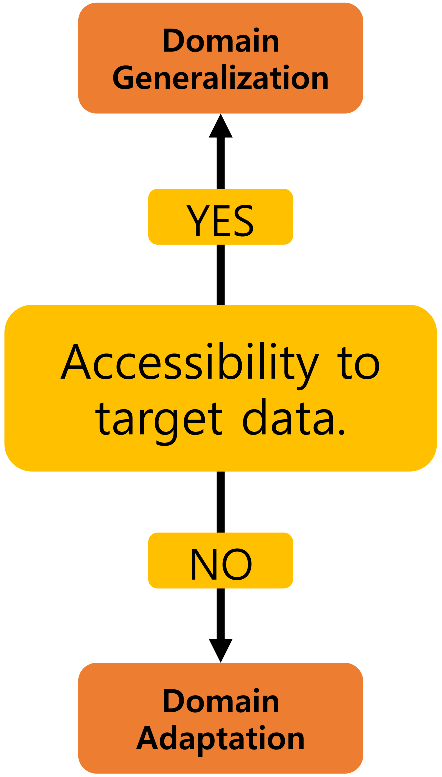
Separate to Adapt: Open Set Domain Adaptation via Progressive Separation

Hong Liu, et al., CVPR, 2019

2019/07/18, Kangyeol Kim

Data and Visual Analytics Lab

Domain adaptation taxonomy



Domain adaptation taxonomy

Domain Generalization

(-) Multi-source

(+) Attributes

Zero-shot DA

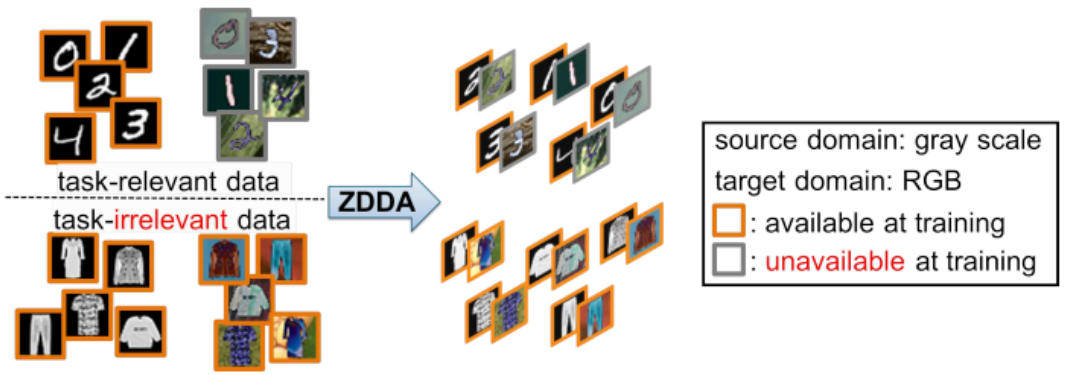
ZDDA, 2018

YES

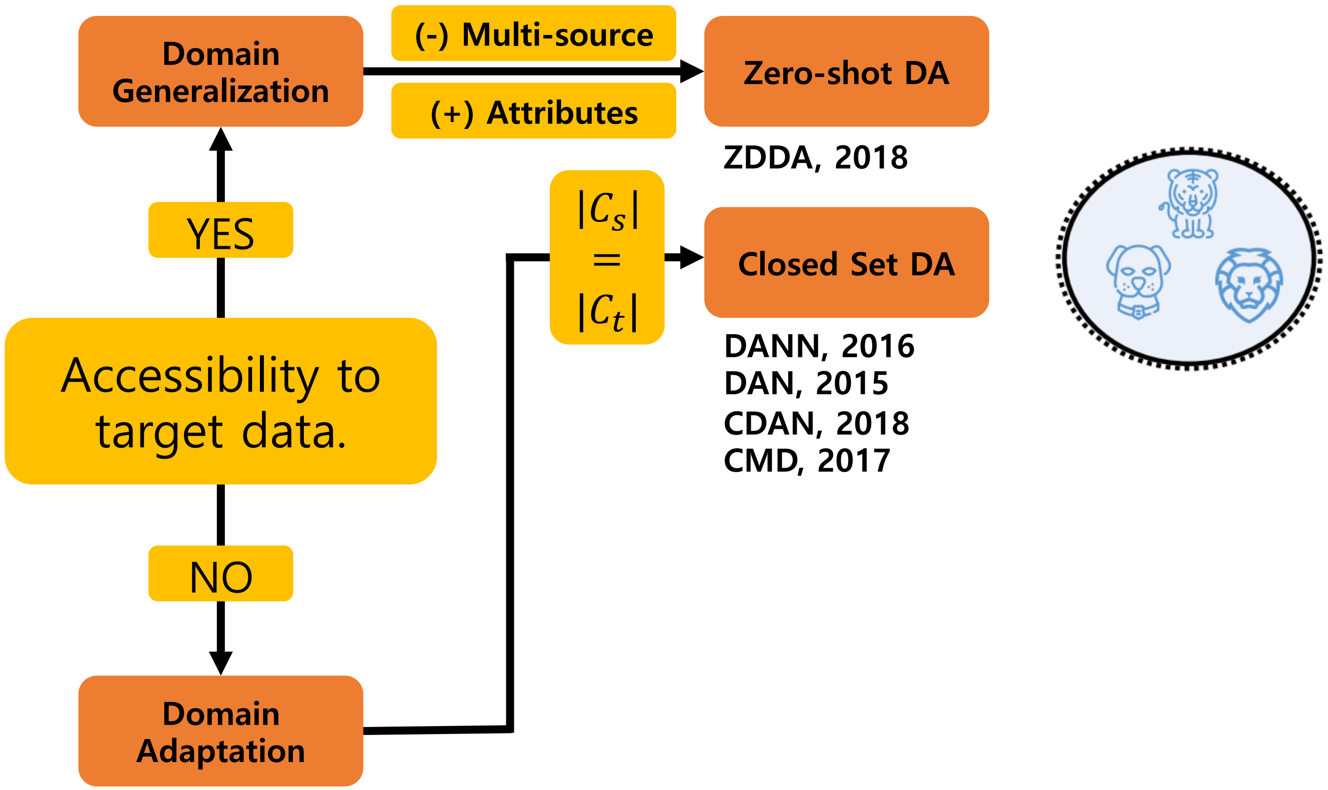
Accessibility to target data.

NO

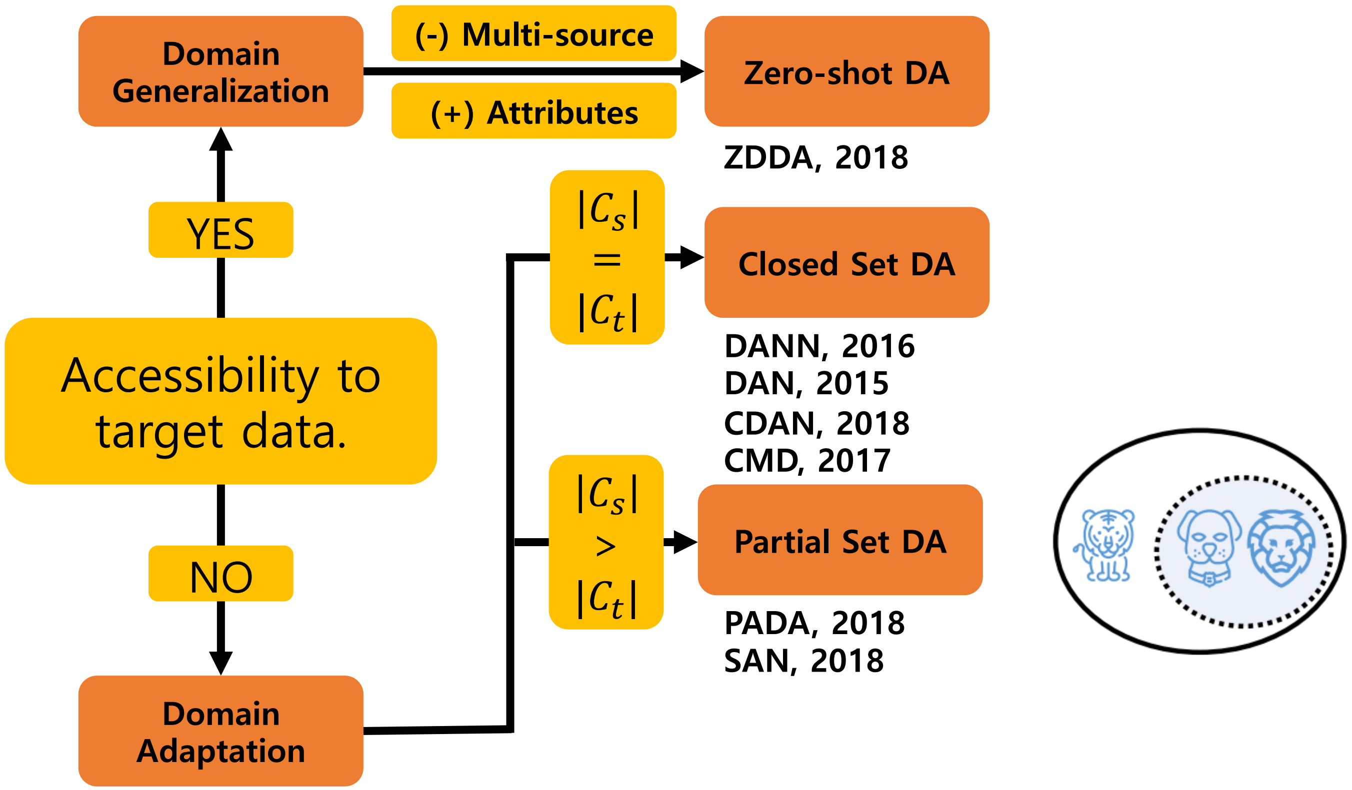
Domain Adaptation



Domain adaptation taxonomy



Domain adaptation taxonomy



Domain adaptation taxonomy

Accessibility to target data.

YES → Domain Generalization

Domain Generalization → (-) Multi-source / (+) Attributes → Zero-shot DA

ZDDA, 2018

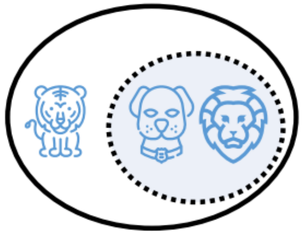
NO → Domain Adaptation

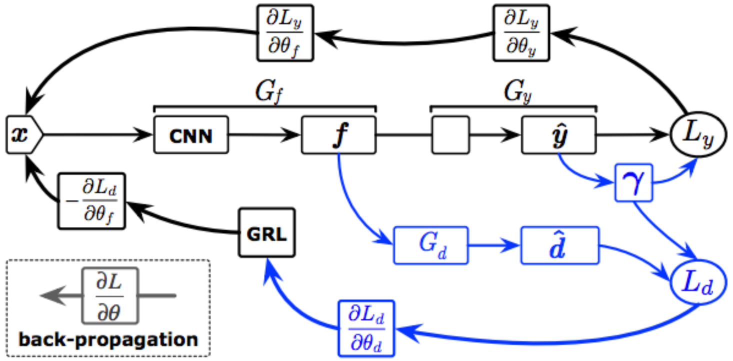
Domain Adaptation → $|C_s| = |C_t|$ → Closed Set DA

DANN, 2016
DAN, 2015
CDAN, 2018
CMD, 2017

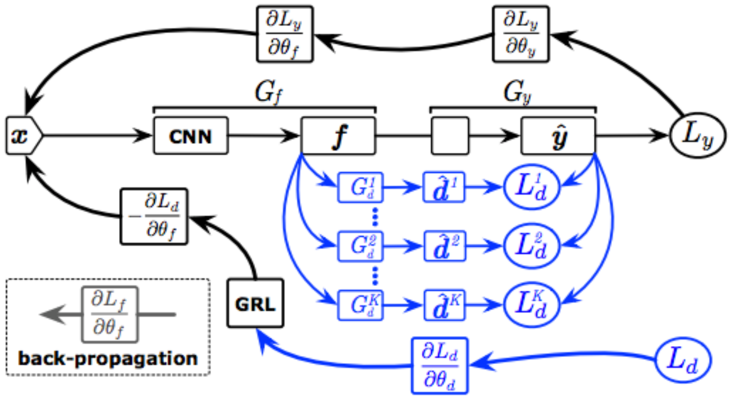
Domain Adaptation → $|C_s| > |C_t|$ → Partial Set DA

PADA, 2018
SAN, 2018



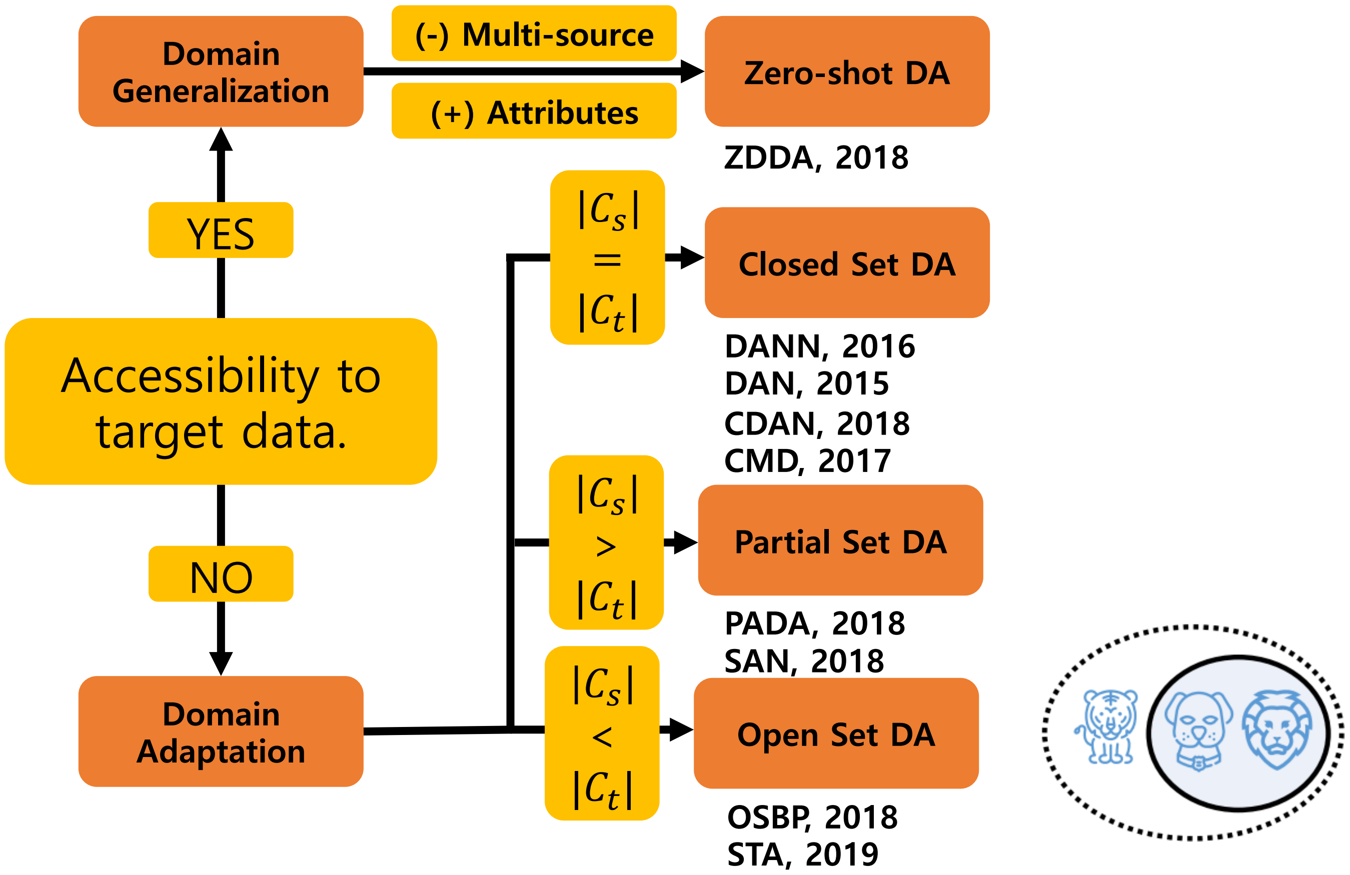


PADA, 2018 / Class-level weight

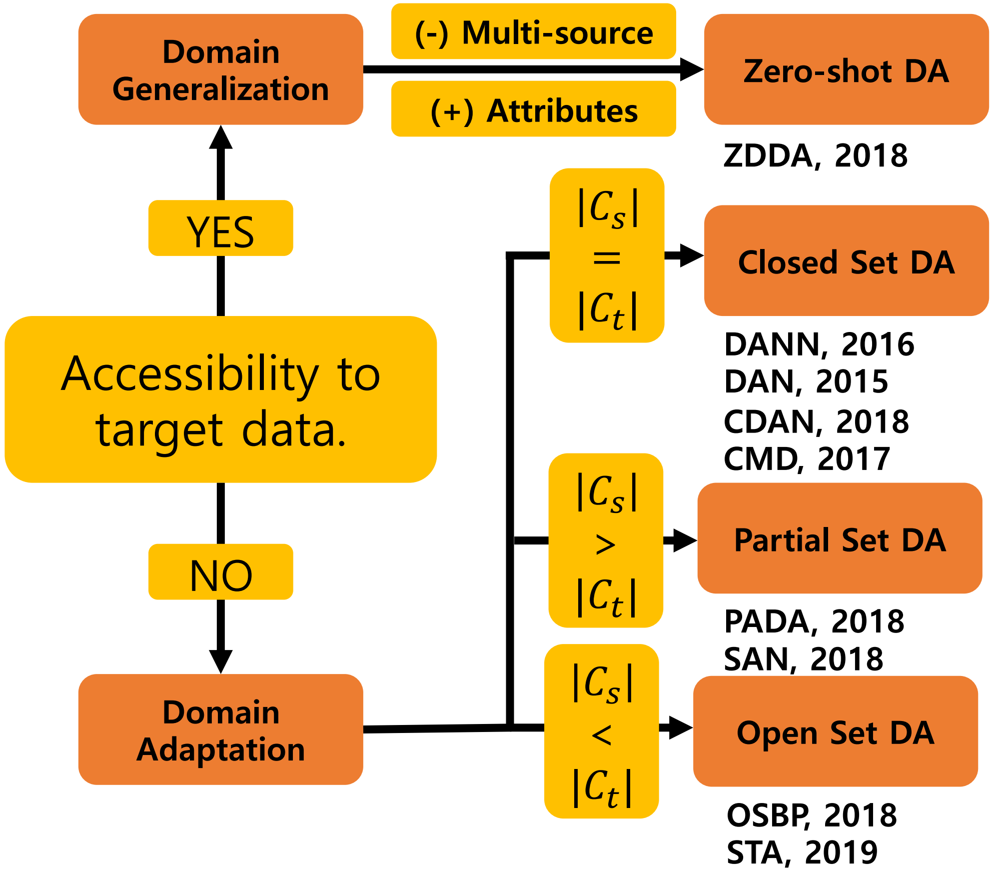


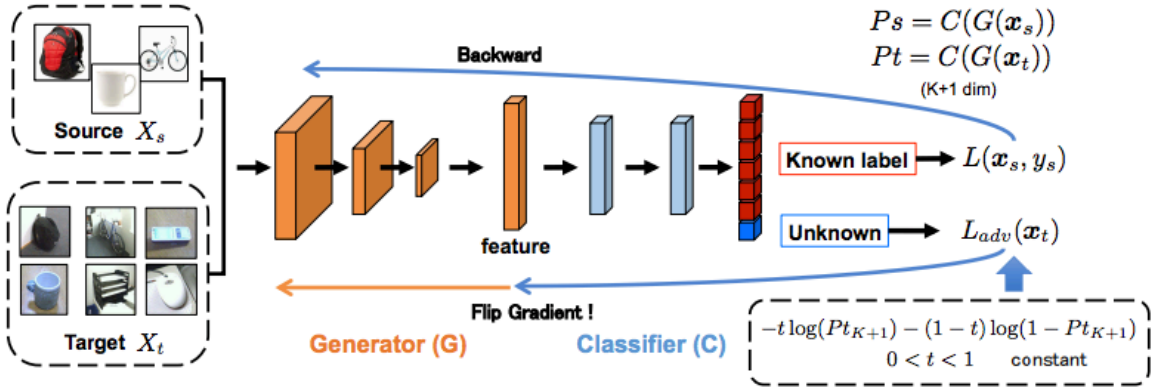
SAN, 2018 / Class + Instance-level weight

Domain adaptation taxonomy

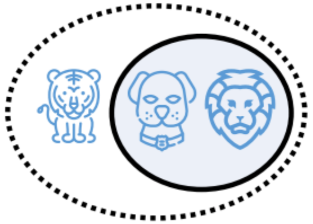


Domain adaptation taxonomy

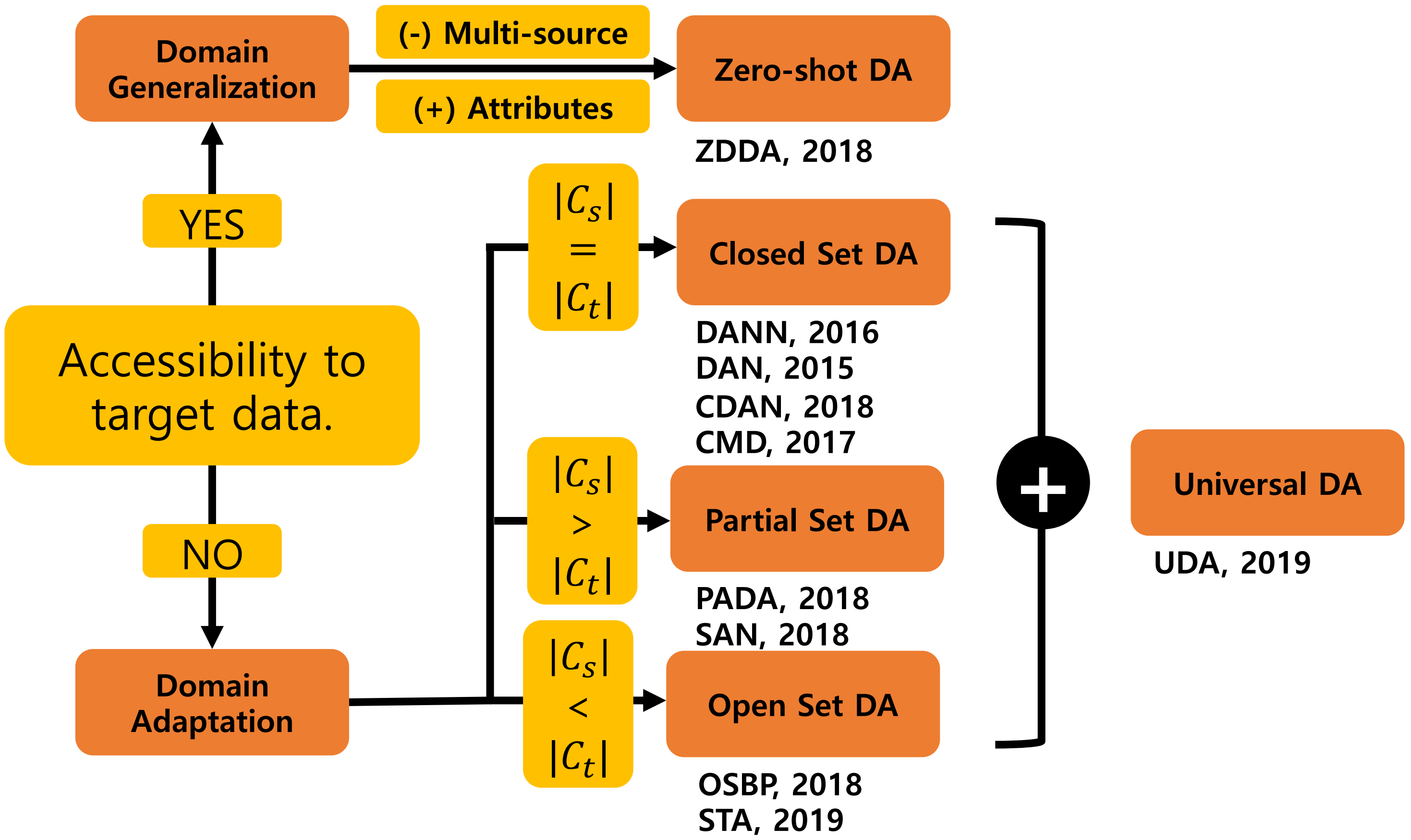




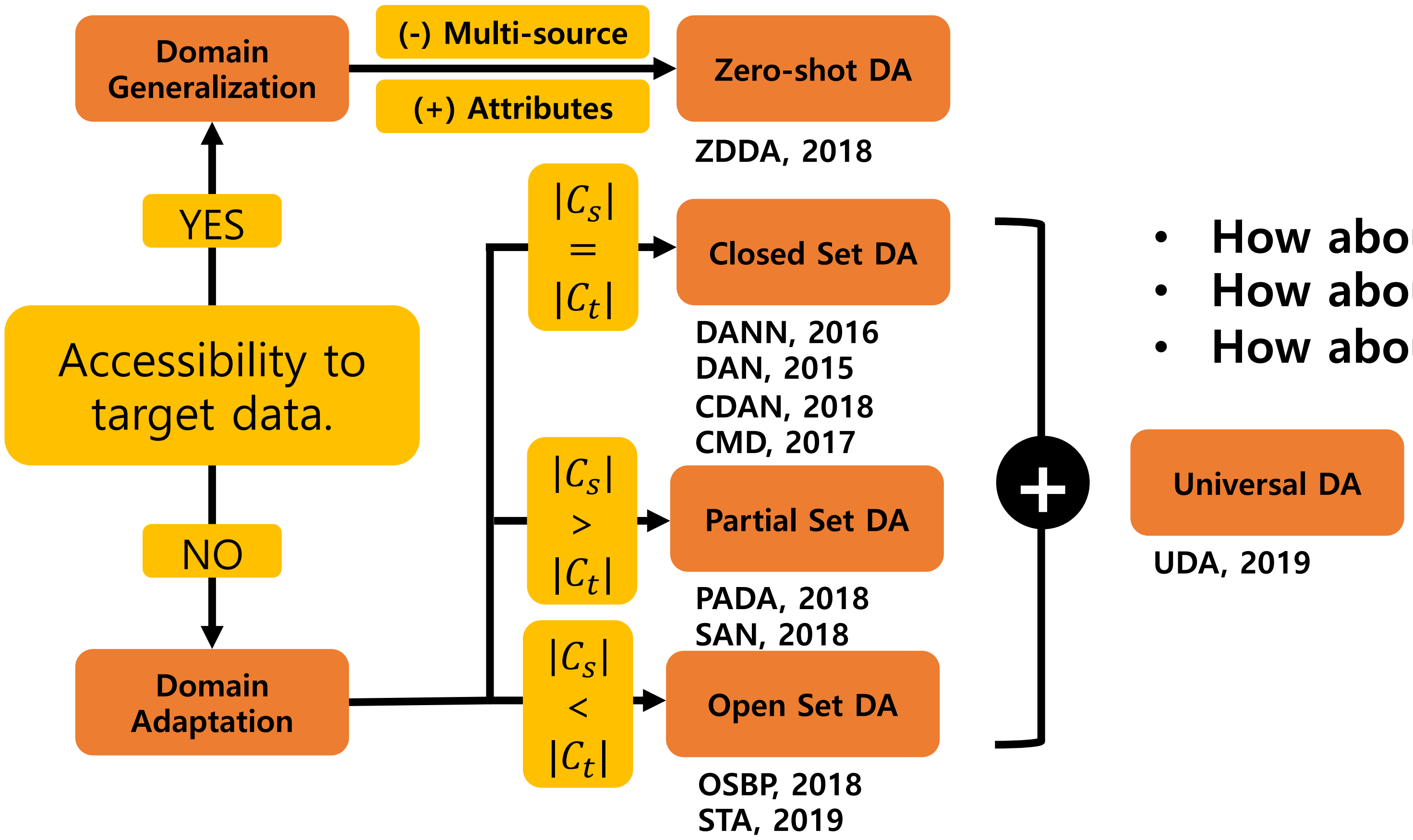
OSBP, 2018 / Augment an unknown class



Domain adaptation taxonomy

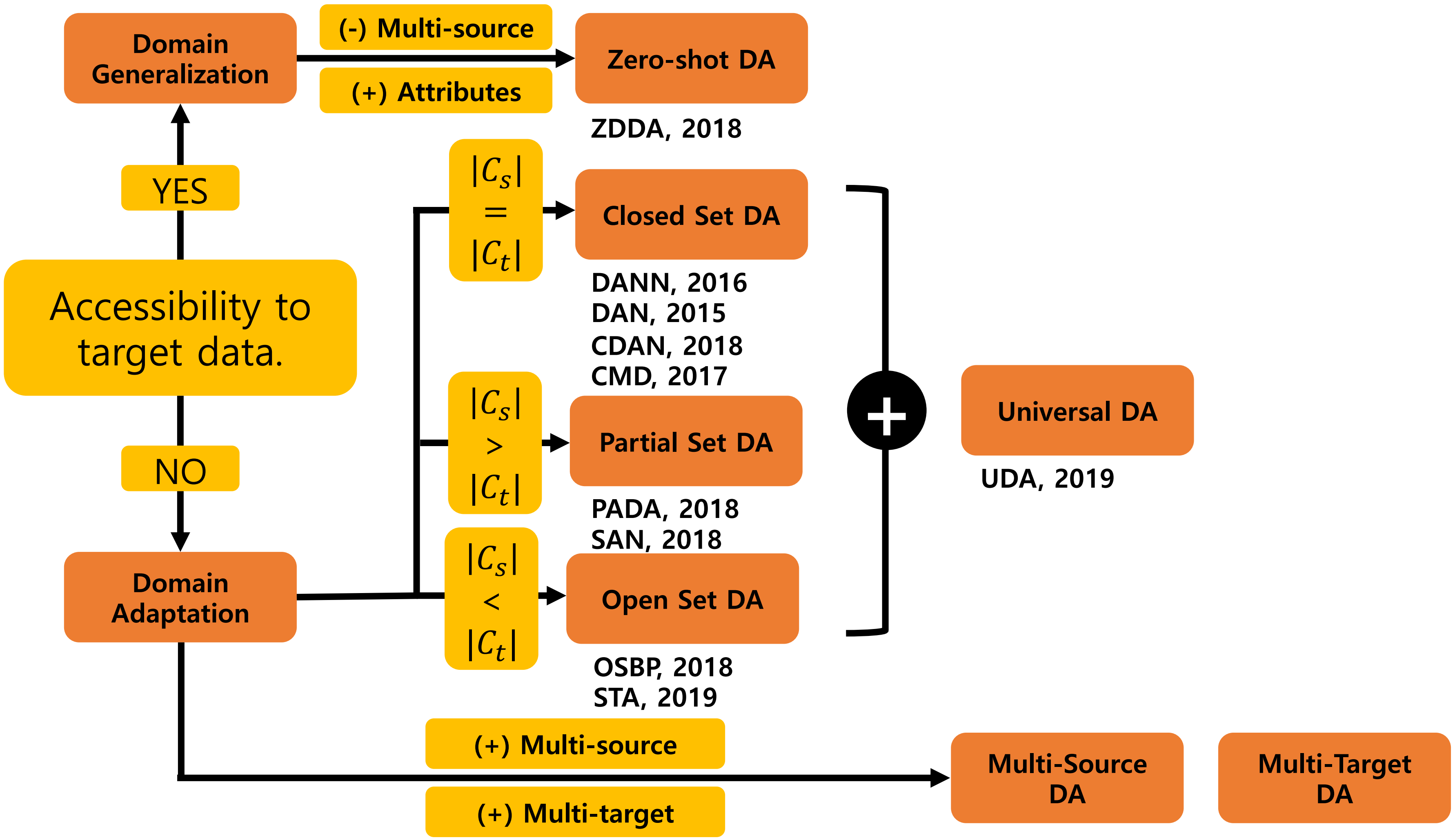


Domain adaptation taxonomy

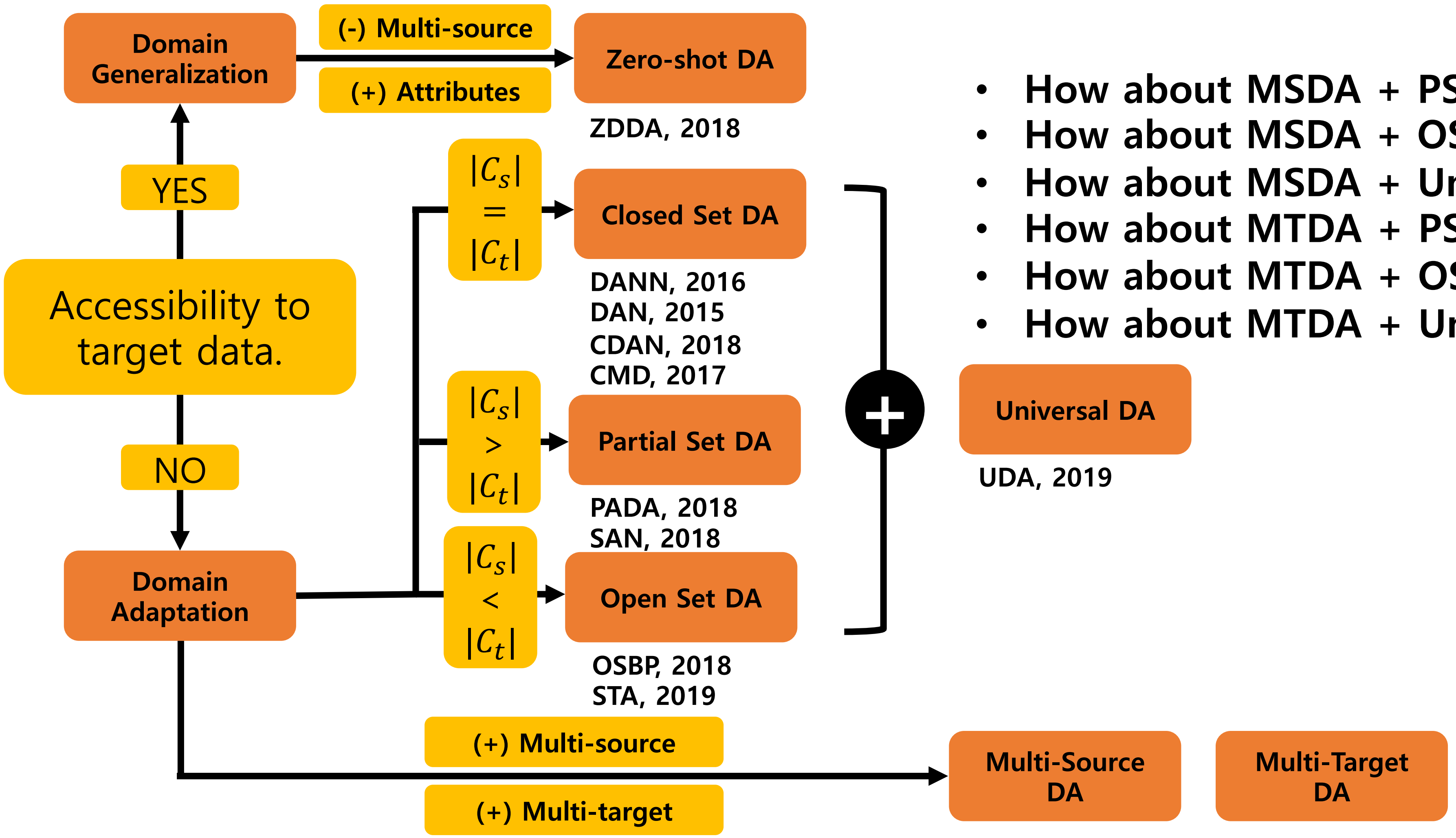


- **How about ZS + PSDA ?**
- **How about ZS + OSDA ?**
- **How about ZS + UniDA ?**

Domain adaptation taxonomy

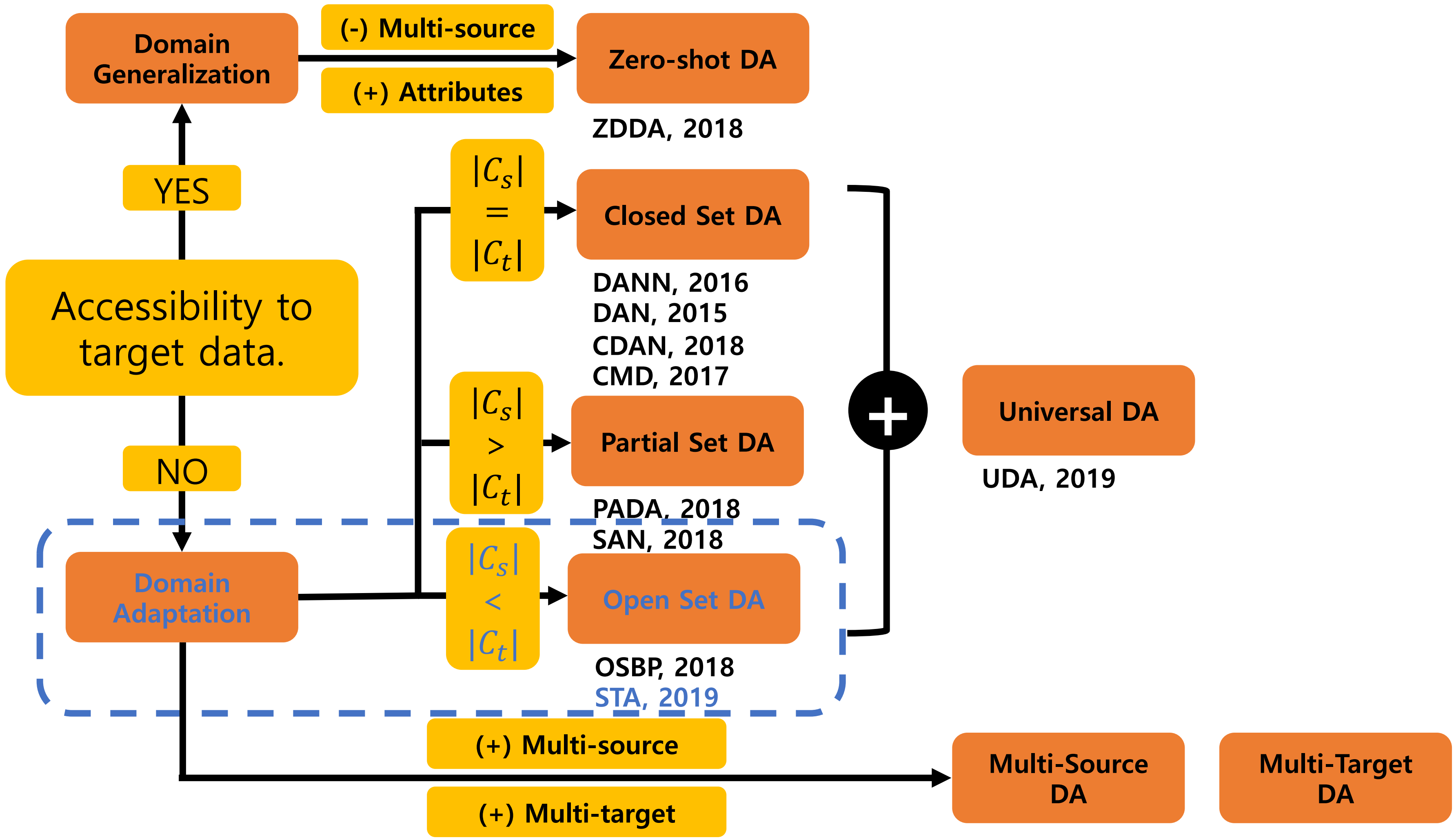


Domain adaptation taxonomy



- How about MSDA + PSDA ?
- How about MSDA + OSDA ?
- How about MSDA + UniDA ?
- How about MTDA + PSDA ?
- How about MTDA + OSDA ?
- How about MTDA + UniDA ?

Today, OSDA, Separate to Adapt



Main challenge of OSDA

- Aligning the whole distribution of source and target domains will be risky.
  - Unknown class in target domain impairs the alignment **(negative transfer)**

Main challenge of OSDA

- Aligning the whole distribution of source and target domains will be risky.
    - Unknown class in target domain impairs the alignment **(negative transfer)**

- **We need to identify the boundary between known and unknown classes. (Without any labels in target class)**

Progressive separation step

- For this, this paper adopt **multi-binary classifiers ($\boldsymbol{G_c}$)** and **unknown class identifier ($\boldsymbol{G_b}$)**, we want that :
  - G_c measures **the similarity between each target sample and each source class**, thus introduce loss L_s

$$L_s = \sum_{c=1}^{|\mathcal{C}_s|} \frac{1}{n_s} \sum_{i=1}^{n_s} L_{\text{bce}} \left(G_c \left(G_f \left(\mathbf{x}_i^s \right) \right), I \left(y_i^s, c \right) \right)$$

Progressive separation step

- For this, this paper adopt **multi-binary classifiers ($\boldsymbol{G_c}$)** and **unknown class identifier ($\boldsymbol{G_b}$)**, we want
  - G_c to measure **the similarity between each target sample and each source class**, thus introduce loss L_s

$$L_s = \sum_{c=1}^{|\mathcal{C}_s|} \frac{1}{n_s} \sum_{i=1}^{n_s} L_{\text{bce}} \left(G_c \left(G_f \left(\mathbf{x}_i^s \right) \right), I \left(y_i^s, c \right) \right)$$

  - G_b to measure the **probability that a sample belongs to unknown class,** thus introduce the following steps:

Progressive separation step

- For this, this paper adopt **multi-binary classifiers (G_c)** and **unknown class identifier (G_b)**, we want
  - G_c to measure **the similarity between each target sample and each source class**, thus introduce loss L_s

$$L_s = \sum_{c=1}^{|\mathcal{C}_s|} \frac{1}{n_s} \sum_{i=1}^{n_s} L_{\text{bce}} \left(G_c \left(G_f \left(\mathbf{x}_i^s \right) \right), I \left(y_i^s, c \right) \right)$$

  - G_b to measure the **probability that a sample belongs to unknown class,** thus introduce the following steps:
    - Pick highest probability from G_c (confidence to source class)
    - Cluster into 3 groups depending on confidence: highest, medium, lowest group and compute the means of highest, lowest group (s_h, s_l)
    - Annotate $s_j > s_h$ to known class, $s_j < s_l$ to unknown class.

Progressive separation step

- For this, this paper adopt **multi-binary classifiers ($\boldsymbol{G_c}$)** and **unknown class identifier ($\boldsymbol{G_b}$)**, we want
  - G_c to measure **the similarity between each target sample and each source class**, thus introduce loss L_s

$$L_s = \sum_{c=1}^{|\mathcal{C}_s|} \frac{1}{n_s} \sum_{i=1}^{n_s} L_{\text{bce}} \left(G_c \left(G_f \left(\mathbf{x}_i^s \right) \right), I \left(y_i^s, c \right) \right)$$

  - G_b to measure the **probability that a sample belongs to unknown class,** thus introduce the following steps:
    - Pick highest probability from G_c (confidence to source class)
    - Cluster into 3 groups depending on confidence: highest, medium, lowest group and compute the means of highest, lowest group (s_h, s_l)
    - Annotate $s_j > s_h$ to known class ($d_j = 0$), $s_j < s_l$ to unknown class ($d_j = 0$) => set X'

Progressive separation step

- For this, this paper adopt **multi-binary classifiers ($\boldsymbol{G_c}$)** and **unknown class identifier ($\boldsymbol{G_b}$)**, we want
  - G_c to measure **the similarity between each target sample and each source class**, thus introduce loss L_s

$$L_s = \sum_{c=1}^{|\mathcal{C}_s|} \frac{1}{n_s} \sum_{i=1}^{n_s} L_{\text{bce}} \left(G_c \left(G_f \left(\mathbf{x}_i^s \right) \right), I \left(y_i^s, c \right) \right)$$

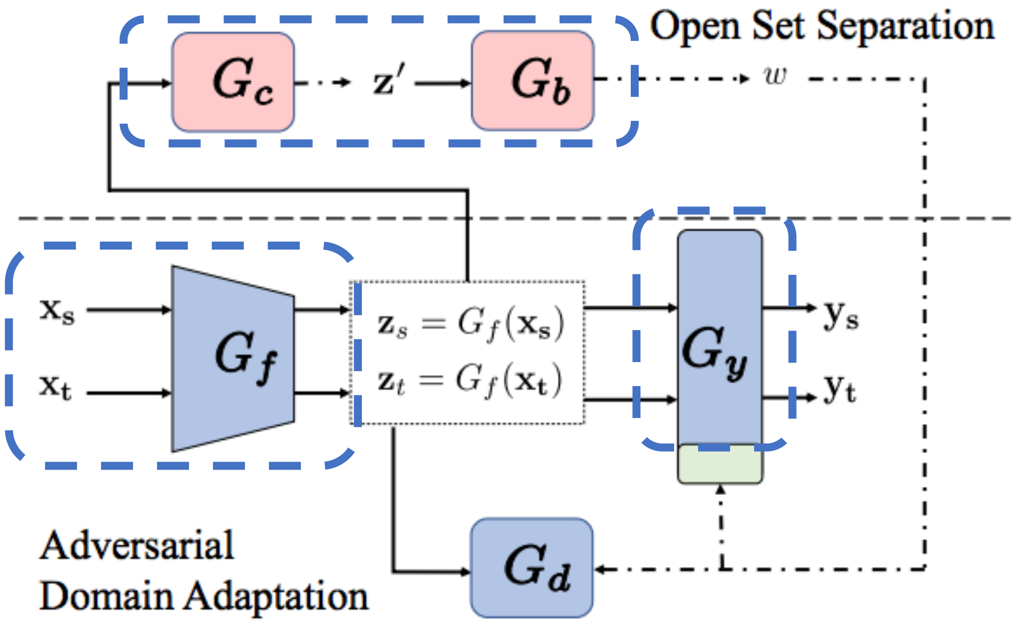
  - G_b to measure the **probability that a sample belongs to unknown class,** thus introduce the following steps:
    - Pick highest probability from G_c (confidence to source class)
    - Cluster into 3 groups depending on confidence: highest, medium, lowest group and compute the means of highest, lowest group (s_h, s_l)
    - Annotate $s_j > s_h$ to known class ($d_j = 0$), $s_j < s_l$ to unknown class ($d_j = 0$) => set X', finally train G_b with

$$L_b = \frac{1}{|\mathbf{X}'|} \sum_{\mathbf{x}_j \in \mathbf{X}'} L_{\text{bce}} \left(G_b \left(G_f \left(\mathbf{x}_j \right) \right), d_j \right).$$

Progressive separation step

- Adding supervision loss using source dataset, whole losses in this step:

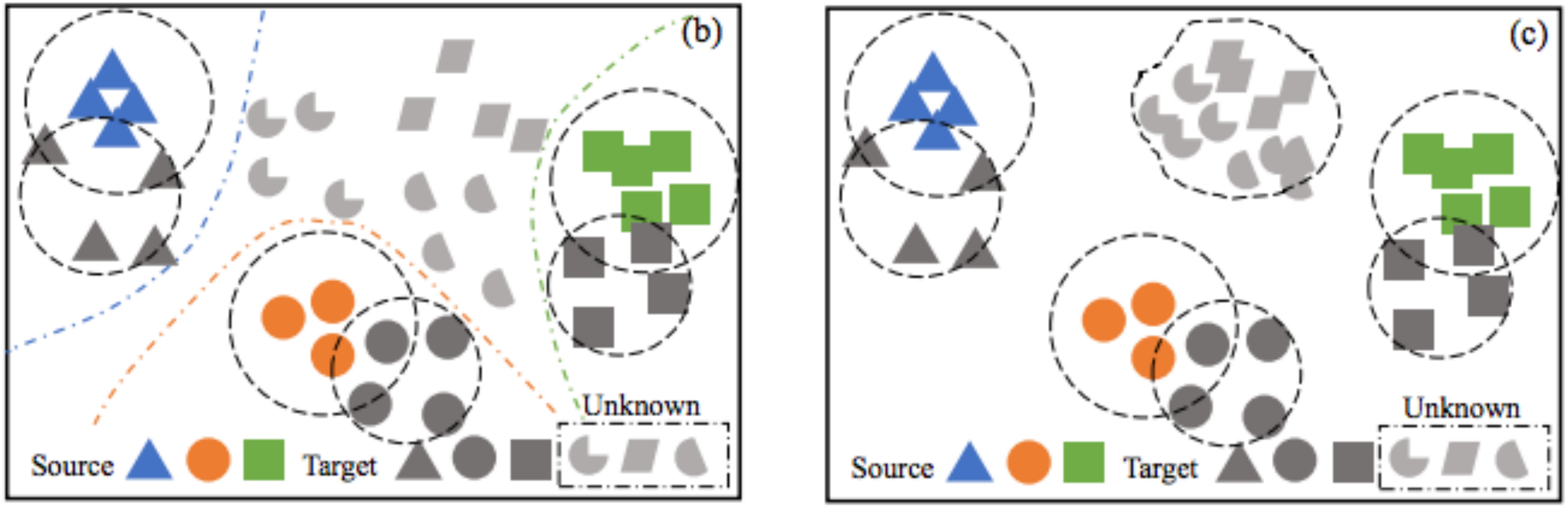
$$L^s_{\text{cls}} = \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y \left(G_y^{1:|\mathcal{C}_s|} \left(G_f \left(\mathbf{x}_i \right) \right), y_i \right)$$



$$(\hat{\theta}_f, \hat{\theta}_y, \hat{\theta}_b, \hat{\theta}_c|_{c=1}^{|\mathcal{C}_s|}) = \underset{\theta_f, \theta_y, \theta_b, \theta_c|_{c=1}^{|\mathcal{C}_s|}}{\arg\min} \quad L^s_{\text{cls}} + L_s + L_b.$$

Progressive separation step

- **Meaning of progressive separation step**
  - By multi-binary classifiers, we can get **the decision boundaries for each class.** (whether a sample belongs to the class or not)
  - Furthermore, by G_b, **multi-binary classifiers roughly need to filter the unknown class samples in target domain.**



Weighted adaptation step

- Next, we need to focus the model on aligning the distribution of source and target data in the shared label space.
  - Using G_b, we can obtain instance-level a probability to be unknown class: $\boldsymbol{w_j} = \boldsymbol{G_b}(\boldsymbol{G_f}(\boldsymbol{x_j}))$.
  - Exploiting w_j, a weighted loss for adversarial adaptation is defined . Note that $(\boldsymbol{1} - \boldsymbol{w_j})$ **is the probability to be known class.**

$$
\begin{aligned}
L_d = & \frac{1}{n_s} \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_{\mathrm{bce}}\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right) \\
& + \frac{1}{\sum_{\mathbf{x}_j \in \mathcal{D}_t}\left(1 - w_j\right)} \sum_{\mathbf{x}_j \in \mathcal{D}_t}\left(1 - w_j\right) L_{\mathrm{bce}}\left(G_d\left(G_f\left(\mathbf{x}_j\right)\right), d_j\right).
\end{aligned}
$$

Weighted adaptation step

- Additionally, to get more accurate unknown class boundary, unknown class branch of G_f is trained using weights of w_j, and $l_{uk} \in \{0, 1\}$ which is determined by w_j (? Thresholding)

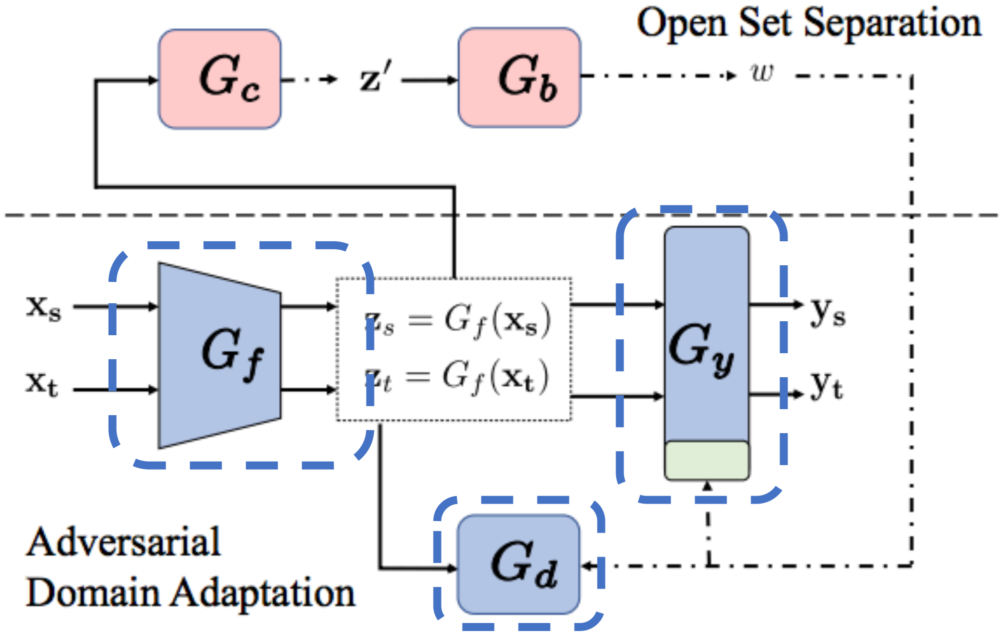
$$L_{\mathrm{cls}}^{t} = \frac{1}{|\mathcal{C}_s|} \frac{1}{\sum_{\mathbf{x}_j \in \mathcal{D}_t} w_j} \sum_{\mathbf{x}_j \in \mathcal{D}_t} w_j L_y \left(G_y^{|\mathcal{C}_s|+1} \left(G_f \left(\mathbf{x}_j \right) \right), l_{\mathrm{uk}} \right)$$

- Moreover, entropy minimization loss on the known classes of target domain is incorporated to enforce the decision boundary to pass through low-density area in the target domain (? ref.)

$$L_e = \frac{1}{\sum_{\mathbf{x}_j \in \mathcal{D}_t} (1 - w_j)} \sum_{\mathbf{x}_j \in \mathcal{D}_t} (1 - w_j) H \left(G_y^{1:|\mathcal{C}_s|} \left(G_f \left(\mathbf{x}_j \right) \right) \right)$$

Weighted adaptation step

- Then, the total losses in this step is described as:

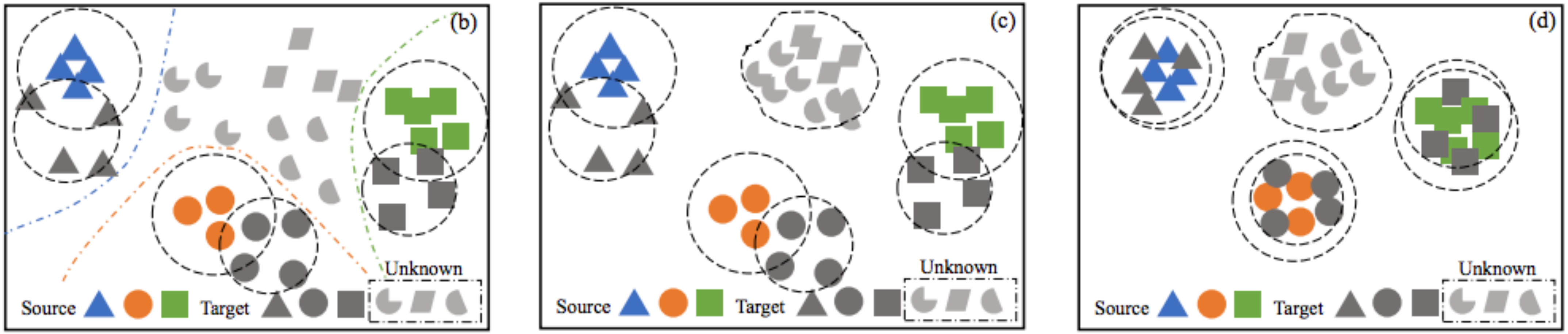


$$(\hat{\theta}_y, \hat{\theta}_d) = \underset{\theta_y, \theta_d}{\arg\min}\, L^s_{\text{cls}} + L^t_{\text{cls}} + L_d + \lambda L_e,$$

$$(\hat{\theta}_f) = \underset{\theta_f}{\arg\min}\, L^s_{\text{cls}} + L^t_{\text{cls}} - L_d + \lambda L_e,$$

Separation and adaptation alternatively

- The two steps are implemented alternately resulting more accurate separation boundary and feature alignment between source and target domain within the known classes



Metric for open set DA

- **OS**
  - normalized accuracy for all the classes including the unknown as one class
- **OS***
  - normalized accuracy only on known classes
- **ALL**
  - the accuracy of all instances (without averaging accuracy over the classes)
- **UNK**
  - the accuracy of unknown samples

Experiment, Comparison with baselines

Table 1. Classification accuracy (%) of open set domain adaptation tasks on Digits (LeNet) and VisDA-2017 (VGGNet)

Method	Digits SVHN → MNIST				USPS → MNIST				MNIST → USPS				Avg				VisDA-2017 Synthetic → Real								
	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	OS	OS*	ALL	UNK	bicycle	bus	car	motorcycle	train	truck	UNK	OS	OS*
OSVM [13]	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	31.7	51.6	66.5	70.4	**88.5**	20.8	38.0	52.5	54.9
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	39.0	50.1	64.2	79.9	86.6	16.3	44.8	54.4	56.0
DANN+OSVM	62.9	75.3	39.2	0.70	84.4	92.4	72.9	0.90	33.8	40.5	21.4	44.3	60.4	69.4	44.5	15.3	31.8	56.6	**71.7**	77.4	87.0	22.3	41.9	55.5	57.8
ATI-λ	67.6	66.5	69.8	73.0	82.4	81.5	84.0	86.7	86.8	89.6	82.8	73.0	78.9	79.2	78.9	77.6	46.2	57.5	56.9	79.1	81.6	**32.7**	65.0	59.9	59.0
OSBP	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	51.1	67.1	42.8	84.2	81.8	28.0	**85.1**	62.9	59.2
STA	**76.9**	**75.4**	**80.0**	**84.4**	92.2	**91.3**	93.9	96.5	**93.0**	**94.9**	**90.3**	83.5	**87.3**	**87.2**	**88.1**	**88.1**	**52.4**	**69.6**	59.9	**87.8**	86.5	27.2	84.1	**66.8**	**63.9**

Table 2. Classification Accuracy (%) of open set domain adaptation tasks on Office-31 (ResNet-50)

Method	A → W		A → D		D → W		W → D		D → A		W → A		Avg	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
ResNet [9]	82.5±1.2	82.7±0.9	85.2±0.3	85.5±0.9	94.1±0.3	94.3±0.7	96.6±0.2	97.0±0.4	71.6±1.0	71.5±1.1	75.5±1.0	75.2±1.6	84.2	84.4
RTN [19]	85.6±1.2	88.1±1.0	89.5±1.4	90.1±1.6	94.8±0.3	96.2±0.7	97.1±0.2	98.7±0.9	72.3±0.9	72.8±1.5	73.5±0.6	73.9±1.4	85.4	86.8
DANN [4]	85.3±0.7	87.7±1.1	86.5±0.6	87.7±0.6	**97.5**±0.2	**98.3**±0.5	**99.5**±0.1	**100.0**±.0	75.7±1.6	76.2±0.9	74.9±1.2	75.6±0.8	86.6	87.6
OpenMax [2]	87.4±0.5	87.5±0.3	87.1±0.9	88.4±0.9	96.1±0.4	96.2±0.3	98.4±0.3	98.5±0.3	83.4±1.0	82.1±0.6	82.8±0.9	82.8±0.6	89.0	89.3
ATI-λ [25]	87.4±1.5	88.9±1.4	84.3±1.2	86.6±1.1	93.6±1.0	95.3±1.0	96.5±0.9	98.7±0.8	78.0±1.8	79.6±1.5	80.4±1.4	81.4±1.2	86.7	88.4
OSBP [30]	86.5±2.0	87.6±2.1	88.6±1.4	89.2±1.3	97.0±1.0	96.5±0.4	97.9±0.9	98.7±0.6	88.9±2.5	90.6±2.3	85.8±2.5	84.9±1.3	90.8	91.3
STA	**89.5**±0.6	**92.1**±0.5	**93.7**±1.5	**96.1**± 0.4	**97.5**±0.2	96.5±0.5	**99.5**±0.2	99.6±0.1	**89.1**±0.5	**93.5**±0.8	**87.9**±0.9	**87.4**±0.6	**92.9**	**94.1**

Experiment, Ablation study

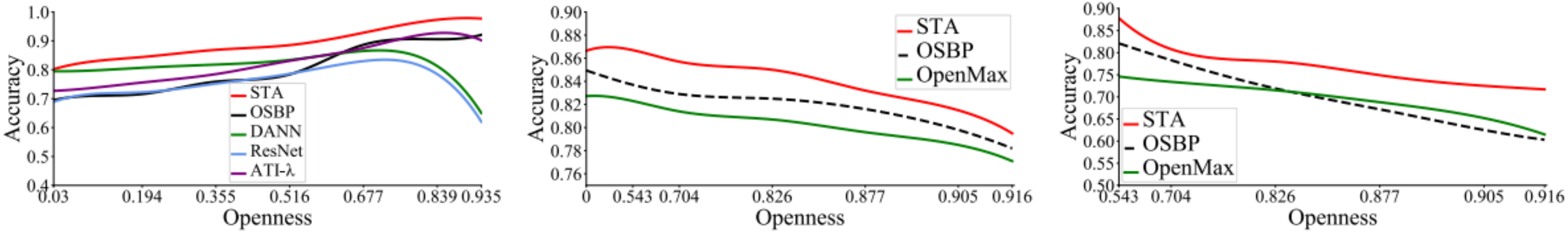
- **Without weight (w),** negative transfer
- **Without multi-binary classifiers (c) just softmax classifier,** worsen to measure to compute the similarites with source class independently
- **Without G_b (b),** naïve results from multi-binary classifiers are not enough
- **Without alternation (j)**

Table 4. Classification accuracy (%) of STA and its three variants on Office-31 (ResNet-50)

Method	A → W		A → D		D → W		W → D		D → A		W → A		Avg	
	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*	OS	OS*
STA w/o w	87.5±1.4	91.4±1.1	83.0±1.2	89.6±1.2	96.2±0.9	97.3±0.4	98.1±0.7	**100.0**±.0	80.3±1.5	79.3±1.5	71.2±1.2	74.3±1.2	86.1	88.7
STA w/o c	**90.4**±1.7	90.6±1.7	91.5±1.4	91.3±1.4	95.9±1.0	96.7±1.1	98.8±0.6	98.7±0.5	87.4±1.5	87.8±1.5	84.6±1.7	85.2±1.7	91.5	91.8
STA w/o b	85.0±1.5	89.0±1.5	90.6±1.2	91.5±1.3	94.8±1.9	**97.6**±0.8	96.2±0.6	98.2±0.5	77.7±2.2	82.5±2.4	78.9±2.6	83.6±3.5	87.2	90.4
STA w/o j	89.0±1.3	**92.8**±1.2	**94.8**±1.5	95.9±1.0	96.4±0.6	96.2±0.3	98.8±0.7	99.4±0.2	**89.7**±1.4	**93.6**±1.4	85.1±1.1	86.7±1.1	92.5	93.9
STA	89.5±0.6	92.1±0.5	93.7±1.5	**96.1**± 0.4	**97.5**±0.2	96.5±0.5	**99.5**±0.2	99.6±0.1	89.1±0.5	93.5±0.8	**87.9**±0.9	**87.4**±0.6	**92.9**	**94.1**

Experiment, Openness variation

- Define openness as $\boldsymbol{O} = \mathbf{1} - \frac{|C_s|}{|C_t|}$
- STA is robust to any openness, even the O = 0 (meaning no unknown class), which indicates the framework can play a role as **filtering a noisy target data.**



(a) Office-31 OS (b) Caltech-ImageNet Known (c) Caltech-ImageNet Unknown

Figure 4. Accuracy (OS) w.r.t. different openness levels in the target domain.

Experiment, Weight visualization

(a) **A** → **W** (Office-31)

(b) **VisDA-2017**

Thanks a lot !!
Any Questions?