Deep Transfer Learning with Joint Adaptation Networks

Mingsheng Long[1], Han Zhu[1], Jianmin Wang[1]
Michael I. Jordan[2]

[1]School of Software, Institute for Data Science
Tsinghua University

[2]Department of EECS, Department of Statistics
University of California, Berkeley

https://github.com/thuml

International Conference on Machine Learning, 2017

Outline

Deep Learning

Learner: $f : \boldsymbol{x} \rightarrow y$ **Distribution:** $(\boldsymbol{x}, y) \sim P(\boldsymbol{x}, y)$

fish

bird

mammal

tree

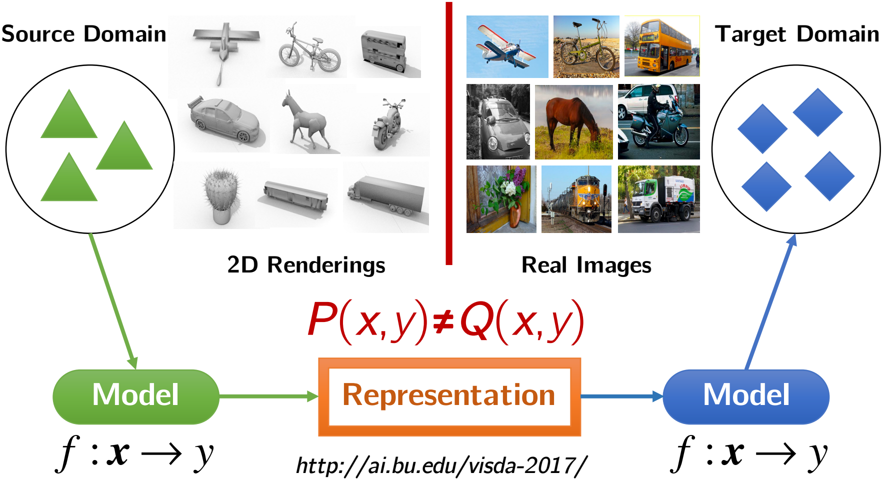
flower

......

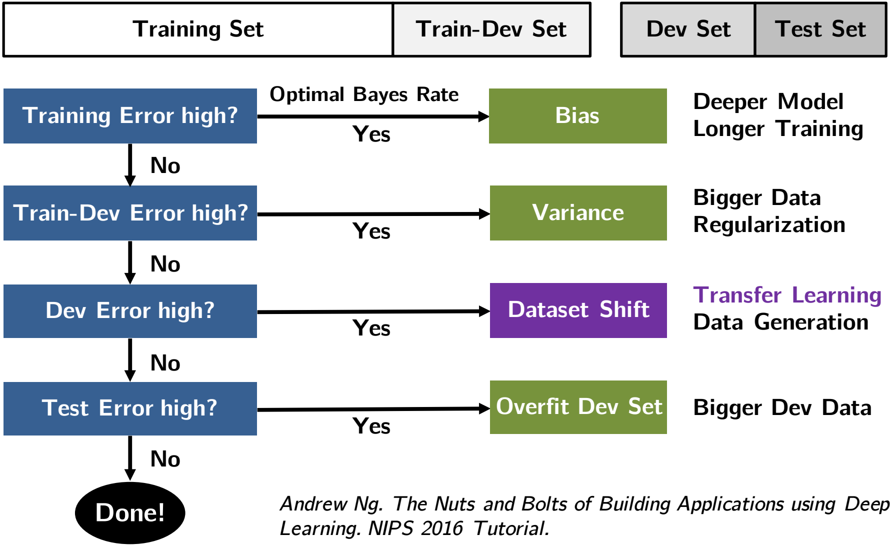
Error Bound: $\epsilon_{\text{test}} \leq \hat{\epsilon}_{\text{train}} + \sqrt{\frac{\text{complexity}}{n}}$

Deep Transfer Learning

- Deep learning across domains of different distributions $P \neq Q$



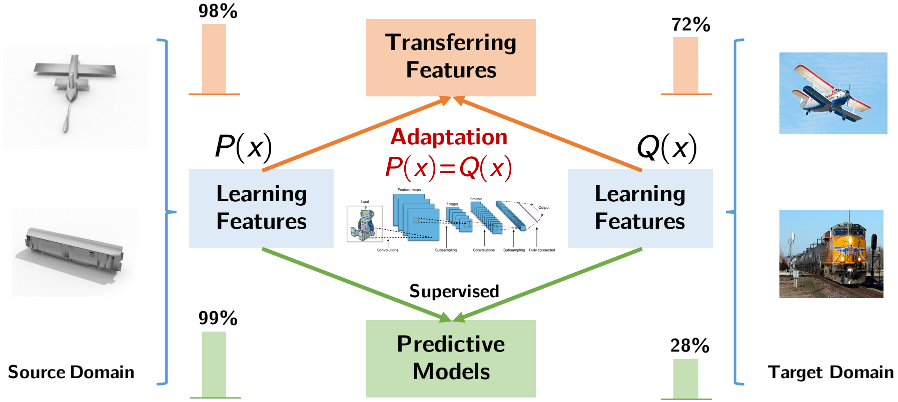
Deep Transfer Learning: Why?



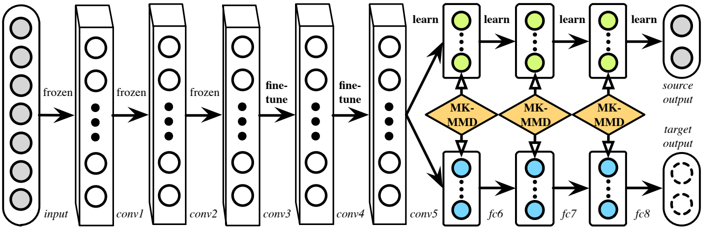
Andrew Ng. The Nuts and Bolts of Building Applications using Deep Learning. NIPS 2016 Tutorial.

Deep Transfer Learning: How?

- Learning predictive models on transferable features s.t. $P(\mathbf{x}) = Q(\mathbf{x})$
- Distribution matching: **MMD** (ICML'15), **GAN** (ICML'15, JMLR'16)



Deep Adaptation Network (DAN)



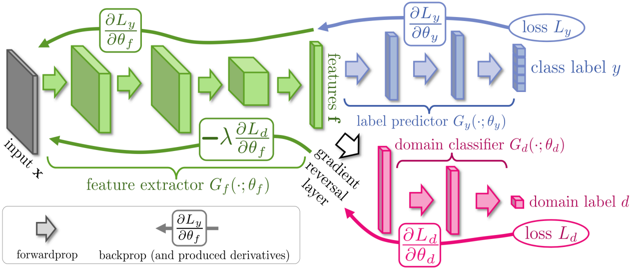
Deep adaptation: match distributions in multiple domain-specific layers

Optimal matching: maximize two-sample test power by multiple kernels

$$d_k^2\left(P, Q\right) \triangleq \left\| \mathbf{E}_P\left[\phi\left(\mathbf{x}^s\right)\right] - \mathbf{E}_Q\left[\phi\left(\mathbf{x}^t\right)\right] \right\|_{\mathcal{H}_k}^2 \tag{1}$$

$$\min_{\theta \in \Theta} \max_{k \in \mathcal{K}} \frac{1}{n_a} \sum_{i=1}^{n_a} J\left(\theta\left(\mathbf{x}_i^a\right), y_i^a\right) + \lambda \sum_{\ell=l_1}^{l_2} d_k^2\left(\mathcal{D}_s^\ell, \mathcal{D}_t^\ell\right) \tag{2}$$

Domain Adversarial Neural Network (DANN)



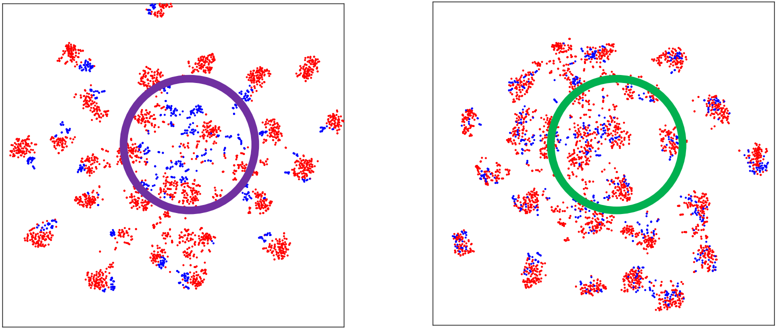
Adversarial adaptation: learning features indistinguishable across domains

$$E(\theta_f, \theta_y, \theta_d) = \sum_{\mathbf{x}_i \in \mathcal{D}_s} L_y(G_y(G_f(\mathbf{x}_i)), y_i) - \lambda \sum_{\mathbf{x}_i \in \mathcal{D}_s \cup \mathcal{D}_t} L_d(G_d(G_f(\mathbf{x}_i)), d_i) \quad (3)$$

$$(\hat{\theta}_f, \hat{\theta}_y) = \arg\min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \theta_d) \quad (\hat{\theta}_d) = \arg\max_{\theta_d} E(\theta_f, \theta_y, \theta_d) \quad (4)$$

Behavior of Existing Work

- Adaptation of marginal distributions $P(\mathbf{x})$ and $Q(\mathbf{x})$ is not sufficient



Before Adaptation

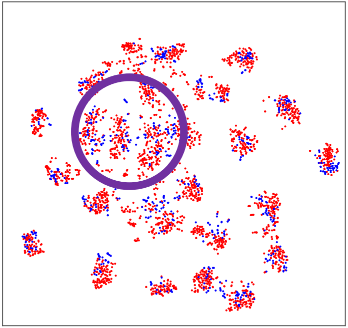
$P(x) \neq Q(x)$

After Adaptation

$P(x) \approx Q(x)$

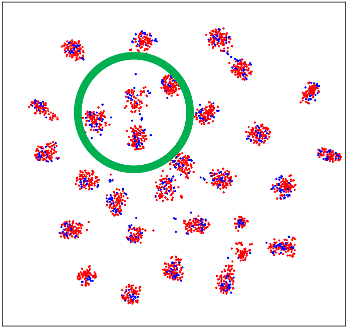
Main Idea of This Work

- **Directly model and match joint distributions $P(\mathbf{x}, y)$ and $Q(\mathbf{x}, y)$**



Match Marginal Distributions

$P(x) \approx Q(x)$



Match Joint Distributions

$P(x,y) \approx Q(x,y)$

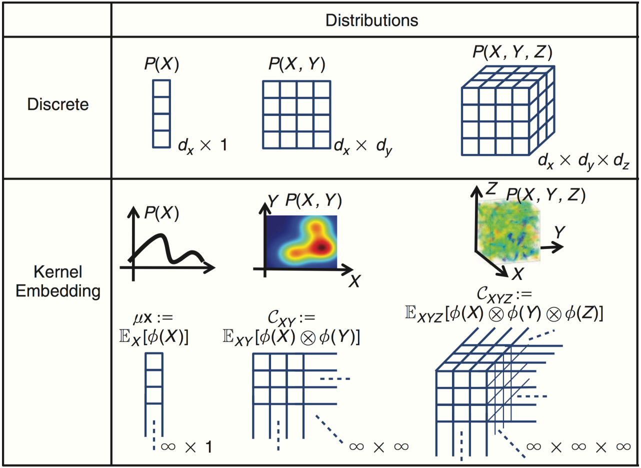
Outline

1. Motivation
   - Deep Transfer Learning
   - Related Work
   - Main Idea

2. **Method**
   - Kernel Embedding
   - JMMD
   - JAN

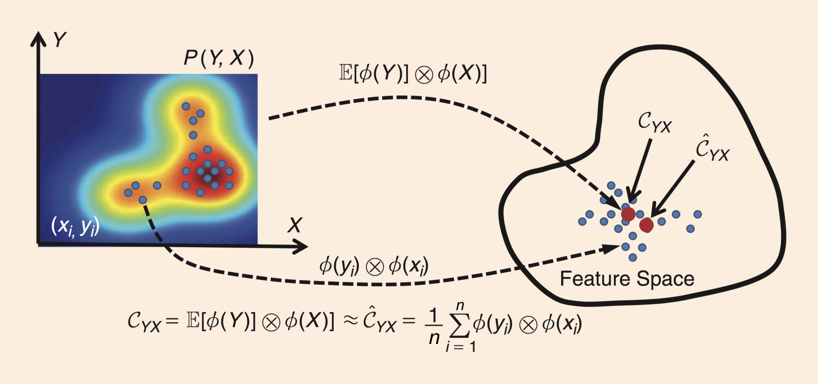
3. Experiments

Kernel Embedding of Distributions



Le Song et al. Kernel Embeddings of Conditional Distributions. IEEE, 2013.

Kernel Embedding of Joint Distributions



$$\mathcal{C}_{\mathbf{X}^{1:m}}(P) \triangleq \mathbb{E}_{\mathbf{X}^{1:m}}\left[\otimes_{\ell=1}^{m} \phi^{\ell}(\mathbf{X}^{\ell})\right] \approx \widehat{\mathcal{C}}_{\mathbf{X}^{1:m}} = \frac{1}{n}\sum_{i=1}^{n} \otimes_{\ell=1}^{m} \phi^{\ell}(\mathbf{x}_i^{\ell}) \tag{5}$$

Le Song et al. Kernel Embeddings of Conditional Distributions. IEEE, 2013.

Joint Maximum Mean Discrepancy (JMMD)

Distance between *embeddings* of $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ and $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$

$$D_{\mathcal{L}}(P, Q) \triangleq \left\| \mathcal{C}_{\mathbf{Z}^{s,1:|\mathcal{L}|}}(P) - \mathcal{C}_{\mathbf{Z}^{t,1:|\mathcal{L}|}}(Q) \right\|^2_{\otimes_{\ell=1}^{|\mathcal{L}|} \mathcal{H}^\ell} . \tag{6}$$

$$\begin{aligned} \widehat{D}_{\mathcal{L}}(P, Q) = & \frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} \prod_{\ell \in \mathcal{L}} k^\ell \left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{s\ell} \right) \\ & + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} \prod_{\ell \in \mathcal{L}} k^\ell \left(\mathbf{z}_i^{t\ell}, \mathbf{z}_j^{t\ell} \right) \\ & - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_t} \prod_{\ell \in \mathcal{L}} k^\ell \left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{t\ell} \right). \end{aligned} \tag{7}$$

Theorem (Two-Sample Test (Gretton et al. 2012))

- $P = Q$ *if and only if* $\widehat{D}_{\mathcal{L}}(P, Q) = 0$ *(In practice,* $\widehat{D}_{\mathcal{L}}(P, Q) < \varepsilon$*)*

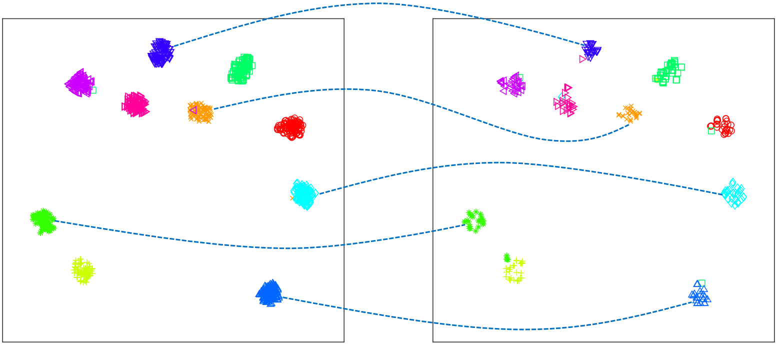
How to Understand JMMD?

- Set last-layer features $\mathbf{Z} = \mathbf{Z}^{L-1}$, classifier predictions $\mathbf{Y} = \mathbf{Z}^{L} \in \mathbb{R}^{C}$
- We can understand JMMD$(\mathbf{Z}, \mathbf{Y})$ by simplifying it to linear kernel
- This interpretation assumes classifier predictions $\mathbf{Y}$ be one-hot vector

$$
\begin{aligned}
\widehat{D}_{\mathcal{L}}(P, Q) &\triangleq \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{z}_i^s \otimes \mathbf{y}_i^s - \frac{1}{n_t} \sum_{j=1}^{n_t} \mathbf{z}_j^t \otimes \mathbf{y}_j^t \right\|^2 \\
&= \sum_{c=1}^{C} \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} y_{i,c}^s \mathbf{z}_i^s - \frac{1}{n_t} \sum_{j=1}^{n_t} y_{j,c}^t \mathbf{z}_j^t \right\|^2 \\
&\approx \sum_{c=1}^{C} \widehat{D}\left(P_{Z|y=c}, Q_{Z|y=c}\right)
\end{aligned}
\tag{8}
$$

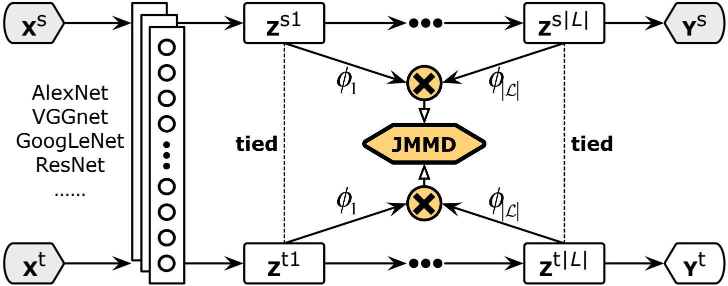
Equivalent to matching distributions P and Q conditioned on each class!

How to Understand JMMD?



- **JMMD can process continuous softmax activations (probability)**
- In practice, Gaussian kernel is used for matching all orders of moments

Joint Adaptation Network (JAN)



Joint adaptation: match joint distributions of multiple task-specific layers

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \widehat{D}_{\mathcal{L}}\left(P, Q\right) \tag{9}$$

$$D_{\mathcal{L}}\left(P, Q\right) \triangleq \left\| \mathcal{C}_{\mathbf{Z}^{s,1:|\mathcal{L}|}}\left(P\right) - \mathcal{C}_{\mathbf{Z}^{t,1:|\mathcal{L}|}}\left(Q\right) \right\|^2_{\otimes_{\ell=1}^{|\mathcal{L}|}\mathcal{H}^\ell} \tag{10}$$

Learning Algorithm

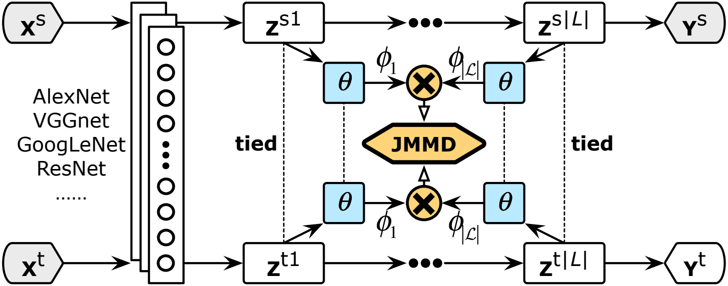
Linear-Time $O(n)$ Algorithm of JMMD (Streaming Algorithm)

$$
\begin{aligned}
\widehat{D}_{\mathcal{L}}(P, Q) = & \frac{2}{n} \sum_{i=1}^{n/2} \left(\prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}) + \prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}) \right) \\
& - \frac{2}{n} \sum_{i=1}^{n/2} \left(\prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{t\ell}) + \prod_{\ell \in \mathcal{L}} k^{\ell}(\mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{s\ell}) \right) \\
= & \frac{2}{n} \sum_{i=1}^{n/2} d\left(\{\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\}_{\ell \in \mathcal{L}} \right)
\end{aligned}
\tag{11}
$$

SGD: for each layer ℓ and for each quad-tuple $\left(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\right)$

$$
\nabla_{W^{\ell}} = \frac{\partial J\left(\mathbf{z}_{2i-1}^{s}, \mathbf{z}_{2i}^{s}, y_{2i-1}^{s}, y_{2i}^{s}\right)}{\partial W^{\ell}} + \lambda \frac{\partial d\left(\{\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\}_{\ell \in \mathcal{L}}\right)}{\partial W^{\ell}} \tag{12}
$$

Adversarial Joint Adaptation Network (JAN-A)



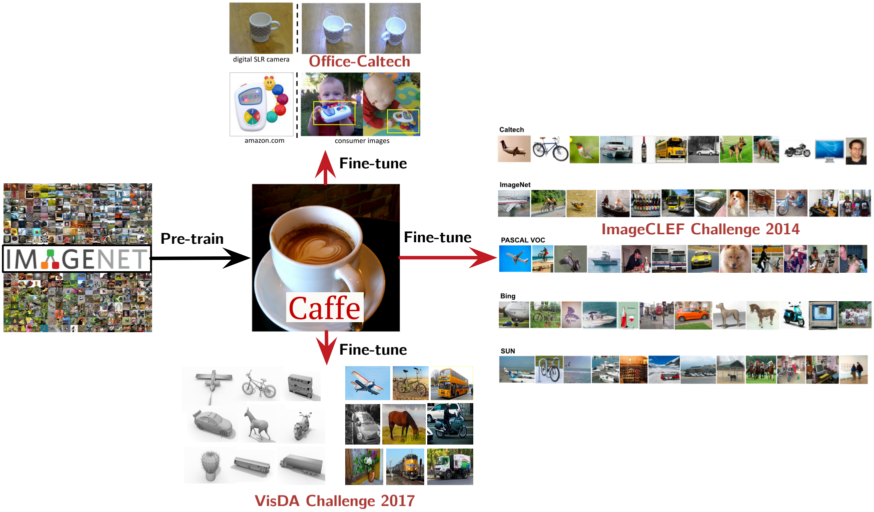
Optimal matching: maximize JMMD as semi-parametric domain adversary

$$\min_f \max_\theta \frac{1}{n_s} \sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda \widehat{D}_{\mathcal{L}}\left(P, Q; \theta\right) \tag{13}$$

$$\widehat{D}_{\mathcal{L}}\left(P, Q; \theta\right) = \frac{2}{n} \sum_{i=1}^{n/2} d\left(\left\{\theta^\ell\left(\mathbf{z}_{2i-1}^{s\ell}, \mathbf{z}_{2i}^{s\ell}, \mathbf{z}_{2i-1}^{t\ell}, \mathbf{z}_{2i}^{t\ell}\right)\right\}_{\ell \in \mathcal{L}}\right) \tag{14}$$

Outline

Datasets

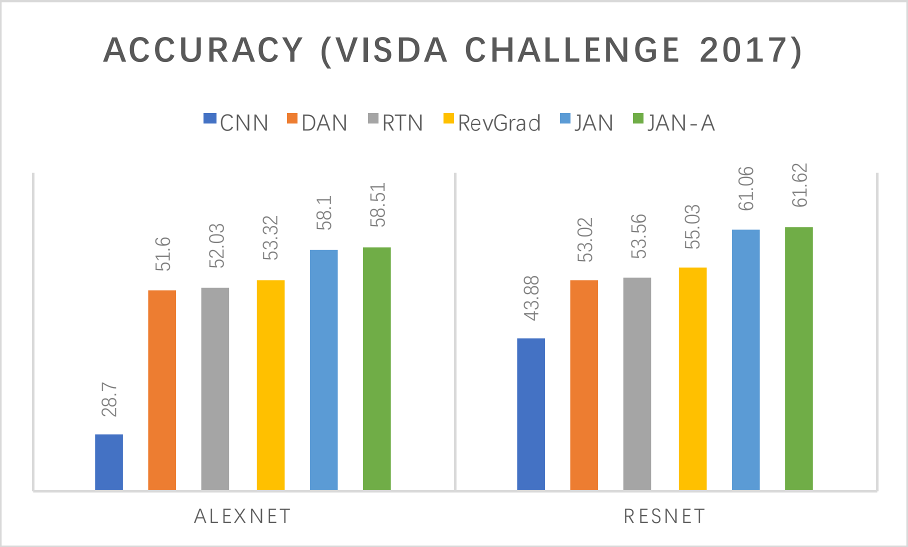

digital SLR camera
Office-Caltech
amazon.com
consumer images
Fine-tune
IMAGENET
Pre-train
Caffe
Fine-tune
Fine-tune
VisDA Challenge 2017
Caltech
ImageNet
ImageCLEF Challenge 2014
PASCAL VOC
Bing
SUN

Results

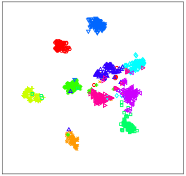
Learning transferable features with joint adaptation and optimal matching

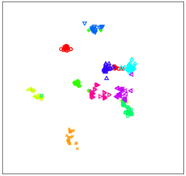
Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet	61.6±0.5	95.4±0.3	99.0±0.2	63.8±0.5	51.1±0.6	49.8±0.4	70.1
TCA	61.0±0.0	93.2±0.0	95.2±0.0	60.8±0.0	51.6±0.0	50.9±0.0	68.8
GFK	60.4±0.0	95.6±0.0	95.0±0.0	60.6±0.0	52.4±0.0	48.1±0.0	68.7
DDC	61.8±0.4	95.0±0.5	98.5±0.4	64.4±0.3	52.1±0.6	52.2±0.4	70.6
DAN	68.5±0.5	96.0±0.3	99.0±0.3	67.0±0.4	54.0±0.5	53.1±0.5	72.9
RTN	73.3±0.3	**96.8**±0.2	**99.6**±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
RevGrad	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	53.4±0.4	51.2±0.5	74.3
JAN	74.9±0.3	96.6±0.2	99.5±0.2	71.8±0.2	**58.3**±0.3	55.0±0.4	76.0
JAN-A	**75.2**±0.4	96.6±0.2	**99.6**±0.1	**72.8**±0.3	57.5±0.2	**56.3**±0.2	**76.3**
ResNet	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
TCA	72.7±0.0	96.7±0.0	99.6±0.0	74.1±0.0	61.7±0.0	60.9±0.0	77.6
GFK	72.8±0.0	95.0±0.0	98.2±0.0	74.5±0.0	63.4±0.0	61.0±0.0	77.5
DDC	75.6±0.2	96.0±0.2	98.2±0.1	76.5±0.3	62.2±0.4	61.5±0.5	78.3
DAN	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
RTN	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
RevGrad	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
JAN	85.4±0.3	**97.4**±0.2	**99.8**±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
JAN-A	**86.0**±0.4	96.7±0.3	99.7±0.1	**85.1**±0.4	**69.2**±0.4	**70.7**±0.5	**84.6**

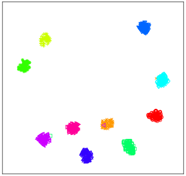
Results

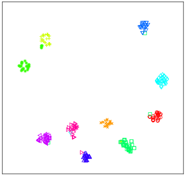

ACCURACY (VISDA CHALLENGE 2017)
CNN
DAN
RTN
RevGrad
JAN
JAN-A
28.7
51.6
52.03
53.32
58.1
58.51
43.88
53.02
53.56
55.03
61.06
61.62
ALEXNET
RESNET

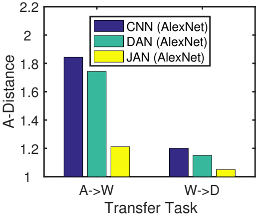
Analysis


(a) DAN: A

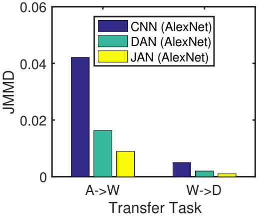

(b) DAN: W


(c) JAN: A

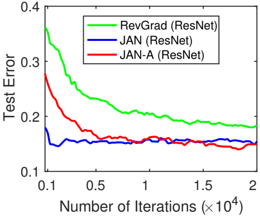

(d) JAN: W



(e) $\mathcal{A}$-distance



(f) JMMD



(g) Convergence

Summary

- A joint adaptation network framework for deep transfer learning
- Two main contributions:
  - Joint adaptation of multilayer features and classifier predictions
  - Adversarial adaptation with semi-parametric domain discriminator
- State-of-the-art results on cross-domain & simulation-to-real datasets

- Open Problems
  - Randomized method for the multilinear operation across feature maps
  - Kernel approximation of the universal kernel for distribution matching

- Code available at: `https://github.com/thuml/transfer-caffe`