Feature-based Transfer Learning via Kernel Embedding of Distributions

Sinno Jialin Pan (Ph.D.)
Nanyang Assistant Professor
School of Computer Science and Engineering
Nanyang Technological University, Singapore

NANYANG TECHNOLOGICAL UNIVERSITY

Transfer of Learning

A psychological point of view

- The study of dependency of human conduct, learning or performance on prior experience.
  - [Thorndike and Woodworth, 1901] explored how individuals would transfer in one context to another context that share similar characteristics

Transfer Learning

In the machine learning community

- The ability of a system to recognize and apply knowledge and skills learned in previous domains/tasks to novel tasks/domains, which share some commonality
- Given a target domain/task, how to identify the commonality between the domain/task and previous domains/tasks, and transfer knowledge from the previous domains/tasks to the target one?

Transfer learning for classification, and regression problems.

- [**Pan** and Yang, A Survey on Transfer Learning, IEEE TKDE 2010]
- [**Pan**, Transfer learning, Book Chapter 2014]

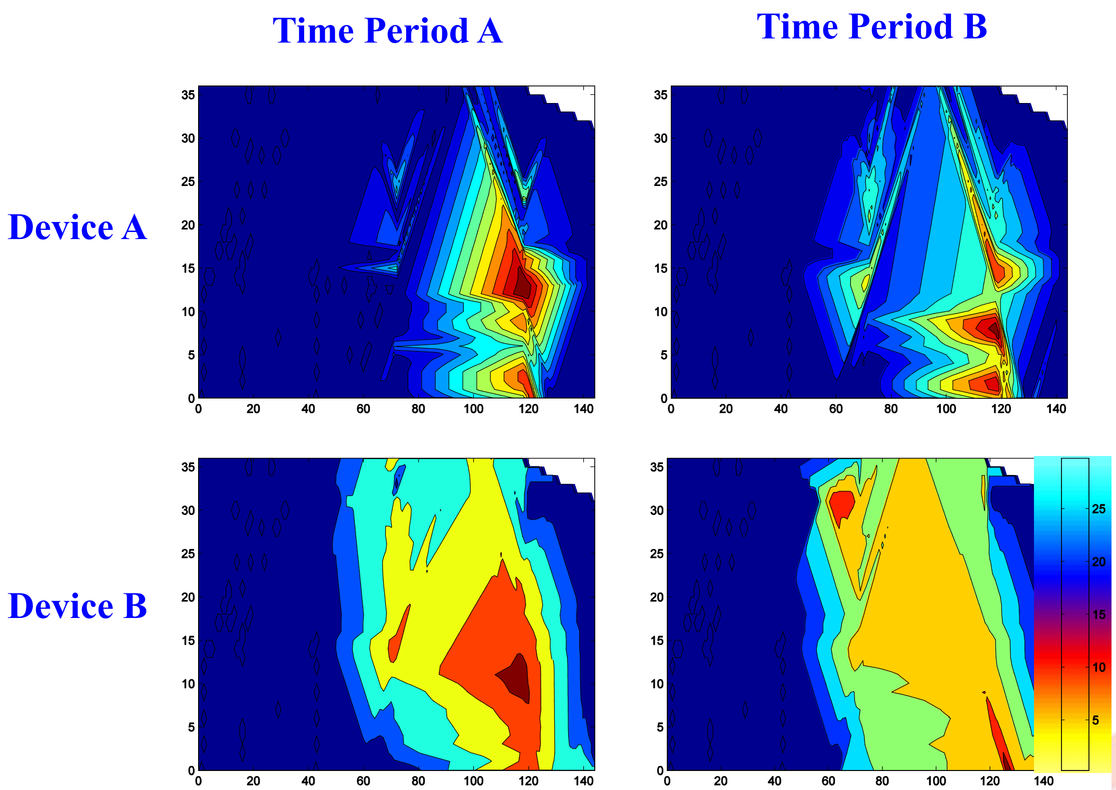
Focus!

Transfer learning for reinforcement learning problems.

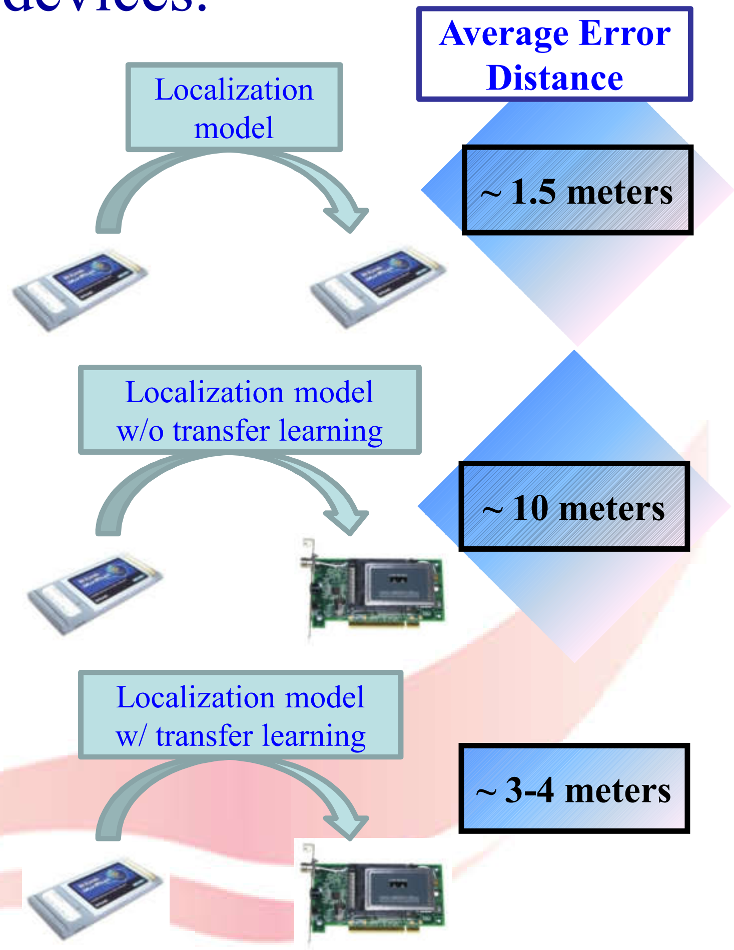
- [Taylor and Stone, Transfer Learning for Reinforcement Learning Domains: A Survey, JMLR 2009]

Applications

- **WiFi localization:** signal strength changes a lot over different time periods, or across different mobile devices.



Contour of signal strength values

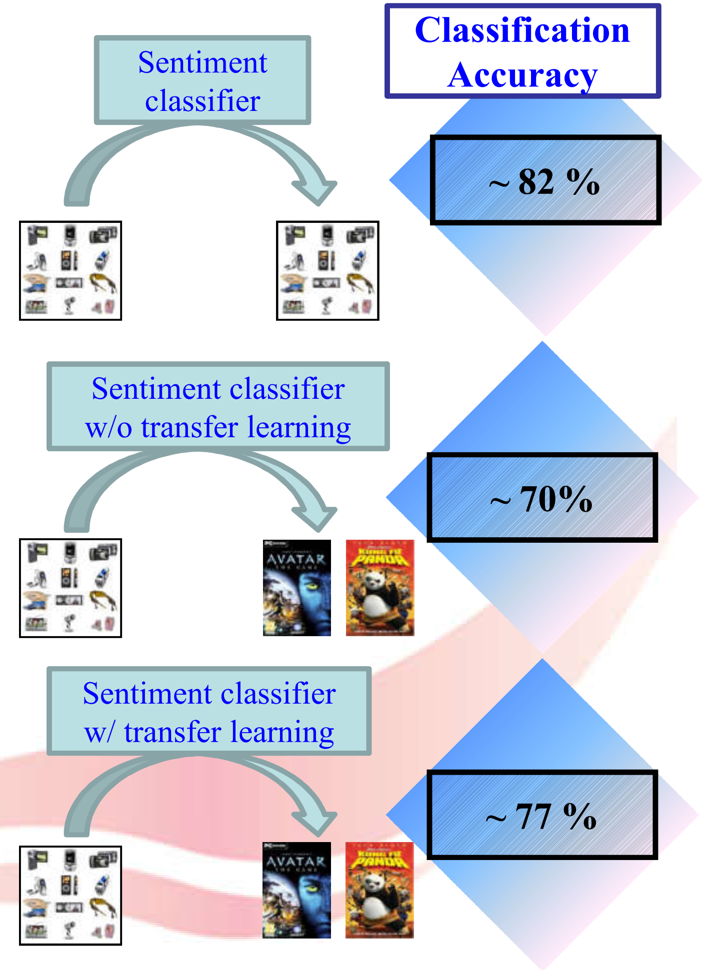


Applications (cont.)

- **Sentiment analysis:** users may use different sentiment words across different domains.

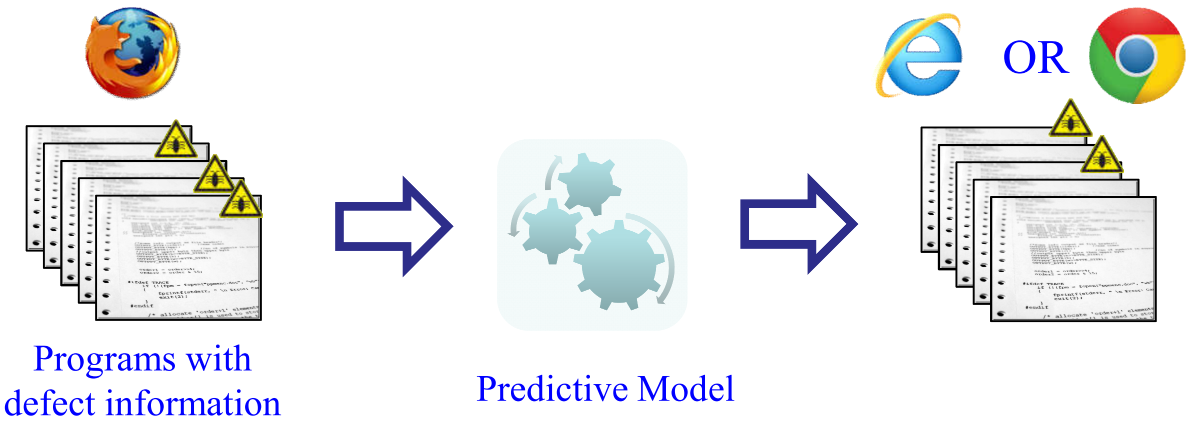
Electronics	Video Games
(1) **Compact**; easy to operate; very good picture quality; looks **sharp**!	(2) A very good game! It is action packed and full of excitement. I am very much **hooked** on this game.
(3) I purchased this unit from Circuit City and I was very excited about the quality of the picture. It is really nice and **sharp**.	(4) Very **realistic** shooting action and good plots. We played this and were **hooked**.
(5) It is also quite **blurry** in very dark settings. I will never buy HP again.	(6) The game is so **boring**. I am extremely unhappy and will probably never buy UbiSoft again.

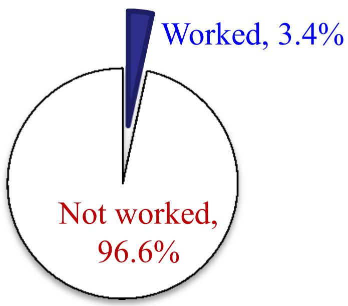
Product reviews on different domains



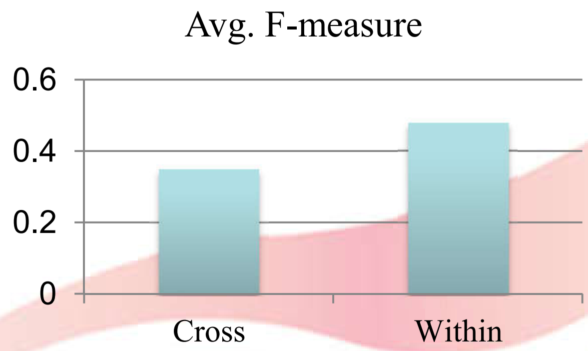
Applications (cont.)

- **[Defect prediction:]** development processes can be very different across different projects





[Zimmerman etal. FSE-09] "We ran 622 cross-project predictions and found only 3.4% actually worked."

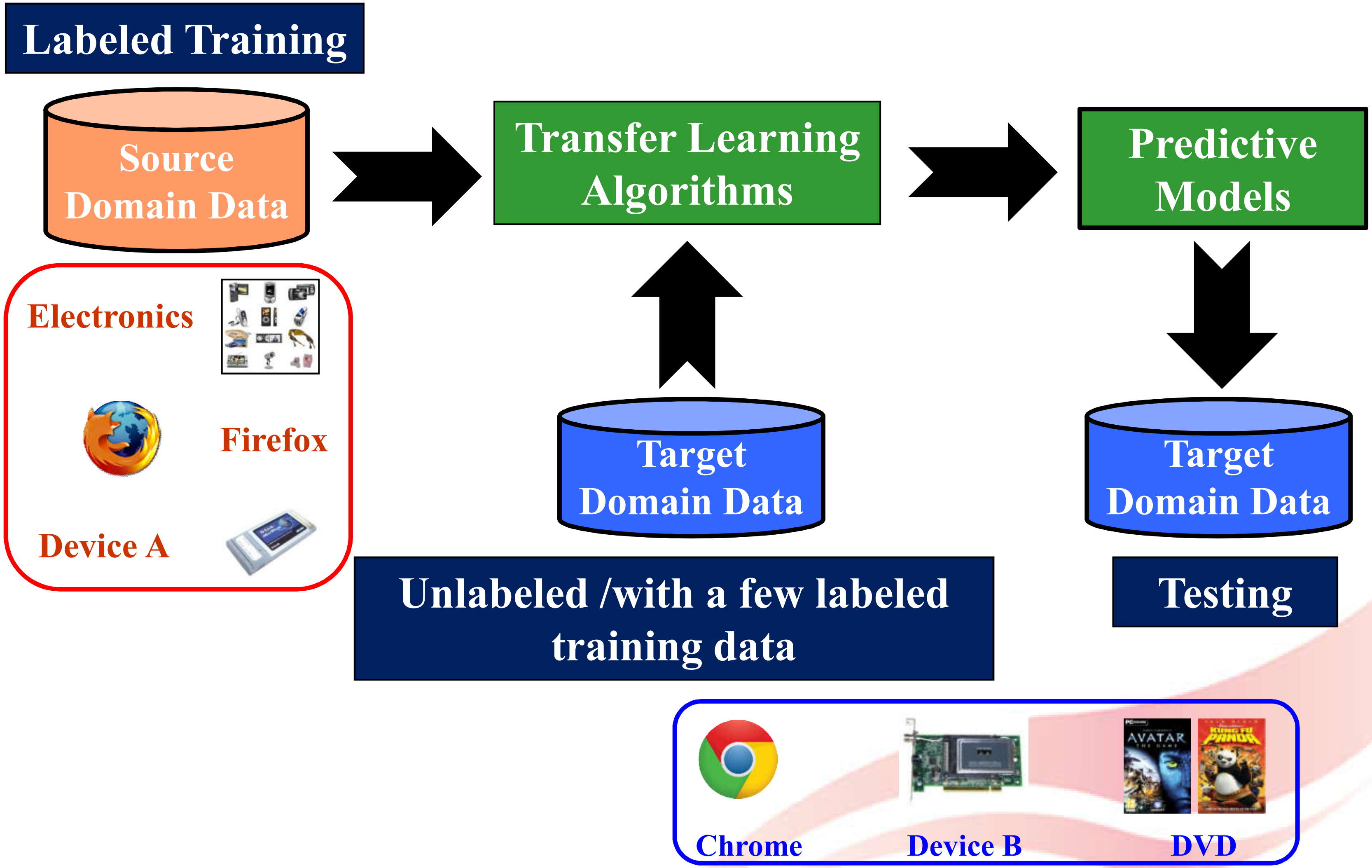


[Rahman, Posnett, and Devanbu. FSE-12]

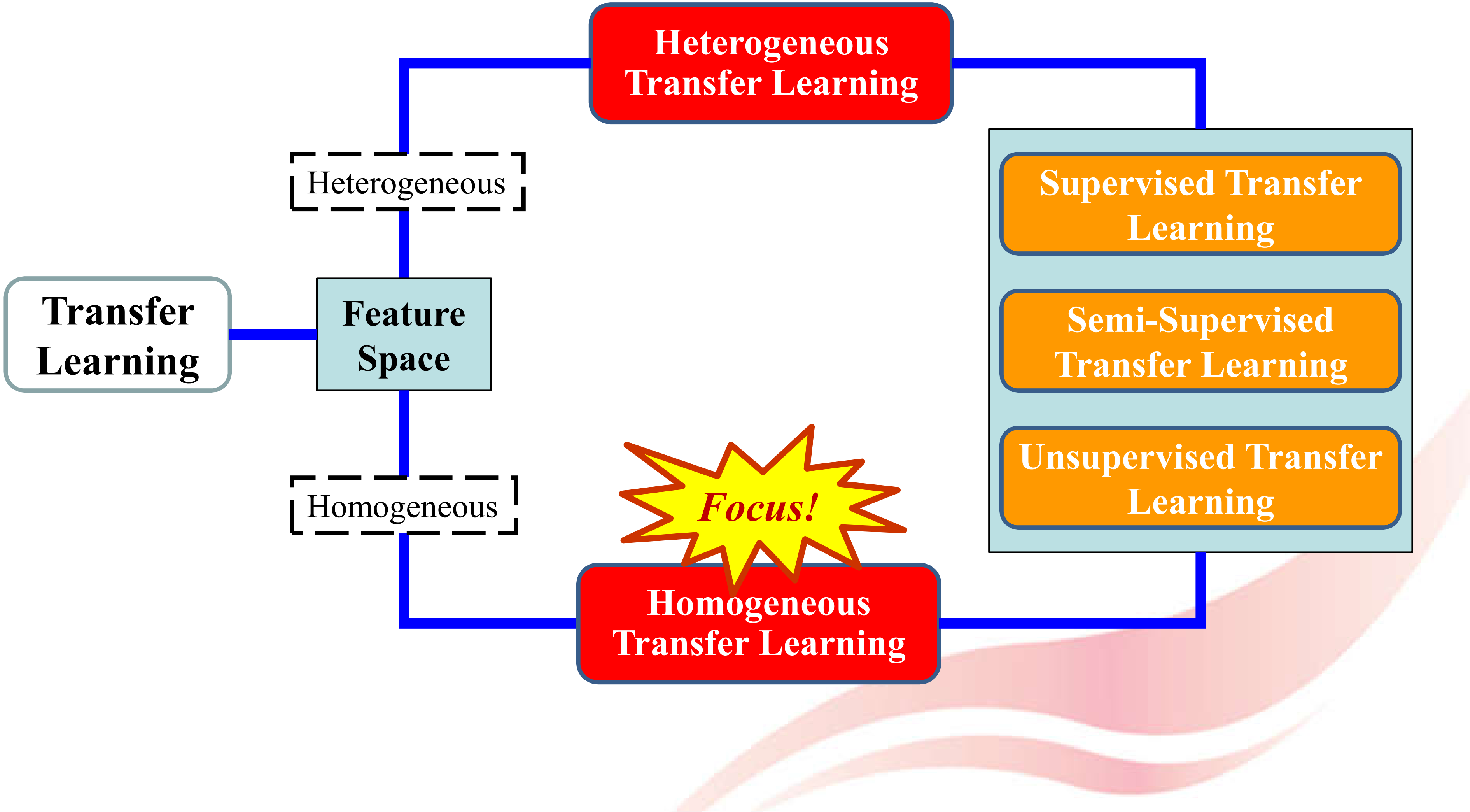
Why Models Perform Poor across Domains

- Fundamental assumption in machine learning: training and test data are assumed to be
  - Represented in the same feature space, AND
  - Follow the same data distribution
- Training and test data from different domains may be
  - Represented in different feature spaces, OR
  - Follow different data distributions

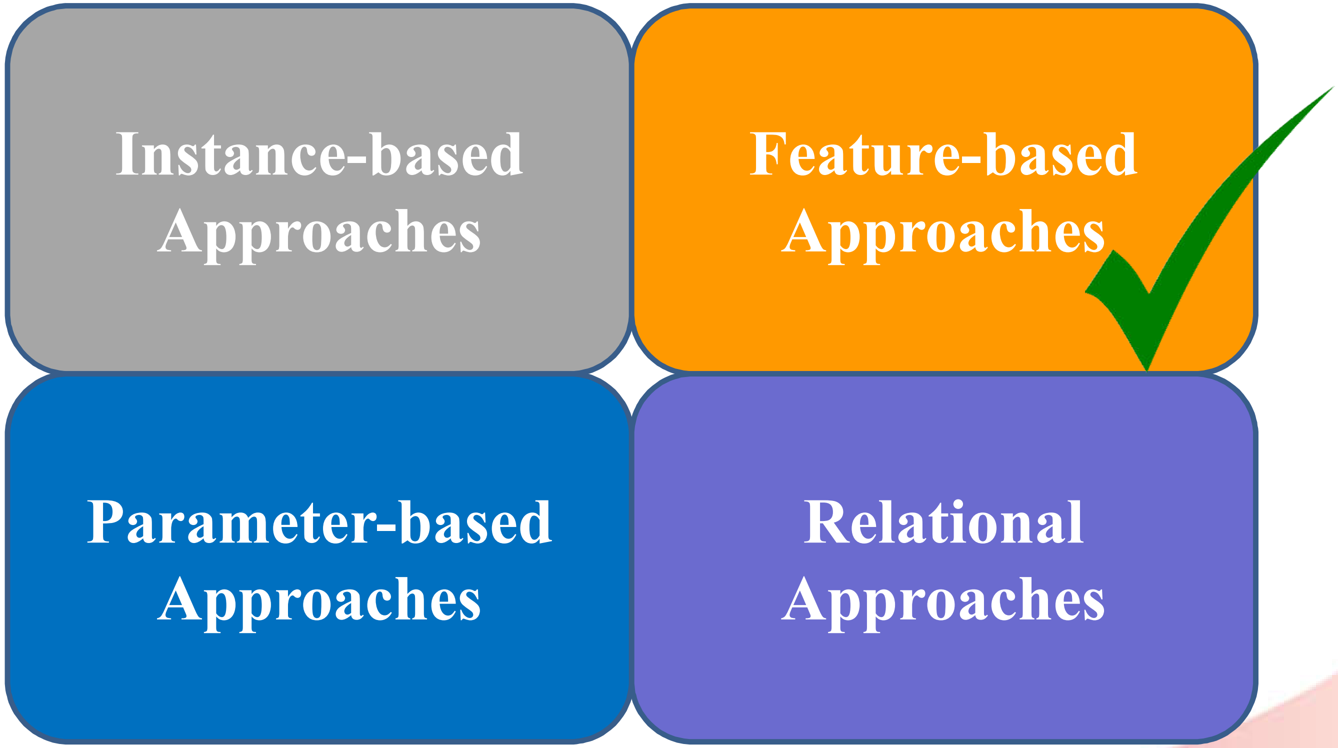
The Goal of Transfer Learning



Transfer Learning Settings



Transfer Learning Approaches



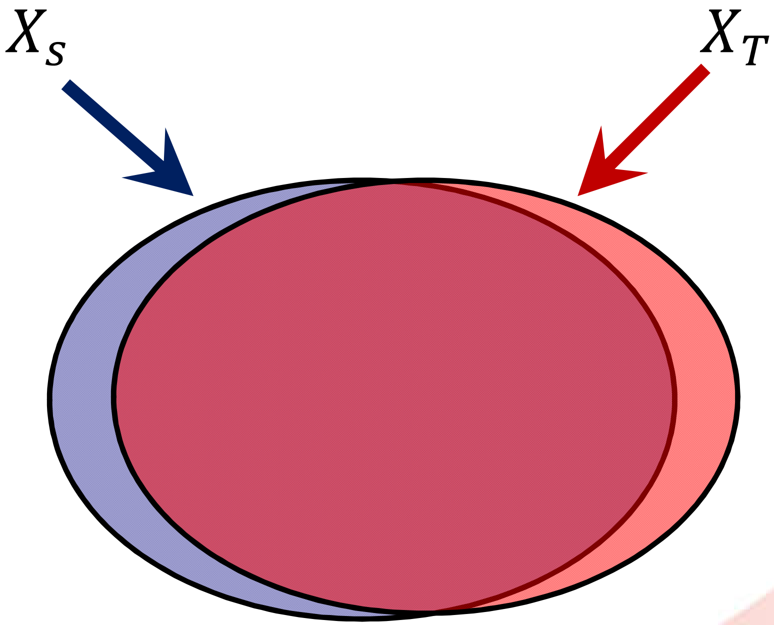
Instance-based Approaches

General Assumption

Source and target domains have a lot of overlapping features (domains share the same/similar support)

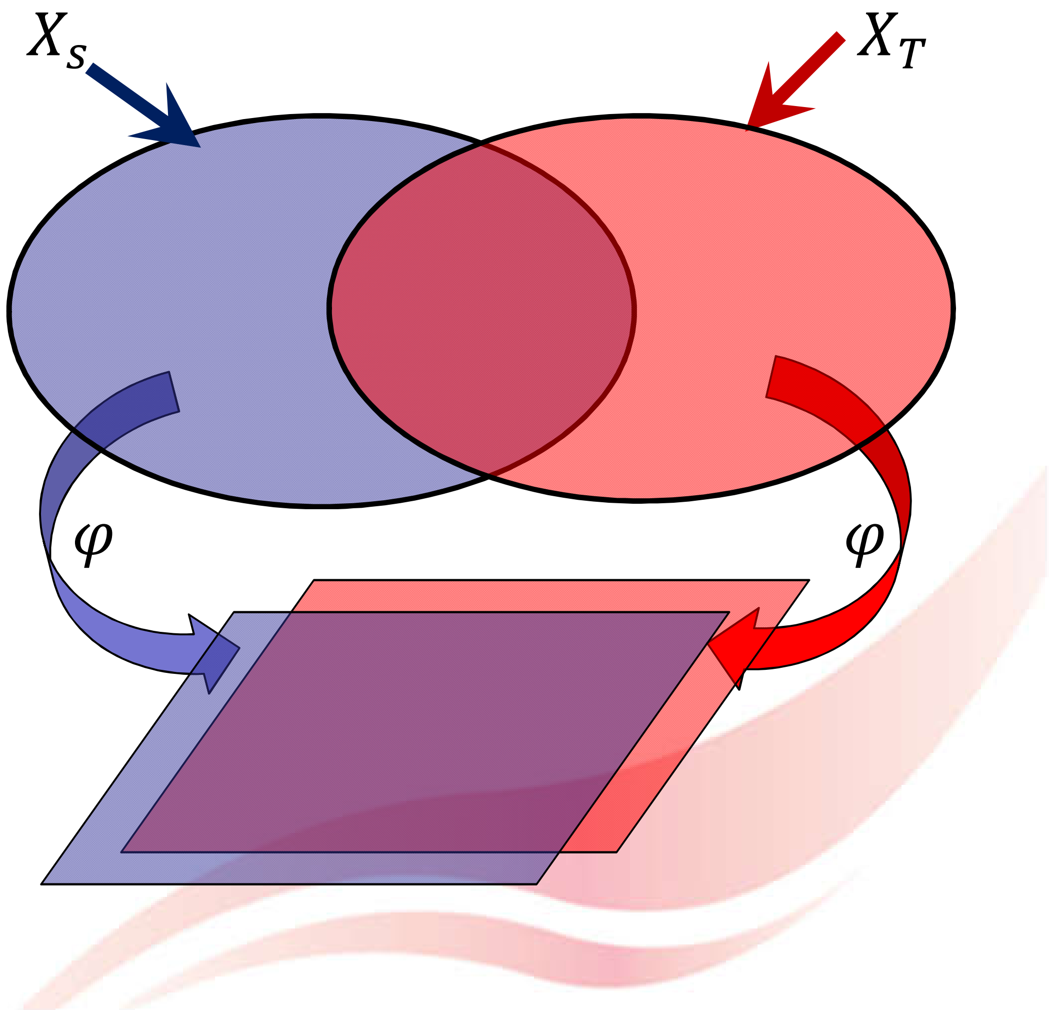
Motivation

Reweight source-domain labeled data to be reused for the target domain



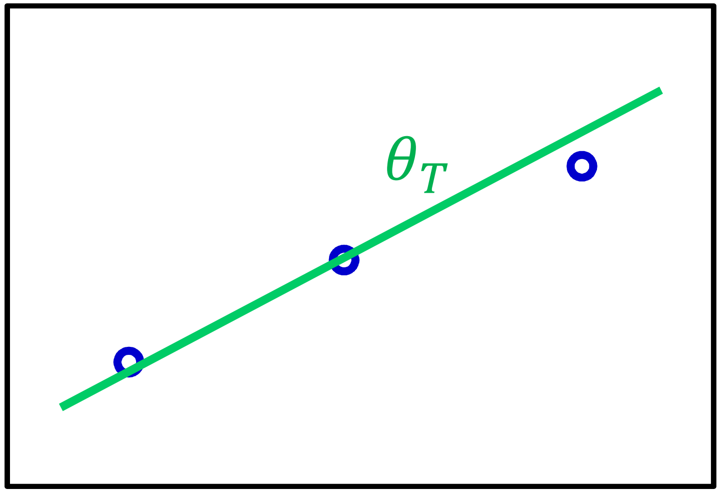
Feature-based Approaches

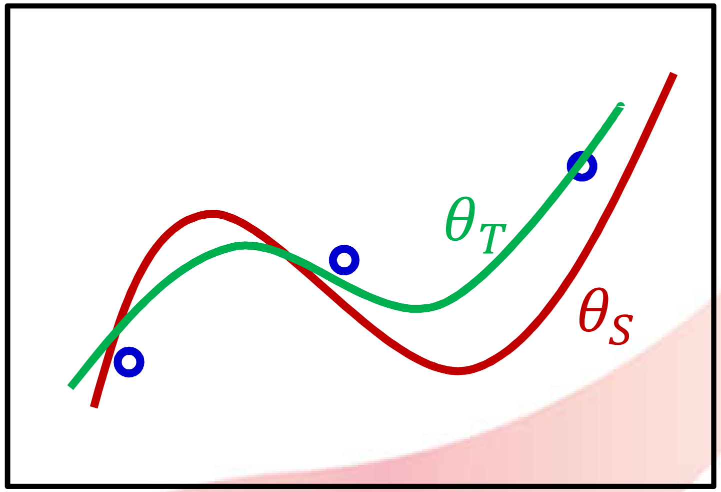
When source and target domains only have some overlapping features. (lots of features only have support in either the source or the target domain)



Parameter-based Approaches

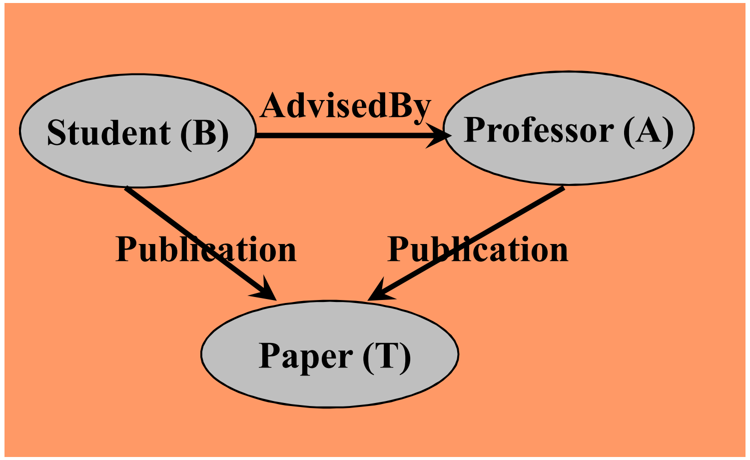
- **Motivation:** A well-trained source model θ_S has captured a lot of structure from data. If two tasks are related, this structure can be transferred to learn a more precise target model θ_T with a few labeled data in the target domain



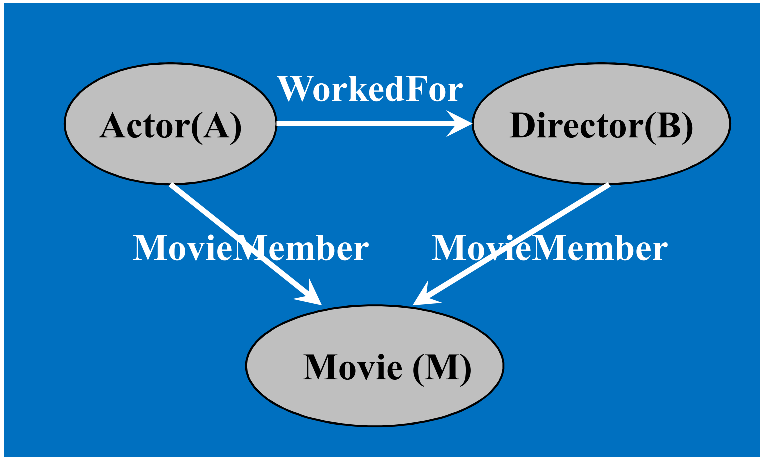


Relational Approaches

Academic domain (source)



Movie domain (target)



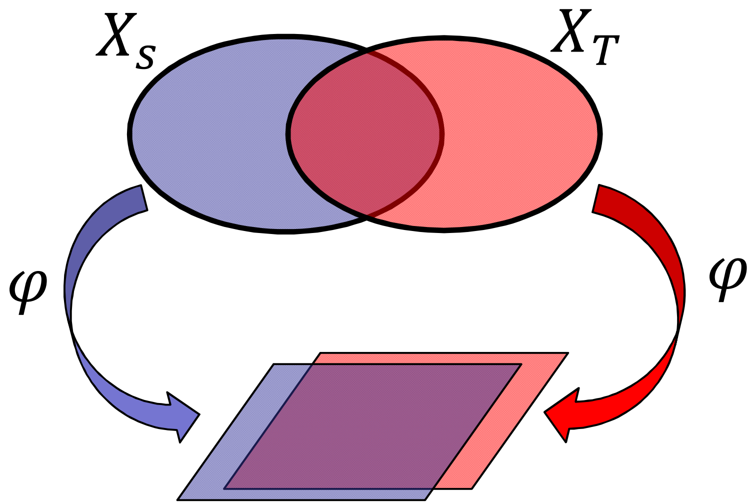
AdvisedBy (B, A) ∧ Publication (B, T)
=> Publication (A, T)

WorkedFor (A, B) ∧ MovieMember (A, M)
=> MovieMember (B, M)

P1(x, y) ∧ P2 (x, z) => P2 (y, z)

Feature-based Approaches (cont.)

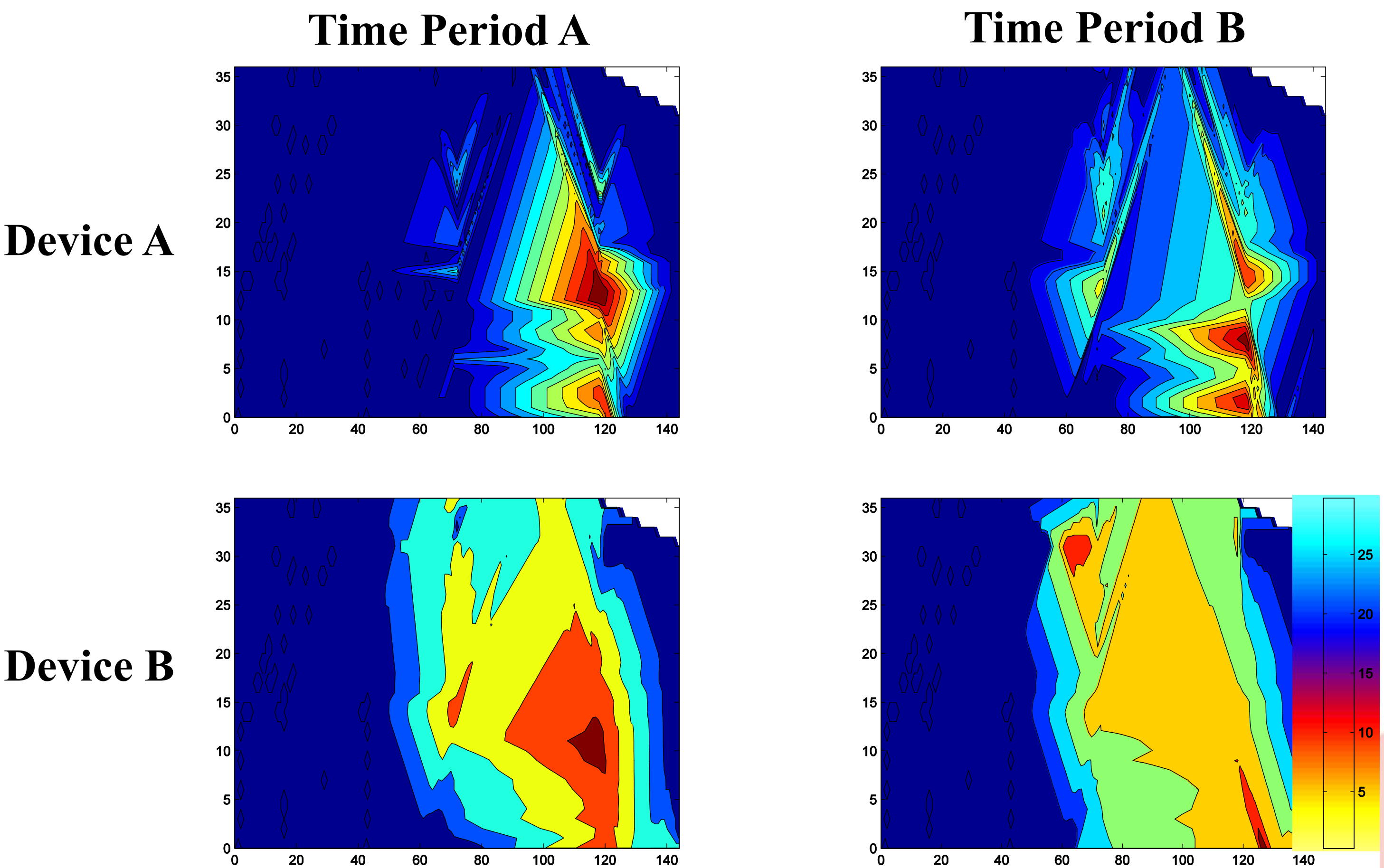
How to learn φ?



- Solution 1: Encode application-specific knowledge to learn the transformation, e.g., sentiment analysis
- Solution 2: General approaches to learning the transformation

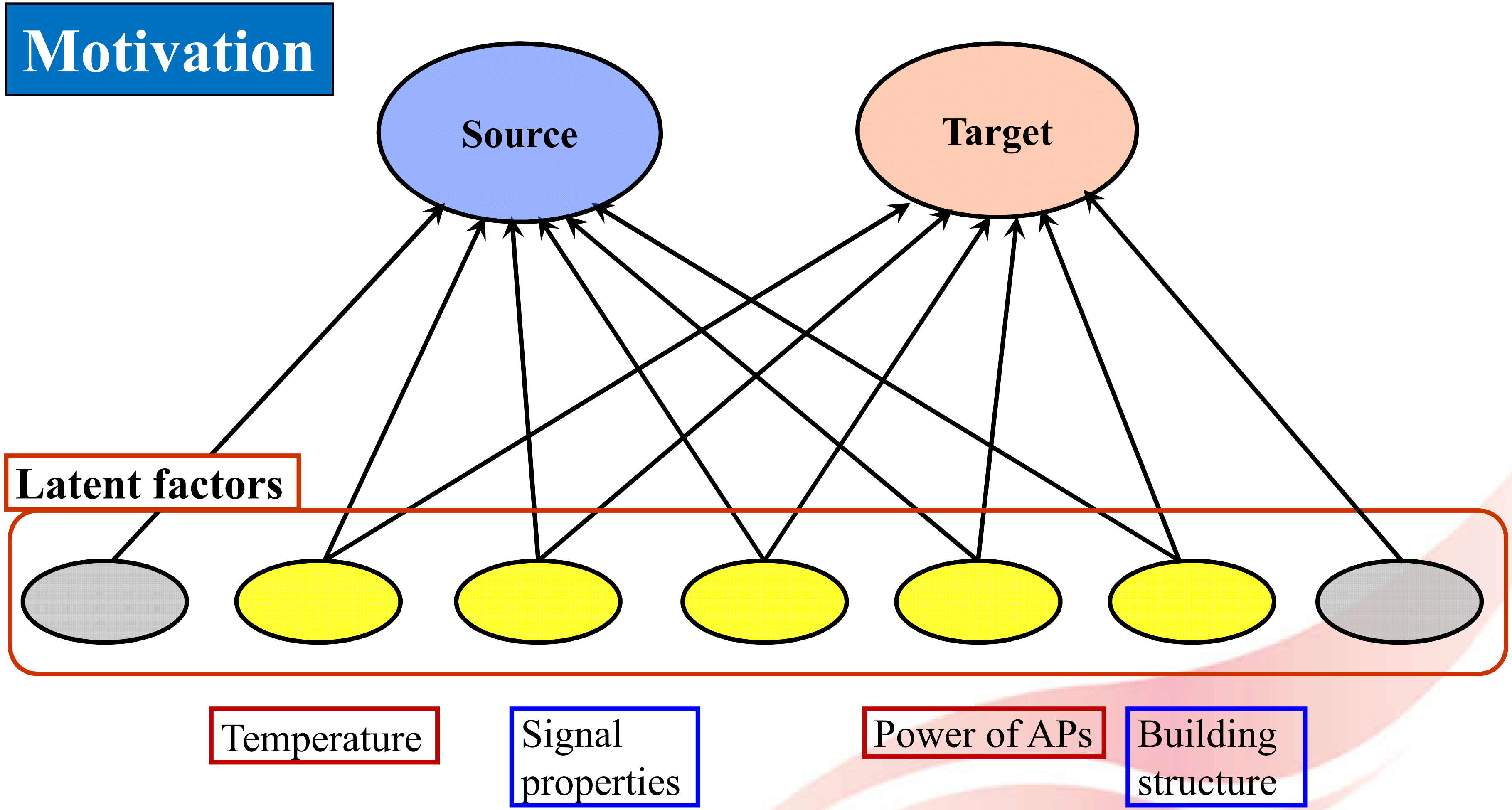
Developing General Approaches

An illustrating Example

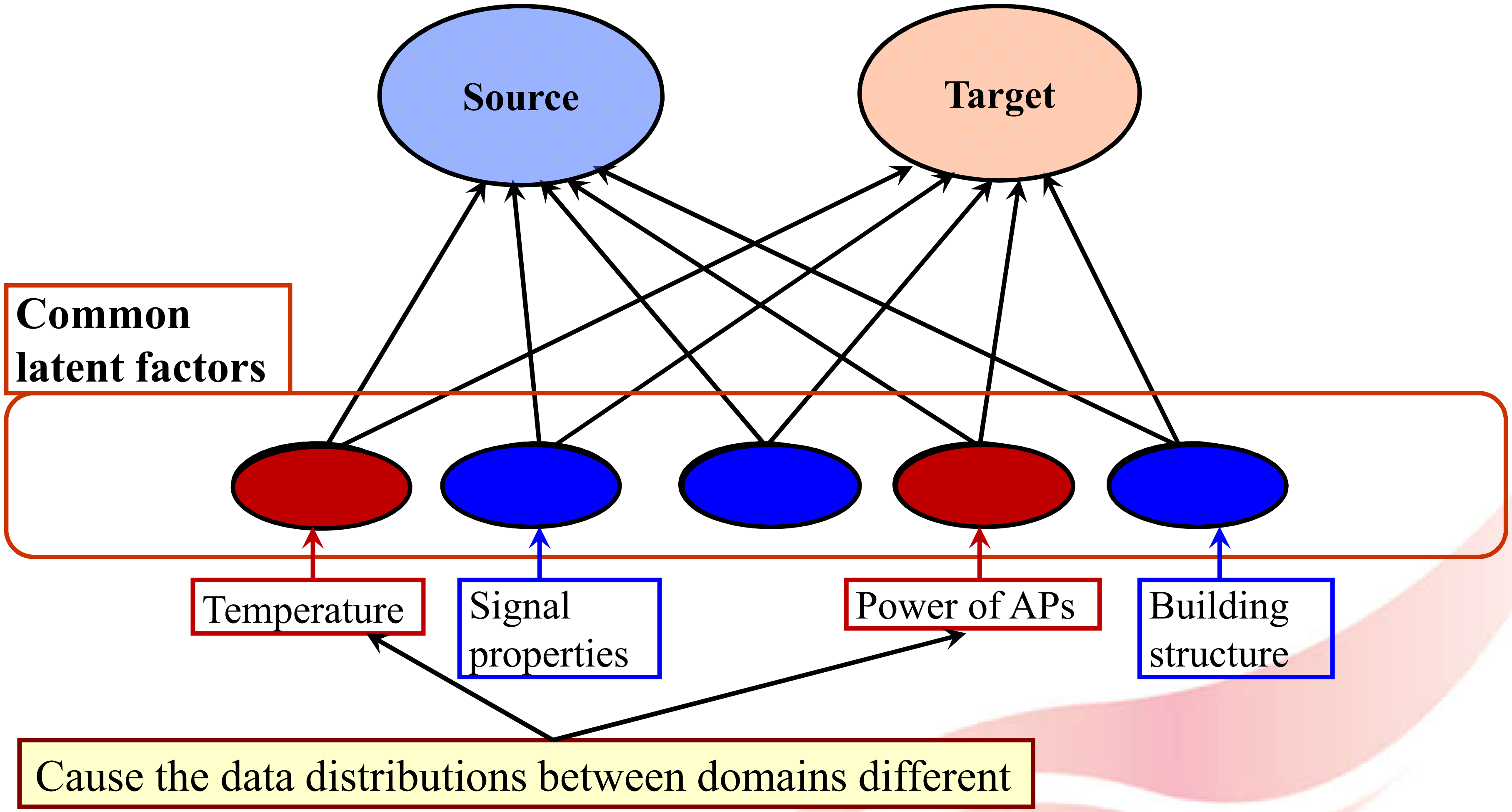


Learning Features via Kernel Embedding of Distributions

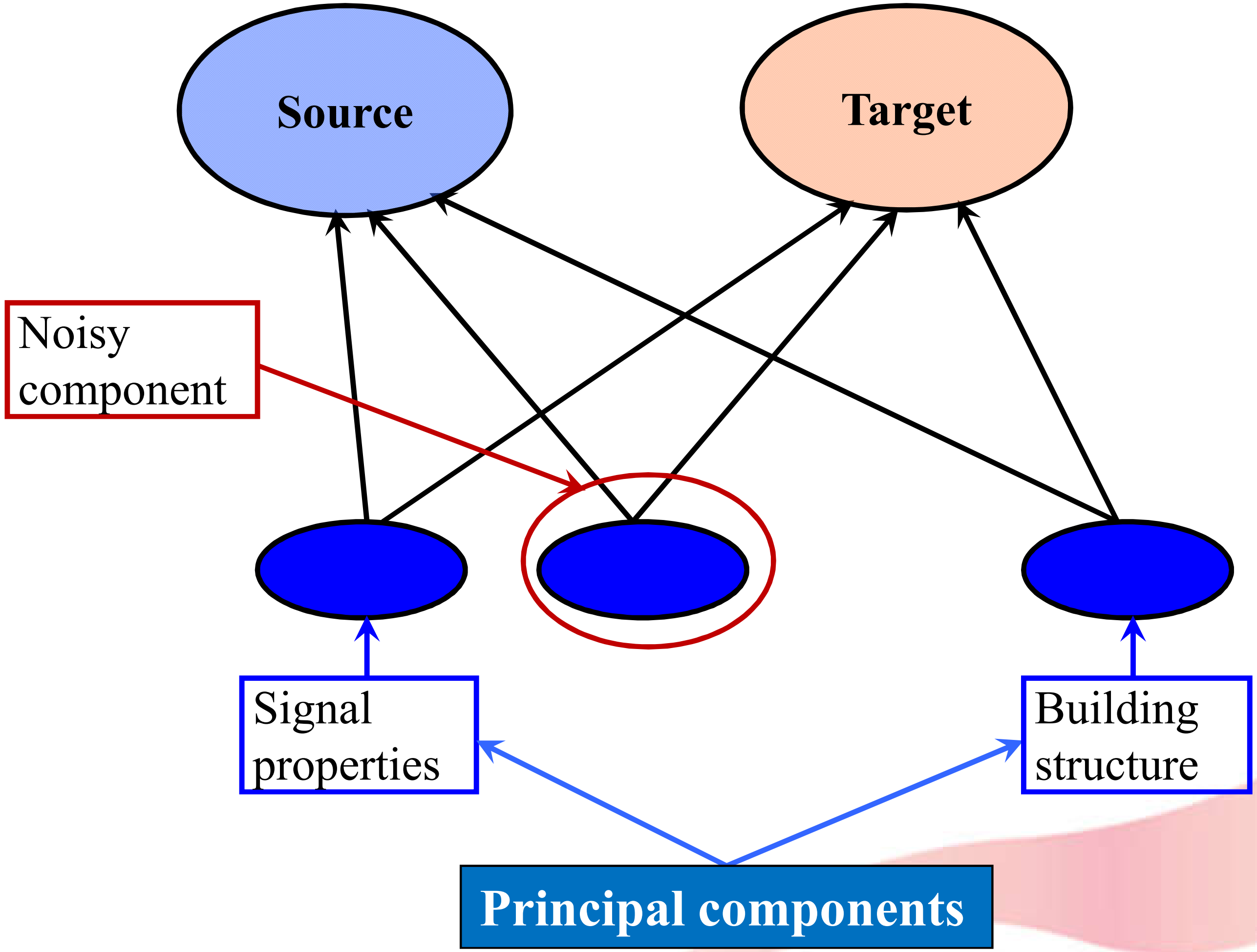
Transfer Component Analysis (TCA) [Pan et al., IJCAI-09, TNN-11]



Transfer Component Analysis (cont.)

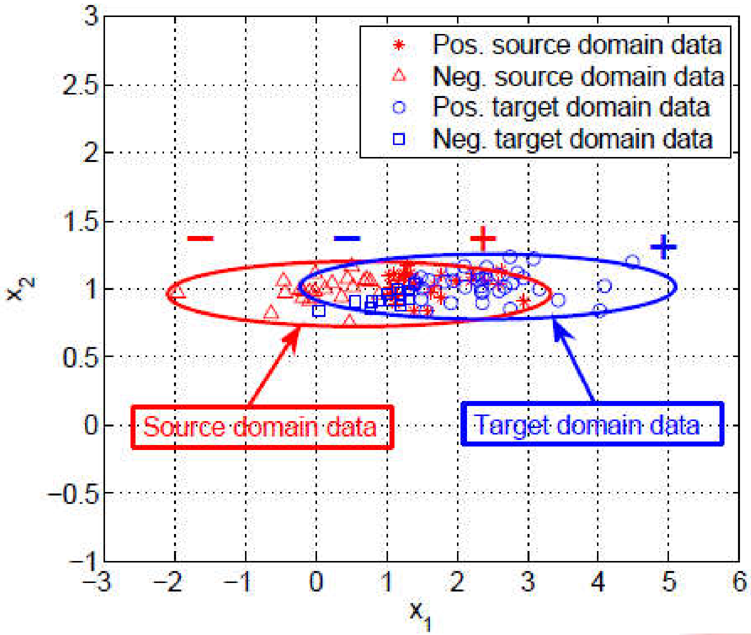


Transfer Component Analysis (cont.)



Transfer Component Analysis (cont.)

- Learning φ by only minimizing distance between distributions may map the data onto noisy factors



Transfer Component Analysis (cont.)

- **Main idea:** the learned φ should map the source domain and target domain data to a latent space spanned by the factors that reduce domain distance as well as preserve data structure
- **High level optimization problem**

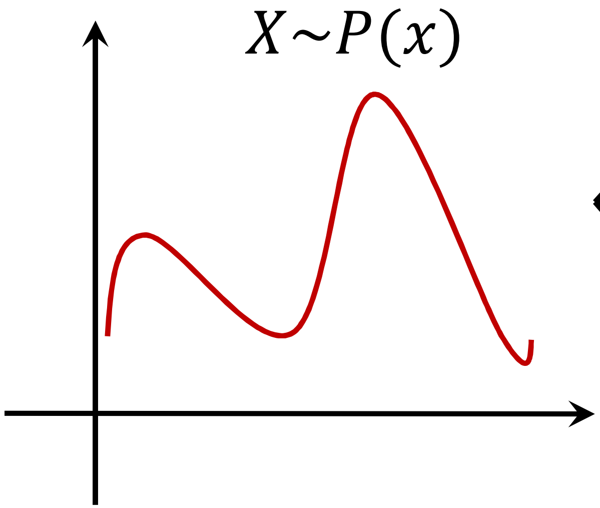
$$\min_{\varphi} \ \boxed{\mathrm{Dist}\big(\varphi(X_S), \varphi(X_T)\big)} + \lambda\Omega(\varphi)$$

$$\text{s.t.} \quad \text{constraints on } \varphi(X_S) \text{ and } \varphi(X_T)$$

Maximum Mean Discrepancy (MMD)

Representing Distributions in RKHS

Can we use a vector μ_X to represent the distribution?



$$\mu_X = (\mathbb{E}[X])$$

$$\mu_X = \begin{pmatrix} \mathbb{E}[X] \\ \mathbb{E}[X^2] \end{pmatrix}$$

$$\mu_X = \begin{pmatrix} \mathbb{E}[X] \\ \mathbb{E}[X^2] \\ \mathbb{E}[X^3] \end{pmatrix}$$

$$\mu_X = \begin{pmatrix} \mathbb{E}[X] \\ \mathbb{E}[X^2] \\ \mathbb{E}[X^3] \\ \cdots \\ \cdots \end{pmatrix}$$

But, infinite, cannot be explicitly computed!

What about $\mu_X \in \mathcal{H}$ (RKHS)?

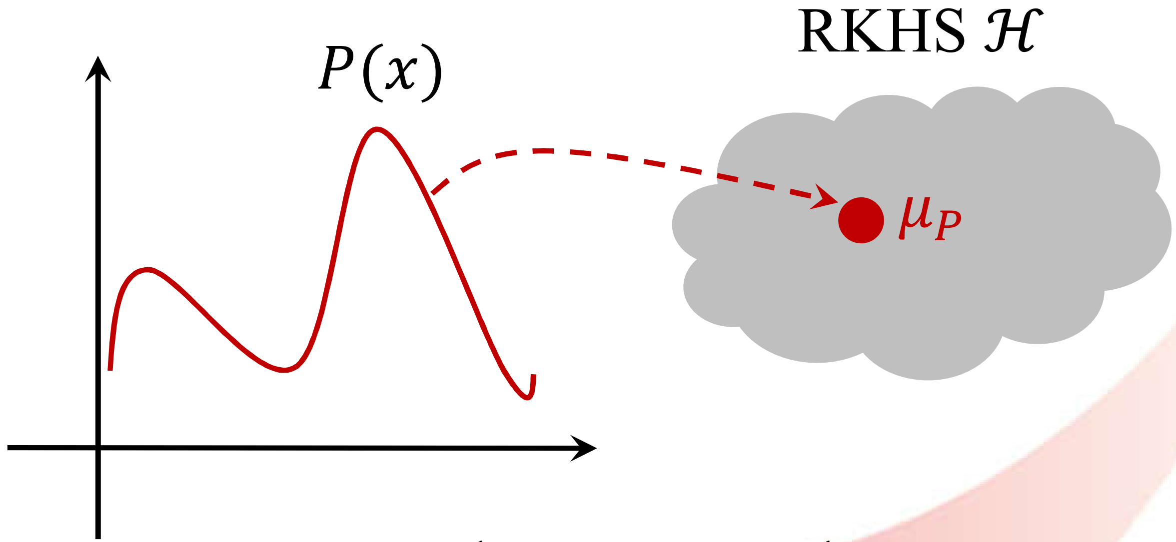
Kernel trick can be applied!

Mean Map in RHKS

Suppose $X \sim P(x)$, and denote $k(x,\cdot) = \phi_x$

Mean map: $\mu_P = \mathbb{E}_{x\sim P(x)}[\phi_x]$

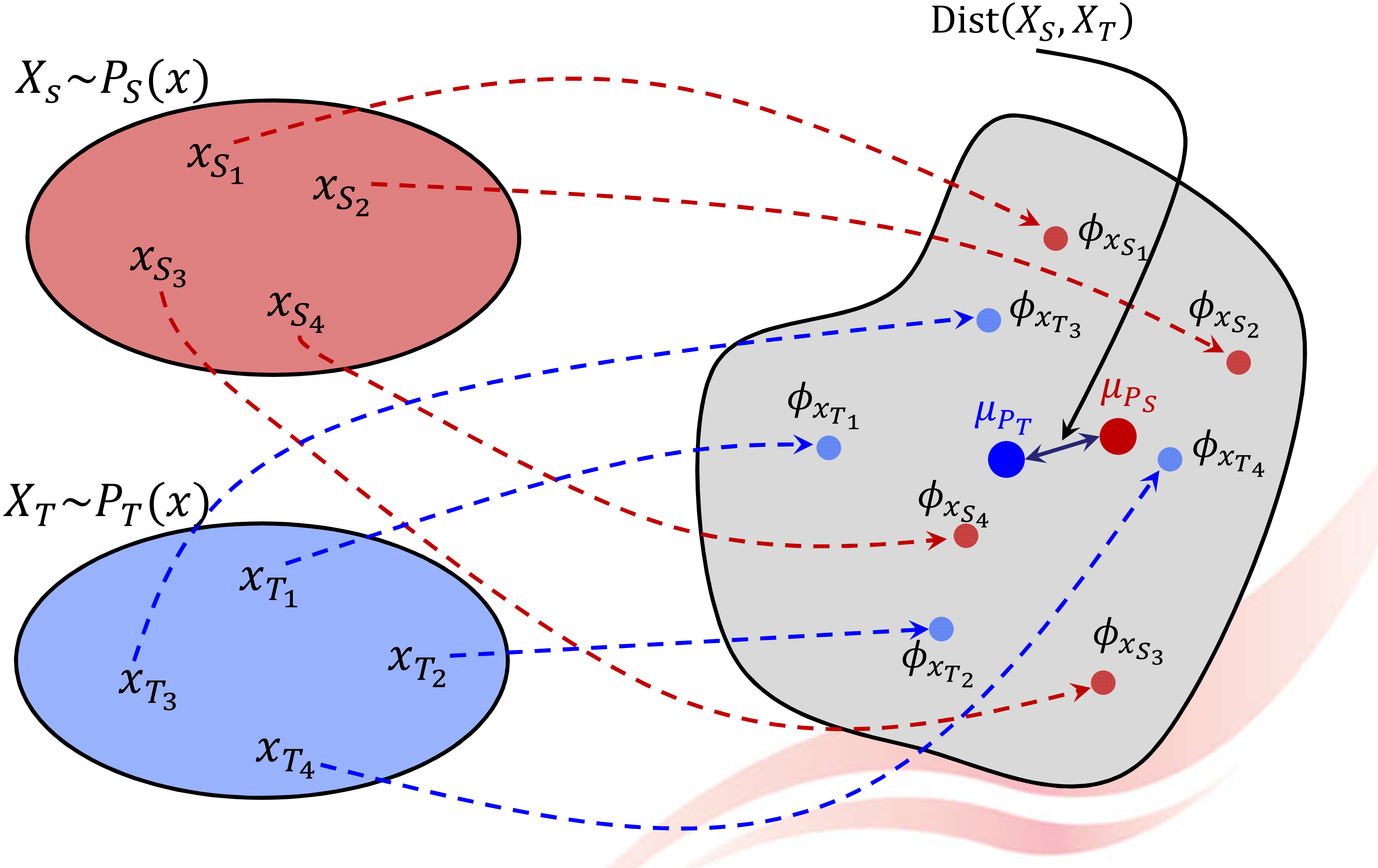
$$\mu_P : P \rightarrow \int \phi_x \mathrm{d}P(x)$$



Empirical mean map for $\{x_i\}_{i=1}^n$: $\hat{\mu}_P = \frac{1}{n}\sum_{i=1}^n \phi_{x_i} = \frac{1}{n}\sum_{i=1}^n k(x_i,\cdot)$

[Berlinet and Thomas-Agnan 2004; Smola et al. ALT-07]

Mean Map in RHKS (cont.)



Distance Measure via MMD

$$\mathrm{Dist}\big(\varphi(X_S),\varphi(X_T)\big) = \left\|\mathbb{E}_{x\sim P_T(x)}\big[\phi\big(\varphi(x)\big)\big] - \mathbb{E}_{x\sim P_S(x)}\big[\phi\big(\varphi(x)\big)\big]\right\|_{\mathcal{H}}$$

$$\approx \left\|\frac{1}{n_T}\sum_{i=1}^{n_T}\phi\left(\varphi\big(x_{T_i}\big)\right) - \frac{1}{n_S}\sum_{i=1}^{n_S}\phi\left(\varphi\big(x_{S_i}\big)\right)\right\|_{\mathcal{H}}$$

Assume $\psi = \phi \circ \varphi$ be a RKHS with kernel $k\big(x_i, x_j\big) = \psi(x_i)^T\psi(x_i)$

$$\mathrm{Dist}\big(\varphi(X_S),\varphi(X_T)\big)^2 = \left\|\frac{1}{n_T}\sum_{i=1}^{n_T}\psi\big(x_{T_i}\big) - \frac{1}{n_S}\sum_{i=1}^{n_S}\psi\big(x_{S_i}\big)\right\|_{\mathcal{H}}^2 = \mathrm{tr}(KL)$$

$$K = \begin{bmatrix} K_{S,S} & K_{S,T} \\ K_{T,S} & K_{T,T} \end{bmatrix}$$

$$L_{ij} = \begin{cases} \dfrac{1}{n_S^2} & x_i, x_j \in X_S \\ \dfrac{1}{n_T^2} & x_i, x_j \in X_T \\ -\dfrac{1}{n_S n_T} & \text{otherwise} \end{cases}$$

Transfer Component Analysis (cont.)

$$\min_{\varphi} \quad \mathrm{Dist}\big(\varphi(X_S), \varphi(X_T)\big) + \lambda\Omega(\varphi)$$
$$\text{s.t.} \quad \text{constraints on } \varphi(X_S) \text{ and } \varphi(X_T)$$

$$\Downarrow$$

$$\min_{\varphi} \quad \mathrm{tr}(KL) + \lambda\Omega(\varphi)$$
$$\text{s.t.} \quad \text{constraints on } \varphi(X_S) \text{ and } \varphi(X_T)$$

- In general, the kernel function $k(\varphi(x_i), \varphi(x_j))$ can be a highly nonlinear function of φ that is unknown
- A direct optimization of minimizing the quantity w.r.t. φ may get stuck in poor local minima

Solution I [Pan *etal*., AAAI-08]

Learning φ →

1) Learning K
2) Low-dimensional reconstructions of X_S and X_T based on K

Minimize the distance between domains

Maximize data variance

Preserve the local geometric structure

1) $$\min_{K \succcurlyeq 0} \operatorname{tr}(KL) + \lambda \operatorname{tr}(K)$$
$$\text{s.t. } K_{ii} + K_{jj} - 2K_{ij} = d_{ij}^2, \forall (i,j) \in \mathcal{N},$$
$$K\mathbf{1} = 0.$$

2) Perform PCA on K

- It is a SDP problem, expensive!
- It is transductive, cannot generalize on unseen instances!
- PCA is post-processed on the learned kernel matrix, which may potentially discard useful information

Solution II [Pan *et al.*, IJCAI-09, IEEE TNN-11]

Known, given by user

Assume K be low-rank, then $K = \bar{K}WW^T\boxed{\bar{K}}$

$W \in \mathbb{R}^{(n_S+n_T)\times m}$ and $m \ll n_S + n_T$

Learning K ➡ Learning a low-rank matrix W

Minimize distance between domains

$\mathrm{tr}(W^T\bar{K}L\bar{K}W)$

$=$

$$\min_W \ \mathrm{tr}(\bar{K}WW^T\bar{K}L) + \lambda\,\mathrm{tr}(W^TW)$$

Regularization term on W

$$\text{s.t.} \ W^T\bar{K}H\bar{K}W = I$$

Maximize data variance

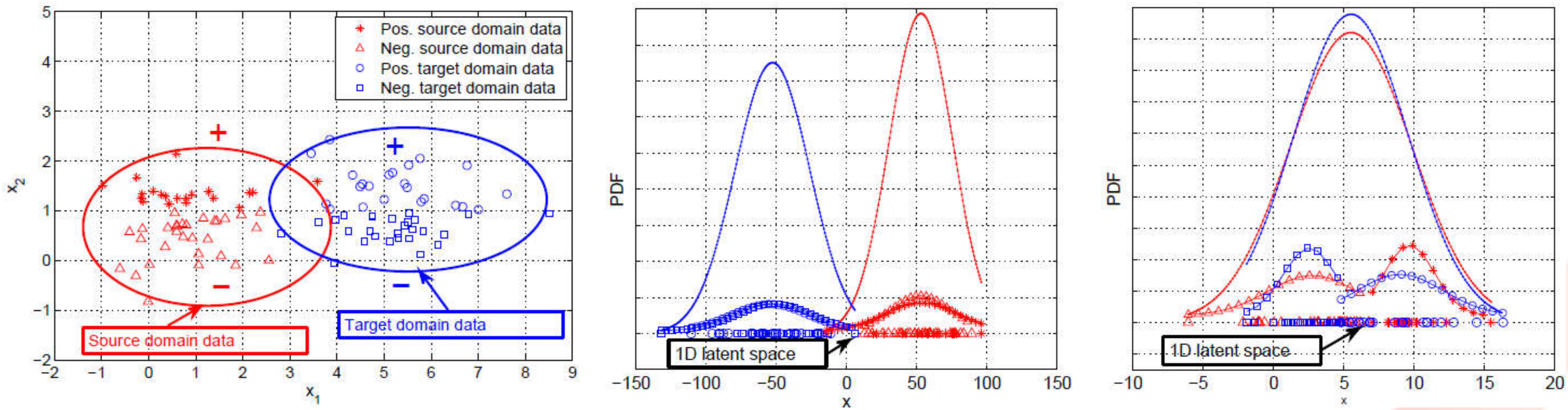
Closed form solution for W^*:

m leading eigenvectors of $(\bar{K}L\bar{K} + \lambda I)^{-1}\bar{K}H\bar{K}$

Transfer Component Analysis (cont.)

An illustrative example

Latent features learned by **PCA** *and* **TCA**



Original feature space

PCA

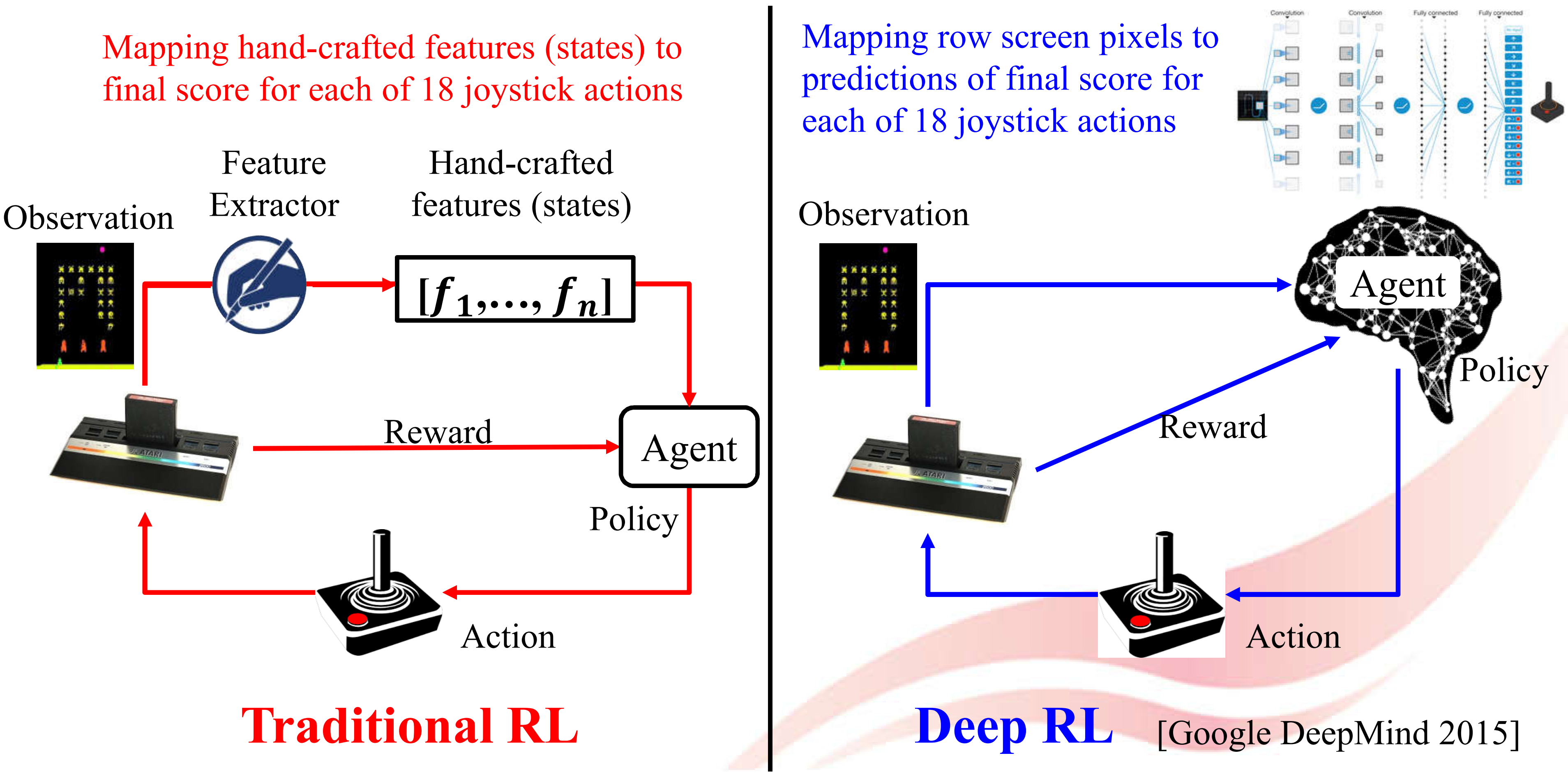
TCA

Future Direction

- Theoretical study beyond generalization error bound
  - Given a source domain and a target domain, determine whether transfer learning should be performed
  - For a specific transfer learning method, given a source and a target domain, determine whether the method should be used for knowledge transfer

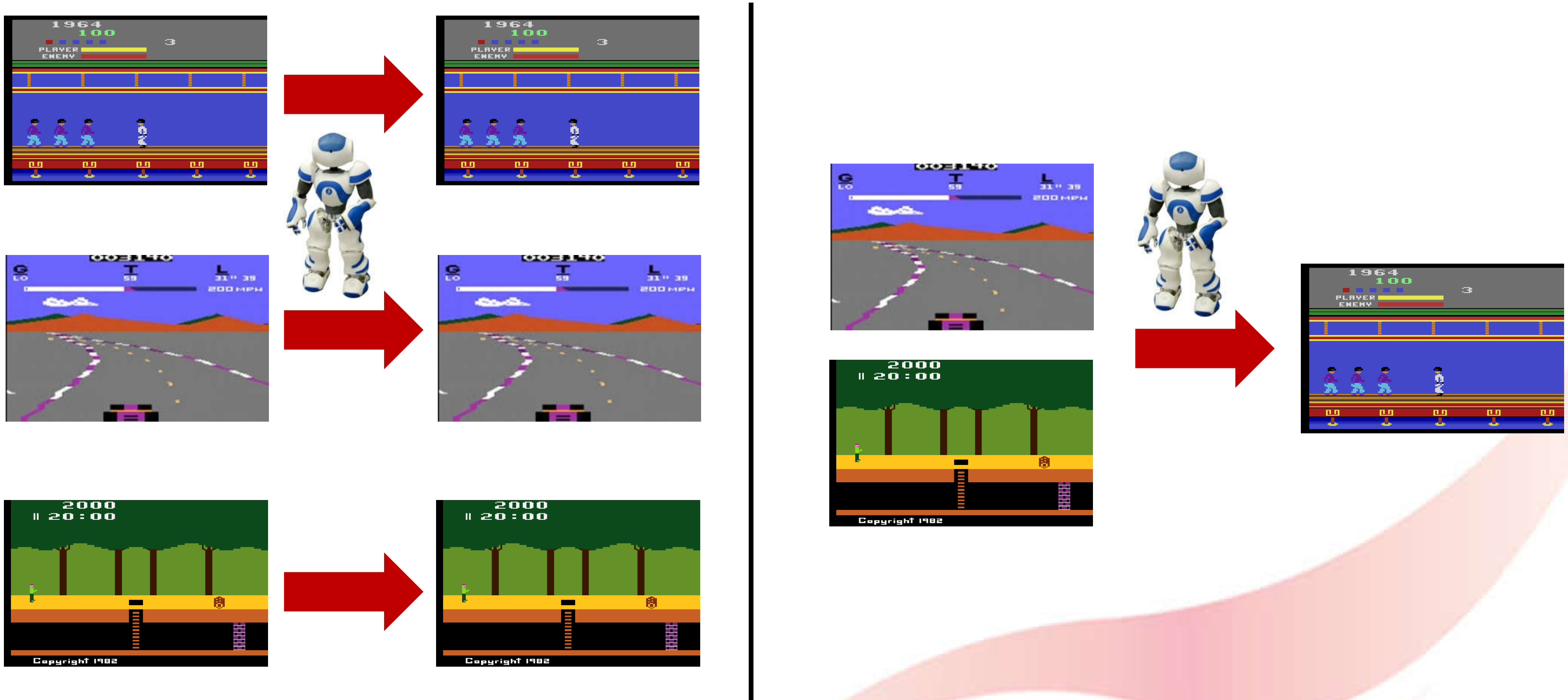
Future Direction (cont.)

- Transfer learning for deep reinforcement learning



Future Direction (cont.)

- Transfer learning for deep reinforcement learning



Deep RL | **Transfer Learning for Deep RL**

Reference

- **Pan** and Yang, A Survey on Transfer Learning, IEEE TKDE 2010
- **Pan**, Transfer learning, Data Classification: Algorithms and Applications (Chapter 21), 2014
- Taylor and Stone, Transfer Learning for Reinforcement Learning Domains: A Survey, JMLR 2009
- **Pan**, Kwok, and Yang, Transfer Learning via Dimensionality Reduction, AAAI 2008
- **Pan**, Tsang, Kwok, and Yang, Domain Adaptation via Transfer Component Analysis, IJCAI 2009, IEEE TNN 2011
- Alex Smola, Arthur Gretton, Le Song, and Bernhard Scholkopf, A Hilbert Space Embedding for Distributions, ALT 2007
- A. Berlinet and C. Thomas-Agnan, Reproducing Kernel Hilbert Spaces in Probability and Statistics. Kluwer Academic Publisher 2004

Thank You!