Walk through Deep Transfer Learning

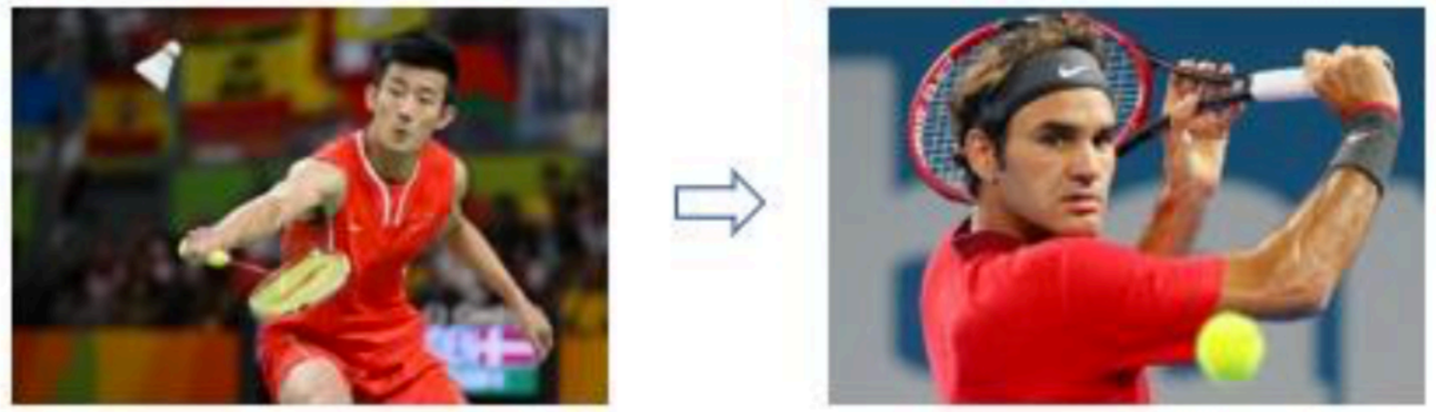
Jiaru Zhang
12.7.2018

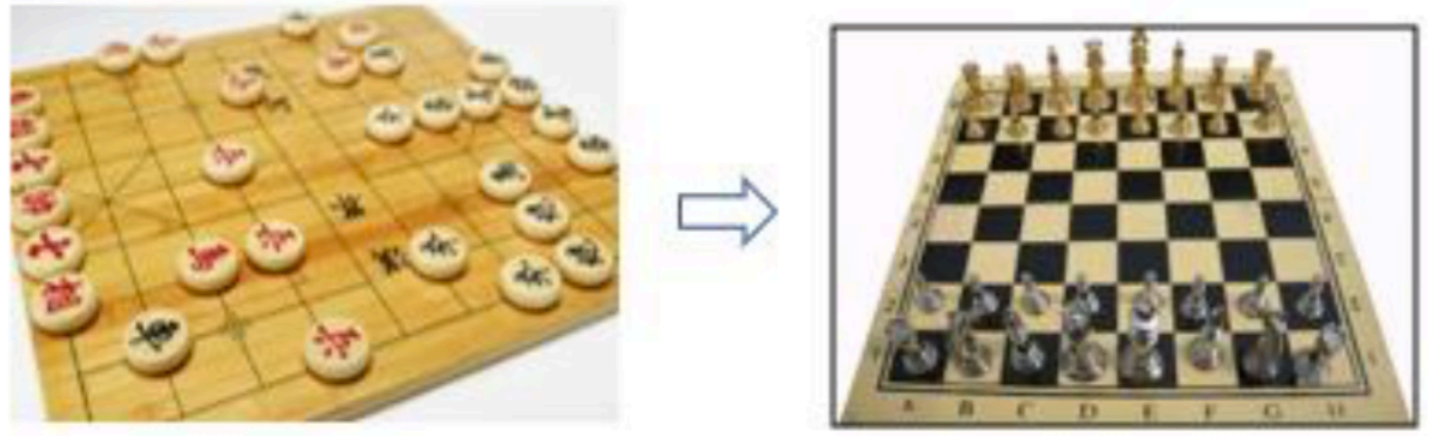
Contents

Introduction

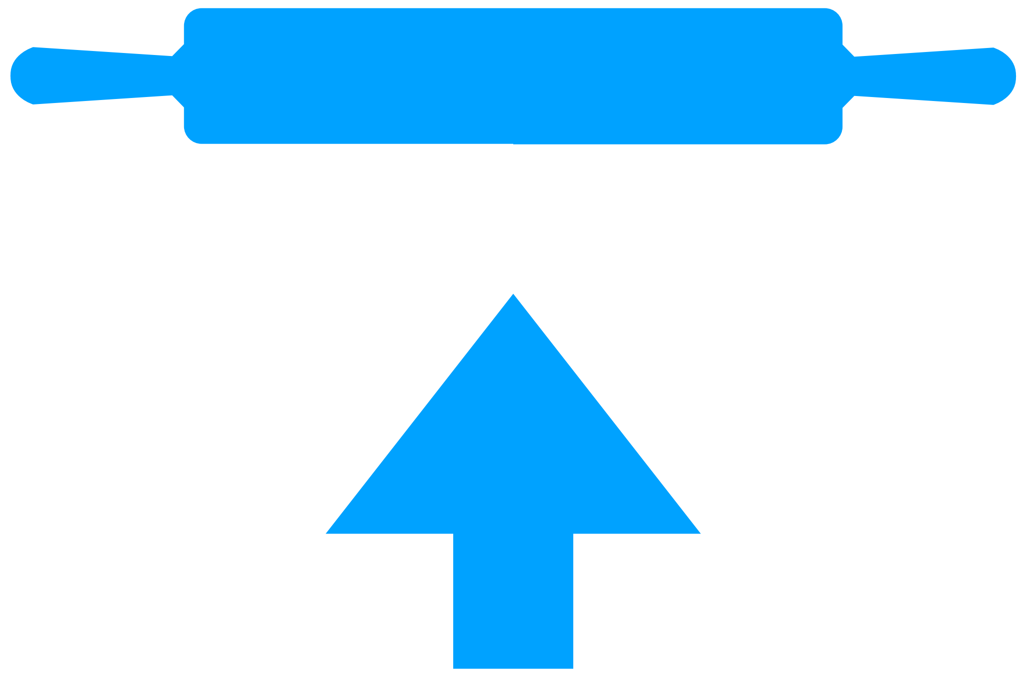
Transfer Learning

The application of skills, knowledge, and/or attitudes that were learned in one situation to another **learning** situation (Perkins, 1992)





Introduction

Transfer Learning  **Deep Learning**

- Big data
- Powerful computation
- New algorithmic techniques
- Mature software packages and architectures
- ……

Introduction

Why is deep learning so significant?

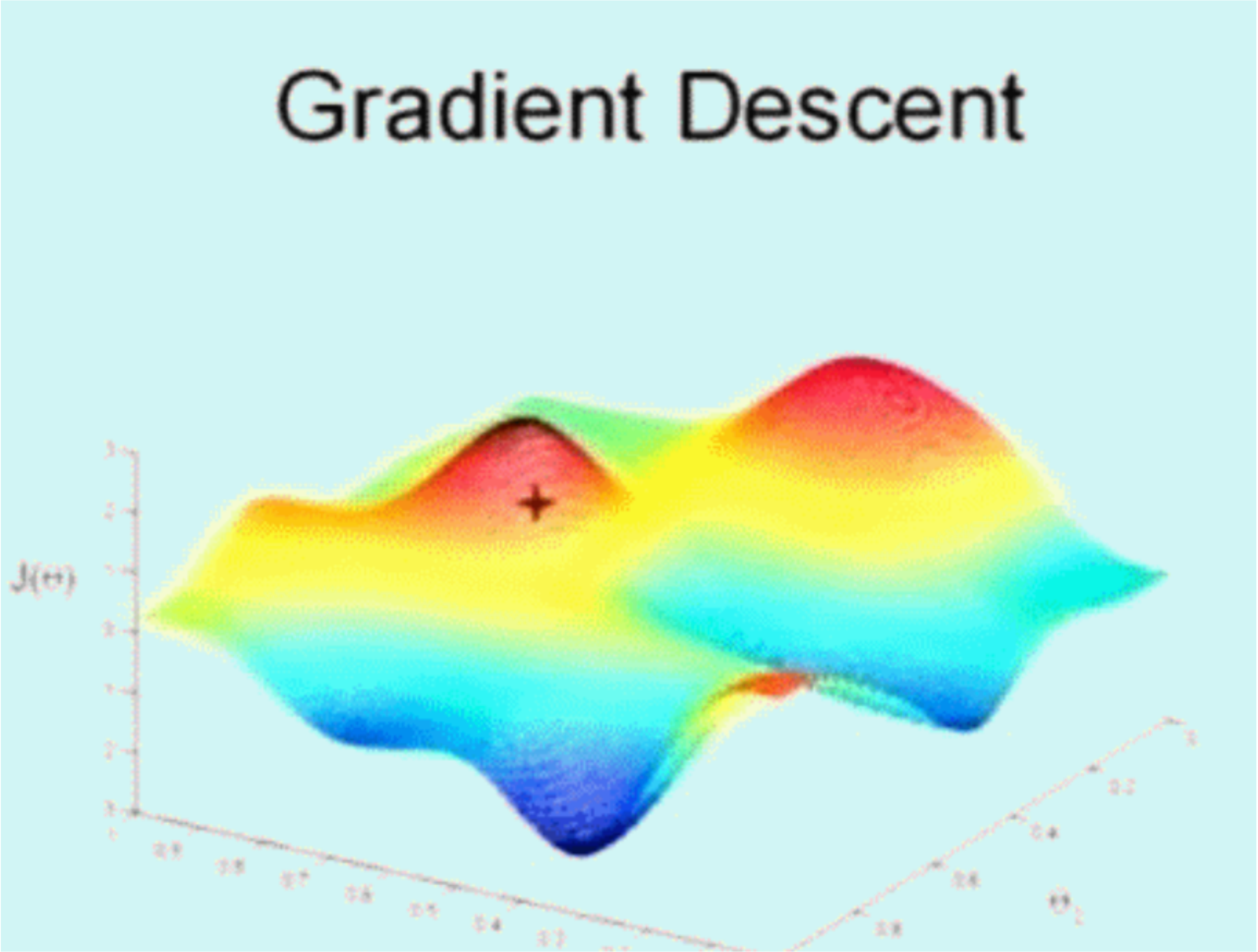
Deep learning

Features:
Radish/Red
Fruit
Shape
etc…

Features:
Sky Blue
Logo
Shape
etc…

Features:
Yellow
Fruit
Shape
etc…

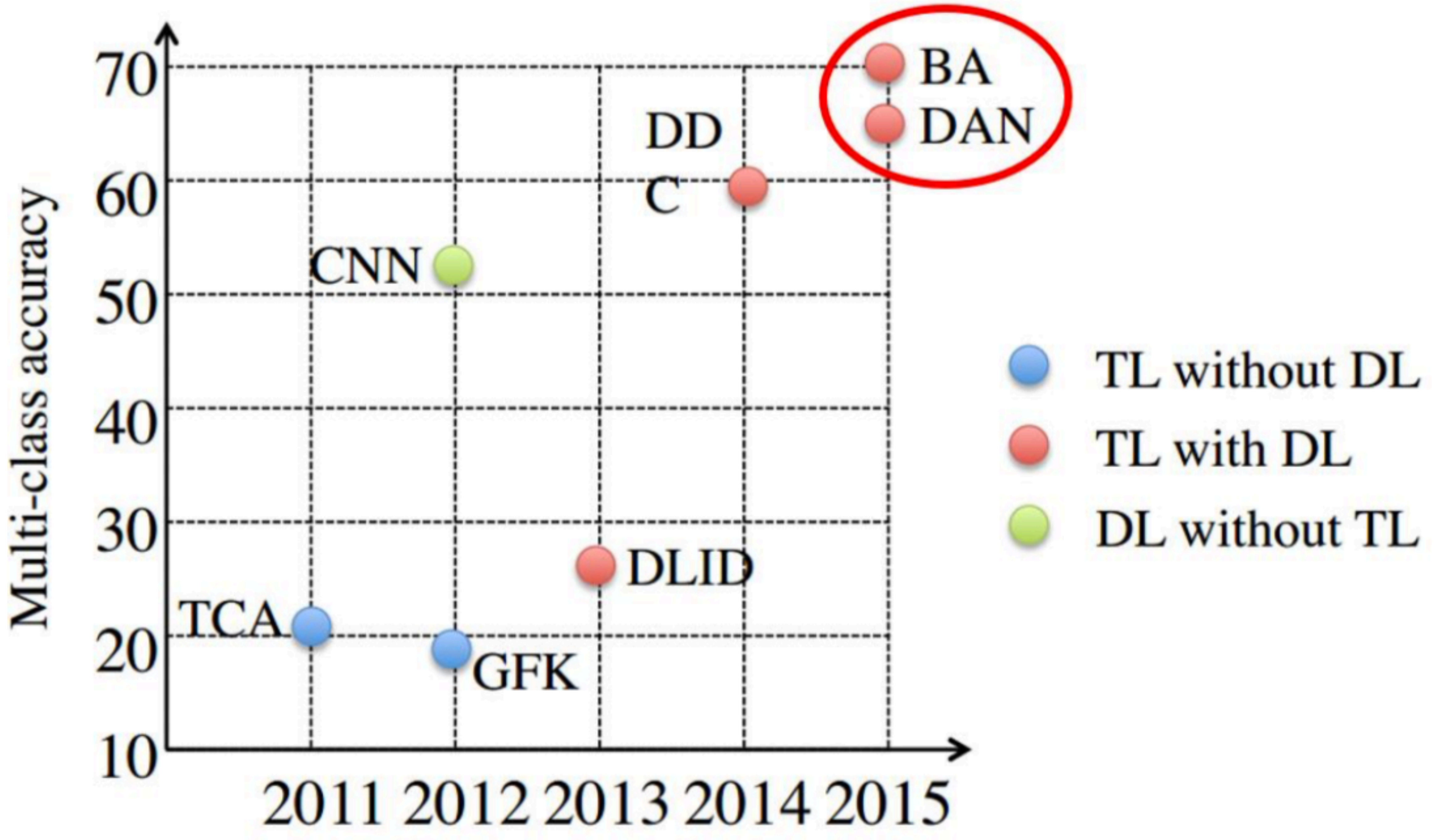
Feature Engineering



End-to-end learning through gradient descent

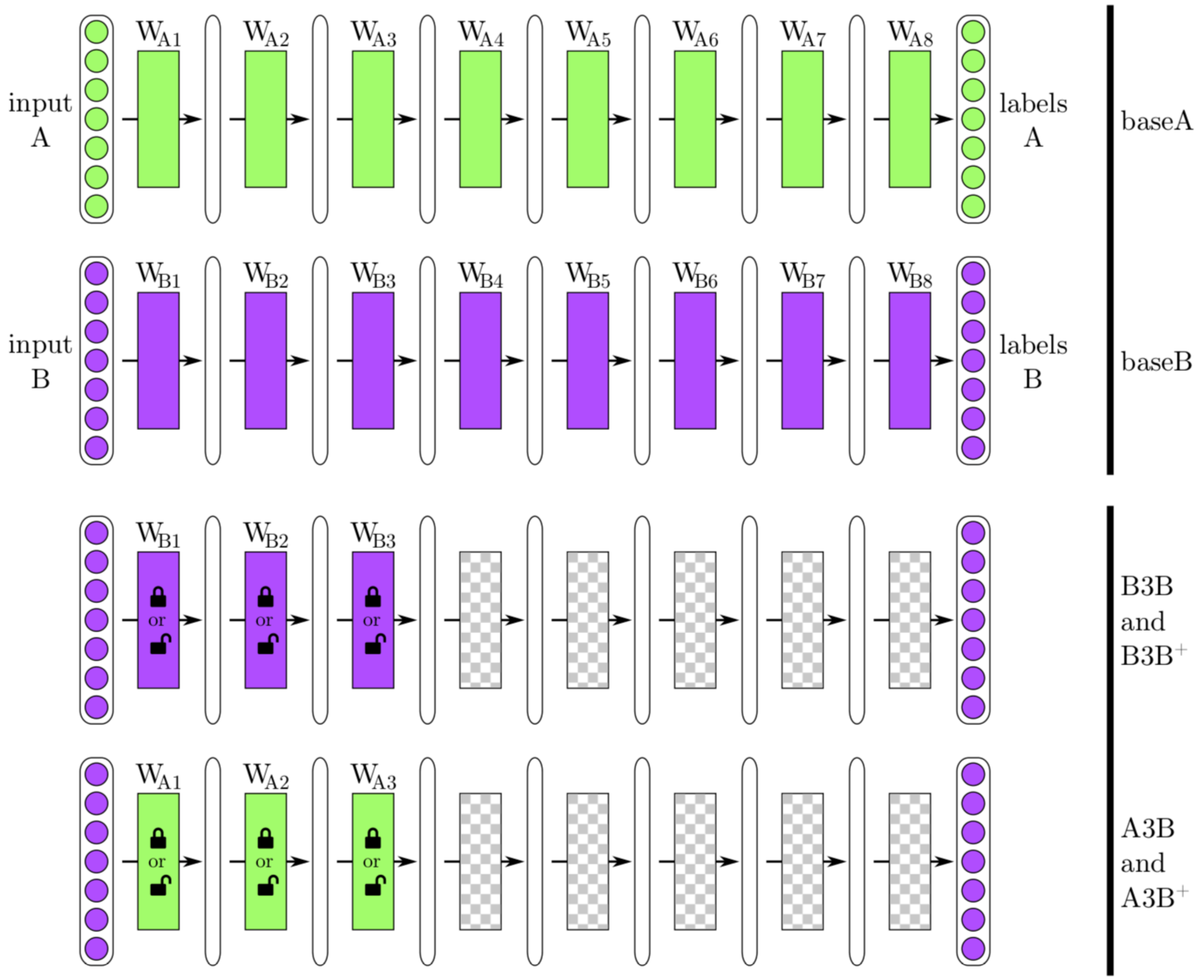
Introduction

Comparison



Introduction

How transferable are features in deep neural networks? [1]

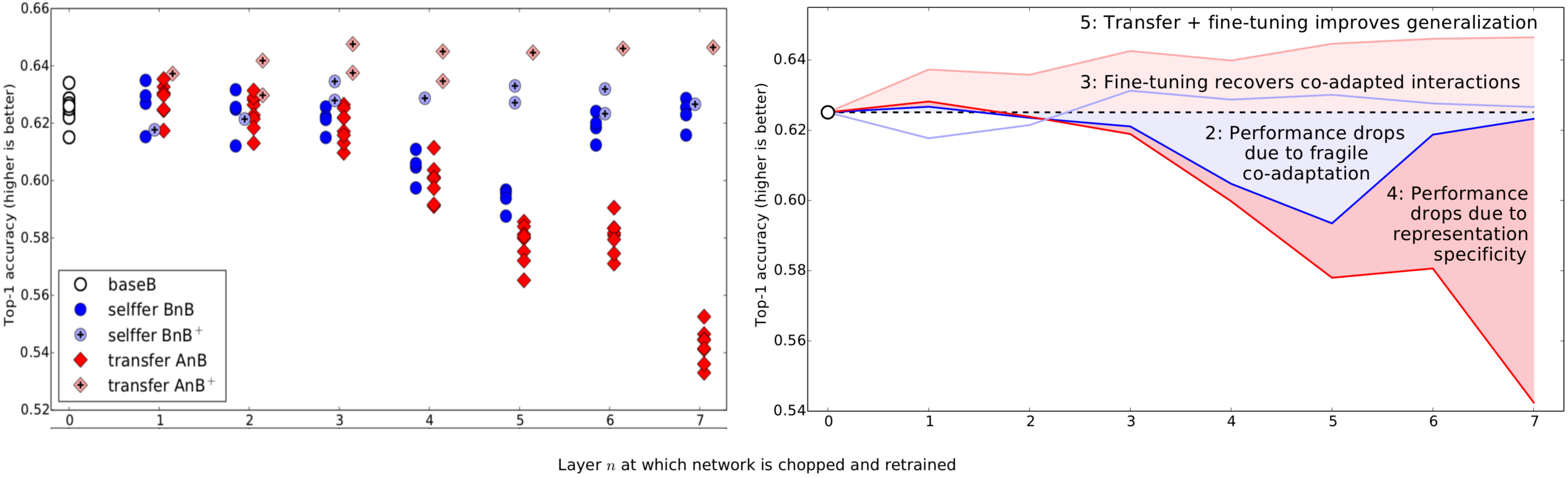


- BnB: First n layers are copied from base B and frozen. Others are randomly initialized.
- AnB: First n layers are copied from base A and frozen. Others are randomly initialized.
- BnB+: BnB but all layers trainable.
- AnB+: AnB but all layers trainable.

[1] Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In NeurIPS, 2014

Introduction

How transferable are features in deep neural networks?



Introduction

How transferable are features in deep neural networks?

Conclusion of the paper:

- The first 3 layers are general.
- Fine-tune improves performance notably.
- By Fine-tuning data from different domain can be used.
- Deep transfer networks are better than randomly initialized ones.

Contents

Core Methods

Why we need domain transfer methods?

	Train set		Test set	
Source domain	x_s	y_s	\	\
Target domain	x_T	y_T	x_T	?

In fine-tune method, y_T is needed!

Core Methods

Domain Adaptive Neural Networks for Object Detection [2]

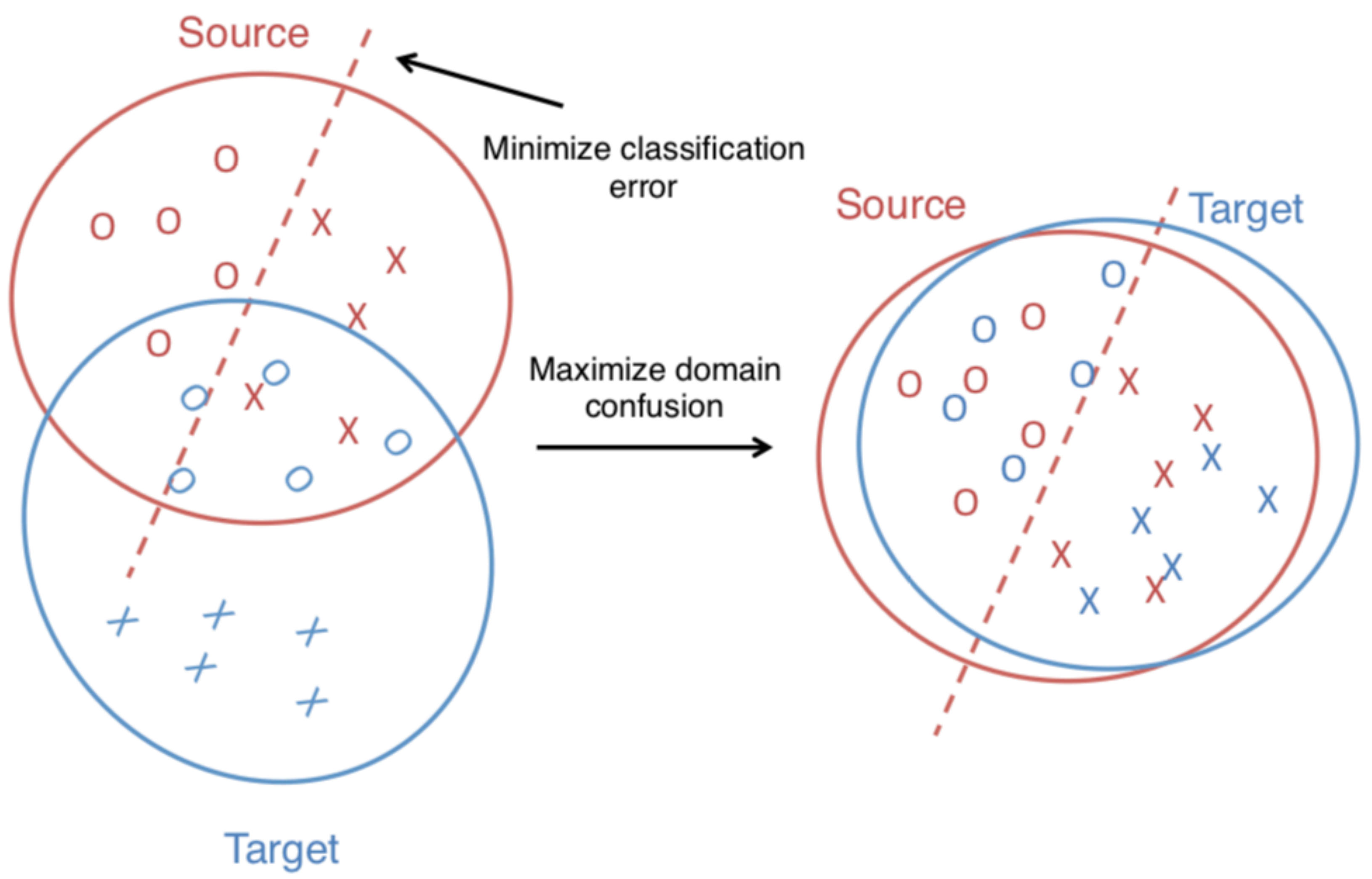
Maximum Mean Discrepancy (MMD):

$$
\begin{aligned}
\mathcal{MMD}_{\mathrm{e}}(\mathbf{x}_s, \mathbf{x}_t) &= \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\mathbf{x}_s^{(i)}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(\mathbf{x}_t^{(j)}) \right\|_{\mathcal{H}} \\
&= \left(\frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(\mathbf{x}_s^{(i)}, \mathbf{x}_s^{(j)}) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}) \right. \\
&\quad \left. - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(\mathbf{x}_s^{(i)}, \mathbf{x}_t^{(j)}) \right)^{\frac{1}{2}} \\
&= \left(\frac{\mathrm{Tr}\left(\mathbf{K}_{xss}\right)}{n_s^2} + \frac{\mathrm{Tr}\left(\mathbf{K}_{xtt}\right)}{n_t^2} - 2\frac{\mathrm{Tr}\left(\mathbf{K}_{xst}\right)}{n_s n_t} \right)^{\frac{1}{2}},
\end{aligned}
$$

[2] Muhammad Ghifary, W. Bastiaan Kleijn, and Mengjie Zhang. Domain Adaptive Neural Networks for Object Recognition. In PRICAI, 2014

Core Methods

Domain Adaptive Neural Networks for Object Detection



Joint loss function:

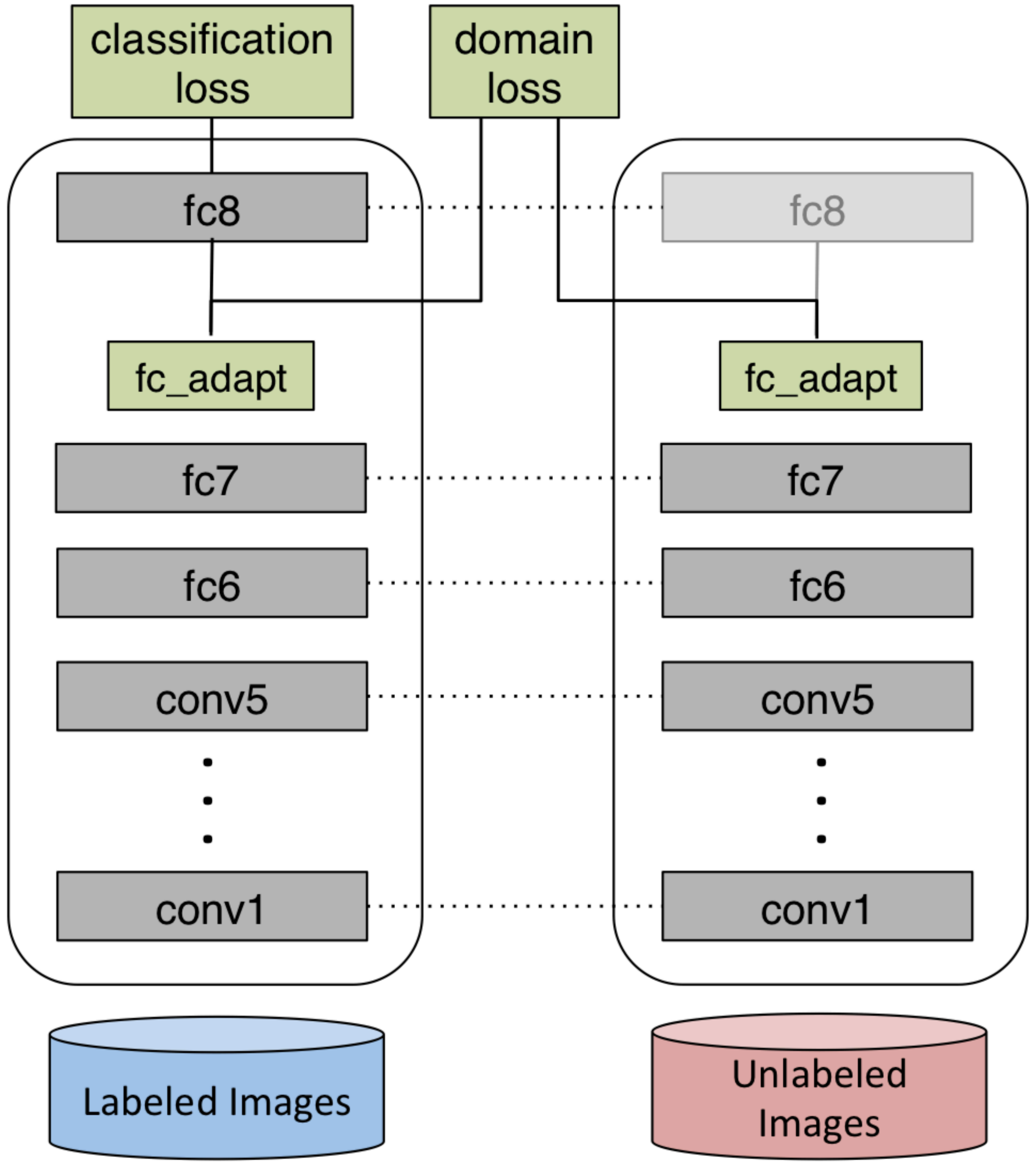
$$J_{\mathrm{DaNN}} = J_{\mathrm{NNs}} + \gamma \mathcal{MMD}_e^2(\mathbf{q}_s, \bar{\mathbf{q}}_t),$$

where

$$\mathbf{q}_s = \mathbf{W}_1^\top \mathbf{x}_s + \mathbf{b},\ \bar{\mathbf{q}}_t = \mathbf{W}_1^\top \mathbf{x}_t + \mathbf{b}$$

Core Methods

Deep Domain Confusion: Maximizing for Domain Invariance [3]



Improvement: Deeper network (Alexnet).

[3] Tzeng E, Hoffman J, Zhang N, et al. Deep domain confusion: Maximizing for domain invariance. arXiv preprint arXiv:1412.3474, 2014

Core Methods

Learning Transferable Features with Deep Adaption Networks [4]

Multiple Kernel variant of Maximum Mean Discrepancy (MMD):

$$
\begin{aligned}
\mathcal{MMD}_{\mathrm{e}}(\mathbf{x}_s, \mathbf{x}_t) &= \left\| \frac{1}{n_s} \sum_{i=1}^{n_s} \phi(\mathbf{x}_s^{(i)}) - \frac{1}{n_t} \sum_{j=1}^{n_t} \phi(\mathbf{x}_t^{(j)}) \right\|_{\mathcal{H}} \\
&= \left(\frac{1}{n_s^2} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(\mathbf{x}_s^{(i)}, \mathbf{x}_s^{(j)}) + \frac{1}{n_t^2} \sum_{i=1}^{n_t} \sum_{j=1}^{n_t} k(\mathbf{x}_t^{(i)}, \mathbf{x}_t^{(j)}) \right. \\
&\left. - \frac{2}{n_s n_t} \sum_{i=1}^{n_s} \sum_{j=1}^{n_s} k(\mathbf{x}_s^{(i)}, \mathbf{x}_t^{(j)}) \right)^{\frac{1}{2}} \\
&= \left(\frac{\mathrm{Tr}\,(\mathbf{K}_{xss})}{n_s^2} + \frac{\mathrm{Tr}\,(\mathbf{K}_{xtt})}{n_t^2} - 2\frac{\mathrm{Tr}\,(\mathbf{K}_{xst})}{n_s n_t} \right)^{\frac{1}{2}},
\end{aligned}
$$

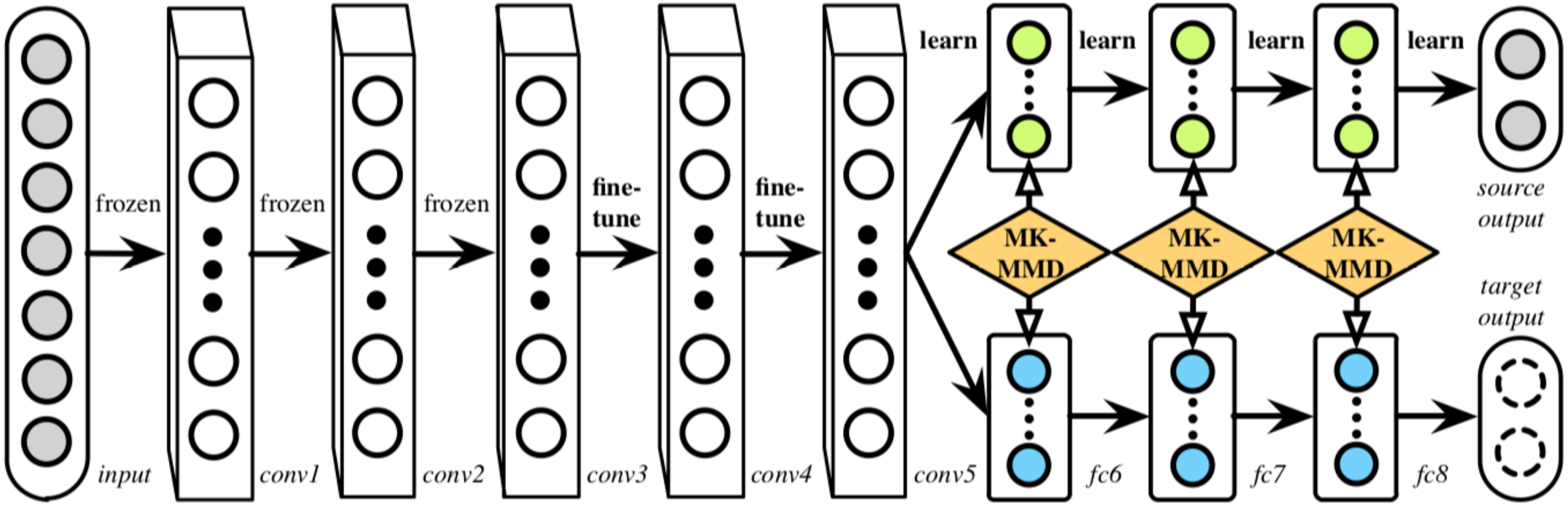
$$
\mathcal{K} := \left\{ k : k = \sum_{u=1}^{d} \beta_u k_u, \sum_{u=1}^{d} \beta_u = D, \beta_u \geq 0, \forall u \in \{1, \ldots, d\} \right\}
$$

[4] Long M, Cao Y, Wang J, et al. Learning transferable features with deep adaptation networks. In ICML, 2015.

Core Methods

Learning Transferable Features with Deep Adaption Networks

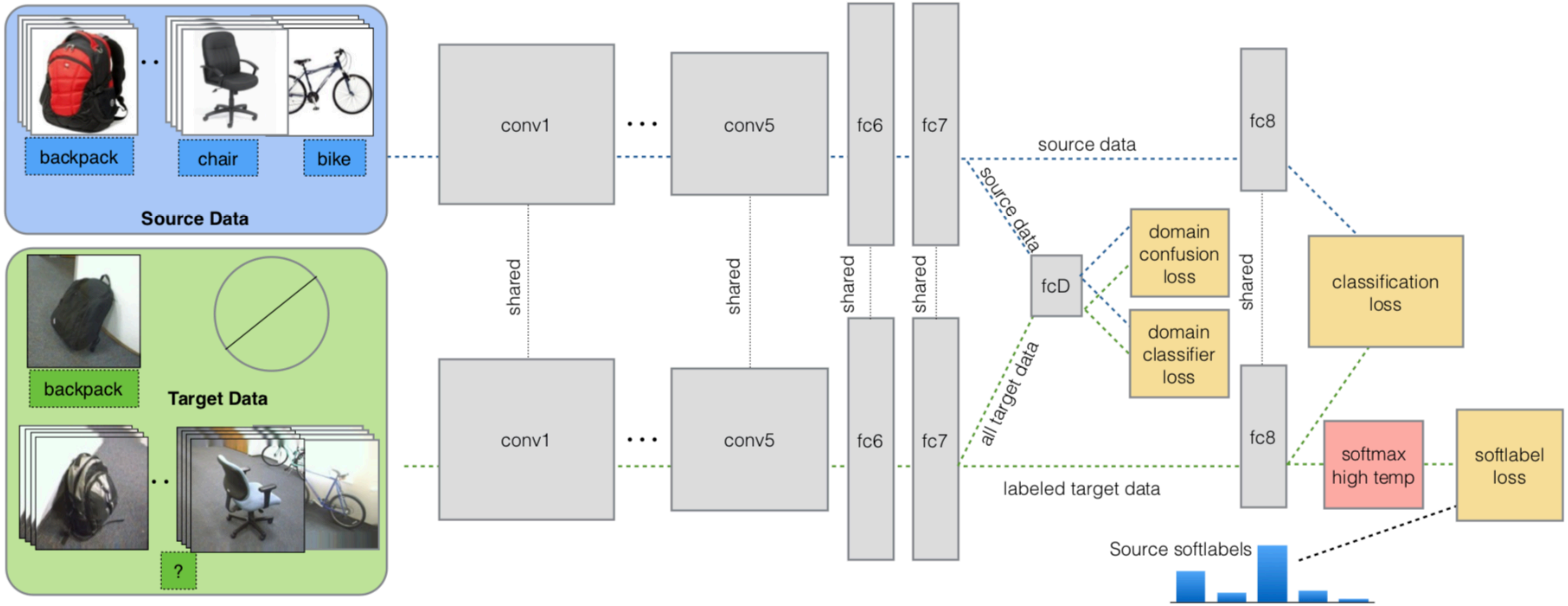
Adaption on multiple layers:



$$\min_{\Theta} \frac{1}{n_a} \sum_{i=1}^{n_a} J(\theta(\mathbf{x}_i^a), y_i^a) + \lambda \sum_{l=l_1}^{l_2} d_k^2(\mathcal{D}_s^l, \mathcal{D}_t^l)$$

Core Methods

Simultaneous deep transfer across domains and tasks [5]



[5] Tzeng, E., Hoffman, J., Darrell, T., and Saenko, K. (2015). Simulta- neous deep transfer across domains and tasks. In ICCV, 2015.

Core Methods

Simultaneous deep transfer across domains and tasks

$$\mathcal{L}_C(x, y; \theta_{\text{repr}}, \theta_C) = -\sum_k \mathbb{1}[y = k] \log p_k$$

$$\mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D) = -\sum_d \mathbb{1}[y_D = d] \log q_d$$

$$\mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) = -\sum_d \frac{1}{D} \log q_d.$$

$$\mathcal{L}_{\text{soft}}(x_T, y_T; \theta_{\text{repr}}, \theta_C) = -\sum_i l_i^{(y_T)} \log p_i$$

1

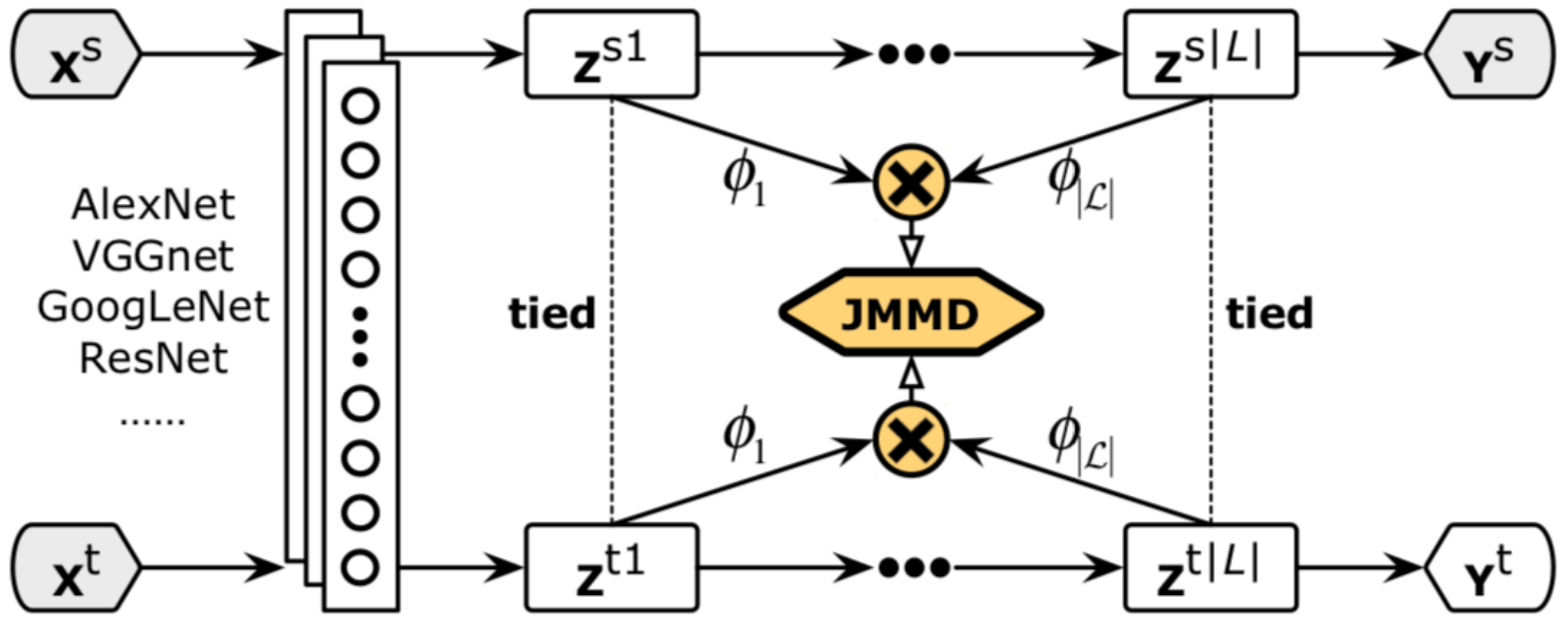
$$\min_{\theta_D} \mathcal{L}_D(x_S, x_T, \theta_{\text{repr}}; \theta_D)$$
$$\min_{\theta_{\text{repr}}} \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}).$$

2

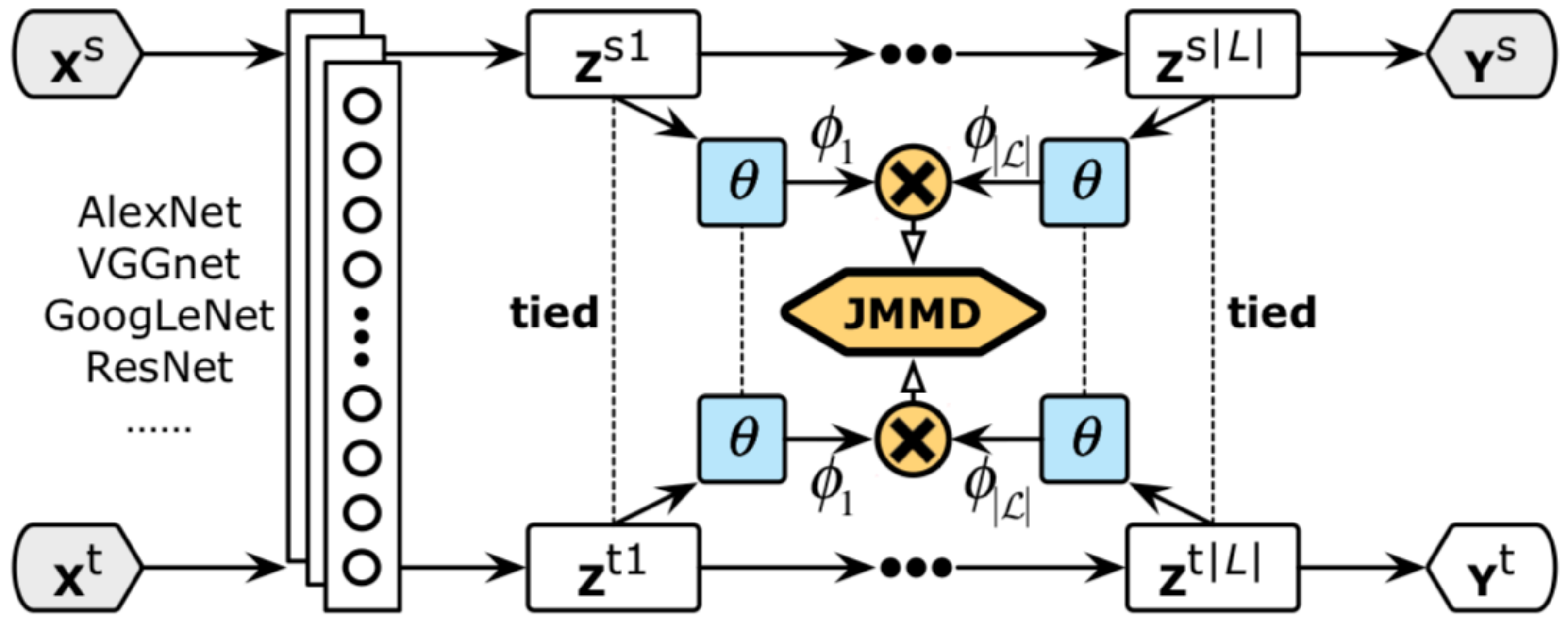
$$\begin{aligned}\mathcal{L}(x_S, y_S, x_T, y_T, \theta_D; \theta_{\text{repr}}, \theta_C) = \\ & \mathcal{L}_C(x_S, y_S, x_T, y_T; \theta_{\text{repr}}, \theta_C) \\ & + \lambda \mathcal{L}_{\text{conf}}(x_S, x_T, \theta_D; \theta_{\text{repr}}) \\ & + \nu \mathcal{L}_{\text{soft}}(x_T, y_T; \theta_{\text{repr}}, \theta_C).\end{aligned}$$

Core Methods

Deep Transfer Learning with Joint Adaptation Networks [6]



(a) Joint Adaptation Network (JAN)



(b) Adversarial Joint Adaptation Network (JAN-A)

$$\widehat{\mathcal{C}}_{\mathbf{X}^{1:m}} = \frac{1}{n}\sum_{i=1}^{n} \otimes_{\ell=1}^{m} \phi^{\ell}\left(\mathbf{x}_i^{\ell}\right).$$

$$\begin{aligned} D_{\mathcal{L}}(P,Q) &\triangleq \left\|\mathcal{C}_{\mathbf{Z}^{s,1:|\mathcal{L}|}}(P) - \mathcal{C}_{\mathbf{Z}^{t,1:|\mathcal{L}|}}(Q)\right\|^2_{\otimes_{\ell=1}^{|\mathcal{L}|}\mathcal{H}^{\ell}} \\ &= \frac{1}{n_s^2}\sum_{i=1}^{n_s}\sum_{j=1}^{n_s}\prod_{\ell\in\mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{s\ell},\mathbf{z}_j^{s\ell}\right) \\ &+ \frac{1}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\prod_{\ell\in\mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{t\ell},\mathbf{z}_j^{t\ell}\right) \\ &- \frac{2}{n_s n_t}\sum_{i=1}^{n_s}\sum_{j=1}^{n_t}\prod_{\ell\in\mathcal{L}} k^{\ell}\left(\mathbf{z}_i^{s\ell},\mathbf{z}_j^{t\ell}\right). \end{aligned}$$

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

[6] Long, M., Wang, J., and Jordan, M. I. Deep transfer learning with joint adaptation networks. In ICML, 2017.

Contents

Other Methods

Adaptive Batch Normalization for practical domain adaptation [7]

Algorithm 1 Adaptive Batch Normalization (AdaBN)

---

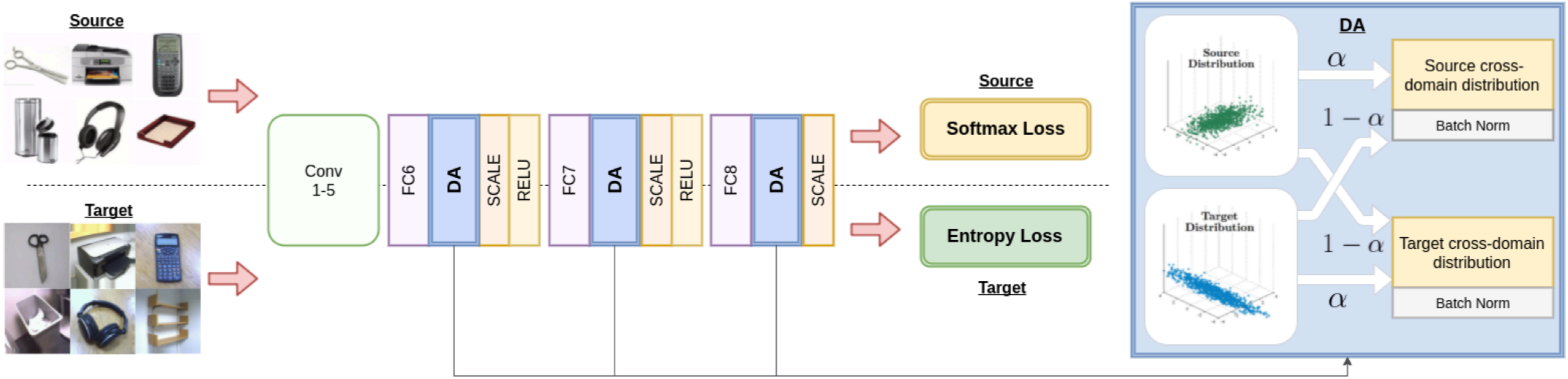
for neuron j in DNN **do**
  Concatenate neuron responses on all images of target domain t: $\mathbf{x}_j = [\ldots, x_j(m), \ldots]$
  Compute the mean and variance of the target domain: $\mu_j^t = \mathbb{E}(\mathbf{x}_j^t)$, $\sigma_j^t = \sqrt{\mathrm{Var}(\mathbf{x}_j^t)}$.
end for
for neuron j in DNN, testing image m in target domain **do**
  Compute BN output $y_j(m) := \gamma_j \frac{\left(x_j(m) - \mu_j^t\right)}{\sigma_j^t} + \beta_j$
end for

Utilize the same variance and bias on both domains.

[7] Li, Y., Wang, N., Shi, J., Hou, X., and Liu, J. Adaptive batch normalization for practical domain adaptation. Pattern Recognition, 2018, 80:109–117

Other Methods

AutoDIAL: Automatic DomaIn Alignment Layers [8]



[8] Carlucci, F. M., Porzi, L., Caputo, B., Ricci, E., and Bulò, S. R. AutoDIAL: Automatic domaIn Alignment Layers. In ICCV, 2017

Contents

Outlook on Future

- Combination with human knowledge
- Transitive transfer learning
- Online transfer learning
- Transfer reinforcement learning
- …