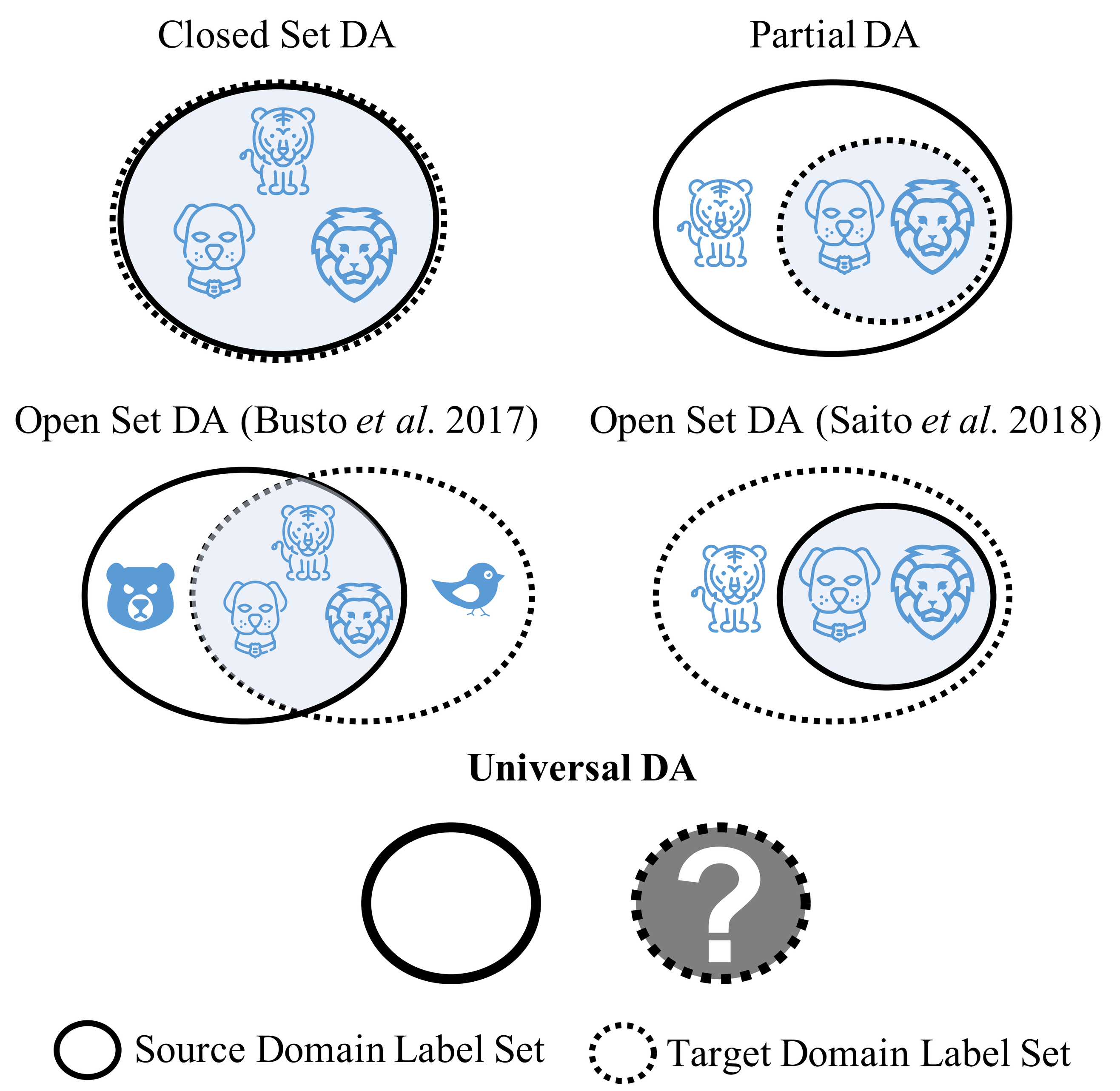
Universal Domain Adaptation

Kaichao You, Mingsheng Long, Zhangjie Cao, Jianmin Wang, and Michael I. Jordan

CVPR LONG BEACH CALIFORNIA June 16-20, 2019

Domain Adaptation Setting

- Domain adaptation: transfer from the source domain to the target domain
  - domain gap: The target domain differs with the source domain
- Several domain adaptation settings
  - Classified by the relationship between label sets of source and target domains

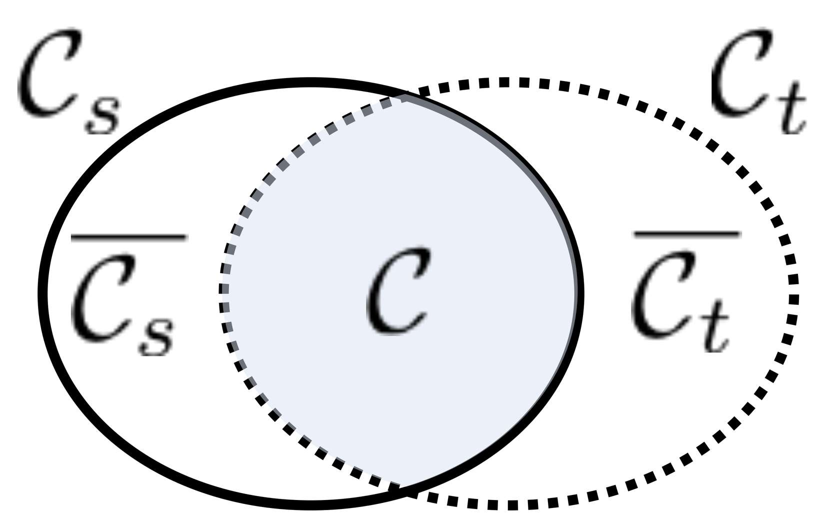


Practical Value of Universal Domain Adaptation

- Domain adaptation in the wild
- E.g., transfer an animal recognizer to an unknown area
  - Unlabeled data is easily accessible by setting up a camera
  - There may exist some unknown species that are not in the training set
  - Domain gap exists and the trained animal recognizer cannot be directly used

Challenges in Universal Domain Adaptation

- **Category gap** arises from the difference of the label sets
- Commonness: $\xi = \frac{|\mathcal{C}_s \cap \mathcal{C}_t|}{|\mathcal{C}_s \cup \mathcal{C}_t|}$



- **Domain gap** between the source and target data in the common label set $\mathcal{C}$
- **Recognize unknown classes** is hard for neural networks

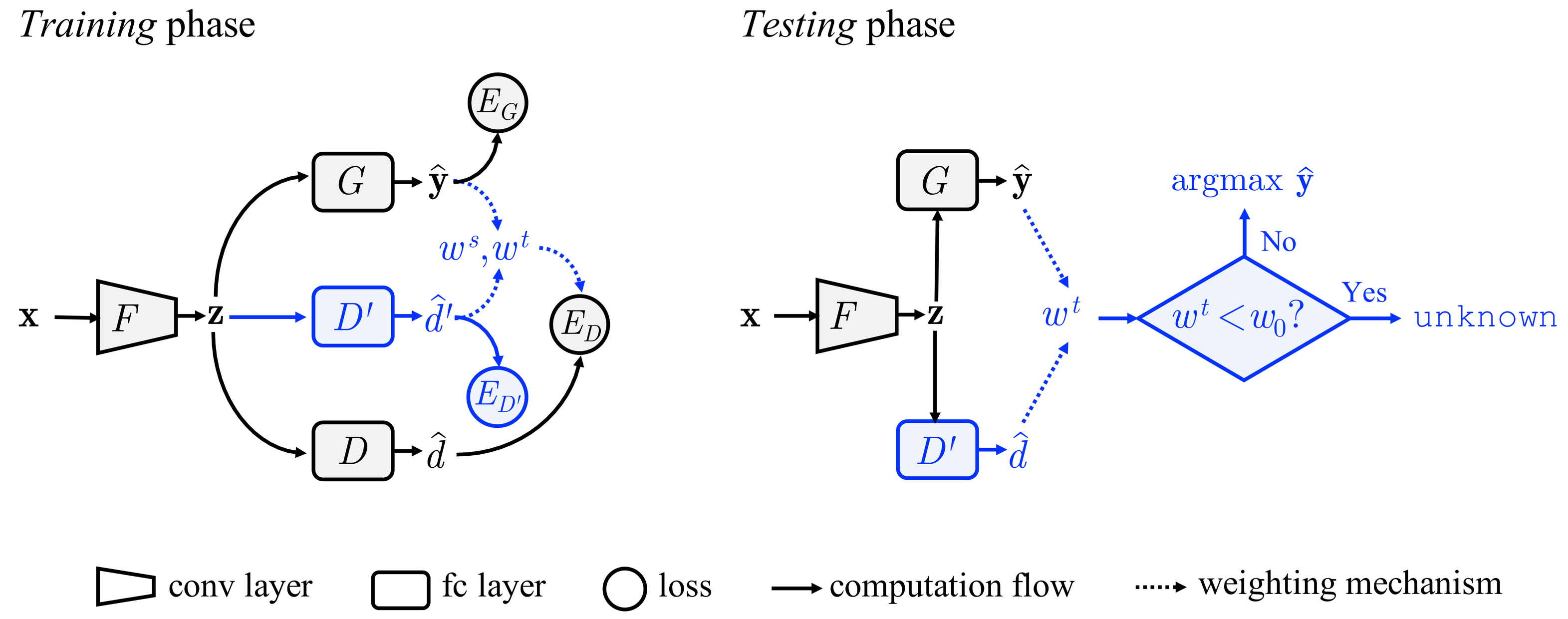
Universal Adaptation Network

- Insights
  - Need a transferability criterion to identify data from $\mathcal{C}$, $\overline{\mathcal{C}}_s$, $\overline{\mathcal{C}}_t$
  - Align data in the common label set $\mathcal{C}$ maximally
  - Ignored data in the private label set $\overline{\mathcal{C}}_t$ and $\overline{\mathcal{C}}_s$ during adaptation
  - Reject target examples with low transferability as "unknown" class
- Ideal transferability criterion

  These inequalities should hold for ideal transferability criterion.

$$\mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} w^s(\mathbf{x}) > \mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} w^s(\mathbf{x})$$
$$\mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} w^t(\mathbf{x}) > \mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} w^t(\mathbf{x})$$

- Architecture



- $\mathbf{x}$: input image
- $\mathbf{z}$: extracted feature
- D: domain discriminator
- E_G, E_D: loss function
- F: feature extractor (ResNet-50)
- G: classifier
- D': auxiliary discriminator
- w_0: threshold
- End-to-End training

$$E_G = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim p} L\left(\mathbf{y}, G(F(\mathbf{x}))\right)$$
$$E_{D'} = -\mathbb{E}_{\mathbf{x}\sim p} \log D'\left(F(\mathbf{x})\right) - \mathbb{E}_{\mathbf{x}\sim q} \log\left(1 - D'\left(F(\mathbf{x})\right)\right)$$
$$E_D = -\mathbb{E}_{\mathbf{x}\sim p} w^s(\mathbf{x}) \log D\left(F(\mathbf{x})\right) - \mathbb{E}_{\mathbf{x}\sim q} w^t(\mathbf{x}) \log\left(1 - D\left(F(\mathbf{x})\right)\right)$$
$$\max_{D} \min_{F,G} E_G - \lambda E_D \qquad \min_{D'} E_{D'}$$

- Transferability criterion
  - Domain Similarity

    $\hat{d}'$: the domain similarity of each sample
    - For a source sample, smaller $\hat{d}'$ means that it is more similar to the target domain
    - For a target sample, larger $\hat{d}'$ means that it is more similar to the source domain.
    - $\mathbb{E}_{\mathbf{x}\sim p_{\overline{\mathcal{C}}_s}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim p_{\mathcal{C}}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim q_{\mathcal{C}}} \hat{d}' > \mathbb{E}_{\mathbf{x}\sim q_{\overline{\mathcal{C}}_t}} \hat{d}'$
  - Prediction Uncertainty

    $H(\hat{\mathbf{y}})$: the entropy of each prediction
    - Predictions are certain for source samples
    - Predictions are uncertain for target samples
    - Influenced by the high entropy samples from $q_{\mathcal{C}}$, the entropy of samples from $p_{\mathcal{C}}$ becomes larger
    - Samples from $p_{\overline{\mathcal{C}}_s}$ keep highest certainty

$$w^s(\mathbf{x}) = \frac{H(\hat{\mathbf{y}})}{\log|\mathcal{C}_s|} - \hat{d}'(\mathbf{x})$$
$$w^t(\mathbf{x}) = \hat{d}'(\mathbf{x}) - \frac{H(\hat{\mathbf{y}})}{\log|\mathcal{C}_s|}$$

Experimental Results

- State-of-the-art performance

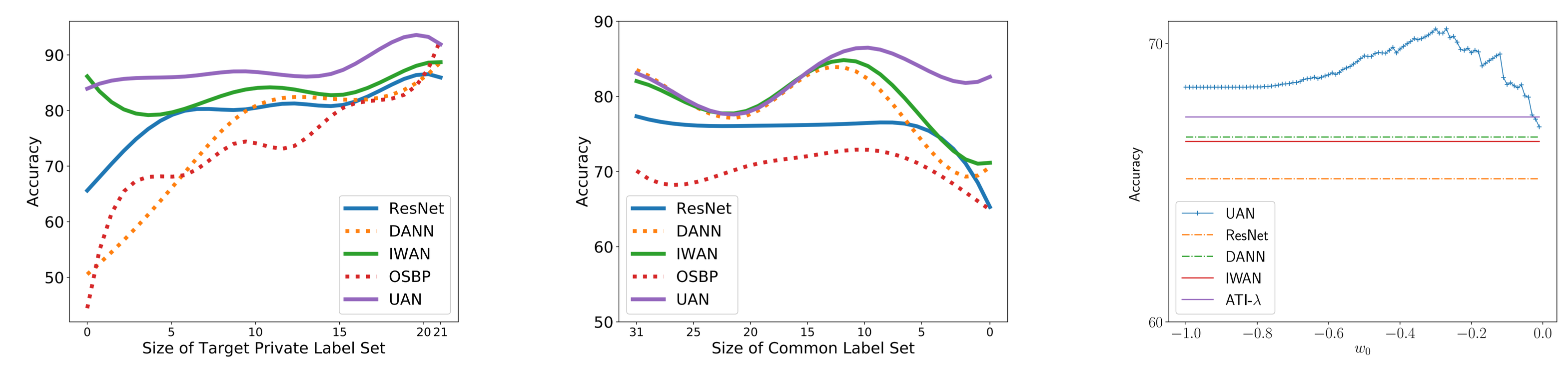
Table: Average class accuracy (%) on **Office-Home**($\xi = 0.15$) dataset (ResNet)

Method	Office-Home												
	Ar → Cl	Ar → Pr	Ar → Rw	Cl → Ar	Cl → Pr	Cl → Rw	Pr → Ar	Pr → Cl	Pr → Rw	Rw → Ar	Rw → Cl	Rw → Pr	Avg
ResNet	59.37	76.58	87.48	69.86	71.11	81.66	73.72	56.30	86.07	78.68	59.22	78.59	73.22
DANN	56.17	81.72	86.87	68.67	73.38	83.76	69.92	56.84	85.80	79.41	57.26	78.26	73.17
RTN	50.46	77.80	86.90	65.12	73.40	85.07	67.86	45.23	85.50	79.20	55.55	78.79	70.91
IWAN	52.55	81.40	86.51	70.58	70.99	85.29	74.88	57.33	85.07	77.48	59.65	78.91	73.39
PADA	39.58	69.37	76.26	62.57	67.39	77.47	48.39	35.79	79.60	75.94	44.50	78.10	62.91
ATI	52.90	80.37	85.91	71.08	72.41	84.39	74.28	57.84	85.61	76.06	60.17	78.42	73.29
OSBP	47.75	60.90	76.78	59.23	61.58	74.33	61.67	44.50	79.31	70.59	54.95	75.18	63.90
UAN w/o d	61.60	81.86	87.67	74.52	73.59	84.88	73.65	57.37	86.61	81.58	62.15	79.14	75.39
UAN w/o y	56.63	77.51	87.61	71.96	69.08	83.18	71.40	56.10	84.24	79.27	60.59	78.35	72.91
UAN	**63.00**	**82.83**	**87.85**	**76.88**	**78.70**	**85.36**	**78.22**	**58.59**	**86.80**	**83.37**	**63.17**	**79.43**	**77.02**

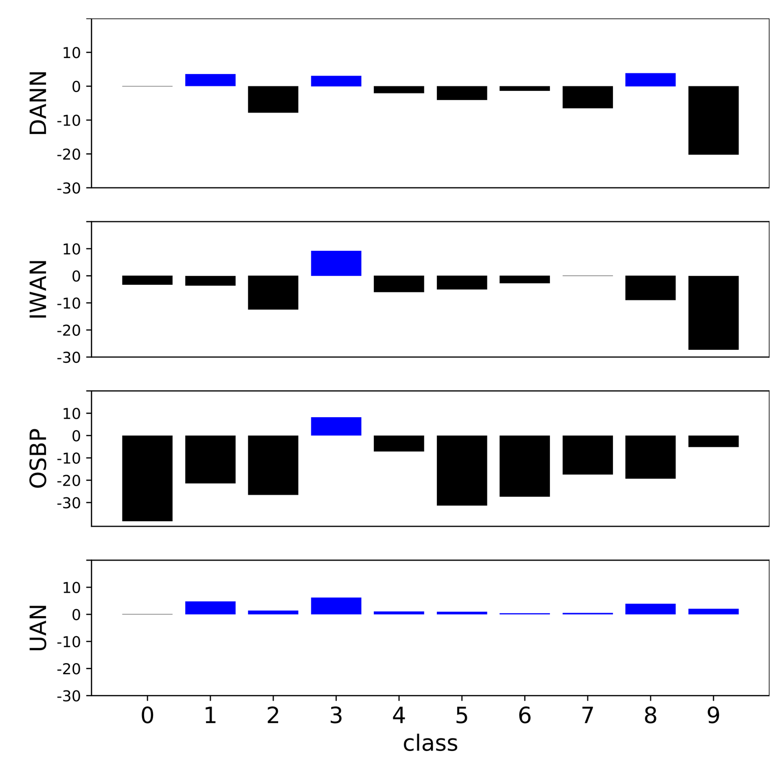
Table: Results on **Office-31**($\xi = 0.32$), **ImageNet-Caltech**($\xi = 0.07$), **VisDA2017**($\xi = 0.50$)

Method	Office-31							ImageNet-Caltech		VisDA
	A → W	D → W	W → D	A → D	D → A	W → A	Avg	I → C	C → I	
ResNet	75.94	89.60	90.91	80.45	78.83	81.42	82.86	70.28	65.14	52.80
DANN	80.65	80.94	88.07	82.67	74.82	83.54	81.78	71.37	66.54	52.94
RTN	85.70	87.80	88.91	82.69	74.64	83.26	84.18	71.94	66.15	53.92
IWAN	85.25	90.09	90.00	84.27	84.22	86.25	86.68	72.19	66.48	58.72
PADA	85.37	79.26	90.91	81.68	55.32	82.61	79.19	65.47	58.73	44.98
ATI	79.38	92.60	90.08	84.40	78.85	81.57	84.48	71.59	67.36	54.81
OSBP	66.13	73.57	85.62	72.92	47.35	60.48	67.68	62.08	55.48	30.26
UAN	**85.62**	**94.77**	**97.99**	**86.50**	**85.45**	**85.12**	**89.24**	**75.28**	**70.17**	**60.83**

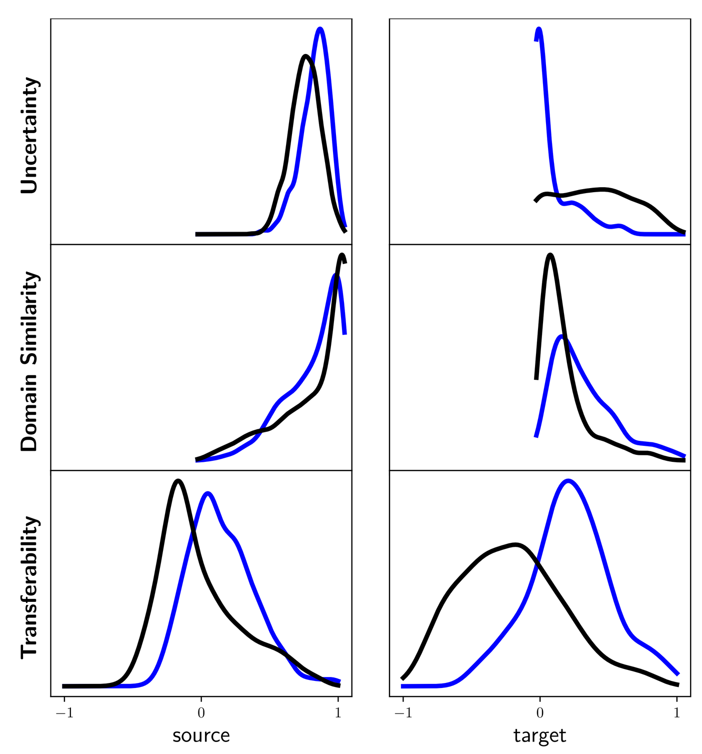
- Analysis
  - Existing methods perform well when their assumptions hold but worse when violated
  - UAN works stably without prior knowledge on the relationship of label sets
  - Mainstream methods suffer from negative transfer in most classes
  - Only UAN promotes positive transfer for all classes
  - Quality of the transferability criterion can be verified



(a) Accuracy w.r.t. $|\overline{\mathcal{C}}_t|$ (b) Accuracy w.r.t. $|\mathcal{C}|$ (c) Sensitivity to w_0



(d) Negative Transfer in UDA



(e) Hypotheses Quality (blue for *common* and black for *private*)

School of Software - Tsinghua University - China National Engineering Lab for Big Data Software University of California, Berkeley Mail: youkaichao@gmail.com