Asymmetric tri-training for Unsupervised Domain Adaptation

ICML 2017

Kuniaki Saito[1], Yoshitaka Ushiku[1] and Tatsuya Harada[1,2]

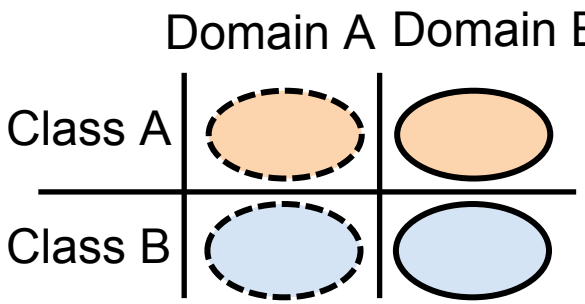
1.The University of Tokyo, 2.RIKEN

Outline

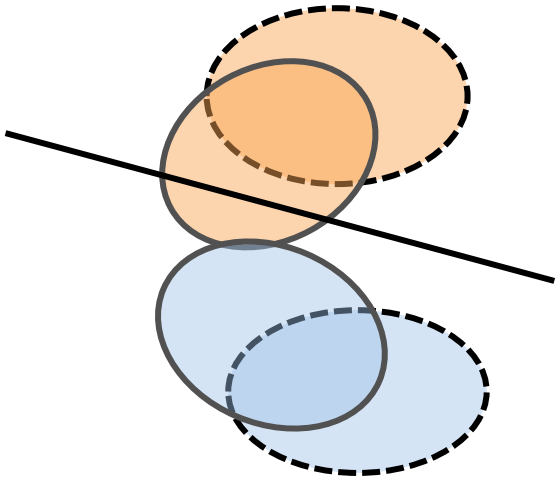
- Background
- Theoretical insight (Motivation)
- Proposed Method
  - Brief overview
  - Network, Training Procedure, Objective
- Experiments
- Summary

Background

- Supervised learning using a lot of samples
- Cost to collect samples in various domains
- Classifiers suffer from domain-shift

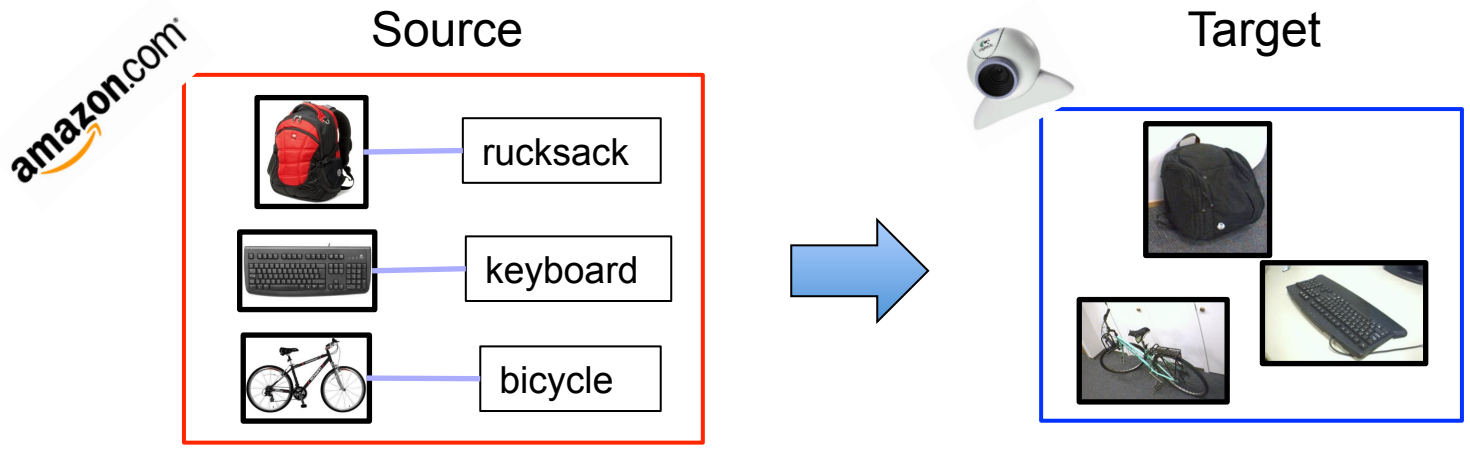


Classifier suffers from domain difference



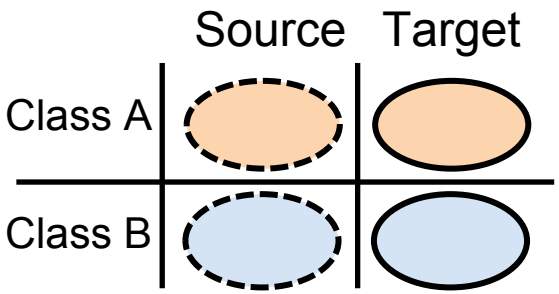
Domain Adaptation (DA)

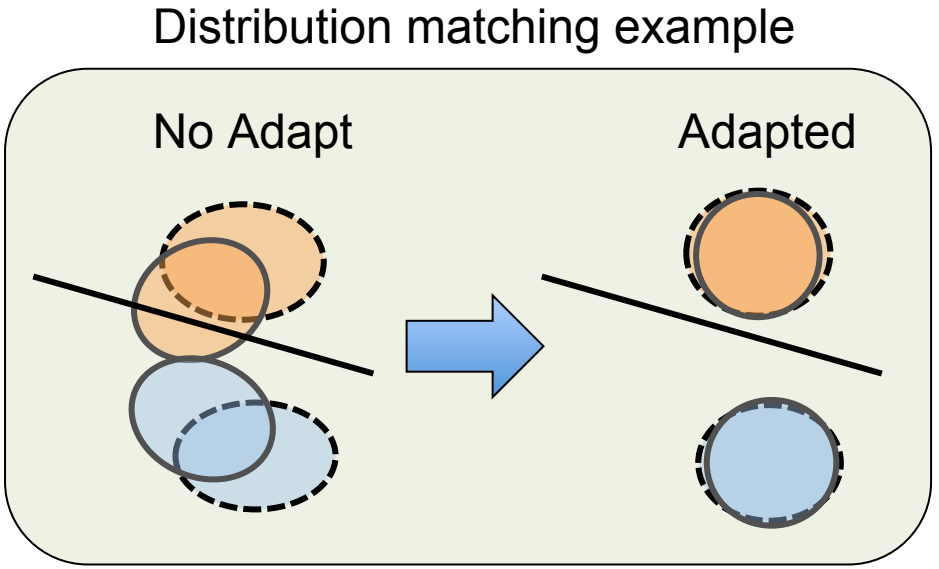
- The purpose of DA
  - Transfer knowledge from **source** domain to **target** domain
  - Purpose
    - Train a classifier that works well on **target** domain.
- Unsupervised Domain Adaptation
  - Labeled source samples and **unlabeled** target samples



Related Work

- Domain adaptation in deep learning
  - Matching hidden features of different domains
    - MMD [Long et al., ICML 2015]
    - Domain Classifier [Ganin et al., ICML 2014]





Theoretical Insight

Theorem [Ben David et al., 2010]

① ② ③

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

Theoretical Insight

Theorem [Ben David et al., 2010]

① ② ③

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

①・・・Error on source domain

Theoretical Insight

Theorem [Ben David et al., 2010]

① ② ③

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

①・・・Error on source domain
②・・・Divergence between domains

Theoretical Insight

Theorem [Ben David et al., 2010]

① ② ③

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

①・・・Error on source domain
②・・・Divergence between domains
③・・・How much features are discriminative

Theoretical Insight

Theorem [Ben David et al., 2010]

① ② ③

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

①・・・Error on source domain
②・・・Divergence between domains
③・・・How much features are discriminative

・Distribution matching approaches aim to minimize ① and ②.
・They regard ③ as sufficiently small.

Theoretical Insight

Theorem [Ben David et al., 2010]

① ② ③

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

①・・・Error on source domain
②・・・Divergence between domains
③・・・How much features are discriminative

There is no guarantee that ③ is small enough.
Consider this term in our method!!

Theoretical Insight

Theorem [Ben David et al., 2010]

$$R_{\mathcal{T}}(h) \leq \underbrace{R_{\mathcal{S}}(h)}_{①} + \underbrace{d_{\mathcal{H}}(\mathcal{S},\mathcal{T})}_{②} + \underbrace{\lambda}_{③}$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

Problem: No labels are provided for target ...

Theoretical Insight

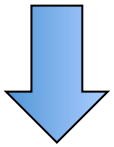
Theorem [Ben David et al., 2010]

① ② ③

$$R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + d_{\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

$$\lambda = \min_{h} \left[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h) \right]$$

Problem: No labels are provided for target ...



Attach pseudo-labels to target, use them for training !

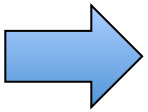
Requirements and Solutions

Requirements and Solutions

- Accurate pseudo labels

Requirements and Solutions

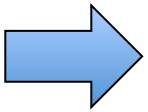
- Accurate pseudo labels



Two classifiers to label target samples

Requirements and Solutions

- Accurate pseudo labels

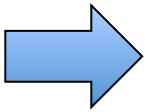


Two classifiers to label target samples

- Discriminative features for target

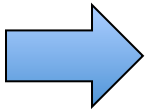
Requirements and Solutions

- Accurate pseudo labels



Two classifiers to label target samples

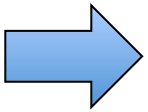
- Discriminative features for target



One classifier to learn mainly from pseudo-labeled samples

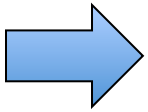
Requirements and Solutions

- Accurate pseudo labels

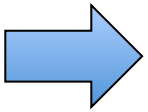


Two classifiers to label target samples

- Discriminative features for target



One classifier to learn mainly from pseudo-labeled samples

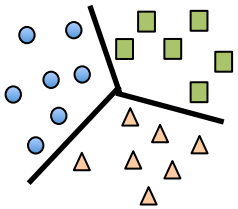


Share low-level features

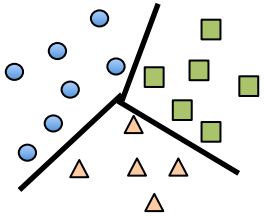
Outline of Proposed Method

Train two classifiers from source samples

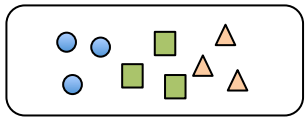
Classifier1



Classifier2



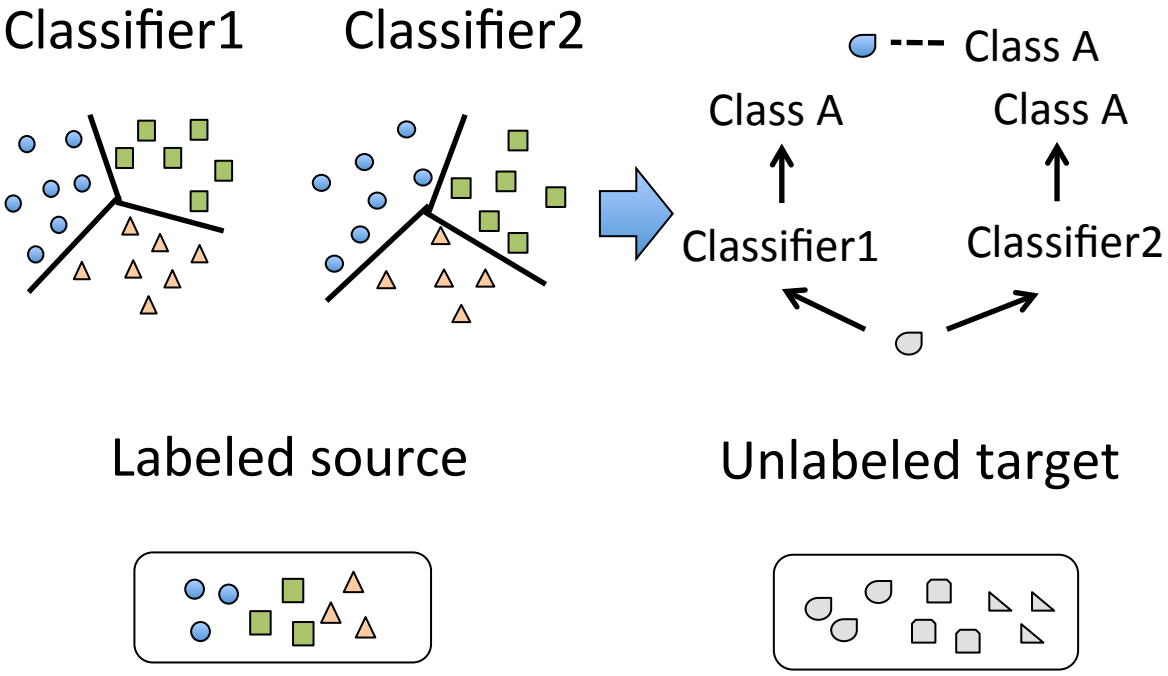
Labeled source



Outline of Proposed Method

Train two classifiers from source samples

Give pseudo labels to target samples

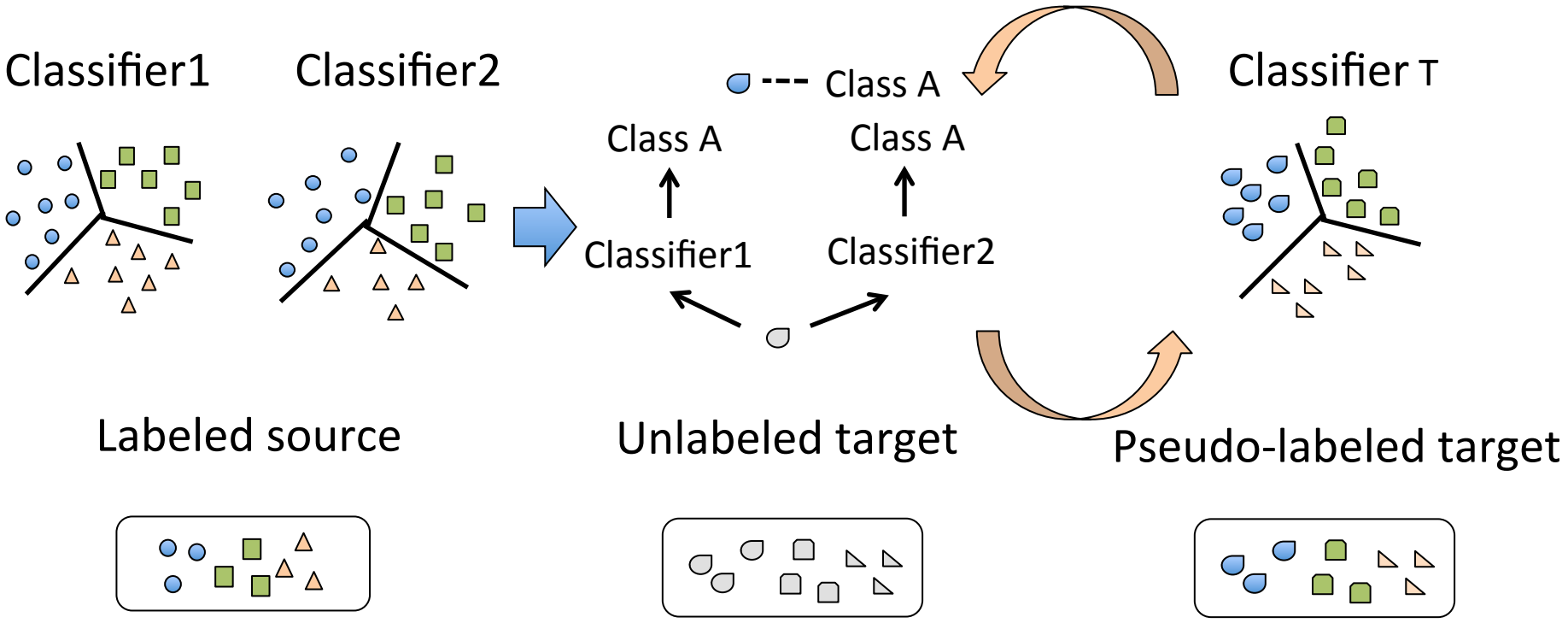


Outline of Proposed Method

Train two classifiers from source samples

Give pseudo labels to target samples

Train target classifier by pseudo-labeled samples



Relationship with Tri-training

- Tri-training [Zhou et al., 2005]
  - Use three classifiers equally
    1, Use two classifiers to give labels to unlabeled samples
    2, Train one classifier by the labeled samples
    3, Repeat in all combination of classifiers
- Our proposed method
  - Use three classifiers asymmetrically
    - Use fixed two classifiers to give labels
    - Train a fixed one classifier by the pseudo-labeled samples

Outline

- Background
- Theoretical insight (Motivation)
- Proposed Method
  - Brief overview
  - Network, Training Procedure, Objective
- Experiments
- Summary

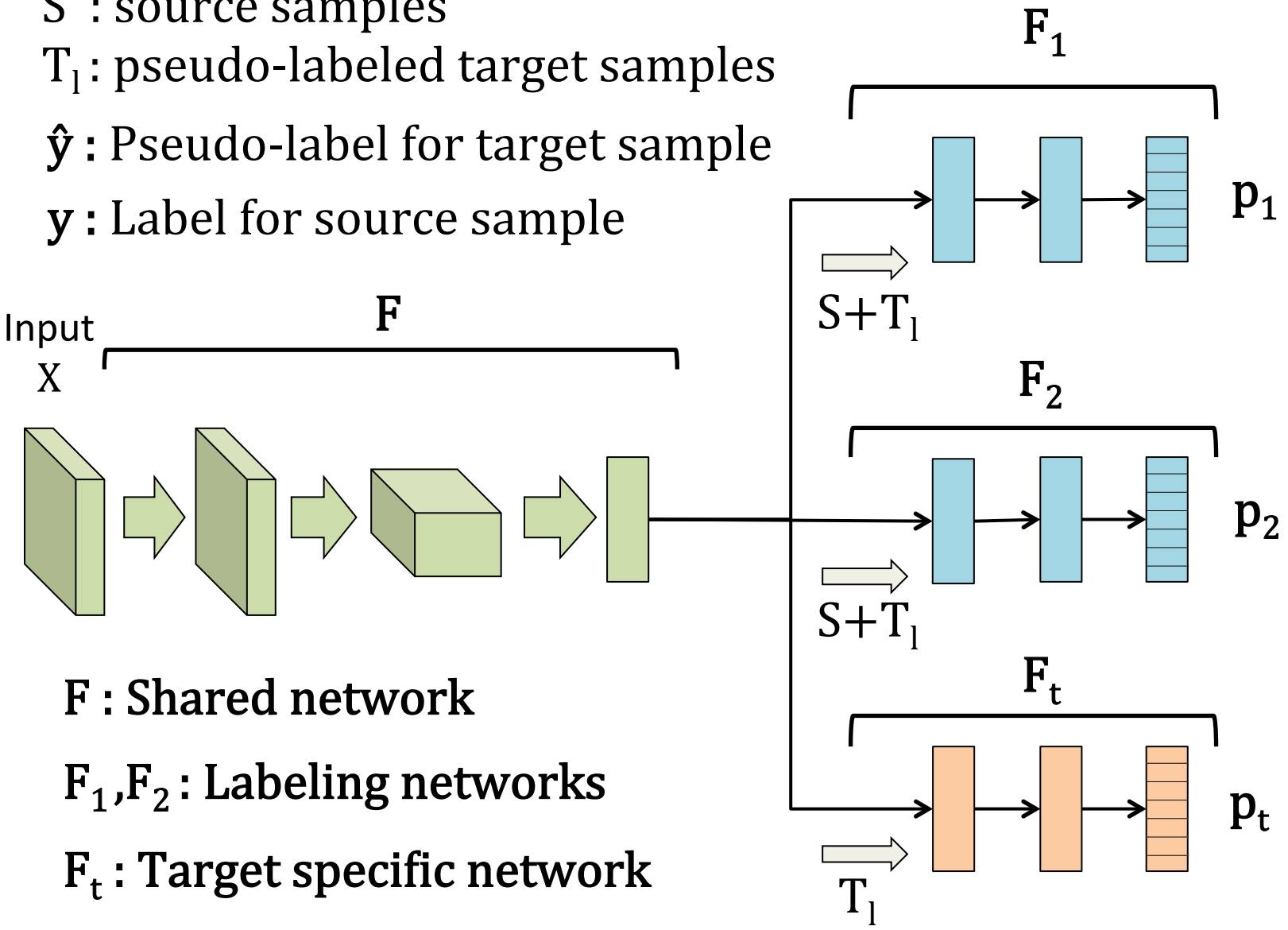
Proposed Architecture

S : source samples

T_l : pseudo-labeled target samples

$\hat{y}$: Pseudo-label for target sample

y : Label for source sample



F : Shared network

F_1,F_2 : Labeling networks

F_t : Target specific network

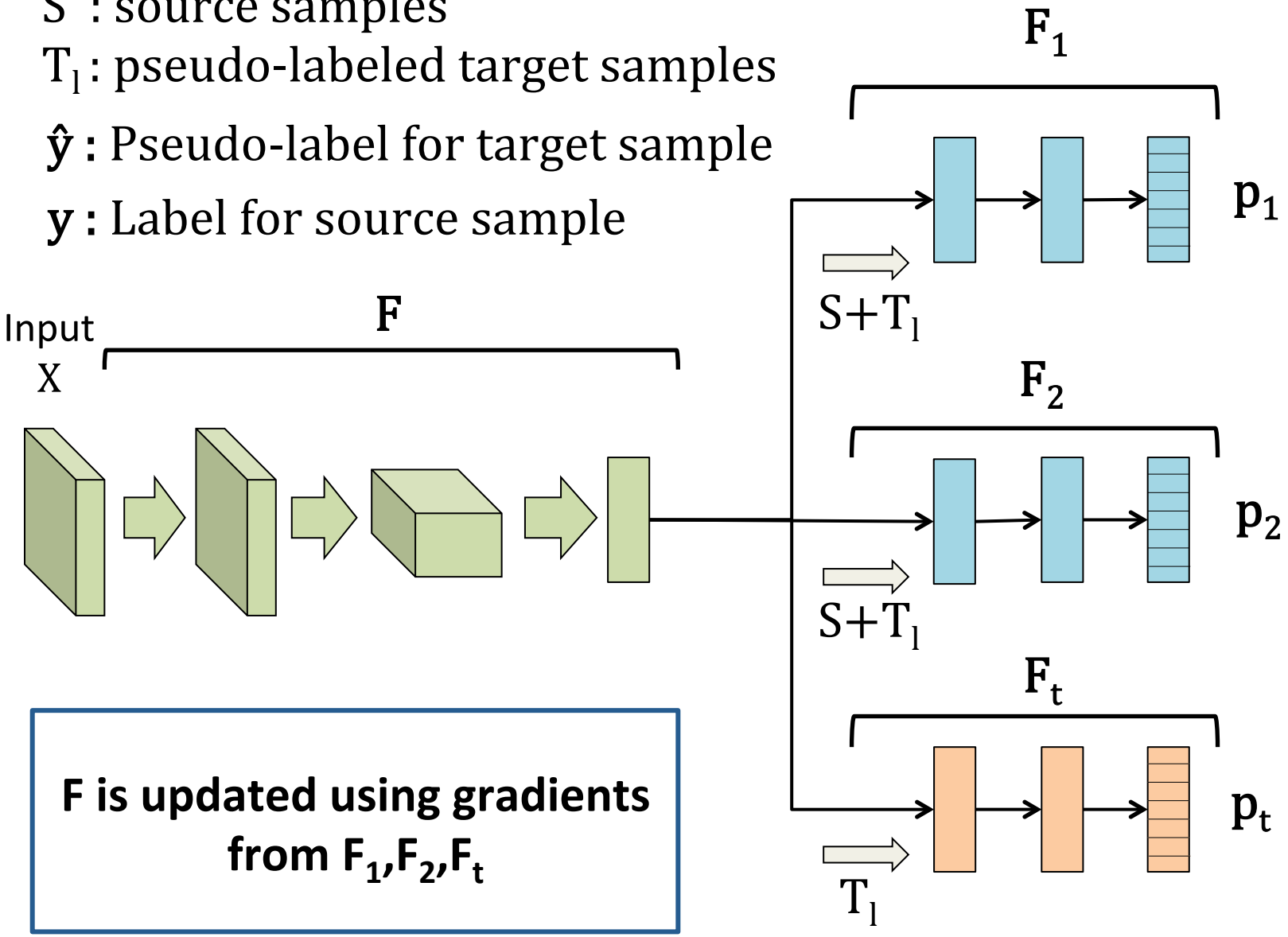
Proposed Architecture

S : source samples

T_l : pseudo-labeled target samples

$\hat{y}$: Pseudo-label for target sample

y : Label for source sample



F is updated using gradients from F_1,F_2,F_t

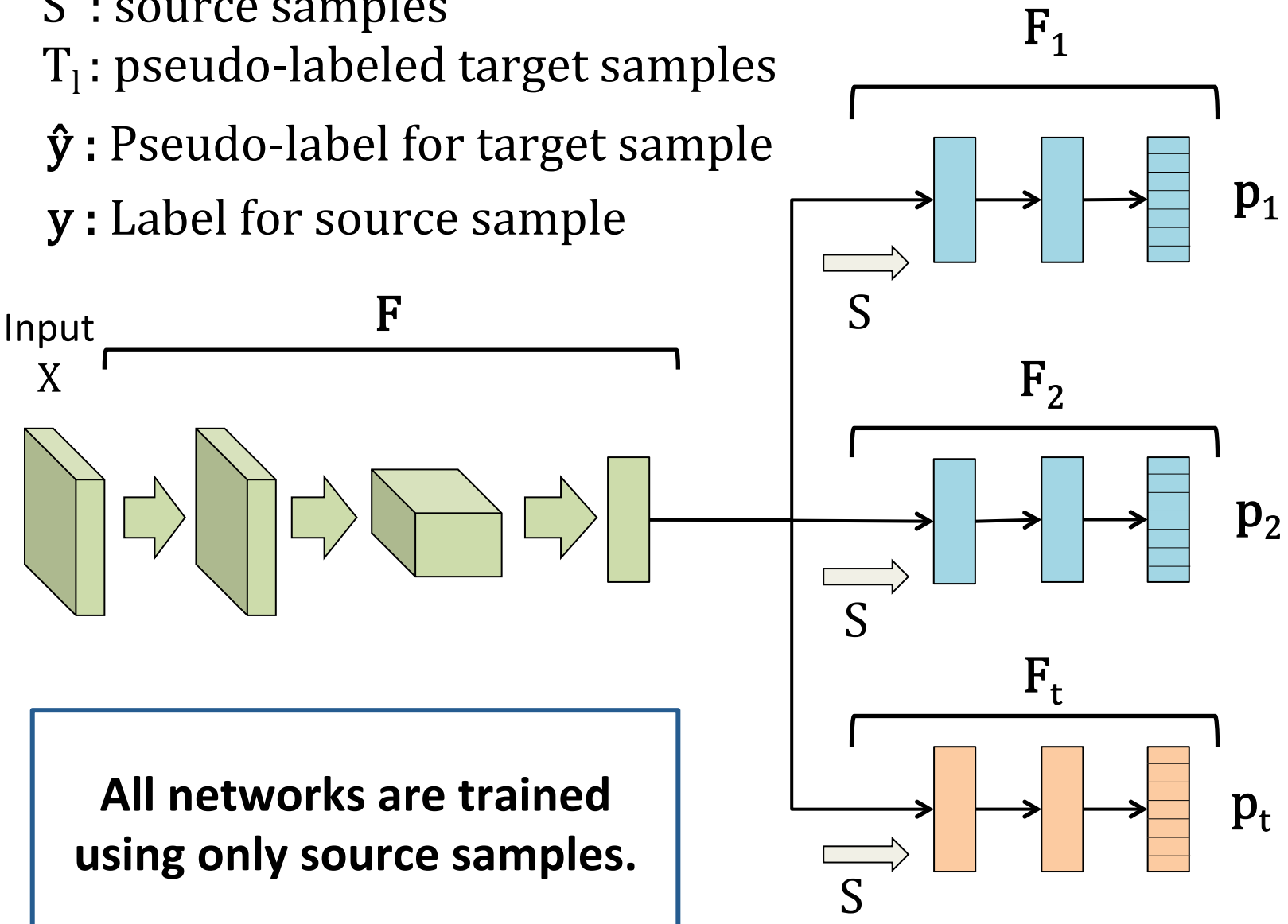
1. Training by source

S : source samples

T_l : pseudo-labeled target samples

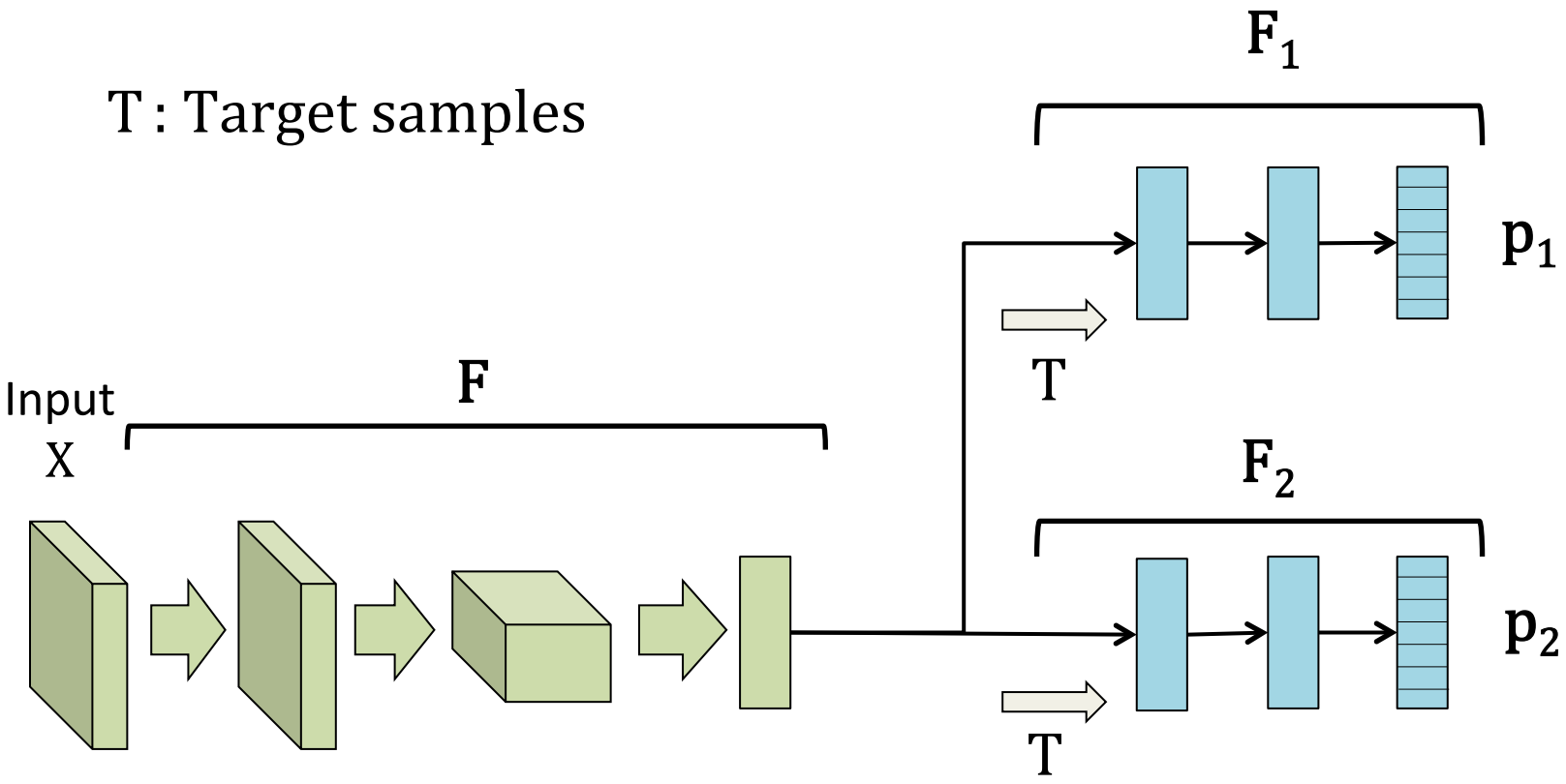
$\hat{\mathbf{y}}$: Pseudo-label for target sample

$\mathbf{y}$: Label for source sample



All networks are trained using only source samples.

2. Labeling target samples



If F_1 and F_2 agree on their predictions, and either of their probability is larger than threshold value, corresponding labels are given to the target sample.

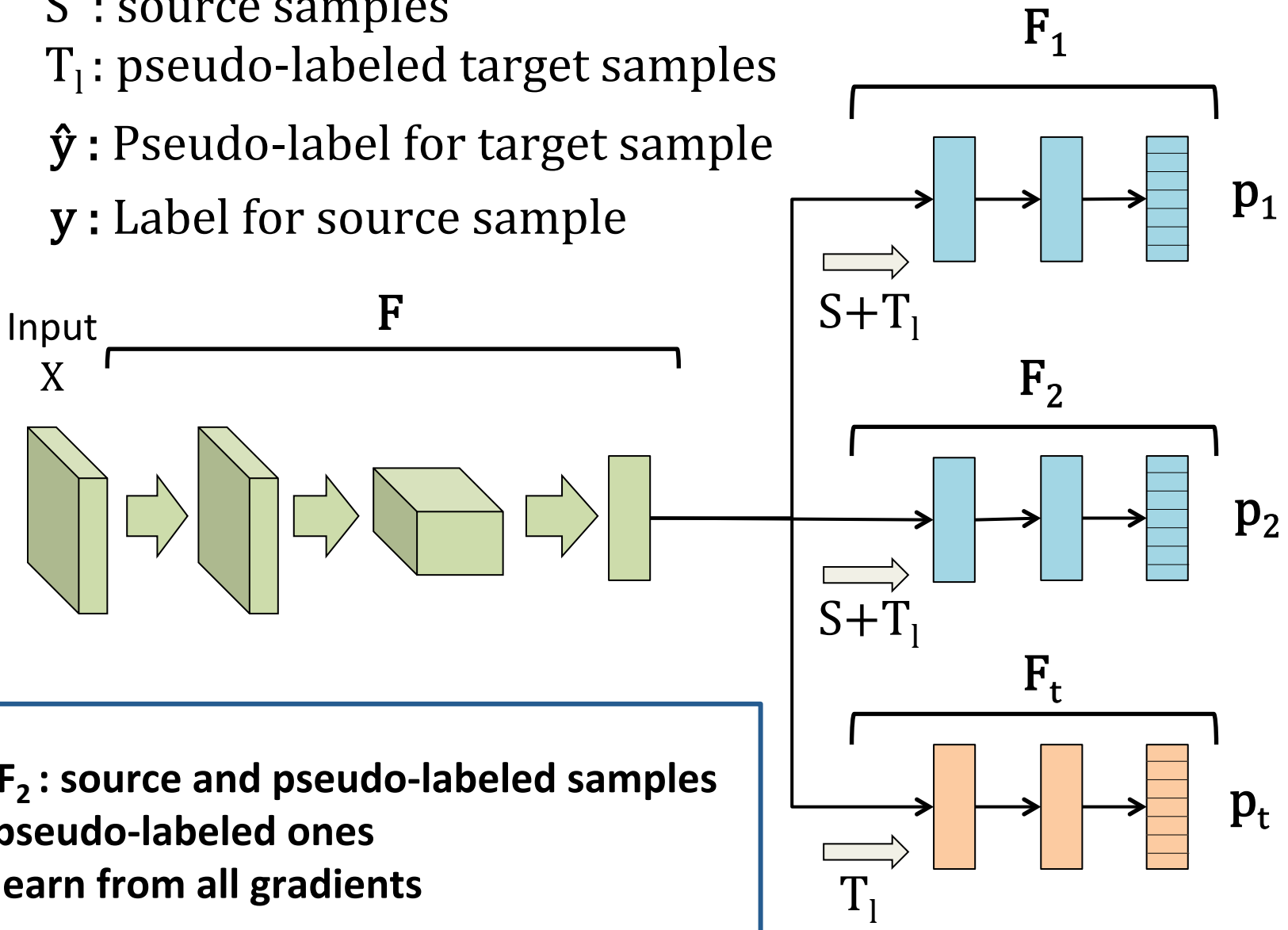
3. Retraining network using pseudo-labeled target samples

S : source samples

T_l : pseudo-labeled target samples

$\hat{y}$: Pseudo-label for target sample

y : Label for source sample

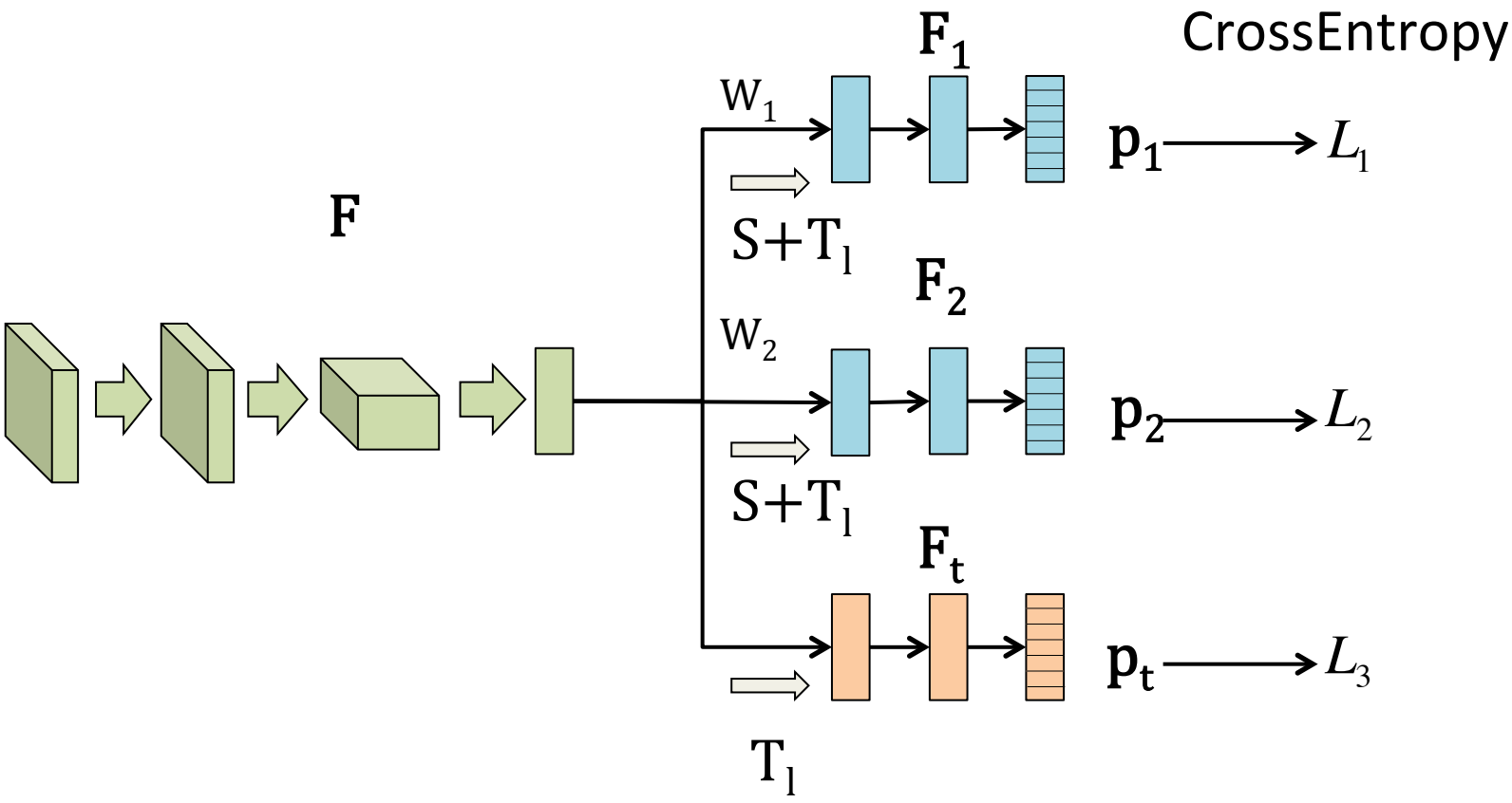


F_1, F_2 : source and pseudo-labeled samples

F_t: pseudo-labeled ones

F : learn from all gradients

Objective



Overall Objective $\lambda_1 | W_1^T W_2 | + L_1 + L_2 + L_3$

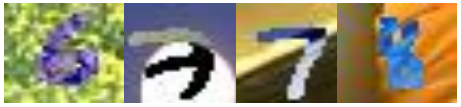
To force F_1 and F_2 to learn from different features.

Experiments

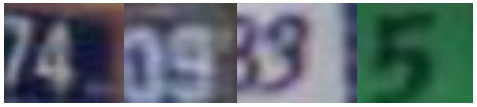
- Four adaptation scenarios between digits datasets
  - MNIST, SVHN, SYN DIGIT (synthesized digits)
- One adaptation scenario between traffic signs datasets
  - GTSRB (real traffic signs), SYN SIGN (synthesized signs)
- Comparison
  - Source only model (w/ BN, w/o BN) without adaptation
  - Other DA methods

MNIST 

SYN SIGNS 

MNIST-M 

GTSRB 

SVHN 

SYN DIGITS 

Accuracy on Target Domain

Source	MNIST	MNIST	SVHN	SYNDIG	SYN NUM
Method / Target	MN-M	SVHN	MNIST	SVHN	GTSRB
Source Only (w/o BN)	59.1	37.2	68.1	84.1	79.2
Source Only (with BN)	57.1	34.9	70.1	85.5	75.7
DANN [Ganin et al., 2014]	81.5	35.7	71.1	90.3	88.7
MMD [Long et al., 2015 ICML]	76.9	-	71.1	88.0	91.1
DSN [Bousmalis et al, 2016 NIPS]	83.2	-	82.7	91.2	93.1
K-NN Labeling [Sener et al., 2016 NIPS]	86.7	40.3	78.8	-	-
Ours (w/o BN)	85.3	39.8	79.8	93.1	96.2
Ours (w/o Weight constraint)	94.2	49.7	86.0	92.4	94.0
Ours	94.0	52.8	86.8	92.9	96.2

- **Our method outperformed other methods.**
- **The effect of BN is obvious in some settings.**
- **The effect of weight constraint is not obvious.**

Accuracy on Target Domain

Method \ Source / Target	MNIST / MN-M	MNIST / SVHN	SVHN / MNIST	SYNDIG / SVHN	SYN NUM / GTSRB
Source Only (w/o BN)	59.1	37.2	68.1	84.1	79.2
Source Only (with BN)	57.1	34.9	70.1	85.5	75.7
DANN [Ganin et al., 2014]	81.5	35.7	71.1	90.3	88.7
MMD [Long et al., 2015 ICML]	76.9	-	71.1	88.0	91.1
DSN [Bousmalis et al, 2016 NIPS]	83.2	-	82.7	91.2	93.1
K-NN Labeling [Sener et al., 2016 NIPS]	86.7	40.3	78.8	-	-
Ours (w/o BN)	85.3	39.8	79.8	93.1	96.2
Ours (w/o Weight constraint)	94.2	49.7	86.0	92.4	94.0
Ours	94.0	52.8	86.8	92.9	96.2

- **Our method outperformed other methods.**
- **The effect of BN is obvious in some settings.**
- **The effect of weight constraint is not obvious.**

Accuracy on Target Domain

Method \ Source / Target	MNIST / MN-M	MNIST / SVHN	SVHN / MNIST	SYNDIG / SVHN	SYN NUM / GTSRB
Source Only (w/o BN)	59.1	37.2	68.1	84.1	79.2
Source Only (with BN)	57.1	34.9	70.1	85.5	75.7
DANN [Ganin et al., 2014]	81.5	35.7	71.1	90.3	88.7
MMD [Long et al., 2015 ICML]	76.9	-	71.1	88.0	91.1
DSN [Bousmalis et al, 2016 NIPS]	83.2	-	82.7	91.2	93.1
K-NN Labeling [Sener et al., 2016 NIPS]	86.7	40.3	78.8	-	-
Ours (w/o BN)	85.3	39.8	79.8	93.1	96.2
Ours (w/o Weight constraint)	94.2	49.7	86.0	92.4	94.0
Ours	94.0	52.8	86.8	92.9	96.2

- **Our method outperformed other methods.**
- **The effect of BN is obvious in some settings.**
- **The effect of weight constraint is not obvious.**

Accuracy on Target Domain

Source	MNIST	MNIST	SVHN	SYNDIG	SYN NUM
Method / Target	MN-M	SVHN	MNIST	SVHN	GTSRB
Source Only (w/o BN)	59.1	37.2	68.1	84.1	79.2
Source Only (with BN)	57.1	34.9	70.1	85.5	75.7
DANN [Ganin et al., 2014]	81.5	35.7	71.1	90.3	88.7
MMD [Long et al., 2015 ICML]	76.9	-	71.1	88.0	91.1
DSN [Bousmalis et al, 2016 NIPS]	83.2	-	82.7	91.2	93.1
K-NN Labeling [Sener et al., 2016 NIPS]	86.7	40.3	78.8	-	-
Ours (w/o BN)	85.3	39.8	79.8	93.1	96.2
Ours (w/o Weight constraint)	94.2	49.7	86.0	92.4	94.0
Ours	94.0	52.8	86.8	92.9	96.2

- **Our method outperformed other methods.**
- **The effect of BN is obvious in some settings.**
- **The effect of weight constraint is not obvious.**

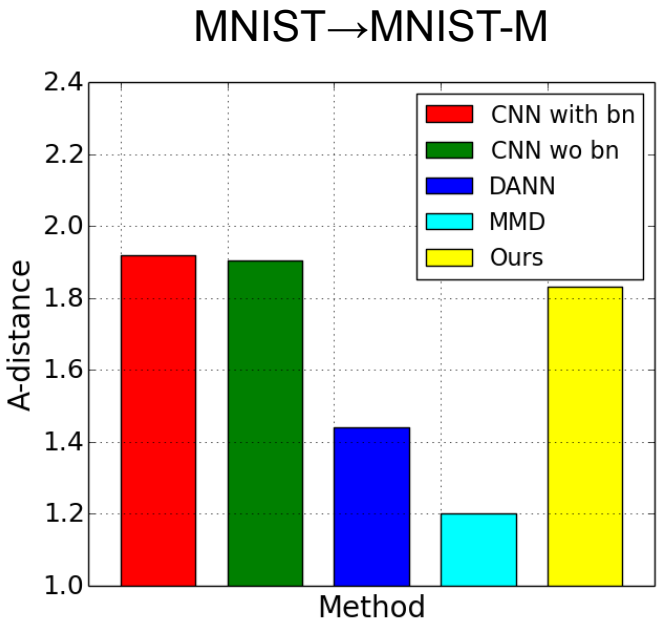
Accuracy on Target Domain

Method \ Source / Target	MNIST / MN-M	MNIST / SVHN	SVHN / MNIST	SYNDIG / SVHN	SYN NUM / GTSRB
Source Only (w/o BN)	59.1	37.2	68.1	84.1	79.2
Source Only (with BN)	57.1	34.9	70.1	85.5	75.7
DANN [Ganin et al., 2014]	81.5	35.7	71.1	90.3	88.7
MMD [Long et al., 2015 ICML]	76.9	-	71.1	88.0	91.1
DSN [Bousmalis et al, 2016 NIPS]	83.2	-	82.7	91.2	93.1
K-NN Labeling [Sener et al., 2016 NIPS]	86.7	40.3	78.8	-	-
Ours (w/o BN)	85.3	39.8	79.8	93.1	96.2
Ours (w/o Weight constraint)	94.2	49.7	86.0	92.4	94.0
Ours	94.0	52.8	86.8	92.9	96.2

- **Our method outperformed other methods.**
- **The effect of BN is obvious in some settings.**
- **The effect of weight constraint is not obvious.**

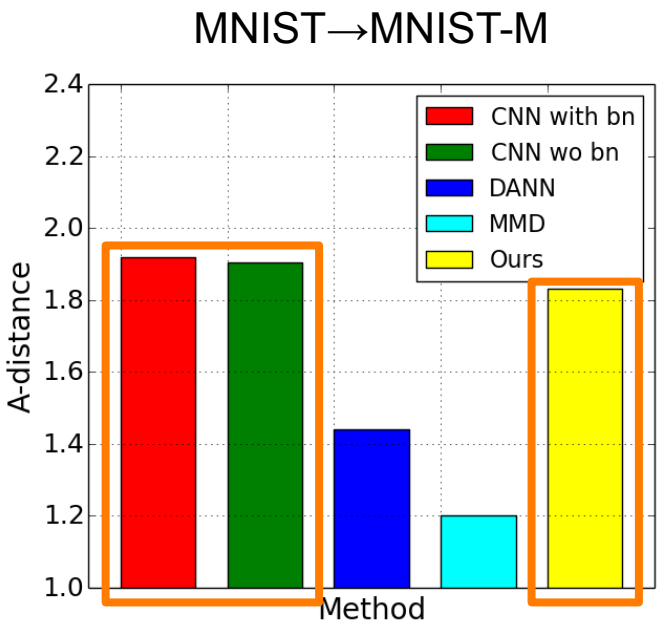
A-distance between Domains

- A-distance
  - Calculated by domain classifier's error
  - As the distance is smaller, features are similar



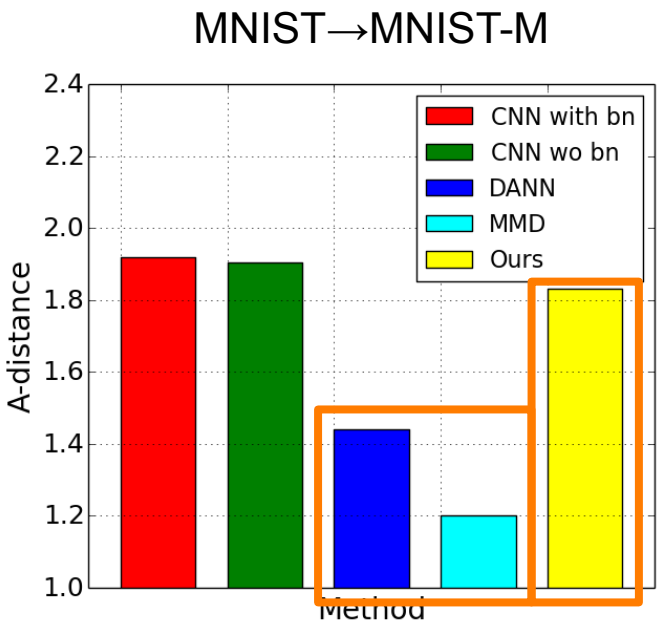
A-distance between Domains

- A-distance
  - Calculated by domain classifier's error
  - As the distance is smaller, features are similar



A-distance between Domains

- A-distance
  - Calculated by domain classifier's error
  - As the distance is smaller, features are similar



- Proposed method does not make the divergence small.

Minimizing the divergence is not a only way to achieve a good adaptation

Summary

- Method for target discriminative domain adaptation
- Proposal of “Asymmetric tri-training”
- Effectiveness is shown in experiments