Model Uncertainty for Unsupervised Domain Adaptation

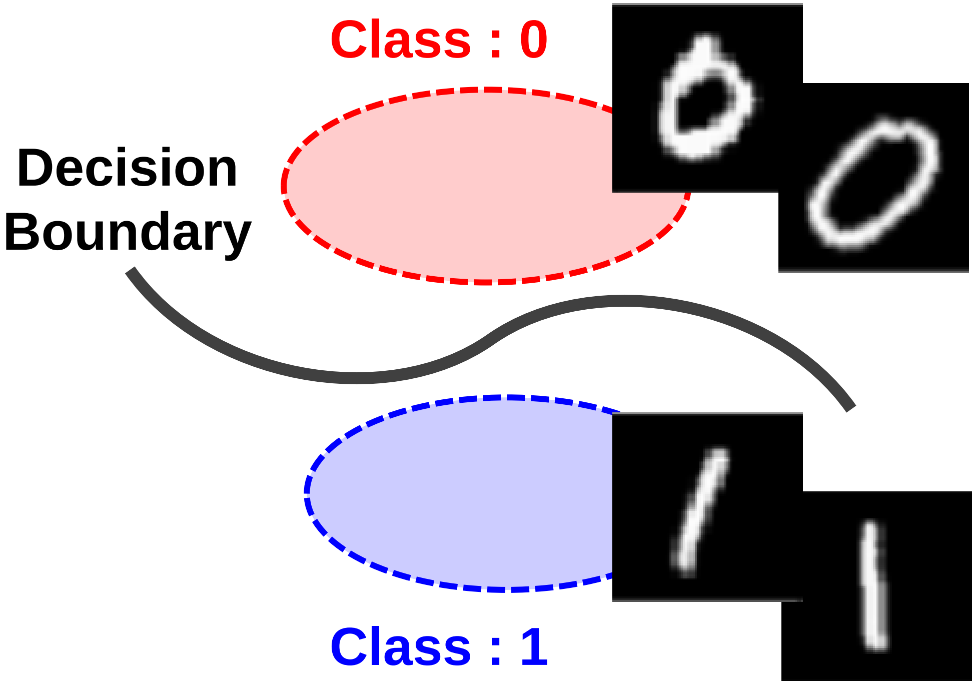
JoonHo Lee[1,2]

Gyemin Lee[1]

[1]Seoul National University of Science and Technology
[2]Samsung SDS Artificial Intelligence Research Center

Motivation

- Deep neural networks (DNNs) have shown great successes for the data that have similar distributions to the training data (*source*).

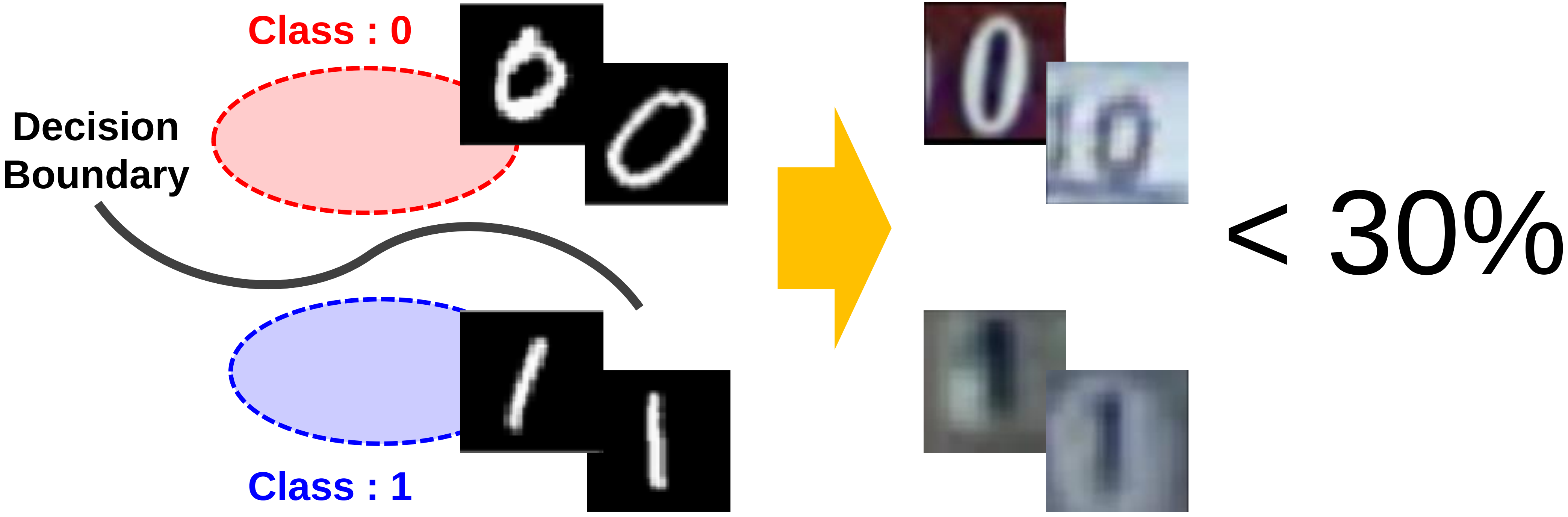


MNIST

> 99%

Motivation

- But the DNN model sufficiently trained on a source domain often doesn't work well on some *target* domains.

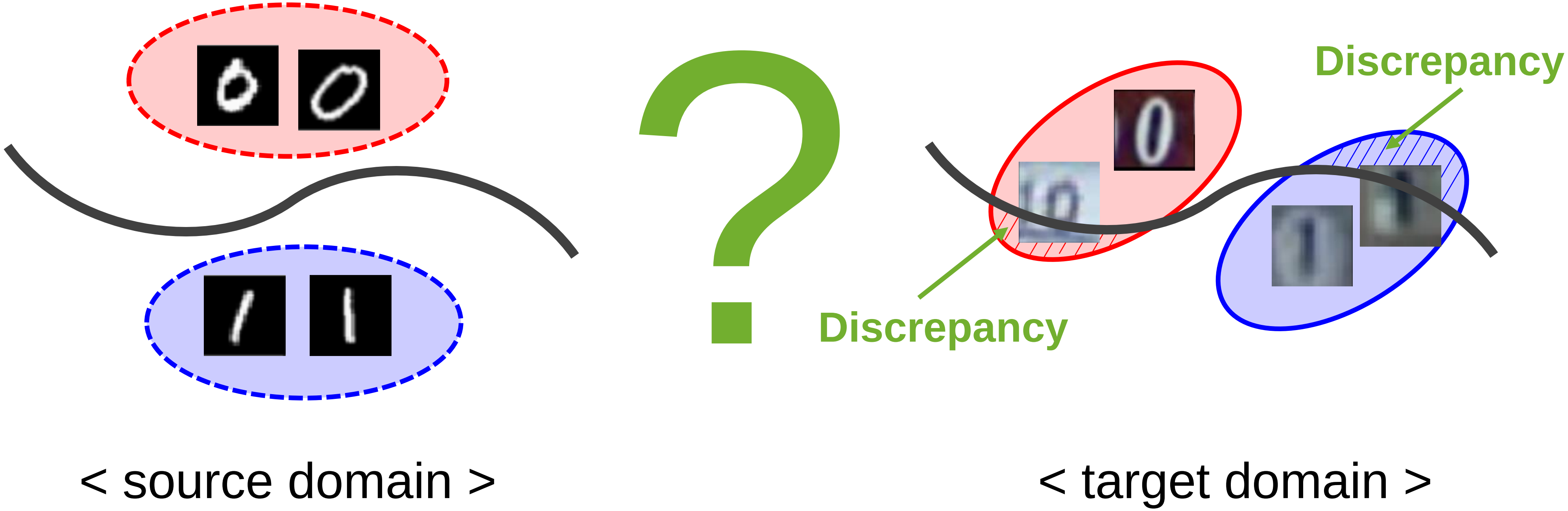


MNIST

Street View House Numbers (SVHN)

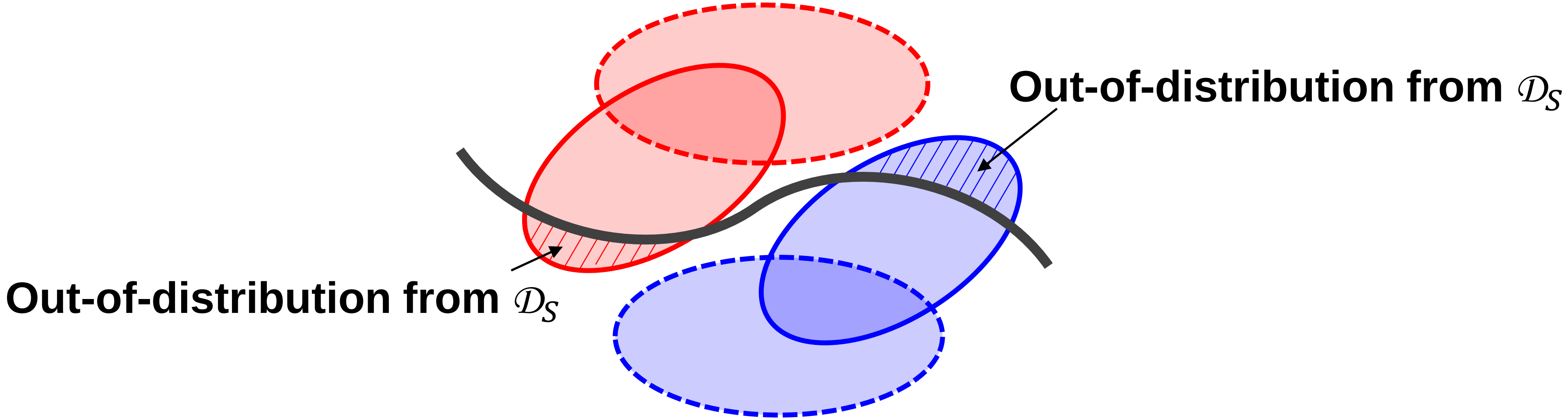
Motivation

- Domain Adaptation (DA) addresses this *domain shift* by adapting a model trained on a source domain to a target domain.



Key Idea

- We presume that domain shift occurs when target samples are out of the source domain distribution ($\mathcal{D}_S$).

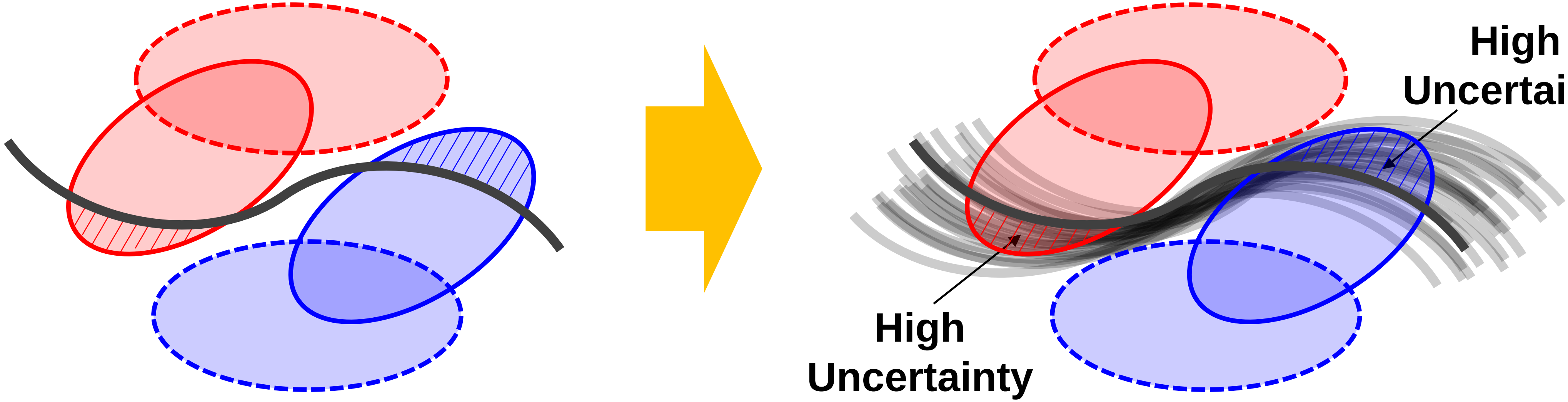


Out-of-distribution from $\mathcal{D}_S$

Out-of-distribution from $\mathcal{D}_S$

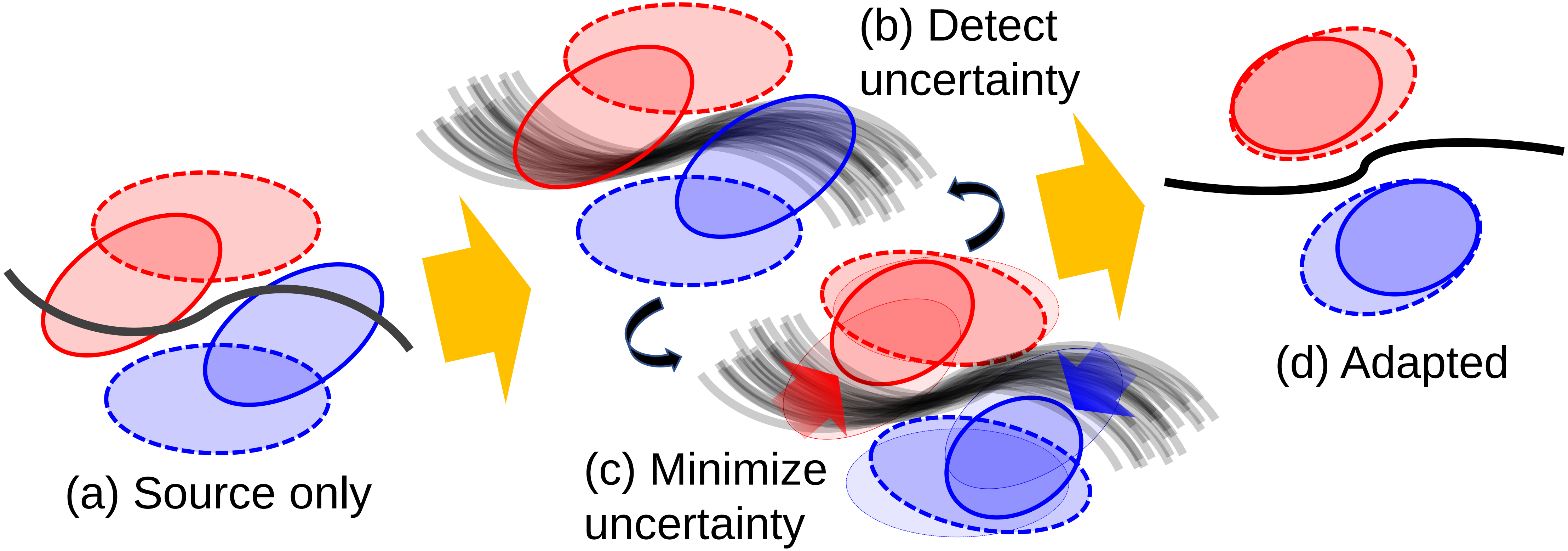
Key Idea

- The predictive uncertainty of the model (*model uncertainty*) should be measured ‘high’ at the target samples outside $\mathcal{D}_S$.



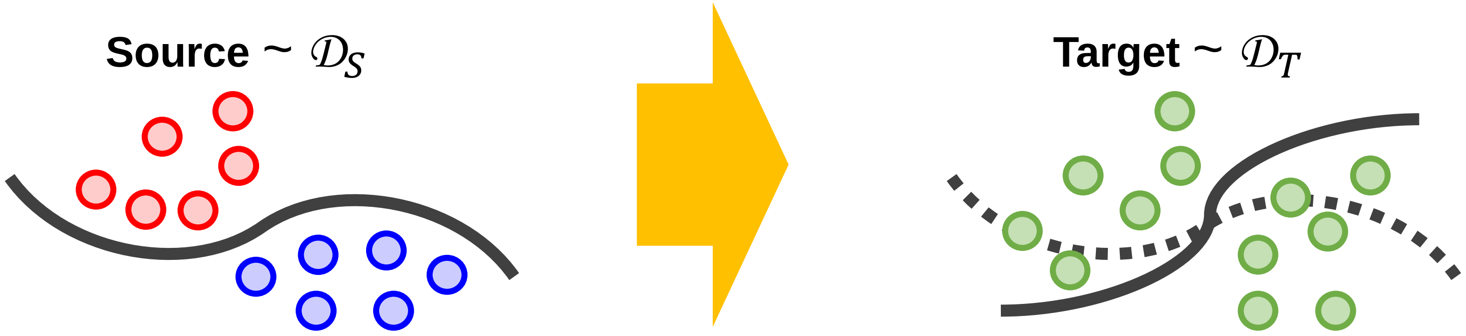
Key Idea

- If the model uncertainty can be quantified and reduced properly, we expect a DNN model learns domain-invariant feature representations.



Unsupervised Domain Adaptation

- Unsupervised Domain Adaptation (UDA) assumes:
  - ① Fully-labeled source data $(X_s, Y_s) \sim \mathcal{D}_S$
  - ② *Unlabeled* target data $X_t \sim \mathcal{D}_T$
  - ③ Both domains share a label space : $y \in \{1, \ldots, K\}$
- The goal of UDA is to build a classifier that correctly predicts y_t of a new target sample x_t drawn from $\mathcal{D}_T$.



Theoretical Insight : Ben-David et al.

- Ben-David et al. [1] has proposed a theory that upperbounds the expected target error $\epsilon_T(h)$ as follows:

$$\boxed{\epsilon_T(h) \leq \epsilon_S(h) + \frac{1}{2}\, d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) + \lambda, \qquad \text{for all } h \in \mathcal{H}}$$

$\epsilon_S(h)$: the expected source error

$d_{\mathcal{H}\triangle\mathcal{H}}$: divergence in the symmetric difference hypothesis space

λ : the combined error of the ideal joint hypothesis (constant)

$\mathcal{H}$: a hypothesis space

[1] S. Ben-David et al., "A theory of learning from different domains", *Machine Learning*, vol. 79(1-2), pp.151-175, 2010

Theoretical Insight : Ben-David et al.

- Ben-David et al. [1] has proposed a theory that upperbounds the expected target error $\epsilon_T(h)$ as follows:

$$\epsilon_T(h) \le \epsilon_S(h) + \frac{1}{2}\boxed{d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T)} + \lambda, \qquad \text{for all } h \in \mathcal{H}$$

$\epsilon_S(h)$: the expected source error

$d_{\mathcal{H}\triangle\mathcal{H}}$: divergence in the symmetric difference hypothesis space

λ : the combined error of the ideal joint hypothesis (constant)

$\mathcal{H}$: a hypothesis space

[1] S. Ben-David et al., “A theory of learning from different domains”, *Machine Learning*, vol. 79(1-2), pp.151-175, 2010

Derivation of Our Method

- From [1], $d_{\mathcal{H}\triangle\mathcal{H}}$ is defined by

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \coloneqq 2 \sup_{h,h' \in \mathcal{H}} \left| \mathbb{E}_{x\sim\mathcal{D}_S}\left[\mathbf{1}\left(h(x) \neq h'(x)\right)\right] - \mathbb{E}_{x\sim\mathcal{D}_T}\left[\mathbf{1}\left(h(x) \neq h'(x)\right)\right] \right|$$

[1] S. Ben-David et al., "A theory of learning from different domains", *Machine Learning*, vol. 79(1-2), pp.151-175, 2010

Derivation of Our Method

- From [1], $d_{\mathcal{H}\triangle\mathcal{H}}$ is defined by

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) := 2 \sup_{h,h' \in \mathcal{H}} \left| \underbrace{\mathbb{E}_{x\sim\mathcal{D}_S}\left[\mathbf{1}\left(h(x) \neq h'(x)\right)\right]}_{①} - \mathbb{E}_{x\sim\mathcal{D}_T}\left[\mathbf{1}\left(h(x) \neq h'(x)\right)\right] \right|$$

- We simplify $d_{\mathcal{H}\triangle\mathcal{H}}$ to incorporate our idea:

① If h and h' can correctly classify source samples, they will agree on source samples. This enables us to neglect the term

$$\mathbb{E}_{x\sim\mathcal{D}_S}\left[\mathbf{1}\left(h(x) \neq h'(x)\right)\right].$$

→ small (≈ 0)

[1] S. Ben-David et al., “A theory of learning from different domains”, *Machine Learning*, vol. 79(1-2), pp.151-175, 2010

Derivation of Our Method

- From [1], $d_{\mathcal{H}\triangle\mathcal{H}}$ is defined by

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T) \coloneqq 2 \sup_{h,h'\in\mathcal{H}} \left| \underbrace{\mathbb{E}_{x\sim\mathcal{D}_S}\left[\mathbb{1}\left(h(x) \neq h'(x)\right)\right]}_{\textcircled{1}} - \underbrace{\mathbb{E}_{x\sim\mathcal{D}_T}\left[\mathbb{1}\left(h(x) \neq h'(x)\right)\right]}_{\textcircled{2}} \right|$$

- We simplify $d_{\mathcal{H}\triangle\mathcal{H}}$ to incorporate our idea:

① If h and h' can correctly classify source samples, they will agree on source samples. This enables us to neglect the term ①.

② For binary classification $h(x) \in \{0,1\}$,

$$\mathbb{1}\left(h(x) \neq h'(x)\right) = \left(h(x) - h'(x)\right)^2.$$

[1] S. Ben-David et al., “A theory of learning from different domains”, *Machine Learning*, vol. 79(1-2), pp.151-175, 2010

Derivation of Our Method

- Hence,

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \approx 2 \sup_{h,h'\in\mathcal{H}} \mathbb{E}_{x\sim\mathcal{D}_T}\left[\left(h(x) - h'(x)\right)^2\right].$$

- We argue, by narrowing our attention to the set of h that minimizes ϵ_S, we can achieve the same goal as the supremum of expectation:

$$\sup_{h,h'\in\mathcal{H}} \mathbb{E}_{x\sim\mathcal{D}_T}\left[\left(h(x) - h'(x)\right)^2\right].$$

Derivation of Our Method

- Hence,

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T) \approx 2 \sup_{h,h'\in\mathcal{H}} \mathbb{E}_{x\sim\mathcal{D}_T}\left[\left(h(x)-h'(x)\right)^2\right].$$

- We can define $\mathcal{D}_{\mathcal{H}} \coloneqq P(h|X_s,Y_s)$ and replace the supremum with the expectation to obtain

$$\begin{aligned} d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S,\mathcal{D}_T) &\approx 2\,\mathbb{E}_{h,h'\sim\mathcal{D}_{\mathcal{H}}}\mathbb{E}_{x\sim\mathcal{D}_T}\left[\left(h(x)-h'(x)\right)^2\right] \\ &= 4\,\mathbb{E}_{x\sim\mathcal{D}_T}\mathbb{E}_{h\sim\mathcal{D}_{\mathcal{H}}}\left[\left(h(x)-\mathbb{E}_{h\sim\mathcal{D}_{\mathcal{H}}}[h(x)]\right)^2\right]. \end{aligned}$$

Derivation of Our Method

- Hence,

$$d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) \approx 2 \sup_{h,h'\in\mathcal{H}} \mathbb{E}_{x\sim\mathcal{D}_T}\left[\left(h(x) - h'(x)\right)^2\right].$$

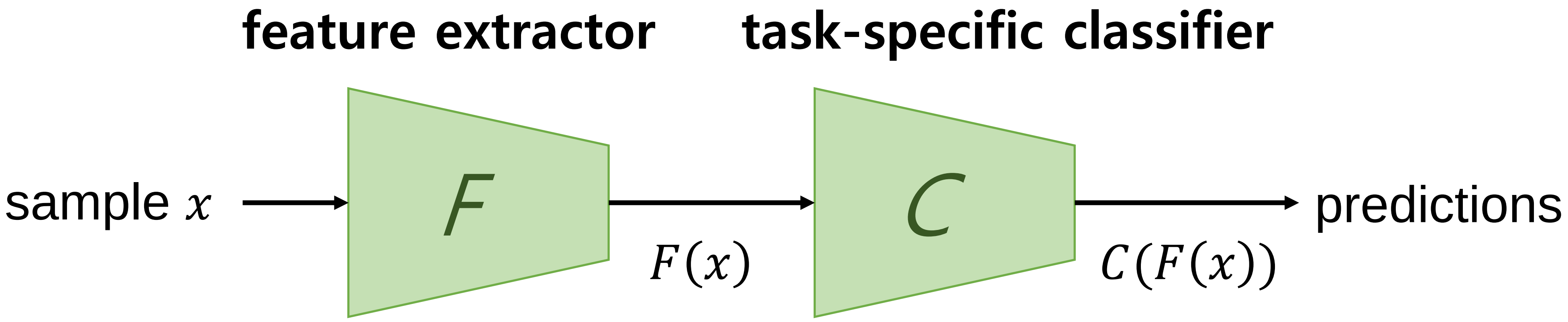
- We can define $\mathcal{D}_\mathcal{H} := P(h|X_s, Y_s)$ and replace the supremum with the expectation to obtain

$$\begin{aligned} d_{\mathcal{H}\triangle\mathcal{H}}(\mathcal{D}_S, \mathcal{D}_T) &\approx 2\, \mathbb{E}_{h,h'\sim\mathcal{D}_\mathcal{H}} \mathbb{E}_{x\sim\mathcal{D}_T}\left[\left(h(x) - h'(x)\right)^2\right] \\ &= 4 \boxed{\mathbb{E}_{x\sim\mathcal{D}_T} \mathbb{E}_{h\sim\mathcal{D}_\mathcal{H}}\left[\left(h(x) - \mathbb{E}_{h\sim\mathcal{D}_\mathcal{H}}[h(x)]\right)^2\right]}. \end{aligned}$$

"the predictive variance of a hypothesis (model uncertainty)"

Model Preliminaries

- We formulate the UDA problem under the shared latent space framework.



feature extractor

task-specific classifier

sample x

F

$F(x)$

C

$C(F(x))$

predictions

Proposed Method (MUDA)

- From the previous derivation, the problem is

  minimizing $d_{\mathcal{H}\Delta\mathcal{H}}$ → minimizing the model uncertainty on target samples.

- Now our objectives are

  ① to find *F* that minimizes the model uncertainty :

$$\min_{F} \mathbb{E}_{x\sim\mathcal{D}_T}\mathbb{E}_{h\sim\mathcal{D}_{\mathcal{H}}}\left[\left(h(x)-\mathbb{E}_{h\sim\mathcal{D}_{\mathcal{H}}}[h(x)]\right)^2\right]$$

  ② to keep minimizing the source error ϵ_S :

$$\min_{F,C} \epsilon_S(h)$$

Proposed Method (MUDA)

① to find *F* that minimizes the model uncertainty :

$$\min_{F} \mathbb{E}_{x\sim\mathcal{D}_T}\mathbb{E}_{h\sim\mathcal{D}_{\mathcal{H}}}\left[\left(h(x)-\mathbb{E}_{h\sim\mathcal{D}_{\mathcal{H}}}[h(x)]\right)^2\right]$$

We perform *M* stochastic forward passes via MC dropout [2] to obtain $\{\hat{y}_1, \ldots, \hat{y}_M\}$ that lead to the estimation :

$$\hat{\sigma}_{MC}^2 = \text{diag}\left(\frac{1}{M}\sum_{m=1}^{M}\hat{y}_m\,\hat{y}_m^{\mathrm{T}} - \bar{y}_{MC}\bar{y}_{MC}^{\mathrm{T}}\right)$$

$$\text{where } \bar{y}_{MC} = \frac{1}{M}\sum_{m=1}^{M}\hat{y}_m$$

➔ Model uncertainty loss $\mathcal{L}_{div}(\mathcal{D}_T) \coloneqq \|\hat{\sigma}_{MC}\|$

[2] Y. Gal et al., “Dropout as a Bayesian approximation: Representing model uncertainty in deep learning”, *In ICML 2016*

Proposed Method (MUDA)

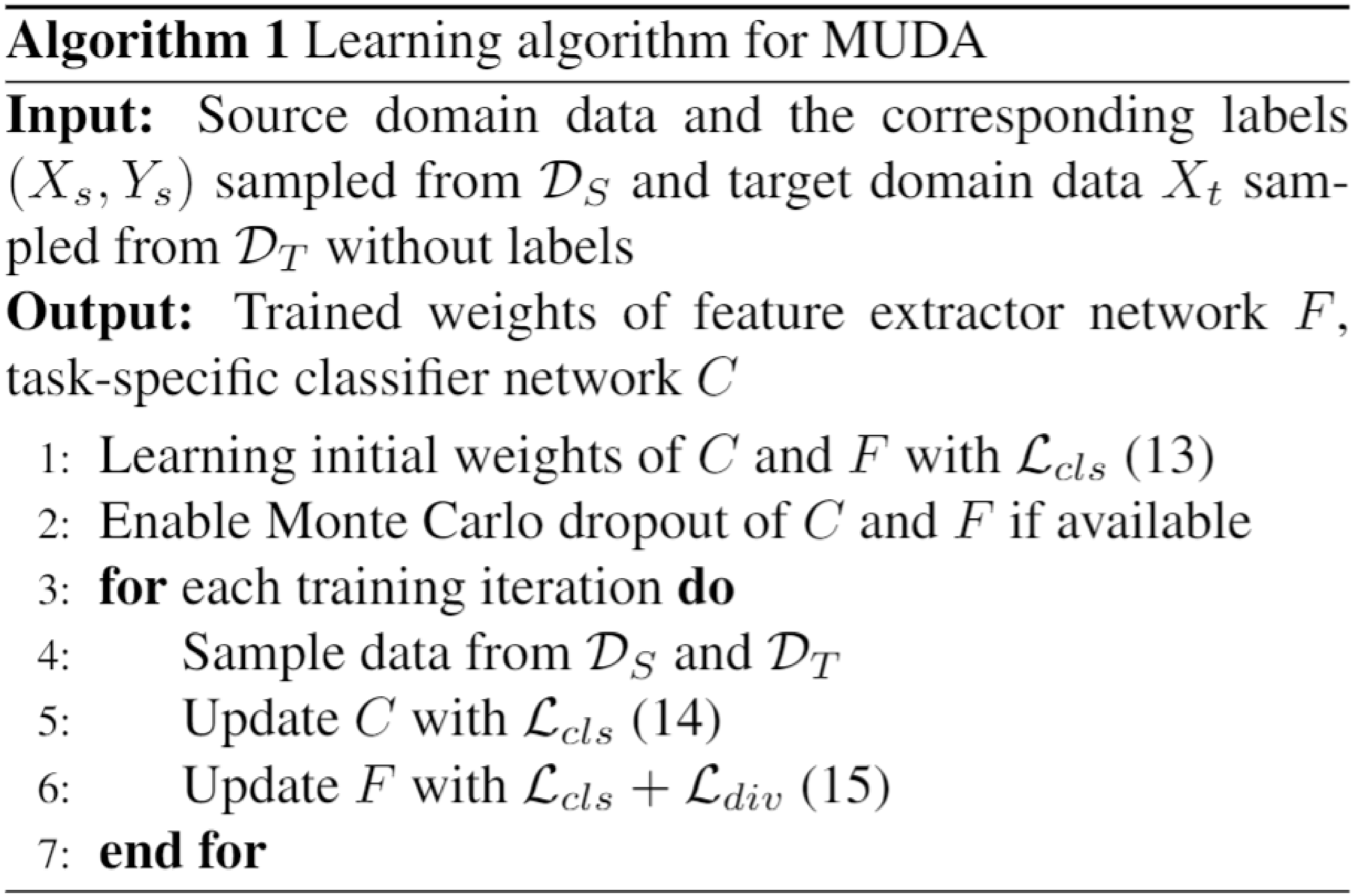
② to keep minimizing the source error ϵ_S :

$$\min_{F,C} \epsilon_S(h)$$

We use cross entropy loss for classification.

➔ Classification loss $\mathcal{L}_{cls}(\mathcal{D}_S) \coloneqq -\mathbb{E}_{(x_s,y_s)\sim\mathcal{D}_S}[y_s^{\mathrm{T}} \log C(F(x_s))]$

Proposed Method (MUDA)



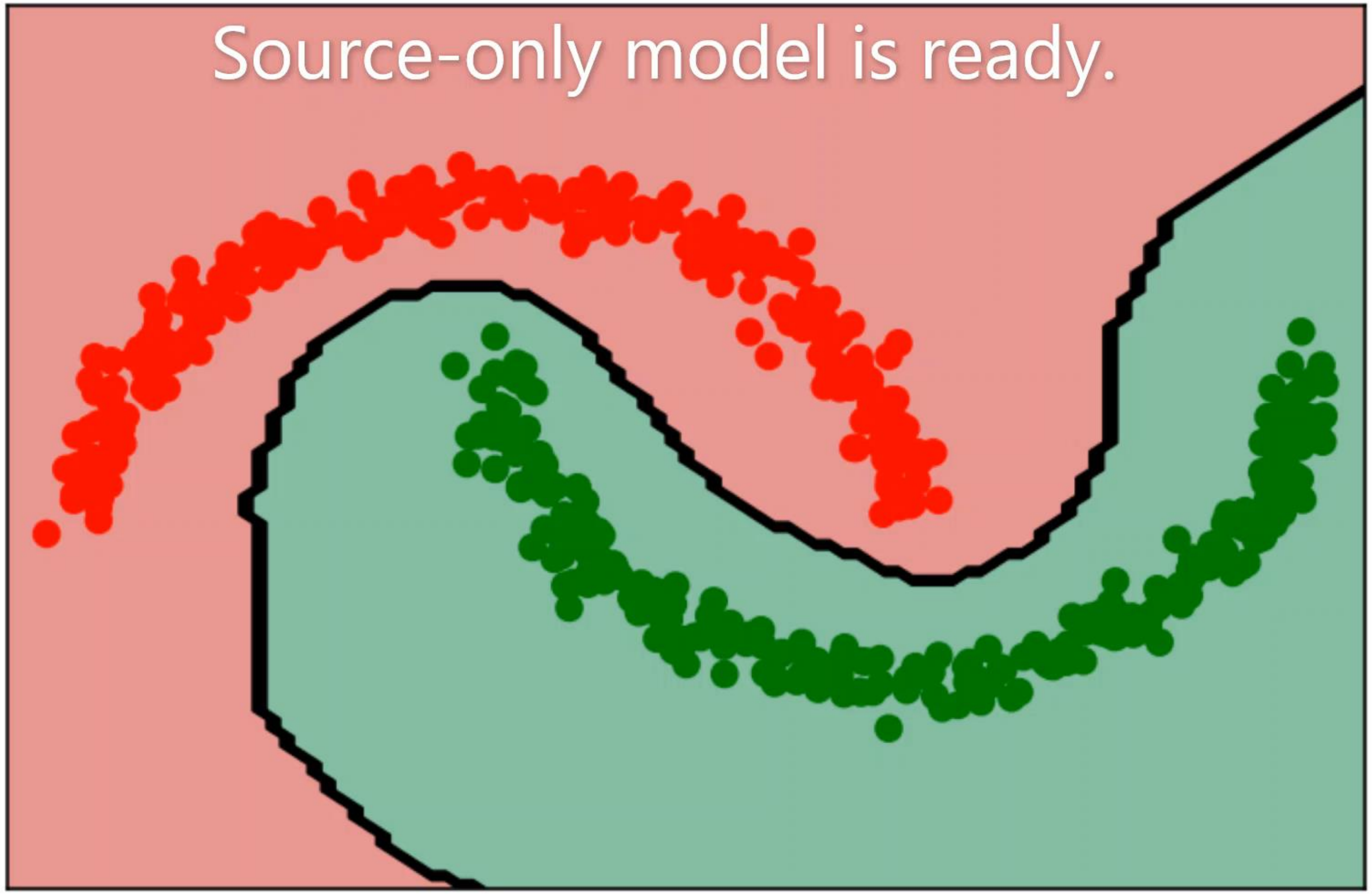
Algorithm 1 Learning algorithm for MUDA

Input: Source domain data and the corresponding labels (X_s, Y_s) sampled from $\mathcal{D}_S$ and target domain data X_t sampled from $\mathcal{D}_T$ without labels

Output: Trained weights of feature extractor network F, task-specific classifier network C

```
Learning initial weights of $C$ and $F$ with $\mathcal{L}_{cls}$ (13)
Enable Monte Carlo dropout of $C$ and $F$ if available
for each training iteration do
    Sample data from $\mathcal{D}_S$ and $\mathcal{D}_T$
    Update $C$ with $\mathcal{L}_{cls}$ (14)
    Update $F$ with $\mathcal{L}_{cls} + \mathcal{L}_{div}$ (15)
end for
```

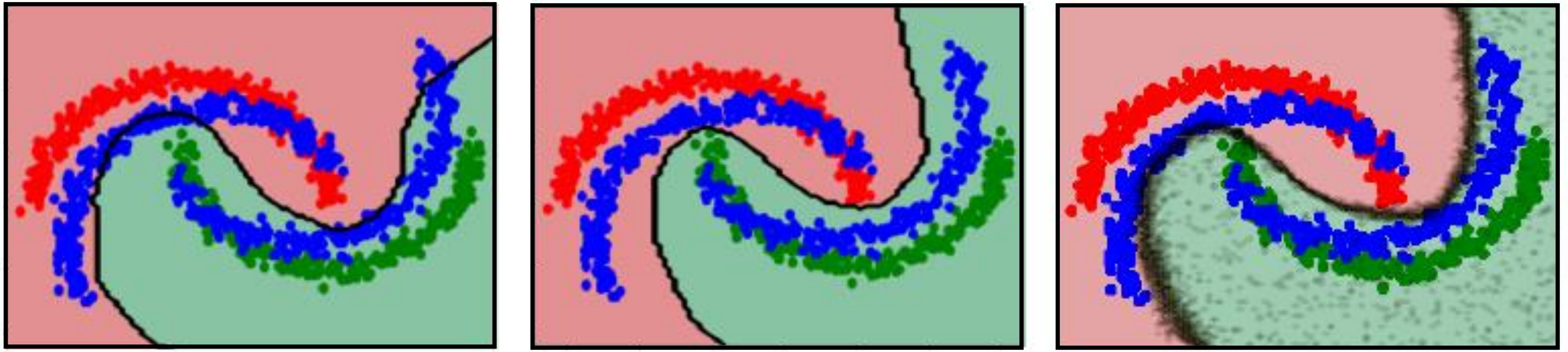
Proof of Concept



Source-only model is ready.

Proof of Concept

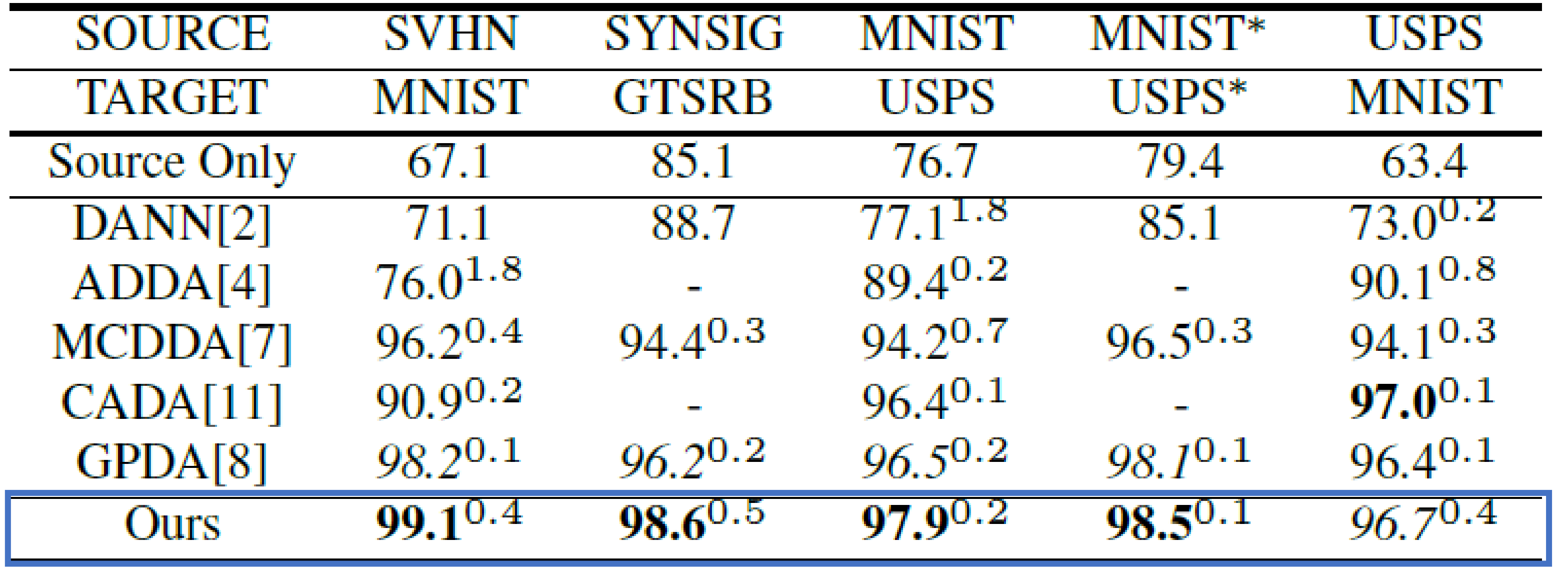
- Snapshots from the previous video clip



(a) Source-only (b) Ours (c) Ours (MC dropout)

Results: Digits & Traffic Signs datasets

Table 1. Average accuracy from 10 random experiments



SOURCE	SVHN	SYNSIG	MNIST	MNIST*	USPS
TARGET	MNIST	GTSRB	USPS	USPS*	MNIST
Source Only	67.1	85.1	76.7	79.4	63.4
DANN[2]	71.1	88.7	77.1$^{1.8}$	85.1	73.0$^{0.2}$
ADDA[4]	76.0$^{1.8}$	-	89.4$^{0.2}$	-	90.1$^{0.8}$
MCDDA[7]	96.2$^{0.4}$	94.4$^{0.3}$	94.2$^{0.7}$	96.5$^{0.3}$	94.1$^{0.3}$
CADA[11]	90.9$^{0.2}$	-	96.4$^{0.1}$	-	**97.0**$^{0.1}$
GPDA[8]	*98.2*$^{0.1}$	*96.2*$^{0.2}$	*96.5*$^{0.2}$	*98.1*$^{0.1}$	96.4$^{0.1}$
Ours	**99.1**$^{0.4}$	**98.6**$^{0.5}$	**97.9**$^{0.2}$	**98.5**$^{0.1}$	*96.7*$^{0.4}$

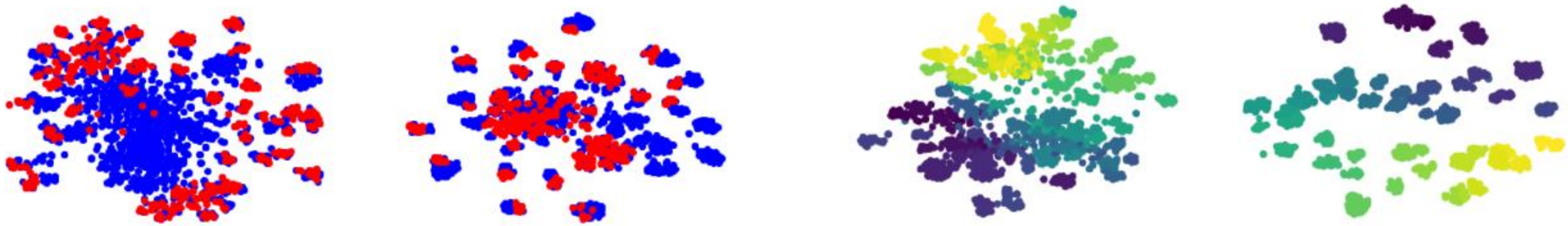
"MUDA shows superior performances to others"

- Small sized images : 28 x 28, 32 x 32, or 40 x 40
- Three conv layers for *F* & two FC layers for *C* are used.

* Transductive setting

Qualitative Analysis

- The embeddings of the learned features are depicted by t-SNE.
- MUDA significantly reduces divergence between domains and makes target samples more discriminative on the feature space.



(a) Before (by domain) (b) After (by domain) (c) Before (by class) (d) After (by class)

- Red & blue dots represent the source & target samples, respectively (a, b).
- SYNSIG (source) to GTSRB (target) setting is analyzed.

Results: Office-31 dataset

Table 2. Average accuracy from 10 random experiments

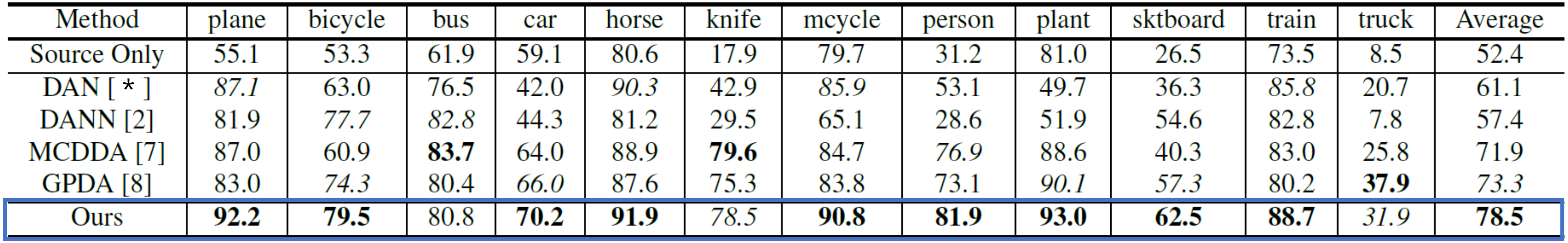
PROTOCOL	A→W	D→W	W→D	A→D	D→A	W→A	Avg.
Source Only	68.4	96.7	99.3	68.9	62.5	60.7	76.1
DANN [2]	82.0	96.9	99.1	79.7	68.2	67.4	82.2
ADDA [4]	86.2	96.2	98.4	77.8	69.5	68.9	82.8
JAN [18]	85.4	*97.4*	*99.8*	84.7	68.6	**70.0**	84.3
MADA [19]	**90.0**	*97.4*	99.6	*87.8*	70.3	66.4	*85.3*
GPDA [8]	83.9	97.3	**100.0**	85.5	**72.3**	68.8	84.6
Ours	*88.2*	**98.7**	*99.8*	**90.0**	*71.2*	*69.0*	**86.1**

“MUDA performs best or second best in all settings.”

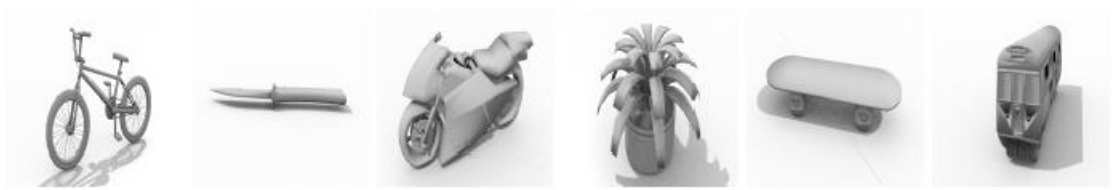
- Large sized images : 224 x 224 (4,562 images with 31 classes)
- ResNet-50 is employed for F & three FC layers are used for C.

Results: VisDA-17 dataset *(preliminary)*

Table 3. Per-category average accuracy from 10 random experiments



Method	plane	bicycle	bus	car	horse	knife	mcycle	person	plant	sktboard	train	truck	Average
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
DAN [*]	*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 [2]	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
MCDDA [7]	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
GPDA [8]	83.0	*74.3*	80.4	*66.0*	87.6	75.3	83.8	73.1	*90.1*	*57.3*	80.2	**37.9**	*73.3*
Ours	**92.2**	**79.5**	80.8	**70.2**	**91.9**	*78.5*	**90.8**	**81.9**	**93.0**	**62.5**	**88.7**	*31.9*	**78.5**



Source

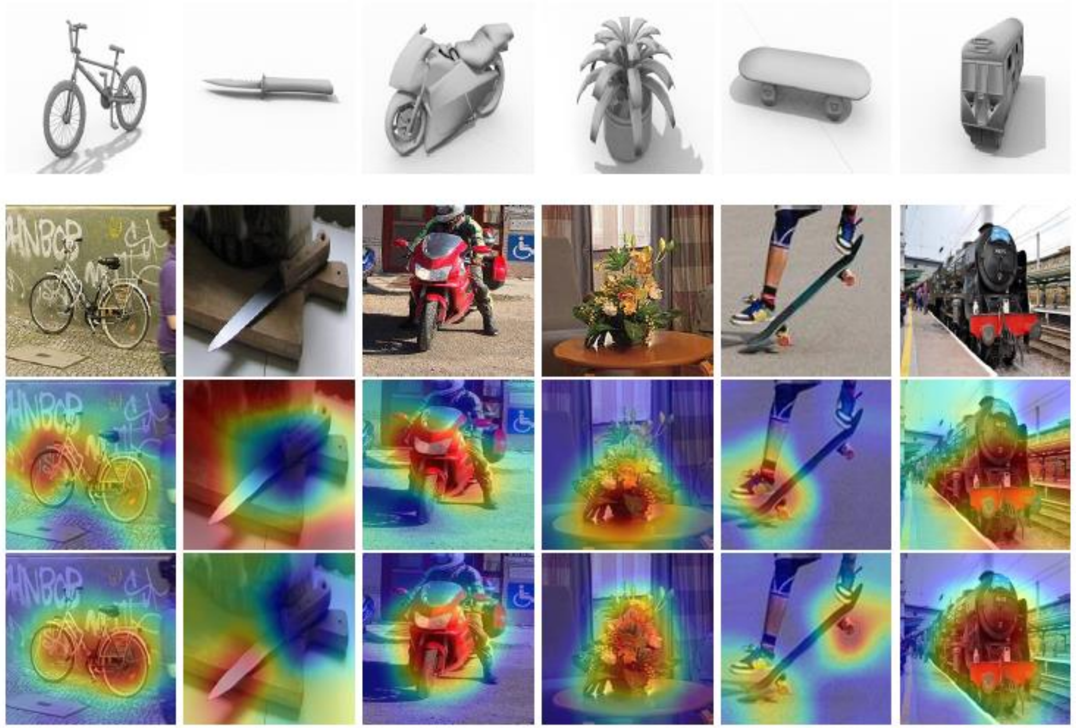
Target

- 152K synthetic images (source) to 55K real images (target) with 12 categories
- ResNet-101 is employed for F & three FC layers are used for C.

* Long et al., *In ICML*, 2015

Qualitative Analysis *(preliminary)*

- The saliency map for a prediction is analyzed using Grad-CAM.
- MUDA makes the model highlight the semantically meaningful regions properly thus generate robust decision boundary.



Source

Target

Source-only

MUDA

* The most discriminative regions are emphasized by deep red color, and the least relevant regions by deep blue color.

Conclusion

- We have presented that MUDA outperforms state-of-the-art methods via novel interpretation of the divergence between domains using the predictive uncertainty of model.

- We also devised an efficient method for computing model uncertainty using MC dropout.

- Model uncertainty is a useful surrogate to tackle domain shift.
  By minimizing it on the feature space, we can shrink the classifier hypothesis to contain only consistent classifiers on target domain.

Thanks for your attention!

JoonHo Lee, Gyemin Lee

{joonholee,gyemin}
@seoultech.ac.kr

MODEL UNCERTAINTY FOR UNSUPERVISED DOMAIN ADAPTATION

IEEE ICIP 2020 Presentation