Transfer Learning: Theories and Algorithms

Mingsheng Long

School of Software, Tsinghua University
Research Center for Big Data, Tsinghua University
National Engineering Laboratory for Big Data Software

mingsheng@tsinghua.edu.cn
http://ise.thss.tsinghua.edu.cn/~mlong
Outline

1. Transfer Learning

2. $\mathcal{H}\Delta\mathcal{H}$-Divergence
   - DAN: Deep Adaptation Network
   - DANN: Domain Adversarial Neural Network
   - MCD: Maximum Classifier Discrepancy

3. Margin Disparity Discrepancy
   - MDD: Margin Disparity Discrepancy

4. Transfer Model Selection
   - DEV: Deep Embedded Validation

5. Evaluation and Implementation
Supervised Learning

Learner: $f : x \rightarrow y$

Distribution: $(x, y) \sim P(x, y)$

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

- Machine learning across domains of different distributions $P \neq Q$
  - Independent and Differently Distributed (IDD)
- How to effectively bound the generalization error on target domain?

Source Domain

$\ell_S$  
Model $f : x \rightarrow y$

Simulation

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

Target Domain

$\ell_T$

Model $f : x \rightarrow y$

Real
Bias-Variance-Shift Tradeoff

<table>
<thead>
<tr>
<th>Training Set</th>
<th>Train-Dev Set</th>
<th>Dev Set</th>
<th>Test Set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Error high?</td>
<td>Optimal Bayes Rate</td>
<td>Yes</td>
<td>Bias</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>Deeper Model</td>
<td></td>
</tr>
<tr>
<td>Train-Dev Error high?</td>
<td></td>
<td>Yes</td>
<td>Longer Training</td>
</tr>
<tr>
<td>No</td>
<td>Variance</td>
<td>Bigger Data</td>
<td></td>
</tr>
<tr>
<td>Dev Error high?</td>
<td></td>
<td>Yes</td>
<td>Regularization</td>
</tr>
<tr>
<td>No</td>
<td>Dataset Shift</td>
<td>Transfer Learning</td>
<td></td>
</tr>
<tr>
<td>Test Error high?</td>
<td></td>
<td>Yes</td>
<td>Data Generation</td>
</tr>
<tr>
<td>No</td>
<td>Overfit Dev Set</td>
<td>Bigger Dev Data</td>
<td></td>
</tr>
<tr>
<td>No</td>
<td>Done!</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Everything should be made as simple as possible, but no simpler.
—Albert Einstein

There is nothing more practical than a good theory.
—Vladimir Vapnik
Outline

1 Transfer Learning

2 $\mathcal{H}\Delta\mathcal{H}$-Divergence
   - DAN: Deep Adaptation Network
   - DANN: Domain Adversarial Neural Network
   - MCD: Maximum Classifier Discrepancy

3 Margin Disparity Discrepancy
   - MDD: Margin Disparity Discrepancy

4 Transfer Model Selection
   - DEV: Deep Embedded Validation

5 Evaluation and Implementation
Notations and Assumptions

- Source risk: \( \epsilon_P(h) = \mathbb{E}_{(x,y) \sim P} [h(x) \neq y], \{(x_i, y_i)\}_{i=1}^n \sim P^n \)
- Target risk: \( \epsilon_Q(h) = \mathbb{E}_{(x,y) \sim Q} [h(x) \neq y], \{(x_i, y_i)\}_{i=1}^m \sim Q^m \)
- Source disparity: \( \epsilon_P(h_1, h_2) = \mathbb{E}_{(x,y) \sim P} [h_1(x) \neq h_2(x)] \)
- Target disparity: \( \epsilon_Q(h_1, h_2) = \mathbb{E}_{(x,y) \sim Q} [h_1(x) \neq h_2(x)] \)

- Ideal joint hypothesis: \( h^* = \arg \min_h \epsilon_P(h) + \epsilon_Q(h) \)
- Assumption: ideal hypothesis has small risk \( \epsilon_{ideal} = \epsilon_P(h^*) + \epsilon_Q(h^*) \)

Ideal joint hypothesis

Ideal hypothesis with small error
Relating the Target Risk to the Source Risk

**Theorem**

Assuming small $\epsilon_{\text{ideal}}$, the bound of the target risk $\epsilon_Q(h)$ of hypothesis $h \in \mathcal{H}$ is given by the source risk $\epsilon_P(h)$ plus the disparity difference:

$$\epsilon_Q(h) \leq \epsilon_P(h) + [\epsilon_P(h^*) + \epsilon_Q(h^*)] + |\epsilon_P(h, h^*) - \epsilon_Q(h, h^*)|$$  \hspace{1cm} (1)

**Proof.**

Simply by using the triangle inequalities, we have

$$\epsilon_Q(h) \leq \epsilon_Q(h^*) + \epsilon_Q(h, h^*)$$

$$\leq \epsilon_Q(h^*) + \epsilon_P(h, h^*) + \epsilon_Q(h, h^*) - \epsilon_P(h, h^*)$$

$$\leq \epsilon_Q(h^*) + \epsilon_P(h, h^*) + |\epsilon_Q(h, h^*) - \epsilon_P(h, h^*)|$$

$$\leq \epsilon_P(h) + [\epsilon_P(h^*) + \epsilon_Q(h^*)] + |\epsilon_P(h, h^*) - \epsilon_Q(h, h^*)|$$  \hspace{1cm} (2)
How to Bound the Disparity Difference?

- We can illustrate the disparity difference $|\epsilon_P(h, h^*) - \epsilon_Q(h, h^*)|$ as

\[ \mathcal{H}\Delta\mathcal{H}\text{-Divergence}\]

\[ d_{\mathcal{H}\Delta\mathcal{H}}(P, Q) \triangleq \sup_{h, h' \in \mathcal{H}} |\epsilon_P(h, h') - \epsilon_Q(h, h')| \]

- Hypothesis-independent discrepancy—depending on hypothesis space.

---

Theorem (Generalization Bound)

Denote by $d$ the VC-dimension of hypothesis space $\mathcal{H}$. For any hypothesis $h \in \mathcal{H}$,

$$\epsilon_Q(h) \leq \epsilon_{\hat{P}}(h) + d_{\mathcal{H}\Delta\mathcal{H}}(\hat{P}, \hat{Q}) + \epsilon_{\text{ideal}}$$

$$+ O\left(\sqrt{d \log n \over n} + \sqrt{d \log m \over m}\right)$$

- $\epsilon_{\hat{P}}(h)$ depicts the performance of $h$ on source domain.
- $d_{\mathcal{H}\Delta\mathcal{H}}$ bounds the generalization gap caused by domain shift.
- $\epsilon_{\text{ideal}}$ quantifies the inverse of “adaptability” between domains.
- The order of the complexity term is $O\left(\sqrt{d \log n / n} + \sqrt{d \log m / m}\right)$. 
Approximating $\mathcal{H}\Delta\mathcal{H}$-Divergence by Statistical Distance

For binary hypothesis $h$, the $\mathcal{H}\Delta\mathcal{H}$-Divergence can be bounded by

$$d_{\mathcal{H}\Delta\mathcal{H}}(P, Q) \triangleq \sup_{h, h' \in \mathcal{H}} |\epsilon_P(h, h') - \epsilon_Q(h, h')|$$

$$= \sup_{h, h' \in \mathcal{H}} |\mathbb{E}_P[|h(x) - h'(x)| \neq 0] - \mathbb{E}_Q[|h(x) - h'(x)| \neq 0]|$$

$$= \sup_{\delta \in \mathcal{H}\Delta\mathcal{H}} |\mathbb{E}_P[\delta(x) \neq 0] - \mathbb{E}_Q[\delta(x) \neq 0]|$$

The last term takes the form of Integral Probability Metric (IPM):

$$d_F(P, Q) = \sup_{f \in \mathcal{F}} |\mathbb{E}_{x \sim P} f(x) - \mathbb{E}_{x \sim Q} f(x)|$$

Assuming $\mathcal{F}$ can be approximated by kernel functions in RKHS, $d_F(P, Q)$ turns into Maximum Mean Discrepancy (MMD) (a statistical distance)
DAN: Deep Adaptation Network\textsuperscript{2}

\[ d_k^2 (P, Q) \triangleq \| E_P [\phi (x^s)] - E_Q [\phi (x^t)] \|^2_{\mathcal{H}_k} \] \hfill (6)

\[ \min_{\theta \in \Theta} \max_{k \in \mathcal{K}} \frac{1}{n_a} \sum_{i=1}^{n_a} L (\theta (x_i^a), y_i^a) + \lambda \sum_{\ell=l_1}^{l_2} d_k^2 (\hat{P}_\ell, \hat{Q}_\ell) \] \hfill (7)

Approximating $\mathcal{H}\Delta\mathcal{H}$-Divergence by Domain Discriminator

For binary hypothesis $h$, the $\mathcal{H}\Delta\mathcal{H}$-Divergence can be bounded by

$$d_{\mathcal{H}\Delta\mathcal{H}}(P, Q) \triangleq \sup_{h, h' \in \mathcal{H}} |\epsilon_P(h, h') - \epsilon_Q(h, h')|$$

$$= \sup_{\delta \in \mathcal{H}\Delta\mathcal{H}} |\mathbb{E}_P[\delta(x) \neq 0] - \mathbb{E}_Q[\delta(x) \neq 0]|$$

$$\leq \sup_{D \in \mathcal{H}_D} |\mathbb{E}_P[D(x) = 1] + \mathbb{E}_Q[D(x) = 0]|$$

This upper-bound can be yielded by training a domain discriminator $D(x)$.
**DANN: Domain Adversarial Neural Network**

**Adversarial adaptation:** learning features indistinguishable across domains

\[
E(\theta_f, \theta_y, \theta_d) = \sum_{x_i \sim \hat{P}} L_y(G_y(G_f(x_i)), y_i) - \lambda \sum_{x_i \sim \hat{P} \cup \hat{Q}} L_d(G_d(G_f(x_i)), d_i) \tag{9}
\]

\[
(\hat{\theta}_f, \hat{\theta}_y) = \arg \min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \theta_d) \quad (\hat{\theta}_d) = \arg \max_{\theta_d} E(\theta_f, \theta_y, \theta_d) \tag{10}
\]

3 Ganin et al. Domain Adversarial Training of Neural Networks, JMLR 2016.

---

**Ganin et al. Domain Adversarial Training of Neural Networks. JMLR 2016.**

**Mingsheng Long**

**Transfer Learning**

**August 21, 2019**
Approximating $\cal{H}\Delta\cal{H}$-Divergence by Classifier Consistency

- Use two classifiers $G_1$, $G_2$ to approximate $\sup_{h,h' \in \cal{H}} |\epsilon_P(h, h') - \epsilon_Q(h, h')|$
- Assume $G_1 = h$ and $G_2 = h'$ should agree on source domain.
- Use $L_1$-loss of two classifiers’ outputs to approximate disagreement:

$$\min\{ \min_{\phi, G_1, G_2} \mathbb{E}_P [L(G_1(x), y) + L(G_2(x), y)] + \max_{G_1, G_2} \mathbb{E}_Q |G_1(x) - G_2(x)| \}$$

---

Outline

1. Transfer Learning

2. $\mathcal{H}\Delta\mathcal{H}$-Divergence
   - DAN: Deep Adaptation Network
   - DANN: Domain Adversarial Neural Network
   - MCD: Maximum Classifier Discrepancy

3. Margin Disparity Discrepancy
   - MDD: Margin Disparity Discrepancy

4. Transfer Model Selection
   - DEV: Deep Embedded Validation

5. Evaluation and Implementation
Towards Informative Margin Theory

- Towards a rigorous **multiclass** domain adaptation theory.
  - All existing theories are only applicable to binary classification.

- Towards an informative **margin theory**.
  - Explore the idea of margin in measuring domain discrepancy.

- Towards a certain **function class** in the theoretical bound.
  - Eliminate approximation assumptions in all existing methods.

- Towards **bridging** the existing gap between theories and algorithms.

---

Notations

- **Scoring function**: \( f \in \mathcal{F} : \mathcal{X} \times \mathcal{Y} \to \mathbb{R} \)
- **Labeling function** induced by \( f \): \( h_f : x \mapsto \arg \max_{y \in \mathcal{Y}} f(x, y) \)
- **Labeling function class**: \( \mathcal{H} = \{ h_f | f \in \mathcal{F} \} \)
- **Margin of a hypothesis**:
  \[
  \rho_f(x, y) = \frac{1}{2}(f(x, y) - \max_{y' \neq y} f(x, y'))
  \]
- **Margin Loss**:
  \[
  \Phi_{\rho}(x) = \begin{cases} 0 & \rho \leq x \\ 1 - x/\rho & 0 \leq x \leq \rho \\ 1 & x \leq 0 \end{cases}
  \]
**MDD: Margin Disparity Discrepancy**

- **Margin risk:** $\epsilon_{D}^{(\rho)}(f) = \mathbb{E}_{(x,y) \sim D}[\Phi_{\rho}(\rho_{f}(x,y))]$
- **Margin disparity:** $\epsilon_{D}^{(\rho)}(f', f) \triangleq \mathbb{E}_{x \sim D_{X}}[\Phi_{\rho}(\rho_{f'}(x, h_{f}(x)))]

**Definition (Margin Disparity Discrepancy, MDD)**

With above definitions, we define Margin Disparity Discrepancy (MDD) and its empirical version by

$$d_{f,\mathcal{F}}^{(\rho)}(P, Q) \triangleq \sup_{f' \in \mathcal{F}} \left( \epsilon_{Q}^{(\rho)}(f', f) - \epsilon_{P}^{(\rho)}(f', f) \right),$$

and

$$d_{f,\mathcal{F}}^{(\rho)}(\hat{P}, \hat{Q}) \triangleq \sup_{f' \in \mathcal{F}} \left( \epsilon_{\hat{Q}}^{(\rho)}(f', f) - \epsilon_{\hat{P}}^{(\rho)}(f', f) \right).$$

(12)

MDD satisfies $d_{f,\mathcal{F}}^{(\rho)}(P, P) = 0$ as well as nonnegativity and subadditivity.
Bounding the Target Risk by MDD

Theorem

Let $\mathcal{F} \subseteq \mathbb{R}^X \times Y$ be a hypothesis set with label set $Y = \{1, \cdots, k\}$ and $\mathcal{H} \subseteq Y^X$ be the corresponding $Y$-valued labeling function class. For every scoring function $f \in \mathcal{F}$,

$$\epsilon_Q(f) \leq \epsilon_P^{(\rho)}(f) + d_{f,\mathcal{F}}(P, Q) + \epsilon_{\text{ideal}}^{(\rho)},$$

(13)

where $\epsilon_{\text{ideal}}^{(\rho)}$ is the margin error of ideal joint hypothesis $f^*$:

$$\epsilon_{\text{ideal}}^{(\rho)} = \min_{f^* \in \mathcal{F}} \{\epsilon_P^{(\rho)}(f^*) + \epsilon_Q^{(\rho)}(f^*)\}.$$  

(14)

- Main proof difficulties: margin loss does not satisfy triangle inequality.
  - Solution: One-sided triangle inequality for the margin loss.
- A new tool for analyzing transfer learning with margin theory.
Definitions

Definition (Function Class $\Pi_1 \mathcal{F}$)
Given a class of scoring functions $\mathcal{F}$, $\Pi_1 \mathcal{F}$ is defined as

$$\Pi_1 \mathcal{F} = \{ x \mapsto f(x, y) | y \in \mathcal{Y}, f \in \mathcal{F} \}.$$  \hfill (15)

Definition (Function Class $\Pi_\mathcal{H} \mathcal{F}$)
Given a class of scoring functions $\mathcal{F}$ and a class of the induced labeling functions $\mathcal{H}$, we define $\Pi_\mathcal{H} \mathcal{F}$ as

$$\Pi_\mathcal{H} \mathcal{F} \triangleq \{ x \mapsto f(x, h(x)) | h \in \mathcal{H}, f \in \mathcal{F} \}.$$  \hfill (16)

By applying the margin error over each entry in $\Pi_\mathcal{H} \mathcal{F}$, we obtain the "scoring" version of $\mathcal{H} \Delta \mathcal{H}$ (symmetric difference hypothesis space)
Definitions

Definition (Rademacher Complexity)

The empirical Rademacher complexity of function class $\mathcal{G}$ with respect to the sample $\hat{D}$ is defined as

$$\hat{\mathcal{R}}_{\hat{D}}(\mathcal{G}) = \mathbb{E}_{\sigma} \sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i g(z_i).$$  \hspace{1cm} (17)

where $\sigma_i$’s are independent uniform random variables taking values in $\{-1, +1\}$. The Rademacher complexity is

$$\mathcal{R}_{n,D}(\mathcal{G}) = \mathbb{E}_{\hat{D} \sim D^n} \hat{\mathcal{R}}_{\hat{D}}(\mathcal{G}).$$  \hspace{1cm} (18)

Definition (Covering Number)

(Informal) A covering number $\mathcal{N}_2(\tau, \mathcal{G})$ is the minimal number of $L_2$ balls of radius $\tau > 0$ needed to cover a class $\mathcal{G}$ of bounded functions $g : \mathcal{X} \to \mathbb{R}$
Theorem (Generalization Bound with Rademacher Complexity)

Let $\mathcal{F} \subseteq \mathbb{R}^{X \times Y}$ be a hypothesis set with label set $Y = \{1, \cdots, k\}$ and $\mathcal{H} \subseteq Y^X$ be the corresponding $Y$-valued labeling function class. Fix $\rho > 0$. For all $\delta > 0$, with probability $1 - 3\delta$ the following inequality holds for all hypothesis $f \in \mathcal{F}$:

$$
\epsilon_Q(f) \leq \epsilon_{\hat{P}}^{(\rho)}(f) + d_f^{(\rho)}(\hat{P}, \hat{Q}) + \epsilon_{\text{ideal}}
$$

$$
+ \frac{2k^2}{\rho} \mathcal{R}_{n,P}(\Pi_1 \mathcal{F}) + \frac{k}{\rho} \mathcal{R}_{n,P}(\Pi_\mathcal{H} \mathcal{F}) + 2\sqrt{\frac{\log \frac{2}{\delta}}{2n}}
$$

$$
+ \frac{k}{\rho} \mathcal{R}_{m,Q}(\Pi_\mathcal{H} \mathcal{F}) + \sqrt{\frac{\log \frac{2}{\delta}}{2m}}.
$$

(19)
Theorem (Generalization Bound with Covering Numbers)

Let $\mathcal{F} \subseteq \mathbb{R}^{X \times Y}$ be a hypothesis set with label set $Y = \{1, \cdots, k\}$ and $\mathcal{H} \subseteq Y^X$ be the corresponding $Y$-valued labeling function class. Suppose $\Pi_1 \mathcal{F}$ is bounded in $L_2$ by $L$. Fix $\rho > 0$. For all $\delta > 0$, with probability $1 - 3\delta$ the following inequality holds for all hypothesis $f \in \mathcal{F}$:

$$
\epsilon_Q(f) \leq \epsilon_{\hat{P}}^{(\rho)}(f) + d_{f,\mathcal{F}}^{(\rho)}(\hat{P}, \hat{Q}) + \epsilon_{\text{ideal}} + 2\sqrt{\frac{\log \frac{2}{\delta}}{2n}} + \sqrt{\frac{\log \frac{2}{\delta}}{2m}} + \frac{16k^2 \sqrt{k}}{\rho} \inf_{\epsilon \geq 0} \left\{ \epsilon + 3 \left( \frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}} \right) \left( \int_{\epsilon}^{L} \sqrt{\log \mathcal{N}_2(\tau, \Pi_1 \mathcal{F})} d\tau + \int_{\epsilon/L}^{1} \sqrt{\log \mathcal{N}_2(\tau, \Pi_1 \mathcal{H})} d\tau \right) \right\}.
$$

(20)
MDD: Margin Disparity Discrepancy

Minimax game: Adversarial learning induced by informative margin theory

\[
\begin{align*}
\min_{f, \psi} & \quad \epsilon_{\psi(\hat{P})}^{(\rho)}(f) + \epsilon_{\psi(\hat{Q})}^{(\rho)}(f^*, f) - \epsilon_{\psi(\hat{P})}^{(\rho)}(f^*, f)), \\
\max_{f'} & \quad (\epsilon_{\psi(\hat{Q})}^{(\rho)}(f', f) - \epsilon_{\psi(\hat{P})}^{(\rho)}(f', f)).
\end{align*}
\]
Transfer Model Selection

Outline

1. Transfer Learning

2. $\mathcal{H}\Delta\mathcal{H}$-Divergence
   - DAN: Deep Adaptation Network
   - DANN: Domain Adversarial Neural Network
   - MCD: Maximum Classifier Discrepancy

3. Margin Disparity Discrepancy
   - MDD: Margin Disparity Discrepancy

4. Transfer Model Selection
   - DEV: Deep Embedded Validation

5. Evaluation and Implementation
Model Selection in Domain Adaptation

- **Supervised Learning**

\[(x_1, y_1) \sim p\]

Training

\[(x_2, y_2) \sim p\]

Validation

\[(x_3, y_3) \sim p\]

Test

- **Semi-Supervised Learning (SSL)?**

- **Unsupervised Domain Adaptation (UDA)?**
**IWCV: Importance-Weighted Cross-Validation\(^6\)**

- **Covariate shift assumption**: \( P(y|\mathbf{x}) = Q(y|\mathbf{x}) \)
- **Model selection by estimating** Target Risk \( \epsilon_Q(h) = \mathbb{E}_Q[h(\mathbf{x}) \neq y] \)
- **Importance-Weighted Cross-Validation (IWCV)**
  
  \[
  \mathbb{E}_P w(\mathbf{x}) \cdot [h(\mathbf{x}) \neq y] = \mathbb{E}_P \frac{Q(\mathbf{x})}{P(\mathbf{x})} \cdot [h(\mathbf{x}) \neq y] = \mathbb{E}_Q [h(\mathbf{x}) \neq y] = \epsilon_Q(h)
  \]

- The estimation is unbiased but the **variance is unbounded**
- Density ratio is not accessible due to unknownness of \( P \) and \( Q \)

---

\(^6\) Covariate shift adaptation by importance weighted cross validation, JMLR'2007
DEV: Deep Embedded Validation

- Variance of IWCV (bounded by Rényi divergence):
  \[
  \text{Var}_{x \sim P} [w(x) \cdot [h(x) \neq y]] \leq d_{\alpha+1}(Q \| P)\epsilon_Q(h)^{1-\frac{1}{\alpha}} - \epsilon_Q(h)^2
  \]

- Density ratio \( w(x) = \frac{Q(x)}{P(x)} \) is estimated by discriminative learning.

- Feature adaptation reduces distribution discrepancy \( d_{\alpha+1}(Q \| P) \).

- Control variate explicitly reduces the variance of \( \mathbb{E}_P w(x) \cdot [h(x) \neq y] \):
  \[
  \mathbb{E}[z] = \zeta, \mathbb{E}[t] = \tau \\
  z^* = z + \eta(t - \tau) \\
  \mathbb{E}[z^*] = \mathbb{E}[z] + \eta \mathbb{E}[t - \tau] = \zeta + \eta(\mathbb{E}[t] - \mathbb{E}[\tau]) = \zeta. \\
  \text{Var}[z^*] = \text{Var}[z + \eta(t - \tau)] = \eta^2 \text{Var}[t] + 2\eta \text{Cov}(z, t) + \text{Var}[z] \\
  \min \text{Var}[z^*] = (1 - \rho_{z,t}^2)\text{Var}[z], \text{ when } \hat{\eta} = -\frac{\text{Cov}(z, t)}{\text{Var}[t]}
  \]

---

7 Learning Bounds for Importance Weighting, NeurIPS 2010
8 Discriminative learning for differing training and test distributions, ICML 2007
9 Conditional Adversarial Domain Adaptation, NeurIPS 2018
Outline

1. Transfer Learning

2. $\mathcal{H}\Delta\mathcal{H}$-Divergence
   - DAN: Deep Adaptation Network
   - DANN: Domain Adversarial Neural Network
   - MCD: Maximum Classifier Discrepancy

3. Margin Disparity Discrepancy
   - MDD: Margin Disparity Discrepancy

4. Transfer Model Selection
   - DEV: Deep Embedded Validation

5. Evaluation and Implementation
## Results

**Table: Accuracy (%) on Office-31 for unsupervised domain adaptation**

<table>
<thead>
<tr>
<th>Method</th>
<th>A → W</th>
<th>D → W</th>
<th>W → D</th>
<th>A → D</th>
<th>D → A</th>
<th>W → A</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>68.4±0.2</td>
<td>96.7±0.1</td>
<td>99.3±0.1</td>
<td>68.9±0.2</td>
<td>62.5±0.3</td>
<td>60.7±0.3</td>
<td>76.1</td>
</tr>
<tr>
<td>DAN</td>
<td>80.5±0.4</td>
<td>97.1±0.2</td>
<td>99.6±0.1</td>
<td>78.6±0.2</td>
<td>63.6±0.3</td>
<td>62.8±0.2</td>
<td>80.4</td>
</tr>
<tr>
<td>DANN</td>
<td>82.0±0.4</td>
<td>96.9±0.2</td>
<td>99.1±0.1</td>
<td>79.7±0.4</td>
<td>68.2±0.4</td>
<td>67.4±0.5</td>
<td>82.2</td>
</tr>
<tr>
<td>CDAN</td>
<td>93.0±0.2</td>
<td>98.4±0.2</td>
<td>100.0±0.0</td>
<td>89.2±0.3</td>
<td>70.2±0.4</td>
<td>69.4±0.4</td>
<td>86.7</td>
</tr>
<tr>
<td>CDAN+E</td>
<td>93.1±0.1</td>
<td><strong>98.6±0.1</strong></td>
<td><strong>100.0±0.0</strong></td>
<td>93.4±0.2</td>
<td>71.0±0.3</td>
<td>70.3±0.3</td>
<td>87.7</td>
</tr>
<tr>
<td>MDD</td>
<td><strong>94.5±0.3</strong></td>
<td>98.4±0.1</td>
<td><strong>100.0±0.0</strong></td>
<td><strong>93.5±0.2</strong></td>
<td><strong>74.6±0.3</strong></td>
<td><strong>72.2±0.1</strong></td>
<td><strong>88.9</strong></td>
</tr>
</tbody>
</table>

**Table: Accuracy (%) on Office-Home for unsupervised domain adaptation**

| Method     | 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-50  | 34.9  | 50.0  | 58.0  | 37.4  | 41.9  | 46.2  | 38.5  | 31.2  | 60.4  | 53.9  | 41.2  | 59.9  | 46.1  |
| DAN        | 43.6  | 57.0  | 67.9  | 45.8  | 56.5  | 60.4  | 44.0  | 43.6  | 67.7  | 63.1  | 51.5  | 74.3  | 56.3  |
| DANN       | 45.6  | 59.3  | 70.1  | 47.0  | 58.5  | 60.9  | 46.1  | 43.7  | 68.5  | 63.2  | 51.8  | 76.8  | 57.6  |
| CDAN       | 50.7  | 70.6  | 76.0  | 57.6  | 70.0  | 70.0  | 57.4  | 50.9  | 77.3  | 70.9  | 56.7  | 81.6  | 65.8  |
| MDD        | **54.9** | **73.7** | **77.8** | **60.0** | **71.4** | **71.8** | **61.2** | **53.6** | **78.1** | **72.5** | **60.2** | **82.3** | **68.1** |
## Results

**Table: Accuracy (%) on VisDA-2017 (ResNet-50)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Synthetic → Real</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCD</td>
<td>69.2</td>
</tr>
<tr>
<td>GTA</td>
<td>69.5</td>
</tr>
<tr>
<td>CDAN</td>
<td>70.0</td>
</tr>
<tr>
<td>MDD</td>
<td><strong>74.6</strong></td>
</tr>
</tbody>
</table>

**Table: Accuracy (%) of MCD by different validation methods on VisDA-2017**

<table>
<thead>
<tr>
<th>Method</th>
<th>plane</th>
<th>bicyl</th>
<th>bus</th>
<th>car</th>
<th>horse</th>
<th>knife</th>
<th>mcycl</th>
<th>person</th>
<th>plant</th>
<th>sktbrd</th>
<th>train</th>
<th>truck</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>87.00</td>
<td>60.90</td>
<td>83.70</td>
<td>64.00</td>
<td>88.90</td>
<td>79.60</td>
<td>84.70</td>
<td>76.90</td>
<td>88.60</td>
<td>40.30</td>
<td>83.00</td>
<td>25.80</td>
<td>71.90</td>
</tr>
<tr>
<td>Source Risk</td>
<td>84.39</td>
<td>54.11</td>
<td>69.15</td>
<td>46.37</td>
<td>80.49</td>
<td>80.45</td>
<td>85.04</td>
<td>65.24</td>
<td>87.22</td>
<td>36.86</td>
<td>78.04</td>
<td>28.91</td>
<td>66.36</td>
</tr>
<tr>
<td>IWCV</td>
<td>81.21</td>
<td>60.95</td>
<td>76.00</td>
<td>56.53</td>
<td>82.83</td>
<td>72.06</td>
<td>84.05</td>
<td>68.65</td>
<td>86.85</td>
<td>44.37</td>
<td>69.29</td>
<td>23.81</td>
<td>67.22</td>
</tr>
<tr>
<td><strong>DEV (w/o control variate)</strong></td>
<td>84.21</td>
<td>63.95</td>
<td>79.00</td>
<td>59.53</td>
<td>85.83</td>
<td>75.06</td>
<td>87.05</td>
<td>71.65</td>
<td>89.85</td>
<td>47.37</td>
<td>72.29</td>
<td>26.81</td>
<td>70.22</td>
</tr>
<tr>
<td><strong>DEV</strong></td>
<td>81.83</td>
<td>53.48</td>
<td>82.95</td>
<td>71.62</td>
<td>89.16</td>
<td>72.03</td>
<td>89.36</td>
<td>75.73</td>
<td>97.02</td>
<td>55.48</td>
<td>71.19</td>
<td>29.17</td>
<td>72.42</td>
</tr>
<tr>
<td>Target Risk (Upper Bound)</td>
<td>81.95</td>
<td>53.60</td>
<td>83.07</td>
<td>72.02</td>
<td>89.25</td>
<td>72.15</td>
<td>89.55</td>
<td>75.83</td>
<td>97.10</td>
<td>55.57</td>
<td>71.19</td>
<td>29.27</td>
<td>72.55</td>
</tr>
</tbody>
</table>
Transfer Learning Systems

Tsinghua Dataway Big Data Software Stack

<table>
<thead>
<tr>
<th>企业数据应用软件</th>
<th>工程机械</th>
<th>能源</th>
<th>电子制造</th>
<th>气象</th>
<th>遥感</th>
<th>环保</th>
</tr>
</thead>
<tbody>
<tr>
<td>中车四方所</td>
<td>金风科技</td>
<td>英业达集团</td>
<td>联想集团</td>
<td>国家气象局</td>
<td>中科遥感所</td>
<td>福建环保厅</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>领域大数据平台软件</th>
<th>PAS</th>
<th>KStone</th>
<th>iCast</th>
<th>MDFS</th>
<th>BDIPS</th>
<th>DATA-X</th>
</tr>
</thead>
<tbody>
<tr>
<td>工业大数据平台</td>
<td>气象大数据平台</td>
<td>环保大数据平台</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>大数据系统软件</th>
<th>大数据系统软件</th>
<th>大数据系统软件</th>
<th>大数据系统软件</th>
<th>大数据系统软件</th>
<th>大数据系统软件</th>
<th>大数据系统软件</th>
</tr>
</thead>
<tbody>
<tr>
<td>核心构件</td>
<td>Proto</td>
<td>Flok</td>
<td>Quality</td>
<td>Xlearn</td>
<td>Vis</td>
<td>DEV</td>
</tr>
</tbody>
</table>
| 计算框架 | Storm | Spark | HadoopMR | Tensorflow | PyTorch | ...
| 数据存储 | TsFile/IoTDB | Cassandra | HDFS | PostgreSQL | Kafka | ...
| 异构硬件 | 边缘设备 | 私有云 | 公有云 |

清华数为框架DWF 组件元数据

DWF-Enterprise-Application
企业应用
DWF-Domain-Application
领域支撑组件
DWF-Application-Foundation
应用基础组件
DWF-Optional-Component
大数据可选组件
DWF-Essential-Component
大数据基础组件
National Engineering Lab for Big Data Software

Jiaguang Sun (孙家广)
Director
Tsinghua University

Jianmin Wang (王建民)
Deputy Director
Dean, School of Software
Tsinghua University

Michael I. Jordan
Academic Committee Chair
UC Berkeley

Machine Learning Group @ NELBDS

Mingsheng Long (龙明盛)
Yuchen Zhang
Yue Cao
Han Zhu
Zhangjie Cao