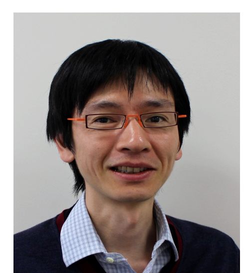
THE UNIVERSITY OF TOKYO AIP Machine Intelligence Laboratory

Maximum Classifier Discrepancy for Unsupervised Domain Adaptation

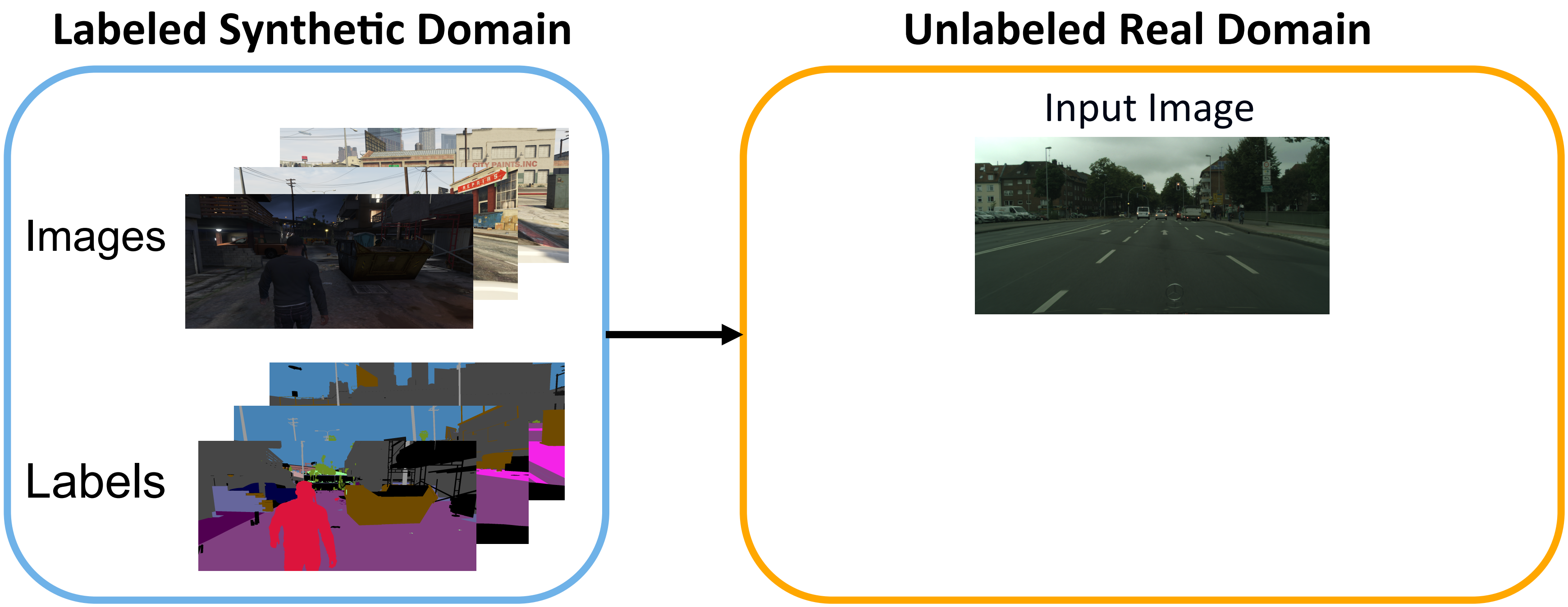
Kuniaki Saito[1], Kohei Watanabe[1], Yoshitaka Ushiku[1] and Tatsuya Harada[1,2]

The University of Tokyo[1], RIKEN AIP[2]

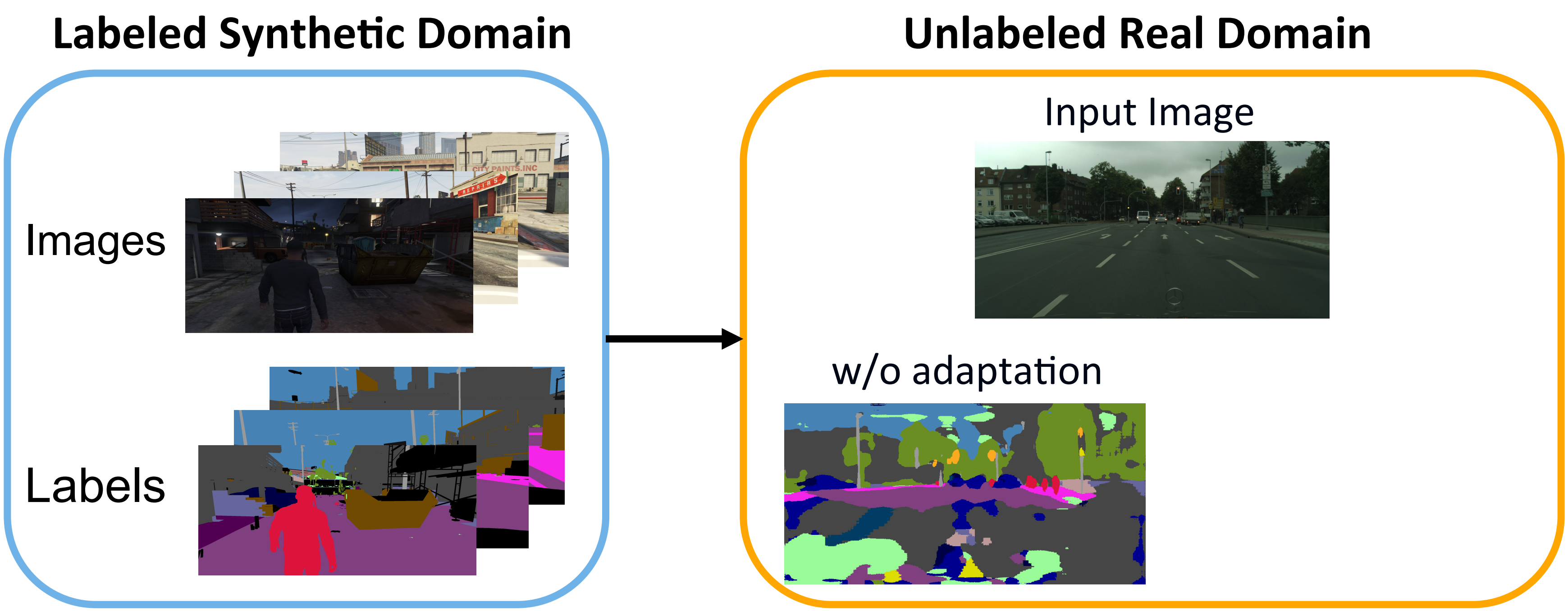
Background

- Problem: Cost to collect many labeled samples
- Solution: Transferring knowledge between different domains
  - Difficulty: Difference of domains



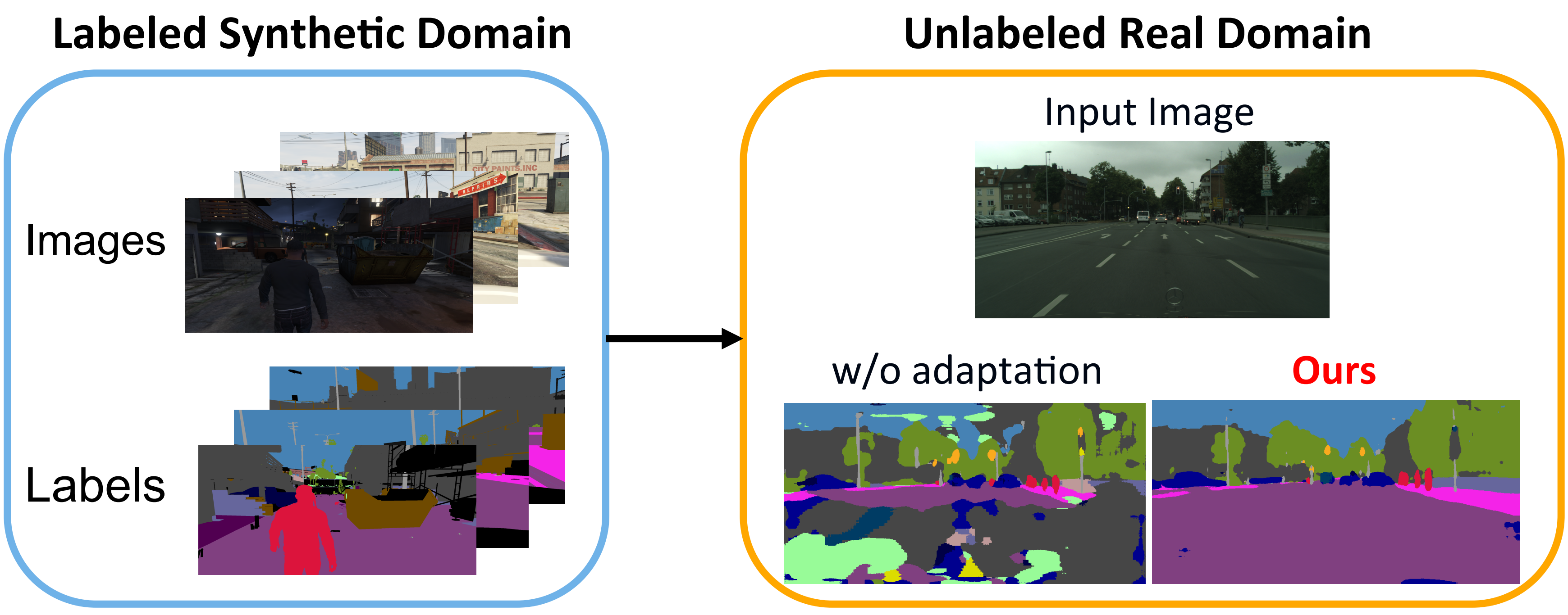
Background

- Problem: Cost to collect many labeled samples
- Solution: Transferring knowledge between different domains
  - Difficulty: Difference of domains



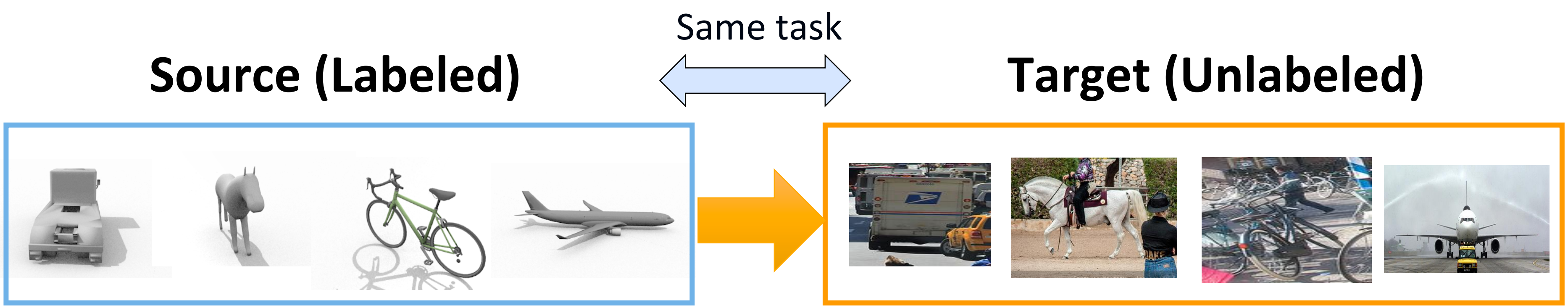
Background

- Problem: Cost to collect many labeled samples
- Solution: Transferring knowledge between different domains
  - Difficulty: Difference of domains



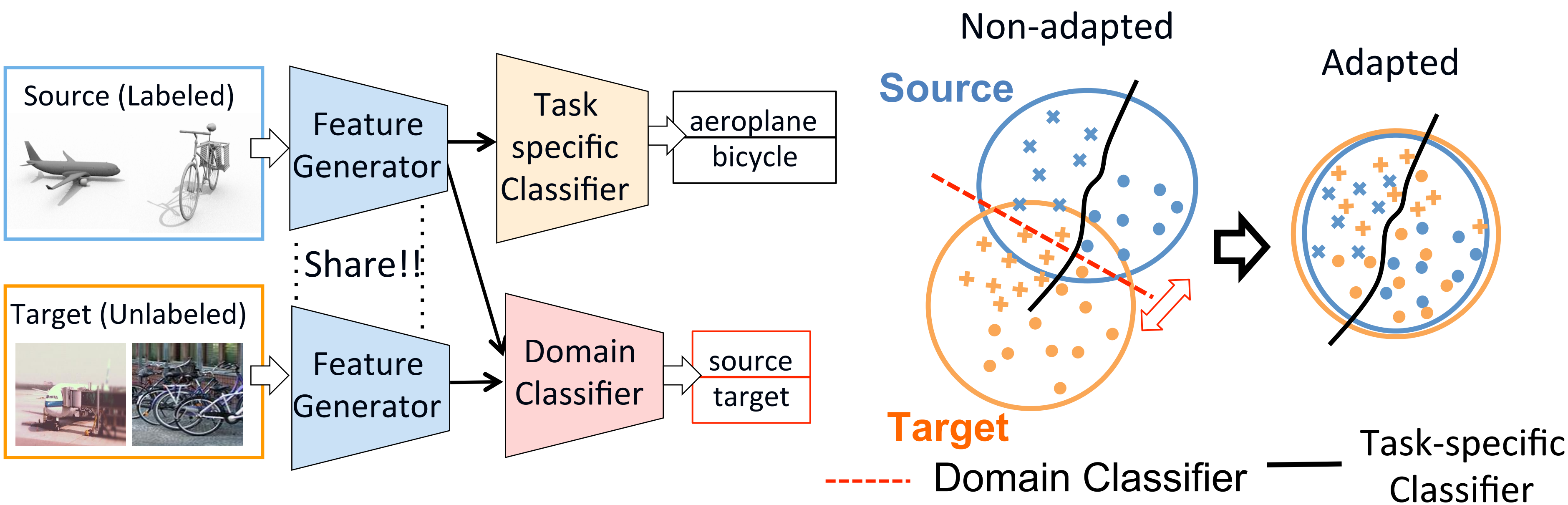
Domain Adaptation

- Goal
  - Transfer knowledge from source to target domain
  - Classifier that works well on target domain
- Unsupervised Domain Adaptation
  - Labeled source and unlabeled target samples



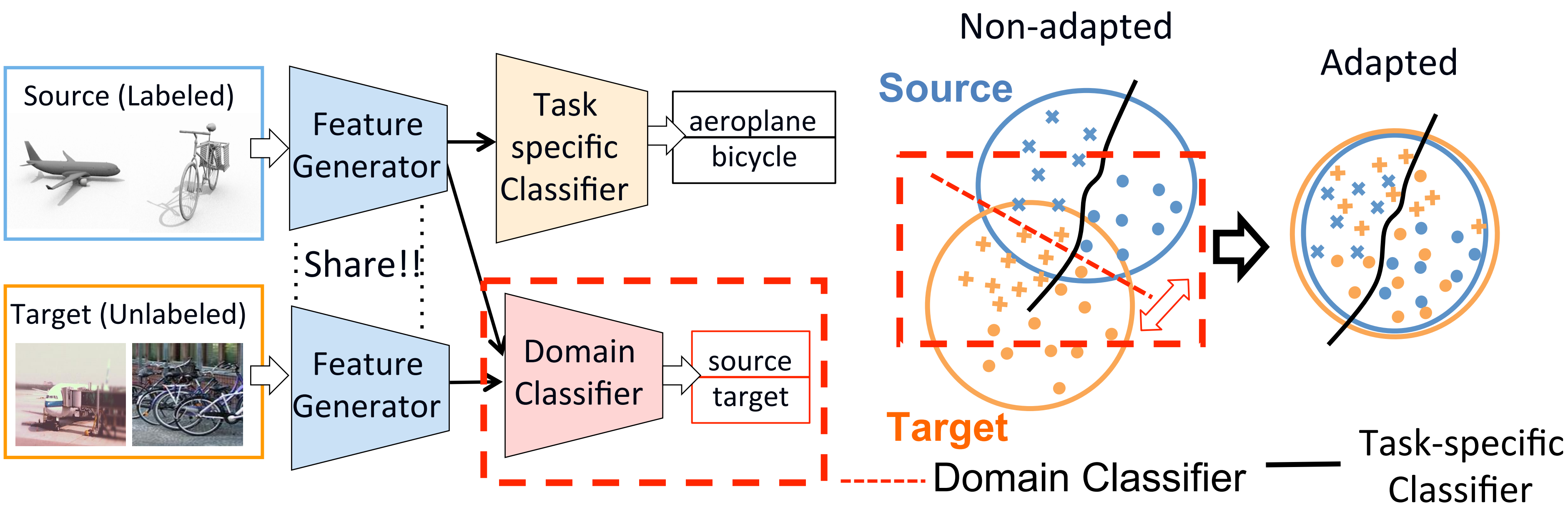
Popular Approach for UDA

- Distribution matching using a domain classifier
  - Domain Adversarial Neural Network [Ganin et al., ICML 2015]
  - Domain Classifier: Discriminate the domain of features
  - Feature Generator: Deceive the domain classifier



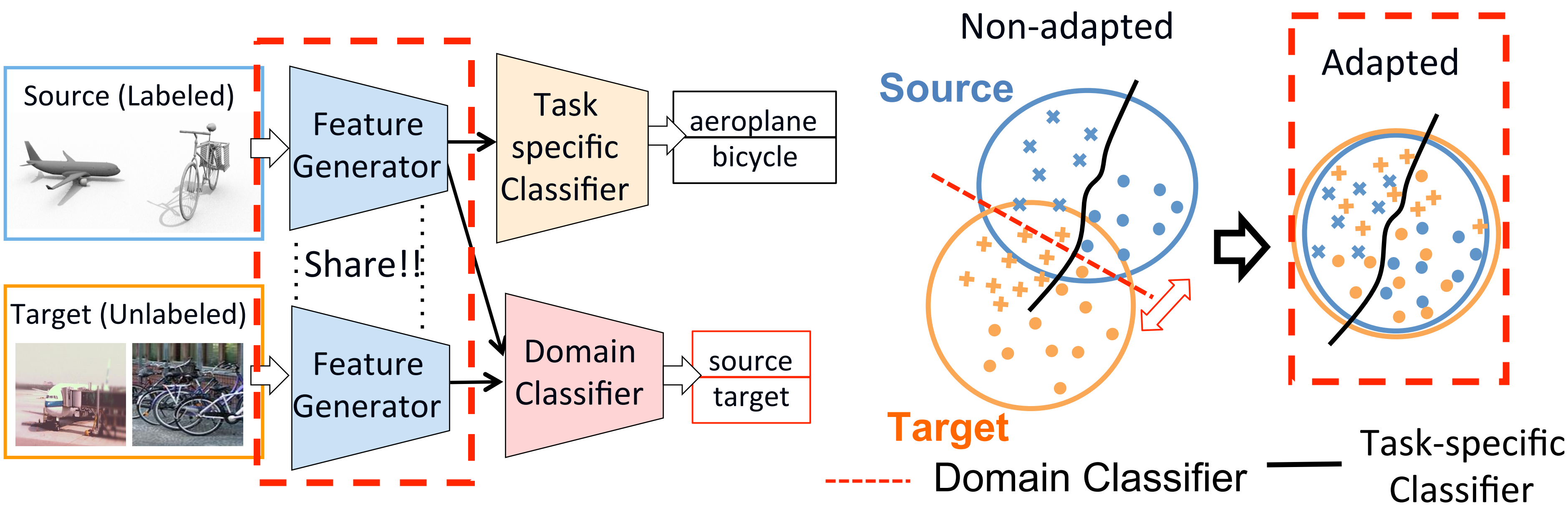
Popular Approach for UDA

- Distribution matching using a domain classifier
  - Domain Adversarial Neural Network [Ganin et al., ICML 2015]
  - Domain Classifier: Discriminate the domain of features
  - Feature Generator: Deceive the domain classifier



Popular Approach for UDA

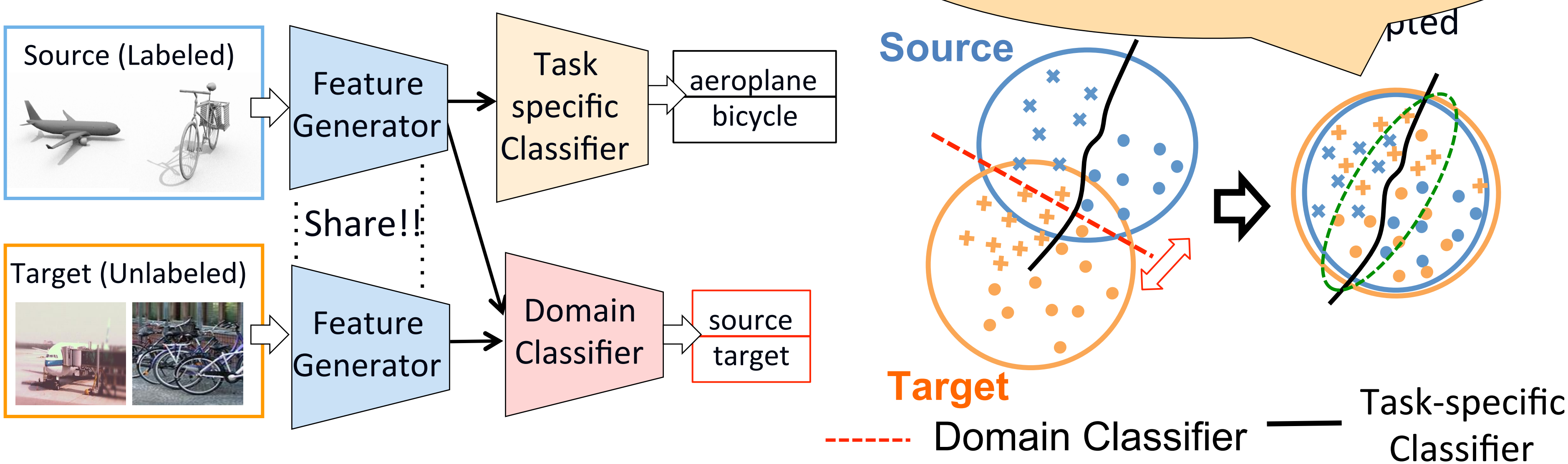
- Distribution matching using a domain classifier
  - Domain Adversarial Neural Network [Ganin et al., ICML 2015]
  - Domain Classifier: Discriminate the domain of features
  - Feature Generator: Deceive the domain classifier



Popular Approach for UDA

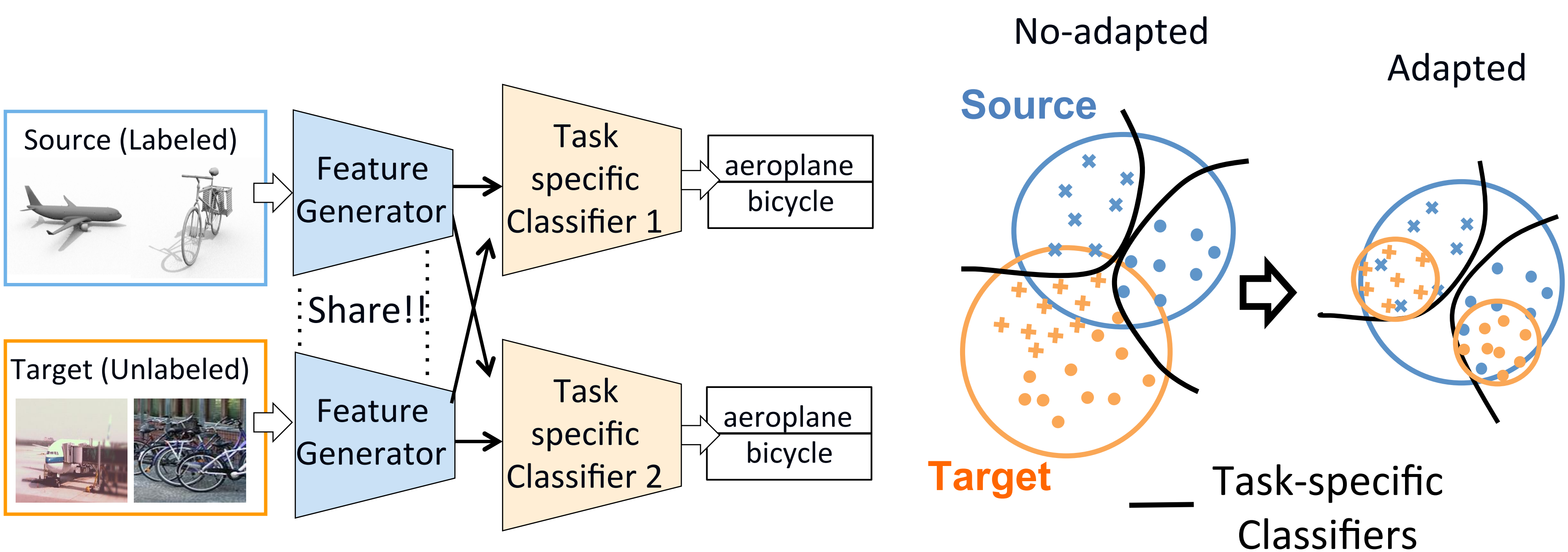
- Distribution matching using a domain classifier
  - Domain Adversarial Neural Network [Ganin et al., ICML 2015]
  - Domain Classifier: Discriminate the domain of features
  - Feature Generator: Deceive

Target features can be near a task-specific classifier's boundary.



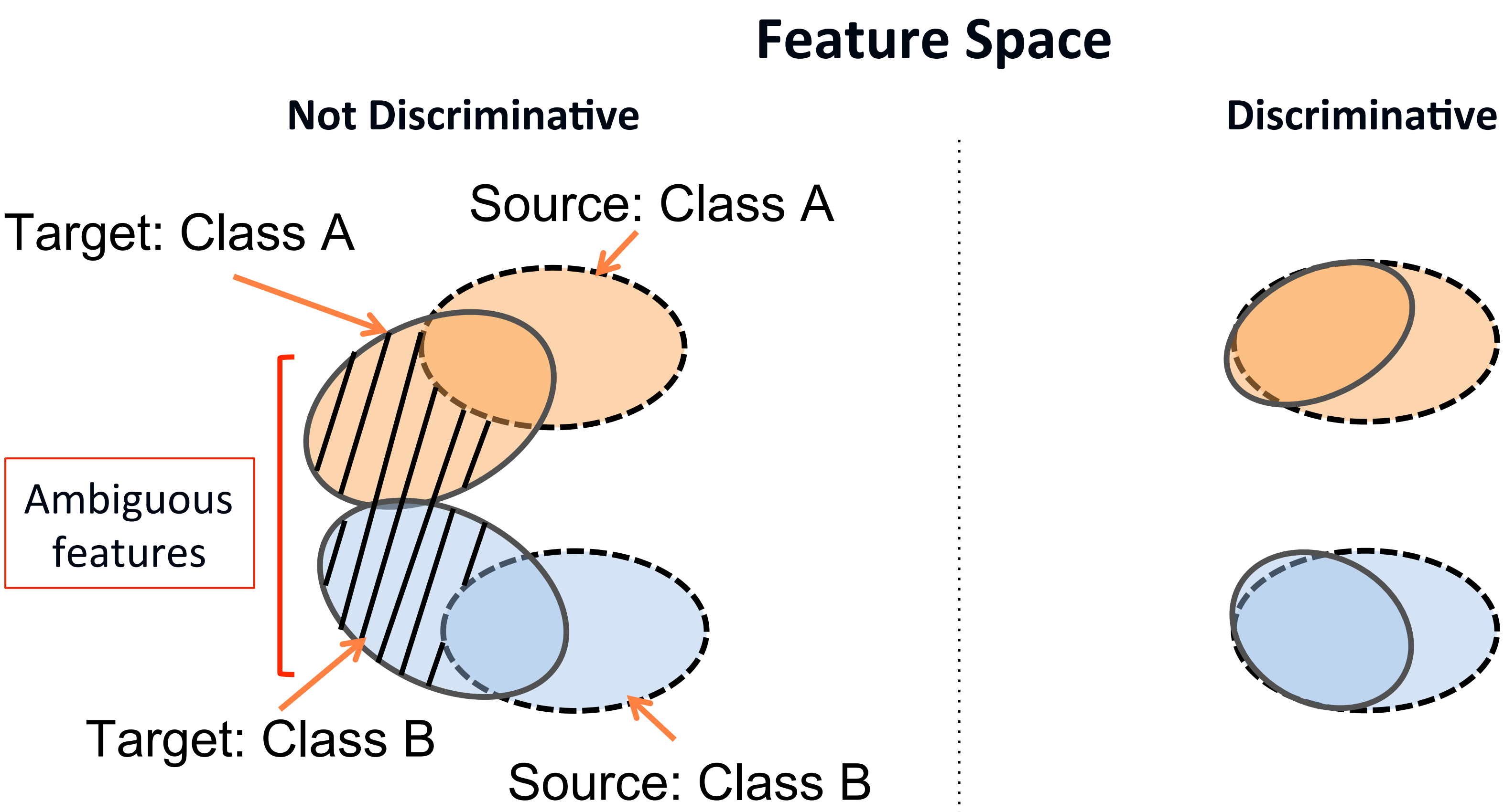
Proposed Method

- Task-specific classifier based distribution alignment method
  - Relationship between decision boundary and target features
  - Discriminative features



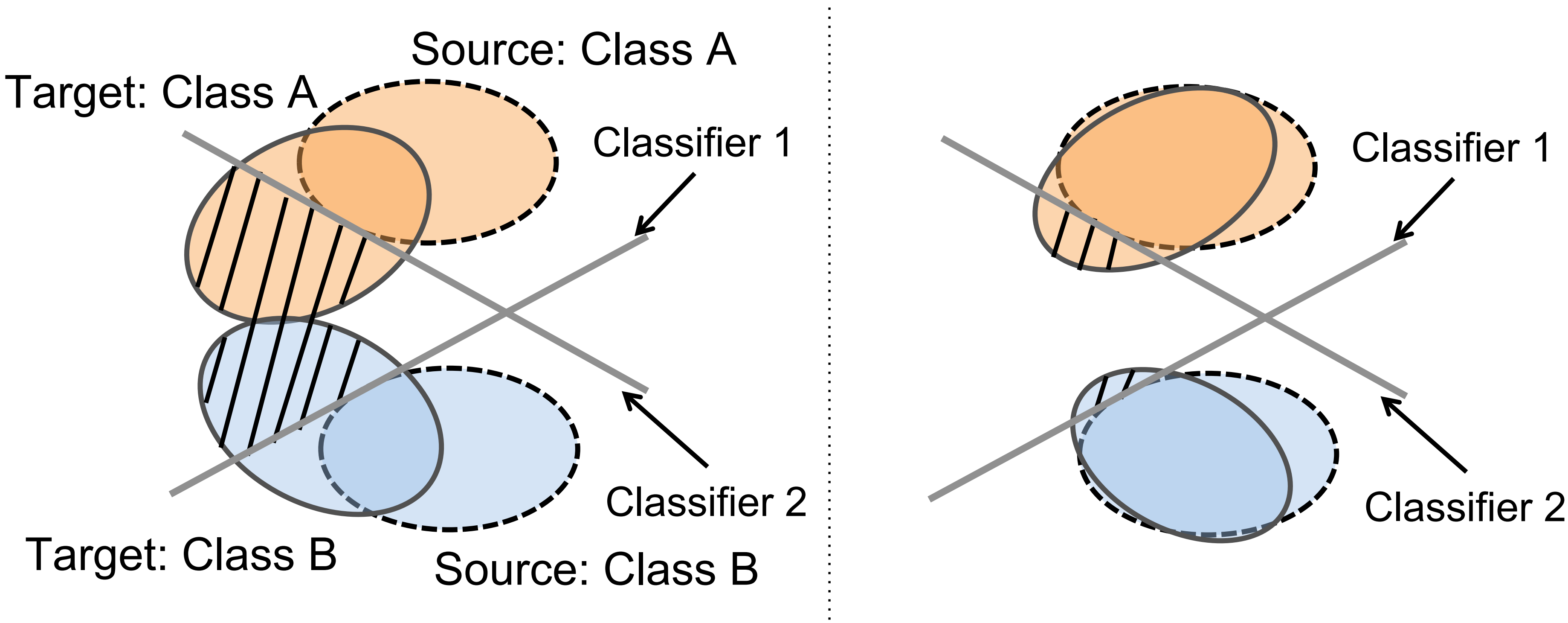
Requirement

- Detect target samples far from source ones
  - Such samples are likely to be misclassified



Key idea: Discrepancy

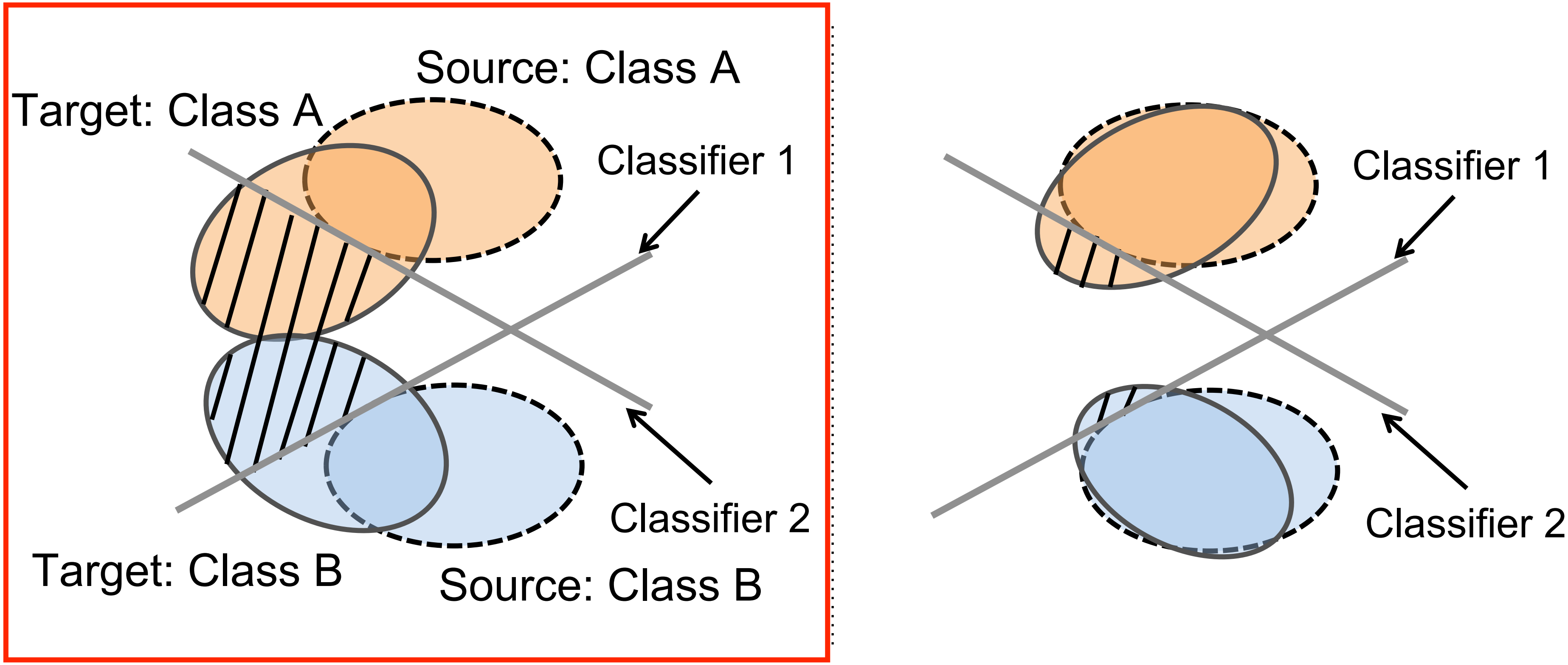
Discrepancy: Disagreement of task-specific classifiers



- Two distinct classifiers that classify source features correctly.
- Agree: Features are near source (discriminative).
- Disagree: Features are ambiguous (non-discriminative).

Key idea: Discrepancy

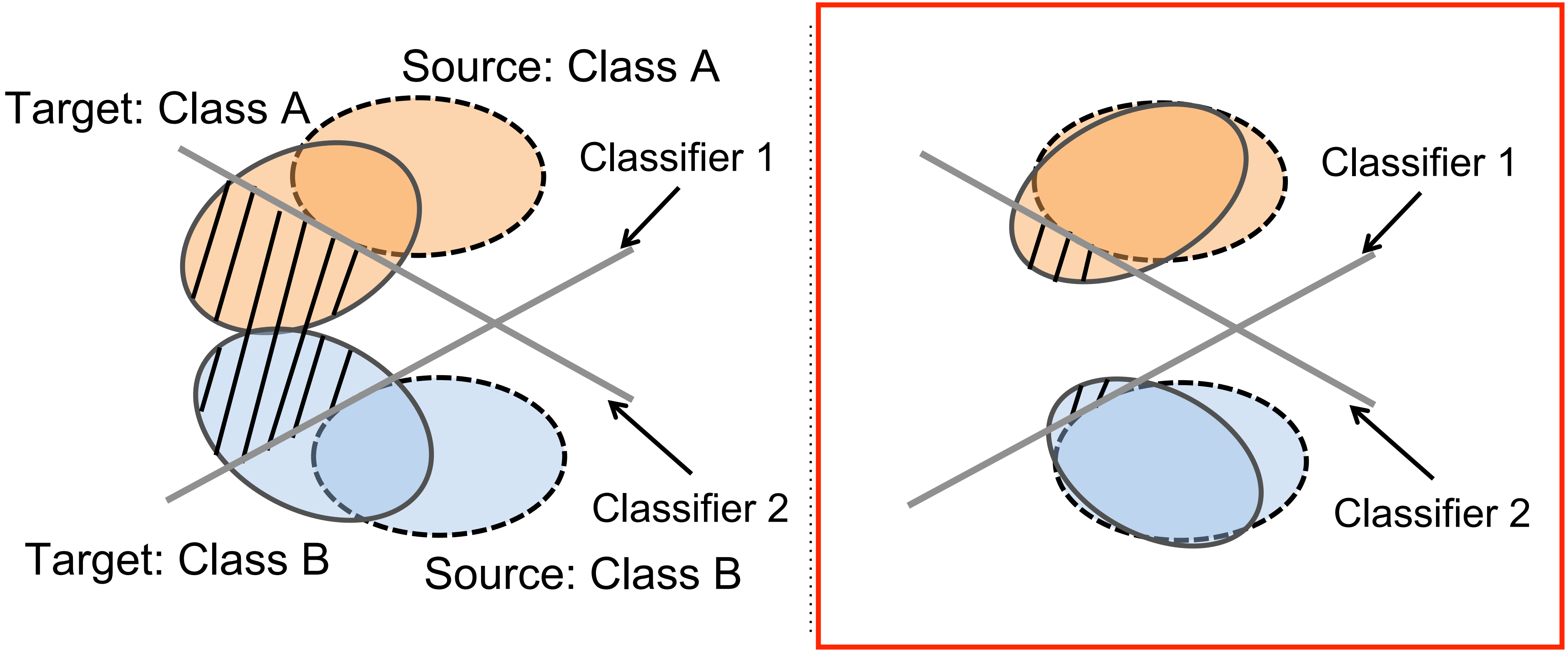
Discrepancy: Disagreement of task-specific classifiers



- Two distinct classifiers that classify source features correctly.
- Agree: Features are near source (discriminative).
- Disagree: Features are ambiguous (non-discriminative).

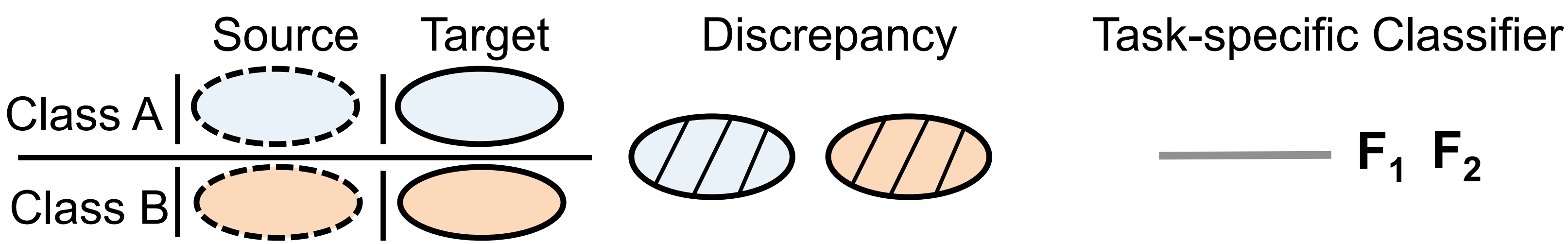
Key idea: Discrepancy

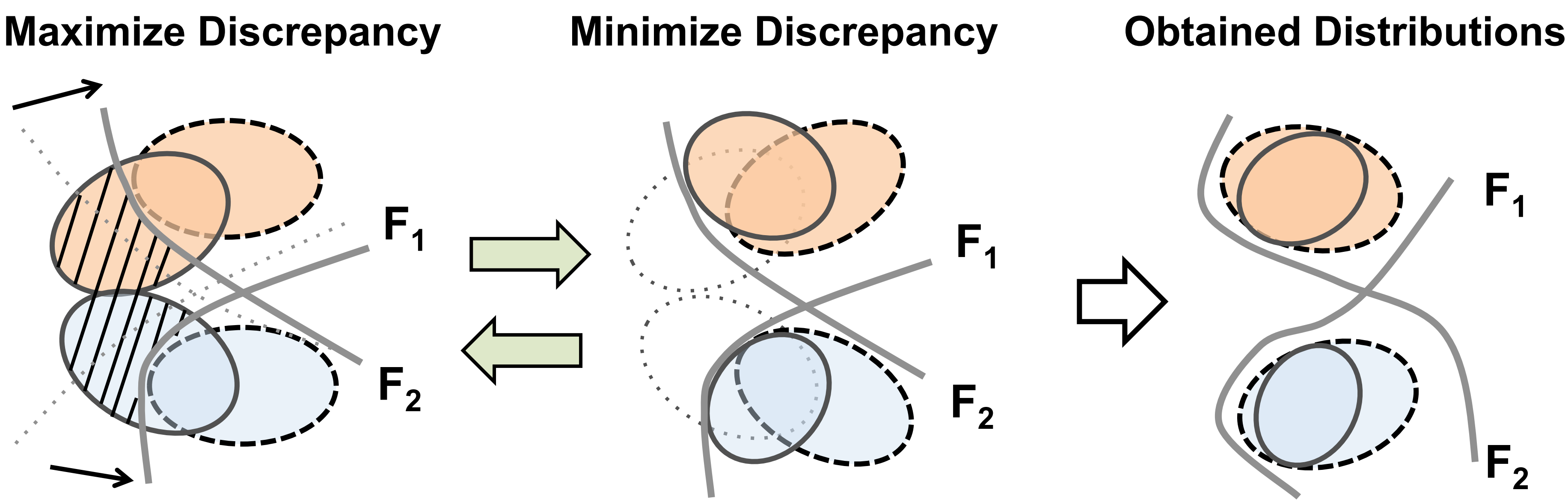
Discrepancy: Disagreement of task-specific classifiers



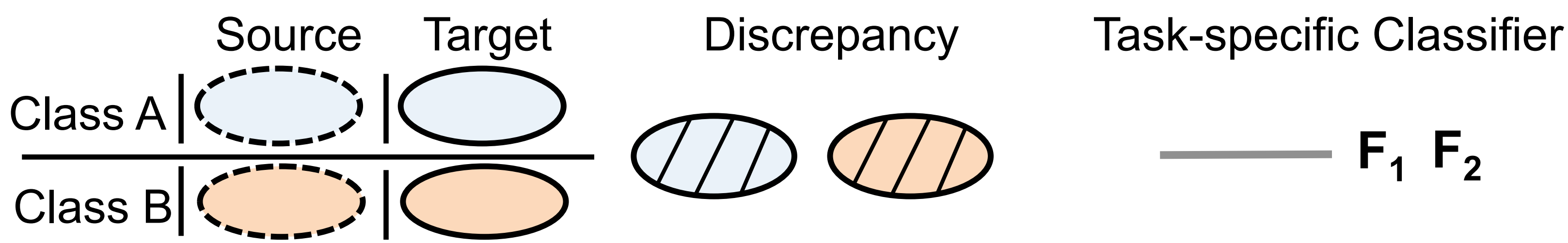
- Two distinct classifiers that classify source features correctly.
- Agree: Features are near source (discriminative).
- Disagree: Features are ambiguous (non-discriminative).

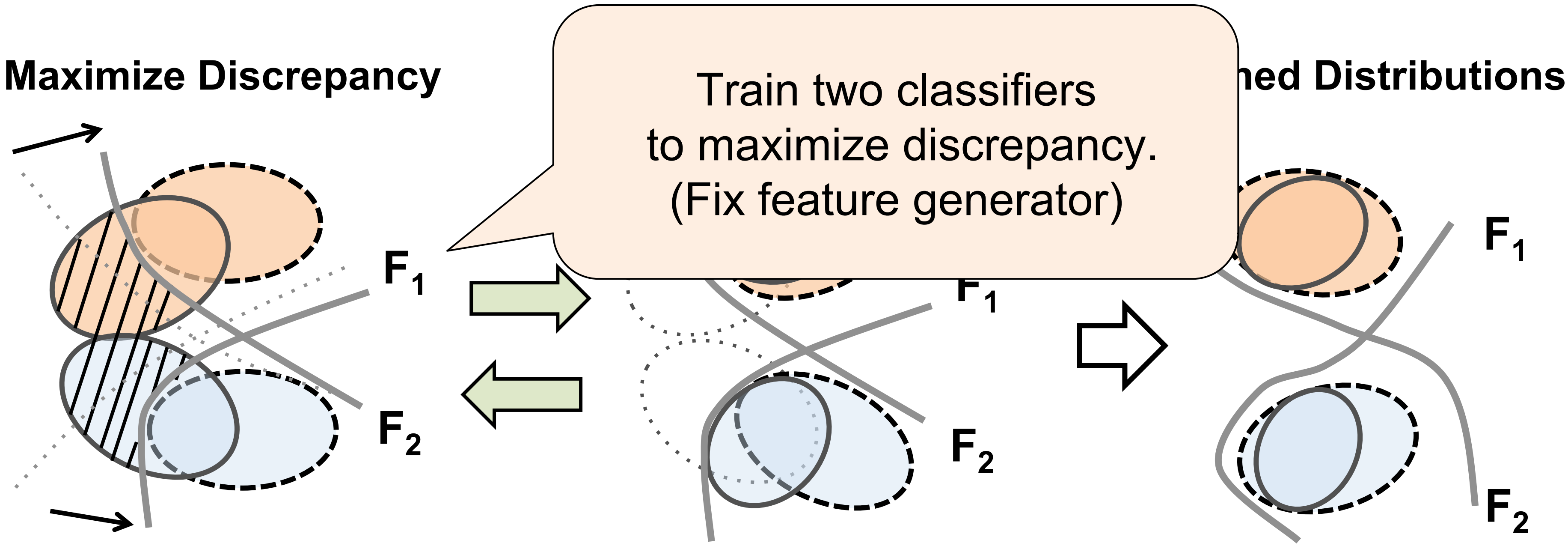
Main Procedure



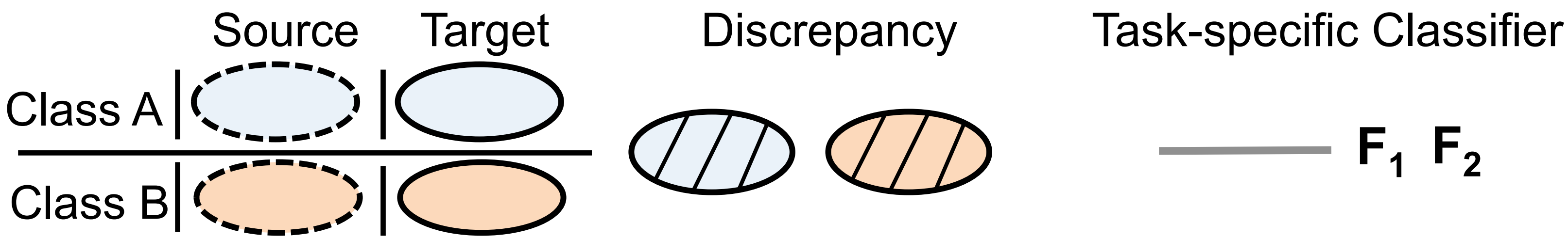


Adversarial Training Step 1: Increase Discrepancy

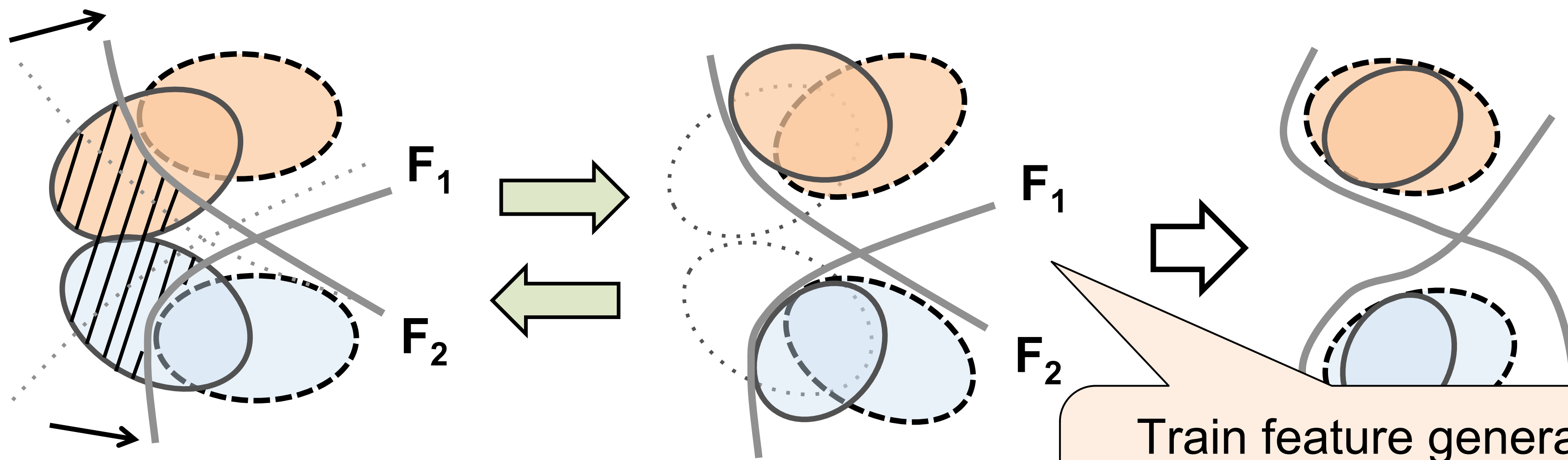




Adversarial Training Step 2: Reduce Discrepancy

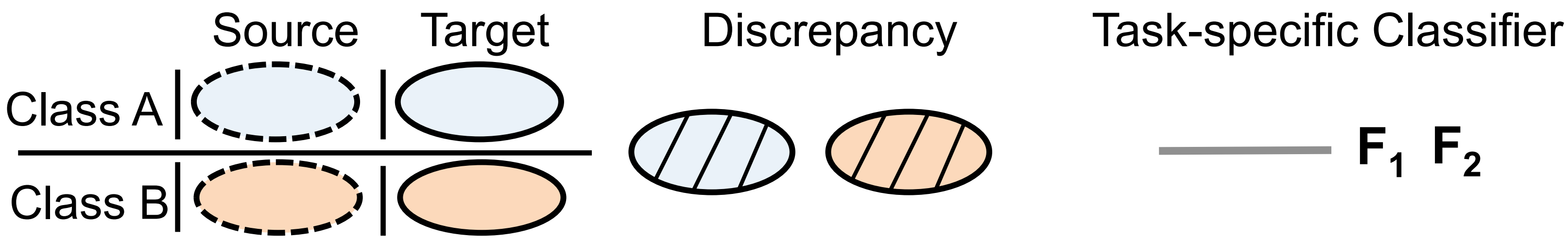


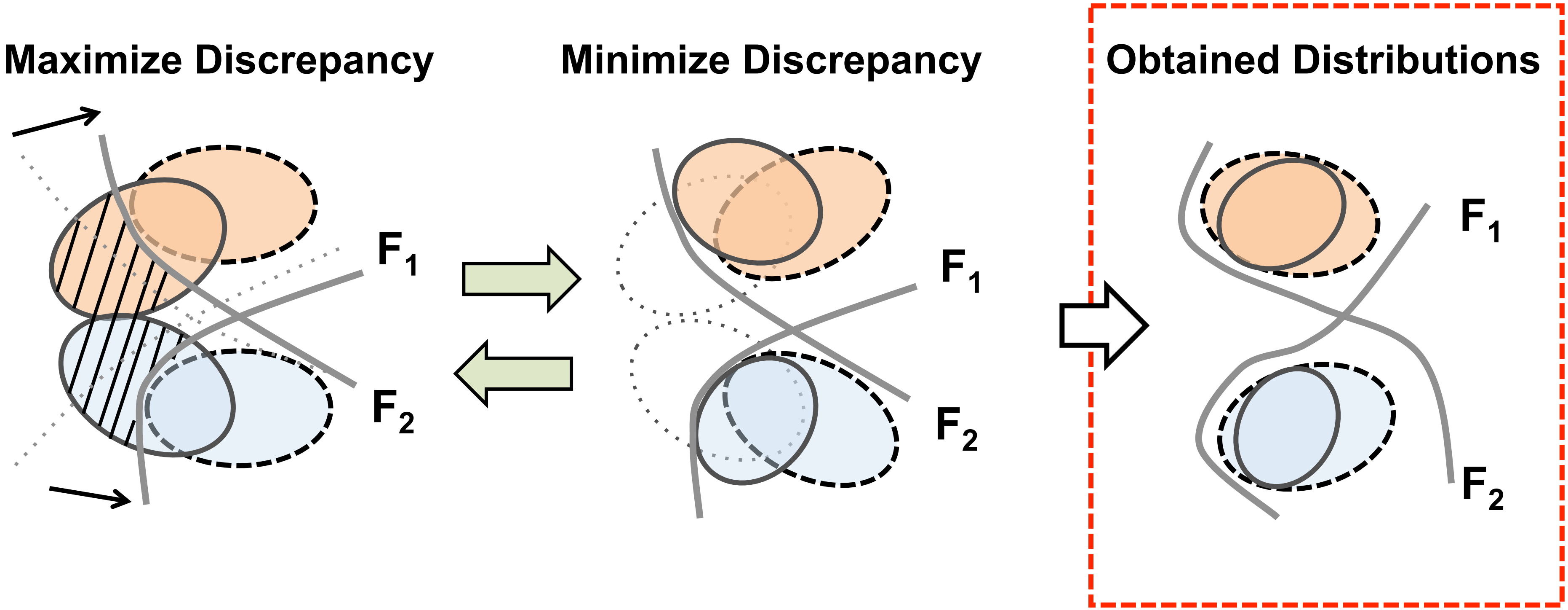
Maximize Discrepancy | **Minimize Discrepancy** | **Obtained Distributions**



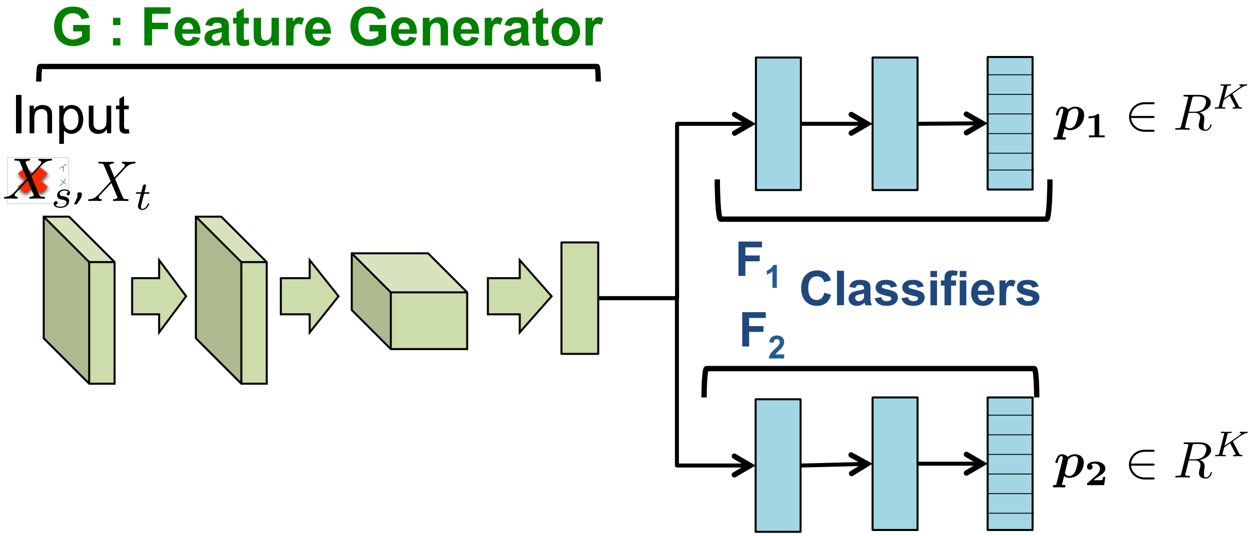
Train feature generator to minimize discrepancy. (Fix F_1, F_2)

Adversarial Training Step 2: Reduce Discrepancy





Entire Training Procedure



Loss Function

Source

L1: $L_{crossentropy}(\boldsymbol{p_1}, y_s)$

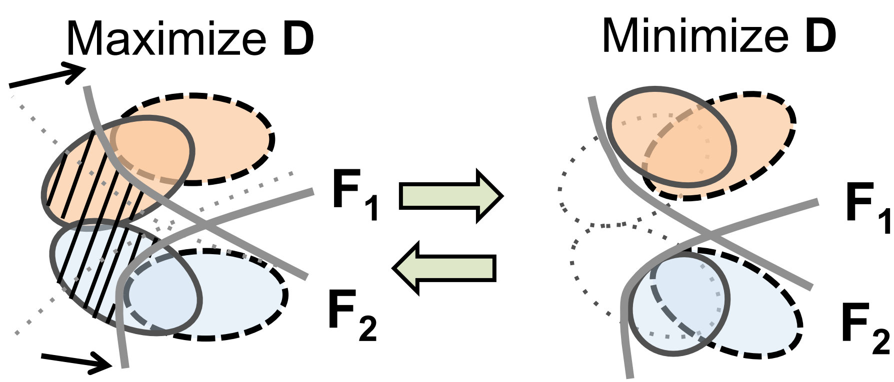
L2: $L_{crossentropy}(\boldsymbol{p_2}, y_s)$

Target

D: $\frac{1}{K}|\boldsymbol{p_1} - \boldsymbol{p_2}|_1$

Training per one mini-batch (sample images from both domains)

1, Fix G and update F1, F2 to decrease **L1+L2-D (maximize the discrepancy)**
2, Update G,F1,F2 to decrease **L1+L2 (minimize error on source)**
3, Fix F1,F2 and update G to decrease **D (minimize the discrepancy)**



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 an ideal hypothesis
$\lambda = \min[R_{\mathcal{S}}(h) + R_{\mathcal{T}}(h)]$

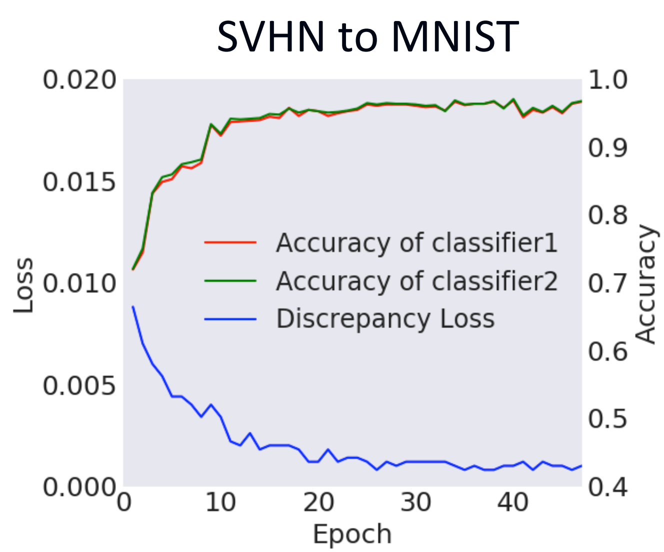
$$\sup_{(h,h') \in \mathcal{H}^2} \left| \mathbb{E}_{\boldsymbol{x} \sim \mathcal{S}} I[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x} \sim \mathcal{T}} I[h(\boldsymbol{x}) \neq h'(\boldsymbol{x})] \right|$$

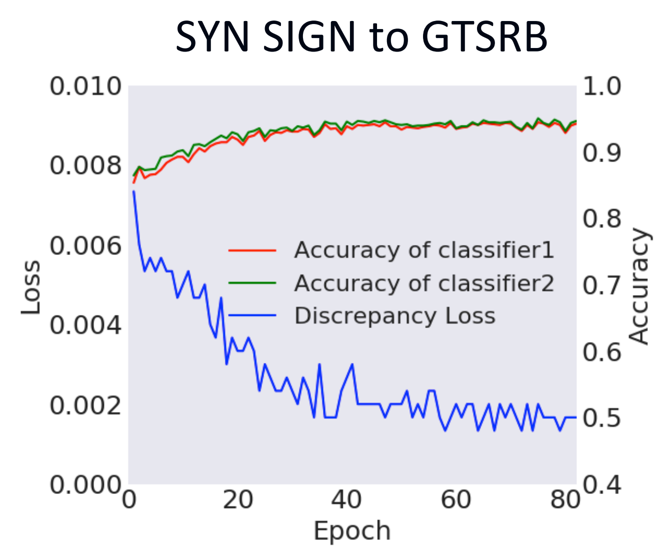
$$\sup_{F_1, F_2} \mathbb{E}_{\boldsymbol{x} \sim \mathcal{T}} I[F_1 \circ G(\boldsymbol{x}) \neq F_2 \circ G(\boldsymbol{x})]$$

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

Experiments 1: Image Classification

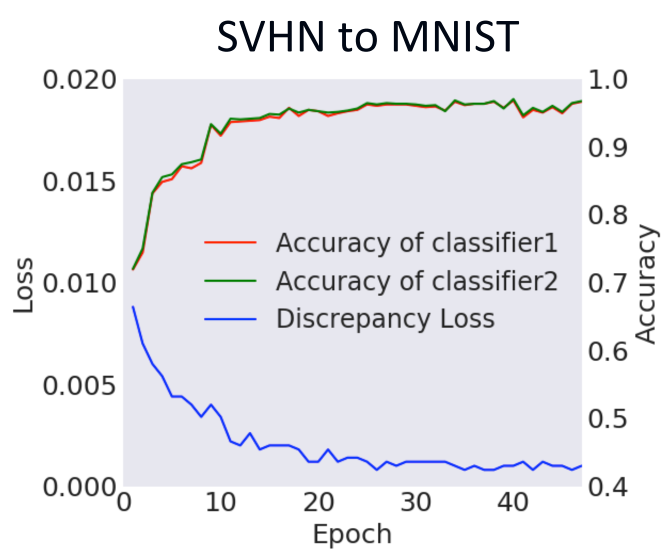
METHOD	SVHN to MNIST	SYNSIG to GTSRB	MNIST to USPS	MNIST* to USPS*	USPS to MNIST
Source Only	67.1	85.1	76.7	79.4	63.4
MMD [Long et al., ICML 2015]	71.1	91.1	-	81.1	-
DANN [Ganin et al., ICML 2015]	71.1	88.7	77.1±1.8	85.1	73.0±0.2
DSN [Bousmalis et al., NIPS 2016]	82.7	93.1	91.3	-	-
ADDA [Tzeng et al., CVPR 2017]	76.0±1.8	-	89.4±0.2	-	90.1±0.8
Ours	**96.2**±0.4	**94.4**±0.3	**94.2**±0.7	**96.5**±0.3	**94.1**±0.3

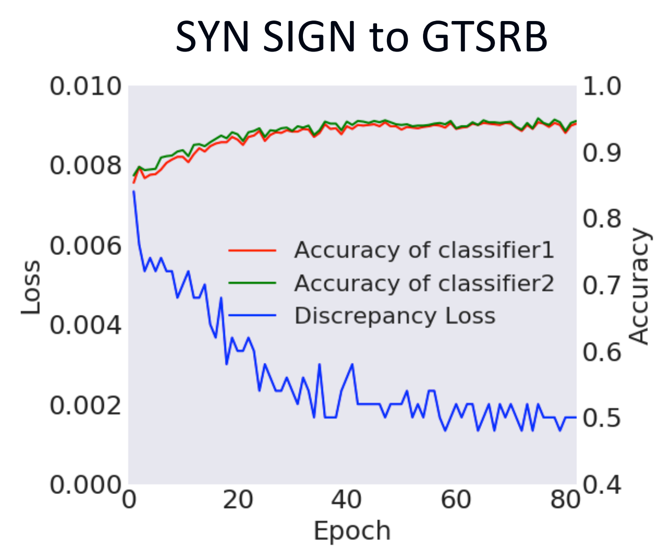




Experiments 1: Image Classification

METHOD	SVHN to MNIST	SYNSIG to GTSRB	MNIST to USPS	MNIST* to USPS*	USPS to MNIST
Source Only	67.1	85.1	76.7	79.4	63.4
MMD [Long et al., ICML 2015]	71.1	91.1	-	81.1	-
DANN [Ganin et al., ICML 2015]	71.1	88.7	77.1±1.8	85.1	73.0±0.2
DSN [Bousmalis et al., NIPS 2016]	82.7	93.1	91.3	-	-
ADDA [Tzeng et al., CVPR 2017]	76.0±1.8	-	89.4±0.2	-	90.1±0.8
Ours	**96.2**±0.4	**94.4**±0.3	**94.2**±0.7	**96.5**±0.3	**94.1**±0.3

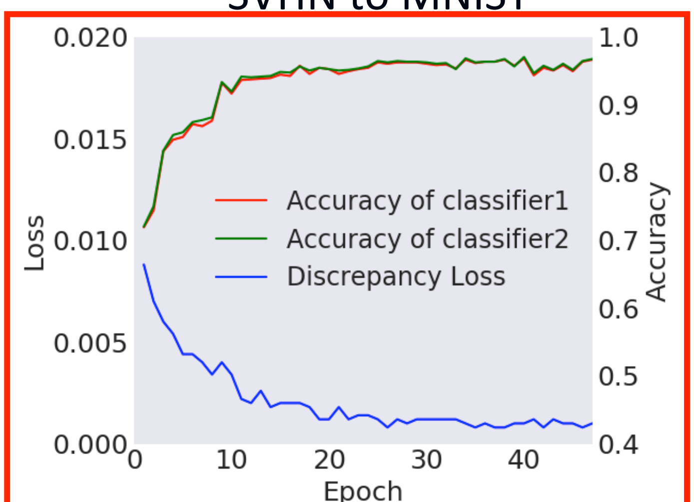




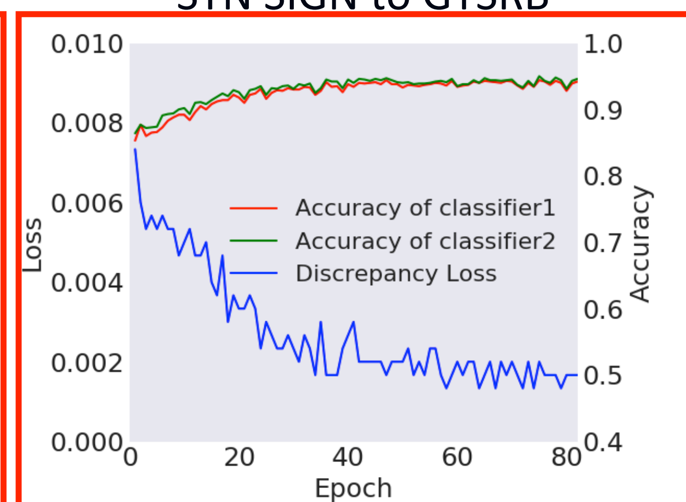
Experiments 1: Image Classification

METHOD	SVHN to MNIST	SYNSIG to GTSRB	MNIST to USPS	MNIST* to USPS*	USPS to MNIST
Source Only	67.1	85.1	76.7	79.4	63.4
MMD [Long et al., ICML 2015]	71.1	91.1	-	81.1	-
DANN [Ganin et al., ICML 2015]	71.1	88.7	77.1±1.8	85.1	73.0±0.2
DSN [Bousmalis et al., NIPS 2016]	82.7	93.1	91.3	-	-
ADDA [Tzeng et al., CVPR 2017]	76.0±1.8	-	89.4±0.2	-	90.1±0.8
Ours	**96.2**±0.4	**94.4**±0.3	**94.2**±0.7	**96.5**±0.3	**94.1**±0.3

SVHN to MNIST

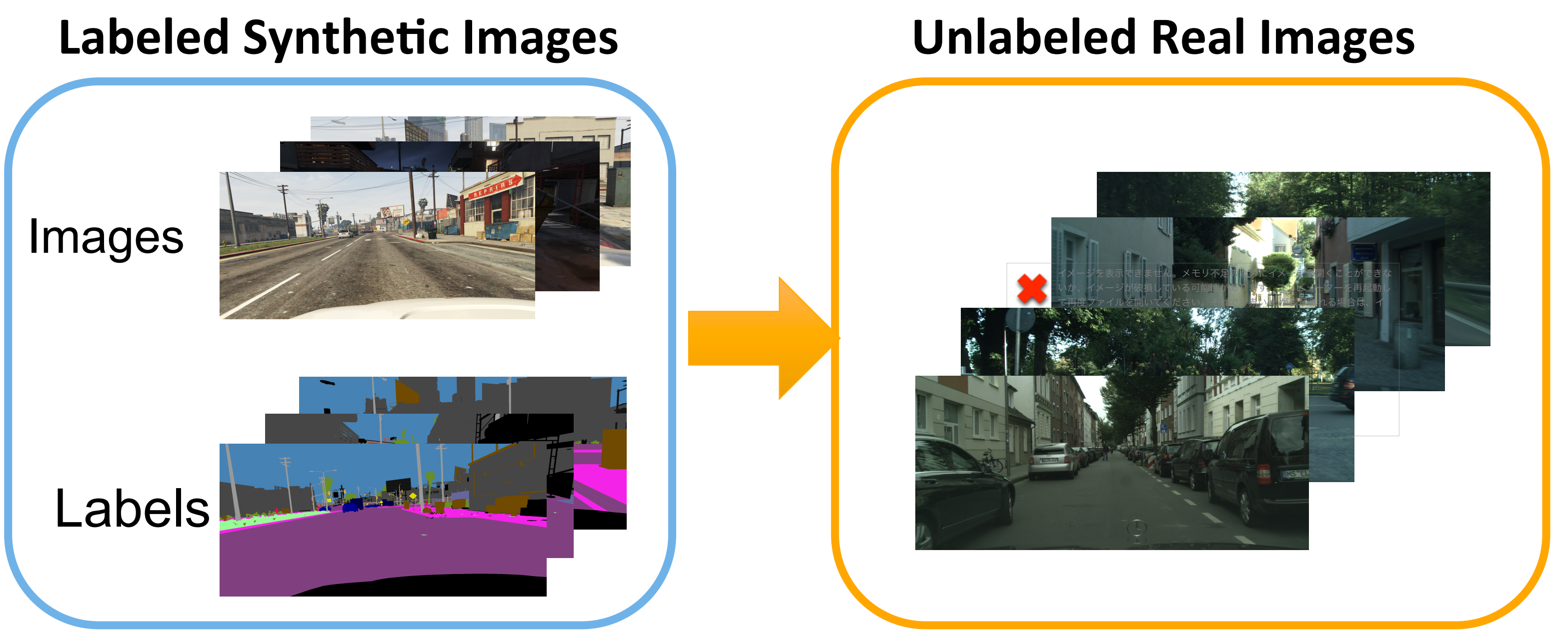


SYN SIGN to GTSRB



Experiments 2: Semantic Segmentation

- Simulated Image (GTA5) to Real Image (CityScape)
- Finetuning of pre-trained VGG, Dilated Residual Network [Yu et al., 2017]
- Discrepancy is calculated in a pixel-wise way.



Experiments 2: Semantic Segmentation

- Simulated Image (GTA5) to Real Image (CityScape)
- Finetuning of pre-trained VGG, Dilated Residual Network [Yu et al., 2017]
- Discrepancy is calculated in a pixel-wise way.

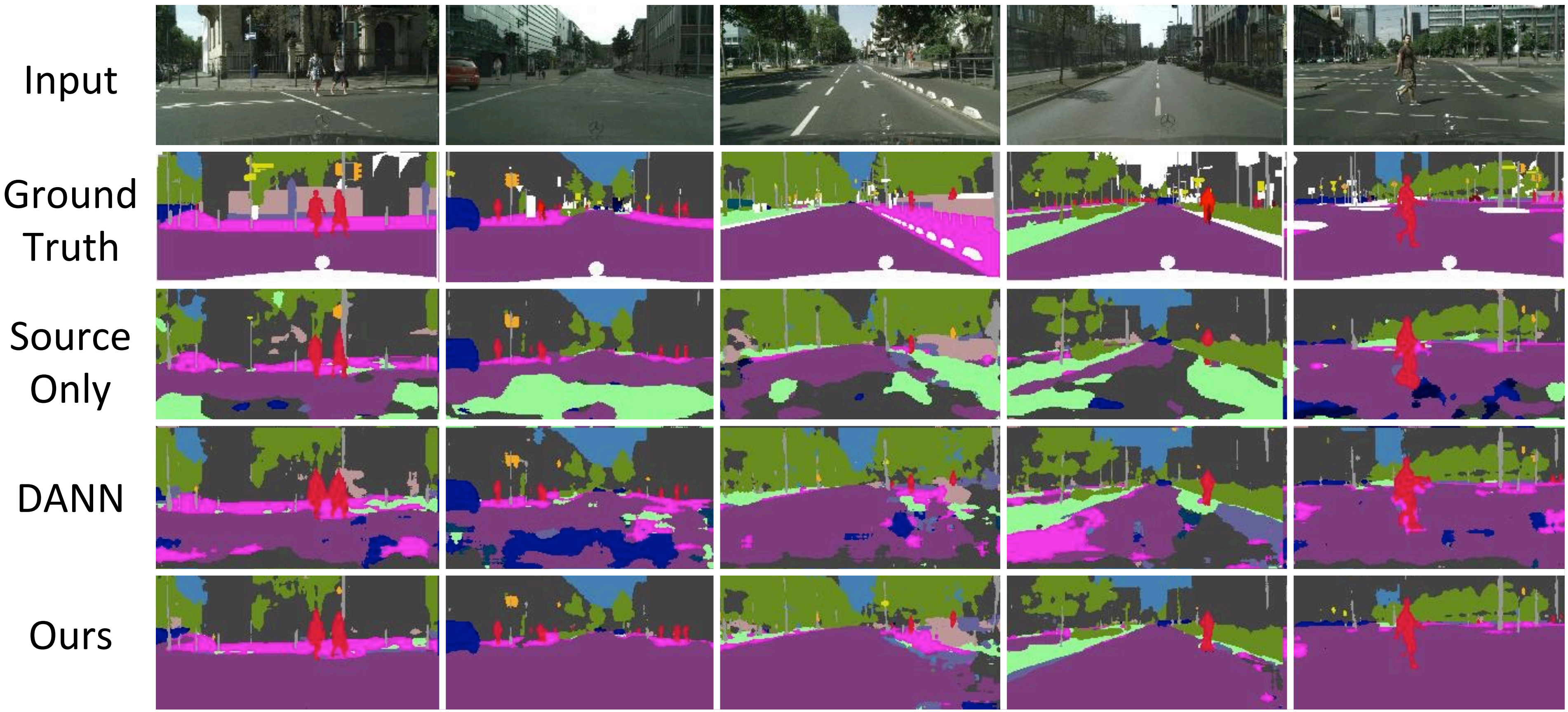
Network	Method	mIoU
VGG-16	Source Only	24.9
	FCN Wld [Hoffman et al., Arxiv 2017]	27.1
	CDA (I) [Zhang et al., ICCV 2017]	23.1
	Ours	28.8
DRN-105	Source Only	22.2
	DANN [Ganin et al., ICML 2015]	32.8
	Ours	**39.7**

Experiments 2: Semantic Segmentation

- Simulated Image (GTA5) to Real Image (CityScape)
- Finetuning of pre-trained VGG, Dilated Residual Network [Yu et al., 2017]
- Discrepancy is calculated in a pixel-wise way.

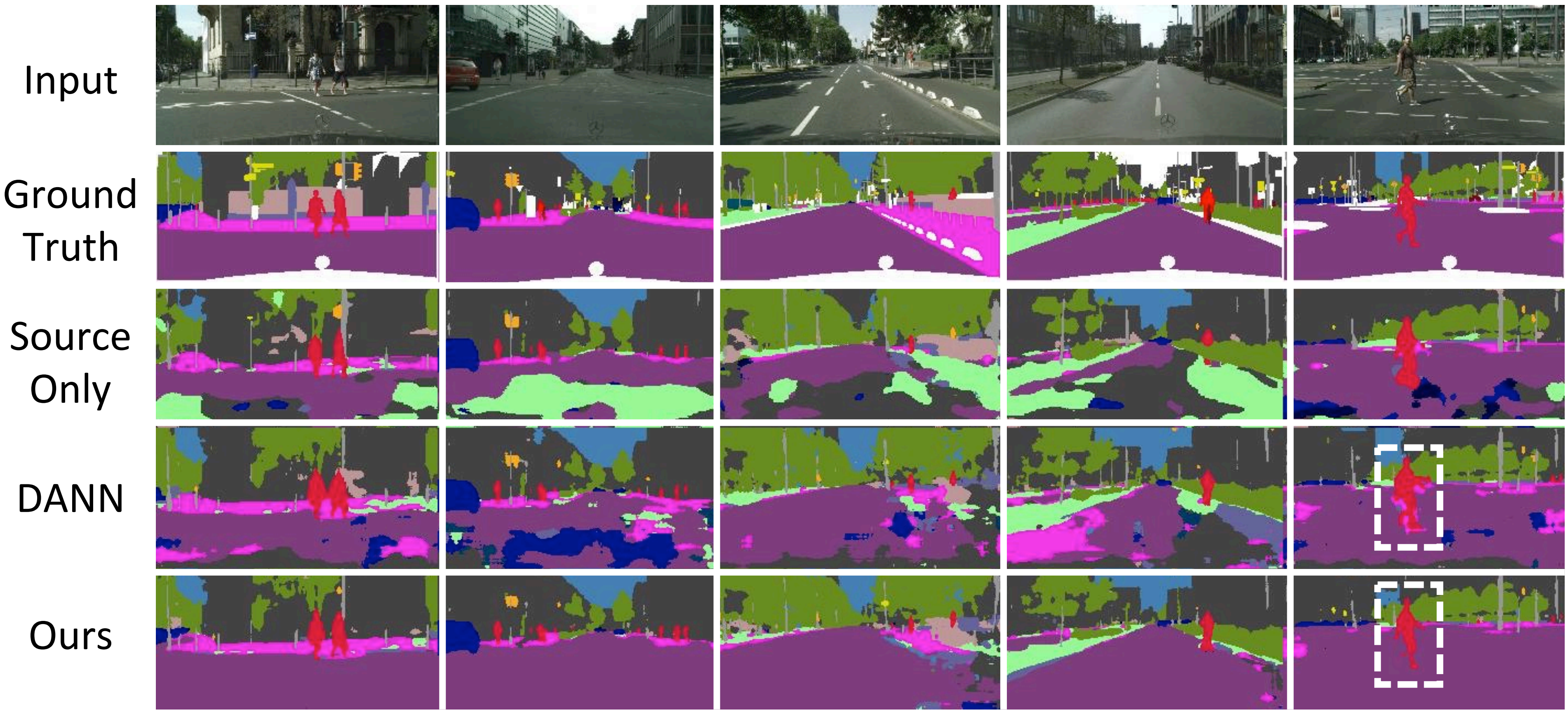
Network	Method	mIoU
VGG-16	Source Only	24.9
	FCN Wld [Hoffman et al., Arxiv 2017]	27.1
	CDA (I) [Zhang et al., ICCV 2017]	23.1
	Ours	28.8
DRN-105	Source Only	22.2
	DANN [Ganin et al., ICML 2015]	32.8
	Ours	**39.7**

Qualitative Result



- ➢ Boundary between other classes is made clear.
- ➢ The effectiveness is obvious for segmentation of road.

Qualitative Result



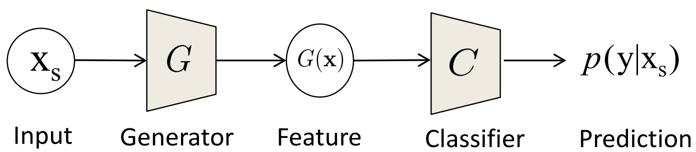
- Boundary between other classes is made clear.
- The effectiveness is obvious for segmentation of road.

Do we need two networks?

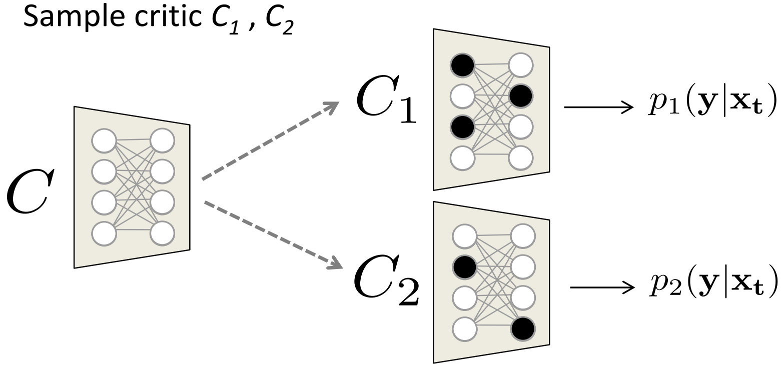
- Adversarial Dropout Regularization [Saito et al., ICLR 2018]
  - Sample two classifiers from one network using dropout

Critic sampling using dropout

Train *G, C* on source inputs using classification loss

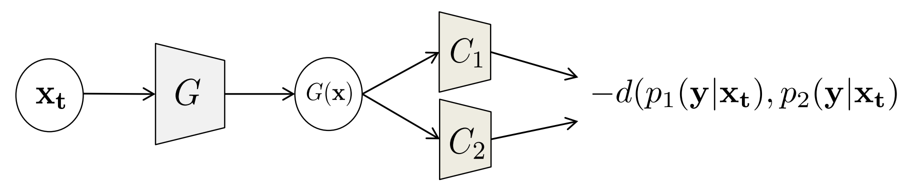


Sample critic C_1 , C_2

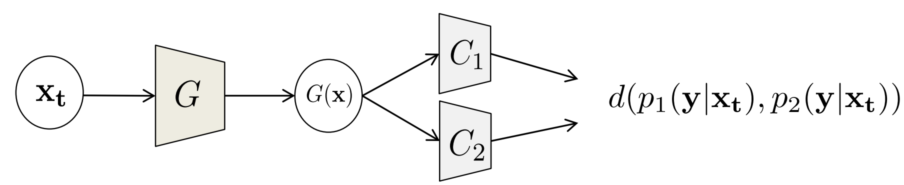


Adversarial learning

Update *C* to ***maximize*** sensitivity on target inputs (Fix *G*)



Update *G* to ***minimize*** sensitivity on target inputs (Fix *C*)



THANK YOU FOR LISTENING!!
#POSTER C19