Kernel Methods for Domain Adaptation

Boqing Gong
University of Southern California

*Joint work with **Kristen Grauman** and **Fei Sha***

Vision datasets

Key to computer vision research

Instrumental to benchmark different methods

But, datasets are **biased**

Need **undo bias** when developing vision systems

IMAGENET

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

SUN

Unsupervised domain adaptation (DA)

Setup

Source domain (with labeled data)

$$D_{\mathcal{S}} = \{(x_m, y_m)\}_{m=1}^{\mathsf{M}} \sim P_{\mathcal{S}}(X, Y)$$

Target domain (no labels for training)

$$D_{\mathcal{T}} = \{(x_n, \textbf{?})\}_{n=1}^{\mathsf{N}} \sim P_{\mathcal{T}}(X, Y)$$

Unsupervised domain adaptation (DA)

Setup

Source domain (with labeled data)

$$D_{\mathcal{S}} = \{(x_m, y_m)\}_{m=1}^{\mathsf{M}} \sim \boxed{P_{\mathcal{S}}(X, Y)}$$

Target domain (no labels for training)

$$D_{\mathcal{T}} = \{(x_n, \textbf{?})\}_{n=1}^{\mathsf{N}} \sim \boxed{P_{\mathcal{T}}(X, Y)}$$

Different distributions

Objective

Learn classifier to work well on target

Unsupervised DA is ill-posed

Image credits: https://www.indiegogo.com/projects/more-than-enough

Unsupervised DA is ill-posed

Make assumptions

- Covariate shift, target shift, sample selection bias, etc.

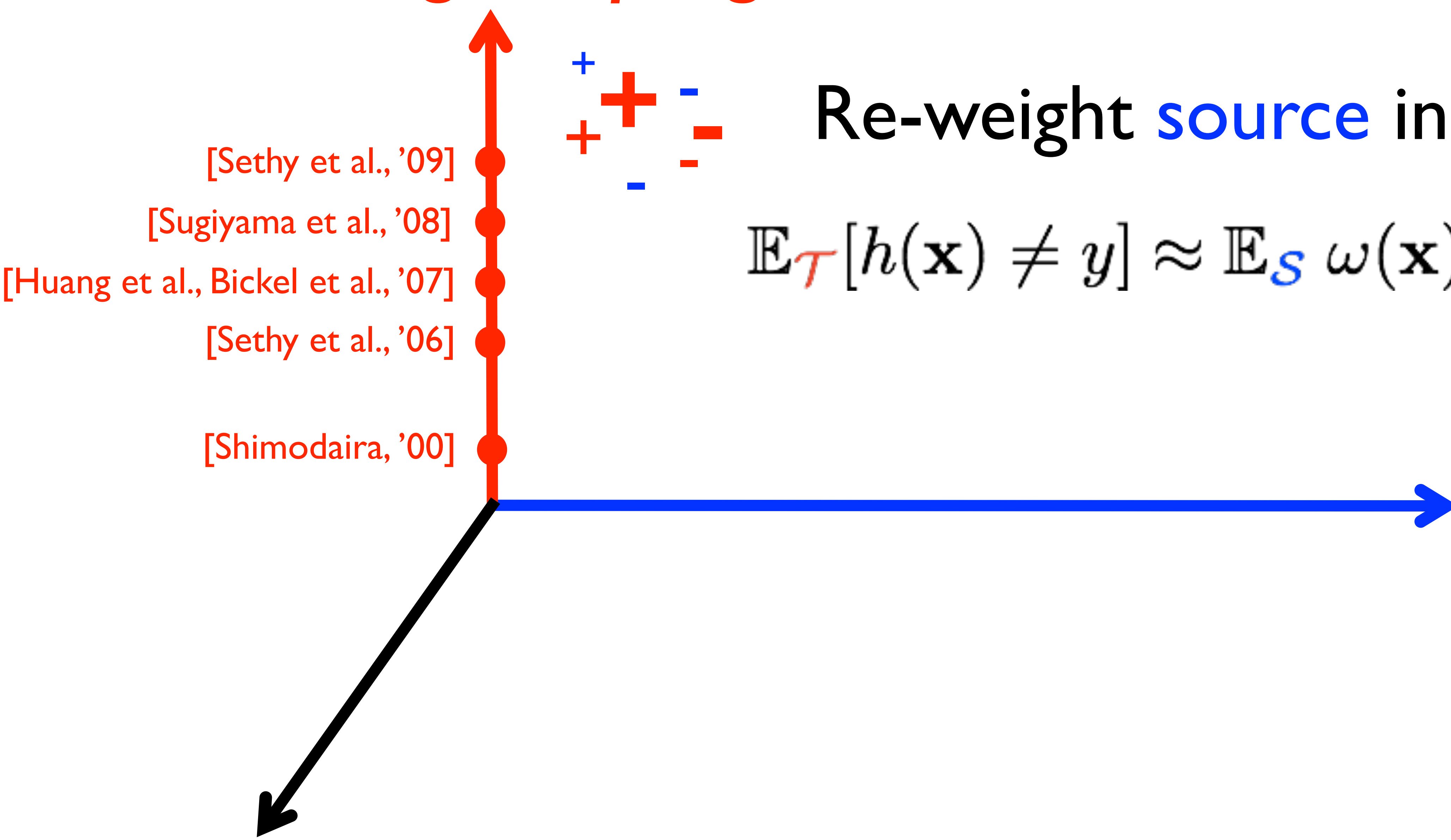
Need domain knowledge

- Exploring intrinsic structures in data (Subspace, cluster, manifold, **landmarks**, etc.)

Image credits: https://www.indiegogo.com/projects/more-than-enough

Background - quick review

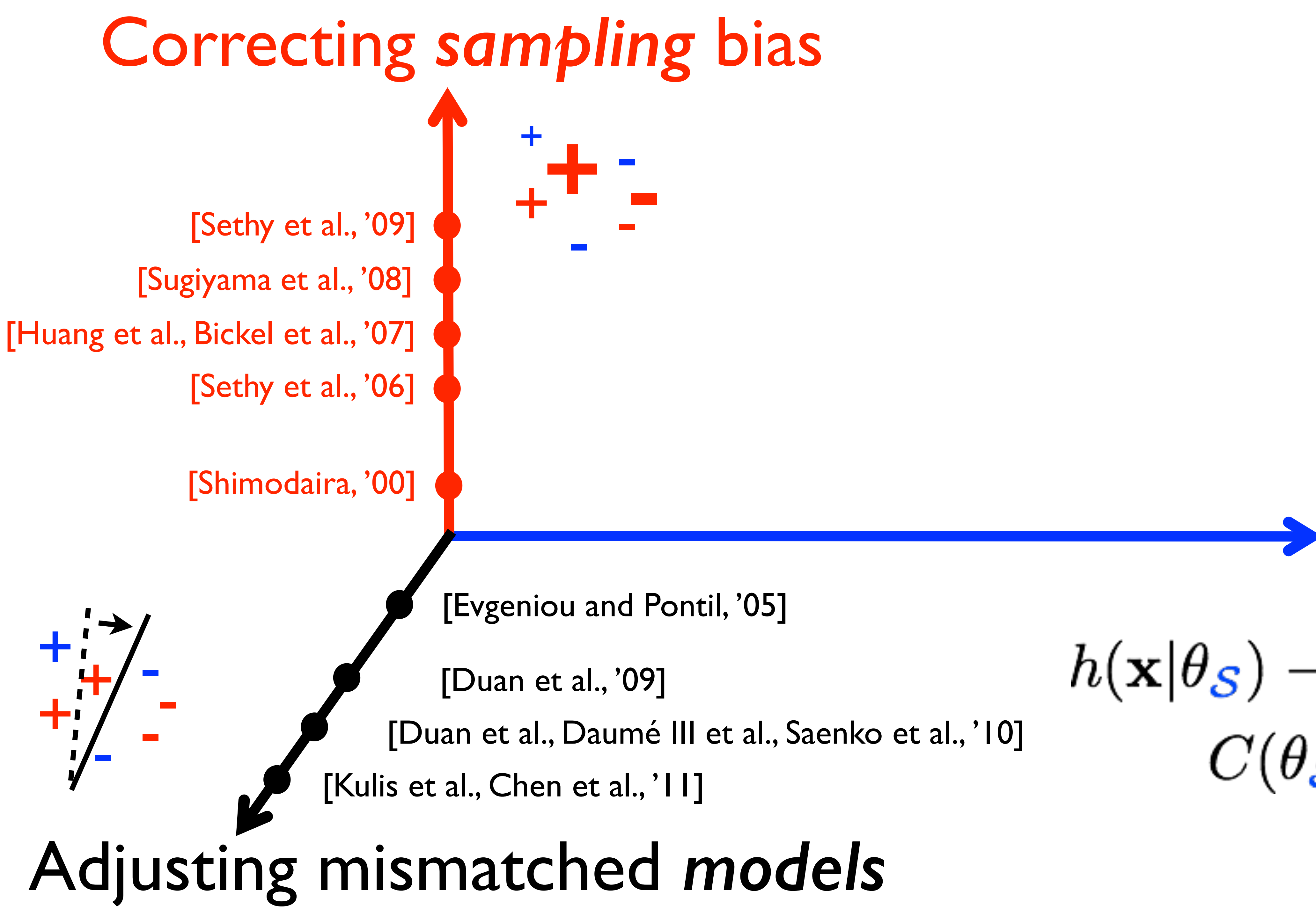
Correcting *sampling* bias



Re-weight source instances

$$\mathbb{E}_{\mathcal{T}}[h(\mathbf{x}) \neq y] \approx \mathbb{E}_{\mathcal{S}}\, \omega(\mathbf{x})\ [h(\mathbf{x}) \neq y]$$

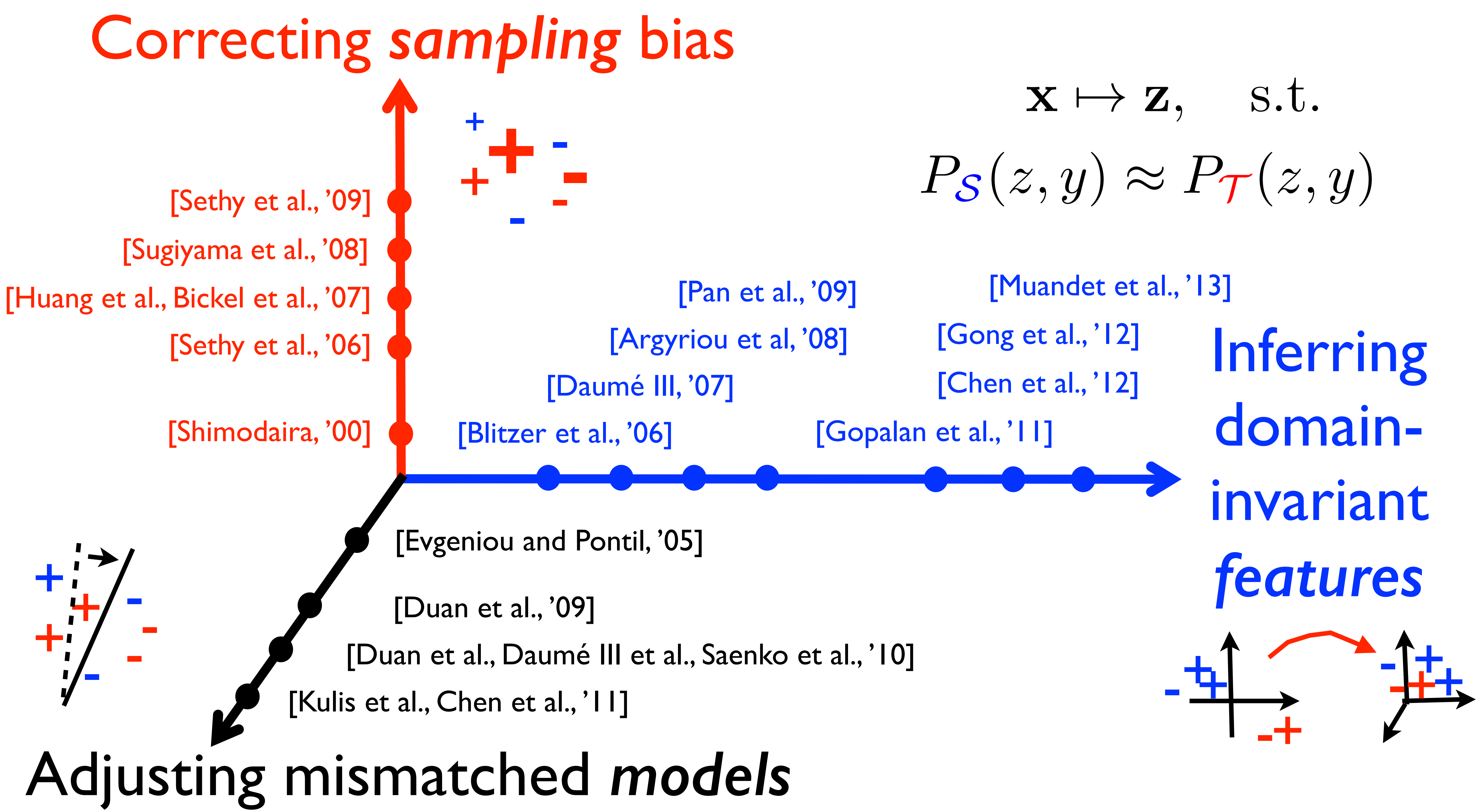
Background - quick review



$$h(\mathbf{x}|\theta_{\mathcal{S}}) \rightarrow h(\mathbf{x}|\theta_{\mathcal{T}})$$

$$C(\theta_{\mathcal{S}}, \theta_{\mathcal{T}}) \leq \epsilon$$

Background - quick review

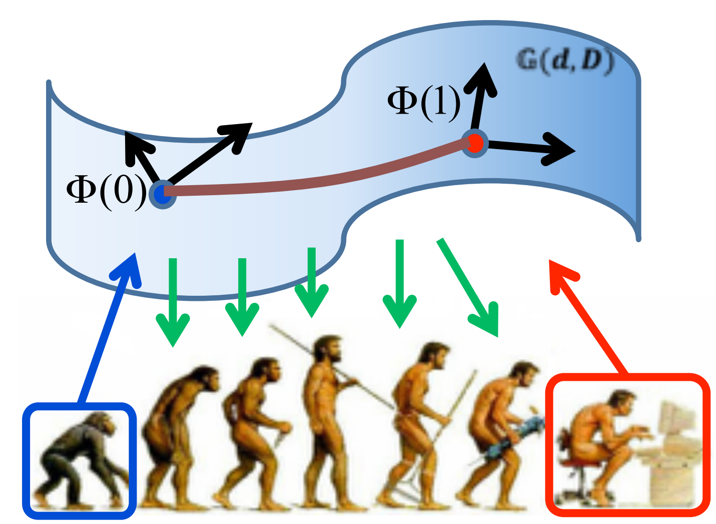


Key: to reduce source-target discrepancy

Our solution: **kernel** methods for

Inferring domain-invariant features

Directly matching distributions



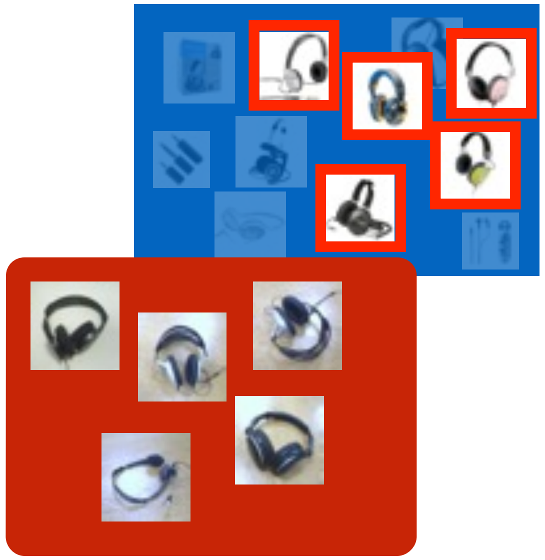
Geodesic flow kernel

Key: to reduce source-target discrepancy

Our solution: **kernel** methods for

Inferring domain-invariant features

Directly matching distributions

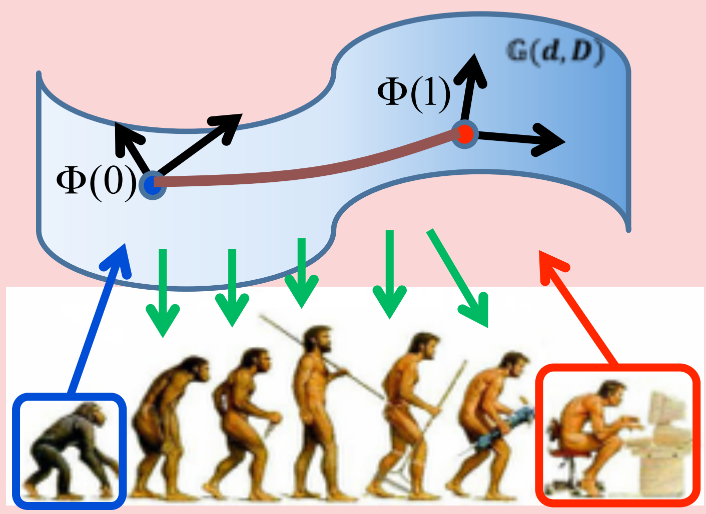


Landmarks

Latent domains

Kernel methods for DA

Inferring domain-invariant features



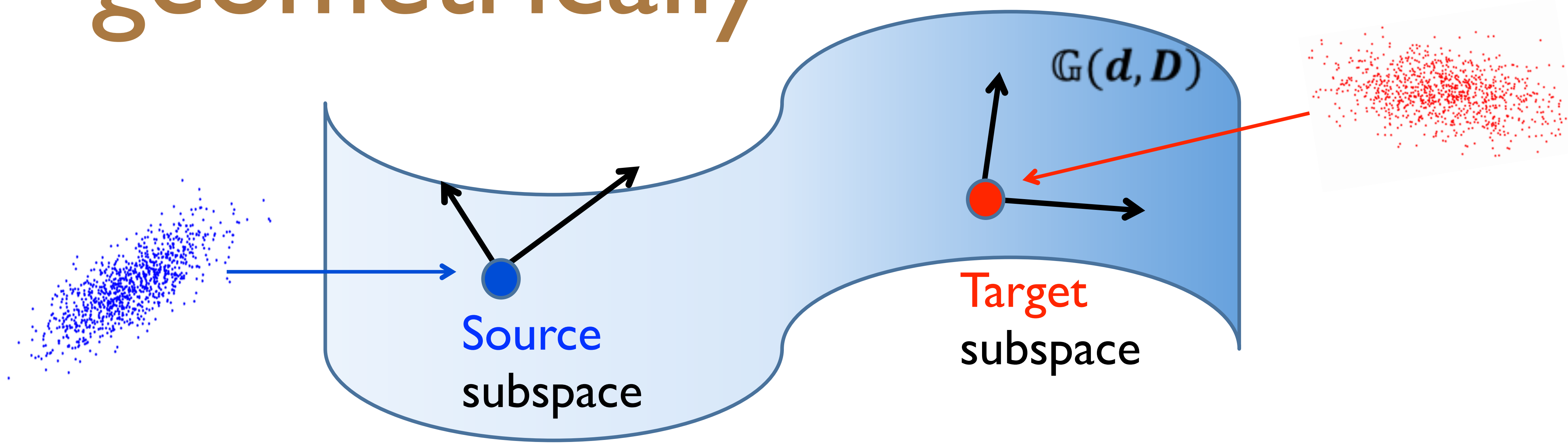
Geodesic flow kernel

Modeling data via subspaces

Assume low-dimensional structure

E.g., PCA, LDA, partial least squares

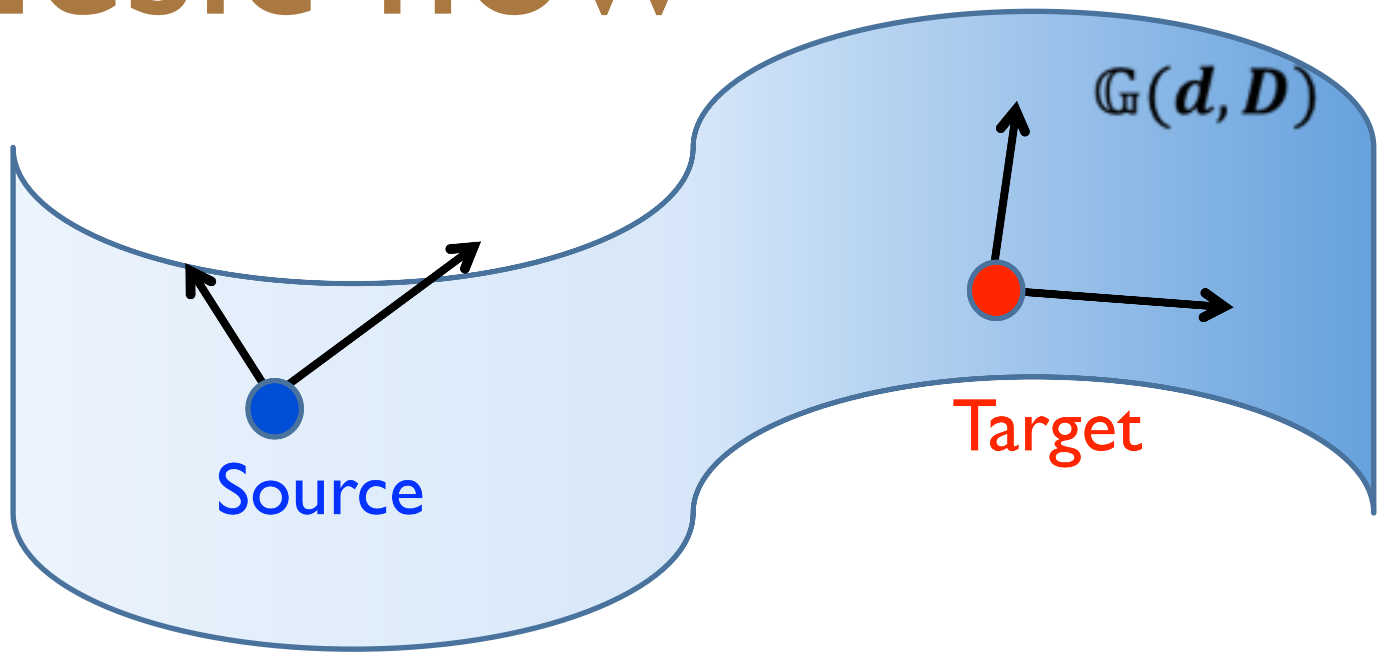
Characterizing domains geometrically



Grassmann manifold $\mathrm{G}(d, D)$

- Collection of *d*-dim subspaces of a vector space $\mathbf{R}^D$ $(d < D)$
- Each point corresponds to a subspace

Modeling domain shift with geodesic flow



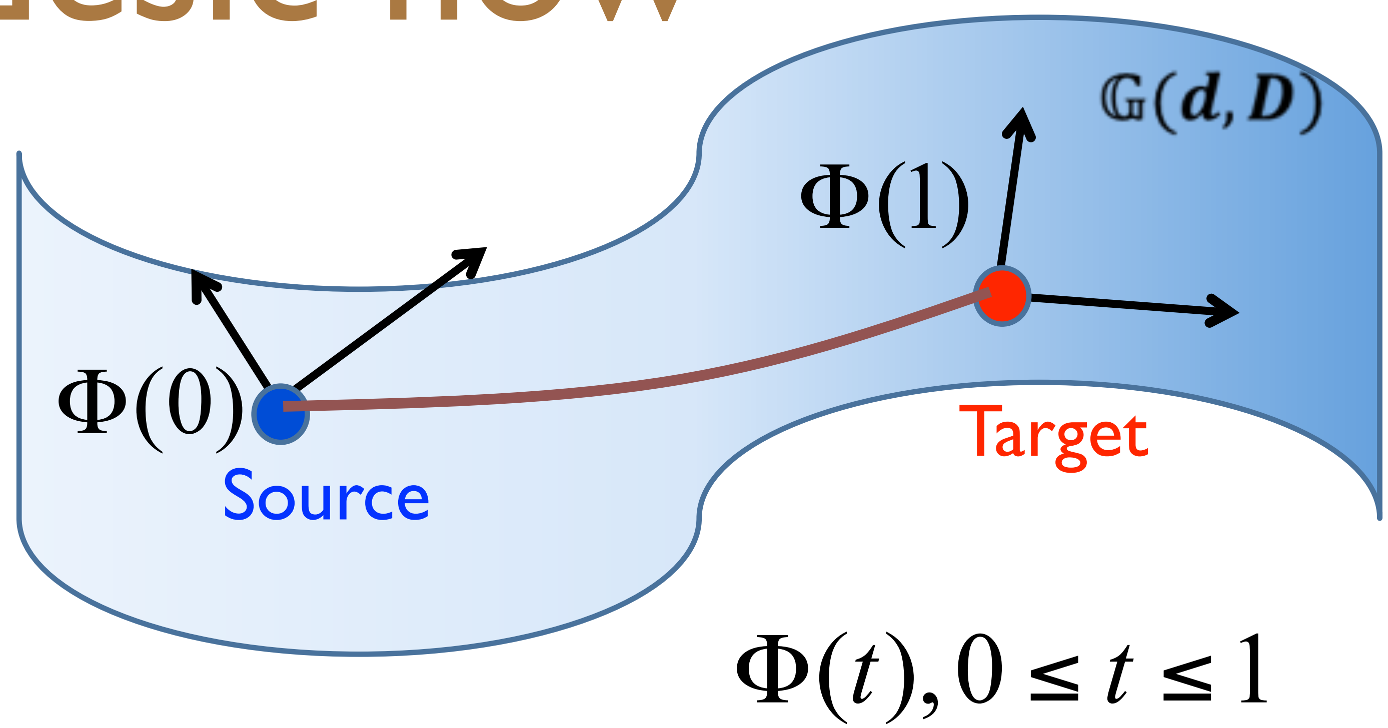
Geodesic flow on the manifold

– starting at source & arriving at target in unit time

– flow parameterized with one parameter t

– closed-form, easy to compute with SVD

Modeling domain shift with geodesic flow

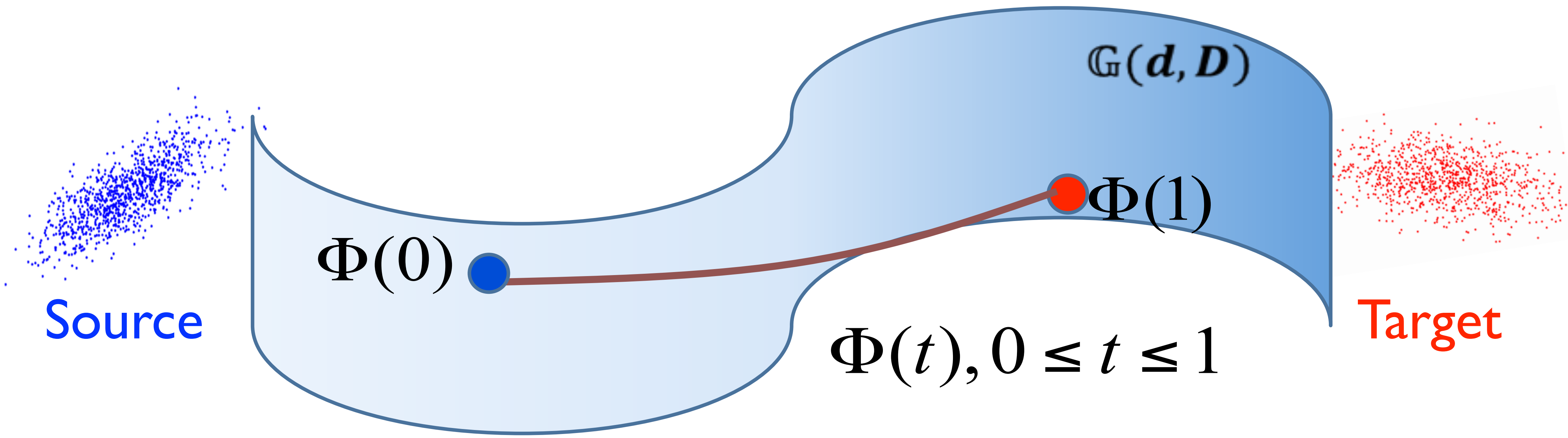


Geodesic flow on the manifold

- starting at source & arriving at target in unit time
- flow parameterized with one parameter ***t***
- closed-form, easy to compute with SVD

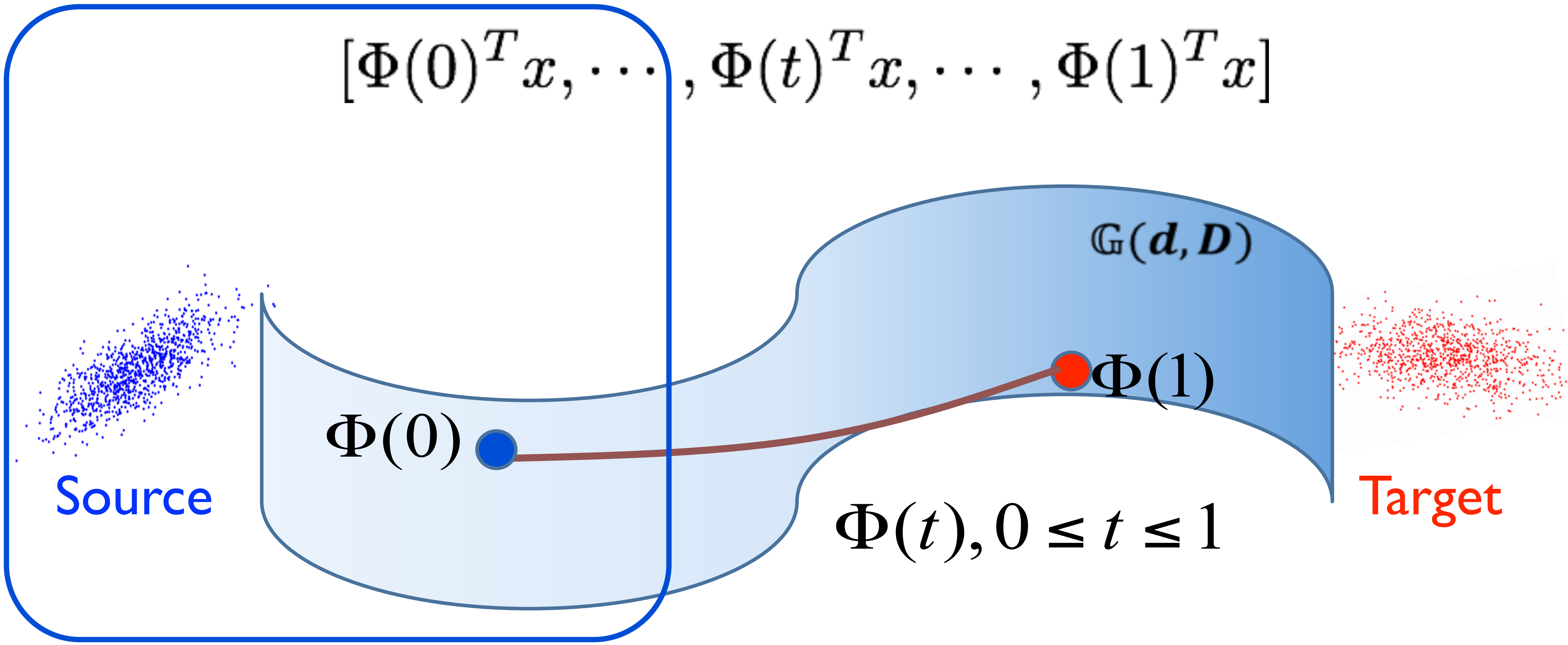
Domain-invariant features

$$z^{\infty} \quad = \quad [\Phi(0)^T x, \cdots, \Phi(t)^T x, \cdots, \Phi(1)^T x]$$



Domain-invariant features

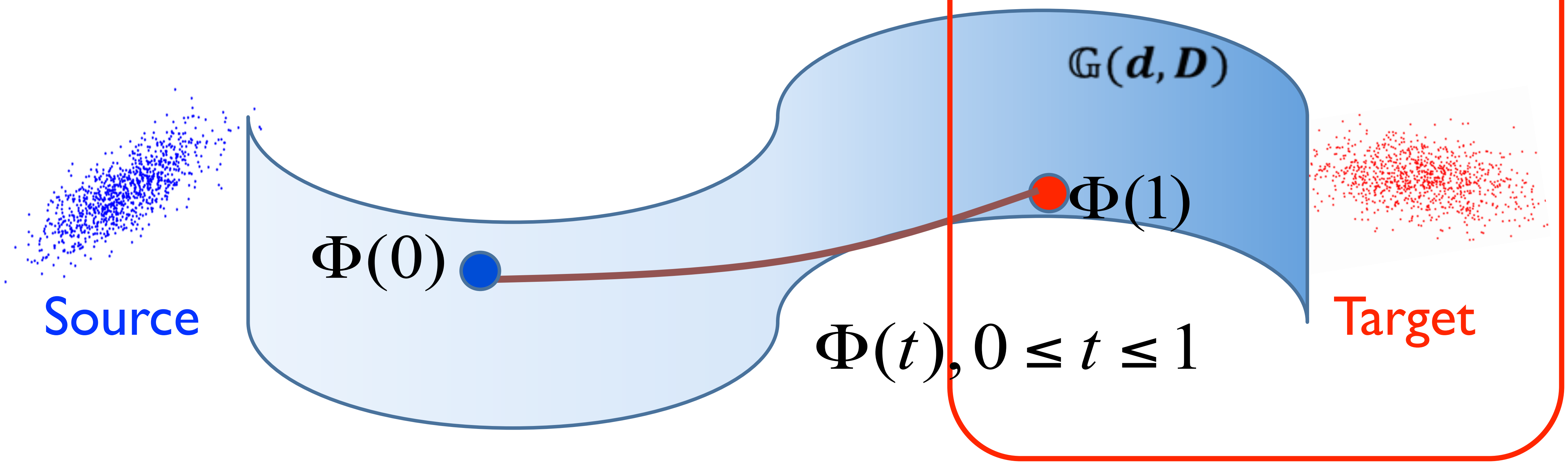
$$z^{\infty} = [\Phi(0)^T x, \cdots, \Phi(t)^T x, \cdots, \Phi(1)^T x]$$



More similar to source.

Domain-invariant features

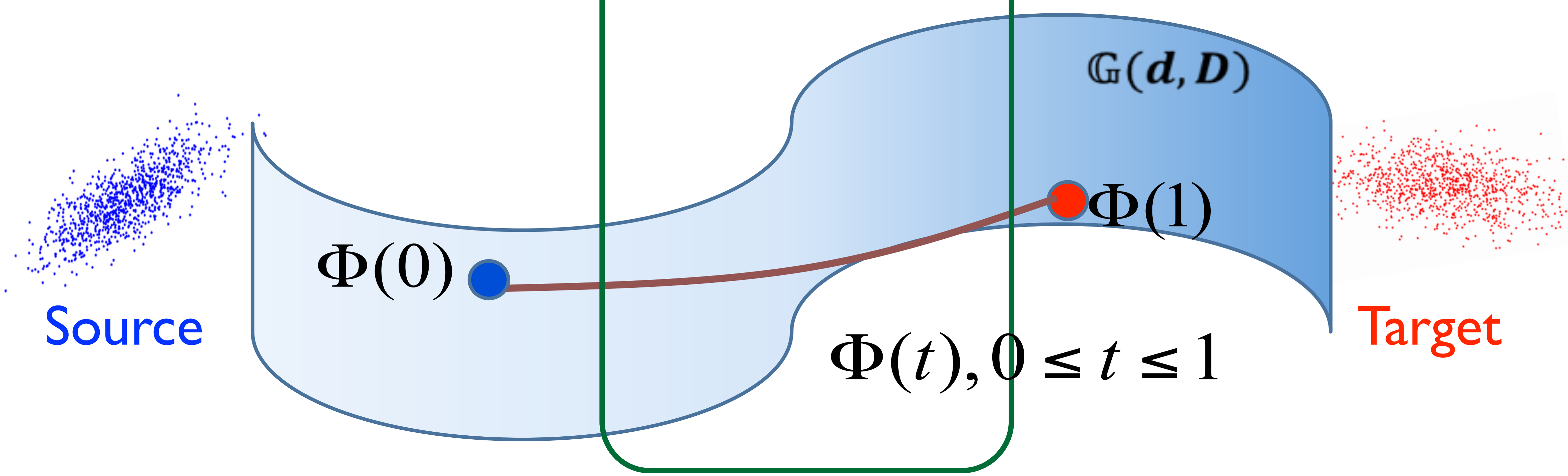
$$z^{\infty} = [\Phi(0)^T x, \cdots, \Phi(t)^T x, \cdots, \Phi(1)^T x]$$



More similar to target.

Domain-invariant features

$$z^{\infty} \quad = \quad [\Phi(0)^T x, \cdots, \Phi(t)^T x, \cdots, \Phi(1)^T x]$$



Blend the two.

The kernel trick

$$z^{\infty} \rightarrow \langle z_i^{\infty}, z_j^{\infty} \rangle$$

Avoiding the explicit mapping

$$z^{\infty} = [\Phi(0)^T x, \cdots, \Phi(t)^T x, \cdots, \Phi(1)^T x]$$

Domain-invariant kernel

We define the **geodesic flow kernel (GFK)**:

$$\langle z_i^\infty, z_j^\infty \rangle = \int_0^1 \left(\Phi(t)^T x_i\right)^T \left(\Phi(t)^T x_j\right) \mathrm{d}t = x_i^T \mathbf{G} x_j$$

Advantages

- Analytically computable, clean formulation
- Only one hyperparameter, automatically determined
- Broadly applicable: can kernelize many classifiers

[Gong *et al.*, CVPR'12]

Experimental study

Four vision datasets/domains on visual **object recognition**

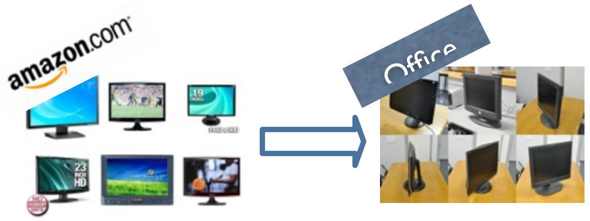
[Griffin et al. '07, Saenko et al. 10']

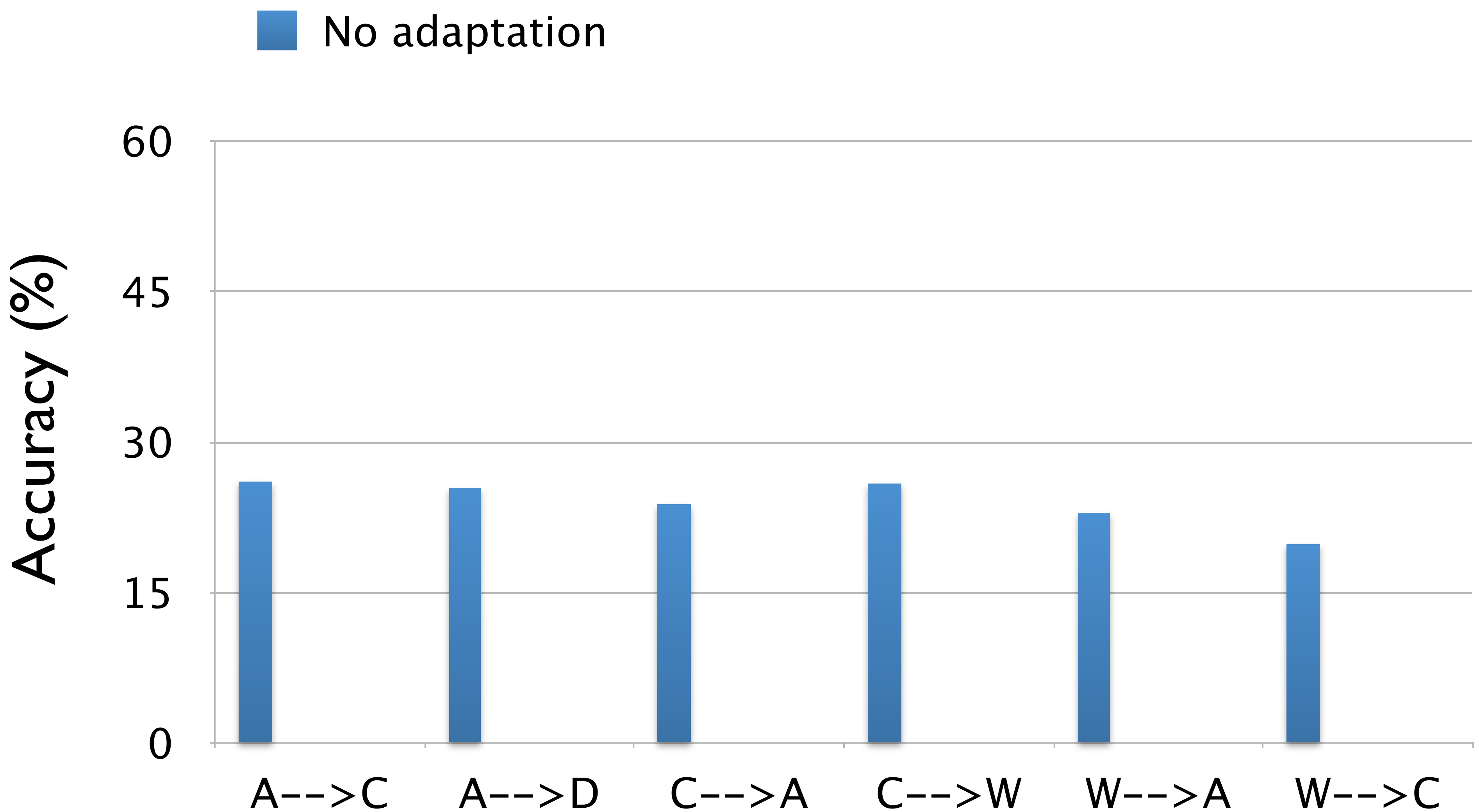
10 common classes

10~100 images per class

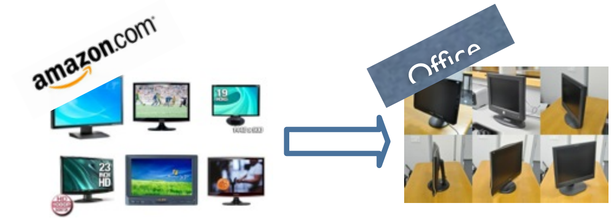
Bag-of-words and SURF features

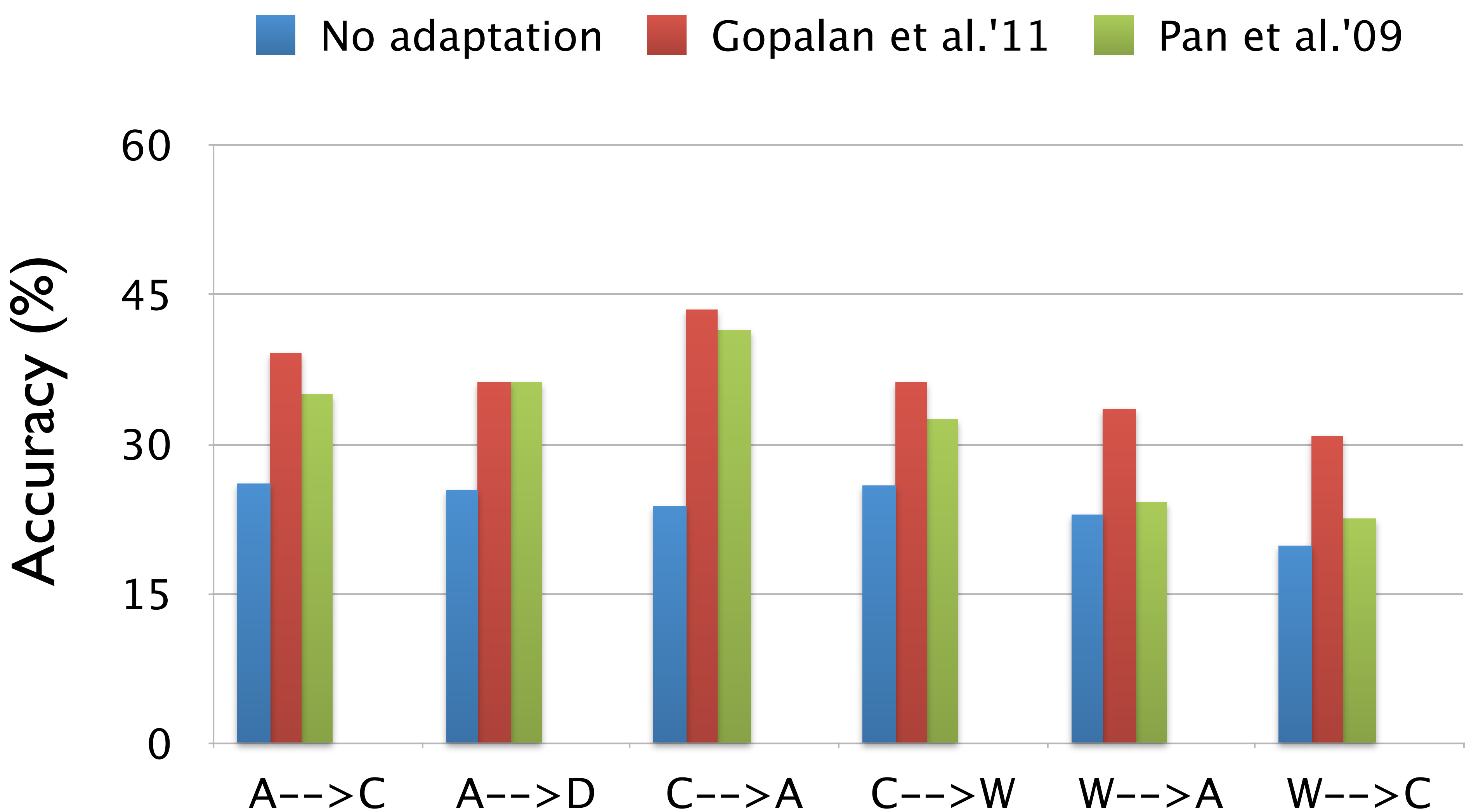
Comparison results

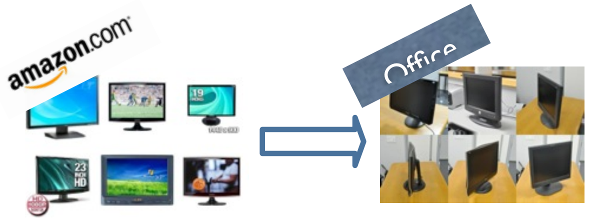




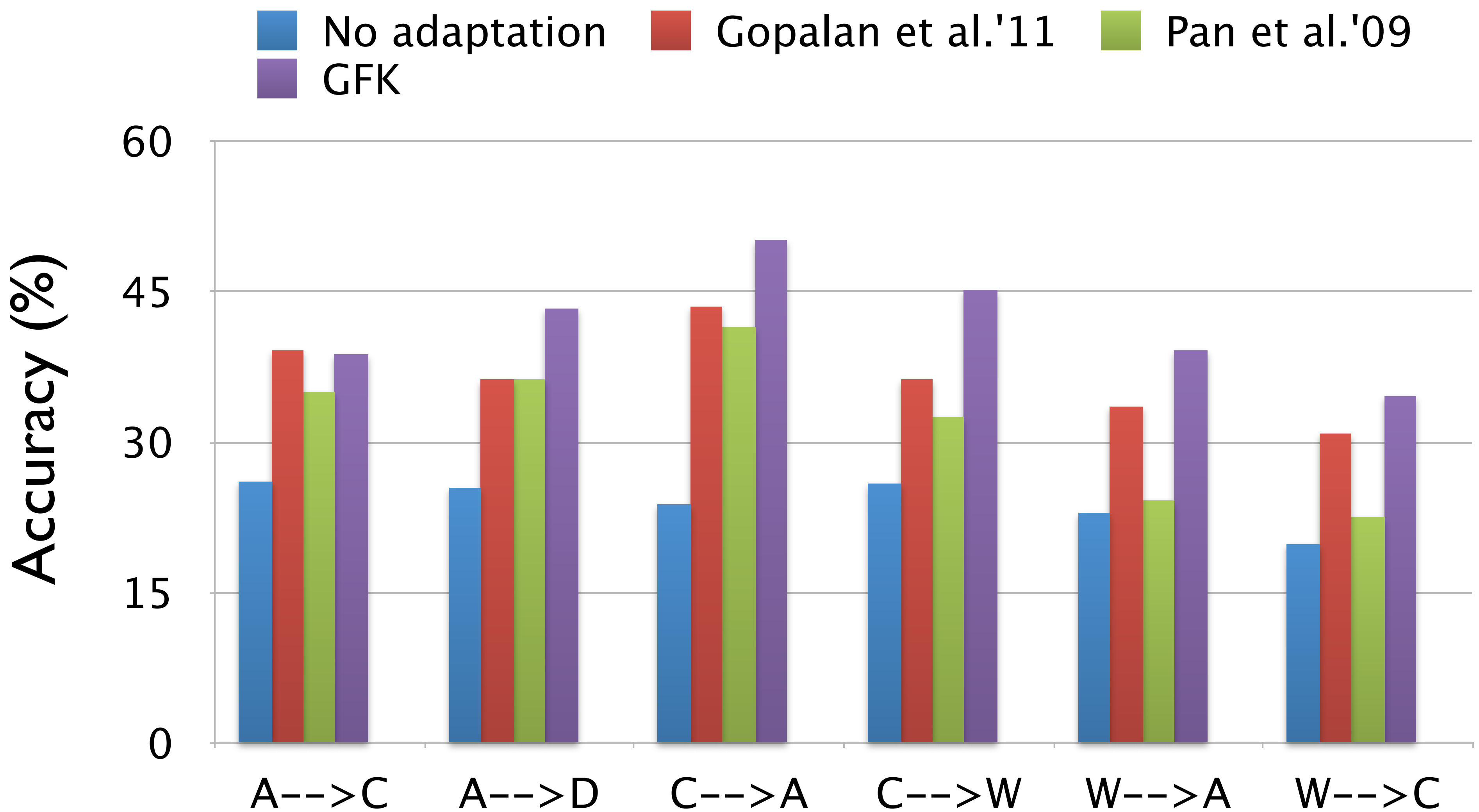
Comparison results





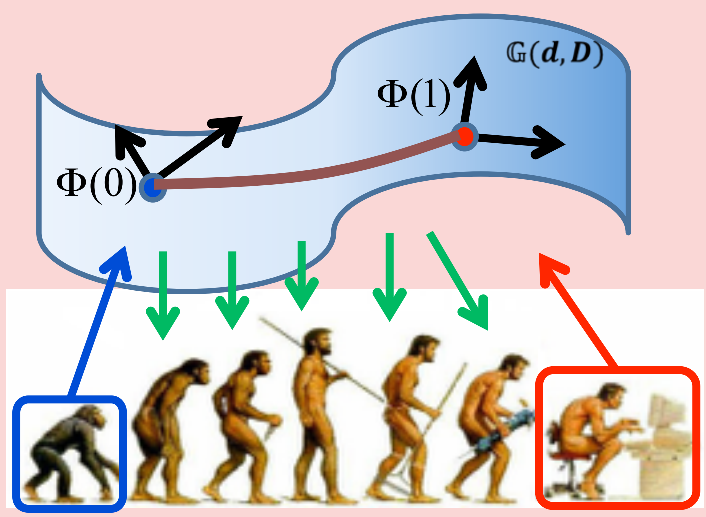


Comparison results



Kernel methods for DA

Inferring domain-invariant features

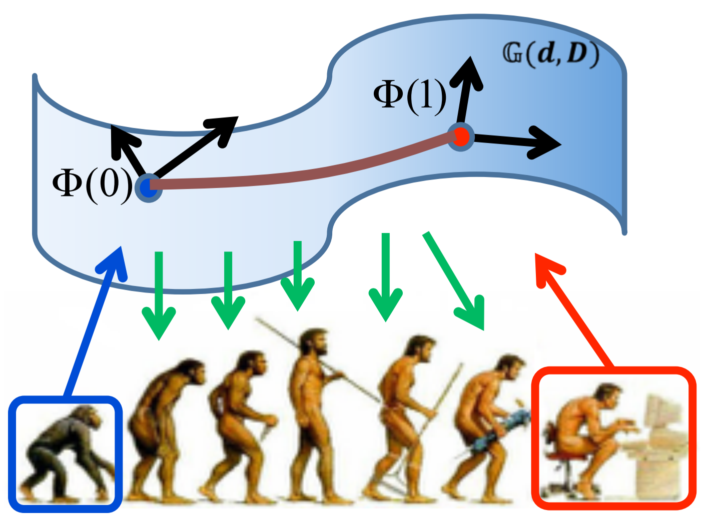


Geodesic flow kernel

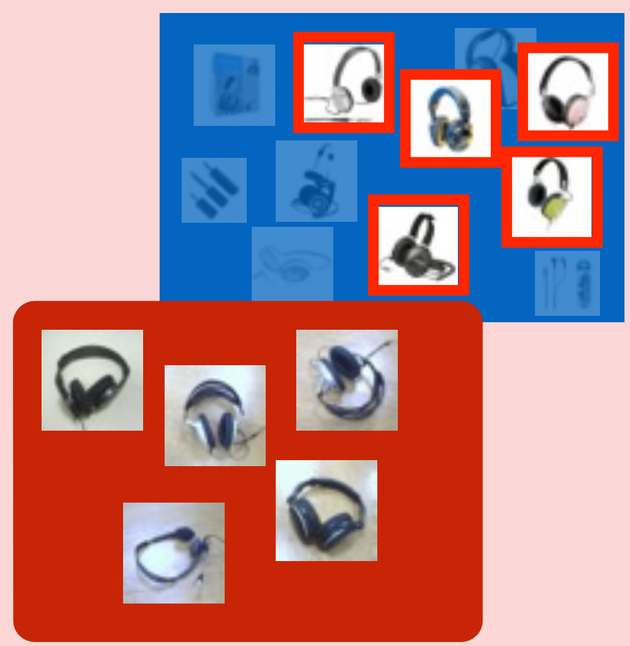
$$\langle z_i^{\infty}, z_j^{\infty} \rangle = \int_0^1 \left(\Phi(t)^T x_i\right)^T \left(\Phi(t)^T x_j\right) \mathrm{d}t = x_i^T \mathbf{G} x_j$$

Kernel methods for DA

Directly matching distributions



Geodesic flow kernel



Landmarks

Latent domains

A case study: building vision systems for Amazon images

Image credits: Nadine DeNinno (http://www.ibtimes.com)

Not all source instances equally adaptable/useful

Pose: Canonical

Lighting: *Uniform*

background: *clean*

Selecting most adaptable source instances

Landmarks are labeled source instances distributed similarly to the target domain.

Source

Target

[Gong *et al.*, ICML'13]

Selecting most adaptable source instances

Landmarks are labeled source instances distributed similarly to the target domain.

Source

Target

[Gong *et al.*, ICML'13]

Selecting most adaptable source instances

Landmarks are labeled source instances distributed similarly to the target domain.

Identifying landmarks:

$$P_{\mathcal{L}}(\text{landmarks}) \approx P_{\mathcal{T}}(\text{target})$$

$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

[Gong *et al.*, ICML'13]

Source

Target

Selecting most adaptable source instances

Landmarks are labeled source instances distributed similarly to the target domain.

Source

Identifying landmarks:

$$P_{\mathcal{L}}(\text{landmarks}) \approx P_{\mathcal{T}}(\text{target})$$

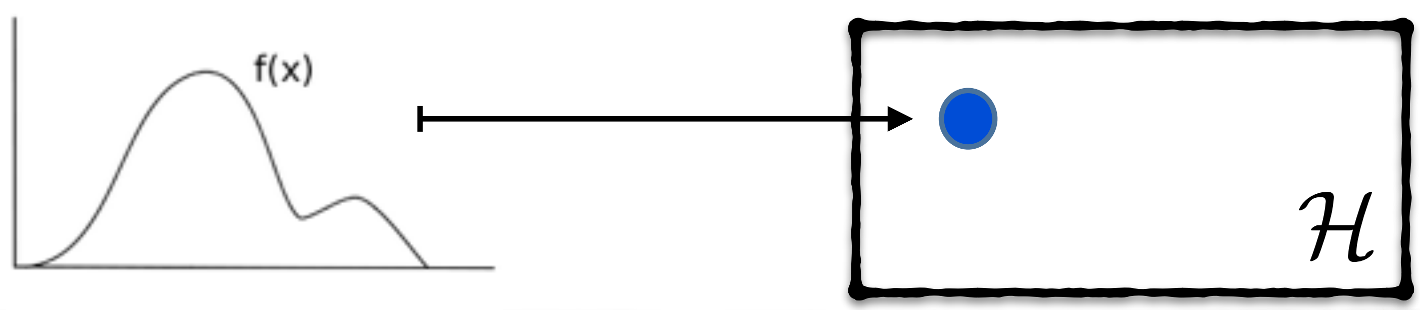
$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

Target

[Gong *et al.*, ICML'13]

Kernel embedding of distributions

$$\mu[P] \triangleq \mathbb{E}_x[\phi(x)]$$

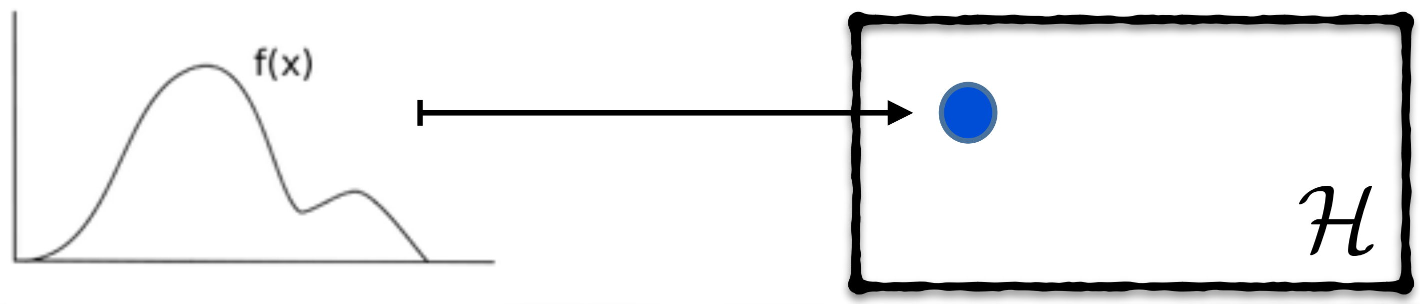


μ maps distributions to RKHS

RKHS associated with kernel $k(,)$, and $\phi(x)=k(x,\cdot)$

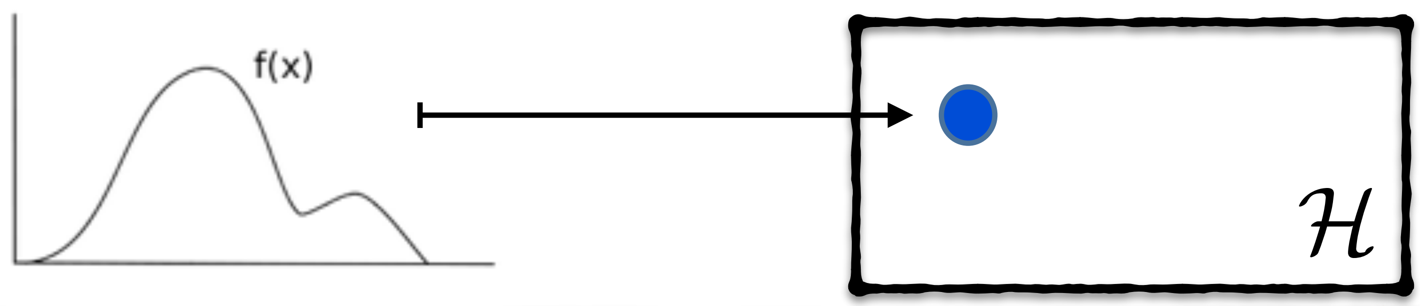
Kernel embedding of distributions

$$\mu[P] \triangleq \mathbb{E}_x[\phi(x)]$$



Kernel embedding of distributions

$$\mu[P] \triangleq \mathbb{E}_x[\phi(x)]$$

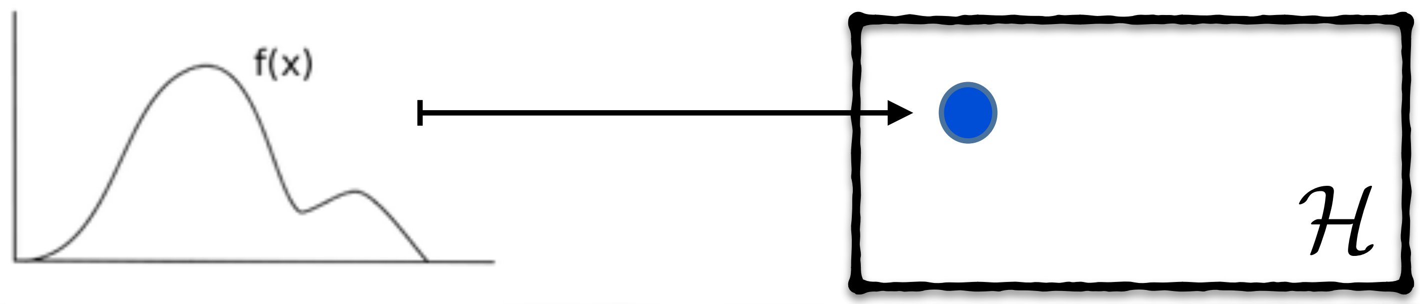


The mapping μ is injective, if $k(,)$ is characteristic

$\mu[P]$ preserves all statistical features of $P(x)$

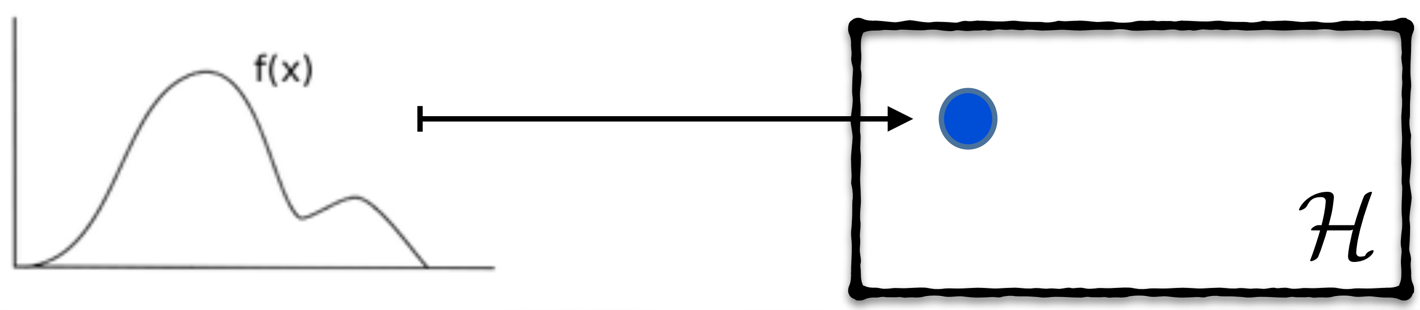
Kernel embedding of distributions

$$\mu[P] \triangleq \mathbb{E}_x[\phi(x)]$$



Kernel embedding of distributions

$$\mu[P] \triangleq \mathbb{E}_x[\phi(x)]$$

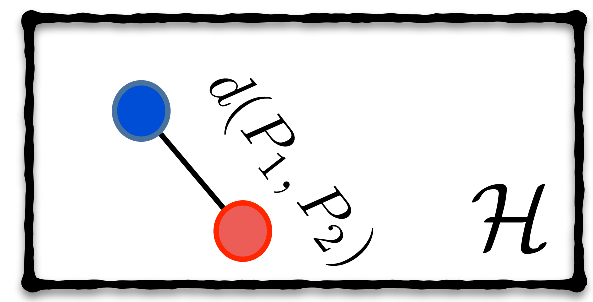


Empirical kernel embedding:

$$\hat{\mu}[P] = \frac{1}{n}\sum_{i=1}^{n}\phi(x_i), \quad x_i \sim P$$

A distance of distributions

$$
\begin{aligned}
& d(P_1, P_2) \\
\triangleq & \sup_{\|f\|_{\mathcal{H}} \leq 1} \left(\mathbb{E}_{x \sim P_1} f(x) - \mathbb{E}_{x \sim P_2} f(x) \right) \\
= & \cdots \\
= & \|\mu[P_1] - \mu[P_2]\|_{\mathcal{H}}
\end{aligned}
$$



d(,) is a metric of distributions, if *k(,)* is characteristic

Maximum mean discrepancy (MMD): the sup() operation

[Müller’97, Gretton et al.’07, Sriperumbudur et al.’10]

Identifying landmarks by empirical MMD

Integer programming

$$\min_{\{\alpha_m\}} \left\| \frac{1}{\sum_i \alpha_i} \sum_{m=1}^{\mathsf{M}} \alpha_m \phi(x_m) - \frac{1}{\mathsf{N}} \sum_{n=1}^{\mathsf{N}} \phi(x_n) \right\|_{\mathcal{H}}^2$$

where

$$\alpha_m = \begin{cases} 1 & \text{if } x_m \text{ is a landmark wrt target} \\ 0 & \text{else} \end{cases}$$

$$m = 1, 2, \cdots, \mathsf{M}$$

[Gong *et al.*, ICML'13]

Identifying landmarks by empirical MMD

Convex relaxation

$$\min_{\{\alpha_m\}} \quad \left\| \frac{1}{\sum_i \alpha_i} \sum_{m=1}^{\mathsf{M}} \alpha_m \phi(x_m) - \frac{1}{\mathsf{N}} \sum_{n=1}^{\mathsf{N}} \phi(x_n) \right\|_{\mathcal{H}}^2$$

Identifying landmarks by empirical MMD

Convex relaxation

$$\min_{\{\alpha_m\}} \quad \left\| \frac{1}{\sum_i \alpha_i} \sum_{m=1}^{\mathsf{M}} \alpha_m \phi(x_m) - \frac{1}{\mathsf{N}} \sum_{n=1}^{\mathsf{N}} \phi(x_n) \right\|_{\mathcal{H}}^2$$

$$\beta_m = \frac{\alpha_m}{\sum_i \alpha_i}$$

Identifying landmarks by empirical MMD

Convex relaxation

$$\min_{\{\alpha_m\}} \quad \left\| \frac{1}{\sum_i \alpha_i} \sum_{m=1}^{\mathsf{M}} \alpha_m \phi(x_m) - \frac{1}{\mathsf{N}} \sum_{n=1}^{\mathsf{N}} \phi(x_n) \right\|_{\mathcal{H}}^2$$

$$\beta_m = \frac{\alpha_m}{\sum_i \alpha_i}$$

$$\min_{\beta} \quad \beta^T K^s \beta - \frac{2}{\mathsf{N}} \beta^T K^{st} \mathbf{1}$$

Experimental study (cont'd)

Four vision datasets/domains on visual **object recognition**

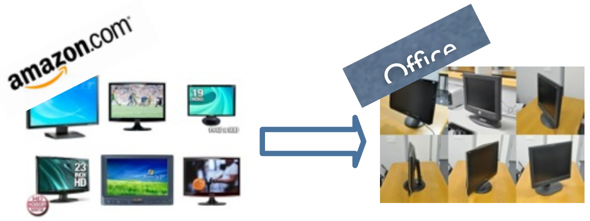
[Griffin et al. '07, Saenko et al. 10']

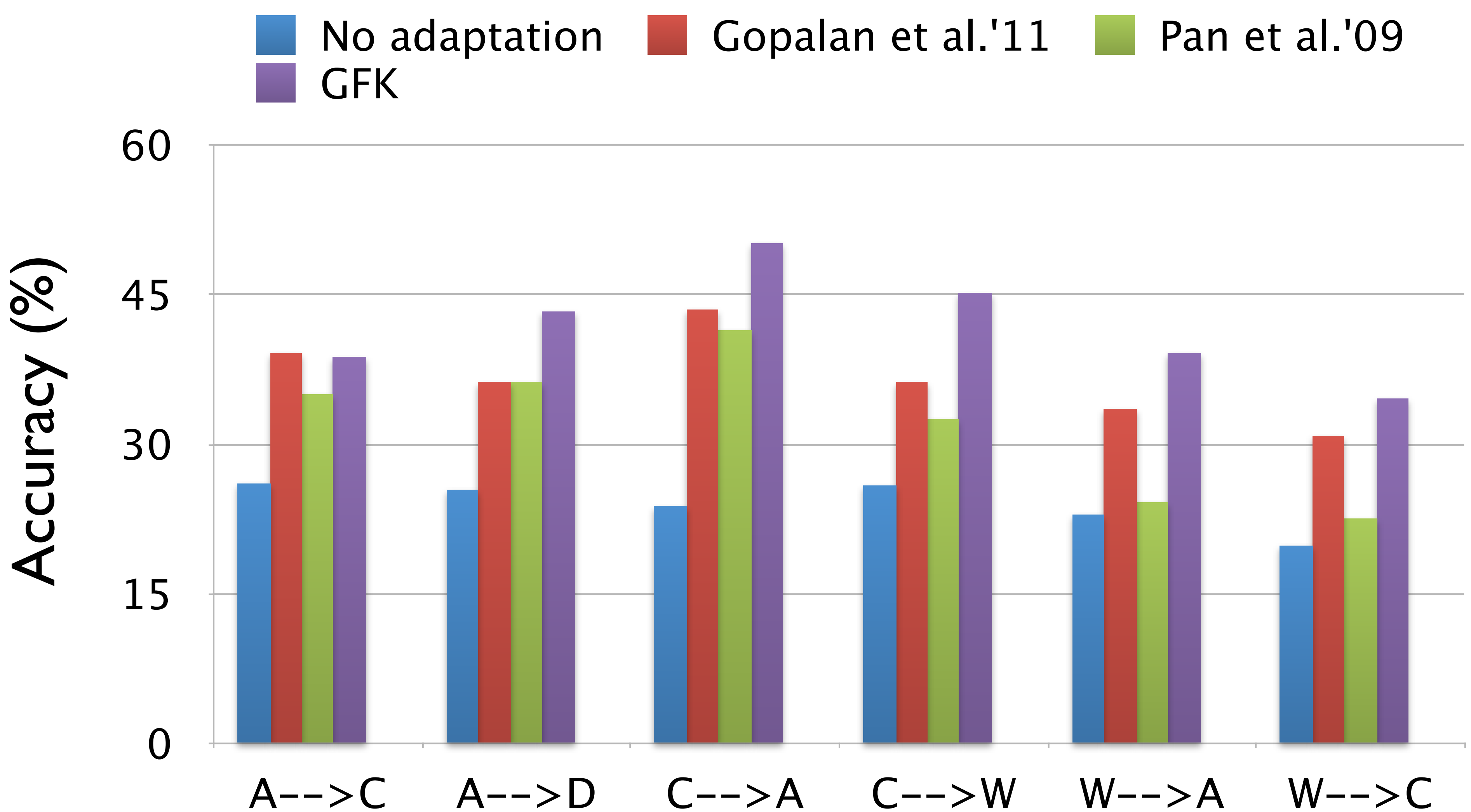
10 common classes

10~100 images per class

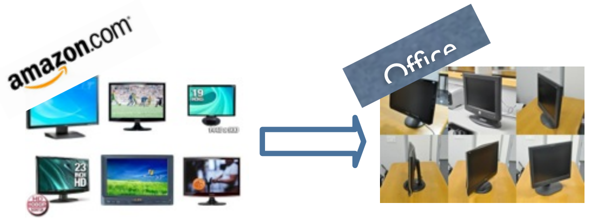
Bag-of-words and SURF features

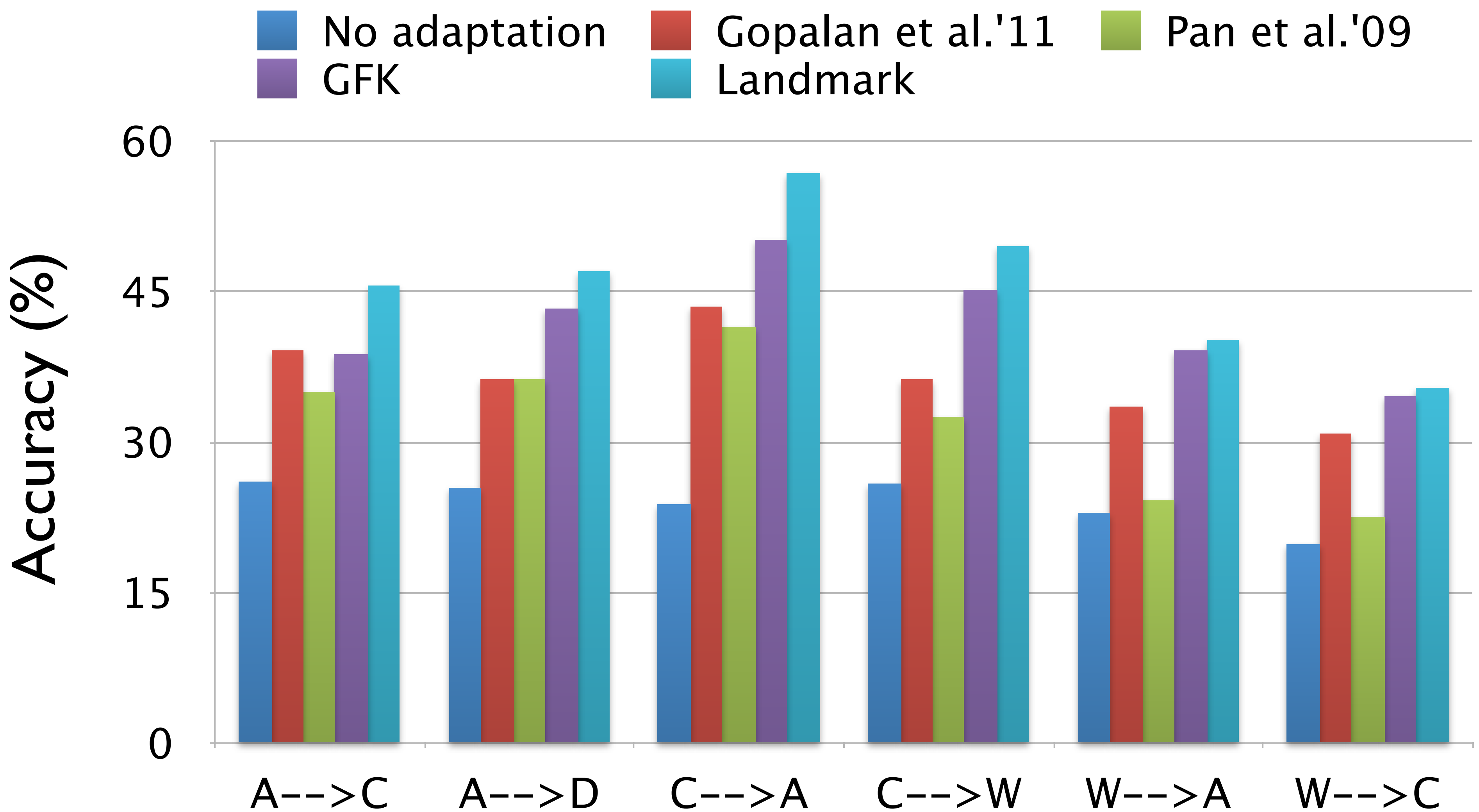
Comparison results: object recognition



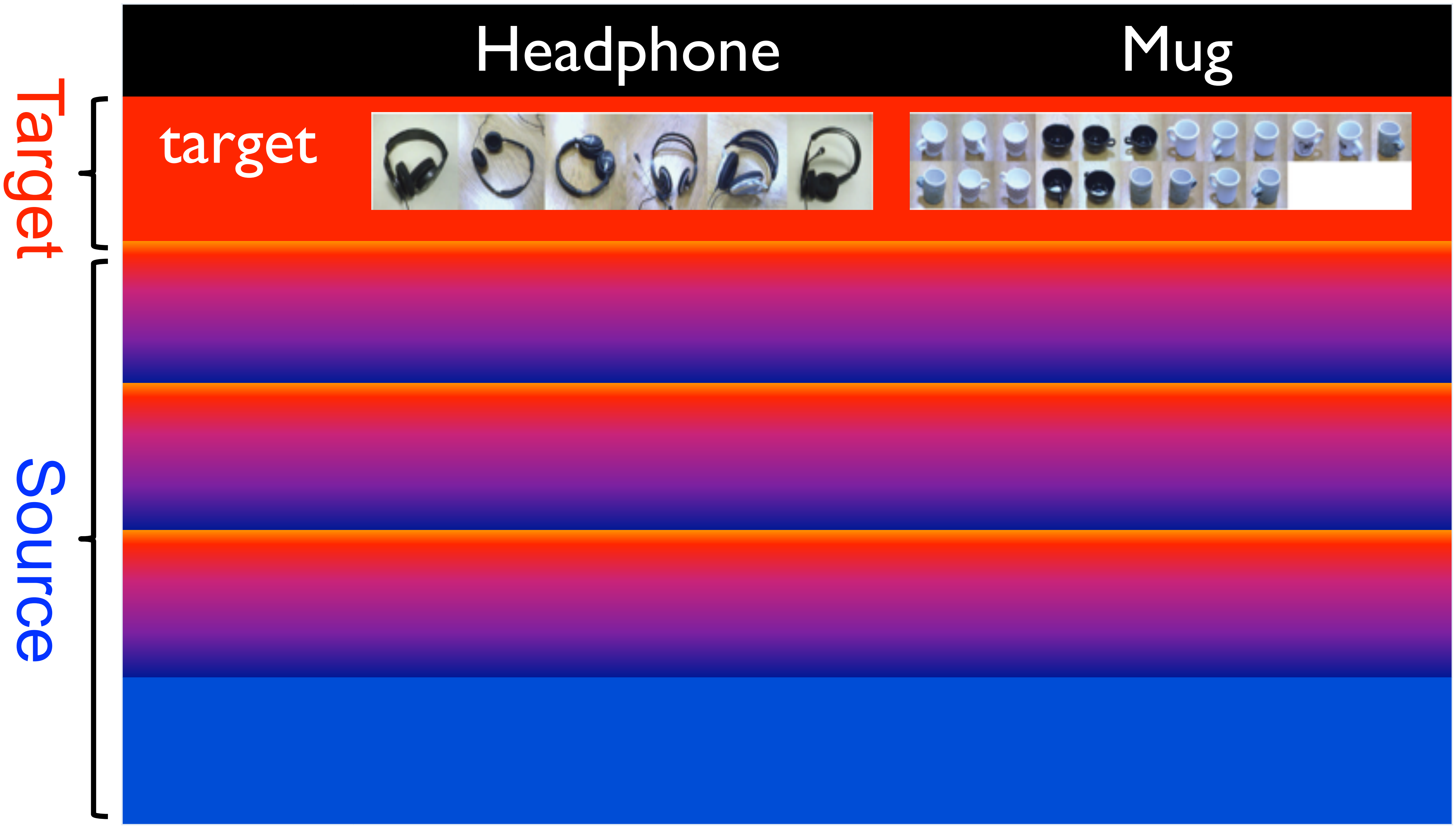


Comparison results: object recognition





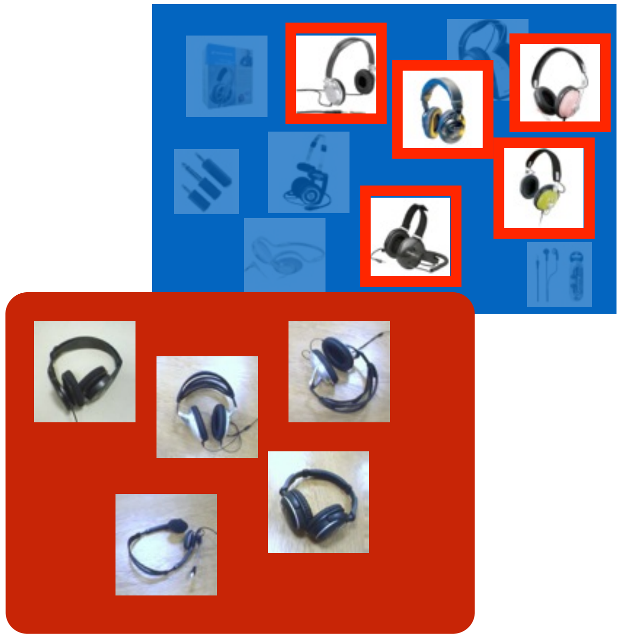
How do landmarks look like?



How do landmarks look like?

How do landmarks look like?

Landmarks



Labeled source instances

Distributed similarly to the target

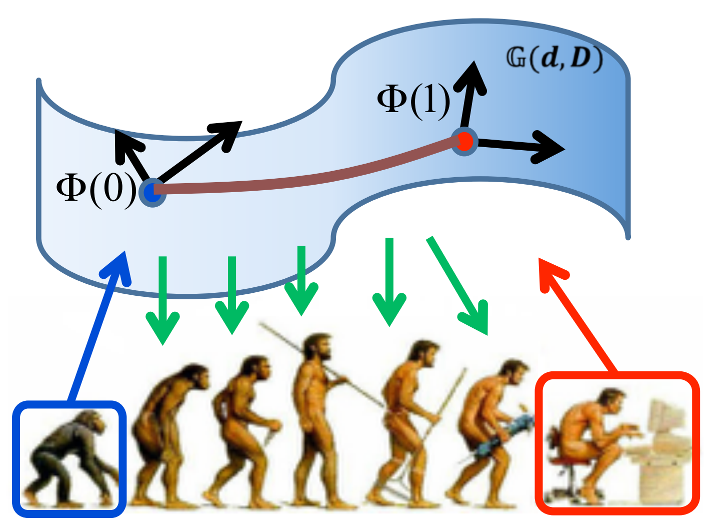
New intrinsic structure shared between domains

Proxy of discriminative loss of target

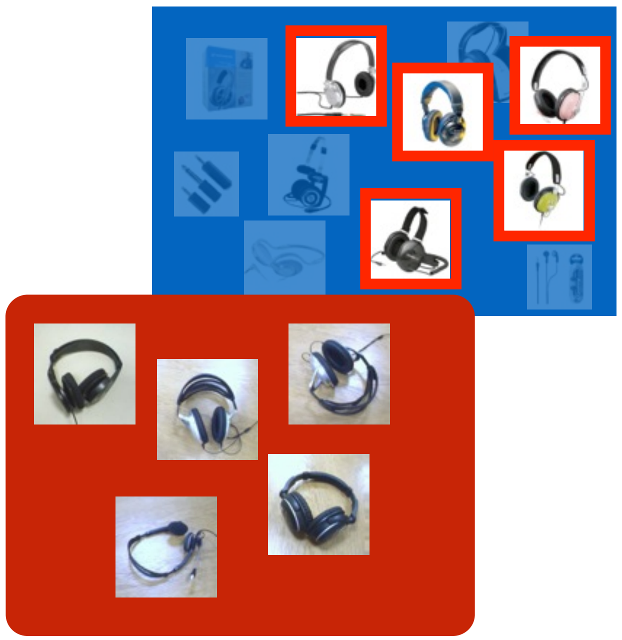
Outperformed the state-of-the-arts

Kernel methods for DA

Directly matching distributions



Geodesic flow kernel



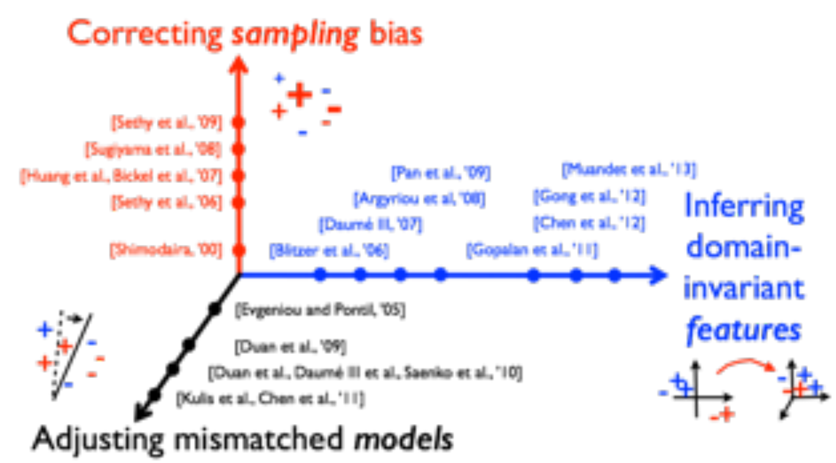
Landmarks

Latent domains

Domains = datasets?

Most DA methods

- Assume good-quality domains
- Evaluated as cross-dataset adaptation

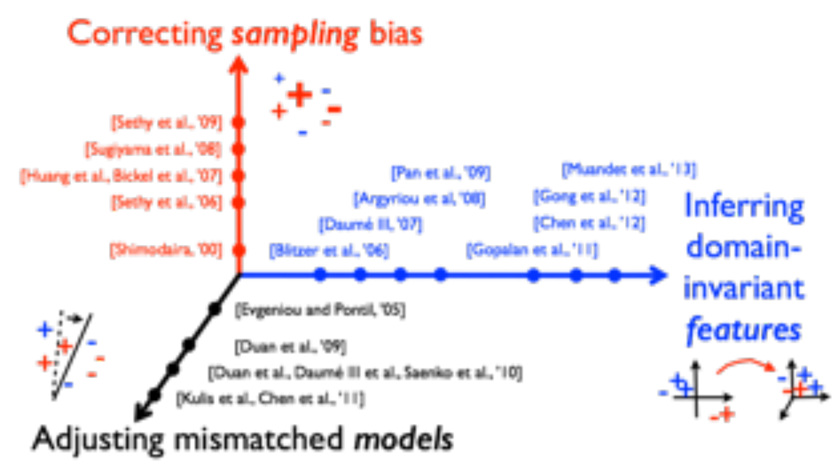


Domains = datasets?

Most DA methods

Assume good-quality domains

Evaluated as cross-dataset adaptation



Common **mistake**: equating datasets with domains

Suboptimal to use DA methods for cross-dataset generalization

What constitutes a domain?

In speech and NLP:

- Speakers
- Languages
- Article topics
- …other factors

What constitutes a domain?

In speech and NLP:

Speakers

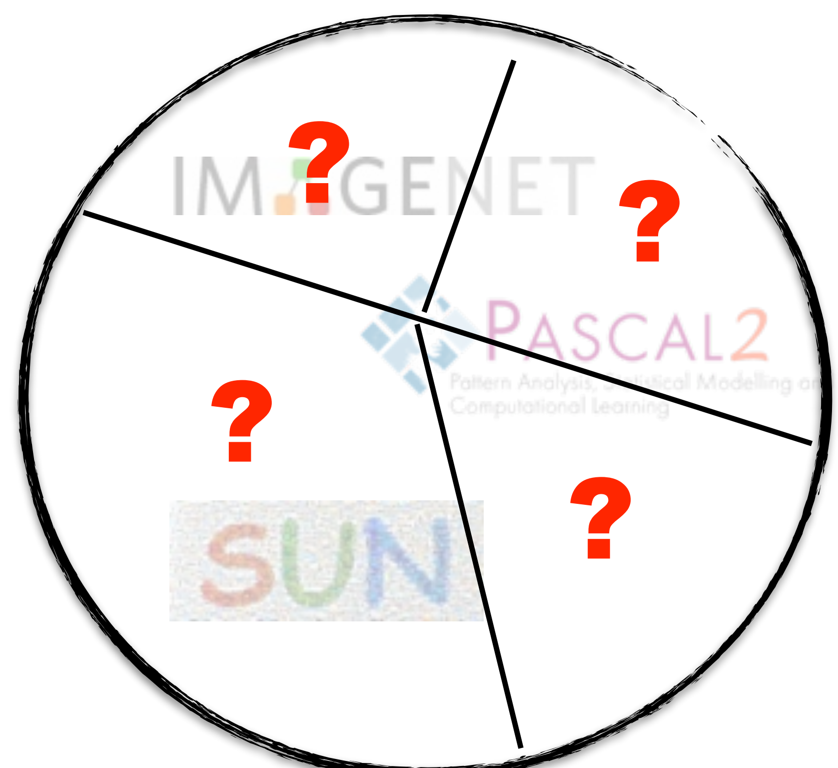
Languages

Article topics

…other factors

In computer vision:

Factors?



What constitutes a domain?

In speech and NLP:

Speakers

Languages

Article topics

…other factors

In computer vision:

Pose

What constitutes a domain?

In speech and NLP:

Speakers

Languages

Article topics

…other factors

In computer vision:

What constitutes a domain?

In speech and NLP:

- Speakers
- Languages
- Article topics
- …other factors

In computer vision:

Pose

Lighting

Fore/Background

What constitutes a domain?

In speech and NLP:

- Speakers
- Languages
- Article topics
- …other factors

In computer vision:

Pose

Lighting

Fore/ Background

Occlusion

What constitutes a domain?

In speech and NLP:

- Speakers
- Languages
- Article topics
- …other factors

In computer vision:

Many factors overlap & interact

What constitutes a domain?

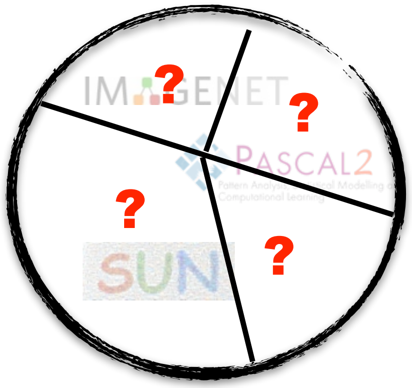
In speech and NLP:

Speakers

Languages

Article topics

…other factors



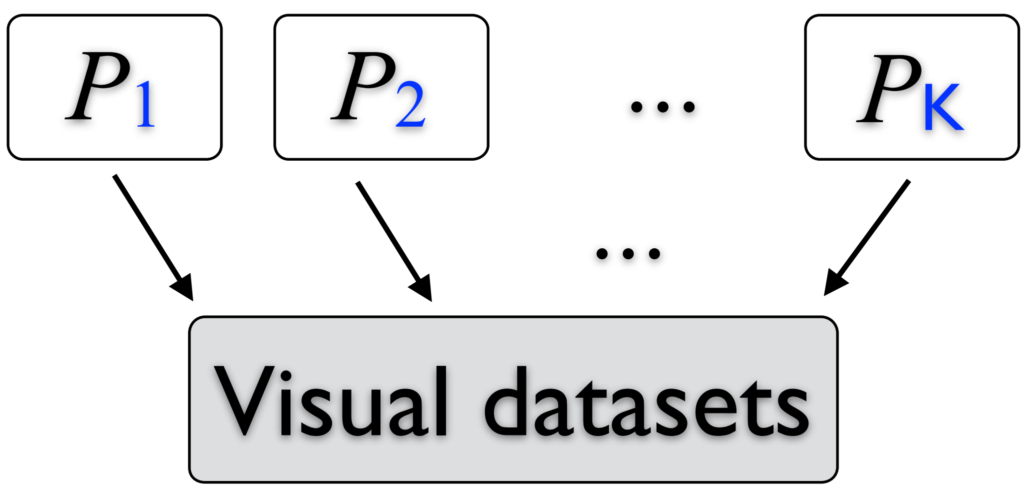
In computer vision:

Hard to manually enumerate/discretize visual factors

Our solution: working with distributions via **kernels**

Domains as distributions

Labeled data could be drawn from multiple domains/distributions



Reshaping data to domains before adaptation

Two axiomatic properties

I. Maximum **distinctiveness**:

Identifying distinct domains maximally different **in distribution** from each other

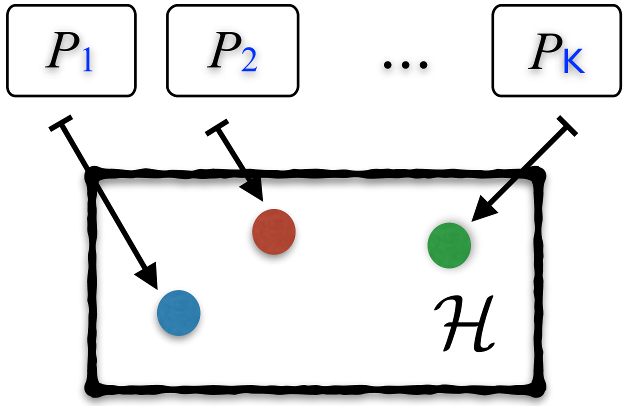
II. Maximum **learnability**

Being able to derive strong discriminative models from the identified domains

I. Maximum distinctiveness

Domains maximally different in distribution from each other

$$\max_{\{z_{mk}\}} \sum_{k \neq k'} \hat{d}(P_k, P_{k'}; \{z_{mk}\})$$



$$z_{mk} = \begin{cases} 1 & \text{if } x_m \in \text{the } k\text{-th domain} \\ 0 & \text{else} \end{cases}$$

$$m = 1, 2, \cdots, \mathrm{M}, \quad k = 1, 2, \cdots, \mathrm{K}$$

[Gong *et al.*, NIPS'13]

II. Maximum **learnability**

Being able to learn strong classifiers from domains

Within-domain cross-validation

$$\mathrm{Accuracy}(\mathsf{K}) = \sum_{k=1}^{\mathsf{K}} \frac{\mathsf{M}_k}{\mathsf{M}} \mathrm{Accuracy}_k$$

- Determining the number of domains K

[Gong *et al.*, NIPS'13]

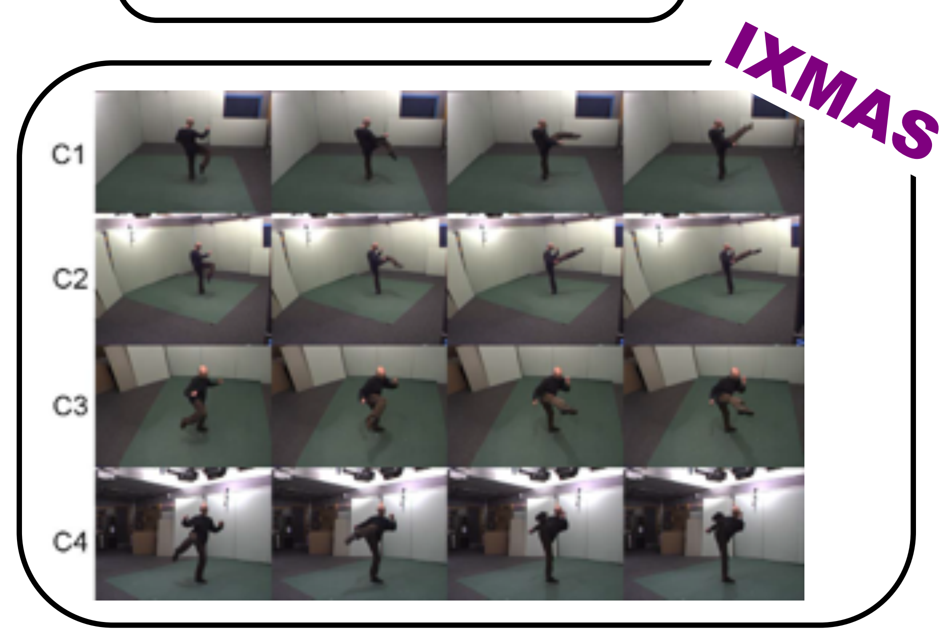
Experimental study

Four vision datasets/domains on visual **object recognition**

[Griffin et al. ’07, Saenko et al. 10’]

Five views/domains on cross-view human **action recognition**

[Weinland et al.’07]



Comparison results

Sources/datasets	A, C	D, W	C, D, W	View 0,1	View 2,3,4
Targets	D, W	A, C	A	View 2,3,4	View 0,1
From	41.0	32.6	41.8	44.6	47.1

Comparison results

Sources/datasets	A, C	D, W	C, D, W	View 0,1	View 2,3,4
Targets	D, W	A, C	A	View 2,3,4	View 0,1
From	41.0	32.6	41.8	44.6	47.1
From domains	42.6	35.5	44.6	47.3	50.3

Comparison results

Sources/datasets	A, C	D, W	C, D, W	View 0,1	View 2,3,4
Targets	D, W	A, C	A	View 2,3,4	View 0,1
From	41.0	32.6	41.8	44.6	47.1
From domains	42.6	35.5	44.6	47.3	50.3

Cross-**domain** adaptation

> cross-**dataset** adaptation

Reshaping test set

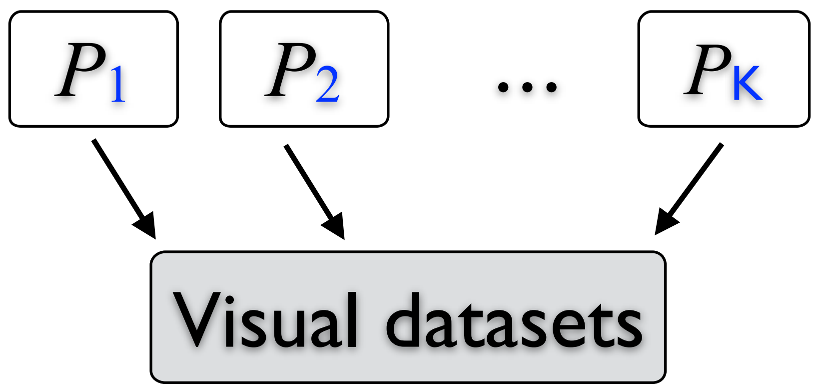
Assigning test data to discovered domains

$$\min_{\{t_{nk}\}} \quad \sum_k \hat{d}\left(\textsc{domain}_k, \textsc{target}; \{t_{nk}\}\right)$$

$$t_{nk} = \mathbb{I}(x_n \text{ is assigned to the } k\text{-th domain})$$

[Gong *et al.*, NIPS'13]

Latent domains



Dataset ≠ domain

Suboptimal to adapt across datasets using DA methods

Cross-**domain** adaptation > cross-**dataset** adaptation

Identifying latent domains

Maximum **distinctiveness** & maximum **learnability**

A non-parametric kernel method

Summary (1/2)

(unsupervised) Domain adaptation is **ill-posed**

Potentially successful solutions come with

- Appropriate assumptions
- Well-modeled domain knowledge

Dataset$\neq$domain, cross-dataset generalization by

- Identifying landmarks from dataset
- Reshaping data to obtain good-quality domains

Summary (2/2)

Kernel methods for domain adaptation:

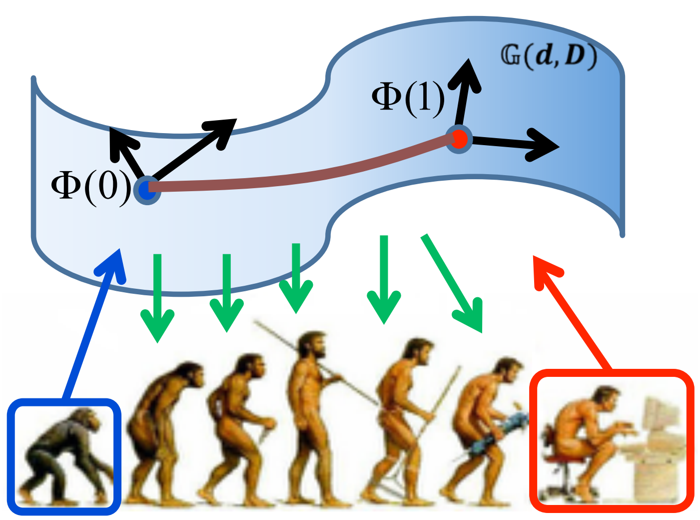
Inferring domain invariant features

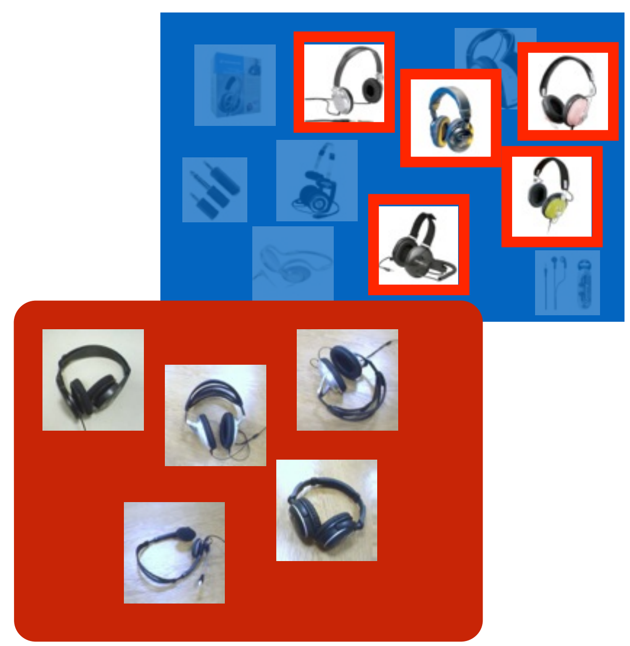
- Geodesic flow kernel (GFK)
- *Kernel trick*

Directly matching distributions

- Landmarks and Latent domains
- *Kernel embedding of distributions*

Code available:
http://www-scf.usc.edu/~boqinggo





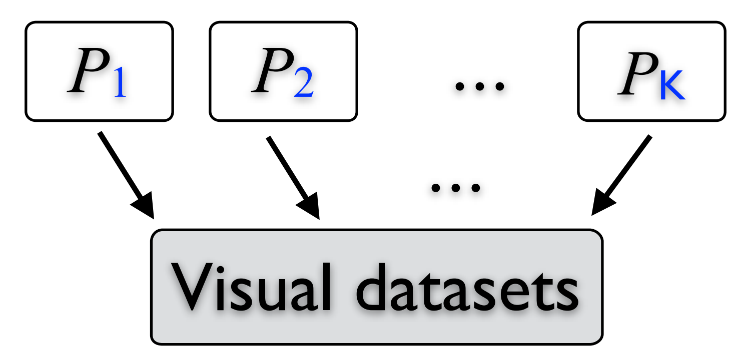
Geodesic flow kernel

$$\langle z_i^\infty, z_j^\infty \rangle = \int_0^1 \left(\Phi(t)^T x_i\right)^T \left(\Phi(t)^T x_i\right) \mathrm{d}t = x_i^T G x_j$$

Landmarks

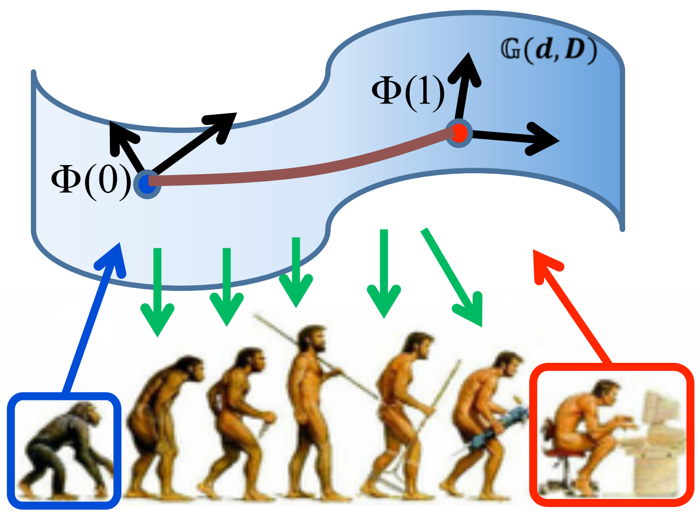
$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

Latent domains



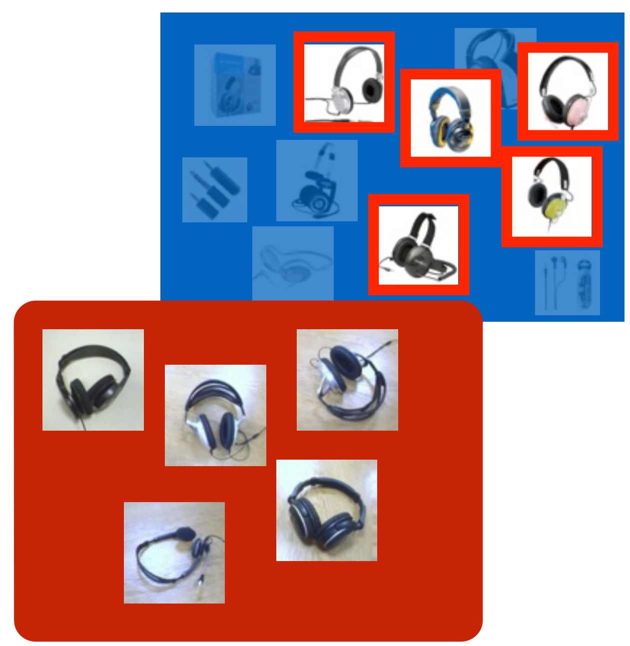
Thanks!

Code available:
http://www-scf.usc.edu/~boqinggo



Geodesic flow kernel

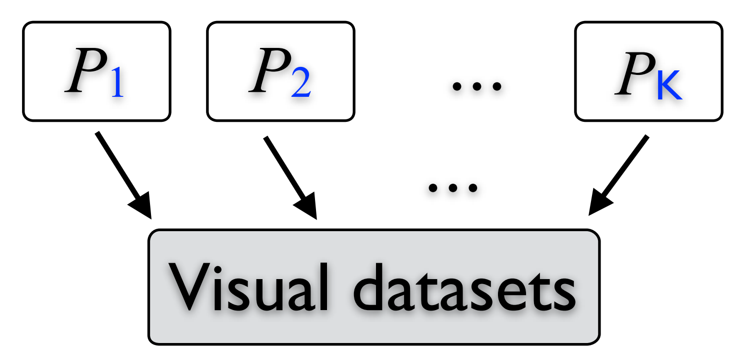
$$\begin{aligned}&\langle z_i^\infty, z_j^\infty\rangle \\ &= \int_0^1 \left(\Phi(t)^T x_i\right)^T \left(\Phi(t)^T x_i\right) \mathrm{d}t \\ &= x_i^T G x_j\end{aligned}$$



Landmarks

$$\min_{\text{landmarks}} \quad d(P_{\mathcal{L}}, P_{\mathcal{T}})$$

Latent domains



Latent domains

Evaluation strategy

Domain adaptation from **discovered domains**

vs. from original source domains/datasets

to all possible target domains

Evaluation metric: expected/averaged accuracy

$$\mathbb{E}_{\mathcal{T}}\left[\text{domains}|\text{datasets}\right] \rightarrow \mathcal{T}$$

Hard to manually define discrete domains

amazon.com

CALTECH 256

Amazon images from [Saenko et al.'10].

Our reshaped domains

Domain I **Domain II**

Results: reshaping test data

S	S	Best domain	Reshaping
View 012	37.3	37.7	38.5
View 123	39.9	40.4	41.1
View 234	47.8	46.5	49.2
View 340	52.3	50.7	54.9
View 401	43.3	43.9	44.8

Reshaping both training and test data gives rise to the best performance.