HUAWEI | ZURICH RESEARCH CENTER

Efficient Inference With Model Cascades

AAAI, Edge Intelligence Workshop, 2024-02-26

Lukas Cavigelli

Paper PDF

L. Lebovitz, L. Cavigelli, M. Magno, L. Müller, "Efficient Inference With Model Cascades", TMLR, 2023-09. https://t.ly/NtOVA.

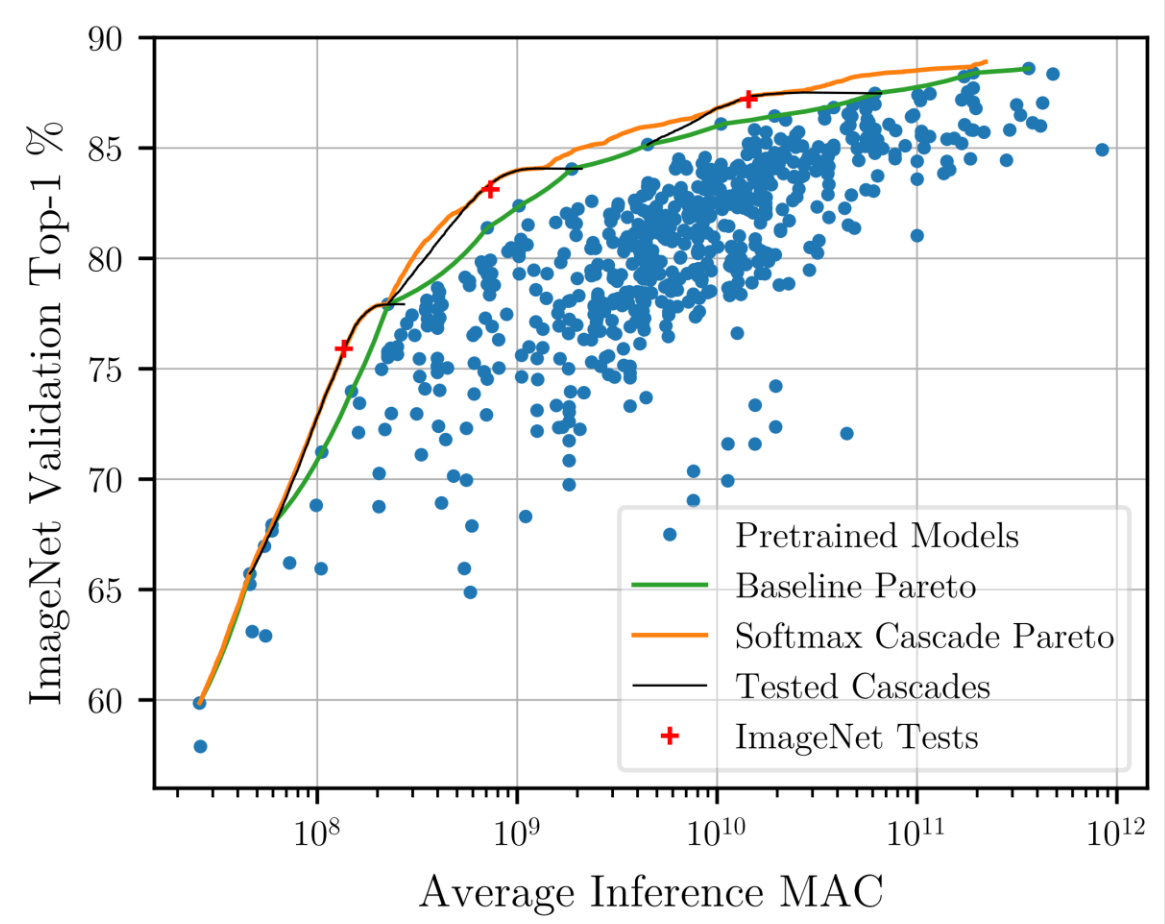
Outline

- Motivation
- Efficient models, Accurate models
- Datasets: Not all queries/classes are equal
- Related work: Early exit models
- Early Exit Cascades
  - Concept
  - Decision criteria & ensembling
  - Multi-model cascades
  - Distribution shifts
  - Real execution time
- Conclusion

Motivation

- Deployment of CV tasks is a **quality-cost trade-off**
  - Specialized chips/devices
  - Quantization, pruning, …
  - Most effective: different models

    → 500 models of the TIMM database



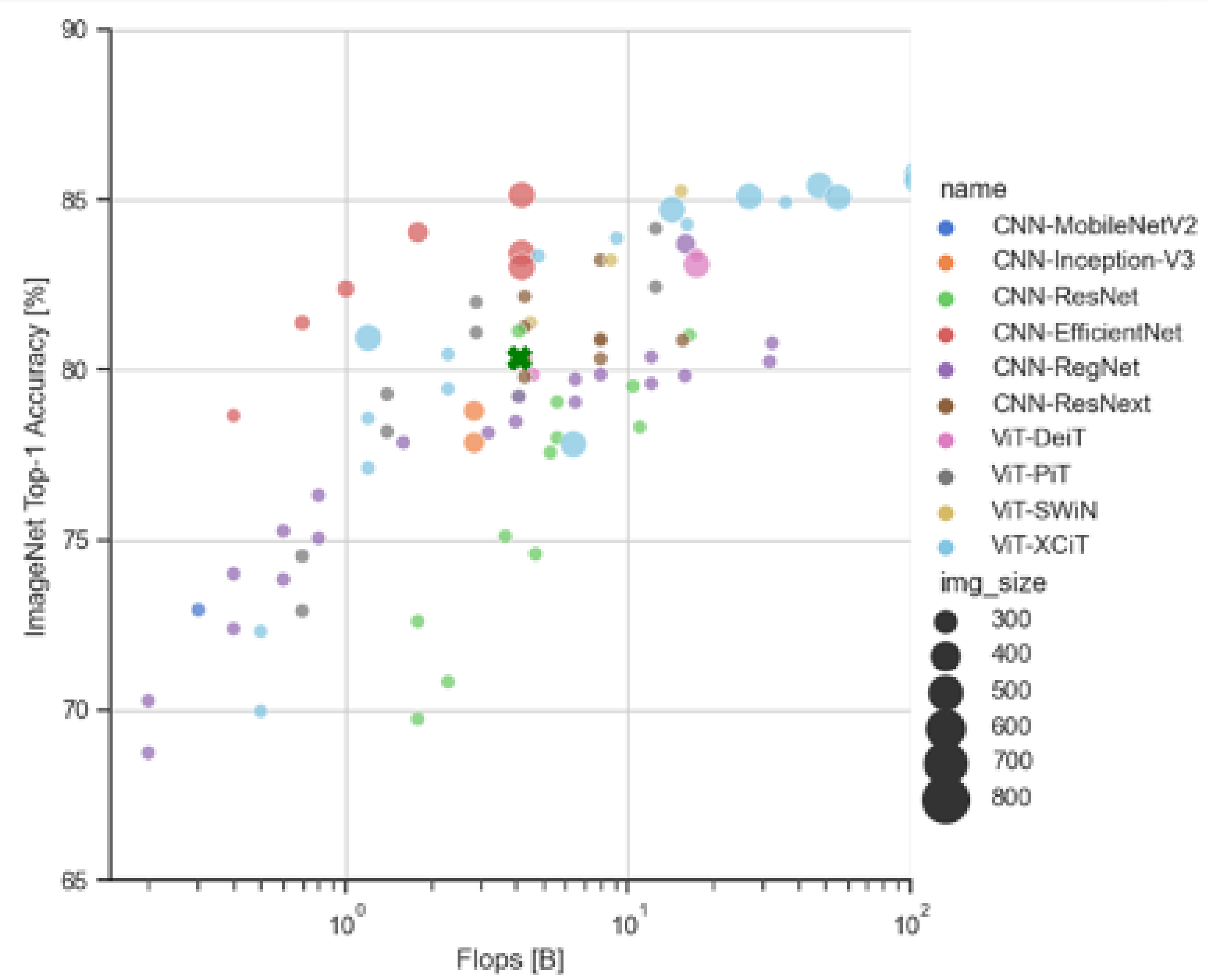
- We will show how to
  - gain **~3x speed-up**
  - at **equal accuracy**,
  - compatible with all other optimizations!

Efficient Models, Accurate Models

- How to measure efficiency?
  - Simplified metric for compute: MatMul & Conv
    → count number of multiply-accum. operations

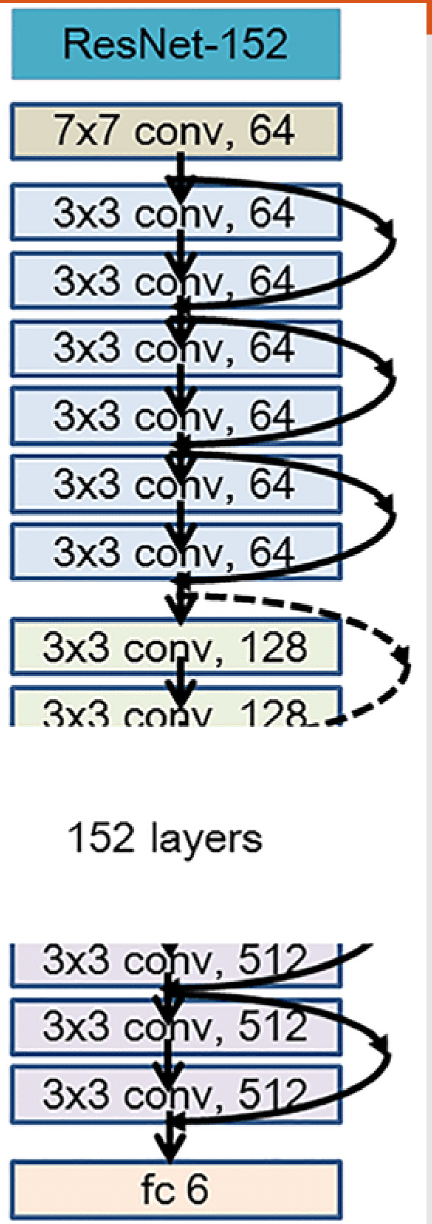
- Different types of DNNs
  - basic: ResNet
  - compute-optimized: MobileNet/EfficientNet, …
  - top-accuracy: various ViT

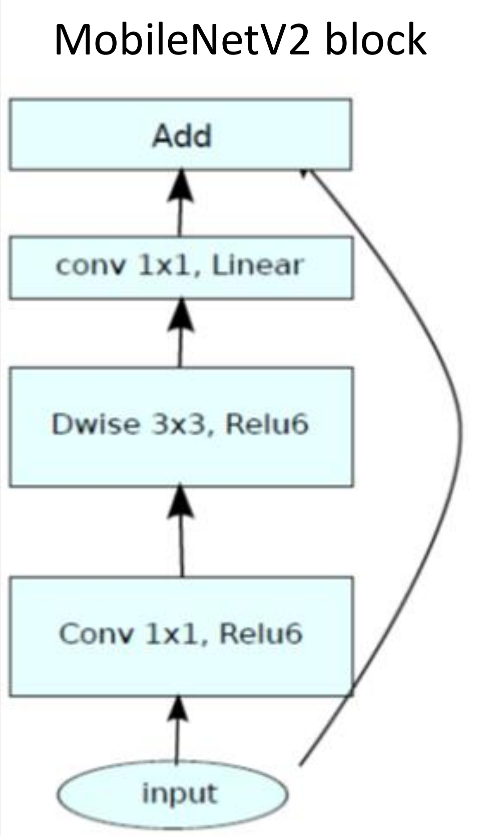
- How are they built?
  - normal convolution layers / residual layers
  - depth-wise separable convolutions
  - transformer blocks

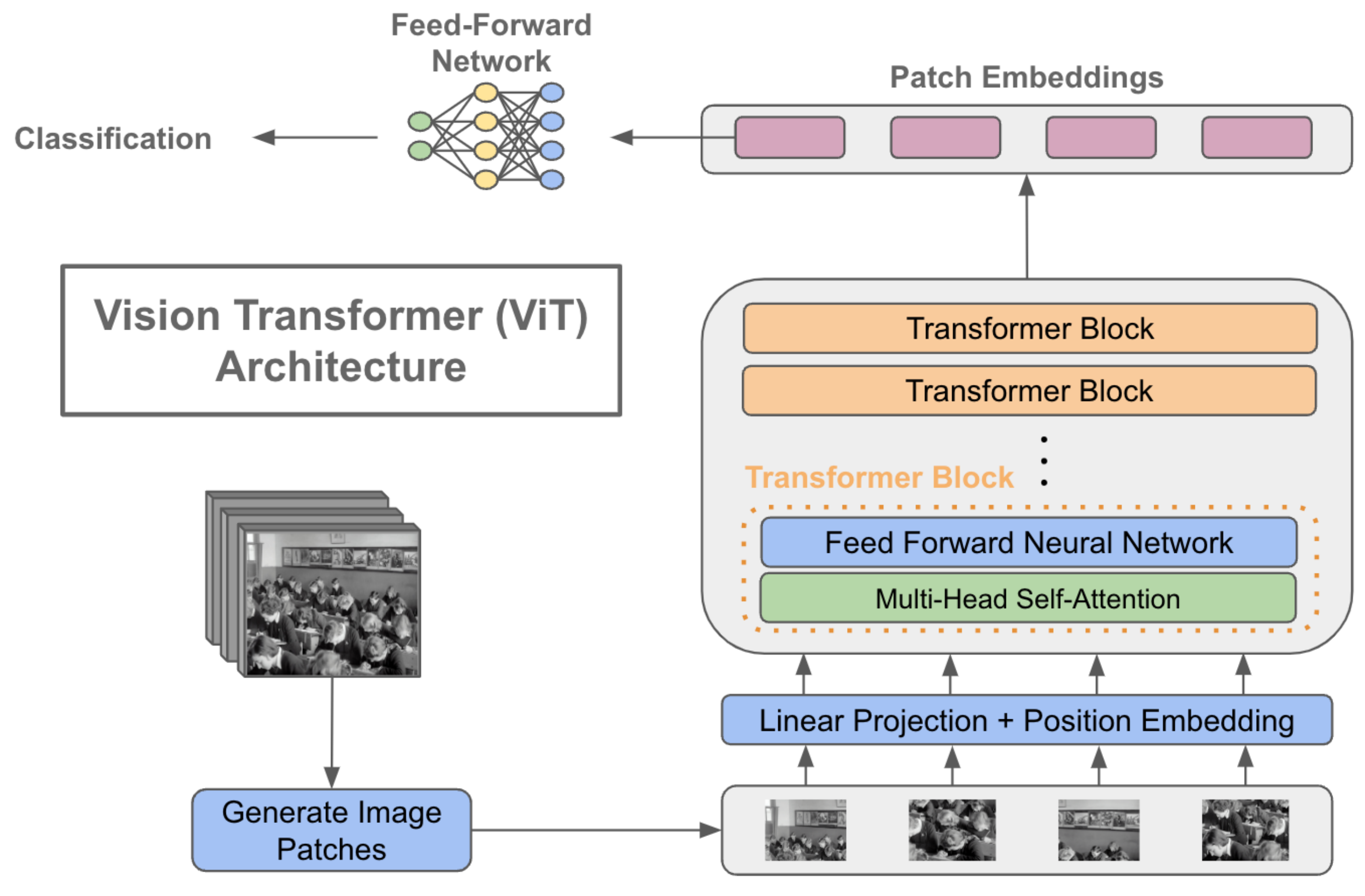


HUAWEI

Efficient Models, Accurate Models

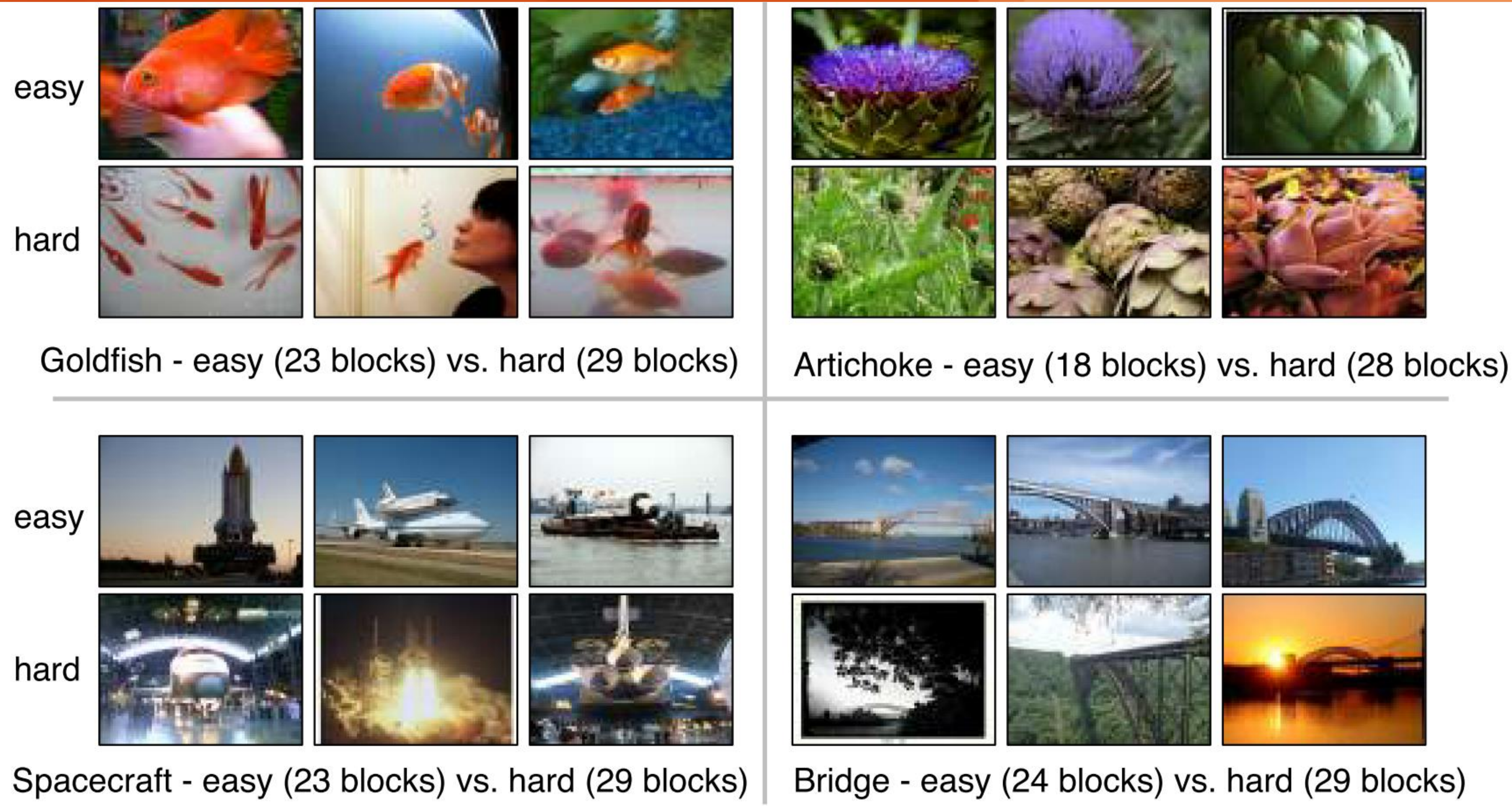






Dataset Intricacies

- Easy and hard examples
- Fine-grained distinction is harder
- 1000 classes



Easy & Hard Dataset Items

- What are these?
  - Basketball & dog. Easy!
  - Assuming simple classes…
- How about now?
  - Basketball. Still easy!
  - But what dog breed?!?!
    That is… hard!
    (it’s a Norfolk terrier)
  - ImageNet: 1k classes, 118 dogs
- Some inputs are just harder
  - more detailed classification
  - harder to identify

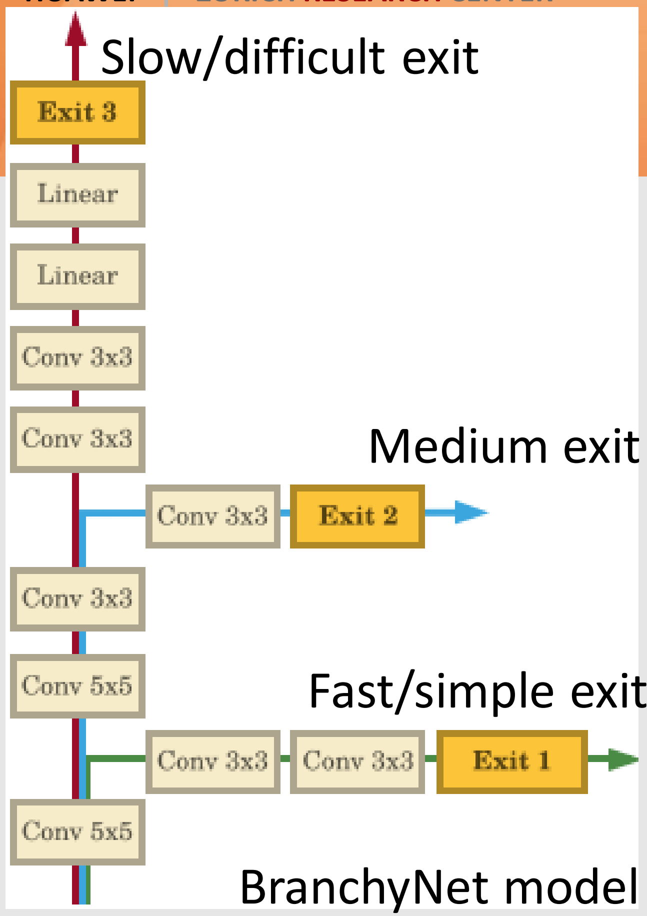
Do we need to spend the same effort for all (easy & hard) queries?

21: 'kite',
22: 'bald eagle,
23: 'vulture’,
…
173: 'Ibizan hound',
174: 'Norwegian elkhound’,
…
181: 'Bedlington terrier',
182: 'Border terrier',
183: 'Kerry blue terrier',
184: 'Irish terrier',
185: 'Norfolk terrier’,
…
275: 'African hunting dog',
…
415: 'bakery',
416: 'balance beam',
417: 'balloon',
418: 'ballpoint pen', …
430: 'basketball’,
…
999: toilet paper'

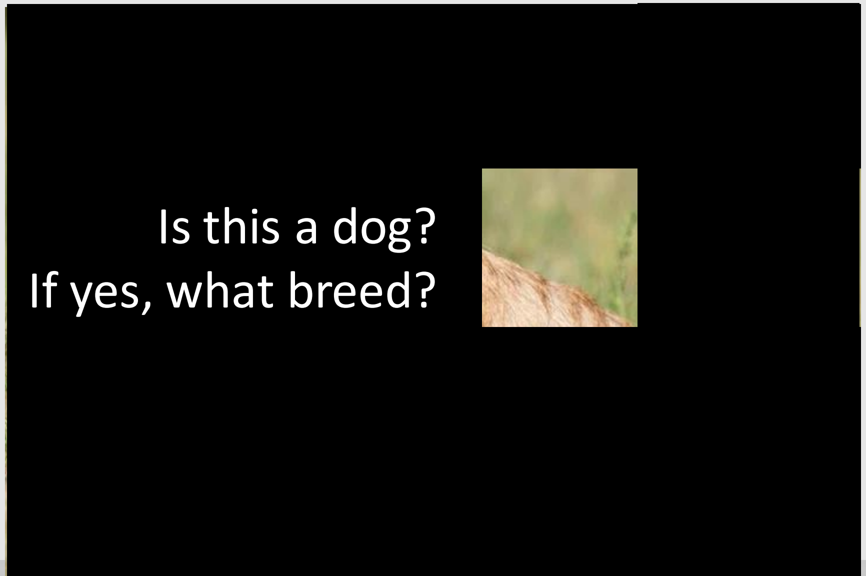
Dealing with Varying Complexity: Early Exit Models

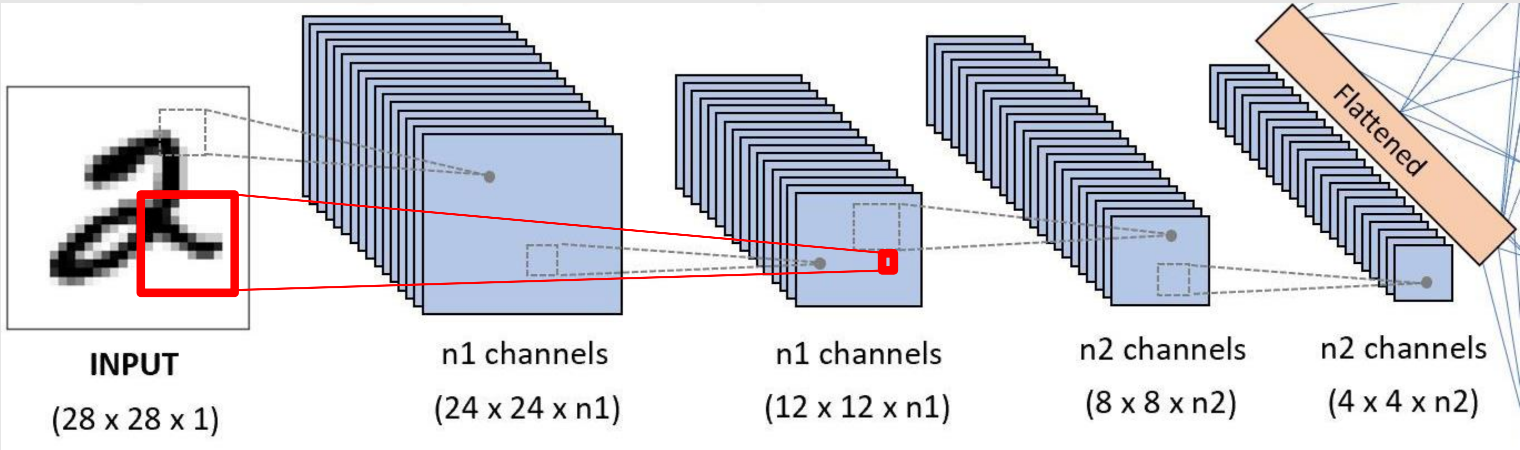
- Simple idea: let the DNN stop computation (exit) early when sufficiently confident
  - Many papers on this… e.g., BranchyNet
- Fundamental problems:
  - we need to train a specialized model, find a good structure
  - field-of-view/receptive field:
    after 2 layers, the output cannot “see” the entire input

→ fundamental limitation!



BranchyNet model





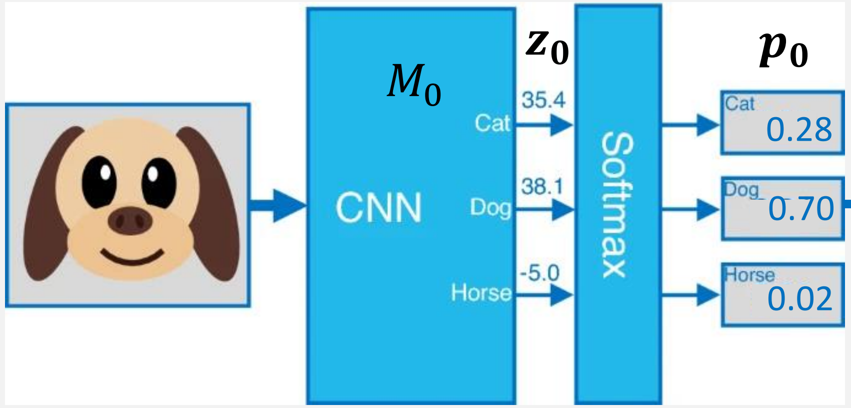
DNN Cascades: Concept

- Back to the high level, simple intuition:

Why ask the master (expensive) when you can ask the apprentice (cheaper)?

  - The student can tell you if they are confident enough in their answer, refer you to the expert if needed
  - Just use existing pre-trained models – fully optimized, no “guess work”

Small model (cost $c_0 = 1$ GMAC/frame)



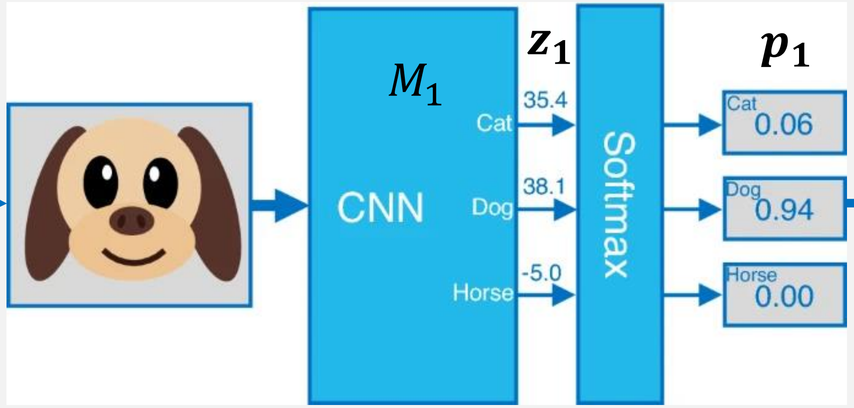
$\max_i(\boldsymbol{p}_{0,i}) \geq \tau_0$?

yes

$y = \arg\max_i(\boldsymbol{p}_{0,i})$

no

Large model (cost $c_1 = 10$ GMAC/frame)



$y = \arg\max_i(\boldsymbol{p}_{1,i})$

- What do we save? If τ_0 such that $\beta_0 = \Pr\left[\max_i(\boldsymbol{p}_{0,i}) \geq \tau_0\right] = 75\%$ of cases can exit early:

$c_{avg} = c_0\beta_0 + (c_0 + c_1)(1 - \beta_0) = 3.5$ GFLOP/frame vs. $c_1 = 10$ GFLOP/frame

HUAWEI

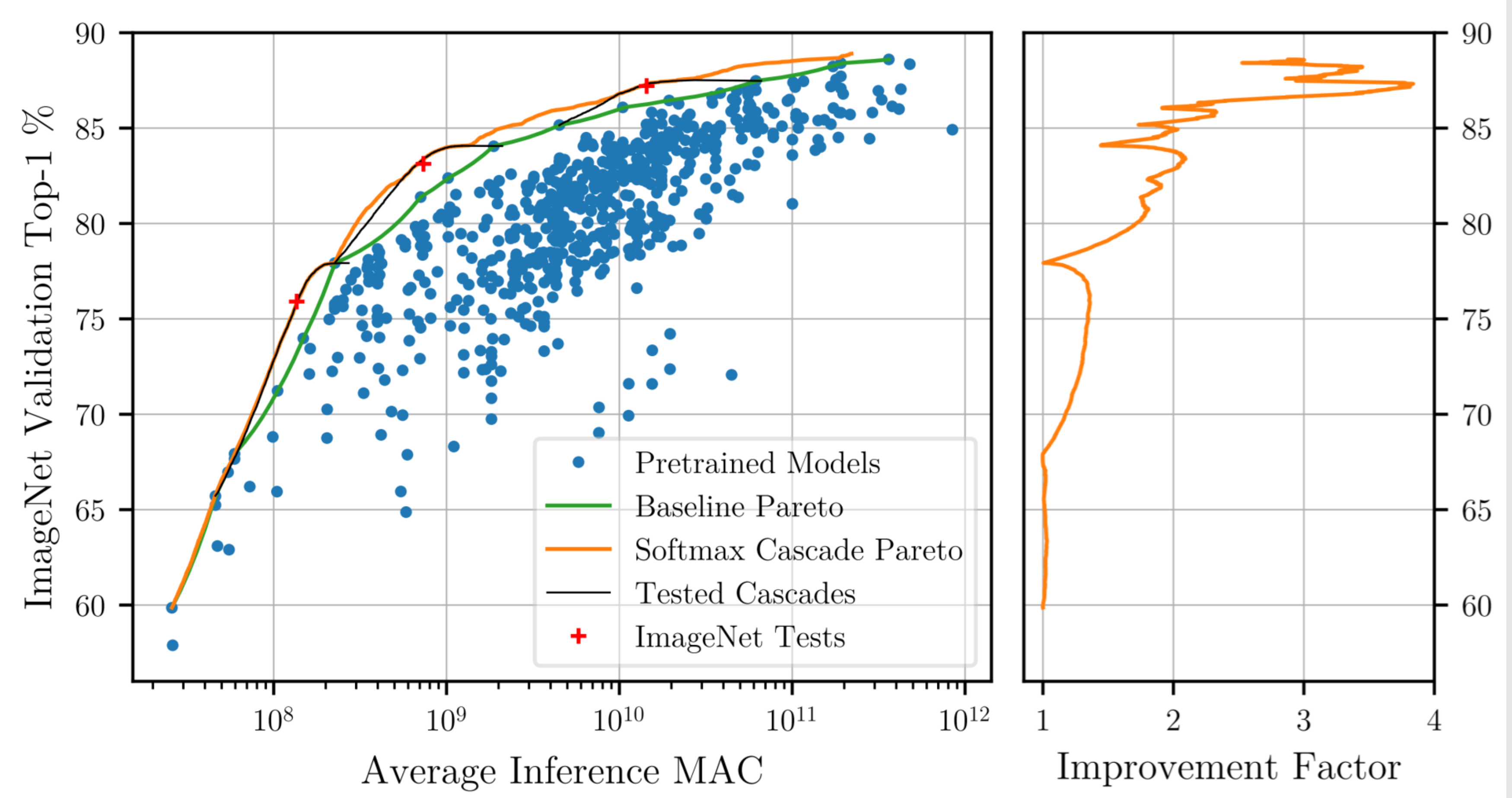
DNN Cascade: Result of Basic Method

→ Massive (2-3.8×!!) speed-ups for >80% accuracy

→ spans entire Pareto-front (always use this!)

→ no re-training or manual engineering

- base models: TIMM database
  - 500+ pre-trained(!) models
- baseline pareto
  - interpolate between 2 models by randomly switching between them
- individual cascades
  - Cascade trade-off (sweep τ_0) of 2 models
- cascade pareto front
  - The best trade-off among all pairs of models
- experimental setup
  - sweep validation set to find best model combination & evaluate

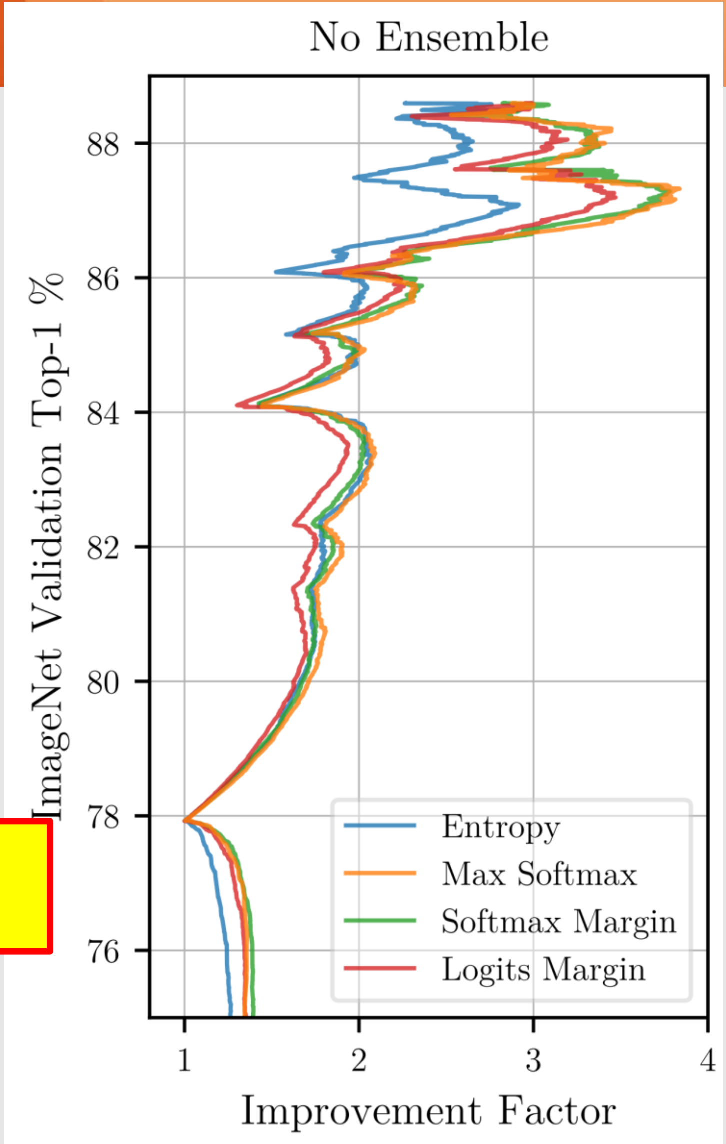


DNN Cascade: Decision Criteria & Ensembling

- **Decision Criteria**
  - Max Softmax (so far): $\max_i(\boldsymbol{p}_{0,i}) \geq \tau_0$
  - Shannon entropy (information/uncertainty): $-\sum_i \boldsymbol{p}_{0,i} \log(\boldsymbol{p}_{0,i}) \geq \tau_0$
  - Softmax margin (margin to 2nd best guess): $\max_i(\boldsymbol{p}_{0,i}) - \max_{j \neq i}(\boldsymbol{p}_{0,\boldsymbol{j}}) \geq \tau_0$
  - Logits margin (Softmax margin w/o norm.): $\max_i(\boldsymbol{z}_{0,i}) - \max_{j \neq i}(\boldsymbol{z}_{0,\boldsymbol{j}}) \geq \tau_0$
- **Ensembling**
  - Multiple DNNs (experts) can form better consensus
    - applies when *no early exit*
    - majority-voting or averaging
  - Weight of experts has to consider their skill level:

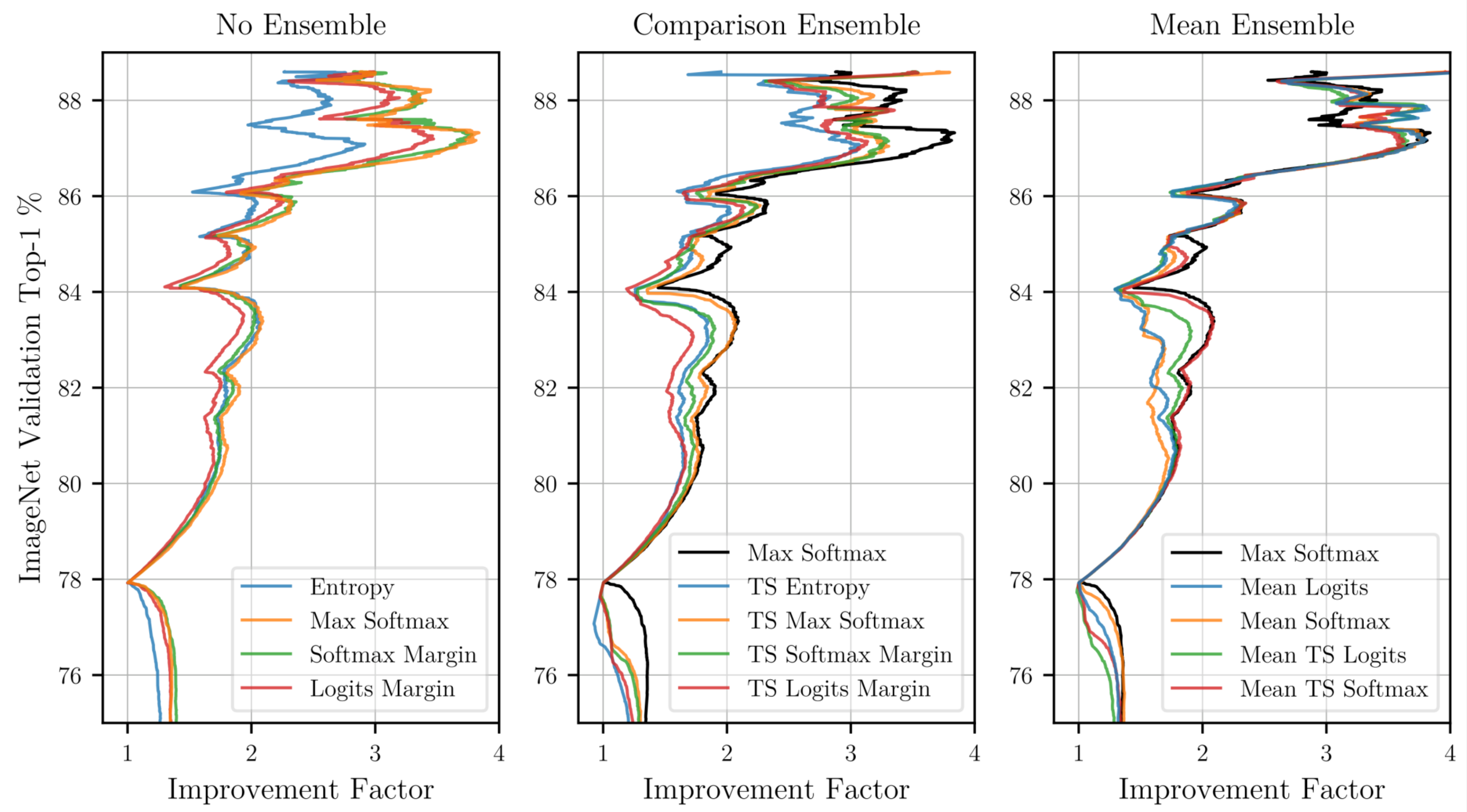
→ use max softmax

→ temperature scaling to calibrate confidence $\boldsymbol{p}_{0,i} = \frac{e^{z_i/T}}{\sum_j e^{z_j/T}}$



HUAWEI

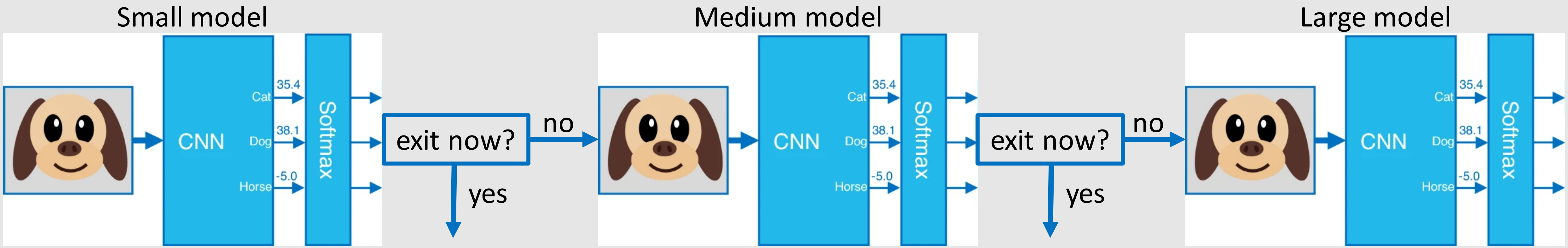
DNN Cascade: Results with Ensembling



- ensembling provides a gain at the top
- ensembling only worthwhile if out of alternatives (need similar-sized models)

DNN Cascade: Multi-Model Cascades

- So far: one small and one large model
- We can also use 3 models:



- Or more generally:

Algorithm 1 Early exit model cascade with maximum softmax confidence metric and no ensembling

Require: input tensor $\mathbf{X}$, models $\{M_1, ..., M_n\}$ ordered by increasing cost, thresholds $\{t_1, ..., t_{n-1}\}$, $n \geq 2$

for $i = 1, ..., n$ **do**

  $\boldsymbol{z}_i = M_i(\mathbf{X})$

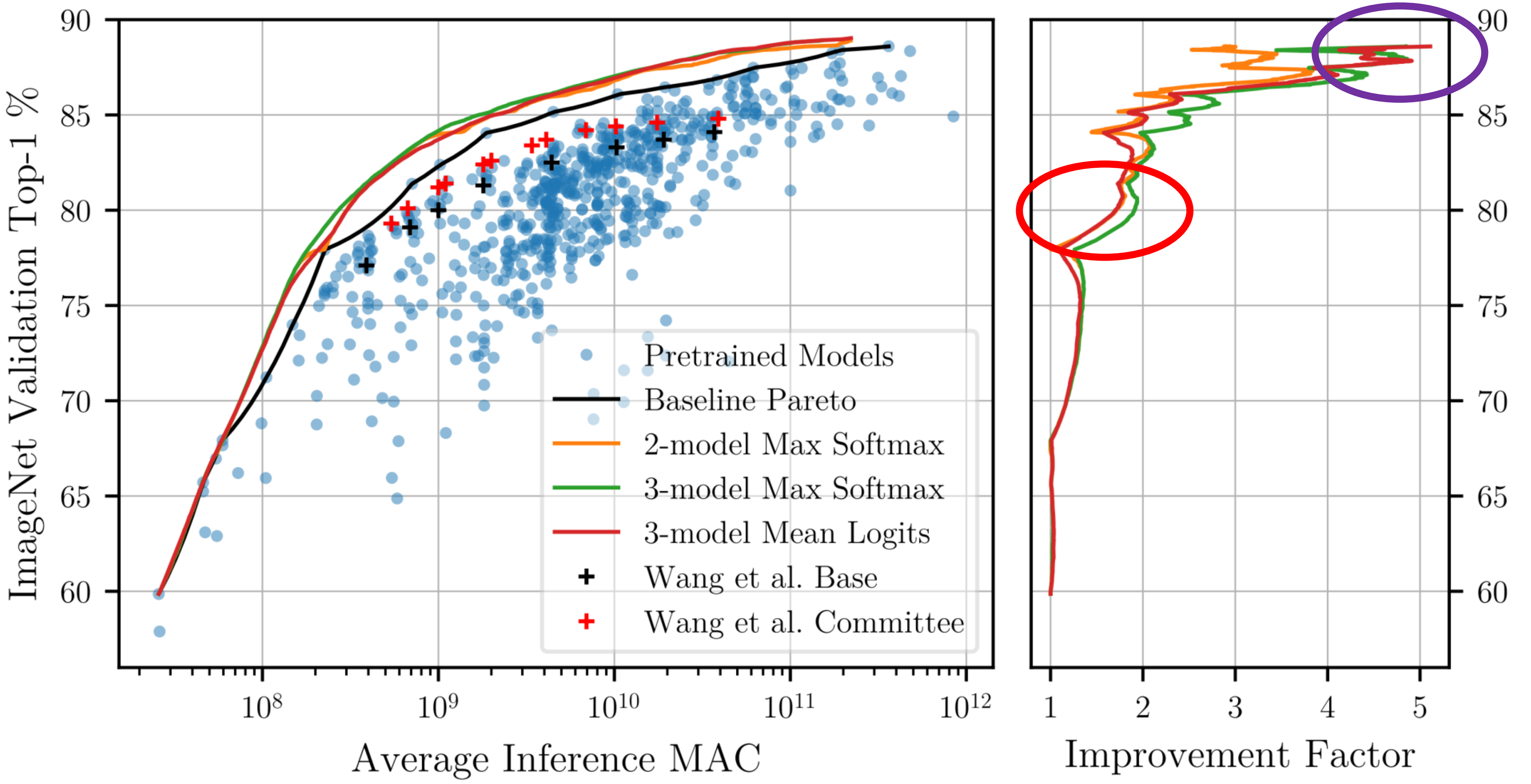
  $\boldsymbol{p}_i = \mathrm{softmax}(\boldsymbol{z}_i)$

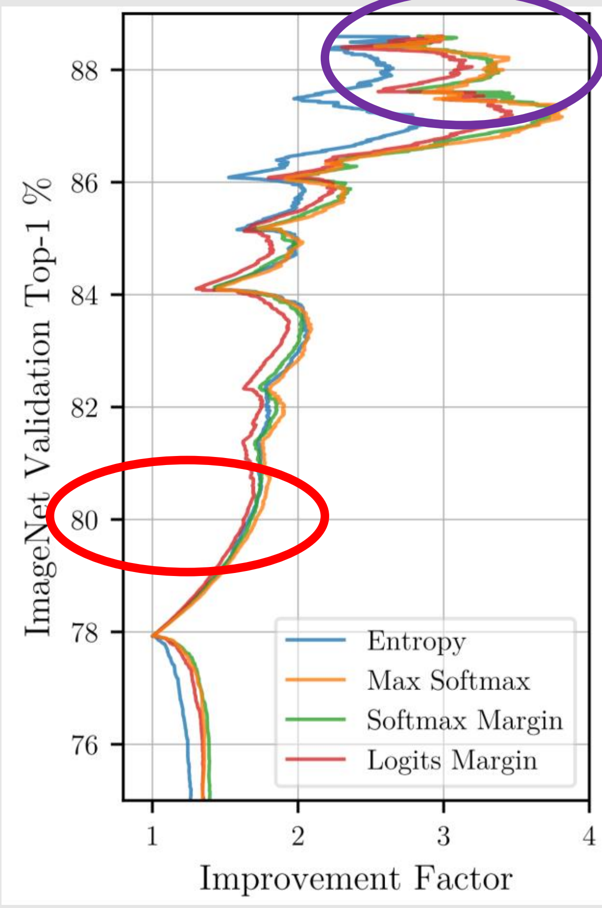
  if $i == n$ **or** $\max(\boldsymbol{p}_i) \geq t_i$ **then**

    return $\arg\max(\boldsymbol{p}_i)$ ▷ cascade returns predicted class

DNN Cascade: Results with 3 Models

previously:
2 model cascades



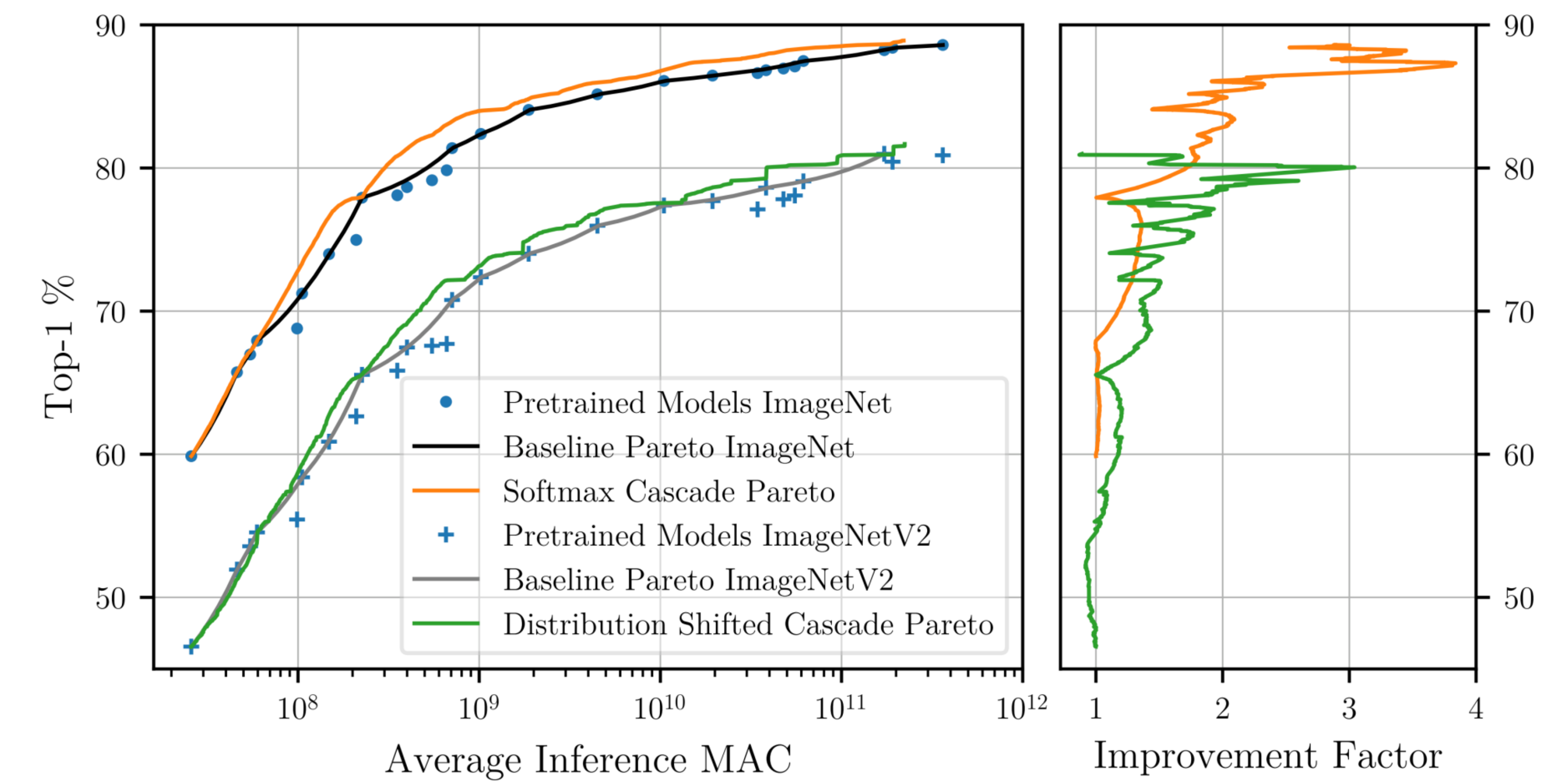


- Clearly above state-of-the-art
- Another clear performance leap: 1.7x to 1.95x at 80%
- Even more effective at the top: from ~3x to ~4.5x speed-up

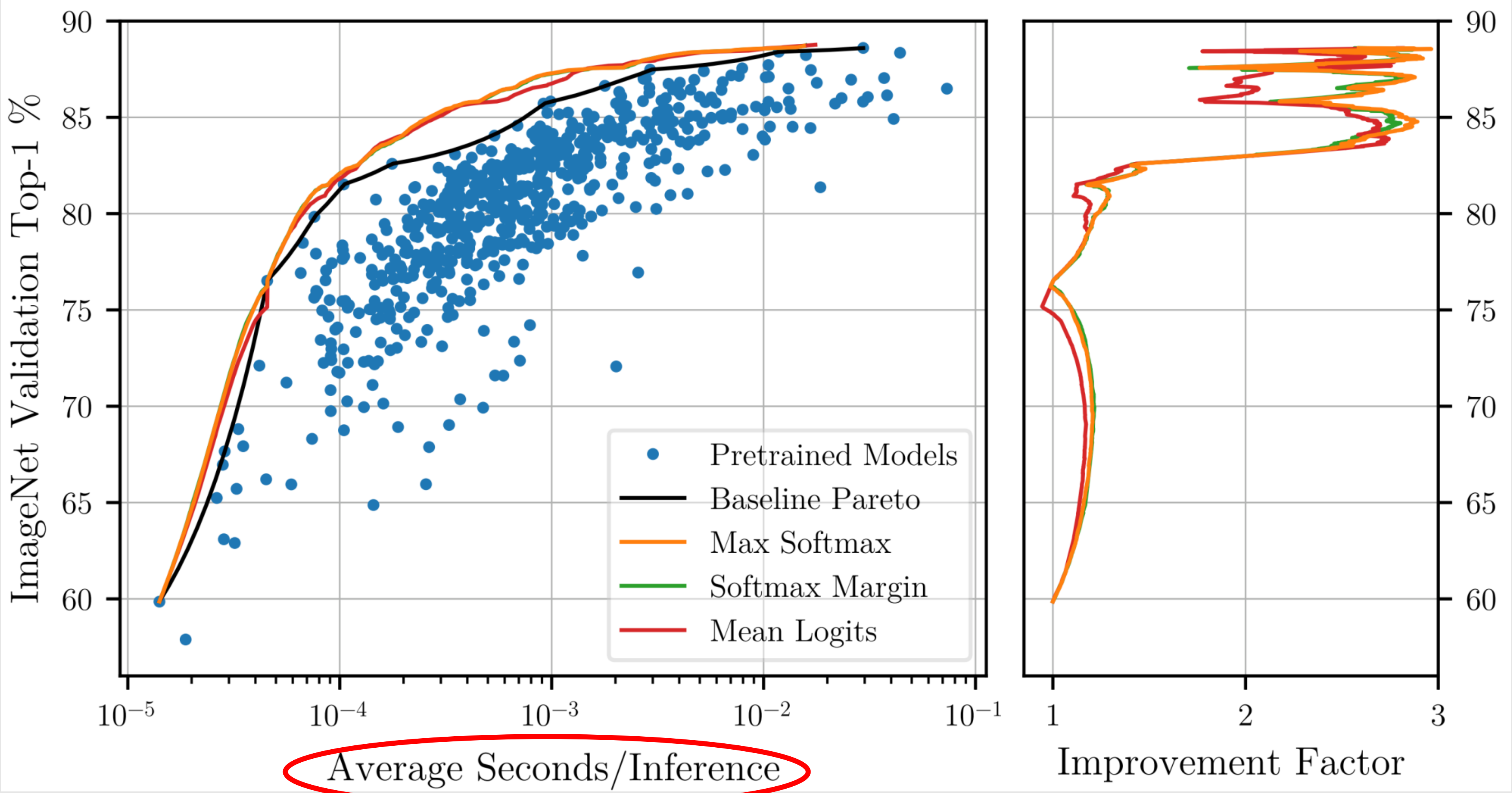
HUAWEI

The world is not ideal: Distribution Shifts

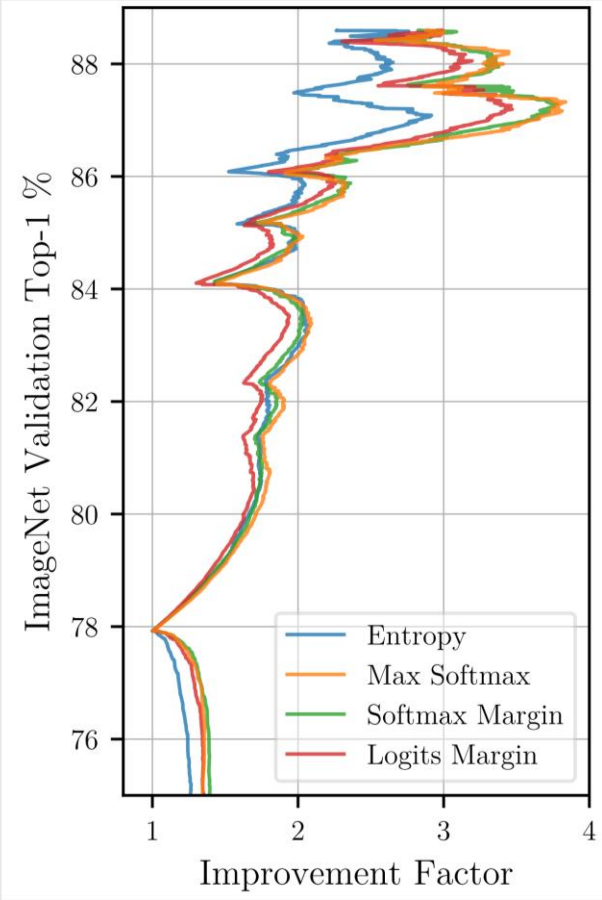
- Data during execution can be different than during training and threshold selection
- Test with ImageNetV2 (much harder dataset) w/o fine-tuning → more hard cases



The world is not ideal: MAC operations v. real execution time



MAC-based speed-up



- The smaller/optimized models are not as much faster as MAC count suggests (not cascade-related)
- 2.8-3x speed-up can be achieved in practice with real device measurements

HUAWEI

Only Image Classification? NLP Results

- SST-2: Stanford Dataset for predicting Sentiment from longer Movie Reviews

The method generalizes to other datasets, task types

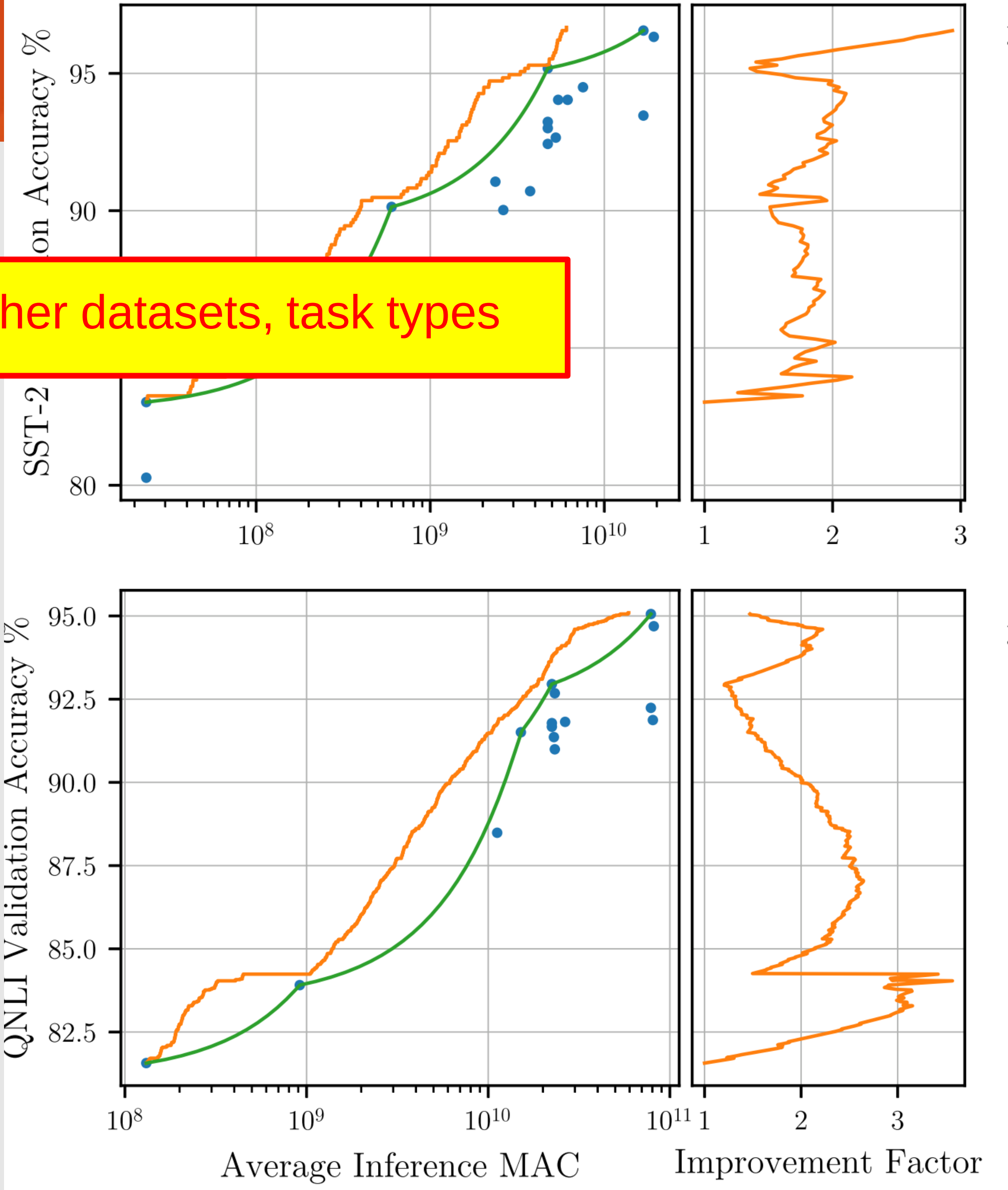
- QNLI: Question-answering Natural Language Inference (based on SQuAD v1.1 – Stanford Question Answering Dataset)

  Context: “As at most other universities, Notre Dame's students run a number of news media outlets. The nine student-run outlets (…)”

  Q: “When did the Scholastic Magazine of Notre dame begin publishing?”

  A: “September 1876”

- Challenge: fewer datasets – more possible?



Conclusion

A simple concept: Don't bother the master with questions the apprentice can answer

The good

- **~3x speed-up** (~energy savings, ~cost reduction) **at equal accuracy**
- **No modifications** to hardware or low- to medium-level software
- **No (re-)training** of any models
- **No engineering effort**

Apply this whenever possible,
save 3x cost/energy almost for free

The bad / limitations:

- Need to **store the smaller model**, too (~10% more); **need multiple models** to be available
- **Worst-case execution time** is worse (~10% longer), but average is much better (~3x)
  → good for data center (it evens out) & embedded/mobile (far less energy), no benefit for real-time
- **Distribution shifts** can impact effectiveness

HUAWEI

ZURICH RESEARCH CENTER