Deep Cocktail Network: Multi-source Unsupervised Domain Adaptation with Category Shift

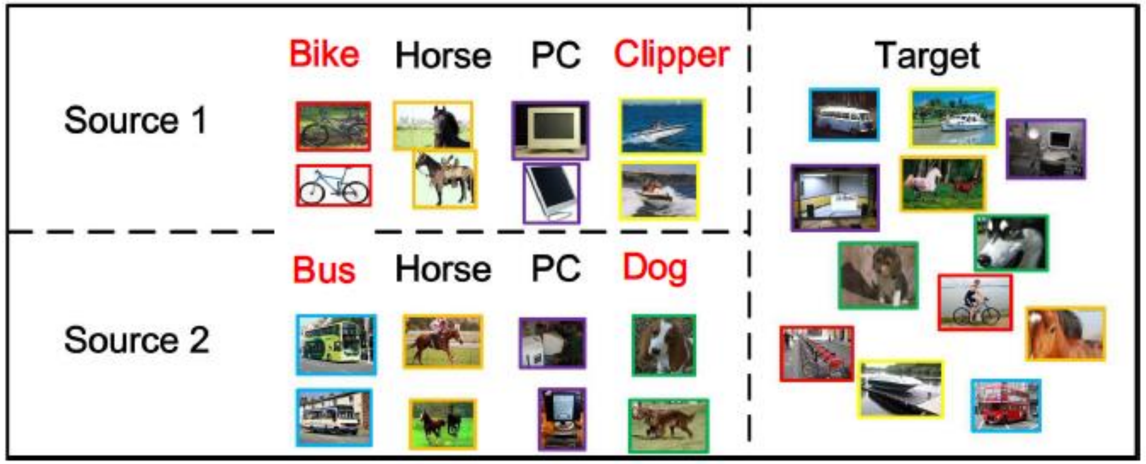
Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, Liang Lin;
The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3964-3973

Paper Reading

Jingge Wang

2020/5/15

Overview



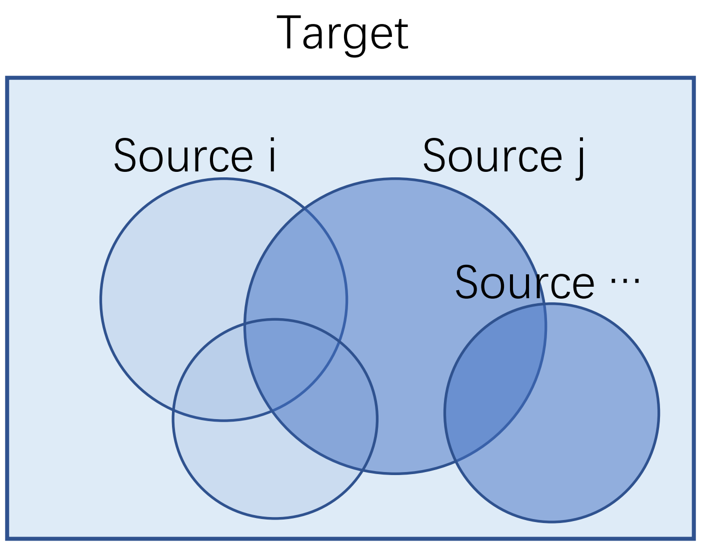
(b) Multi-source domain adaptation with category shift

- Motivation
  - Multi source domains
    - Domain shift
    - Category shift
- Challenges
  - eliminate the distribution discrepancy between target and each source maybe too strict, and harmful
  - cannot simply apply same UDA via combining all source domains since there are possible domain shifts among sources
  - category shift in sources
- DCTN (deep cocktail network)
  - multi-way adversarial learning
    - minimize the discrepancy
  - Weighted source predictions
    - source-specific perplexity scores

Settings

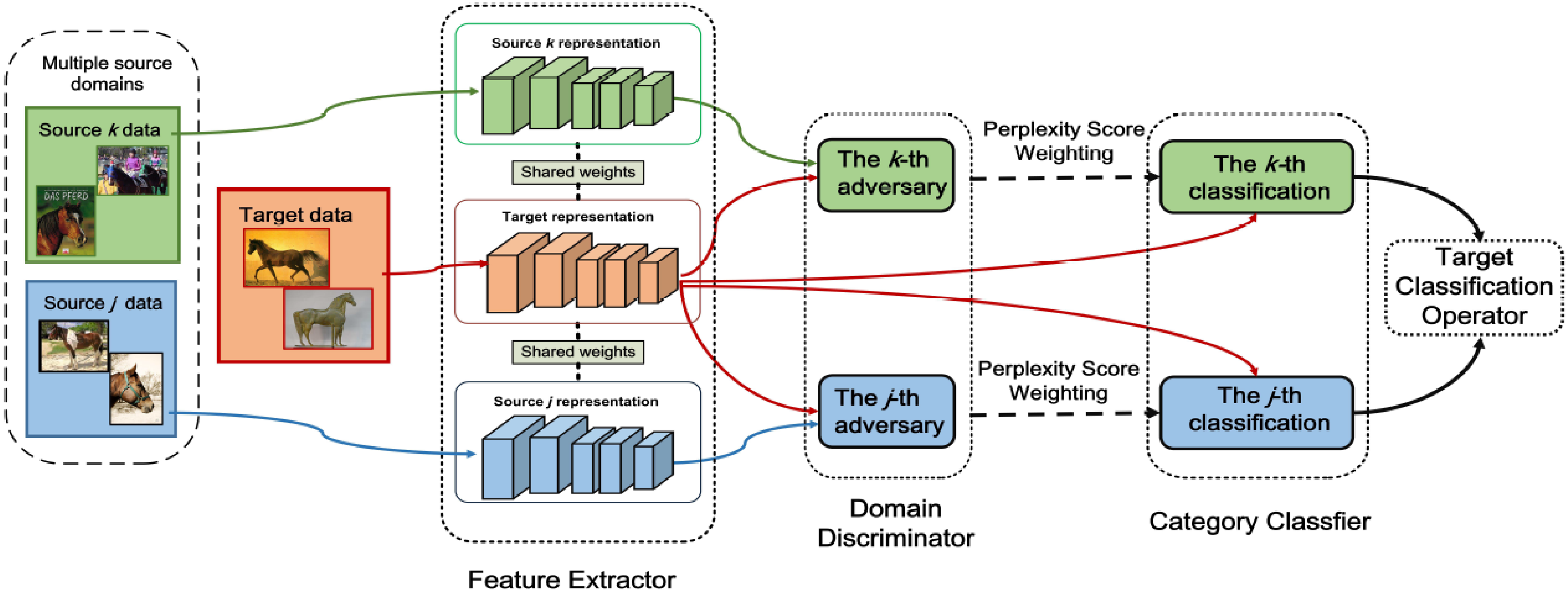
- Vanilla MDA
  - N different underlying source distributions $\{p_{s_j}(x, y)\}_{j=1}^{N}$
    - $X_{s_j} = \{x_i^{s_j}\}_{i=1}^{|X_{s_j}|}$
    - $Y_{s_j} = \{y_i^{s_j}\}_{i=1}^{|Y_{s_j}|}$
  - 1 target distribution $p_t(x, y)$, no label
    - $X_t = \{x_i^t\}_{i=1}^{|X_t|}$
- Category Shift

$$\mathcal{C}_t = \bigcup_{j=1}^{N} \mathcal{C}_{s_j}$$



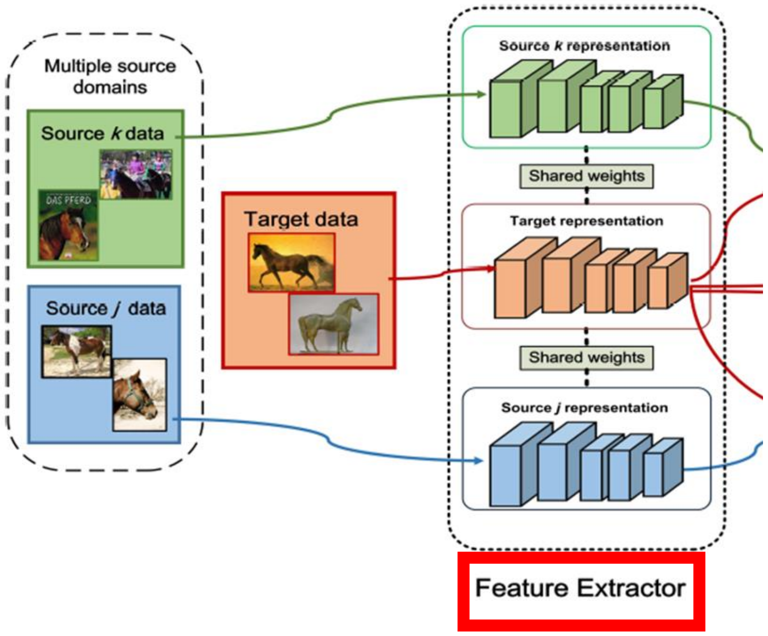
DCTN(deep cocktail network)

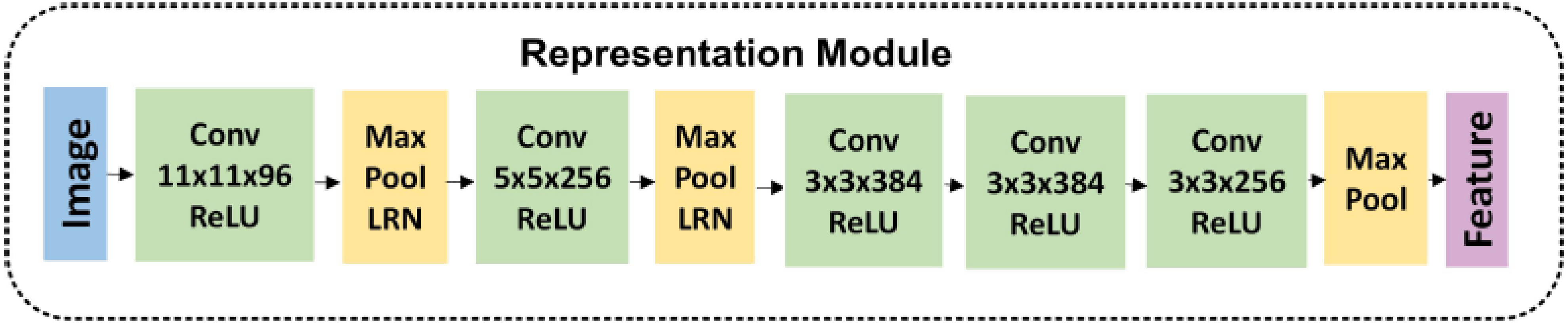
- Architecture
  - Feature extractor F
  - Domain discriminator D
  - Category classifier C
  - Target classification operator



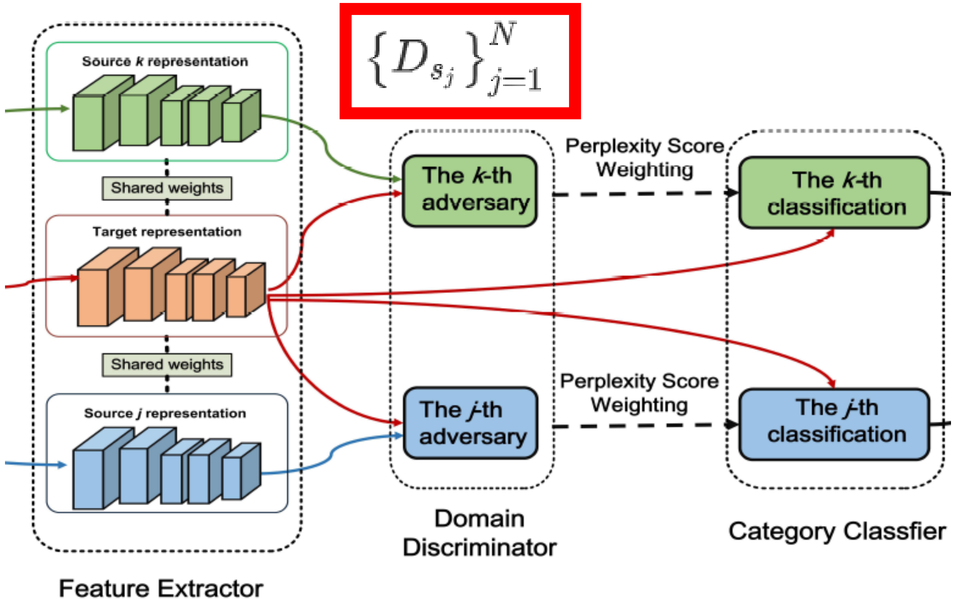
DCTN(deep cocktail network)

- **Feature extractor F**
  - Fully convolutional network
  - map all N sources and target into a common feature space
  - employ adversarial learning to obtain the optimal mapping
  - domain-invariant features
  - related to each source domain





DCTN(deep cocktail network)

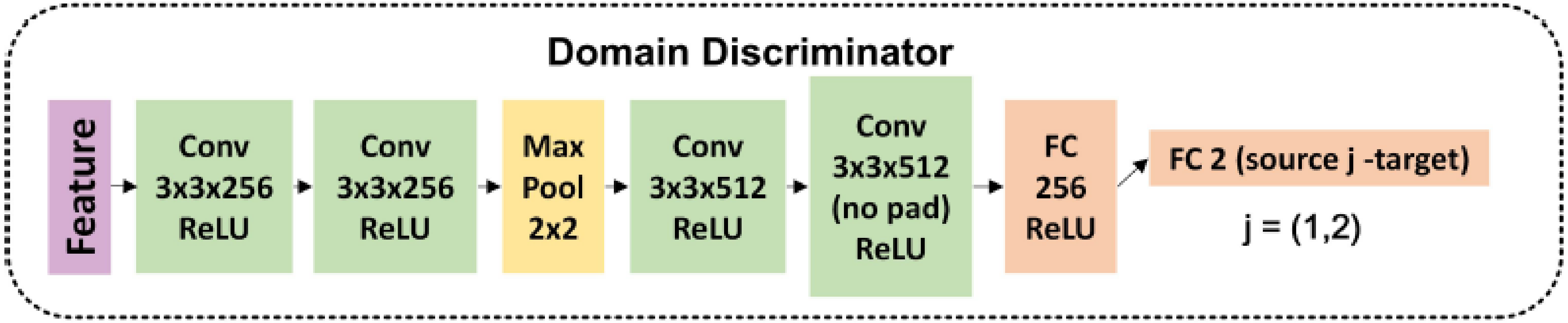


- (Multi-source) **domain discriminators** $\{D_{s_j}\}_{j=1}^{N}$
  - two-output classifier
  - N source-specific discriminative results:

$$\{D_{s_j}(F(x^t))\}_{j=1}^{N}$$

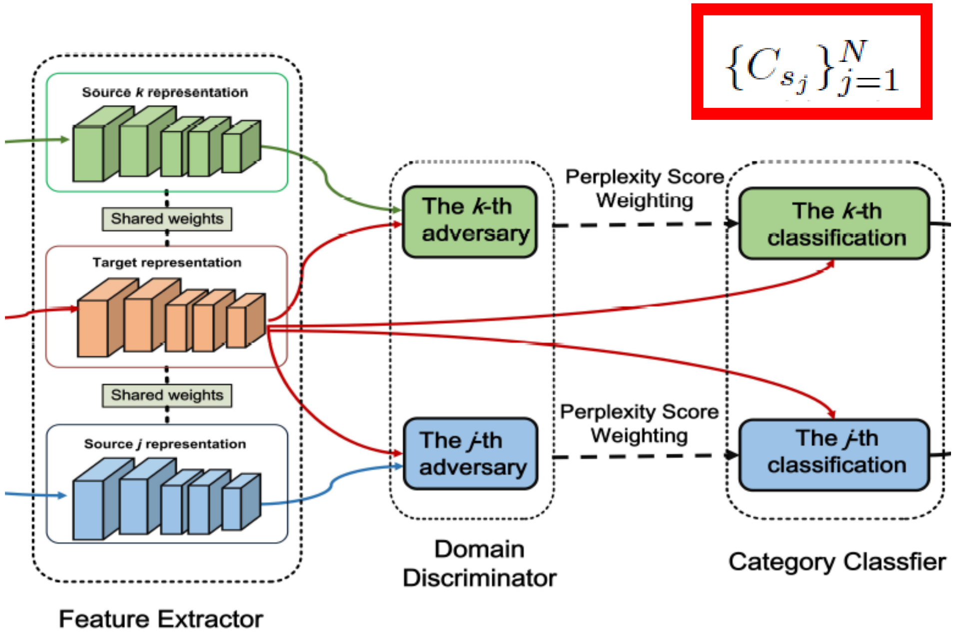
  - target-source perplexity scores for x^t

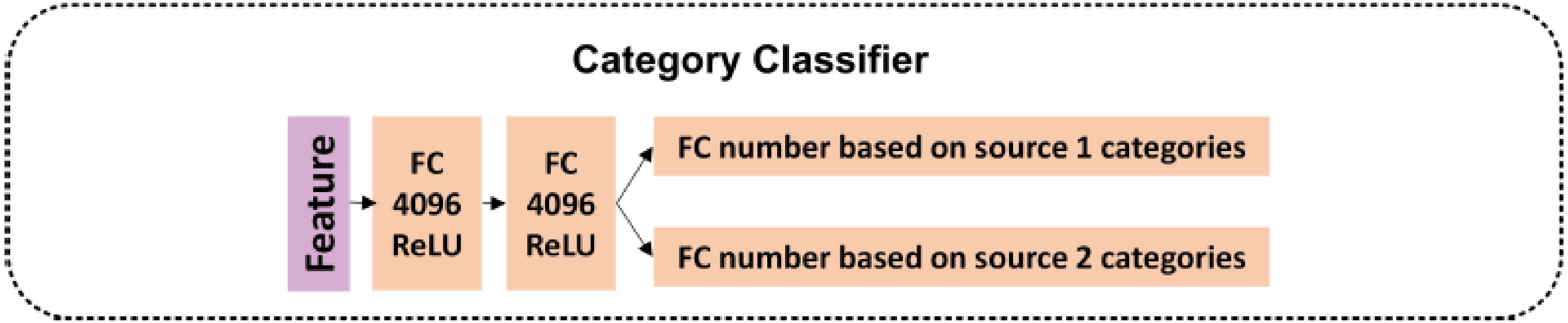
$$\mathcal{S}_{cf}(x^t; F, D_{s_j}) = -\log(1 - D_{s_j}(F(x^t))) + \alpha_{s_j} \qquad (1)$$



DCTN(deep cocktail network)

- (Multi-source) **category classifiers** $\{C_{s_j}\}_{j=1}^{N}$





DCTN(deep cocktail network)

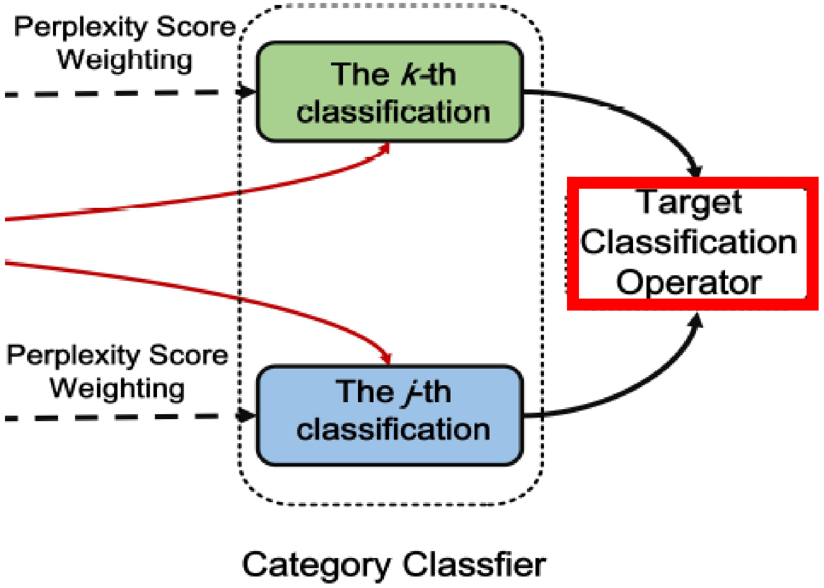
- **Target classification operator**
  - takes each source perplexity score $\mathcal{S}_{cf}(x^t; F, D_{s_j})$

$$\mathcal{S}_{cf}(x^t; F, D_{s_j}) = -\log(1 - D_{s_j}(F(x^t))) + \alpha_{s_j} \qquad (1)$$

  - re-weight each source-specific classification $\{C_{s_j}\}_{j=1}^{N}$

$$Confidence\left(c|x^t\right) := \sum_{c \in \mathcal{C}_{s_j}} \frac{\mathcal{S}_{cf}\left(x^t; F, D_{s_j}\right)}{\sum_{c \in \mathcal{C}_{s_k}} \mathcal{S}_{cf}\left(x^t; F, D_{s_k}\right)} C_{s_j}\left(c|F\left(x^t\right)\right) \qquad (2)$$

$$where\ c \in \bigcup_{j=1}^{N} \mathcal{C}_{s_j}$$



Learning

- Pre-training feature extractor F and category classifier C
  - jointly train F and C
    - all source images
  - perplexity scores
    - uniform distribution simplex weight
  - predict pseudo label for target
    - categories with high confidence
  - obtain the pre-trained F and C
    - further fine-tuning with sources and the pseudo-labeled target

Learning

- Alternative adaptation pipeline
  - Multi-way Adversarial Adaptation

$$\min_{F}\max_{D} V(F, D; \overline{C}) = \mathcal{L}_{adv}(F, D) + \mathcal{L}_{cls}(F, \overline{C}) \qquad (4)$$

  - Target Discriminative Adaptation

$$\min_{F,\, C} \mathcal{L}_{cls}(F, C) = \sum_{j}^{N} \mathbb{E}_{(x,y)\sim(X_{s_j}, Y_{s_j})}[\mathcal{L}(C_{s_j}(F(x)), y)] + \mathbb{E}_{(x^t,\hat{y})\sim(X_t^P, Y_t^P)}[\sum_{\hat{y}\in\mathcal{C}_{\hat{s}}} \mathcal{L}(C_{\hat{s}}(F(x^t)), \hat{y})] \qquad (8)$$

Algorithm 2 Learning algorithm for DCTN

Input: N source labeled datasets $\{X_{s_j}, Y_{s_j}\}_{j=1}^{N}$; target unlabeled dataset X_t; initiated feature extractor F; category classifier C and domain discriminator D; confidence threshold γ; adversarial iteration threshold β.

Output: well-trained feature extractor F^*, domain discriminator D^* and category classifier C^*.

```
Pre-train C and F
while not converged do
    Multi-way Adversarial Adaptation:
    for 1:β do
        Sample mini-batch from {X_{s_j}}_{j=1}^N and X_t;
        Update D by Eq.4;
        Update F by Algorithm.1;sequentially
    end for
    Target Discriminative Adaptation:
    Estimate confidence for X_t by Eq.2 with perplexity scores
      offered by Eq.1. Samples X_t^P ⊂ X_t with confidence larger
      than γ get annotations Y_T^P;
    Update F and C by Eq.8.
end while
return F* = F; C* = C; D* = D.
```

Learning

- Online hard domain batch mining
  - batch M
  - compute D_{s_j} ' s loss

$$\sum_{i}^{M} -\log D_{s_j}(F(x_i^{s_j})) - \log(1 - D_{s_j}(F(x_i^t)))$$

  - find hard source domain j^*

$$\mathcal{L}_{cf}\left(x; F, D_{s_j}\right) = \frac{1}{2}\log D_{s_j}(F(x)) + \frac{1}{2}\log(1 - D_{s_j}(F(x))) \quad (7)$$

$$\mathcal{L}_{adv}(F, D) = \frac{1}{N}\sum_j^N \mathbb{E}_{x\sim X_{s_j}}\mathcal{L}_{cf}\left(x; F, D_{s_j}\right) + \mathbb{E}_{x\sim X_t}\mathcal{L}_{cf}\left(x; F, D_{s_j}\right)$$

$$\min_F \max_D V(F, D; \overline{C}) = \mathcal{L}_{adv}(F, D) + \mathcal{L}_{cls}(F, \overline{C}) \quad (4)$$

$$\mathcal{L}_{adv}^{s_j^*}(F, D) = \sum_i^M \mathcal{L}_{cf}\left(x_i^{s_j^*}; F, D_{s_j^*}\right) + \mathcal{L}_{cf}\left(x_i^t; F, D_{s_j^*}\right)$$

$$\min_F \max_D V(F, D; \bar{C}) = \mathcal{L}_{adv}^{s_j^*}(F, D) + \mathcal{L}_{cls}(F, \bar{C})$$

Algorithm 1 Mini-batch Learning via online hard domain batch mining

Input: Mini-batch $\{x_i^t, \{x_i^{s_j}, y_i^{s_j}\}_{j=1}^N\}_{i=1}^M$ sampled from X_t and $\{(X_{s_j}, Y_{s_j})\}_{j=1}^N$ respectively; feature extractor F; domain discriminator D; category classifier $\overline{C}$.

Output: Updated F'.

1: Select the source domain $j^* \in [N]$, where $j^* = \arg\max_j^N\{\sum_i^M -\log D_{s_j}(F(x_i^{s_j})) - \log(1 - D_{s_j}(F(x_i^t)))\}_{j=1}^N$;

2: $\mathcal{L}_{adv}^{s_j*} = \sum_i^M \mathcal{L}_{cf}(x_i^{s_j*}; F, D_{s_{j^*}}) + \mathcal{L}_{cf}(x_i^t; F, D_{s_{j^*}})$

3: Replace $\mathcal{L}_{adv}$ in Eq.4 with $\mathcal{L}_{adv}^{s_j*}$, update F by Eq.4.

4: **return** $F' = F$.

Experiments

- Datasets
  - Office-31
    - 3 domains: A (Amazon), D (DSLR), W (Webcam).
    - 31 categories
  - ImageCLEF-DA
    - 3 domains: I (ImageNet ILSVRC 2012), P (Pascal VOC 2012), C (Caltech-256)
    - 12 categories
  - Digits-five
    - 5 digit domains: mt (MNIST) [26], mm (MNIST-M) [11], sv(SVHN) [36], up (USPS), sy (Synthetic Digits)
    - 10 categories
- Vanilla setting
  - samples from diverse sources share a same category set
- Category shift setting
  - categories from different sources might be also different

Experiments

- Vanilla setting

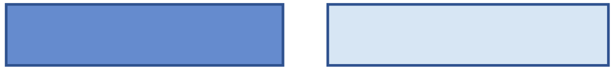
Table 1. Classification accuracy (%) on Office-31 dataset for MDA in the vanilla setting.

Standards	Models	A,W→D	A,D→W	D,W→A	Avg
Single best	TCA	95.2	93.2	51.6	68.8
	GFK	95.0	95.6	52.4	68.7
	DDC	98.5	95.0	52.2	70.7
	DRCN	99.0	96.4	**56.0**	73.6
	RevGrad	99.2	96.4	53.4	74.3
	DAN	99.0	96.0	54.0	72.9
	RTN	**99.6**	96.8	51.0	73.7
Source combine	Source only	98.1	93.2	50.2	80.5
	RevGrad	98.8	96.2	54.6	83.2
	DAN	98.8	95.2	53.4	82.5
Multi-source	Source only	98.2	92.7	51.6	80.8
	sFRAME	54.5	52.2	32.1	46.3
	SGF	39.0	52.0	28.0	39.7
	DCTN (ours)	**99.6**	**96.9**	54.9	**83.8**

Experiments

- Category shift setting
  - Overlap

  - Disjoint



- reduces performance drops
- resist negative transfer

Table 4. Evaluations on Office-31 (A,D→ W) for MDA in the category shift setting.

Category Shift	Models	Accuracy	Degraded Accuracy	Transfer Gain
Overlap	Source only	84.4	-8.3	0
	RevGrad	86.3	-7.9	1.9
	DAN	87.8	**-6.4**	3.4
	DCTN(ours)	90.2	-6.7	**5.8**
Disjoint	Source only	78.1	-14.6	0
	RevGrad	78.6	-15.6	0.5
	DAN	75.5	-18.7	-2.6
	DCTN(ours)	82.9	**-14.0**	**4.8**

Table 5. Evaluations on ImageCLEF-DA (I,P→ C) for MDA in the category shift settings.

Category Shift	Models	Accuracy	Degraded Accuracy	Transfer Gain
Overlap	Source only	86.3	-3.0	0
	RevGrad	85.7	-4.5	-0.6
	DAN	85.5	-4.0	-0.8
	DCTN(ours)	88.7	**-1.3**	**2.4**
Disjoint	Source only	81.5	**-7.8**	0
	RevGrad	71.5	-18.7	-10.0
	DAN	71.0	-18.5	-10.5
	DCTN(ours)	82.0	-8.0	**0.5**

Feature visualization

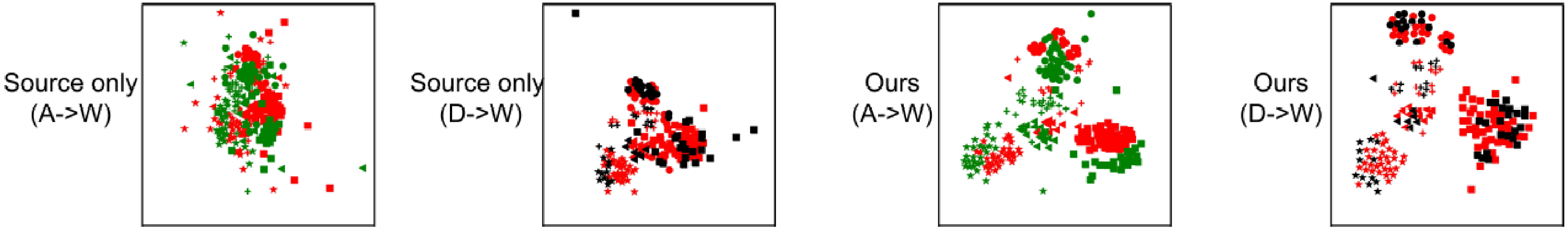


Figure 3. The t-SNE [32] visulization of A,D$\rightarrow$W. Green, black and red represent domains A, D and W respectively. We use different markers to denote 5 categories, e.g., bookcase, calculator, monitor, printer, ruler. Best viewed in color.

Discussion

- Inspired from the distribution weighted combining rule, this paper proposed the deep cocktail network (DCTN) together with the alternating adaptation algorithm to learn transferable and discriminative representation.
- multi-way adversarial learning to minimize the discrepancy between the target and each of the multiple source domains
- Did not show completed comparison for multi-source combination results
- Hard domain batch mining may be not necessary or not enough
- Not consider unknow classes in target