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

Supervised Learning

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

fish

bird

mammal

tree

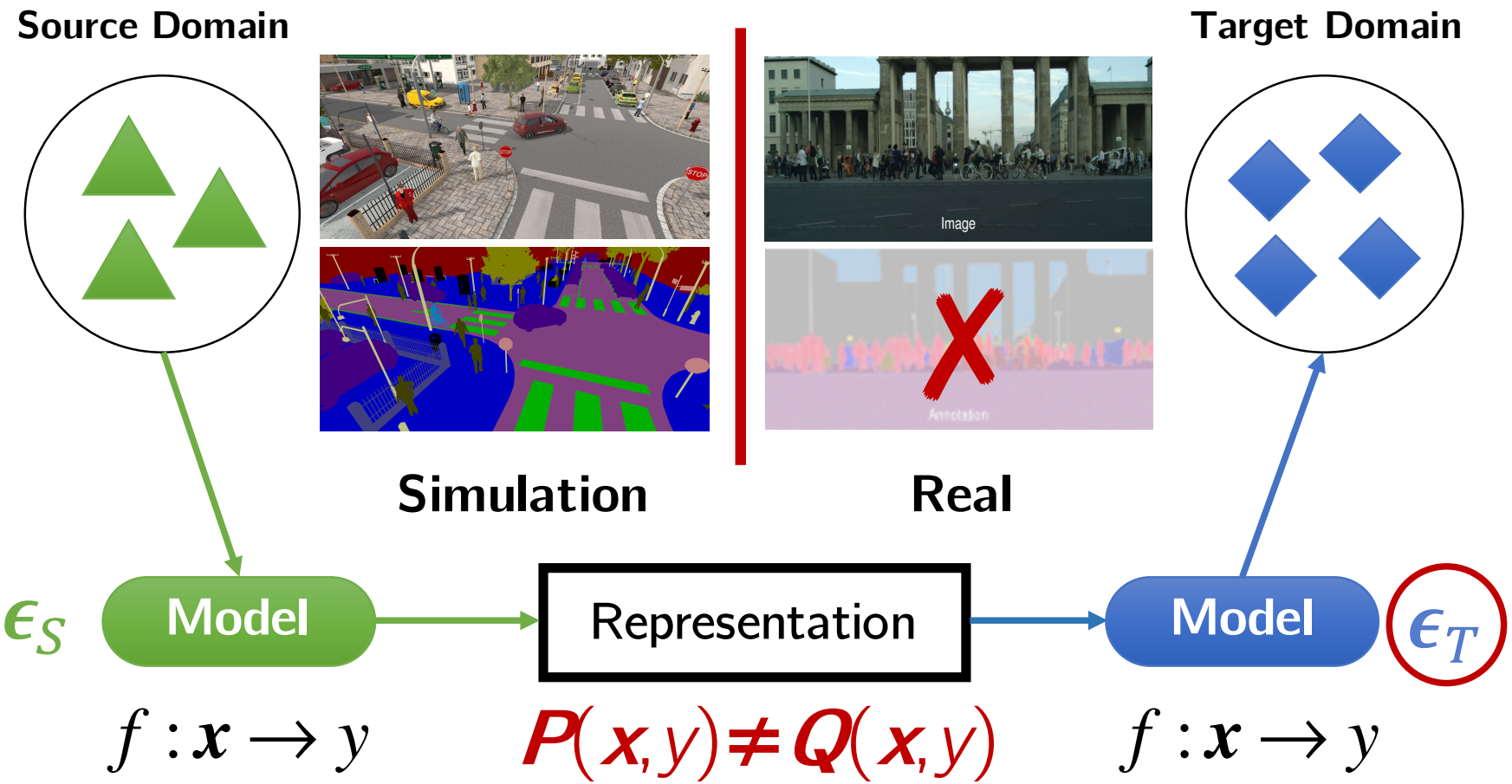
flower

.....

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

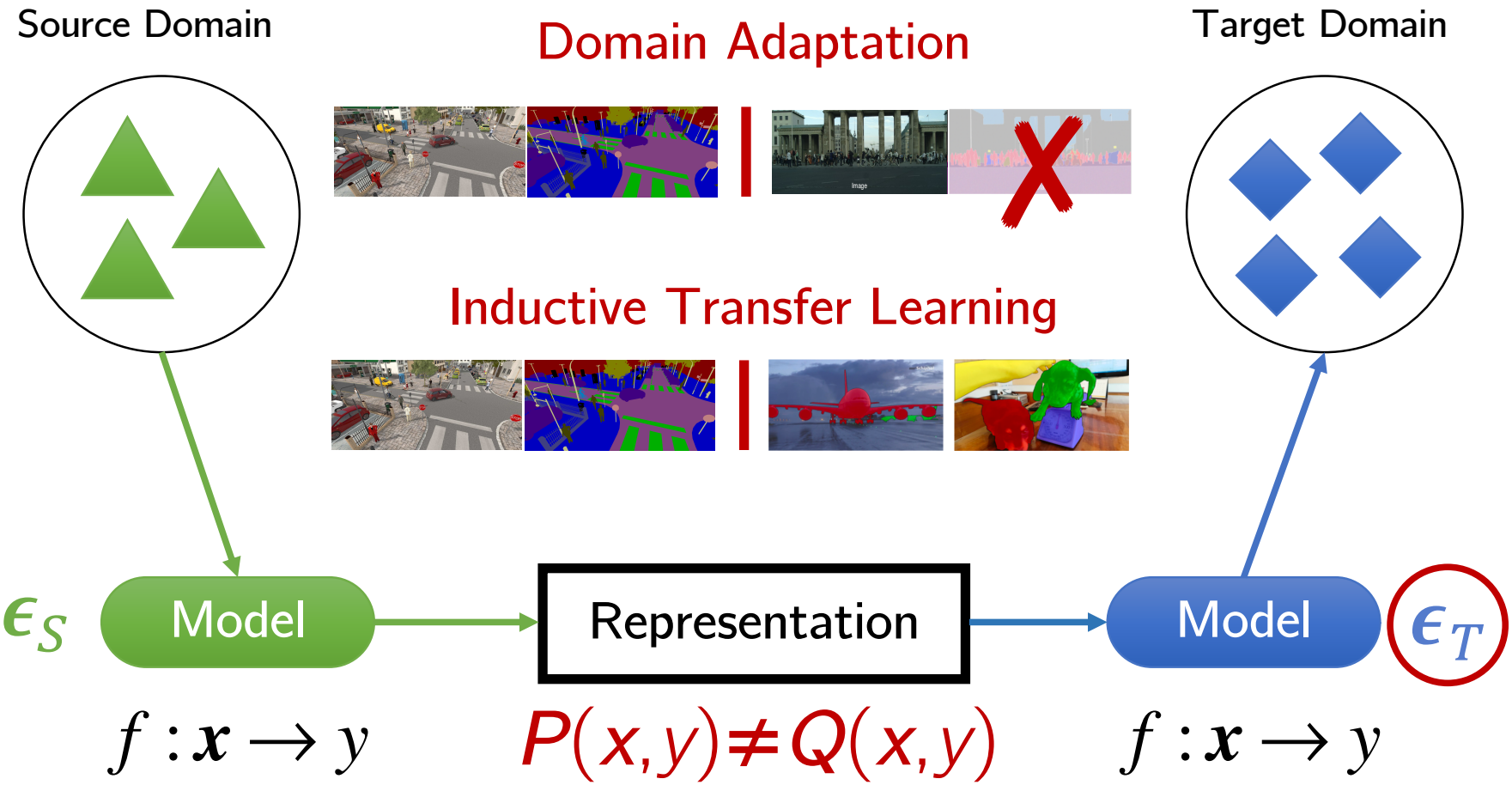
Transfer Learning

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

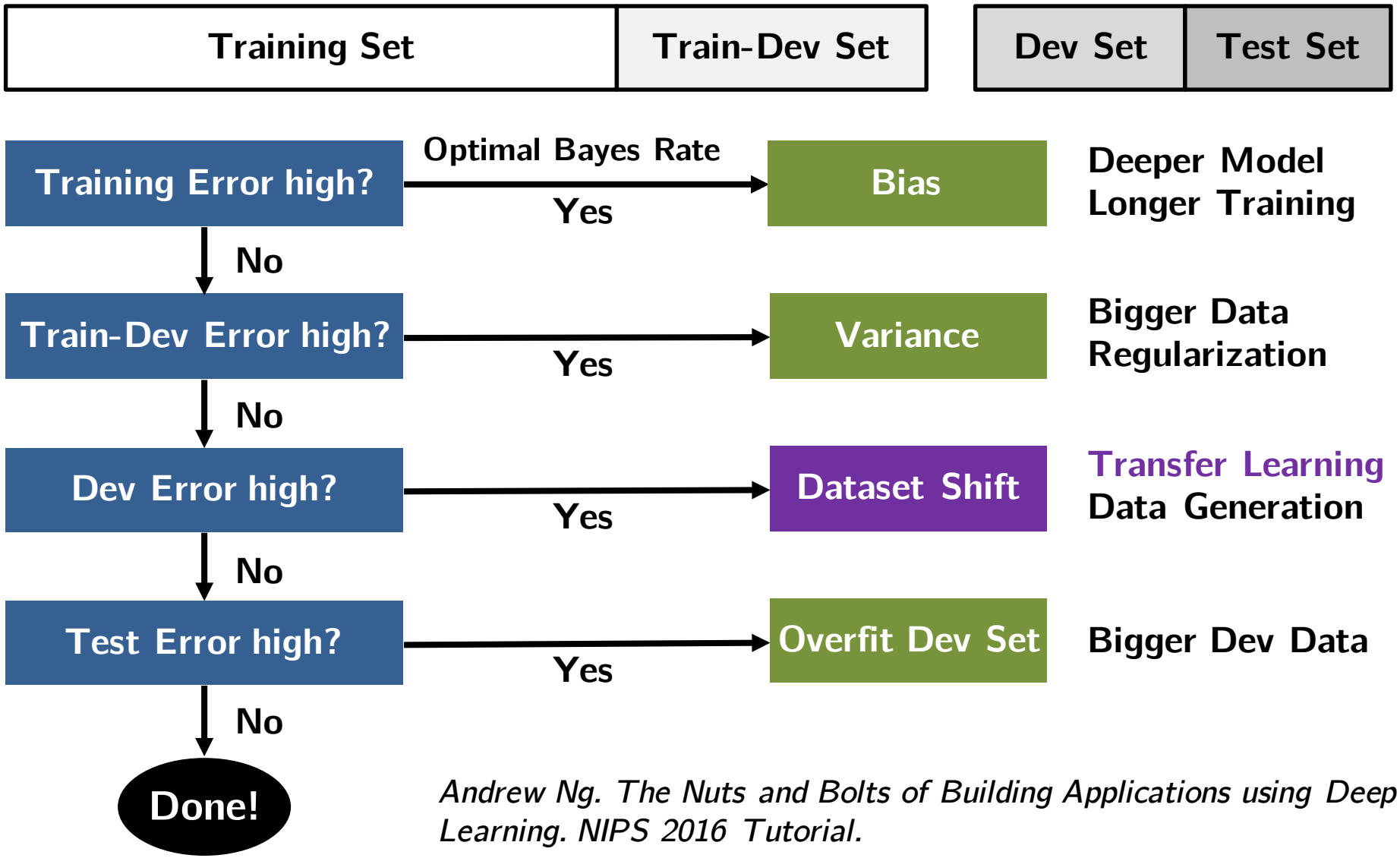


Transfer Learning

- Transfer learning setups ($P \neq Q$): Feature Space X, Label Space Y
  - **Domain Adaptation:** common X, common Y, unlabeled T
  - **Inductive Transfer Learning:** common X, different Y, labeled T

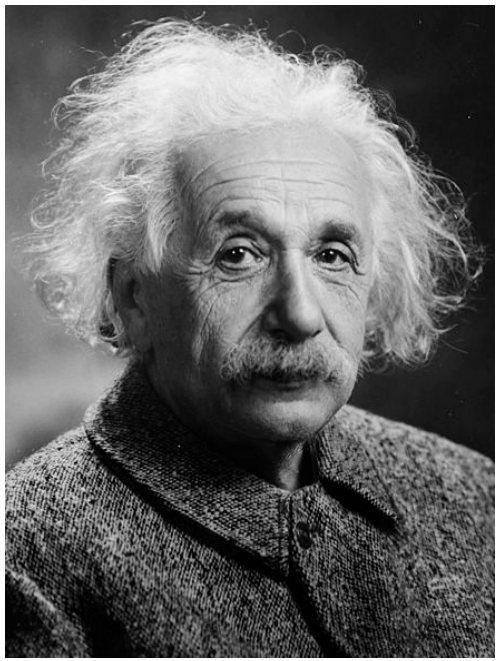


Bias-Variance-Shift Tradeoff



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

Bridging Theory and Algorithm



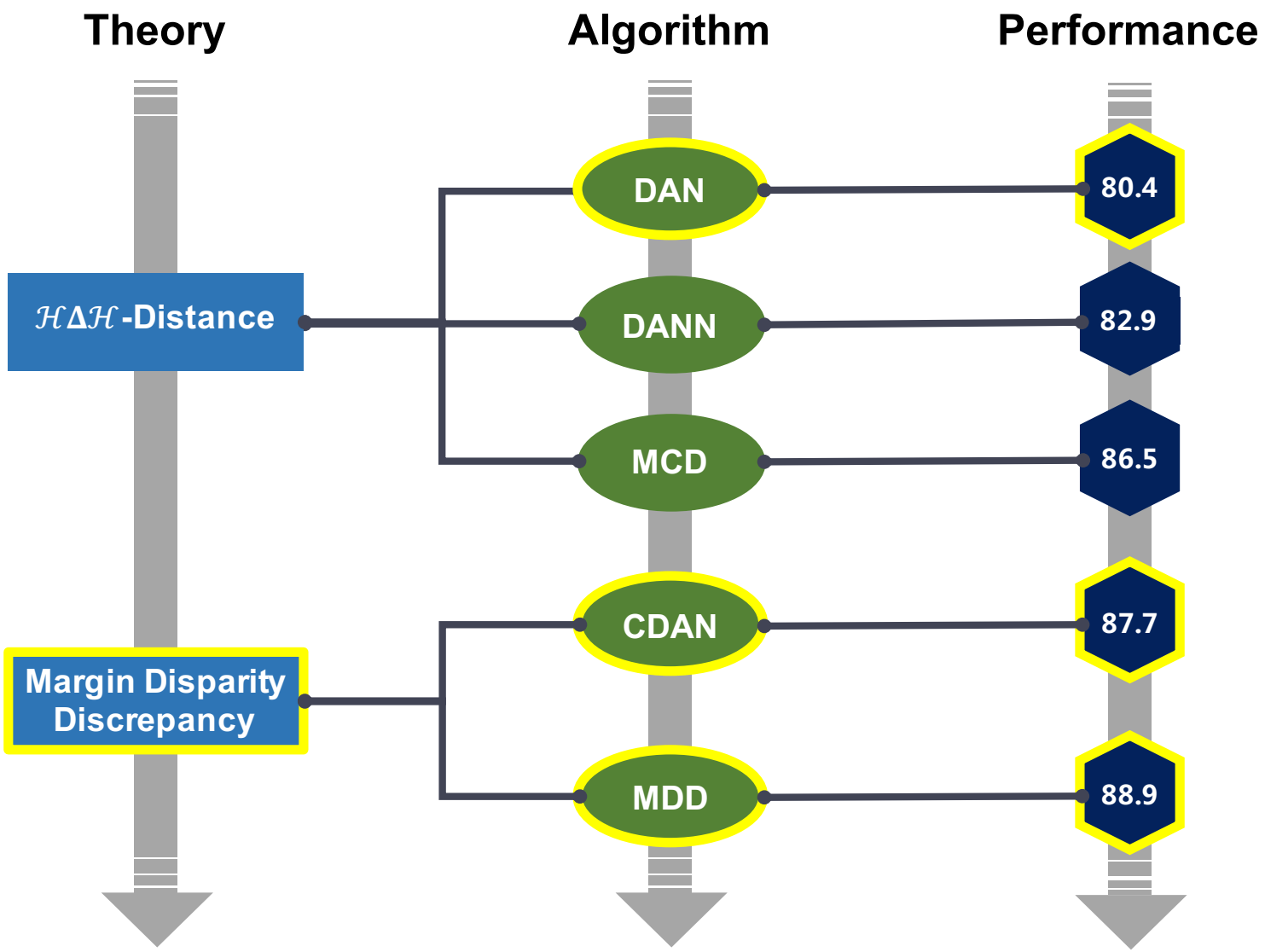
Everything should be made as simple as possible, but no simpler.

—Albert Einstein

There is nothing more practical than a good theory.

—Vladimir Vapnik

Bridging Theory and Algorithm

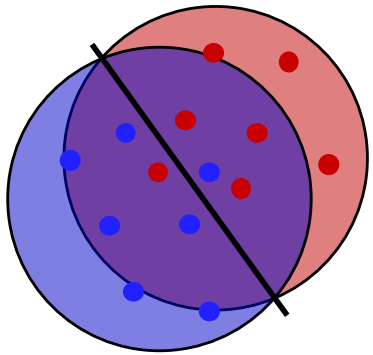


Outline

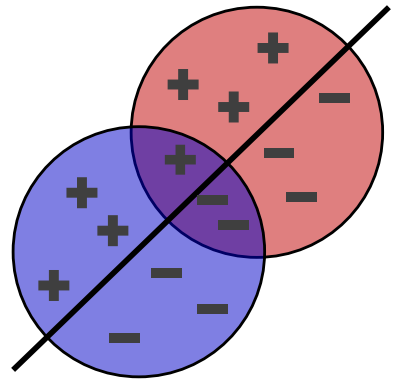
Notations and Assumptions

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

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



Distribution discrepancy



Ideal hypothesis with small error

Relating the Target Risk to the Source Risk

Theorem

Assuming small ϵ_{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:

$$\epsilon_Q(h) \leqslant \epsilon_P(h) + [\epsilon_P(h^*) + \epsilon_Q(h^*)] + |\epsilon_P(h, h^*) - \epsilon_Q(h, h^*)| \tag{1}$$

Proof.

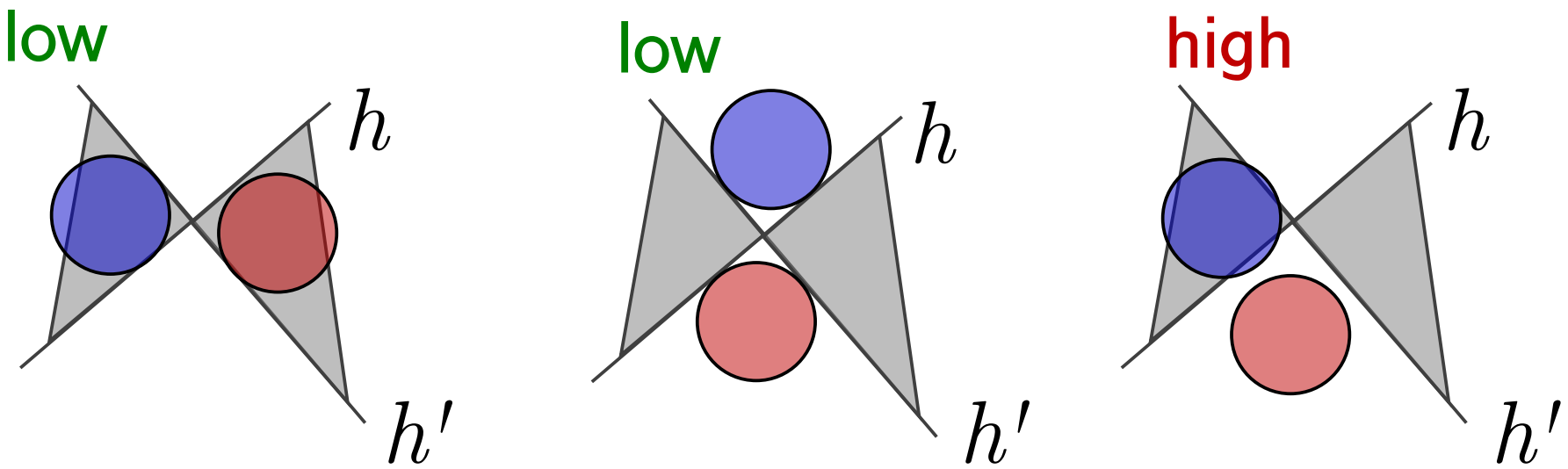
Simply by using the triangle inequalities, we have

$$\begin{aligned}
\epsilon_Q(h) &\leqslant \epsilon_Q(h^*) + \epsilon_Q(h, h^*) \\
&\leqslant \epsilon_Q(h^*) + \epsilon_P(h, h^*) + \epsilon_Q(h, h^*) - \epsilon_P(h, h^*) \\
&\leqslant \epsilon_Q(h^*) + \epsilon_P(h, h^*) + |\epsilon_Q(h, h^*) - \epsilon_P(h, h^*)| \\
&\leqslant \epsilon_P(h) + [\epsilon_P(h^*) + \epsilon_Q(h^*)] + |\epsilon_P(h, h^*) - \epsilon_Q(h, h^*)|
\end{aligned} \tag{2}$$

□

How to Bound the Disparity Difference?

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



- $\mathcal{H}\Delta\mathcal{H}$-Divergence[1]: $d_{\mathcal{H}\Delta\mathcal{H}}(P, Q) \triangleq \sup_{h,h' \in \mathcal{H}} |\epsilon_P(h, h') - \epsilon_Q(h, h')|$
- Hypothesis-independent discrepancy—depending on hypothesis space.

[1] *Ben-David et al. A Theory of Learning from Different Domains. Machine Learning, 2010.*

Generalization Bound with $\mathcal{H}\Delta\mathcal{H}$-Divergence

Theorem (Generalization Bound)

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

$$\epsilon_Q(h) \leq \epsilon_{\hat{P}}(h) + d_{\mathcal{H}\Delta\mathcal{H}}(\widehat{P}, \widehat{Q}) + \epsilon_{ideal} + O(\sqrt{\frac{d \log n}{n}} + \sqrt{\frac{d \log m}{m}}) \tag{3}$$

- $\epsilon_P(h)$ depicts the performance of h on source domain.
- $d_{\mathcal{H}\Delta\mathcal{H}}$ bounds the generalization gap caused by domain shift.
- ϵ_{ideal} quantifies the inverse of "*adaptability*" between domains.
- The order of the complexity term is $O(\sqrt{d \log n/n} + \sqrt{d \log m/m})$.

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

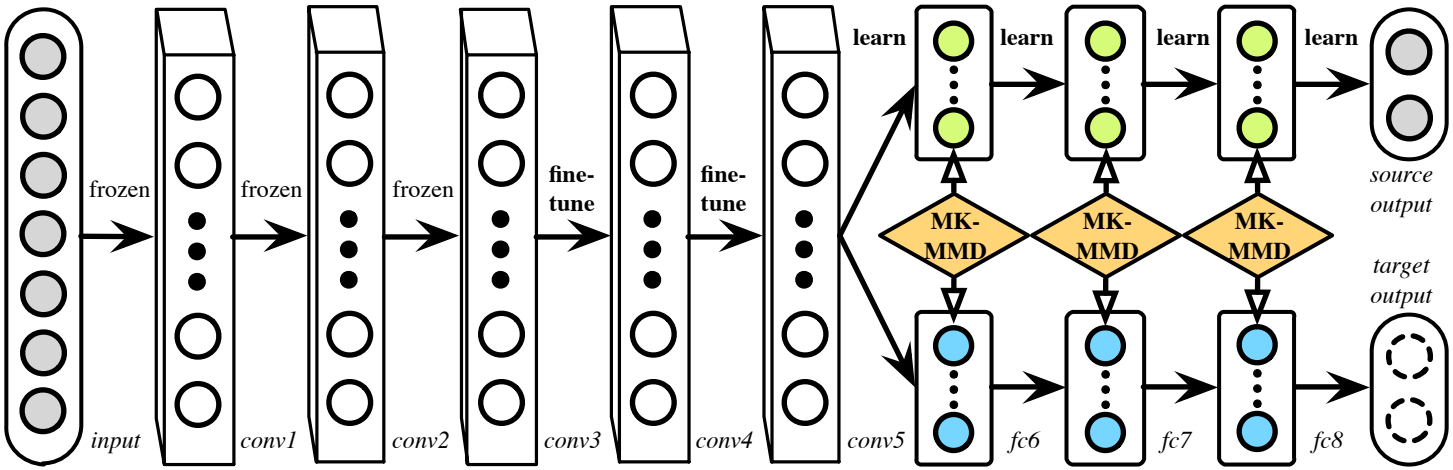
$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(P,Q) &\triangleq \sup_{h,h'\in\mathcal{H}} \left|\epsilon_P\left(h,h'\right)-\epsilon_Q\left(h,h'\right)\right| \\ &= \sup_{h,h'\in\mathcal{H}} \left|\mathbb{E}_P\left[\left|h\left(\mathbf{x}\right)-h'\left(\mathbf{x}\right)\right|\neq 0\right]-\mathbb{E}_Q\left[\left|h\left(\mathbf{x}\right)-h'\left(\mathbf{x}\right)\right|\neq 0\right]\right| \\ &= \sup_{\delta\in\mathcal{H}\Delta\mathcal{H}} \left|\mathbb{E}_P\left[\delta(\mathbf{x})\neq 0\right]-\mathbb{E}_Q\left[\delta\left(\mathbf{x}\right)\neq 0\right]\right| \end{aligned} \tag{4}$$

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

$$d_{\mathcal{F}}(P,Q)=\sup_{f\in\mathcal{F}}\left|\mathbb{E}_{\mathbf{x}\sim P}f(\mathbf{x})-\mathbb{E}_{\mathbf{x}\sim Q}f(\mathbf{x})\right| \tag{5}$$

Assuming $\mathcal{F}$ can be approximated by kernel functions in RKHS, $d_{\mathcal{F}}(P,Q)$ turns into Maximum Mean Discrepancy (MMD) (a statistical distance)

DAN: Deep Adaptation Network[2]



Distribution matching: yield the upper-bound by multiple kernel learning

$$d_k^2(P,Q) \triangleq \left\| \mathbf{E}_P\left[\phi\left(\mathbf{x}^s\right)\right] - \mathbf{E}_Q\left[\phi\left(\mathbf{x}^t\right)\right] \right\|_{\mathcal{H}_k}^2 \tag{6}$$

$$\min_{\theta\in\Theta} \max_{k\in\mathcal{K}} \frac{1}{n_a}\sum_{i=1}^{n_a} L\left(\theta\left(\mathbf{x}_i^a\right), y_i^a\right) + \lambda \sum_{\ell=l_1}^{l_2} d_k^2\left(\widehat{P}_\ell, \widehat{Q}_\ell\right) \tag{7}$$

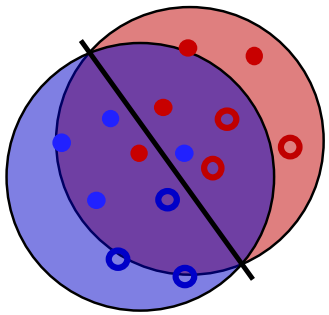
[2] *Long et al. Learning Transferable Features with Deep Adaptation Networks. ICML 2015.*

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

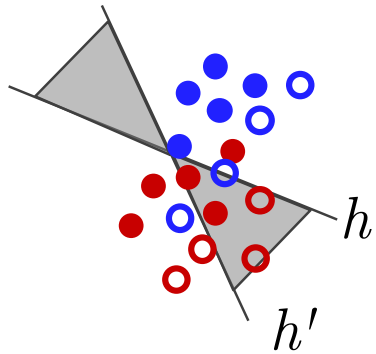
For binary hypothesis h, the $\mathcal{H}\Delta\mathcal{H}$-Divergence can be bounded by

$$\begin{aligned} d_{\mathcal{H}\Delta\mathcal{H}}(P,Q) &\triangleq \sup_{h,h'\in\mathcal{H}} \left|\epsilon_P\left(h,h'\right)-\epsilon_Q\left(h,h'\right)\right| \\ &= \sup_{\delta\in\mathcal{H}\Delta\mathcal{H}} \left|\mathbb{E}_P\left[\delta(\mathbf{x})\neq 0\right]-\mathbb{E}_Q\left[\delta\left(\mathbf{x}\right)\neq 0\right]\right| \\ &\leqslant \sup_{D\in\mathcal{H}_D} \left|\mathbb{E}_P\left[D(\mathbf{x})=1\right]+\mathbb{E}_Q\left[D\left(\mathbf{x}\right)=0\right]\right| \end{aligned} \tag{8}$$

This upper-bound can be yielded by training a domain discriminator $D(\mathbf{x})$

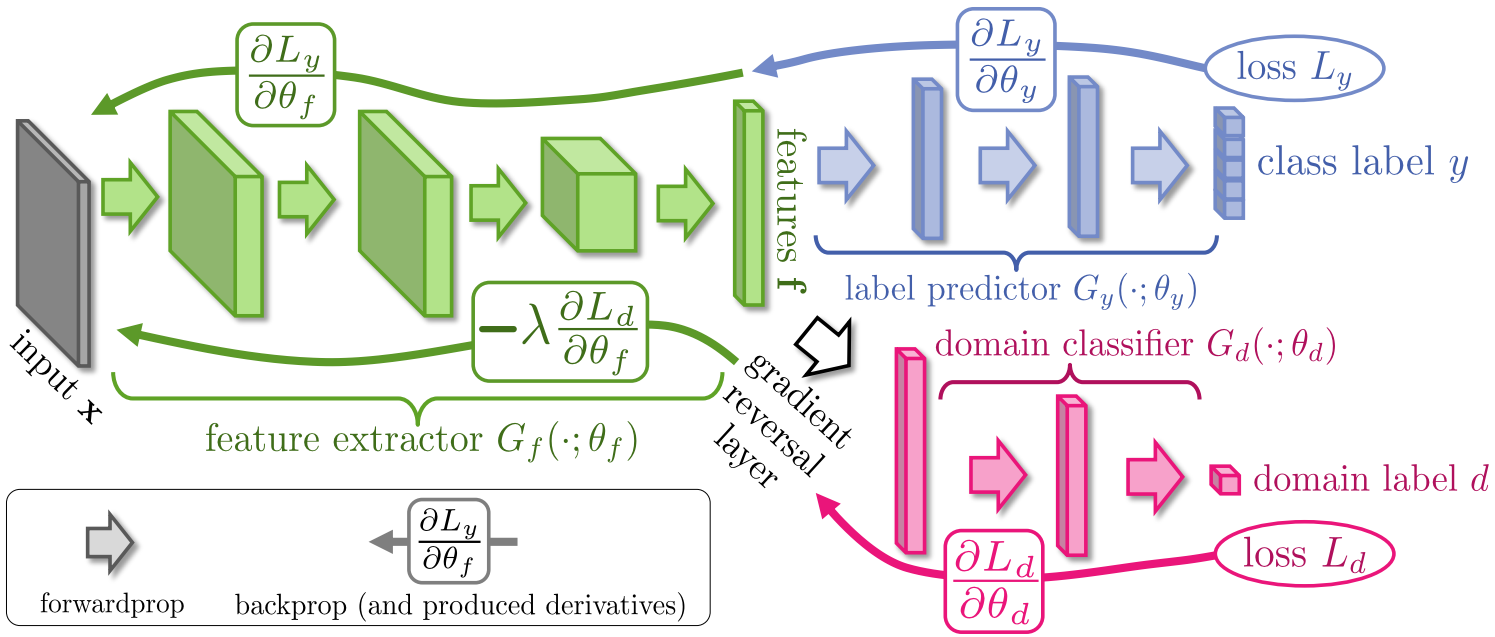


Distribution discrepancy



Hypothesis-based distribution discrepancy

DANN: Domain Adversarial Neural Network[3]



Adversarial adaptation: learning features indistinguishable across domains

$$E\left(\theta_f,\theta_y,\theta_d\right)=\sum_{\mathbf{x}_i\sim\widehat{P}} L_y\left(G_y\left(G_f\left(\mathbf{x}_i\right)\right),y_i\right)-\lambda\sum_{\mathbf{x}_i\sim\widehat{P}\cup\widehat{Q}} L_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right),d_i\right) \quad (9)$$

$$\left(\hat{\theta}_f,\hat{\theta}_y\right)=\arg\min_{\theta_f,\theta_y} E\left(\theta_f,\theta_y,\theta_d\right) \quad \left(\hat{\theta}_d\right)=\arg\max_{\theta_d} E\left(\theta_f,\theta_y,\theta_d\right) \quad (10)$$

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

Outline

Towards Informative Margin Theory

- Towards a rigorous multiclass domain adaptation theory.
  - All existing theories are only applicable to binary classification.
  - Generalization bound with scoring functions has not been studied.
- Towards an informative margin theory.
  - Explore the idea of margin in measuring domain discrepancy.
  - Generalization bound with margin loss has not been studied.
- Towards a certain function class in the theoretical bound.
  - Eliminate approximation assumptions in all existing methods.
  - Computing the supremum in previous discrepancies requires an ergodicity over $\mathcal{H}\Delta\mathcal{H}$ that increases the difficulty of optimization.
- Towards bridging the existing gap between theories and algorithms.

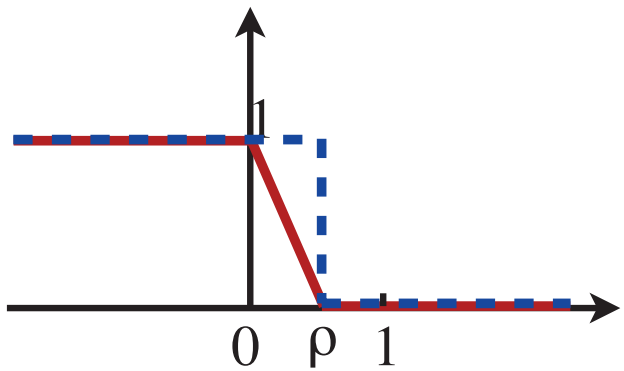
Notations

- 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:

$$\Phi_\rho(x) = \begin{cases} 0 & \rho \leqslant x \\ 1 - x/\rho & 0 \leqslant x \leqslant \rho \\ 1 & x \leqslant 0 \end{cases}$$



DD: Disparity Discrepancy

Definition (Disparity Discrepancy, DD)

Given a hypothesis space $\mathcal{H}$ and a *specific classifier* $h \in \mathcal{H}$, the Disparity Discrepancy (DD) induced by $h' \in \mathcal{H}$ is defined by

$$d_{h,\mathcal{H}}(P, Q) = \sup_{h' \in \mathcal{H}} \left| \mathbb{E}_Q[h' \neq h] - \mathbb{E}_P[h' \neq h] \right|. \tag{11}$$

The supremum in the disparity discrepancy is taken **only over the hypothesis space** $\mathcal{H}$ and thus can be optimized more easily.

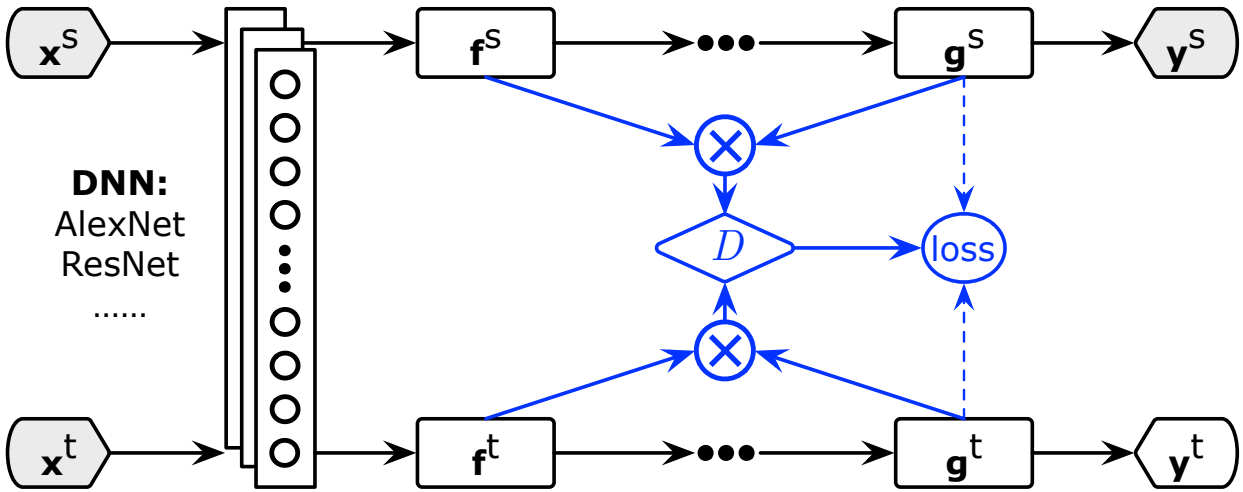
Theorem

For every hypothesis $h \in \mathcal{H}$,

$$\epsilon_Q(h) \leq \epsilon_P(h) + d_{h,\mathcal{H}}(P, Q) + \epsilon_{ideal}, \tag{12}$$

where $\epsilon_{ideal} = \epsilon(\mathcal{H}, P, Q)$ *is the ideal combined loss.*

CDAN: Conditional Domain Adversarial Network[4]



Conditional adaptation of distributions over representation & prediction

$$\begin{aligned} &\min_G \ \mathcal{E}(G) - \lambda \mathcal{E}(D, G) \\ &\min_D \ \mathcal{E}(D, G), \end{aligned} \tag{13}$$

$$\mathcal{E}(D, G) = -\mathbb{E}_{\mathbf{x}_i^s \sim \mathcal{D}_s} \log \left[D\left(\mathbf{f}_i^s \otimes \mathbf{g}_i^s \right) \right] - \mathbb{E}_{\mathbf{x}_j^t \sim \mathcal{D}_t} \log \left[1 - D\left(\mathbf{f}_j^t \otimes \mathbf{g}_j^t \right) \right] \tag{14}$$

[4] *Long et al. Conditional Adversarial Domain Adaptation. NIPS 2018.*

MDD: Margin Disparity Discrepancy[5]

- Margin risk: $\epsilon_D^{(\rho)}(f) = \mathbb{E}_{(x,y)\sim D}\left[\Phi_\rho(\rho_f(x,y))\right]$
- 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

$$\begin{aligned} d_{f,\mathcal{F}}^{(\rho)}(P, Q) &\triangleq \sup_{f'\in\mathcal{F}} \left(\epsilon_Q^{(\rho)}(f', f) - \epsilon_P^{(\rho)}(f', f)\right), \\ d_{f,\mathcal{F}}^{(\rho)}(\widehat{P}, \widehat{Q}) &\triangleq \sup_{f'\in\mathcal{F}} \left(\epsilon_{\widehat{Q}}^{(\rho)}(f', f) - \epsilon_{\widehat{P}}^{(\rho)}(f', f)\right). \end{aligned} \tag{15}$$

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

[5] *Zhang et al. Bridging Theory and Algorithm for Domain Adaptation. ICML 2019.*

Bounding the Target Risk by MDD

Theorem

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

$$\epsilon_Q(f) \leq \epsilon_P^{(\rho)}(f) + d_{f,\mathcal{F}}^{(\rho)}(P, Q) + \epsilon_{ideal}^{(\rho)}, \tag{16}$$

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

$$\epsilon_{ideal}^{(\rho)} = \min_{f^*\in\mathcal{F}} \{\epsilon_P^{(\rho)}(f^*) + \epsilon_Q^{(\rho)}(f^*)\}. \tag{17}$$

- This upper bound has a similar form with previous bound.
  - $\epsilon_P^{(\rho)}(f)$ depicts the performance of f on source domain.
  - MDD bounds the performance gap caused by domain shift.
  - ϵ_{ideal} quantifies the inverse of "*adaptability*".
- **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

$$\Pi_1\mathcal{F} = \{x \mapsto f(x, y) \big| y \in \mathcal{Y}, f \in \mathcal{F}\}. \tag{18}$$

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

$$\Pi_{\mathcal{H}}\mathcal{F} \triangleq \{x \mapsto f(x, h(x)) \big| h \in \mathcal{H}, f \in \mathcal{F}\}. \tag{19}$$

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 $\widehat{D}$ is defined as

$$\widehat{\mathfrak{R}}_{\widehat{D}}(\mathcal{G}) = \mathbb{E}_{\sigma} \sup_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^{n} \sigma_i g(z_i). \tag{20}$$

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

$$\mathfrak{R}_{n,D}(\mathcal{G}) = \mathbb{E}_{\widehat{D} \sim D^n} \widehat{\mathfrak{R}}_{\widehat{D}}(\mathcal{G}). \tag{21}$$

Definition (**Covering Number**)

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

Generalization Bound with Rademacher Complexity

Theorem (Generalization Bound with Rademacher Complexity)

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

$$\begin{aligned}\epsilon_Q(f) \leq & \epsilon_{\widehat{P}}^{(\rho)}(f) + d_{f,\mathcal{F}}^{(\rho)}(\widehat{P}, \widehat{Q}) + \epsilon_{ideal} \\ & + \frac{2k^2}{\rho}\mathfrak{R}_{n,P}(\Pi_1\mathcal{F}) + \frac{k}{\rho}\mathfrak{R}_{n,P}(\Pi_{\mathcal{H}}\mathcal{F}) + 2\sqrt{\frac{\log\frac{2}{\delta}}{2n}} \\ & + \frac{k}{\rho}\mathfrak{R}_{m,Q}(\Pi_{\mathcal{H}}\mathcal{F}) + \sqrt{\frac{\log\frac{2}{\delta}}{2m}}.\end{aligned} \tag{22}$$

Rademacher Bound of Linear Classifier

We need to check the variation of $\mathfrak{R}_{n,D}(\Pi_{\mathcal{H}}\mathcal{F})$ with the growth of n. First, we include a simple example of binary linear classifiers.

Theorem

Let $S \subseteq \mathcal{X} = \{\mathbf{x} \in \mathbb{R}^s | \|\mathbf{x}\|_2 \leq r\}$ be a sample of size m and suppose

$$\mathcal{F} = \left\{ f : \mathcal{X} \times \{\pm 1\} \to \mathbb{R} \;\middle|\; f(\mathbf{x}, y) = \operatorname{sgn}(y)\ \mathbf{w} \cdot \mathbf{x},\ \|\mathbf{w}\|_2 \leq \Lambda \right\},$$
$$\mathcal{H} = \left\{ h \mid h(\mathbf{x}) = \operatorname{sgn}(\mathbf{w} \cdot \mathbf{x}),\ \|\mathbf{w}\|_2 \leq \Lambda \right\}.$$

Then the empirical Rademacher complexity of $\Pi_{\mathcal{H}}\mathcal{F}$ can be bounded as follows:

$$\widehat{\mathfrak{R}}_S(\Pi_{\mathcal{H}}\mathcal{F}) \leq 2\Lambda r \sqrt{\frac{d \log \frac{em}{d}}{m}},$$

where d is the VC-dimension of $\mathcal{H}$.

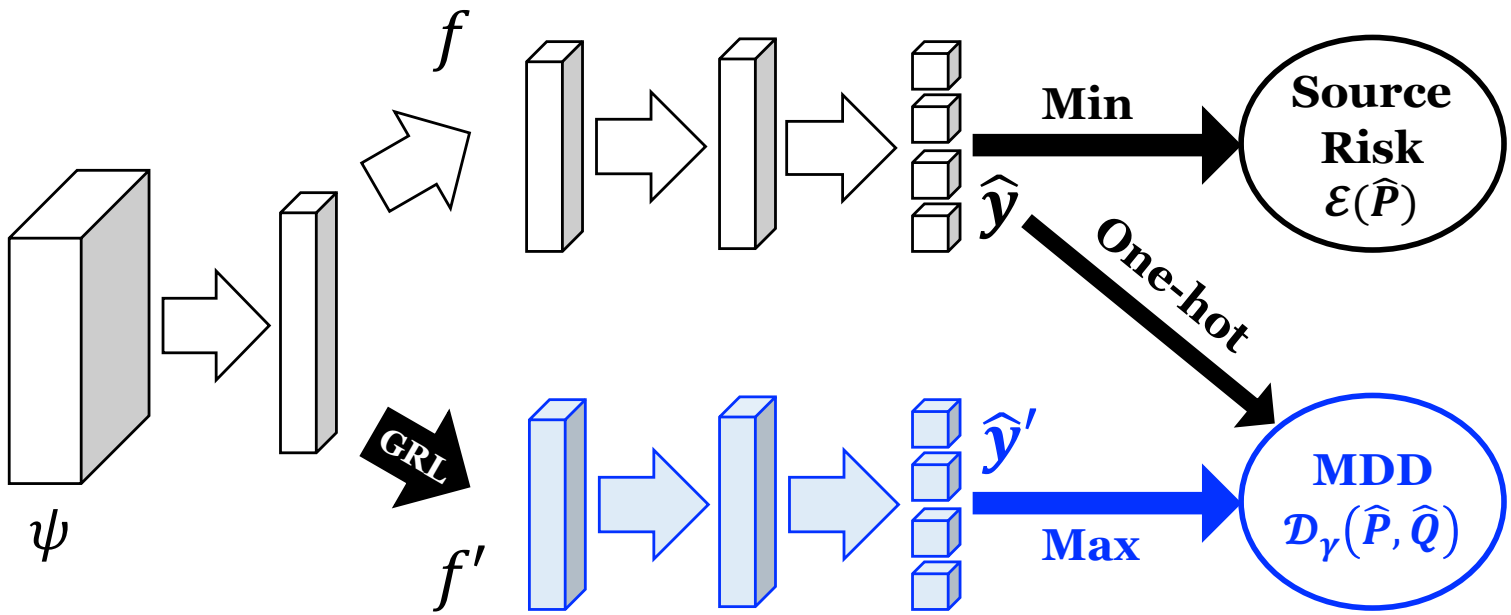
Generalization Bound with Covering Numbers

Theorem (Generalization Bound with Covering Numbers)

Let $\mathcal{F} \subseteq \mathbb{R}^{\mathcal{X}\times\mathcal{Y}}$ be a hypothesis set with label set $\mathcal{Y} = \{1, \cdots, k\}$ and $\mathcal{H} \subseteq \mathcal{Y}^{\mathcal{X}}$ be the corresponding $\mathcal{Y}$-valued labeling function class. Suppose $\Pi_1\mathcal{F}$ is bounded in $\mathcal{L}_2$ by L. Fix $\rho > 0$. For all $\delta > 0$, with probability $1 - 3\delta$ the following inequality holds for all hypothesis $f \in \mathcal{F}$:

$$\begin{aligned}\epsilon_Q(f) \leq & \epsilon_{\widehat{P}}^{(\rho)}(f) + d_{f,\mathcal{F}}^{(\rho)}(\widehat{P}, \widehat{Q}) + \epsilon_{ideal} + 2\sqrt{\frac{\log\frac{2}{\delta}}{2n}} \\ & + \sqrt{\frac{\log\frac{2}{\delta}}{2m}} + \frac{16k^2\sqrt{k}}{\rho} \inf_{\epsilon\geq 0}\Big\{\epsilon + 3\Big(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\Big) \\ & \Big(\int_{\epsilon}^{L}\sqrt{\log \mathcal{N}_2(\tau, \Pi_1\mathcal{F})}\mathrm{d}\tau + L\int_{\epsilon/L}^{1}\sqrt{\log \mathcal{N}_2(\tau, \Pi_1\mathcal{H})}\mathrm{d}\tau\Big)\Big\}.\end{aligned} \tag{23}$$

MDD: Margin Disparity Discrepancy



Minimax game: Adversarial learning induced by informative margin theory

$$\min_{f,\psi} \ \epsilon^{(\rho)}_{\psi(\widehat{P})}(f) + (\epsilon^{(\rho)}_{\psi(\widehat{Q})}(f^*, f) - \epsilon^{(\rho)}_{\psi(\widehat{P})}(f^*, f)),$$
$$f^* = \max_{f'} \ (\epsilon^{(\rho)}_{\psi(\widehat{Q})}(f', f) - \epsilon^{(\rho)}_{\psi(\widehat{P})}(f', f)). \tag{24}$$

Results

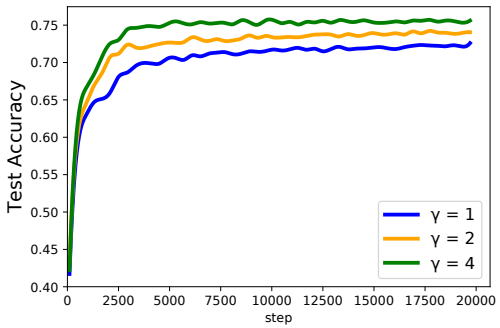
Table: Accuracy (%) on *Office-31* for unsupervised domain adaptation

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
ResNet-50	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
DANN	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
MCD	88.6±0.2	98.5±0.1	**100.0**±.0	92.2±0.2	69.5±0.1	69.7±0.3	86.5
CDAN	94.1±0.1	**98.6**±0.1	**100.0**±.0	92.9±0.2	71.0±0.3	69.3±0.3	87.7
MDD (Proposed)	**94.5**±0.3	98.4±0.1	**100.0**±.0	**93.5**±0.2	**74.6**±0.3	**72.2**±0.1	**88.9**

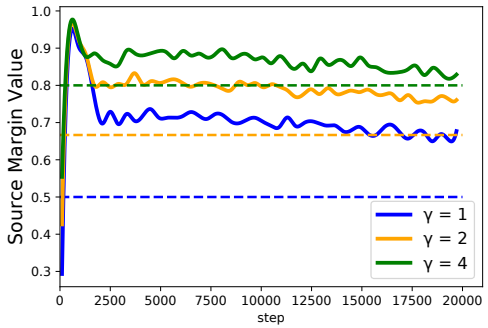
Table: Accuracy (%) on *Office-Home* for unsupervised domain adaptation

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
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
JAN	45.9	61.2	68.9	50.4	59.7	61.0	45.8	43.4	70.3	63.9	52.4	76.8	58.3
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 (Proposed)	**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**

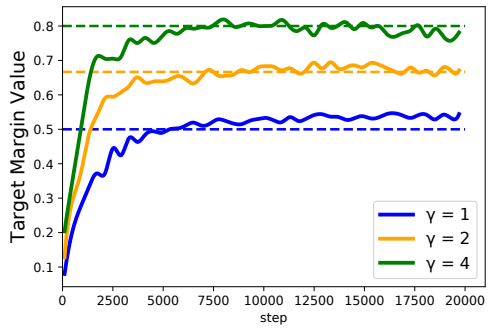
Results



(c) Test Accuracy



(d) Equilibrium on Source



(e) Equilibrium on Target

Figure: Test accuracy and empirical values of $\sigma_{h_f} \circ f'$ (dashed line: $\frac{\gamma}{\gamma+1}$).

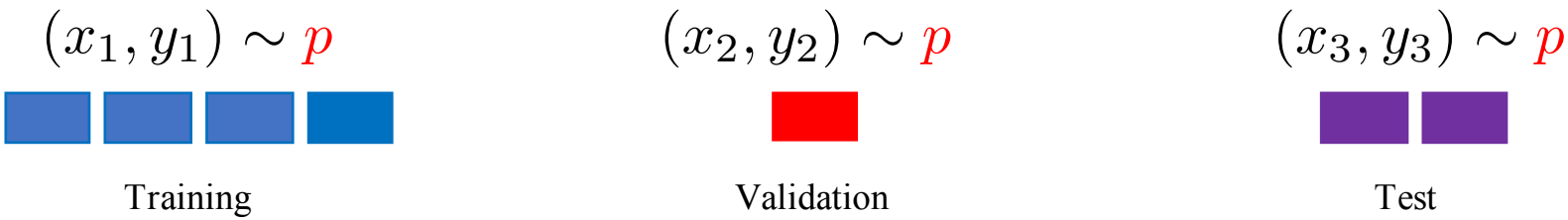
Table: Accuracy (%) on Office-31 by different margins.

Margin γ	1	2	3	**4**	5	6
A $\rightarrow$ W	92.5	93.7	94.0	**94.5**	93.8	93.5
D $\rightarrow$ A	72.4	73.0	73.7	**74.6**	74.3	74.2
Avg on Office-31	87.6	88.1	88.5	**88.9**	88.7	88.6

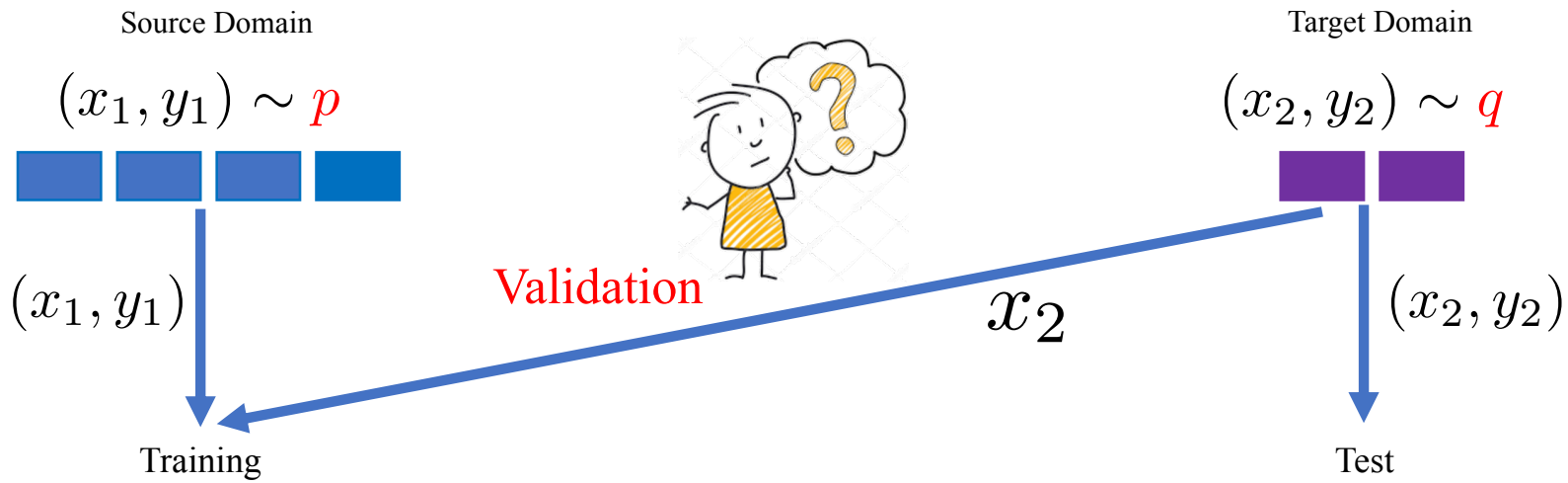
Outline

Model Selection in Domain Adaptation

- Supervised Learning



- Semi-Supervised Learning (SSL)?
- Unsupervised Domain Adaptation (UDA)?

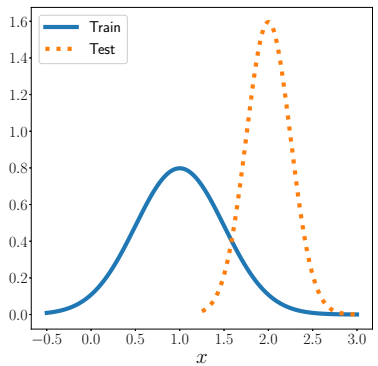


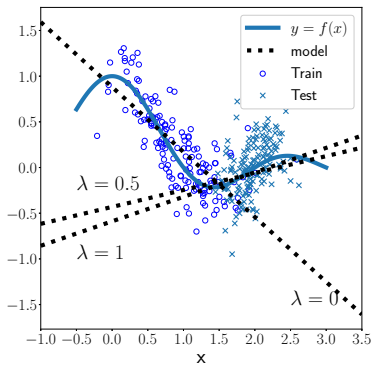
IWCV: Importance-Weighted Cross-Validation[6]

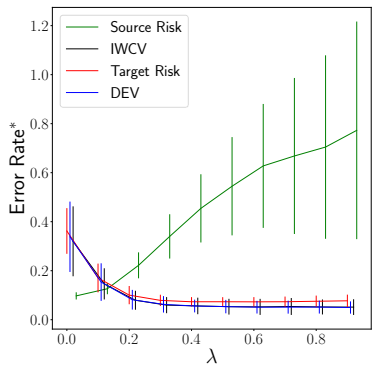
- Covariate shift assumption: $P(y|\mathbf{x}) = Q(y|\mathbf{x})$
- Model selection by estimating Target Risk $\epsilon_Q(h) = \mathbb{E}_Q[h(\mathbf{x}) \neq y]$
- Importance-Weighted Cross-Validation (IWCV)

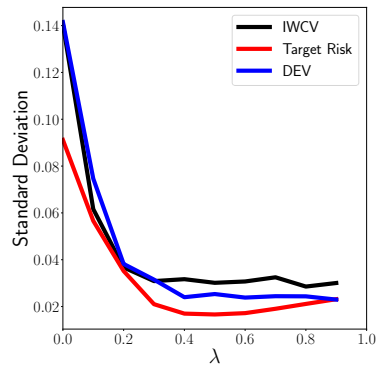
$$\mathbb{E}_P w(\mathbf{x}) \cdot [h(\mathbf{x}) \neq y] = \mathbb{E}_P \frac{Q(\mathbf{x})}{P(\mathbf{x})} \cdot [h(\mathbf{x}) \neq y] = \mathbb{E}_Q [h(\mathbf{x}) \neq y] = \epsilon_Q(h)$$

  - 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[10]

- Variance of IWCV (bounded by Rényi divergence)[7]:

$$\mathrm{Var}_{\mathbf{x}\sim P}[w(\mathbf{x})\cdot[h(\mathbf{x})\neq y]] \leq d_{\alpha+1}(Q\|P)\epsilon_Q(h)^{1-\frac{1}{\alpha}} - \epsilon_Q(h)^2$$

- Density ratio $w(\mathbf{x}) = \frac{Q(\mathbf{x})}{P(\mathbf{x})}$ is estimated by discriminative learning[8]
- Feature adaptation reduces distribution discrepancy $d_{\alpha+1}(Q\|P)$[9]
- Control variate explicitly reduces the variance of $\mathbb{E}_P w(\mathbf{x})\cdot[h(\mathbf{x})\neq y]$
  - $\mathbb{E}[z]=\zeta, \mathbb{E}[t]=\tau$
  - $z^\star = z+\eta(t-\tau)$
  - $\mathbb{E}[z^\star] = \mathbb{E}[z]+\eta\mathbb{E}[t-\tau] = \zeta+\eta(\mathbb{E}[t]-\mathbb{E}[\tau]) = \zeta.$
  - $\mathrm{Var}[z^\star] = \mathrm{Var}[z+\eta(t-\tau)] = \eta^2\mathrm{Var}[t]+2\eta\mathrm{Cov}(z,t)+\mathrm{Var}[z]$
  - min $\mathrm{Var}[z^\star] = (1-\rho_{z,t}^2)\mathrm{Var}[z]$, when $\hat{\eta} = -\frac{\mathrm{Cov}(z,t)}{\mathrm{Var}[t]}$

[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*

[10] *You et al. Towards Accurate Model Selection in Deep Unsupervised Domain Adaptation. ICML 2019.*

DEV: Deep Embedded Validation

Algorithm 1 Deep Embedded Validation (DEV)

1: **Input:** Candidate model $g(\mathbf{x}) = \mathcal{T}(F(\mathbf{x}))$
Training set $\mathcal{D}_{\text{tr}} = \{(\mathbf{x}_i^{\text{tr}}, y_i^{\text{tr}})\}_{i=1}^{n_{\text{tr}}}$
Validation set $\mathcal{D}_{\text{v}} = \{(\mathbf{x}_i^{\text{v}}, y_i^{\text{v}})\}_{i=1}^{n_{\text{v}}}$
Test set $\mathcal{D}_{\text{ts}} = \{(\mathbf{x}_i^{\text{ts}})\}_{i=1}^{n_{\text{ts}}}$

2: **Output:** DEV Risk $\mathcal{R}_{\text{DEV}}(g)$ of model g

3: Compute features and predictions using model g:
$$\mathcal{F}_{\text{tr}} = \{\boldsymbol{f}_i^{\text{tr}}\}_{i=1}^{n_{\text{tr}}},\ \mathcal{F}_{\text{ts}} = \{\boldsymbol{f}_i^{\text{ts}}\}_{i=1}^{n_{\text{ts}}},\ \mathcal{F}_{\text{v}} = \{\boldsymbol{f}_i^{\text{v}}\}_{i=1}^{n_{\text{v}}},\ \mathcal{Y}_{\text{v}} = \{\hat{y}_i^{\text{v}}\}_{i=1}^{n_{\text{v}}}$$

4: Train a two-layer logistic regression model M to classify $\mathcal{F}_{\text{tr}}$ and $\mathcal{F}_{\text{ts}}$ (label $\mathcal{F}_{\text{tr}}$ as 1 and $\mathcal{F}_{\text{ts}}$ as 0)

5: Compute $w_{\boldsymbol{f}}(\mathbf{x}_i^{\text{v}}) = \frac{n_{\text{tr}}}{n_{\text{ts}}} \frac{1 - M(\boldsymbol{f}_i^{\text{v}})}{M(\boldsymbol{f}_i^{\text{v}})}$, $W = \{w_{\boldsymbol{f}}(\mathbf{x}_i^{\text{v}})\}_{i=1}^{n_{\text{v}}}$

6: Compute weighted loss $L = \{w_{\boldsymbol{f}}(\mathbf{x}_i^{\text{v}}) \ell(\hat{y}_i^{\text{v}}, y_i^{\text{v}})\}_{i=1}^{n_{\text{v}}}$

7: Estimate coefficient $\eta = -\frac{\widehat{\text{Cov}}(L, W)}{\widehat{\text{Var}}[W]}$

8: Compute DEV Risk $\mathcal{R}_{\text{DEV}}(g) = \text{mean}(L) + \eta \text{mean}(W) - \eta$

Outline

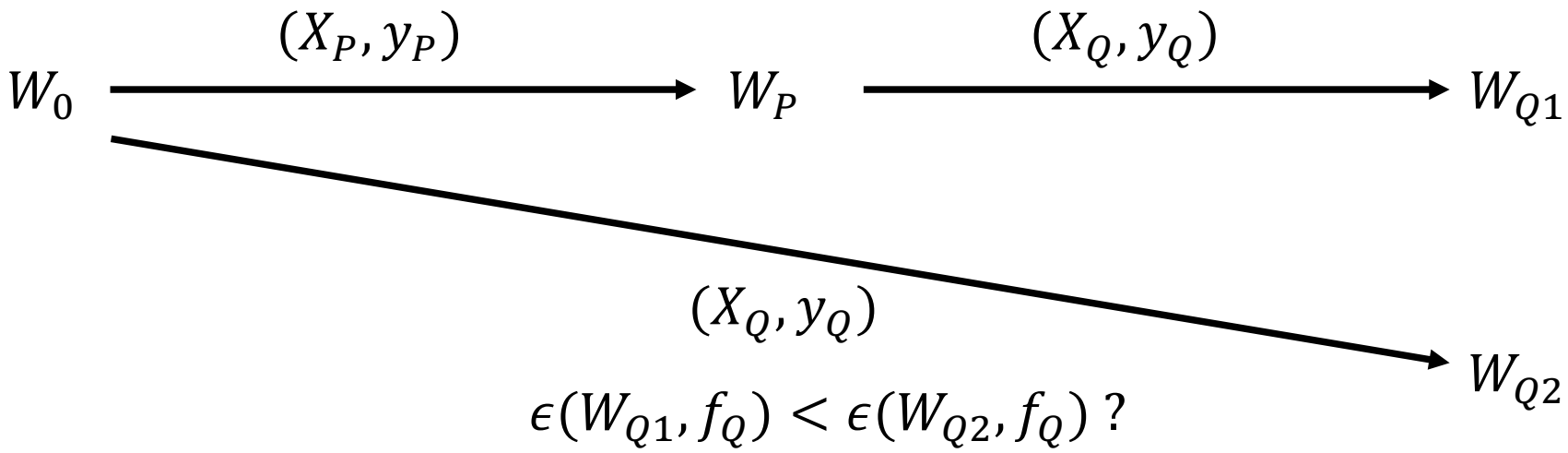
Inductive Transfer Learning[11]

Successful of transfer learning: Pre-train a model on a large-scale source dataset, and use the parameters as initialization for training a target task.

Compared to training from scratch:

- Generalization: better accuracy
- Optimization: faster convergence

How to understand the transferability of deep representations?

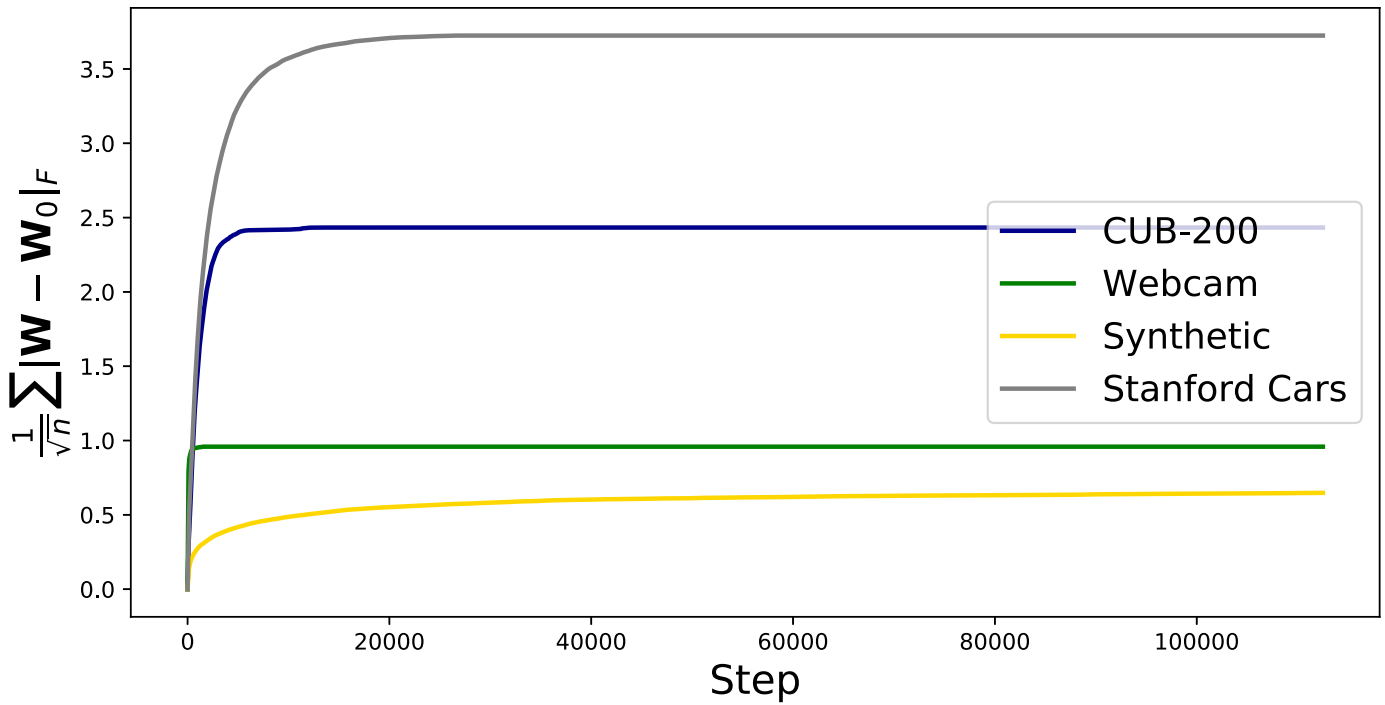


[11] *Liu et al. Towards Understanding the Transferability of Deep Representations, arXiv, 2019*

Transferred Parameters Induce Better Generalization

We can quantify how pre-trained knowledge is preserved when transferring to the target dataset with $\frac{1}{\sqrt{n}}\|\mathbf{W}_Q - \mathbf{W}_P\|_F$.

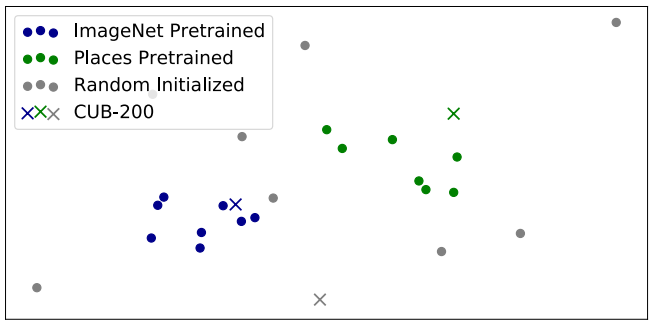
- For more similar target datasets, $\frac{1}{\sqrt{n}}\|\mathbf{W}_Q - \mathbf{W}_P\|_F$ is smaller
- For more similar target datasets, generalization error is smaller
- Is $\frac{1}{\sqrt{n}}\|\mathbf{W}_Q - \mathbf{W}_P\|_F$ implicitly bounded? (we will formally study this)



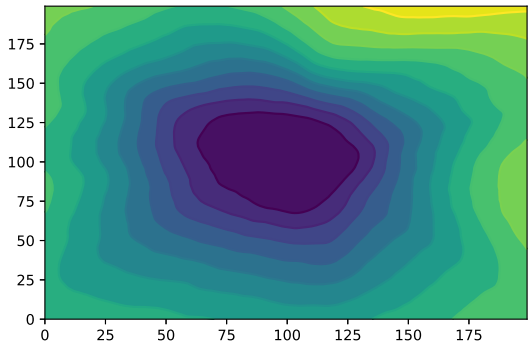
Transferred Parameters Induce Better Generalization

Staying close to the transferred parameters benefits generalization

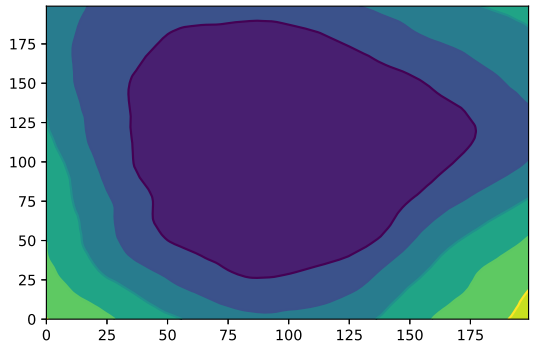
- Even for the same target dataset, different pre-trained parameters lead to significantly different solutions
- At the convergence point, pre-trained networks stay in the original flat region, leading to flatter minima than random initialization



(a) t-SNE of parameters



(b) Randomly initialized

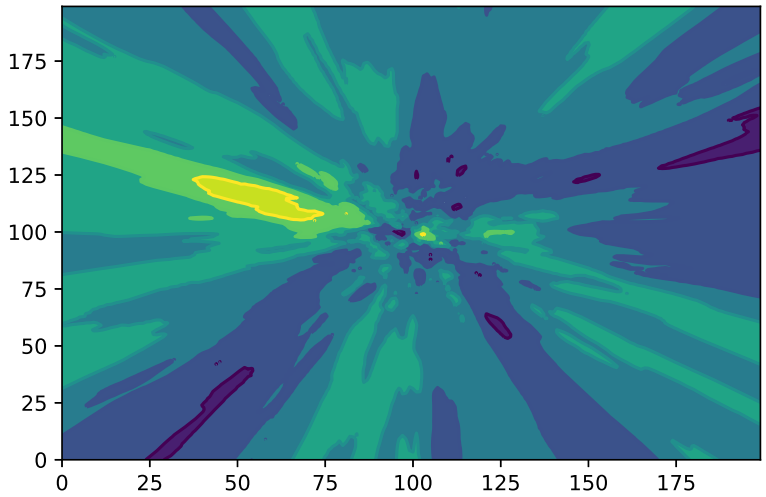


(c) ImageNet pre-trained

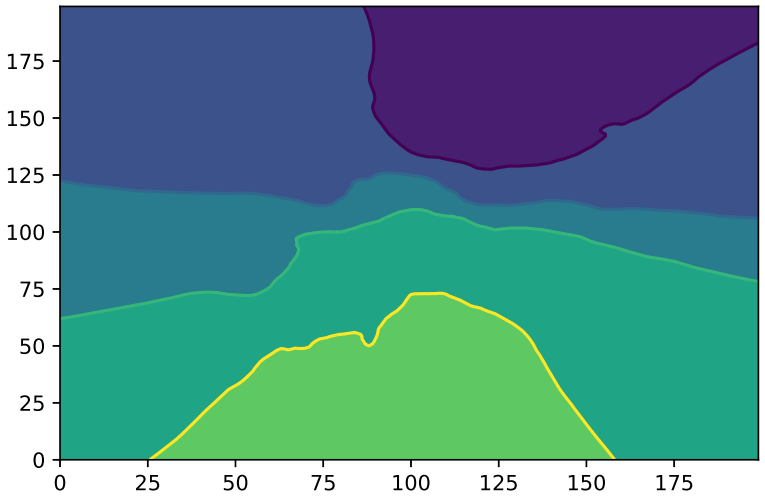
Transferred Parameters Enable Faster Optimization

Modern neural networks are equipped with Batch Normalization (BN) and skip connections to enable better loss landscapes

- However, at the initialization point, the loss landscapes are still very messy even in the presence of Batch-Norm and residual connections



(d) Randomly initialized

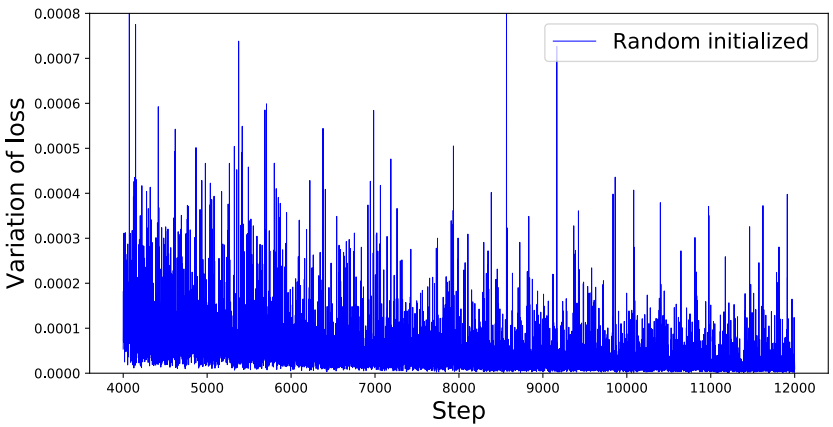


(e) ImageNet pre-trained

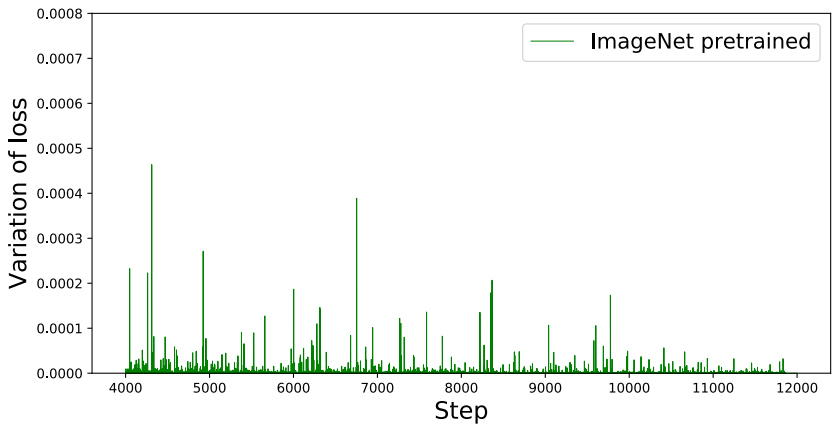
Transferred Parameters Enable Faster Optimization

Pre-trained parameters help smoothen the loss landscape and accelerate training in the early stages

- The landscapes can be described with the Lipschitzness of the loss function, i.e. the magnitude of gradient



(f) Randomly initialized.

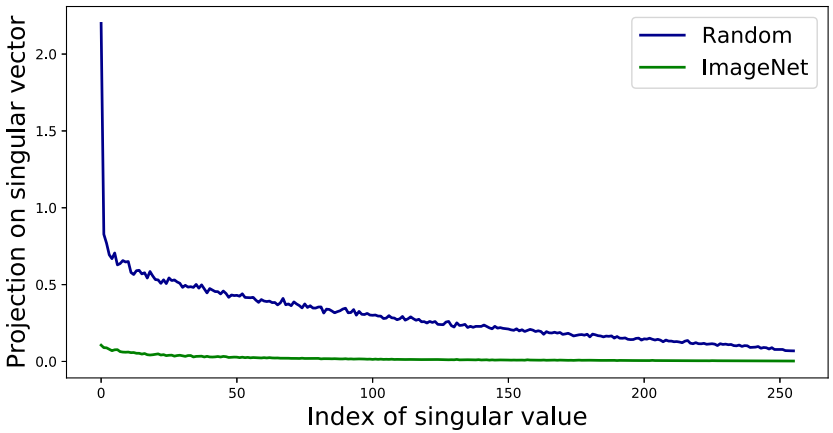


(g) ImageNet pre-trained.

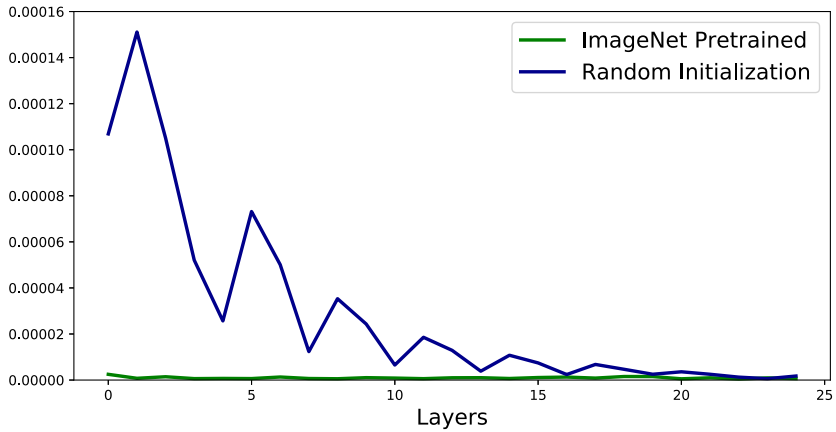
Transferred Parameters Enable Better Optimization

Why is the magnitude of gradient better controlled with the pre-trained representations?

- The gradient is computed through back-prop, $\frac{\partial L}{\partial \mathbf{x}_i^{k-1}} = \mathbf{W}_k \mathbb{I}_i^k \left(\frac{\partial L}{\partial \mathbf{x}_i^k} \right)$. Pre-trained weight matrices provide more stable scaling factors.



(h) Projection on components of gradient



(i) Scaling of gradient

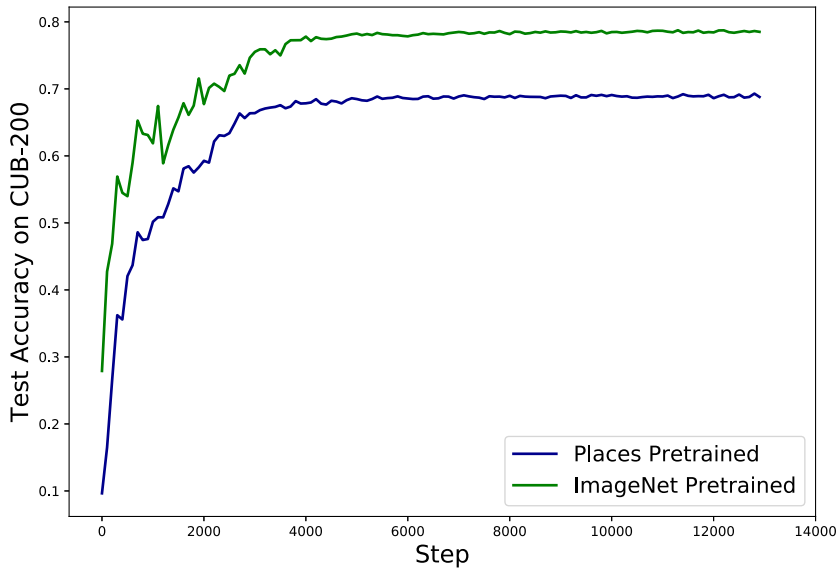
Feasibility of Transfer Learning

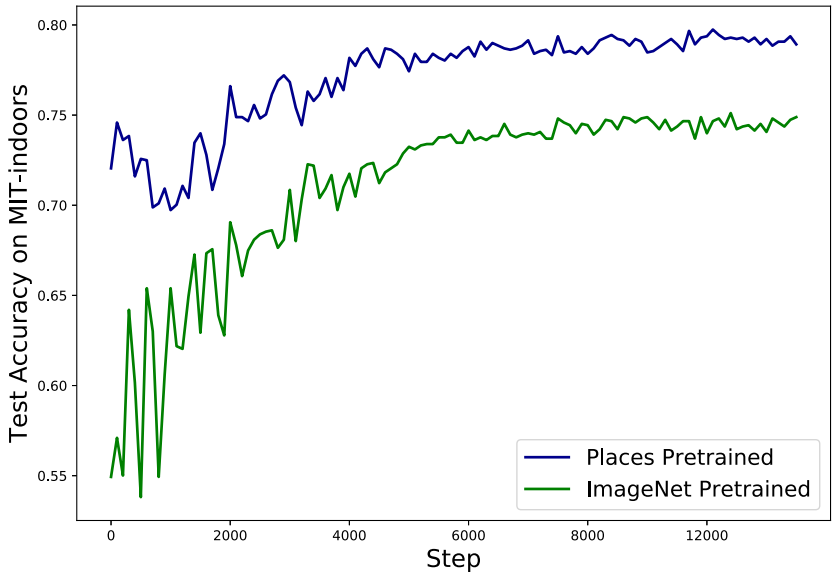
Varying input with fixed labels.

- Choosing a model pre-trained on more similar inputs yields a larger performance gain.

Varying labels with fixed input.

- The similarity of the input (images) is just one point. Another factor of similarity is the relationship between the nature of tasks (labels).

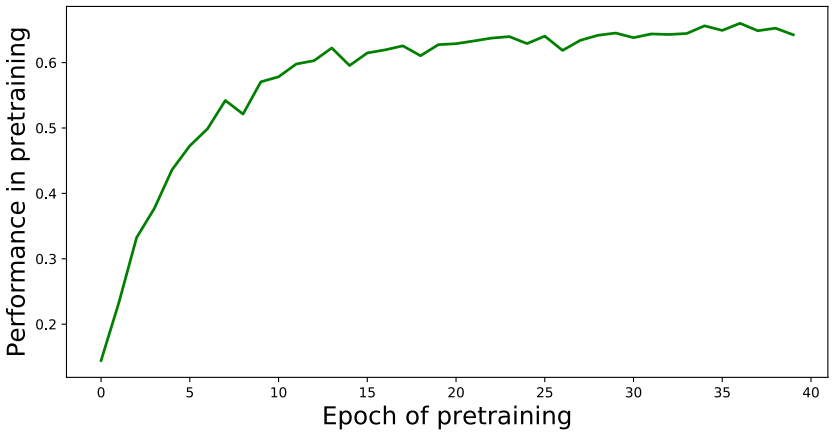




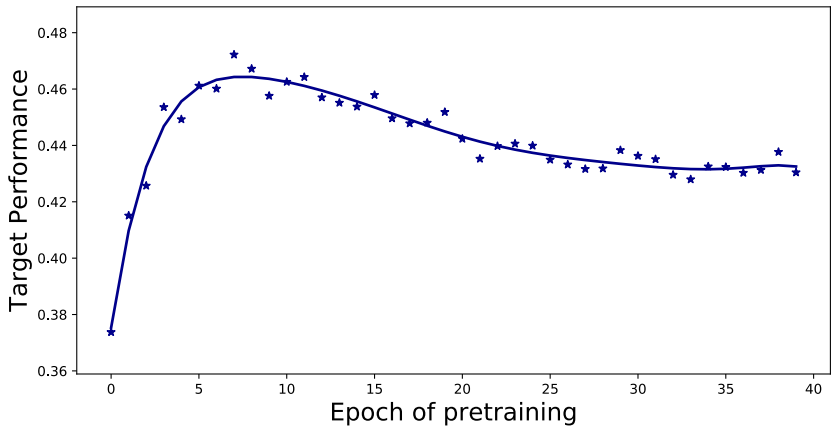
Feasibility of Transfer Learning

Choices of pre-training epochs.

- Although the test accuracy on the pre-training dataset continues increasing, the test accuracy on the target dataset starts to decline.



(l) Accuracy of pre-training



(m) Accuracy of the target dataset

Theoretical Analysis

- Two-layer ReLU network of m hidden units $f_{\mathbf{W},\mathbf{a}}(\mathbf{x}) = \frac{1}{\sqrt{m}}\mathbf{a}^\top \sigma(\mathbf{W}^\top \mathbf{x})$.
- $\mathbf{x} \in \mathbb{R}^d$, $\mathbf{W} = (\mathbf{w}_1, \cdots, \mathbf{w}_m) \in \mathbb{R}^{d\times m}$.
- $\mathbf{w}_r(0) \sim \mathcal{N}(\mathbf{0},\ \kappa^2\mathbf{I})$, $a_r \sim$unif $(\{-1, 1\})$.
- $L(\mathbf{W}) = \frac{1}{2}(\mathbf{y} - f_{\mathbf{W},\mathbf{a}}(\mathbf{X}))^\top(\mathbf{y} - f_{\mathbf{W},\mathbf{a}}(\mathbf{X}))$.

We first pre-train the model on $\{\mathbf{x}_{P,i}, y_{P,i}\}_{i=1}^{n_P}$ drawn i.i.d from P to obtain $\mathbf{W}(P)$, then train on the target dataset $\{\mathbf{x}_{Q,i}, y_{Q,i}\}_{i=1}^{n_Q}$ drawn i.i.d from Q.

Definition (Gram matrix of P and Q)

$$\mathbf{H}^\infty_{P,ij} = \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\mathbf{x}^\top_{P,i}\mathbf{x}_{P,j}\mathbb{I}\{\mathbf{w}^\top\mathbf{x}_{P,i} \geq 0,\ \mathbf{w}^\top\mathbf{x}_{P,j} \geq 0\}]. \tag{25}$$

$$\mathbf{H}^\infty_{Q,ij} = \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\mathbf{x}^\top_{Q,i}\mathbf{x}_{Q,j}\mathbb{I}\{\mathbf{w}^\top\mathbf{x}_{Q,i} \geq 0,\ \mathbf{w}^\top\mathbf{x}_{Q,j} \geq 0\}]. \tag{26}$$

Definition (Gram matrix of transfer learning)

$$\mathbf{H}^\infty_{PQ,ij} = \mathbb{E}_{\mathbf{w}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}[\mathbf{x}^\top_{P,i}\mathbf{x}_{Q,j}\mathbb{I}\{\mathbf{w}^\top\mathbf{x}_{P,i} \geq 0,\ \mathbf{w}^\top\mathbf{x}_{Q,j} \geq 0\}]. \tag{27}$$

Theoretical Analysis

Definition (Transformed labels from source label set to target label set)

$$\mathbf{y}_{P\to Q} \triangleq \mathbf{H}_{PQ}^{\infty\top}\mathbf{H}_{P}^{\infty-1}\mathbf{y}_{P}. \tag{28}$$

Theorem (Improved Lipschitzness)

Denote by $\mathbf{X}^1$ *the activations in the target dataset. If* $m \geq \text{poly}(n_P, n_Q, \delta^{-1}, \lambda_P^{-1}, \lambda_Q^{-1}, \kappa^{-1})$, $\kappa = O\left(\frac{\lambda_P^2\delta}{n_P^2 n_Q^{\frac{1}{2}}}\right)$, *with probability no less than* $1-\delta$ *over the random initialization,*

$$\|\frac{\partial L(\mathbf{W}(P))}{\partial \mathbf{X}^1}\|^2 = \|\frac{\partial L(\mathbf{W}(0))}{\partial \mathbf{X}^1}\|^2 - \mathbf{y}_Q^\top\mathbf{y}_Q + (\mathbf{y}_Q - \mathbf{y}_{P\to Q})^\top(\mathbf{y}_Q - \mathbf{y}_{P\to Q}) + \frac{\text{poly}(n_P, n_Q, \delta^{-1}, \lambda_P^{-1}, \kappa^{-1})}{m^{\frac{1}{4}}} + O\left(\frac{n_P^2 n_Q^{\frac{1}{2}}\kappa}{\lambda_P^2\delta}\right). \tag{29}$$

Theoretical Analysis

Theorem (Improved generalization)

Suppose $m \geq \text{poly}(n_P, n_Q, \delta^{-1}, \lambda_P^{-1}, \lambda_Q^{-1}, \kappa^{-1})$, $\kappa = O\left(\frac{\lambda_P^2 \lambda_Q^2 \delta}{n_P^2 n_Q^{\frac{1}{2}}}\right)$, *with probability no less than* $1-\delta$ *over the random initialization,*

$$\|\mathbf{W}(Q) - \mathbf{W}(P)\|_F \leq \sqrt{(\mathbf{y}_Q - \mathbf{y}_{P\to Q})^\top \mathbf{H}_Q^{\infty\,-1}(\mathbf{y}_Q - \mathbf{y}_{P\to Q})} + O\left(\frac{n_P n_Q^{\frac{1}{2}} \kappa^{\frac{1}{2}}}{\lambda_P \lambda_Q \delta^{\frac{1}{2}}}\right) + \frac{\text{poly}(n_P, n_Q, \delta^{-1}, \lambda_P^{-1}, \lambda_Q^{-1}, \kappa^{-1})}{m^{\frac{1}{4}}}. \quad (30)$$

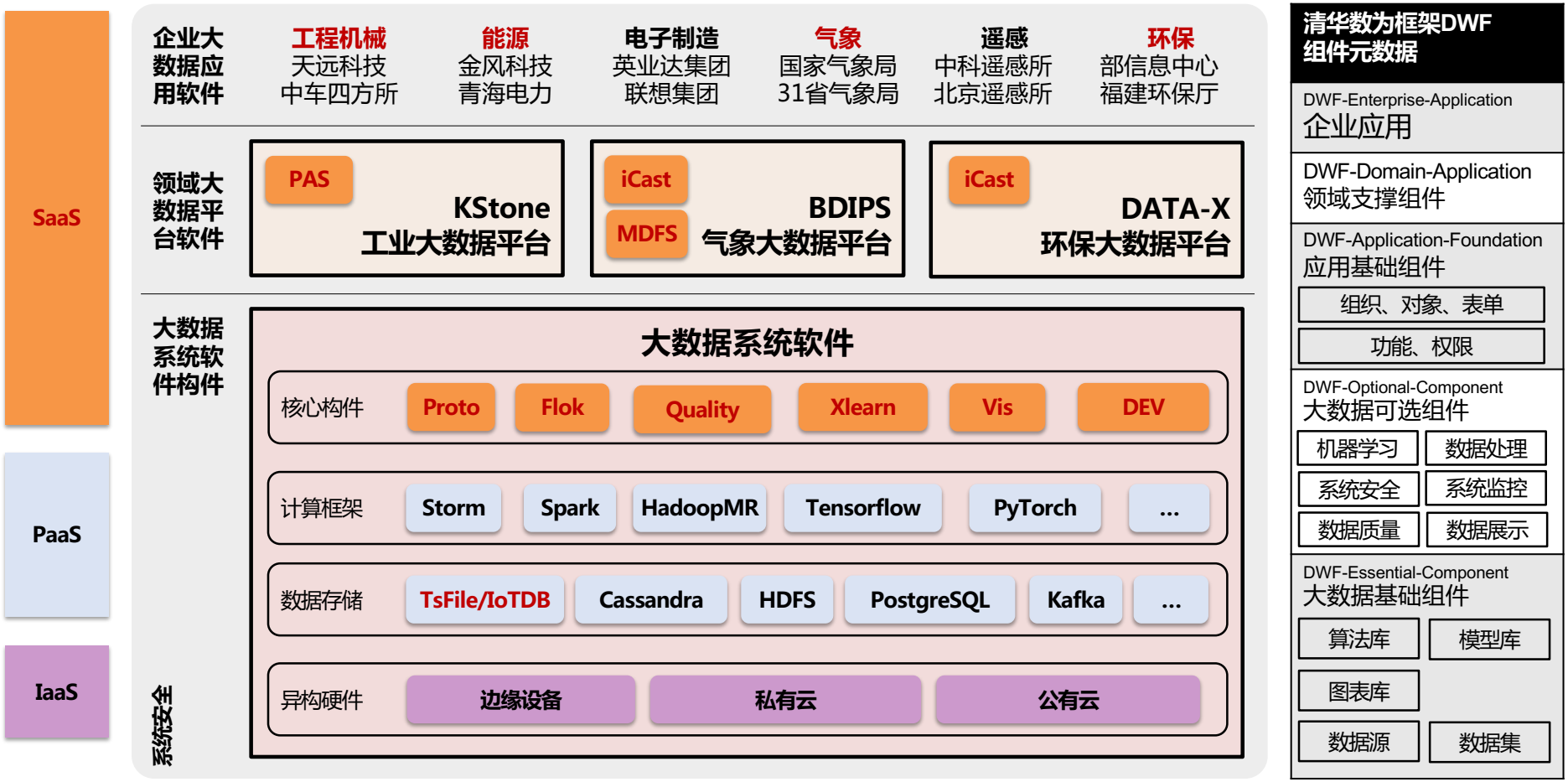
Lemma (Arora et al., 2019)

Under the same conditions as (30), *with probability no less than* $1-\delta$,

$$\mathbb{E}_Q(L(f(\mathbf{x}))) \leq \sqrt{\frac{2}{n_Q}}\|\mathbf{W}(Q) - \mathbf{W}(P)\|_F + O\left(\sqrt{\frac{1}{n_Q}}\right)$$

Transfer Learning System

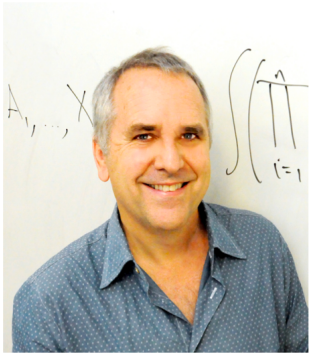
Tsinghua Dataway Big Data Software Stack



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

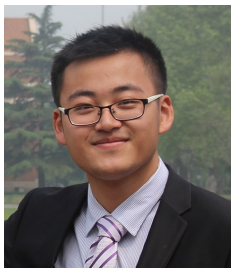


Michael I. Jordan
Academic Committee Chair
UC Berkeley

Machine Learning Group @ NELBDS

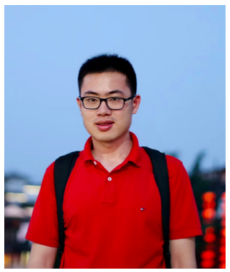
Mingsheng Long (龙明盛)

Yuchen Zhang



Yue Cao

Han Zhu



Zhangjie Cao