Deep Transfer Learning with Joint Adaptation Networks

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

[1]School of Software, Tsinghua University, China [2]Department of EECS, UC Berkeley, USA

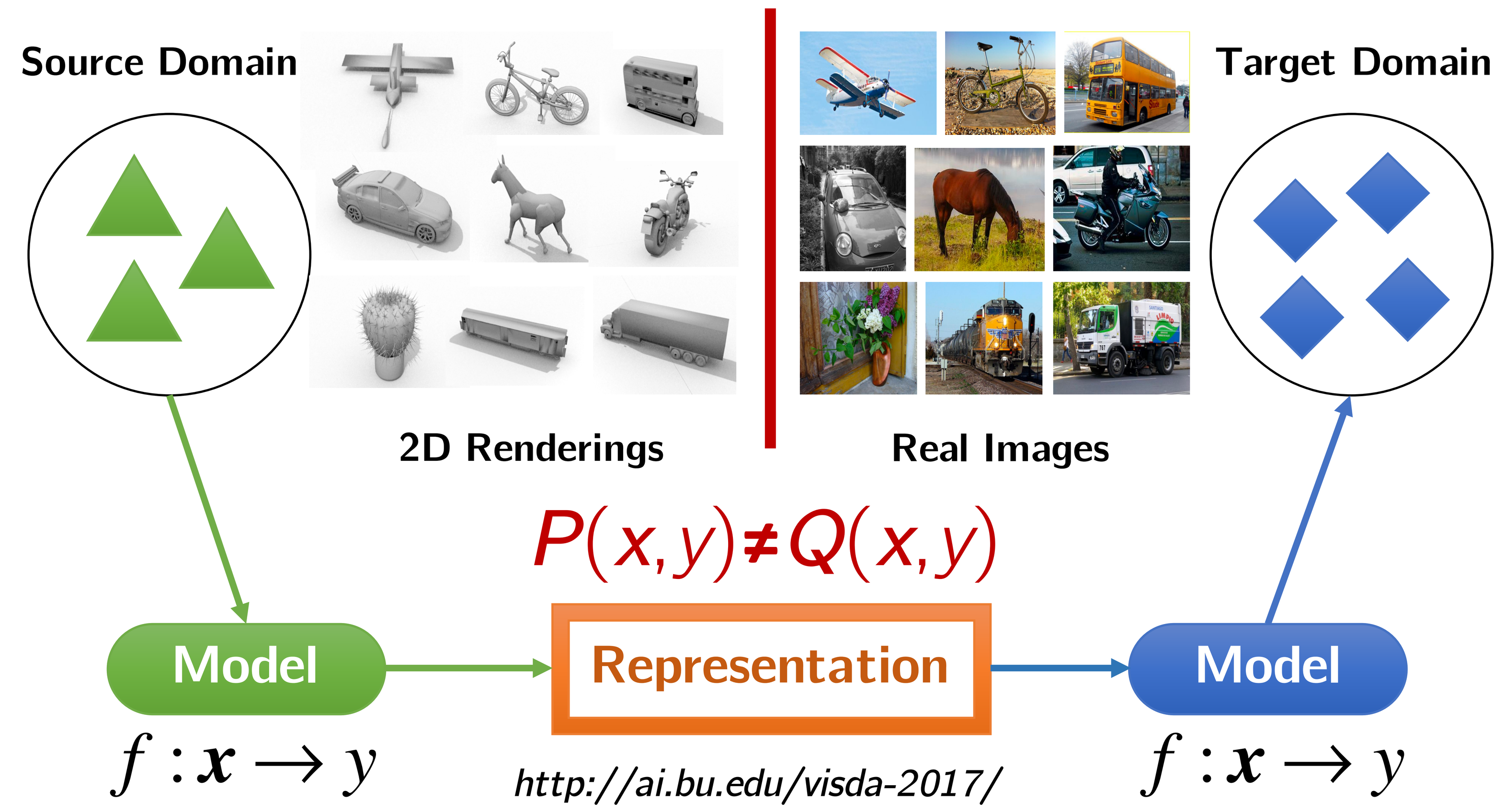
ICML 2017 THE 34TH INTERNATIONAL CONFERENCE ON MACHINE LEARNING SYDNEY, AUSTRALIA AUGUST 6TH - 11TH, 2017

Summary

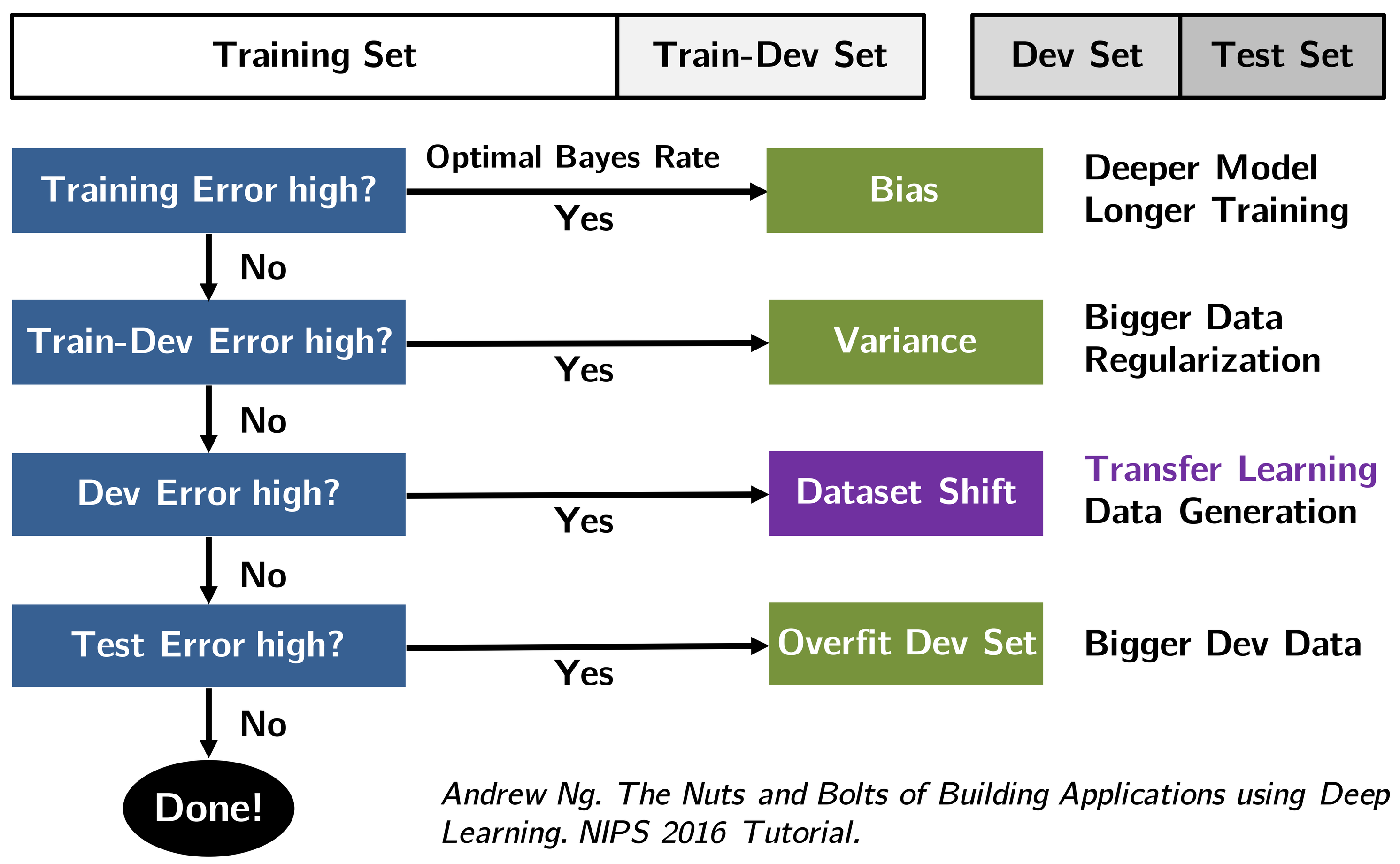
- A joint adaptation network framework for deep transfer learning
- Two main contributions:
  - Joint adaptation of multilayer features and classifier predictions
  - Adversarial adaptation with semi-parametric domain discriminator
- State-of-the-art results on visual & simulation-to-real datasets
- Open Problems
  - Randomized method for the multilinear operation across feature maps
  - Kernel approximation of the universal kernel for distribution matching
- Code@: `https://github.com/thuml/transfer-caffe`

Deep Transfer Learning

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



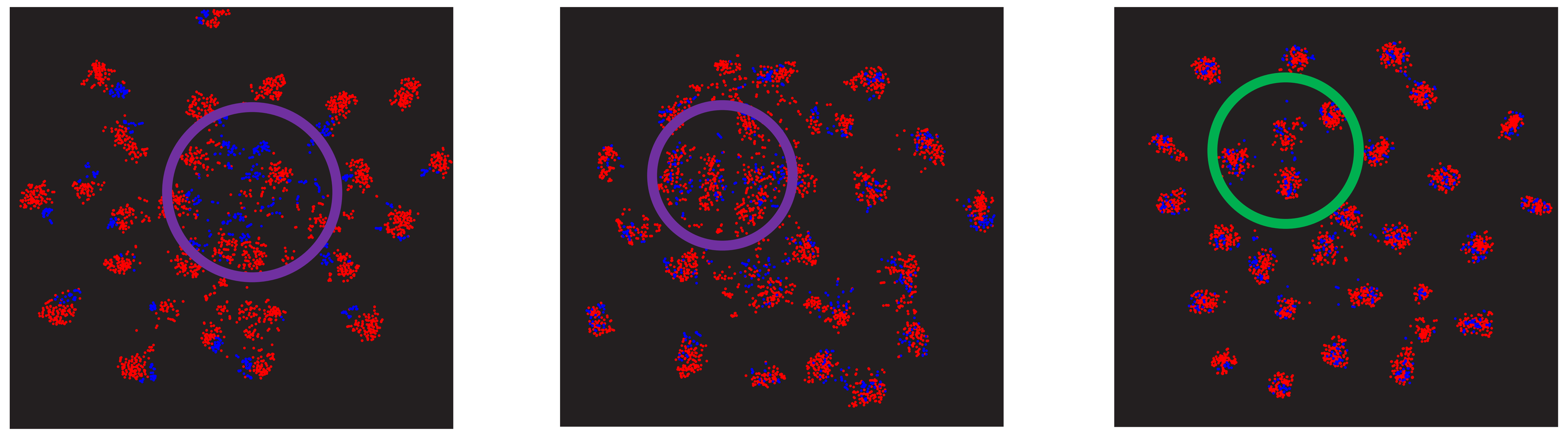
Deep Transfer Learning: Why?



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

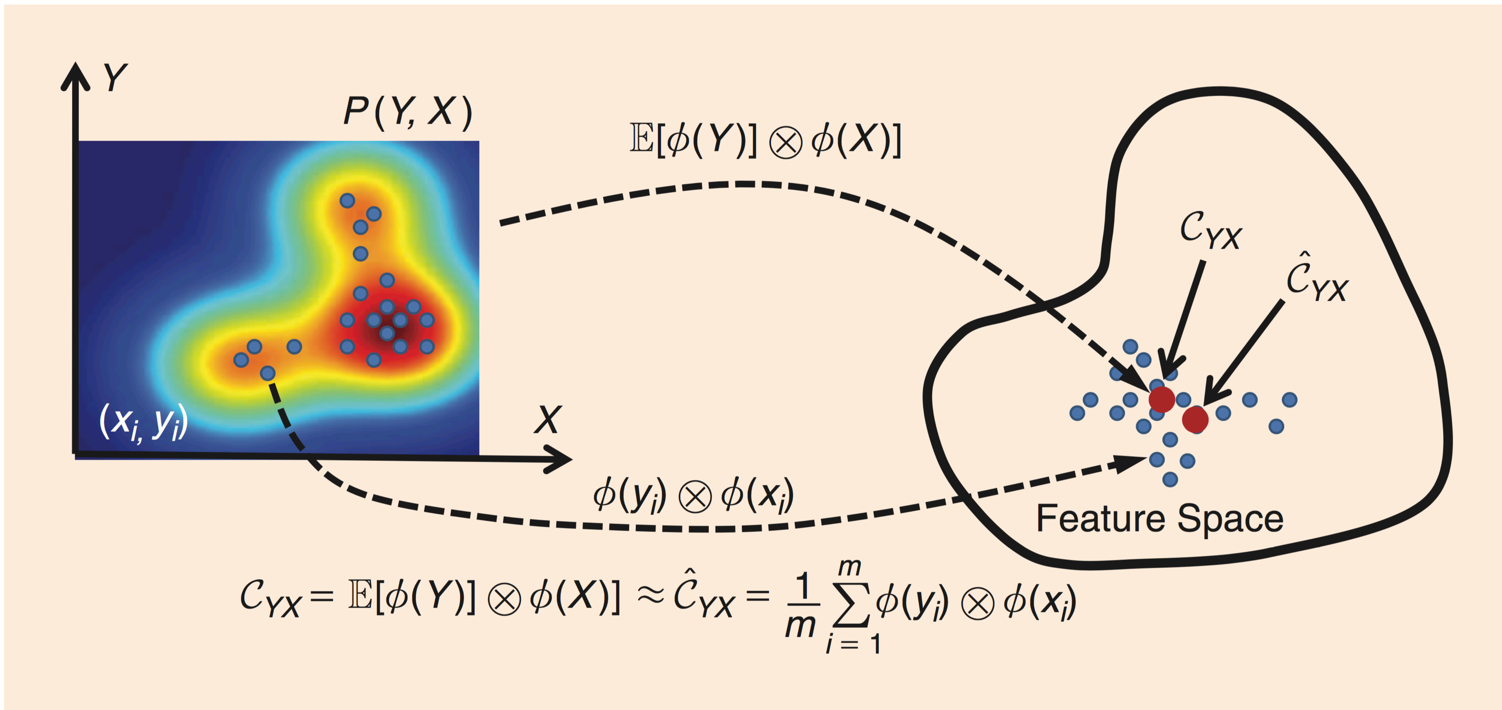
Main Idea of This Work

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



No Adaptation	Match Marginal Distributions	Match Joint Distributions
$P(x) \neq Q(x)$	$P(x) \approx Q(x)$	$P(x,y) \approx Q(x,y)$

Kernel Embedding of Joint Distributions



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

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

Joint Maximum Mean Discrepancy (JMMD)

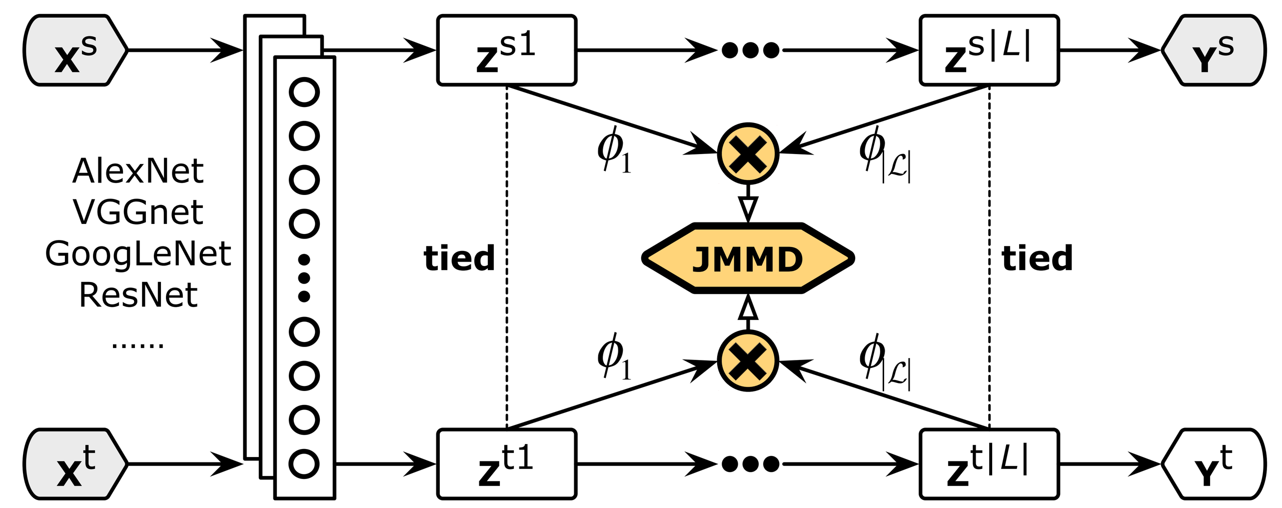
Distance between *embeddings* $P(\mathbf{Z}^{s1}, \ldots, \mathbf{Z}^{s|\mathcal{L}|})$ $Q(\mathbf{Z}^{t1}, \ldots, \mathbf{Z}^{t|\mathcal{L}|})$

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

$$\begin{aligned}\widehat{D}_{\mathcal{L}}(P, Q) = & \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\prod_{\ell\in\mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{s\ell}\right) \\ & + \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\prod_{\ell\in\mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{t\ell}, \mathbf{z}_j^{t\ell}\right) \\ & - \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\prod_{\ell\in\mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{s\ell}, \mathbf{z}_j^{t\ell}\right).\end{aligned} \quad (3)$$

- $P = Q$ iff. $\widehat{D}_{\mathcal{L}}(P, Q) = 0$ (In practice, $\widehat{D}_{\mathcal{L}}(P, Q) < \varepsilon$)

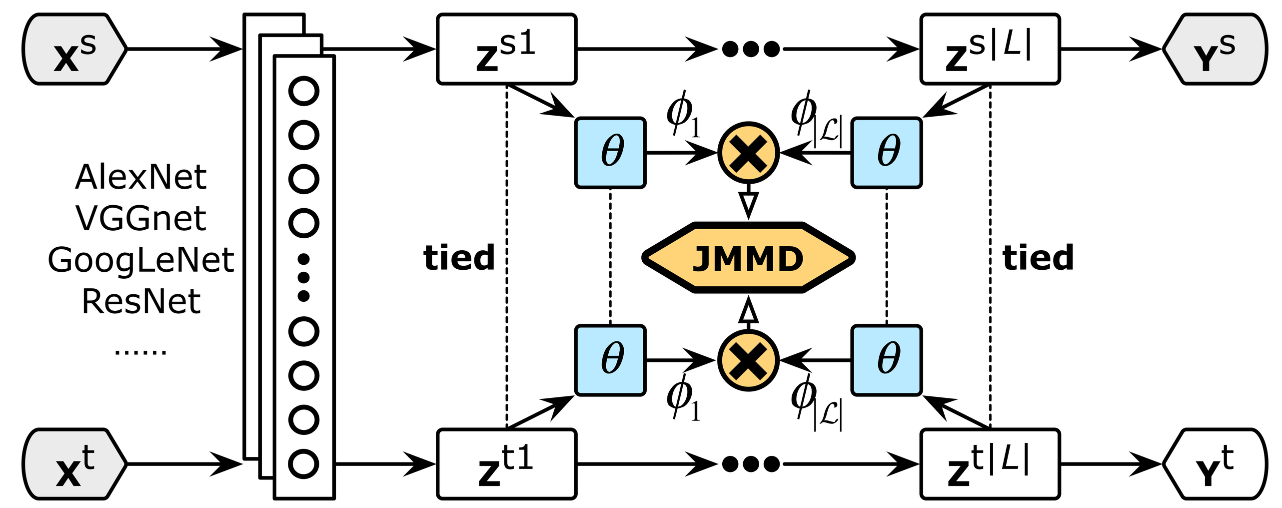
Joint Adaptation Network (JAN)



Joint adaptation: match joint distributions of features/predictions

$$\min_{f} \frac{1}{n_s}\sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda\widehat{D}_{\mathcal{L}}(P, Q) \quad (4)$$

Adversarial Joint Adaptation Network (JAN-A)



Optimal matching: maximize JMMD as semi-parametric adversary

$$\min_{f}\max_{\theta} \frac{1}{n_s}\sum_{i=1}^{n_s} J\left(f\left(\mathbf{x}_i^s\right), \mathbf{y}_i^s\right) + \lambda\widehat{D}_{\mathcal{L}}(P, Q; \theta) \quad (5)$$

Experimental Results

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
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
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**