Transfer Learning
Generalizing Deep Learning across Domains and Tasks

Mingsheng Long

School of Software
National Engineering Lab for Big Data Software
Tsinghua University

https://github.com/thuml
Chinese Conference on Pattern Recognition and Computer Vision
PRCV 2018
Transfer Learning

1. Transfer Learning

2. Problem I: \( P(X) \neq Q(X) \)
   - DAN: Deep Adaptation Network

3. Problem II: \( P(Y|X) \neq Q(Y|X) \)
   - CDAN: Conditional Domain Adversarial Network

4. Theoretical Analysis

5. Benchmarking
Learner: $f : x \rightarrow y$  

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

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

- Machine learning across domains of **Non-IID distributions** $P \neq Q$
- How to design models that effectively bound the **generalization error**?

Source Domain

2D Renderings

Real Images

**Model**

$f : x \rightarrow y$

**Representation**

**Model**

$f : x \rightarrow y$

$P(x, y) \neq Q(x, y)$
# 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><strong>Training Error high?</strong></td>
<td>Optimal Bayes Rate</td>
<td>Bias</td>
<td>Deeper Model</td>
</tr>
<tr>
<td>No</td>
<td>Yes</td>
<td></td>
<td>Longer Training</td>
</tr>
<tr>
<td><strong>Train-Dev Error high?</strong></td>
<td></td>
<td>Variance</td>
<td>Bigger Data</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>Yes</td>
<td>Regularization</td>
</tr>
<tr>
<td><strong>Dev Error high?</strong></td>
<td></td>
<td>Dataset Shift</td>
<td>Transfer Learning</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td>Yes</td>
<td>Data Generation</td>
</tr>
<tr>
<td><strong>Test Error high?</strong></td>
<td></td>
<td>Overfit Dev Set</td>
<td>Bigger Dev Data</td>
</tr>
<tr>
<td>No</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Done!</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Basic Approaches to Transfer Learning

Matching distributions across source and target domains s.t. $P \approx Q$

- Reduce **marginal** distribution mismatch: $P(X) \neq Q(X)$
- Reduce **conditional** distribution mismatch: $P(Y|X) \neq Q(Y|X)$
- Challenge: fail to align different domains of multimodal distributions

Basic Guidelines to Algorithm Design

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

—Albert Einstein
Outline

1. Transfer Learning

2. Problem I: $P(X) \neq Q(X)$
   - DAN: Deep Adaptation Network

3. Problem II: $P(Y|X) \neq Q(Y|X)$
   - CDAN: Conditional Domain Adversarial Network

4. Theoretical Analysis

5. Benchmarking
**Problem I:** \( P(X) \neq Q(X) \)

**DAN: Deep Adaptation Network\(^1\)**

Deep adaptation: match distributions in multiple domain-specific layers

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

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

\[
\min_{\theta \in \Theta} \max_{k \in \mathcal{K}} \frac{1}{n_a} \sum_{i=1}^{n_a} J (\theta (x_i^a), y_i^a) + \lambda \sum_{\ell=l_1}^{l_2} d_k^2 (D_s^\ell, D_t^\ell)\]

\(^1\)Long et al. Learning Transferable Features with Deep Adaptation Networks. ICML '15.
Problem I: \( P(X) \neq Q(X) \)

**DAN: MK-MMD**

Multiple Kernel Maximum Mean Discrepancy (MK-MMD)

RKHS distance between *kernel embeddings* of distributions \( P_X \) and \( Q_X \)

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

(3)

\( k(x^s, x^t) = \langle \phi(x^s), \phi(x^t) \rangle \) is a convex combination of \( m \) PSD kernels

\[
\mathcal{K} \triangleq \left\{ k = \sum_{u=1}^{m} \beta_u k_u : \sum_{u=1}^{m} \beta_u = 1, \beta_u \geq 0, \forall u \right\}.
\]

(4)

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

- \( P = Q \) if and only if \( d^2_k(P, Q) = 0 \) (In practice, \( d^2_k(P, Q) < \epsilon \))
- \( \max_{k \in \mathcal{K}} d^2_k(P, Q) \sigma_k^{-2} \iff \min \text{ Type II Error} \ (d^2_k(P, Q) < \epsilon \text{ when } P \neq Q) \)
**DAN: Feature Learning**

Linear-Time Algorithm of MK-MMD (Streaming Algorithm)

\[ O(n^2): \quad d_k^2(p, q) = \mathbb{E}_{x^s, x'^s} k(x^s, x'^s) + \mathbb{E}_{x^t, x'^t} k(x^t, x'^t) - 2\mathbb{E}_{x^s, x^t} k(x^s, x^t) \]

\[ O(n): \quad d_k^2(p, q) = \frac{2}{n_s} \sum_{i=1}^{n_s/2} g_k(z_i) \rightarrow \text{linear-time unbiased estimate} \]

- **Quad-tuple** \( z_i \triangleq (x^s_{2i-1}, x^s_{2i}, x^t_{2i-1}, x^t_{2i}) \)
- \( g_k(z_i) \triangleq k(x^s_{2i-1}, x^s_{2i}) + k(x^t_{2i-1}, x^t_{2i}) - k(x^s_{2i-1}, x^t_{2i}) - k(x^s_{2i}, x^t_{2i-1}) \)

**Stochastic Gradient Descent (SGD)**

For each layer \( \ell \) and for each quad-tuple \( z_i^\ell = (h^s_{2i-1}, h^s_{2i}, h^t_{2i-1}, h^t_{2i}) \)

\[
\nabla \Theta^\ell = \frac{\partial J(z_i)}{\partial \Theta^\ell} + \lambda \frac{\partial g_k(z_i^\ell)}{\partial \Theta^\ell}
\]
Problem I: $P(X) \neq Q(X)$

DAN: Deep Adaptation Network

**DAN: Kernel Learning**

Learning optimal kernel $k = \sum_{u=1}^{m} \beta_u k_u$

Maximizing test power $\triangleq$ minimizing Type II error (Gretton et al. 2012)

$$
\max_{k \in \mathcal{K}} d_k^2 \left( D_s^\ell, D_t^\ell \right) \sigma_k^{-2},
$$

(6)

where $\sigma_k^2 = \mathbb{E} z g_k^2 (z) - [\mathbb{E} z g_k (z)]^2$ is the estimation variance.

Quadratic Program (QP), scaling linearly to sample size: $O(m^2 n + m^3)$

$$
\min_{d^T \beta = 1, \beta \succeq 0} \beta^T \left( Q + \epsilon I \right) \beta,
$$

(7)

where $d = (d_1, d_2, \ldots, d_m)^T$, and each $d_u$ is MMD using base kernel $k_u$. 
**DANN**: Domain Adversarial Neural Network\(^2\)

Adversarial adaptation: learning features indistinguishable across domains

\[
E(\theta_f, \theta_y, \theta_d) = \sum_{x_i \in D_s} L_y(G_y(G_f(x_i)), y_i) - \lambda \sum_{x_i \in D_s \cup D_t} L_d(G_d(G_f(x_i)), d_i)
\]  
(8)

\[
(\hat{\theta}_f, \hat{\theta}_y) = \arg \min_{\theta_f, \theta_y} E(\theta_f, \theta_y, \theta_d) \quad (\hat{\theta}_d) = \arg \max_{\theta_d} E(\theta_f, \theta_y, \theta_d)
\]  
(9)

\(^2\) Ganin et al. *Domain Adversarial Training of Neural Networks*. JMLR '16.
Outline

1 Transfer Learning

2 Problem I: $P(X) \neq Q(X)$
   - DAN: Deep Adaptation Network

3 Problem II: $P(Y|X) \neq Q(Y|X)$
   - CDAN: Conditional Domain Adversarial Network

4 Theoretical Analysis

5 Benchmarking
**CDAN: Conditional Domain Adversarial Network**

Main Idea of This Work: Distribution Embeddings with Statistics

- **Capture** cross-covariance statistics across multiple random vectors
- **Concatenation**: \( \mathbb{E}_{XY}[X \oplus Y] = \mathbb{E}_X[X] \oplus \mathbb{E}_Y[Y] \)
- **Multilinear**: \( \mathbb{E}_{XY}[X \otimes Y] = \mathbb{E}_X[X|Y = 1] \oplus \ldots \oplus \mathbb{E}_X[X|Y = C] \)

<table>
<thead>
<tr>
<th>Distributions</th>
<th>Probabilistic Operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discrete</td>
<td></td>
</tr>
<tr>
<td>( P(X) )</td>
<td>( \mathbb{E}_X[\phi(X)] )</td>
</tr>
<tr>
<td>( P(X, Y) )</td>
<td>( \mathbb{E}_{XY}[\phi(X) \otimes \phi(Y)] )</td>
</tr>
<tr>
<td>( P(X, Y, Z) )</td>
<td>( \mathbb{E}_{XYZ}[\phi(X) \otimes \phi(Y) \otimes \phi(Z)] )</td>
</tr>
<tr>
<td>Kernel</td>
<td></td>
</tr>
<tr>
<td>Embedding</td>
<td></td>
</tr>
<tr>
<td>( P(X) )</td>
<td>( \mathbb{E}_X[\phi(X)] )</td>
</tr>
<tr>
<td>( P(X, Y) )</td>
<td>( \mathbb{E}_{XY}[\phi(X) \otimes \phi(Y)] )</td>
</tr>
<tr>
<td>( P(X, Y, Z) )</td>
<td>( \mathbb{E}_{XYZ}[\phi(X) \otimes \phi(Y) \otimes \phi(Z)] )</td>
</tr>
</tbody>
</table>

\[ \text{Problem II: } P(Y|X) \neq Q(Y|X) \]

---

**CDAN: Multilinear Conditioning**

Conditional adaptation of distributions over representation & prediction

\[
\min_G E_G - \lambda E_{D,G}
\]

\[
\min_D E_{D,G}
\]

\[
E_{D,G} = -\frac{1}{n_s} \sum_{i=1}^{n_s} \log (D(f^s_i \otimes g^s_i)) - \frac{1}{n_t} \sum_{j=1}^{n_t} \log (1 - D(f^t_j \otimes g^t_j))
\]
Problem II: $P(Y|X) \neq Q(Y|X)$

CDAN: Conditional Domain Adversarial Network

CDAN: Randomized Multilinear Conditioning

Conditional adaptation of distributions over representation & prediction

\[
T \otimes (f, g) = f \otimes g
\]

\[
T \circ (f, g) = \frac{1}{\sqrt{d}} (R_f f) \circ (R_g g)
\]

\[
T(h) = \begin{cases}
T \otimes (f, g) & \text{if } d_f \times d_g \leq 4096 \\
T \circ (f, g) & \text{otherwise}
\end{cases}
\]
CDAN: Entropy Conditioning

Control the uncertainty of classifier prediction to guarantee transferability

\[
\max_D \frac{1}{n_s} \sum_{i=1}^{n_s} e^{-H(g^s_i)} \log [D(T(h^s_i))] + \frac{1}{n_t} \sum_{j=1}^{n_t} e^{-H(g^t_j)} \log [1 - D(T(h^t_j))] 
\]

(15)
CDAN: Minimax Optimization Problem

Principled approaches: Conditional Domain Adversarial Networks (CDAN)

- **Multilinear Conditioning**: capture the cross-covariance between feature representation & classifier prediction to boost discriminability
- **Entropy Conditioning**: control the uncertainty of classifier prediction to guarantee transferability (entropy minimization principle)

\[
\min_G \frac{1}{n_s} \sum_{i=1}^{n_s} L(G(x_i^s), y_i^s) \\
+ \frac{\lambda}{n_s} \sum_{i=1}^{n_s} e^{-H(g_i^s)} \log [D(T(h_i^s))] + \frac{\lambda}{n_t} \sum_{j=1}^{n_t} e^{-H(g_j^t)} \log [1 - D(T(h_j^t))] \\
\]

\[
\max_D \frac{1}{n_s} \sum_{i=1}^{n_s} e^{-H(g_i^s)} \log [D(T(h_i^s))] + \frac{1}{n_t} \sum_{j=1}^{n_t} e^{-H(g_j^t)} \log [1 - D(T(h_j^t))] \\
\] (16)

\[
\] (17)
Outline

1 Transfer Learning

2 Problem I: $P(X) \neq Q(X)$
   ◦ DAN: Deep Adaptation Network

3 Problem II: $P(Y|X) \neq Q(Y|X)$
   ◦ CDAN: Conditional Domain Adversarial Network

4 Theoretical Analysis

5 Benchmarking
Notations and Assumptions

- Source risk: $\epsilon_P (G) = \mathbb{E}_{(f,y) \sim P} [G (f) \neq y]$
- Target risk: $\epsilon_Q (G) = \mathbb{E}_{(f,y) \sim Q} [G (f) \neq y]$
- Disagreement on source: $\epsilon_P (G_1, G_2) = \mathbb{E}_{(f,y) \sim P} [G_1 (f) \neq G_2 (f)]$
- Disagreement on target: $\epsilon_Q (G_1, G_2) = \mathbb{E}_{(f,y) \sim Q} [G_1 (f) \neq G_2 (f)]$
- Idea hypothesis: $G^* = \arg\min_G \epsilon_P (G) + \epsilon_Q (G)$
- Assumption: idea hypothesis has small risk $\epsilon_{ideal} = \epsilon_P (G^*) + \epsilon_Q (G^*)$
Generalization Bound

Theorem

The probabilistic bound of the target risk $\epsilon_Q(G)$ of hypothesis $G$ is given by the source risk $\epsilon_P(G)$ plus the distribution discrepancy:

$$
\epsilon_Q(G) \leq \epsilon_P(G) + [\epsilon_P(G^*) + \epsilon_Q(G^*)] + |\epsilon_P(G, G^*) - \epsilon_Q(G, G^*)| \quad (18)
$$

Proof.

By using the triangle inequalities, we have

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

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

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

$$
\leq \epsilon_P(G) + [\epsilon_P(G^*) + \epsilon_Q(G^*)] + |\epsilon_P(G, G^*) - \epsilon_Q(G, G^*)| \quad (19)
$$
Joint Distribution Discrepancy

Define the proxies of the joint distributions $P(x, y)$ and $Q(x, y)$

- $P_G = (f, G(f))_{f \sim P(f)}$, $Q_G = (f, G(f))_{f \sim Q(f)}$
- $\epsilon_P (G, G^*) = \epsilon_{P_G}(G^*)$, $\epsilon_Q (G, G^*) = \epsilon_{Q_G}(G^*)$

**Proof.**

$\epsilon_P (G, G^*) = \mathbb{E}_{(f,y) \sim P} [G(f) \neq G^*(f)] = \mathbb{E}_{(f,g) \sim P_G} [g \neq G^*(f)] = \epsilon_{P_G}(G^*)$

How to bound the distribution discrepancy $|\epsilon_P (G, G^*) - \epsilon_Q (G, G^*)|$?
The distribution discrepancy $|\epsilon_P (G, G^*) - \epsilon_Q (G, G^*)|$ is bounded by

$$|\epsilon_P (G, G^*) - \epsilon_Q (G, G^*)| = |\mathbb{E}_{(f,g) \sim P_G} [g \neq G^*(f)] - \mathbb{E}_{(f,g) \sim Q_G} [g \neq G^*(f)]|$$

$$\leq \sup_{G^* \in \mathcal{H}} |\mathbb{E}_{(f,g) \sim P_G} [|g - G^*(f)| \neq 0] - \mathbb{E}_{(f,g) \sim Q_G} [|g - G^*(f)| \neq 0]|$$

$$\leq \sup_{\delta \in \Delta} |\mathbb{E}_{(f,g) \sim P_G} [\delta (f, g) \neq 0] - \mathbb{E}_{(f,g) \sim Q_G} [\delta (f, g) \neq 0]|$$

$$\leq \sup_{D \in \mathcal{H}_D} |\mathbb{E}_{(f,g) \sim P_G} [D (f, g) \neq 0] - \mathbb{E}_{(f,g) \sim Q_G} [D (f, g) \neq 0]|$$

The upper-bound can be yielded by training the domain discriminator $D$. 

---

**Distribution discrepancy**

**Hypothesis-based distribution discrepancy**
Outline

1. Transfer Learning
2. Problem I: $P(X) \neq Q(X)$
   - DAN: Deep Adaptation Network
3. Problem II: $P(Y|X) \neq Q(Y|X)$
   - CDAN: Conditional Domain Adversarial Network
4. Theoretical Analysis
5. Benchmarking
Datasets

- Benchmarking
- Pre-train
- Fine-tune
- VisDA Challenge 2017
- Fine-tune
- Office-Caltech
- OfficeHome
- Mingsheng Long
- Transfer Learning
- May 18, 2019 27 / 31
### Table: Accuracy (%) on Office-31 for unsupervised domain adaptation

<table>
<thead>
<tr>
<th>Method</th>
<th>$A \rightarrow W$</th>
<th>$D \rightarrow W$</th>
<th>$W \rightarrow D$</th>
<th>$A \rightarrow D$</th>
<th>$D \rightarrow A$</th>
<th>$W \rightarrow A$</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>AlexNet</td>
<td>61.6±0.5</td>
<td>95.4±0.3</td>
<td>99.0±0.2</td>
<td>63.8±0.5</td>
<td>51.1±0.6</td>
<td>49.8±0.4</td>
<td>70.1</td>
</tr>
<tr>
<td>DAN</td>
<td>68.5±0.5</td>
<td>96.0±0.3</td>
<td>99.0±0.3</td>
<td>67.0±0.4</td>
<td>54.0±0.5</td>
<td>53.1±0.5</td>
<td>72.9</td>
</tr>
<tr>
<td>RTN</td>
<td>73.3±0.3</td>
<td>96.8±0.2</td>
<td>99.6±0.1</td>
<td>71.0±0.2</td>
<td>50.5±0.3</td>
<td>51.0±0.1</td>
<td>73.7</td>
</tr>
<tr>
<td>DANN</td>
<td>73.0±0.5</td>
<td>96.4±0.3</td>
<td>99.2±0.3</td>
<td>72.3±0.3</td>
<td>53.4±0.4</td>
<td>51.2±0.5</td>
<td>74.3</td>
</tr>
<tr>
<td>ADDA</td>
<td>73.5±0.6</td>
<td>96.2±0.4</td>
<td>98.8±0.4</td>
<td>71.6±0.4</td>
<td>54.6±0.5</td>
<td>53.5±0.6</td>
<td>74.7</td>
</tr>
<tr>
<td>JAN</td>
<td>74.9±0.3</td>
<td>96.6±0.2</td>
<td>99.5±0.2</td>
<td>71.8±0.2</td>
<td><strong>58.3±0.3</strong></td>
<td>55.0±0.4</td>
<td>76.0</td>
</tr>
<tr>
<td>CDAN-RM</td>
<td><strong>77.9±0.3</strong></td>
<td>96.9±0.2</td>
<td><strong>100.0±0.0</strong></td>
<td><strong>74.6±0.2</strong></td>
<td>55.1±0.3</td>
<td><strong>57.5±0.4</strong></td>
<td><strong>77.0</strong></td>
</tr>
<tr>
<td>CDAN-M</td>
<td>77.6±0.2</td>
<td><strong>97.2±0.1</strong></td>
<td><strong>100.0±0.0</strong></td>
<td>73.0±0.1</td>
<td>57.3±0.2</td>
<td>56.1±0.3</td>
<td>76.9</td>
</tr>
<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>RTN</td>
<td>84.5±0.2</td>
<td>96.8±0.1</td>
<td>99.4±0.1</td>
<td>77.5±0.3</td>
<td>66.2±0.2</td>
<td>64.8±0.3</td>
<td>81.6</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>ADDA</td>
<td>86.2±0.5</td>
<td>96.2±0.3</td>
<td>98.4±0.3</td>
<td>77.8±0.3</td>
<td>69.5±0.4</td>
<td>68.9±0.5</td>
<td>82.9</td>
</tr>
<tr>
<td>JAN</td>
<td>85.4±0.3</td>
<td>97.4±0.2</td>
<td>99.8±0.2</td>
<td>84.7±0.3</td>
<td>68.6±0.3</td>
<td>70.0±0.4</td>
<td>84.3</td>
</tr>
<tr>
<td>JAN-A</td>
<td>92.6±0.2</td>
<td>98.2±0.1</td>
<td>99.8±0.2</td>
<td>86.3±0.1</td>
<td>71.4±0.2</td>
<td>72.4±0.1</td>
<td>86.8</td>
</tr>
<tr>
<td>CDAN-RM</td>
<td>93.0±0.2</td>
<td>98.4±0.2</td>
<td><strong>100.0±0.0</strong></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-M</td>
<td><strong>93.1±0.1</strong></td>
<td><strong>98.6±0.1</strong></td>
<td><strong>100.0±0.0</strong></td>
<td><strong>93.4±0.2</strong></td>
<td><strong>71.0±0.3</strong></td>
<td><strong>70.3±0.3</strong></td>
<td><strong>87.7</strong></td>
</tr>
</tbody>
</table>
Results

VISDA-2017

<table>
<thead>
<tr>
<th>Method</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-50</td>
<td>40.13</td>
</tr>
<tr>
<td>DAN</td>
<td>62.6</td>
</tr>
<tr>
<td>DANN</td>
<td>64.5</td>
</tr>
<tr>
<td>JAN</td>
<td>66.9</td>
</tr>
<tr>
<td>CDA-N-M</td>
<td>70.8</td>
</tr>
</tbody>
</table>
Analysis

Figure: T-SNE on features by (a) ResNet, (b) DANN, (c) CDAN-f, (d) CDAN-fg.

Figure: Analysis of CDAN: (a) Conditioning, (b) Discrepancy, (c) Convergence.
Open Problems

- Conditional Shift: \( P(Y^s|X^s) \neq Q(Y^t|X^t) \)
- Simulation-to-Real: \( P(X^s_{\text{low-level}}) \neq Q(X^t_{\text{low-level}}) \)
- Open-Set/Zero-Shot (auxiliary info): \( Y^s \neq Y^t \)
- Heterogeneous (almost impossible): \( X^s \neq X^t \)
- Learning Transferable Architectures: BN, Skip-connection, etc.

\[ X_{\text{learn}} \text{ library is available: } \text{https://github.com/thuml/Xlearn} \]