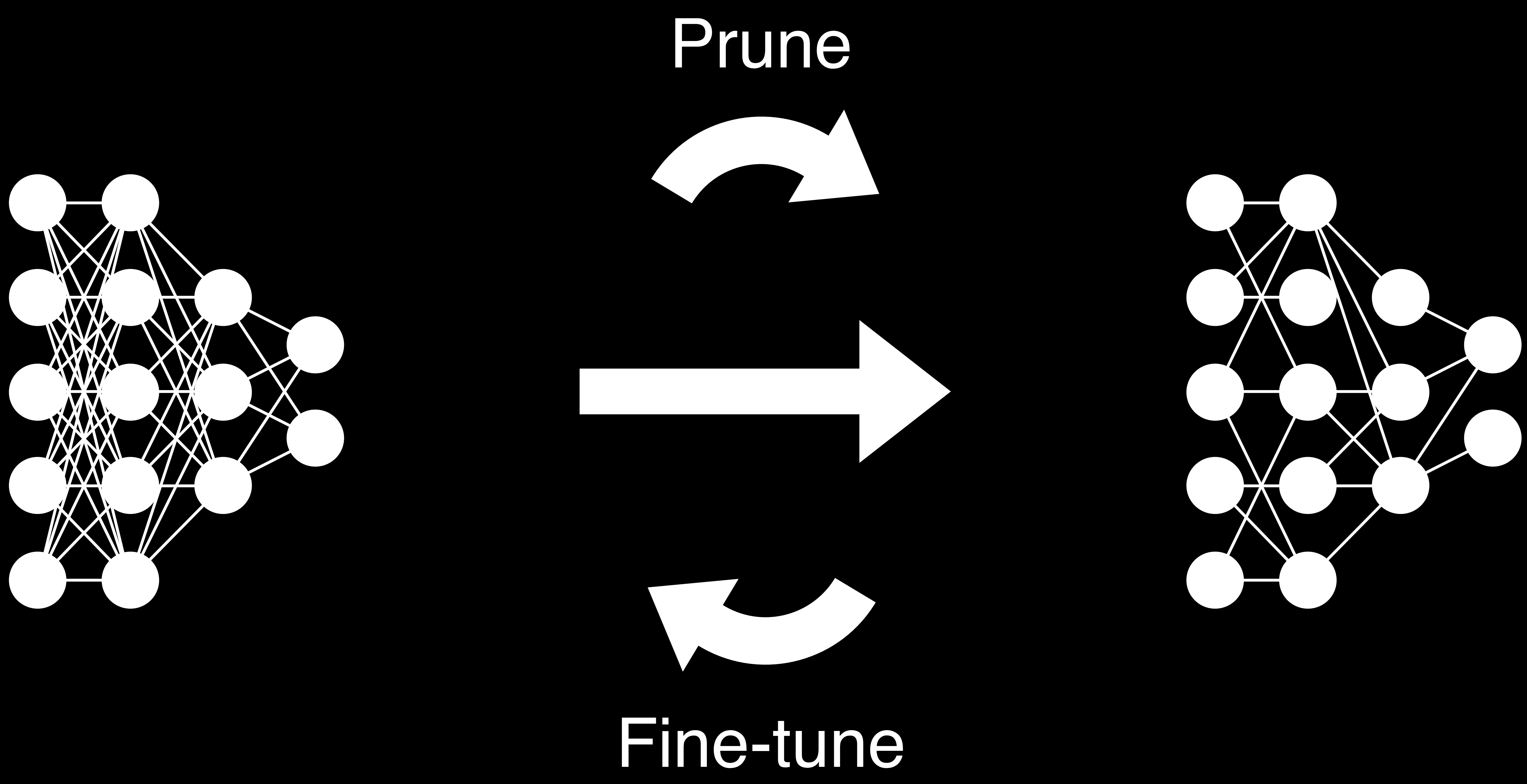
1. Designed E^2CM, a simple and lightweight early exit algorithm to reduce inference cost.

2. Demonstrated how early exit networks can be combined with model pruning.

Pruning Early Exit Networks

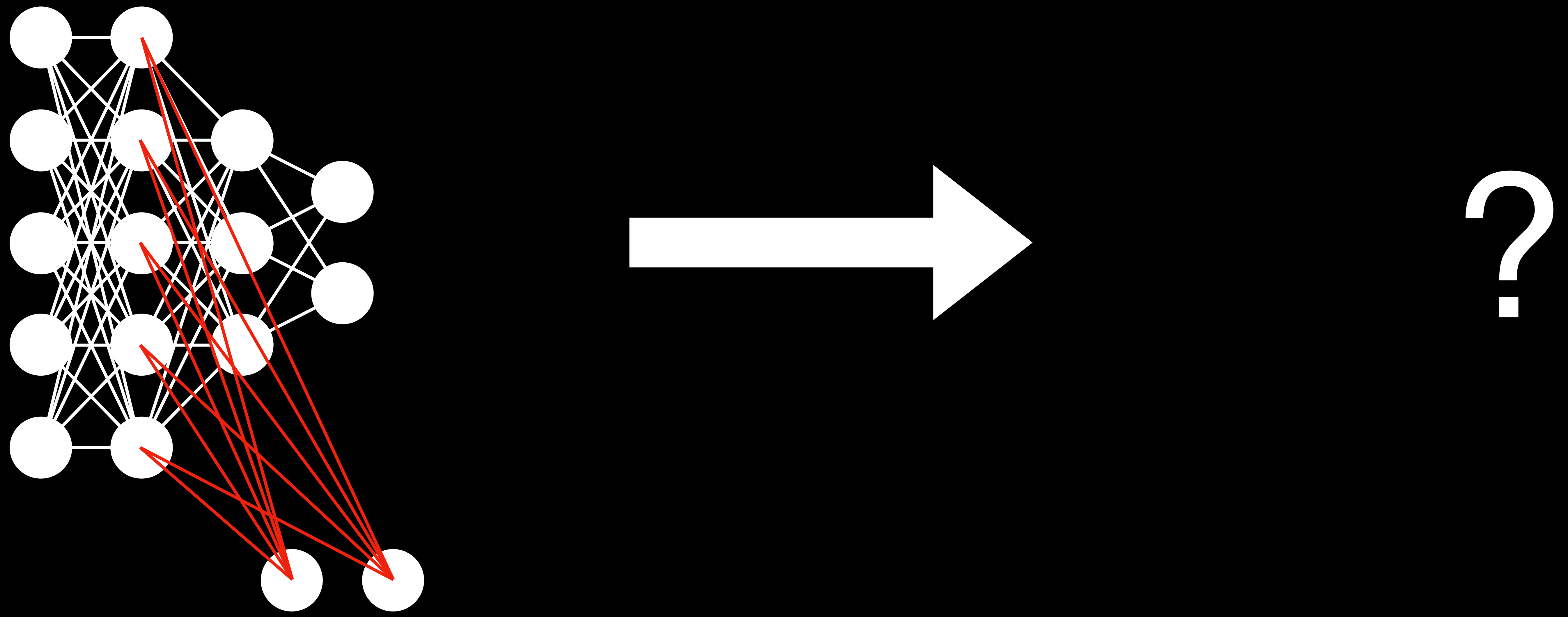
A. Görmez and E. Koyuncu, "Pruning Early Exit Networks," *2022 Sparsity in Neural Networks*, 2022, doi: 10.48559/arXiv.2207.03644.

Pruning Early Exit Networks



Pruning reduces the model size by setting weights to zero.

Pruning Early Exit Networks



How to prune the early exit weights?

Pruning Early Exit Networks

Approach 1

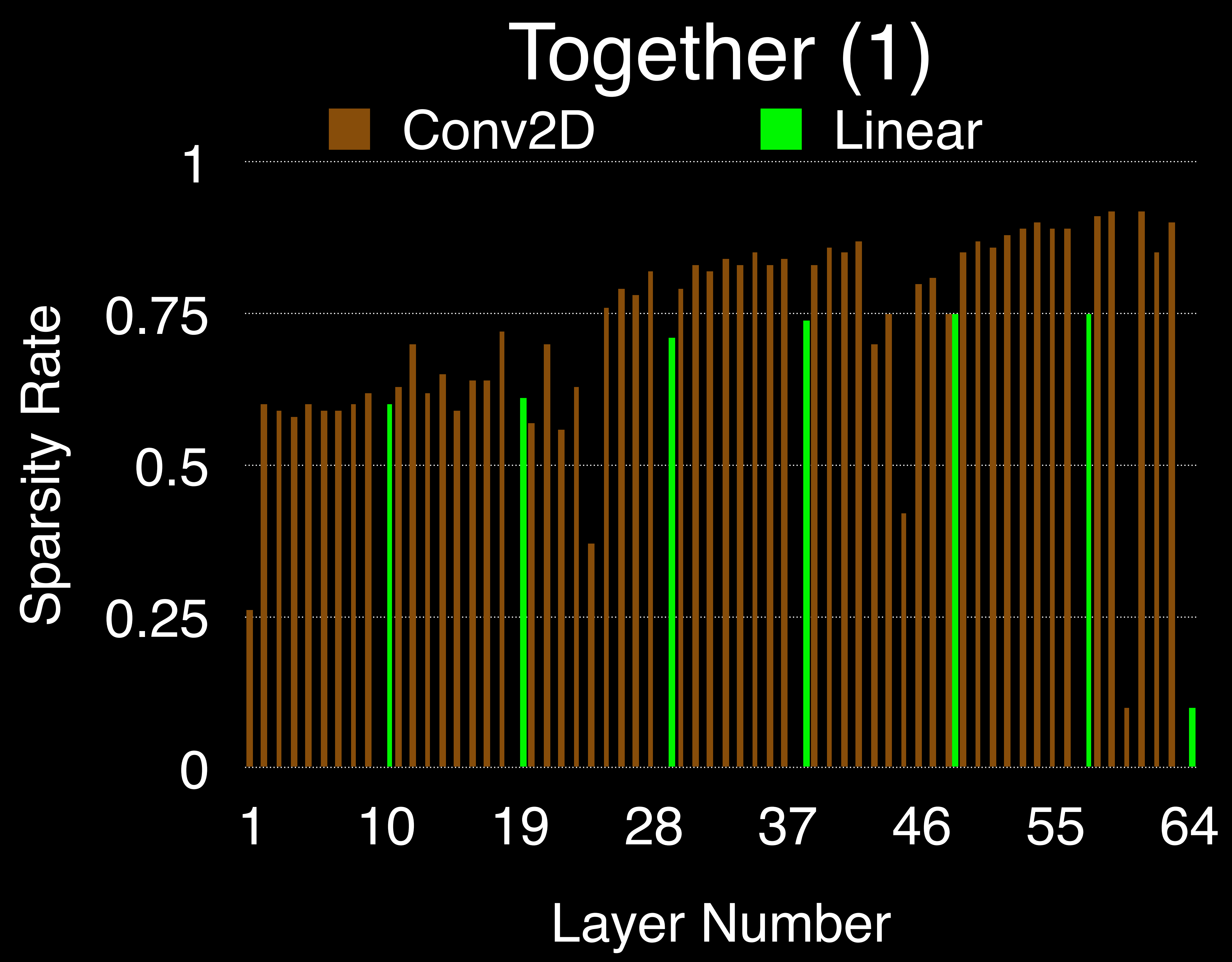
- Prune backbone & exit layers together
- Finetune everything together
- Repeat

Approach 2

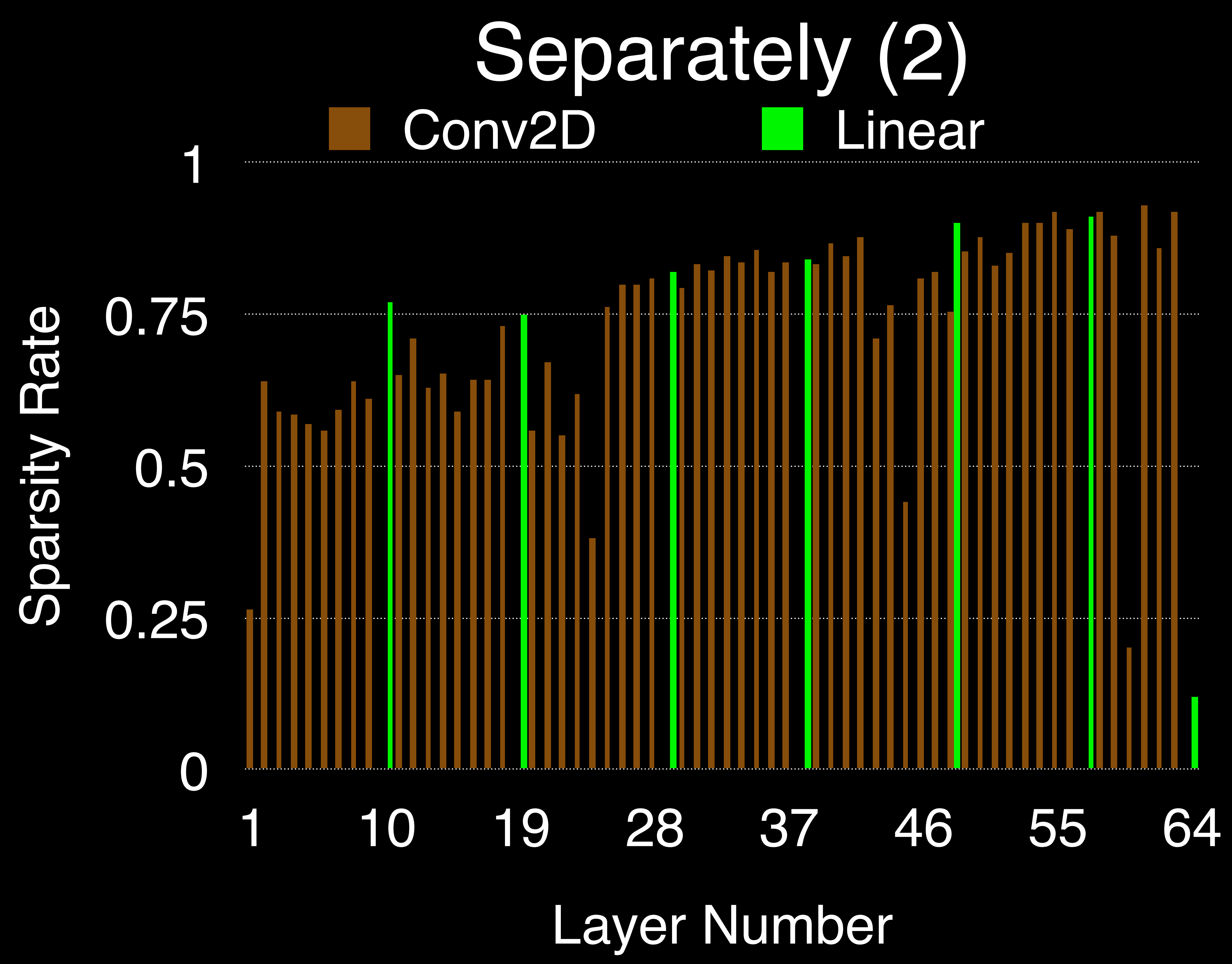
- Prune backbone
- Finetune backbone
- Repeat
- Once done, separately prune exit layers
- Finetune
- Repeat

How to prune the early exit weights?

Pruning Early Exit Networks



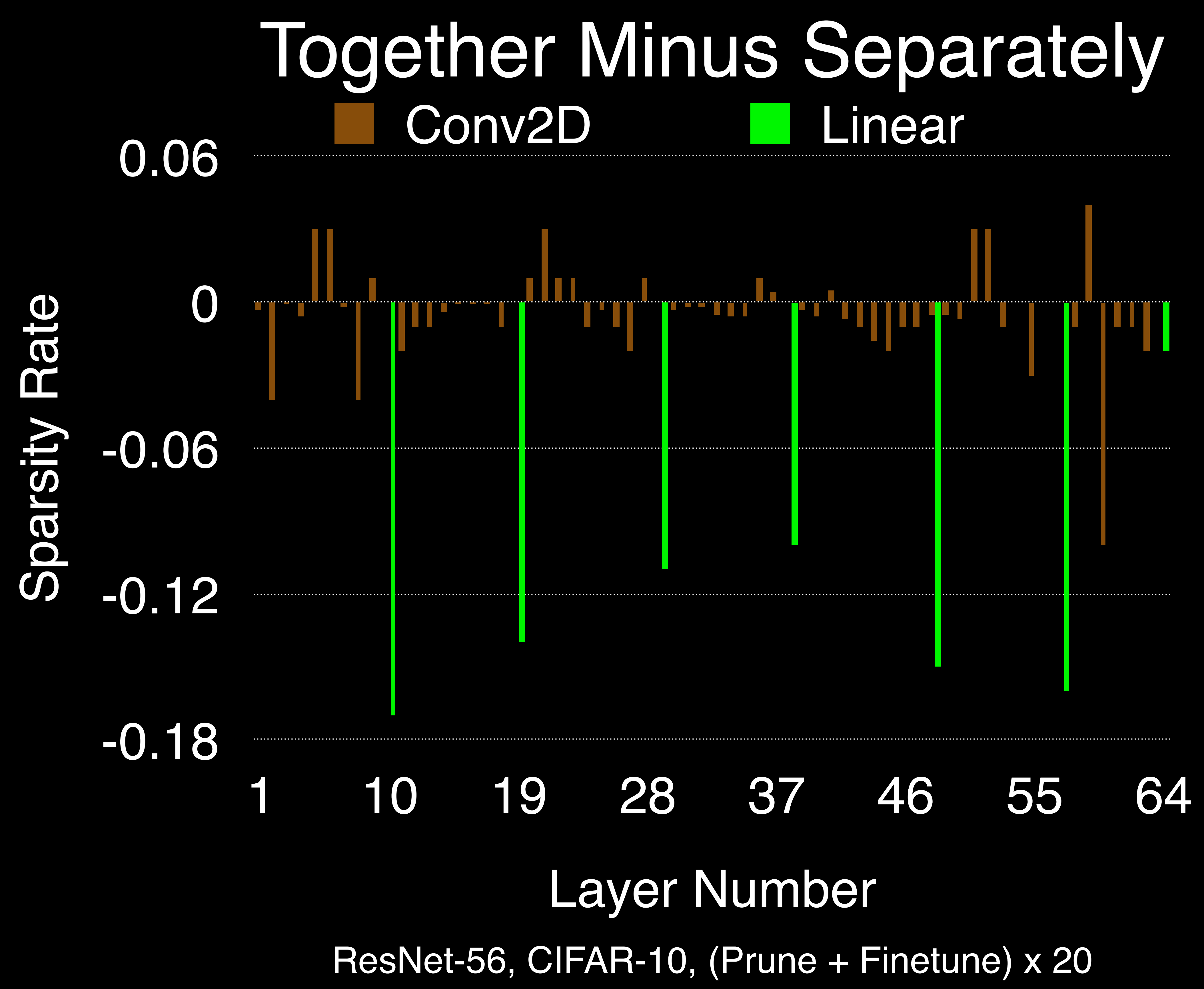
ResNet-56, CIFAR-10, (Prune + Finetune) x 20



ResNet-56, CIFAR-10, (Prune + Finetune) x 20

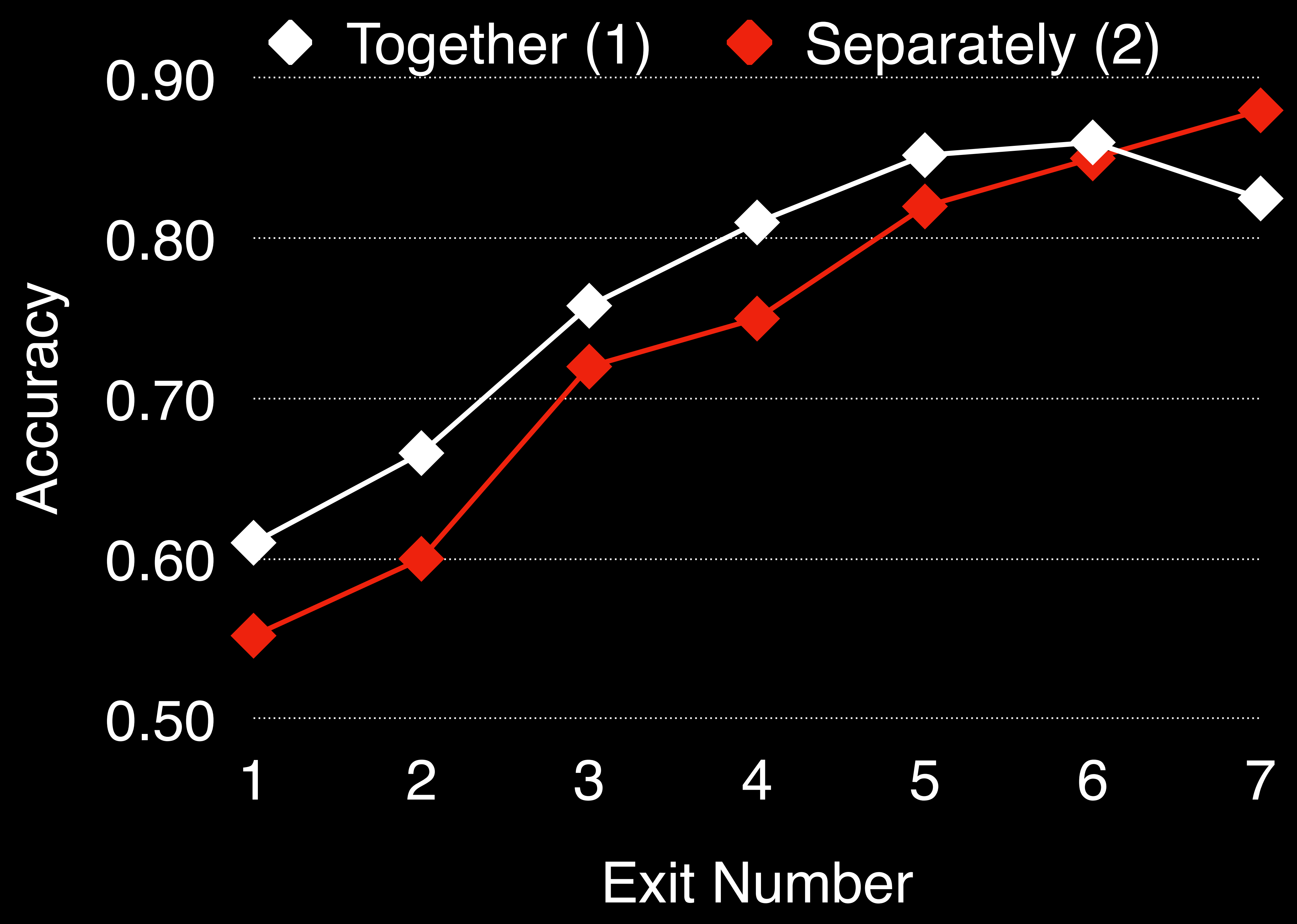
How to prune the early exit weights?

Pruning Early Exit Networks



Pruning backbone & exit layers separately leads to sparser exit weights.

Pruning Early Exit Networks



ResNet-56, CIFAR-10, (Prune + Finetune) x 20

Pruning everything together gives the best outcome.

Thesis Contributions

1. Designed E^2CM, a simple and lightweight early exit algorithm to reduce inference cost.

2. Demonstrated how early exit networks can be combined with model pruning.

3. Designed CBT, a new algorithm to further decrease the inference cost of early exit semantic segmentation networks.

Class Based Thresholding in Early Exit Semantic Segmentation Networks

A. Görmez and E. Koyuncu, "Class Based Thresholding in Early Exit Semantic Segmentation Networks," *IEEE Signal Processing Letters*, vol. 31, pp. 1184-1188, 2024. Also presented in IEEE MLSP 2024.

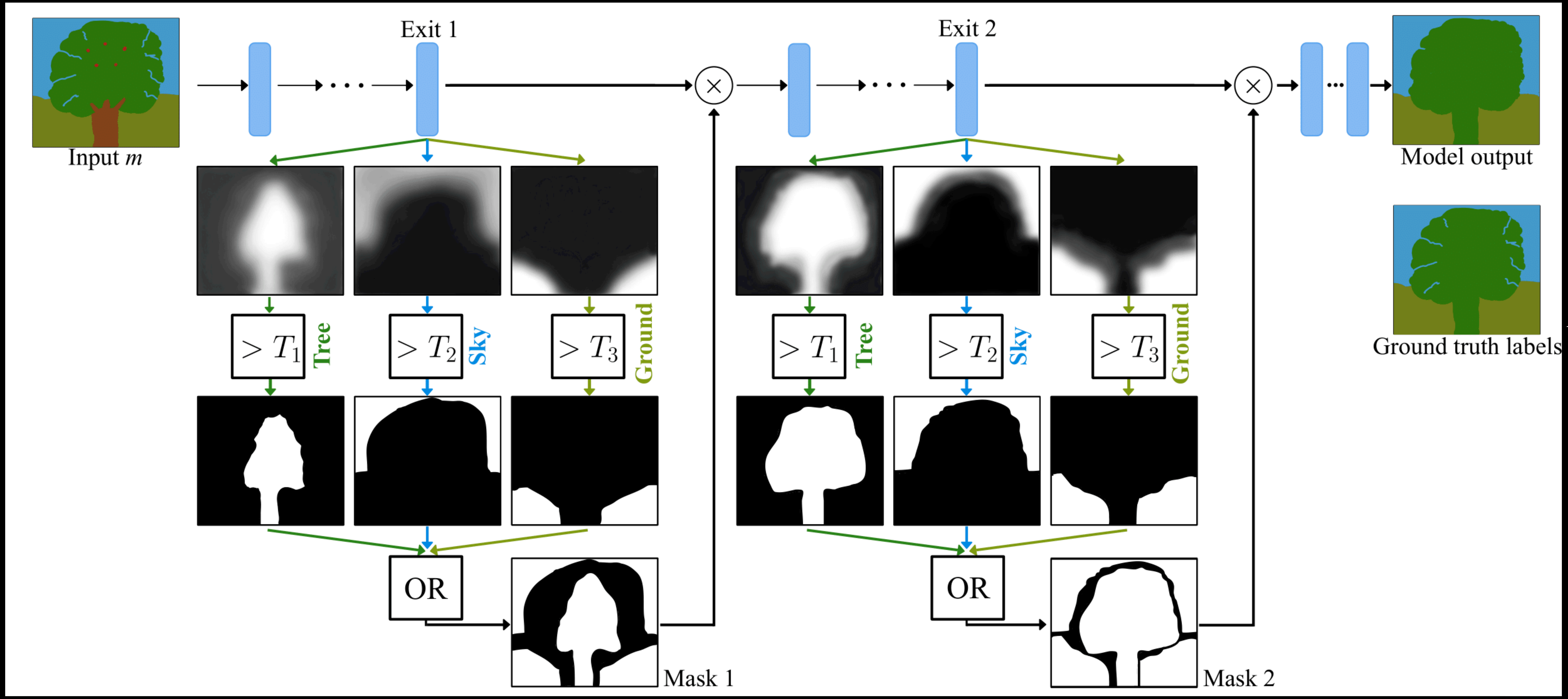
Class Based Thresholding in Early Exit Semantic Segmentation Networks

$T = 0.52$

Class Based Thresholding in Early Exit Semantic Segmentation Networks

Not all classes have the same classification difficulty.

Class Based Thresholding in Early Exit Semantic Segmentation Networks



$n \in \{1,2,\ldots,N\}$: The exit layer.

$k \in \{1,2,\ldots,K\}$: The pixel class.

$\phi_n(\cdot)$: Prob. vec. for pixel at exit n, $\in [0,1]^K$

$\|S_k\|$: Set of all pixels with ground truth class k.

$$p_{n,k} = \frac{1}{\|S_k\|} \sum_{(\cdot)\in S_k} \phi_n(\cdot)$$

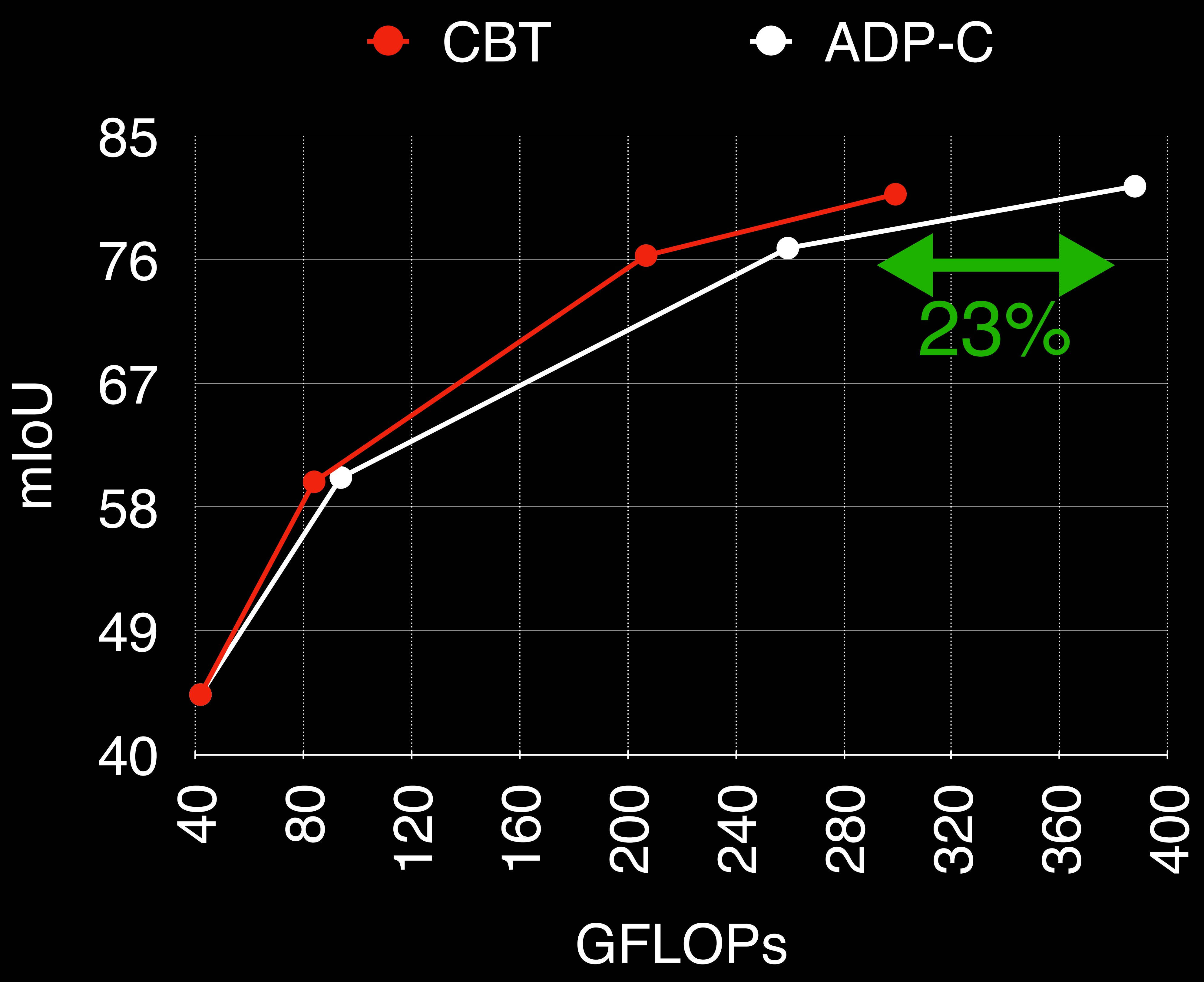
$$P_k = \frac{1}{N} \sum_{n=1}^{N} p_{n,k}$$

$$T_k = max(P_k,1) - max(P_k,2)$$

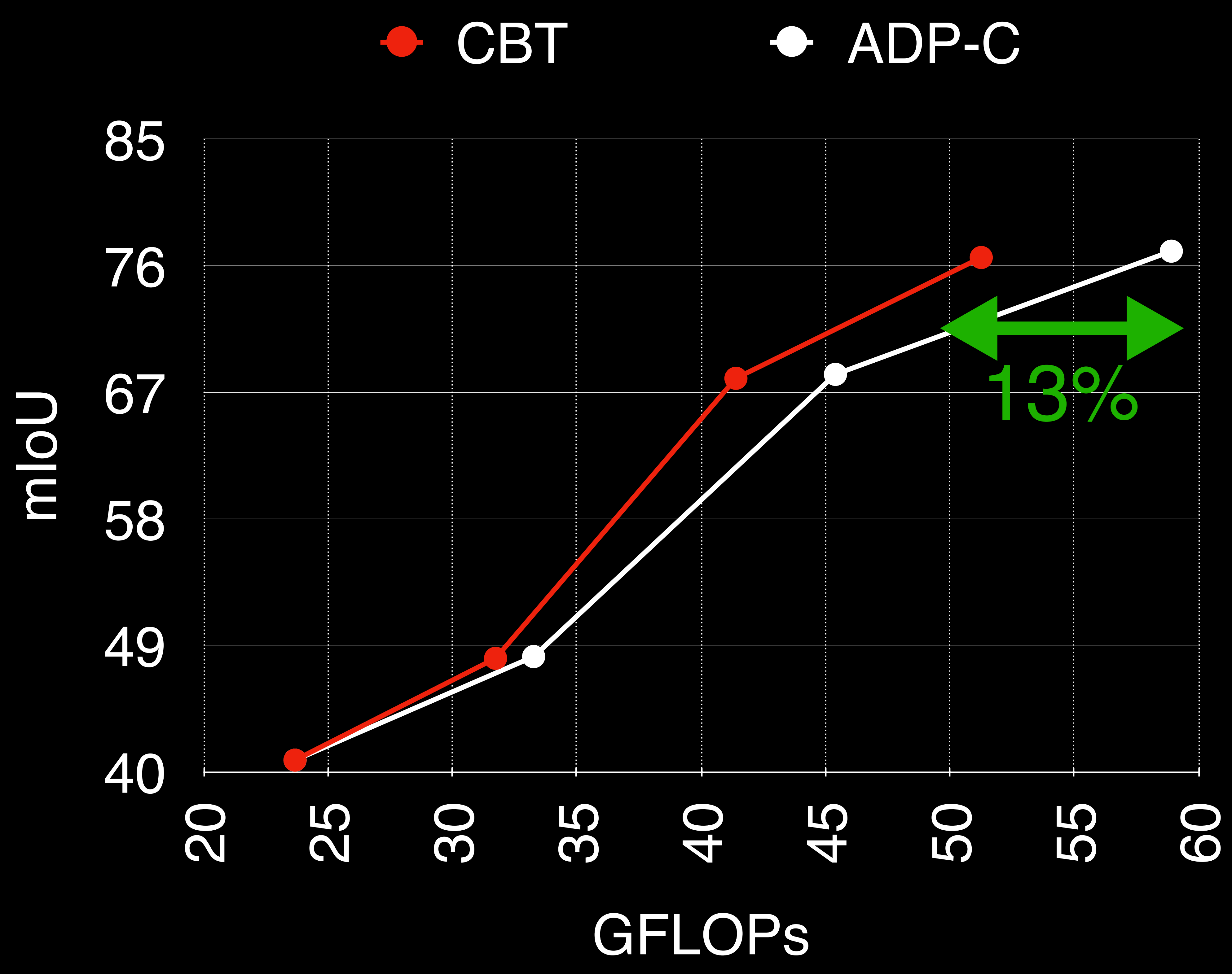
$$T_k \leftarrow \left(1 - \frac{T_k - \min T}{\max T - \min T}\right)(\beta - \alpha) + \alpha$$

CBT utilizes different thresholds considering the varying levels of inherent difficulty.

Class Based Thresholding in Early Exit Semantic Segmentation Networks



HRNetV2-W48, Cityscapes, [0.99, 0.998]



HRNetV2-W18, Cityscapes, [0.99, 0.998]

Class Based Thresholding in Early Exit Semantic Segmentation Networks

Method	Model	Exit 1 mIoU	Exit 1 GFLOPs	Exit 2 mIoU	Exit 2 GFLOPs	Exit 3 mIoU	Exit 3 GFLOPs	Exit 4 mIoU	Exit 4 GFLOPs
ADP-C	HRNetV2-W48	4.12	6.20	5.16	15.42	12.15	52.47	42.82	100.28
CBT [0.9, 0.998]	HRNetV2-W48	4.12	6.20	5.15	15.07	12.09	50.48	41.85	94.31
CBT-ns [0.9, 0.998]	HRNetV2-W48	4.12	6.20	5.15	15.06	12.08	50.48	41.87	94.34
CBT [0.8, 0.998]	HRNetV2-W48	4.12	6.20	5.14	14.80	11.90	48.81	40.17	90.25
CBT [0.7, 0.998]	HRNetV2-W48	4.12	6.20	5.12	14.55	11.58	47.27	37.54	86.52
ADP-C	HRNetV2-W18	4.89	5.88	6.83	7.84	8.94	12.73	9.74	19.04
CBT [0.9, 0.998]	HRNetV2-W18	4.89	5.88	6.80	7.73	10.07	12.24	11.78	17.89
CBT [0.8, 0.998]	HRNetV2-W18	4.89	5.88	6.75	7.67	10.17	11.98	11.95	17.26
CBT [0.7, 0.998]	HRNetV2-W18	4.89	5.88	6.70	7.62	10.09	11.75	11.88	16.71

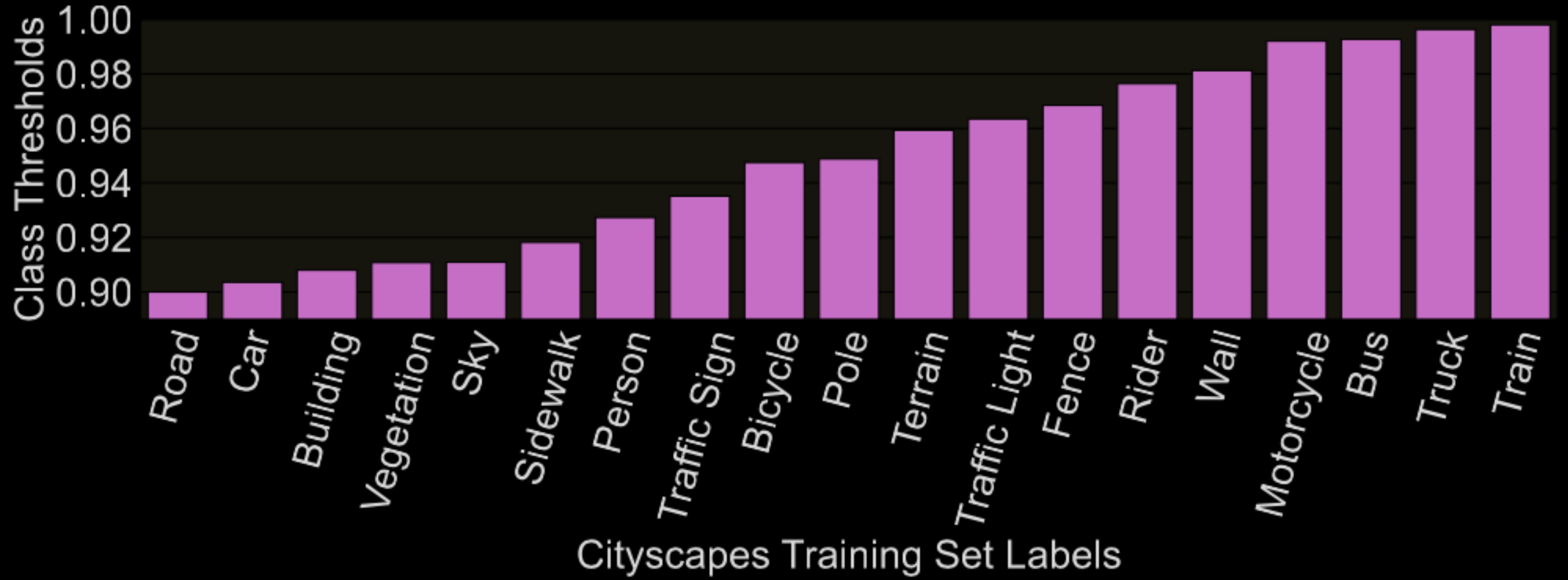
Results on ADE20K.

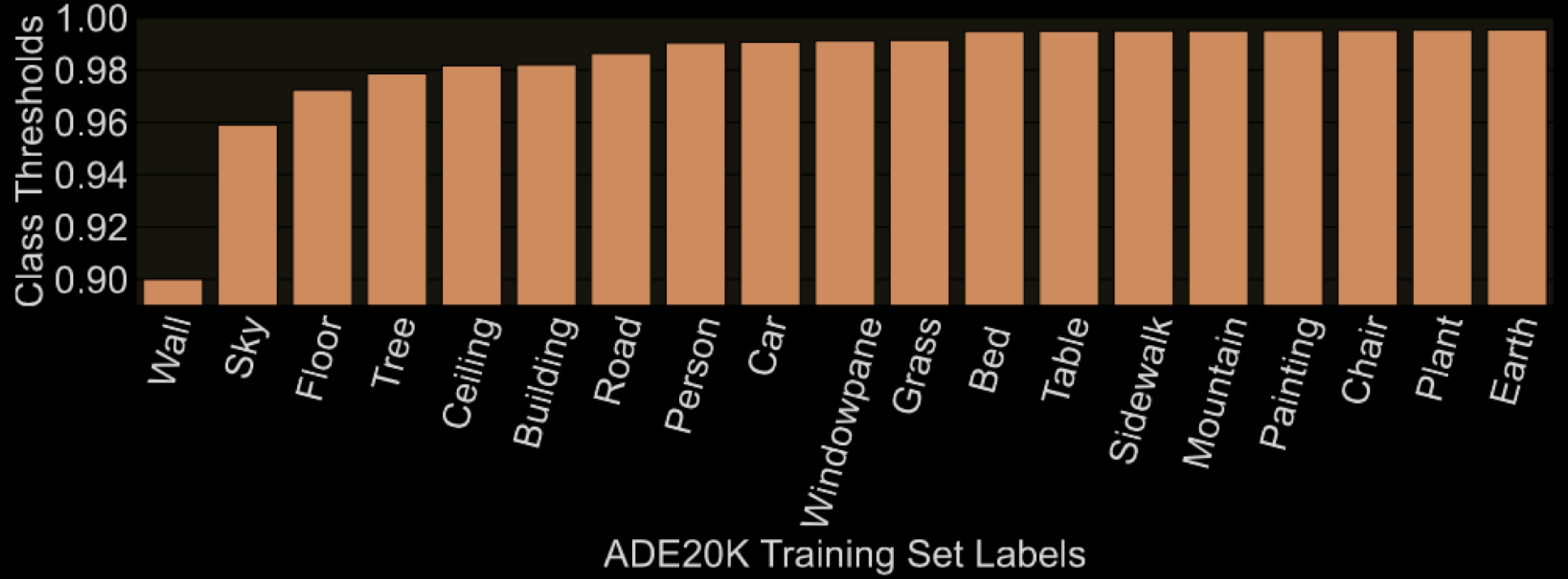
Class Based Thresholding in Early Exit Semantic Segmentation Networks

Method	Dataset	Model	Exit 1 mIoU	Exit 1 GFLOPs	Exit 2 mIoU	Exit 2 GFLOPs	Exit 3 mIoU	Exit 3 GFLOPs
DToP	ADE20K	ViT-Base	41.79	55.70	45.85	66.60	49.21	83.52
CBT [0.85, 0.9]	ADE20K	ViT-Base	41.79	55.70	45.52	65.60	49.04	80.80
DToP	ADE20K	ViT-Large	37.86	208.96	47.97	352.32	52.18	452.3
CBT [0.9, 0.95]	ADE20K	ViT-Large	37.86	208.96	47.82	336.01	51.69	421.93
DToP	COCOStuff10K	ViT-Large	31.89	124.94	41.71	205.14	45.64	266.17
CBT [0.9, 0.95]	COCOStuff10K	ViT-Large	31.89	124.94	41.09	197.53	45.29	252.04

Comparison of CBT against Dynamic Token Pruning (DToP).

Class Based Thresholding in Early Exit Semantic Segmentation Networks





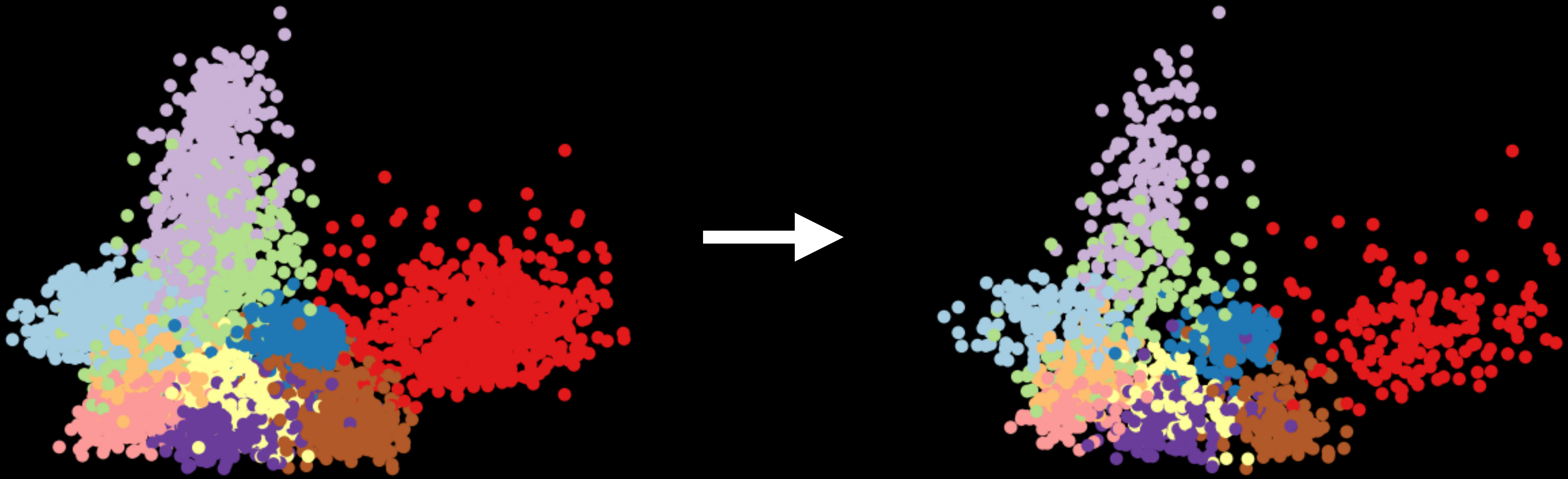
Thesis Contributions

1. Designed E^2CM, a simple and lightweight early exit algorithm to reduce inference cost.
2. Introduced EEPrune, a novel dataset pruning algorithm that uses early exit networks to reduce training cost.
3. Designed CBT, a new algorithm to further decrease the inference cost of early exit semantic segmentation networks.
4. Introduced EEPrune, a novel dataset pruning algorithm that uses early exit networks to reduce training cost.

Dataset Pruning Using Early Exit Networks

A. Görmez and E. Koyuncu, "Dataset Pruning Using Early Exit Networks," *ICML Workshop on Localized Learning (LLW)*, 2023. Also presented in Cohere for AI ML Efficiency Group, and in Mediterranean Machine Learning Summer School.

Dataset Pruning Using Early Exit Networks

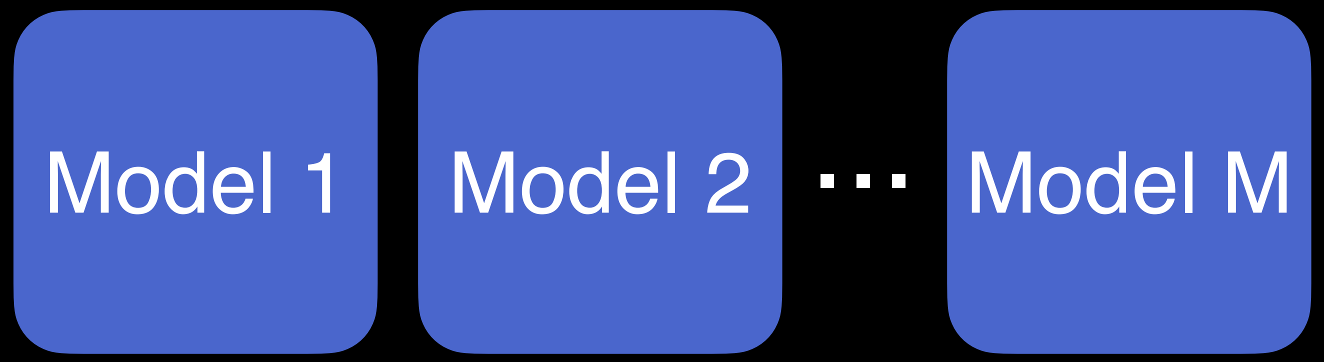


Reduce training set size.

Dataset Pruning Using Early Exit Networks

Dataset Pruning Using Early Exit Networks

Existing dataset pruning algorithms:



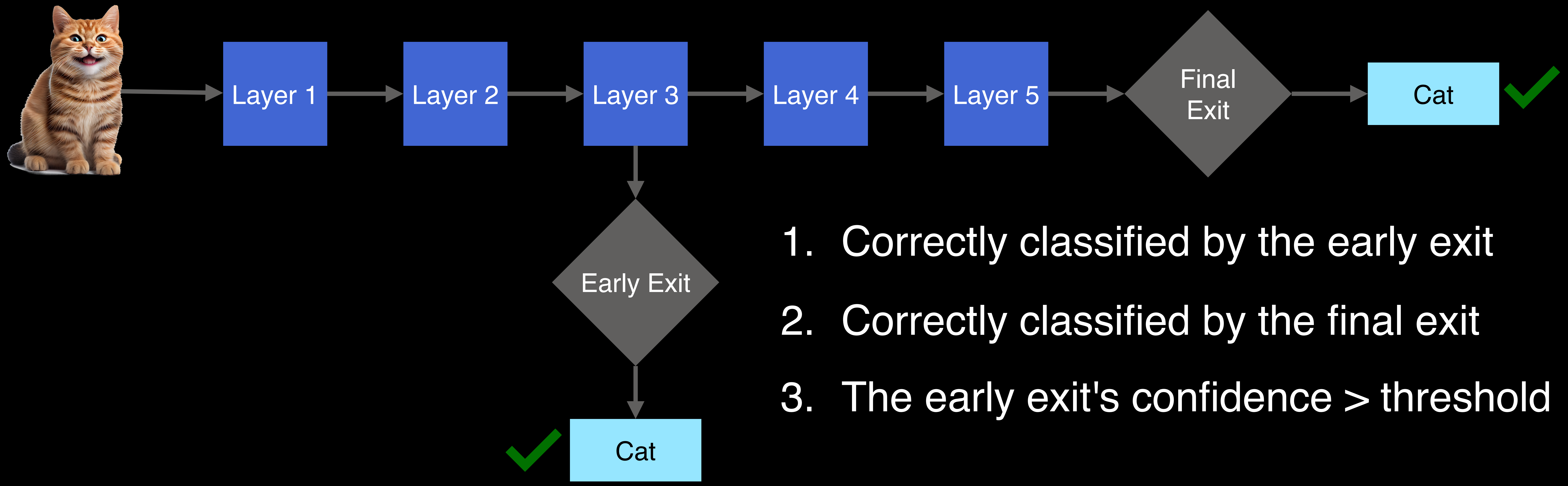
Train ensemble of models.

Perform full training on the original dataset.

Cannot beat random pruning.

Dataset Pruning Using Early Exit Networks

After a <u>short</u> period of training an early exit network on the entire dataset, prune a sample if:

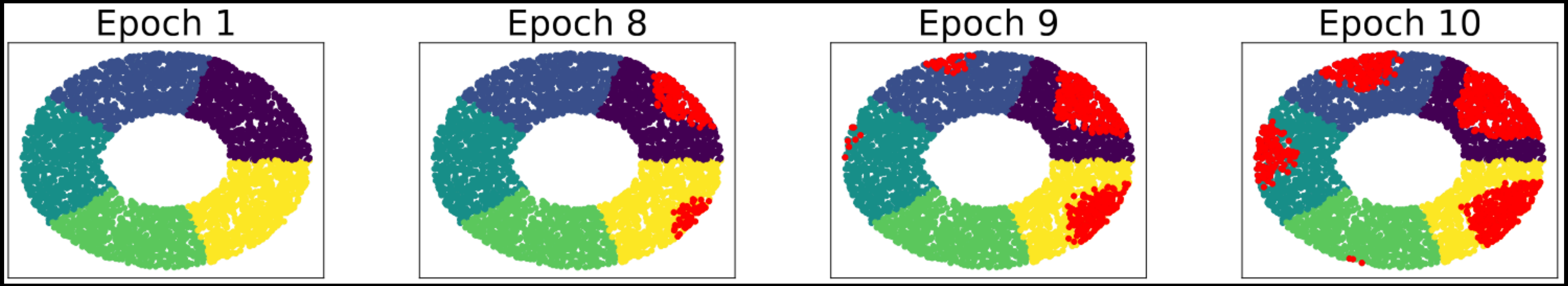


1. Correctly classified by the early exit
2. Correctly classified by the final exit
3. The early exit's confidence > threshold

$0.71 > T = 0.52$

Early exit networks can detect easy samples.

Dataset Pruning Using Early Exit Networks



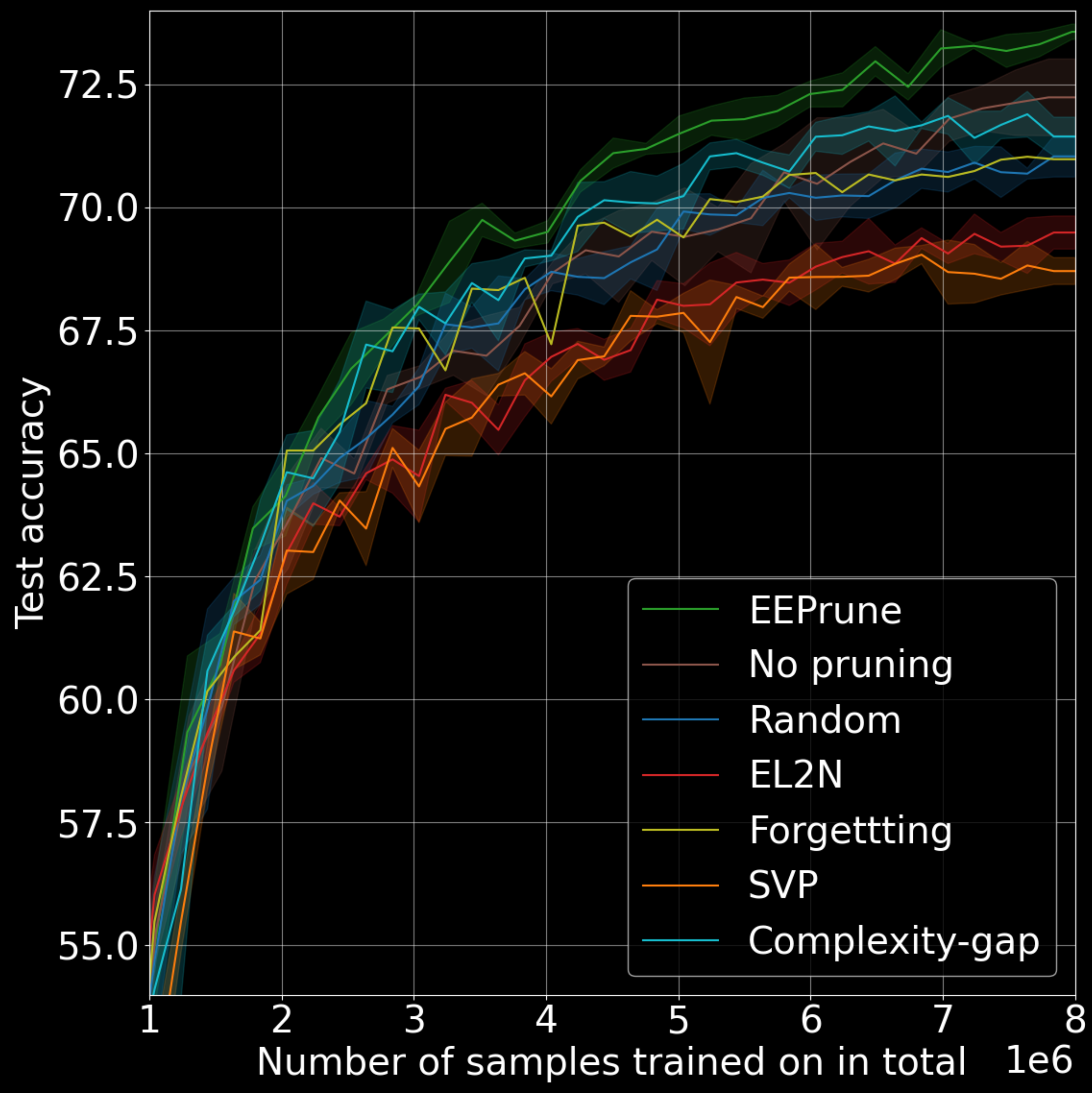
EEPrune discards samples that are furthest away from the decision boundaries.

Dataset Pruning Using Early Exit Networks

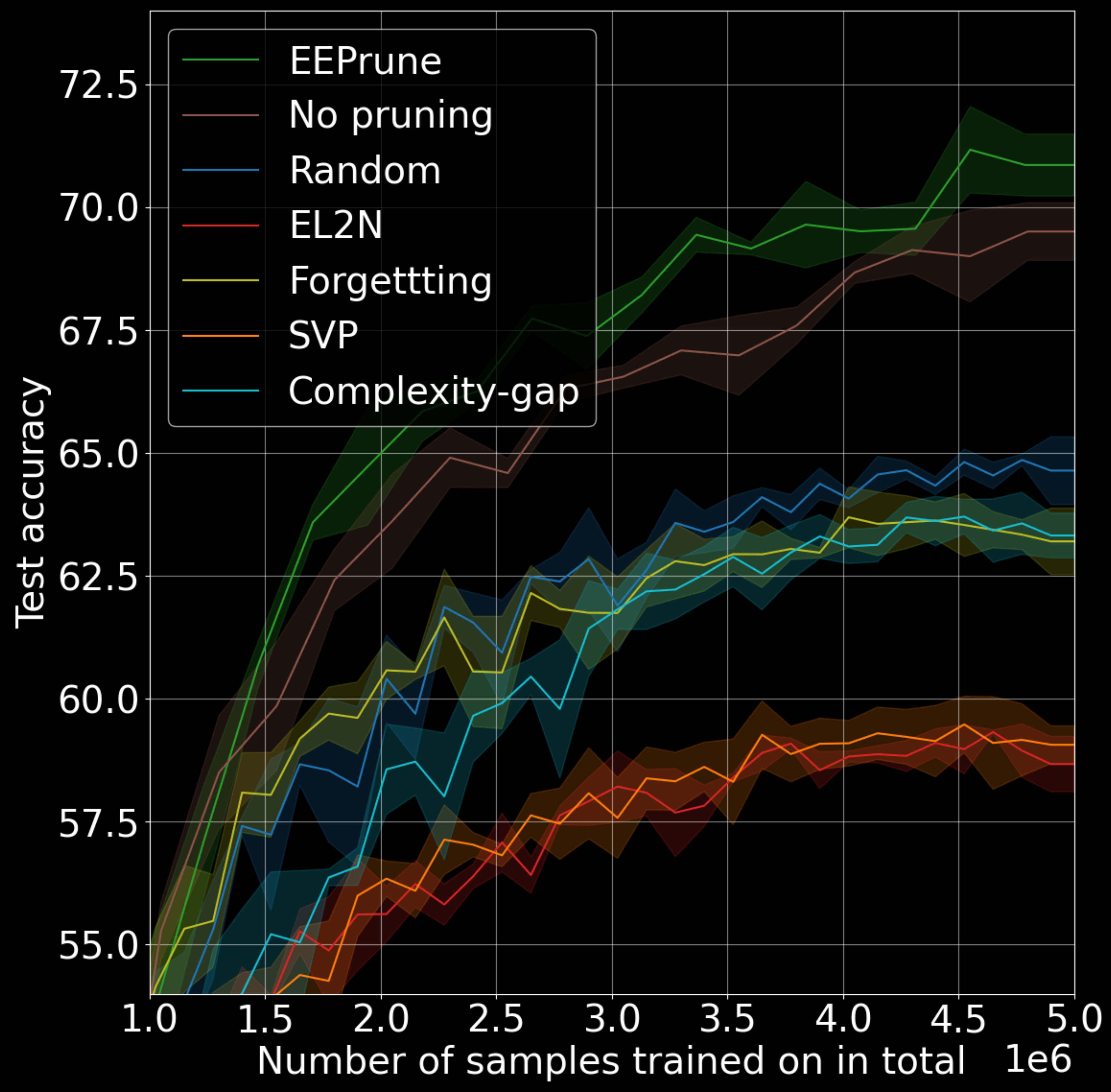
Experiment axis	Choices
Methods	EEPrune, No pruning, Random, EL2N, Forgetting, SVP, Complexity gap
Datasets	CIFAR-10, CIFAR-100, Tiny ImageNet, KMNIST, ImageNet
Models	EfficientNetV2-M, MobileNetV3-large, ResNet-50
Pruning ratios	10%, 20%, 30%, 40%, 50%, 60%
Metrics	Top-1 accuracy, cumulative number of samples seen
Number of repeats	3

Summary of the experiments.

Dataset Pruning Using Early Exit Networks

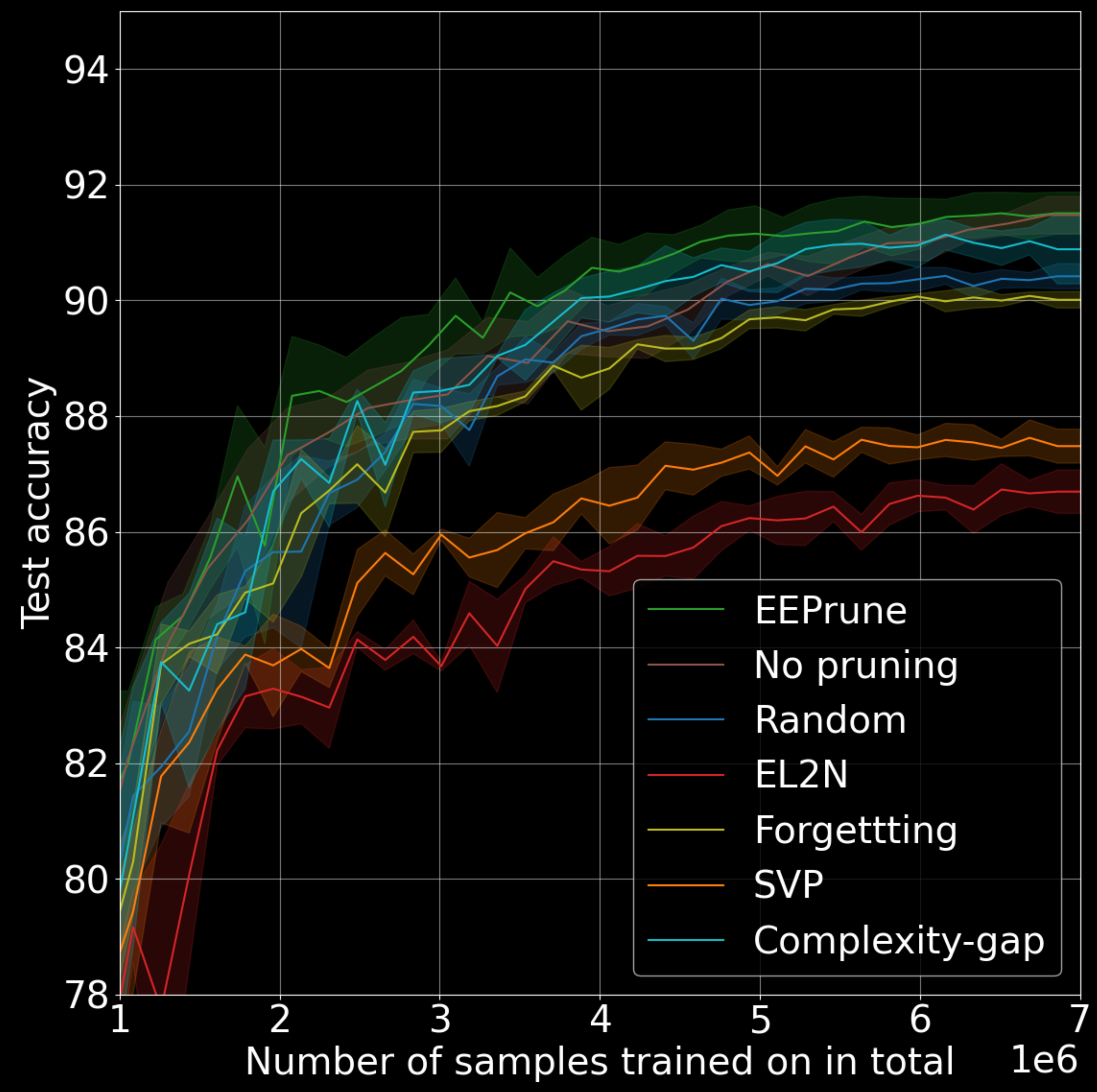


MobileNetV3-large, CIFAR-100, 20%

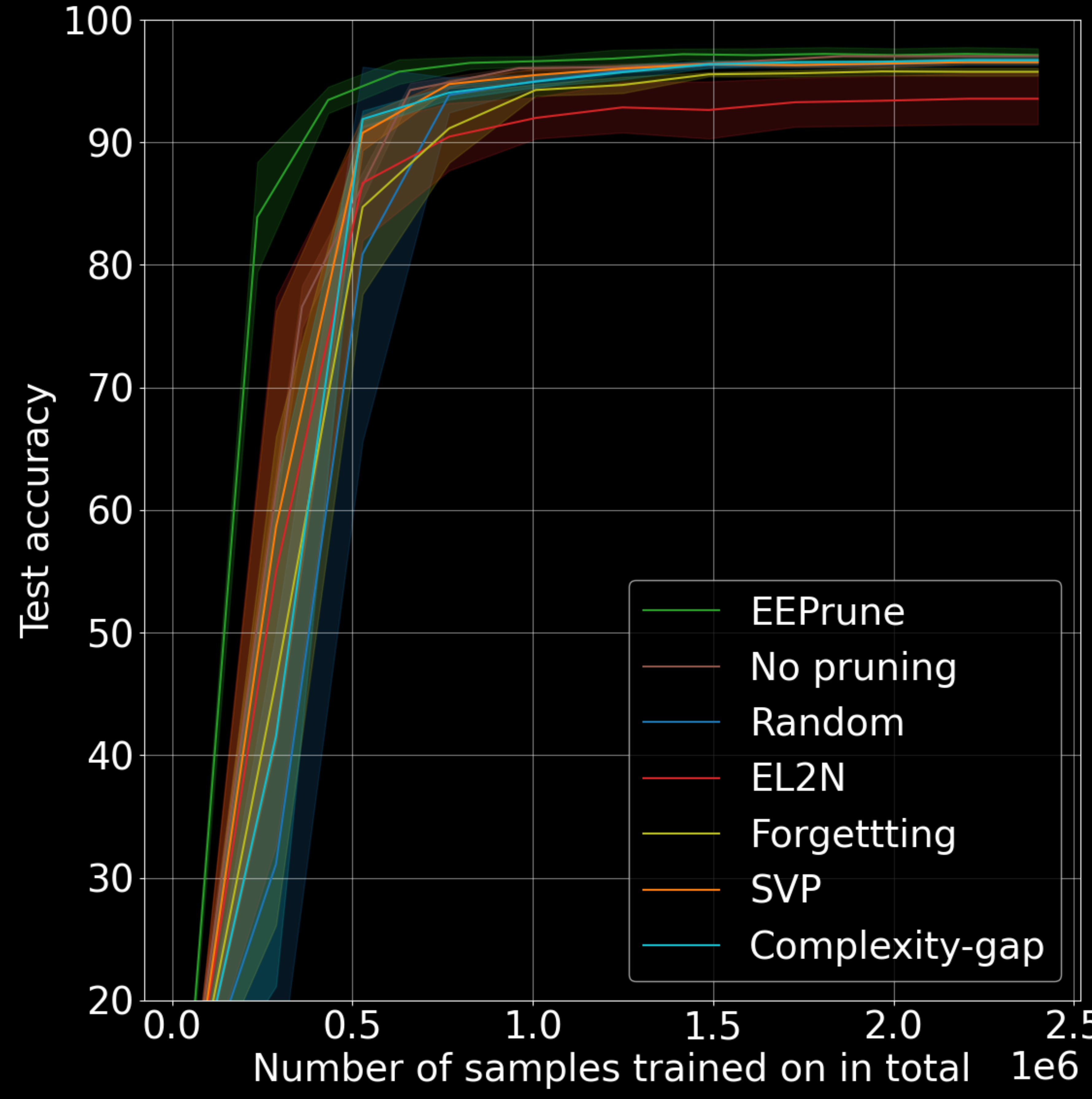


MobileNetV3-large, CIFAR-100, 50%

Dataset Pruning Using Early Exit Networks

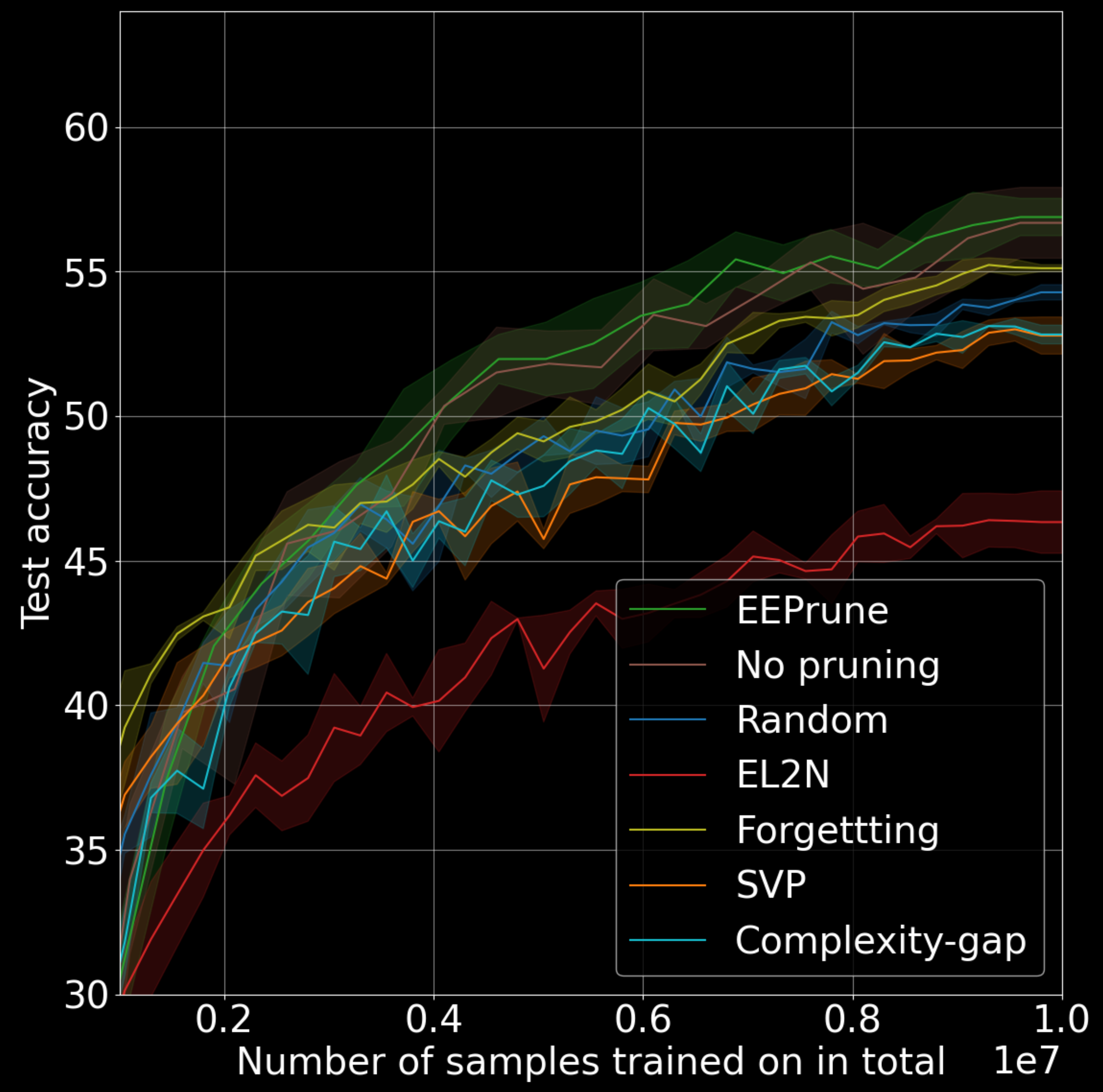


MobileNetV3-large, CIFAR-10, 30%

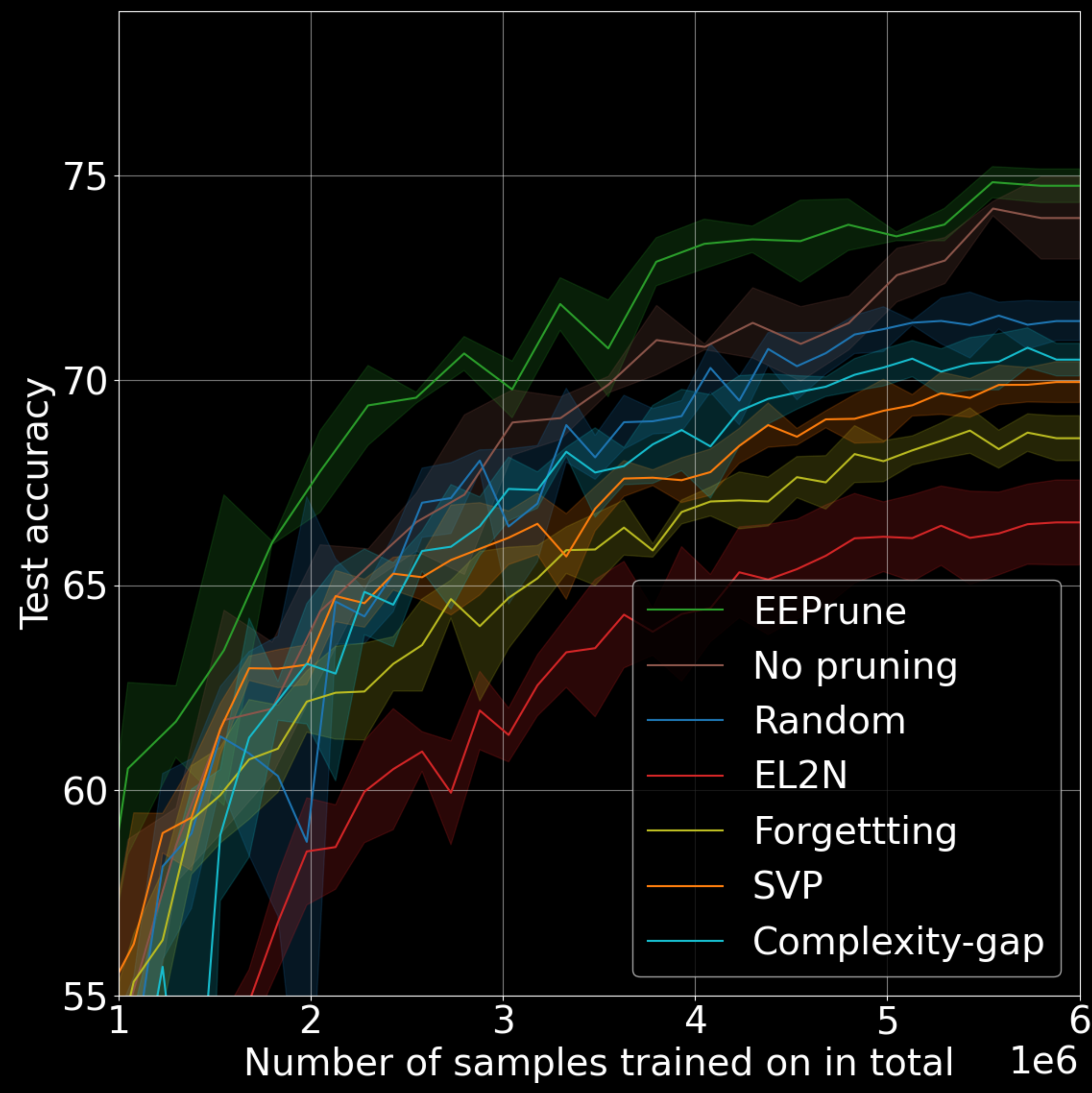


ResNet-50, KMNIST, 20%

Dataset Pruning Using Early Exit Networks



MobileNetV3-large, Tiny ImageNet, 50%



ResNet-50, CIFAR-100, 40%

Dataset Pruning Using Early Exit Networks

EEPrune consumes less energy than the other dataset pruning methods.

Dataset Pruning Using Early Exit Networks

Dataset Pruning Using Early Exit Networks

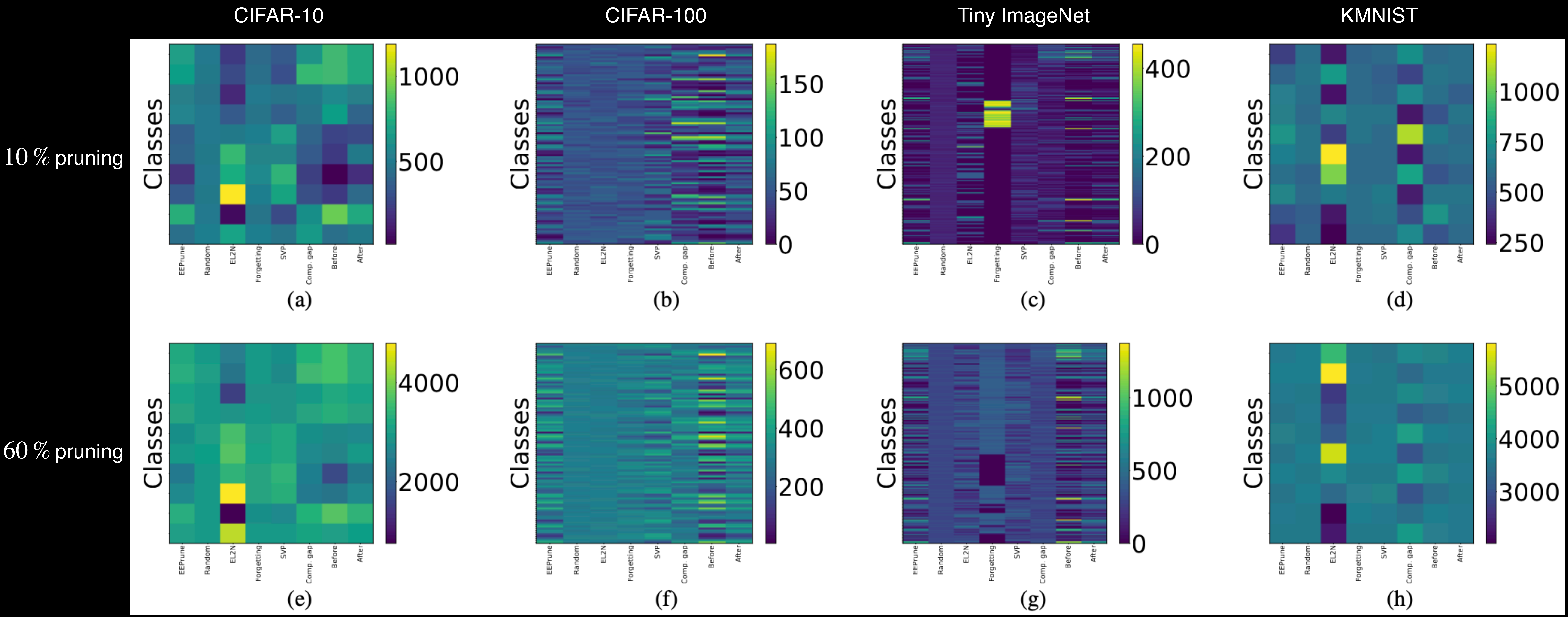
MobileNetV3-Large performance when the model is trained on D_p instead of $D_{tr} \setminus D_p$ for 50% pruning with EEPrune.

Dataset Pruning Using Early Exit Networks

Exit Location	**Pruning Ratio**		
	10%	**30%**	**60%**
Before	67.98 ± 0.37	67.66 ± 0.45	67.59 ± 0.56
Mid	$\mathbf{73.39 \pm 0.48}$	$\mathbf{72.32 \pm 0.41}$	$\mathbf{68.52 \pm 0.77}$
After	65.57 ± 0.37	67.01 ± 0.59	67.57 ± 0.48

Comparison of exit locations for EEPrune on MobileNetV3-large and CIFAR-100.

Dataset Pruning Using Early Exit Networks



Number of samples each dataset pruning method discards from the training set shown as heat map for MobileNetV3-large.
The 8 columns correspond to EEPrune, Random, EL2N, Forgetting, SVP, Complexity Gap, EEPrune-Before and EEPrune-After.

Thesis Contributions

1. Designed E^2CM, a simple and lightweight early exit algorithm to reduce inference cost.

2. Demonstrated how early exit networks can be combined with model pruning.

3. Designed CBT, a new algorithm to further decrease the inference cost of early exit semantic segmentation networks.

4. Introduced EEPrune, a novel dataset pruning algorithm that uses early exit networks to reduce training cost.

5. Developed a novel class-aware weight initialization technique for early exit LLMs with the purpose of accelerating pre-training.

Class-aware Initialization of Early Exits for Pre-training Large Language Models

A. Görmez and E. Koyuncu, "Class-aware Initialization of Early Exits for Pre-training Large Language Models," *WANT@ICML*, 2024.

Class-aware Initialization of Early Exits for Pre-training Large Language Models

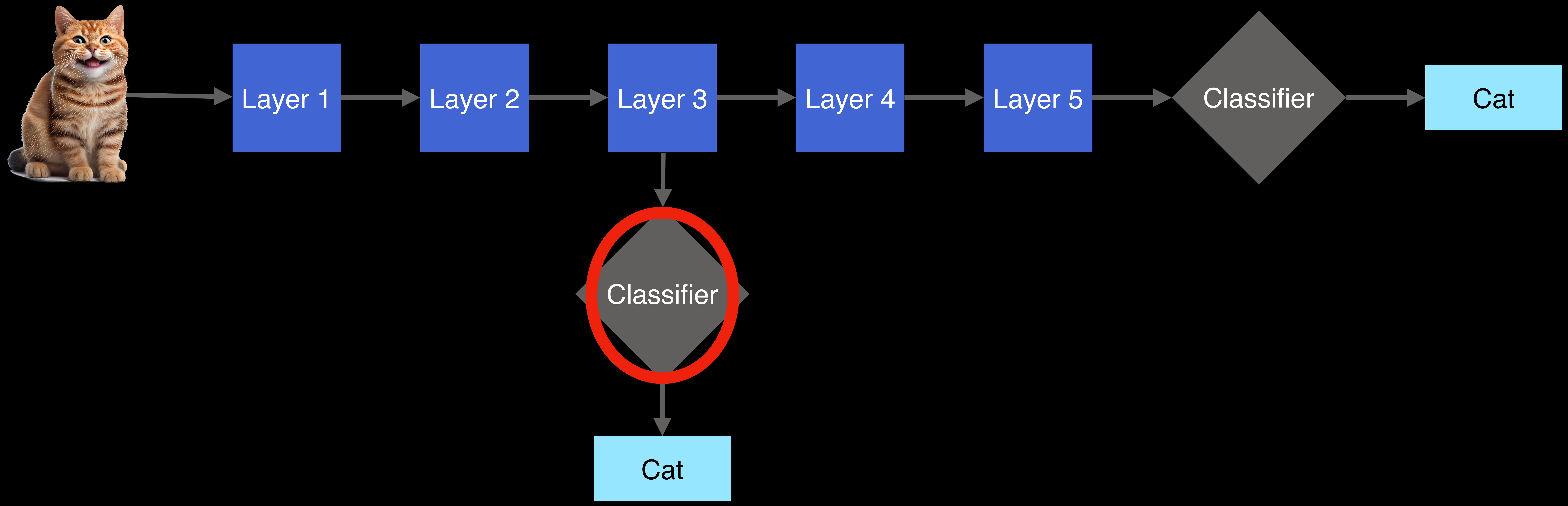
HuggingFace Open LLM Leaderboard

T	Model	Average	ARC	HellaSwag	MMLU
🔶	davidkim205/Rhea-72b-v0.5	81.22	79.78	91.15	77.95
💬	MTSAIR/MultiVerse_70B	81	78.67	89.77	78.22
🔶	MTSAIR/MultiVerse_70B	80.98	78.58	89.74	78.27
🔶	SF-Foundation/Ein-72B-v0.11	80.81	76.79	89.02	77.2
🔶	SF-Foundation/Ein-72B-v0.13	80.79	76.19	89.44	77.07
🔶	SF-Foundation/Ein-72B-v0.12	80.72	76.19	89.46	77.17
🔶	abacusai/Smaug-72B-v0.1	80.48	76.02	89.27	77.15
🔶	ibivibiv/alpaca-dragon-72b-v1	79.3	73.89	88.16	77.4
💬	moreh/MoMo-72B-lora-1.8.7-DPO	78.55	70.82	85.96	77.13
🔶	cloudyu/TomGrc_FusionNet_34Bx2_MoE_v0.1_DPO_f16	77.91	74.06	86.74	76.65
🔶	saltlux/luxia-21.4b-alignment-v1.0	77.74	77.47	91.88	68.1
🔶	cloudyu/TomGrc_FusionNet_34Bx2_MoE_v0.1_full_linear_DPO	77.52	74.06	86.67	76.69

~140 GB

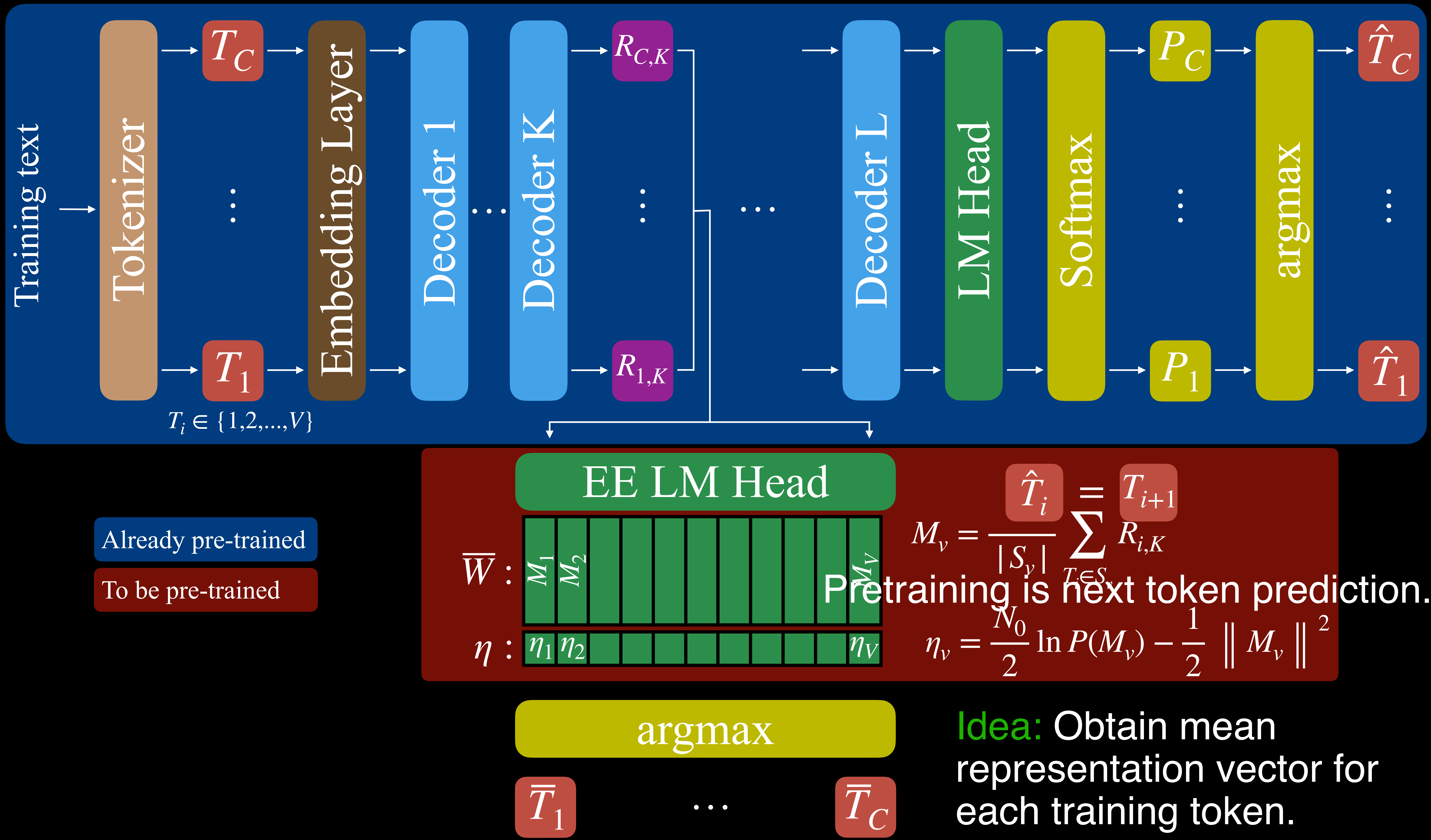
LLMs are too big.
Therefore, inference times are too high.

Class-aware Initialization of Early Exits for Pre-training Large Language Models



The early exit has to be trained.

Goal: Find a smart initialization, so we do not have to do much training.



Pretraining is next token prediction.

Idea: Obtain mean representation vector for each training token.

Class-aware Initialization of Early Exits for Pre-training Large Language Models

Optimal detection for the vector AWGN channel

$$r = s_m + n \qquad \in \mathbb{R}^N$$

$$m \in 1, \ldots, M$$

$$n_i \sim \mathcal{N}(0, \frac{N_0}{2})$$

$$\hat{m} = argmax\ P_m\, p(r \mid s_m)$$

$$\hat{m} = argmax\ P_m\, p_n(r - s_m)$$

$$\vdots$$

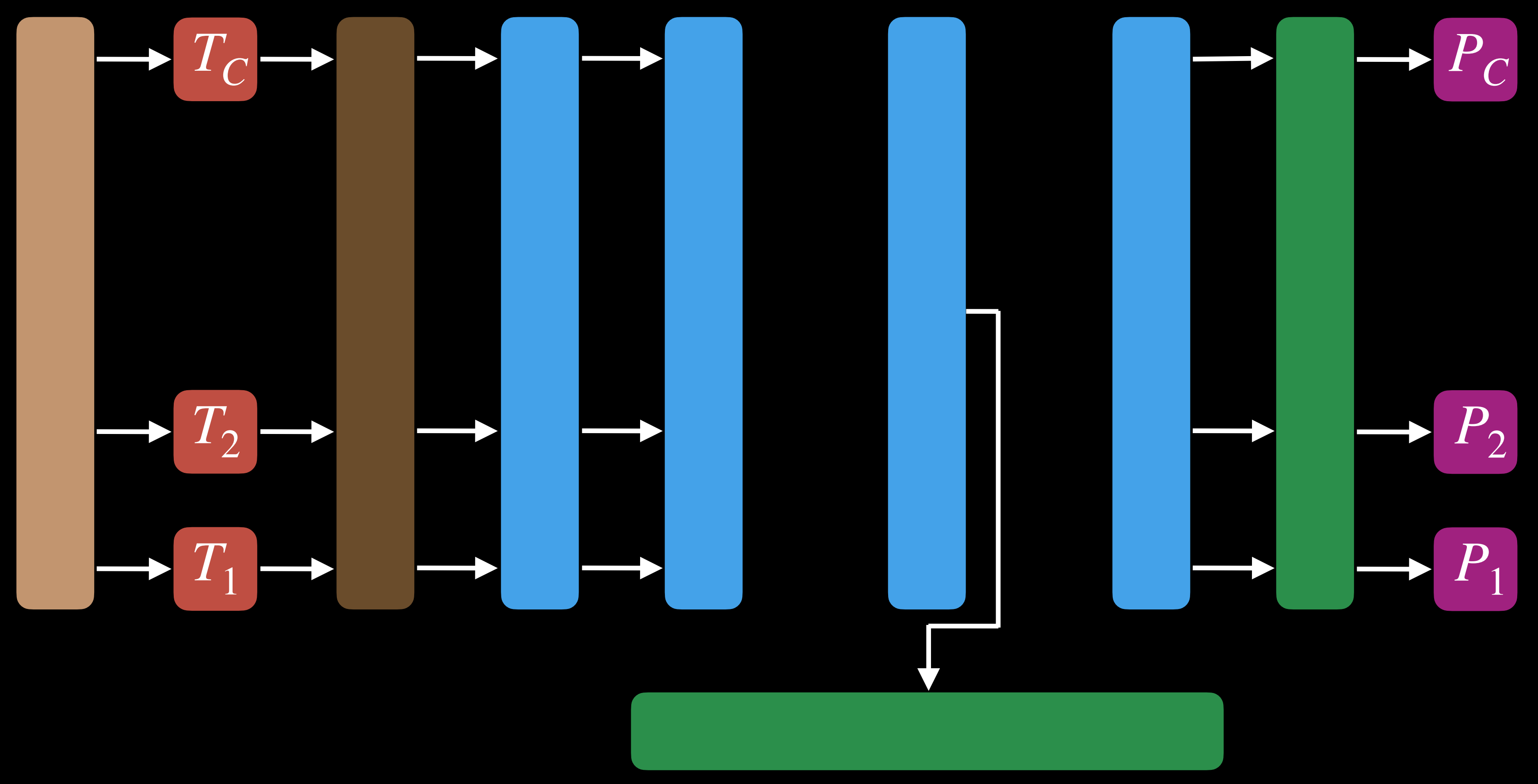
$$\hat{m} = argmax\ \left[\frac{N_0}{2} lnP_m - \frac{1}{2} \|r - s_m\|^2\right]$$

$$\hat{m} = argmax\ \left[\underline{\frac{N_0}{2} lnP_m - \frac{\|s_m\|^2}{2}} + r \cdot s_m\right]$$

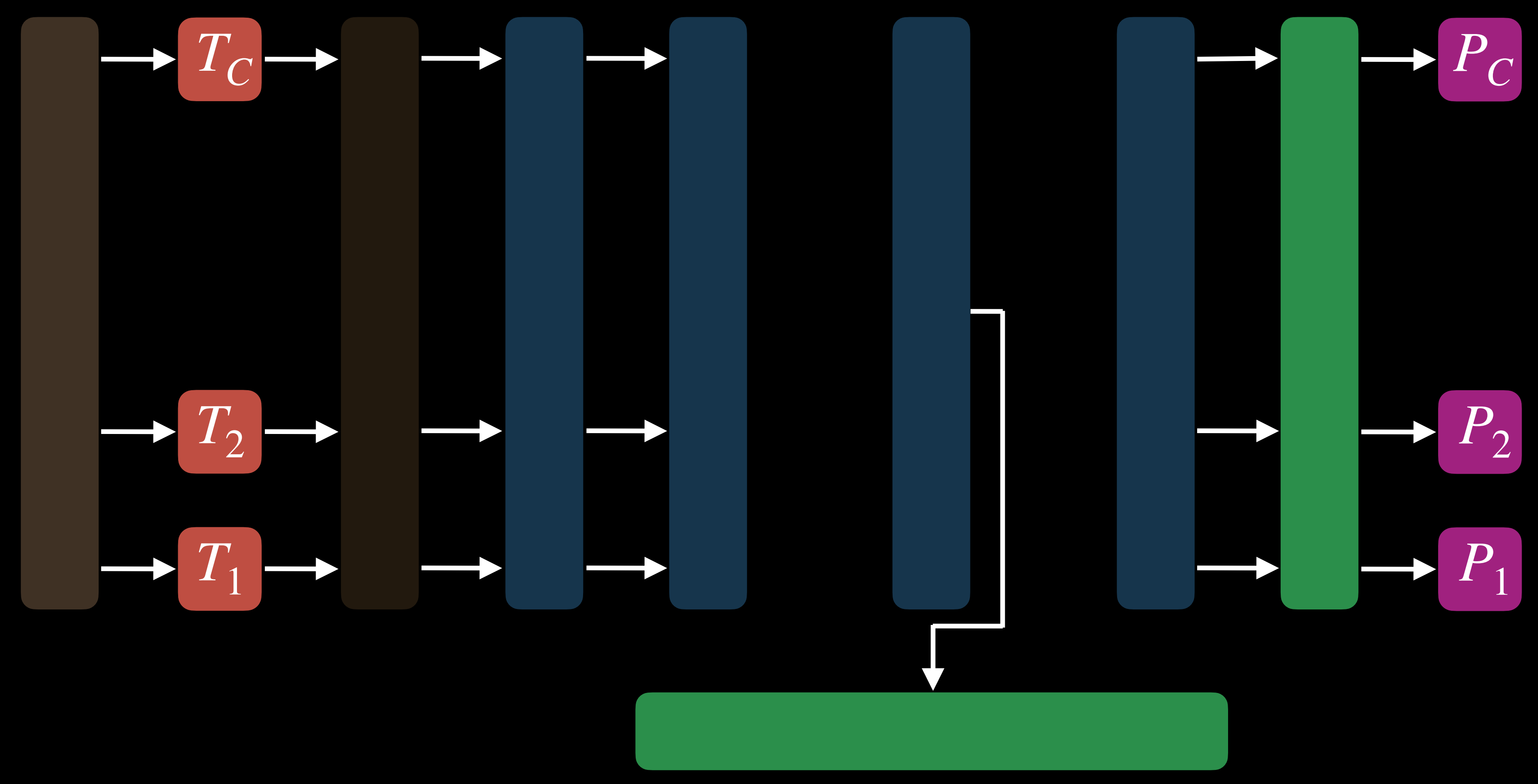
$$\frac{N_0}{2} \ln P(M_v) - \frac{1}{2} \parallel M_v \parallel^2 + R_{i,K} M_v$$

Class-aware Initialization of Early Exits for Pre-training Large Language Models

Two settings:

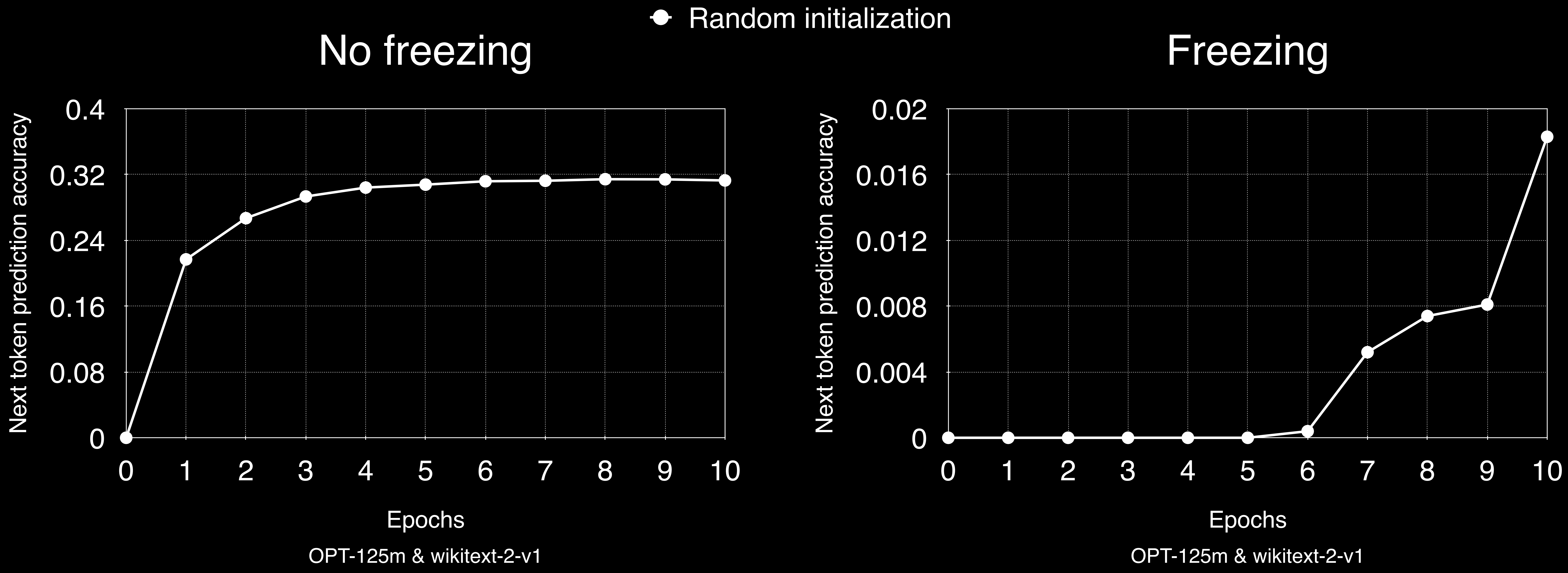


No freezing

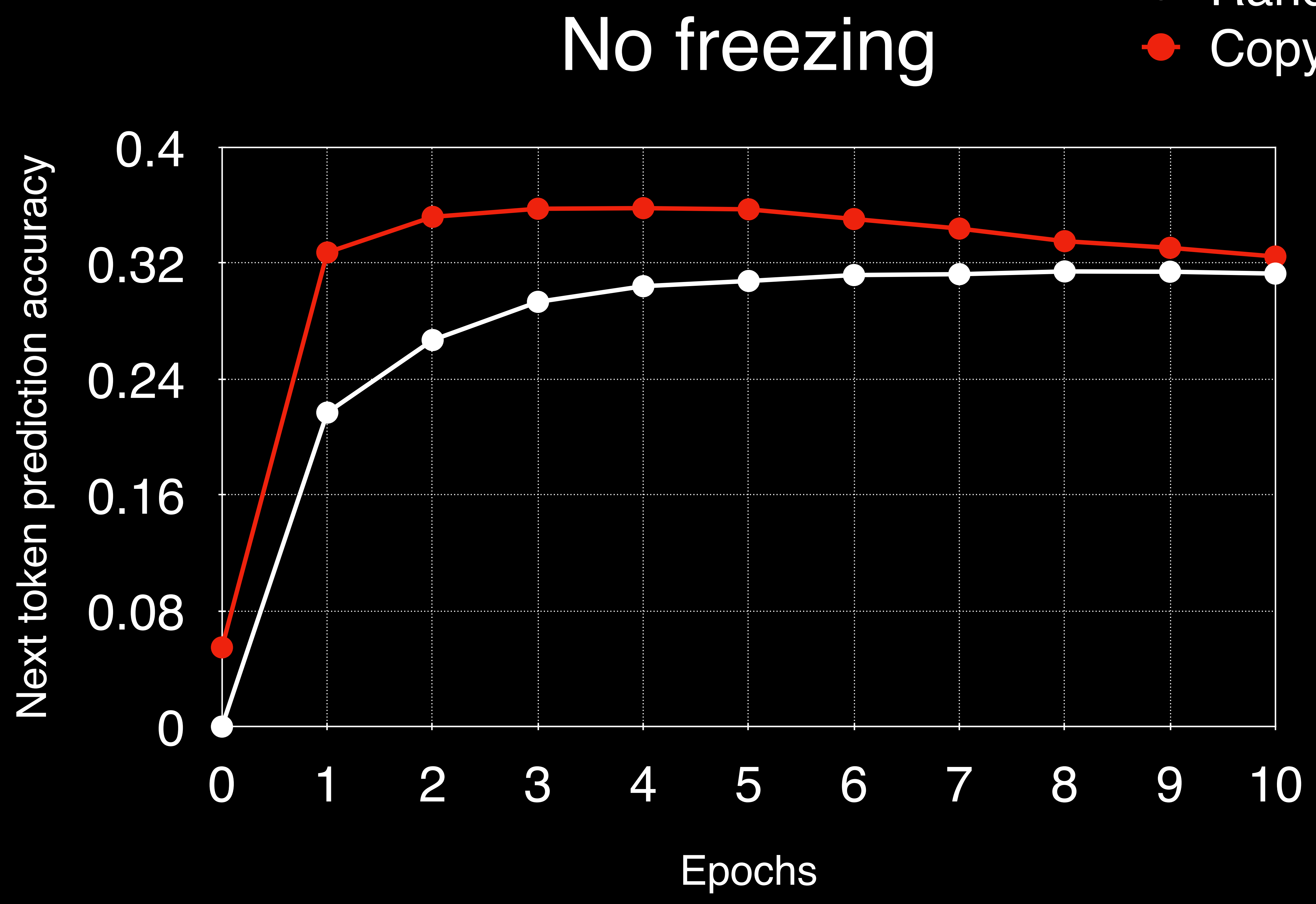


Freezing everything except LM Heads

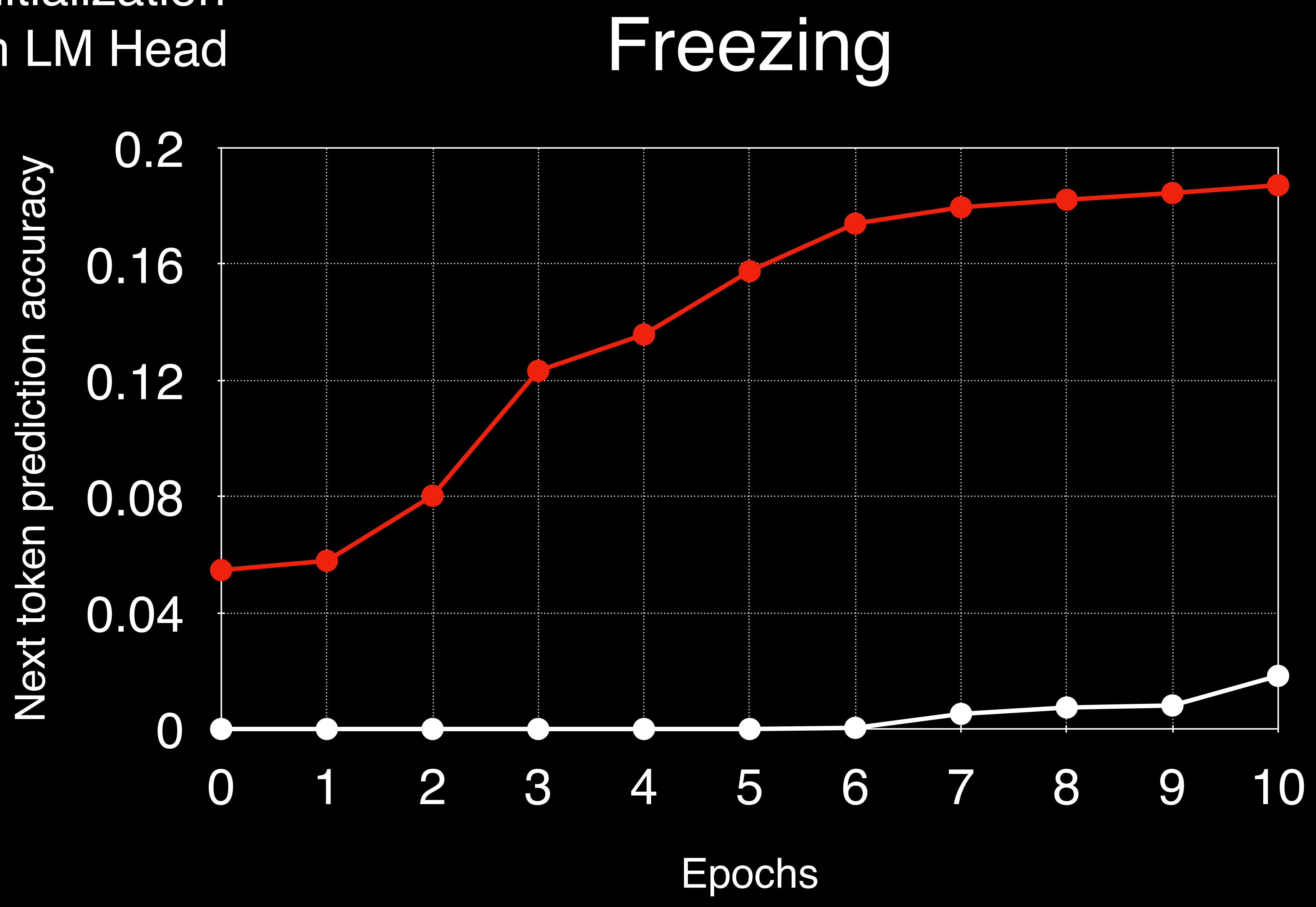
Class-aware Initialization of Early Exits for Pre-training Large Language Models



Class-aware Initialization of Early Exits for Pre-training Large Language Models

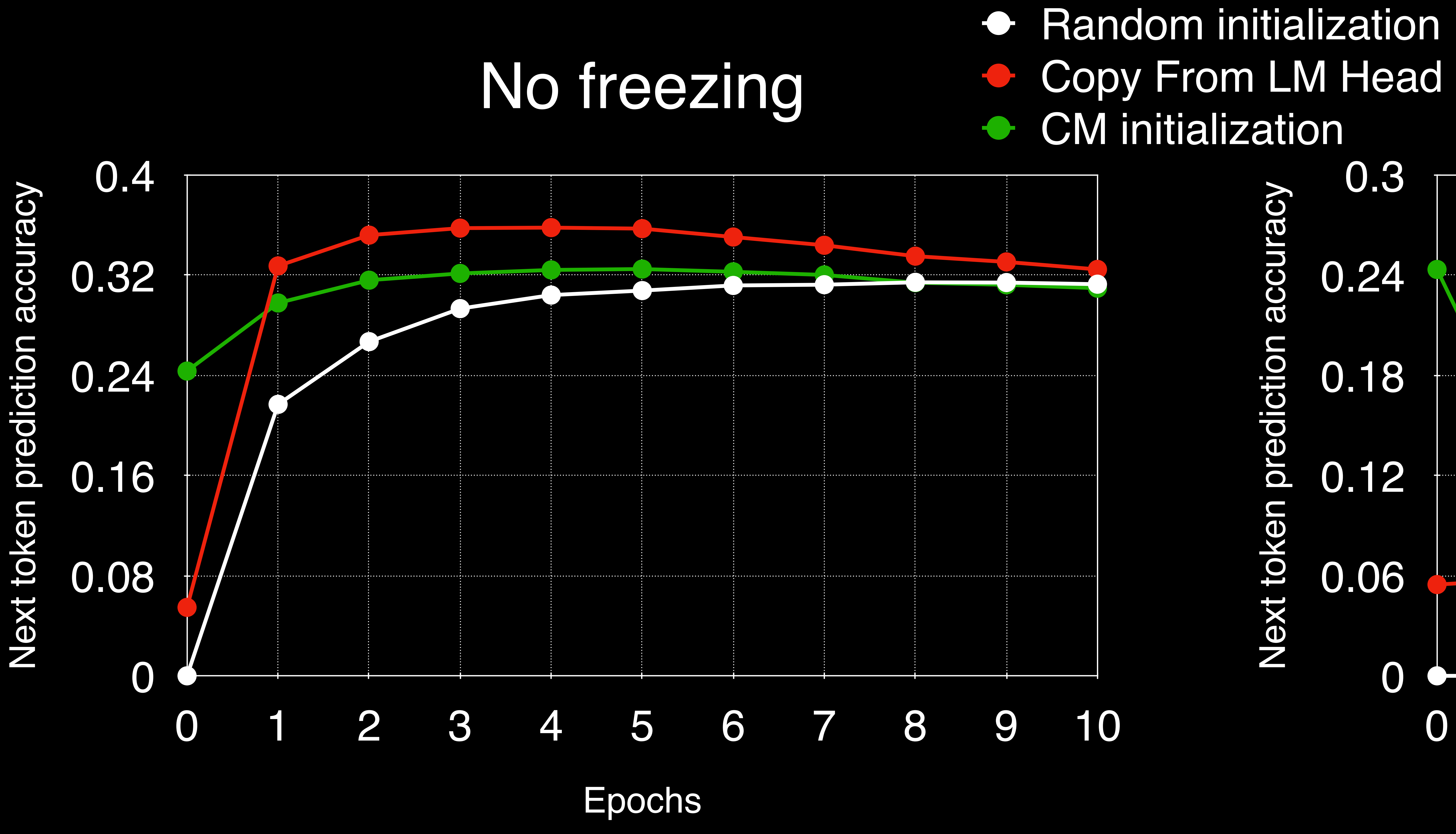


OPT-125m & wikitext-2-v1

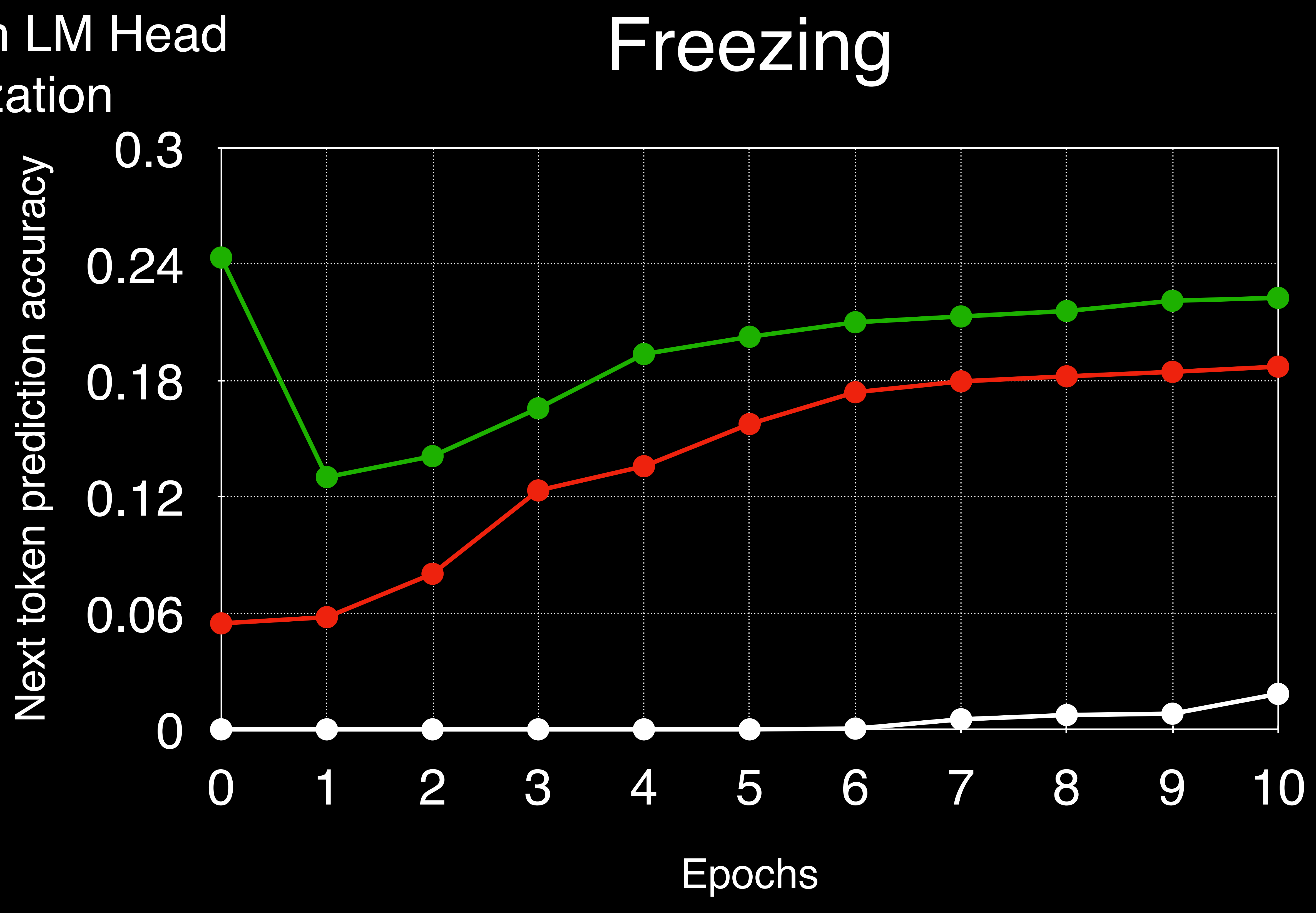


OPT-125m & wikitext-2-v1

Class-aware Initialization of Early Exits for Pre-training Large Language Models

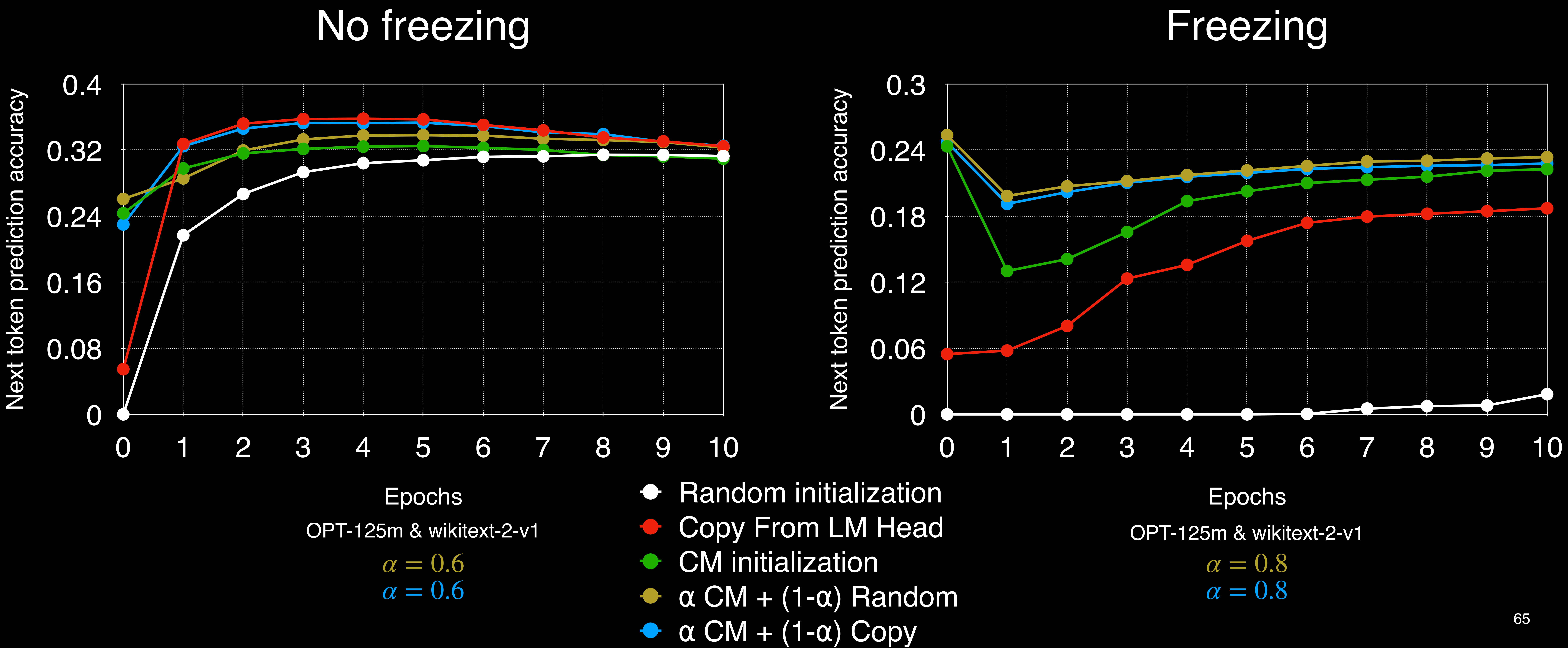


OPT-125m & wikitext-2-v1



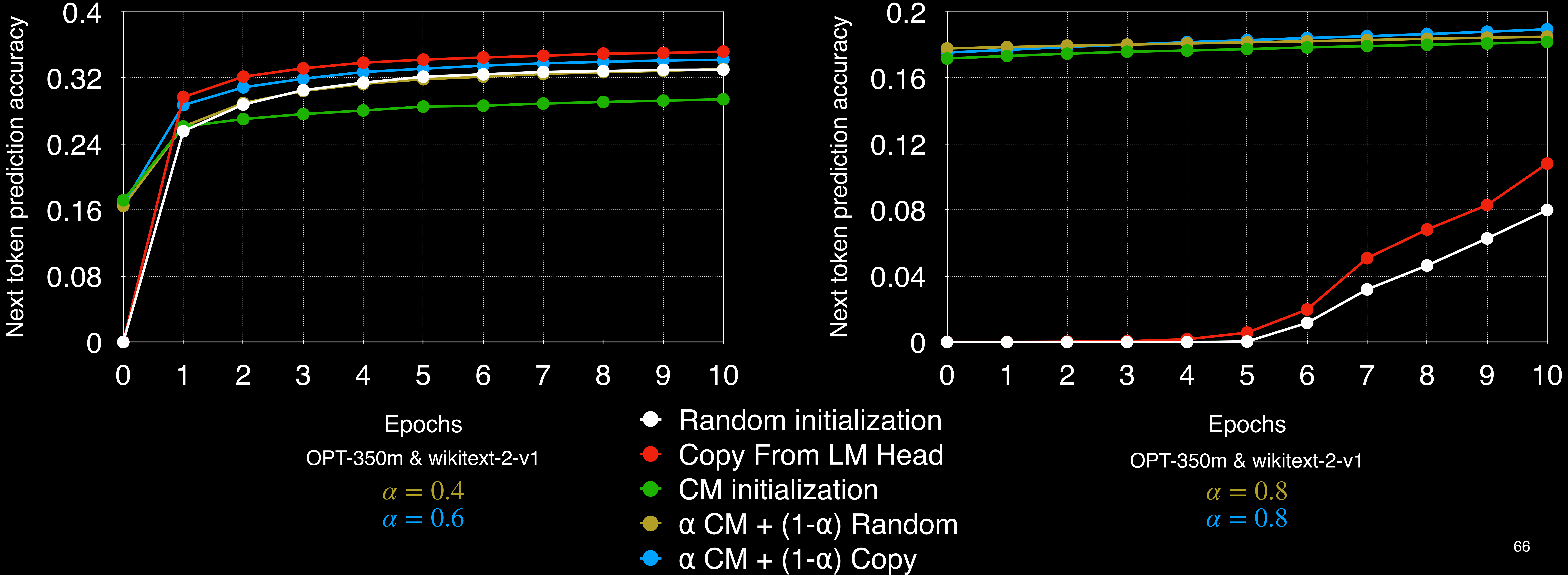
OPT-125m & wikitext-2-v1

Class-aware Initialization of Early Exits for Pre-training Large Language Models

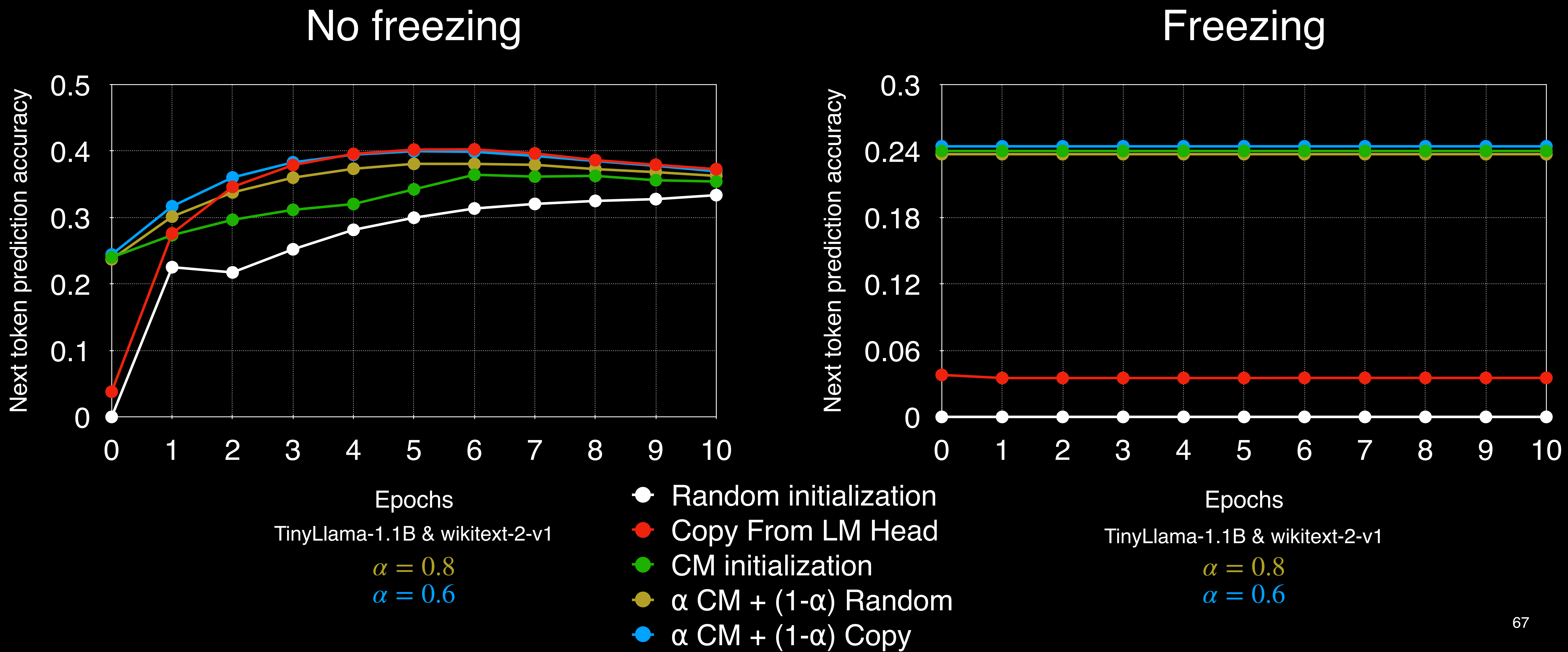


Class-aware Initialization of Early Exits for Pre-training Large Language Models

No freezing | Freezing



Class-aware Initialization of Early Exits for Pre-training Large Language Models

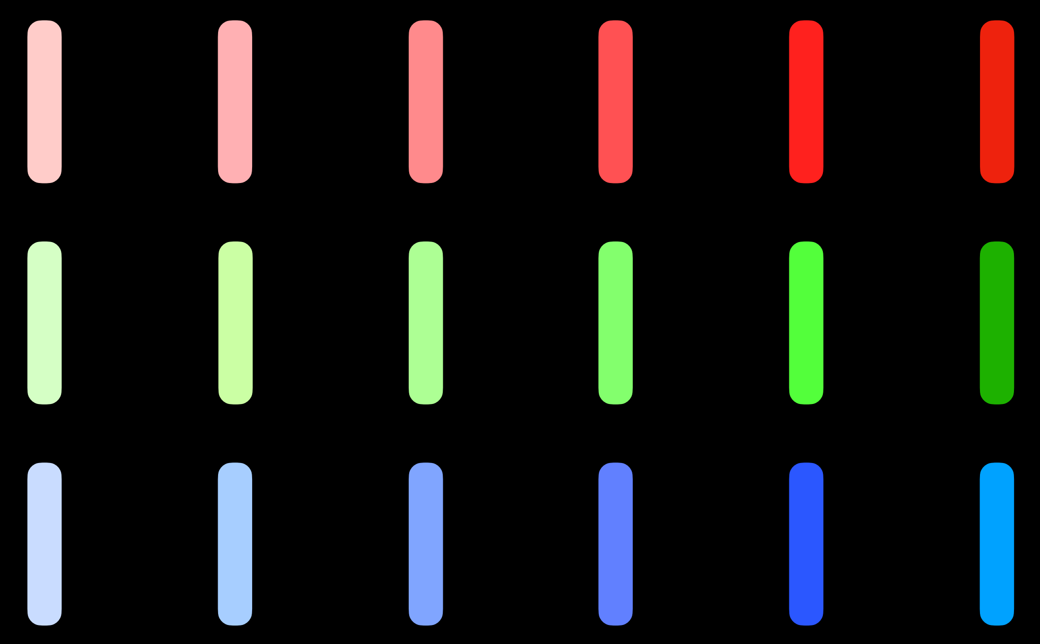


TinyLlama-1.1B & wikitext-2-v1

$\alpha = 0.8$

$\alpha = 0.6$

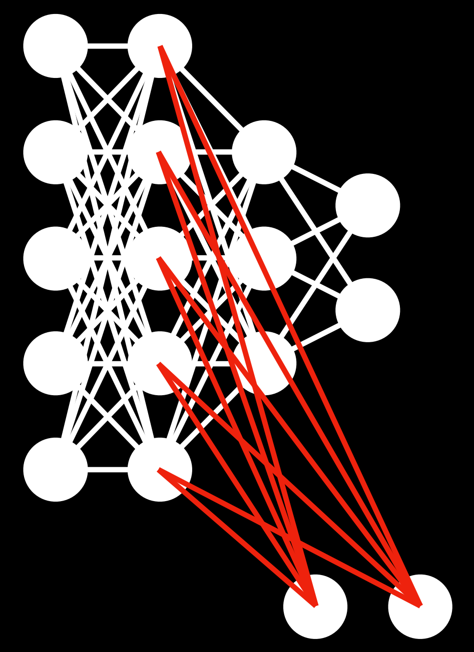
Future Work



- E^{2}CM in open world scenarios
- E^{2}CM efficiency via pooling, quantization

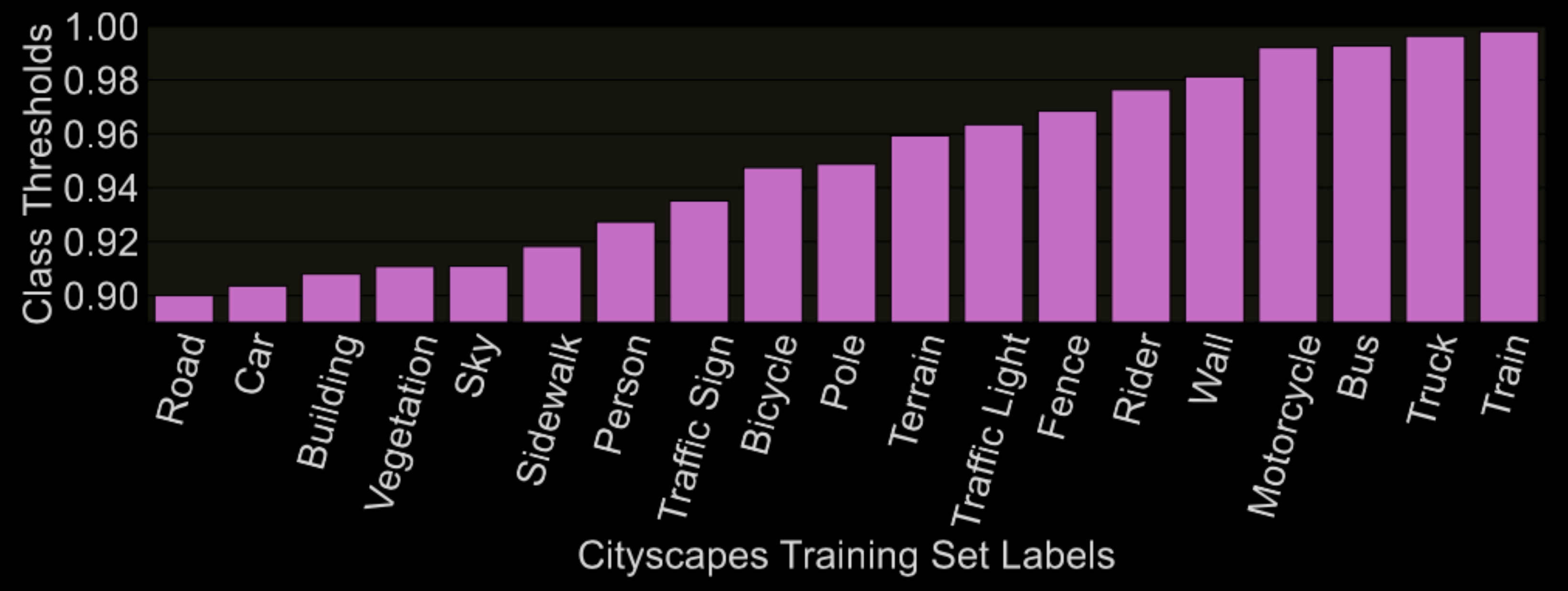
Future Work

- E^2CM in open world scenarios
- E^2CM efficiency via pooling, quantization
- Quantize + prune + distill early exit networks



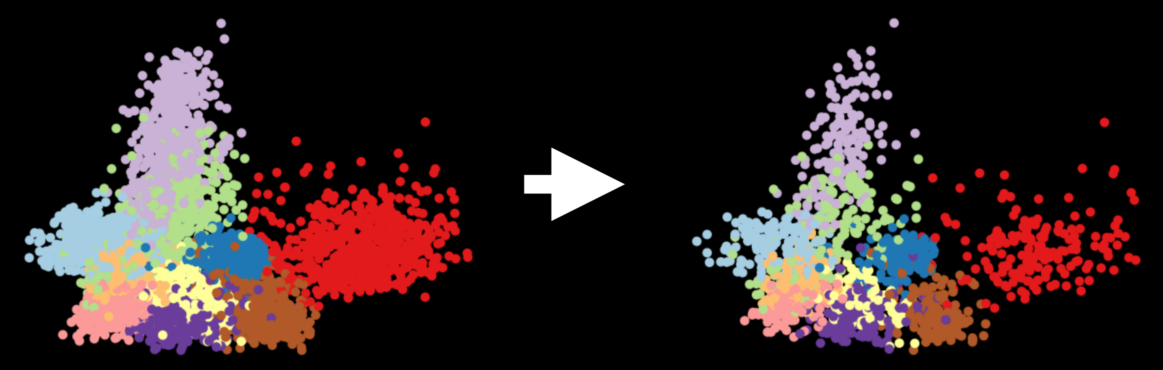
Future Work

- E^2CM in open world scenarios
- E^2CM efficiency via pooling, quantization
- Quantize + prune + distill early exit networks
- CBT for multimodal data



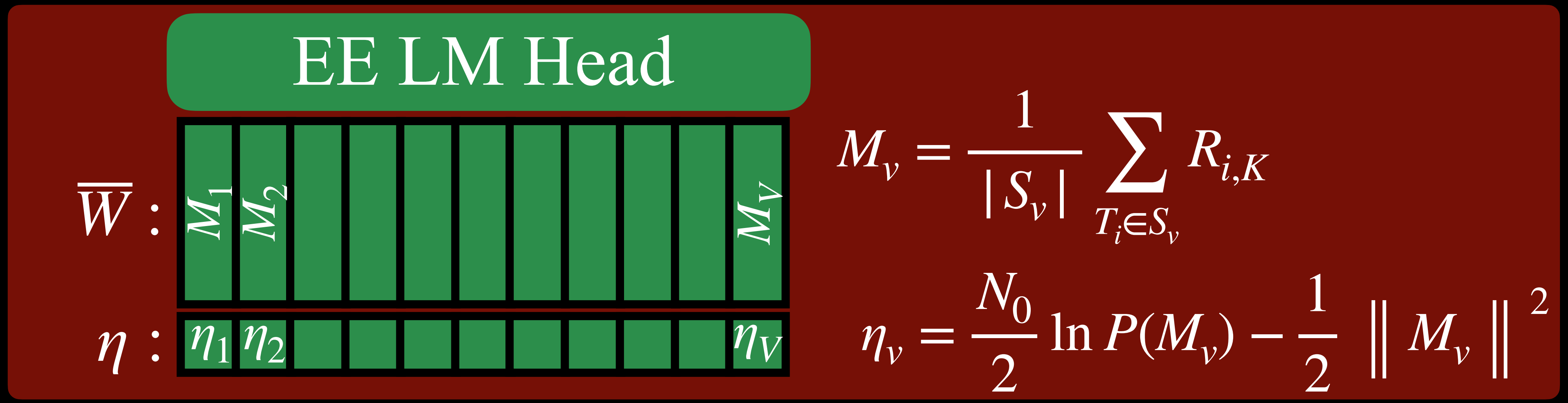
Future Work

- E^2CM in open world scenarios
- E^2CM efficiency via pooling, quantization
- Quantize + prune + distill early exit networks
- CBT for multimodal data
- EEPrune for unsupervised learning settings, e.g. clustering
- EEPrune for filtering LLM pre-training datasets

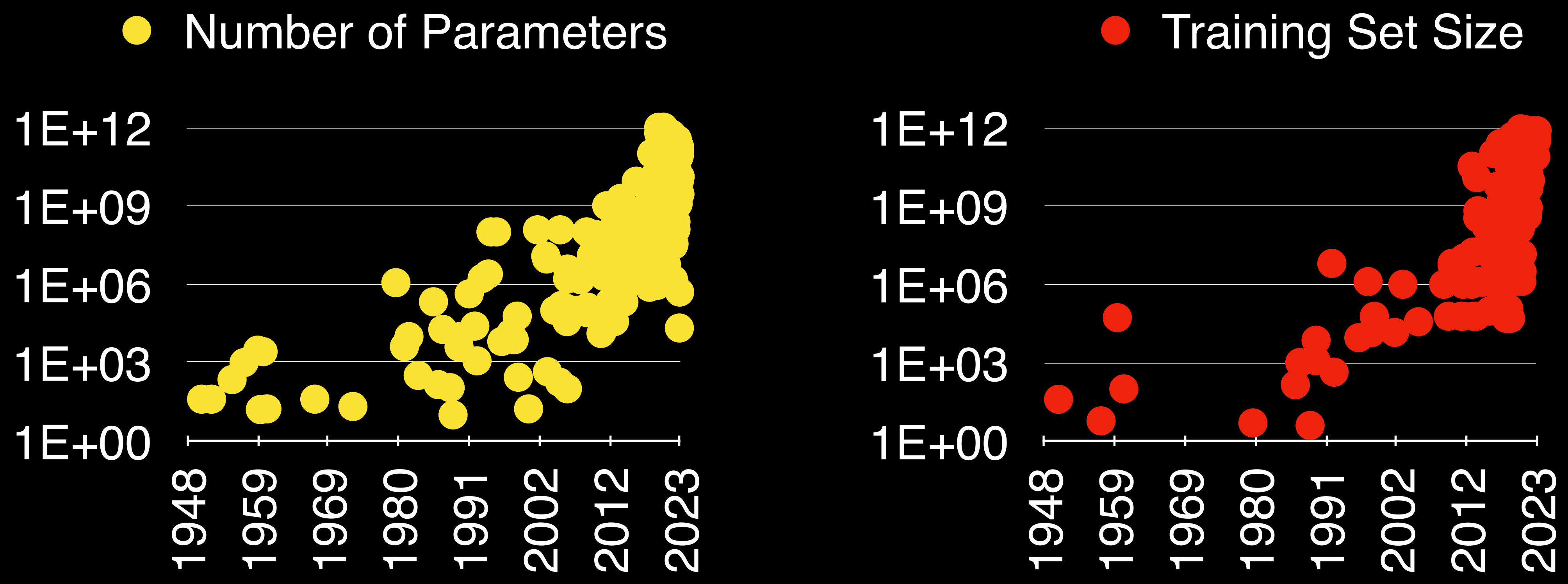


Future Work

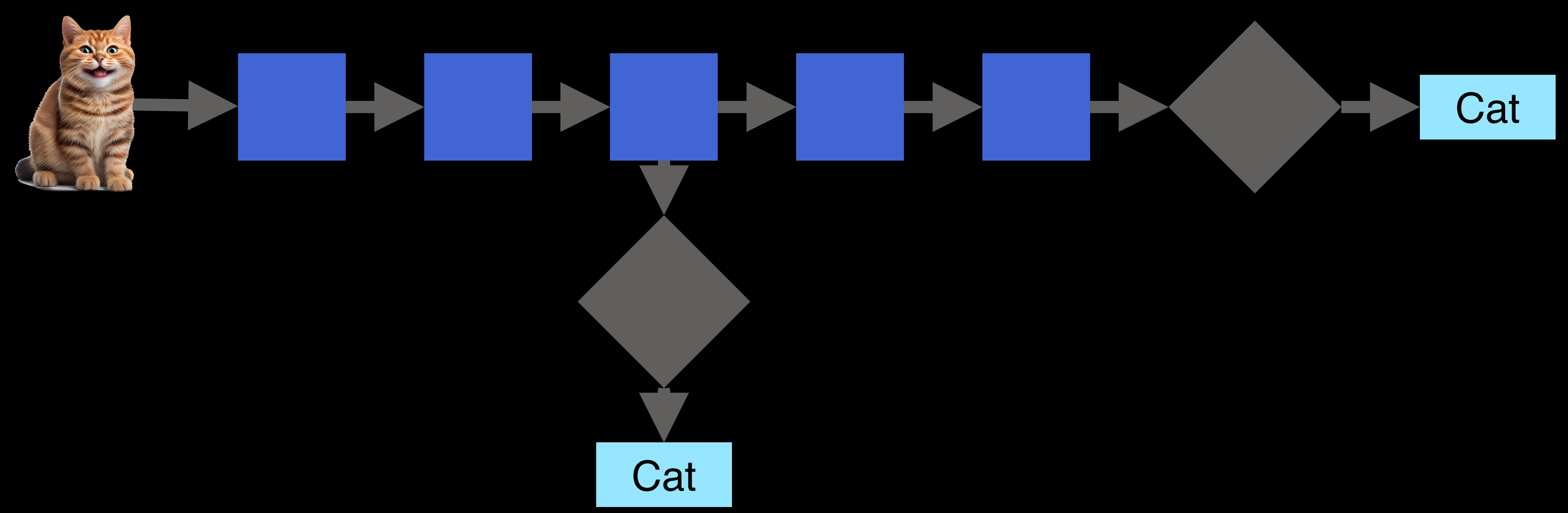
- E²CM in open world scenarios
- E²CM efficiency via pooling, quantization
- Quantize + prune + distill early exit networks
- CBT for multimodal data
- EEPrune for unsupervised learning settings, e.g. clustering
- EEPrune for filtering LLM pre-training datasets
- EE LLM initialization for SFT & RLHF steps



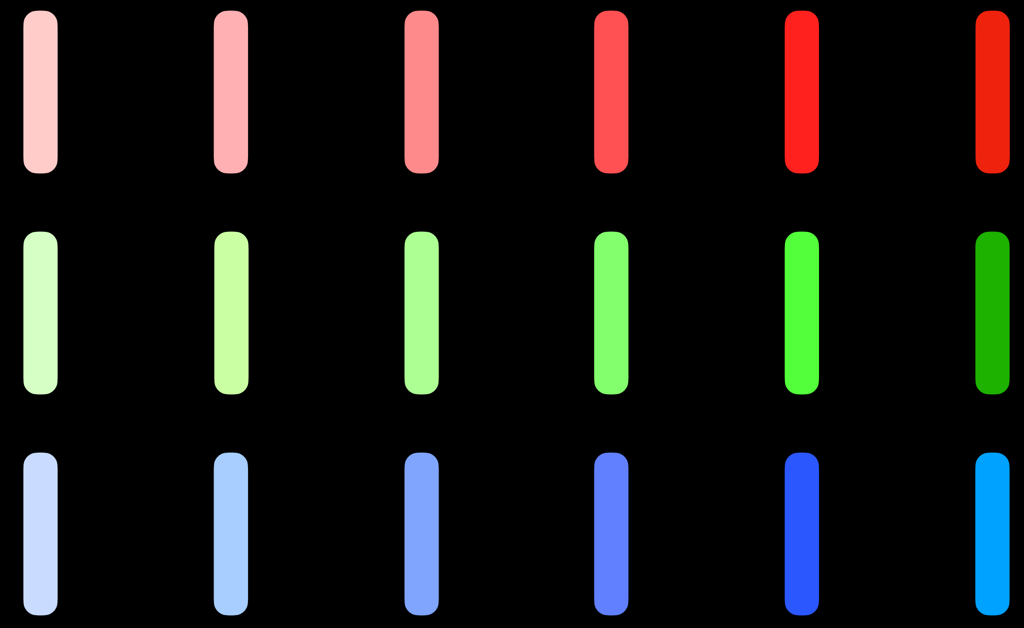
Conclusion

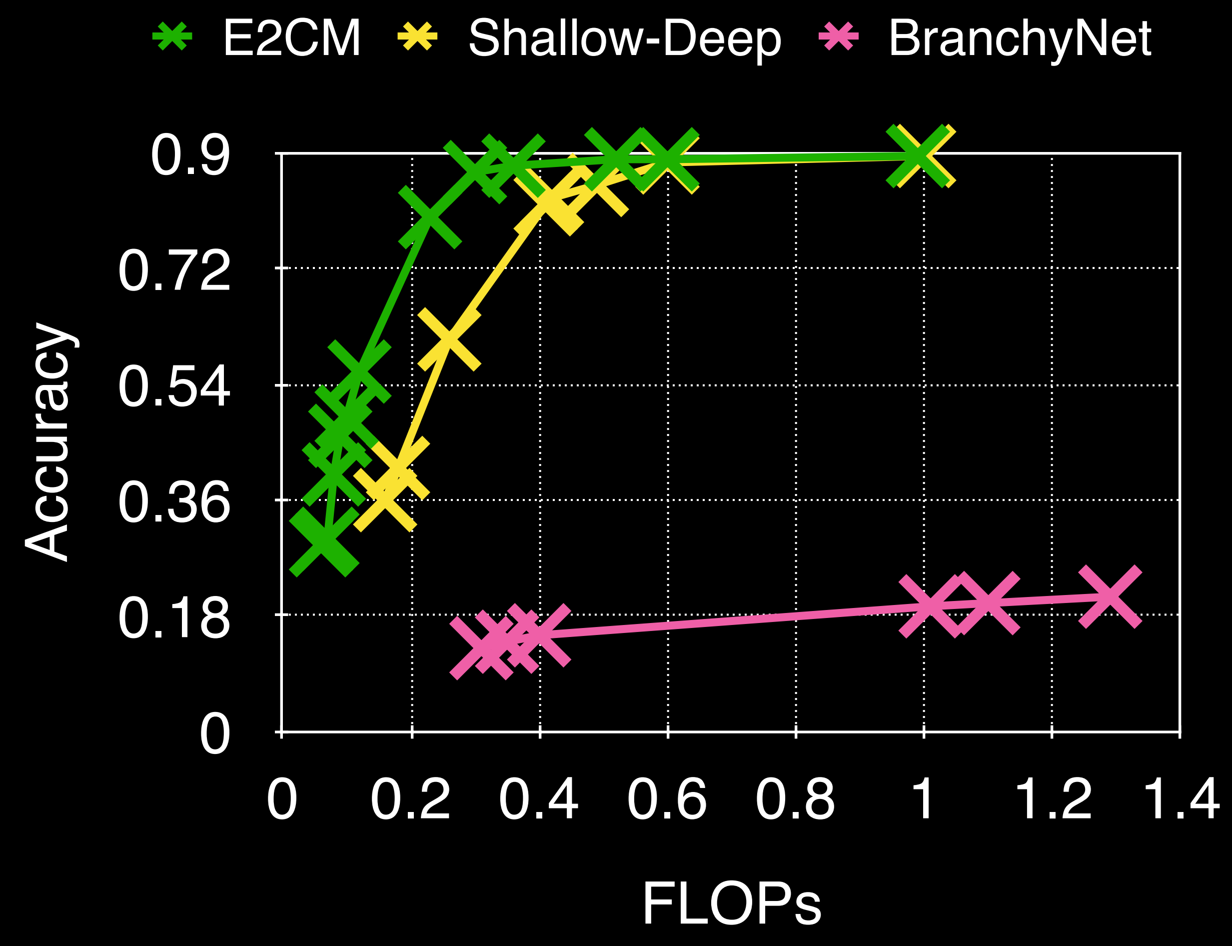


Number of Parameters
Training Set Size
1E+12
1E+09
1E+06
1E+03
1E+00
1948
1959
1969
1980
1991
2002
2012
2023

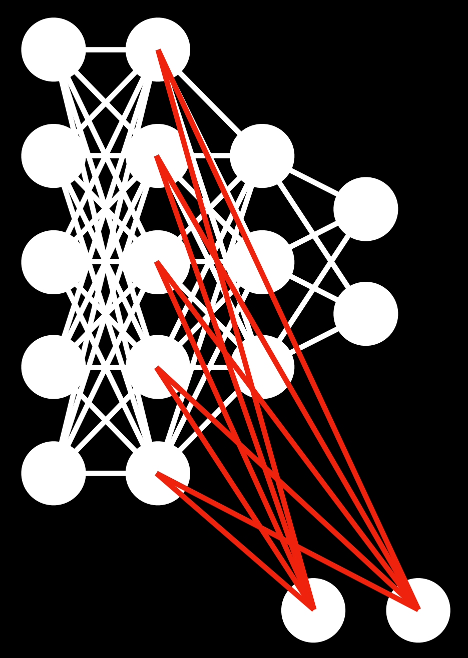


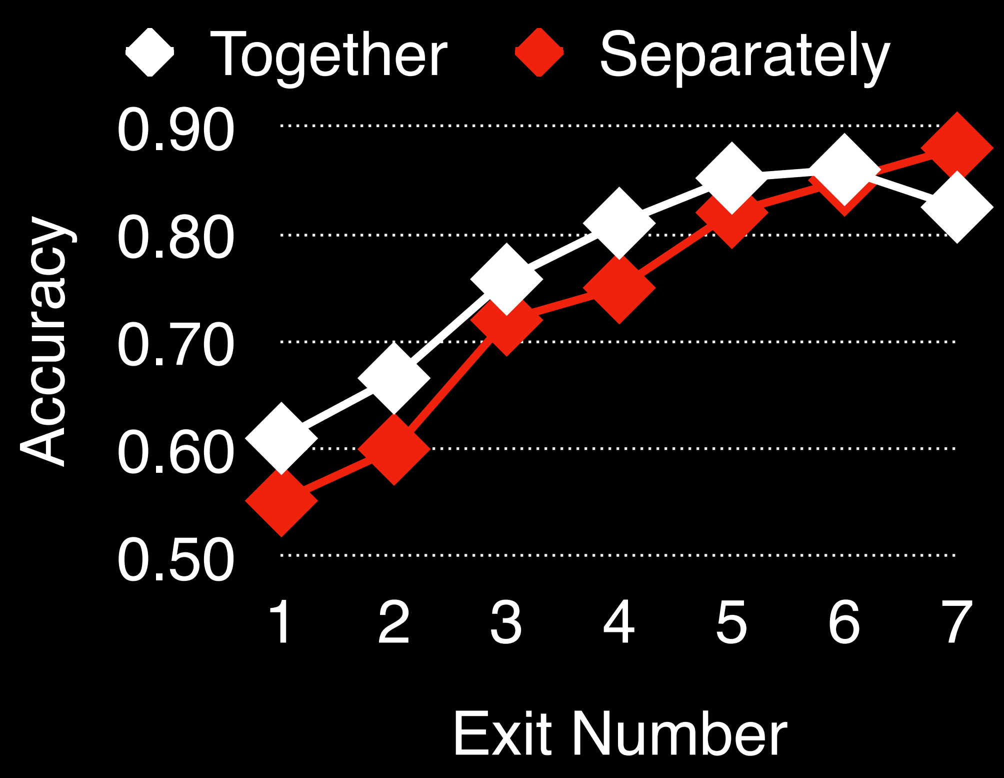
Cat
Cat



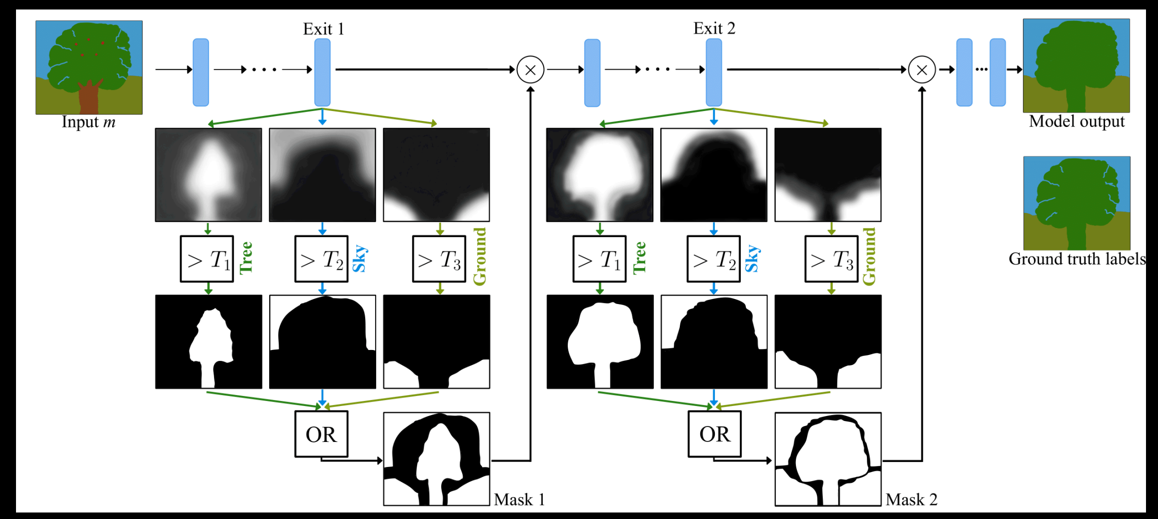


E2CM
Shallow-Deep
BranchyNet
Accuracy
0.9
0.72
0.54
0.36
0.18
0
0
0.2
0.4
0.6
0.8
1
1.2
1.4
FLOPs

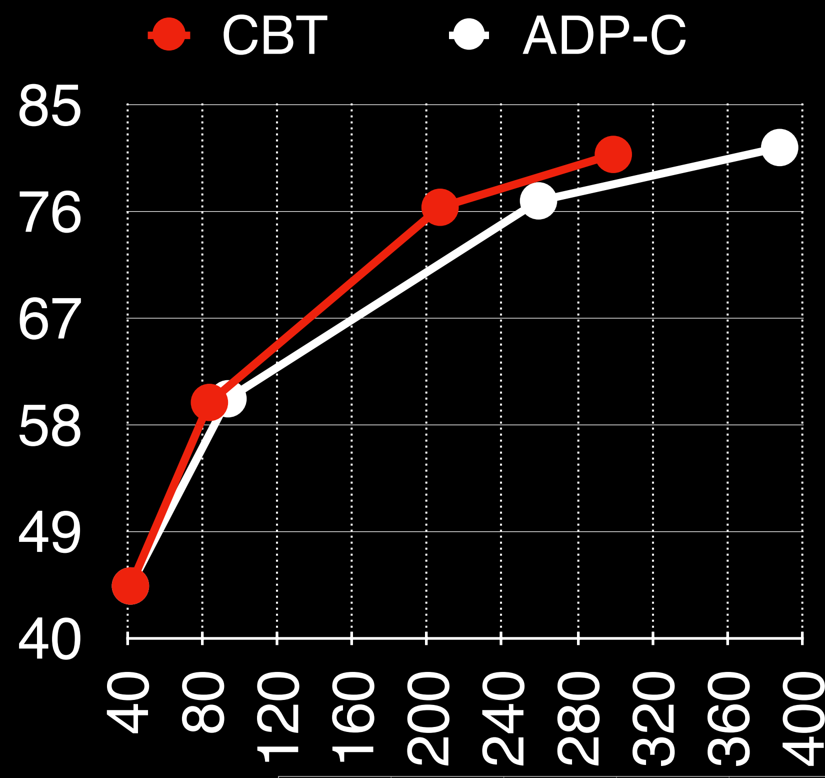




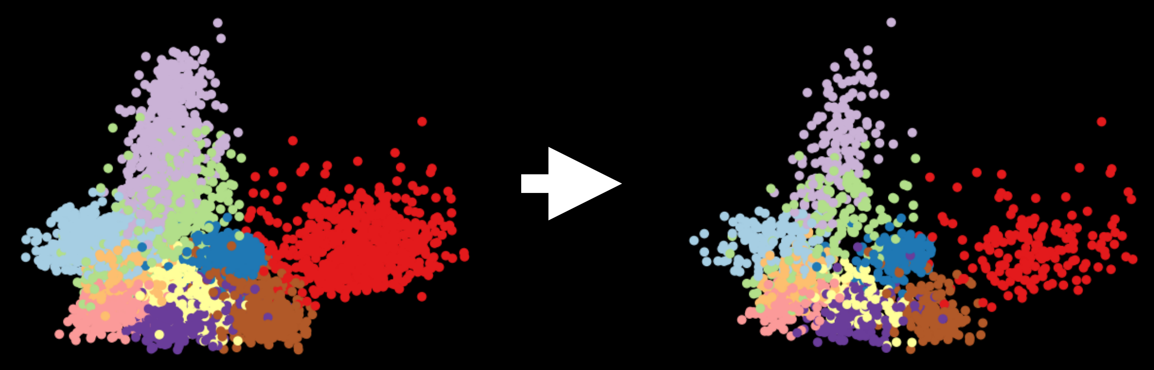
Together
Separately
Accuracy
0.90
0.80
0.70
0.60
0.50
1
2
3
4
5
6
7
Exit Number

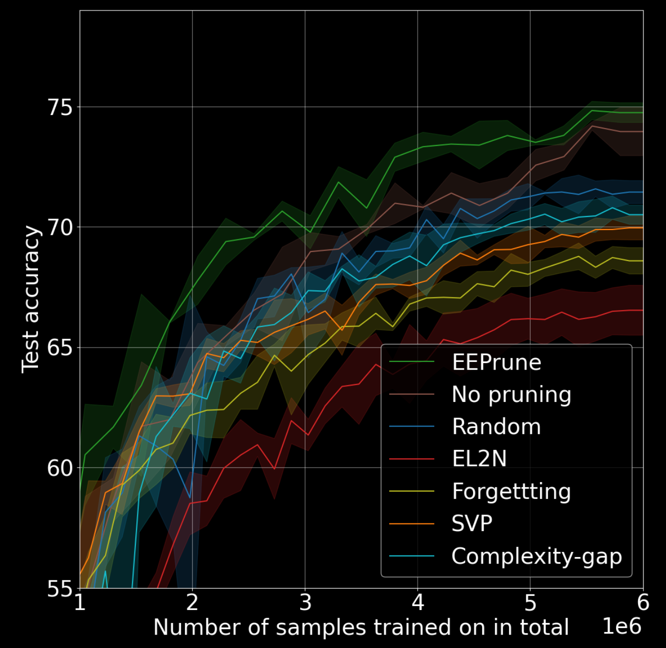


Exit 1
Exit 2
Input m
Model output
Ground truth labels
OR
Mask 1
Mask 2

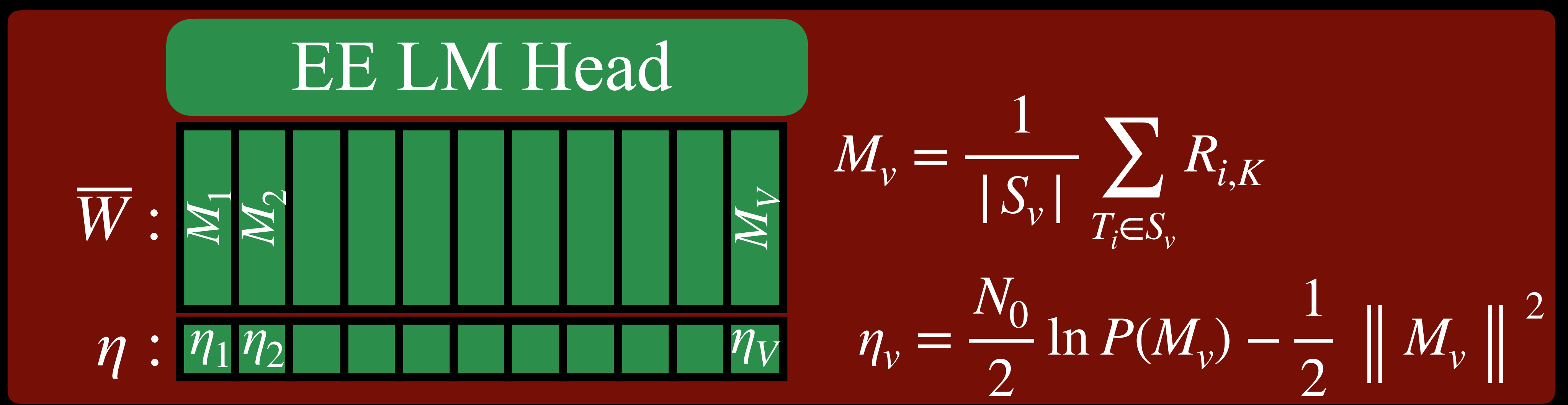


CBT
ADP-C
85
76
67
58
49
40
40
80
120
160
200
240
280
320
360
400





Test accuracy
75
70
65
60
55
EEPrune
No pruning
Random
EL2N
Forgettting
SVP
Complexity-gap
1
2
3
4
5
6
Number of samples trained on in total
1e6



EE LM Head
$\overline{W}$:
M_1 M_2 M_V
η :
η_1 η_2 η_V
$M_v = \frac{1}{|S_v|} \sum_{T_i \in S_v} R_{i,K}$
$\eta_v = \frac{N_0}{2} \ln P(M_v) - \frac{1}{2} \| M_v \|^2$