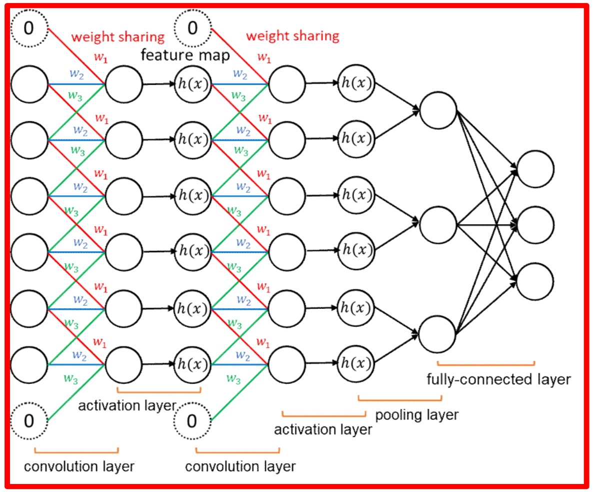
GTC March 18, 2019

Learning from Limited Data

The University of Tokyo / RIKEN AIP
Tatsuya Harada

Deep Neural Networks for Visual Recognition

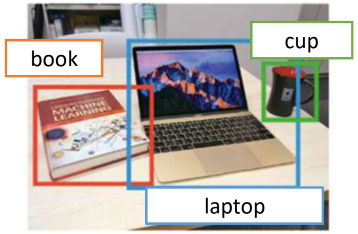
Deep Neural Networks



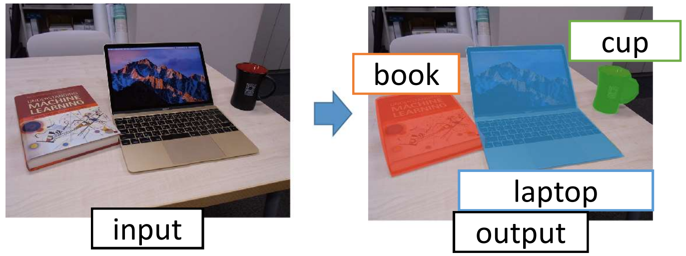
- Tasks in the visual recognition field
  - Object class recognition
  - Object detection
  - Image caption generation
  - Semantic and instance segmentation
  - Image generation
  - Style transfer
- DNNs becomes an indispensable module.
- A large amount of labeled data is needed to train DNNs.
- Reducing annotation cost is highly required.

Applications

cellphone



A yellow train on the tracks near a train station.



Can we learn Deep Neural Networks from limited Supervised Information?

Topics

- Recent progresses in our team (MIL, the University of Tokyo) for learning from limited data
- Between-class learning (BC learning)
- Unsupervised domain adaptation
    - Close domain adaptation
    - Open set domain adaptation
    - Adaptive Object Detection

Learning from Limited Data

Between-class Learning

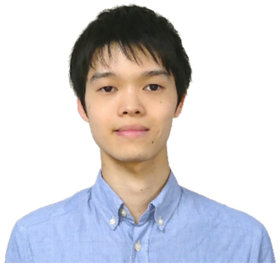
Yuji Tokozume, Yoshitaka Ushiku, Tatsuya Harada

Learning from Between-class Examples for Deep Sound Recognition

ICLR 2018

Between-class Learning for Image Classification

CVPR 2018



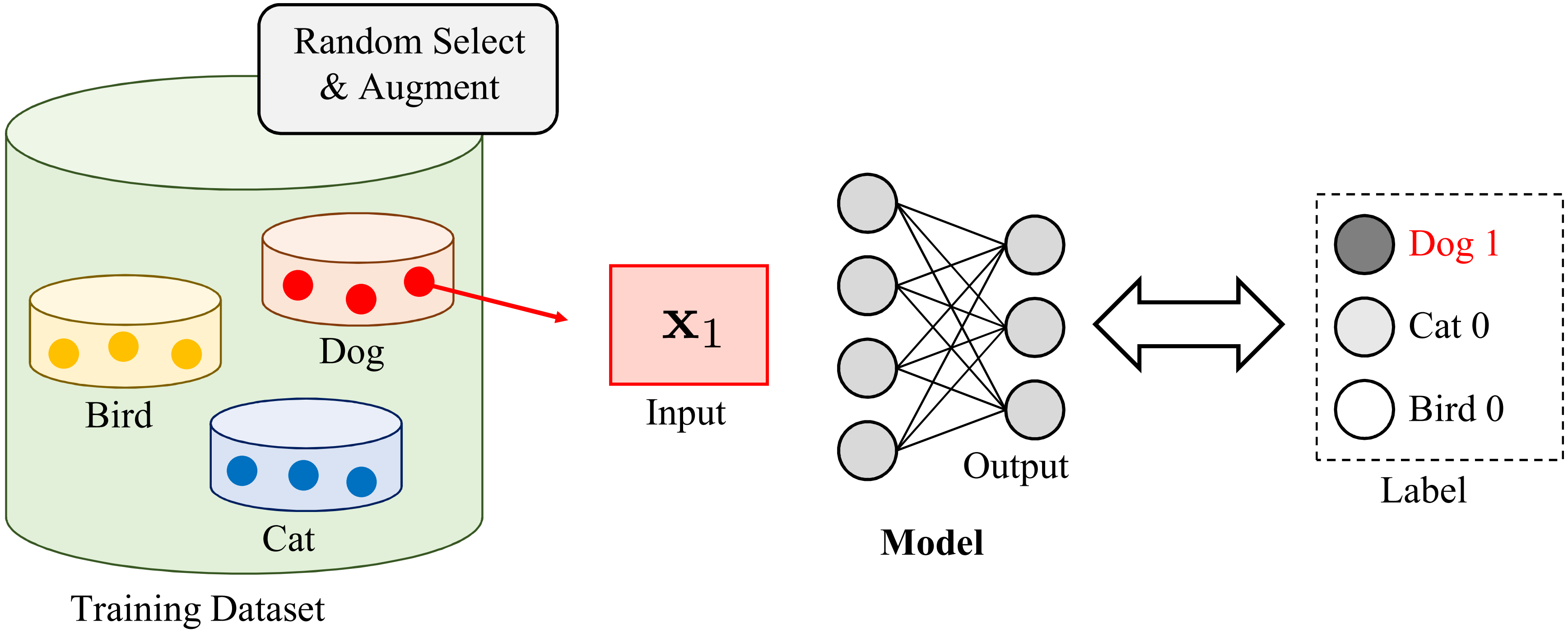
Y. Tokozume

Y. Ushiku

T. Harada

Standard Supervised Learning

1. Select one example from training dataset
2. Train the model to output 1 for the corresponding class and 0 for the other classes

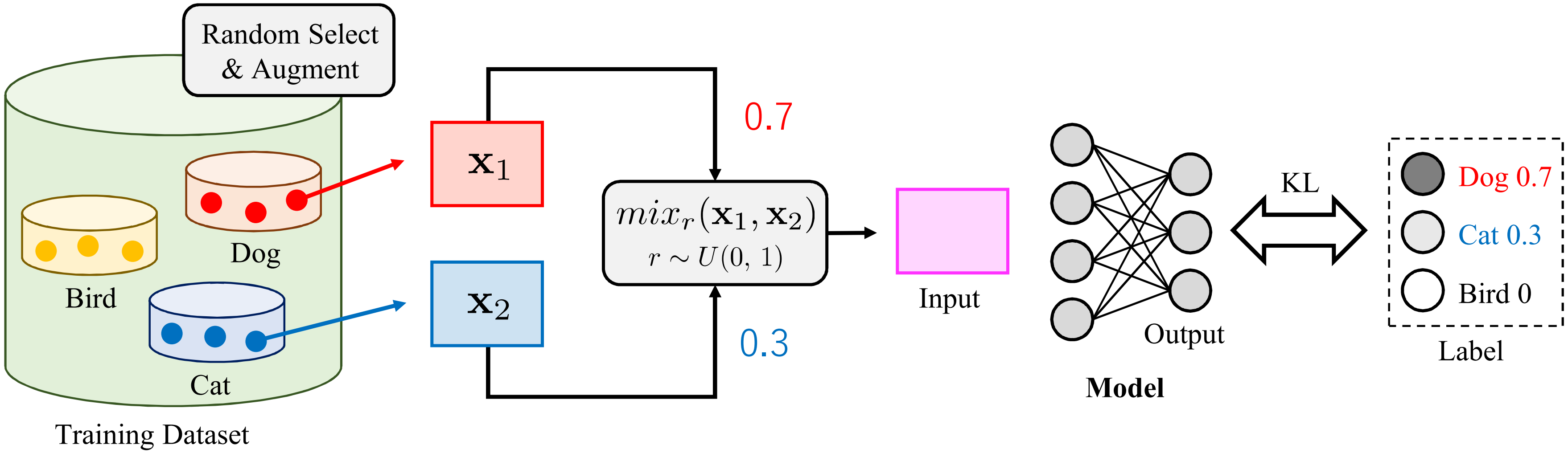


Between-class (BC) Learning

Proposed method

1. Select two training examples from different classes
2. Mix those examples with a random ratio
3. Train the model to output the mixing ratio and mixing classes

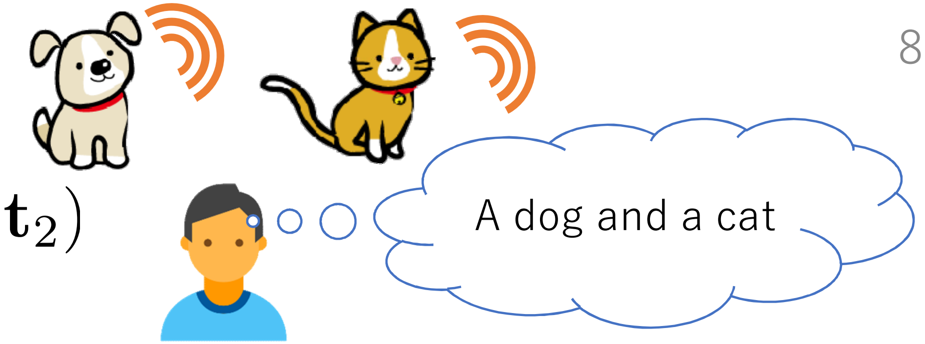
On test phase, we input a single example into the network.



Merits

- Generate infinite training data from limited data
- Learn more discriminative feature space than standard learning

BC learning for sounds

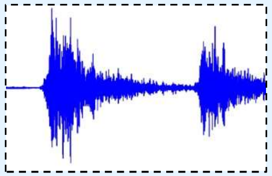


- Two training examples $(\mathbf{x}_1,\, \mathbf{t}_1),\, (\mathbf{x}_2,\, \mathbf{t}_2)$
- Random ratio $r \sim U(0,\, 1)$

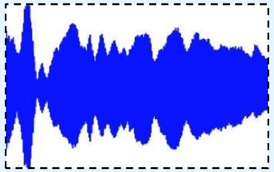
labels

$\mathbf{t}_1$	$\mathbf{t}_2$	$r\,\mathbf{t}_1 + (1-r)\,\mathbf{t}_2$
Dog: 1	Dog: 0	Dog: r
Cat: 0	Cat: 1	Cat: $1-r$
Bird: 0	Bird: 0	Bird: 0

sounds



$\mathbf{x}_1$



$\mathbf{x}_2$

$$\Rightarrow \quad mix_r(\mathbf{x}_1,\, \mathbf{x}_2) = \frac{p\,\mathbf{x}_1 + (1-p)\,\mathbf{x}_2}{\sqrt{p^2 + (1-p)^2}} \quad \text{where} \quad p = \frac{1}{1 + 10^{\frac{G_1 - G_2}{20}} \cdot \frac{1-r}{r}}$$

G_1, G_2: sound pressure level of $\boldsymbol{x}_1, \boldsymbol{x}_2$[dB]

Results of Sound Recognition

② Various datasets

① Various models

③ Compatible with strong data augmentation

④ Surpass the human level

Model	Learning	Error rate (%) on ESC-50	ESC-10	UrbanSound8K
EnvNet (Tokozume & Harada, 2017)	Standard	29.2 ± 0.1	12.8 ± 0.4	33.7
	BC (ours)	24.1 ± 0.2	11.3 ± 0.6	28.9
SoundNet5 (Aytar et al., 2016)	Standard	33.8 ± 0.2	16.4 ± 0.8	33.3
	BC (ours)	27.4 ± 0.3	13.9 ± 0.4	30.2
M18 (Dai et al., 2017)	Standard	31.5 ± 0.5	18.2 ± 0.5	28.8
	BC (ours)	26.7 ± 0.1	14.2 ± 0.9	26.5
Logmel-CNN (Piczak, 2015a) + BN	Standard	27.6 ± 0.2	13.2 ± 0.4	25.3
	BC (ours)	23.1 ± 0.3	$\mathbf{9.4 \pm 0.4}$	23.5
EnvNet-v2 (ours)	Standard	25.6 ± 0.3	14.2 ± 0.8	30.9
	BC (ours)	$\mathbf{18.2 \pm 0.2}$	10.6 ± 0.6	**23.4**
EnvNet-v2 (ours) + strong augment	Standard	21.2 ± 0.3	10.9 ± 0.6	24.9
	BC (ours)	$\mathbf{15.1 \pm 0.2}$	$\mathbf{8.6 \pm 0.1}$	**21.7**
SoundNet8 + Linear SVM (Aytar et al., 2016)		25.8	7.8	-
Human (Piczak, 2015b)		18.7	4.3	-

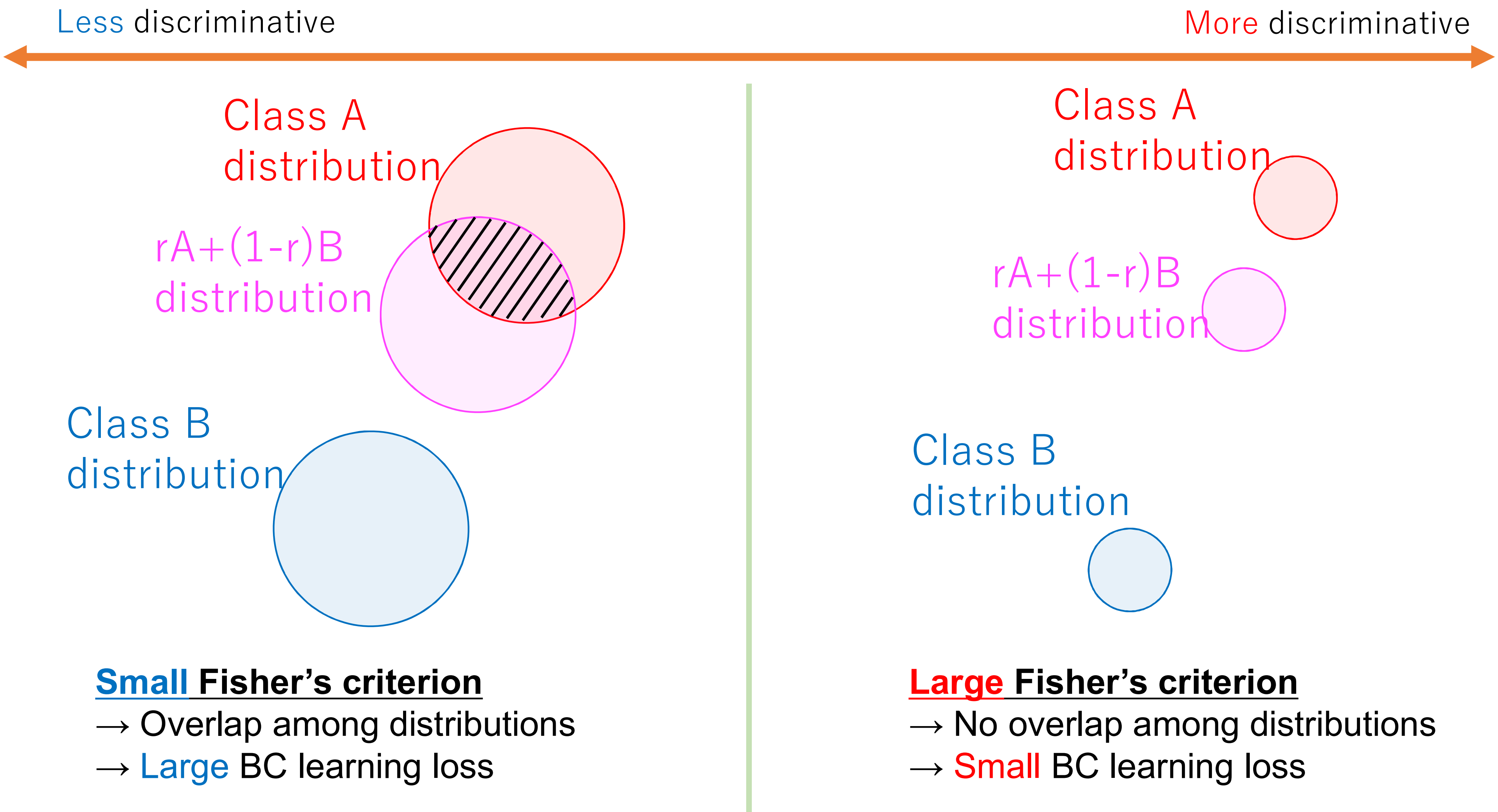
We can improve recognition performance for any sound networks,
if we apply the BC learning.

Results on CIFAR

Our preliminary results were presented in ILSVRC2017 on July 26, 2017.

Model	Learning	Error rate (%) on CIFAR-10	Error rate (%) on CIFAR-100
11-layer CNN	Standard	6.07 ± 0.04	26.68 ± 0.09
	BC (ours)	5.40 ± 0.07	24.28 ± 0.11
	BC+ (ours)	$\mathbf{5.22 \pm 0.04}$	$\mathbf{23.68 \pm 0.10}$
ResNet-29† [28]	Standard	4.24 ± 0.06 / 4.39 [28]	20.18 ± 0.07
	BC (ours)	3.75 ± 0.04	19.56 ± 0.10
	BC+ (ours)	$\mathbf{3.55 \pm 0.03}$	$\mathbf{19.41 \pm 0.07}$
ResNeXt-29 (16×64d)† [28]	Standard	3.54 ± 0.04 / 3.58 [28]	$\mathbf{16.99 \pm 0.06}$ / 17.31 [28]
	BC (ours)	$\mathbf{2.79 \pm 0.06}$	18.21 ± 0.12
	BC+ (ours)	2.81 ± 0.06	17.93 ± 0.09
DenseNet-BC ($k = 40$)† [13]	Standard	3.61 ± 0.10 / 3.46 [13]	17.28 ± 0.12 / 17.18 [13]
	BC (ours)	2.68 ± 0.03	16.36 ± 0.10
	BC+ (ours)	$\mathbf{2.57 \pm 0.06}$	$\mathbf{16.23 \pm 0.07}$
Shake-Shake Regularization [9]	Standard	2.86 [9]	**15.85** [9]
	BC (ours)	2.38 ± 0.04	15.90 ± 0.06
	BC+ (ours)	$\mathbf{2.26 \pm 0.01}$	16.00 ± 0.10

How BC Learning Works



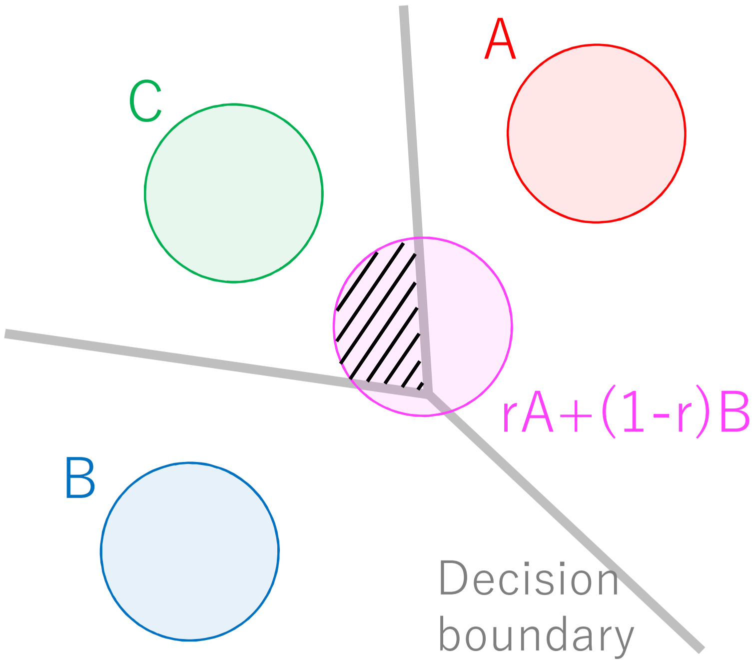
Small Fisher's criterion
→ Overlap among distributions
→ Large BC learning loss

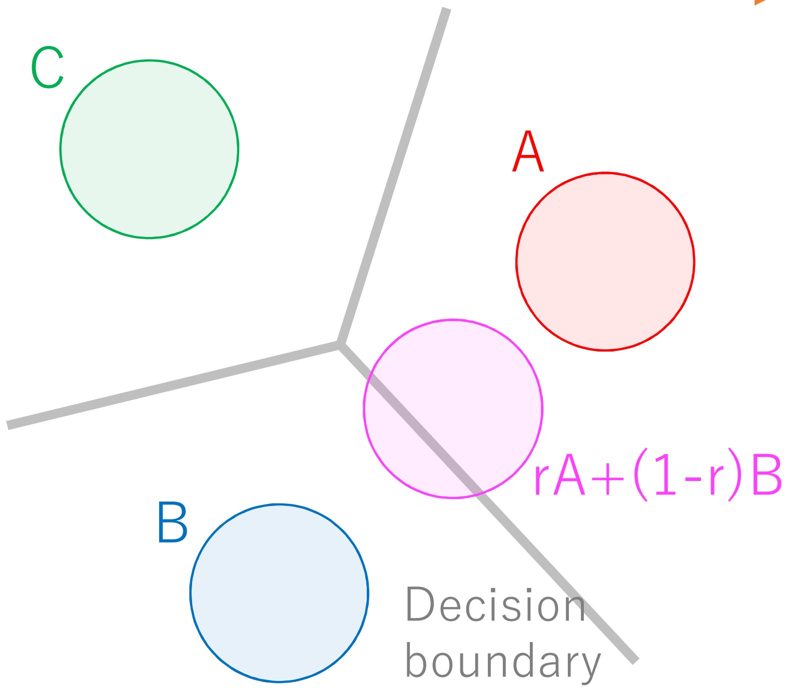
Large Fisher's criterion
→ No overlap among distributions
→ Small BC learning loss

How BC Learning Works

In the classification, the distributions must be uncorrelated because the teaching signal is discrete.

Large correlation ⟷ Small correlation





Large correlation among classes

→ Mixing class of A and B may be classified into class C.

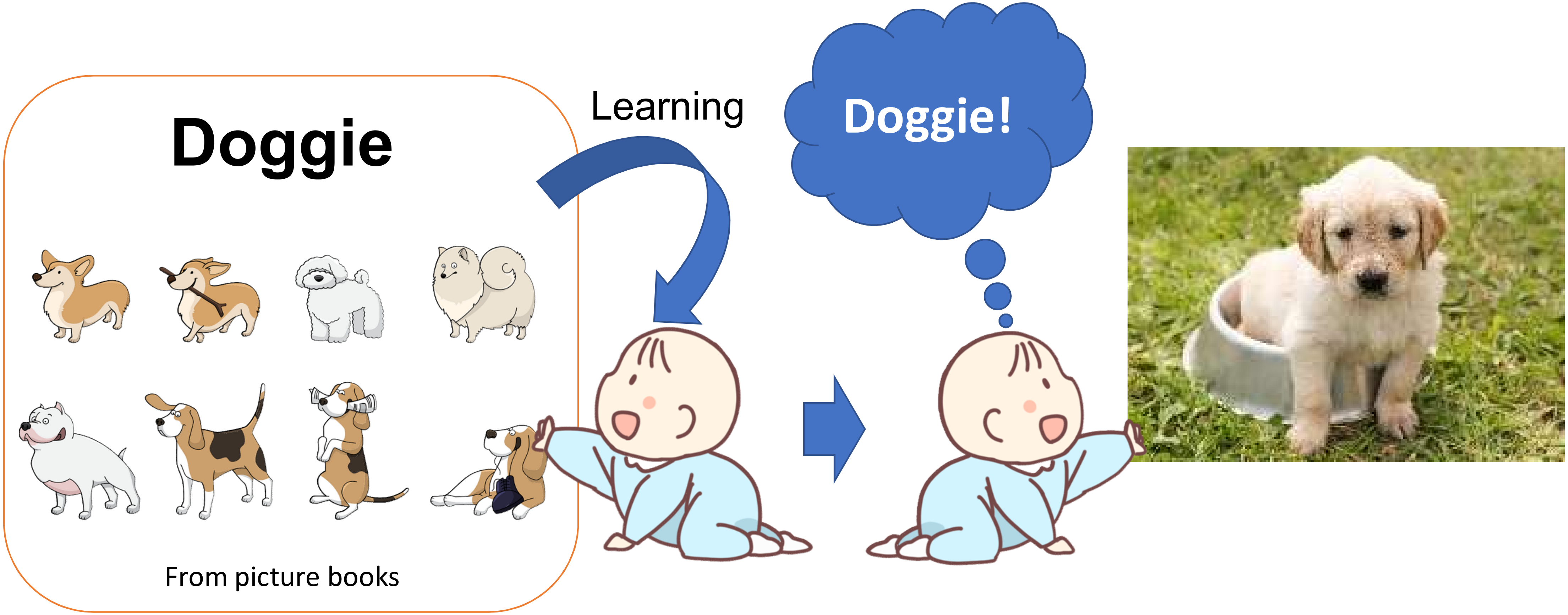
→ Large BC learning loss

Small correlation among classes

→ Mixing class of A and B is not classified into class C.

→ Small BC learning loss

Knowledge Transfer



Domain Adaptation

Image by Chiemsee2016 on Pixabay

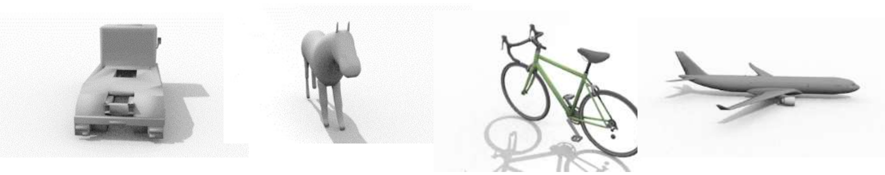
Image by GraphicMama-team on Pixabay

Domain Adaptation (DA)

- Problems
  - Supervised learning model needs many labeled examples
  - Cost to collect them in various domains
- Goal
  - Transfer knowledge from source (rich supervised data) to target (small supervised data) domain
  - Classifier that works well on target domain.
- Unsupervised Domain Adaptation (UDA)
  - Labeled examples are given only in the source domain.
  - There are no labeled examples in the target domain.

Source domain

Synthetic images, labeled

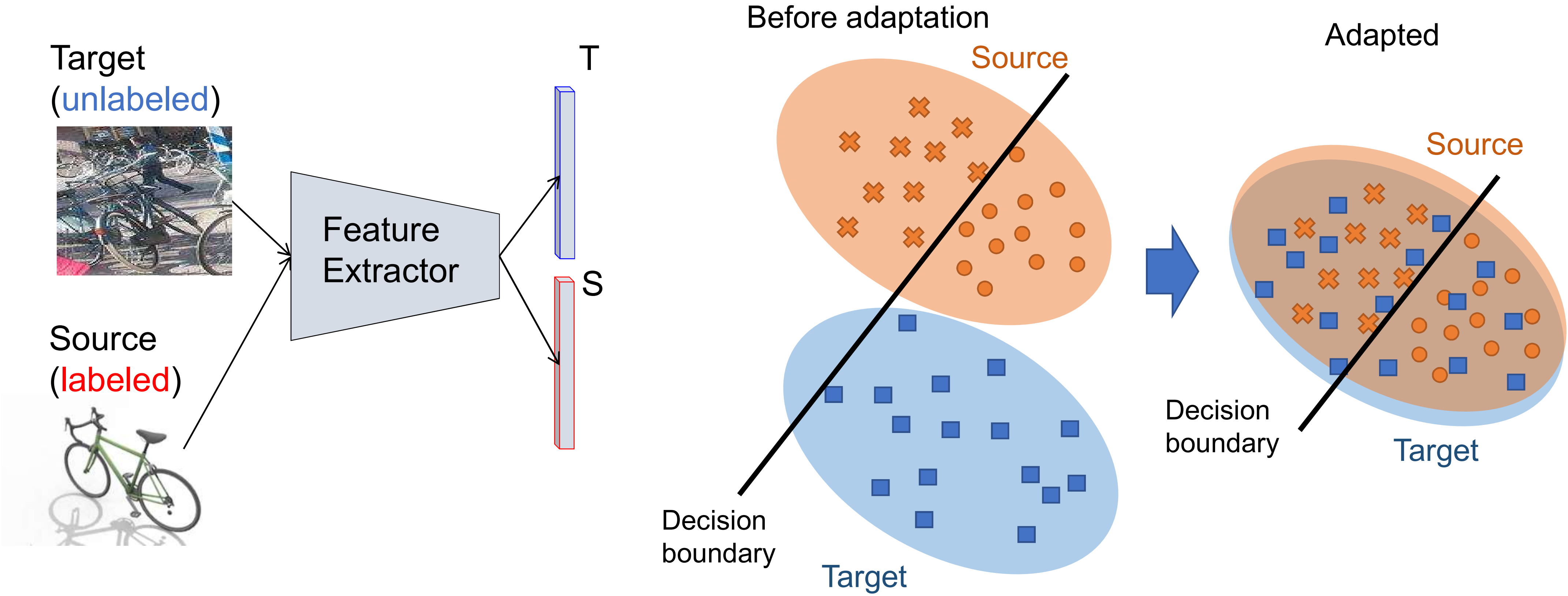


Target domain

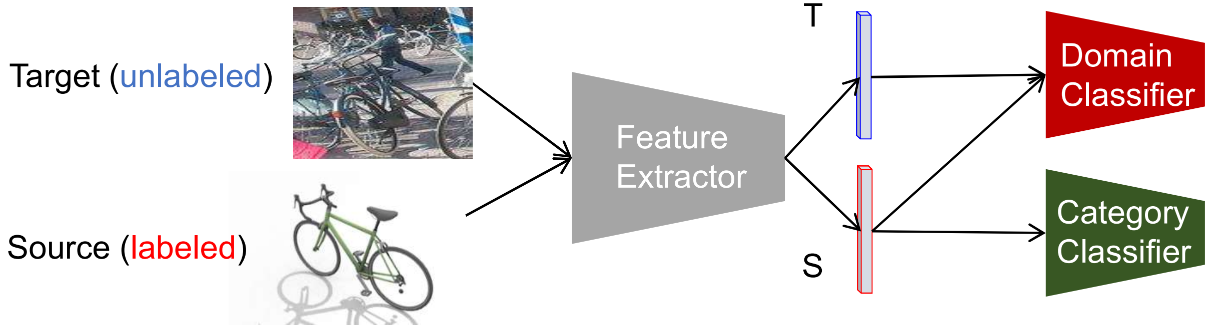
Real images, unlabeled

Distribution Matching for Unsupervised Domain Adaptation

- **Distribution matching based method**
  - Match distributions of source and target features
    - Domain Classifier (GAN) [Ganin et al., 2015]
    - Maximum Mean Discrepancy [Long et al., 2015]

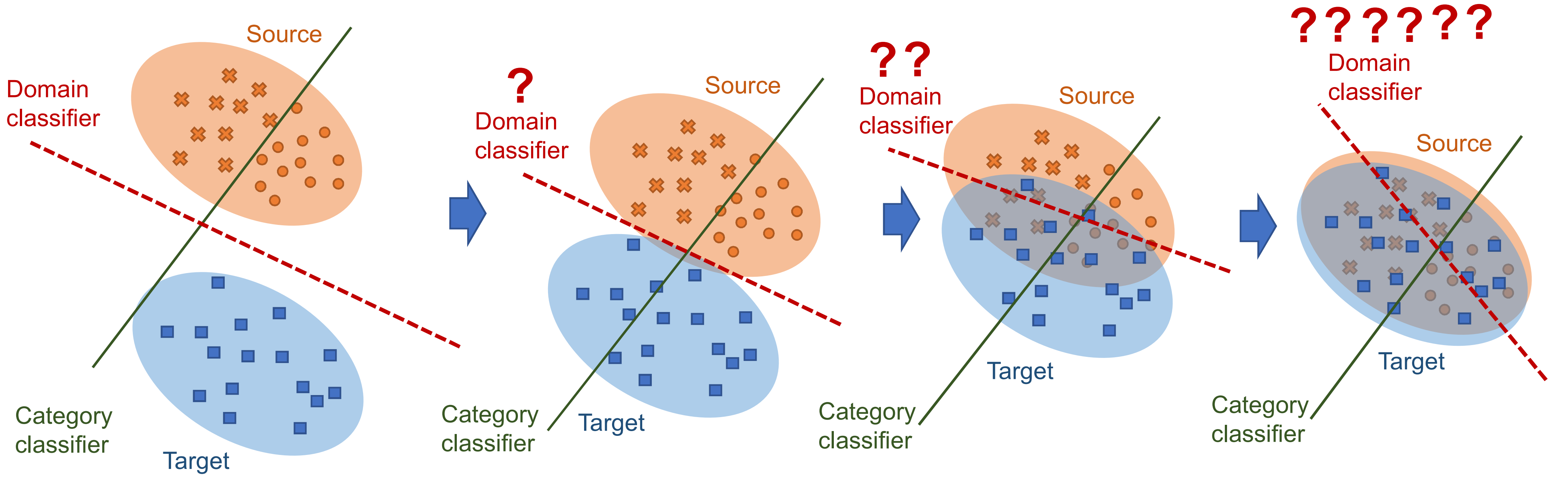


Adversarial Domain Adaptation



- Training the feature generator in a adversarial way works well!
- Category classifier, domain classifier, feature extractor
- Problems
  - Whole distribution matching
  - Ignorance of category information in source domain

Tzeng, Eric, et al. Adversarial discriminative domain adaptation. CVPR, 2017.



Unsupervised Domain Adaptation using Classifier Discrepancy

Kuniaki Saito[1], Kohei Watanabe[1], Yoshitaka Ushiku[1], Tatsuya Harada[1, 2]
1: The University of Tokyo, 2: RIKEN

CVPR 2018, oral presentation


K. Saito

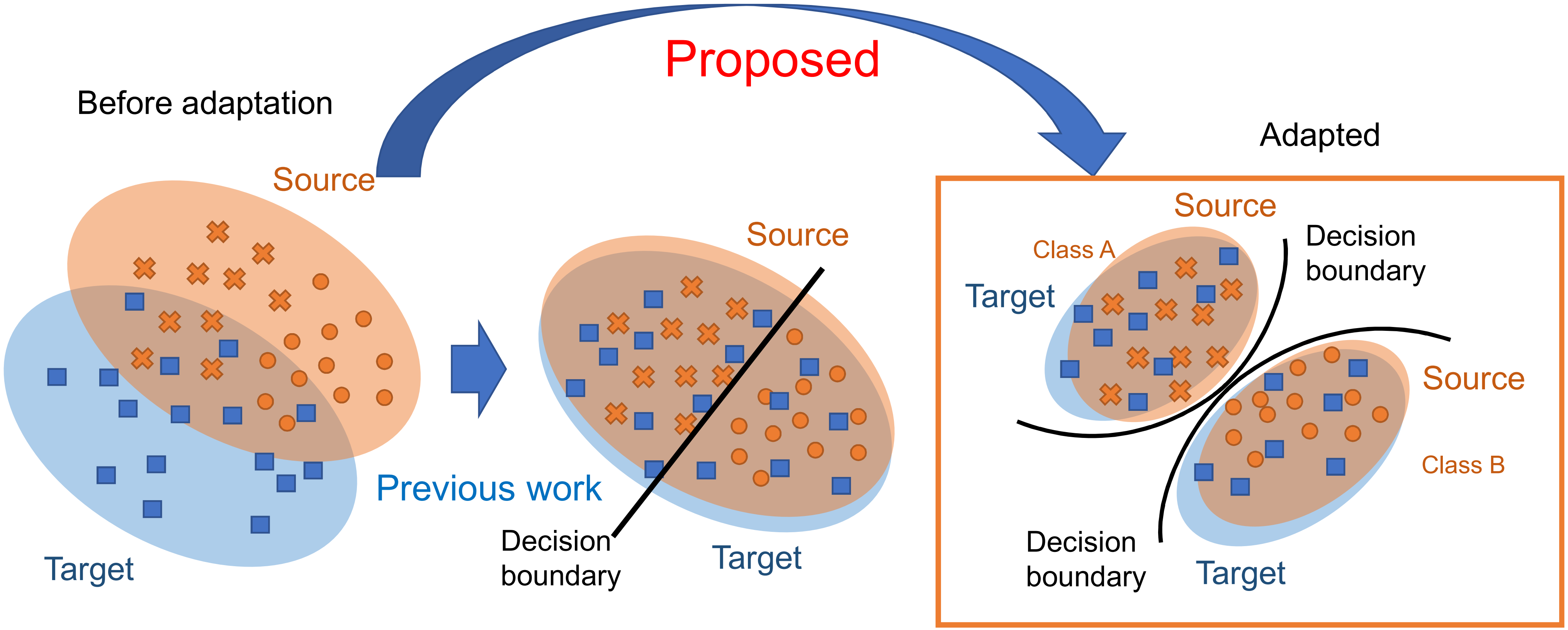

K. Watanabe


Y. Ushiku


T. Harada

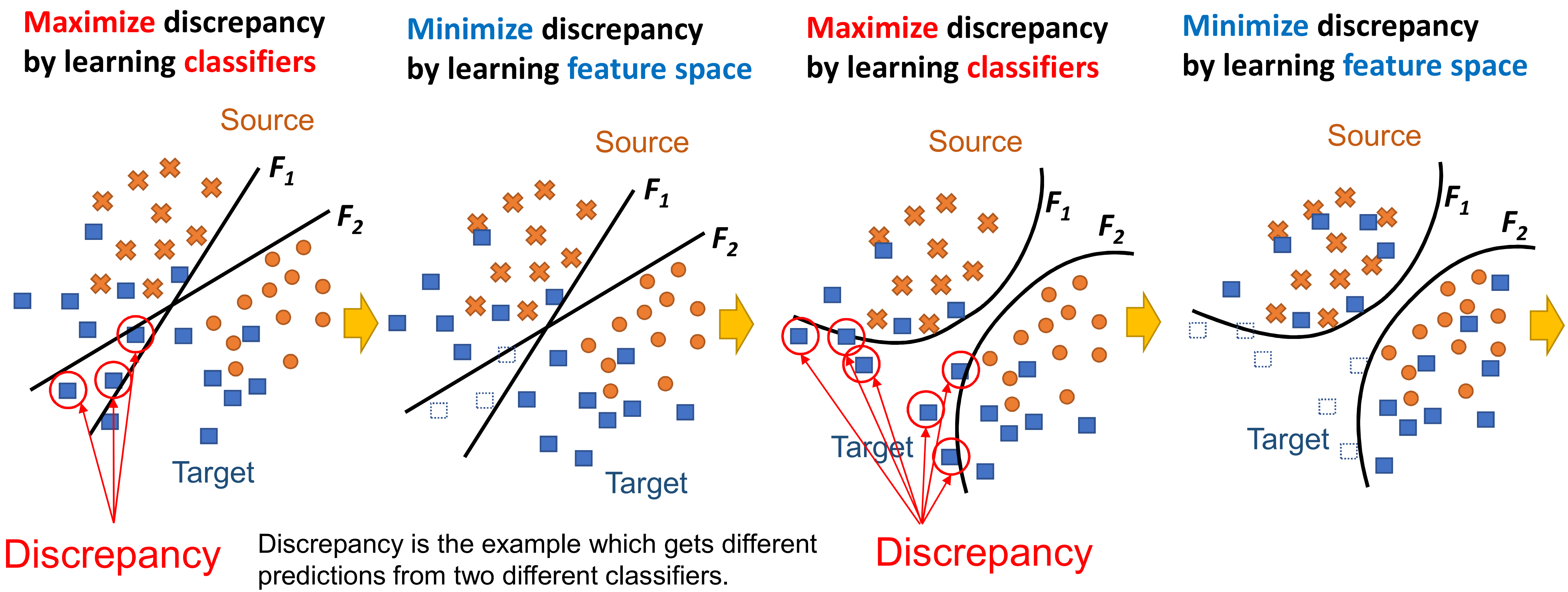
Proposed Approach

- Considering class specific distributions
- Using decision boundary to align distributions

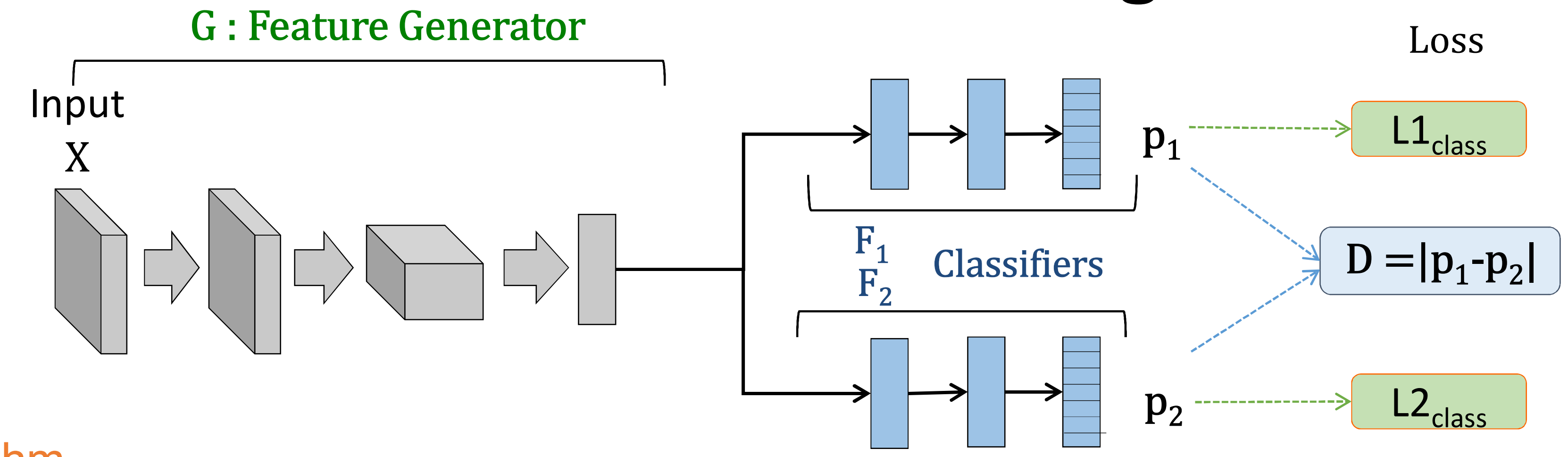


Key Idea

- Maximizing discrepancy by learning two classifiers
- Minimizing discrepancy by learning feature space



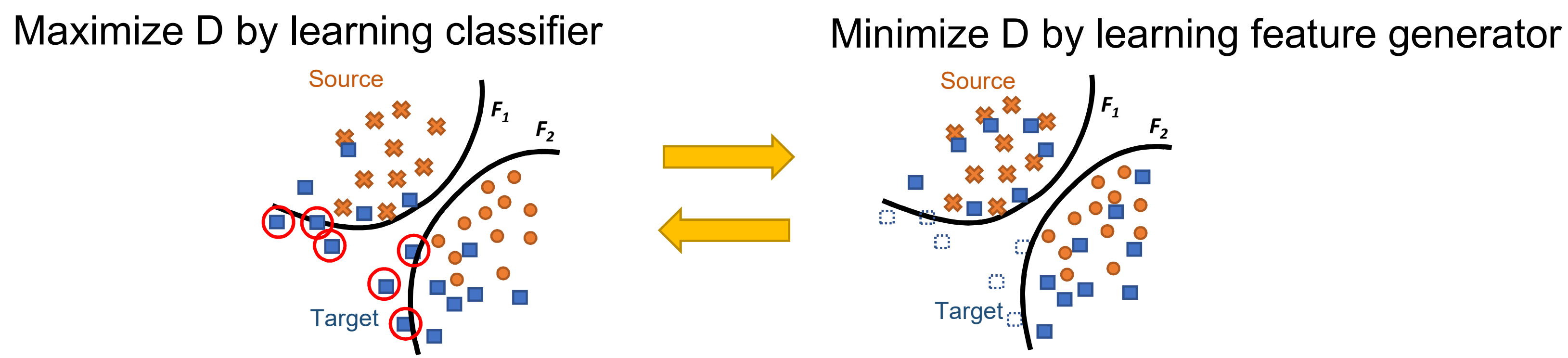
Network Architecture and Training



Algorithm

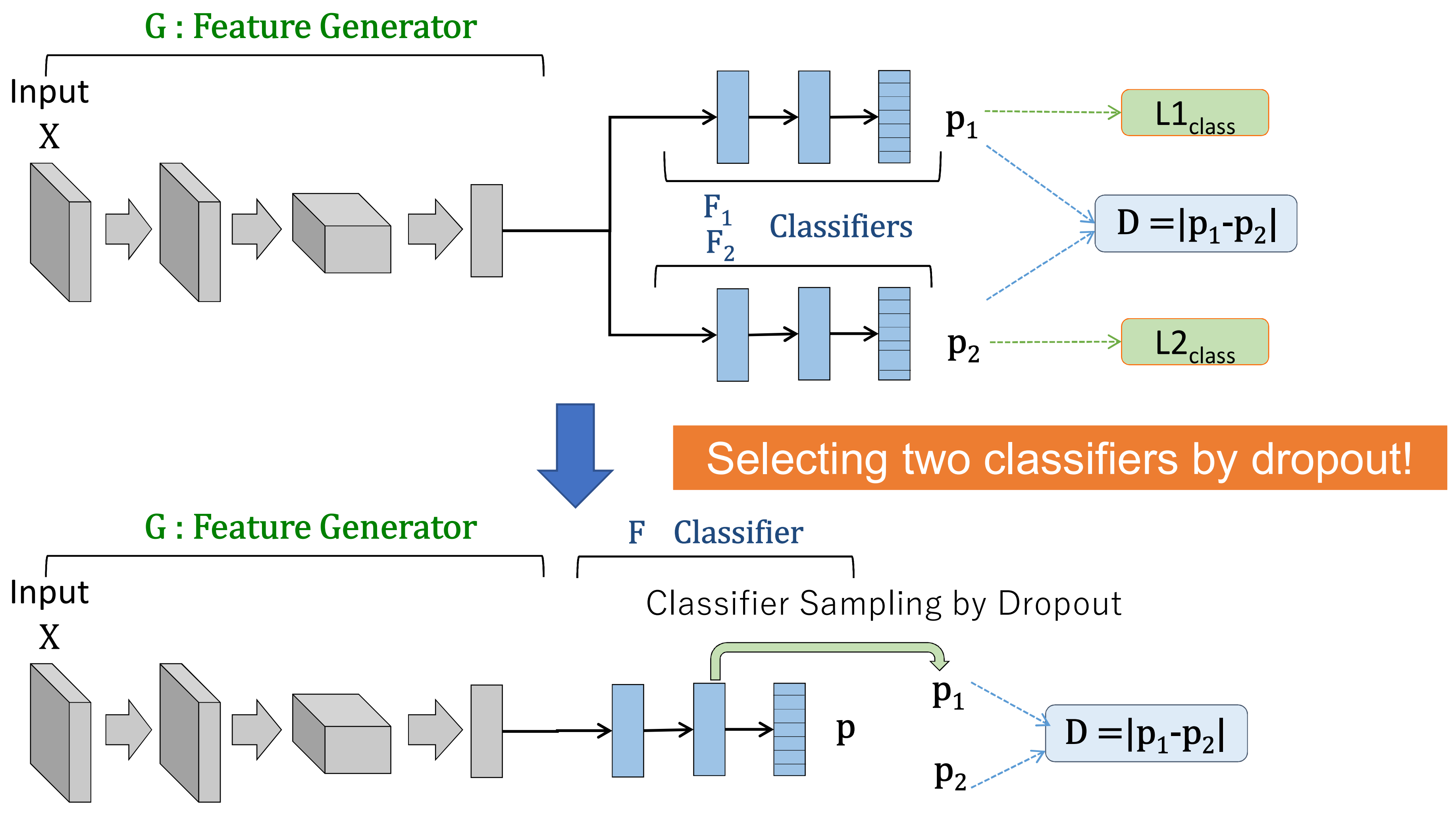
1. Fix generator G, and find classifiers F_1, F_2 that **maximize $D - (L_1 + L_2)$**
2. for $k = 1{:}n$

   Fix classifiers F_1, F_2, and find feature generator G that **minimizes D**



Improving by Dropout

Adversarial Dropout Regularization
Kuniaki Saito, Yoshitaka Ushiku, Tatsuya Harada, Kate Saenko
ICLR 2018



Why Discrepancy Method Works Well?

Theorem [Ben et al., 2010]
Let H be the hypothesis class. Given two domains $\mathcal{S}$ and $\mathcal{T}$, we have

$$\forall h \in H, R_{\mathcal{T}}(h) \leq R_{\mathcal{S}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{S}, \mathcal{T}) + \lambda$$

Hypothesis

Expected error in target domain

Expected error in source domain

Shared error of the ideal hypothesis
$\lambda = \min[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h)]$

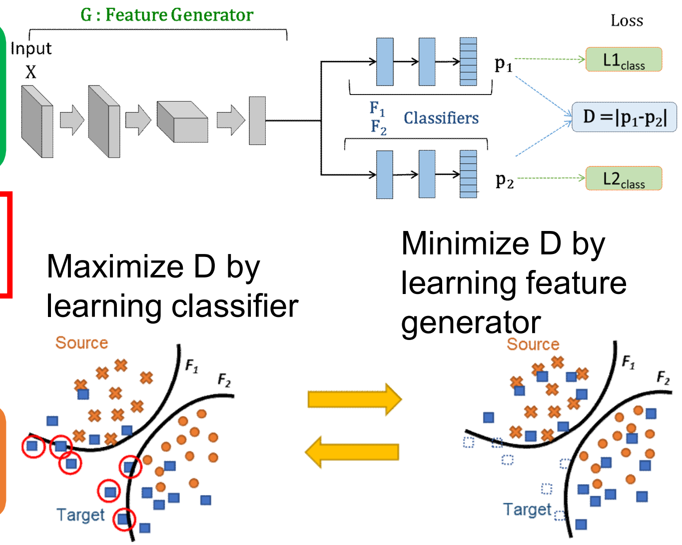
$$\sup_{(h,h') \in \mathcal{H}^2} \left| \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{S}} I[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})] - \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{T}} I[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})] \right|$$

Minimal upper-bound

This term is assumed to be low, if h and h' can classify source samples correctly.

$h(x) = F_1 \circ G(x)$,
$h'(x) = F_2 \circ G(x)$

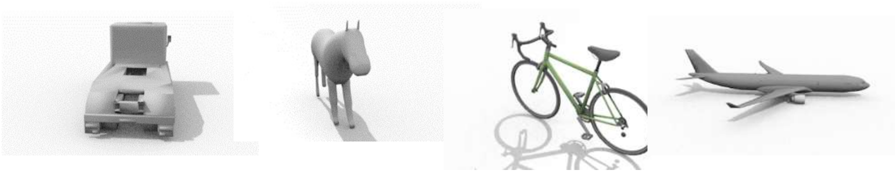
$$\min_{G} \max_{F_1, F_2} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathcal{T}} I[F_1 \circ G(\boldsymbol{x}) \neq F_2 \circ G(\boldsymbol{x})]$$



Object Classification

- Synthetic images to Real images (12 Classes)
- Finetune pre-trained ResNet101 [He et al., CVPR 2016] (ImageNet)
- Source:images, Target:images

Source (Synthetic images)



Target (Real images)

Method	plane	bcycl	bus	car	hrs	knf	mcycl	prsn	plnt	sktbrd	trn	trck	mean
Source Only	55.1	53.3	61.9	59.1	80.6	17.9	79.7	31.2	81.0	26.5	73.5	8.5	52.4
MMD [Long et al., ICML 2015]	87.1	63.0	76.5	42.0	**90.3**	42.9	**85.9**	53.1	49.7	36.3	**85.8**	20.7	61.1
DANN [Ganin et al., ICML 2015]	81.9	**77.7**	82.8	44.3	81.2	29.5	65.1	28.6	51.9	**54.6**	82.8	7.8	57.4
Ours ($n = 4$)	**87.0**	60.9	**83.7**	**64.0**	88.9	**79.6**	84.7	**76.9**	**88.6**	40.3	83.0	**25.8**	**71.9**

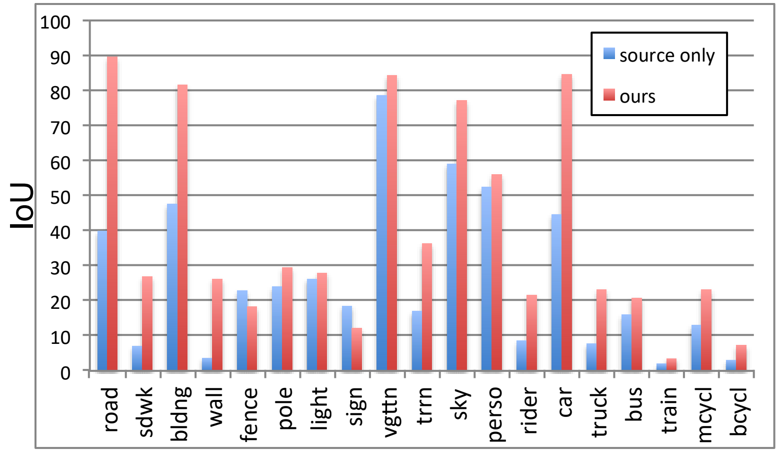
Semantic Segmentation

- Simulated Image (GTA5) to Real Image (CityScape)
- Finetuning of pre-trained VGG, Dilated Residual Network [Yu et al., 2017] (ImageNet)
  - Calculate discrepancy pixel-wise
- Evaluation by mean IoU (TP/(TP+FP+FN))

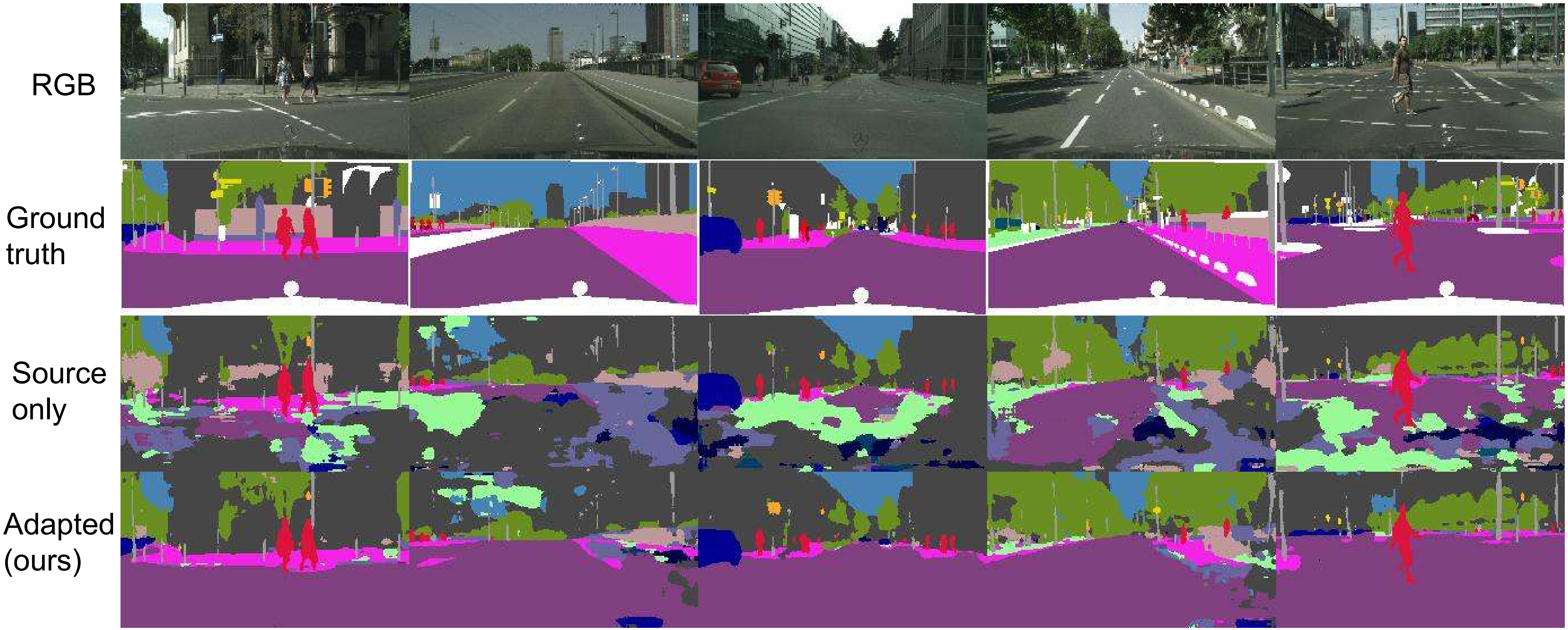
GTA 5 (Source)

CityScape(Target)

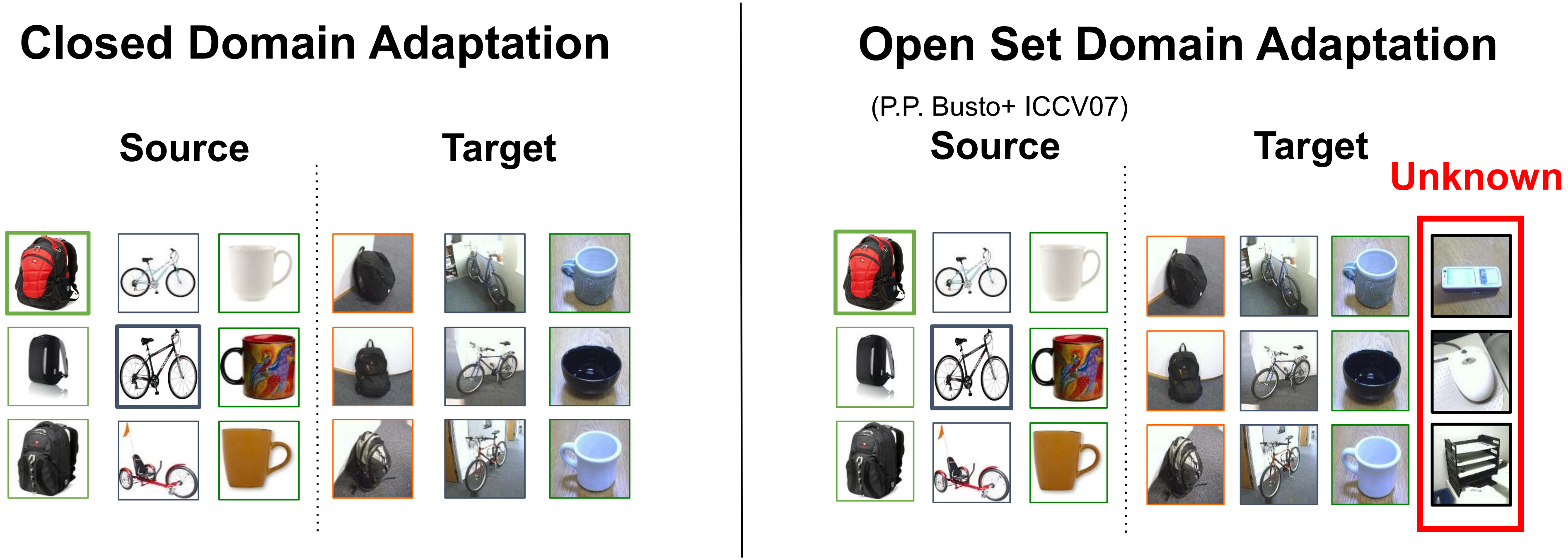
Network	Method	mIoU
VGG-16	Source Only	21.2
	FCN Wld [Hoffman et al., Arxiv 2017]	27.1
VGG-16	Source Only	22.3
	CrrclmDA (I) [Zhang el al., ICCV 2017]	23.1
VGG-16	Source Only	24.9
	Ours	28.8
DRN-105	Source Only	22.2
	Ours	**39.7**



Qualitative Results



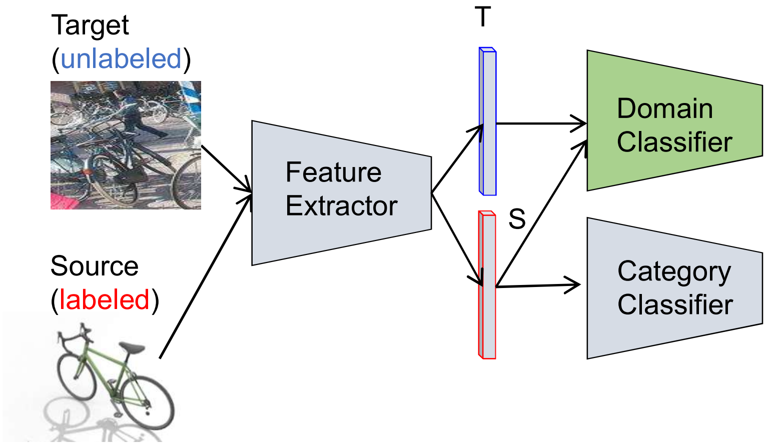
Open Set Domain Adaptation (OSDA)

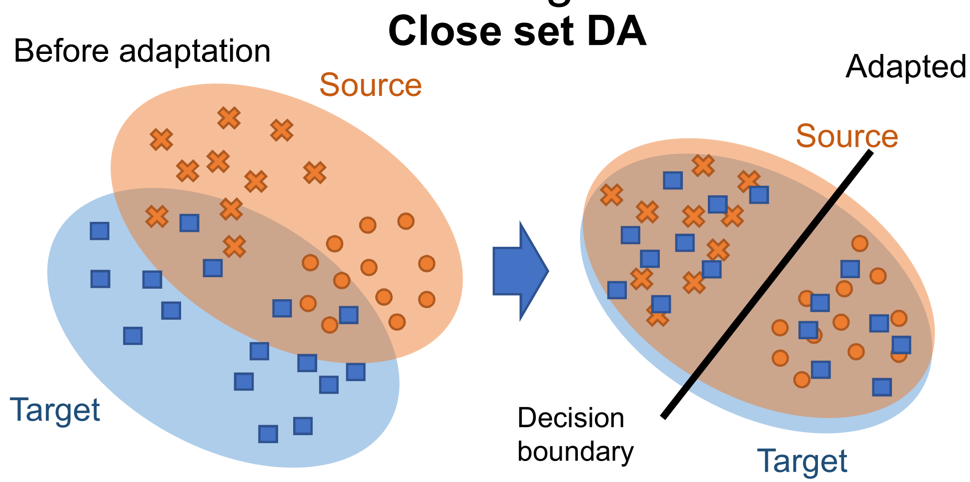


- Source and target completely share classes in domain adaptation.
- Target examples are unlabeled. Open set situation is more realistic.
- Open set ･･･Target contains unknown category.

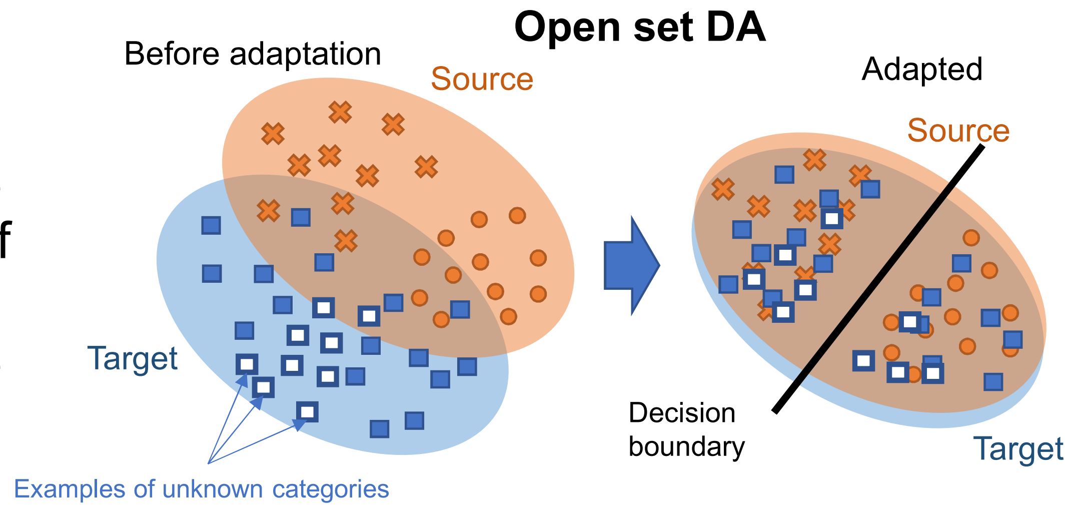
Distribution Matching for Open Set DA

- Close set domain adaptation: match distributions of source and target features





- Problem in open set
  - Examples of unknown category are also aligned with the distributions of known categories.
  - Examples of unknown category are classified into known categories.



Open Set Domain Adaptation by Backpropagation

Kuniaki Saito[1], Shohei Yamamoto[1], Yoshitaka Ushiku[1], Tatsuya Harada[1, 2]
1: The University of Tokyo, 2: RIKEN

ECCV 2018


K. Saito

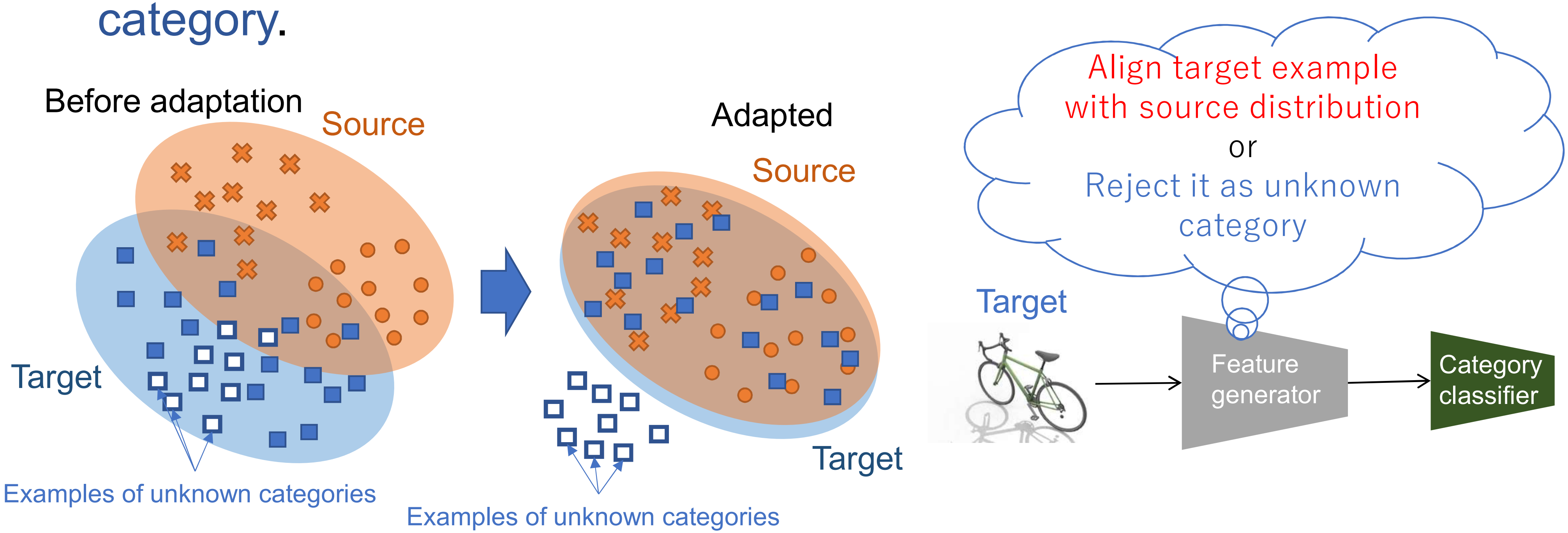

S. Yamamoto


Y. Ushiku


T. Harada

Idea

- Separation of examples of unknown category from these of known categories in target domain
- Alignment between the distribution of known category in target domain and the distribution of source domain
- The feature generator should have option to align target examples with source distribution or to reject target examples as the unknown category.



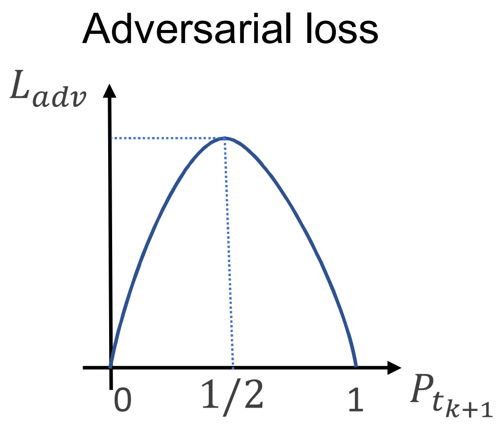
Proposed Method

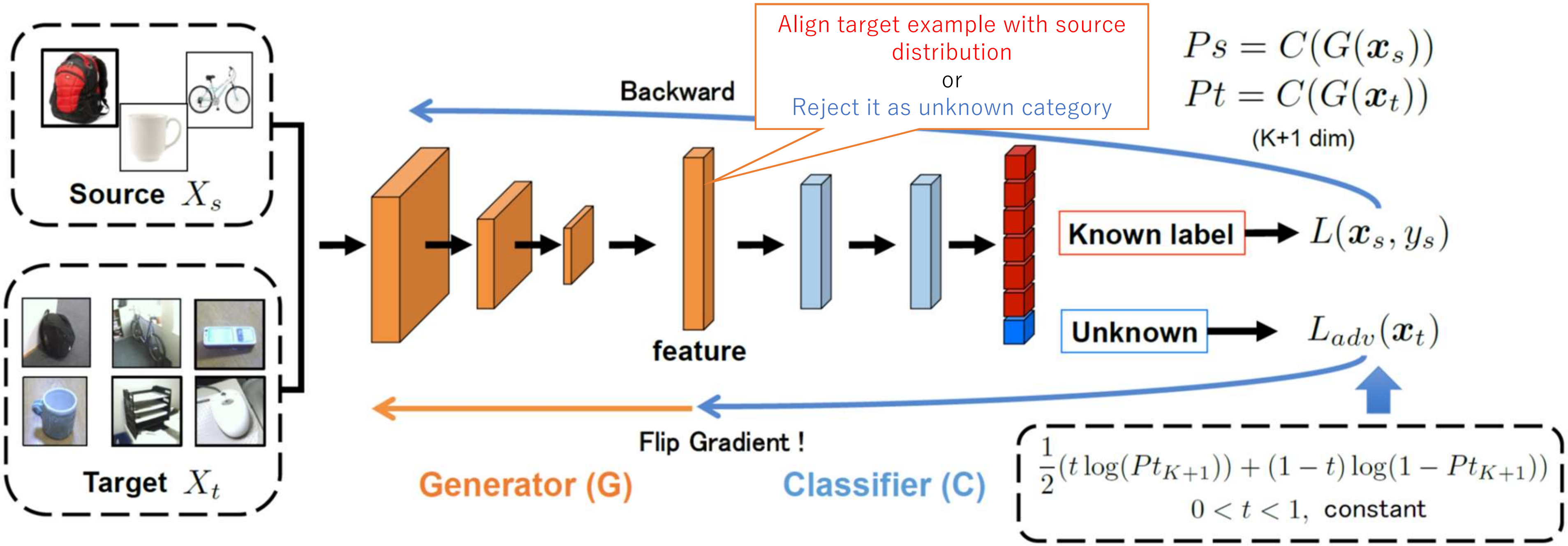
Classifier C

- minimize classification loss to correctly categorize source examples
- maximize adversarial loss ($P(y = K + 1|\boldsymbol{x}_t) = 1/2$) for target examples

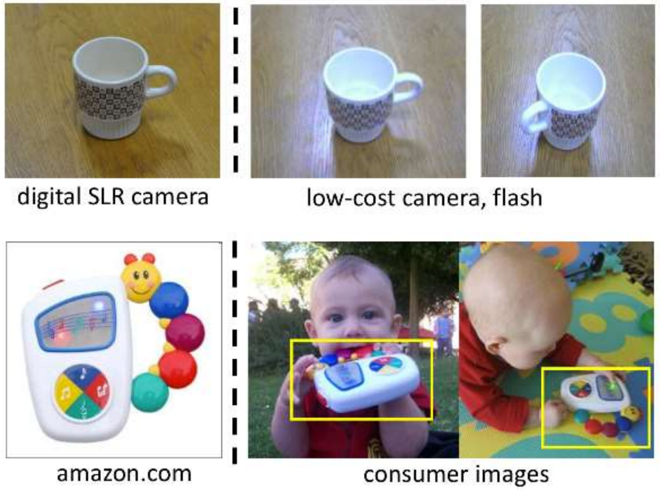
Feature generator G

- minimize adversarial loss to deceive the classier for target examples
- is trained to output $P(y = K + 1|\boldsymbol{x}_t) = 1$ or 0





Experimental Results for Office Dataset



・11 categories classification

・The dataset consists of 31 classes, and 10 classes were selected as shared classes. 21-31 classes are used as unknown samples in the target domain.

・BP, MMD are distribution matching based method.

・OS* is measured only for known class.

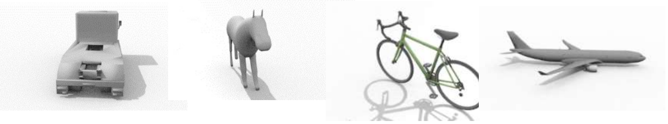
	Adaptation Scenario													
	A-D		A-W		D-A		D-W		W-A		W-D		AVG	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
Method w/ unknown classes in source domain (AlexNet)														
BP [4]	78.3	77.3	75.9	73.8	57.6	54.1	89.8	88.9	64.0	61.8	98.7	98.0	77.4	75.7
ATI-λ [2]	79.8	79.2	77.6	76.5	71.3	70.0	93.5	93.2	76.7	76.5	98.3	99.2	82.9	82.4
Method w/o unknown classes in source domain (AlexNet)														
OSVM	59.6	59.1	57.1	55.0	14.3	5.9	44.1	39.3	13.0	4.5	62.5	59.2	40.6	37.1
MMD + OSVM	47.8	44.3	41.5	36.2	9.9	0.9	34.4	28.4	11.5	2.7	62.0	58.5	34.5	28.5
BP+OSVM	40.8	35.6	31.0	24.3	10.4	1.5	33.6	27.3	11.5	2.7	49.7	44.8	29.5	22.7
ATI-λ[2] + OSVM	72.0	-	65.3	-	**66.4**	-	82.2	-	71.6	-	92.7	-	75.0	-
Ours	**76.6**	**76.4**	**74.9**	**74.3**	62.5	**62.3**	**94.4**	**94.6**	**81.4**	**81.2**	**96.8**	**96.9**	**81.1**	**80.9**
Method w/o unknown classes in source domain (VGGNet)														
OSVM	82.1	83.9	75.9	75.8	38.0	33.1	57.8	54.4	54.5	50.7	83.6	83.3	65.3	63.5
MMD + OSVM	84.4	**85.8**	75.6	75.7	41.3	35.9	61.9	58.7	50.1	45.6	84.3	83.4	66.3	64.2
BP+OSVM	83.1	84.7	76.3	76.1	41.6	36.5	61.1	57.7	53.7	49.9	82.9	82.0	66.4	64.5
Ours	**85.8**	**85.8**	**85.3**	**85.1**	**88.7**	**89.6**	**94.6**	**95.2**	**83.4**	**83.1**	**97.1**	**97.3**	**89.1**	**89.4**

Table 1. Accuracy (%) of each method in 10 shared class situation. A, D and W correspond to Amazon, DSLR and Webcam respectively.

Experimental Results for VisDA Dataset

Source domain

Labeled synthetic images



Target domain

Unlabeled real images

- VisDA dataset consists of 12 categories in total.
- We choose 6 categories from them and set other 6 categories as the unknown class.

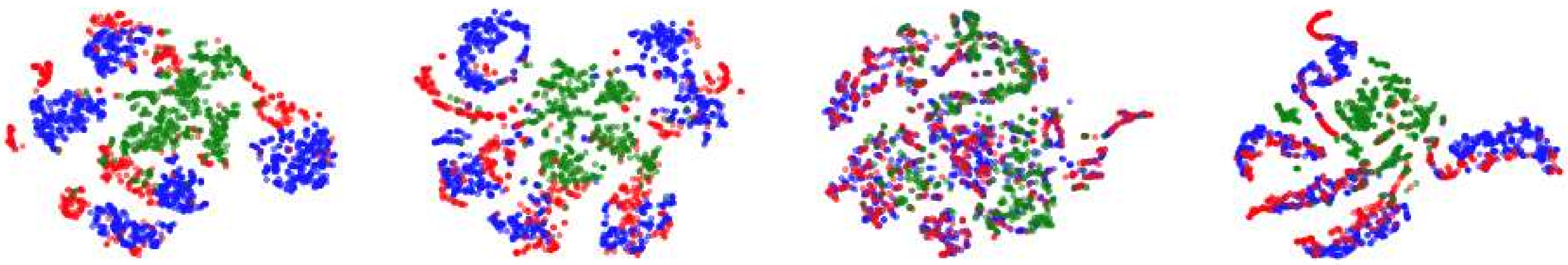
Ground Truth Class → Predicted Class			
Known → Unknown ×	Unknown → Known ×	Known → Known √	Unknown → Unknown √
Train → Unknown 	Unknown → Motorcycle 	Truck → Truck 	Unknown → Unknown 
Motorcycle → Unknown 	Unknown → Motorcycle 	Bicycle → Bicycle 	Unknown → Unknown 
Car → Unknown 	Unknown → Motorcycle 	Motorcycle → Motorcycle 	Unknown → Unknown 

Table 4. Examples of recognition results on VisDA dataset.

Experimental Results on Digits Dataset

Method	SVHN-MNIST				USPS-MNIST				MNIST-USPS				Average			
	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK
OSVM	54.3	63.1	37.4	10.5	43.1	32.3	63.5	97.5	79.8	77.9	84.2	**89.0**	59.1	57.7	61.7	65.7
MMD+OSVM	55.9	64.7	39.1	12.2	62.8	58.9	69.5	82.1	80.0	79.8	81.3	81.0	68.0	68.8	66.3	58.4
BP+OSVM	62.9	**75.3**	39.2	0.7	84.4	**92.4**	72.9	0.9	33.8	40.5	21.4	44.3	60.4	69.4	44.5	15.3
Ours	**63.0**	59.1	**71.0**	**82.3**	**92.3**	91.2	**94.4**	**97.6**	**92.1**	**94.9**	**88.1**	78.0	**82.4**	**81.7**	**84.5**	**85.9**

Table 5. Accuracy (%) of experiments on digits datasets.



(a) Source Only (b) MMD (c) BP (d) Ours

Blue: Source Known, Red: Target Known, Green: Target Unknown

BP aligns target unknown with source known whereas ours rejects the target unknown.

Unsupervised Domain Adaptation for Object Detection

- Can we realize object detection using domain matching method?
- Source: w/ category and bounding box
- Target: w/o category and bounding box

Strong Global Distribution Alignment

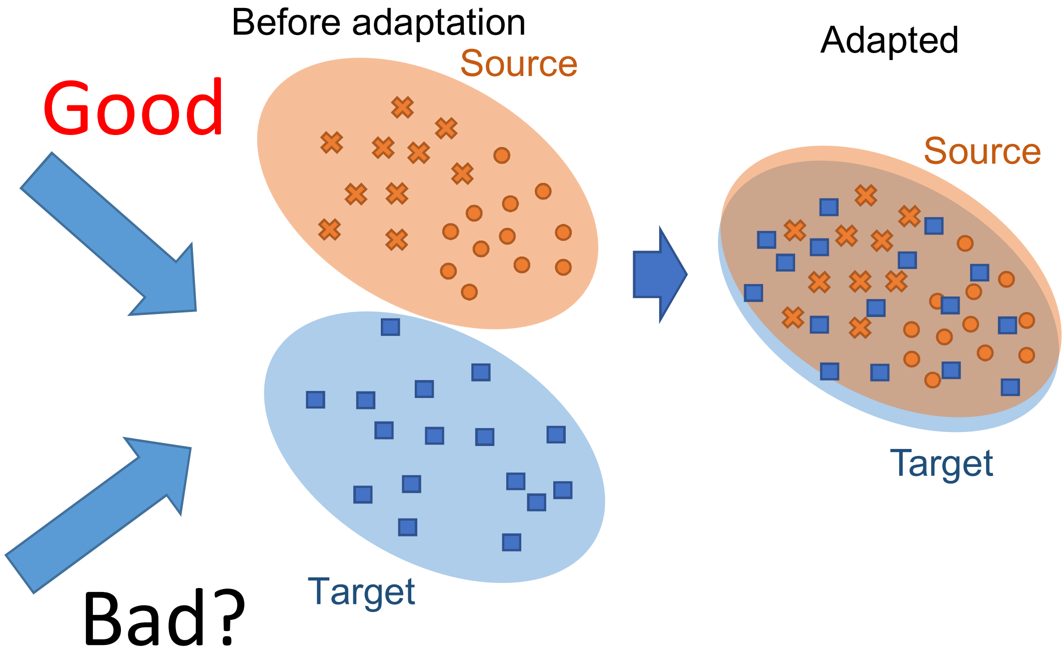
Similar Domains

Source **Target**

Dissimilar Domains

Source **Target**

Strong global distribution alignment



Layout, number and combination of objects can be different.

Strong Instance Distribution Alignment

Similar Domains

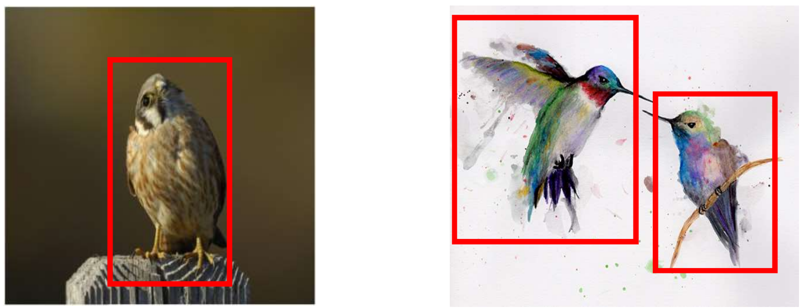
Source **Target**

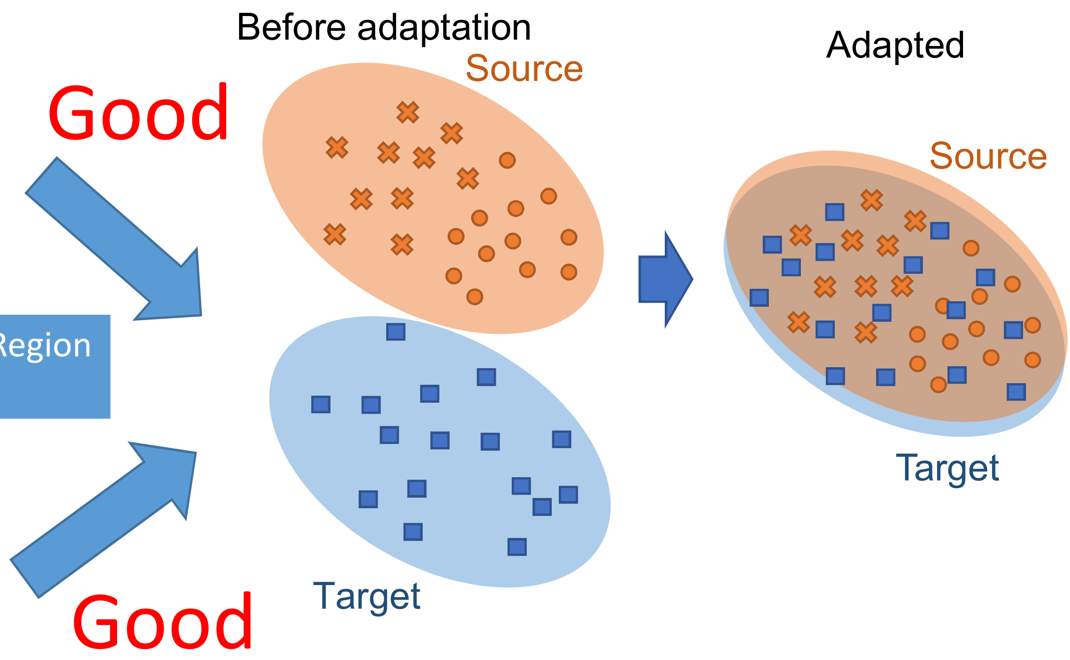
Strong instance distribution alignment

Dissimilar Domains

How to obtain good Region Proposal Networks?

Source **Target**





❑Problem

- To effectively conduct feature alignment, region proposals have to precisely localize objects of interest.

Problems of UDA for Object Detection

- Global distribution alignment
  - Strong global distribution alignment is not appropriate for object detection.
- Instance distribution alignment
  - Strong instance distribution alignment might be appropriate.
  - However, it is hard to obtain good region proposals in the target domain, because there are no ground truth bounding boxes in the target domain.

Strong-Weak Distribution Alignment for Adaptive Object Detection

Kuniaki Saito[1], Yoshitaka Ushiku[2], Tatsuya Harada[2, 3], Kate Sanenko[1]
1: Boston University, 2: The University of Tokyo, 3: RIKEN

To be appeared in CVPR 2019

K. Saito

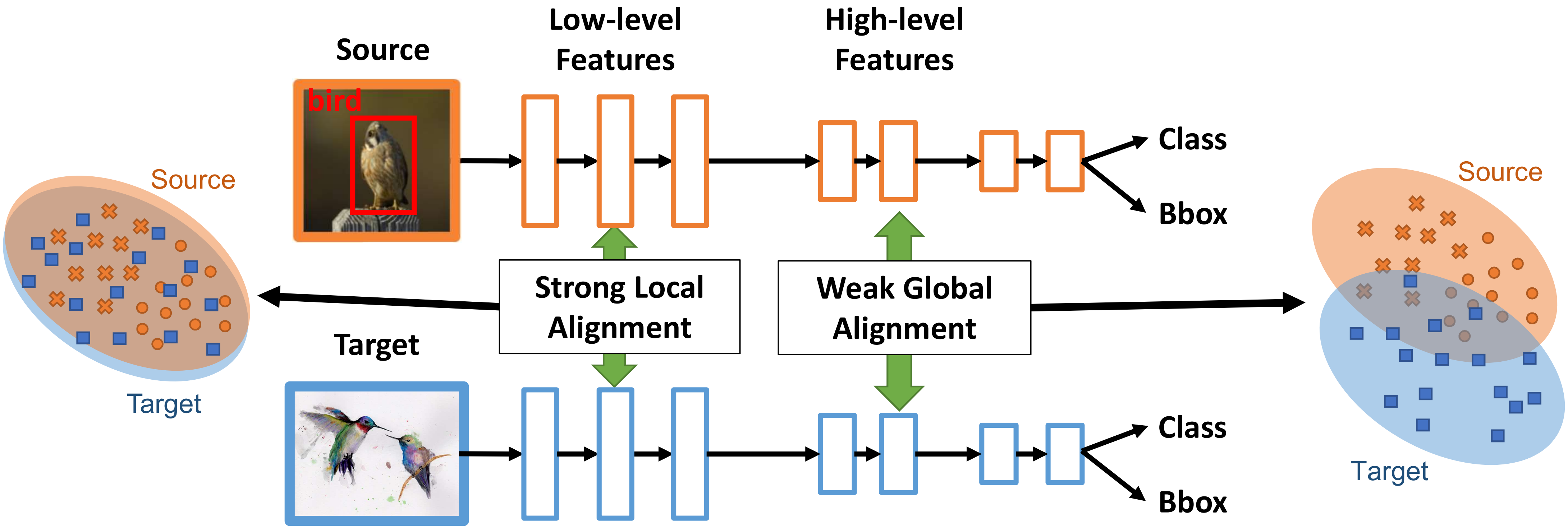
Y. Ushiku

T. Harada

K. Sanenko

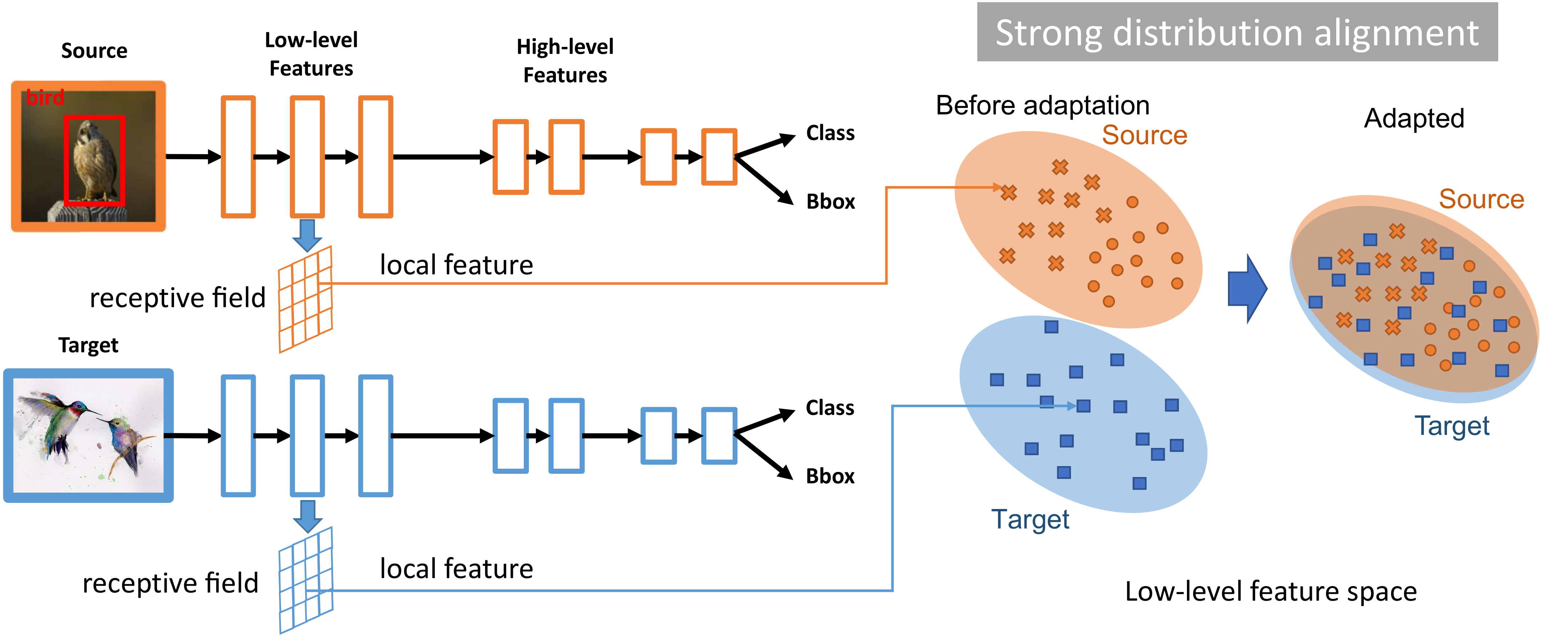
Key Idea

- **Weak Global Alignment** ⟶ High Level Feature (Category)
- **Strong Local Alignment** ⟶ Low Level Feature (Texture, Color)



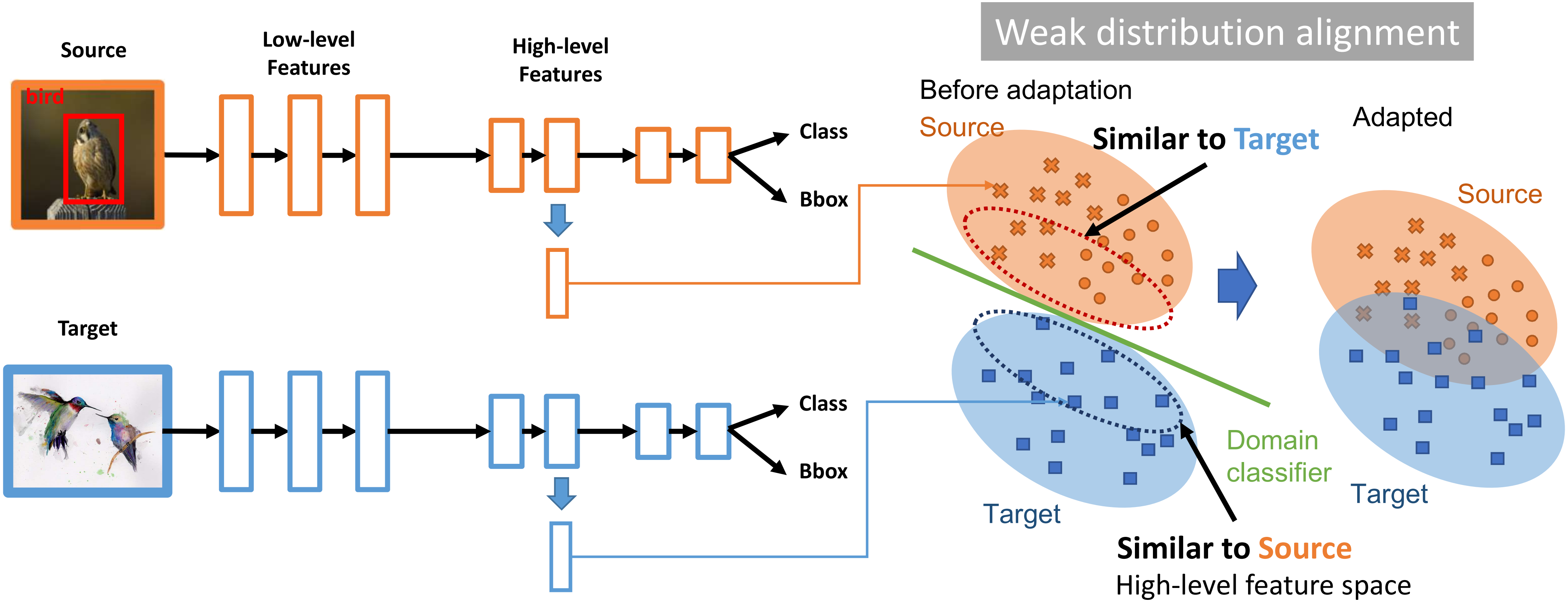
Proposal: Strong Local Alignment

- Domain invariant local features
- Extraction of local feature from each receptive field in low-level layer



Proposal: Weak Global Alignment

- Alignment of high level-features by force degrades DA performance.
- Partial alignment of high-level features



Proposal: Weak Global Alignment

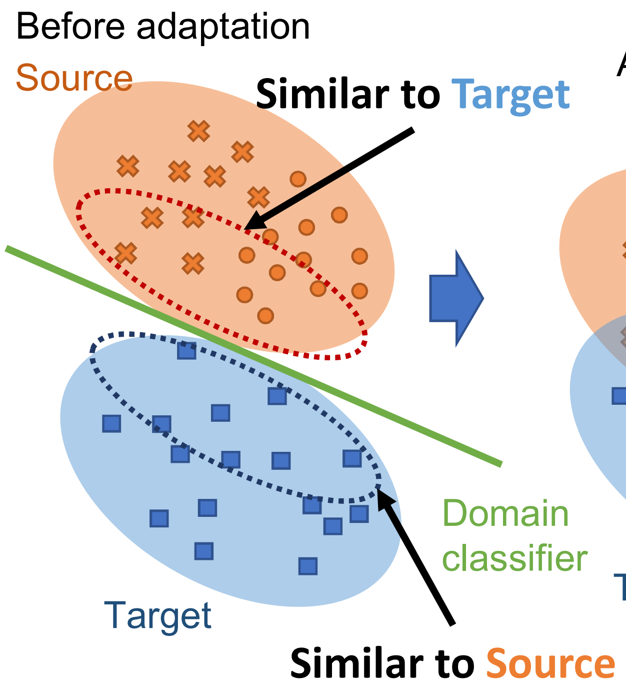
- Similar examples for each domain are hard-to-classify examples with domain classifier.
- Objective of domain classifier
  - Higher weight on hard examples
  - Lower weight on easy examples

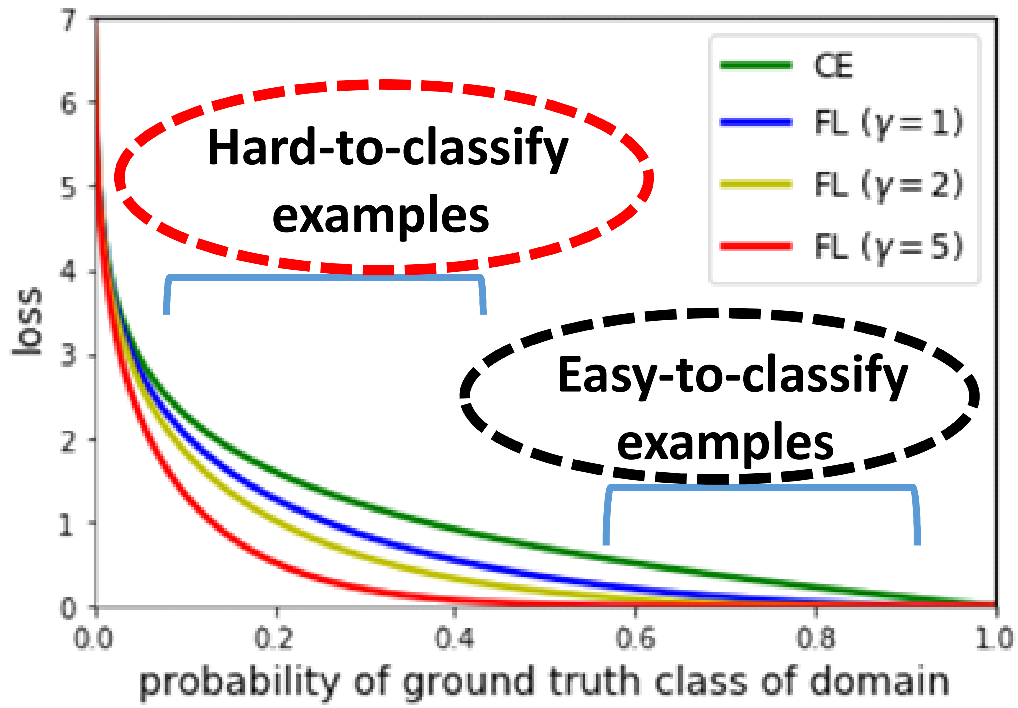
$$\boxed{-f(p_t)\log(p_t)}$$

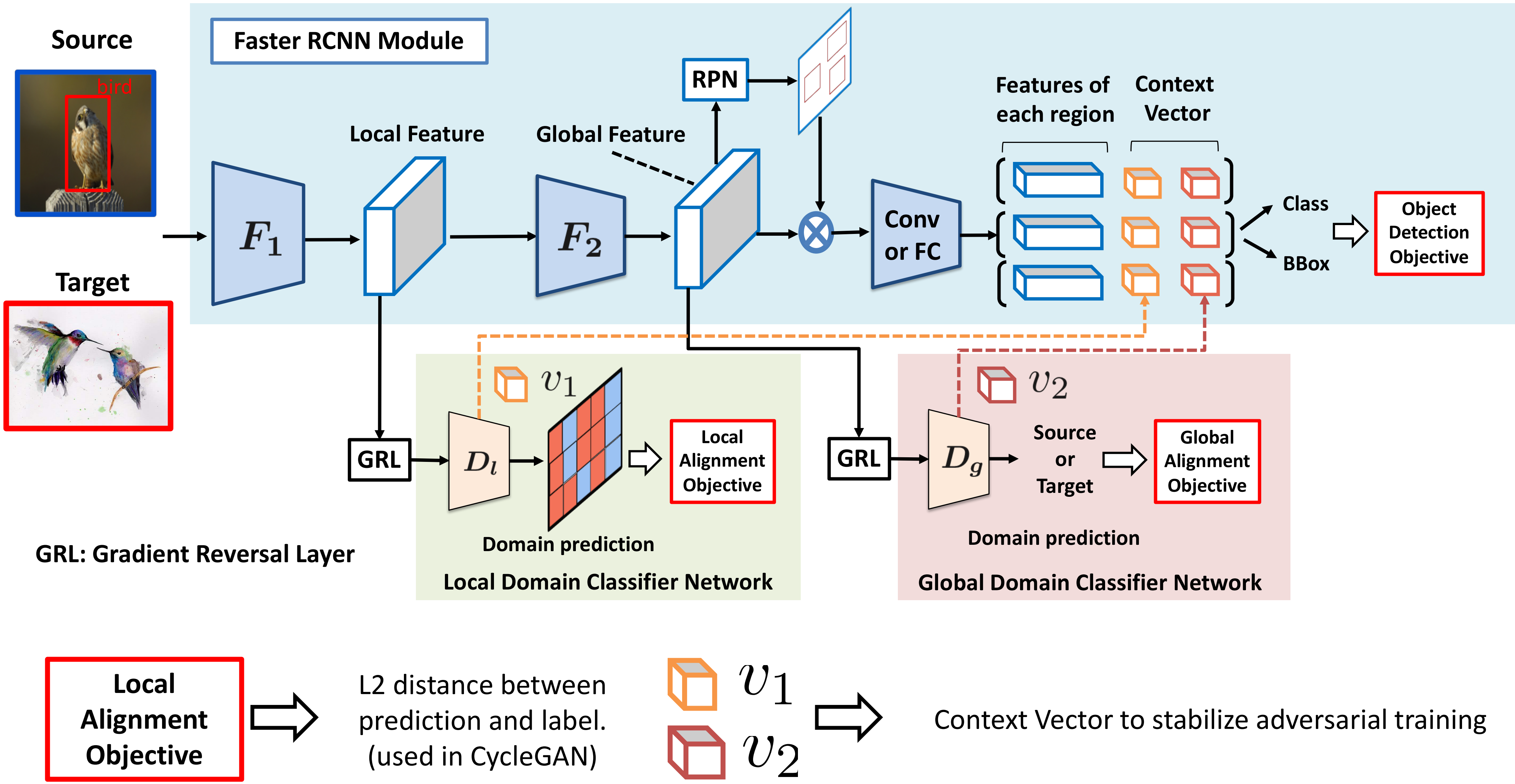
$$p_t = \begin{cases} p & \text{if } d = 1 \\ 1-p & \text{otherwise.} \end{cases}$$

Focal loss (T.-Y. Lin+, ICCV17)

$$f(p_t) = (1-p_t)^{\gamma}$$





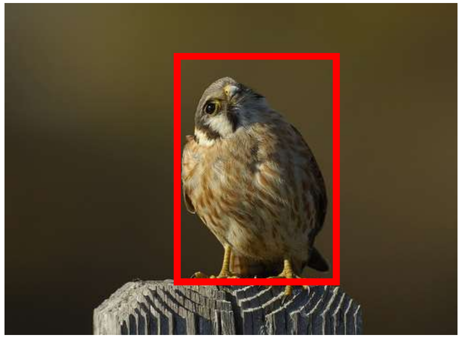

Source
bird
Target
Faster RCNN Module
Local Feature
Global Feature
RPN
F_1
F_2
Conv or FC
Features of each region
Context Vector
Class
BBox
Object Detection Objective
GRL
D_l
v_1
Local Alignment Objective
Domain prediction
Local Domain Classifier Network
GRL
D_g
v_2
Source or Target
Global Alignment Objective
Domain prediction
Global Domain Classifier Network
GRL: Gradient Reversal Layer
Local Alignment Objective
L2 distance between prediction and label. (used in CycleGAN)
v_1
v_2
Context Vector to stabilize adversarial training

Experiment 1: Adaptation Between Dissimilar Domains

- Pascal VOC to Clipart and Watercolor

Source domain

Pascal VOC



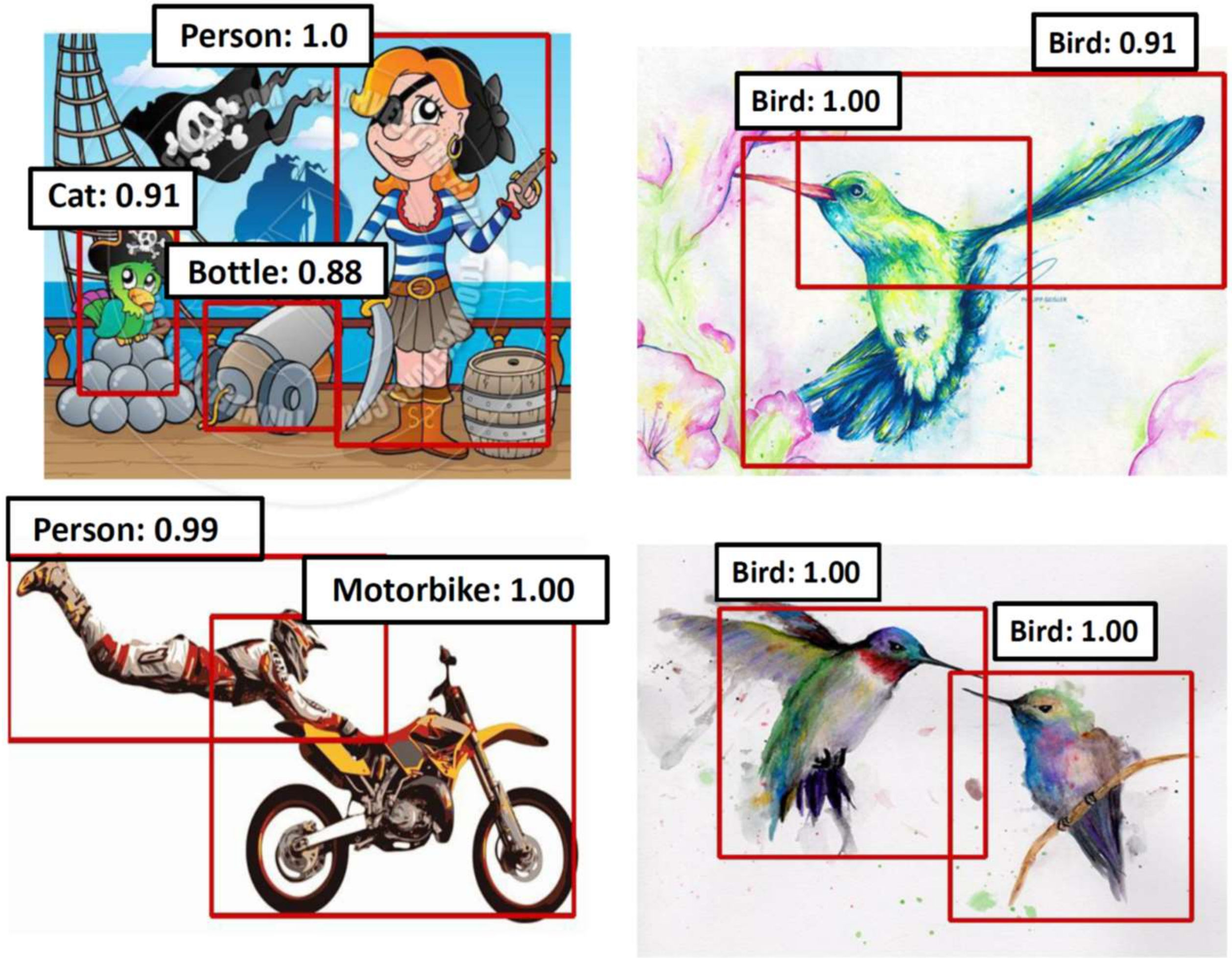
Target domain

Clipart

Watercolor

Experiment 1: Adaptation Between Dissimilar Domains

- Pascal VOC to Clipart and Watercolor



Results on Clipart

G: Global Alignment, I: Instance, CTX: Context Vector, L: Local, P: Pixel

Method	G	I	CTX	L	P	aero	bcycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	hrs	bike	prsn	plnt	sheep	sofa	train	tv	MAP
Faster RCNN						**35.6**	52.5	24.3	23.0	20.0	43.9	32.8	10.7	30.6	11.7	13.8	6.0	**36.8**	45.9	48.7	41.9	**16.5**	7.3	22.9	32.0	27.8
BDC-Faster	✓					20.2	46.4	20.4	19.3	18.7	41.3	26.5	6.4	33.2	11.7	**26.0**	1.7	36.6	41.5	37.7	44.5	10.6	20.4	33.3	15.5	25.6
DA-Faster	✓	✓				15.0	34.6	12.4	11.9	19.8	21.1	23.2	3.1	22.1	26.3	10.6	10.0	19.6	39.4	34.6	29.3	1.0	17.1	19.7	24.8	19.8
	✓					30.5	48.5	**33.6**	24.8	41.2	48.9	32.4	17.2	34.5	55.0	19.0	13.6	35.1	66.2	**63.0**	45.3	12.5	22.6	45.0	38.9	36.4
				✓		19.8	50.7	25.4	21.7	30.2	47.2	27.1	8.5	33.5	26.8	14.0	11.7	31.5	62.0	49.9	39.6	9.1	23.8	39.5	38.4	30.5
Proposed	✓		✓			31.7	**55.2**	30.9	26.8	**43.4**	47.5	**40.0**	7.9	**36.7**	50.0	14.3	**18.0**	29.2	**68.1**	62.3	50.4	13.4	24.5	**54.2**	45.8	37.5
	✓		✓	✓		26.2	48.5	32.6	**33.7**	38.5	**54.3**	37.1	**18.6**	34.8	**58.3**	17.0	12.5	33.8	65.5	61.6	**52.0**	9.3	**24.9**	54.1	**49.1**	**38.1**
	✓		✓	✓	✓	31.1	53.7	28.9	24.9	40.3	49.0	38.1	14.6	41.9	43.8	15.3	7.2	27.9	75.5	57.3	41.8	6.7	23.3	48.5	44.1	35.7

9.8%

- Strong global alignments (BDC-Faster (27.8 -> 25.6 %), DA-Faster (27.8 -> 19.8 %)) degrade performance.
- Weak global alignment improves performance 9.8 % (25.6 -> 36.4 %).
- Strong local alignment improves performance 2.7 % (27.8 -> 30.5 %).
- The method with weak global alignment, strong local alignment and context vector is the best (38.1 %).

Pascal VOC

Clipart

Results on Watercolor

G: Global Alignment, I: Instance, CTX: Context Vector, L: Local, P: Pixel

Table 1: Results on Watercolor.

Method	G	I	CTX	L	P	AP on a target domain bike	bird	car	cat	dog	prsn	MAP
Faster RCNN						68.8	46.8	37.2	32.7	21.3	60.7	44.6
BDC-Faster	✓					68.6	48.3	47.2	26.5	21.7	60.5	45.5
DA-Faster	✓	✓				75.2	40.6	48.0	31.5	20.6	60.0	46.0
	✓					66.4	53.7	43.8	37.9	31.9	65.3	49.8
				✓		79.4	54.8	47.2	37.1	31.5	62.4	52.1
Proposed	✓		✓			71.3	52.0	46.6	36.2	29.2	67.3	50.4
	✓		✓	✓		82.3	**55.9**	46.5	32.7	35.5	66.7	53.3
	✓		✓	✓	✓	**90.5**	54.8	**49.4**	**38.6**	**38.8**	**67.9**	**56.7**
Oracle						83.6	59.4	50.7	43.7	39.5	74.5	58.6

Local-level was effective

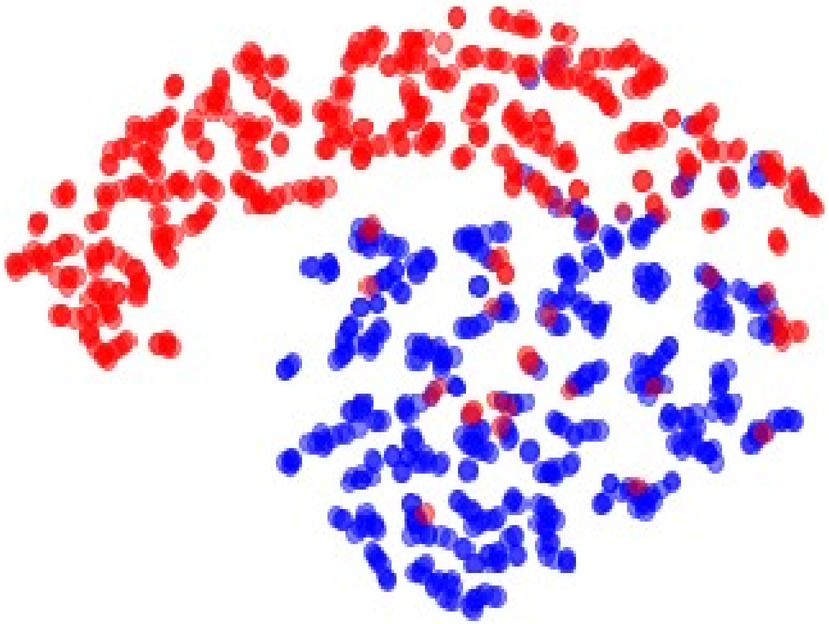
Oracle-level performance

- Weak global alignment improves performance 4.3 % (45.5 -> 49.8 %).
- Strong local alignment improves performance 7.5 % (44.6 -> 52.1 %).
- The method with weak global alignment, strong local alignment, context vector and pixel level alignment is the best (38.1 %).

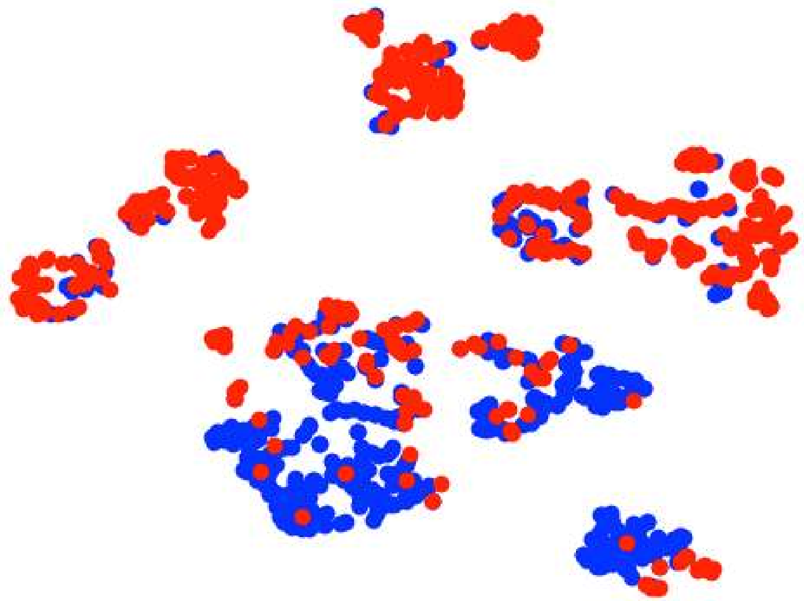
Pascal VOC

Watercolor

Ours (Weak Global Alignment Only)
(MAP: 36.4)



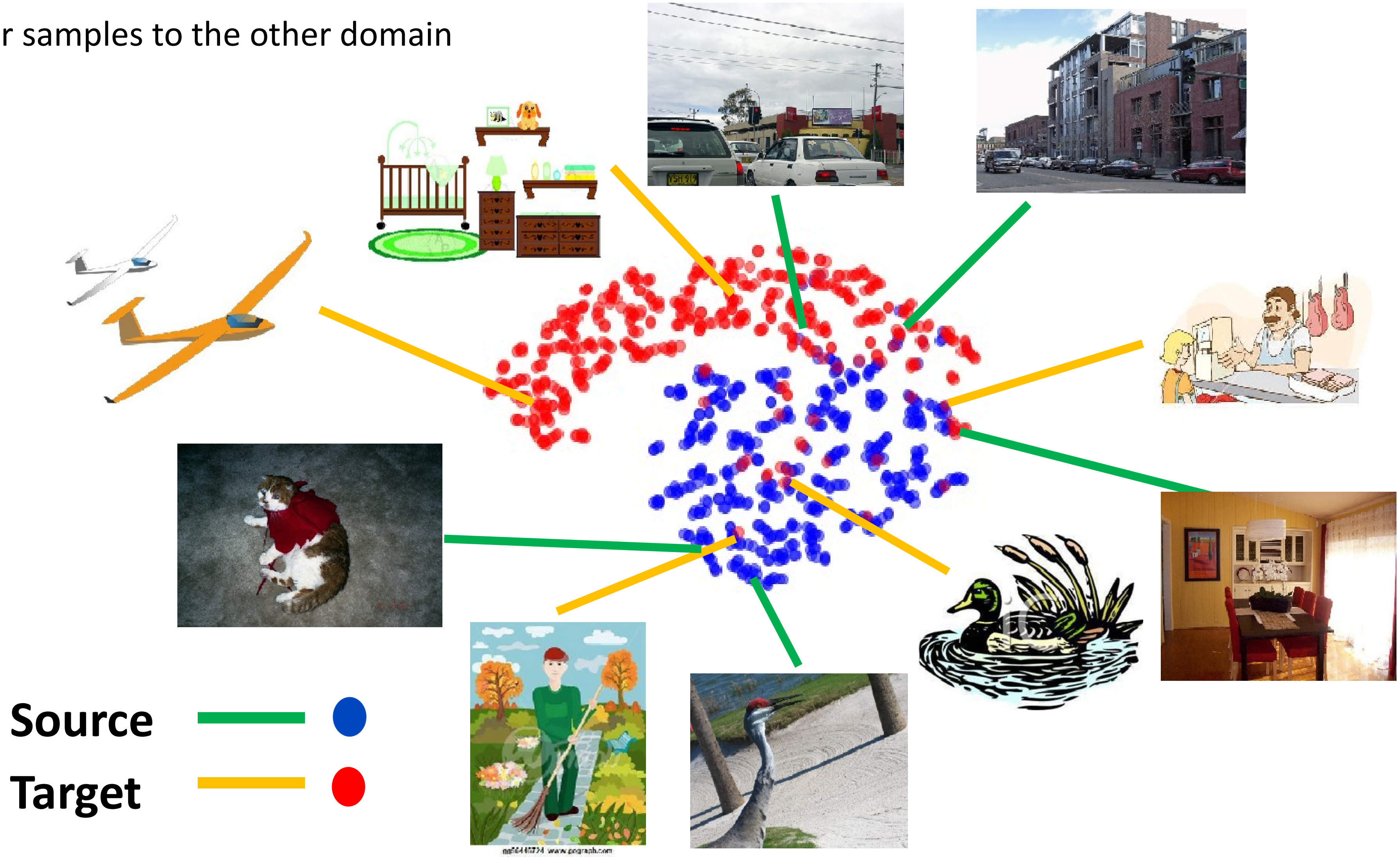
Baseline DC Method
(MAP: 25.6)



- The results of adaptation between dissimilar domains (from pascal to clipart).
- Blue: source examples, Red: target examples

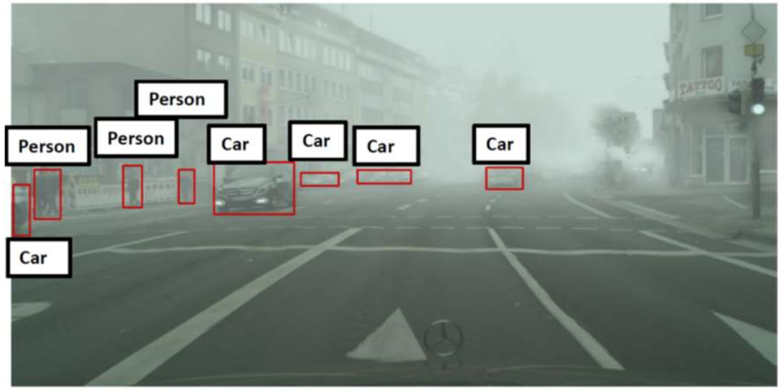
Global-Weak Alignment

- Focus on similar samples to the other domain



Experiment 2: Adaptation Between Similar Domains

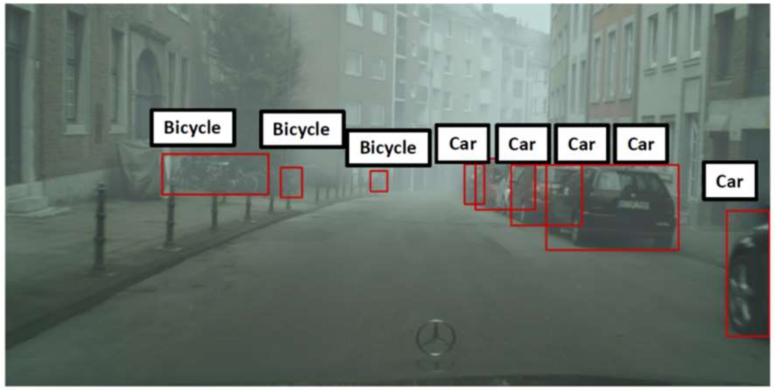
- **Cityscape to FoggyCityscape**



G: Global Alignment, I: Instance, CTX: Context Vector, L: Local

Table 1:

Method	G	I	CTX	L	AP on a target domain bus	bcycle	car	bike	prsn	rider	train	truck	MAP
Faster RCNN					22.3	26.5	34.3	15.3	24.1	33.1	3.0	4.1	20.3
BDC-Faster	✓				25.0	31.0	40.5	22.1	**35.3**	20.2	20.0	**27.1**	27.6
DA-Faster	✓	✓			33.1	23.3	25.5	15.6	23.4	29.0	10.9	19.6	22.5
Proposed	✓				33.5	33.3	42.7	22.2	27.1	40.3	11.6	22.3	29.1
				✓	34.3	32.2	36.2	23.7	27.5	39.3	5.4	24.4	27.9
	✓		✓		**38.0**	31.2	41.8	20.7	26.6	37.6	19.7	20.5	29.5
	✓		✓	✓	36.2	**35.3**	**43.5**	**30.0**	29.9	**42.3**	**32.6**	24.5	**34.3**
Oracle					50.0	36.2	49.7	34.7	33.2	45.9	37.4	35.6	40.3



Experiment 3: Adaptation from Synthetic to Real

G: Global Alignment, I: Instance, CTX: Context Vector, L: Local, P: Pixel

GTA

Cityscape

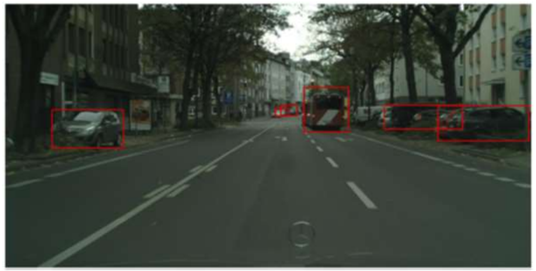


Table 1: .

Method	G	I	CTX	L	P	AP on Car
Faster RCNN						34.6
BDC-Faster	✓					31.8
DA-Faster	✓	✓				34.2
Proposed (FL)	✓					36.4
				✓		40.2
					✓	40.0
	✓		✓			38.2
	✓		✓	✓		40.1
	✓		✓		✓	41.5
	✓		✓	✓	✓	40.7
Proposed Method with different parameters						
EFL	✓		✓			38.7
FL ($\gamma = 3$)	✓		✓			42.3
FL ($\gamma = 3$)*	✓		✓		✓	47.7
Oracle						53.1

- **Pixel-level, local level adaptation are good.**
- **Combining pixel-level and our adaptation is better.**
- **EFL performs better than baselines**

Weak global alignment is effective !

Visualization of Domain Evidence

- Visualization of the evidence for the global-level domain classifier’s prediction using Grad-cam
- Evidence for why the domain classifier thinks the image comes from the source or the target
- The feature extractor seems to focus on cars to deceive the domain classifier.

Take Home Messages

- Learning from Limited Data
  - Knowledge Transfer
  - Domain Adaptation
  - Between-class learning

- Between-class learning (BC learning)
  - Mix two training examples with a random ratio
  - Train the model to output the mixing ratio
  - Simple to implement

- Unsupervised domain adaptation
  - Considering class specific distribution matching and adversarial training are effective for unsupervised domain adaptation.

- Open set domain adaptation
  - Giving an option for the feature extractor to select known or unknown patterns is practical in the open set domain adaptation.

- Adaptive Object Detection
  - Weak global feature alignment and strong local feature alignment are effective for adaptive object detection.