Co-Teaching

Jacob Manuel

About the Paper

Co-teaching: Robust Training Deep Neural Networks with Extremely Noisy Labels

Conference: NIPS 2018

Bo Han[*1,2], Quanming Yao[*3], Xingrui Yu[1], Gang Niu[2], Miao Xu[2], Weihua Hu[2], Ivor Tsang[1], and Masashi Sugiyama[2,4]

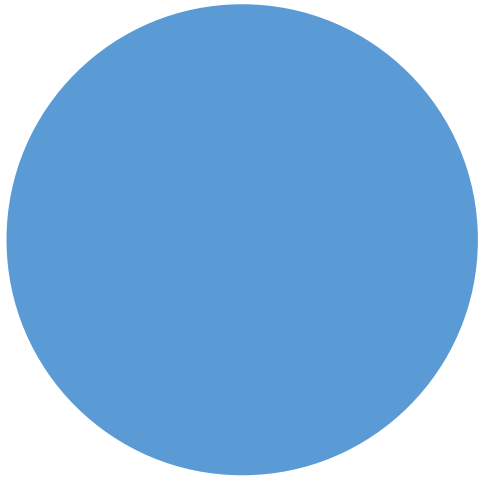
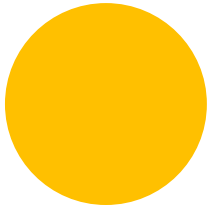
[1]Center for Artificial Intelligence, University of Technology Sydney, Australia

[2]Center for Advanced Intelligence Project, RIKEN, Japan

[3]4Paradigm Inc., Beijing, China

[4]Graduate School of Frontier Sciences, University of Tokyo, Japan

Overview

Introduction

Motivations

Many data collection processes yield noisy labels

- Ground Truth Labels vs. Noisy Labels

DNNs have a high capacity to overfit to noisy labels

- High number of parameters allows for many degrees of freedom in the model
- DNNs tend to learn simple patterns first, but fit to noise as the number of epochs grows

Current Approaches

The paper identifies two main approaches to training under noisy labels:

1

Estimating the noise transition matrix and integrating it into the model

2

Training on selected samples (selecting ‘clean’ instances from the noisy ones)

Contributions

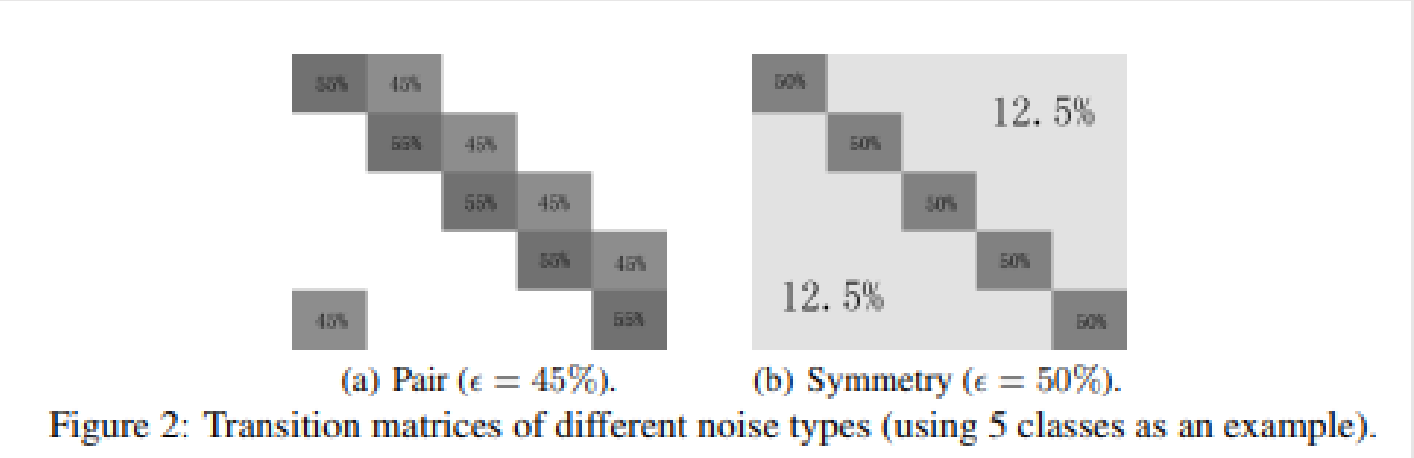
- Novel approach named ‘co-teaching’
  - Performs well under extreme label noise
  - Tested on noisy versions of MNIST, CIFAR-10, and CIFAR-100
  - Flexible: no need to combine with specific network architectures
  - Doesn’t require pre-training

Methodology

Label Corruption

Manual corruption of the datasets is performed using two types of noise transformation matrices:

1. Pair Flipping – Flipping that simulates fine-grained noise
2. Symmetry Flipping – All labels have equal probability of being flipped



(a) Pair ($\epsilon = 45\%$). (b) Symmetry ($\epsilon = 50\%$).

Figure 2: Transition matrices of different noise types (using 5 classes as an example).

Label Corruption (Cont'd)

Method	Noise Rate	Rationale
Pair Flipping	45%	Almost half of instances have noisy labels. Simulates erroneous labels which are similar to true labels.
Symmetry	50%	Half of instances have noisy labels.
Symmetry	20%	Verify the robustness of co-teaching in a low-level noise scenario.

Co-Teaching Method

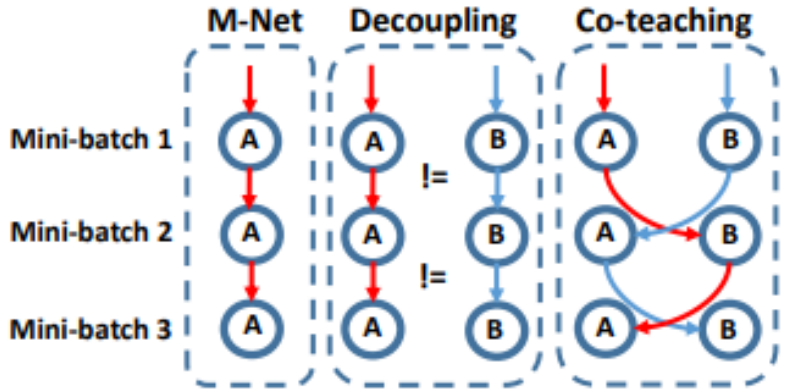


Figure 1: Comparison of error flow among MentorNet (M-Net) [17], Decoupling [26] and Co-teaching. Assume that the error flow comes from the biased selection of training instances, and error flow from network A or B is denoted by red arrows or blue arrows, respectively. **Left panel:** M-Net maintains only one network (A). **Middle panel:** Decoupling maintains two networks (A & B). The parameters of two networks are updated, when the predictions of them disagree (!=). **Right panel:** Co-teaching maintains two networks (A & B) simultaneously. In each mini-batch data, each network samples its small-loss instances as the useful knowledge, and teaches such useful instances to its peer network for the further training. Thus, the error flow in Co-teaching displays the zigzag shape.

Co-Teaching Method (Cont'd)

Algorithm 1 Co-teaching Algorithm.

1: **Input** w_f and w_g, learning rate η, fixed τ, epoch T_k and $T_{\max}$, iteration $N_{\max}$;
for $T = 1, 2, \ldots, T_{\max}$ **do**
  2: **Shuffle** training set $\mathcal{D}$; //noisy dataset
  for $N = 1, \ldots, N_{\max}$ **do**
    3: **Fetch** mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
    4: **Obtain** $\bar{\mathcal{D}}_f = \arg\min_{\mathcal{D}':|\mathcal{D}'| \geq R(T)|\bar{\mathcal{D}}|} \ell(f, \mathcal{D}')$; //sample $R(T)\%$ small-loss instances
    5: **Obtain** $\bar{\mathcal{D}}_g = \arg\min_{\mathcal{D}':|\mathcal{D}'| \geq R(T)|\bar{\mathcal{D}}|} \ell(g, \mathcal{D}')$; //sample $R(T)\%$ small-loss instances
    6: **Update** $w_f = w_f - \eta \nabla \ell(f, \bar{\mathcal{D}}_g)$; //update w_f by $\bar{\mathcal{D}}_g$;
    7: **Update** $w_g = w_g - \eta \nabla \ell(g, \bar{\mathcal{D}}_f)$; //update w_g by $\bar{\mathcal{D}}_f$;
  end
  8: **Update** $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
end
9: **Output** w_f **and** w_g.

Notes:

$\tau = 1 - \epsilon^*$

$T_k = 10^*$

* Updated in most recent revision

Baseline Comparisons

	Bootstrap	S-model	F-correction	Decoupling	MentorNet	Co-teaching
large class	✗	✗	✗	✓	✓	✓
heavy noise	✗	✗	✗	✗	✓	✓
flexibility	✗	✗	✓	✓	✓	✓
no pre-train	✓	✗	✗	✗	✓	✓

Criteria	Description
Large Class	Can deal with a large number of classes
Heavy Noise	Can combat heavy noise (i.e., high noise ratio)
Flexibility	Need not combine with specific network architecture
No Pre-Train	Can be trained from scratch

Implementation

- Implemented with default parameters in PyTorch
- 9 convolutional layers with max pooling and dropout
- Adam optimizer
  - Momentum = 0.9
- Learning rate = 0.001
- Batch size = 128
- Epochs = 200

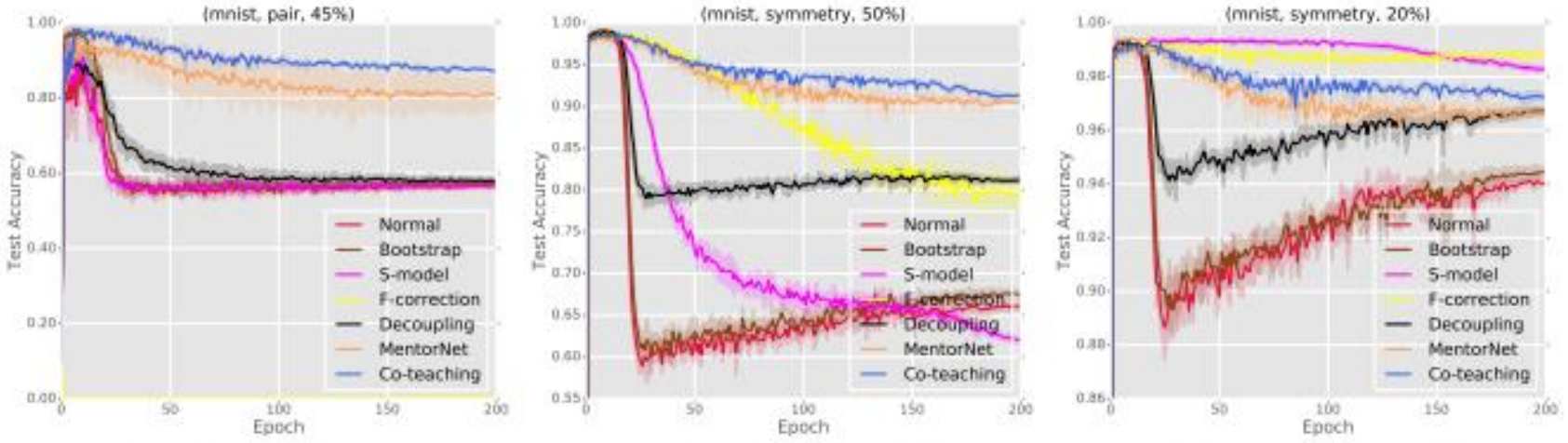
CNN on *MNIST*	CNN on *CIFAR-10*	CNN on *CIFAR-100*
28×28 Gray Image	32×32 RGB Image	32×32 RGB Image
3×3 conv, 128 LReLU 3×3 conv, 128 LReLU 3×3 conv, 128 LReLU		
2×2 max-pool, stride 2 dropout, $p = 0.25$		
3×3 conv, 256 LReLU 3×3 conv, 256 LReLU 3×3 conv, 256 LReLU		
2×2 max-pool, stride 2 dropout, $p = 0.25$		
3×3 conv, 512 LReLU 3×3 conv, 256 LReLU 3×3 conv, 128 LReLU		
avg-pool		
dense 128→10	dense 128→10	dense 128→100

Results

MNIST

Table 4: Average test accuracy on *MNIST* over the last ten epochs.

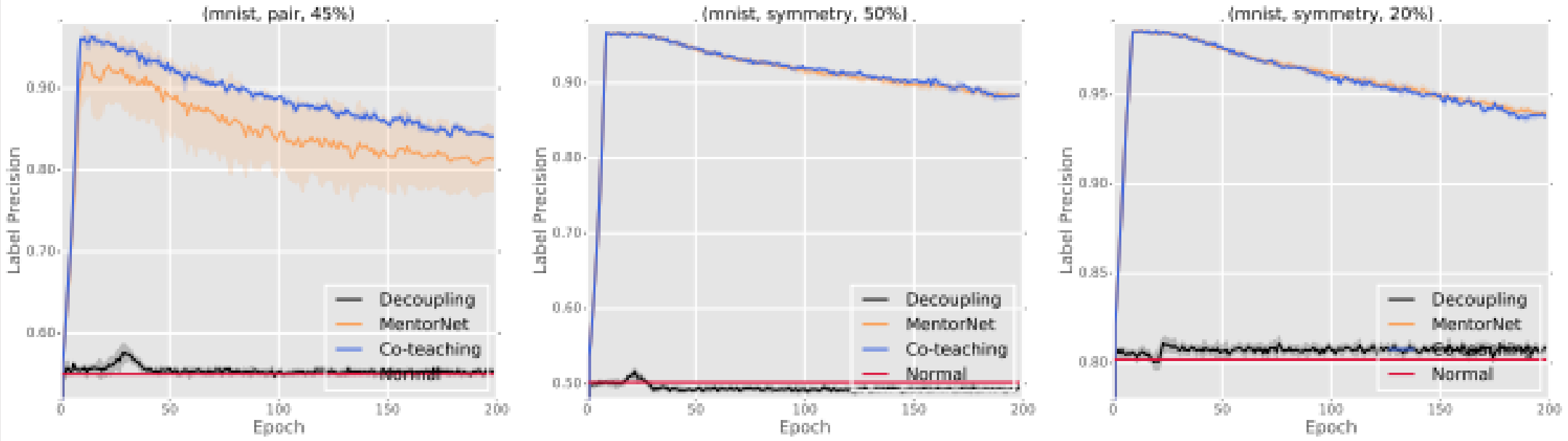
Flipping-Rate	Standard	Bootstrap	S-model	F-correction	Decoupling	MentorNet	Co-teaching
Pair-45%	56.52% ±0.55%	57.23% ±0.73%	56.88% ±0.32%	0.24% ±0.03%	58.03% ±0.07%	80.88% ±4.45%	**87.63%** ±0.21%
Symmetry-50%	66.05% ±0.61%	67.55% ±0.53%	62.29% ±0.46%	79.61% ±1.96%	81.15% ±0.03%	90.05% ±0.30%	**91.32%** ±0.06%
Symmetry-20%	94.05% ±0.16%	94.40% ±0.26%	98.31% ±0.11%	**98.80%** ±0.12%	95.70% ±0.02%	96.70% ±0.22%	97.25% ±0.03%



(a) pair, 45%. (b) symmetry, 50%. (c) symmetry, 20%.

Fig. 3. Test accuracy vs number of epochs on *MNIST* dataset.

MNIST (Cont'd)



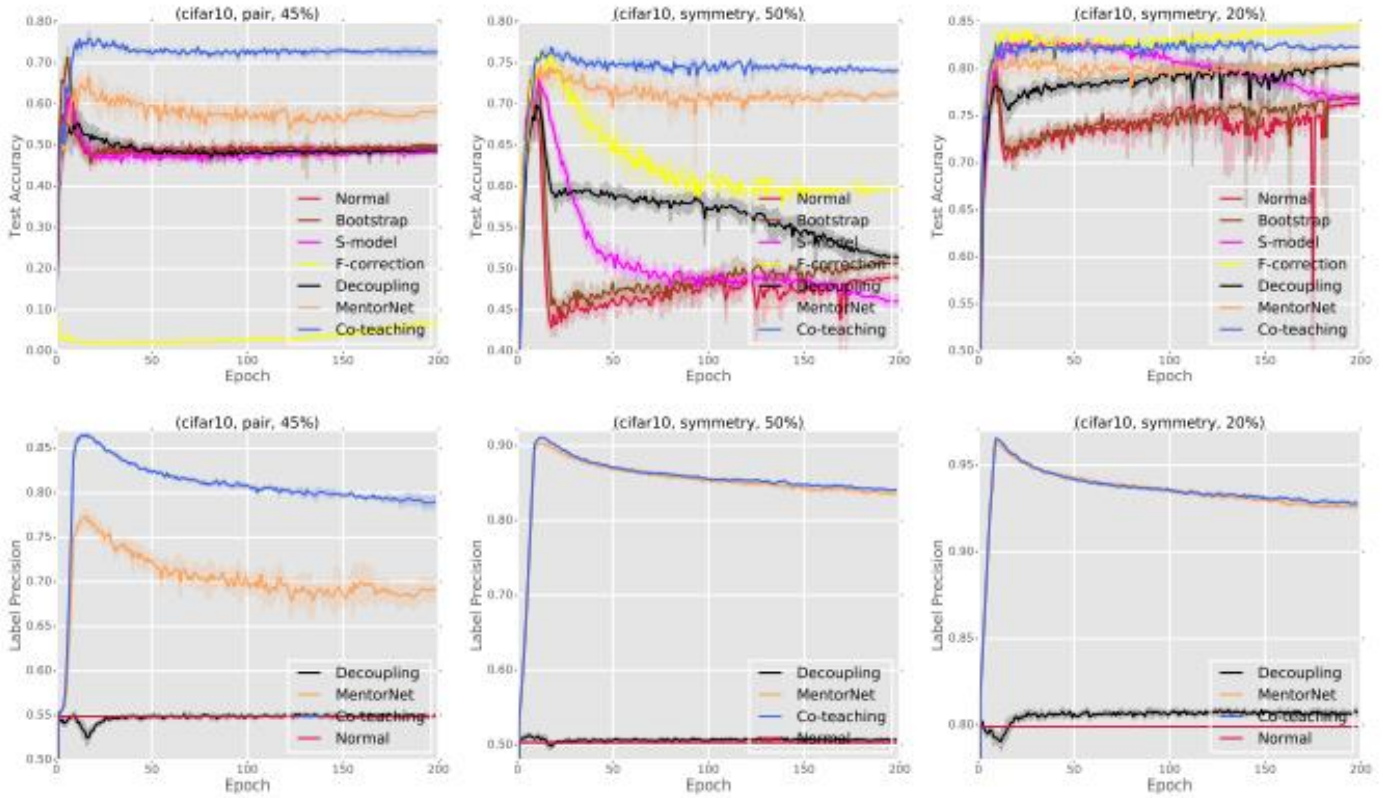
(a) pair, 45%. (b) symmetry, 50%. (c) symmetry, 20%.

Fig. 4. Label precision vs number of epochs on *MNIST* dataset.

CIFAR-10

Table 5: Average test accuracy on *CIFAR-10* over the last ten epochs.

Flipping,Rate	Standard	Bootstrap	S-model	F-correction	Decoupling	MentorNet	Co-teaching
Pair-45%	49.50% ±0.42%	50.05% ±0.30%	48.21% ±0.55%	6.61% ±1.12%	48.80% ±0.04%	58.14% ±0.38%	**72.62%** ±0.15%
Symmetry-50%	48.87% ±0.52%	50.66% ±0.56%	46.15% ±0.76%	59.83% ±0.17%	51.49% ±0.08%	71.10% ±0.48%	**74.02%** ±0.04%
Symmetry-20%	76.25% ±0.28%	77.01% ±0.29%	76.84% ±0.66%	**84.55%** ±0.16%	80.44% ±0.05%	80.76% ±0.36%	82.32% ±0.07%



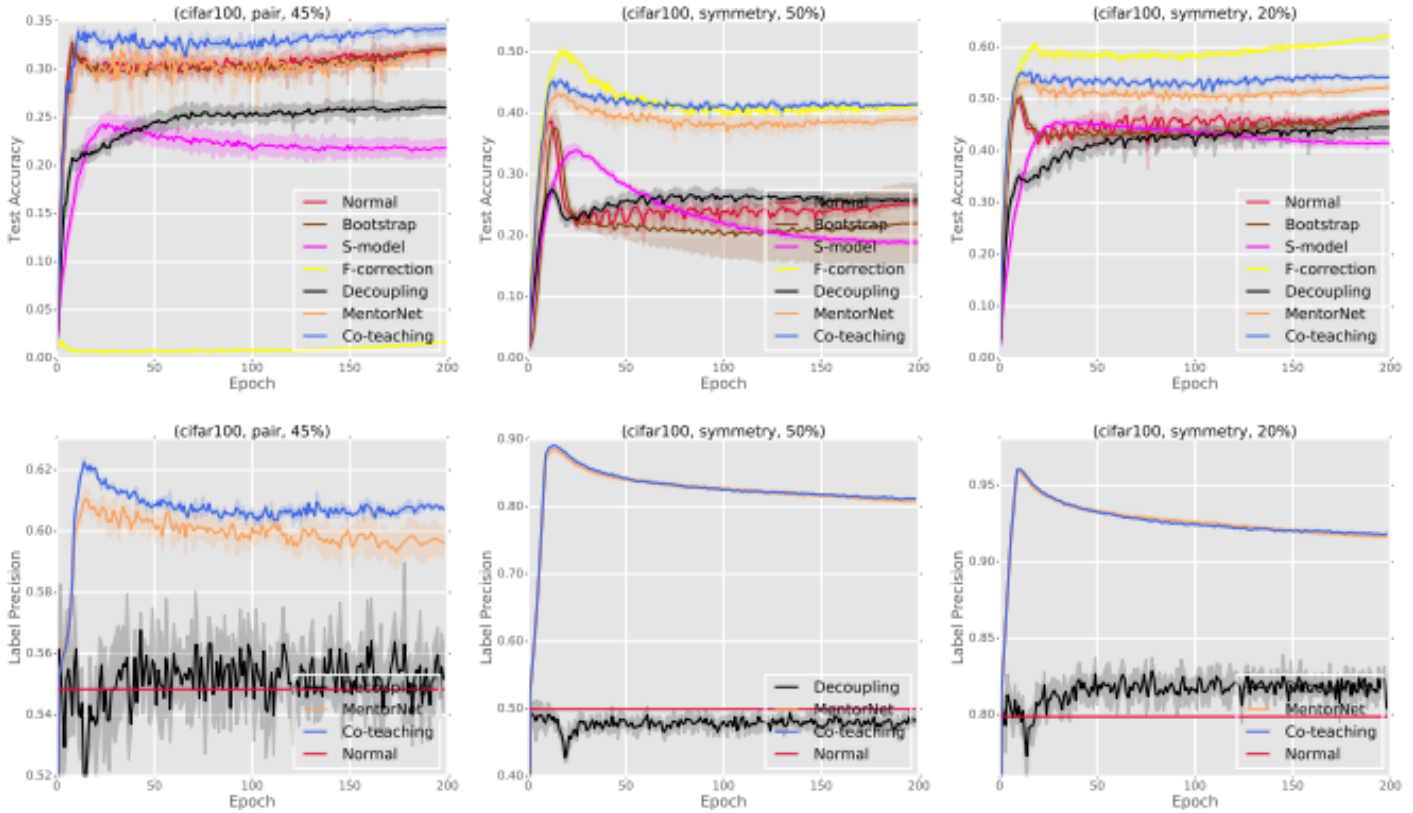
(a) pair, 45%. (b) symmetry, 50%. (c) symmetry, 20%.

Fig. 5. Results on *CIFAR10* dataset. Top: test accuracy vs number of epochs; bottom: label precision vs number of epochs.

CIFAR-100

Table 6: Average test accuracy on *CIFAR-100* over the last ten epochs.

Flipping,Rate	Standard	Bootstrap	S-model	F-correction	Decoupling	MentorNet	Co-teaching
Pair-45%	31.99% ±0.64%	32.07% ±0.30%	21.79% ±0.86%	1.60% ±0.04%	26.05% ±0.03%	31.60% ±0.51%	**34.81%** ±0.07%
Symmetry-50%	25.21% ±0.64%	21.98% ±6.36%	18.93% ±0.39%	41.04% ±0.07%	25.80% ±0.04%	39.00% ±1.00%	**41.37%** ±0.08%
Symmetry-20%	47.55% ±0.47%	47.00% ±0.54%	41.51% ±0.60%	**61.87%** ±0.21%	44.52% ±0.04%	52.13% ±0.40%	54.23% ±0.08%



(a) pair, 45%. (b) symmetry, 50%. (c) symmetry, 20%.

Fig. 6. Results on *CIFAR100* dataset. Top: test accuracy vs number of epochs; bottom: label precision vs number of epochs.

Discussion

Conclusions

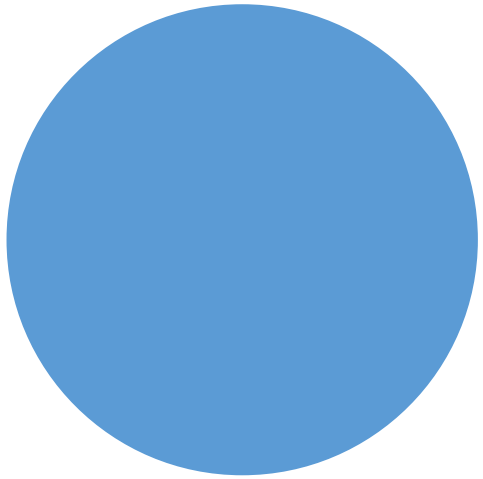
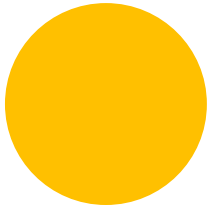
The co-teaching method outperforms baseline methods in terms of accuracy.

Co-teaching still performs competitively in the low-noise (usually 2nd or 3rd best accuracy)

Similar to MentorNet, Co-teaching greatly improves the label precision (training labels more closely match ground truth)

Critiques

- Lack of task diversity
- Lack of theoretical development
- Only two techniques for handling heavy noise are compared
- Arbitrary selection of τ, ϵ, and T_k values (addressed in latest revision)
- Data corruption scenarios (selection of method and ϵ) also arbitrary

Thanks!

Any questions?