Training strategies for noisy labels
Literature review

Dominik Lewy

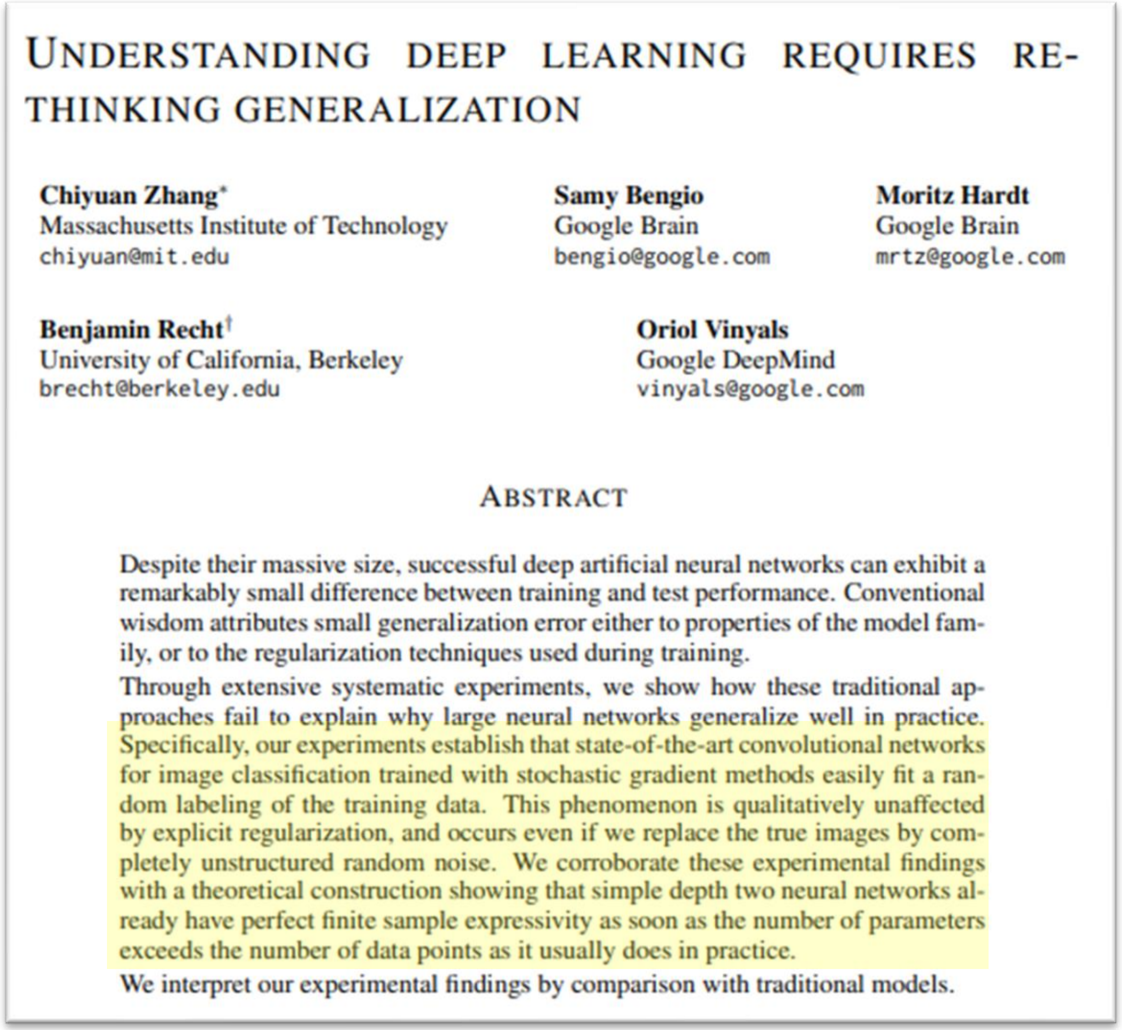
Introduction

Problem statement

Deep Neural Networks suffer from memorization effect.

Memorization effect – learning random not meaningful patterns. The result is like lookup table where we store patterns without deeper understanding of the whole concept.

- DNNs offer good generalization (it is not that well understood; the property is applied either to model family or regularization techniques used in training)
- DNNs can easily fit a random labeling of the training data (unaffected by regularization; if the number of weights surpasses the number of data points, which usually is the case in modern architectures)



UNDERSTANDING DEEP LEARNING REQUIRES RE-THINKING GENERALIZATION

Chiyuan Zhang* Massachusetts Institute of Technology chiyuan@mit.edu

Samy Bengio Google Brain bengio@google.com

Moritz Hardt Google Brain mrtz@google.com

Benjamin Recht† University of California, Berkeley brecht@berkeley.edu

Oriol Vinyals Google DeepMind vinyals@google.com

ABSTRACT

Despite their massive size, successful deep artificial neural networks can exhibit a remarkably small difference between training and test performance. Conventional wisdom attributes small generalization error either to properties of the model family, or to the regularization techniques used during training.

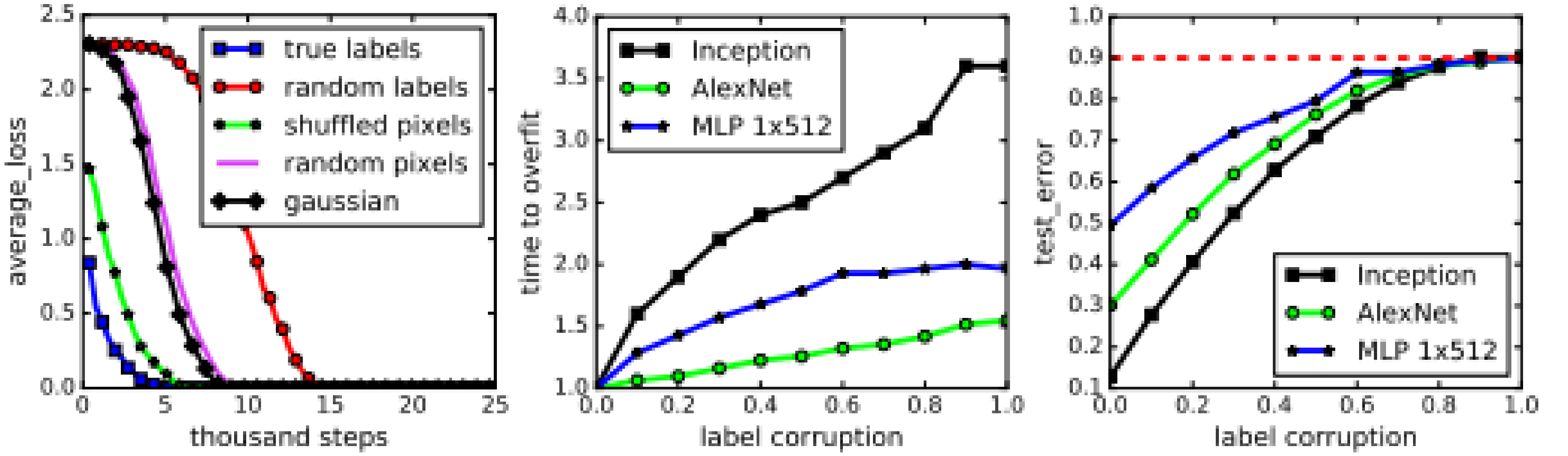
Through extensive systematic experiments, we show how these traditional approaches fail to explain why large neural networks generalize well in practice. Specifically, our experiments establish that state-of-the-art convolutional networks for image classification trained with stochastic gradient methods easily fit a random labeling of the training data. This phenomenon is qualitatively unaffected by explicit regularization, and occurs even if we replace the true images by completely unstructured random noise. We corroborate these experimental findings with a theoretical construction showing that simple depth two neural networks already have perfect finite sample expressivity as soon as the number of parameters exceeds the number of data points as it usually does in practice.

We interpret our experimental findings by comparison with traditional models.

https://arxiv.org/pdf/1611.03530.pdf

February 2017

Problem statement – DNNs can easily fit a random labeling



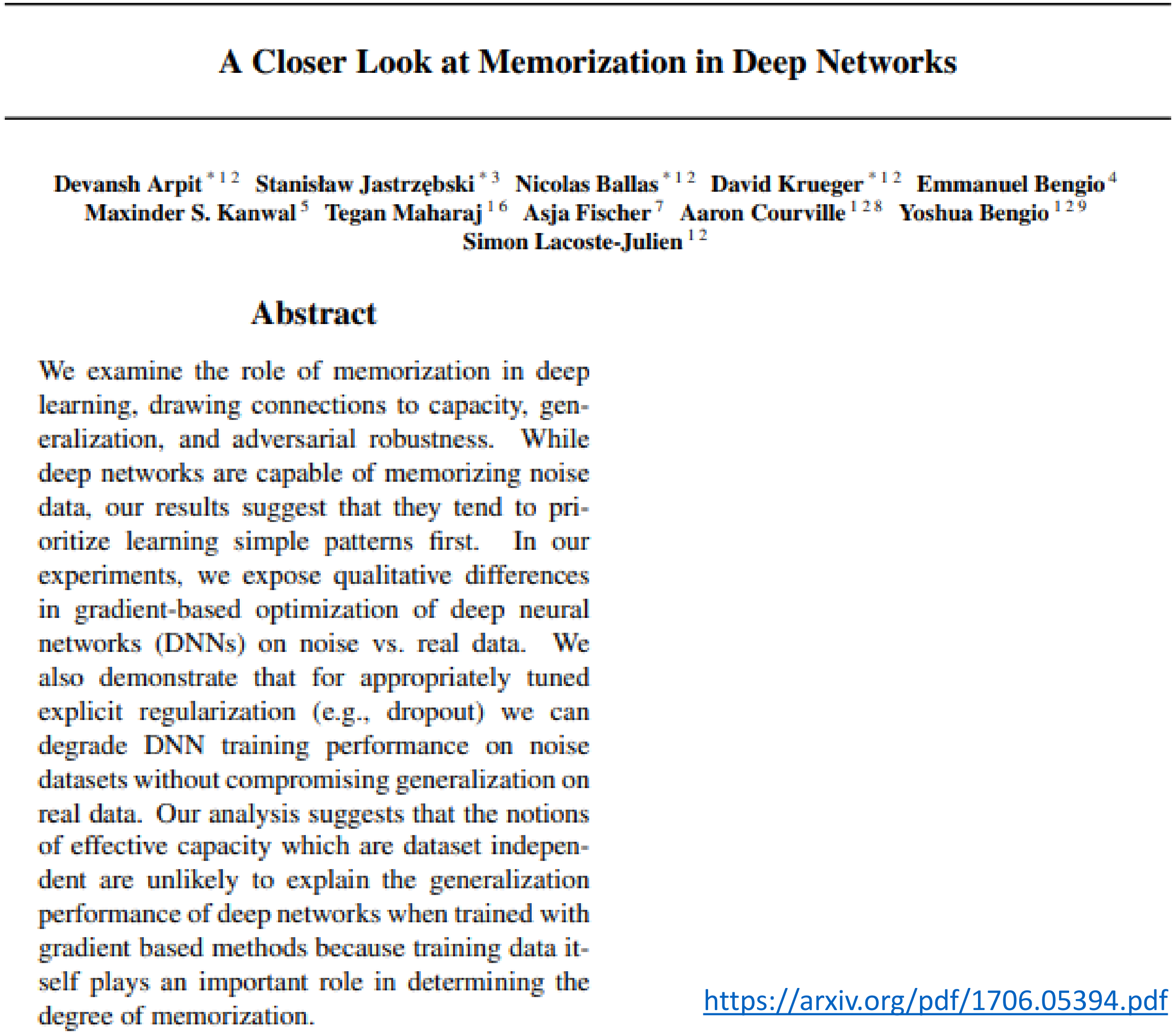
(a) learning curves (b) convergence slowdown (c) generalization error growth

Figure 1: Fitting random labels and random pixels on CIFAR10. (a) shows the training loss of various experiment settings decaying with the training steps. (b) shows the relative convergence time with different label corruption ratio. (c) shows the test error (also the generalization error since training error is 0) under different label corruptions.

Intuition

Content of the paper:

1. **Qualitative differences in training networks on random vs real data**
2. DNNs learn simple patterns first
3. How to use regularization to reduce memorization effect



A Closer Look at Memorization in Deep Networks

Devansh Arpit [*1 2] Stanisław Jastrzębski [*3] Nicolas Ballas [*1 2] David Krueger [*1 2] Emmanuel Bengio [4] Maxinder S. Kanwal [5] Tegan Maharaj [1 6] Asja Fischer [7] Aaron Courville [1 2 8] Yoshua Bengio [1 2 9] Simon Lacoste-Julien [1 2]

Abstract

We examine the role of memorization in deep learning, drawing connections to capacity, generalization, and adversarial robustness. While deep networks are capable of memorizing noise data, our results suggest that they tend to prioritize learning simple patterns first. In our experiments, we expose qualitative differences in gradient-based optimization of deep neural networks (DNNs) on noise vs. real data. We also demonstrate that for appropriately tuned explicit regularization (e.g., dropout) we can degrade DNN training performance on noise datasets without compromising generalization on real data. Our analysis suggests that the notions of effective capacity which are dataset independent are unlikely to explain the generalization performance of deep networks when trained with gradient based methods because training data itself plays an important role in determining the degree of memorization.

https://arxiv.org/pdf/1706.05394.pdf

Intuition – Qualitative differences in training networks on random vs real data

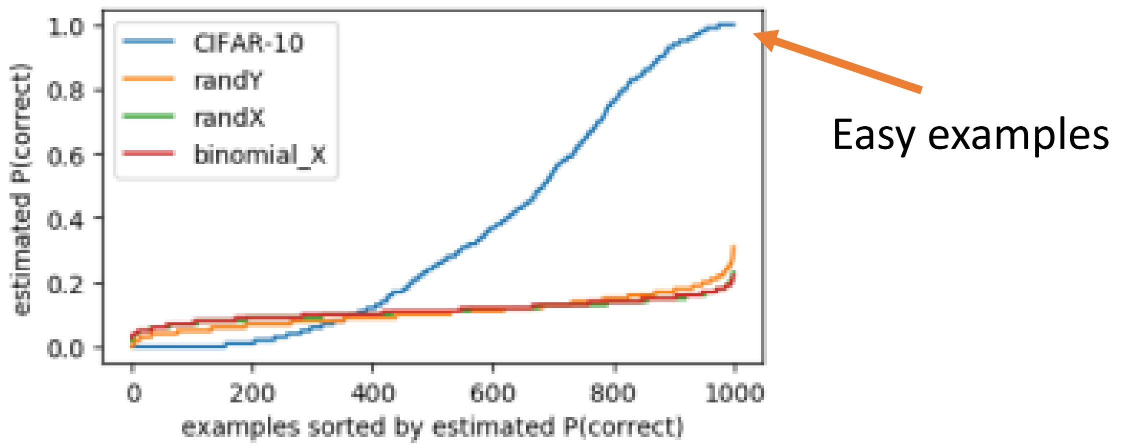
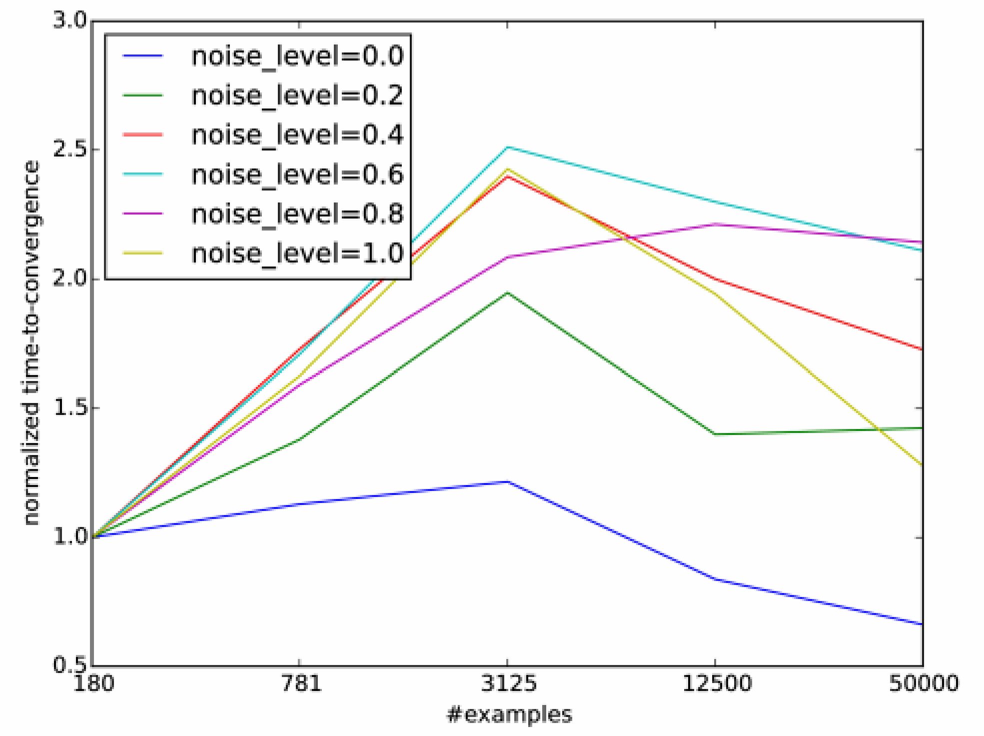


Figure 1. Average (over 100 experiments) misclassification rate for each of 1000 examples after one epoch of training. This measure of an example's difficulty is much more variable in real data. We conjecture this is because the easier examples are explained by some simple patterns, which are reliably learned within the first epoch of training. We include 1000 points samples from a binomial distribution with $n = 100$ and p equal to the average estimated P(correct) for randX, and note that this curve closely resembles the randX curve, suggesting that random inputs are all equally difficult.



Problem statement

Noisy labels

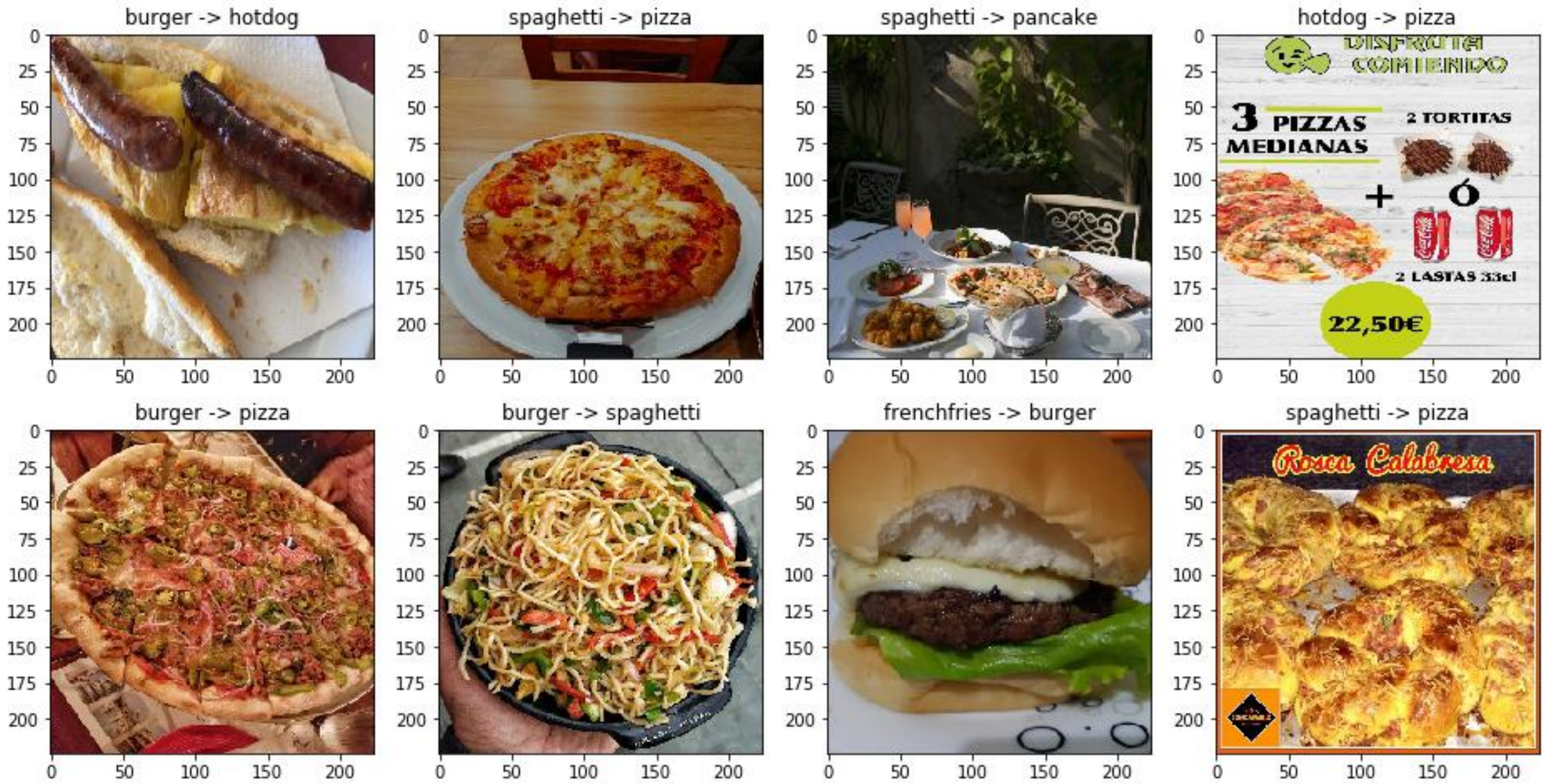
Other approaches of handling noisy data

Other approaches of handling noisy data

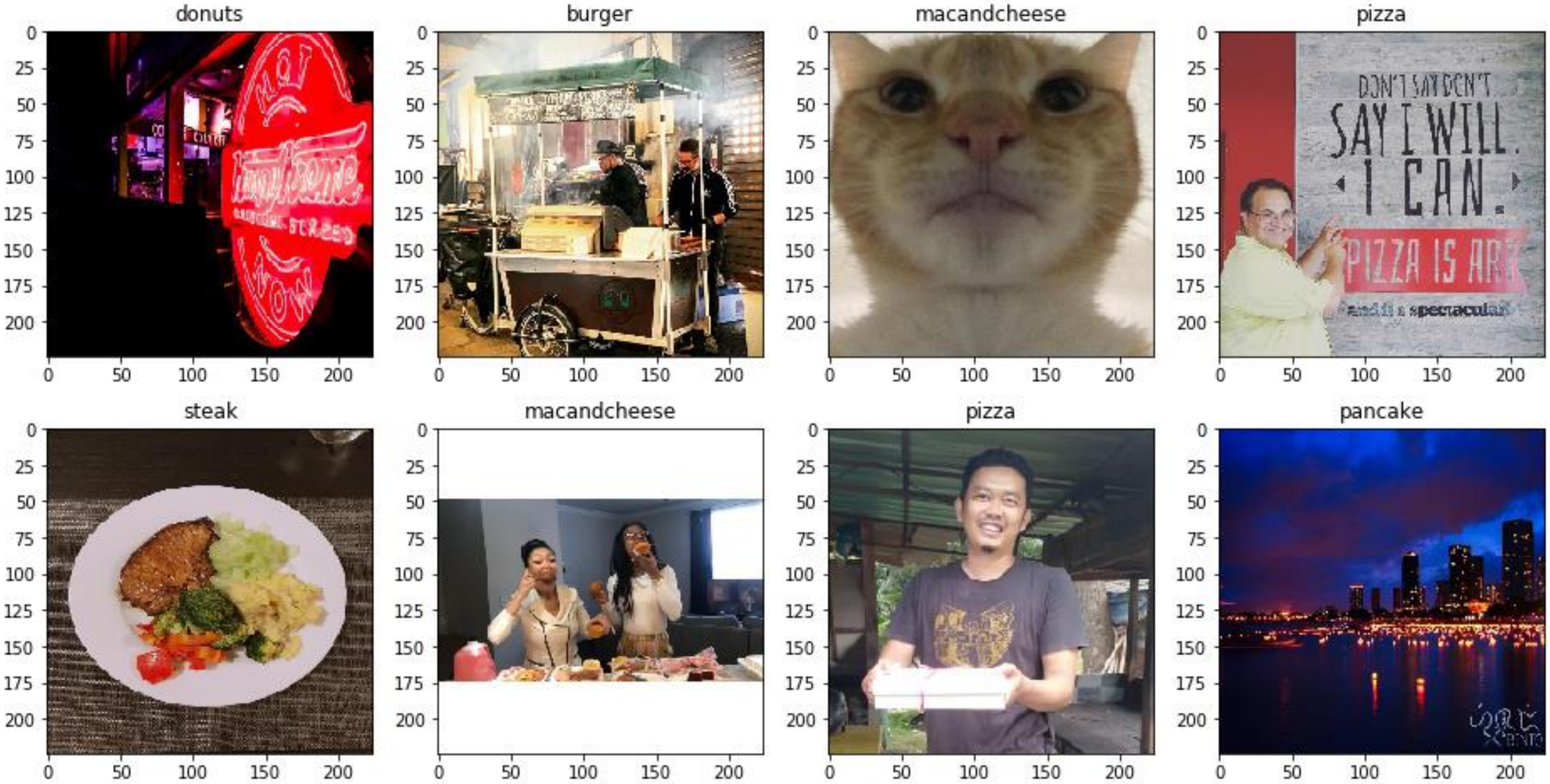
- Preprocessing steps (eliminating observations that are suspected of being mislabeled)
- Buckets of examples (predicting labels for groups of examples rather for single observation)
- Loss function change (adding a regularization term to the loss function)
- Adding layers that mimic noisy behavior (estimating a conditional probability of seeing a wrong label)
- Regularization (adding dropout to the network)

Key concepts

Small-loss samples



Big-loss samples



Update by disagreement

Original image:	**True class:**		**Net 1:**		**Net 2:**
	Boat	=	Boat	=	Boat
	Car	≠	Bike	=	Bike

Decoupling

Decoupling – key concepts

Key concept: decouple the decision of "when to update" from the decision of "how to update".

"when to update" – when 2 classifiers give different predictions (when classifiers "disagree"). This decision is independent of the "true" label.

Details:

- Algorithm uses 2 DNNs (this could be seen as a meta algorithm that decides on which observations should be used for learning).
- The difference stems from random initialization (This is crucial. If we were to initialize both networks in the same way the algorithm would not make any updates).

Pseudo code:

Algorithm 1 Update by Disagreement

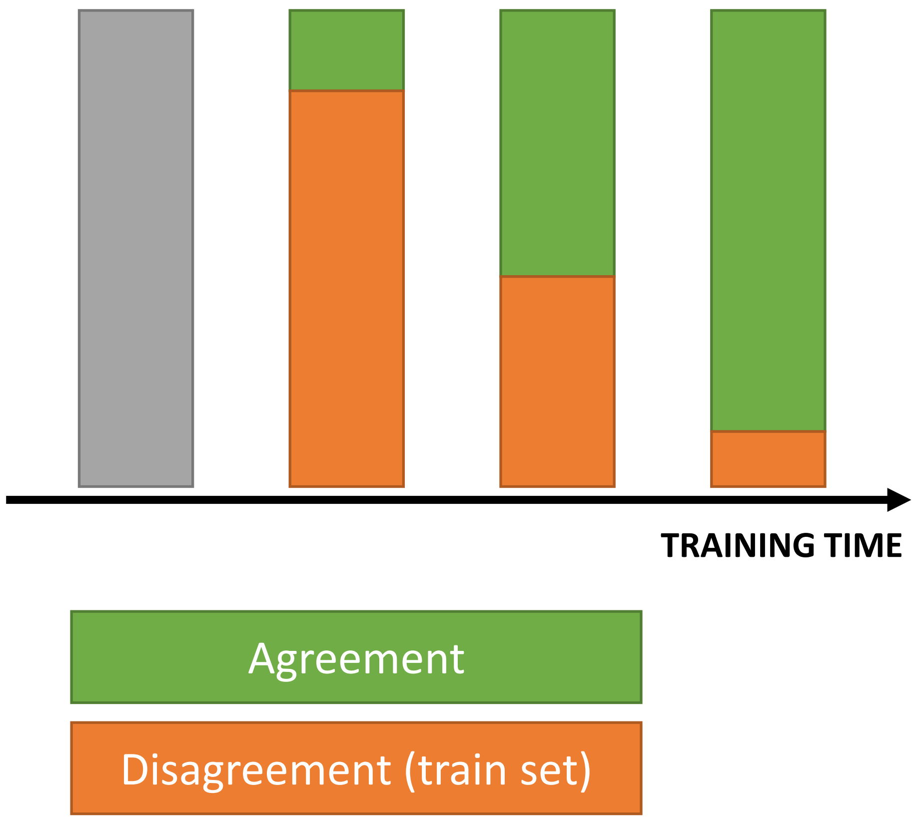
input:
  an update rule U
  batch size b
  two initial predictors $h_1, h_2 \in \mathcal{H}$
for $t = 1, 2, \ldots, N$ **do**
  draw mini-batch $(x_1, y_1), \ldots, (x_b, y_b) \sim \tilde{\mathcal{D}}^b$
  let $S = \{(x_i, y_i) : h_1(x_i) \neq h_2(x_i)\}$
  $h_1 \leftarrow U(h_1, S)$
  $h_2 \leftarrow U(h_2, S)$
end for

Decoupling – practice & intuition

Practice: the procedure suggested by authors is as follows:

1. Initially training each of the two classifiers on a different subset of the data
2. Switching to the suggested update rule in an advanced stage of the training process
3. At the end of the optimization process each of the two classifiers can be used for inference

Training set over training iterations:



Decoupling – experiments

Data set: Labeled Faces in the Wild (LFW). This benchmark consists of 13,233 images of 5,749 different people collected from the web, labeled with the name of the person in the picture. The authors reformulated the problem by using an external algorithm to predict (with some uncertainty) the gender of the person based on the name. This resulted in noisy labels.

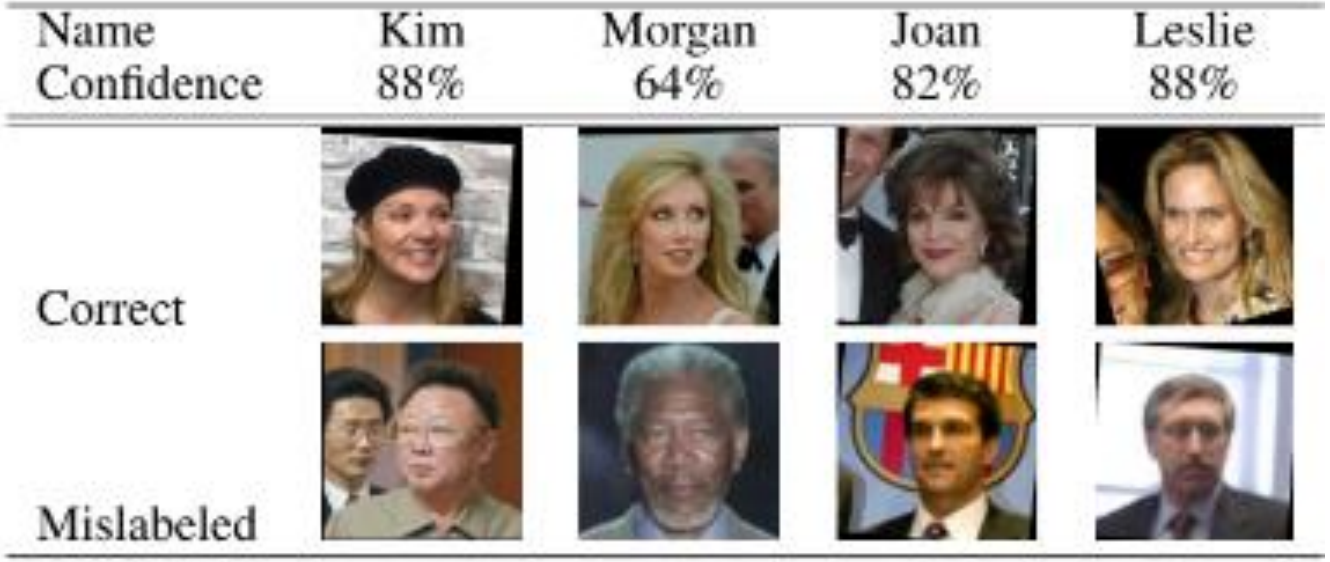
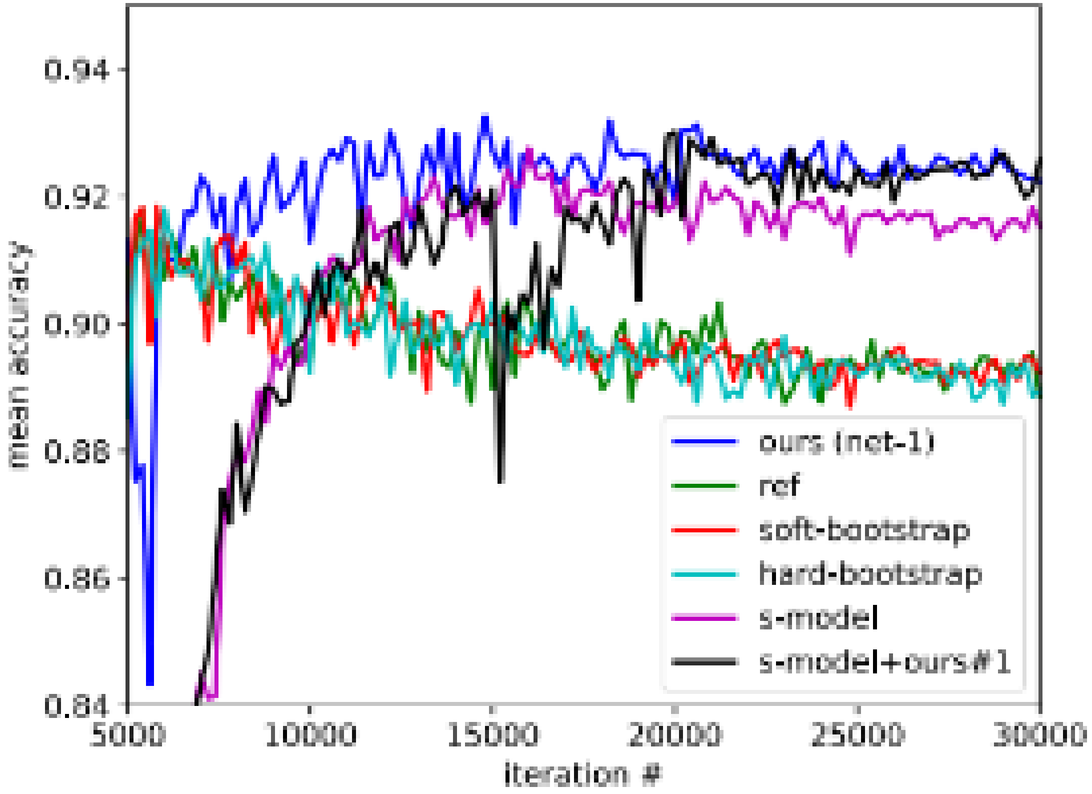


Figure 1: Images from the dataset tagged as female

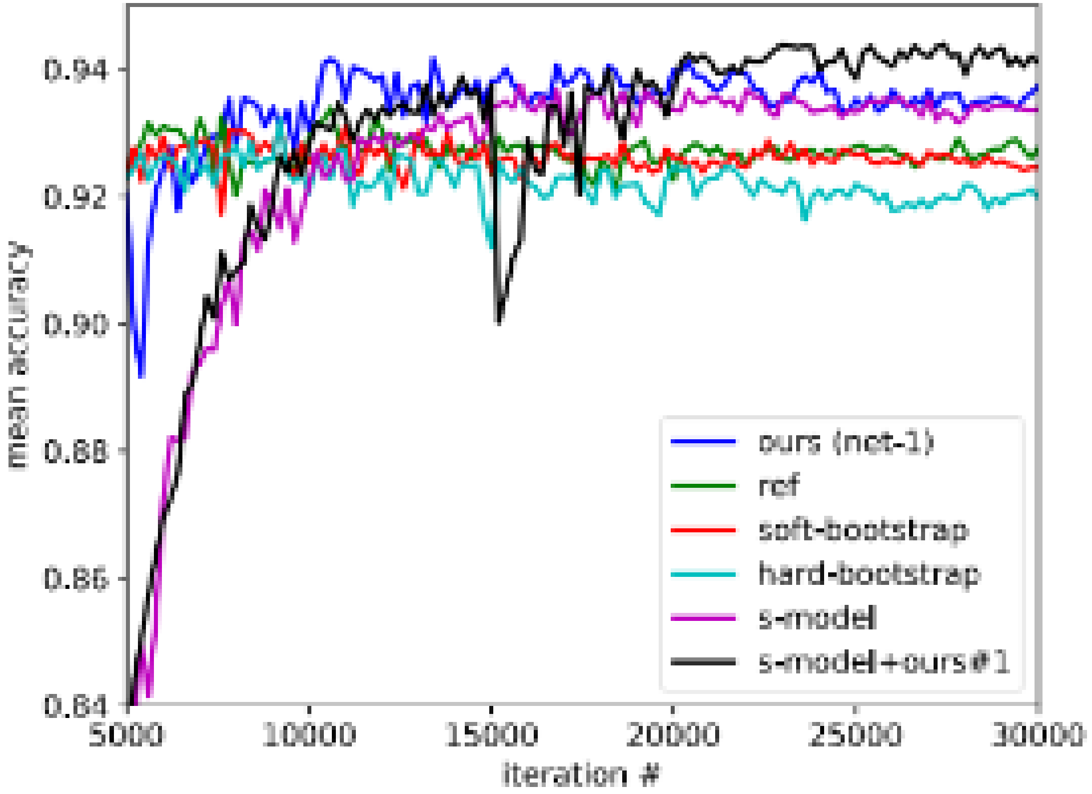
The authors created 5 subsets based on the data set above:

- N_1, N_2, N_3- all the images for which the algorithm was 100% sure about the gender (divided into 3 equal parts)
- N_4- the images where the algorithm was more than 90% sure
- N_5- the algorithm did not provide prediction. All those images were labeled as male hence majority of the images in the data set were of males.

Decoupling – experiments



Dataset #1 - more noise (N$_1$-N$_5$)



Dataset #2 - less noise (N$_1$-N$_4$)

Small loss | Cross update | Joint training | Disagreement | Agreement

Decoupling – experiments

Scenarios: Two alternative scenarios were considered:

- The clean data set was available for model selection (in this case the observed value is the balanced accuracy on the best available iteration)
- The clean data is not available (in this case the observed value is the balanced accuracy of the last iteration)

Dataset #1	Accuracy (best iteration)			Accuracy (last iteration)		
	Male	Female	**Mean**	Male	Female	**Mean**
ours (net #1)	94.4 ± 0.7	92.7 ± 0.2	**93.6 ± 0.2**	94.8 ± 0.8	89.7 ± 1.3	**92.2 ± 0.6**
ours (net #2)	93.5 ± 1.1	93.2 ± 0.6	93.4 ± 0.3	93.7 ± 0.8	90.1 ± 0.9	91.9 ± 0.4
s-model+ours #1	93.3 ± 1.7	93.8 ± 1.4	**93.6 ± 0.4**	93.7 ± 1.1	91.4 ± 1.0	**92.6 ± 0.1**
s-model+ours #2	94.2 ± 0.7	91.7 ± 0.6	93.0 ± 0.2	93.6 ± 1.3	91.6 ± 1.5	92.6 ± 0.1
baseline	91.6 ± 2.2	92.7 ± 1.8	92.2 ± 0.2	94.5 ± 0.7	83.3 ± 3.2	88.9 ± 1.3
bootstrap-soft	92.5 ± 0.6	91.9 ± 0.6	92.2 ± 0.2	94.5 ± 0.7	84.0 ± 1.7	89.2 ± 0.8
bootstrap-hard	92.4 ± 0.7	91.9 ± 1.0	92.1 ± 0.3	94.7 ± 0.2	83.2 ± 1.7	88.9 ± 0.8
s-model	94.5 ± 0.7	91.3 ± 0.4	92.9 ± 0.5	93.3 ± 2.0	89.8 ± 1.3	91.5 ± 0.4

Dataset #2	Accuracy (best iteration)			Accuracy (last iteration)		
	Male	Female	**Mean**	Male	Female	**Mean**
ours (net #1)	95.5 ± 0.8	93.6 ± 0.9	94.5 ± 0.2	95.4 ± 1.1	92.1 ± 0.7	93.7 ± 0.2
ours (net #2)	95.7 ± 1.5	93.0 ± 1.8	94.4 ± 0.2	95.9 ± 0.6	91.6 ± 0.6	93.7 ± 0.3
s-model+ours #1	95.5 ± 0.5	94.0 ± 0.7	**94.8 ± 0.2**	95.3 ± 1.3	92.9 ± 2.2	**94.1 ± 0.4**
s-model+ours #2	95.1 ± 0.8	93.9 ± 1.5	94.5 ± 0.3	95.6 ± 1.2	92.5 ± 1.7	94.0 ± 0.2
baseline	93.6 ± 0.7	93.9 ± 0.8	93.8 ± 0.3	96.2 ± 0.2	89.4 ± 1.6	92.8 ± 0.8
bootstrap-soft	94.8 ± 1.0	92.2 ± 0.6	93.5 ± 0.4	96.2 ± 0.6	88.7 ± 2.0	92.5 ± 0.7
bootstrap-hard	93.9 ± 1.2	92.8 ± 0.7	93.4 ± 0.4	96.1 ± 0.3	87.9 ± 1.6	92.0 ± 0.6
s-model	94.8 ± 1.0	93.3 ± 0.4	94.1 ± 0.3	94.5 ± 0.6	92.3 ± 0.2	93.4 ± 0.4

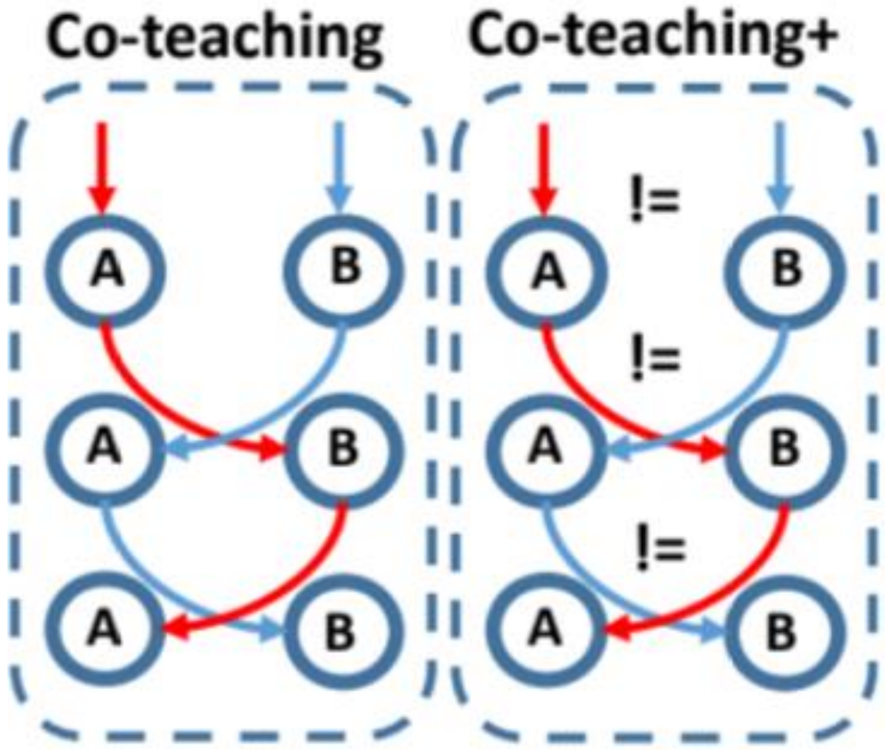
Co-teaching+

Co-teaching+ – key concepts

Key concept: combining the “decoupling” strategy with Co-teaching (based on small-loss trick).

Small-loss trick – using for training only those observations that produce low errors (are “easy” to classify)

High level steps:
- Both networks do prediction for all the data
- Only prediction disagreement is used further
- Both networks select small-loss data from the disagreement sample
- The selection of one network is back propagated through the other net



The motivation for introducing Co-teaching+ is that in the simple Co-teaching both networks gradually converge to a consensus reducing the benefit of two separate experts.

Small loss | Cross update | Joint training | Disagreement | Agreement

Co-teaching+ – key concepts

Mechanisms used:

- Small-loss trick
- Keeping the networks diverged (presented on the right)
- Cross-updating parameters of two networks (intuition comes from culture evolving hypothesis, where a human brain can learn better if guided by the signal produced by other humans)

Figure 1. Comparison of divergence (evaluated by Total Variation) between two networks trained by the "Disagreement" strategy, Co-teaching and Co-teaching+, respectively. Co-teaching+ naturally bridges the "Disagreement" strategy with Co-teaching.

Small loss | Cross update | Joint training | Disagreement | Agreement

Co-teaching+ – key concepts

Algorithm 1 Co-teaching+. Step 4: disagreement-update; Step 5-8: cross-update.

1: **Input** $w^{(1)}$ and $w^{(2)}$, training set $\mathcal{D}$, batch size B, learning rate η, estimated noise rate τ, epoch E_k and $E_{\max}$;
for $e = 1, 2, \ldots, E_{\max}$ **do**
    2: **Shuffle** $\mathcal{D}$ into $\frac{|\mathcal{D}|}{B}$ mini-batches; //noisy dataset
    for $n = 1, \ldots, \frac{|\mathcal{D}|}{B}$ **do**
        3: **Fetch** n-th mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
        4: **Select** prediction disagreement $\bar{\mathcal{D}}'$ by Eq. (1);
        5: **Get** $\bar{\mathcal{D}}'^{(1)} = \arg\min_{\mathcal{D}':|\mathcal{D}'| \geq \lambda(e)|\bar{\mathcal{D}}'|} \ell(\mathcal{D}'; w^{(1)})$; //sample $\lambda(e)\%$ small-loss instances
        6: **Get** $\bar{\mathcal{D}}'^{(2)} = \arg\min_{\mathcal{D}':|\mathcal{D}'| \geq \lambda(e)|\bar{\mathcal{D}}'|} \ell(\mathcal{D}'; w^{(2)})$; //sample $\lambda(e)\%$ small-loss instances
        7: **Update** $w^{(1)} = w^{(1)} - \eta \nabla \ell(\bar{\mathcal{D}}'^{(2)}; w^{(1)})$; //update $w^{(1)}$ by $\bar{\mathcal{D}}'^{(2)}$;
        8: **Update** $w^{(2)} = w^{(2)} - \eta \nabla \ell(\bar{\mathcal{D}}'^{(1)}; w^{(2)})$; //update $w^{(2)}$ by $\bar{\mathcal{D}}'^{(1)}$;
    end
    9: **Update** $\lambda(e) = 1 - \min\{\frac{e}{E_k}\tau, \tau\}$ or $1 - \min\{\frac{e}{E_k}\tau, (1 + \frac{e - E_k}{E_{\max} - E_k})\tau\}$;
end
10: **Output** $w^{(1)}$ and $w^{(2)}$.

9: **Update** $\lambda(e) = 1 - \min\{\frac{e}{E_k}\tau, \tau\}$ or $1 - \min\{\frac{e}{E_k}\tau, (1 + \frac{e - E_k}{E_{\max} - E_k})\tau\}$;

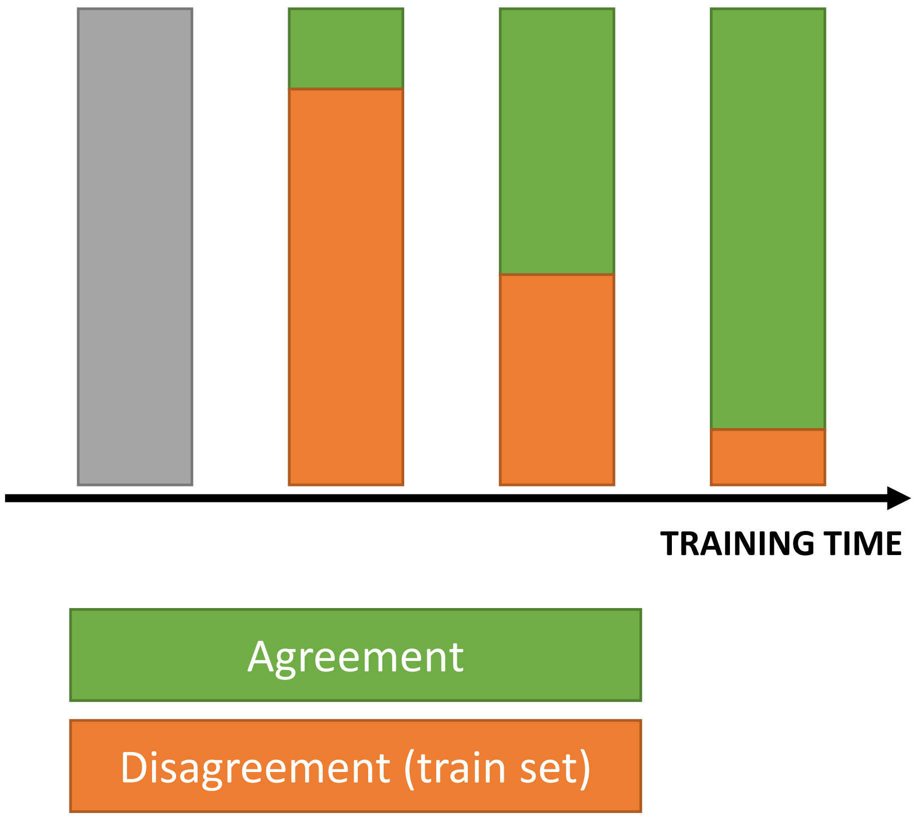
Controlling how many small-loss data should be selected:

- Beginning of the training procedure – we want to keep more small-loss data in each mini-batch, which is equivalent to dropping less data (we need a large $\lambda(e)$
- Advanced part of the training procedure – we want to keep less small-loss data in each mini-batch, which is equivalent to dropping more data (we need a small $\lambda(e)$

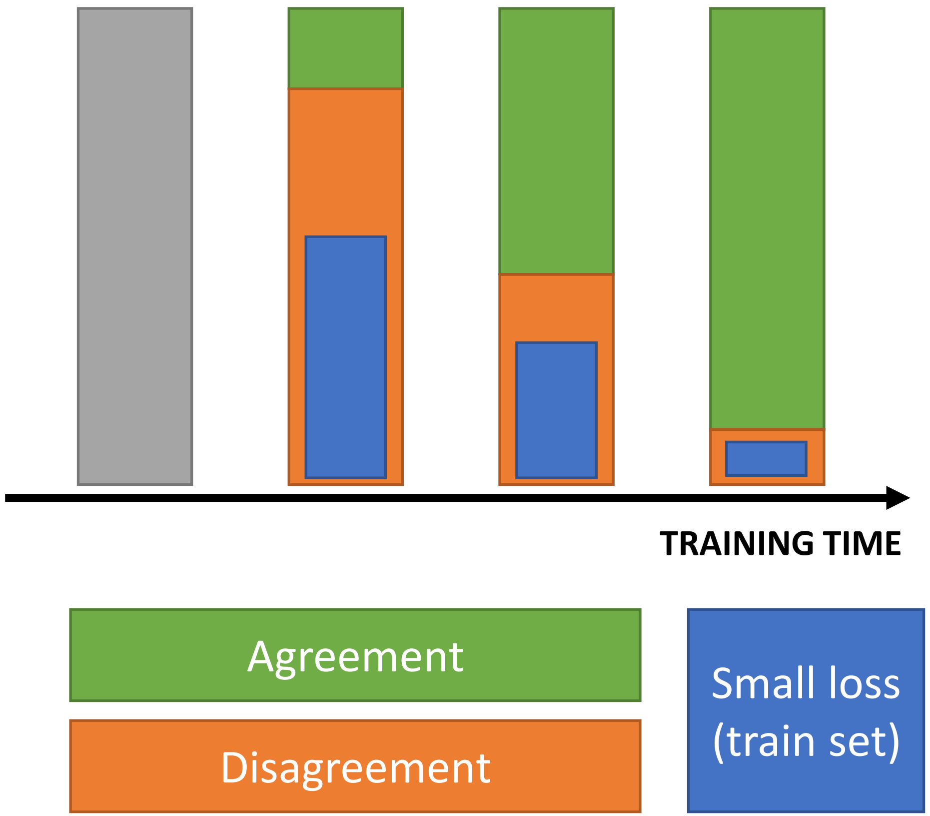
Small loss | Cross update | Joint training | Disagreement | Agreement

Co-teaching+ – training set over training iterations

Decoupling:



Co-teaching+ :

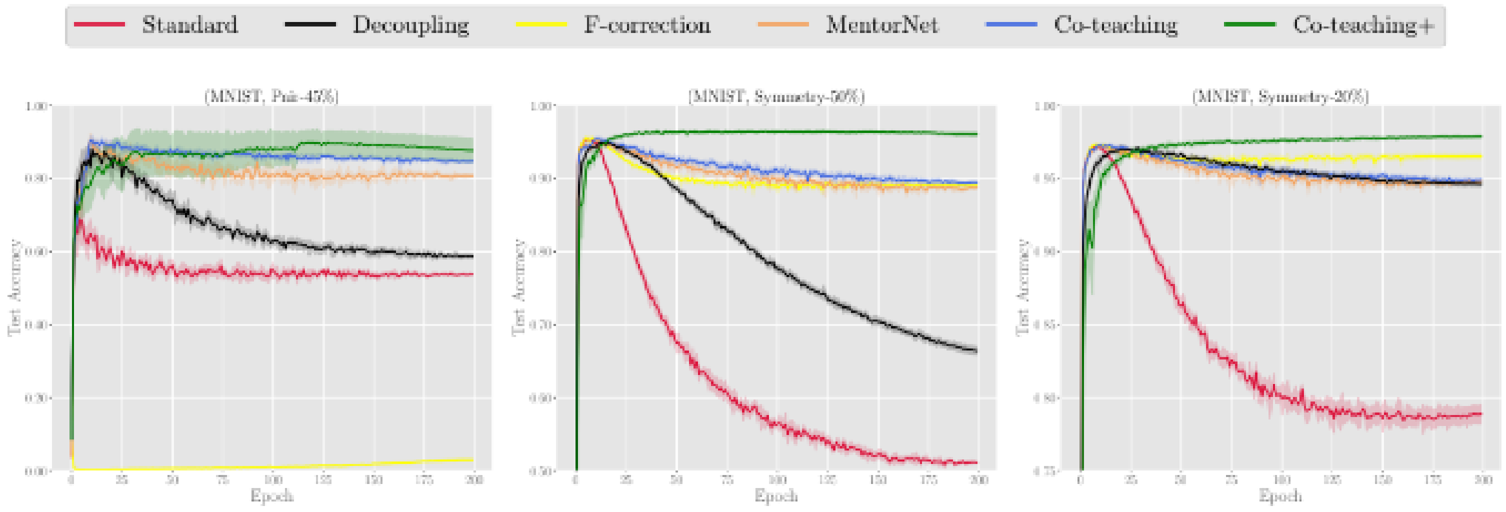


Small loss | Cross update | Joint training | Disagreement | Agreement

Co-teaching+ – experiments

Data set: four benchmark data sets were used: MNIST, CIFAR-10, CIFAR-100 and NEWS. Those data sets were clean, so noise was introduced according to the following scenarios:

- Symmetry flipping
- Pair flipping

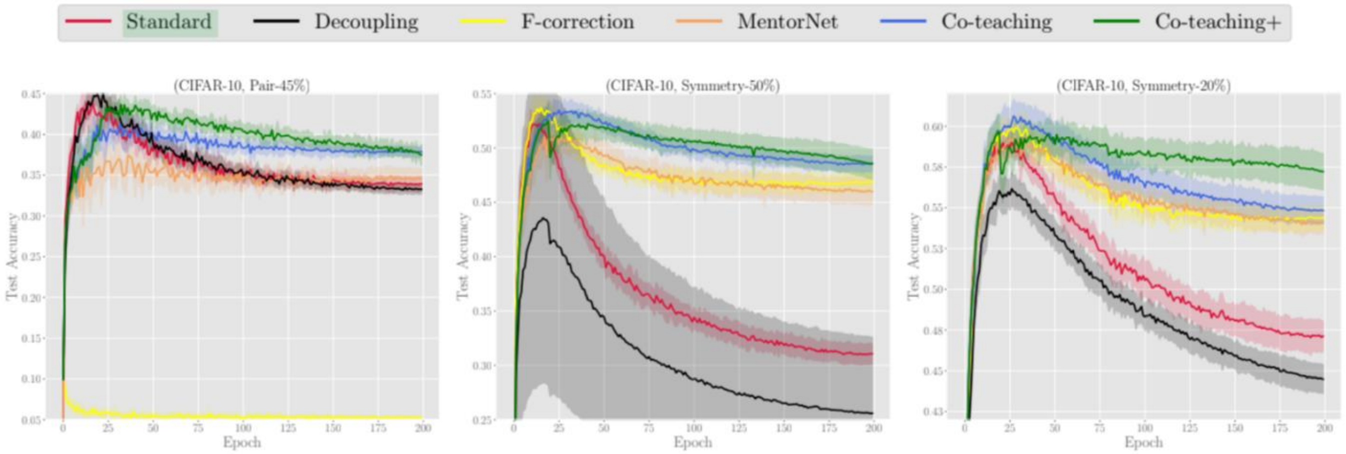


(a) Pair-45%. (b) Symmetry-50%. (c) Symmetry-20%.

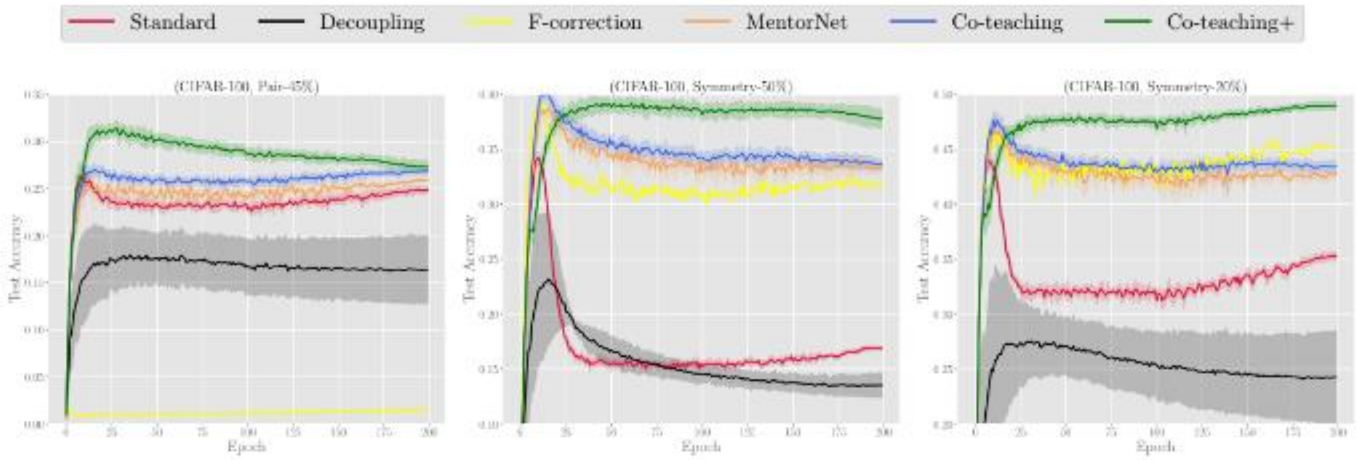
Figure 3. Test accuracy vs. number of epochs on *MNIST* dataset.

Small loss | Cross update | Joint training | Disagreement | Agreement

Co-teaching+ – experiments



(a) Pair-45%. (b) Symmetry-50%. (c) Symmetry-20%.

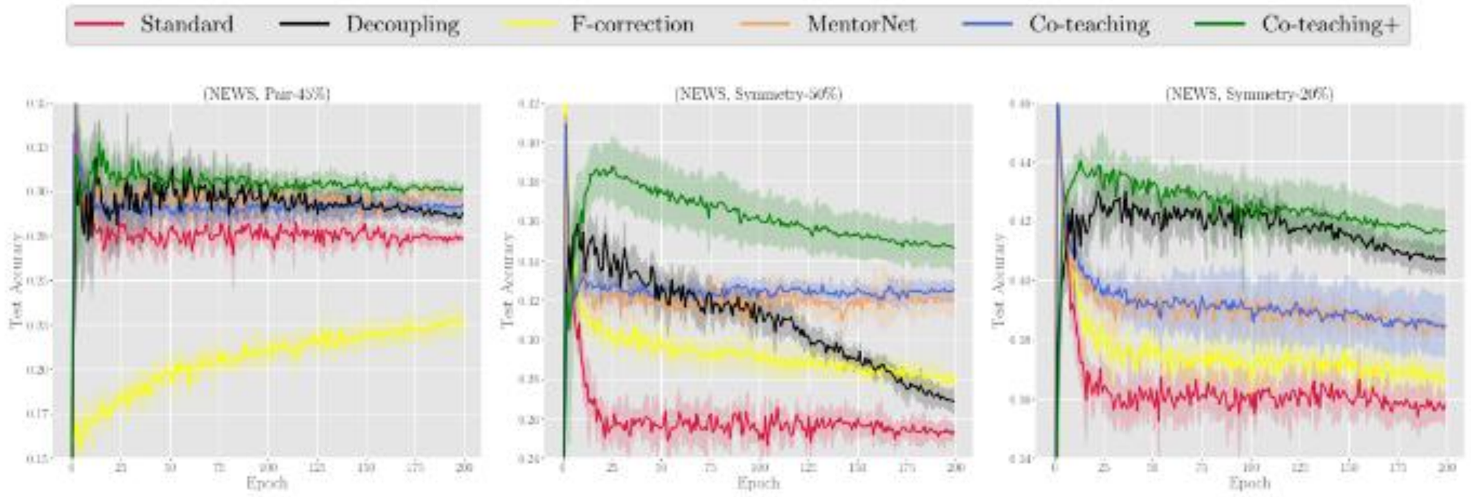


(a) Pair-45%. (b) Symmetry-50%. (c) Symmetry-20%.

Figure 5. Test accuracy vs. number of epochs on *CIFAR-100* dataset.

Small loss | Cross update | Joint training | Disagreement | Agreement

Co-teaching+ – experiments



(a) Pair-45%. (b) Symmetry-50%. (c) Symmetry-20%.

Figure 6. Test accuracy vs. number of epochs on *NEWS* dataset.

Table 4. Averaged/maximal test accuracy (%) of different approaches on *T-ImageNet* over last 10 epochs. The best results are in bold.

Flipping-Rate(%)	Standard	Decoupling	F-correction	MentorNet	Co-teaching	Co-teaching+
Pair-45%	26.14/26.32	26.10/26.61	0.63/0.67	26.22/26.61	27.41/**27.82**	26.54/26.87
Symmetry-50%	19.58/19.77	22.61/22.81	32.84/33.12	35.47/35.76	37.09/37.60	41.19/**41.77**
Symmetry-20%	35.56/35.80	36.28/36.97	44.37/44.50	45.49/45.74	45.60/46.36	47.73/**48.20**

Table 5. Averaged/maximal test accuracy (%) of different approaches on *Open-sets* over last 10 epochs. The best results are in bold.

Open-set noise	Standard	MentorNet	Iterative (Wang et al., 2018)	Co-teaching	Co-teaching+
CIFAR-10+CIFAR-100	62.92	79.27/79.33	79.28	79.43/79.58	79.28/**79.74**
CIFAR-10+ImageNet-32	58.63	79.27/79.40	79.38	79.42/79.60	79.89/**80.52**
CIFAR-10+SVHN	56.44	79.72/79.81	77.73	80.12/80.33	80.62/**80.95**

This paper does not claim to improve the SOTA results. It claims to reduce the memorization effect caused by prolonged training on noisy data.

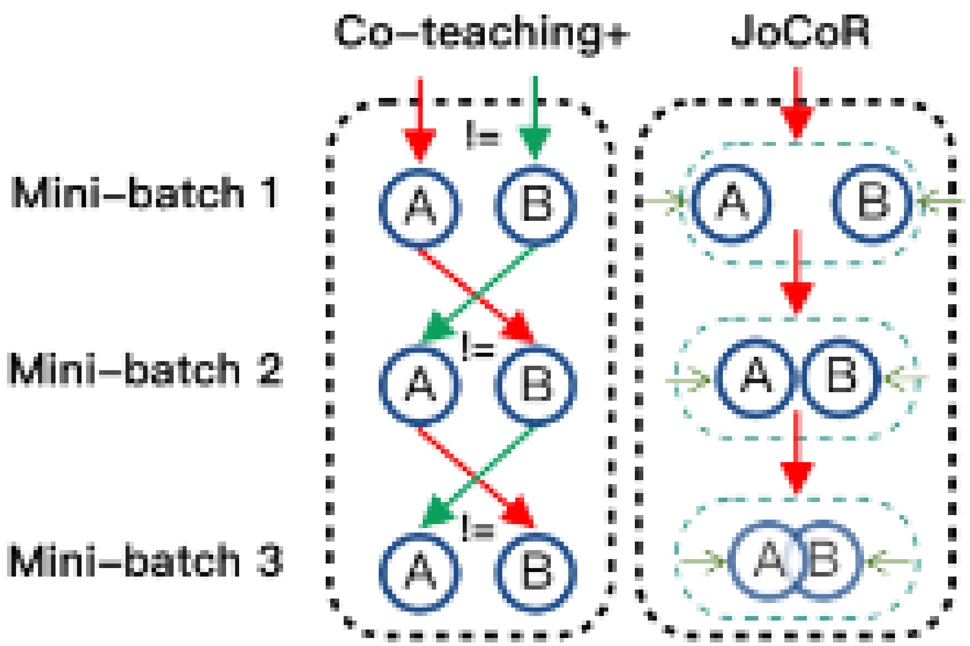
Small loss | Cross update | Joint training | Disagreement | Agreement

Joint training with Co-Regularization

Co-teaching+ – key concepts

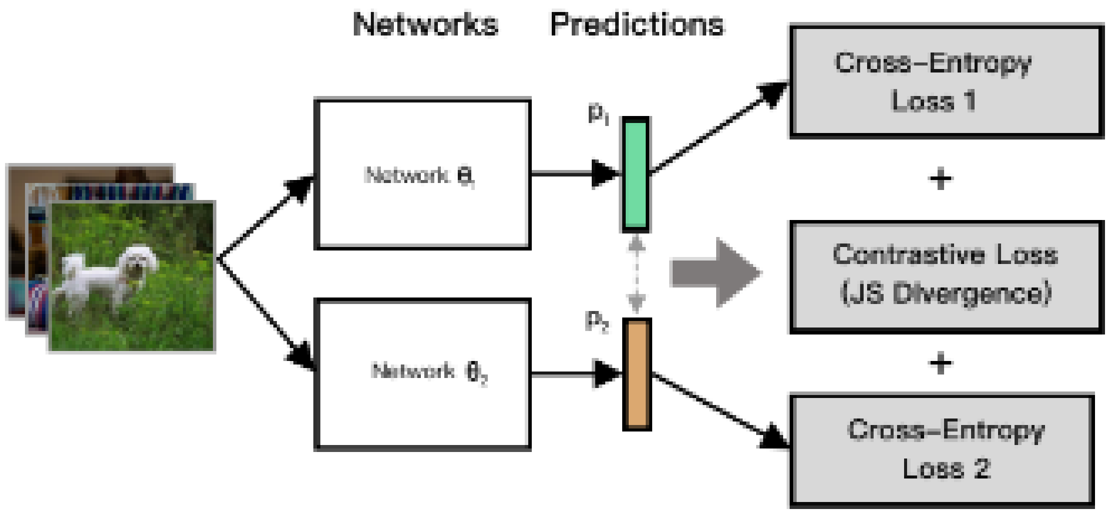
Key concept: reducing the diversity of two networks during training.

High level steps:
- Both networks do prediction for all the data
- A join loss with co-regularization is calculated for each training sample
- Small-loss samples are selected based on the joint function
- The selections are back propagated through both networks simultaneously



Small loss | Cross update | Joint training | Disagreement | Agreement

JoCoR – Joint training with Co-Regularization



Error function:

$$\ell(x_i) = (1-\lambda) * \ell_{\text{sup}}(\boldsymbol{x}_i, y_i) + \lambda * \ell_{\text{con}}(\boldsymbol{x}_i)$$

Classification loss:

$$\begin{aligned}\ell_{\text{sup}}(\boldsymbol{x}_i, y_i) &= \ell_{C1}(\boldsymbol{x}_i, y_i) + \ell_{C2}(\boldsymbol{x}_i, y_i) \\ &= -\sum\nolimits_{i=1}^{N} \sum\nolimits_{m=1}^{M} y_i \log(p_1^m(\boldsymbol{x}_i)) \\ &\quad -\sum\nolimits_{i=1}^{N} \sum\nolimits_{m=1}^{M} y_i \log(p_2^m(\boldsymbol{x}_i))\end{aligned}$$

Contrastive loss:

$$\ell_{\text{con}} = D_{\text{KL}}(\boldsymbol{p}_1 \| \boldsymbol{p}_2) + D_{\text{KL}}(\boldsymbol{p}_2 \| \boldsymbol{p}_1)$$

Small loss | Cross update | Joint training | Disagreement | Agreement

JoCoR – Joint training with Co-Regularization

Algorithm 1 JoCoR

Input: Network f with $\Theta = \{\Theta_1, \Theta_2\}$, learning rate η, fixed τ, epoch T_k and $T_{\max}$, iteration $I_{\max}$;

for $t = 1, 2, \ldots, T_{\max}$ **do**

  Shuffle training set D;

  for $n = 1, \ldots, I_{\max}$ **do**

    Fetch mini-batch D_n from D;

    $p_1 = f(x, \Theta_1), \forall x \in D_n$;

    $p_2 = f(x, \Theta_2), \forall x \in D_n$;

    Calculate the joint loss ℓ by (1) using p_1 and p_2;

    Obtain small-loss sets $\tilde{D}_n$ by (5) from D_n;

    Obtain L by (6) on $\tilde{D}_n$;

    Update $\Theta = \Theta - \eta \nabla L$;

  end for

  Update $R(t) = 1 - \min\left\{\frac{t}{T_k}\tau, \tau\right\}$

end for

Output: Θ_1 and Θ_2

Controlling how many small-loss data should be selected is similar to Co-teaching+ approach.

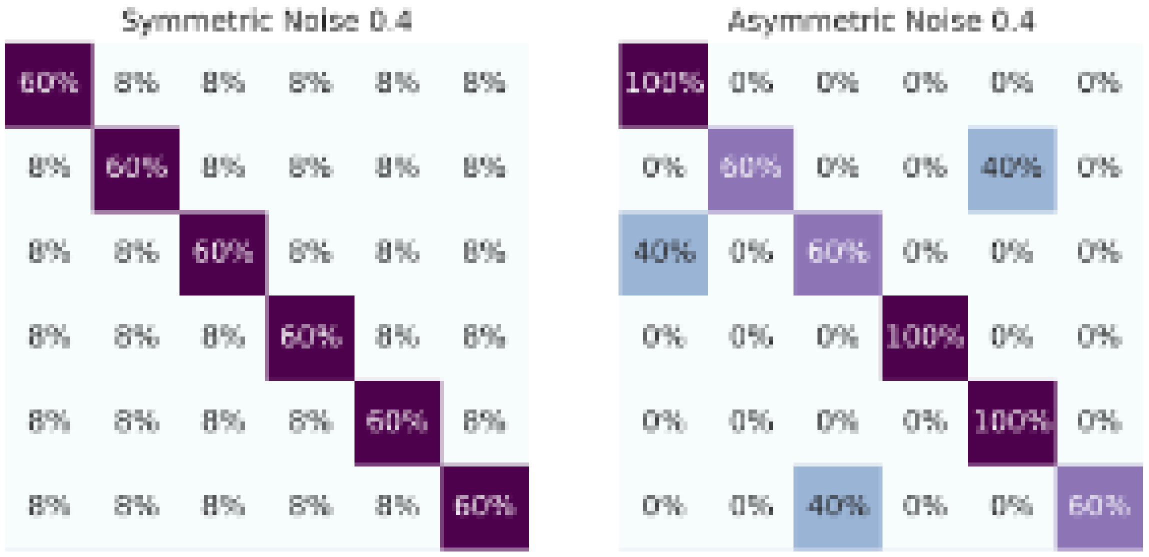
The intuition behind explicit regularization that aims at agreement is that two models are unlikely to agree on a incorrect label.

Small loss | Cross update | Joint training | Disagreement | Agreement

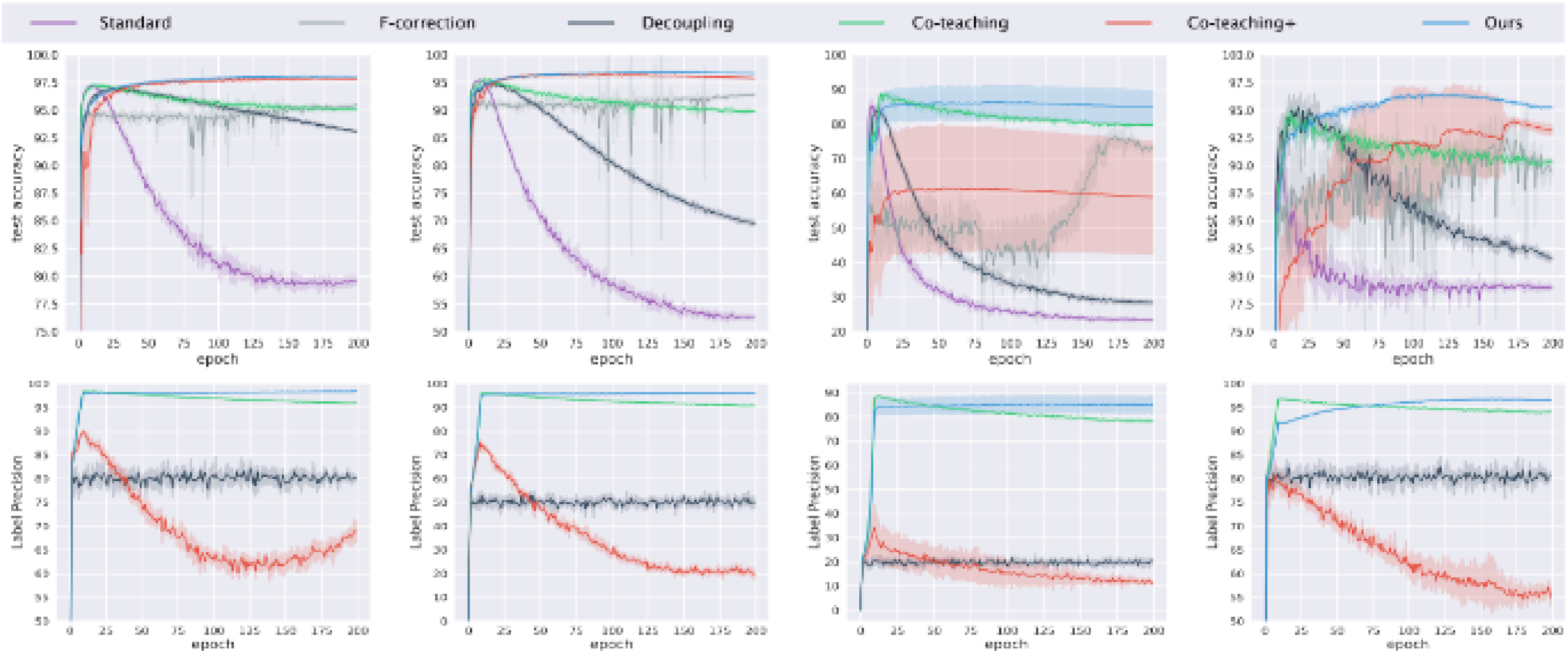
JoCoR – Joint training with Co-Regularization – experiments

Data set: four benchmark data sets were used: MNIST, CIFAR-10, CIFAR-100 and Clothing1M. The first 3 data sets were clean, so noise was introduced according to the following scenarios:

- Symmetric flipping
- Asymmetric flipping



JoCoR – Joint training with Co-Regularization – experiments



(a) Symmetry-20% (b) Symmetry-50% (c) Symmetry-80% (d) Asymmetry-40%

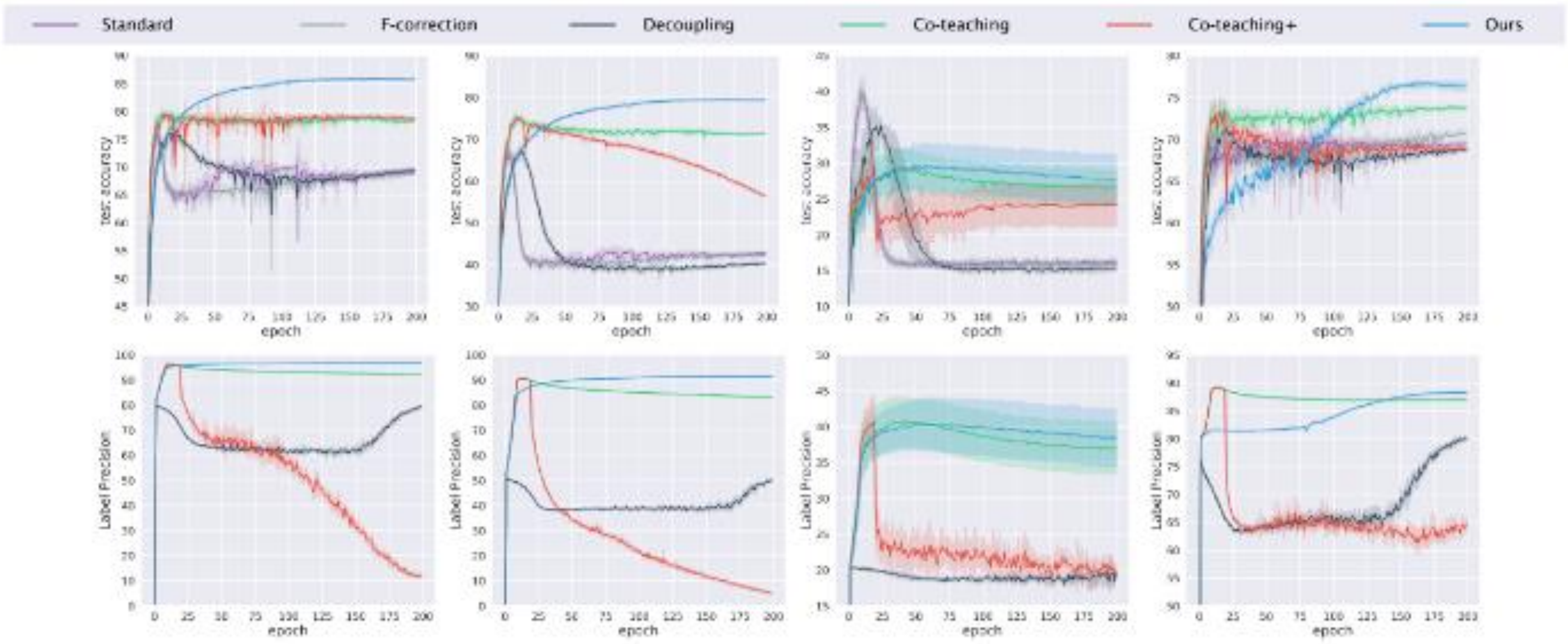
Figure 3. Results on MNIST dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

Table 2. Average test accuracy (%) on *MNIST* over the last 10 epochs.

Flipping-Rate	Standard	F-correction	Decoupling	Co-teaching	Co-teaching+	JoCoR
Symmetry-20%	79.56 ± 0.44	95.38 ± 0.10	93.16 ± 0.11	95.10 ± 0.16	97.81 ± 0.03	$\mathbf{98.06} \pm 0.04$
Symmetry-50%	52.66 ± 0.43	92.74 ± 0.21	69.79 ± 0.52	89.82 ± 0.31	95.80 ± 0.09	$\mathbf{96.64} \pm 0.12$
Symmetry-80%	23.43 ± 0.31	72.96 ± 0.90	28.51 ± 0.65	79.73 ± 0.35	58.92 ± 14.73	$\mathbf{84.89} \pm 4.55$
Asymmetry-40%	79.00 ± 0.28	89.77 ± 0.96	81.84 ± 0.38	90.28 ± 0.27	93.28 ± 0.43	$\mathbf{95.24} \pm 0.10$

Small loss | Cross update | Joint training | Disagreement | Agreement

JoCoR – Joint training with Co-Regularization – experiments



(a) Symmetry-20% (b) Symmetry-50% (c) Symmetry-80% (d) Asymmetry-40%

Figure 5. Results on CIFAR-10 dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

Table 3. Average test accuracy (%) on *CIFAR-10* over the last 10 epochs.

Flipping-Rate	Standard	F-correction	Decoupling	Co-teaching	Co-teaching+	JoCoR
Symmetry-20%	69.18 ± 0.52	68.74 ± 0.20	69.32 ± 0.40	78.23 ± 0.27	78.71 ± 0.34	$\mathbf{85.73} \pm 0.19$
Symmetry-50%	42.71 ± 0.42	42.19 ± 0.60	40.22 ± 0.30	71.30 ± 0.13	57.05 ± 0.54	$\mathbf{79.41} \pm 0.25$
Symmetry-80%	16.24 ± 0.39	15.88 ± 0.42	15.31 ± 0.43	26.58 ± 2.22	24.19 ± 2.74	$\mathbf{27.78} \pm 3.06$
Asymmetry-40%	69.43 ± 0.33	70.60 ± 0.40	68.72 ± 0.30	73.78 ± 0.22	68.84 ± 0.20	$\mathbf{76.36} \pm 0.49$

Small loss | Cross update | Joint training | Disagreement | Agreement

JoCoR – Joint training with Co-Regularization – experiments

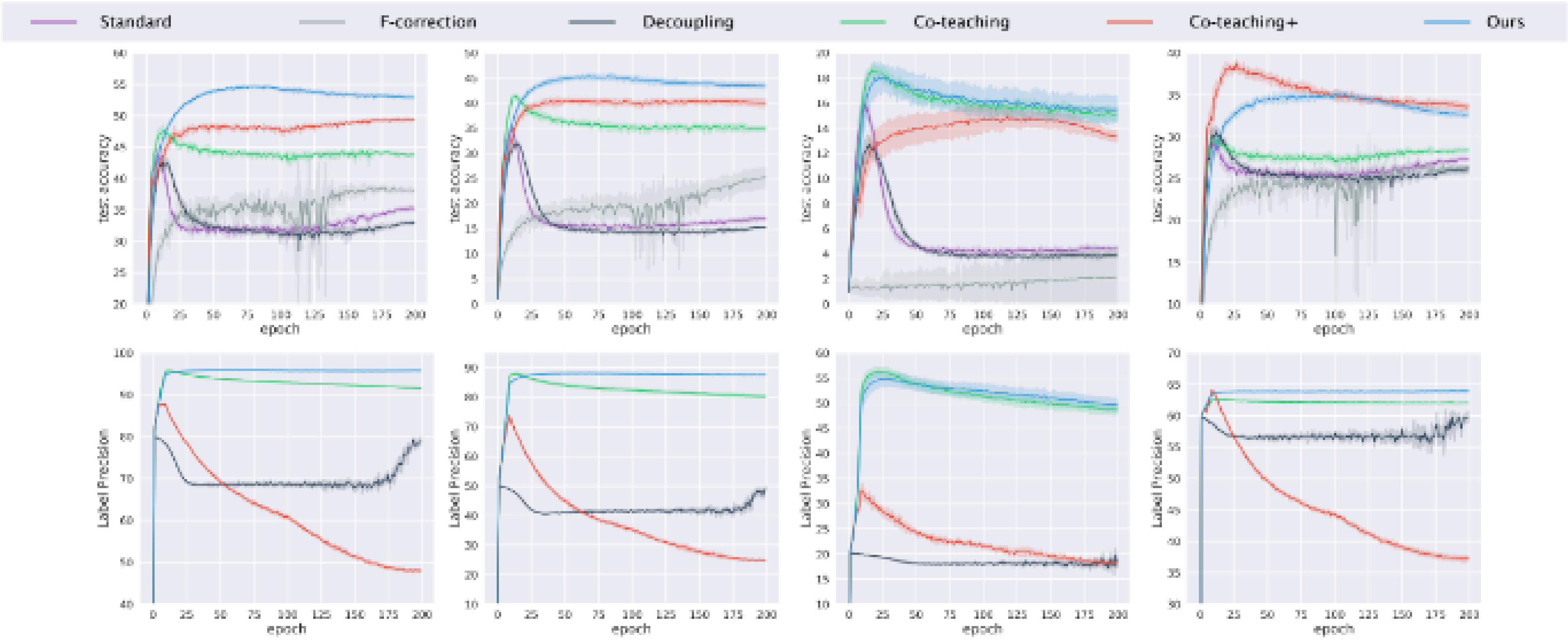


Figure 6. Results on CIFAR-100 dataset. Top: test accuracy(%) vs. epochs; bottom: label precision(%) vs. epochs.

Table 4. Average test accuracy (%) on *CIFAR-100* over the last 10 epochs.

Flipping-Rate	Standard	F-correction	Decoupling	Co-teaching	Co-teaching+	JoCoR
Symmetry-20%	35.14 ± 0.44	37.95 ± 0.10	33.10 ± 0.12	43.73 ± 0.16	49.27 ± 0.03	$\mathbf{53.01} \pm 0.04$
Symmetry-50%	16.97 ± 0.40	24.98 ± 1.82	15.25 ± 0.20	34.96 ± 0.50	40.04 ± 0.70	$\mathbf{43.49} \pm 0.46$
Symmetry-80%	4.41 ± 0.14	2.10 ± 2.23	3.89 ± 0.16	15.15 ± 0.46	13.44 ± 0.37	$\mathbf{15.49} \pm 0.98$
Asymmetry-40%	27.29 ± 0.25	25.94 ± 0.44	26.11 ± 0.39	28.35 ± 0.25	$\mathbf{33.62} \pm 0.39$	32.70 ± 0.35

Small loss | Cross update | Joint training | Disagreement | Agreement

JoCoR – Joint training with Co-Regularization – experiments

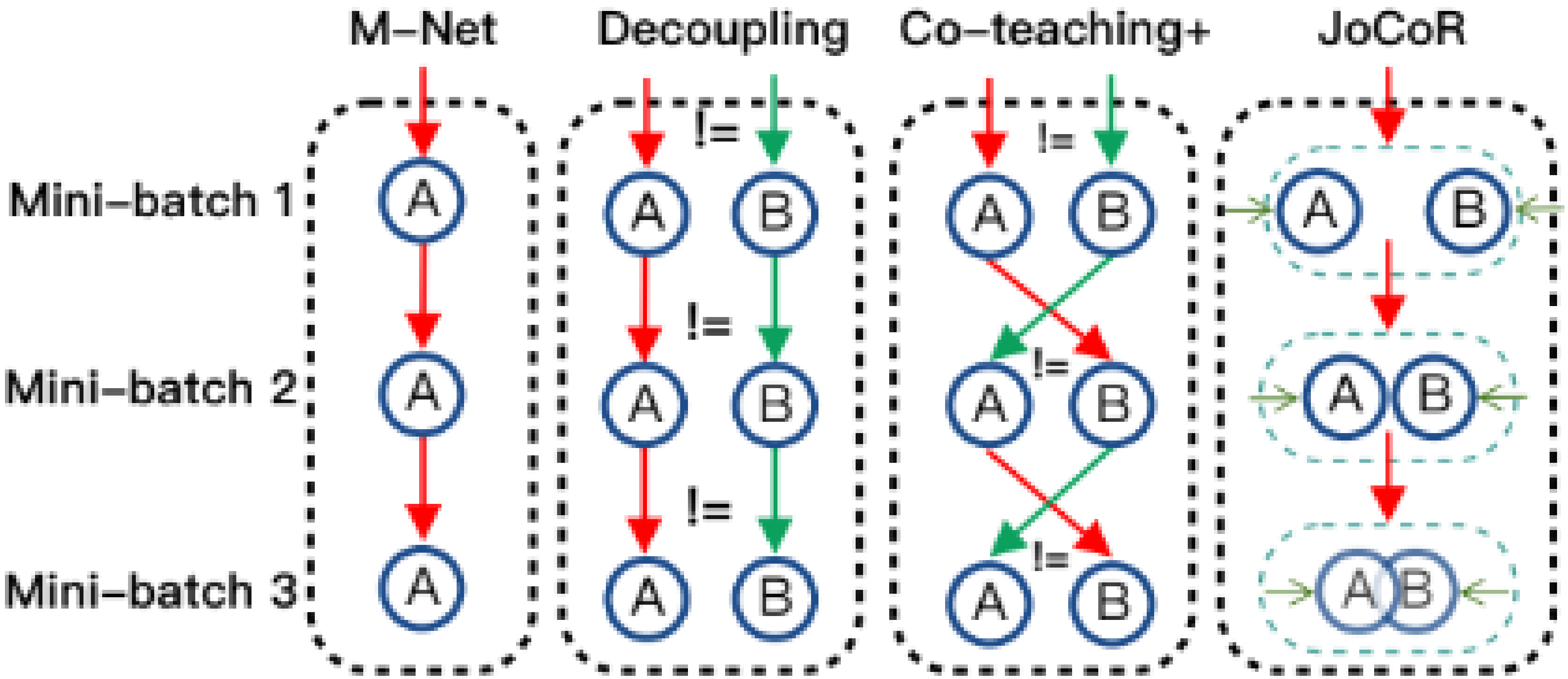
Table 5. Classification accuracy (%) on the *Clothing1M* test set

Methods	*best*	*last*
Standard	67.22	64.68
F-correction	68.93	65.36
Decoupling	68.48	67.32
Co-teaching	69.21	68.51
Co-teaching+	59.32	58.79
JoCoR	**70.30**	**69.79**

Small loss | Cross update | Joint training | Disagreement | Agreement

Advanced training strategies to cope with noisy data

Training strategies for noisy labels – overview



Training strategies for noisy labels

	Decoupling	Co-teaching	Co-teaching+	JoCoR
small loss	✗	✓	✓	✓
cross update	✗	✓	✓	✗
joint training	✗	✗	✗	✓
disagreement	✓	✗	✓	✗
agreement	✗	✗	✗	✓

The end.
Thank you!