Towards Robust Deep Learning

Bo Han

https://bhanml.github.io/
Center for Advanced Intelligence Project, RIKEN, Japan
Centre for Artificial Intelligence Research, University of Technology Sydney, Australia

November 22, 2018

Outline

Real-world Scenario I

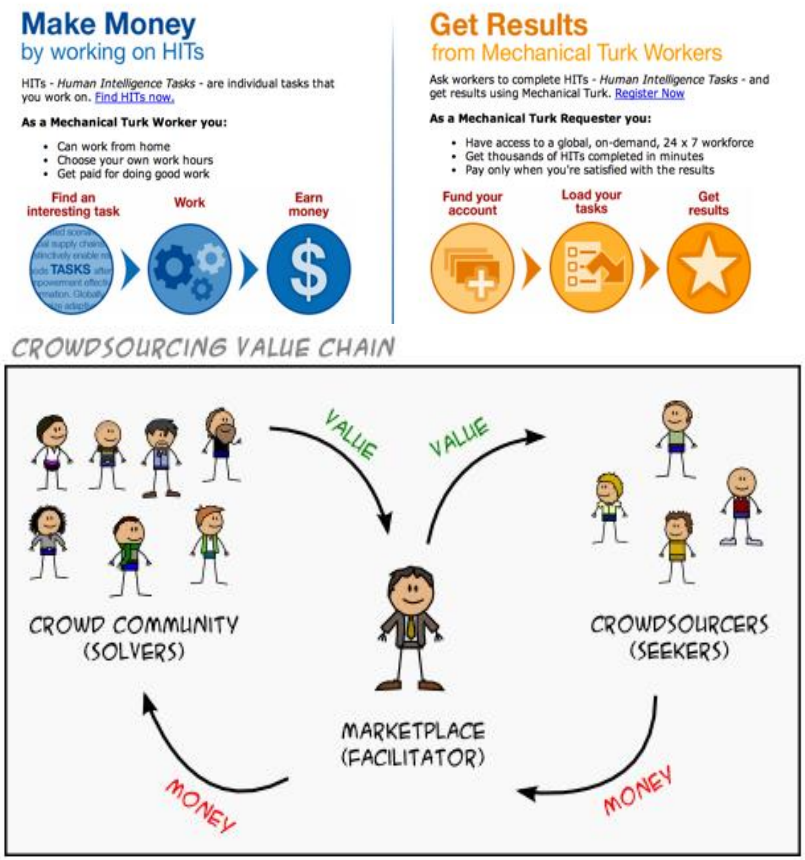


Figure : Labels from crowdsourcing.

Real-world Scenario II

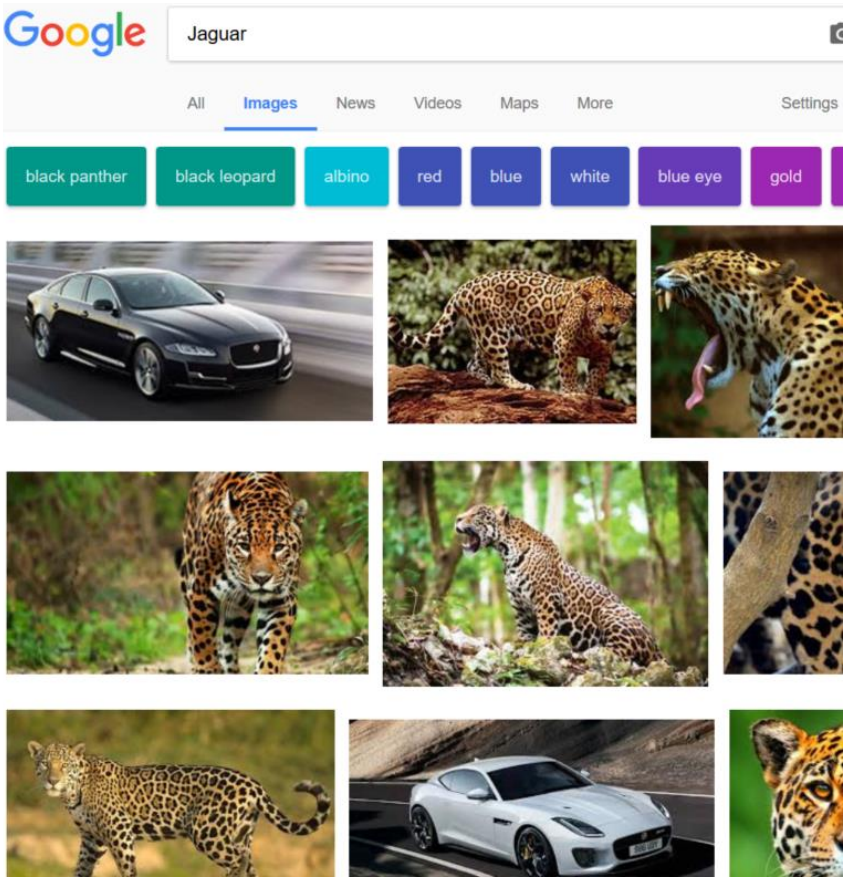


Figure : Labels from web search.

Real-world Scenario III

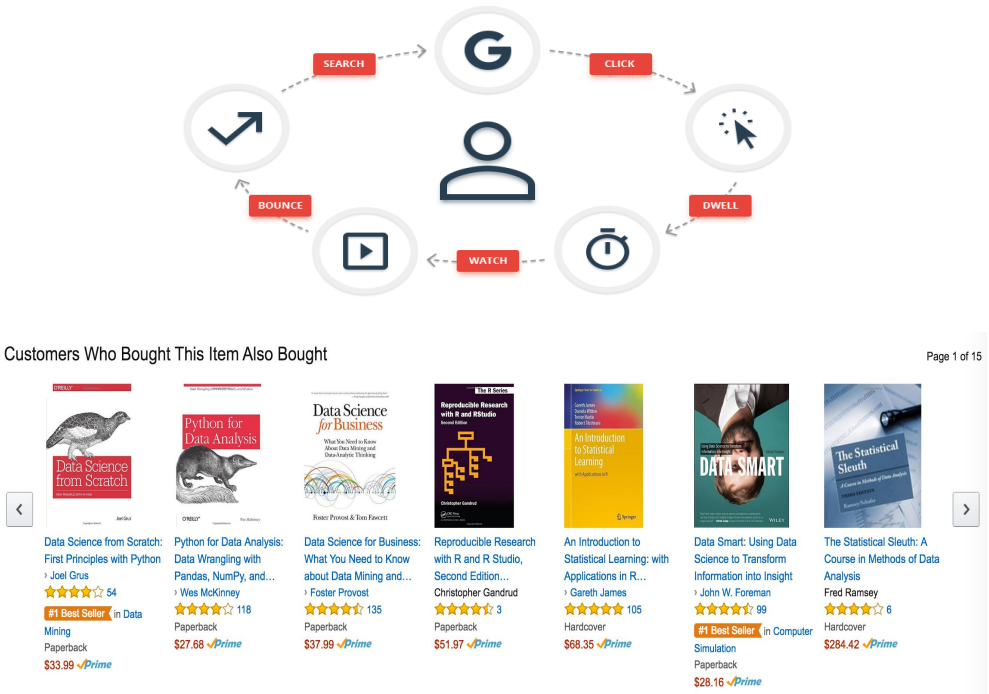


Figure : Labels from implicit feedback.

Simple Example

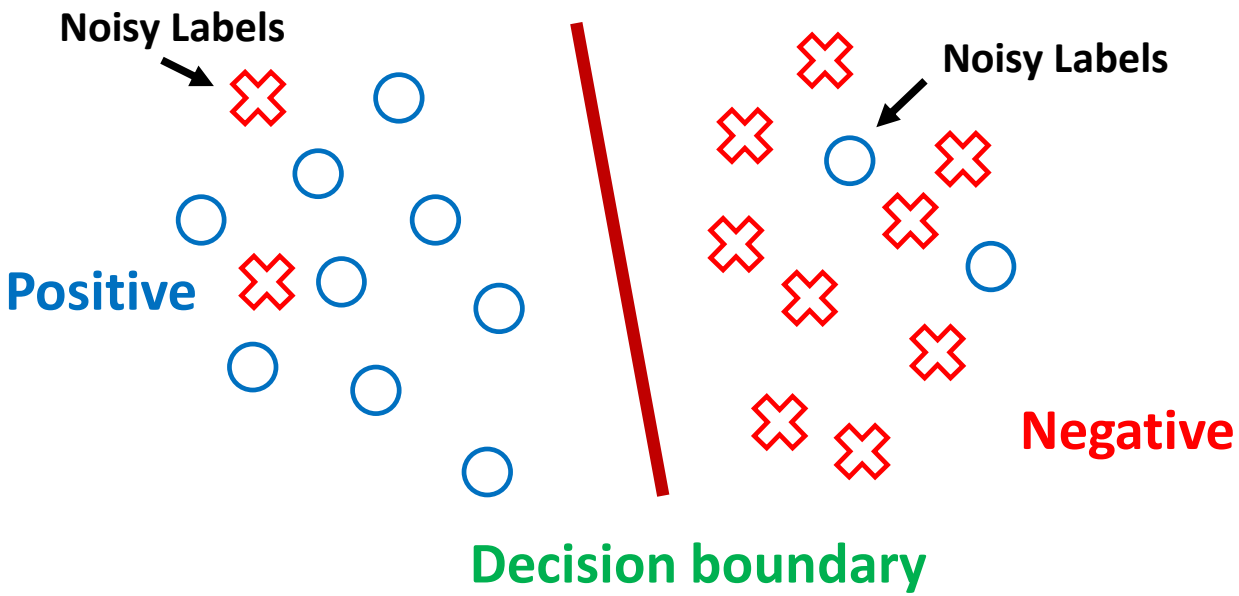


Figure : Noisy labels in binary-class datasets.

Research Directions

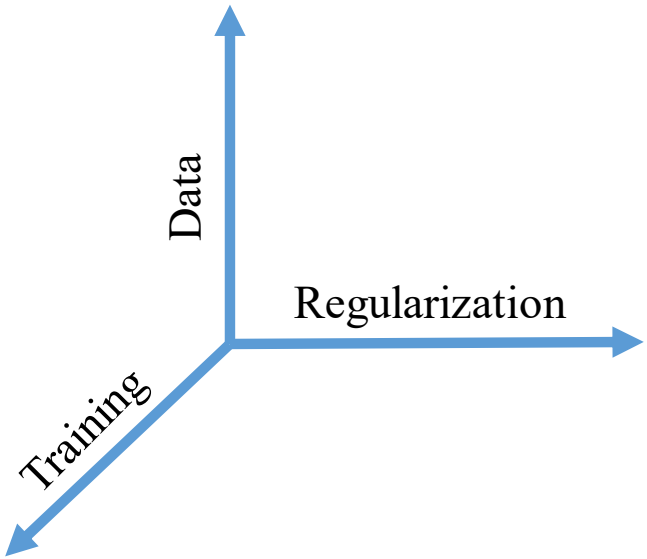


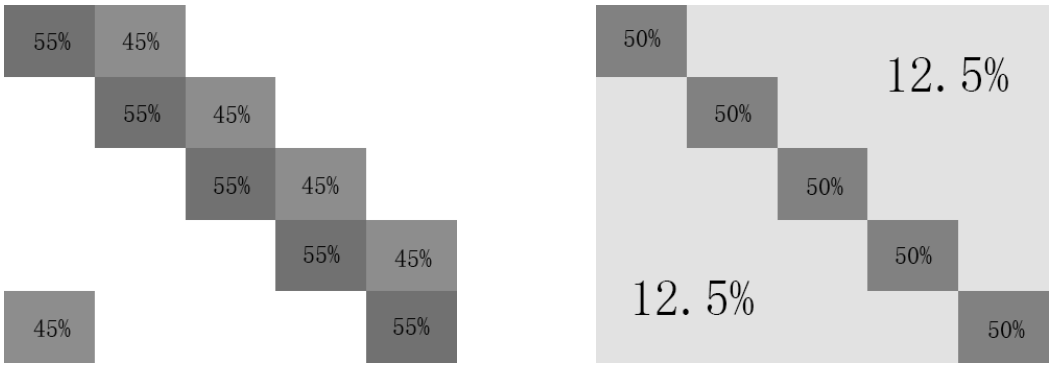
Figure : Three orthogonal directions in deep learning with noisy labels.

Current Progress

- Estimating noise transition matrix:
  - Backward Correction (Australian National University, CVPR'17)
  - S-adaptation (Bar Ilan University, ICLR'17)
- Training on selected samples:
  - MentorNet (Google AI, ICML'18)
  - Learning to Reweight Examples (University of Toronto, ICML'18)
  - Decoupling (Hebrew University of Jerusalem, NIPS'17)
- Designing implicit regularization:
  - Mean Teachers (Curious AI, NIPS'17)
  - Temporal Ensembling (NVIDIA, ICLR'17)
  - Virtual Adversarial Training (Preferred Networks, ICLR'16)

Estimating Noise Transition Matrix

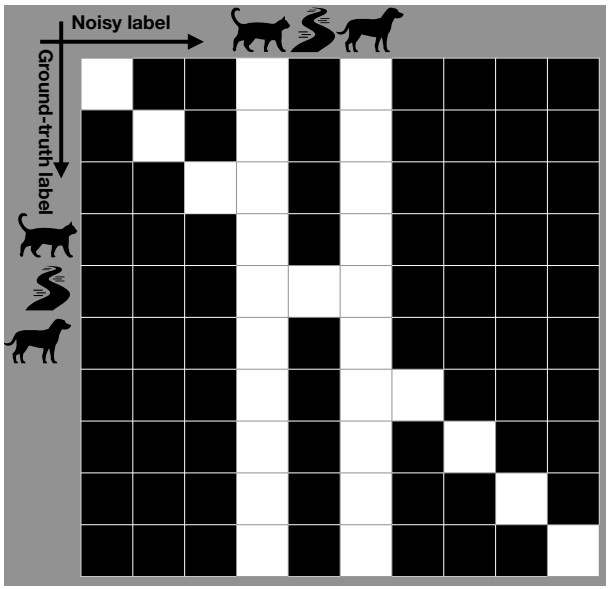
- Main idea: estimate the matrix and learn the classifier
- Benefit: with theoretical guarantees
- Drawback: hard to estimate the matrix for large-class cases



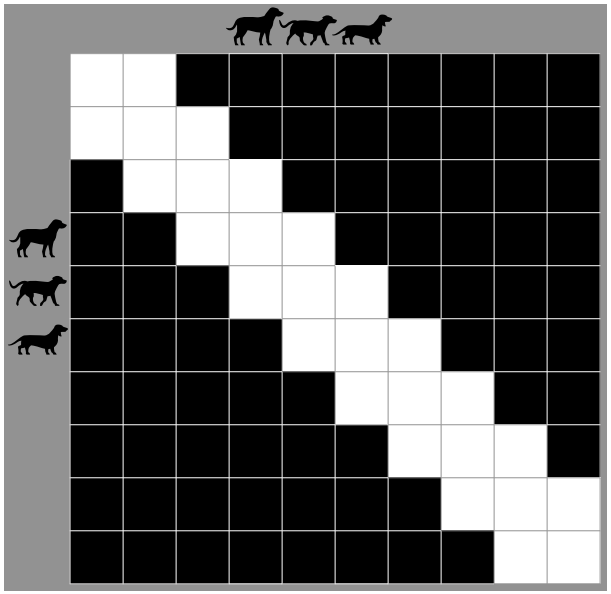
(a) pair ($\epsilon = 45\%$). (b) sym ($\epsilon = 50\%$).

Figure : The noise transition matrix T, where $T_{ij} = \Pr(\tilde{y} = e^j | y = e^i)$.

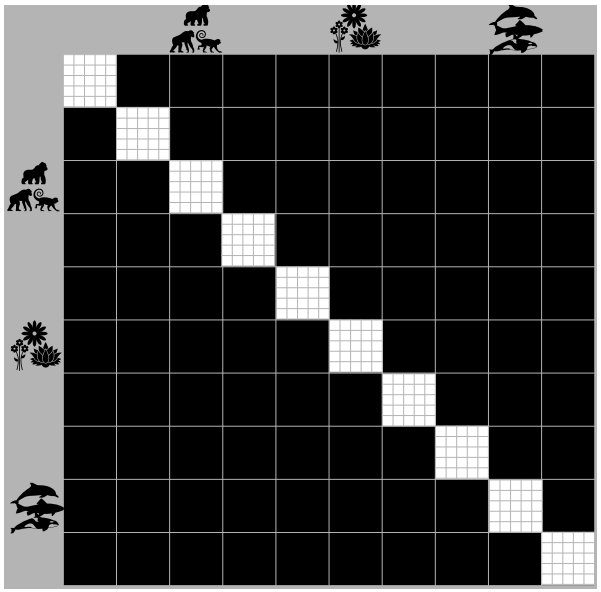
Data Perspective



(a) Column-diagonal



(b) Tri-diagonal



(c) Block-diagonal

Figure : Three types of noise structure.

- (a) beach $\nleftrightarrow$ mountain; beach $\leftrightarrow$ dog.
- (b1) Australian terrier $\nleftrightarrow$ Norwich terrier;
- (b2) Norfolk terrier $\leftrightarrow$ Norwich terrier $\leftrightarrow$ Irish terrier.
- (c) aquatic mammals $\nleftrightarrow$ flowers; beaver $\leftrightarrow$ dolphin.

Deficiency of Benchmarks

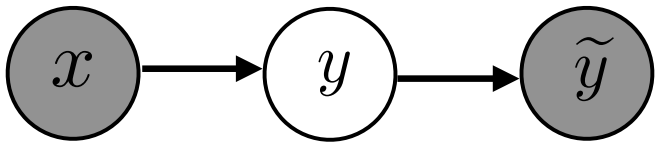


Figure : Benchmark models. $(x, \tilde{y})$ denotes the instance with the noisy label.

- Independent framework: the estimation is not for agnostic noisy data.
- Unified framework: the brute-force estimation suffer local minimums.

Our Solution: Structure-aware probabilistic model

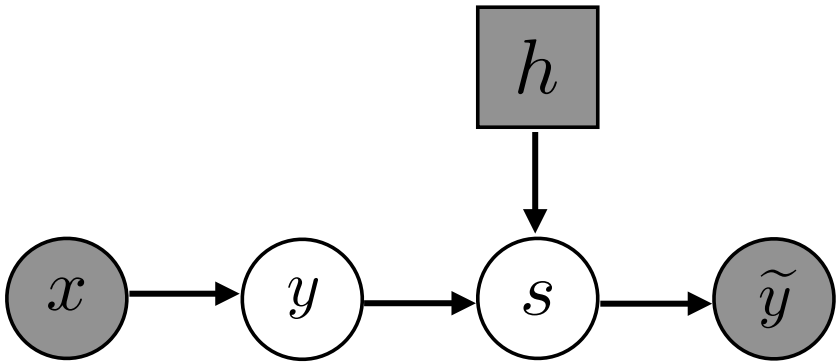


Figure : MASKING models the matrix T, where $T_{ij} = \Pr(\tilde{y} = e^j | y = e^i)$, by an explicit variable s. Thus, we embed a structure constraint (h) on the variable s.

- Human cognition masks the invalid class transitions.
- The model focuses on estimating the noise transition probability.
- The estimation burden will be largely reduced.

When Structure Meets Generative Model

- The latent ground-truth label $y \sim P(y|x)$ (Categorical).
- The transition $s \sim P(s)$ and its structure $s_o \sim P(s_o)$, where $P(s)$ is an implicit distribution modeled by DNN, $P(s_o) = P(s)\frac{ds}{ds_o}\Big|_{s_o=f(s)}$. $f(\cdot)$ is the mapping function from s to s_o.
- The noisy label $\tilde{y} \sim P(\tilde{y}|y,s)$, where $P(\tilde{y}|y,s)$ models the transition from y to $\tilde{y}$ given s.

ELBO of MASKING

$$\ln P(\tilde{y}|x) \geq \mathbb{E}_{Q(s)} \left[\underbrace{\ln \sum_{y} P(\tilde{y}|y,s)P(y|x)}_{\text{previous model}} - \ln \left(Q(s_o) / \underbrace{P(s_o)}_{\text{structure prior}} \right) \Big|_{s_o=f(s)} \right],$$

where $Q(s)$ is the variational distribution to approximate the posterior of the noise transition matrix s, and $Q(s_o) = Q(s)\frac{ds}{ds_o}\big|_{s_o=f(s)}$ is the corresponding variational distribution of the structure s_o.

Remark

MASKING benefits from the human guidance (the second term) in the procedure of learning with noisy supervision (the first term).

Principled Realization

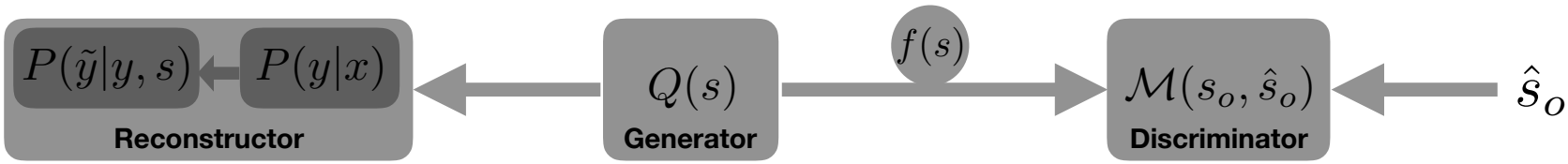
Q1: Challenge from structure extraction.

A1: the tempered sigmoid func as $f(\cdot)$ to map from s to s_o,

$$f(s) = \frac{1}{1 + \exp(-\frac{s-\alpha}{\beta})}, \quad \text{where } \alpha \in (0,1), \beta \ll 1.$$

Q2: Challenge from structure alignment.

A2: GAN-like structure to model the structure instillation.



Network Structures I

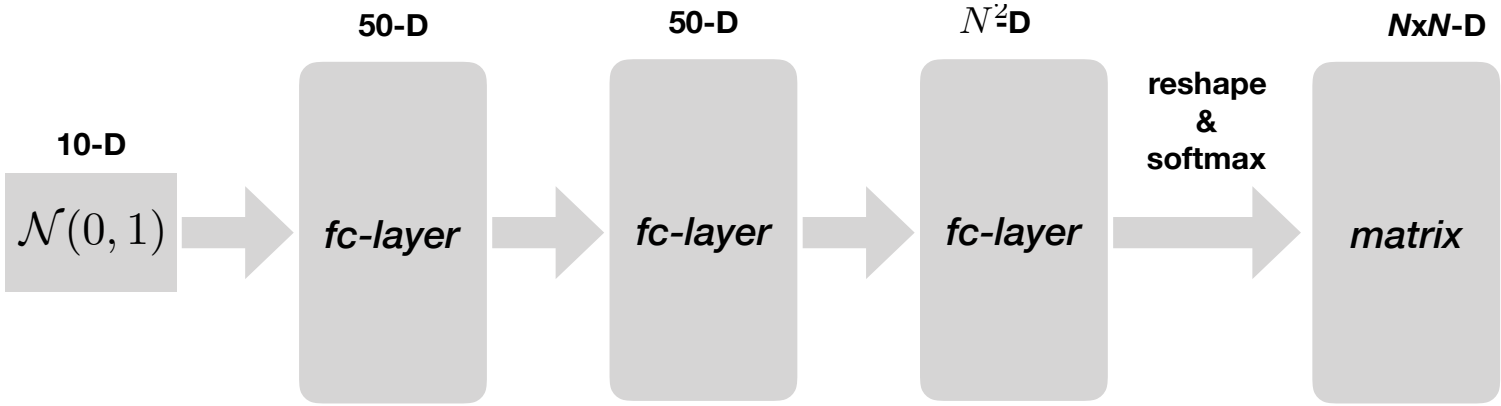


Figure : The network configuration of the generator module.

Network Structures II

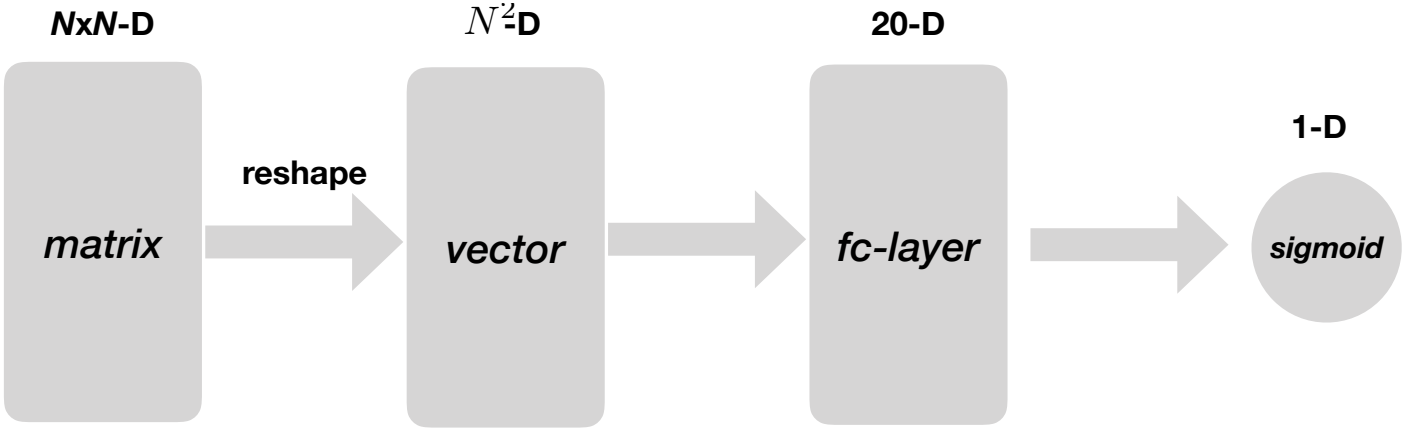


Figure : The network configuration of the discriminator module.

Network Structures III

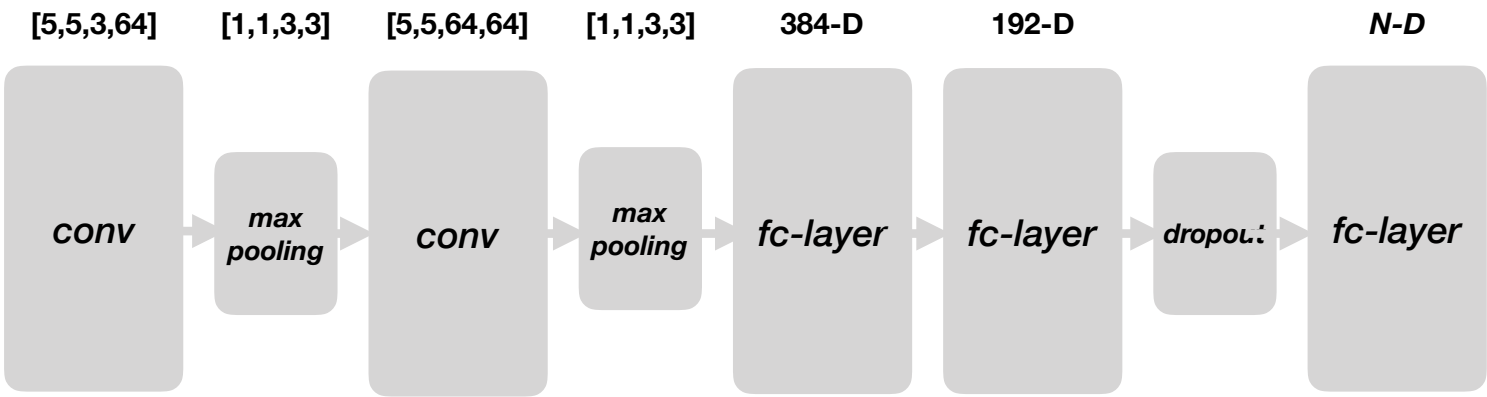


Figure : The network configuration of the reconstructor module.

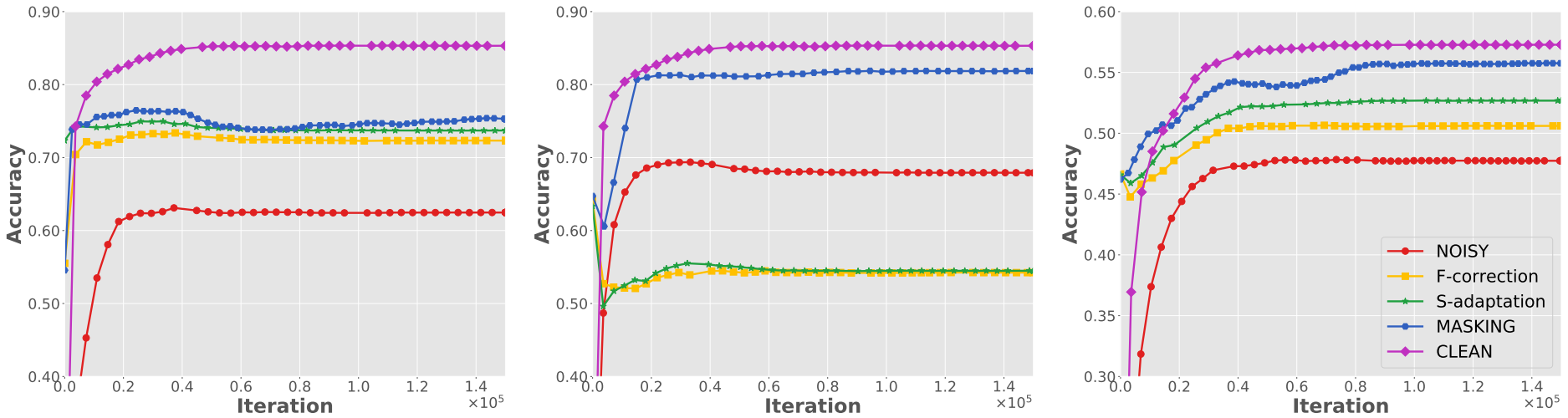
Datasets

Table : Benchmark CIFAR10 and CIFAR100; Industrial-level Clothing1M.

	# of training	# of testing	# of class	size
CIFAR10	50,000	10,000	10	32×32
CIFAR100	50,000	10,000	1000	32×32
Clothing1M	1,000,000(N) + 5,000(C)	1,000	14	256×256

Figure : Mislabeled images often share similar visual patterns in Clothing1M.

CIFAR10 and CIFAR100



(a) Column-diagnoal (b) Tri-diagnoal (c) Block-diagnoal

Figure : The test accuracy vs iterations on benchmark datasets.

Clothing1M

Table : Test accuracy on Clothing1M.

Models	Performance(%)
NOISY	68.9
Loss-correction	69.8
S-adaption	70.3
MASKING	71.1
CLEAN	75.2

Future

Open questions:

- When the initial structure is wrongly set, how does our model correct the initial structure from the finite dataset?

AI landing possibilities:

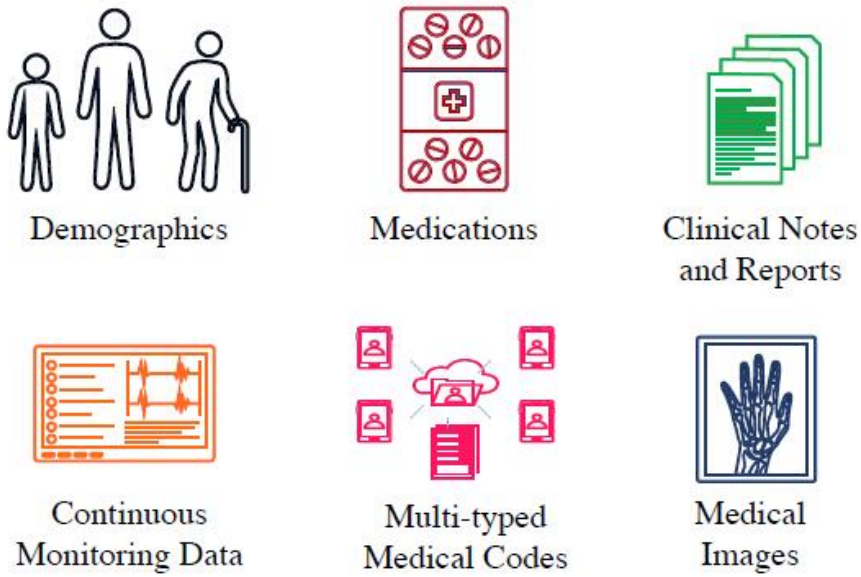


Figure : Healthcare application in HKBU.

Training on Selected Samples

- Main idea: regard small-loss instances as "correct" instances
- Benefit: easy to implement & free of assumptions
- Drawback: accumulated error caused by sample-selection bias

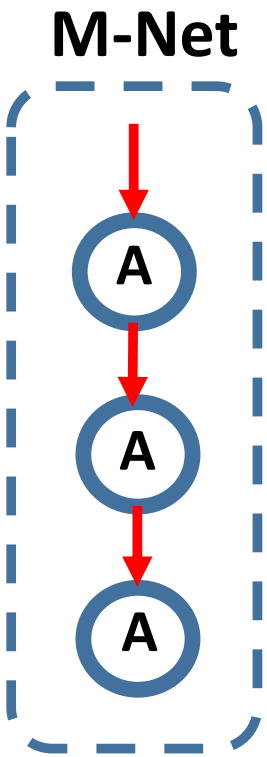
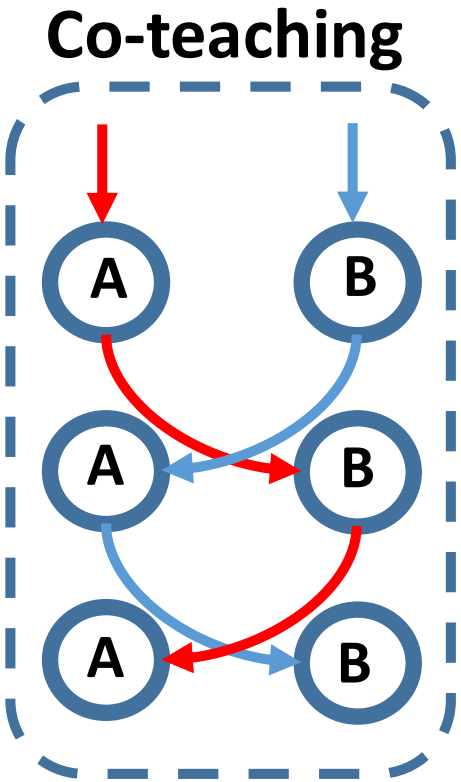


Figure : Self-training MentorNet.

Training Perspective



- Co-teaching maintains two networks (A & B) simultaneously.
- Each network samples its small-loss instances as the useful knowledge.
- Each network teaches such useful instances to its peer network.

Co-teaching Paradigm

Input: w_f and w_g, learning rate η, fixed τ, epoch T_k and $T_{\max}$, iter $N_{\max}$;
for $T = 1, 2, \ldots, T_{max}$ do
  Shuffle: training set $\mathcal{D}$; //noisy dataset;
  for $N = 1, \ldots, N_{\max}$ do
    Draw: mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
    Sample: $\bar{\mathcal{D}}_f = \arg\min_{\bar{\mathcal{D}}} \ell(f, \bar{\mathcal{D}}, R(T))$; //$R(T)$% small-loss;
    Sample: $\bar{\mathcal{D}}_g = \arg\min_{\bar{\mathcal{D}}} \ell(g, \bar{\mathcal{D}}, R(T))$; //$R(T)$% small-loss;
    Update: $w_f = w_f - \eta \nabla f(\bar{\mathcal{D}}_g)$; //update w_f by $\bar{\mathcal{D}}_g$;
    Update: $w_g = w_g - \eta \nabla g(\bar{\mathcal{D}}_f)$; //update w_g by $\bar{\mathcal{D}}_f$;
  end
  Update: $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
end
Output: w_f and w_g

Algorithm 1: Co-teaching Paradigm.

Two Questions

Q1. Why sampling small-loss instances is based on dynamic $R(T)$?

A1. The "memorization" effect of DNN.

Q2. Why do we need two networks and cross-update the parameters?

A2. The peer-review process speeds up the bug findings.

Relations to Co-training

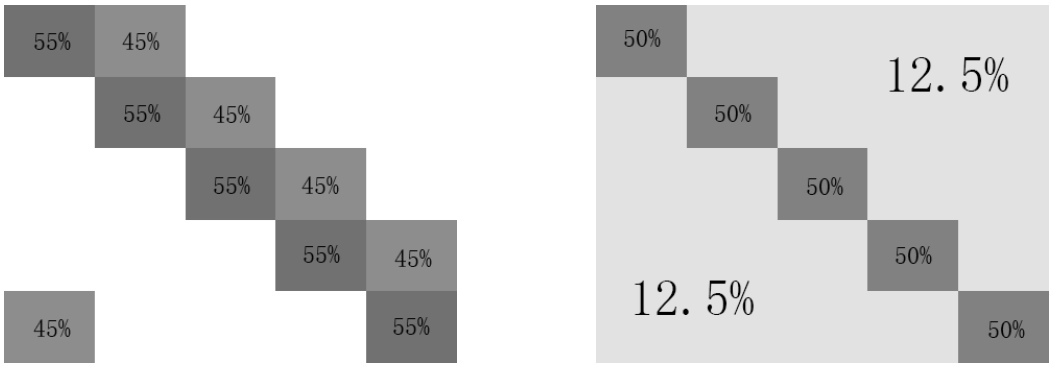
- Co-training needs two views (two independent sets of features), while Co-teaching needs a single view.
- Co-training does not exploit the memorization of deep neural networks, while Co-teaching does.
- Co-training is designed for *semi-supervised learning* (SSL), and Co-teaching is for *learning with noisy labels* (LNL).

Datasets

Table : Summary of datasets used in the experiments.

	# of training	# of testing	# of class	size
MNIST	60,000	10,000	10	28×28
CIFAR10	50,000	10,000	10	32×32
Newsgroup	11,314	7,532	7	300-dim

Noise Types



(a) pair ($\epsilon = 45\%$). (b) sym ($\epsilon = 50\%$).

Figure : Different noise types (using 5 classes as an example).

Baselines

- Bootstrap: weighted combination of predicted and original labels;
- S-model: a softmax layer;
- F-correction: loss correction on transition matrix;
- Decoupling: instances that have different predictions;
- MentorNet: small-loss trick.

Comparison of SOTA Techniques

Table : "large class": can deal with a large number of class; "heavy noise": can combat the heavy noise, i.e., high noise ratio; "flexibility": need not combine with specific network architecture; "no pre-train": can be train from scratch.

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

Network Structures

Table : CNN and MLP models used in our experiments on *MNIST*, *CIFAR10* and *Newsgroups*.

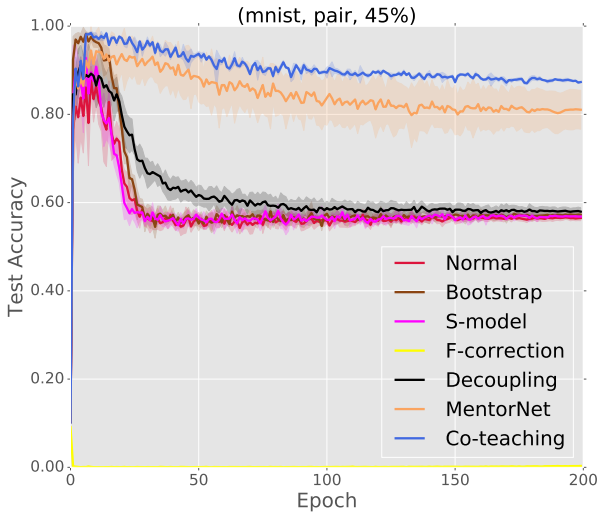
<table>
<tr><th>CNN on MNIST</th><th>CNN on CIFAR10</th><th>MLP on 20 Newsgroups</th></tr>
<tr><td>28×28 Gray Image</td><td>32×32 RGB Image</td><td>300D Embedding</td></tr>
<tr><td colspan="2">3×3 conv, 128 LReLU
3×3 conv, 128 LReLU
3×3 conv, 128 LReLU</td><td rowspan="3">dense 300→300,
Softsign</td></tr>
<tr><td colspan="2">2×2 max-pool, stride 2
dropout, $p = 0.25$</td></tr>
<tr><td colspan="2">3×3 conv, 256 LReLU
3×3 conv, 256 LReLU
3×3 conv, 256 LReLU</td></tr>
<tr><td colspan="2">2×2 max-pool, stride 2
dropout, $p = 0.25$</td><td rowspan="3">dense 300→300</td></tr>
<tr><td colspan="2">3×3 conv, 512 LReLU
3×3 conv, 256 LReLU
3×3 conv, 128 LReLU</td></tr>
<tr><td colspan="2">avg-pool</td></tr>
<tr><td>dense 128→10</td><td>dense 128→10</td><td>dense 300→2</td></tr>
</table>

MNIST I

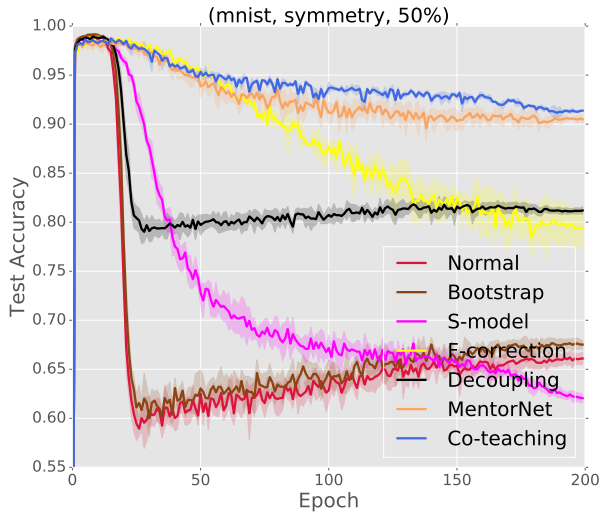
Table : Average test accuracy on *MNIST* over the last ten epoch.

(Flipping, Rate)	Normal	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%

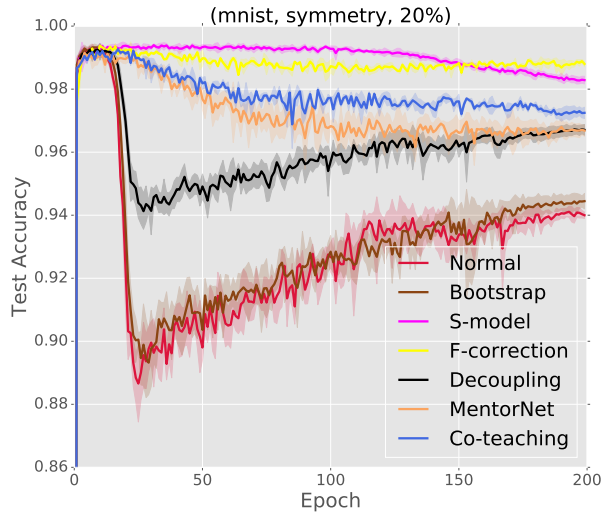
MNIST II



(a) pair, 45%.



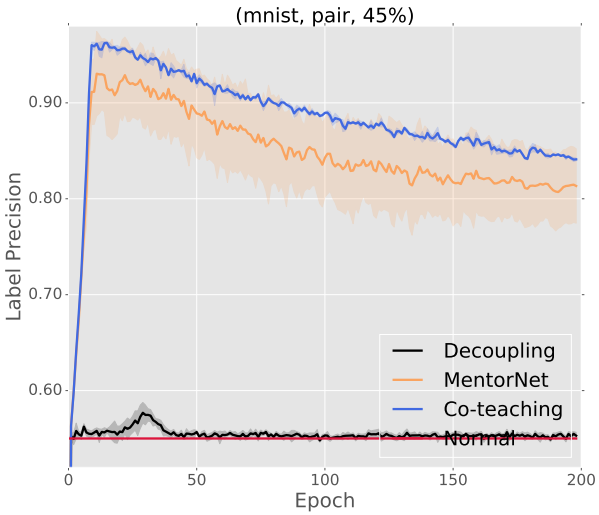
(b) symmetry, 50%.



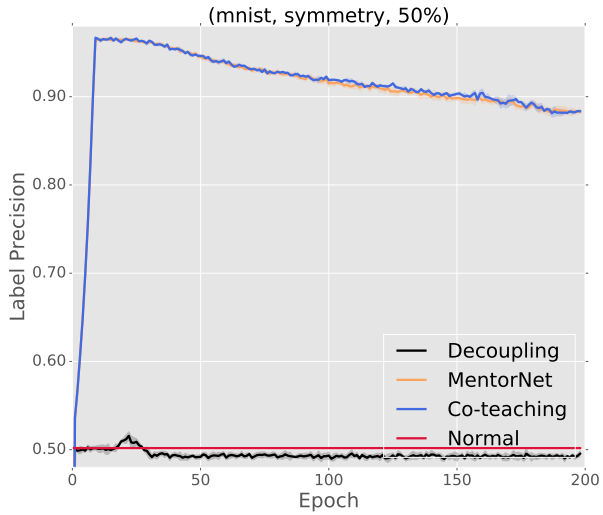
(c) symmetry, 20%.

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

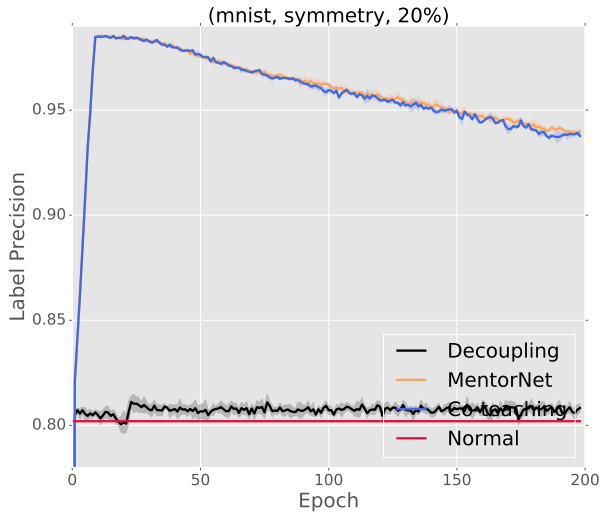
MNIST III



(a) pair, 45%.



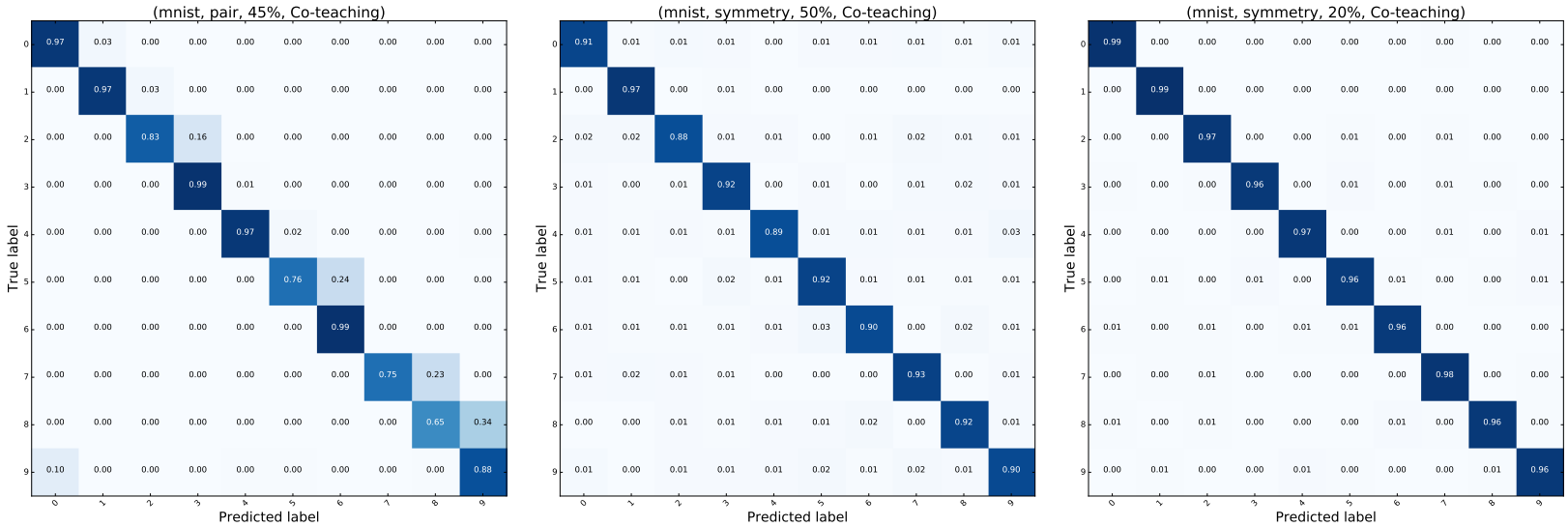
(b) symmetry, 50%.



(c) symmetry, 20%.

Figure : Label precision vs number of epochs on *MNIST* dataset.

MNIST IV



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

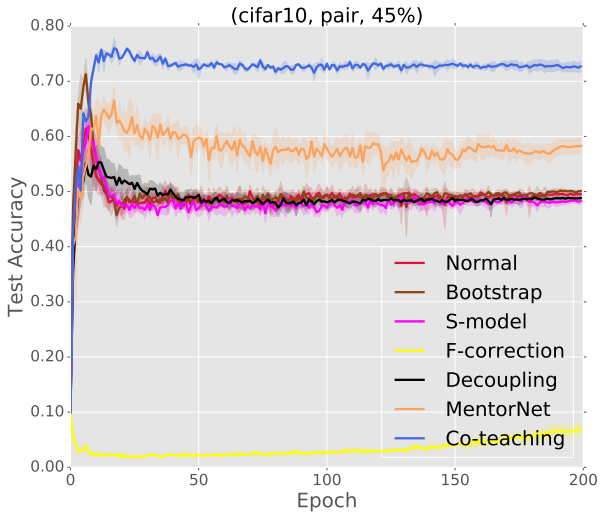
Figure : Confusion matrix of co-teaching on *MNIST* dataset.

CIFAR10 I

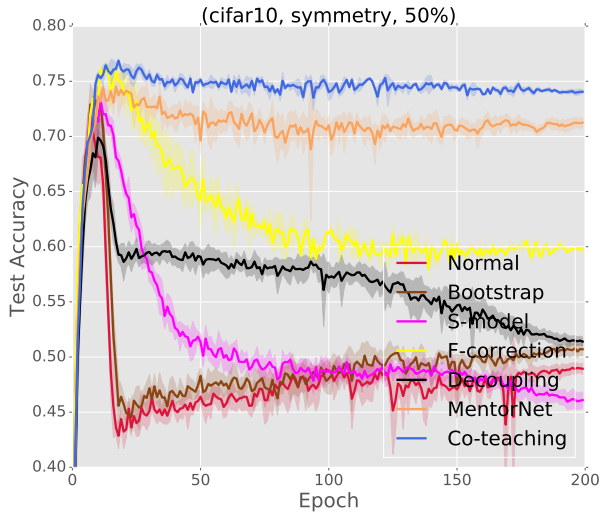
Table : Average test accuracy on *CIFAR10* over the last ten epoch.

(Flipping, Rate)	Normal	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%

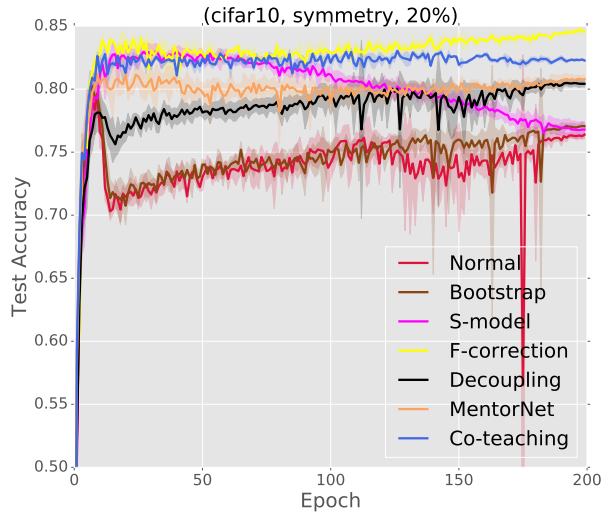
CIFAR10 II



(a) pair, 45%.



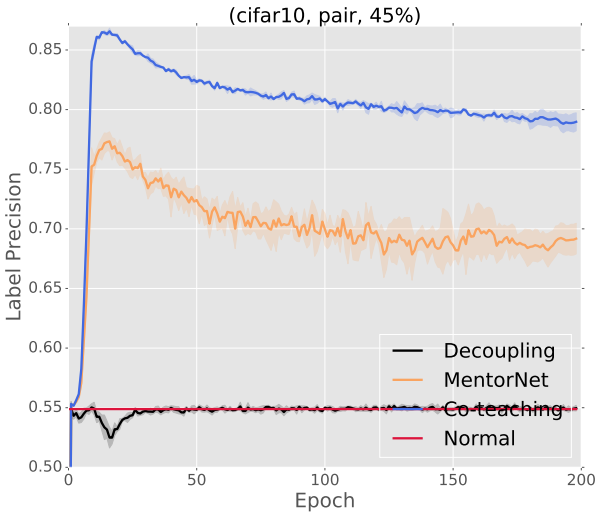
(b) symmetry, 50%.



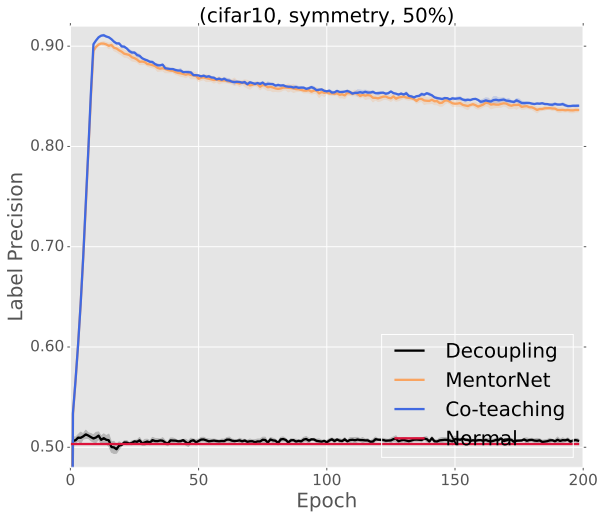
(c) symmetry, 20%.

Figure : Test accuracy vs number of epochs on *CIFAR10* dataset.

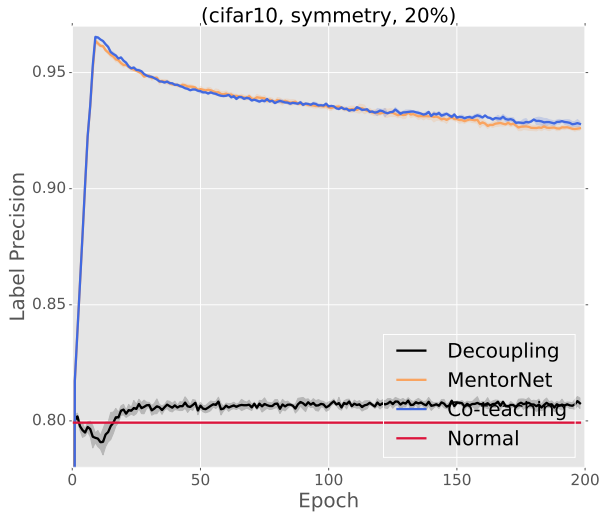
CIFAR10 III



(a) pair, 45%.



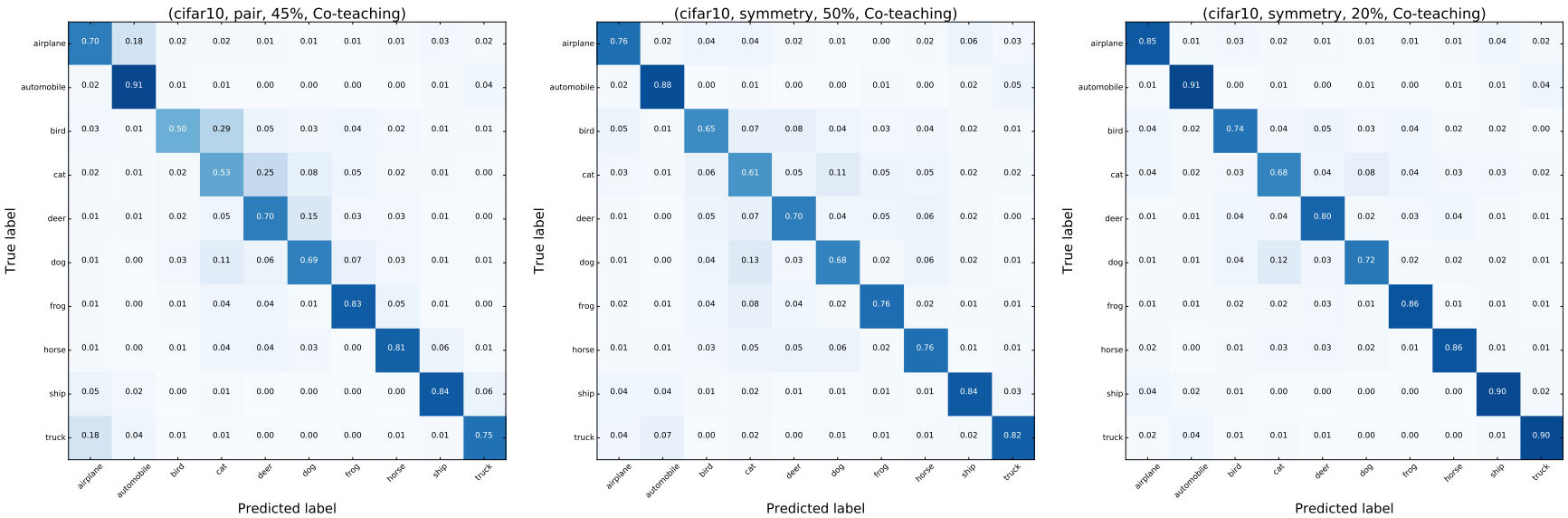
(b) symmetry, 50%.



(c) symmetry, 20%.

Figure : Label precision vs number of epochs on *CIFAR10* dataset.

CIFAR10 IV



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

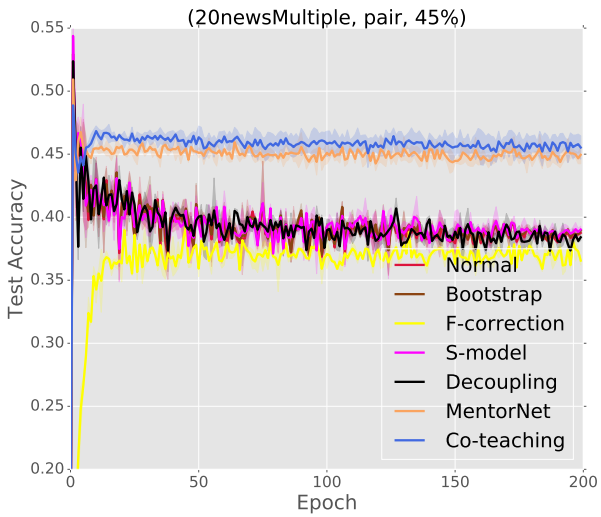
Figure : Confusion matrix of co-teaching on *CIFAR10* dataset.

Newsgroup I

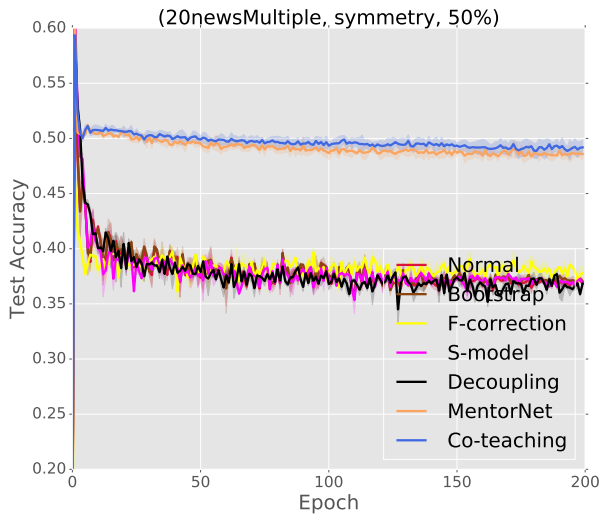
Table : Average test accuracy on *Newsgroup* over the last ten epoch.

(Flipping, Rate)	Normal	Bootstrap	S-model	F-correction	Decoupling	MentorNet	Co-teaching
(Pair, 45%)	38.53% ±0.22%	38.67% ±0.15%	38.80% ±0.37%	37.00% ±0.38%	38.17% ±0.16%	44.90% ±0.51%	45.66% ±0.25%
(Symmetry, 50%)	36.92% ±0.22%	37.21% ±0.28%	36.97% ±0.22%	37.92% ±0.37%	36.45% ±0.31%	48.61% ±0.44%	49.10% ±0.12%
(Symmetry, 20%)	57.20% ±0.20%	57.42% ±0.28%	57.49% ±0.21%	59.13% ±0.26%	56.74% ±0.12%	64.37% ±0.20%	64.64% ±0.07%

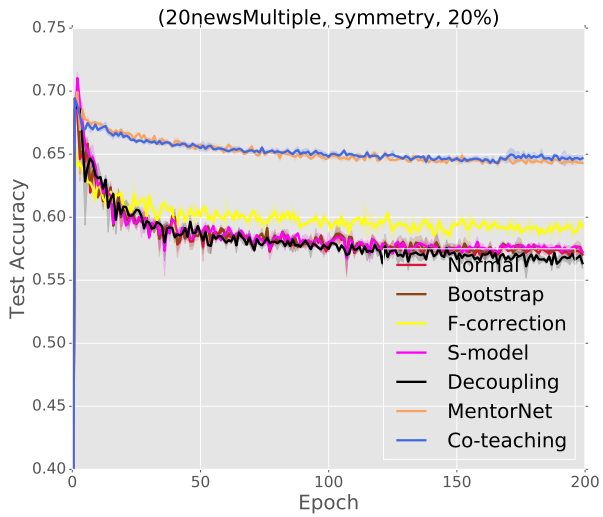
Newsgroup II



(a) pair, 45%.



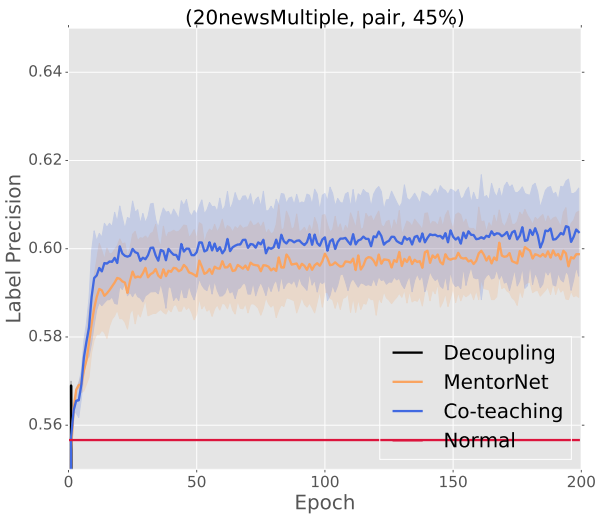
(b) symmetry, 50%.



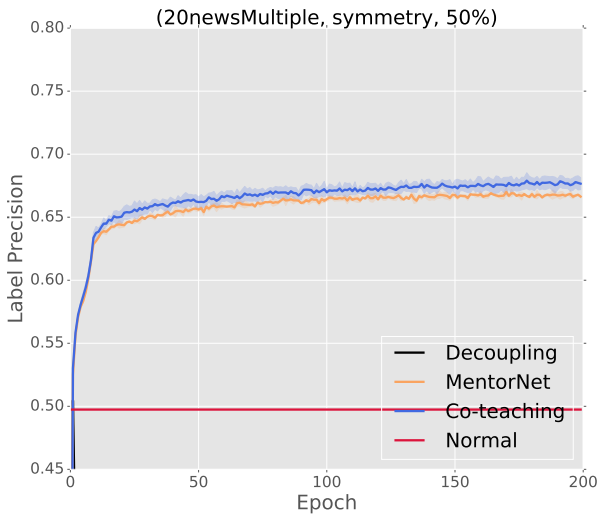
(c) symmetry, 20%.

Figure : Test accuracy vs number of epochs on *Newsgroup* dataset.

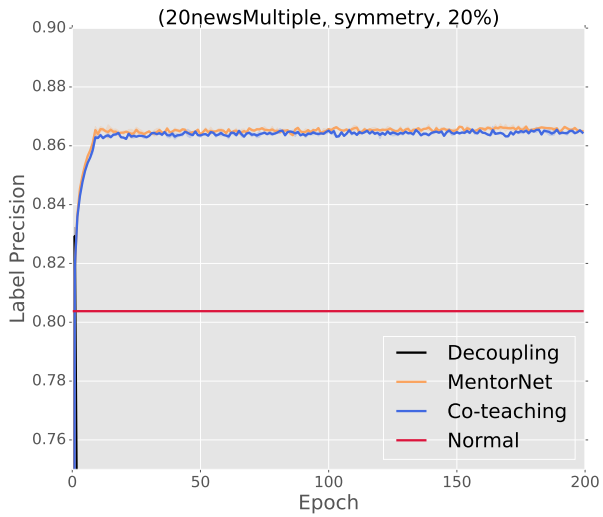
Newsgroup III



(a) pair, 45%.



(b) symmetry, 50%.



(c) symmetry, 20%.

Figure : Label precision vs number of epochs on *Newsgroup* dataset.

Choice of $R(T)$ I

- $R(T) = 1 - \tau \cdot \min\left\{\frac{T^c}{T_k}, 1\right\}$, three choices of c should be considered, $c = \{0.5, 1, 2\}$. We use $c = 1$ in previous experiments.
- three values of T_k should be considered, i.e., $T_k = \{5, 10, 15\}$. We fix $T_k = 10$ in previous experiments.

Choice of $R(T)$ II

Table : Average test accuracy on *MNIST* - (Pair, 45%).

	$c = 0.5$	$c = 1$	$c = 2$
$T_k = 5$	75.56%±0.33%	87.59%±0.26%	87.54%±0.23%
$T_k = 10$	88.43%±0.25%	87.56%±0.12%	87.93%±0.21%
$T_k = 15$	88.37%±0.09%	87.29%±0.15%	88.09%±0.17%

Table : Average test accuracy on *MNIST* - (Symmetry, 50%).

	$c = 0.5$	$c = 1$	$c = 2$
$T_k = 5$	91.75%±0.13%	91.75%±0.12%	92.20%±0.14%
$T_k = 10$	91.70%±0.21%	91.55%±0.08%	91.27%±0.13%
$T_k = 15$	91.74%±0.14%	91.20%±0.11%	91.38%±0.08%

Table : Average test accuracy on *MNIST* - (Symmetry, 20%).

	$c = 0.5$	$c = 1$	$c = 2$
$T_k = 5$	97.05%±0.06%	97.10%±0.06%	97.41%±0.08%
$T_k = 10$	97.33%±0.05%	96.97%±0.07%	97.48%±0.08%
$T_k = 15$	97.41%±0.06%	97.25%±0.09%	97.51%±0.05%

Choice of τ

Table : Average test accuracy of Co-teaching with different τ on *MNIST*.

(Flipping, Rate)	$0.5\tau^*$	$0.75\tau^*$	τ^*	$1.25\tau^*$	$1.5\tau^*$
(Pair, 45%)	66.74% $\pm$0.28%	77.86% $\pm$0.47%	87.63% $\pm$0.21%	97.89% $\pm$0.06%	69.47% $\pm$0.02%
(Symmetry, 50%)	75.89% $\pm$0.21%	82.00% $\pm$0.28%	91.32% $\pm$0.06%	98.62% $\pm$0.05%	79.43% $\pm$0.02%
(Symmetry, 20%)	94.94% $\pm$0.09%	96.25% $\pm$0.06%	97.25% $\pm$0.03%	98.90% $\pm$0.03%	99.39% $\pm$0.02%

Future

Open questions:

- Can we provide theoretical guarantees for Co-teaching?
- Do more peer networks boost the performance? e.g., Tri-teaching?

AI landing possibilities:

Figure : Financial application in 4Paradigm Inc.

Designing Implicit Regularization

- Main idea: leverage regularization bias to combat with noisy labels
- Benefit: easy to implement & free of assumptions
- Drawback: without theoretical guarantees

Regularization Perspective

- The idea of Pumpout is to actively squeeze out the negative effects of noisy labels from the training model.
- On clean labels, Pumpout conducts stochastic gradient descent.
- On noisy labels, Pumpout conducts scaled stochastic gradient ascent, instead of stopping gradient computation as usual.

Figure : How to drain the negative effects of noisy labels from training models? Yes, Pumpout them actively!

Pumpout Paradigm

Input: network parameter w_f, learning rate η, maximum epoch $T_{\max}$, hyper parameter $0 \leq \gamma \leq 1$;
for $t = 1, 2, \ldots, T_{\max}$ do
  Shuffle: training set $\mathcal{D}$; //noisy dataset
  for $i = 1, \ldots, |\mathcal{D}|$ do
    Select: $\{x_i, y_i\}$ from $\mathcal{D}$ sequentially;
    if $\{x_i, y_i\}$ *is fitting* then
      Update: $w_f = w_f - \eta \nabla f(x_i, y_i)$; //gradient descent
    end
    else
      Update: $w_f = w_f + \gamma\eta \nabla f(x_i, y_i)$; //scaled gradient ascent
    end
  end
end
Output: w_f.

Algorithm 2: Meta Algorithm Pumpout.

Three Questions

Q1. What is the fitting condition?

A1. if $\{x_i, y_i\}$ satisfies a fitting condition, fitting on this point will benefit training the robust model.

Q2. Why do we need gradient ascent on non-fitting data?

A2. We hope to actively squeeze out the negative effects from the training model.

Q3. Why do we scale the stochastic gradient ascent on non-fitting data?

A3. When $\gamma = 1$, the fast squeezing rate will negatively affect the convergence of our algorithm; while when $\gamma = 0$, our Pumpout will not squeeze out any negative effects.

Network Structures

Table : 9-layer CNN used in our experiments on *MNIST*.

CNN on MNIST
28×28 Gray 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

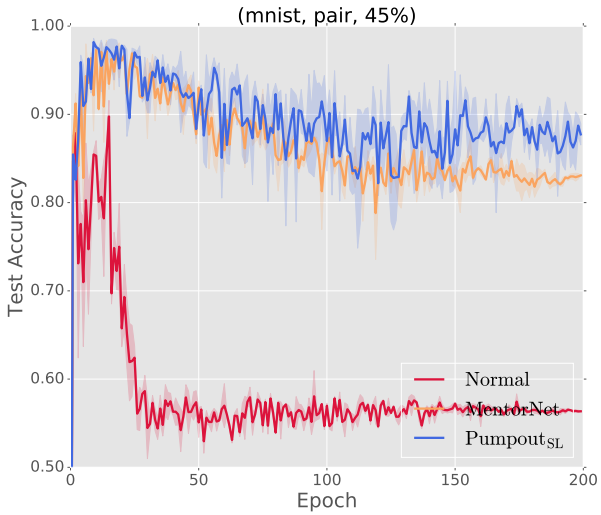
For CIFAR10, 32×32 RGB image, the structure is ResNet-32.

Realization 1: Upgraded MentorNet

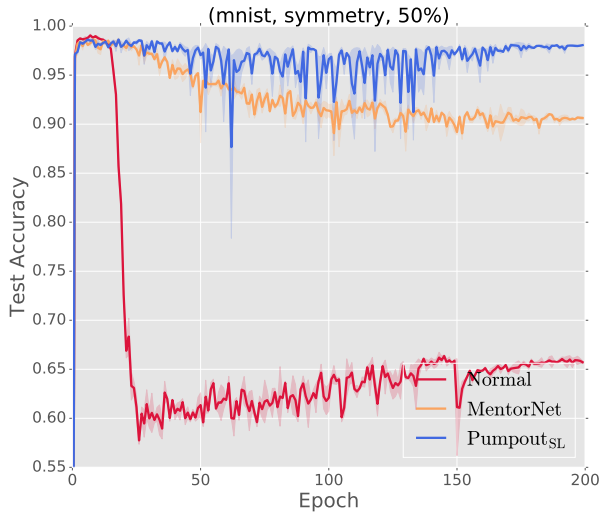
Input: network para w_f, learning rate $\eta > 0$, estimated rate τ, epoch $T_{\max}$, iter $N_{\max}$, hyper para $0 \leq \gamma \leq 1$;
for $T = 1, 2, \ldots, T_{\max}$ do
  Shuffle: training set $\mathcal{D}$;
  for $N = 1, \ldots, N_{\max}$ do
    Draw: mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
    Sample: $\bar{\mathcal{D}}_{\mathrm{s}} = \arg\min_{\bar{\mathcal{D}}} \ell(f, \bar{\mathcal{D}}, R(T))$;
    Sample: $\bar{\mathcal{D}}_{\mathrm{b}} = \arg\max_{\bar{\mathcal{D}}} \ell(f, \bar{\mathcal{D}}, 1 - R(T))$;
    Update: $w_f = w_f - \eta \nabla f(\bar{\mathcal{D}}_{\mathrm{s}})$; //update w_f on $\bar{\mathcal{D}}_{\mathrm{s}}$;
    Update: $w_f = w_f + \gamma\eta \nabla f(\bar{\mathcal{D}}_{\mathrm{b}})$; //update w_f on $\bar{\mathcal{D}}_{\mathrm{b}}$;
  end
  Update: $R(T) = 1 - \min\left\{\frac{T}{T_k}\tau, \tau\right\}$;
end
Output: w_f.

Algorithm 3: Pumpout for MentorNet. The fitting condition is whether a point belongs to small-loss instances.

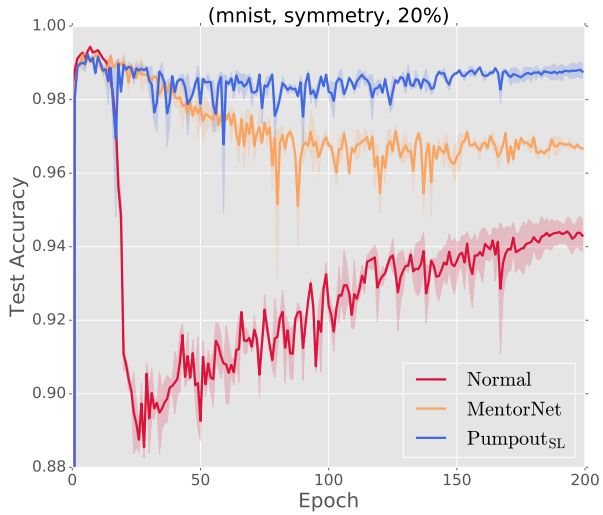
MNIST I



(a) pair, 45%.



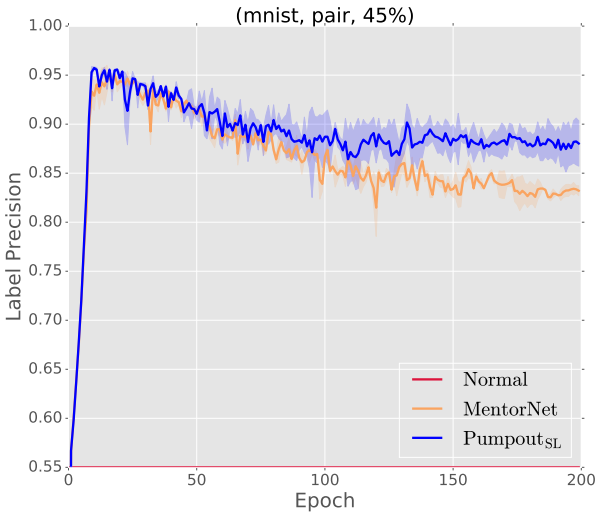
(b) symmetry, 50%.



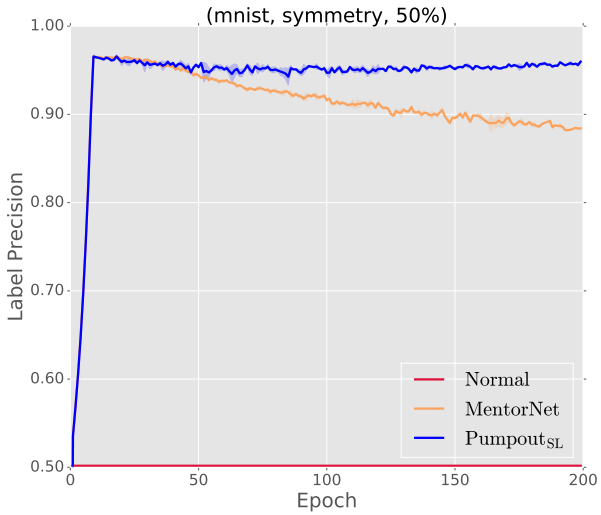
(c) symmetry, 20%.

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

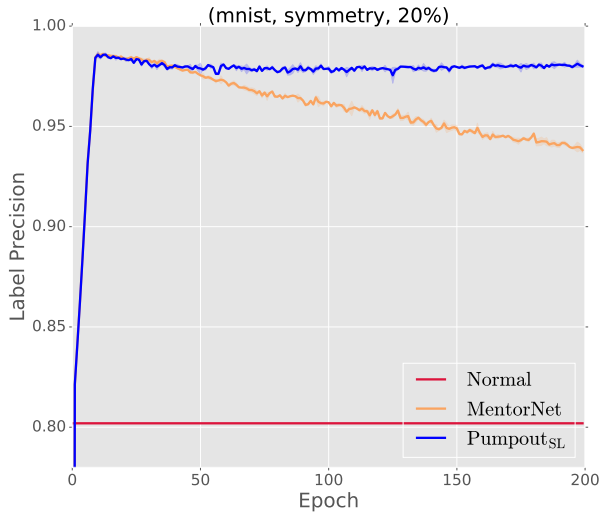
MNIST II



(a) pair, 45%.



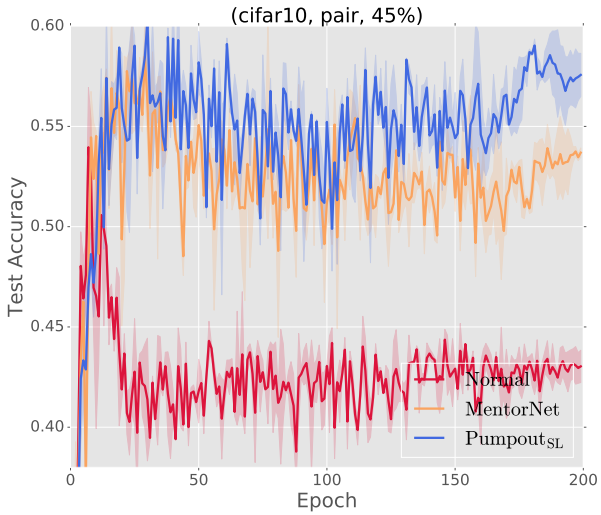
(b) symmetry, 50%.



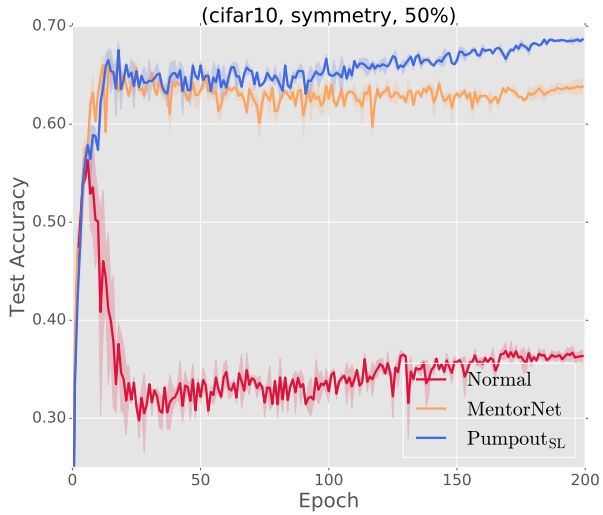
(c) symmetry, 20%.

Figure : Label precision vs number of epochs on *MNIST* dataset.

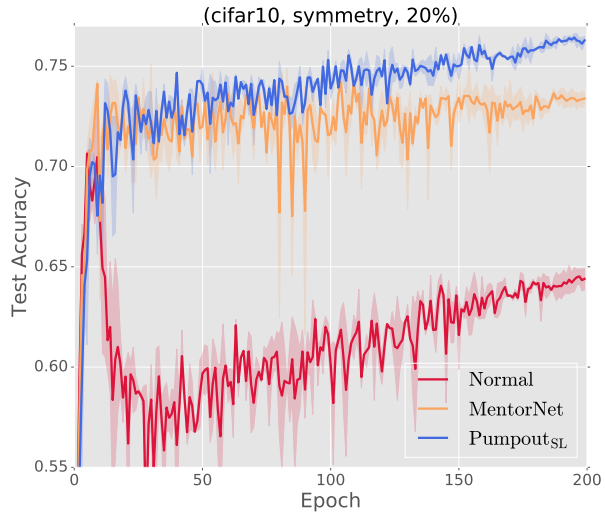
CIFAR10 I



(a) pair, 45%.



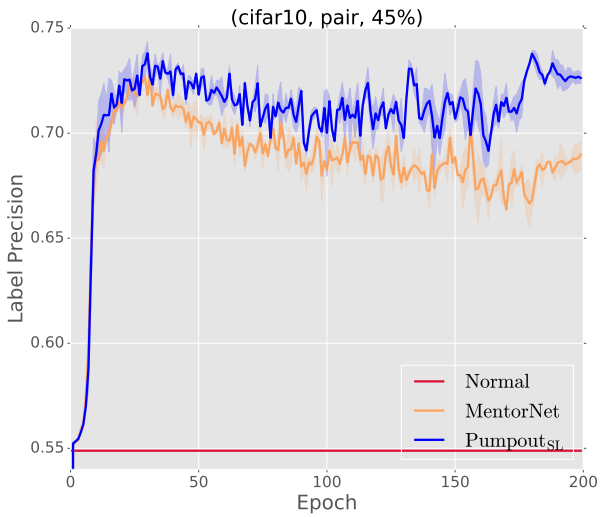
(b) symmetry, 50%.



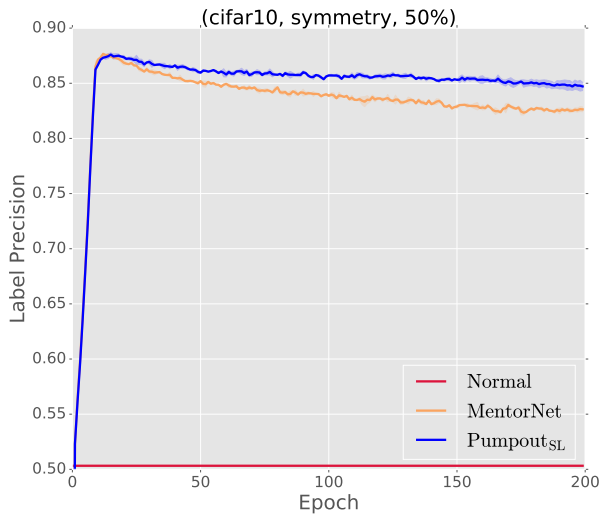
(c) symmetry, 20%.

Figure : Test accuracy vs number of epochs on *CIFAR10* dataset.

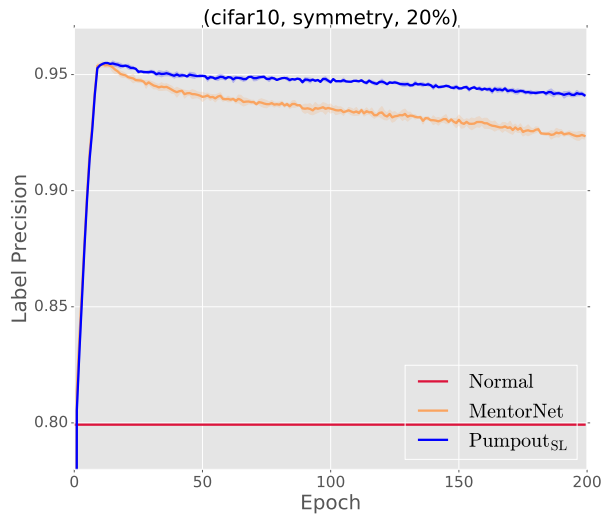
CIFAR10 II



(a) pair, 45%.



(b) symmetry, 50%.



(c) symmetry, 20%.

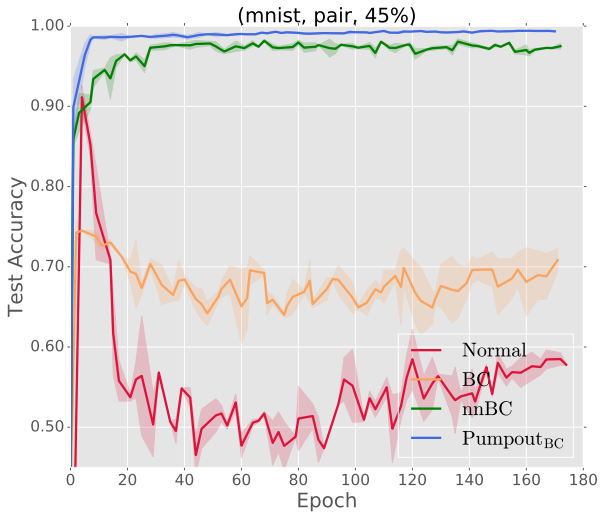
Figure : Label precision vs number of epochs on *CIFAR10* dataset.

Realization 2: Upgraded Backward Correction

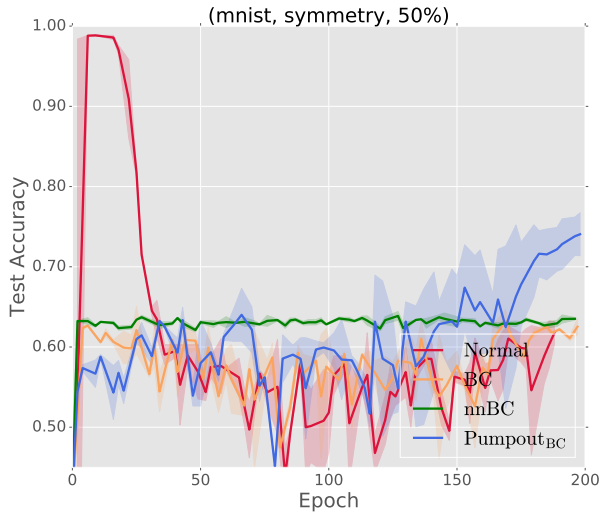
Input: network para w_f, learning rate $\eta > 0$, maximum epoch $T_{\max}$, hyper para $\beta \geq 0$ and $0 \leq \gamma \leq 1$;
for $T = 1, 2, \ldots, T_{\max}$ do
  Shuffle: training set $\mathcal{D}$ into n-mini batches with batch size k;
  for $j = 1, \ldots, n$ do
    Reset: $G_a = 0$;
    Draw: j-th mini-batch $\bar{\mathcal{D}}$ from $\mathcal{D}$;
    for $i = 1, \ldots, k$ do
      Select: $\{\mathrm{x}_i, y_i\}$ from $\bar{\mathcal{D}}$ as i-th data point;
      Set: $g_t = \nabla_{w_f}\{1^\top \mathrm{T}^{-1}\ell(\mathrm{x}_i, y_i; w_f)\}$;
      if $1^\top \mathrm{T}^{-1}\ell(\mathrm{x}_i, y_i; w_f) \geq \beta$ then
        Update: $G_a = G_a + g_t$; //gradient descent
      end
      else
        Update: $G_a = G_a - \gamma g_t$; //scaled gradient ascent
      end
    end
    Average: $g_a = G_a/k$;
    Update: $w_f = w_f - \eta g_a$;
  end
end
Output: w_f.

Algorithm 4: Pumpout for Backward Correction. The fitting condition is whether a point satisfies $1^\top \mathrm{T}^{-1}\ell(\mathrm{x}_i, y_i; w_f) \geq \beta$.

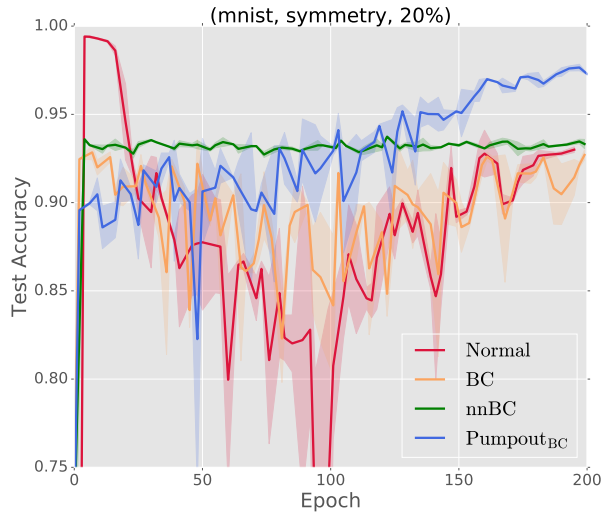
MNIST I



(a) pair, 45%.



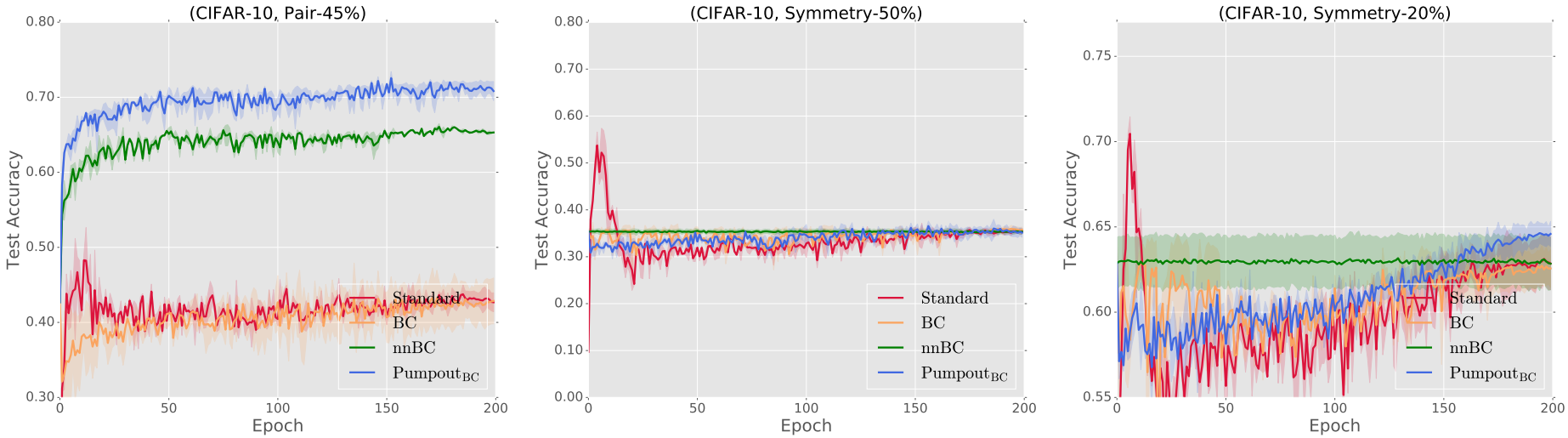
(b) symmetry, 50%.



(c) symmetry, 20%.

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

CIFAR10



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

Figure : Test accuracy vs number of epochs on *CIFAR10* dataset.

Future

Open questions:

- Can we provide theoretical guarantees for Pumpout?
- Can we apply Pumpout to another weak supervision?

AI landing possibilities:

- We hope to attract AI companies using our Pumpout to handle their agnostic noisy data.

References

- Masking: A New Perspective of Noisy Supervision. *NeurIPS*, 2018.
- Co-teaching: Robust Training of Deep Neural Networks with Extremely Noisy Labels. *NeurIPS*, 2018.
- Pumpout: A Meta Approach for Robustly Training Deep Neural Networks with Noisy Labels. *arXiv*, 2018.



Thanks Co-authors and Grants

- Dr. Gang Niu: Research Scientist in RIKEN-AIP.
- Dr. Mingyuan Zhou: A/Professor in University of Texas Austin.
- Dr. Quanming Yao: Research Scientist in 4Paradigm Inc.
- Dr. Miao Xu: Postdoctoral Researcher in RIKEN-AIP.
- Mr. Jiangchao Yao: PhD in University of Technology Sydney.
- Mr. Xingrui Yu: PhD in University of Technology Sydney.
- Mr. Weihua Hu: PhD in Stanford University.
- Dr. Ivor W. Tsang: Professor in University of Technology Sydney.
- Dr. Masashi Sugiyama: Professor in RIKEN-AIP/University of Tokyo.

These works were supported by WPI-IRCN at The University of Tokyo Institutes for Advanced Study, JST CREST JPMJCR1403, ARC FT130100746, DP180100106 and LP150100671.

RIKEN-AIP

- RIKEN Center for Advanced Intelligence Project (RIKEN-AIP) is a national research center founded in 2016 in Japan. RIKEN-AIP has memorandum of understanding with many world-class research institutes, such as MILA, Vector and CMU.

Figure : RIKEN-AIP.

REsearch CO-supervision Program (bo.han@riken.jp)

Research Interests:

- Deep learning under weak supervision (2015–)
- Adversarial machine learning (2018–)

Requirements:

- Strong motivation (willing to catch multiple dues)
- Exceptional coding ability (Pytorch or Tensorflow)
- Tough and easygoing

Offers:

- Half-baked idea sharing (with story)
- 1 vs 1 meeting (once or twice a week)
- Paper writing and revising (each paper 100+)