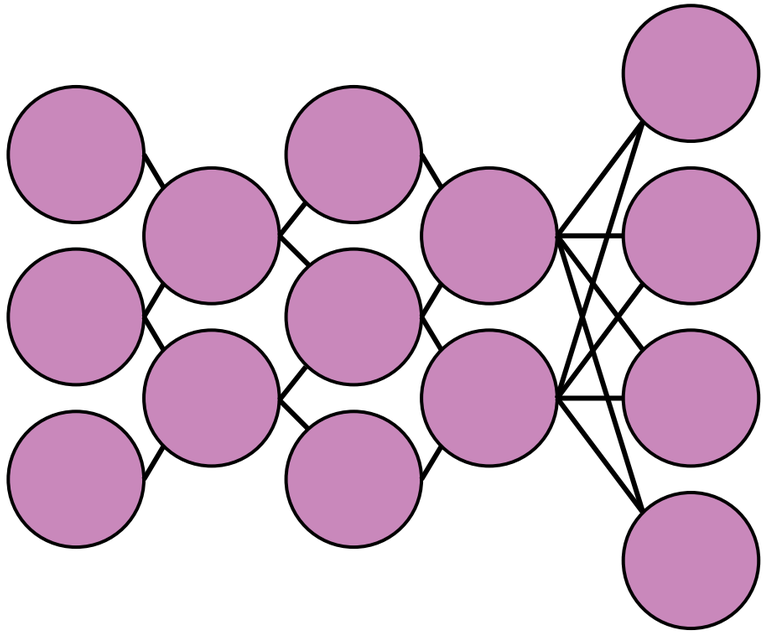
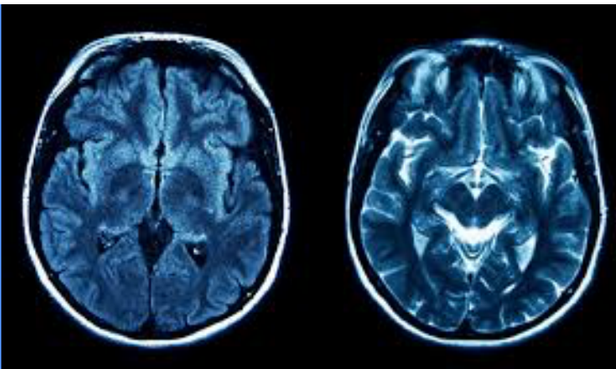
Accelerating Deep Learning with the Biggest Losers

Angela H. Jiang, Daniel L.-K. Wong, Giulio Zhou,
David G. Andersen, Jeffrey Dean, Gregory R. Ganger,
Gauri Joshi, Michael Kaminsky, Michael A. Kozuch,
Zachary C. Lipton, Padmanabhan Pillai

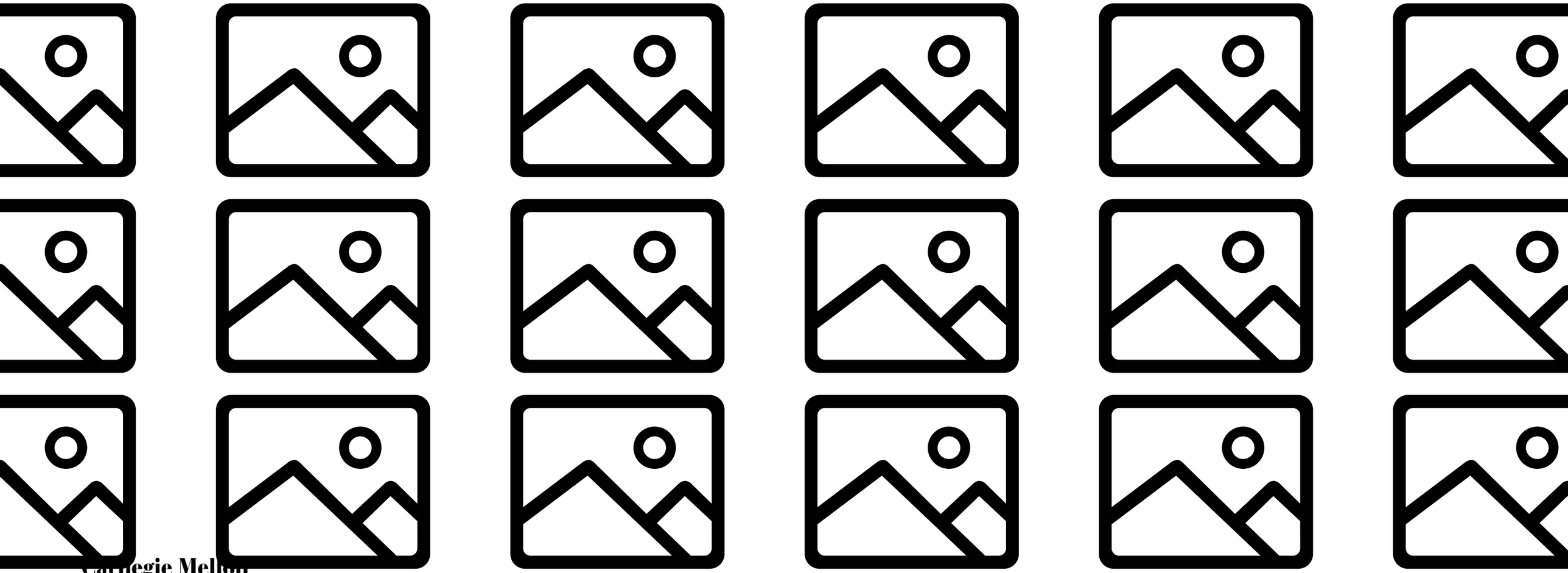
Carnegie Mellon
Parallel Data Laboratory

Deep learning enables emerging applications

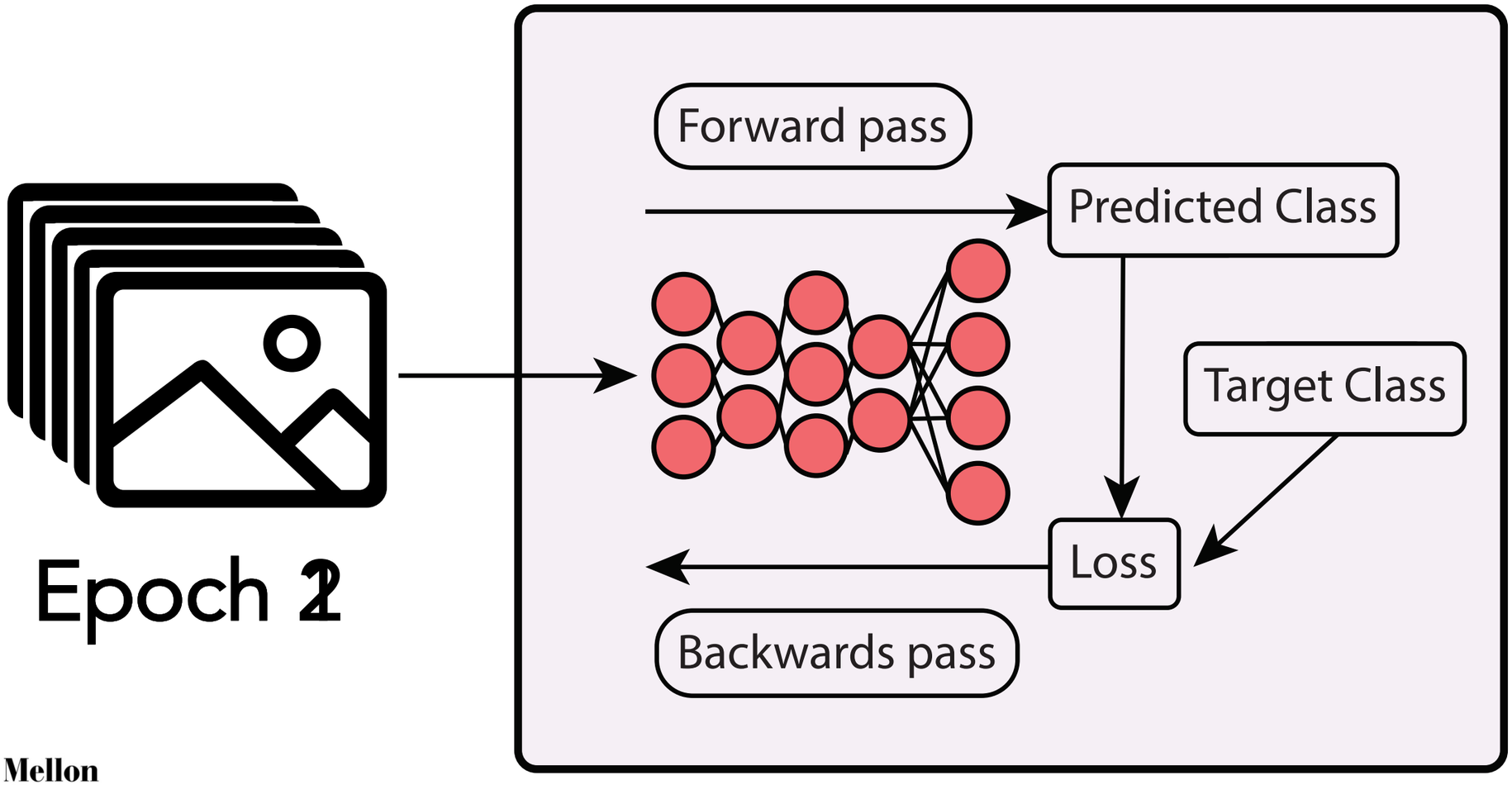




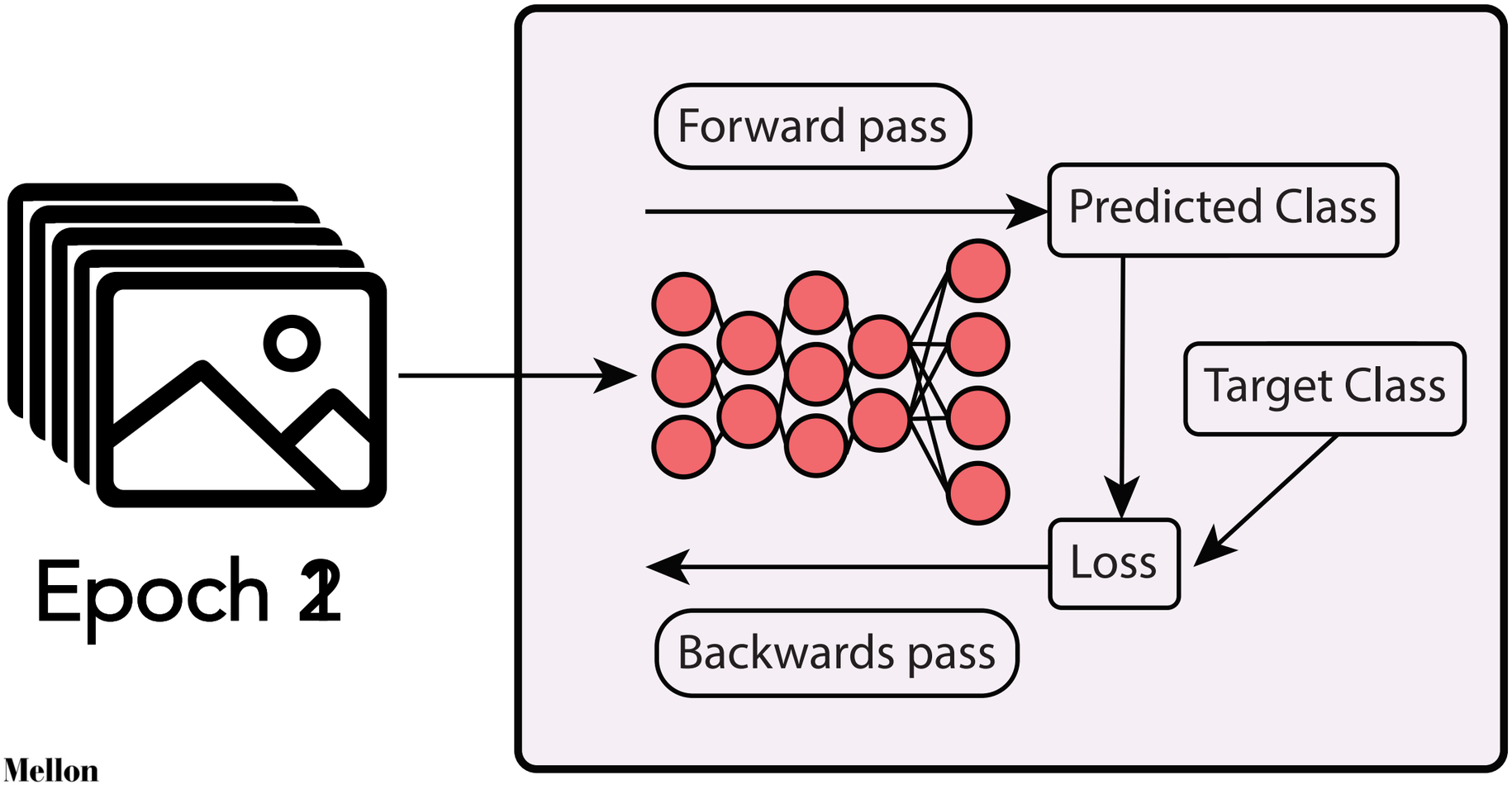
DNN training analyzes many examples



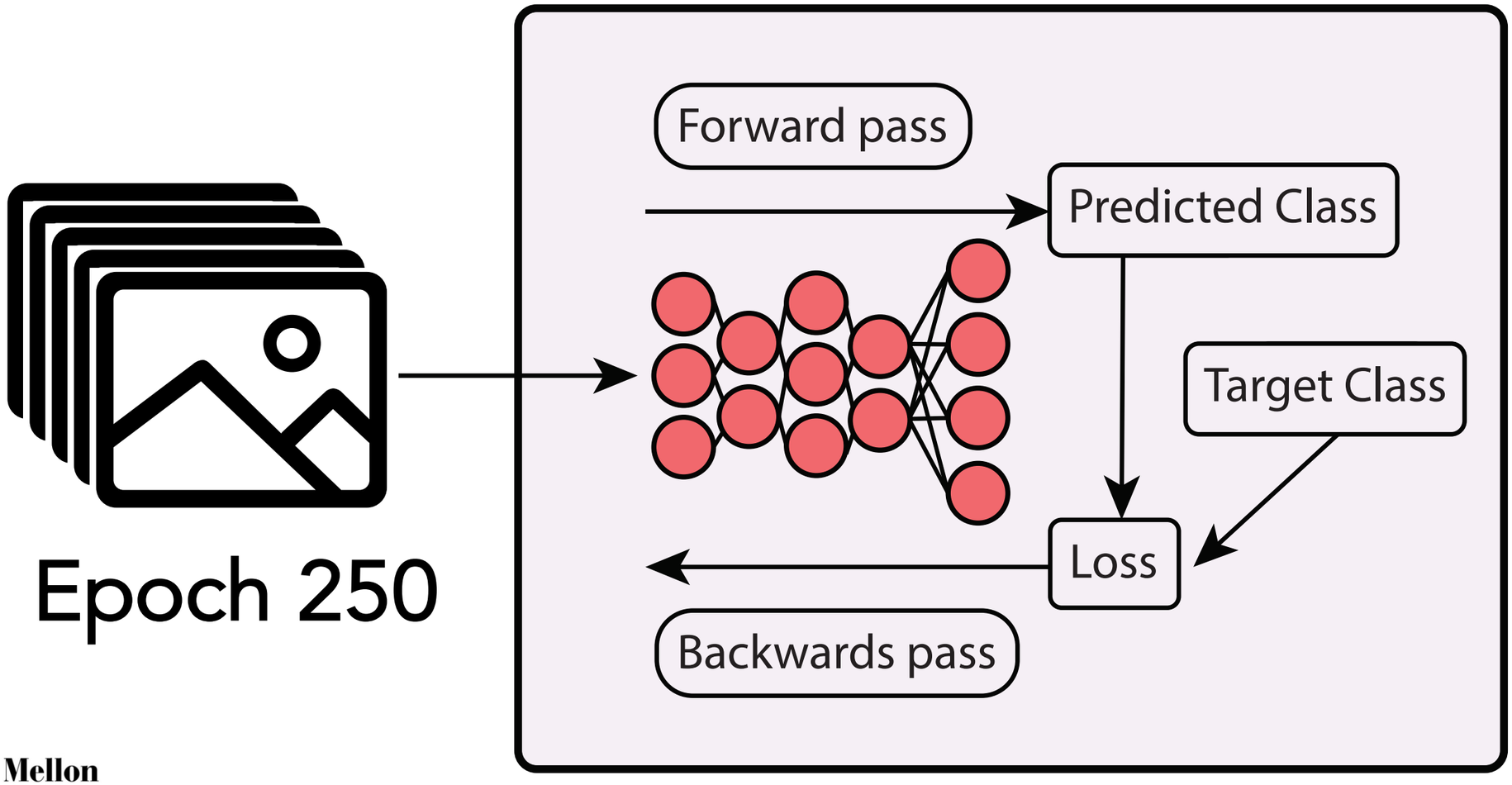
DNN training analyzes an example many times



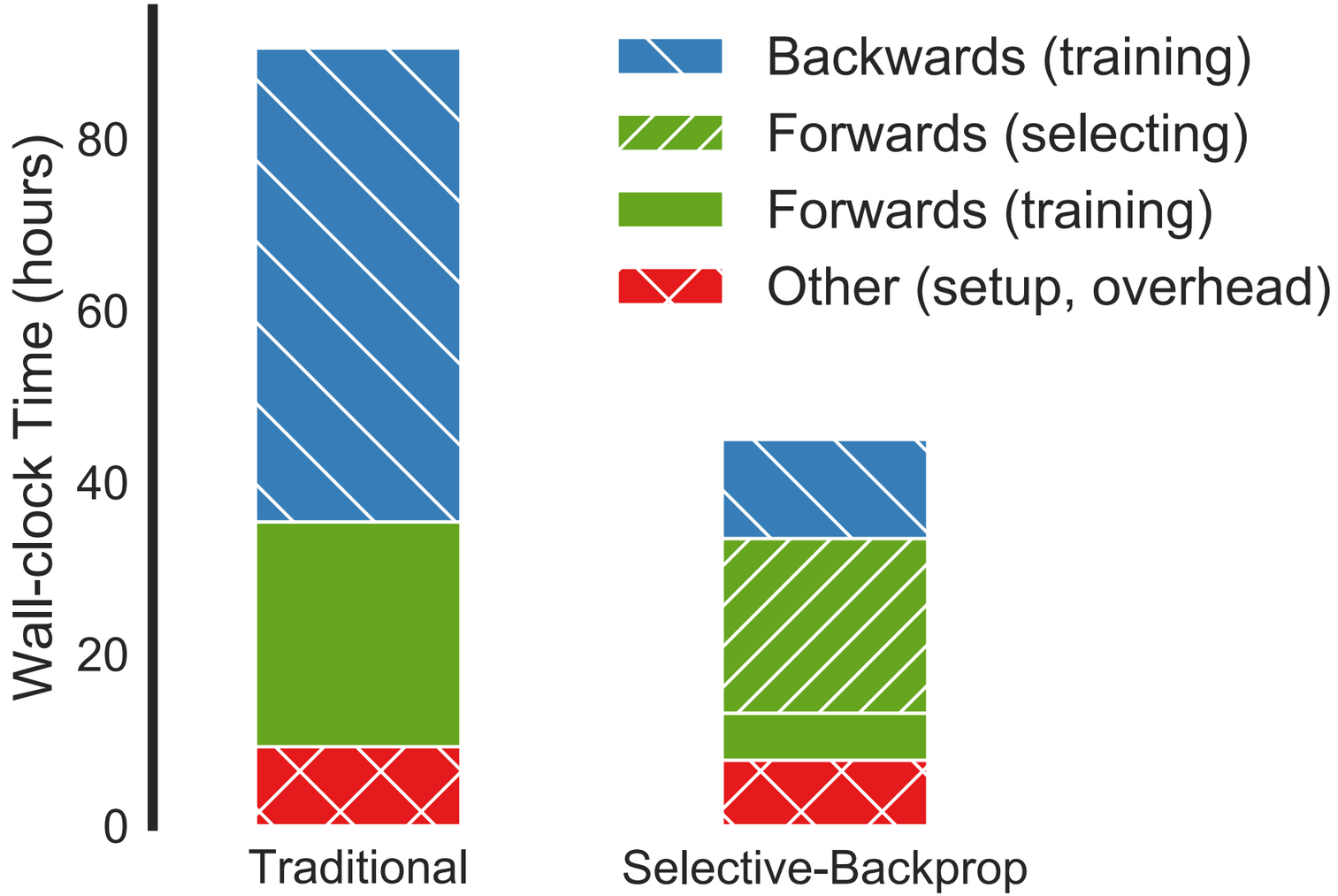
DNN training analyzes an example many times



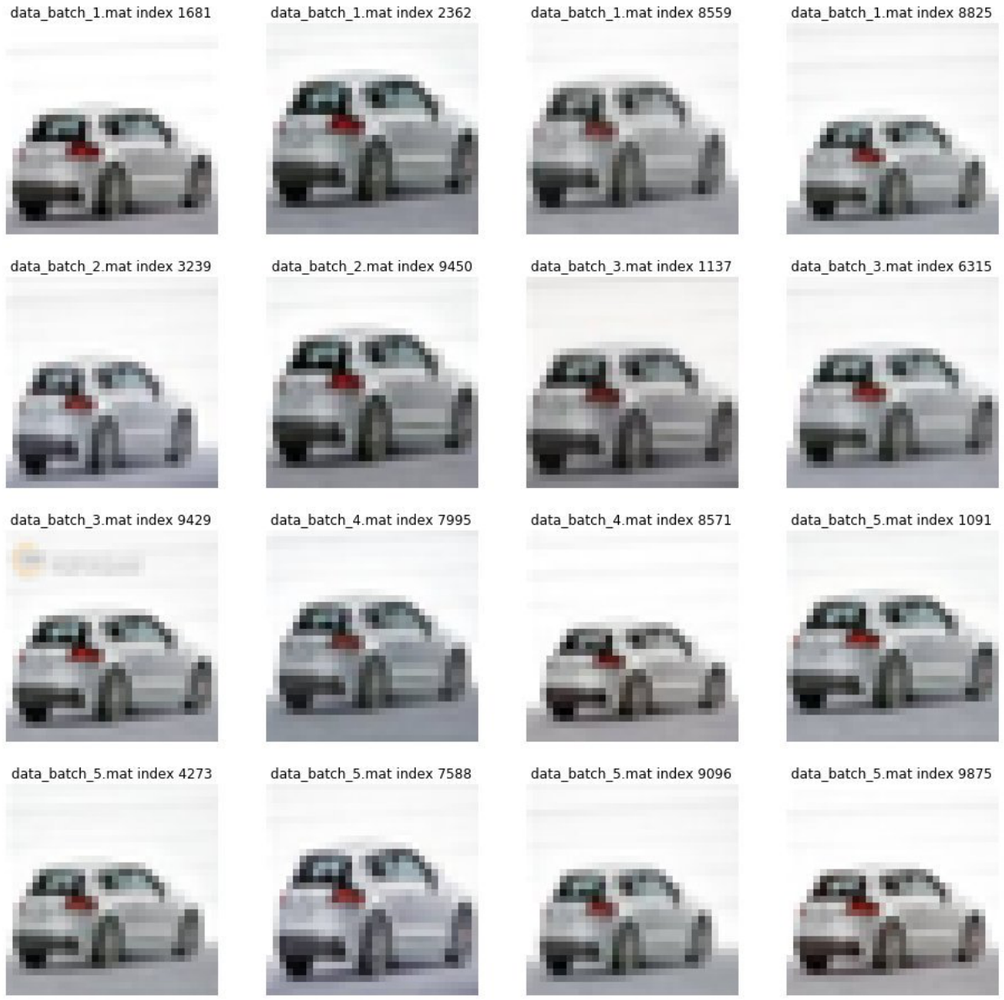
DNN training analyzes many examples



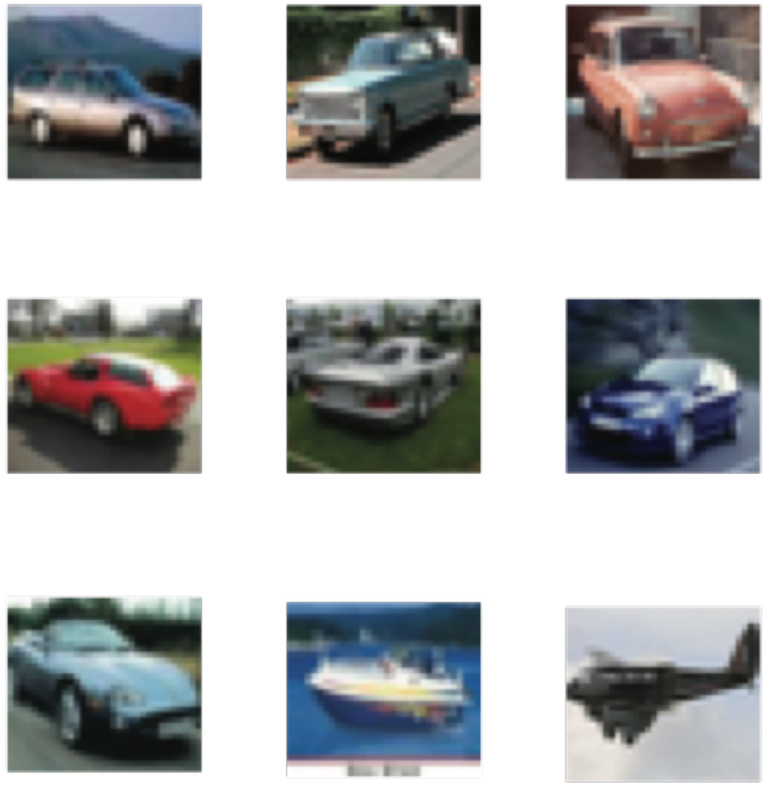
SelectiveBackprop targets slowest part of training



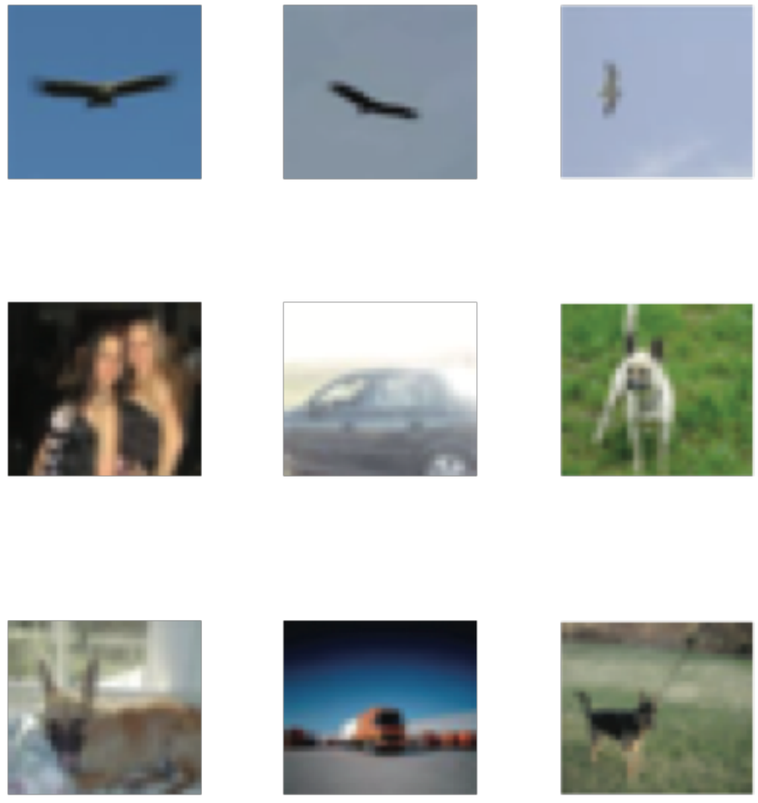
Not all examples are equally useful



Prioritize examples with high loss



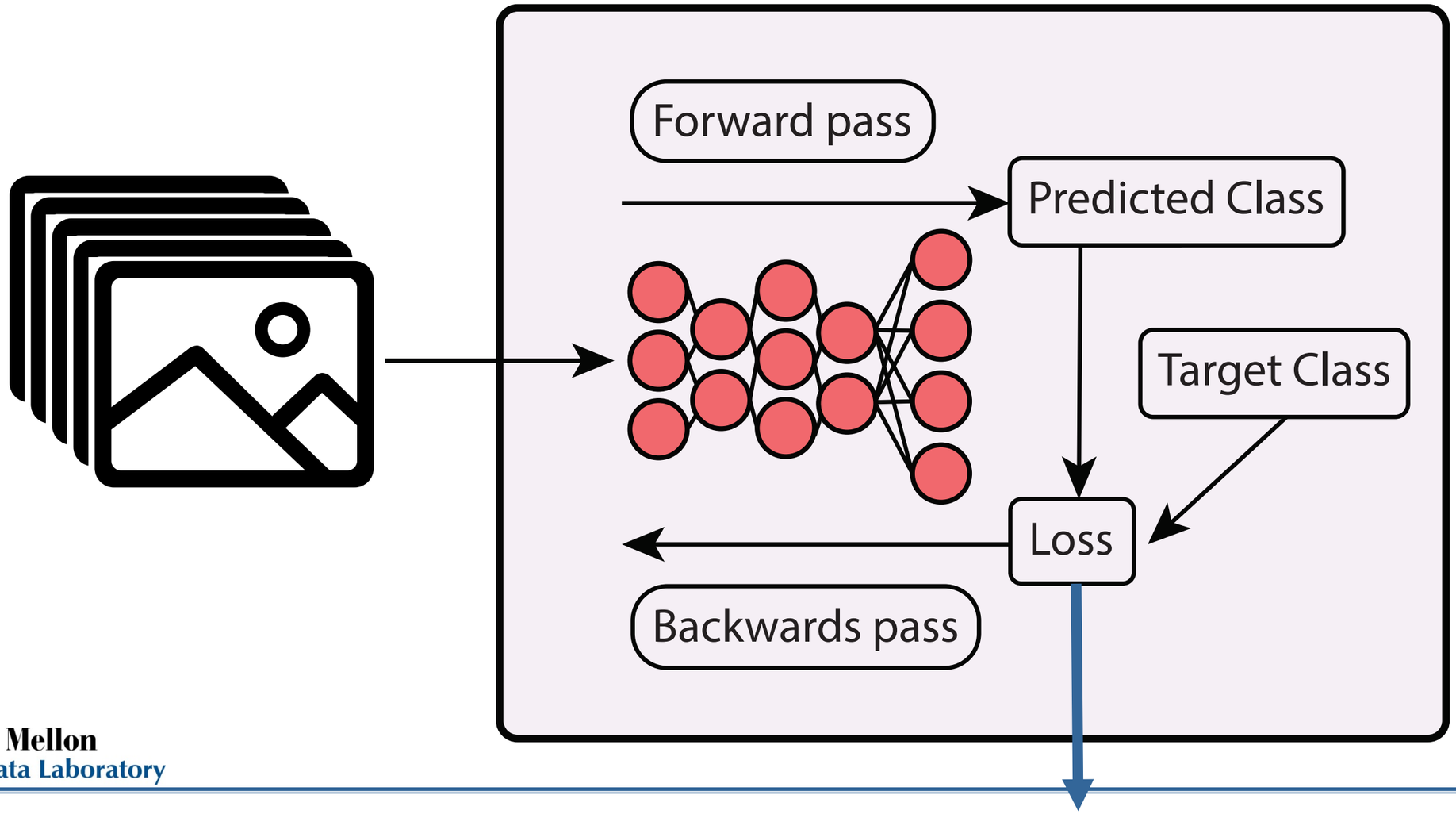
Examples with low loss



Examples with high loss

Selective Backprop algorithm

DNN training analyzes an example many times



Attempt #1: Deciding with a hard threshold

if loss > threshold: backprop()

Attempt #2: Deciding probabilistically with loss

$$P(backprop) = normalize(loss, 0, 1)$$

SB idea: Use relative probabilistic calculation

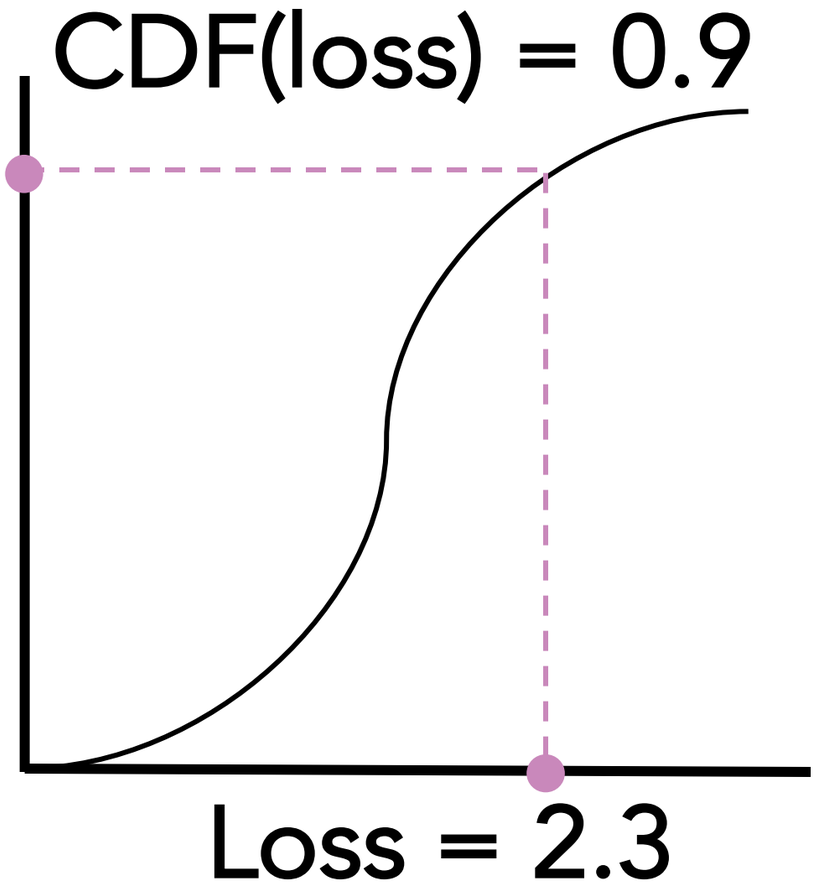
$$P(\text{backprop}) = \text{Percentile}(\text{loss}, \text{recent losses})$$

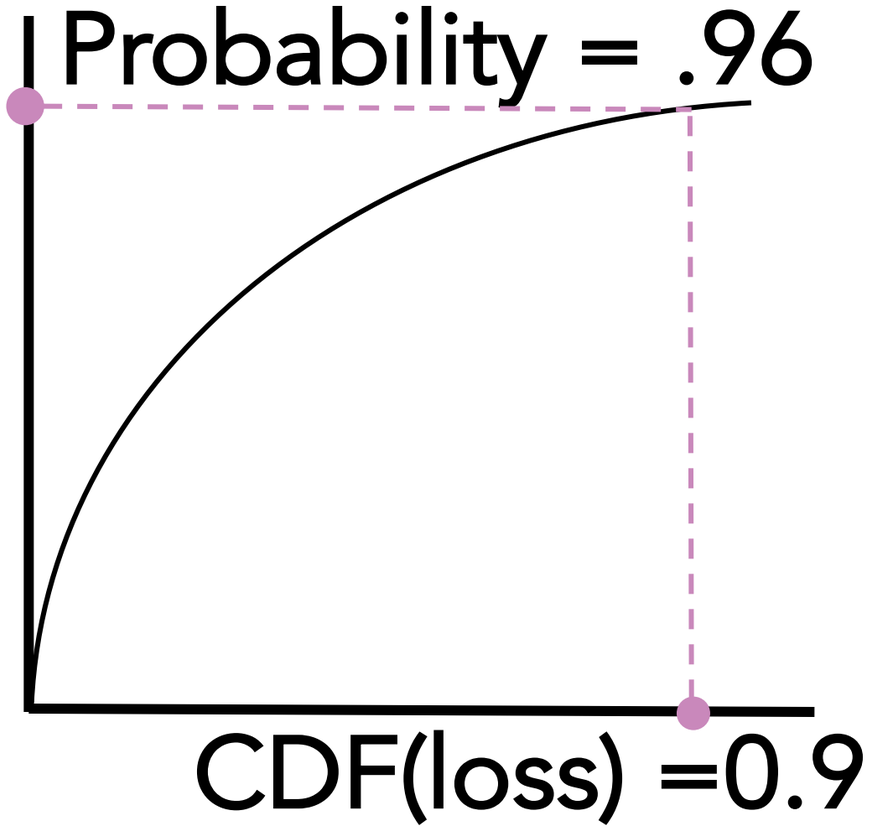
SB idea: Use relative probabilistic calculation

$$P(\text{backprop}) = \text{Percentile}(\text{loss}, \text{recent losses})^{B}$$

Higher B makes SB more selective

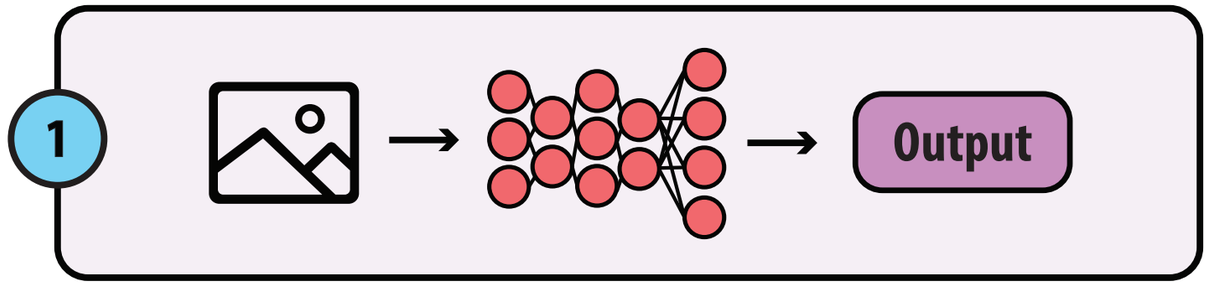
Example of probability calculation



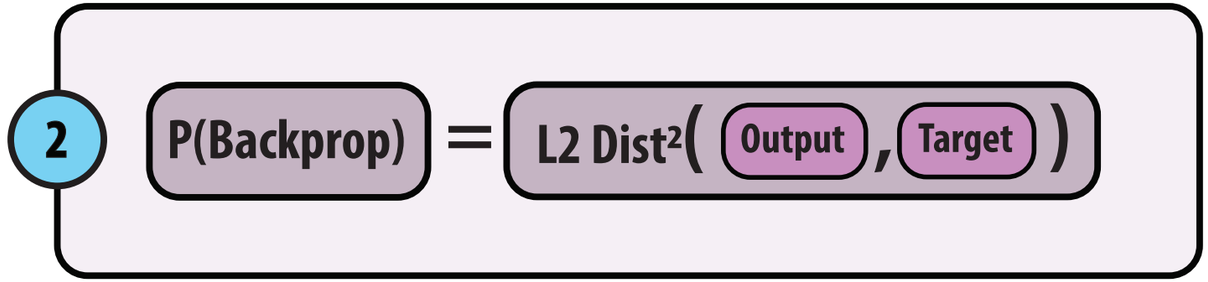


Selective-Backprop approach

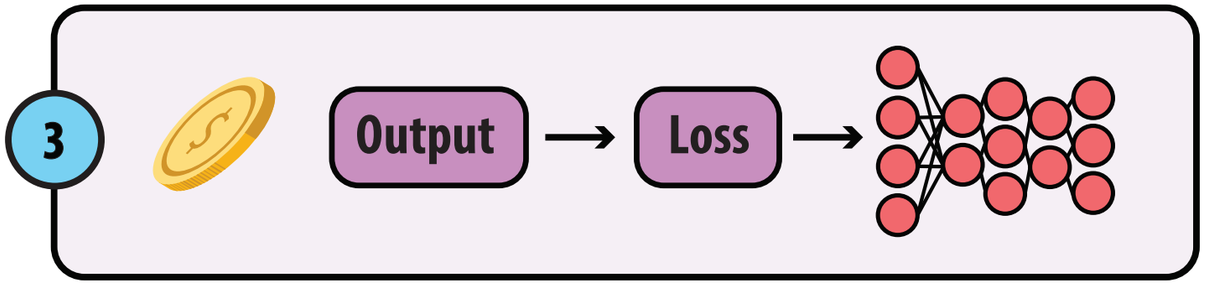
Forward propagate example through the network



Calculate usefulness of backpropping example based on its accuracy

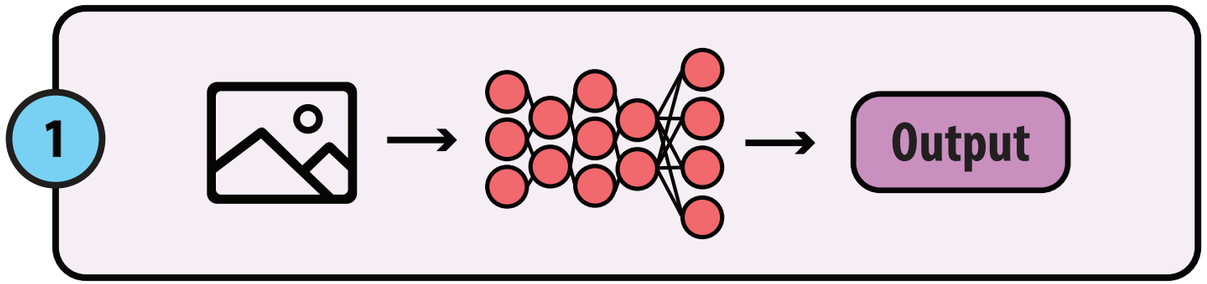


Decide probabilistically if we should backprop



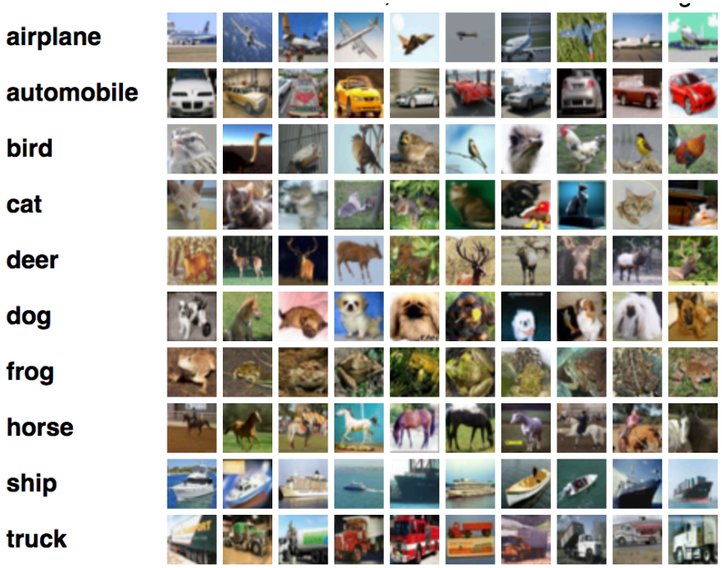
StaleSB reduces forward passes

Forward propagate example through the network
every n epochs



Evaluation of Selective Backprop

Datasets



CIFAR10

60,000 Training Images

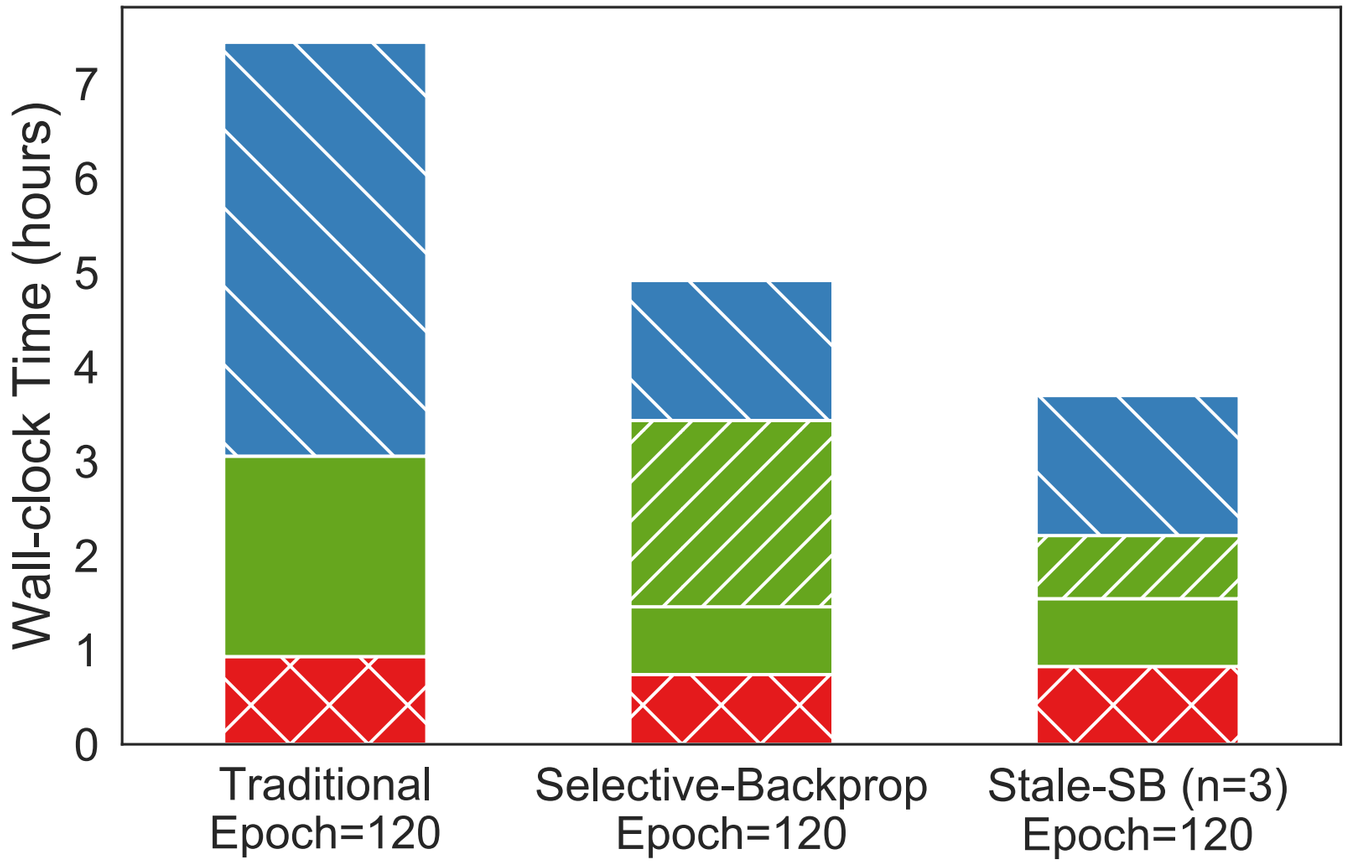
CIFAR100

60,000 Training Images

SVHN

604,388 Training Images

Train CIFAR10 to 4.14% (1.4x Traditional's final error)

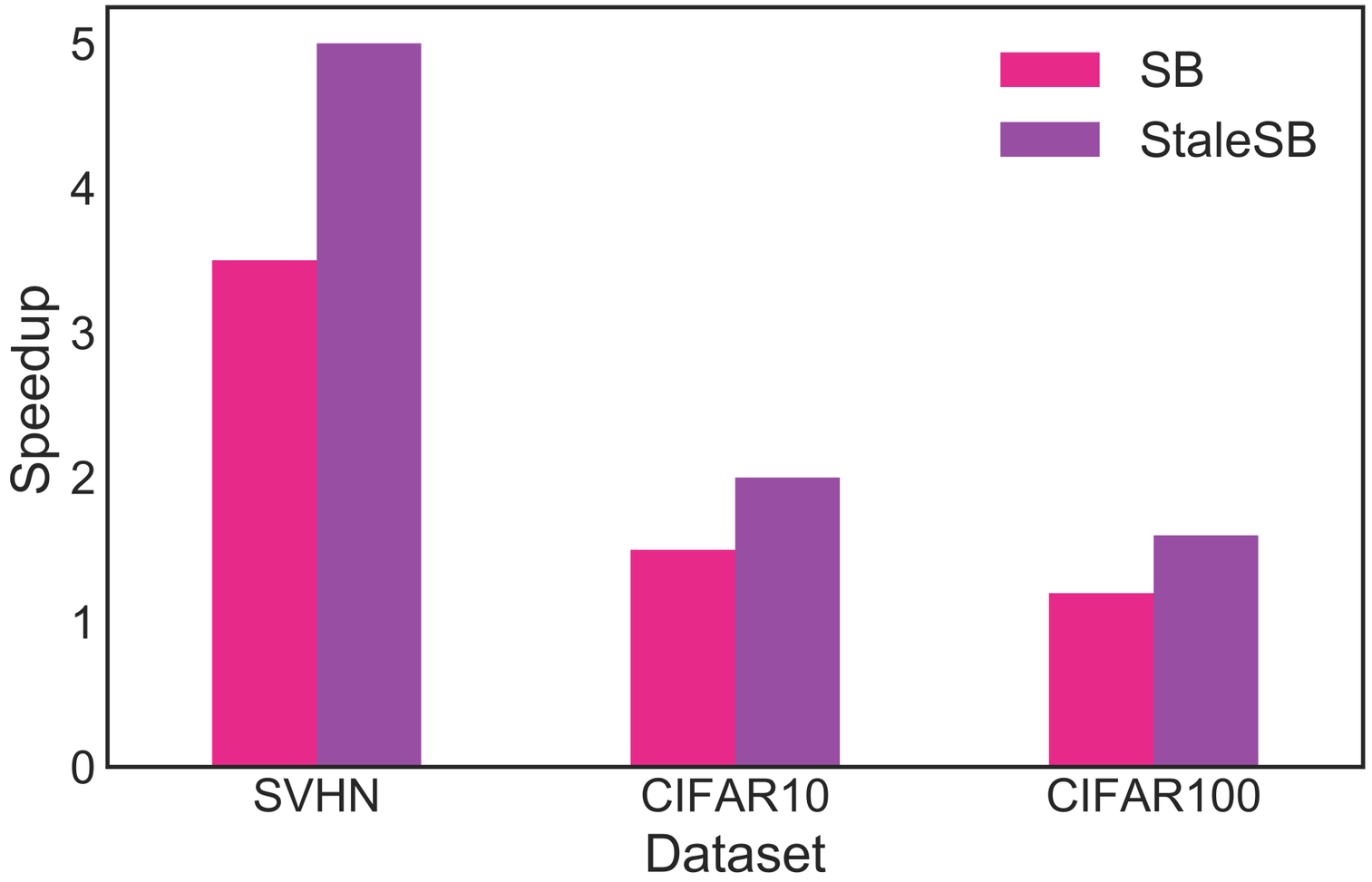


SB: 1.5X faster

StaleSB: 2X faster

Backwards (training)
Forwards (selecting)
Forwards (training)
Other (setup, overhead)

SB accelerates by 1.2-3.5x, StaleSB accelerates by 1.6-5sx



Compared approaches

Traditional

Classic SGD with no filtering

Katharopoulos18

State of the art importance sampling approach

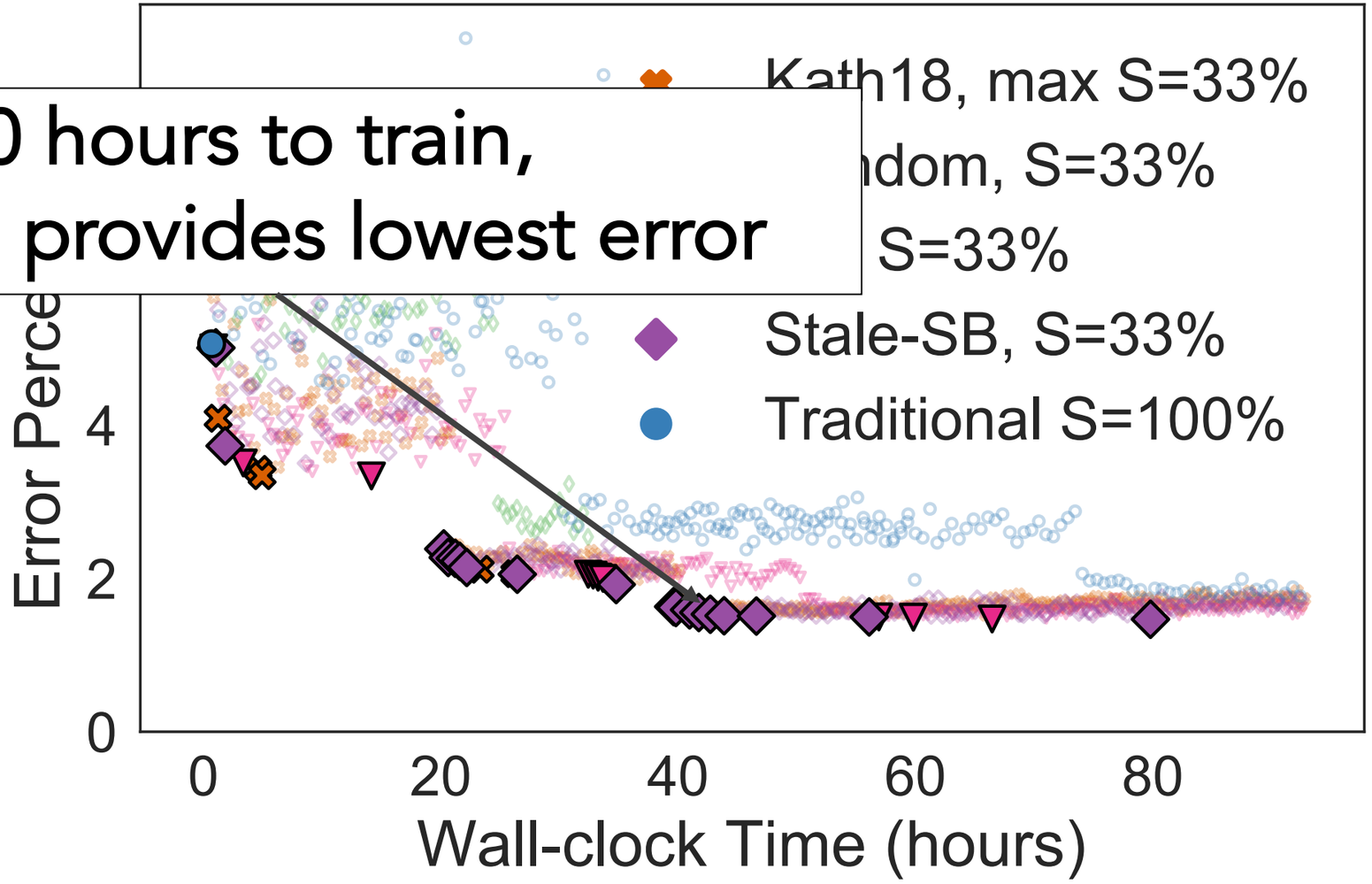
Random

Random sampling approach

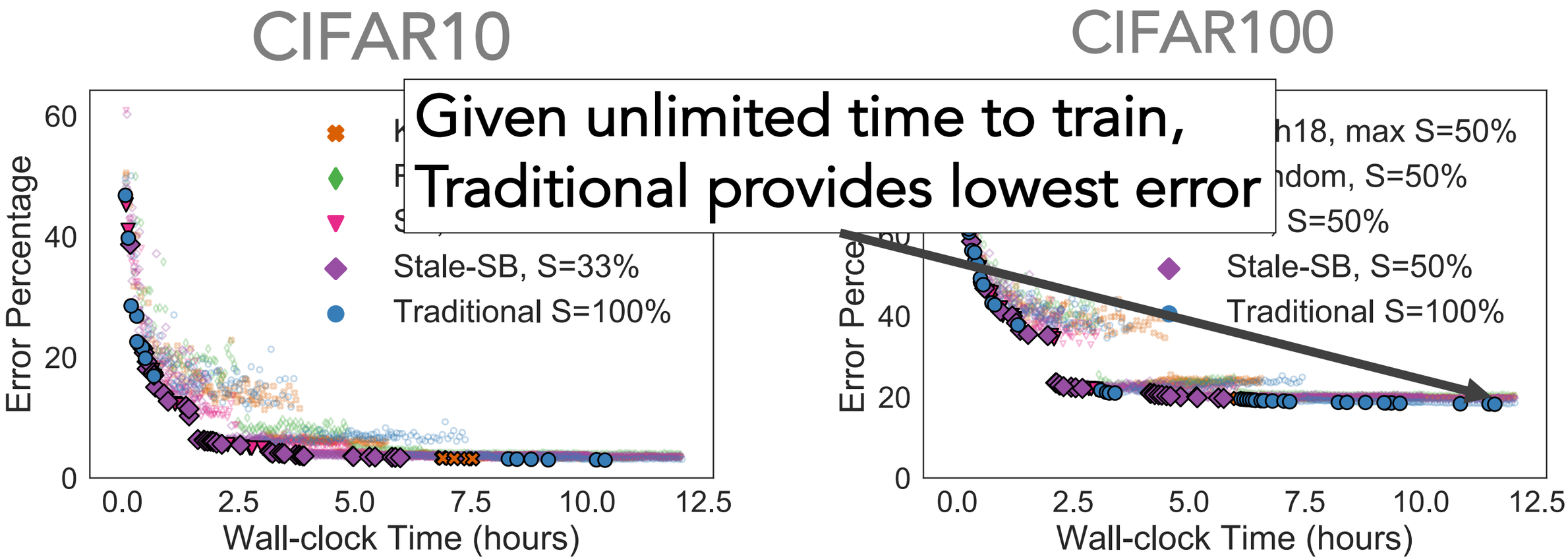
Selective-Backprop (Us)

SVHN

Given 40 hours to train,
Stale-SB provides lowest error



Most Pareto optimal points are SB or StaleSB

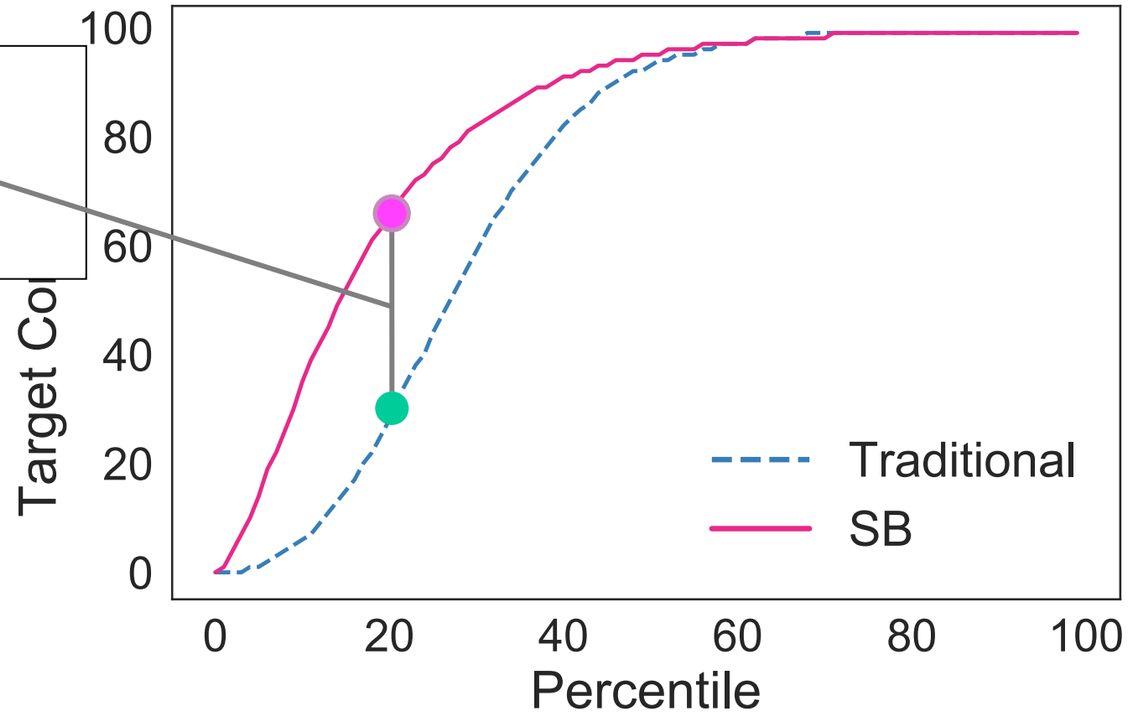


SB on CIFAR10 targets hard examples

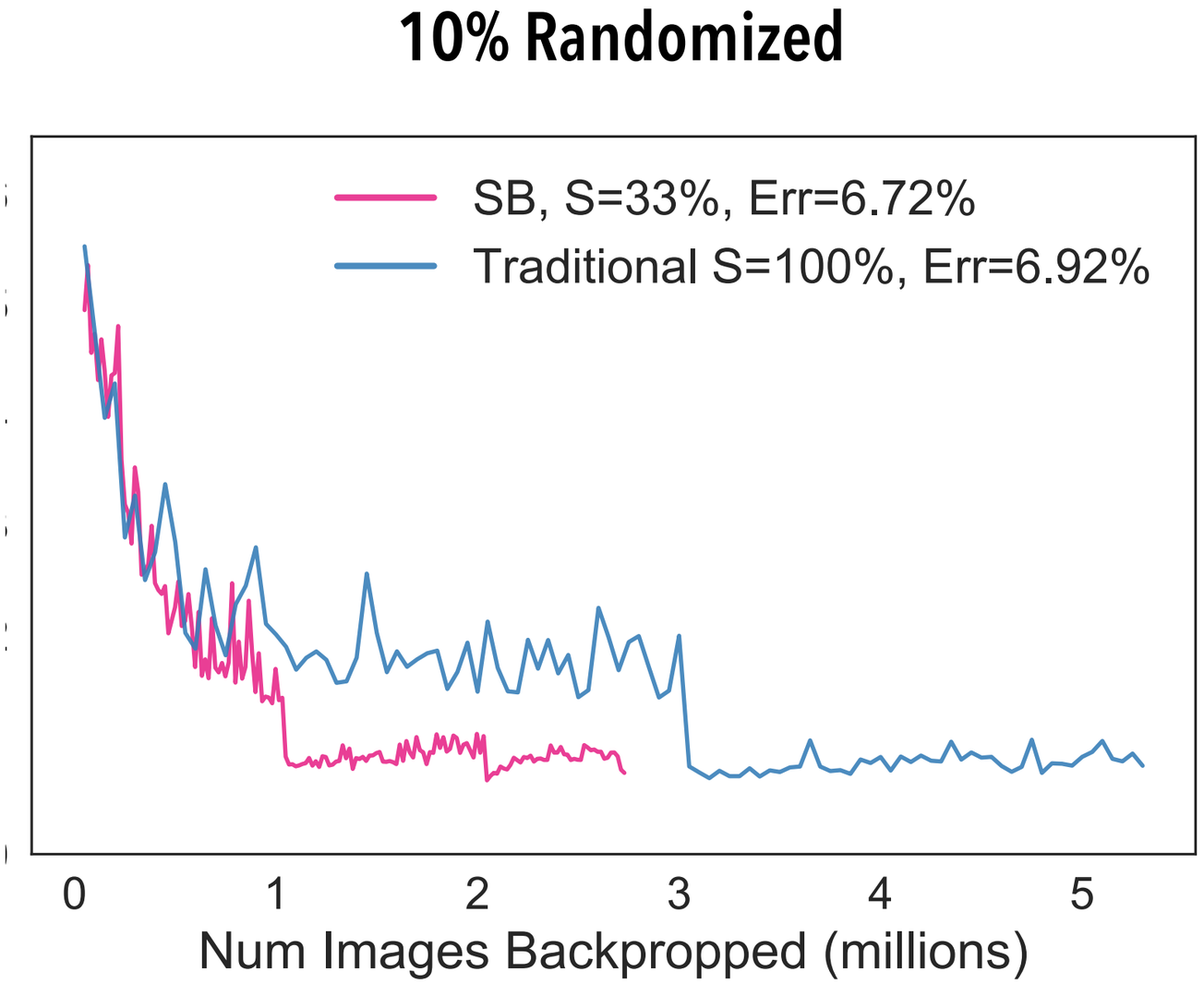
3% correct w/ Traditional
29% correct w/ SB

Output = [0.1, 0.3, 0.6]

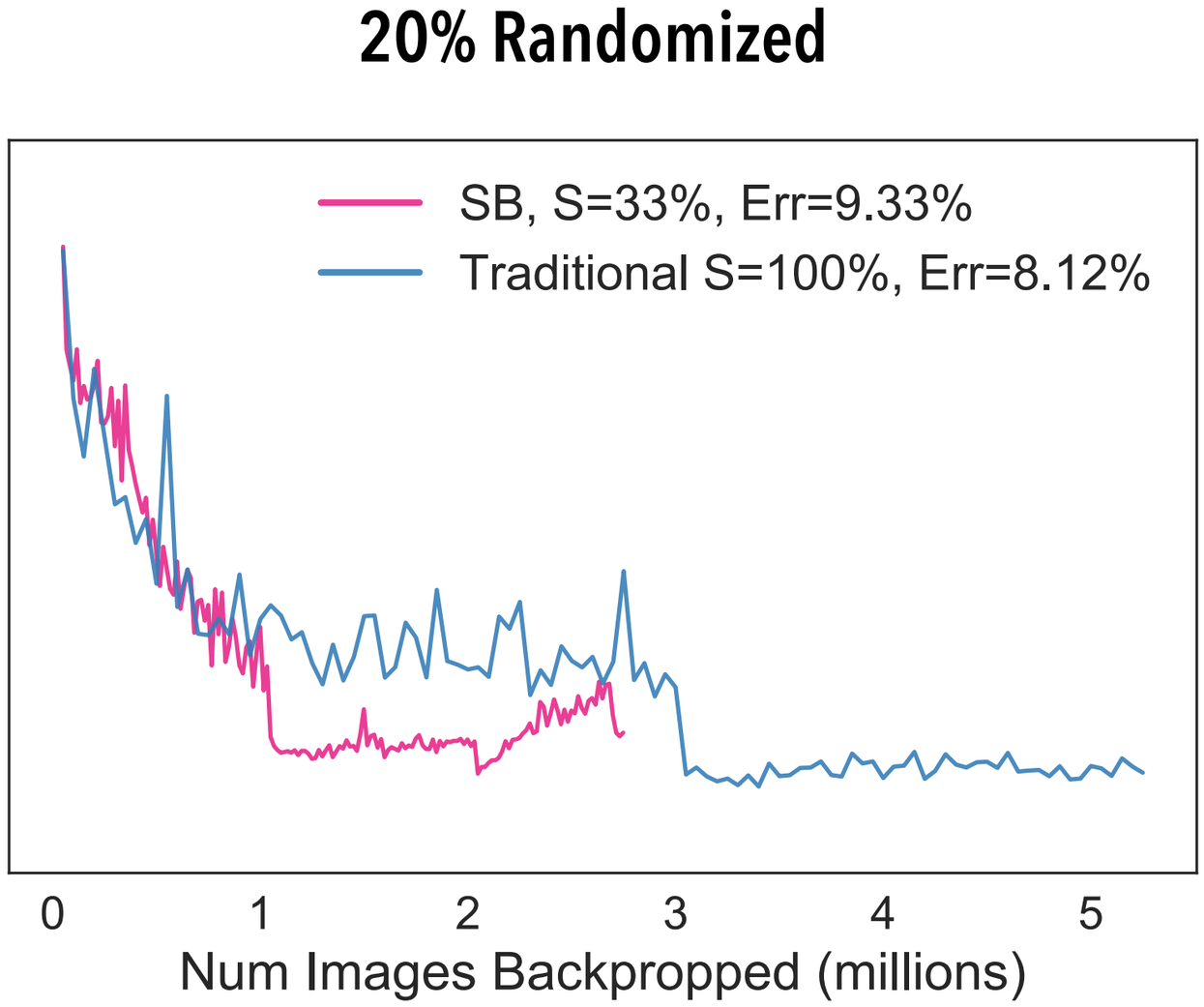
Target confidence = 0.3



SB is robust to modest amounts of error



SB is robust to modest amounts of error



More results at the poster session!

Selective-Backprop accelerates training

Reduces time spent in the backwards pass by prioritizing high-loss examples

SelectiveBackprop outperforms static approaches

Trains up to 3.5x faster compared to standard SGD

Trains 1.02-1.8X faster than state-of-the-art importance sampling approach

Stale-SB further accelerates training

Trains on average 26% faster compared to SB

www.github.com/angelajiang/SelectiveBackprop