Beyond neural scaling laws: beating power law scaling via data pruning

NeurIPS 2022

Author: Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, Ari S. Morcos

Presenter: Wenjin Zhang

Background

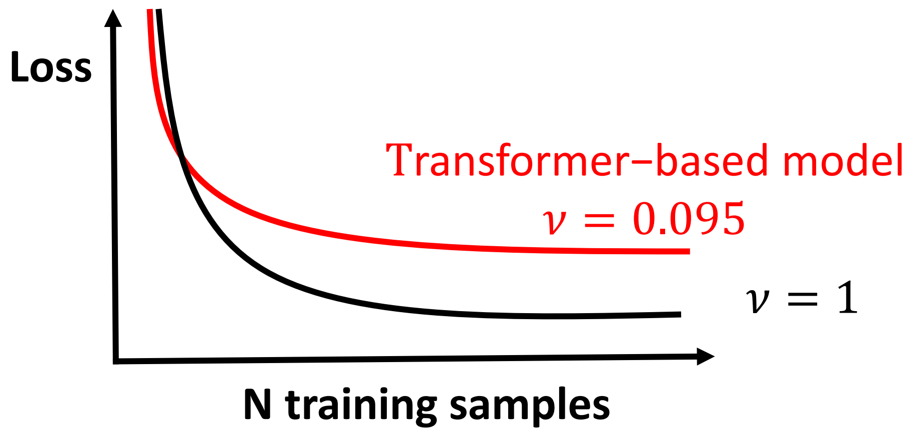
- Neural scaling laws: show the dependency between the error rate of a model and the amount of training data (or model size or compute).
- Recent works show neural scaling laws follow a power law:

$$loss \approx N^{-\nu} = \frac{1}{N^{\nu}}$$

ν **is Problem depend**

Error

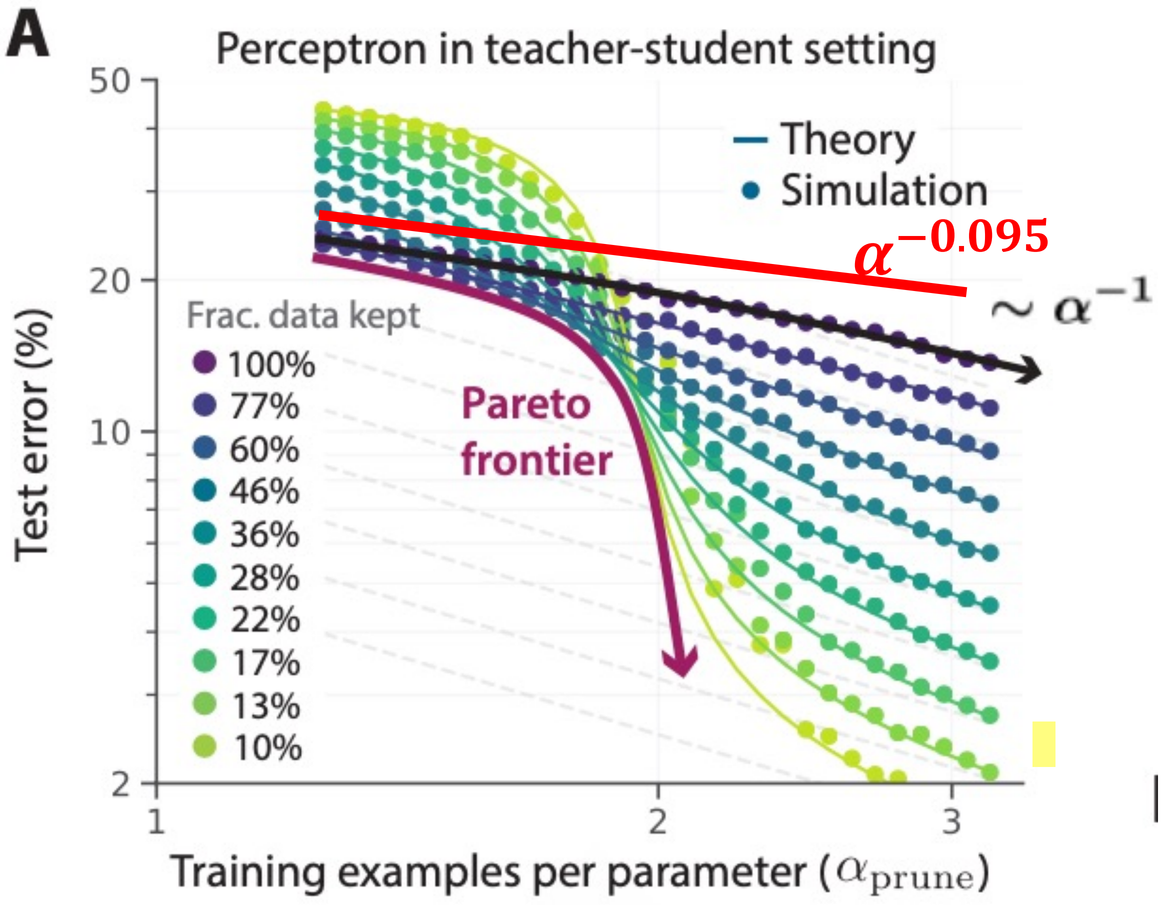
data point #



Problem Statement and Motivation

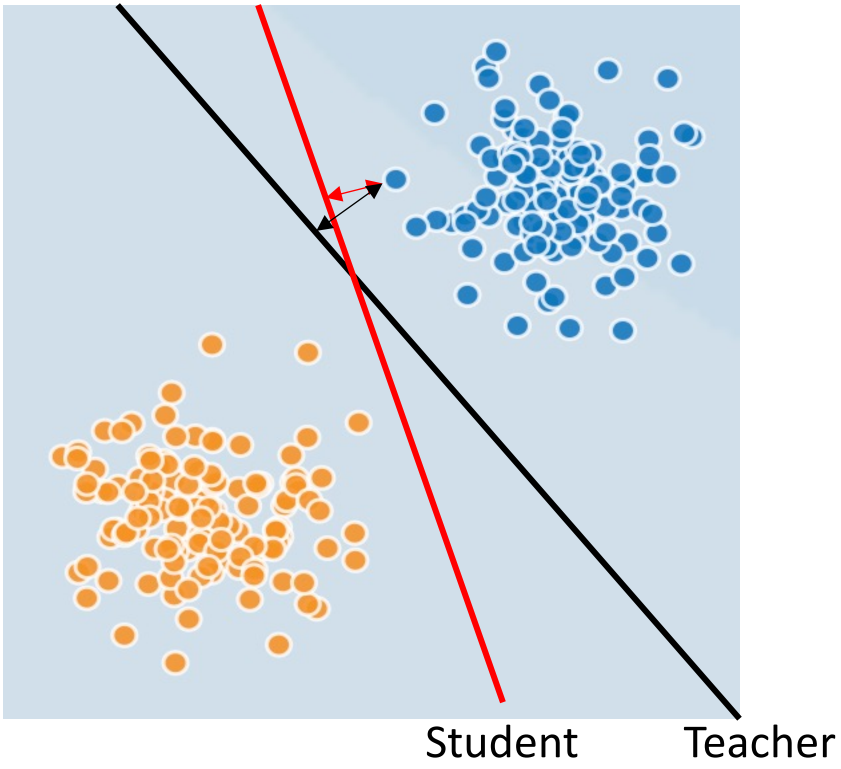
- For large vision transformers: an additional 2 *billion* data points (starting from 1 billion) leads to an accuracy gain on ImageNet of a few percentage points

- Can we do better?
  - Goal: how to make the loss reduce faster than a power law?



Data Pruning Method

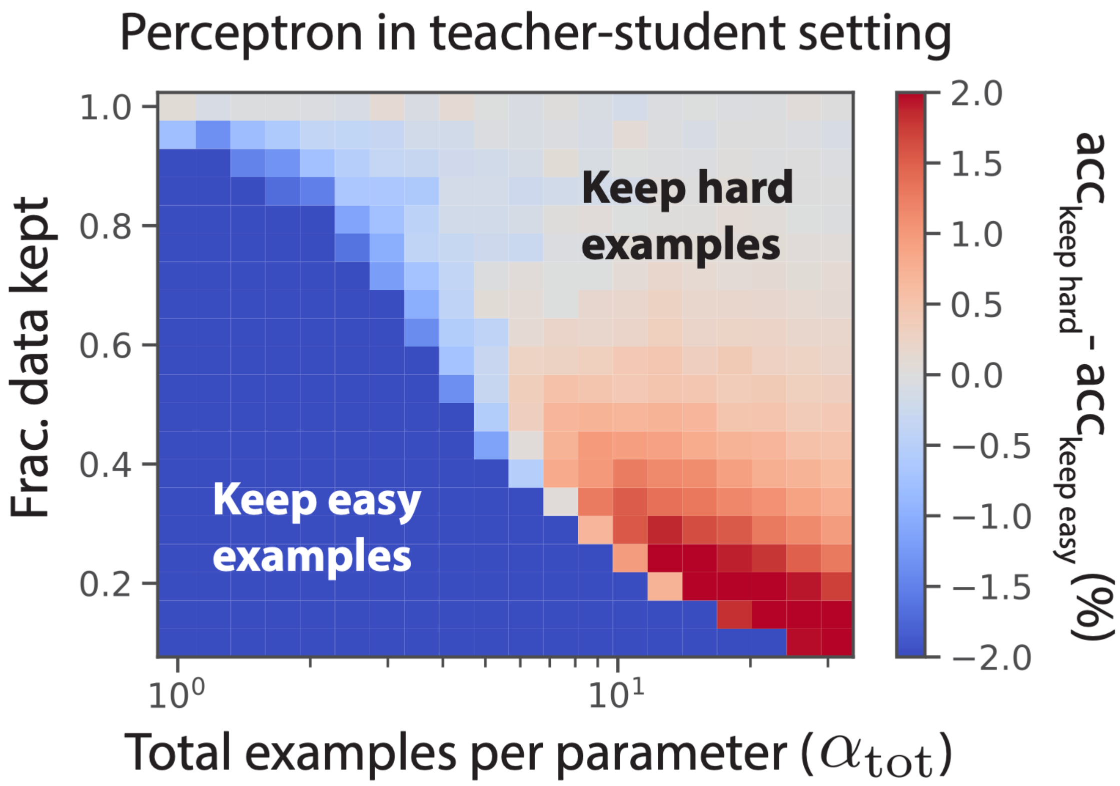
- The authors suggest to train the model with a fraction of hard examples not easy examples
- How to define easy and hard data point
  - Easy: large margin
  - Hard: small margin
- The idea is based on teacher and student setup
  - ➢ Leverage a pre-trained teacher to guide student model.
  - ➢ Teacher is well-trained
  - ➢ Student is only trained a few epoch and under-trained

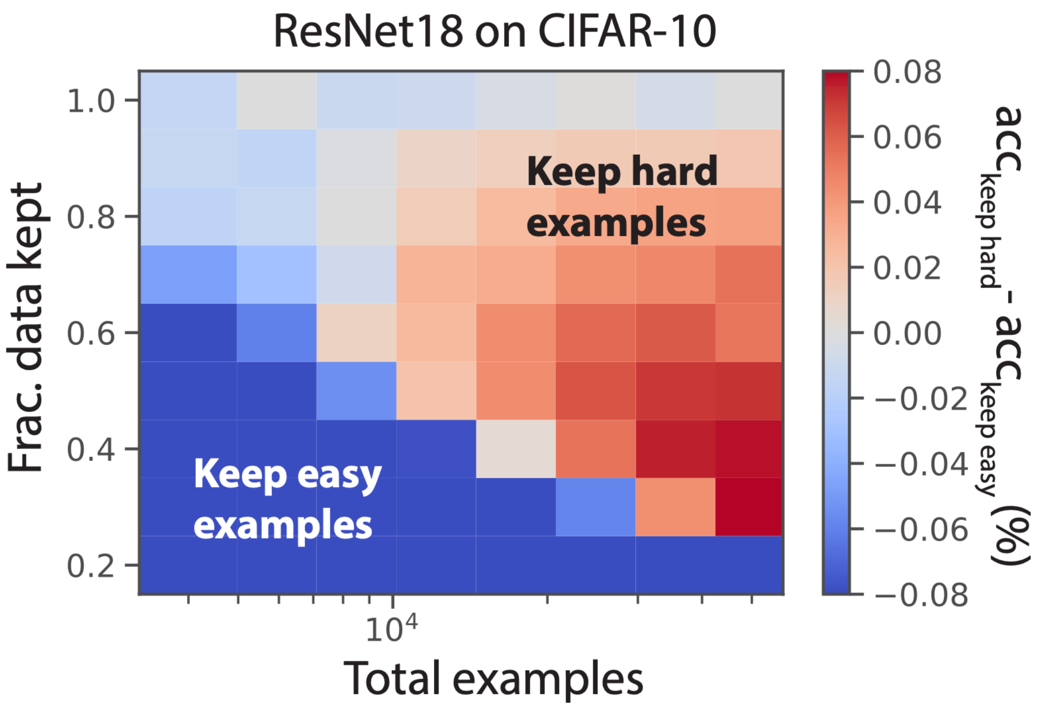


Margin of one sample is defined as difference between the distance to different decision boundary

Important Conclusion 1

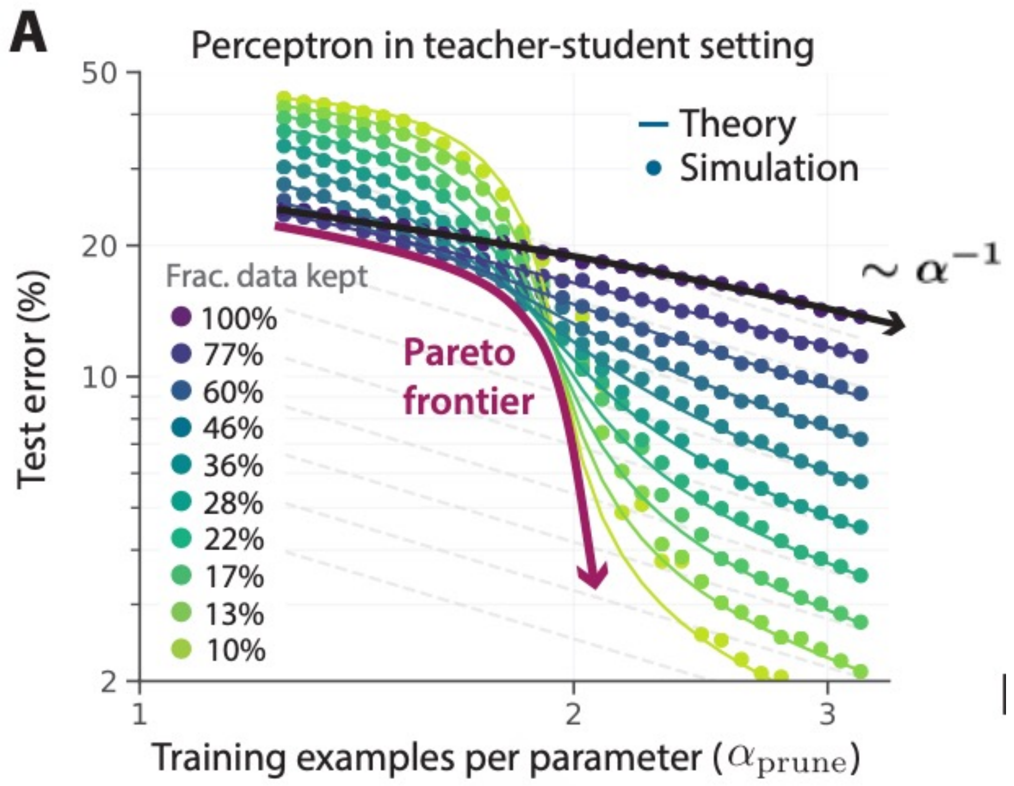
- The best pruning strategy depends on the amount of initial data





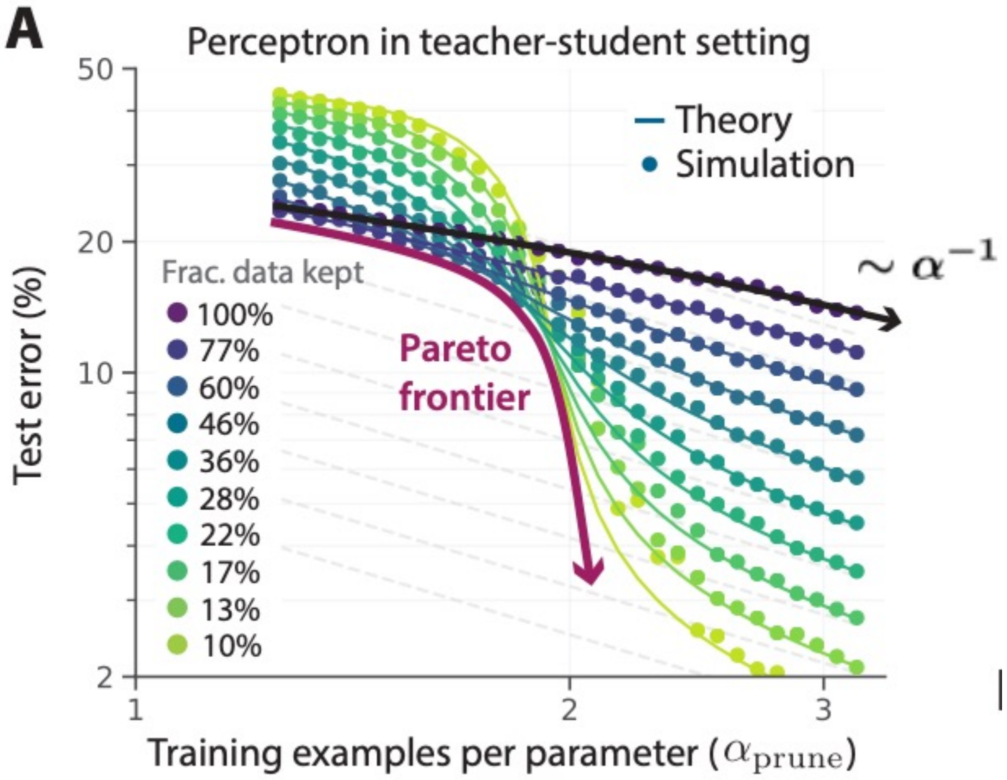
Important Conclusion 2

- Optimal pruning results into an exponential scaling law: Only if we can apply optimal pruning.



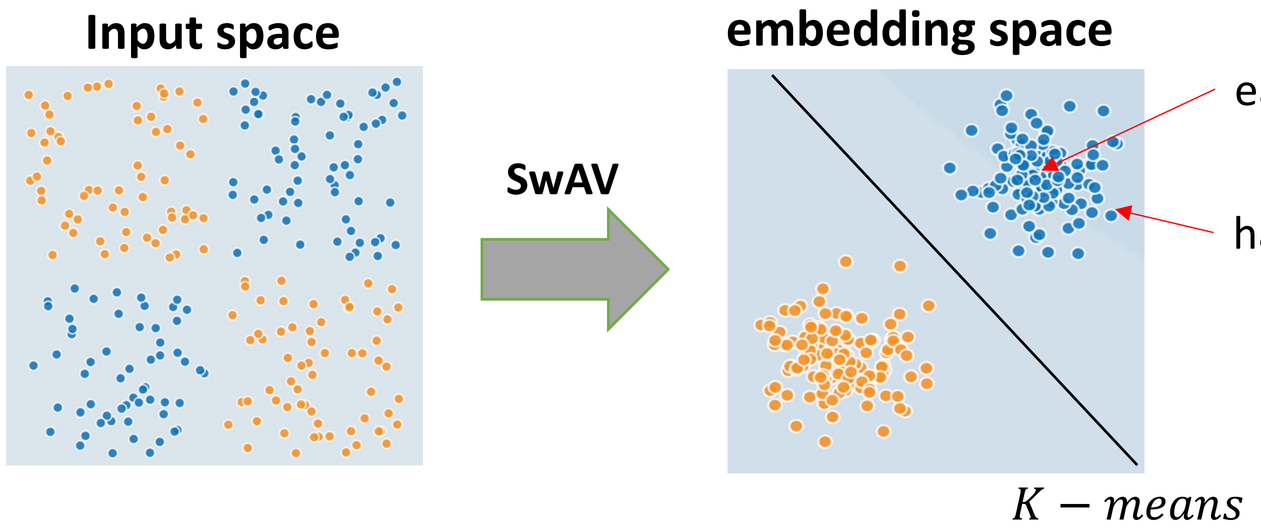
Important Conclusion 3

- An imperfect pruning metric yields a cross over from exponential to power law scaling

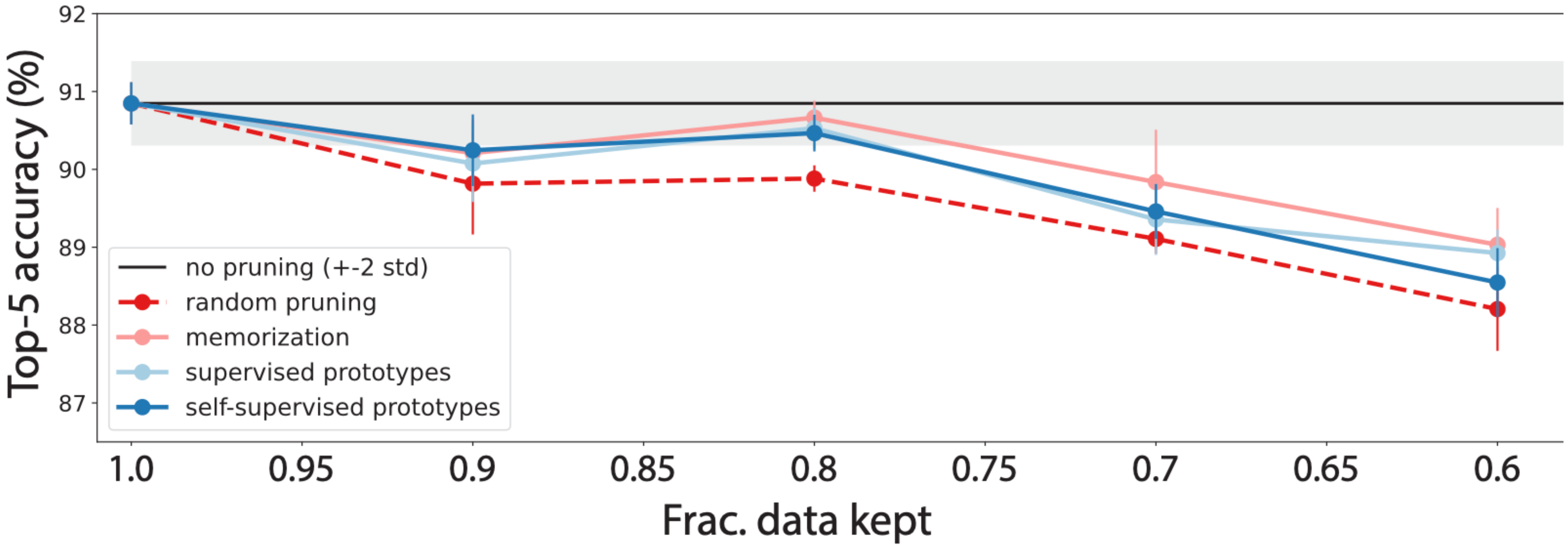


Problem: How to pruning no labeled dataset

- Without labeling, how to define hard or easy samples?
- Solution:



Experiment



92
91
90
89
88
87
Top-5 accuracy (%)
no pruning (+-2 std)
random pruning
memorization
supervised prototypes
self-supervised prototypes
1.0
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
Frac. data kept