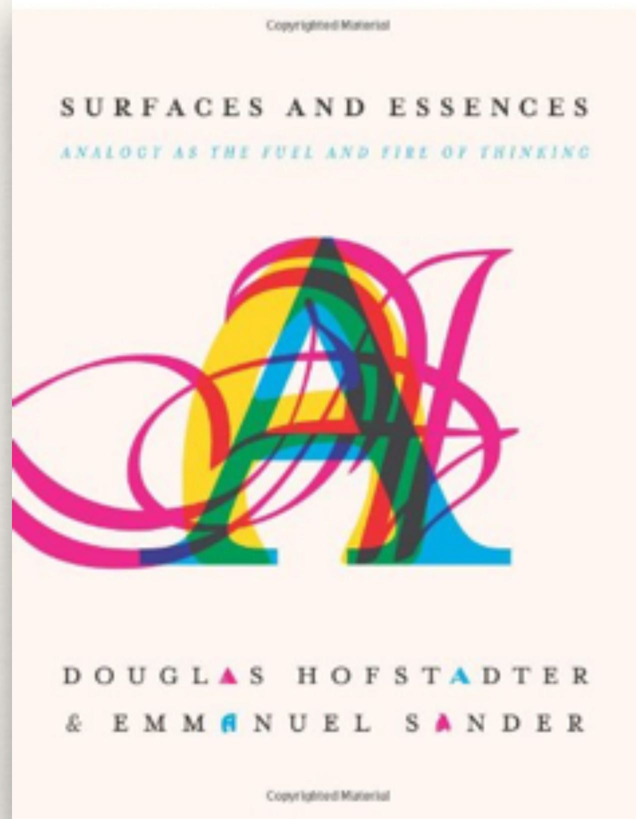
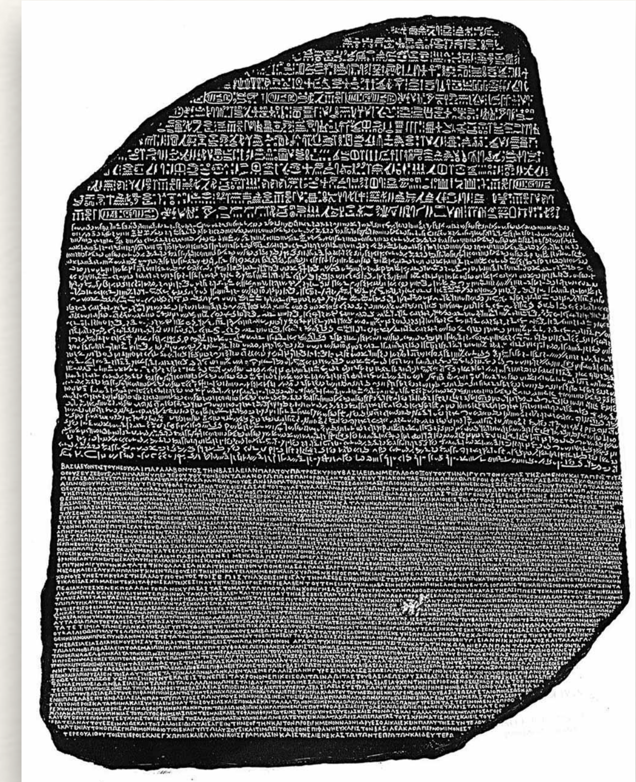


IJCAI 2016 Tutorial
July 11, New York City



Towards a Unified Framework for Transfer Learning: Exploiting Correlations and Symmetries

Sridhar Mahadevan
Autonomous Learning Lab
UMass Amherst

UMASS CS 50 YEARS

College of Information and Computer Sciences

Outline of the Tutorial

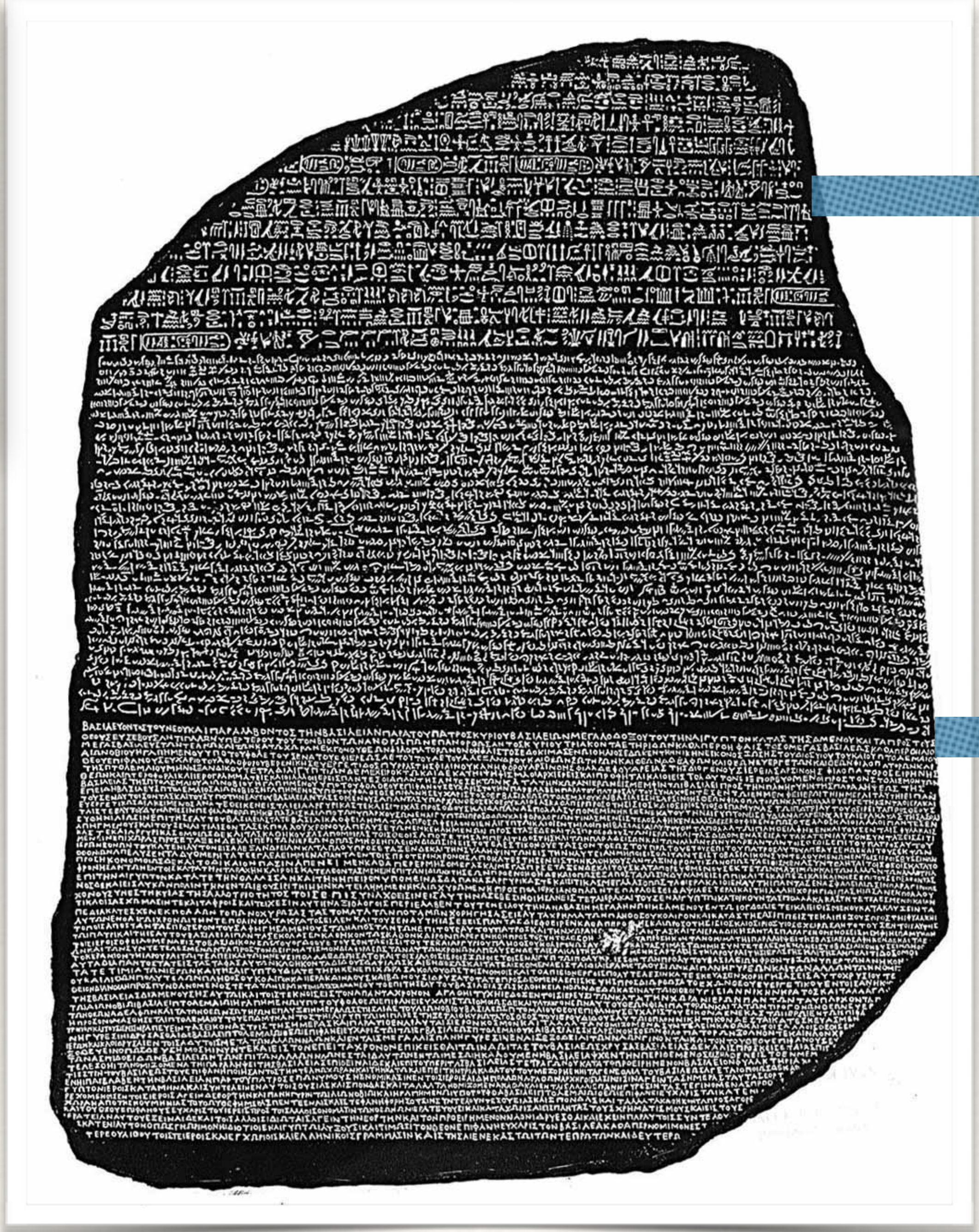
- **Historical review and motivation (20 minutes)**
- Mathematical background (20 minutes)
- Algorithms (30 minutes)
- Applications (30 minutes)
- Questions (5 minutes)

Motivation

- Machine learning assumes the test data is drawn from the same distribution as the training data
- Transfer learning is the class of problems where this assumption is violated (also called **domain adaptation**)
- In many real world problems, there is a lack of adequate labeled datasets, as labeling requires human effort
- In cognitive science, analogies and metaphors have been long studied as a major component of human thought

The Rosetta Stone

Champollion



Undeciphered language (hieroglyphics)

Known language (Coptic, Greek)

Cross-Language IR

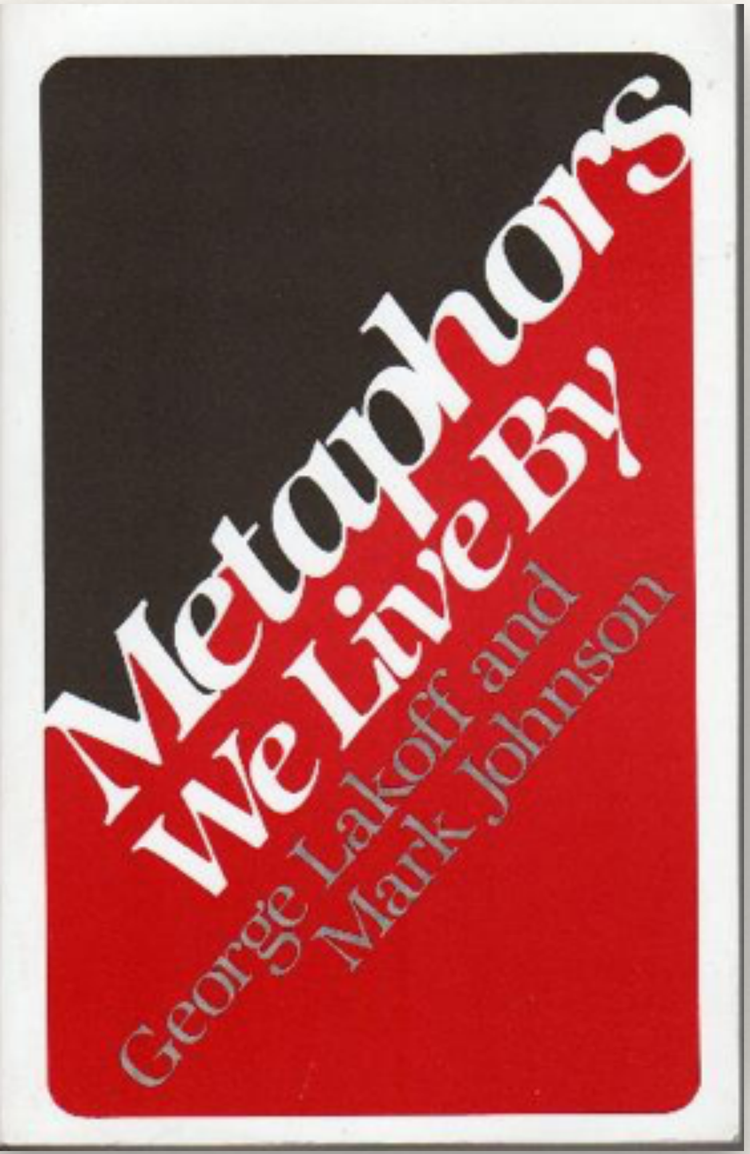
Madam President, on a point of order. You will be aware from the press and television that there have been a number of bomb explosions and killings in Sri Lanka.

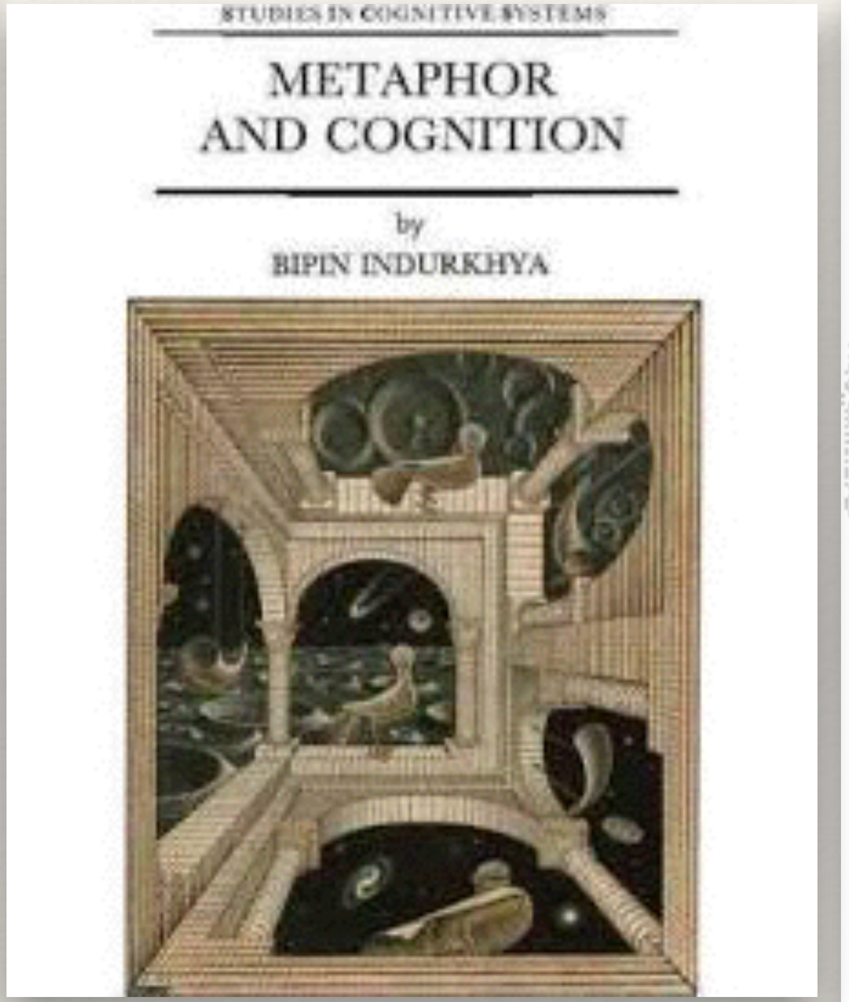
Signora Presidente, intervengo per una mozione d'ordine. Come avrà letto sui giornali o sentito alla televisione, in Sri Lanka si sono verificati numerosi assassinii ed esplosioni di ordigni.

Frau Präsidentin, zur Geschäftsordnung.
Wie Sie sicher aus der Presse und dem Fernsehen wissen, gab es in Sri Lanka mehrere Bombenexplosionen mit zahlreichen Toten.

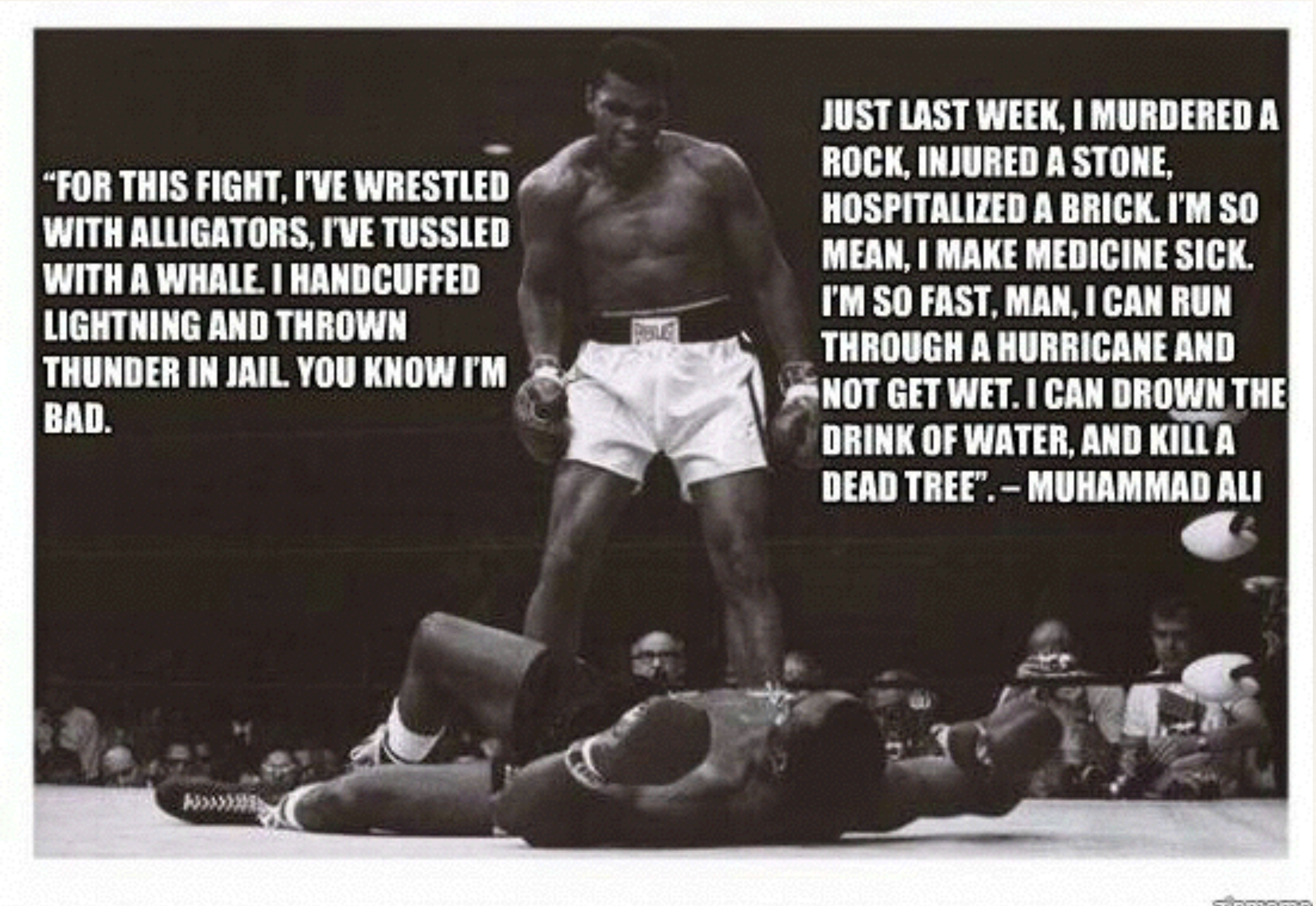
Metaphors in Language

"The stock market crashed today"



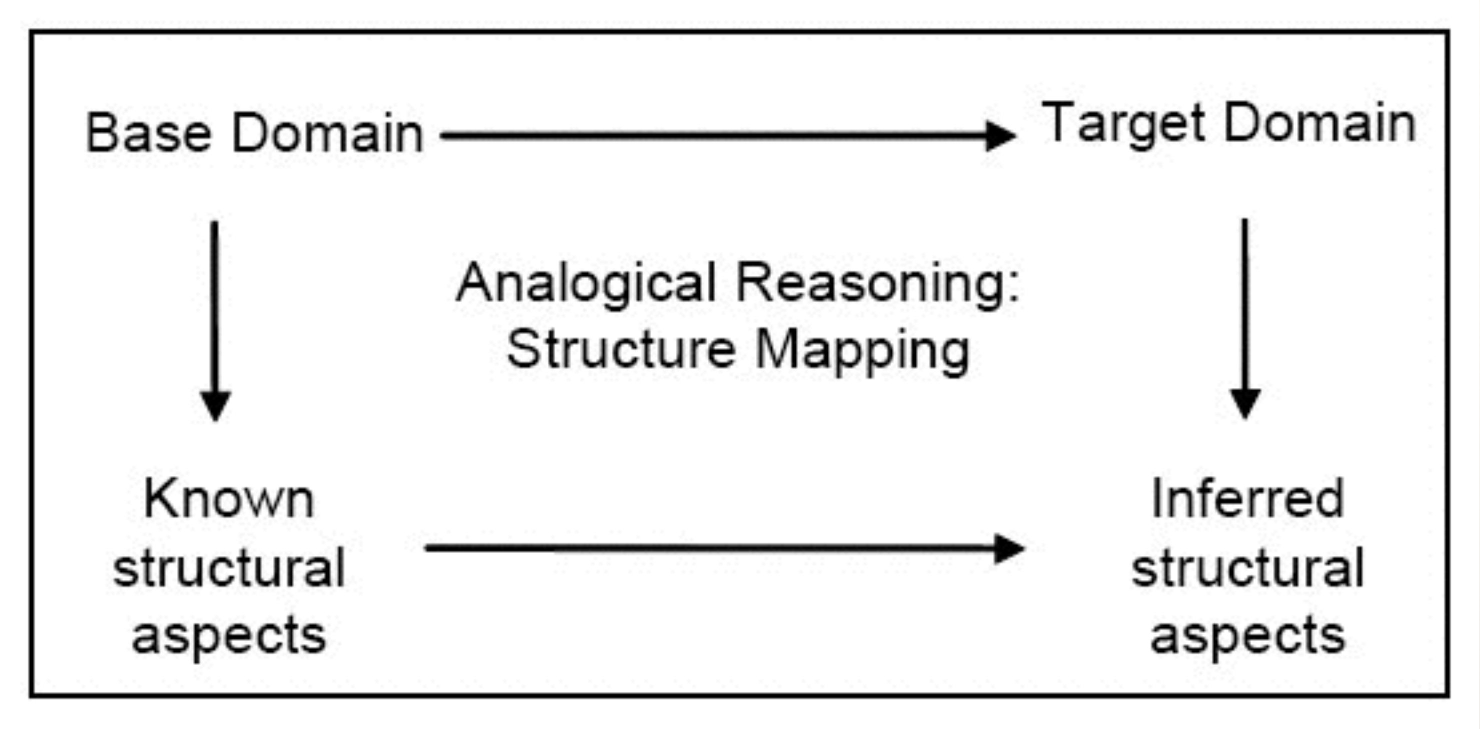


Metaphors in Language

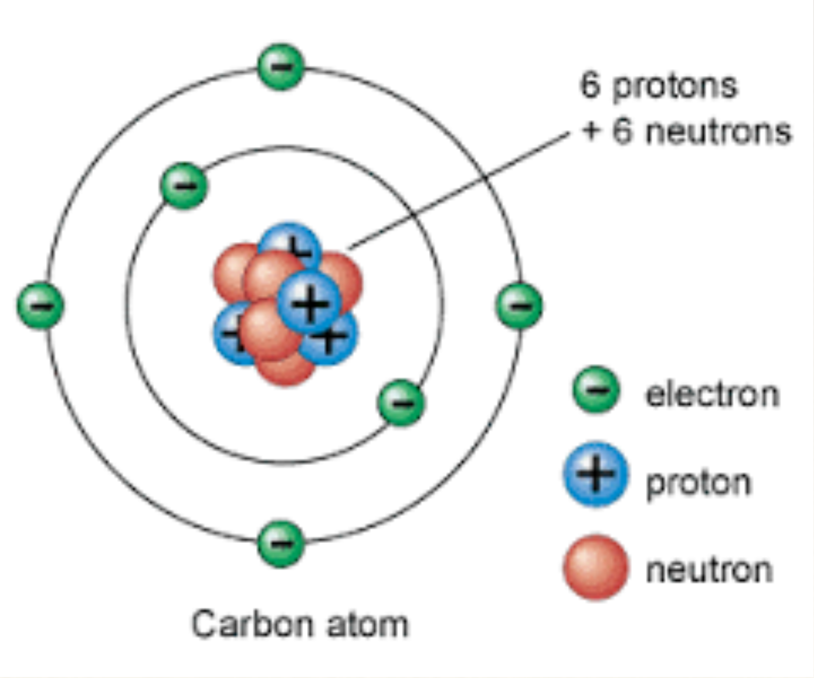


Cognitive Science Models

Gentner

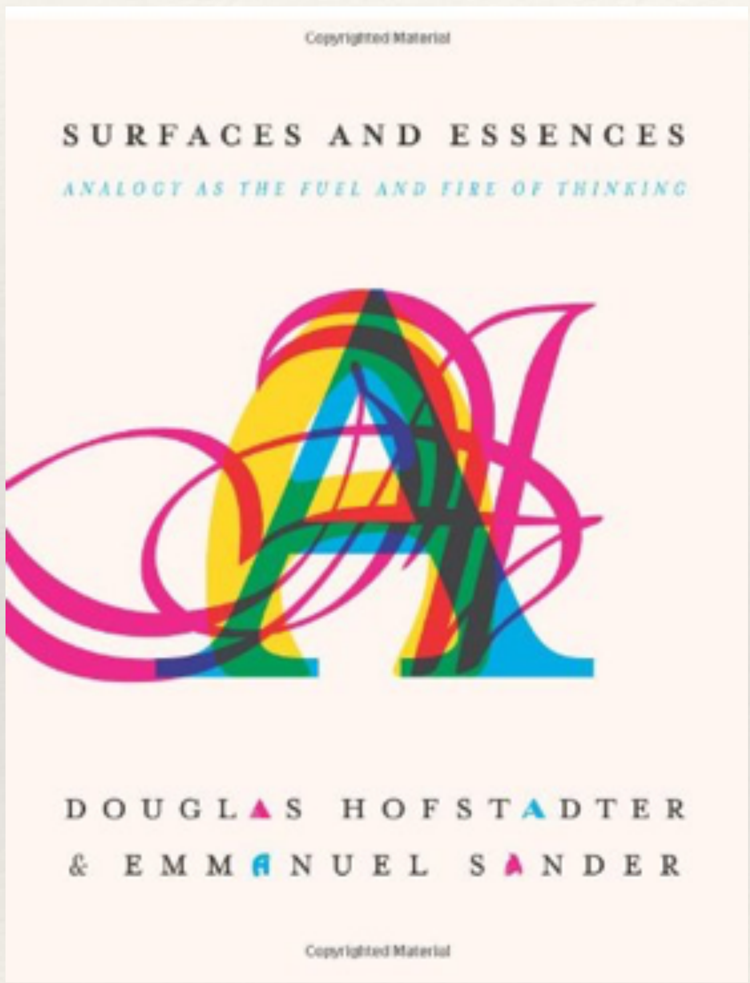


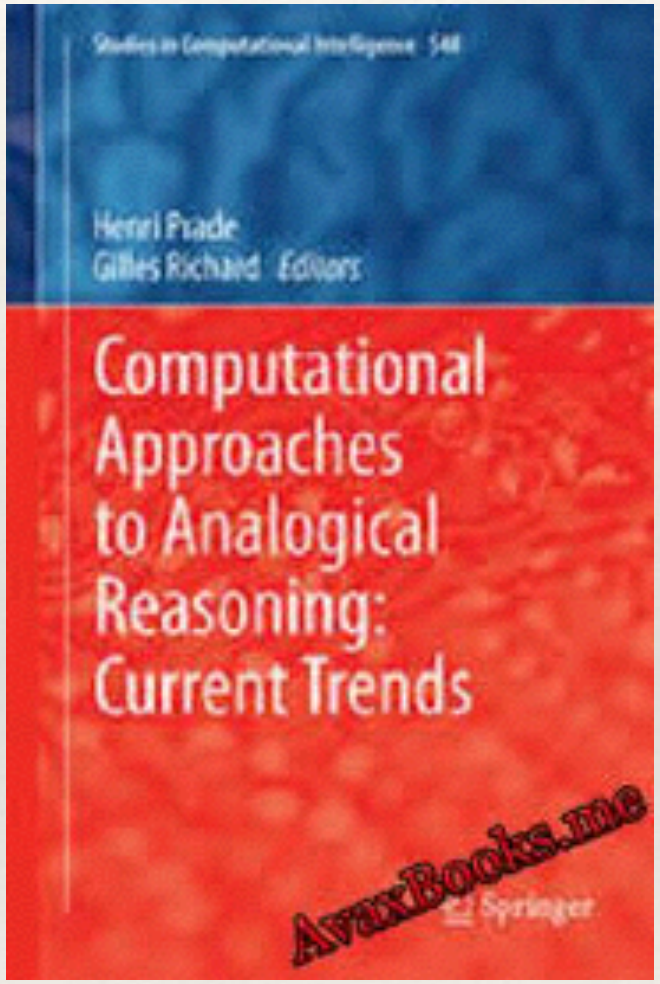
The atom is like the solar system



Solar system

Recent Books on Analogical Reasoning





Logical Approach to Analogy

- In IJCAI 1987, Stuart Russell and Todd Davies proposed the use of **determination rules** as a logical framework for analogy
- Determinations generalize the concept of functional dependencies in databases
- We intuitively think nationality determines language, in that speakers who share a nationality speak the same language

Determination Rules

(Rusell and Davies, IJCAI 87)

THE DEFINITION OF DETERMINATION:

$$\Sigma[\underline{x}, \underline{y}] \succ X[\underline{x}, \underline{z}]$$

iff

$$\forall \underline{y}, \underline{z} (\exists \underline{x}\, \Sigma[\underline{x}, \underline{y}] \wedge X[\underline{x}, \underline{z}]) \Rightarrow (\forall \underline{x}\, \Sigma[\underline{x}, \underline{y}] \Rightarrow X[\underline{x}, \underline{z}]).$$

$Make(Car_B) = Ford \wedge Make(Car_J) = Ford$

$Model(Car_B) = Mustang \wedge Model(Car_J) = Mustang$

$Design(Car_B) = GLX \wedge Design(Car_J) = GLX$

$Engine(Car_B) = V6 \wedge Engine(Car_J) = V6$

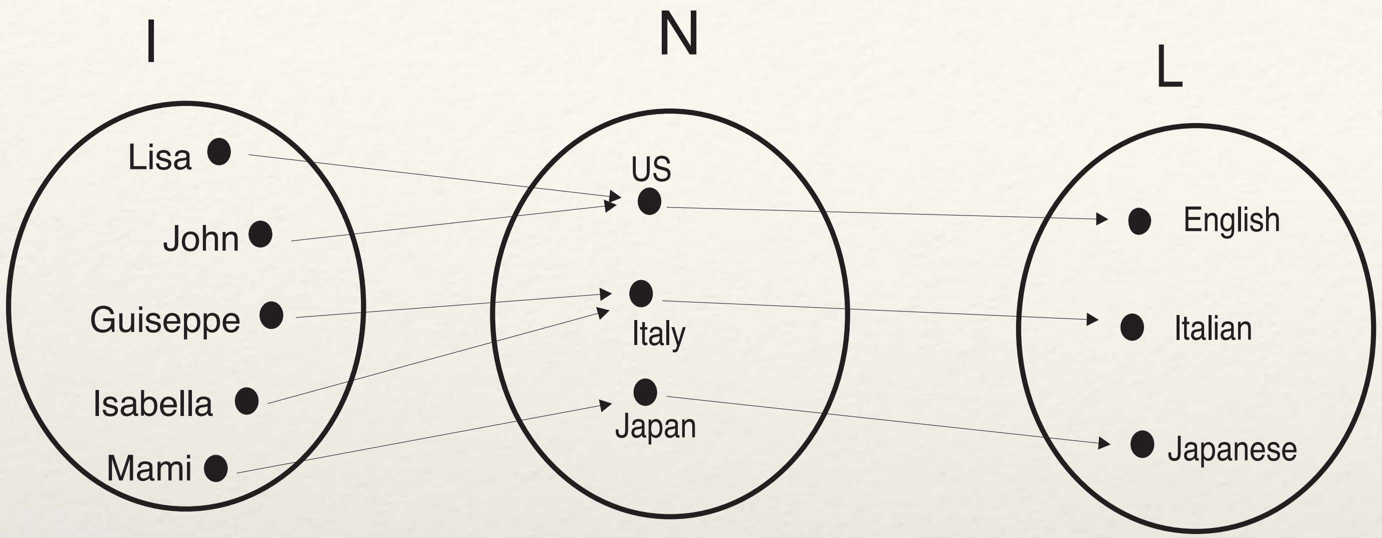
$Condition(Car_B) = Good \wedge Condition(Car_J) = Good$

$Year(Car_B) = 1982 \wedge Year(Car_J) = 1982$

$Value(Car_B) = \$3500$

$Value(Car_J) = \$3500,$

PAC Learning of Determinations



Mahadevan and Tadepalli
MLJ 1994

Theorem 4 *The space of functions $F_{\succ}$ consistent with a determination $P(x,y) \succ Q(x,z)$ is polynomial-time learnable if $|range(P)| \leq c$ and $|range(Q)| \leq l$ are polynomials in $|x| = n$.*

Determ.	*Dimension*	*|Examples| needed*	*P-time learnable if*
$P \succ$ Q	$\leq cl$	$\frac{1}{\epsilon}\{cl \ln 2 + \ln \frac{1}{\delta}\}$	$c \leq O(n^k)$
$P \succ_R Q$	cl	$\frac{1}{\epsilon}\{cl \ln 2 + \ln \frac{1}{\delta}\}$	$c \leq O(n^k)$
$P \succ_\forall Q$	$Min[2^c l, 2^n l]$	$\frac{1}{\epsilon}\ \{Min(2^c l, 2^n l) \ln 2 + \ln \frac{1}{\delta}\}$	$c \leq O(\log n)$
$P \succ_\subseteq Q$	$[2^{c/2} l, Min[2^c l, 2^n l]]$	$\frac{1}{\epsilon}\{2^c l \ln 2 + \ln \frac{1}{\delta}\}$	$c \leq O(\log n)$
$P \succ_\exists Q$	$[2^n(l-1), 2^n l]$	$\frac{1}{\epsilon}\{2^n l \ln 2 + \ln \frac{1}{\delta}\}$	Not Learnable
$P \succ_E^p Q$	$\leq cl + cl^2(p-1) +$ $cln(p-1) + \log(cl(p-1))$	$\frac{1}{\epsilon}\{(cl + cl^2(p-1) + cln(p-1) +$ $\log(cl(p-1))) \ln 2 + \ln \frac{1}{\delta}\}$	$c \leq O(n^k)$
$P \succ_P^\alpha Q$	$\leq cl + 2c^2 l^2 \alpha +$ $2c^2 l \alpha n + \log 2c^2 l \alpha$	$\frac{1}{\epsilon}\{(cl + 2c^2 l^2 \alpha + 2c^2 l \alpha n +$ $\log 2c^2 l \alpha) \ln 2 + \ln \frac{1}{\delta}\}$	$c \leq O(n^k)$

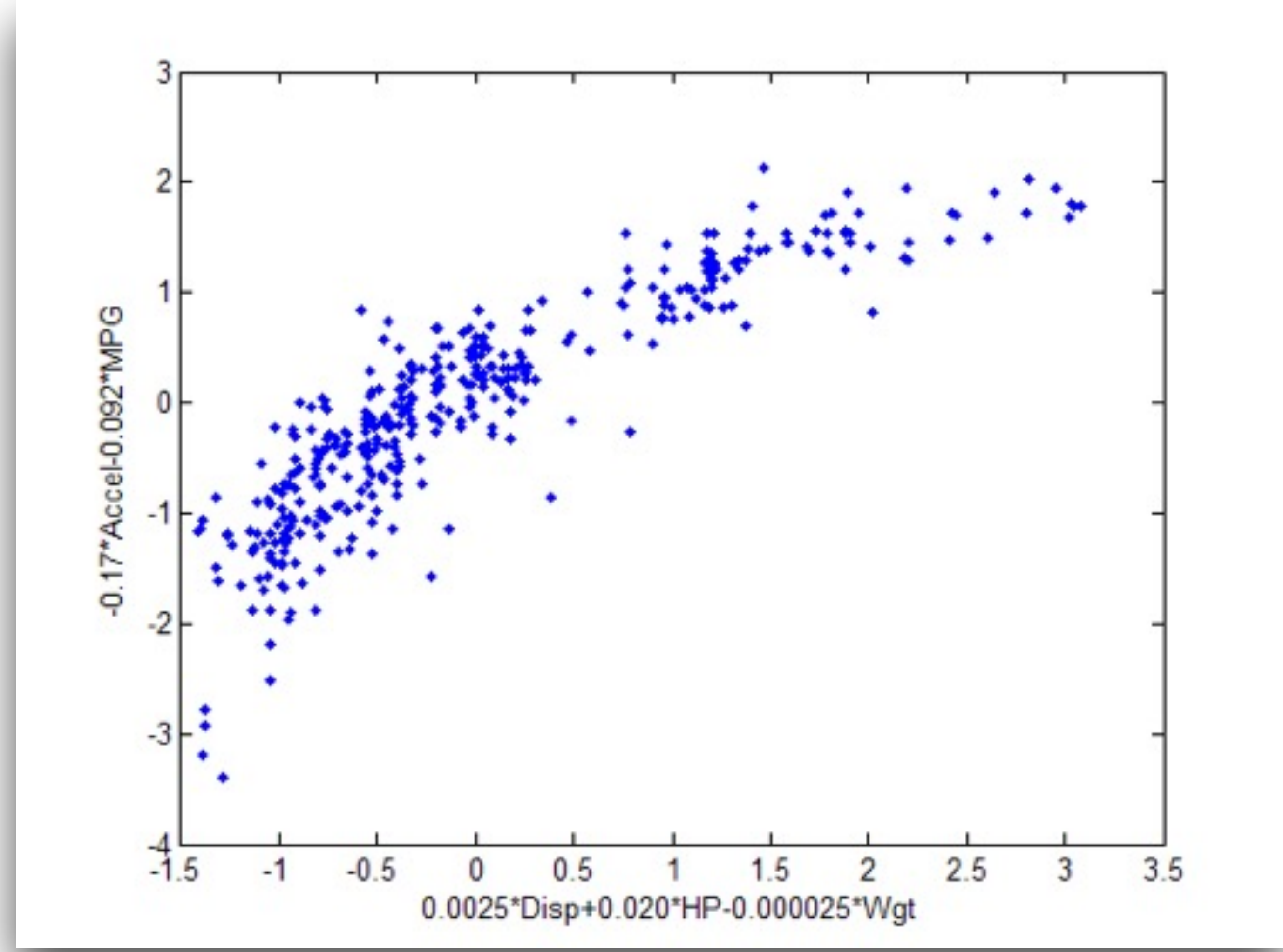
Learning from Multiple Datasets

- In many applications, multiple “views” or multiple datasets are constructed
  - Bioinformatics
  - Activity recognition
  - Computer graphics
  - Scientific exploration (MARS rover)
  - Cross-lingual information retrieval
  - Spectral methods for learning latent variable models

Exploiting Correlations

(Hotelling, 1936)

Acceleration MPG

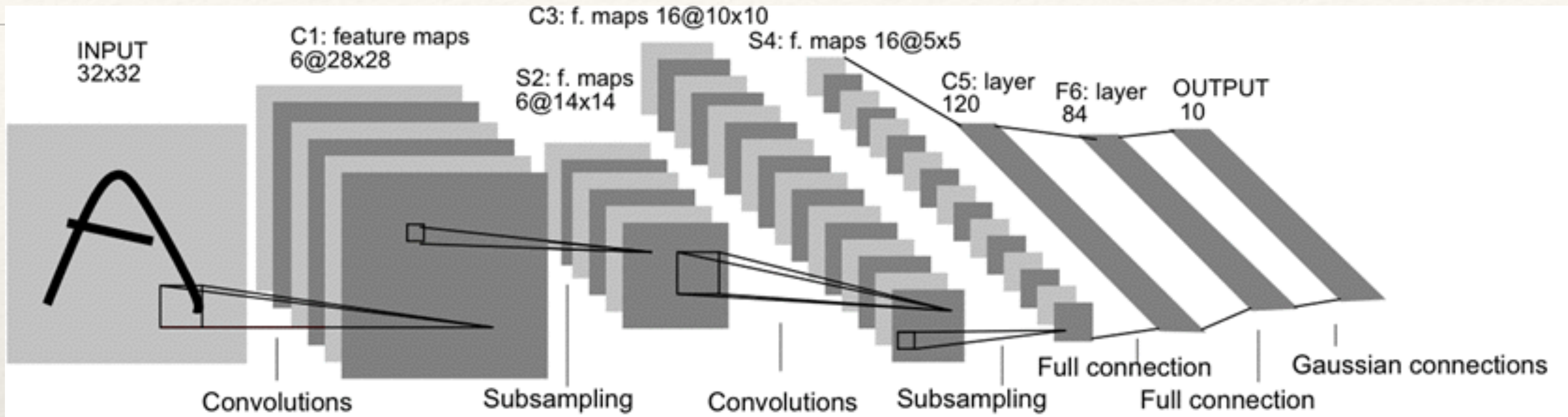


Displacement Horsepower Weight

$$\frac{u^T X^T Y v}{\sqrt{u^T X^T X u}\sqrt{v^T Y^T Y v}}$$

Find a projection of source and target vectors onto common latent space such that projected vectors are maximally correlated

Exploiting Symmetries



An early (Le-Net5) Convolutional Neural Network design, LeNet-5, used for recognition of digits

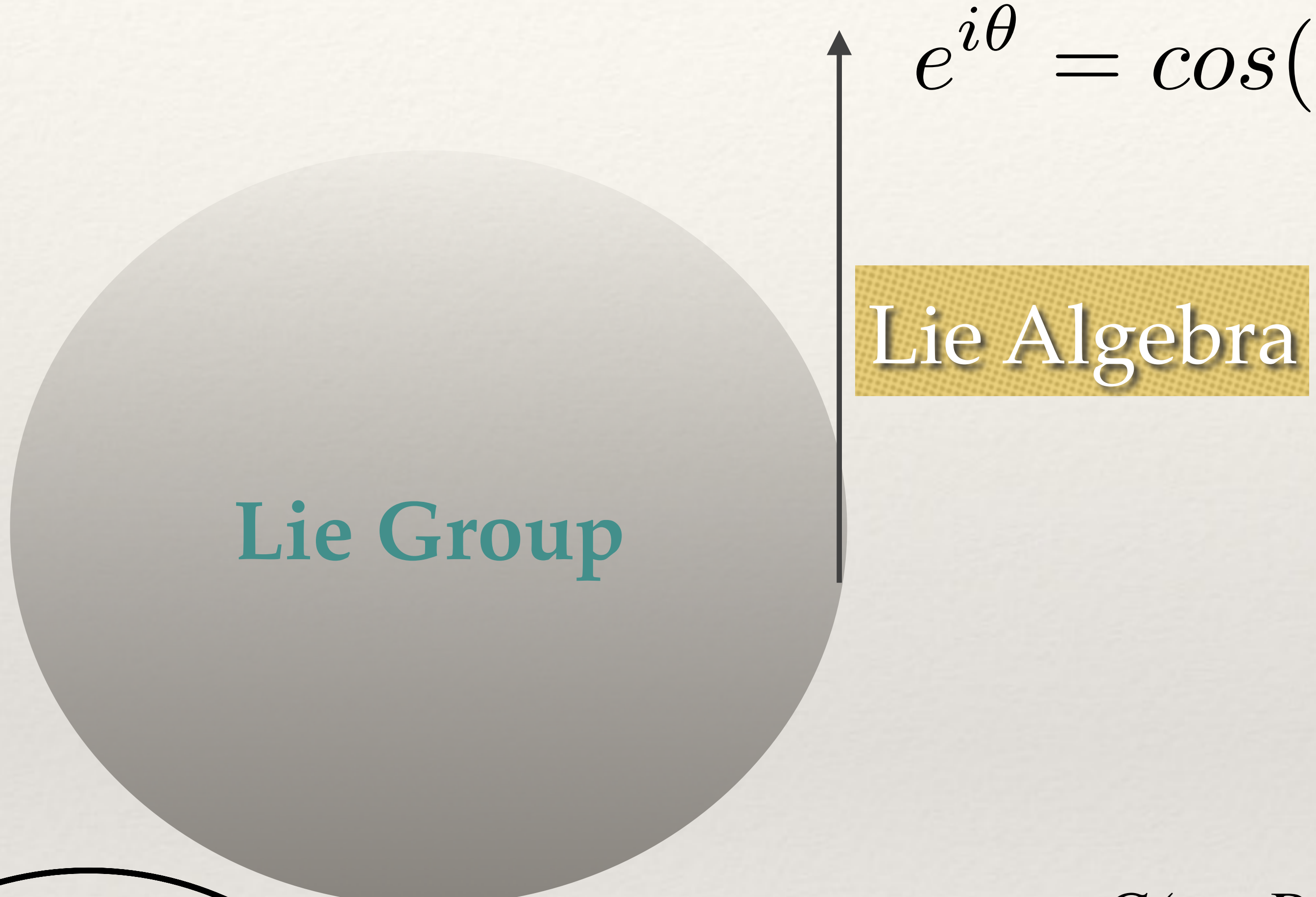
Deep RL in Atari (Mnih et al., Nature 2015)

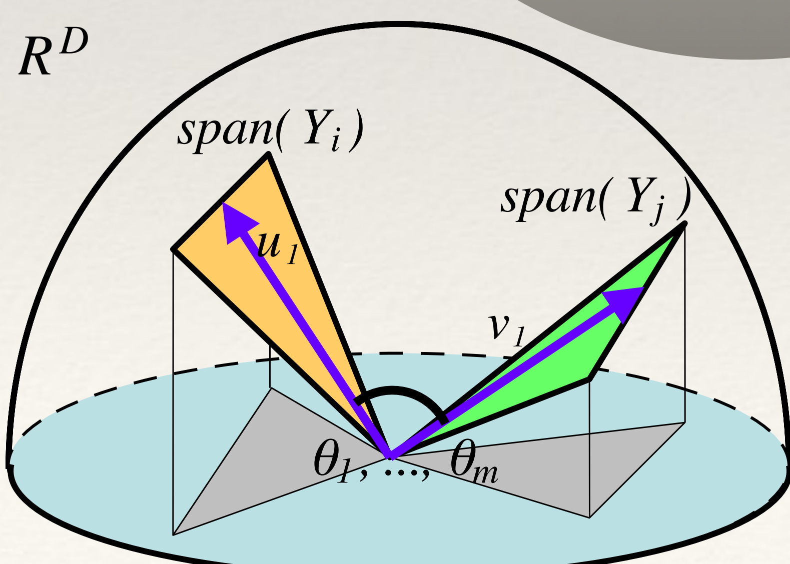
Learned filters

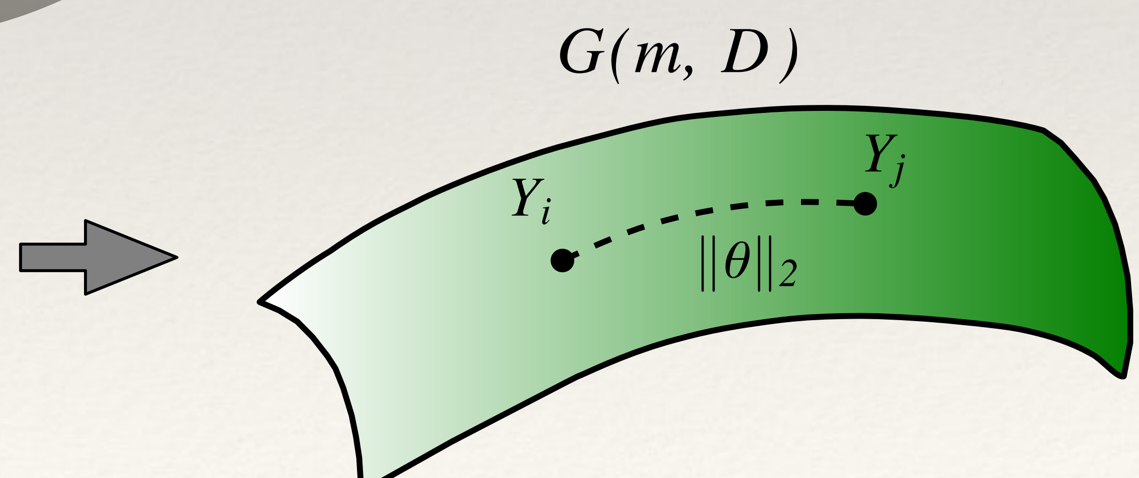
Group Theoretic Approaches

$$e^{i\theta} = cos(\theta) + isin(\theta)$$

Sphere
in n-dim

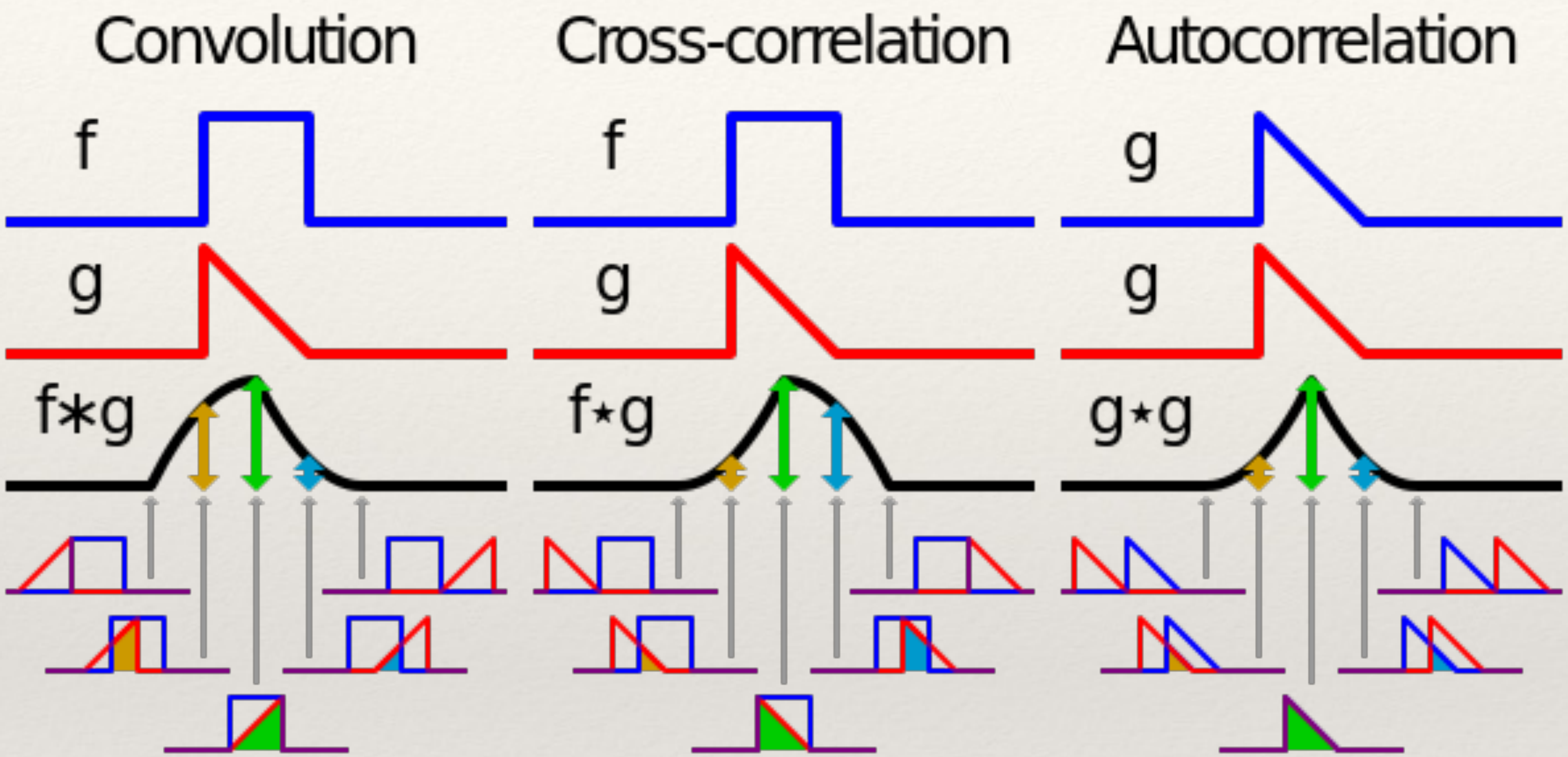






Grassmannian
approaches
(Ham & Lee)

Convolution and Group Theory



(Wikipedia)

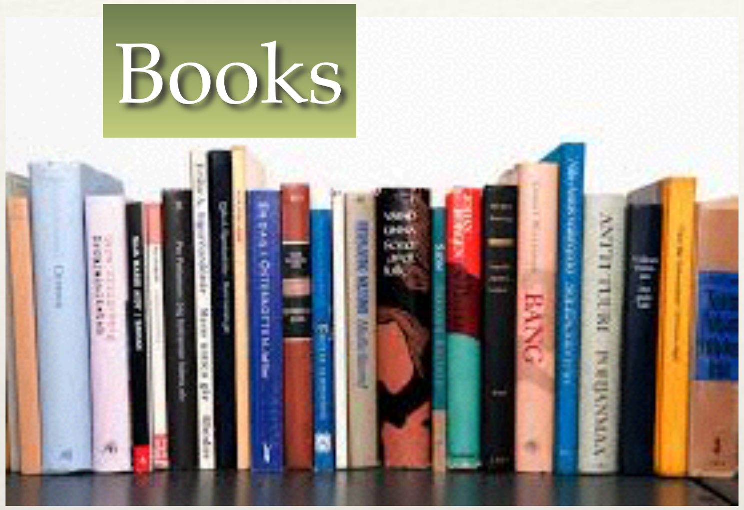
$$(f * g)(t) = \int_{-\infty}^{\infty} f(\tau) g(t - \tau) d\tau$$

Definition of Transfer Learning

Definition 1 *(Transfer Learning)* Given a source domain $\mathcal{D}_S$ and learning task $\mathcal{T}_S$, a target domain $\mathcal{D}_T$ and learning task $\mathcal{T}_T$, *transfer learning* aims to help improve the learning of the target predictive function $f_T(\cdot)$ in $\mathcal{D}_T$ using the knowledge in $\mathcal{D}_S$ and $\mathcal{T}_S$, where $\mathcal{D}_S \neq \mathcal{D}_T$, or $\mathcal{T}_S \neq \mathcal{T}_T$.

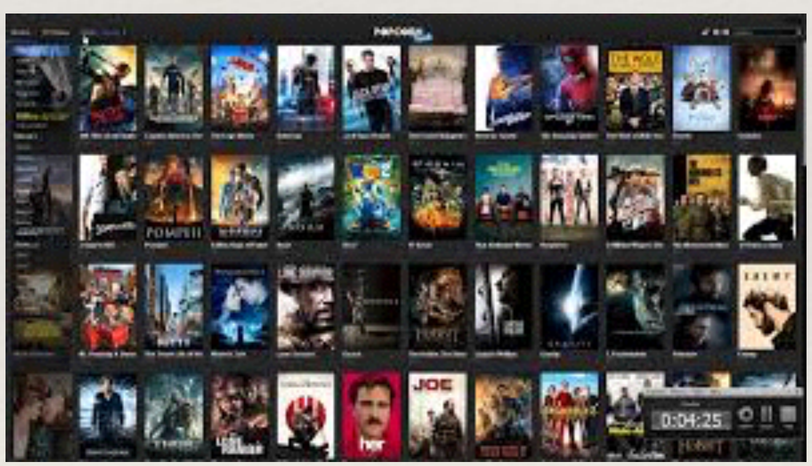
[Pan and Yang, IEEE Trans]

Amazon Sentiment Analysis



Books

"A great read. You get an opportunity to glimpse how a great scientific mind thinks and how the person lived."



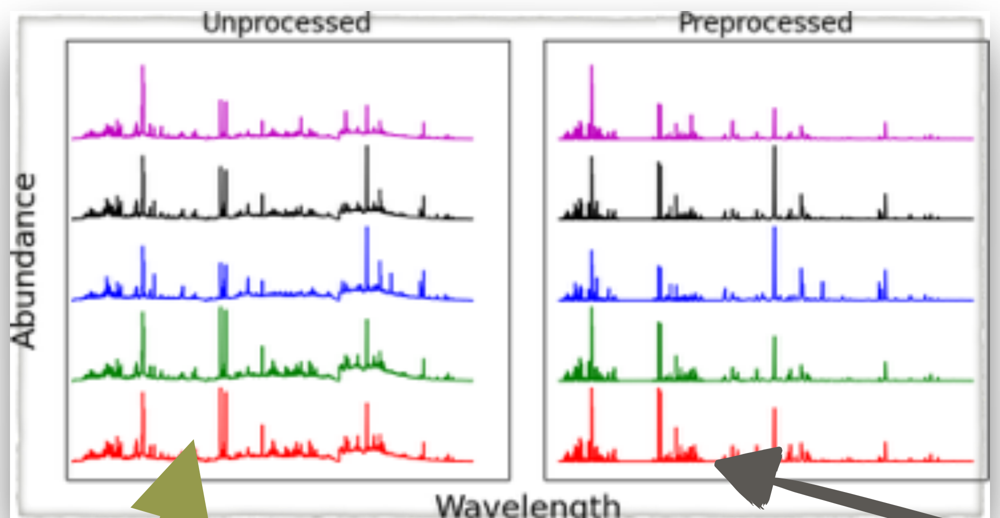
Movies

"Fantastic performances from every actor. I appreciate that this movie doesn't feel that it needs to take an already dramatic topic and dramatize it even more. It takes itself seriously, and presents the story without unnecessary drama. Highly recommended."

Computer Vision Transfer

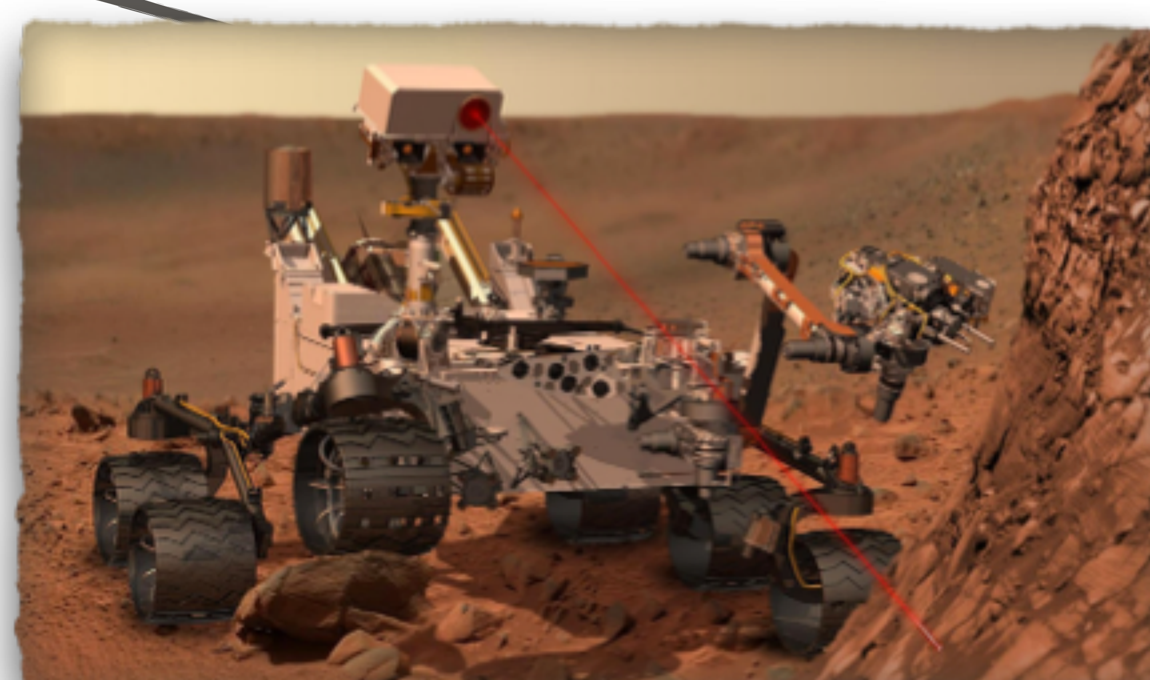
Transfer Learning on Mars

(Dyar, Mahadevan et al.)

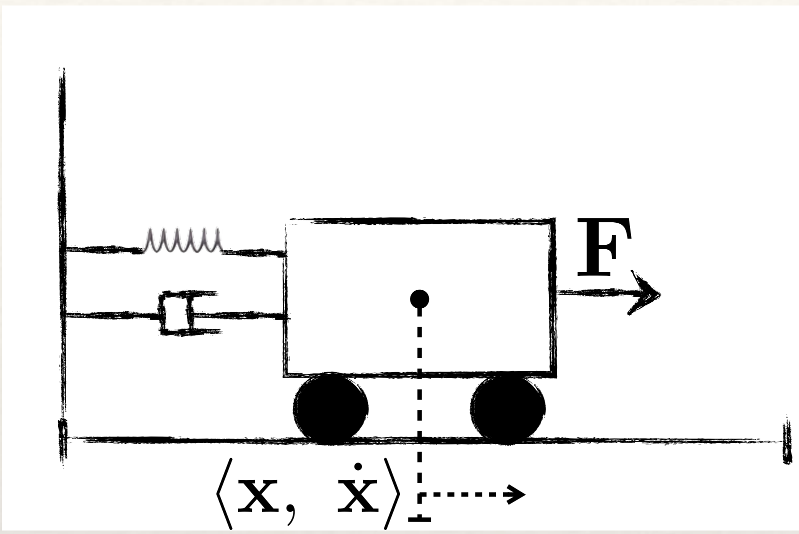


Curiosity zapping a rock with a laser

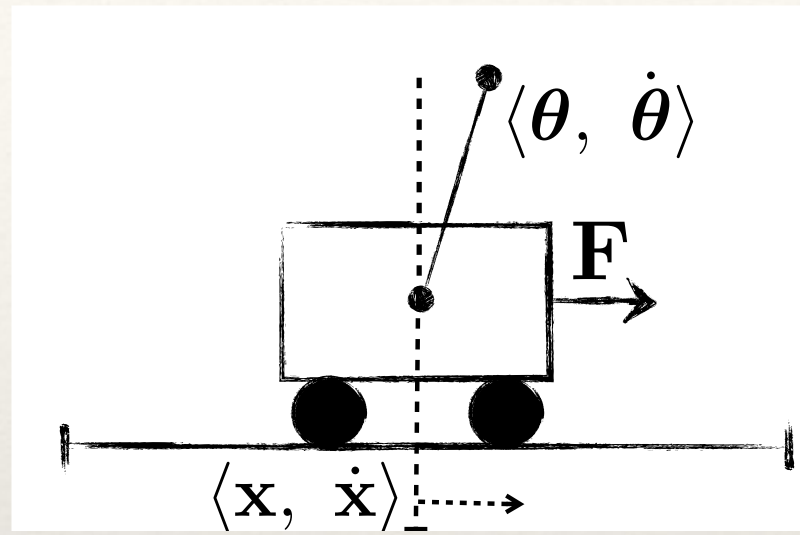
Same laser on Earth as on Mars



Transfer in Reinforcement Learning

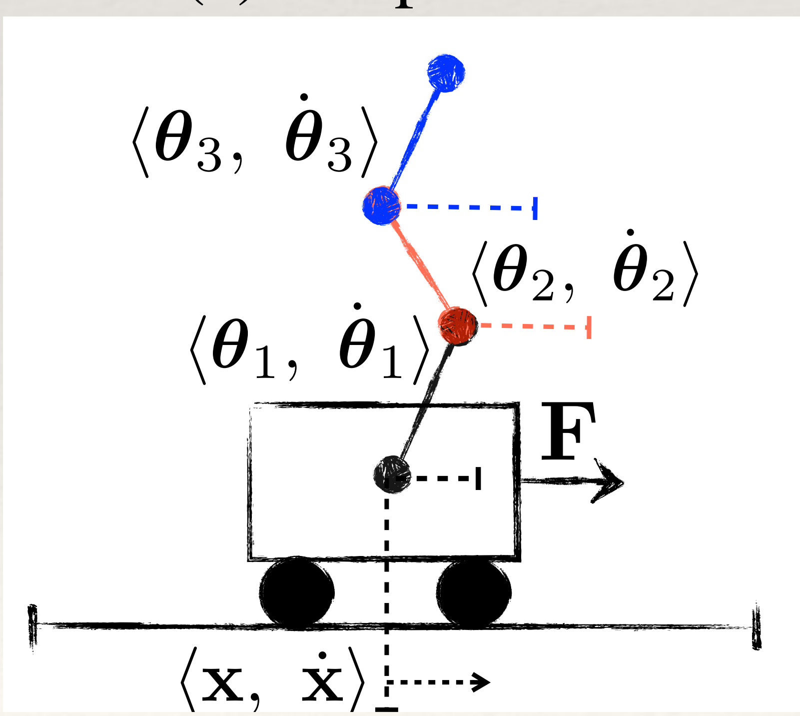


(a) Simple Mass

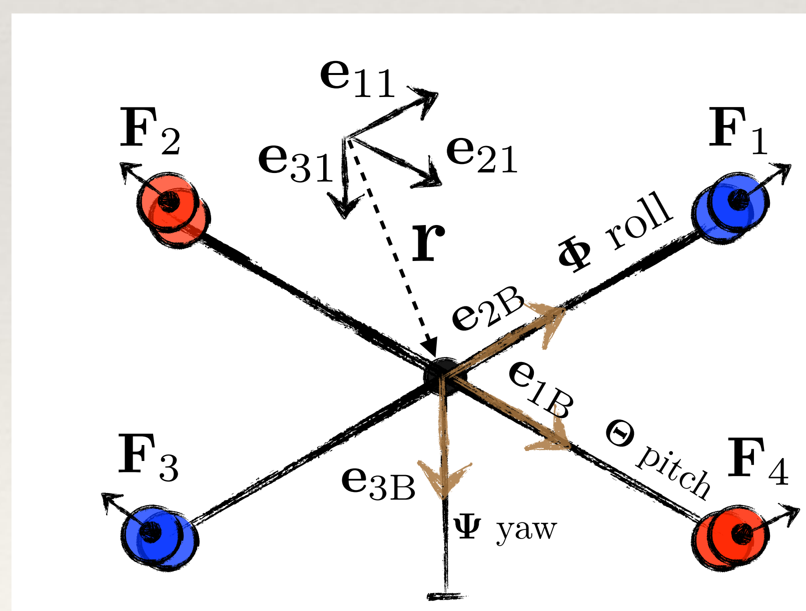


(b) Cart Pole

(Ammar et al., AAAI 2015)

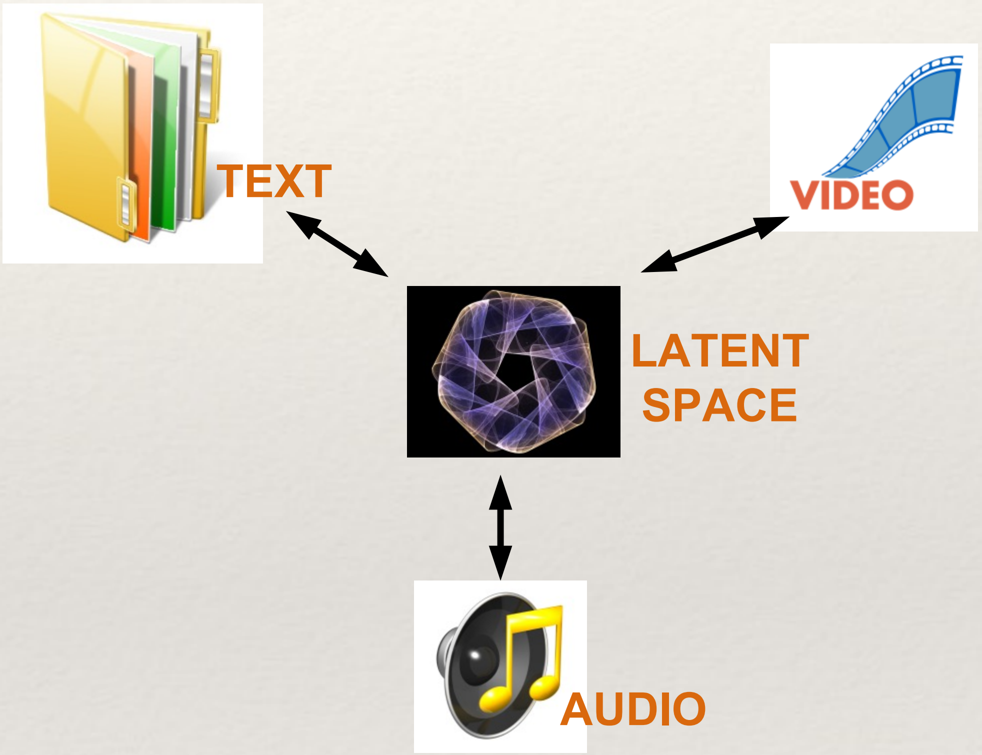


(c) Three-Link Cart Pole



(d) Quadrotor

Multi-modal transfer learning



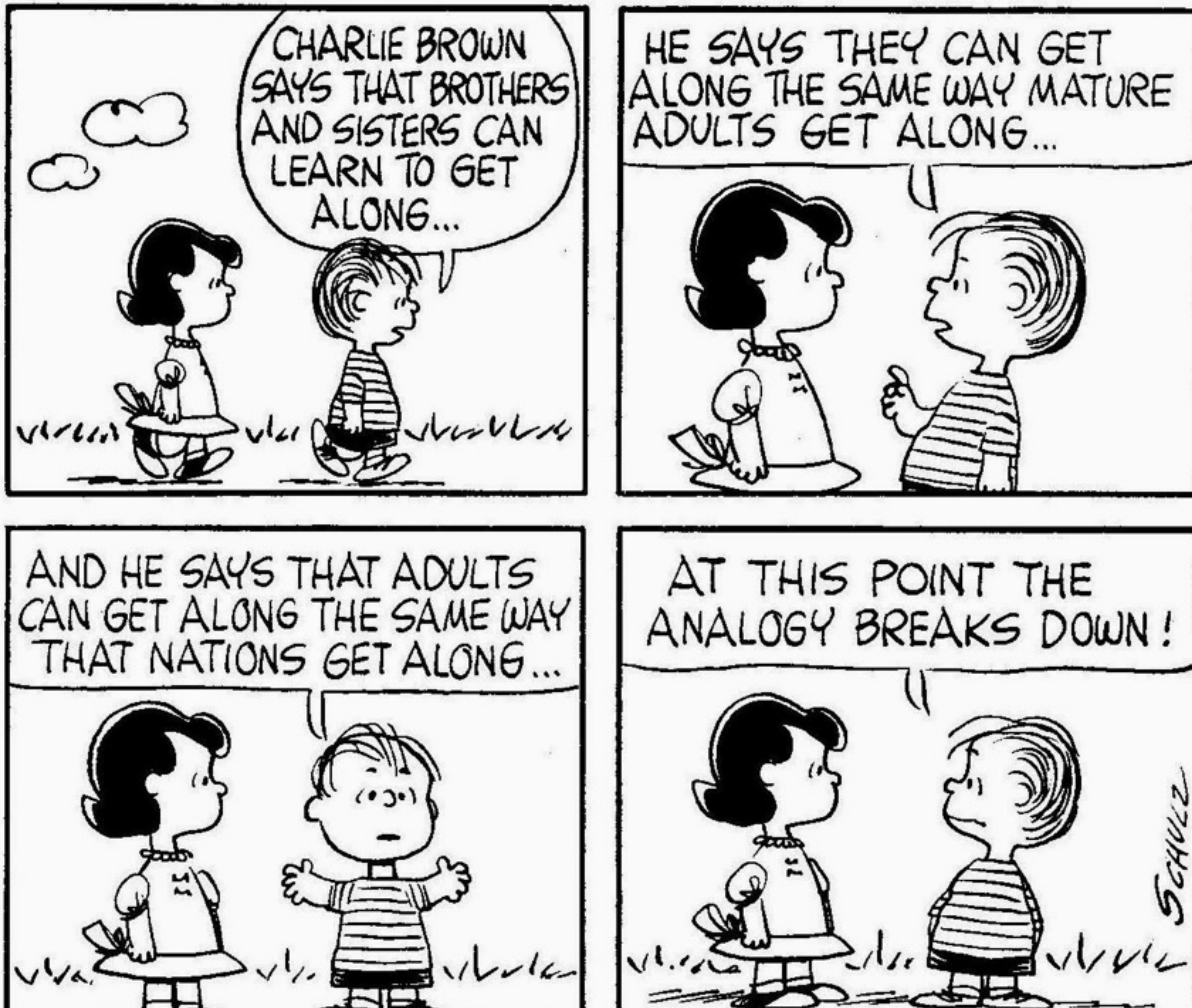
[flowers, grass, tiger, water]

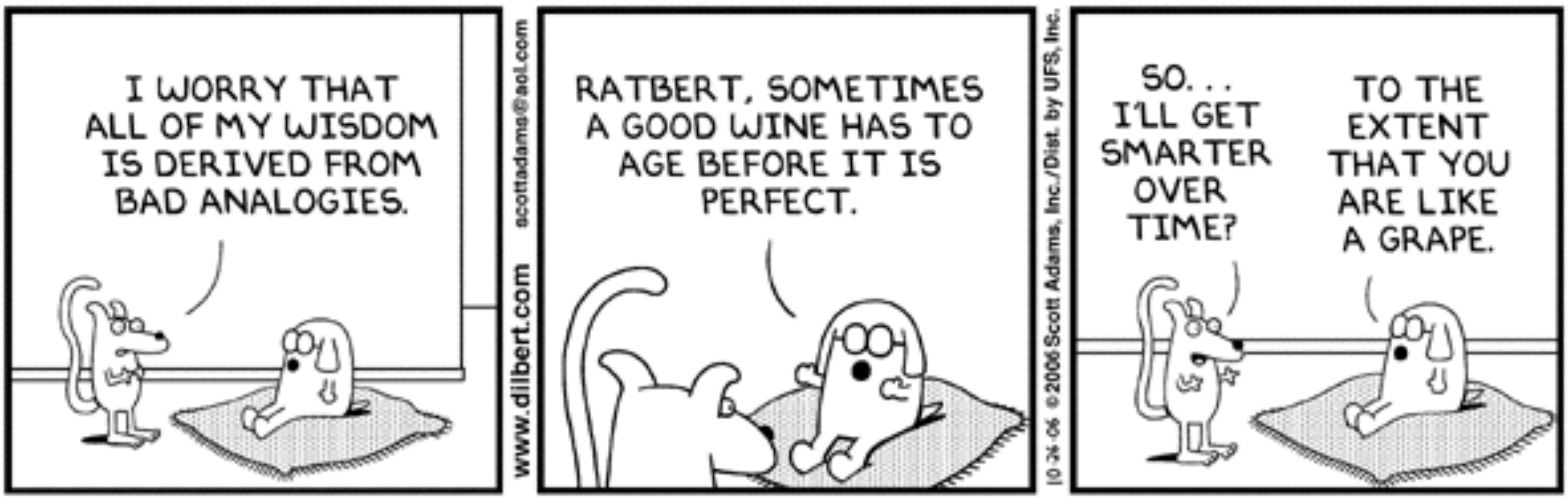
Why is Transfer Learning Difficult?

- High-dimensional datasets (images, text, speech)
- Source and target domains may not share features (e.g., words in English and German)
- Lack of sufficient correspondences
- Limited number of labeled examples in source and target

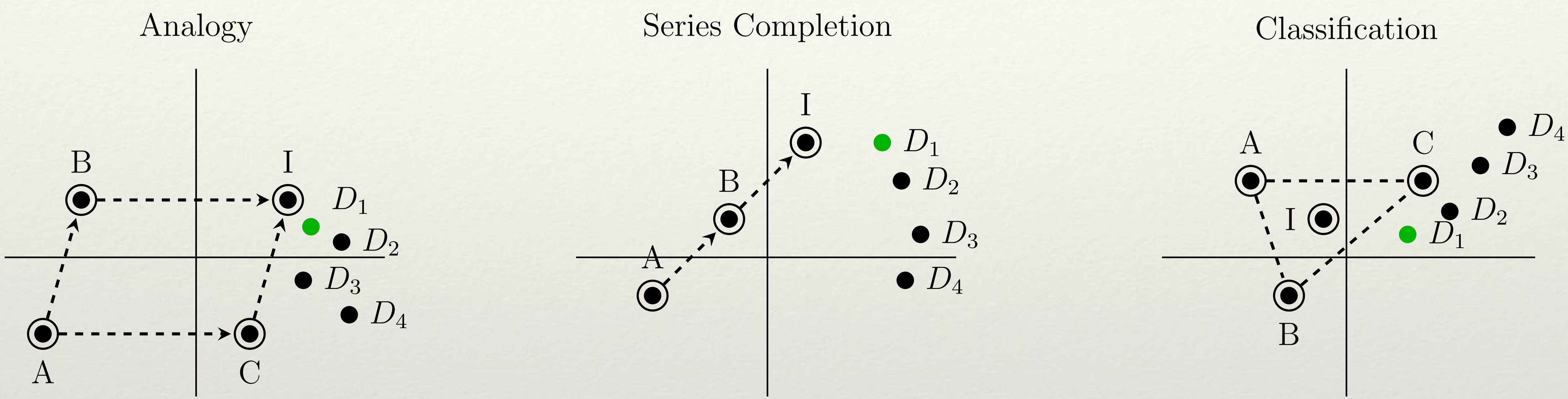
Outline of the Tutorial

- Historical review and motivation (20 minutes)
- **Mathematical background (20 minutes)**
- Algorithms (30 minutes)
- Applications (30 minutes)
- Questions (5 minutes)


CHARLIE BROWN SAYS THAT BROTHERS AND SISTERS CAN LEARN TO GET ALONG...
HE SAYS THEY CAN GET ALONG THE SAME WAY MATURE ADULTS GET ALONG...
AND HE SAYS THAT ADULTS CAN GET ALONG THE SAME WAY THAT NATIONS GET ALONG...
AT THIS POINT THE ANALOGY BREAKS DOWN!
SCHULZ


I WORRY THAT ALL OF MY WISDOM IS DERIVED FROM BAD ANALOGIES.
scottadams@aol.com
www.dilbert.com
RATBERT, SOMETIMES A GOOD WINE HAS TO AGE BEFORE IT IS PERFECT.
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.
SO... I'LL GET SMARTER OVER TIME?
TO THE EXTENT THAT YOU ARE LIKE A GRAPE.

Sternberg's Vector Space Model



he is to she as grandpa is to X?

R. J. Sternberg and M. K. Gardner. Unities in inductive reasoning. *Journal of Experimental Psychology: General*, 112(1):80, 1983.

Analogical Reasoning in NLP

Athens is to Greece as Baghdad is to ?

he is to she as grandpa is to X?

cheap is to cheaper as high is to X?

Europe is to euro as Vietnam is to X?

NLP

Linguistic Reasoning by Vector Arithmetic

$$queen \approx king - man + woman$$

$$\underset{b^* \in V}{\arg\max} \left(\cos \left(b^*, b - a + a^* \right) \right)$$

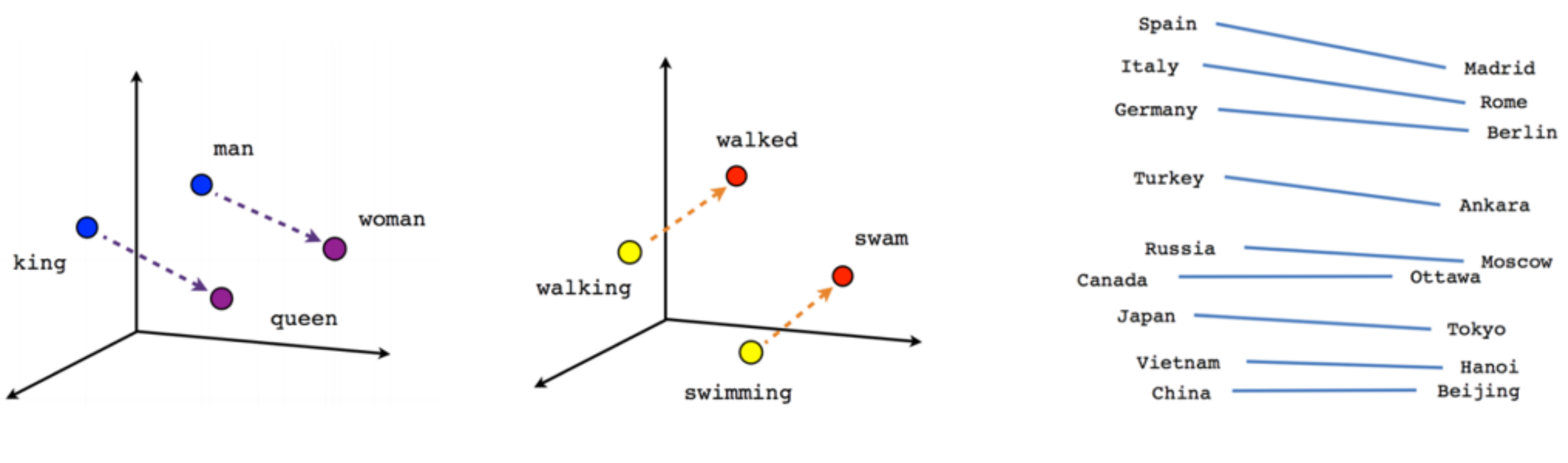
Mikolov et al., 2013

Levy and Goldberg, 2014

To achieve better balance among the different aspects of similarity, we propose switching from an additive to a multiplicative combination:

$$\arg\max_{b^* \in V} \frac{\cos(b^*, b)\cos(b^*, a^*)}{\cos(b^*, a) + \varepsilon}$$

Modeling of Linguistic Relations



Male-Female

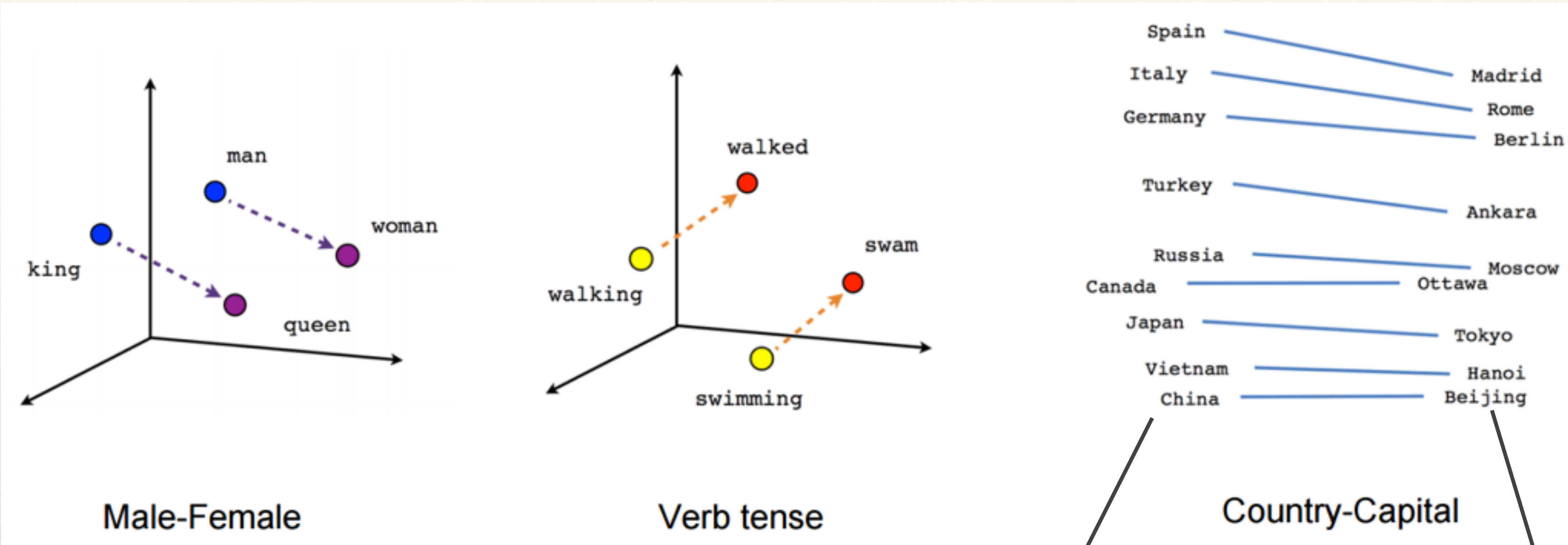
Verb tense

Country-Capital

(Sternberg and Gardner, 1983; Mikolov et al., 2013)

Matrix Manifold Model of Linguistic Relations

(Mahadevan and Chandar, Arxiv, 2015)



We show later that matrix manifold representations of linguistic relations are far superior to linear vector translation approaches

geodesic on Grassmannian manifold

Countries subspace

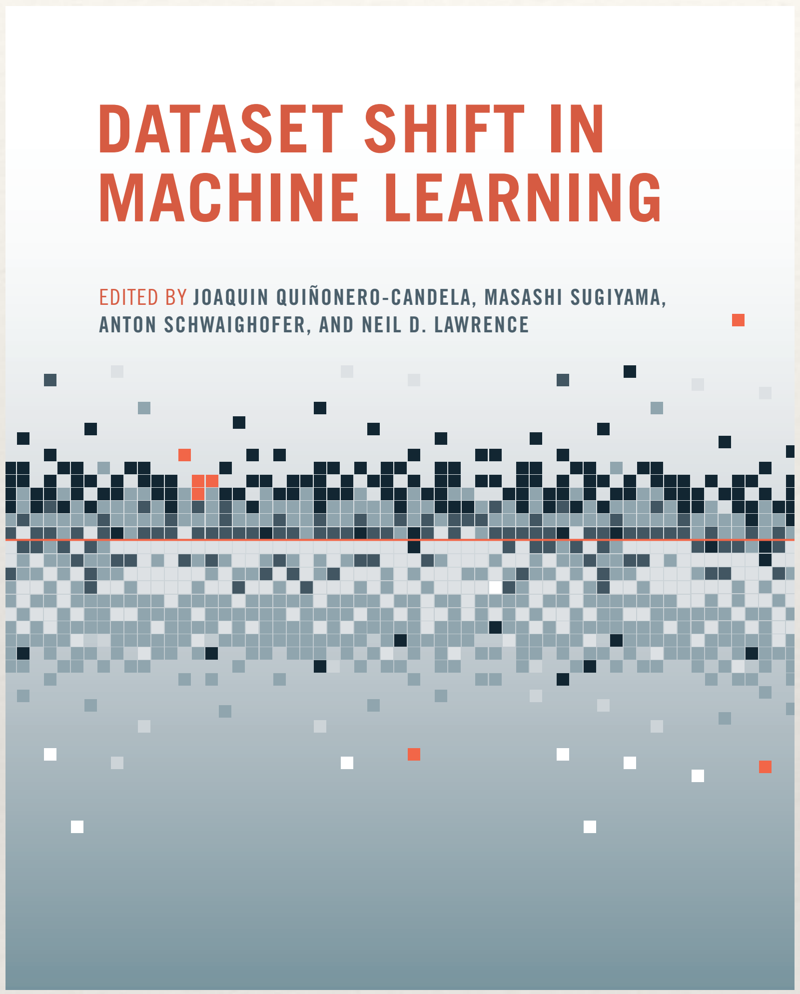
Capitals subspace

PCA derived subspaces of word vectors

ML Techniques

- Instance reweighing methods
  - Domain adaptation
- Linear Feature (subspace) construction methods
  - CCA, Manifold alignment
  - Subspace alignment
  - Geodesic flow kernels
- Nonlinear feature construction approaches
  - Deep learning

Some Surveys



Journal of Artificial Intelligence Research 26 (2006) 101-126 Submitted 8/05; published 5/06

Domain Adaptation for Statistical Classifiers

Hal Daumé III HDAUME@ISI.EDU
Daniel Marcu MARCU@ISI.EDU
Information Sciences Institute
University of Southern California
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292 USA

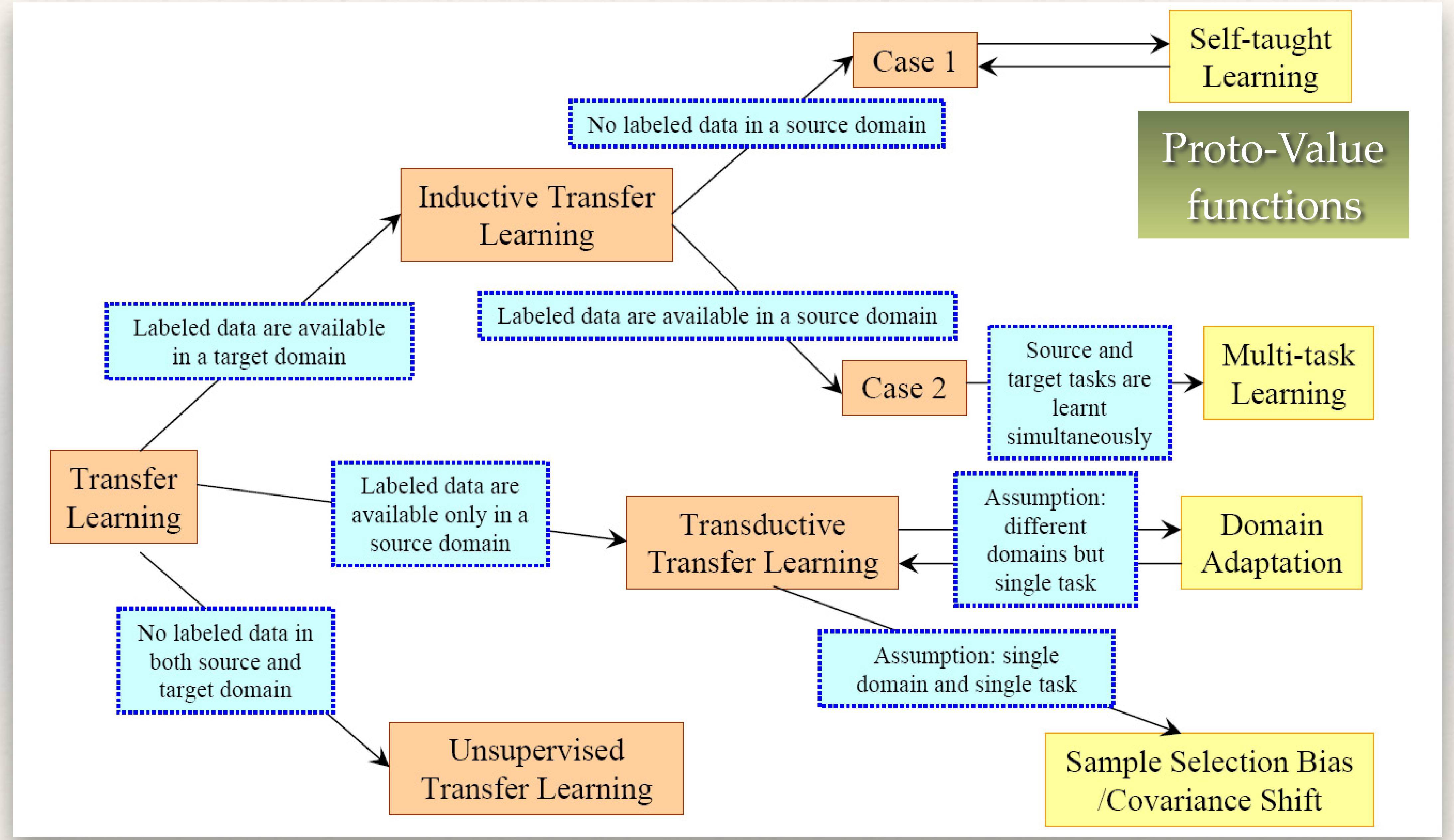
A Survey on Transfer Learning

Sinno Jialin Pan and Qiang Yang *Fellow, IEEE*

Abstract—A major assumption in many machine learning and data mining algorithms is that the training and future data must be in the same feature space and have the same distribution. However, in many real-world applications, this assumption may not hold. For example, we sometimes have a classification task in one domain of interest, but we only have sufficient training data in another domain of interest, where the latter data may be in a different feature space or follow a different data distribution. In such cases, knowledge transfer, if done successfully, would greatly improve the performance of learning by avoiding much expensive data labeling efforts. In recent years, transfer learning has emerged as a new learning framework to address this problem. This survey focuses on categorizing and reviewing the current progress on transfer learning for classification, regression and clustering problems. In this survey, we discuss the relationship between transfer learning and other related machine learning techniques such as domain adaptation, multi-task learning and sample selection bias, as well as co-variate shift. We also explore some potential future issues in transfer learning research.

Index Terms—Transfer Learning, Survey, Machine Learning, Data Mining.

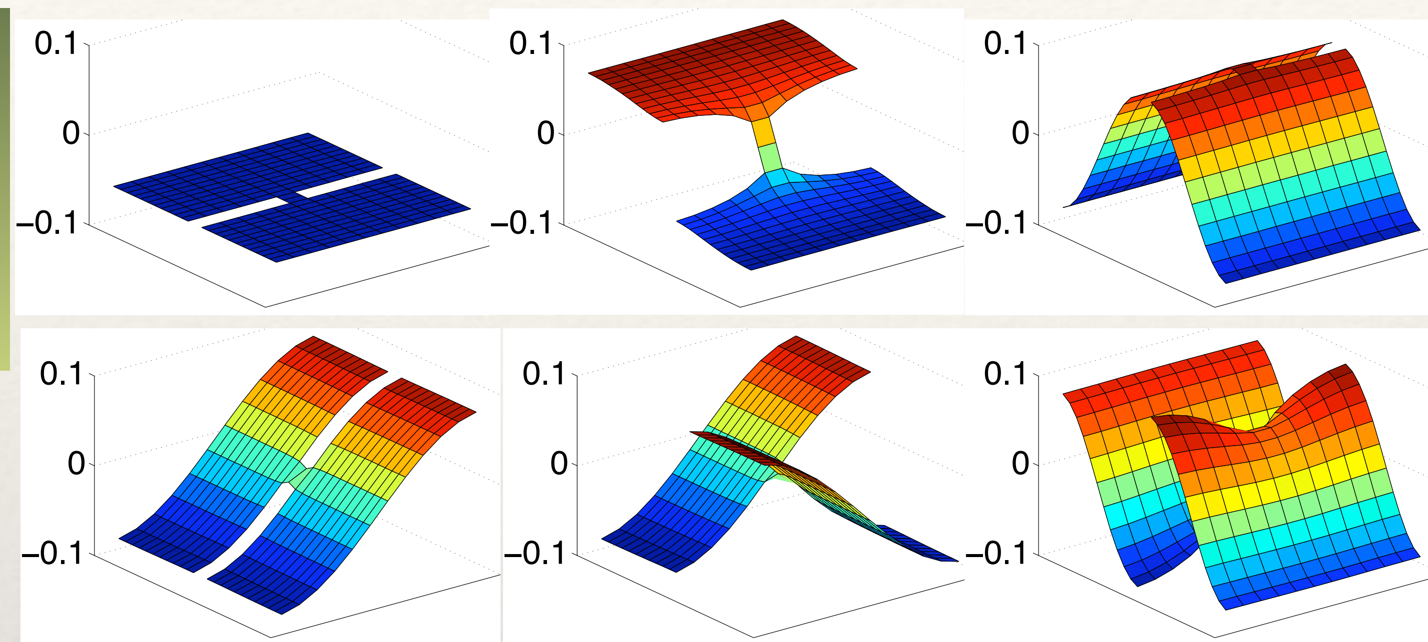
A Taxonomy of Transfer Learning



Pan and Yang, A Survey of Transfer Learning, IEEE TKDE 2010

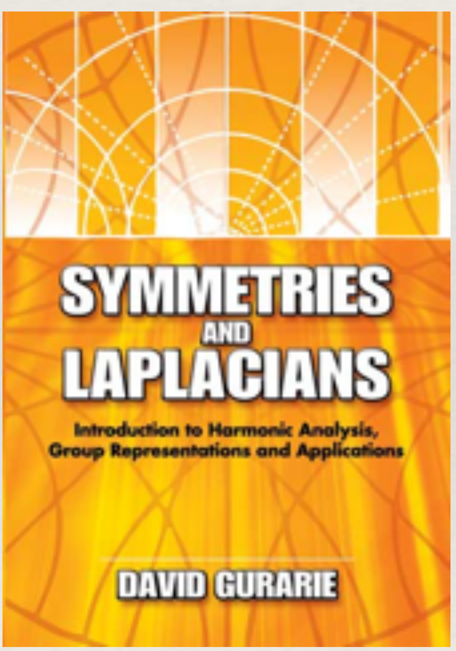
Proto-Value Function Approximation

Eigenvectors of the MDP graph Laplacian
L = D - W

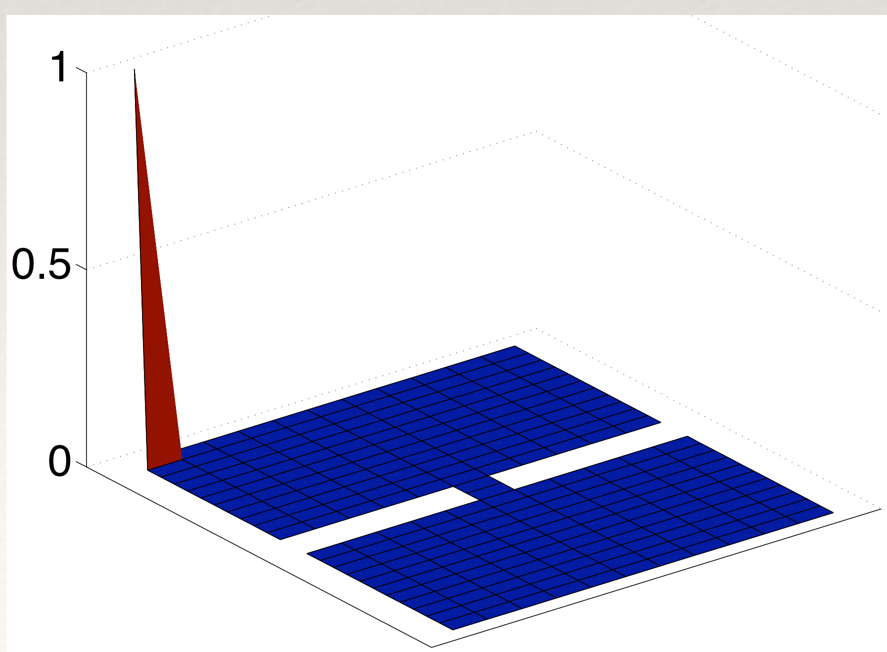


[Mahadevan, ICML 2005;
Mahadevan & Maggioni, JMLR 2007]

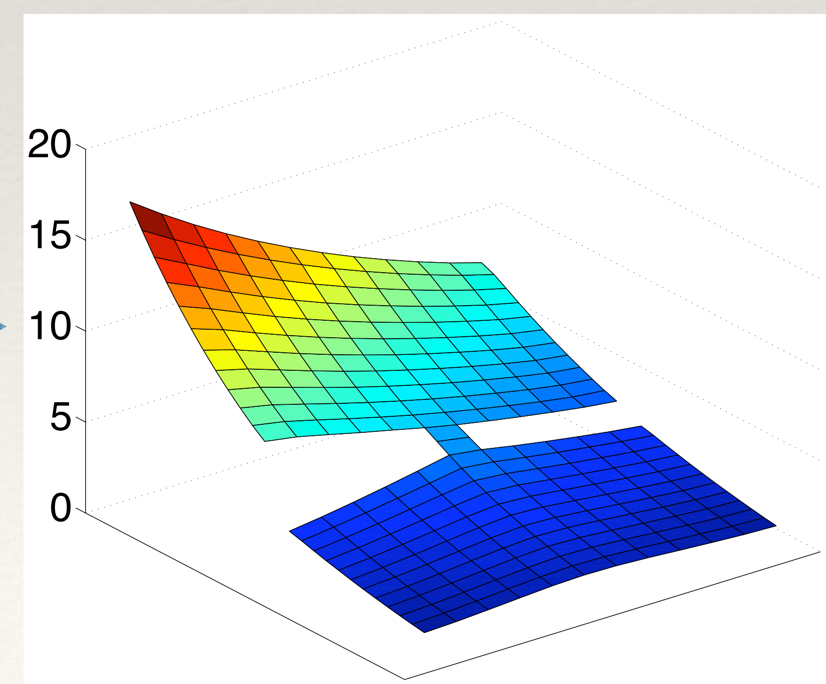
Reward-invariant representations



R

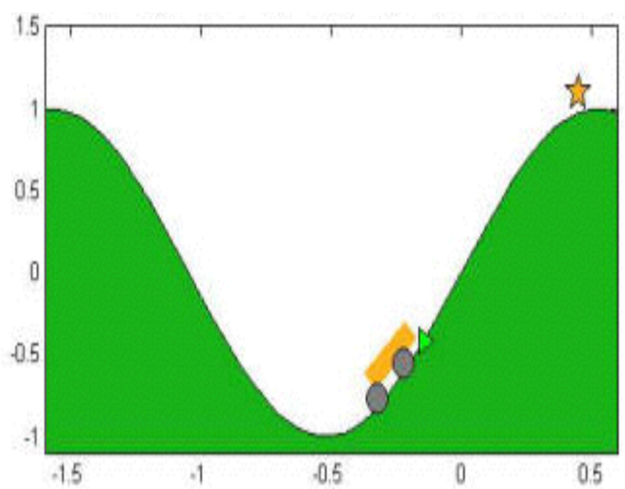


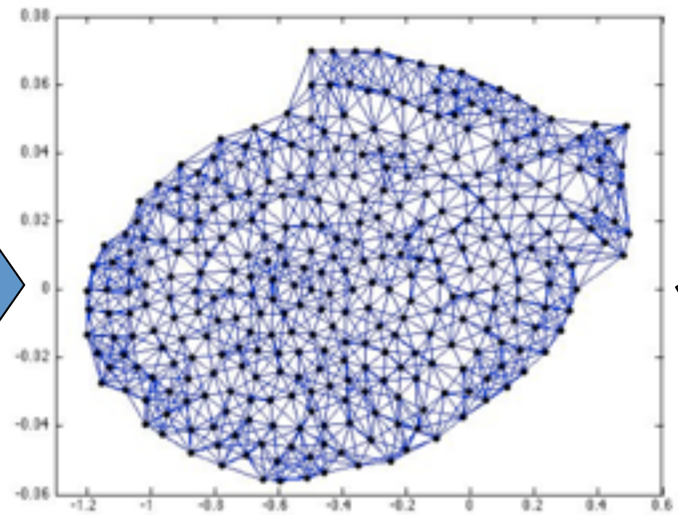
V^*

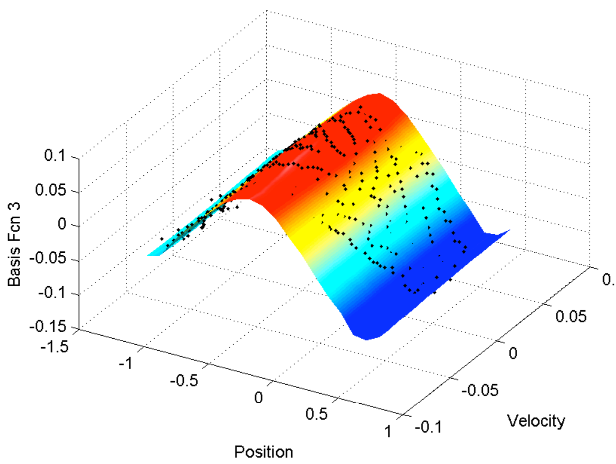


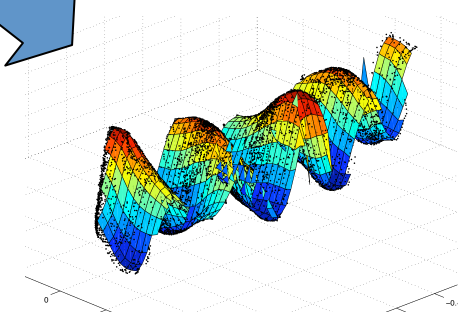
Extensions to Continuous MDPs

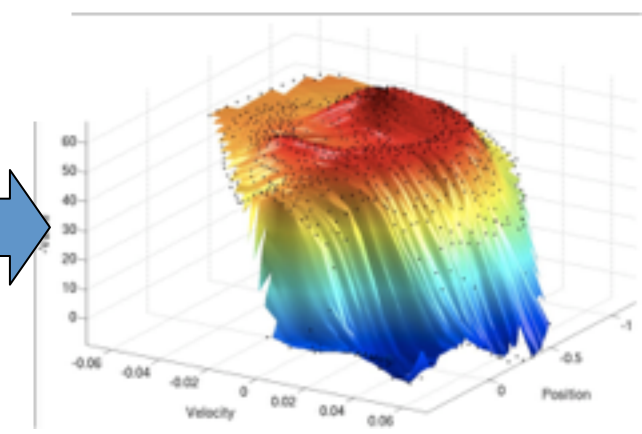
Mountain car





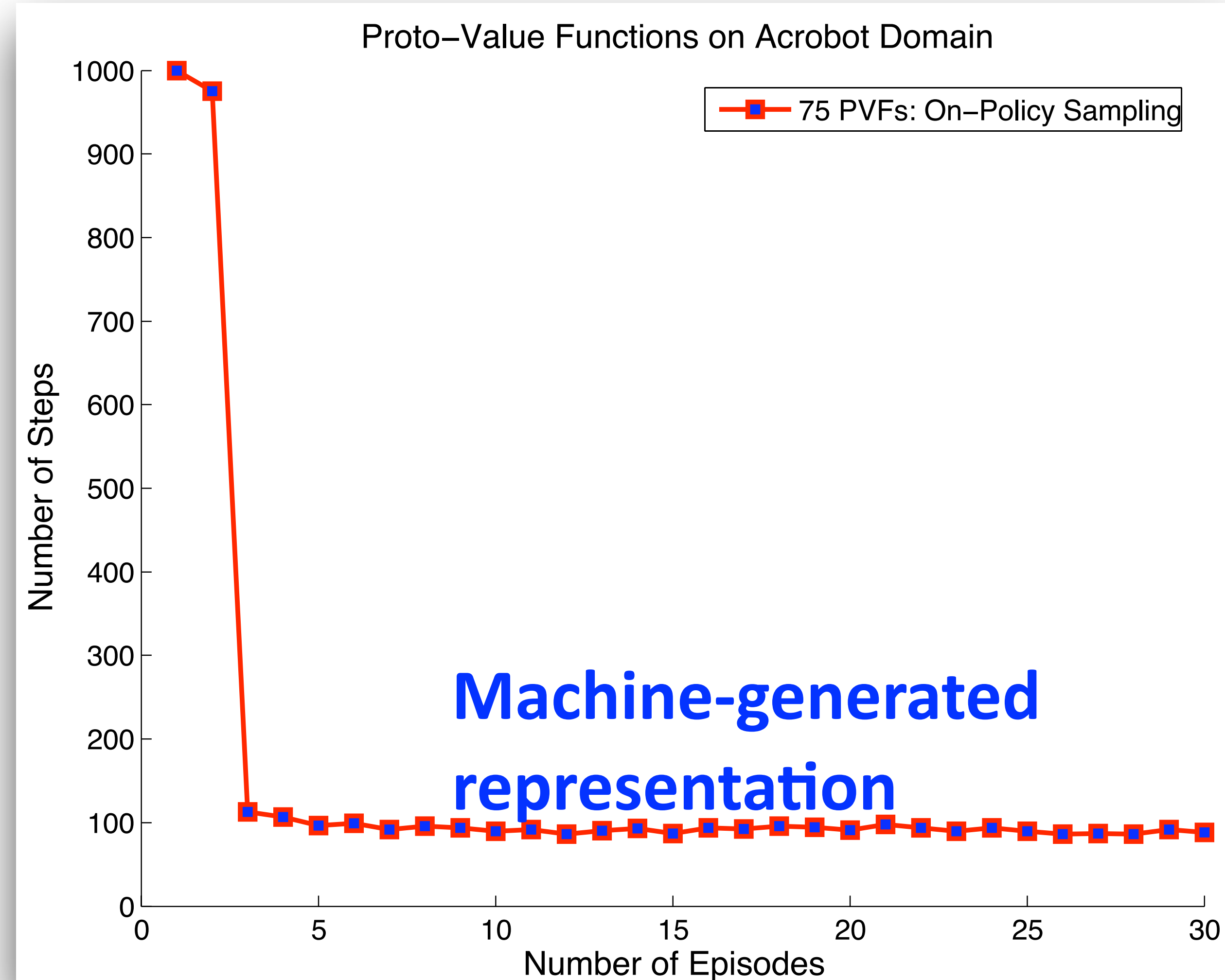






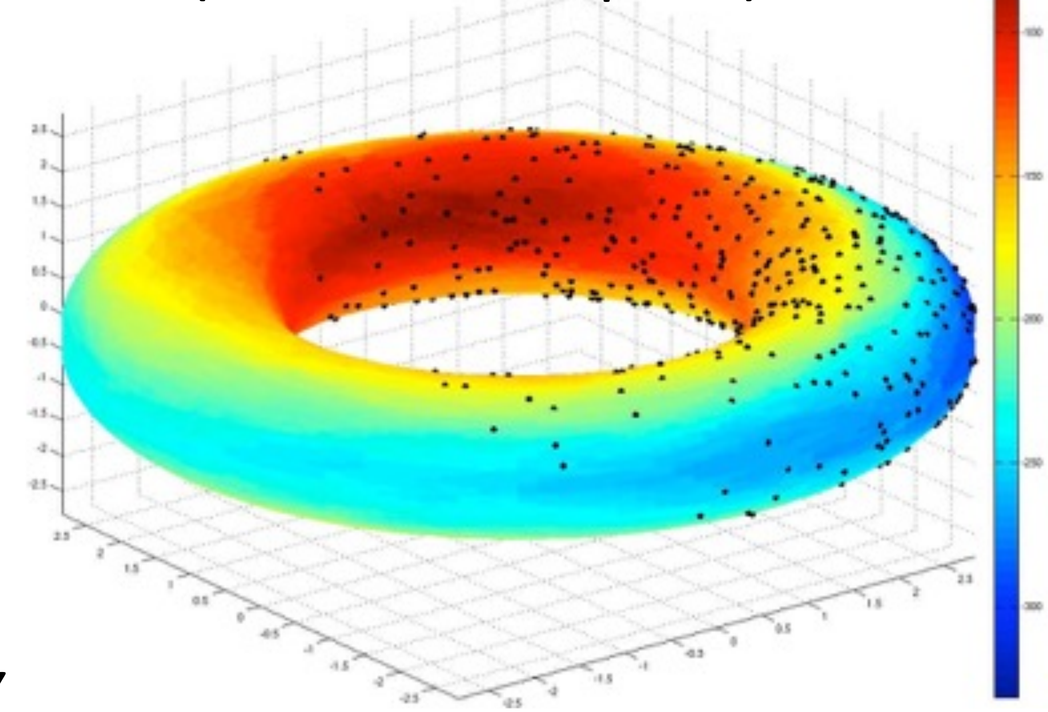
[Mahadevan et al., AAAI 2006; Mahadevan and Maggioni, JMLR 2007]

Continuous MDPs: Acrobot Task

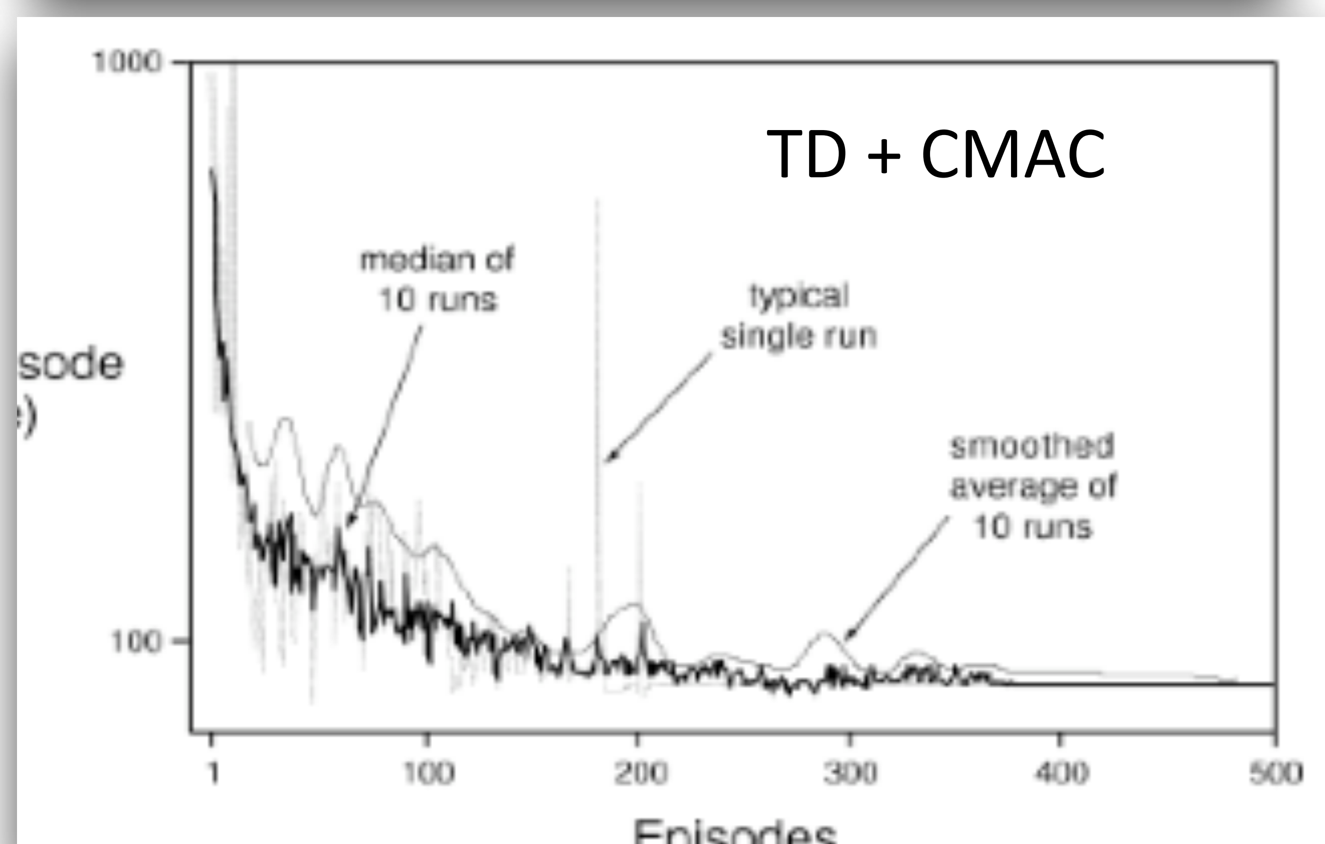


Machine-generated representation

(4-dim state space)



40X faster



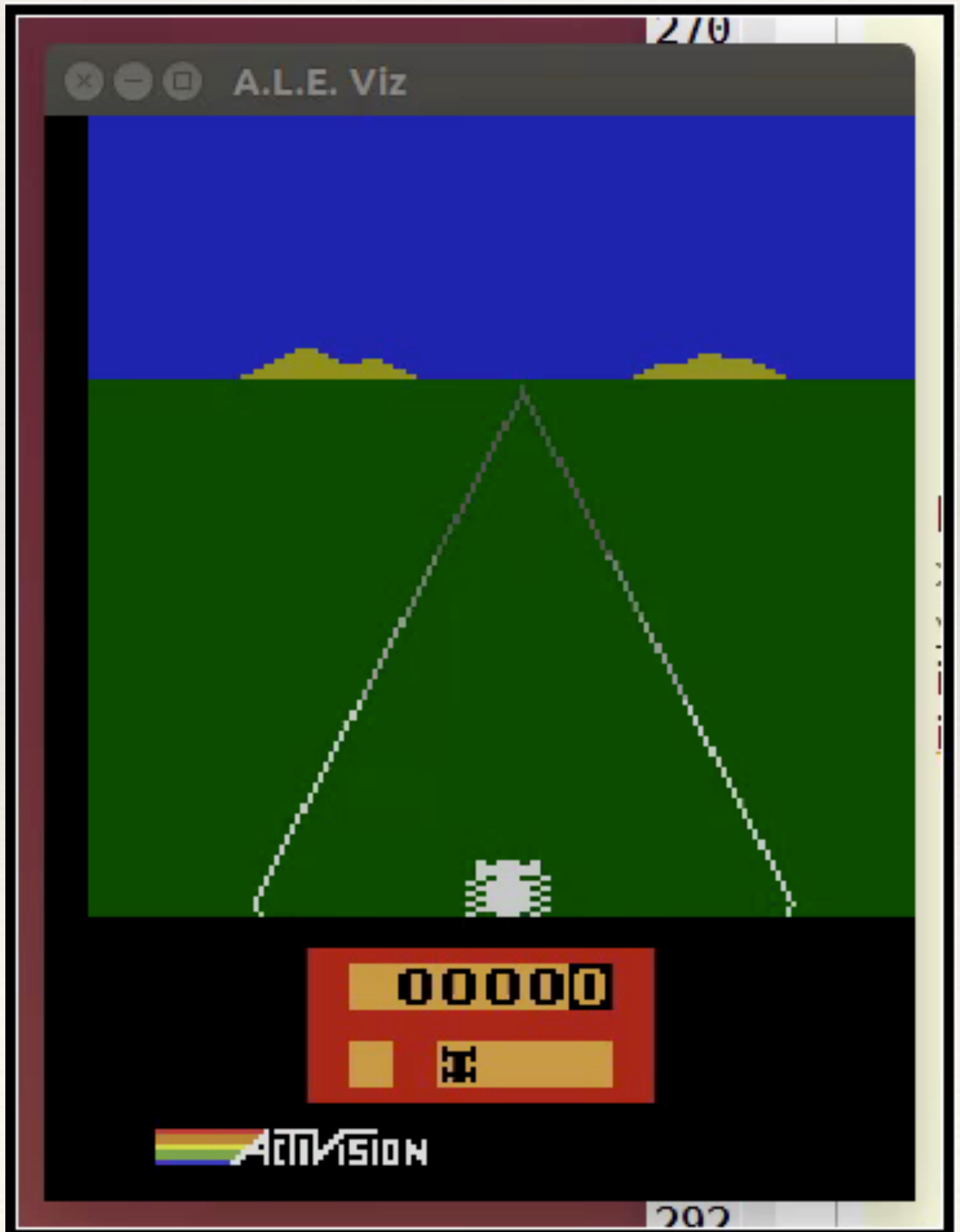
Human-designed representation

Reinforcement Learning for Atari

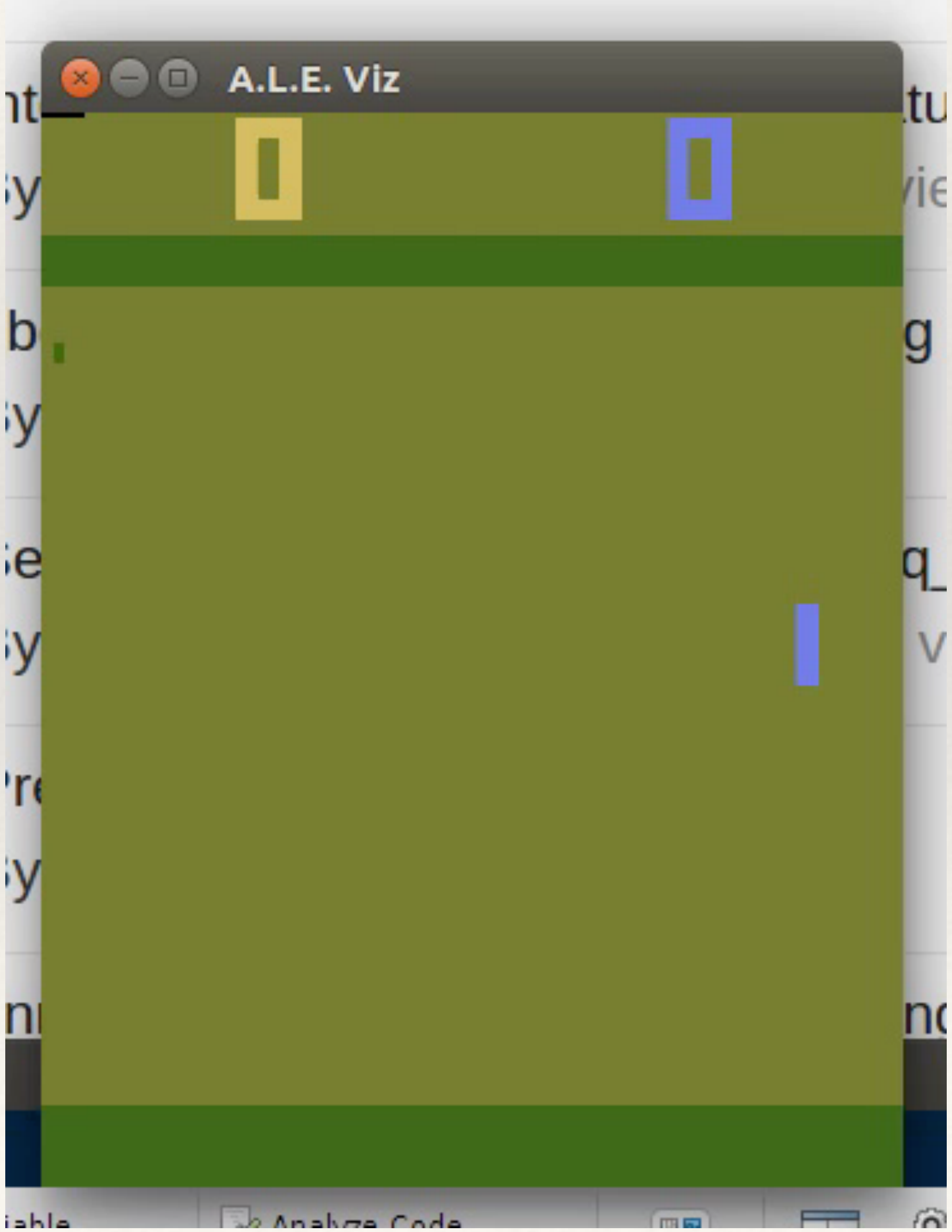
(Mnih et al., Nature 2015)

Representation Discovery by finding symmetries using convolutional neural networks

Enduro

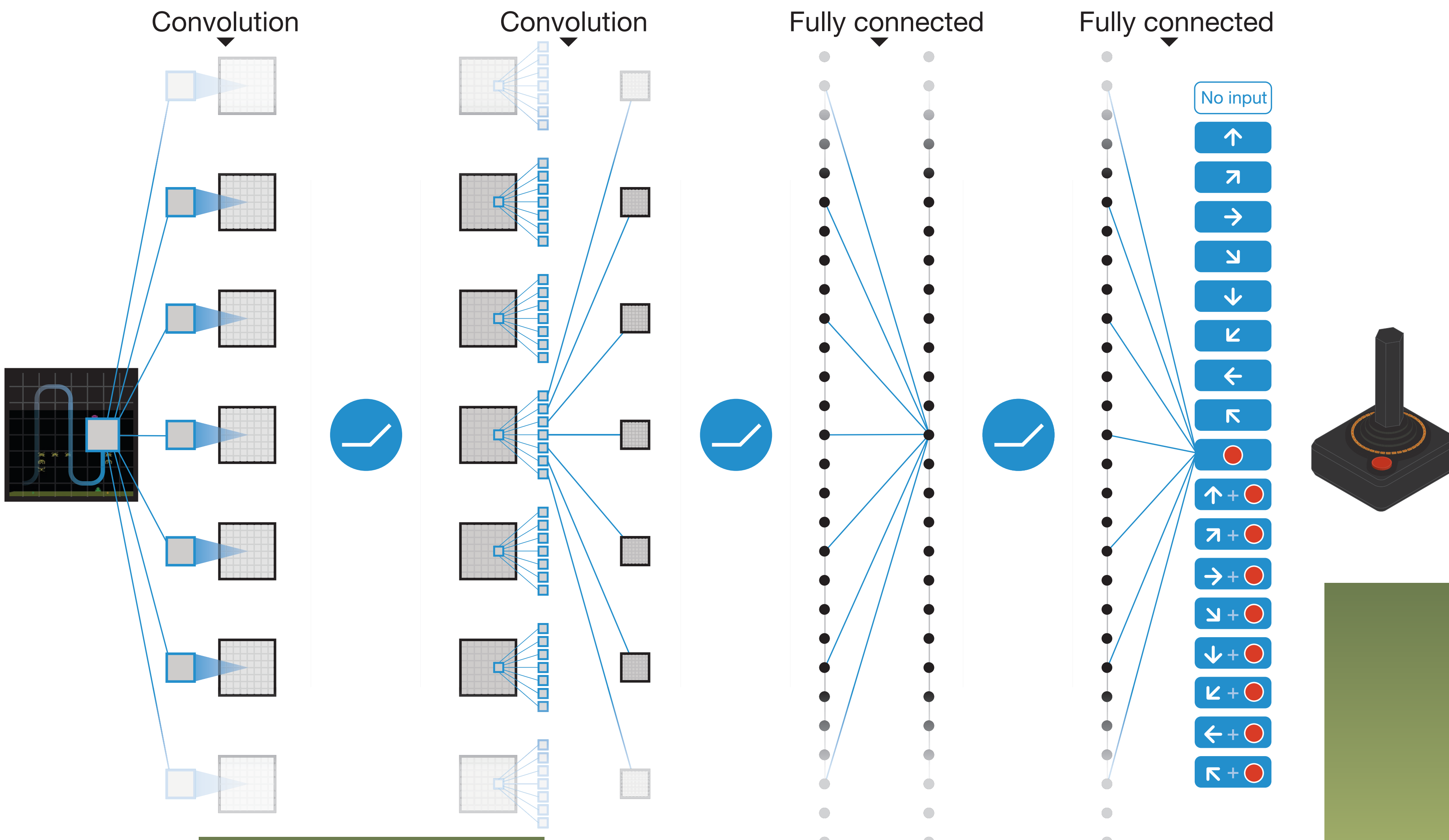


Pong



Atari Deep Learning Architecture

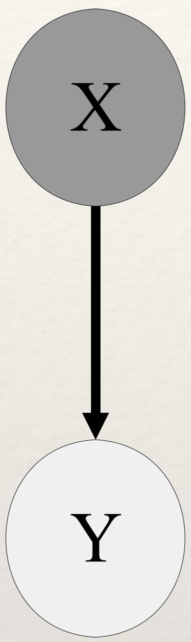
(Mnih et al., Nature 2015)

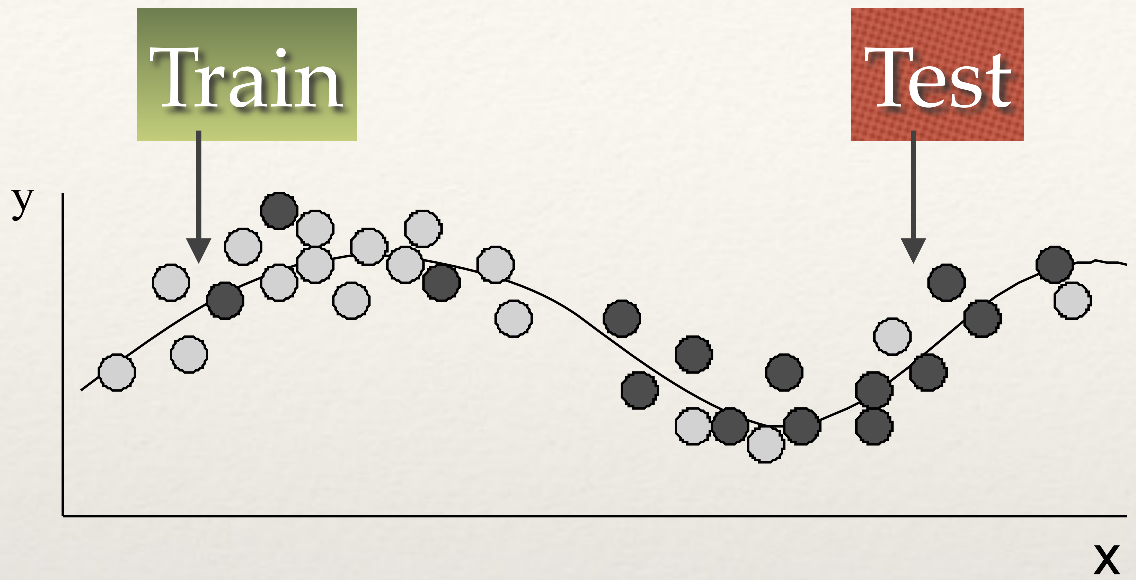


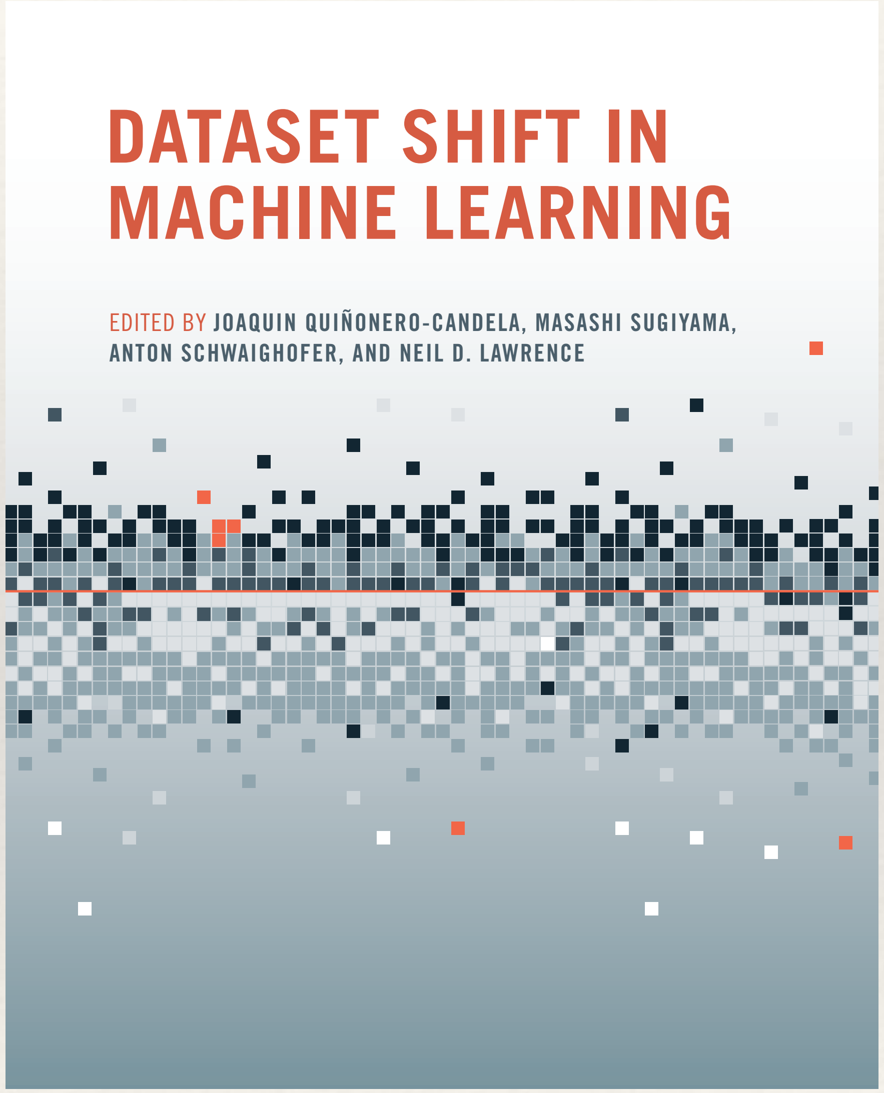
Symmetry detection CNNs

Actor-Mimic Architecture for Transfer in Deep RL (Parisoto et al., ICLR 2016)

Statistical Models of Domain Adaptation







Simple covariate shift is when only the distributions of covariates **x** change and everything else is the same.

Prior probability shift is when only the distribution over **y** changes and everything else stays the same.

Sample selection bias is when the distributions differ as a result of an unknown sample rejection process.

Imbalanced data is a form of deliberate dataset shift for computational or modeling convenience.

Domain shift involves changes in measurement.

Source component shift involves changes in strength of contributing components.

Simple Domain Adaptation Methods

The SRCONLY baseline ignores the target data and trains a single model, only on the source data.

The TGTONLY baseline trains a single model only on the target data.

The ALL baseline simply trains a standard learning algorithm on the union of the two datasets.

A potential problem with the ALL baseline is that if $N \gg M$, then D^s may "wash out" any affect D^t might have. We will discuss this problem in more detail later, but one potential solution is to re-weight examples from D^s. For instance, if $N = 10 \times M$, we may weight each example from the source domain by 0.1. The next baseline, WEIGHTED, is exactly this approach, with the weight chosen by cross-validation.

The PRED baseline is based on the idea of using the output of the source classifier as a feature in the target classifier. Specifically, we first train a SRCONLY model. Then we run the SRCONLY model on the target data (training, development and test). We use the predictions made by the SRCONLY model as additional features and train a second model on the target data, augmented with this new feature.

In the LININT baseline, we linearly interpolate the predictions of the SRCONLY and the TGTONLY models. The interpolation parameter is adjusted based on target development data.

(Daume' and Marcu, 2006)

Task	Dom	SRCONLY	TGTONLY	ALL	WEIGHT	PRED	LININT	PRIOR	AUGMENT	T<S	Win
ACE-NER	bn	4.98	2.37	2.29	2.23	2.11	2.21	2.06	**1.98**	+	+
	bc	4.54	4.07	3.55	3.53	3.89	4.01	**3.47**	**3.47**	+	+
	nw	4.78	3.71	3.86	3.65	3.56	3.79	3.68	**3.39**	+	+
	wl	2.45	2.45	**2.12**	**2.12**	2.45	2.33	2.41	**2.12**	=	+
	un	3.67	2.46	2.48	2.40	2.18	2.10	2.03	**1.91**	+	+
	cts	2.08	0.46	0.40	0.40	0.46	0.44	**0.34**	**0.32**	+	+
CoNLL	tgt	2.49	2.95	1.80	**1.75**	2.13	**1.77**	1.89	**1.76**		+
PubMed	tgt	12.02	4.15	5.43	4.15	4.14	3.95	3.99	**3.61**	+	+
CNN	tgt	10.29	3.82	3.67	3.45	3.46	3.44	**3.35**	**3.37**	+	+
Treebank-Chunk	wsj	6.63	4.35	4.33	4.30	4.32	4.32	4.27	**4.11**	+	+
	swbd3	15.90	4.15	4.50	4.10	4.13	4.09	3.60	**3.51**	+	+
	br-cf	5.16	6.27	4.85	4.80	4.78	**4.72**	5.22	5.15		
	br-cg	4.32	5.36	**4.16**	**4.15**	4.27	4.30	4.25	4.90		
	br-ck	5.05	6.32	5.05	4.98	**5.01**	**5.05**	5.27	5.41		
	br-cl	5.66	6.60	5.42	**5.39**	**5.39**	5.53	5.99	5.73		
	br-cm	3.57	6.59	**3.14**	**3.11**	3.15	3.31	4.08	4.89		
	br-cn	4.60	5.56	4.27	4.22	**4.20**	**4.19**	4.48	4.42		
	br-cp	4.82	5.62	4.63	**4.57**	**4.55**	**4.55**	4.87	4.78		
	br-cr	5.78	9.13	5.71	5.19	5.20	**5.15**	6.71	6.30		
Treebank-brown		6.35	5.75	4.80	4.75	4.81	4.72	4.72	**4.65**	+	+

Mathematical Model of Sample Selection Bias

DEFINITION 2.1. *Let* $\mathcal{F}$ *be a class of functions* $f: \mathcal{X} \to \mathbb{R}$. *Let* p *and* q *be Borel probability distributions, and let* $X = (x_1, \ldots, x_m)$ *and* $Y = (y_1, \ldots, y_n)$ *be samples composed of independent and identically distributed observations drawn from* p *and* q*, respectively. We define the maximum mean discrepancy (MMD) and its empirical estimate as*

$$\mathrm{MMD}[\mathcal{F}, p, q] := \sup_{f \in \mathrm{F}} \left(\mathbf{E}_p[f(x)] - \mathbf{E}_q[f(y)] \right)$$

$$\mathrm{MMD}[\mathcal{F}, X, Y] := \sup_{f \in \mathrm{F}} \left(\frac{1}{m} \sum_{i=1}^{m} f(x_i) - \frac{1}{n} \sum_{i=1}^{n} f(y_i) \right)$$

(Borgwardt et al., Bioinformatics 2006)

Kernel Version of MMD

[Baktashmotlagh et al., ICCV 2013]

Let $\tilde{\boldsymbol{X}}_{\boldsymbol{s}} = \{\tilde{\boldsymbol{x}}_s^1, \cdots, \tilde{\boldsymbol{x}}_s^n\}$ and $\tilde{\boldsymbol{X}}_{\boldsymbol{t}} = \{\tilde{\boldsymbol{x}}_t^1, \cdots, \tilde{\boldsymbol{x}}_t^m\}$ be two sets of observations drawn i.i.d. from s and t, respectively. An empirical estimate of the MMD can be computed as

$$D(\tilde{\boldsymbol{X}}_s, \tilde{\boldsymbol{X}}_t) = \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(\tilde{\boldsymbol{x}}_s^i) - \frac{1}{m} \sum_{j=1}^{m} \phi(\tilde{\boldsymbol{x}}_t^j) \right\|_{\mathcal{H}}$$

$$= \left(\sum_{i,j=1}^{n} \frac{k(\tilde{\boldsymbol{x}}_s^i, \tilde{\boldsymbol{x}}_s^j)}{n^2} + \sum_{i,j=1}^{m} \frac{k(\tilde{\boldsymbol{x}}_t^i, \tilde{\boldsymbol{x}}_t^j)}{m^2} - 2 \sum_{i,j=1}^{n,m} \frac{k(\tilde{\boldsymbol{x}}_s^i, \tilde{\boldsymbol{x}}_t^j)}{nm} \right)^{\frac{1}{2}},$$

where $\phi(\cdot)$ is the mapping to the RKHS $\mathcal{H}$, and $k(\cdot, \cdot) = \langle \phi(\cdot), \phi(\cdot) \rangle$ is the universal kernel associated with this mapping. In short, the MMD between the distributions of two sets of observations is equivalent to the distance between the sample means in a high-dimensional feature space.

Kernel MMD on Orthogonal Subspaces

$$D(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) = \left\| \frac{1}{n}\sum_{i=1}^{n}\phi(\boldsymbol{W}^T\boldsymbol{x}_s^i) - \frac{1}{m}\sum_{j=1}^{m}\phi(\boldsymbol{W}^T\boldsymbol{x}_t^j) \right\|_{\mathcal{H}},$$

[Baktashmotlagh et al., ICCV 2013]

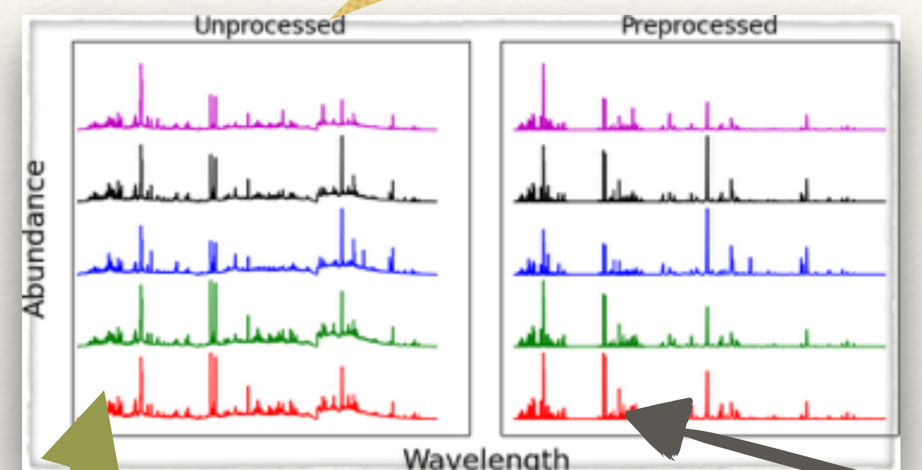
$$\begin{aligned} D^2(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) \; &= \\ &\frac{1}{n^2}\sum_{i,j=1}^{n}\exp\left(-\frac{(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)}{\sigma}\right) \\ &+\frac{1}{m^2}\sum_{i,j=1}^{m}\exp\left(-\frac{(\boldsymbol{x}_t^i - \boldsymbol{x}_t^j)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_t^i - \boldsymbol{x}_t^j)}{\sigma}\right) \\ &-\frac{2}{mn}\sum_{i,j=1}^{n,m}\exp\left(-\frac{(\boldsymbol{x}_s^i - \boldsymbol{x}_t^j)^T\boldsymbol{W}\boldsymbol{W}^T(\boldsymbol{x}_s^i - \boldsymbol{x}_t^j)}{\sigma}\right) \end{aligned}$$

Feature Construction Methods

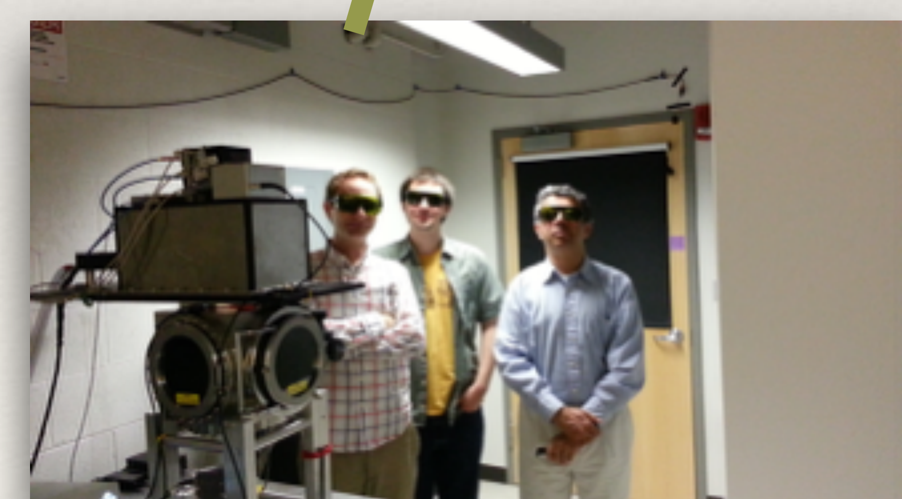
Single Subspace Methods

Map source
and target instances
to latent space

CCA,
manifold alignment



Curiosity zapping a
rock with a laser

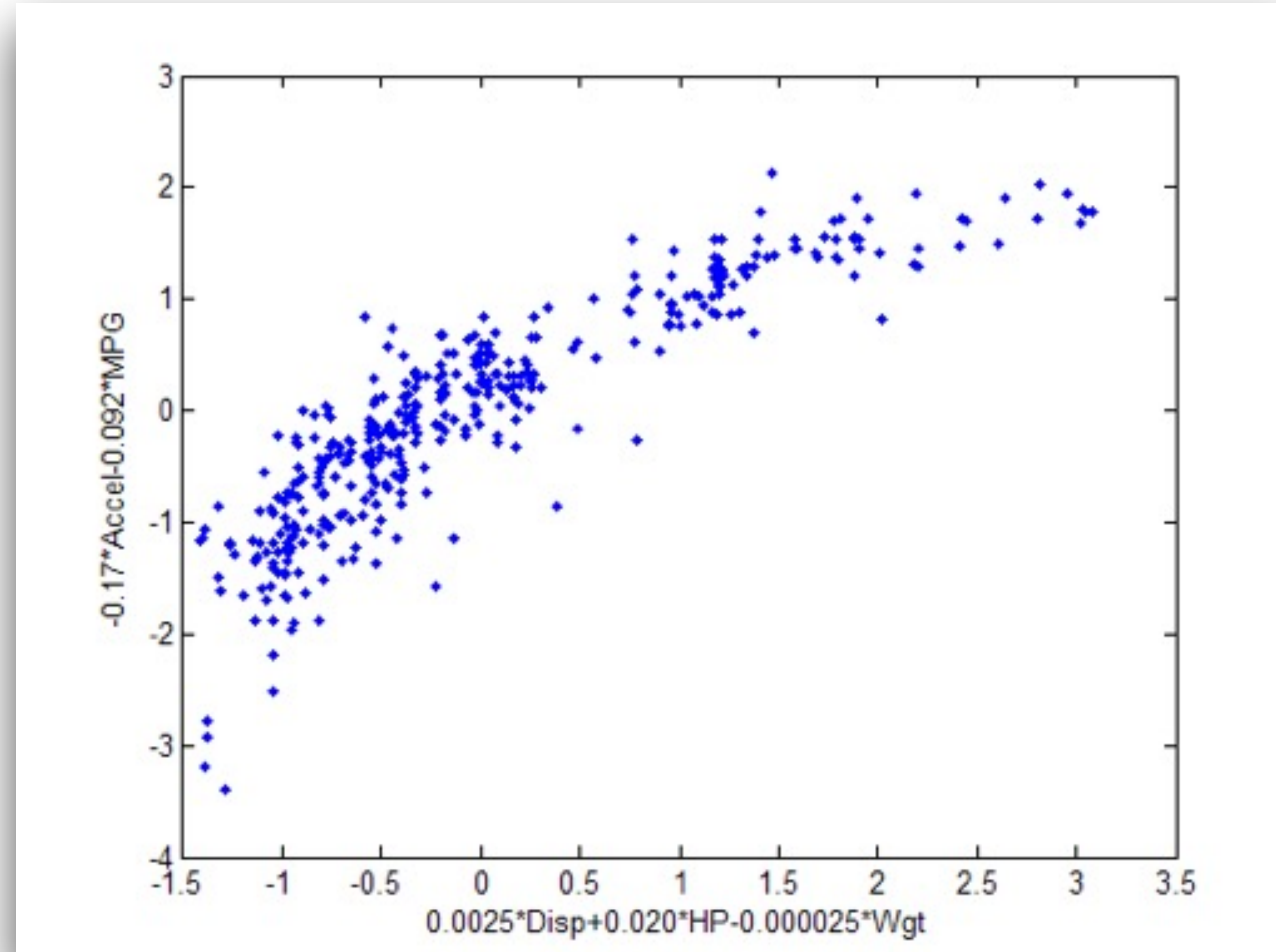


Same laser
on Earth
as on Mars

Canonical Correlational Analysis

(Hotelling, 1936)

Acceleration MPG



Displacement Horsepower Weight

Pioneer of the statistics departments in the US!
UNC, Chapel Hill
Columbia University

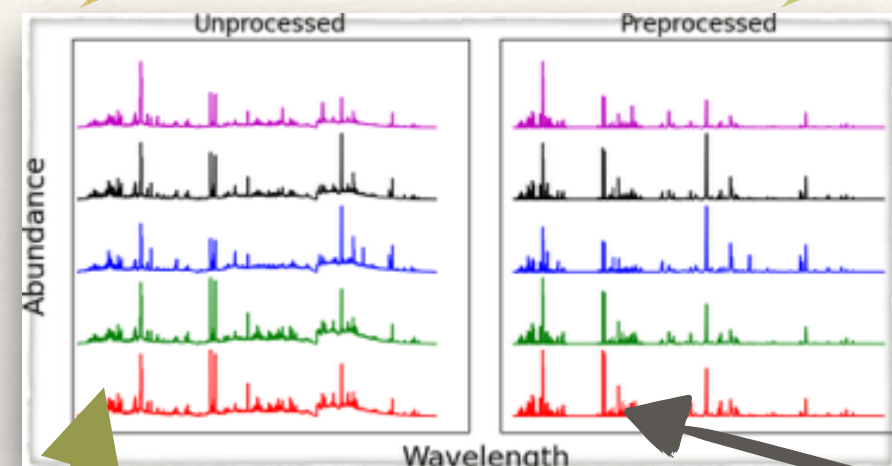
Find u,v that maximizes $$\frac{u^T X^T Y v}{\sqrt{u^T X^T X u}\sqrt{v^T Y^T Y v}}$$

Dual Subspace Methods

Source subspace

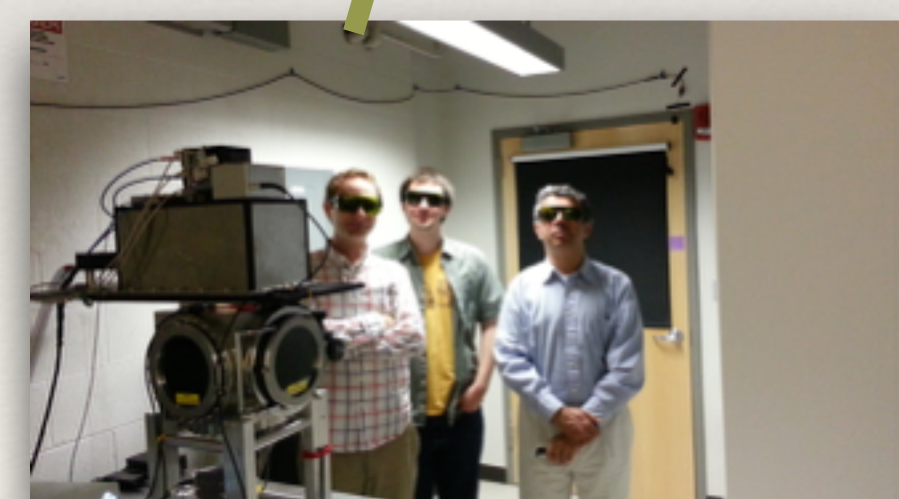
Target subspace

subspace alignment,
geodesic flow
kernels

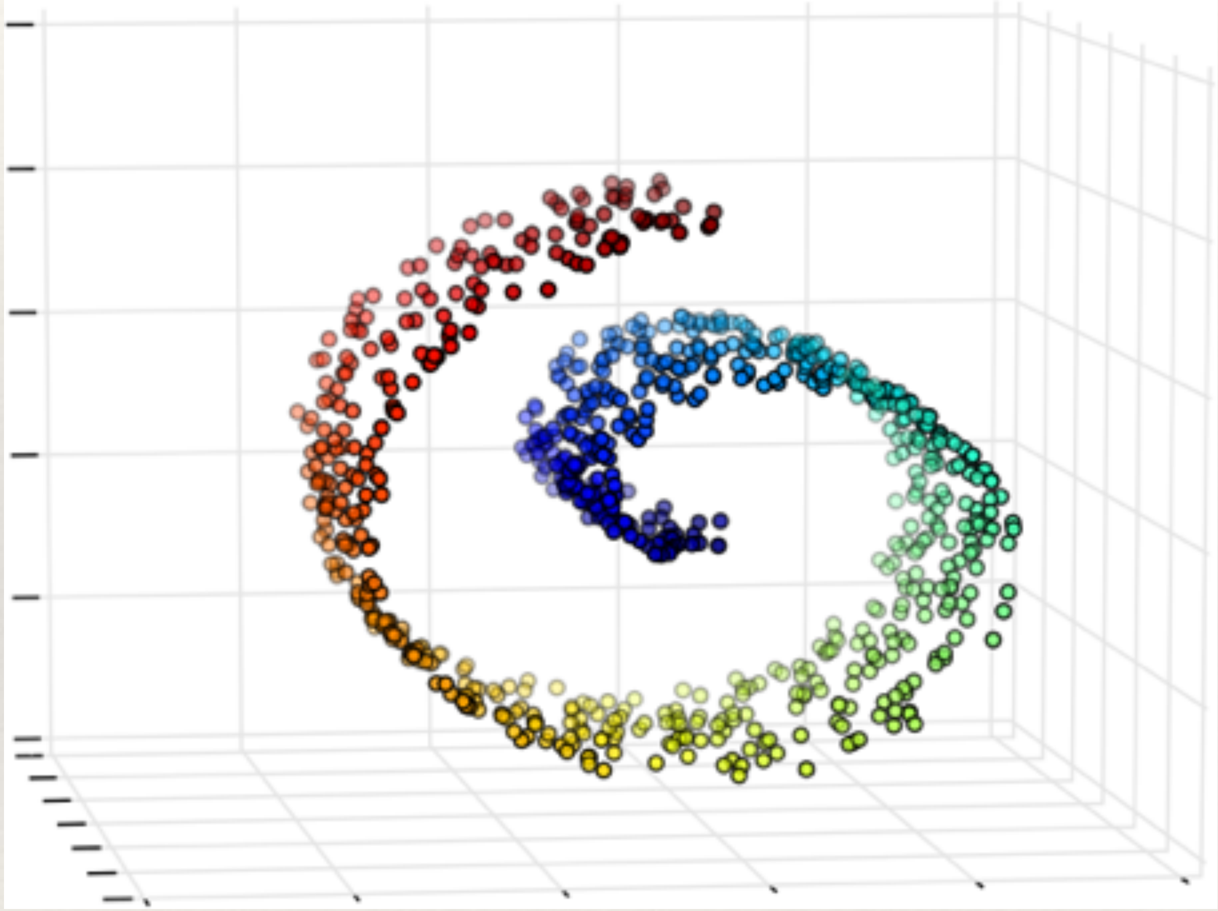


Curiosity zapping a rock with a laser

Same laser on Earth as on Mars



Manifold Learning



LLE, ISOMAP
Laplacian Eigenmaps

A Summary of Manifold Alignment Approaches

	Given correspondences	***Given labels***	***Unsupervised alignment***
Preserve Local geometry	Procrustes alignment, Manifold Projections (MP)	Extensions of MP	Extensions of MP
Preserve Global geometry	Procrustes alignment, Extensions of MP		
One-step alignment	Manifold Projections (MP)	Extensions of MP	Extensions of MP
Two-step alignment	Procrustes alignment		
Feature-level	Procrustes alignment, Manifold Projections (MP)	Extensions of MP	Extensions of MP
Instance-level	Procrustes alignment, Manifold Projections (MP)	Extensions of MP	Extensions of MP

Procrustes alignment *Manifold Projections (MP)* *Extensions of MP*

- Chang Wang, Peter Krafft, and Sridhar Mahadevan, “ Manifold Alignment ", appearing in Manifold Learning: Theory and Applications, Taylor and Francis CRC Press, 2012.

Mathematical Notation

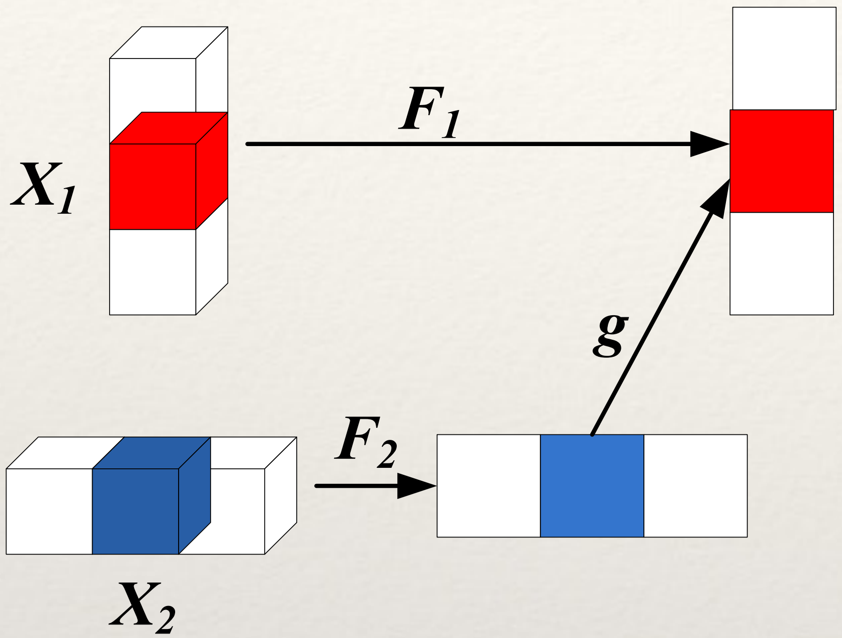
D_x is a diagonal matrix: $D_x^{ii} = \sum_j W_x^{ij}$.
$L_x = D_x - W_x$.
D_y is a diagonal matrix: $D_y^{ii} = \sum_j W_y^{ij}$.
$L_y = D_y - W_y$.
Ω_1 is an $m \times m$ diagonal matrix, and $\Omega_1^{ii} = \sum_j W^{i,j}$.
Ω_2 is an $m \times n$ matrix, and $\Omega_2^{i,j} = W^{i,j}$.
Ω_3 is an $n \times m$ matrix, and $\Omega_3^{i,j} = W^{j,i}$.
Ω_4 is an $n \times n$ diagonal matrix, and $\Omega_4^{ii} = \sum_j W^{j,i}$.

$$Z = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}.$$

$$D = \begin{pmatrix} D_x & 0 \\ 0 & D_y \end{pmatrix}.$$

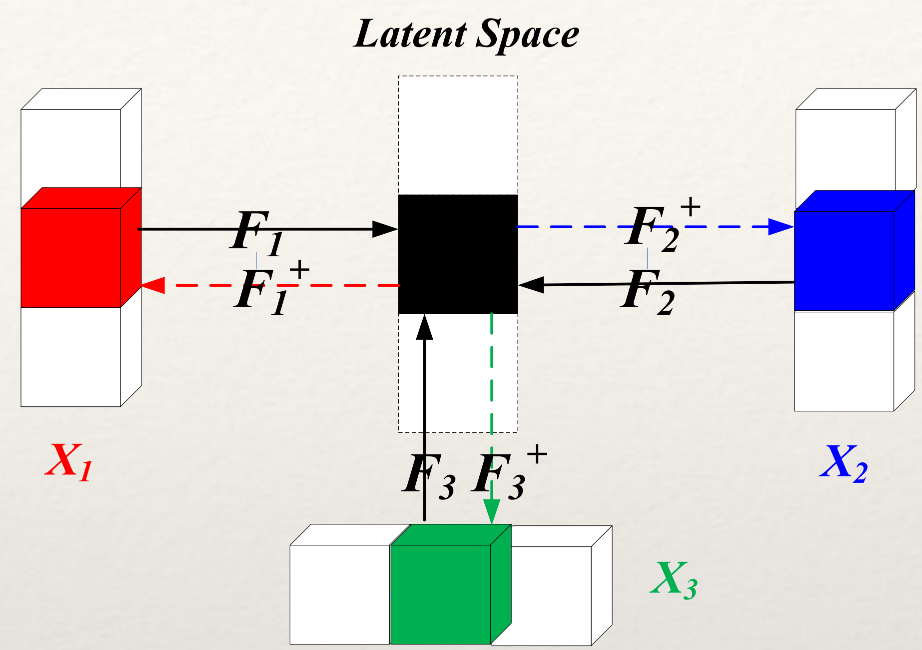
$$L = \begin{pmatrix} L_x + \mu\Omega_1 & -\mu\Omega_2 \\ -\mu\Omega_3 & L_y + \mu\Omega_4 \end{pmatrix}.$$

Manifold Alignment



Two-step alignment

Example: Procrustes alignment

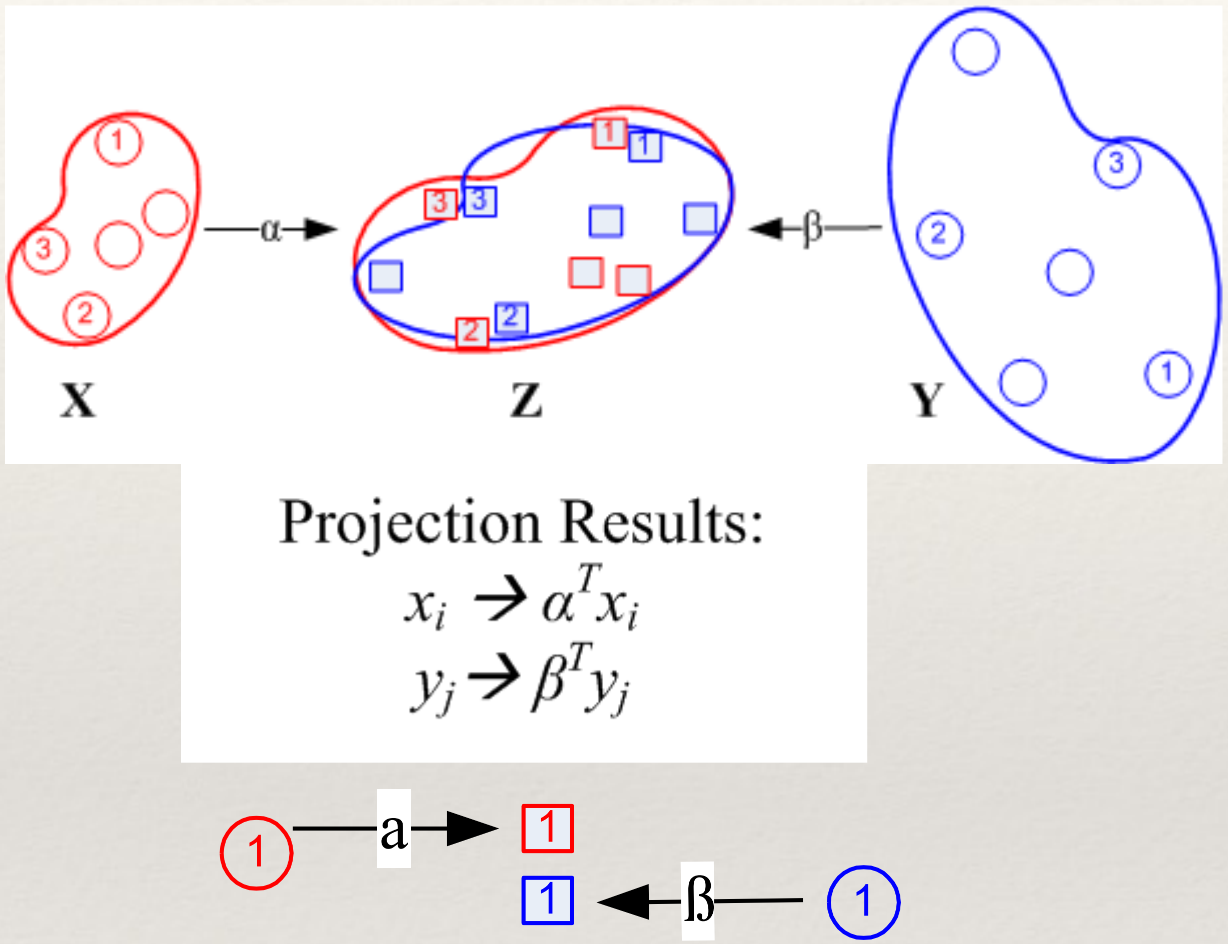


One-step alignment

Example: Manifold Projections

- Chang Wang, Peter Krafft, and Sridhar Mahadevan, " Manifold Alignment ", in Manifold Learning: Theory and Applications, Taylor and Francis CRC Press, 2012.

Feature-Level Manifold Projection

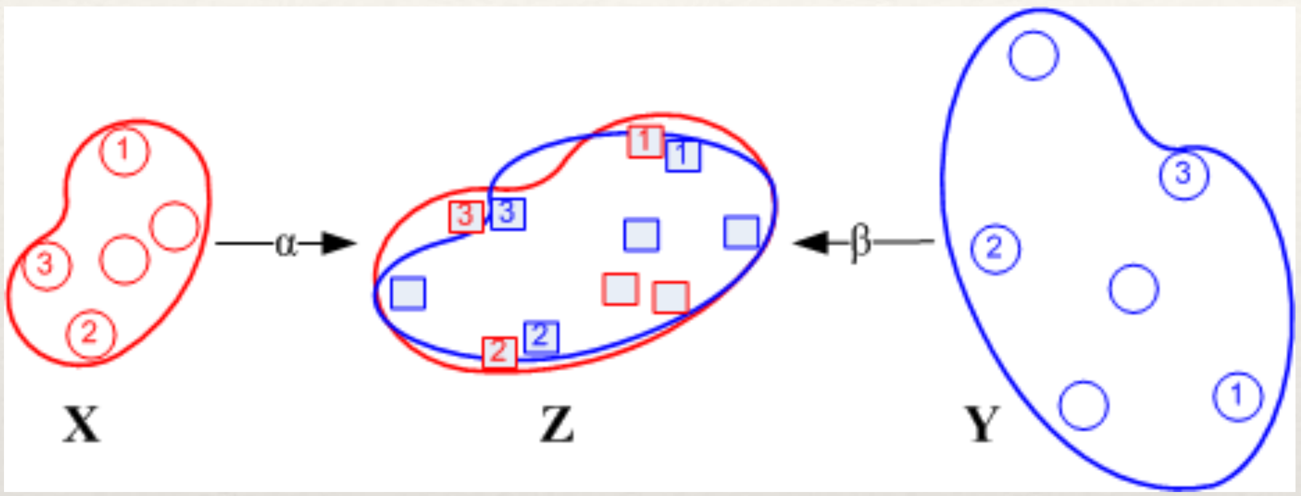


$X = [x_1, \ldots, x_m], x_i \in R^p.$

$Y = [y_1, \ldots, y_n], y_j \in R^q$

$x_i \leftrightarrow y_i$ for $i \in [1, l]$

Manifold Projection

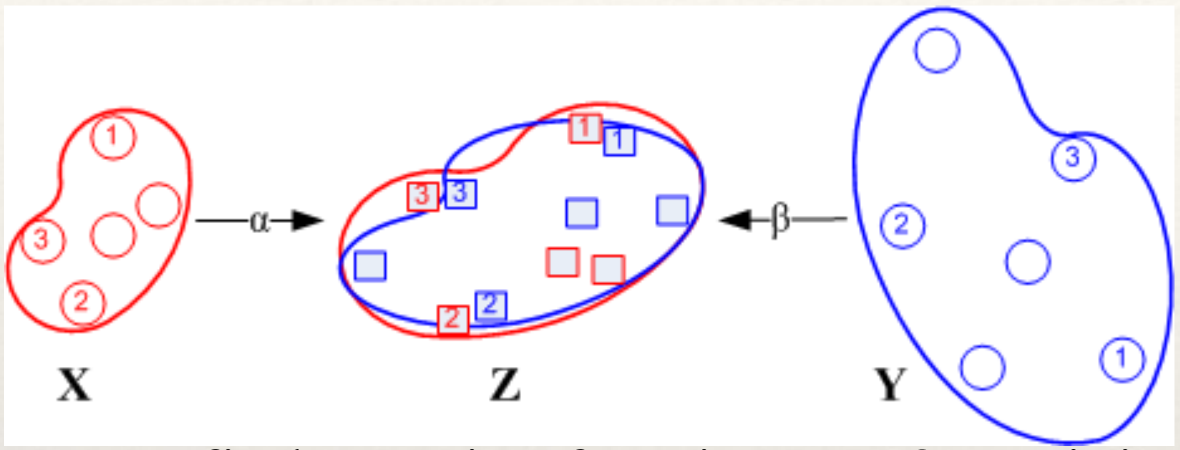


$X = [x_1, \ldots, x_m], x_i \in R^p.$

$Y = [y_1, \ldots, y_n], y_j \in R^q$

$x_i \leftrightarrow y_i$ for $i \in [1, l]$

Manifold Projection



$X = [x_1,\ldots,x_m], x_i \in R^p.$

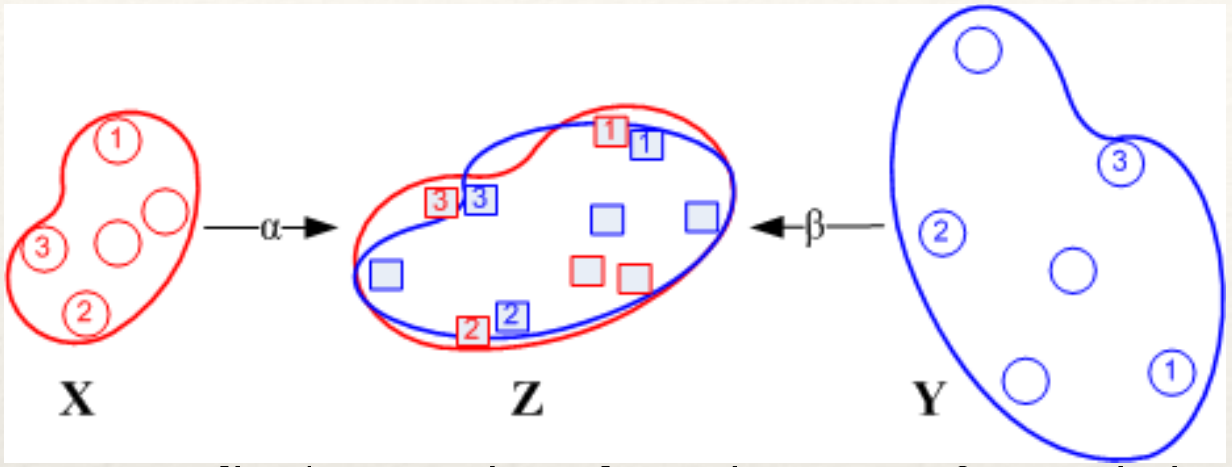
$Y = [y_1,\ldots,y_n], y_j \in R^q$

$x_i \leftrightarrow y_i$ for $i \in [1,l]$

We want to find mapping functions α, β to minimize the cost function $C(\alpha,\beta)$, where

$$C(\alpha,\beta) = \mu \sum_i \sum_j (\alpha^T x_i - \beta^T y_j)^2 W^{i,j} + 0.5 \sum_{i,j} (\alpha^T x_i - \alpha^T x_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (\beta^T y_i - \beta^T y_j)^2 W_y^{i,j}$$

Manifold Projection



$$X = [x_1, \dots, x_m], x_i \in R^p.$$
$$Y = [y_1, \dots, y_n], y_j \in R^q$$
$$x_i \leftrightarrow y_i \text{ for } i \in [1, l]$$

We want to find mapping functions α, β to minimize the cost function $C(\alpha, \beta)$, where

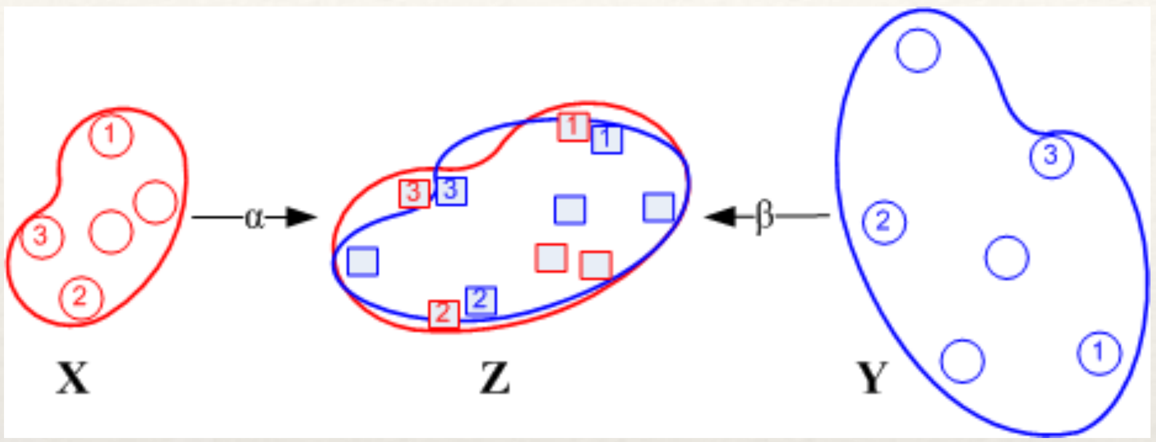
$$C(\alpha,\beta) = \mu \sum_i \sum_j (\alpha^T x_i - \beta^T y_j)^2 W^{i,j} + 0.5 \sum_{i,j} (\alpha^T x_i - \alpha^T x_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (\beta^T y_i - \beta^T y_j)^2 W_y^{i,j}$$

The ***first*** term encourages the corresponding instances from different domains to be projected to similar locations.
$W^{i,j}$=1, when x_i and y_j are in correspondence; 0, otherwise.

$$W = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \dots & & & & & \\ & & & 1 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & \dots & \\ & & & & & & & 0 \end{bmatrix}$$

- ***When 1:1 correspondence is given*** *($x_i \leftarrow\rightarrow y_i$ for $i<=l$):*
- ***When many:many correspondence is given****, set corresponding entries to 1.*
- ***When nothing is given****, we can use local geometry information to fill in this matrix. (IJCAI 2009)*

Comparison with CCA



$X = [x_1, \ldots, x_m], x_i \in R^p.$

$Y = [y_1, \ldots, y_n], y_j \in R^q$

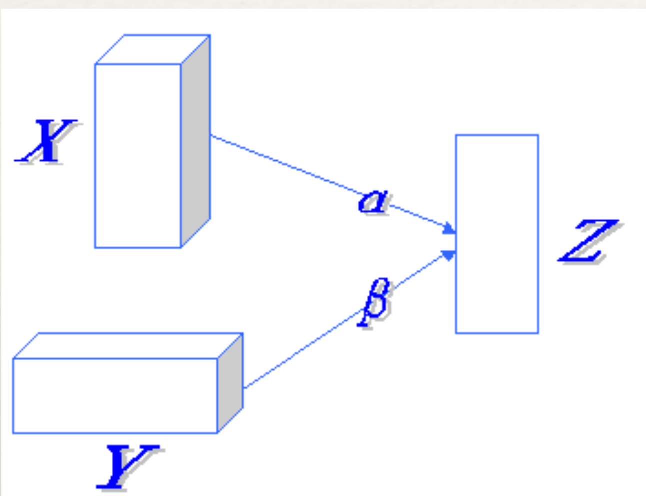
$x_i \leftrightarrow y_i$ for $i \in [1, l]$

We want to find mapping functions α, β to minimize the cost function $C(\alpha, \beta)$, where

$$C(\alpha,\beta) = \mu \sum_i \sum_j (\alpha^T x_i - \beta^T y_j)^2 W^{i,j} + 0.5 \sum_{i,j} (\alpha^T x_i - \alpha^T x_j)^2 W_x^{i,j} + 0.5 \sum_{i,j} (\beta^T y_i - \beta^T y_j)^2 W_y^{i,j}$$

How to compute projections?

Optimal Solution:



$[\alpha, \beta] = F(X, Y, W)$

↑ correspondence

(1) Construct *Z, L, D* using *X, Y* and *W* (the correspondences).

D_x is a diagonal matrix: $D_x^{ii} = \sum_j W_x^{ij}$.
$L_x = D_x - W_x$.
D_y is a diagonal matrix: $D_y^{ii} = \sum_j W_y^{ij}$.
$L_y = D_y - W_y$.
Ω_1 is an $m \times m$ diagonal matrix, and $\Omega_1^{ii} = \sum_j W^{i,j}$.
Ω_2 is an $m \times n$ matrix, and $\Omega_2^{i,j} = W^{i,j}$.
Ω_3 is an $n \times m$ matrix, and $\Omega_3^{i,j} = W^{j,i}$.
Ω_4 is an $n \times n$ diagonal matrix, and $\Omega_4^{ii} = \sum_j W^{j,i}$.

$$Z = \begin{pmatrix} X & 0 \\ 0 & Y \end{pmatrix}.$$
$$D = \begin{pmatrix} D_x & 0 \\ 0 & D_y \end{pmatrix}.$$
$$L = \begin{pmatrix} L_x + \mu\Omega_1 & -\mu\Omega_2 \\ -\mu\Omega_3 & L_y + \mu\Omega_4 \end{pmatrix}.$$

Create a joint domain.
(use correspondences to determine how to join them)

Project the joint domain to a lower dimensional space.

(2) Theorem 1 : α, β ***to minimize*** $C(\alpha, \beta)$ ***are given by the eigenvectors corresponding to the smallest eigenvalues of*** $ZLZ^T\gamma = \lambda ZDZ^T\gamma$.

(3) $\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = [\gamma_1, \ldots, \gamma_d]$, where γ_i is the i^{th} minimum eigenvector.

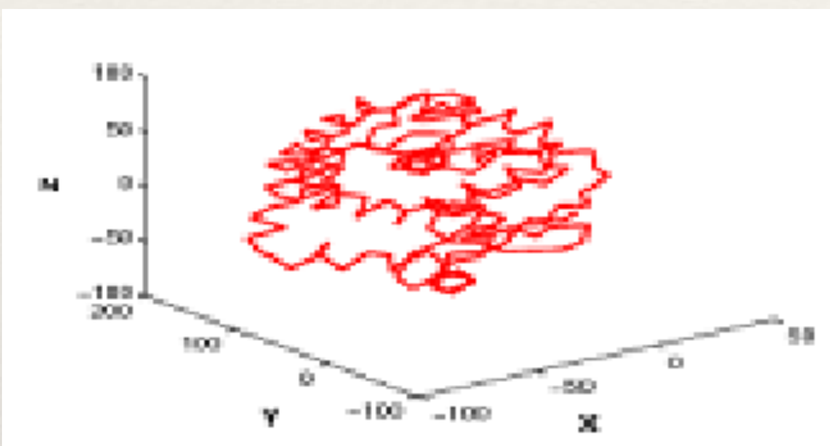
Protein Alignment

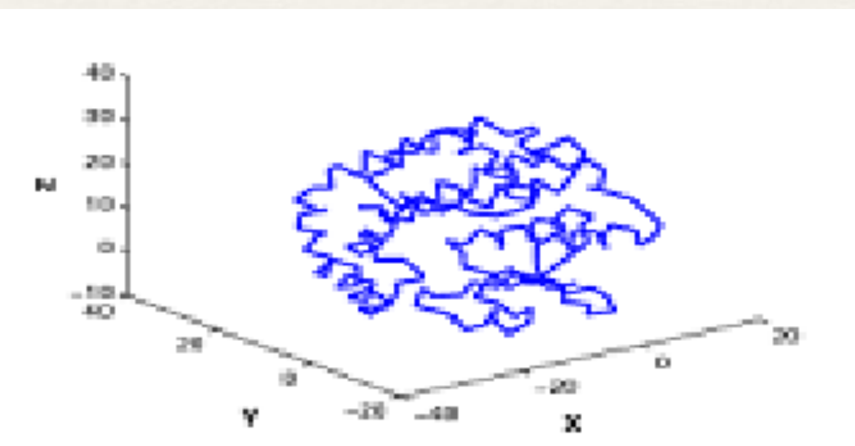
Two datasets:

X: 3*215 matrix

Y: 3*215 matrix

10% points are in correspondence





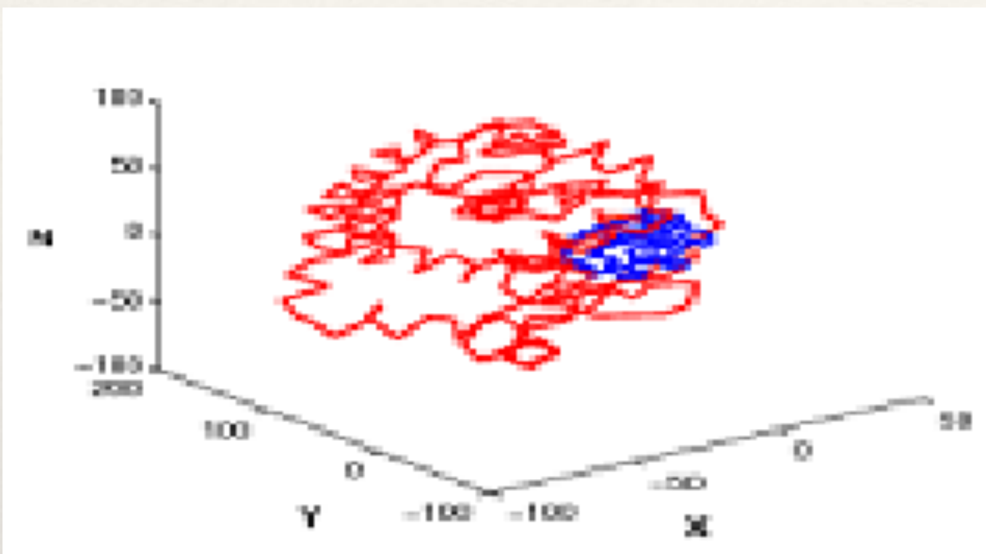
	1	2	3
1	4.388	-3.508	-10.572
2	2.472	8.376	-20.128
3	-6.944	20.36	-18.184
4	-11.984	26.472	-31.232
5	-16.216	40.492	-26.64
6	-18.468	55.324	-29.54
7	-9.84	61.756	-18.424
8	-20.412	71.732	-13.436
9	-31.58	61.296	-12.044
10	-36.328	60.128	2.656
11	-38.652	45.132	0.732
12	-24.248	43.664	-4.044
13	-19.648	51.504	8.288
14	-27.768	41.44	16.744
15	-20.788	29.8	9.572
16	-6.784	35.812	10.756

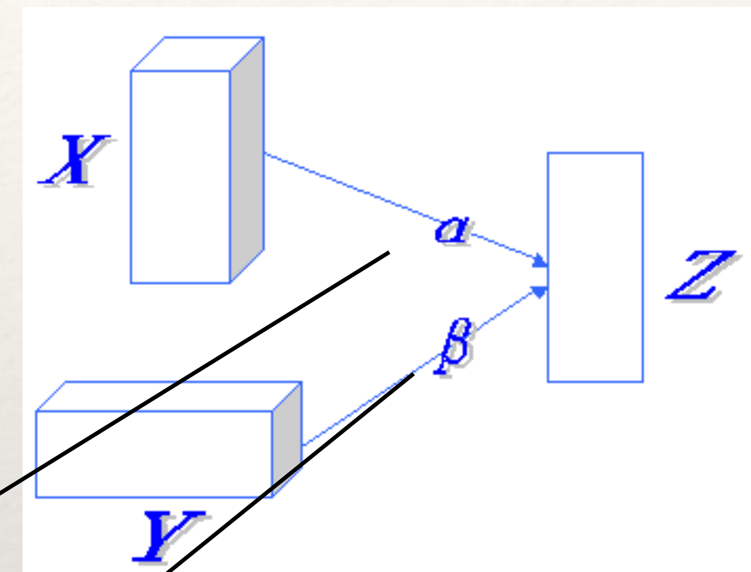
	1	2	3
1	2.756	-2.591	-3.275
2	-0.639	-0.921	-3.985
3	-3.214	0.065	-1.307
4	-6.82	0.065	-2.548
5	-8.344	2.16	0.275
6	-10.785	4.753	1.667
7	-8.798	7.635	3.284
8	-10.872	7.801	6.497
9	-11.08	4.019	7.073
10	-9.42	2.98	10.402
11	-8.042	-0.212	8.806
12	-6.674	1.858	5.9
13	-5.055	4.182	8.427
14	-3.503	1.162	10.245
15	-1.957	-0.001	6.95
16	-0.811	3.496	5.955

$$W = \begin{bmatrix} 1 & & & & & & & \\ & 1 & & & & & & \\ & & \cdots & & & & & \\ & & & 1 & & & & \\ & & & & 0 & & & \\ & & & & & 0 & & \\ & & & & & & \cdots & \\ & & & & & & & 0 \end{bmatrix}$$

Protein Alignment

X and Y



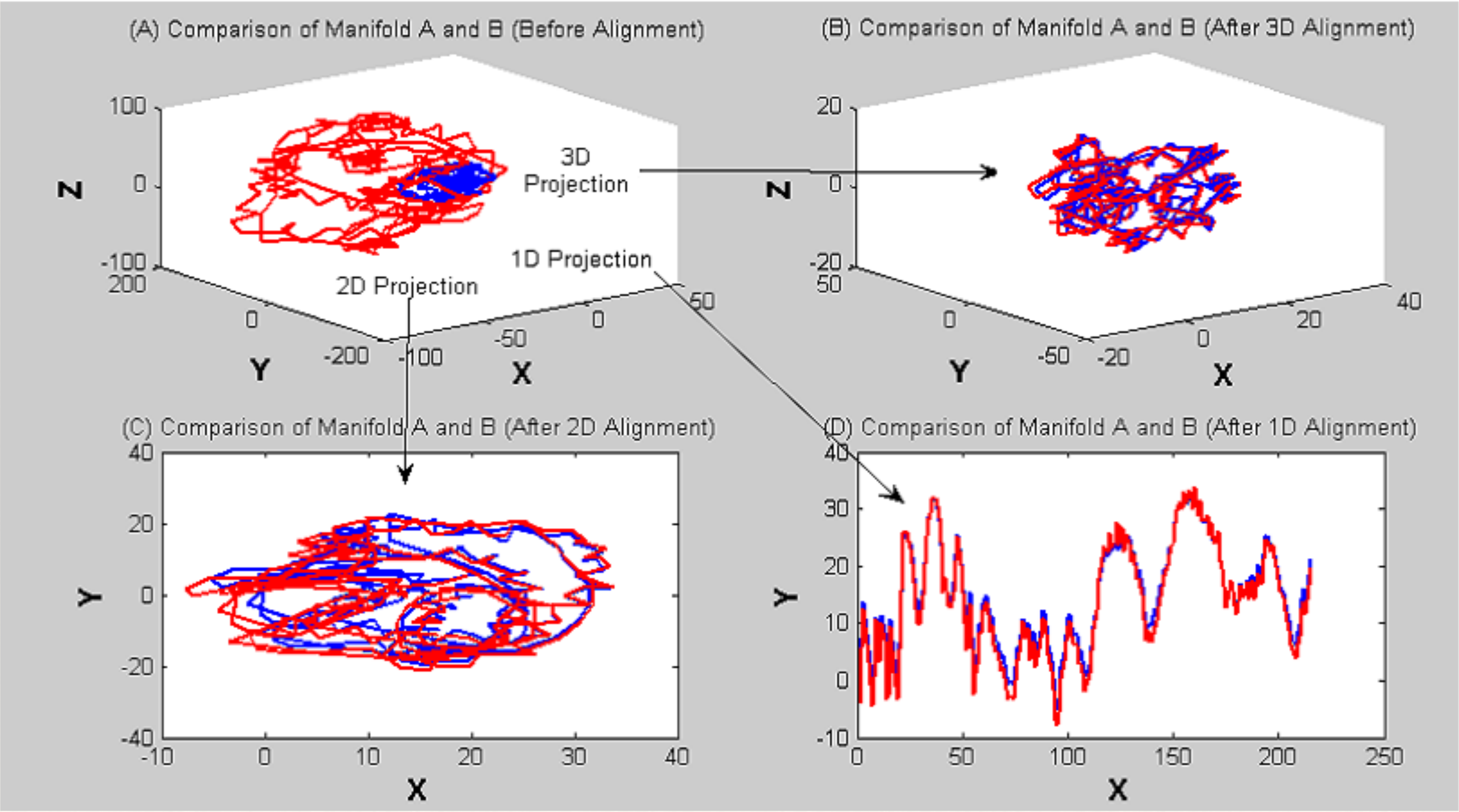


$$\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = [\gamma_1, \ldots, \gamma_s], \text{ where } \gamma_i \text{ is the } i^{th} \text{ minimum eigen solution to } ZLZ^T\gamma = \lambda ZDZ^T\gamma.$$

$$\alpha = \begin{pmatrix} -0.1589 & -0.0181 & -0.2178 \\ 0.1471 & 0.0398 & -0.1073 \\ 0.0398 & -0.2368 & -0.0126 \end{pmatrix},$$

$$\beta = \begin{pmatrix} -0.6555 & -0.7379 & -0.3007 \\ 0.0329 & 0.0011 & -0.8933 \\ 0.7216 & -0.6305 & 0.2289 \end{pmatrix}.$$

Protein Alignment



Reinforcement Learning Transfer using Manifold Alignment

(Ammar et al., AAAI 2015)

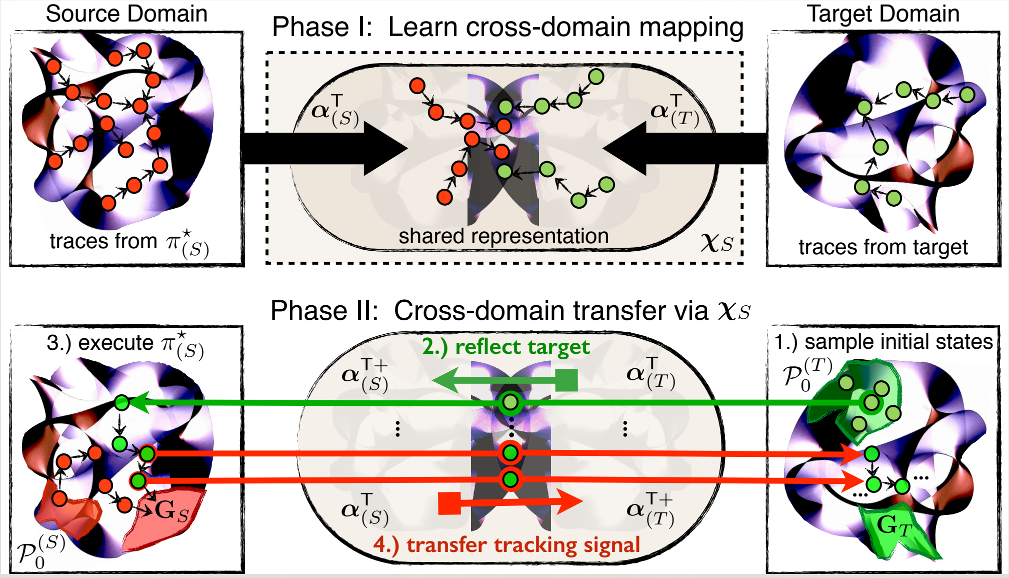


Figure 1: Transfer is split into two phases: (I) learning the inter-state mapping χ_S via manifold alignment, and (II) initializing the target policy via mapping the source task policy.

Algorithm 1 Manifold Alignment Cross-Domain Transfer for Policy Gradients (MAXDT-PG)

Inputs: Source and target tasks $\mathcal{T}^{(S)}$ and $\mathcal{T}^{(T)}$, optimal source policy $\pi^\star_{(S)}$, # source and target traces n_S and n_T, # nearest neighbors k, # target rollouts z_T, initial # of target states m.

Learn χ_S:

1: Sample n_S optimal source traces, $\boldsymbol{\tau}^\star_{(S)}$, and n_T random target traces, $\boldsymbol{\tau}_{(T)}$

2: Using the modified UMA approach, learn $\boldsymbol{\alpha}_{(S)}$ and $\boldsymbol{\alpha}_{(T)}$ to produce $\chi_S = \boldsymbol{\alpha}^{\mathsf{T}+}_{(T)} \boldsymbol{\alpha}^{\mathsf{T}}_{(S)}[\cdot]$

Transfer & Initialize Policy:

3: Collect m initial target states $\boldsymbol{s}^{(T)}_1 \sim \mathcal{P}^{(T)}_0$

4: Project these m states to the source by applying $\chi^+_S[\cdot]$

5: Apply the optimal source policy $\pi^\star_{(S)}$ on these projected states to collect $\mathcal{D}^{(S)} = \left\{ \boldsymbol{\tau}^{(S)}_{(i)} \right\}_{i=1}^{m}$

6: Project the samples in $\mathcal{D}^{(S)}$ to the target using $\chi_S[\cdot]$ to produce tracking target traces $\tilde{\mathcal{D}}^{(T)}$

7: Compute tracking rewards using Eqn. (9)

8: Use policy gradients to minimize Eqn. (8), yielding $\boldsymbol{\theta}^{(0)}_{(T)}$

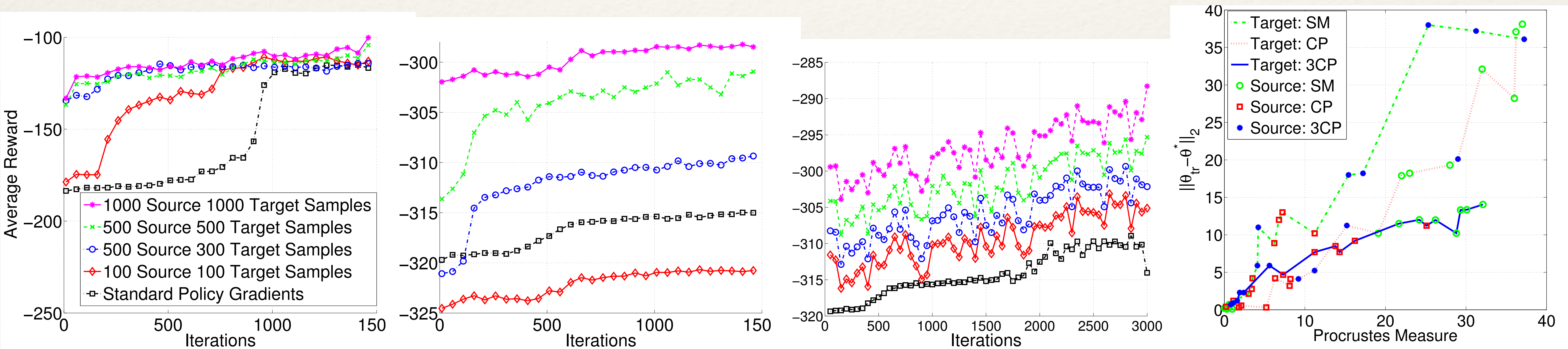
Improve Policy:

9: Start with $\boldsymbol{\theta}^{(0)}_{(T)}$ and sample z_T target rollouts

10: Follow policy gradients (e.g., episodic REINFORCE) but using target rewards $\mathcal{R}^{(T)}$

11: Return optimal target policy parameters $\boldsymbol{\theta}^\star_{(T)}$

Transfer in RL using Manifold Alignment

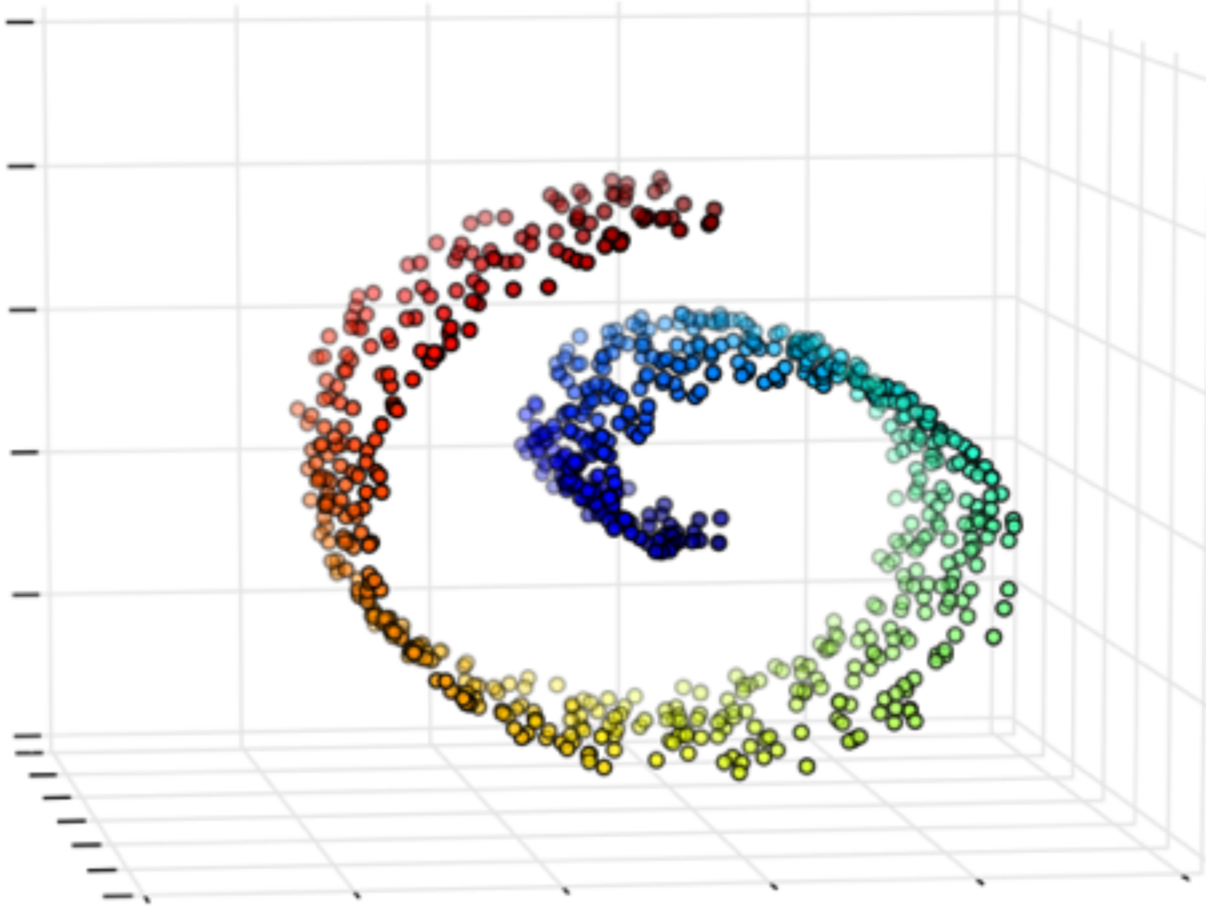


(a) Simple Mass to Cart Pole (b) Cart Pole to Three-Link CP (c) Cart Pole to Quadrotor (d) Alignment Quality vs Transfer

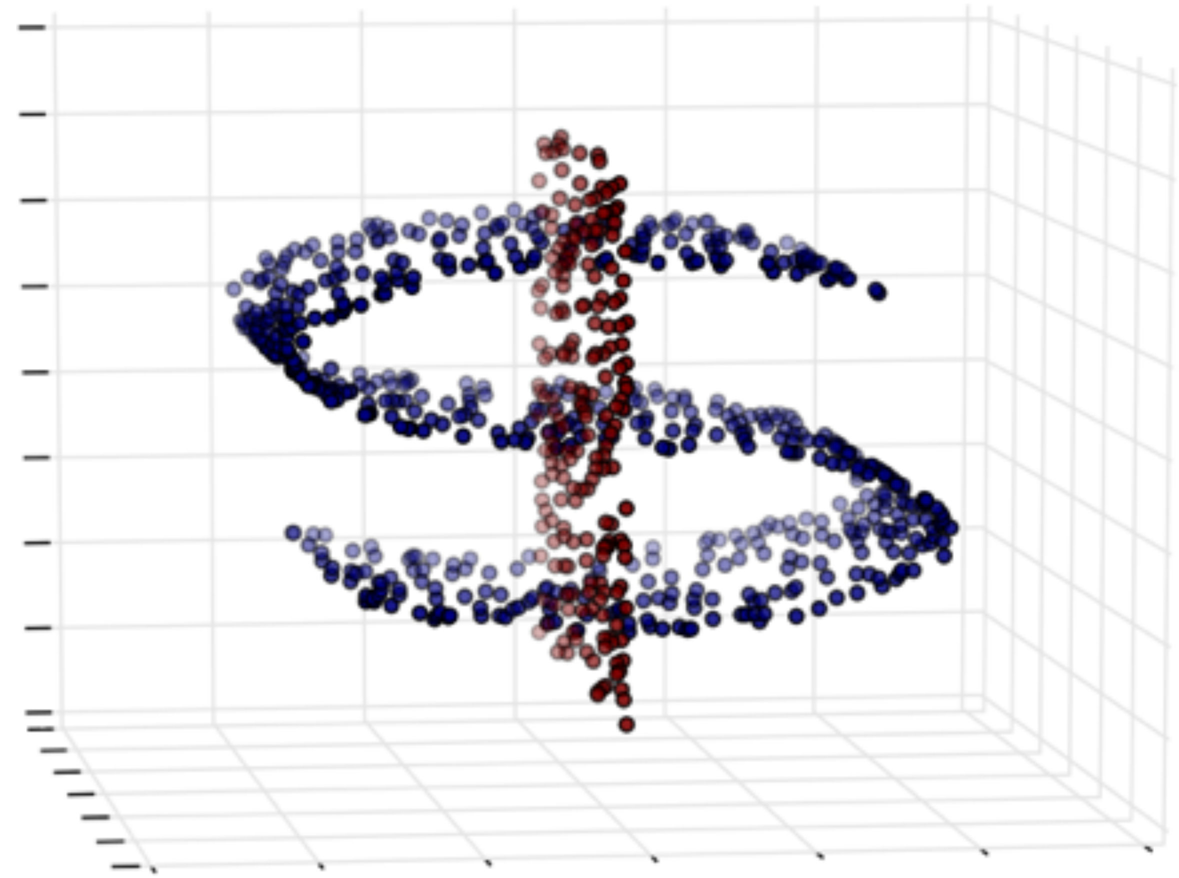
Transfer Learning from Mixture of Manifolds

(Boucher, Carey, Mahadevan, and Dyar, AAAI 2015)

Single manifold
(LLE, Laplacian Eigenmaps, Isomap)



Low-rank Alignment (LRA)



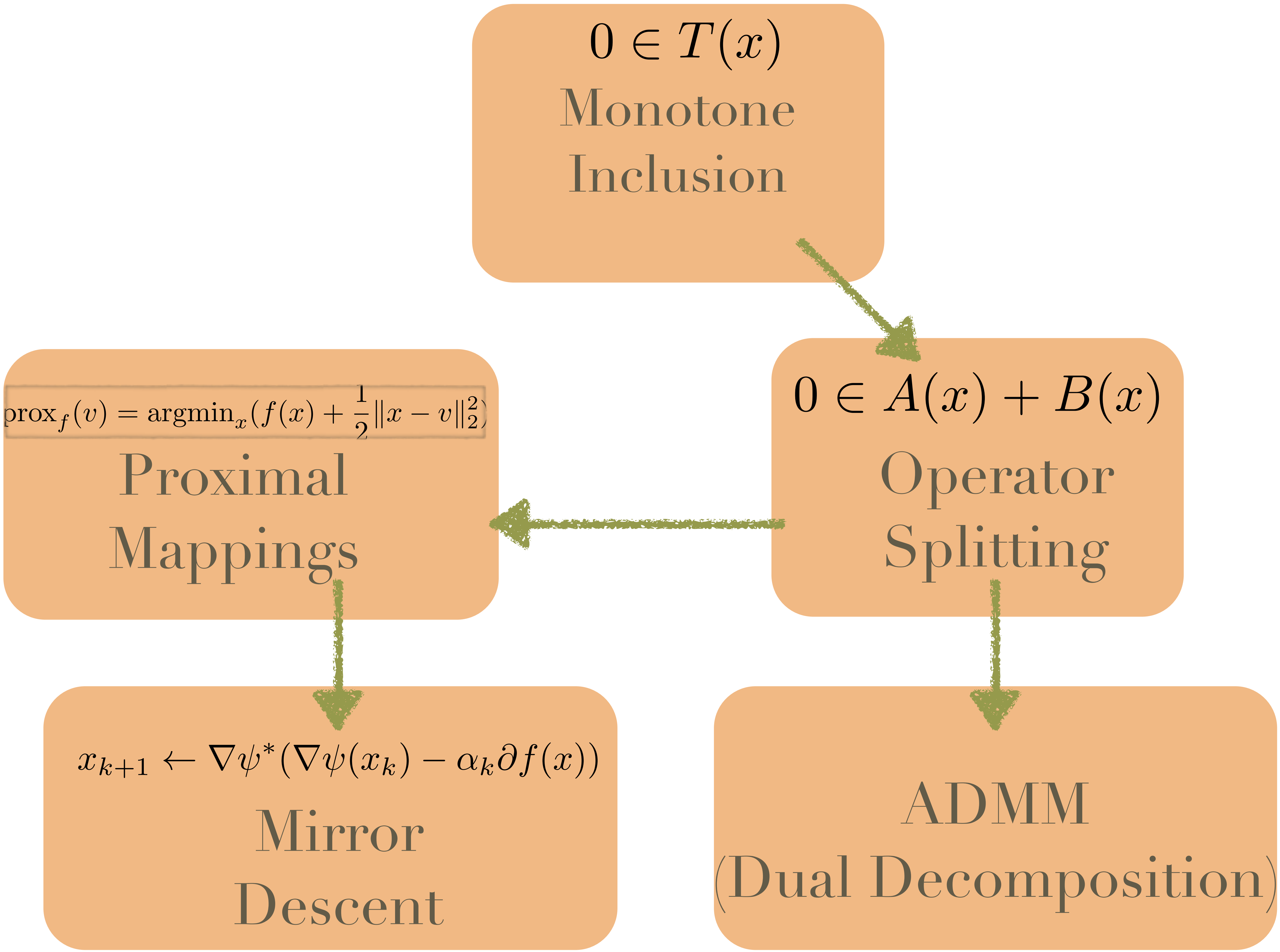
$$\min_R \frac{1}{2}||X - XR||_F^2 + \lambda||R||_*,$$

Multiple Objectives

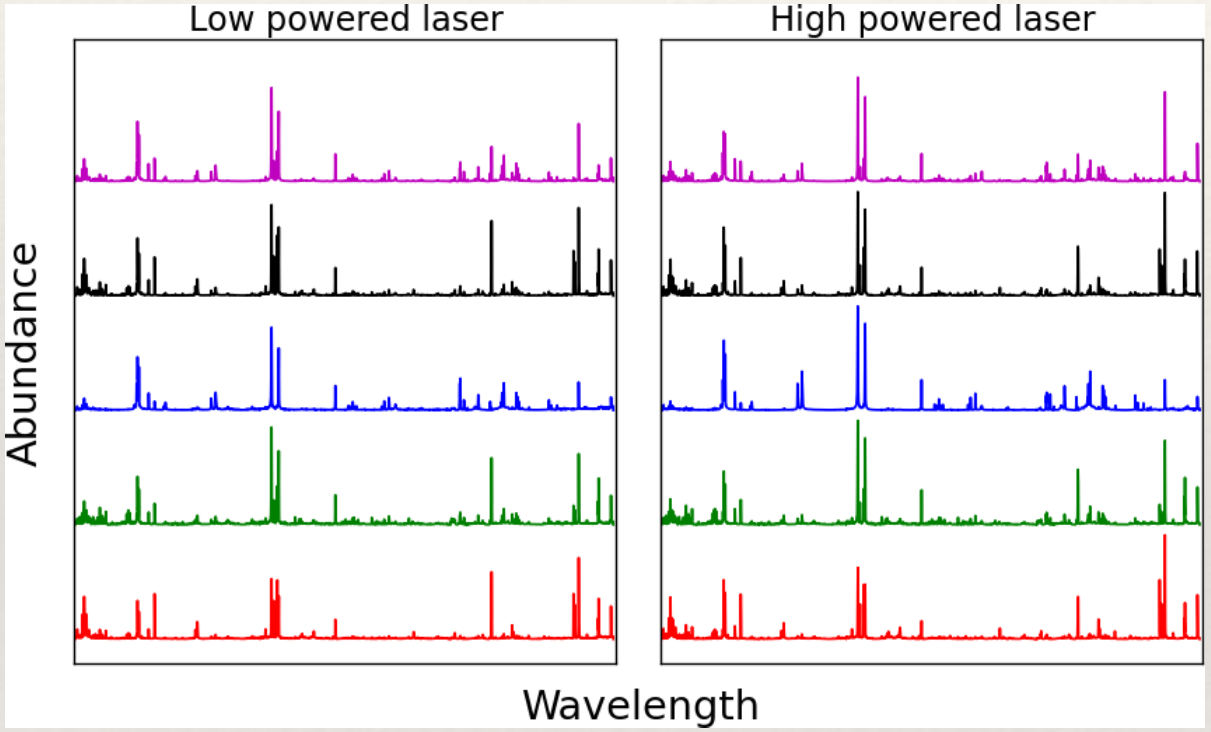
$$\min_R \frac{1}{2}||X - XR||_F^2 + \lambda||R||_*,$$

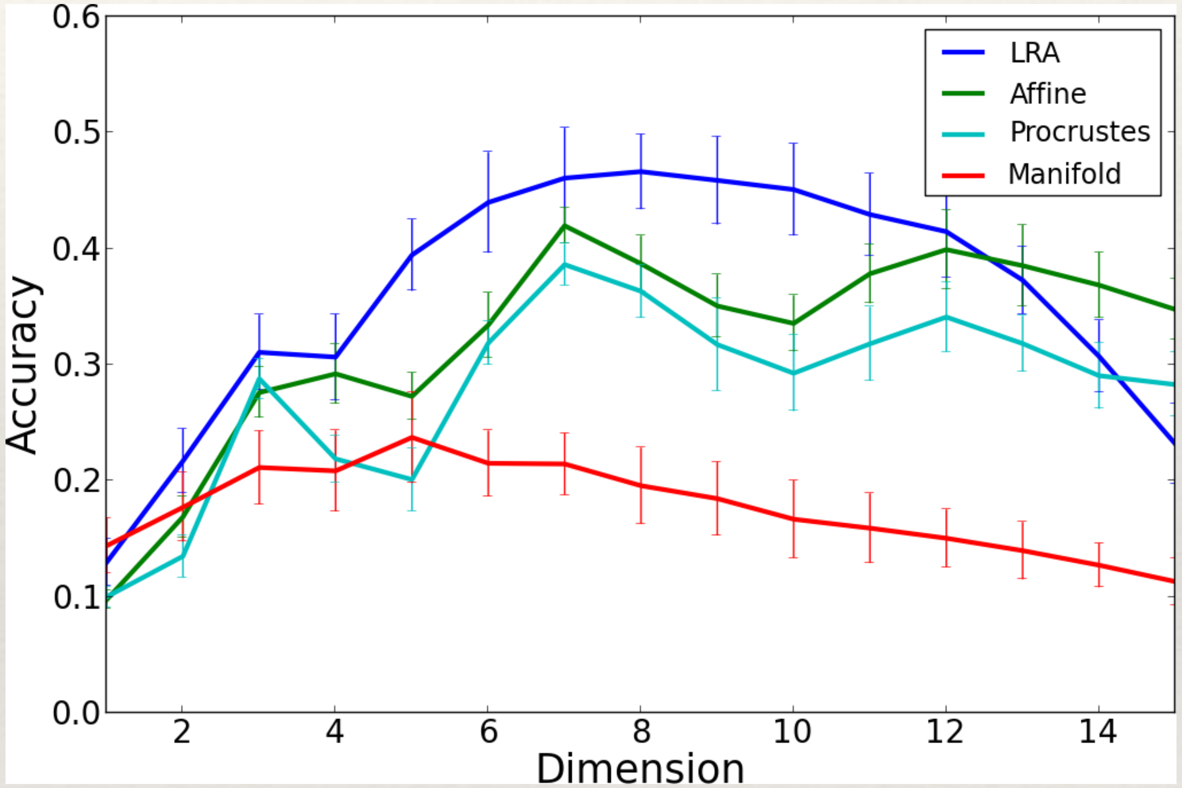
Minimize reconstruction error

Minimize model complexity


$0 \in T(x)$
Monotone Inclusion
$0 \in A(x) + B(x)$
Operator Splitting
$\mathrm{prox}_f(v) = \mathrm{argmin}_x(f(x) + \frac{1}{2}\|x - v\|_2^2)$
Proximal Mappings
$x_{k+1} \leftarrow \nabla\psi^*(\nabla\psi(x_k) - \alpha_k \partial f(x))$
Mirror Descent
ADMM
(Dual Decomposition)

MARS Alignment





Cross-Language IR

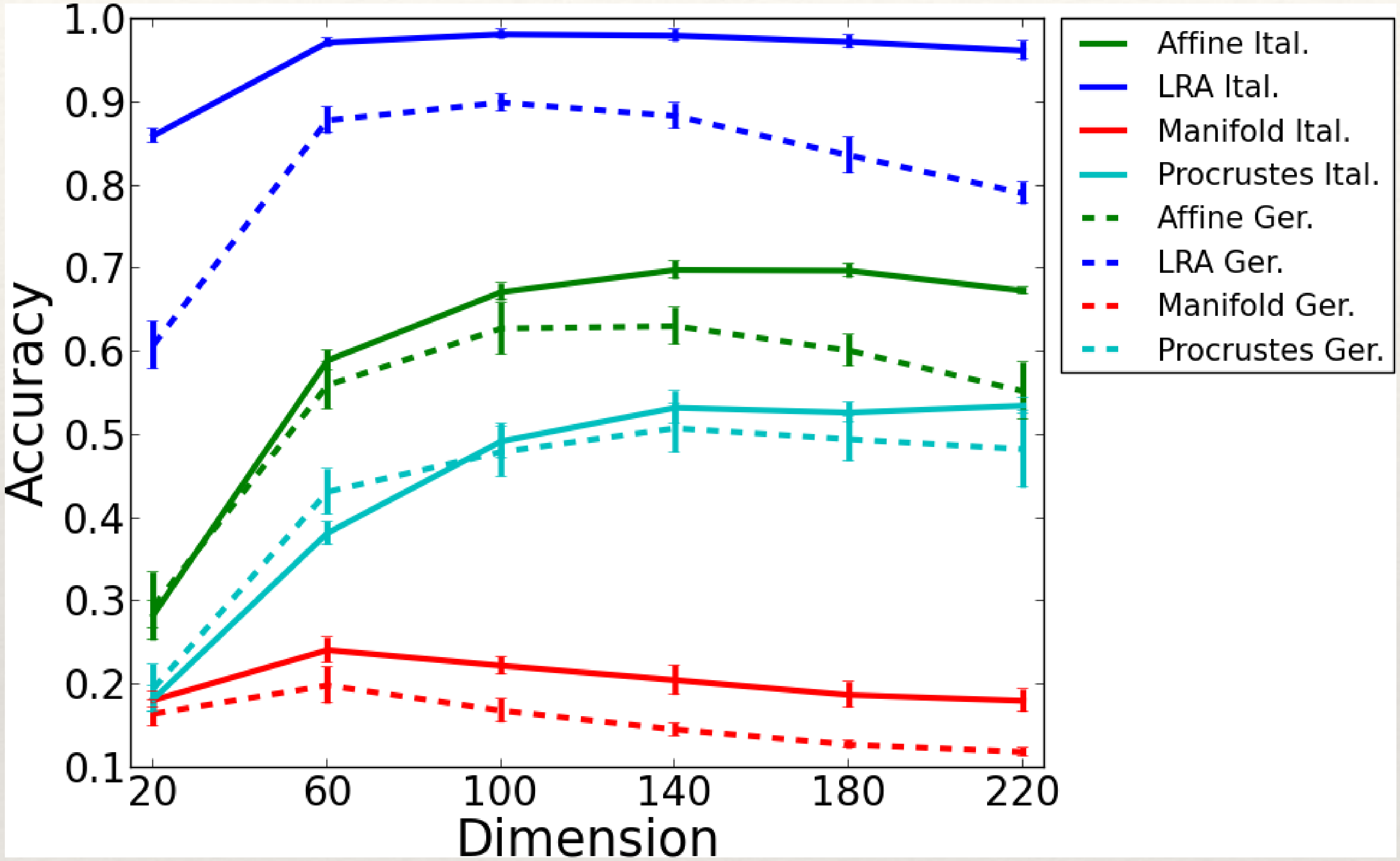
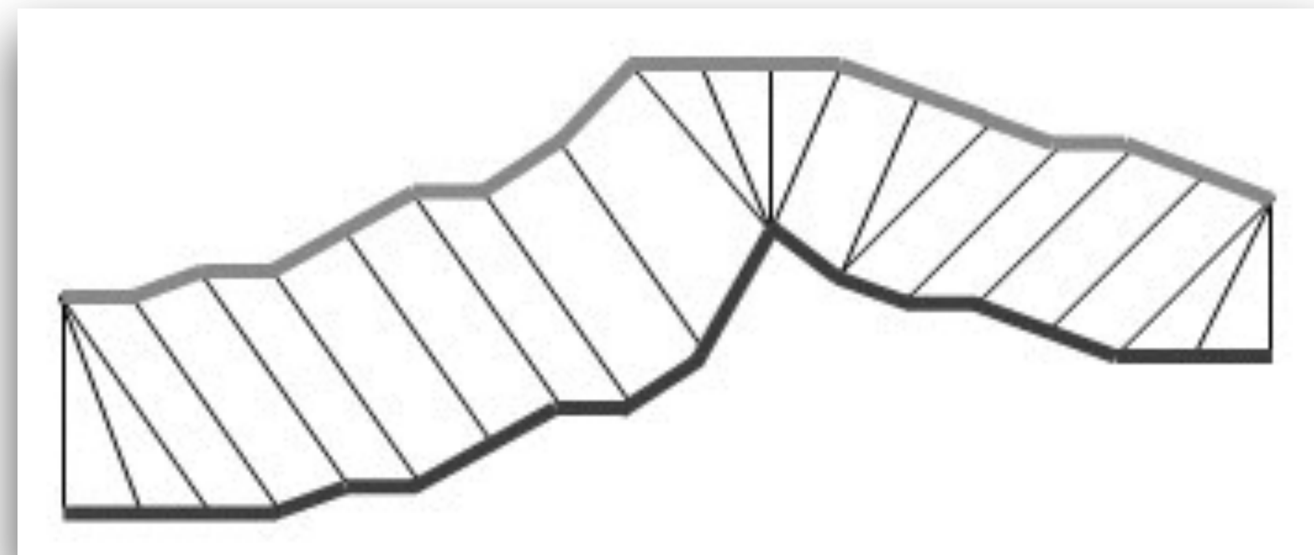


Figure 5: Cross validation results of EU parallel corpus with 2410 Italian-English sentences pairs and 2110 German-English sentences pairs.

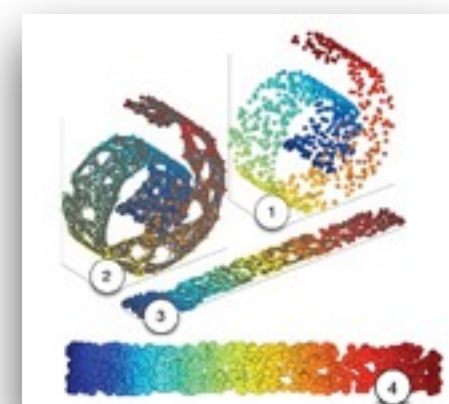
Manifold Warping

(Hoa, Carey, Mahadevan: AAAI, 2012)

Dynamic Time Warping



+



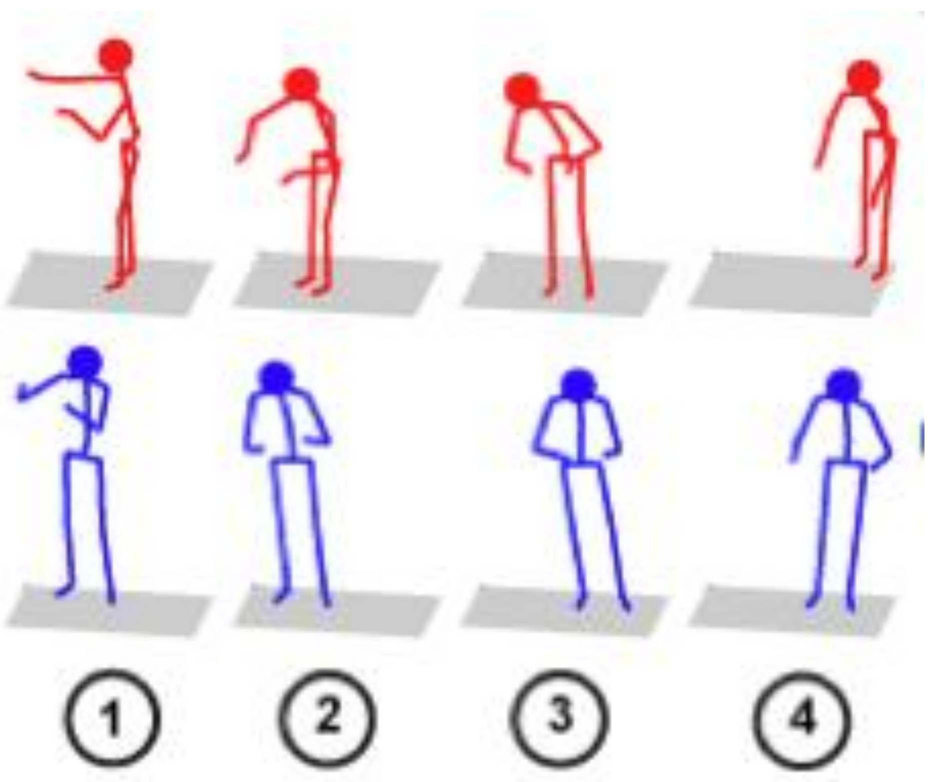
Iterate:

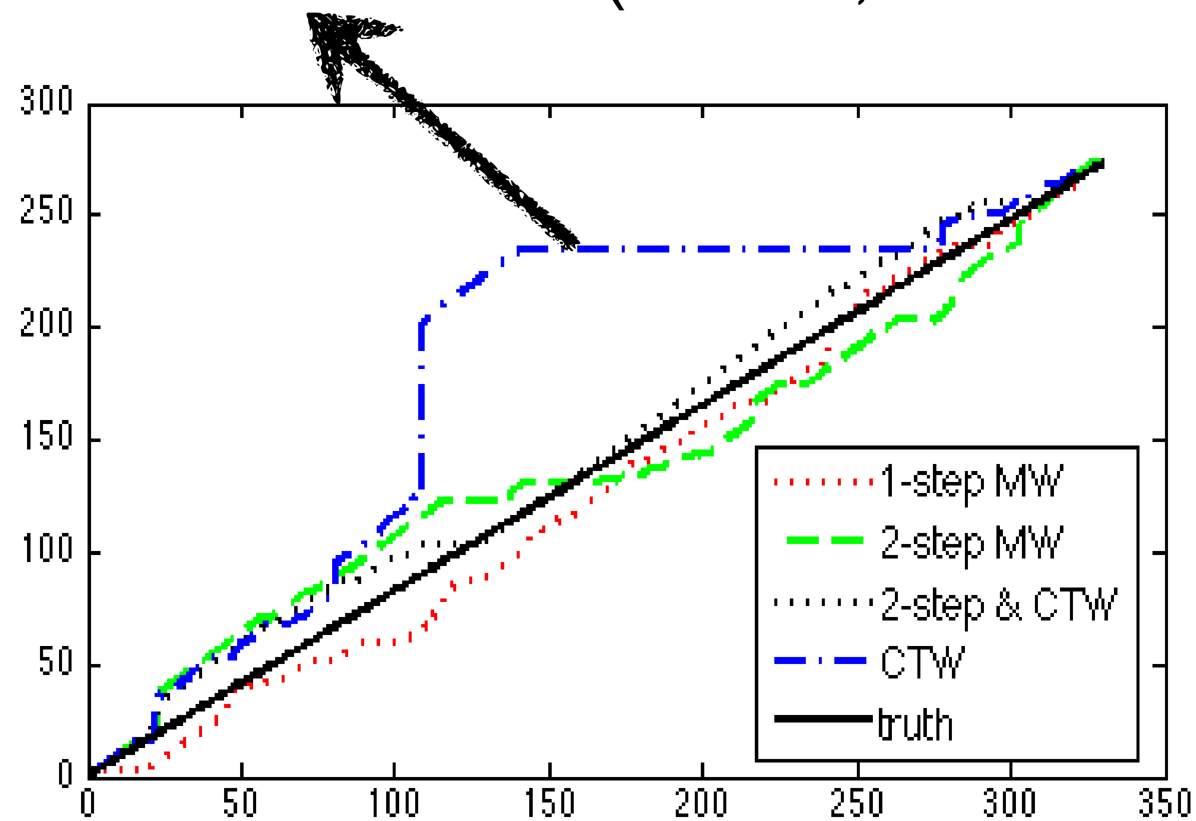
- Find projection to lower-dimensional space
- Find new set of correspondences

Manifold Alignment

Activity Recognition

CCA+DTW (Zhou, NIPS 2009)





The resulted alignment path of manifold warping is much closer to the ground truth alignment

Vu, Carey, and Mahadevan, AAAI 2012

Social Network Alignment

Sparse Manifold Alignment

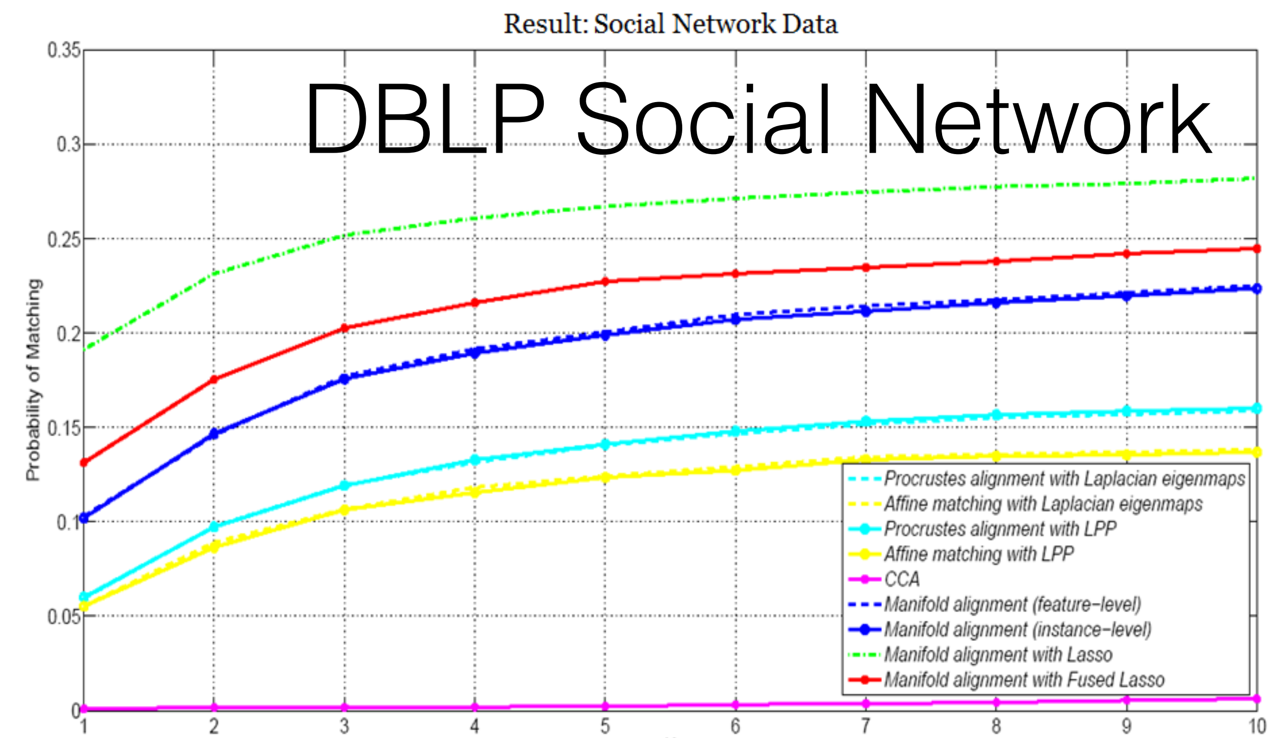
Use Lasso to find a sparse solution.

Manifold Alignment with Lasso

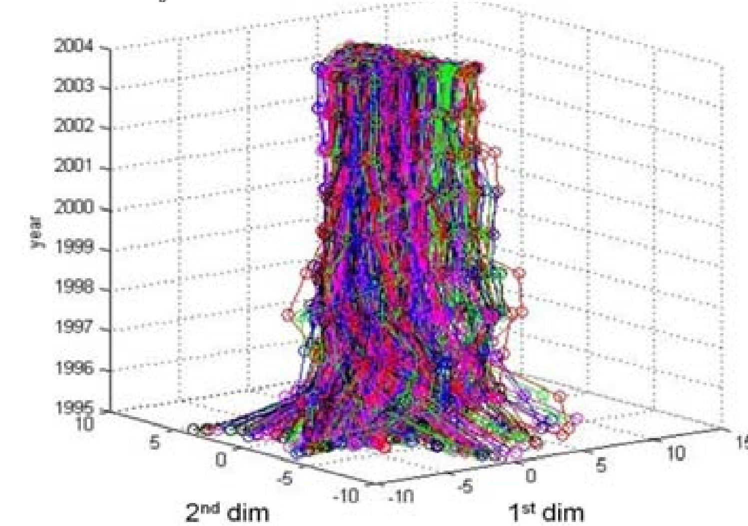
$$\|W^T Z - U_h^T Q^T\|_F^2 + \alpha \|W\|_{1,1}.$$

Manifold Alignment with Fused Lasso

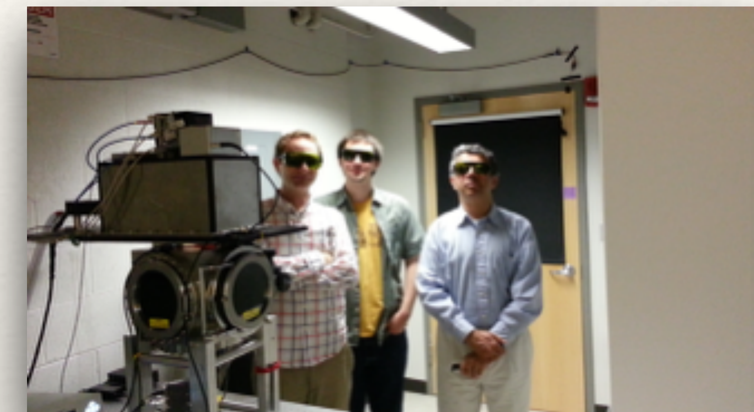
$$\|W^T Z - U_h^T Q^T\|_F^2 \quad + \quad \alpha \|W\|_{1,1} + \beta \sum_{j=1}^{h} \sum_{k=2}^{p+q} |w_{j,k} - w_{j,k-1}|.$$



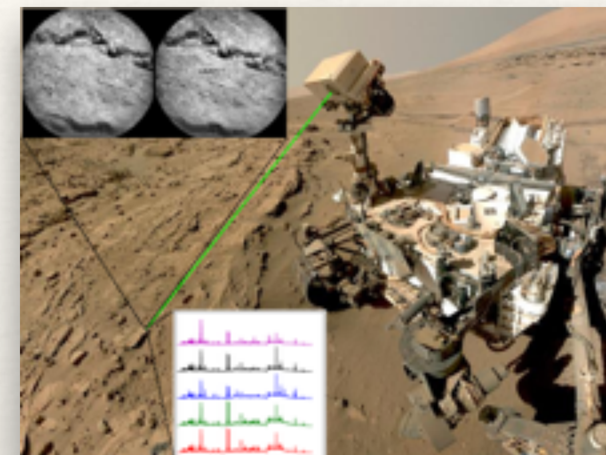
Wang, Liu, Vu, and Mahadevan, 2012

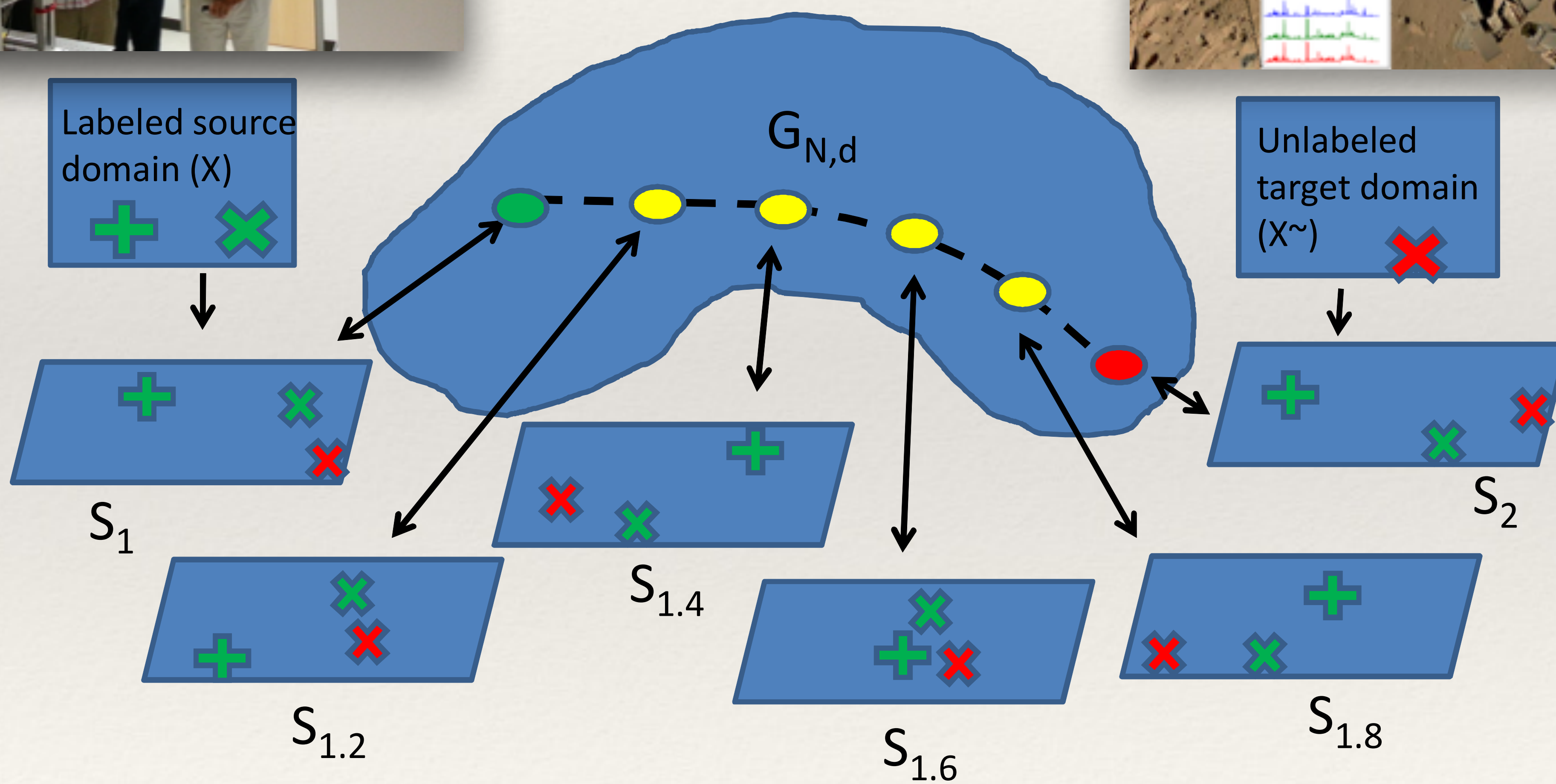


Smooth Transfer Learning



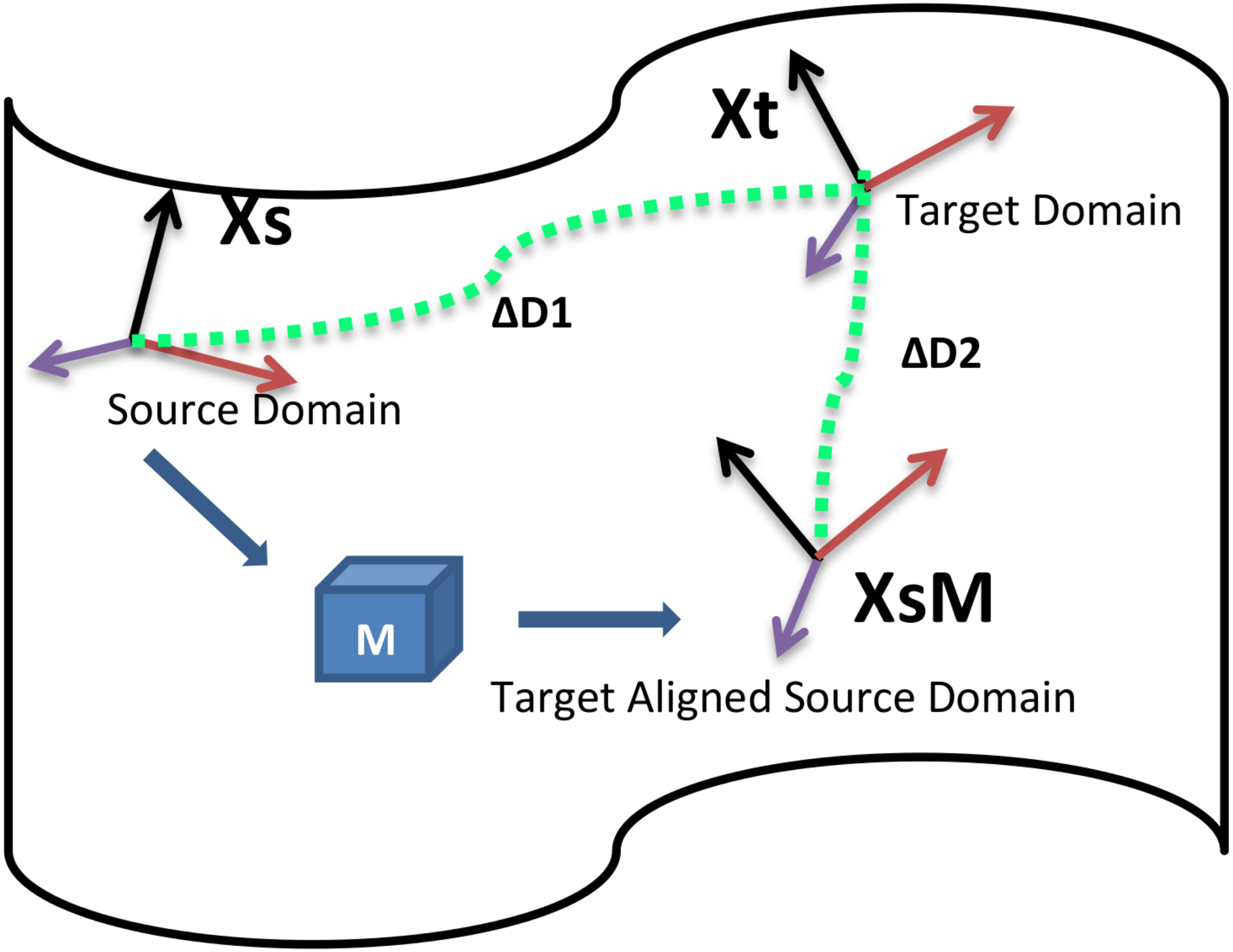
Smooth path between source and target





Subspace Alignment

- CCA and manifold alignment are based on aligning instances
- They assume a discrete source and target domain
- They are non-incremental methods
- We present an alternative approach based on aligning subspaces



Subspace Alignment (Fernando et al., CVPR 2014)

Subspace Alignment

$$F(M) = ||X_S M - X_T||_F^2$$

$$M^* = argmin_M(F(M))$$

$$\begin{aligned} M^* &= argmin_M||X_S' X_S M - X_S' X_T||_F^2 \\ &= argmin_M||M - X_S' X_T||_F^2. \end{aligned}$$

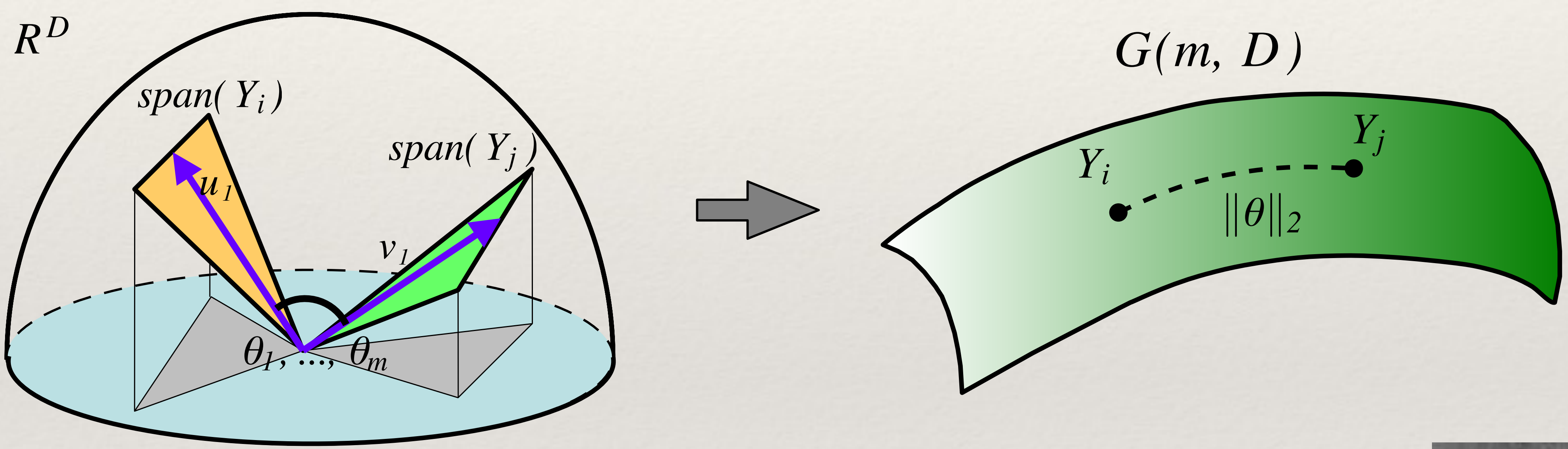
Incremental Subspace Alignment

$$\|(S_0^t + \delta S_0^t)M^{t+1} - (S_1^t + \delta S_1^t)\|_F^2$$

$$M^{t+1} = (S_0^t + \delta S_0^t)^T (S_1^t + \delta S_1^T)$$

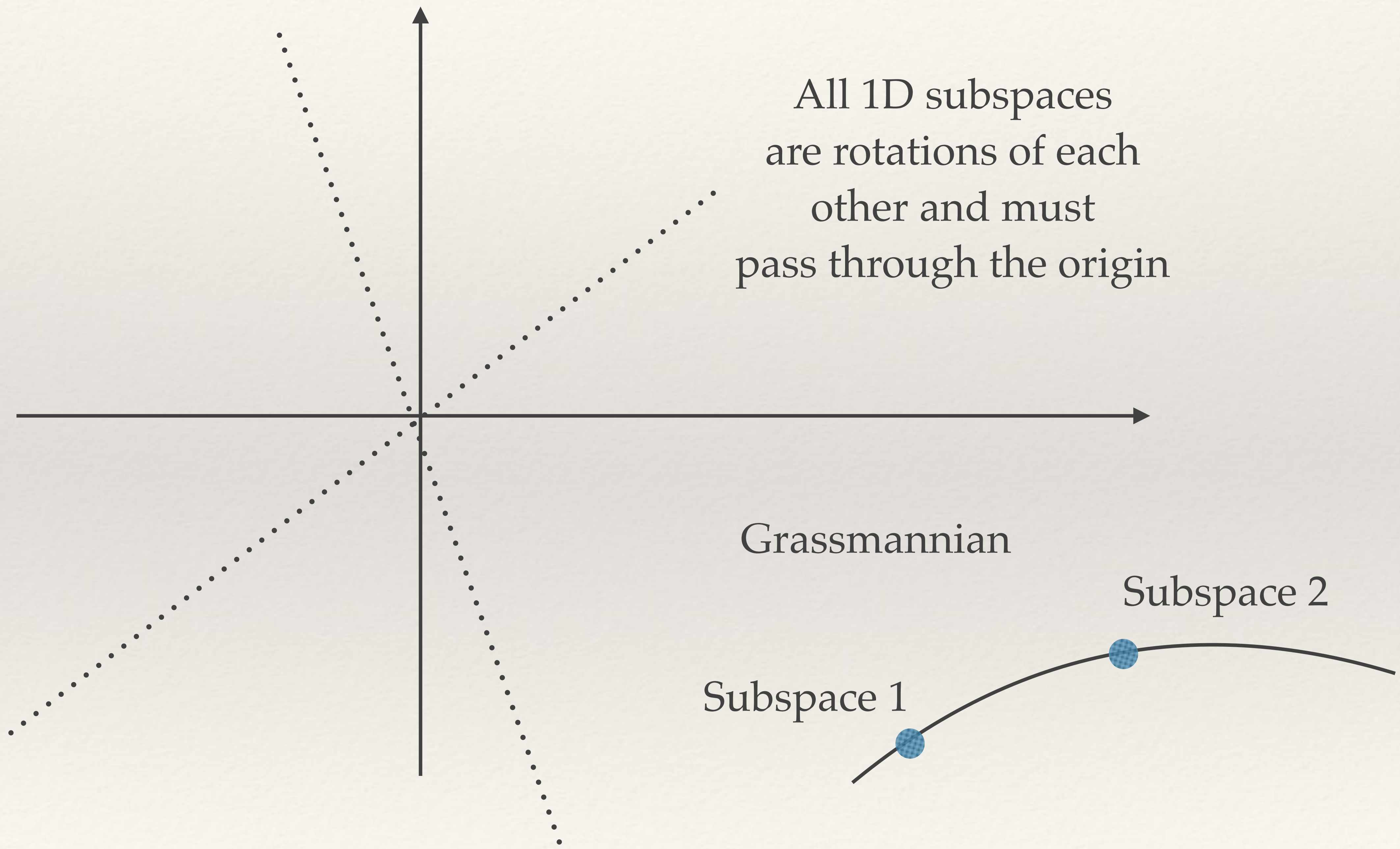
$$M^{t+1} = M^t + \delta M^t$$

Grassmannian Manifolds



1809-1877

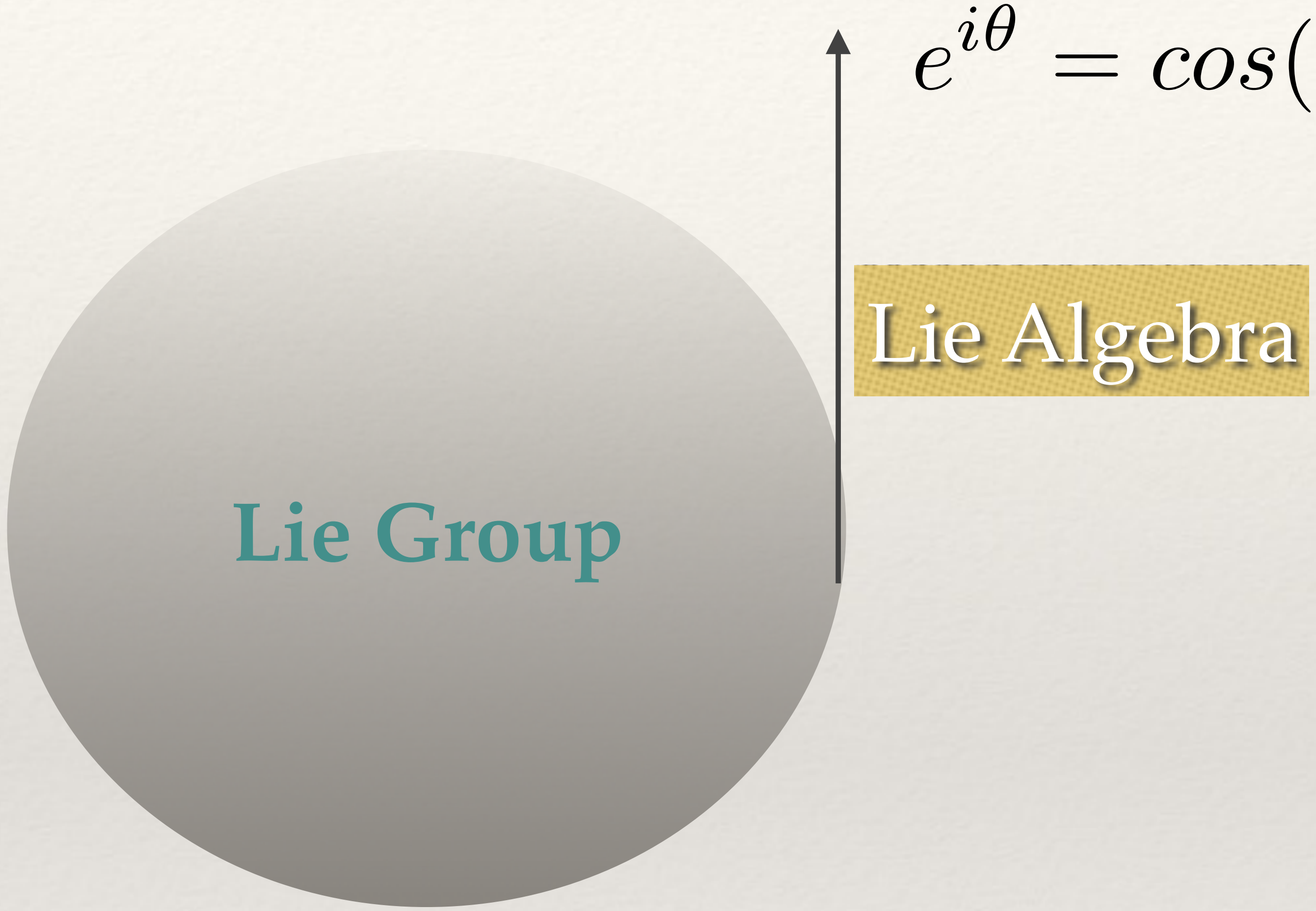
2D Example


All 1D subspaces
are rotations of each
other and must
pass through the origin
Grassmannian
Subspace 2
Subspace 1

Rotations in n-dimensions

$$e^{i\theta} = cos(\theta) + isin(\theta)$$

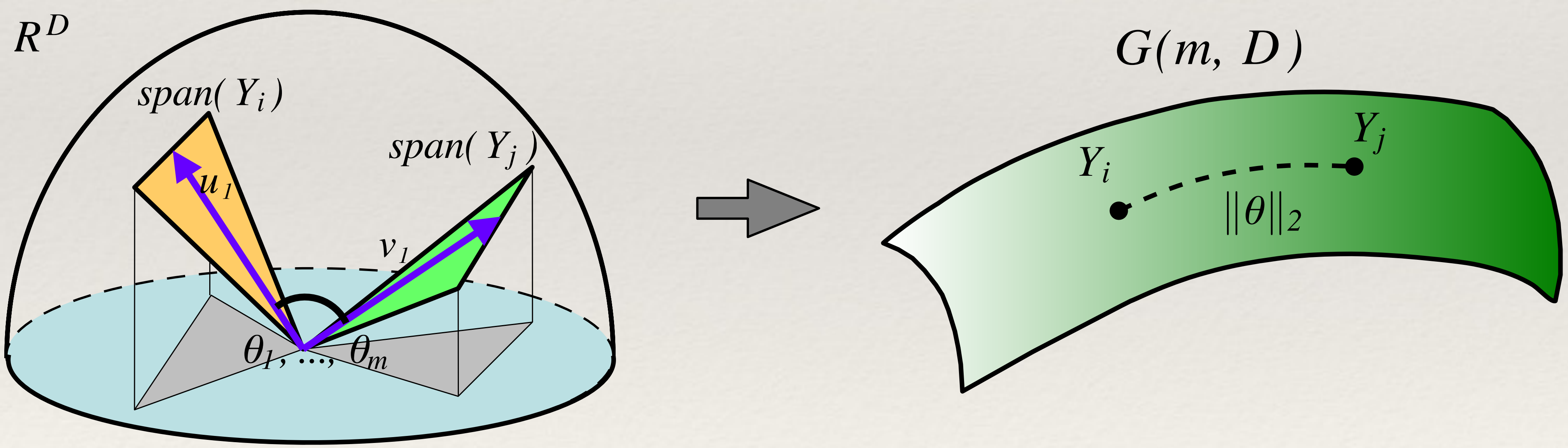
Sphere
in n-dim



Geodesics on Lie Groups

In a Lie group, gradients live in the tangent space, not in the group

Log map: Lie group to tangent space
Exponential map: tangent space to Lie group

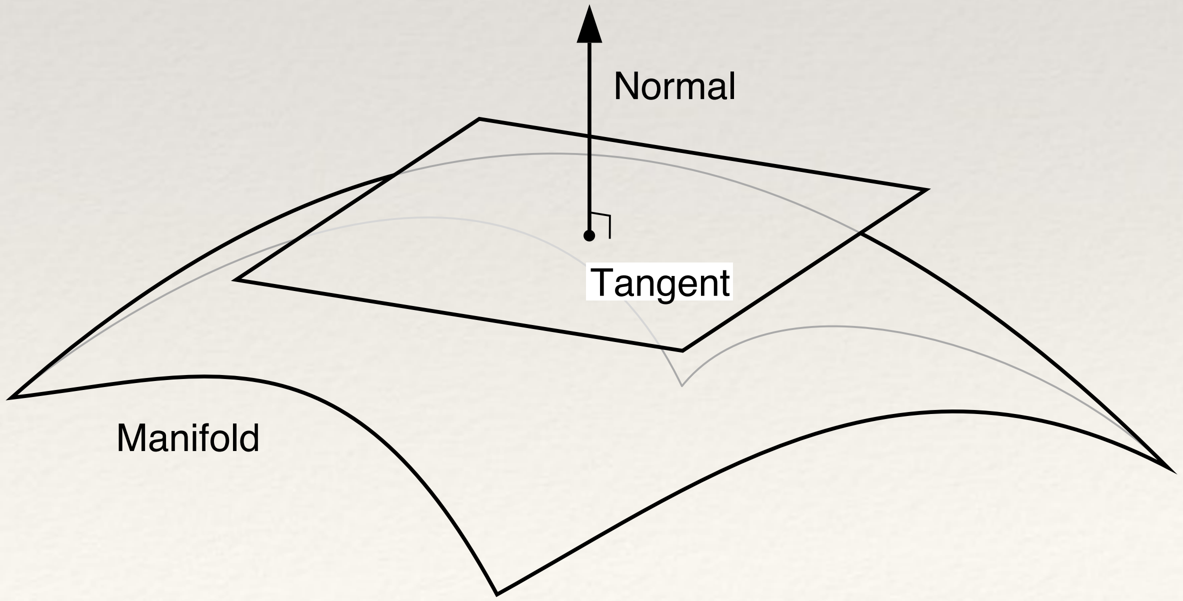


Subspace Manifolds

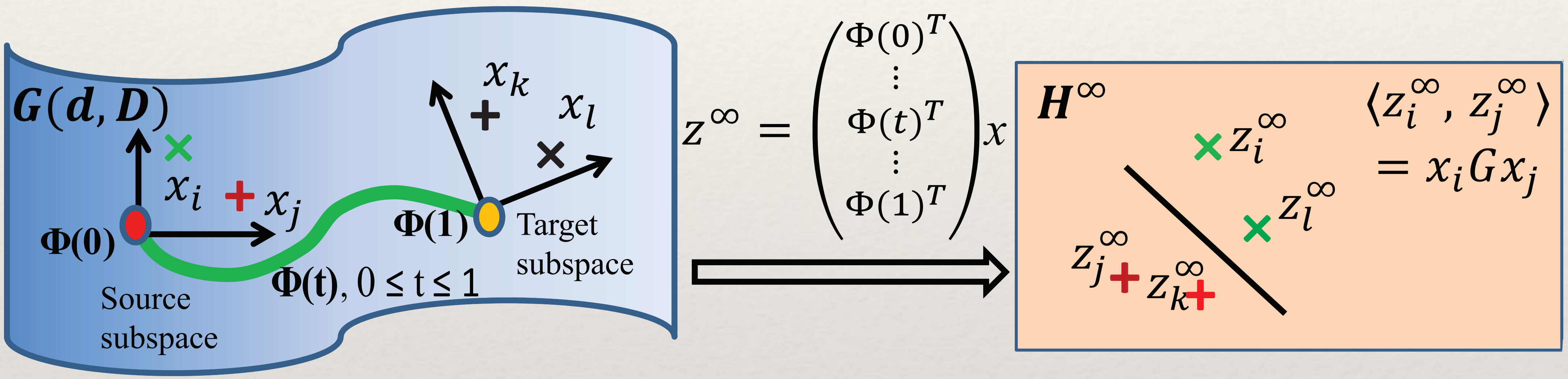
Space	Symbol	Matrix rep.	Quotient rep.
Orthogonal group	O_n	Q	–
Stiefel manifold	$V_{n,\,p}$	Y	O_n/O_{n-p}
Grassmann manifold	$G_{n,\,p}$	None	$\left\{ \begin{array}{c} V_{n,\,p}/O_p \\ \text{or} \\ O_n/\left(O_p \times O_{n-p}\right) \end{array} \right\}$

Tangent Spaces

Space	Data structure	represents	Tangents Δ
Stiefel manifold	Y	one point	$Y^T\Delta$ = skew-symmetric
Grassmann manifold	Y	entire equivalence class	$Y^T\Delta = 0$



Geodesic Flow Kernels



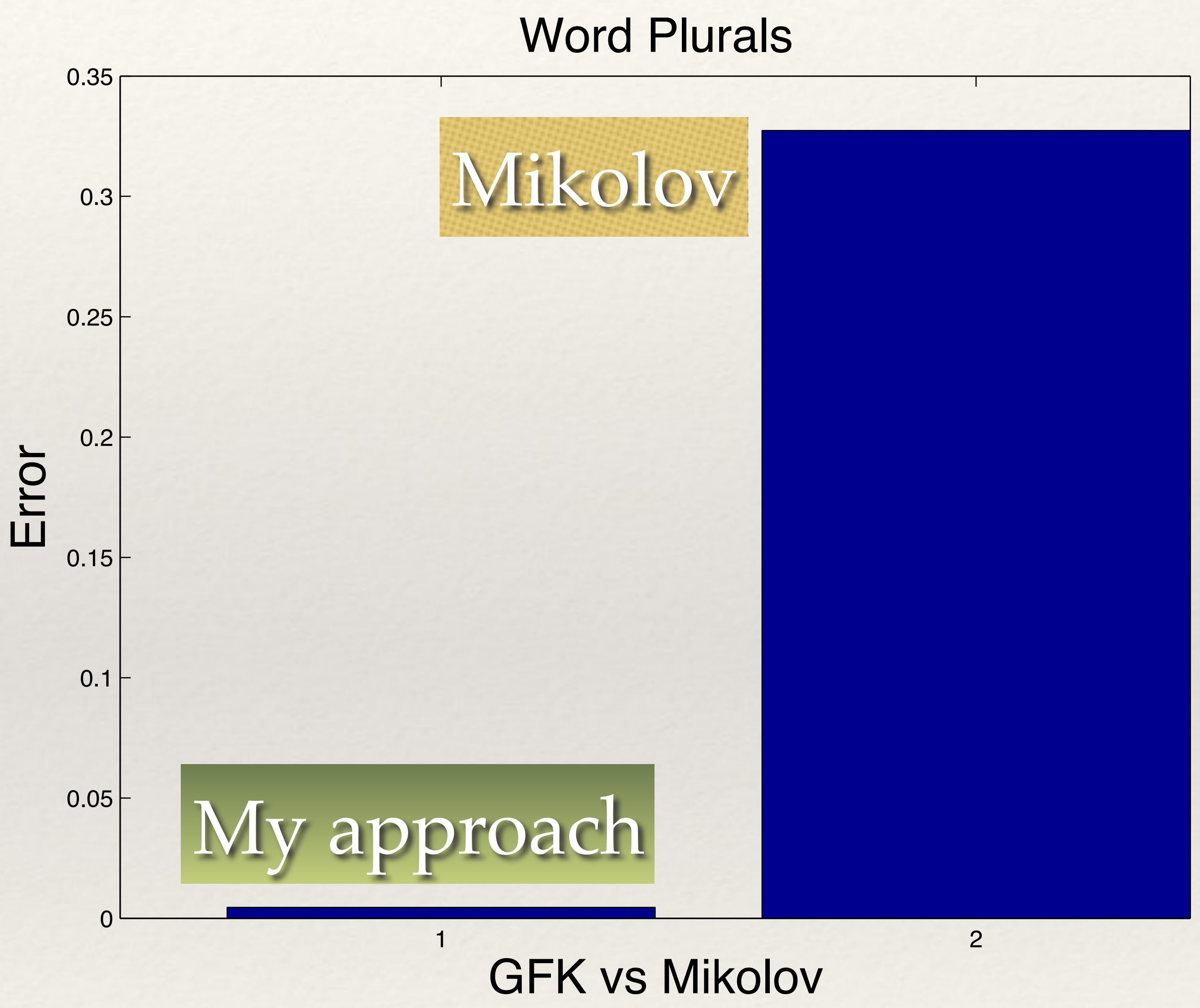
Gong et al., CVPR 2012

Word Analogy Results

$$\langle \boldsymbol{z}_i^{\infty}, \boldsymbol{z}_j^{\infty} \rangle = \int_0^1 (\boldsymbol{\Phi}(t)^{\mathrm{T}} \boldsymbol{x}_i)^{\mathrm{T}} (\boldsymbol{\Phi}(t)^{\mathrm{T}} \boldsymbol{x}_j)\, dt = \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{G} \boldsymbol{x}_j$$

$$\langle x_i, x_j \rangle_G = \frac{x_i^T G x_j}{\|\sqrt{G} x_j\|_2 \|\sqrt{G} x_j\|_2}$$

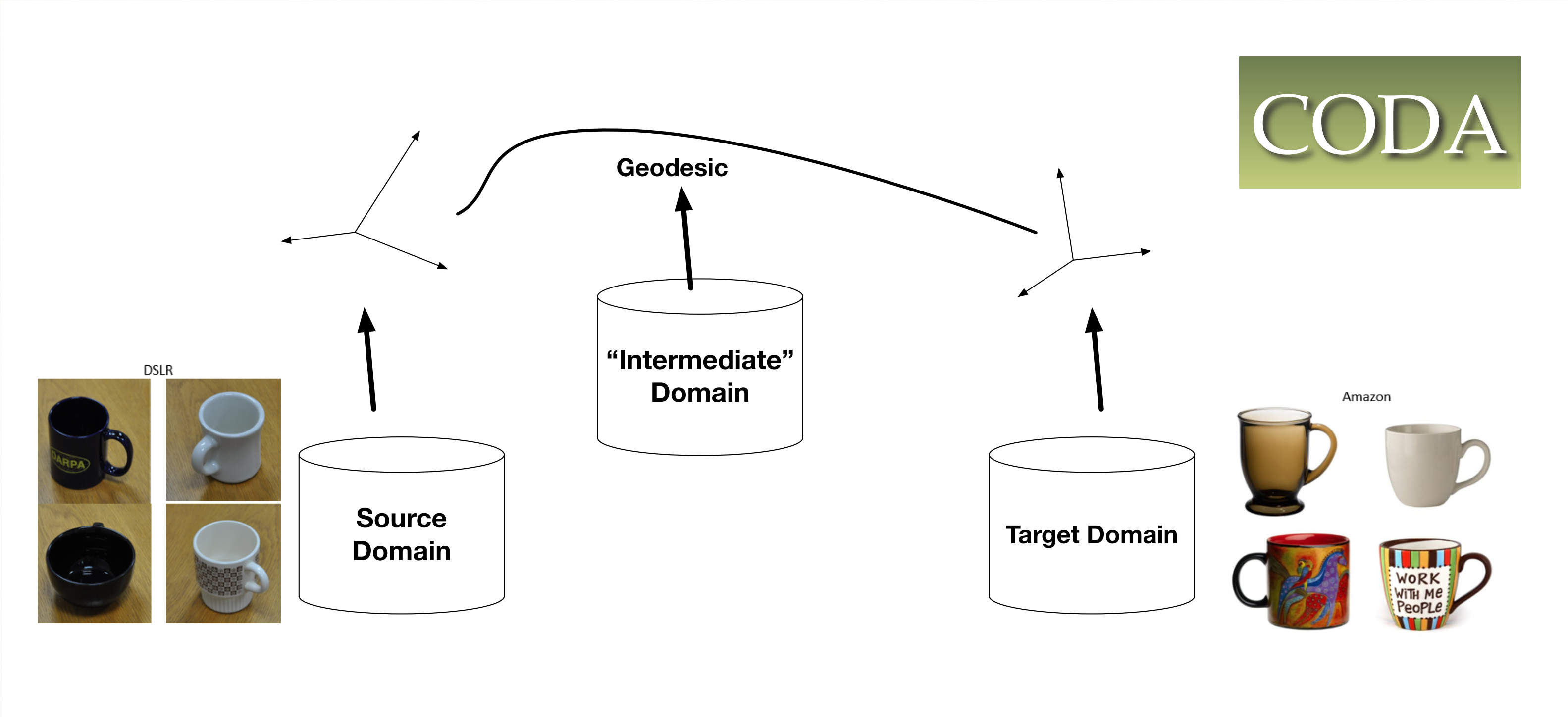
car cars woman X



Comparisons

	Relation	CosADD	CosMUL	GFKCosADD	GFKCosMUL
Google	capital-common-countries	89.52%	98.22%	**100%**	**100%**
	capital-world	51.25%	**80.43%**	72.61%	76.68%
	city-in-state	7.62%	43.12%	46.00%	**69.59%**
	currency	18.57%	15.17%	**33.43%**	27.86%
	family (gender inflections)	69.36%	81.42%	**94.26%**	93.67%
	gram1-adjective-to-adverb	30.54%	39.91%	**89.31%**	86.18%
	gram2-opposite	39.40%	45.32%	**75.00%**	73.02%
	gram3-comparative	73.49%	88.81%	**92.71%**	91.96%
	gram4-superlative	33.80%	67.61%	86.17%	**90.43%**
	gram5-present-participle	80.01%	92.32%	**99.81%**	99.71%
	gram6-nationality-adjective	92.49%	95.30%	**98.93%**	98.43%
	gram7-past-tense	84.29%	93.79%	**99.80%**	99.29%
	gram8-plural (nouns)	80.03%	90.16%	**98.19%**	97.67%
	gram9-pluran-verbs	82.52%	91.72%	**97.81%**	97.58
MSR	adjectives	35.90%	47.19%	59.55%	**60.44%**
	nouns	69.91%	83.04%	**84.10%**	83.90%
	verbs	81.26%	91.86%	**89.03%**	88.86%

Correspondence Optimized DA

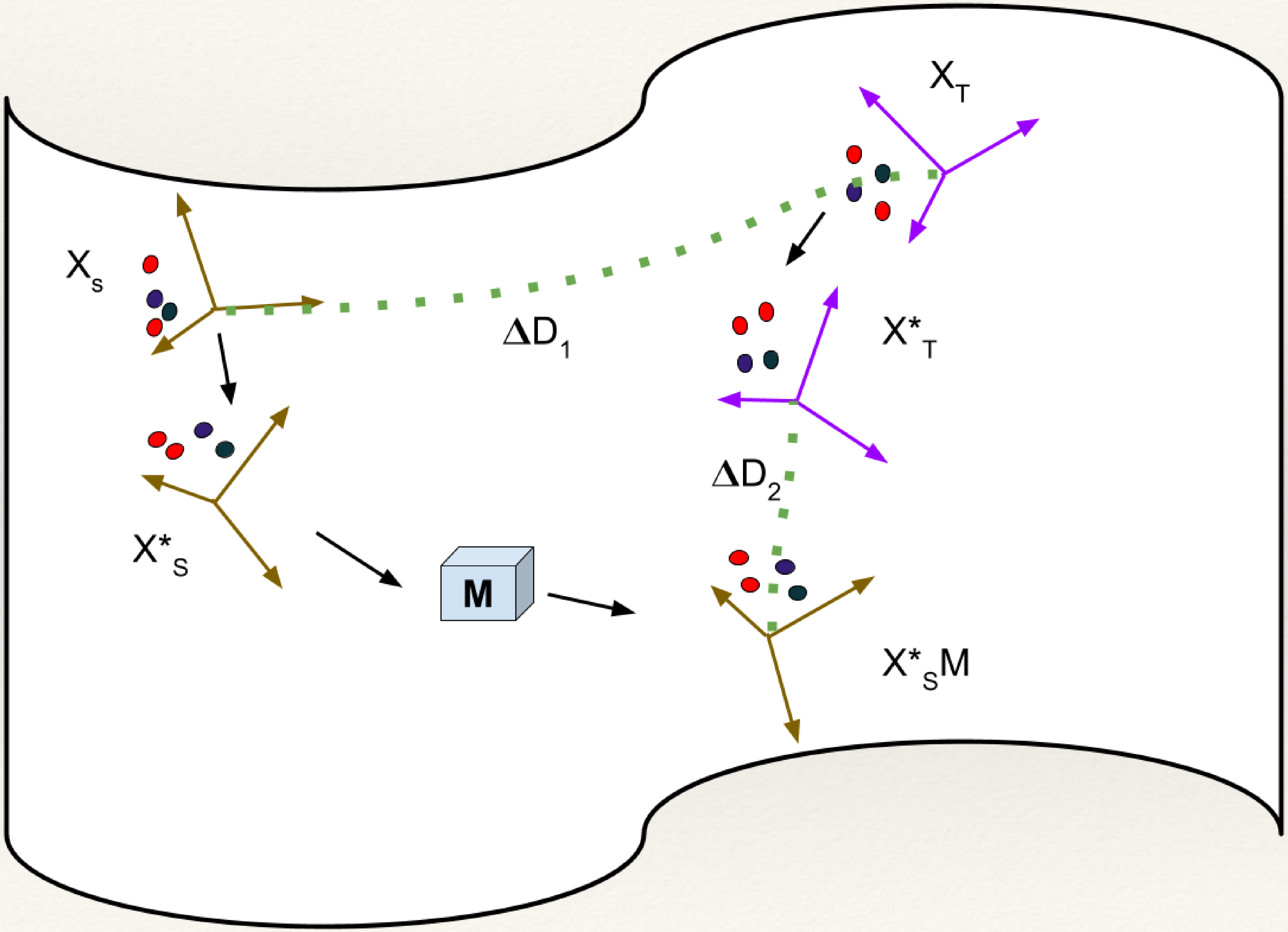


Giguere, 2016

$$f(S_0, S_1) = \sum_{x_i, x_j \in X_t} \frac{x_i S_0 S_0^T S_1 S_1^T x_j^T}{|X_t|}$$

$$- \frac{1}{2} ||X_0 - X_0 S_0 S_0^T||_F^2 - \frac{1}{2} ||X_1 - X_1 S_1 S_1^T||_F^2$$

Correspondence Optimized DA



Computer Vision Testbed

Table 2. Recognition accuracy with semi-supervised DA with SVM classifier(Office dataset + Caltech10).

Method	C→A	D→A	W→A	A→C	D→C	W→C	A→D	C→D	W→D	A→W	C→W	D→W
NA	45.10	32.80	28.20	37.80	28.40	23.80	38.60	39.30	71.80	38.70	64.60	83.10
PCA_S	46.20	37.70	35.60	37.10	31.60	29.30	39.10	33.70	66.80	36.10	76.60	83.10
PCA_T	43.60	38.50	34.30	36.60	31.60	27.80	39.10	34.10	64.20	36.80	67.90	83.10
GFK	45.40	36.30	32.10	38.80	28.50	26.30	39.50	39.10	70.30	41.10	77.70	83.10
SA	44.70	41.60	39.30	**40.60**	34.80	32.60	40.90	41.10	**77.60**	38.20	**82.20**	**87.10**
OSA	**46.51**	**46.38**	**45.86**	36.17	**34.95**	**34.27**	**49.79**	**49.82**	73.16	**58.89**	53.99	78.26

Domain Invariant Projection

[Baktashmotlagh et al., ICCV 2013]

DIP is based on doing gradients on the Grassmannian manifold to optimize the kernelized MMD metric

Compute the gradient $\nabla f_{\boldsymbol{W}}$ of the objective function f on the manifold at the current estimate $\boldsymbol{W}$ as

$$\nabla f_{\boldsymbol{W}} = \partial f_{\boldsymbol{W}} - \boldsymbol{W}\boldsymbol{W}^T \partial f_{\boldsymbol{W}} , \quad (1)$$

Riemannian gradient

Euclidean gradient

Domain Invariant Projection

$$\begin{aligned} \boldsymbol{W}^* \quad &= \quad \underset{\boldsymbol{W}}{\operatorname{argmin}} \; D^2(\boldsymbol{W}^T\boldsymbol{X_s}, \boldsymbol{W}^T\boldsymbol{X_t}) \\ \text{s.t.} \quad & \quad \boldsymbol{W}^T\boldsymbol{W} = \boldsymbol{I}_d \ , \end{aligned}$$

$$\boldsymbol{G}_{ss}(i,j) = -\frac{2}{\sigma} k_G(\boldsymbol{x}_s^i, \boldsymbol{x}_s^j)(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)(\boldsymbol{x}_s^i - \boldsymbol{x}_s^j)^T \boldsymbol{W}$$

$$\frac{\partial f}{\partial \boldsymbol{W}} = \sum_{i,j=1}^{n} \frac{\boldsymbol{G}_{ss}(i,j)}{n^2} + \sum_{i,j=1}^{m} \frac{\boldsymbol{G}_{tt}(i,j)}{m^2} - 2 \sum_{i,j=1}^{n,m} \frac{\boldsymbol{G}_{st}(i,j)}{mn}$$

DIP Results in Computer Vision

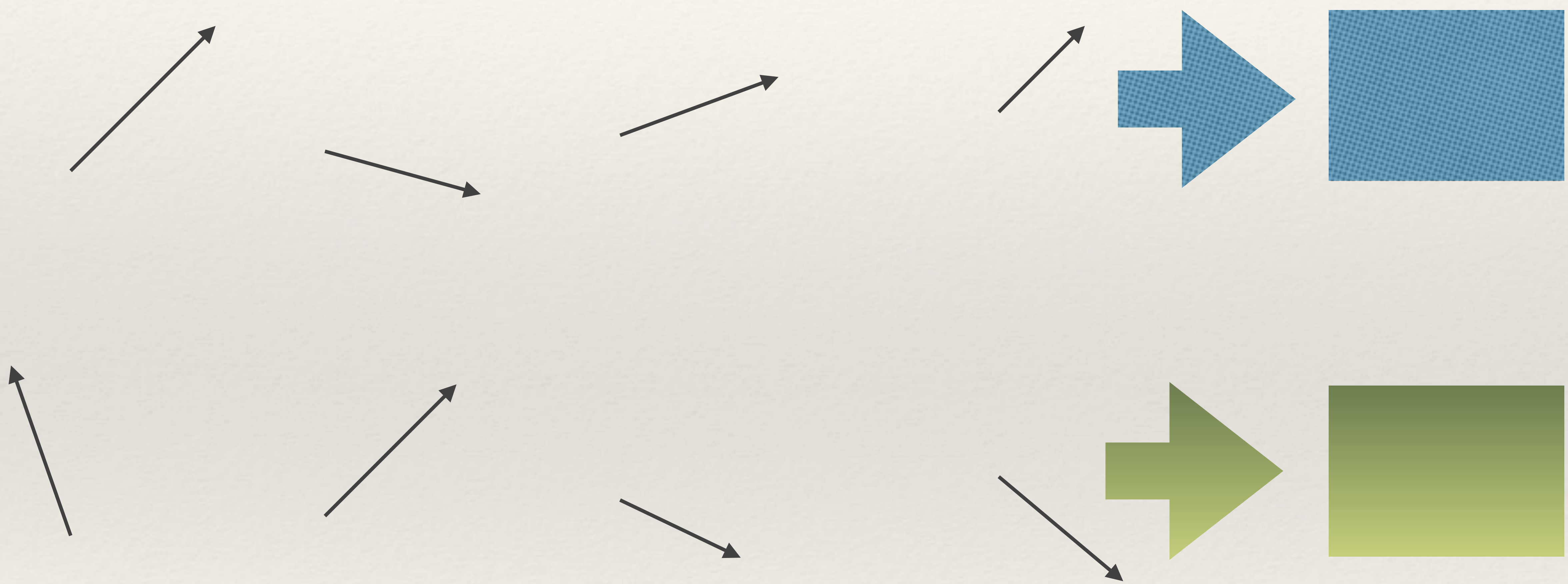
Method	$A \to C$	$A \to D$	$A \to W$	$C \to A$	$C \to D$	$C \to W$	$W \to A$	$W \to C$	$W \to D$
NO ADAPT-1NN	26	25.5	29.8	23.7	25.5	25.8	23	20	59.2
NO ADAPT-SVM	41.7	41.4	34.2	51.8	54.1	46.8	31.1	31.5	70.7
TCA[24]	35.0	36.3	27.8	41.4	45.2	32.5	24.2	22.5	80.2
GFK[15]	42.2	42.7	40.7	44.5	43.3	44.7	31.8	30.8	75.6
SCL[5]	42.3	36.9	34.9	49.3	42.0	39.3	34.7	32.5	83.4
KMM[18]	42.2	42.7	42.4	48.3	53.5	45.8	31.9	29.0	72.0
LM[14]	45.5	47.1	46.1	56.7	57.3	49.5	40.2	35.4	75.2
DIP	**47.4**	**50.3**	47.5	55.7	60.5	**58.3**	**42.6**	34.2	88.5
DIP-CC	47.2	49.04	**47.8**	**58.7**	**61.2**	58	40.9	**37.2**	**91.7**
DIP(Poly)	47.3	49.1	45.1	56.1	58.6	57	42.8	36.5	89.8
DIP-CC(Poly)	47.4	48.4	46.1	56.4	58.6	58	42.7	36.5	89.8

Table 1. Recognition accuracies on 9 pairs of source/target domains using the evaluation protocol of [14]. C**: Caltech,** A**: Amazon,** W**: Webcam,** D**: DSLR**.

Batch vs. Incremental Methods

- Both MA and SA domain adaptation methods are batch mode techniques
- They require having all the data upfront, and involve a matrix eigenvector (SVD) computation
- Given a new instance, the whole solution has to be recomputed
- Can we design an incremental method?

Incremental Subspace Tracking



Subspace Tracking

1. Introduction. We seek to identify an unknown subspace $\mathcal{S}$ of dimension d in $\mathbb{R}^n$, described by an $n \times d$ matrix $\bar{U}$ whose orthonormal columns span $\mathcal{S}$. Our data consist of a sequence of vectors v_t of the form

$$v_t = \bar{U} s_t, \tag{1.1}$$

where $s_t \in \mathbb{R}^d$ is a random vector whose elements are independent and identically distributed (i.i.d.) in $\mathcal{N}(0, 1)$. Critically, we observe only a subset $\Omega_t \subset \{1, 2, \ldots, n\}$ of the components of v_t.

Key Theorem
(Edelman et al, SIAM)

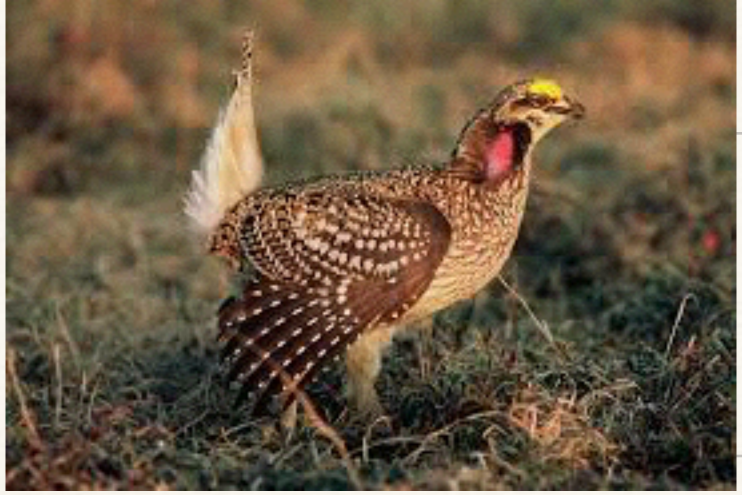
2.5.1. Geodesics (Grassmann). A formula for geodesics on the Grassmann manifold was given via (2.32); the following theorem provides a useful method for computing this formula using n-by-p matrices.

THEOREM 2.3. *If* $Y(t) = Qe^{t\left(\begin{smallmatrix} 0 & -B^T \\ B & 0 \end{smallmatrix}\right)} I_{n,p}$, *with* $Y(0) = Y$ *and* $\dot{Y}(0) = H$, *then*

$$Y(t) = \begin{pmatrix} YV & U \end{pmatrix} \begin{pmatrix} \cos \Sigma t \\ \sin \Sigma t \end{pmatrix} V^T, \tag{2.65}$$

where $U\Sigma V^T$ *is the compact singular value decomposition of* H.

$$Q(t) = Q(0) \exp t \begin{pmatrix} 0 & -B^T \\ B & 0 \end{pmatrix} \tag{2.32}$$



GROUSE

(Balzano et al., 2010)

(Grassmannian Rank-One Update Subspace Estimation)

Algorithm 1 GROUSE

Given U_0, an $n \times d$ orthonormal matrix, with $0 < d < n$;
Set $t := 1$;
repeat
  Take Ω_t and $(v_t)_{\Omega_t}$ from (1);
  Define $w_t := \arg\min_w \|[U_t]_{\Omega_t} w - [v_t]_{\Omega_t}\|_2^2$;
  Define $p_t := U_t w_t$; $[r_t]_{\Omega_t} := [v_t]_{\Omega_t} - [p_t]_{\Omega_t}$;
  $[r_t]_{\Omega_t^C} := 0$; $\sigma_t := \|r_t\| \, \|p_t\|$;
  Choose $\eta_t > 0$ and set

$$U_{t+1} := U_t + (\cos(\sigma_t \eta_t) - 1) \frac{p_t}{\|p_t\|} \frac{w_t^T}{\|w_t\|} + \sin(\sigma_t \eta_t) \frac{r_t}{\|r_t\|} \frac{w_t^T}{\|w_t\|} \,. \tag{2}$$

  $t := t + 1$;
until termination

Derivation of GROUSE

$$F(S;t) = \min_a \|\Delta_{\Omega_t}(Ua - v_t)\|^2$$

$$\frac{dF}{dU} = -2(\Delta_{\Omega_t}(v_t - Uw))w^T$$

$$= -2rw^T \qquad r := \Delta_{\Omega_t}(v_t - Uw)$$

$$\nabla F = (I - UU^T)\frac{dF}{dU}$$

$$= -2(I - UU^T)rw^T = -2rw^T$$

Derivation of GROUSE

$$\sigma = 2||r||||w||$$

$$-2rw^T = \left[\begin{array}{cccc} -\frac{r}{||r||} & x_2 & \dots & x_d \end{array} \right] \times \text{diag}(\sigma, 0, \dots, 0) \times \left[\begin{array}{cccc} \frac{w}{||w||} & y_2 & \dots & y_d \end{array} \right]^T$$

$$\begin{aligned} U(\eta) &= U + \frac{(\cos(\sigma\eta) - 1)}{||w||^2} U w w^T + \sin(\sigma\eta) \frac{r}{||r||} \frac{w^T}{||w||} \\ &= U + \left(\sin(\sigma\eta) \frac{r}{||r||} + (\cos(\sigma\eta) - 1) \frac{p}{||p||} \right) \frac{w^T}{||w||} \end{aligned}$$

Incremental Subspace Alignment

$$\|(S_0^t + \delta S_0^t)M^{t+1} - (S_1^t + \delta S_1^t)\|_F^2$$

$$M^{t+1} = (S_0^t + \delta S_0^t)^T(S_1^t + \delta S_1^T)$$

$$M^{t+1} = M^t + \delta M^t$$

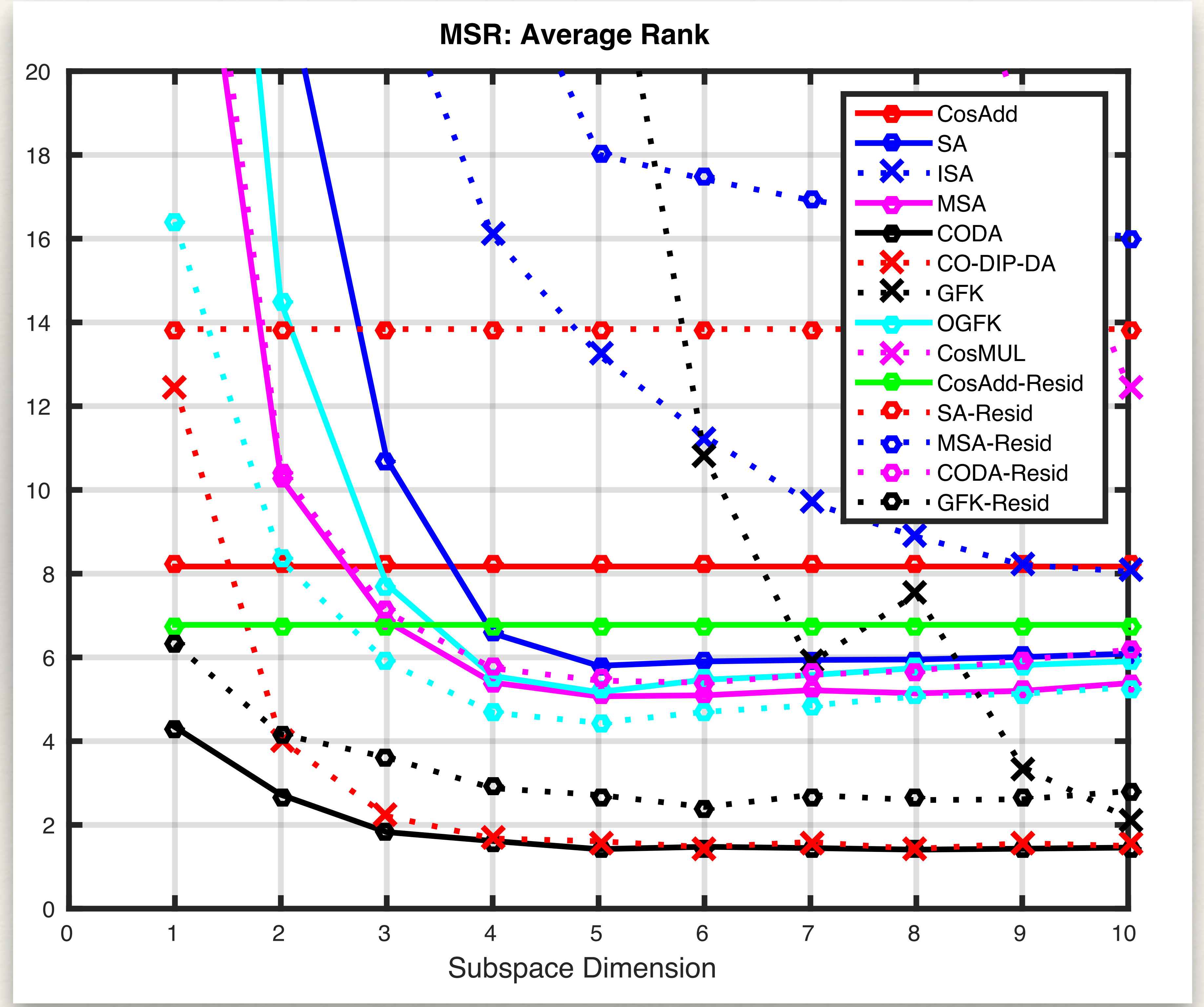
CO-DIP-DA

Correspondence optimized domain invariant projection for domain adaptation (Mahadevan, 2016)

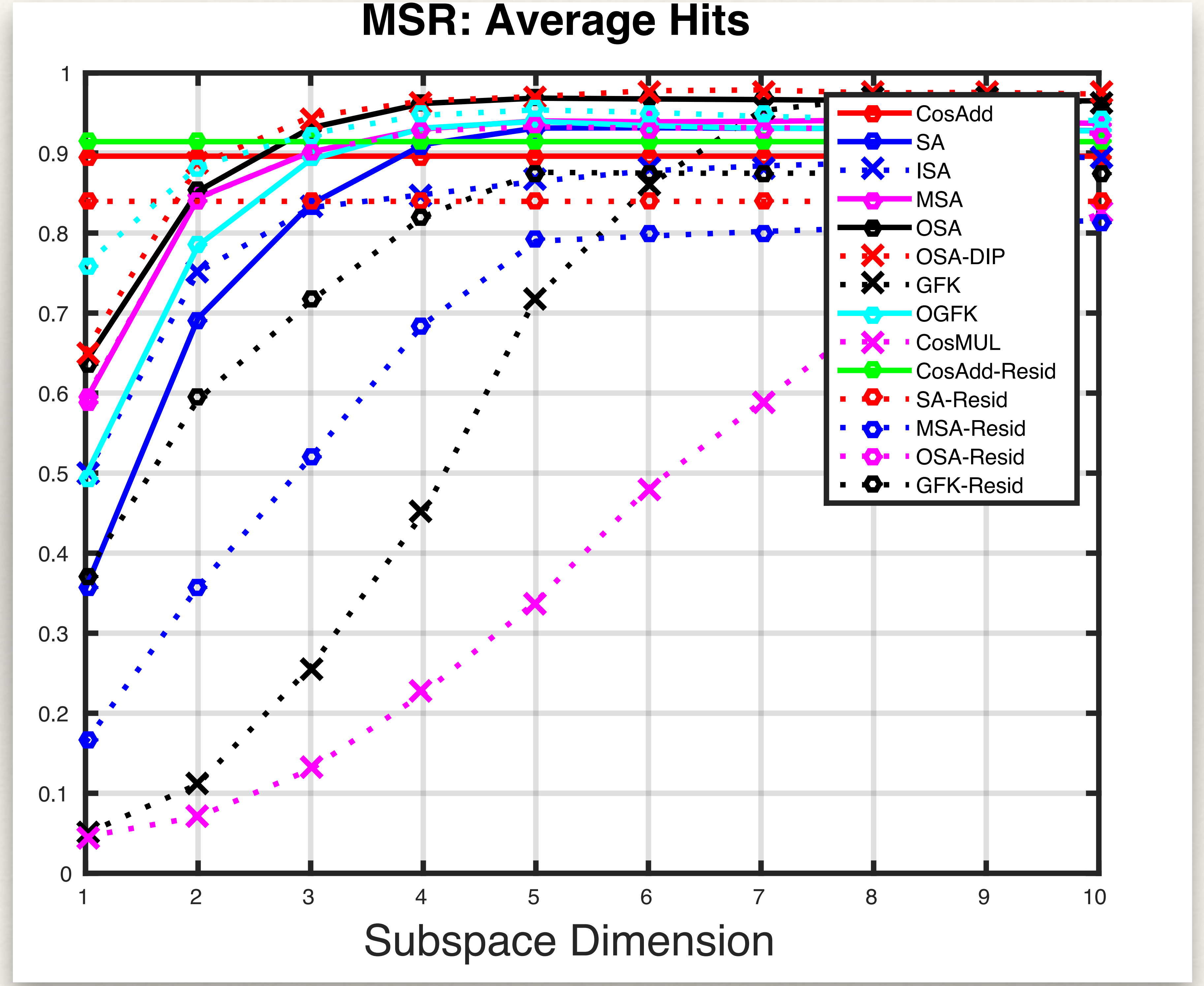
Combines minimization of kernelized maximum mean discrepancy with CODA

Uses a convex combination of gradients from DIP and CODA

Comparison of DA Methods



Comparison of DA Methods



Computer Vision: Comparison of DA Methods

Method	C-->A	D-->A	W-->A	A-->C	D-->C	W-->C	A-->D	C-->D	W-->D	A-->W	C-->W	D-->W
SA-Stnd	39.14	38.67	38.03	28.48	26.53	31.89	47.33	42.17	51.60	56.67	54.69	57.63
SA-CODA	48.46	47.40	44.38	36.40	32.29	35.35	54.14	52.06	58.22	62.98	62.74	68.33
GFK-Stnd	36.11	33.90	38.78	29.03	29.89	32.88	36.03	35.62	57.70	41.15	37.15	62.39
GFK-CODA	41.65	40.45	43.55	33.91	32.89	34.41	41.71	44.44	58.24	50.27	48.16	68.78

Resid:

Method	C-->A	D-->A	W-->A	A-->C	D-->C	W-->C	A-->D	C-->D	W-->D	A-->W	C-->W	D-->W
SA-Stnd	43.41	43.33	40.69	30.74	31.87	33.51	51.48	50.79	62.95	65.36	63.87	79.39
SA-CODA	44.35	45.99	43.80	36.87	31.98	33.32	51.83	47.87	70.78	59.83	58.82	78.67
GFK-Stnd	43.27	40.18	41.45	33.59	32.35	31.07	48.05	51.05	78.75	57.27	55.47	81.17
GFK-CODA	46.10	44.56	41.89	35.85	32.74	32.13	50.90	47.97	78.27	54.23	56.38	81.44

Normalized:

Method	C-->A	D-->A	W-->A	A-->C	D-->C	W-->C	A-->D	C-->D	W-->D	A-->W	C-->W	D-->W
SA-Stnd	44.08	42.89	40.15	31.34	30.78	32.74	49.78	48.97	62.41	64.94	61.71	78.31
SA-CODA	48.00	49.45	46.96	37.31	33.29	36.21	54.43	52.79	70.29	65.05	65.49	80.18
GFK-Stnd	46.07	43.24	43.71	34.37	31.79	33.77	48.51	48.94	77.73	58.26	57.87	82.84
GFK-CODA	49.53	46.59	45.30	39.33	33.70	36.74	51.05	52.10	77.22	60.51	62.20	84.94

```
sridhar@fovea:~/code/OptimizedDomainAdaptation-master$
```

Computer Vision Testbed: Comparison of DA Methods

Standard:

Method	C-->A	D-->A	W-->A	A-->C	D-->C	W-->C	A-->D	C-->D	W-->D	A-->W	C-->W	D-->W
SA-Stnd	36.80	38.58	38.00	28.50	26.61	31.16	45.75	40.19	51.81	50.69	52.91	59.72
SA-CODA	46.02	47.92	44.68	35.35	32.60	33.14	55.08	50.30	56.98	59.59	63.68	69.64
GFK-Stnd	35.60	34.69	38.28	27.93	29.43	30.89	32.22	32.84	58.06	33.47	37.92	63.60
GFK-CODA	40.78	39.34	41.67	31.39	32.47	32.84	41.22	45.05	57.25	44.49	45.80	70.85

Resid:

Method	C-->A	D-->A	W-->A	A-->C	D-->C	W-->C	A-->D	C-->D	W-->D	A-->W	C-->W	D-->W
SA-Stnd	40.84	43.31	42.98	34.19	30.13	32.05	53.87	49.21	63.00	60.81	60.38	78.68
SA-CODA	44.66	45.99	42.51	36.89	32.73	32.51	52.25	51.00	72.98	57.22	58.46	77.65
GFK-Stnd	39.91	42.32	39.68	31.62	31.66	29.52	49.11	48.97	79.57	56.12	53.80	82.05
GFK-CODA	44.44	43.91	41.06	33.67	32.17	32.09	51.87	47.35	77.19	53.30	55.57	81.38

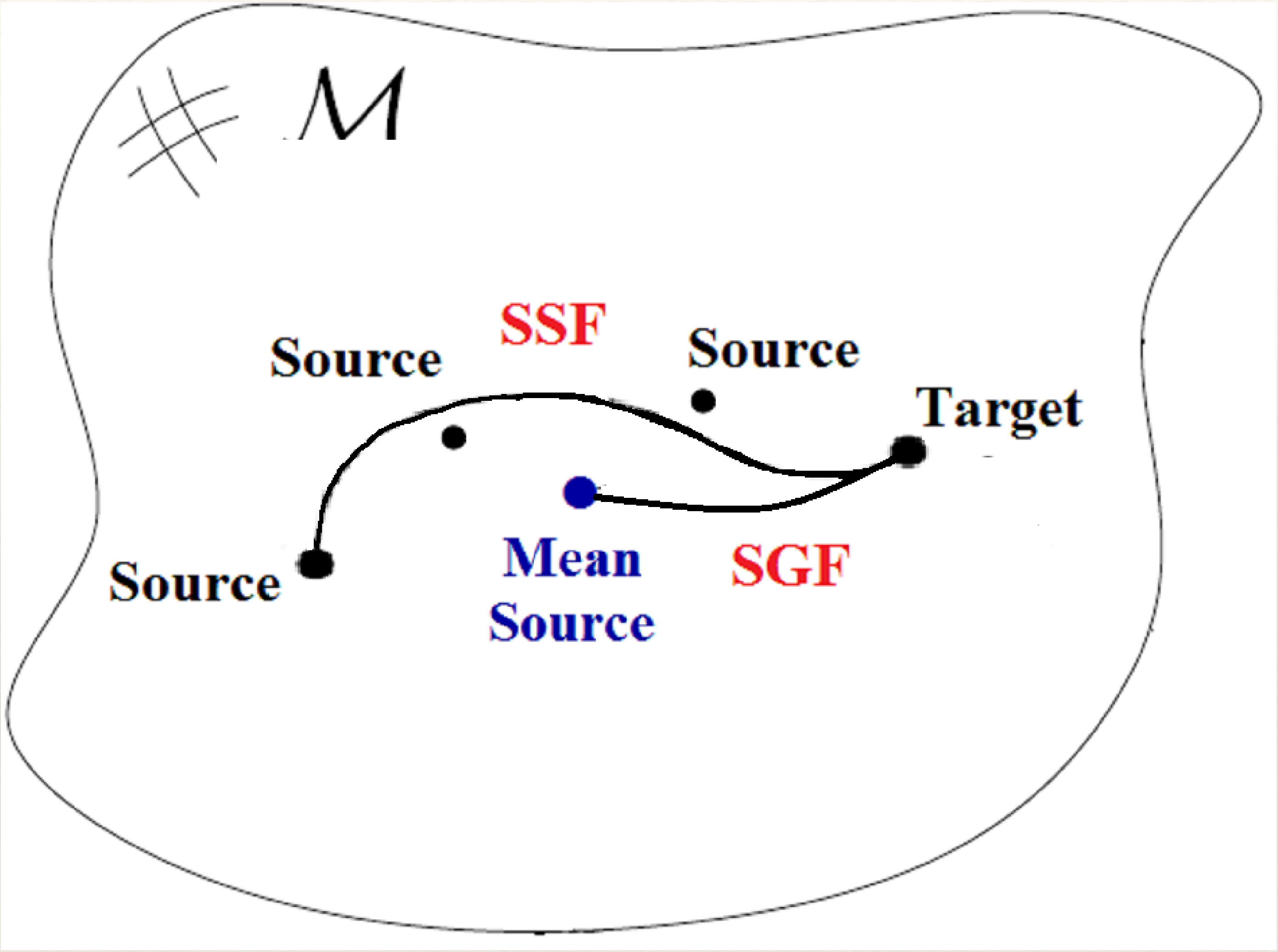
Normalized:

Method	C-->A	D-->A	W-->A	A-->C	D-->C	W-->C	A-->D	C-->D	W-->D	A-->W	C-->W	D-->W
SA-Stnd	39.97	43.58	42.20	33.20	29.36	31.55	51.41	48.30	62.17	61.89	59.94	78.03
SA-CODA	48.57	50.01	47.28	39.96	34.81	34.71	56.06	51.32	70.73	61.98	65.17	80.21
GFK-Stnd	43.50	43.69	42.75	35.62	31.52	32.51	48.90	48.21	77.16	58.62	55.63	83.26
GFK-CODA	47.91	46.18	45.11	37.22	34.60	34.68	53.11	50.97	77.06	58.54	60.89	85.23

sridhar@fovea:~/code/OptimizedDomainAdaptation-master$

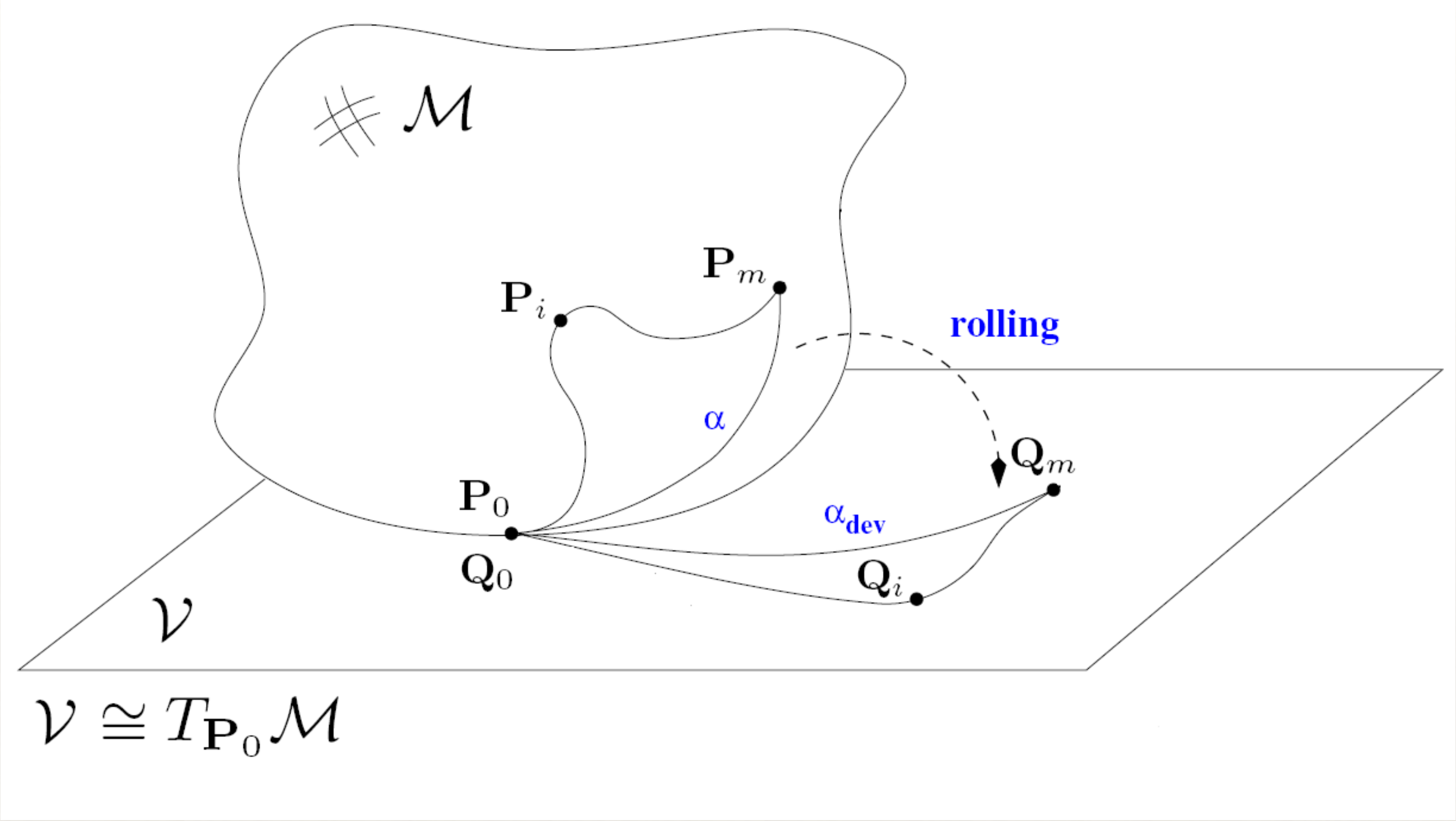
Spine flow along manifolds

How to model multiple source domains?

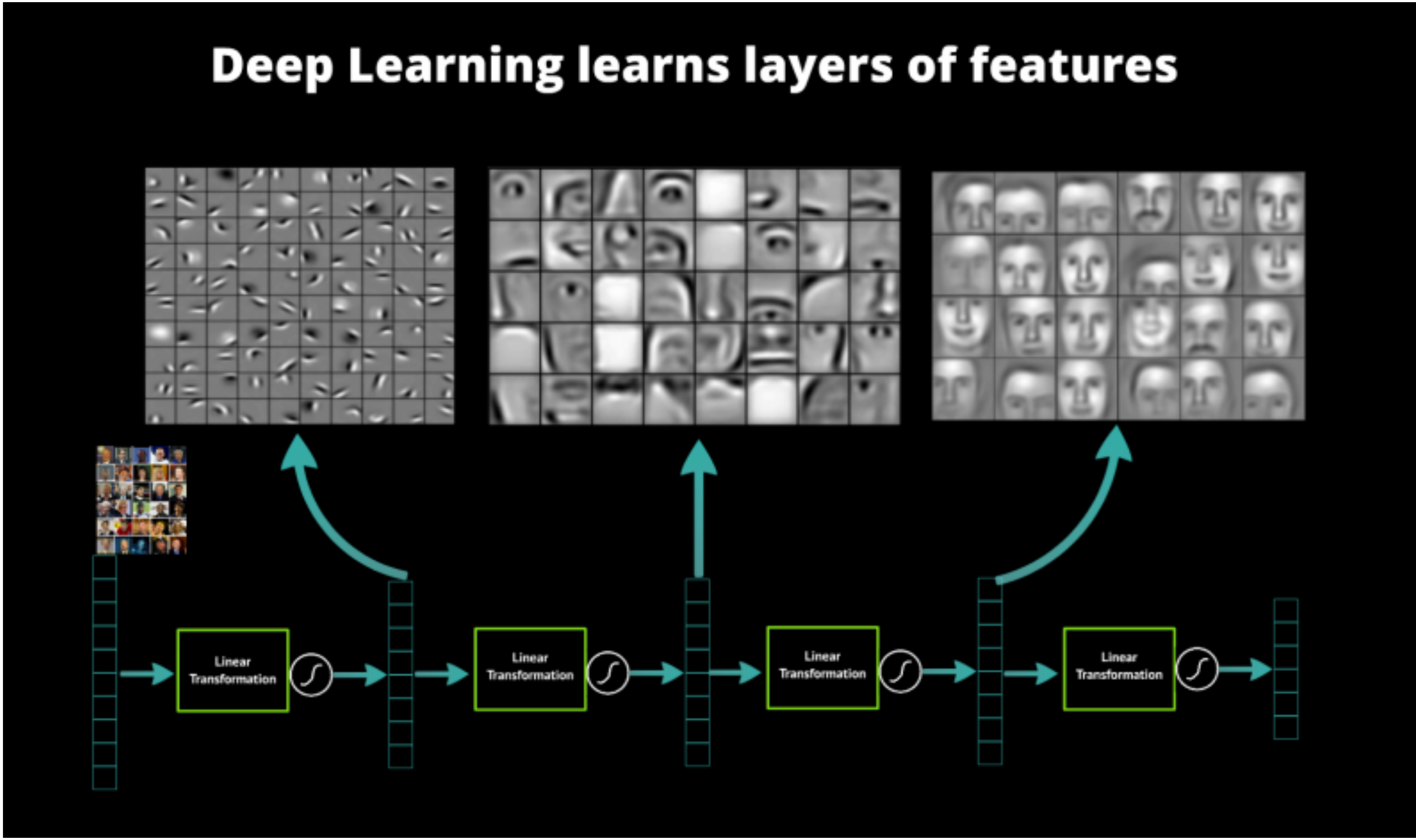


(Caseiro et al., CVPR 2015)

Rolling Riemannian Manifolds

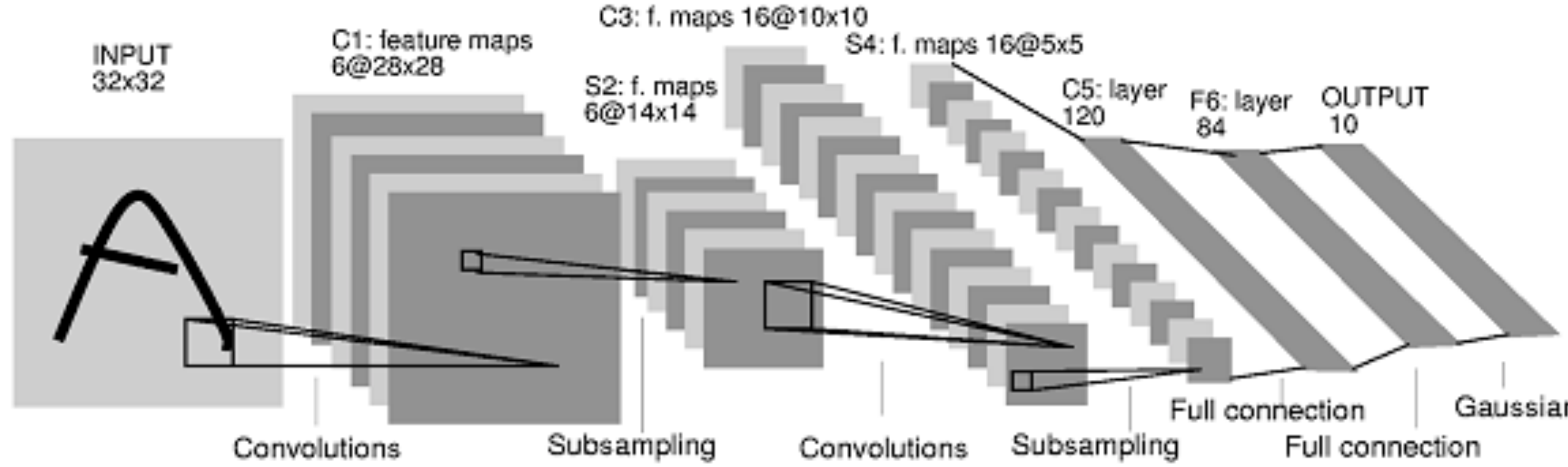


Deep Transfer Learning


Deep Learning learns layers of features
Linear Transformation
Linear Transformation
Linear Transformation
Linear Transformation

1998

LeCun et al.



# of transistors

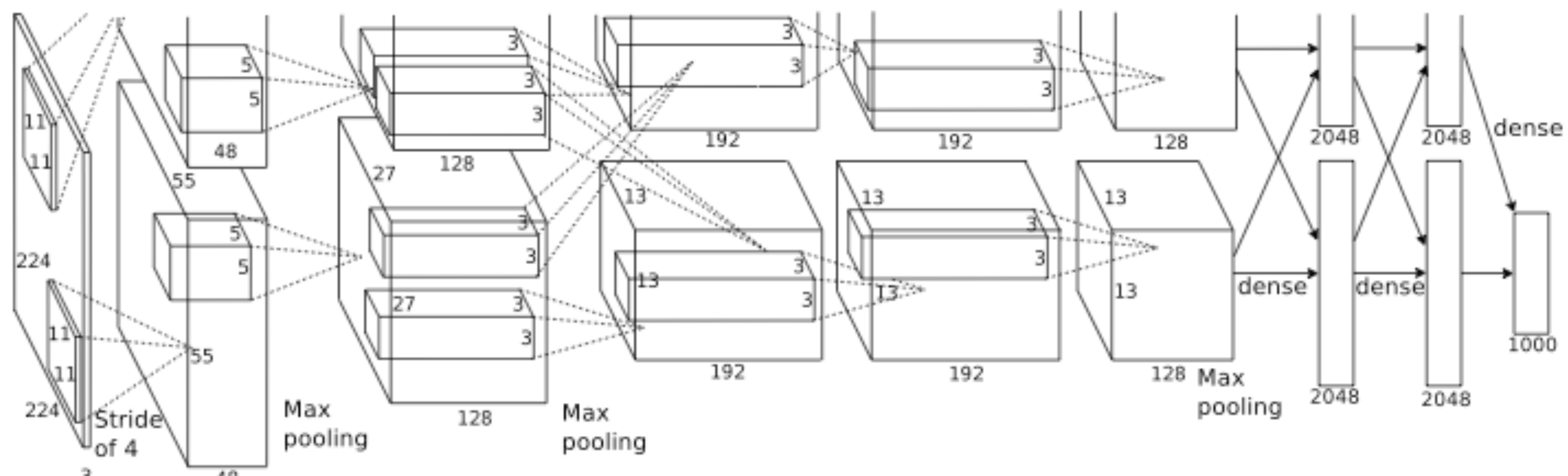
10^6

# of pixels used in training

10^7 NIST

2012

Krizhevsky et al.



# of transistors

10^9

GPUs

# of pixels used in training

10^{14} IMAGENET

Large-scale Image recognition

IMAGENET

www.image-net.org

22K categories and **14M** images

- Animals
  - Bird
  - Fish
  - Mammal
  - Invertebrate
- Plants
  - Tree
  - Flower
- Food
- Materials
- Structures
- Artifact
  - Tools
  - Appliances
  - Structures
- Person
- Scenes
  - Indoor
  - Geological Formations
- Sport Activities


flamingo
cock
ruffed grouse
quail
partridge
Egyptian cat
Persian cat
Siamese cat
tabby
lynx

IMAGENET Large Scale Visual Recognition Challenge

Year 2010

NEC-UIUC

Dense grid descriptor: HOG, LBP

↓

Coding: local coordinate, super-vector

↓

Pooling, SPM

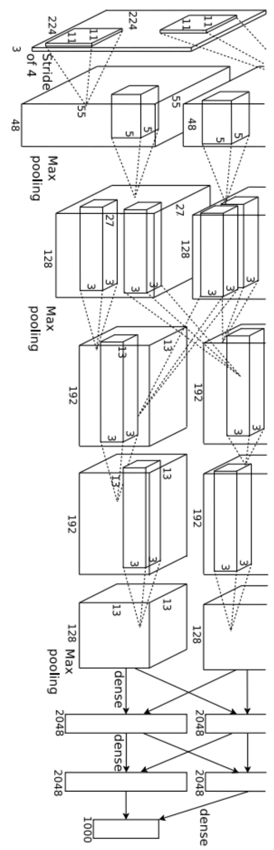
↓

Linear SVM

[Lin CVPR 2011]

Year 2012

SuperVision



[Krizhevsky NIPS 2012]

Year 2014

GoogLeNet

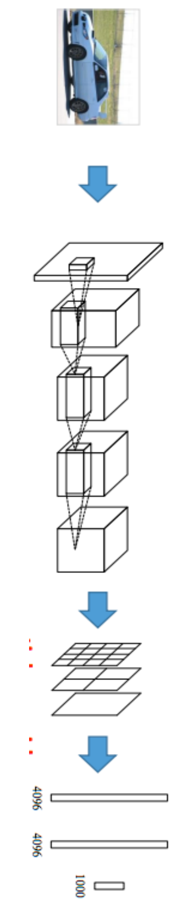
[Szegedy arxiv 2014]

VGG

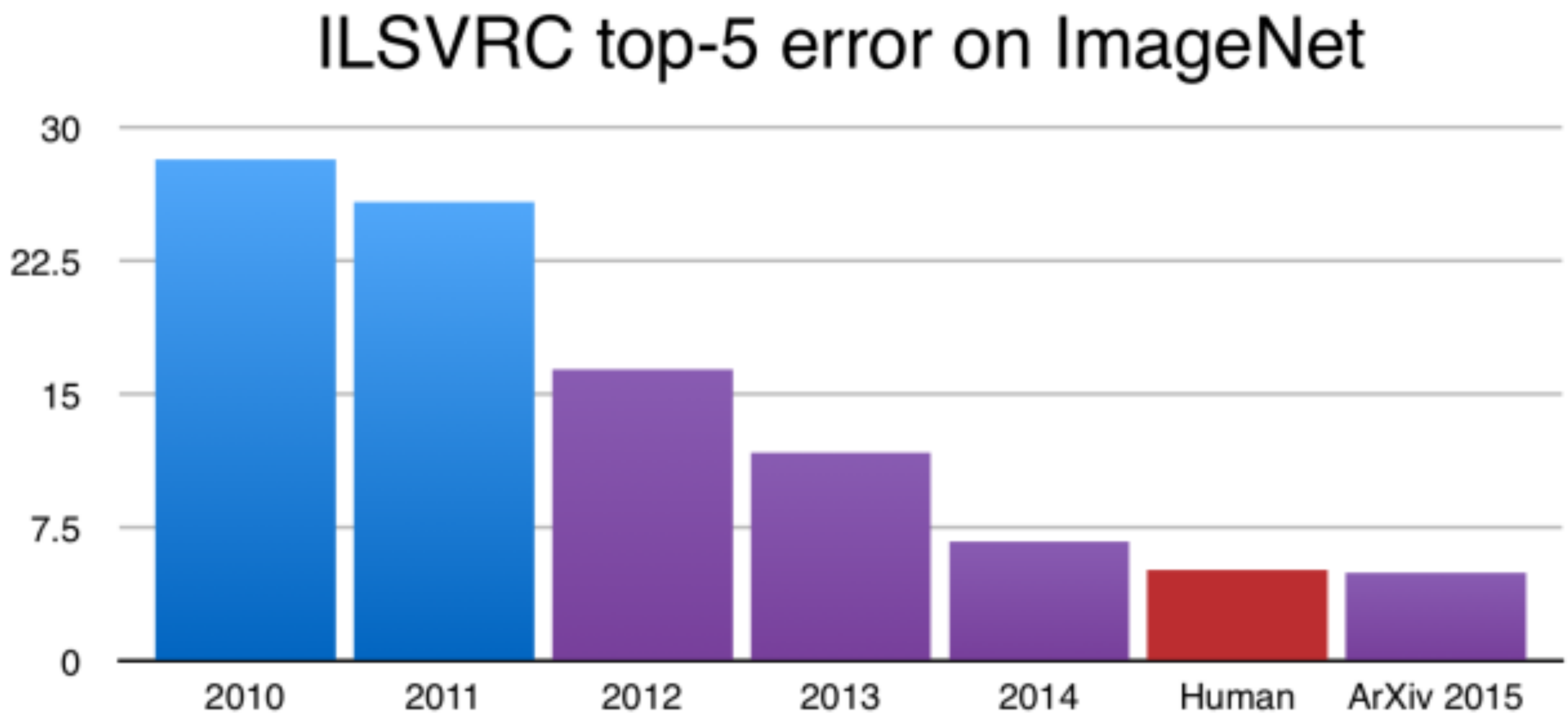
image
conv-64
conv-64
maxpool
conv-128
conv-128
maxpool
conv-256
conv-256
maxpool
conv-512
conv-512
maxpool
conv-512
conv-512
maxpool
FC-4096
FC-4096
FC-1000
softmax

[Simonyan arxiv 2014]

MSRA

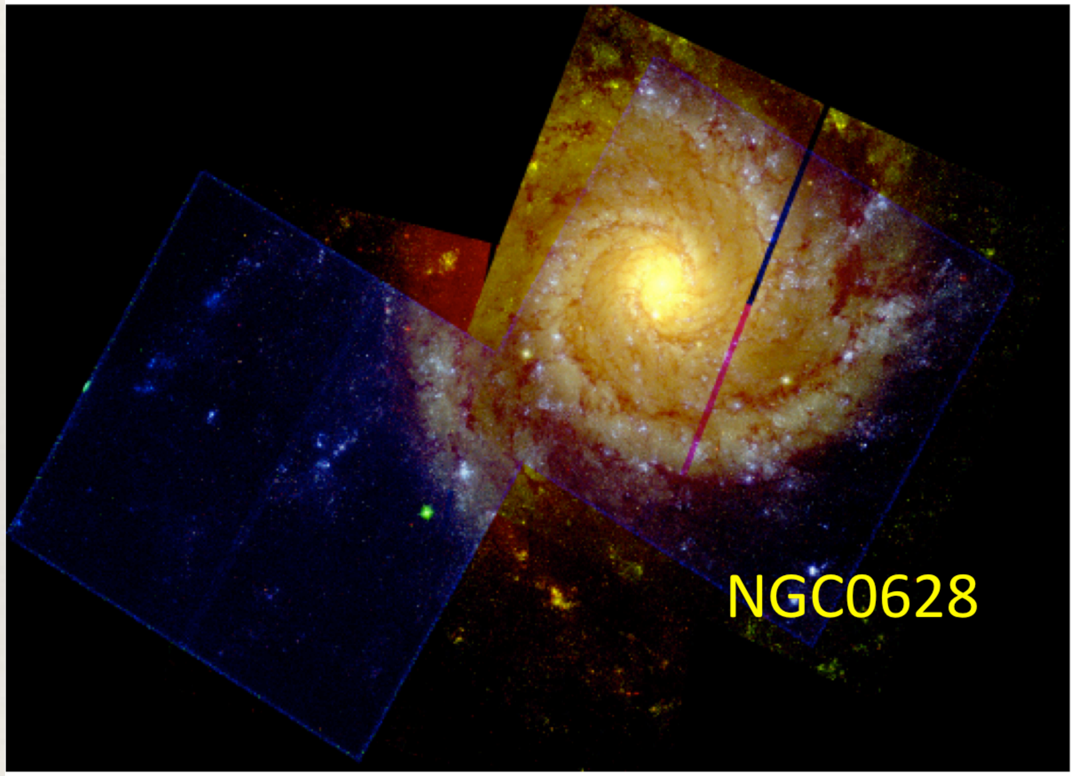


[He arxiv 2014]


ILSVRC top-5 error on ImageNet
30
22.5
15
7.5
0
2010
2011
2012
2013
2014
Human
ArXiv 2015

Classifying Objects in Hubble Images

Ongoing project with Professor Daniela Calzetti, UMass Astronomy



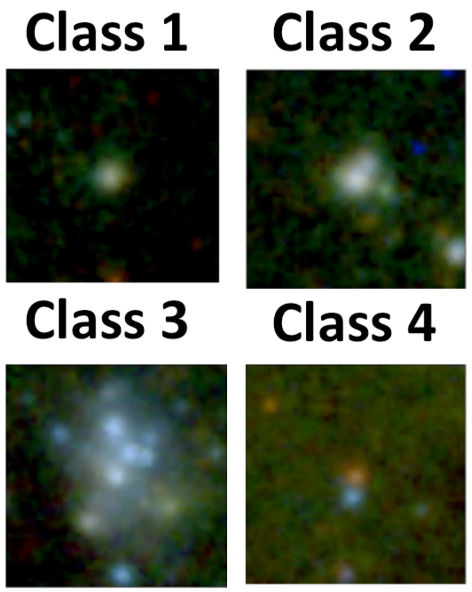
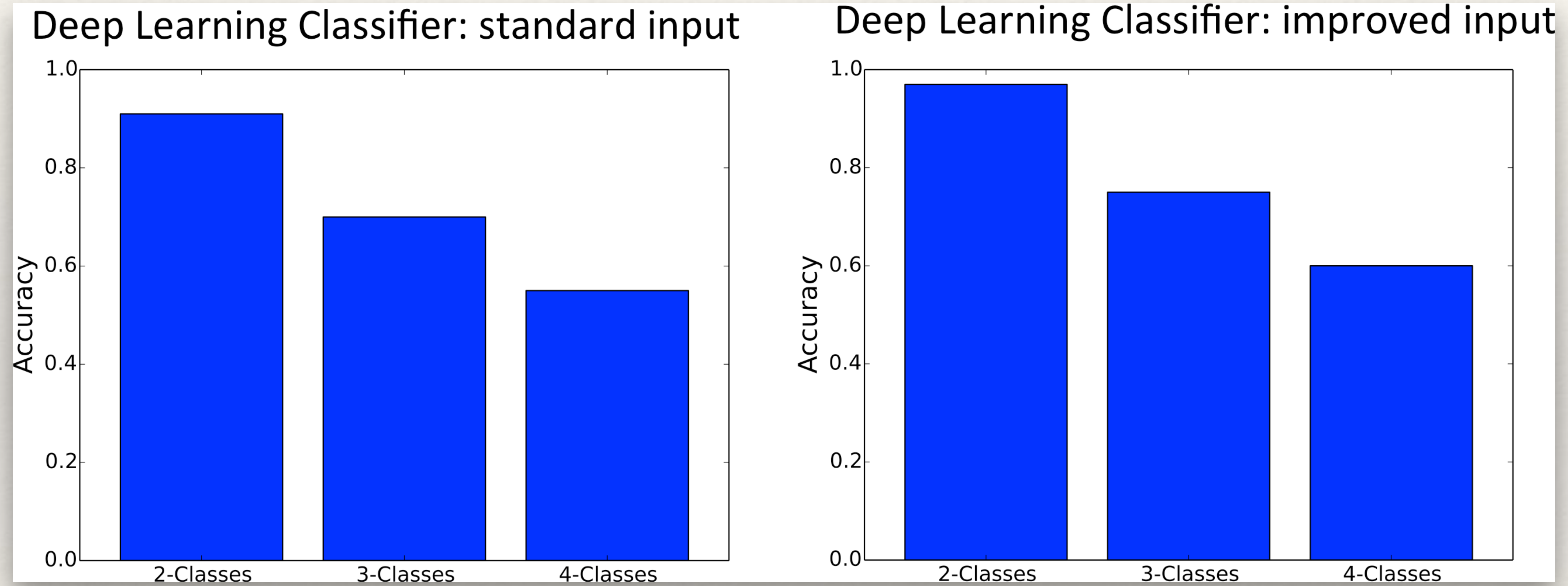


Table 1

Galaxy	Type	# Clusters
NGC0628	SAc	~1,500
NGC1313	SBd	~1,800
NGC1566	SABbc	~2,100
NGC3344	SABbc	~400
NGC3738	Im	~400
NGC4449	IBm	~600*
NGC5194	SAbc	~3,000*
NGC5253	Im	~150*
NGC7793	SAd	~300
~12 Dwarfs	Irr	~300*

Hubble Classification using Deep Learning



Transfer Learning with CNNs

1. Train on Imagenet

2. If small dataset: fix all weights (treat CNN as fixed feature extractor), retrain only the classifier

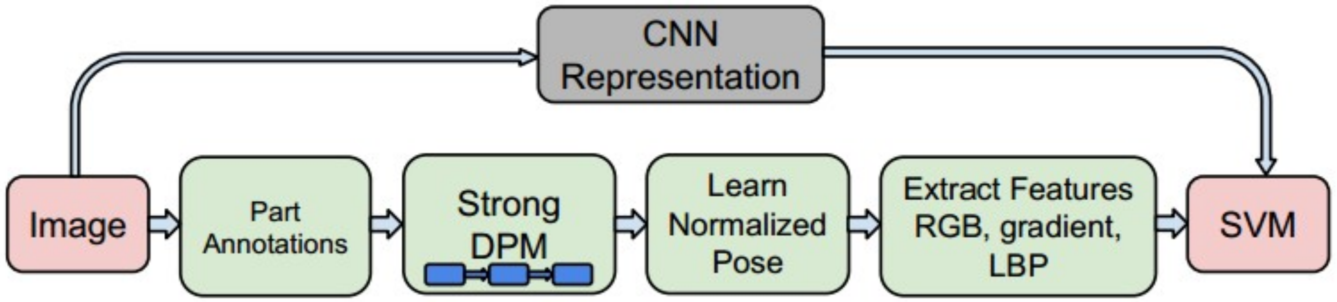
i.e. swap the Softmax layer at the end

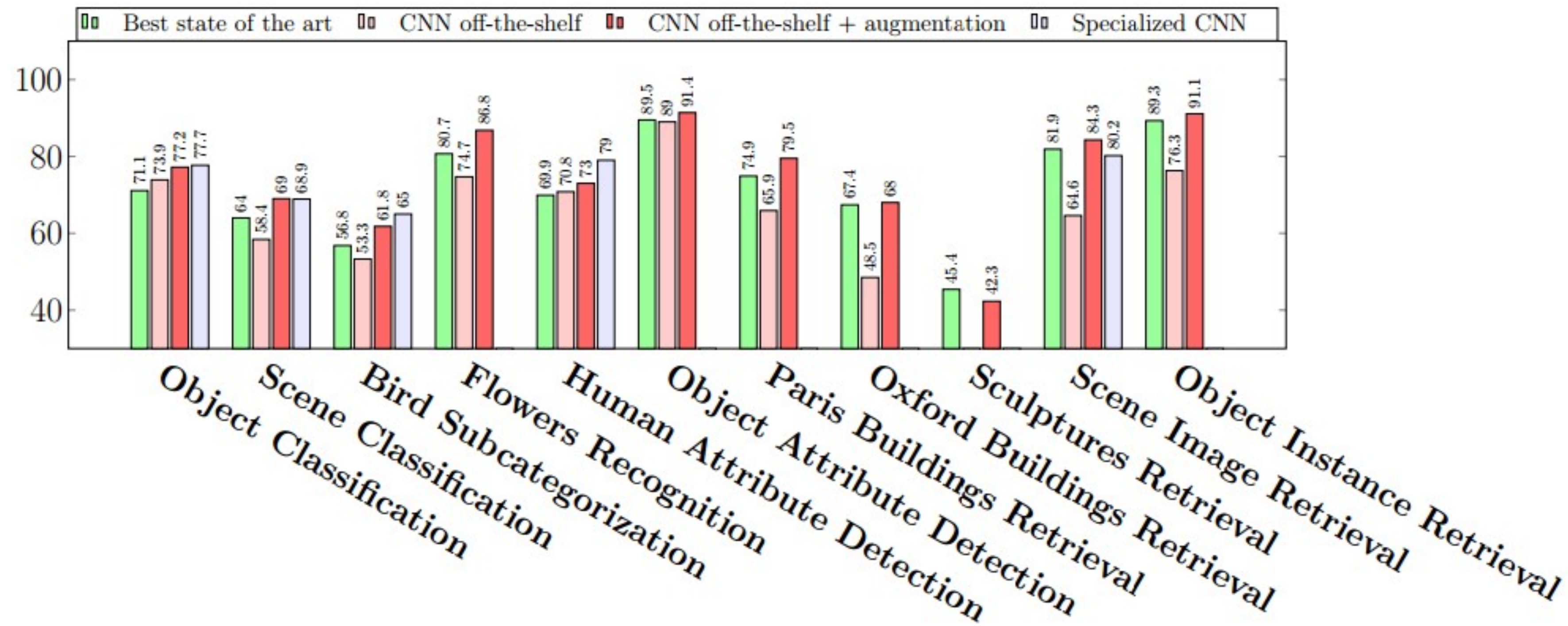
3. If you have medium sized dataset, **"finetune"** instead: use the old weights as initialization, train the full network or only some of the higher layers

retrain bigger portion of the network, or even all of it.

Slide courtesy of Fei Fei Li and Andrej Karpathy

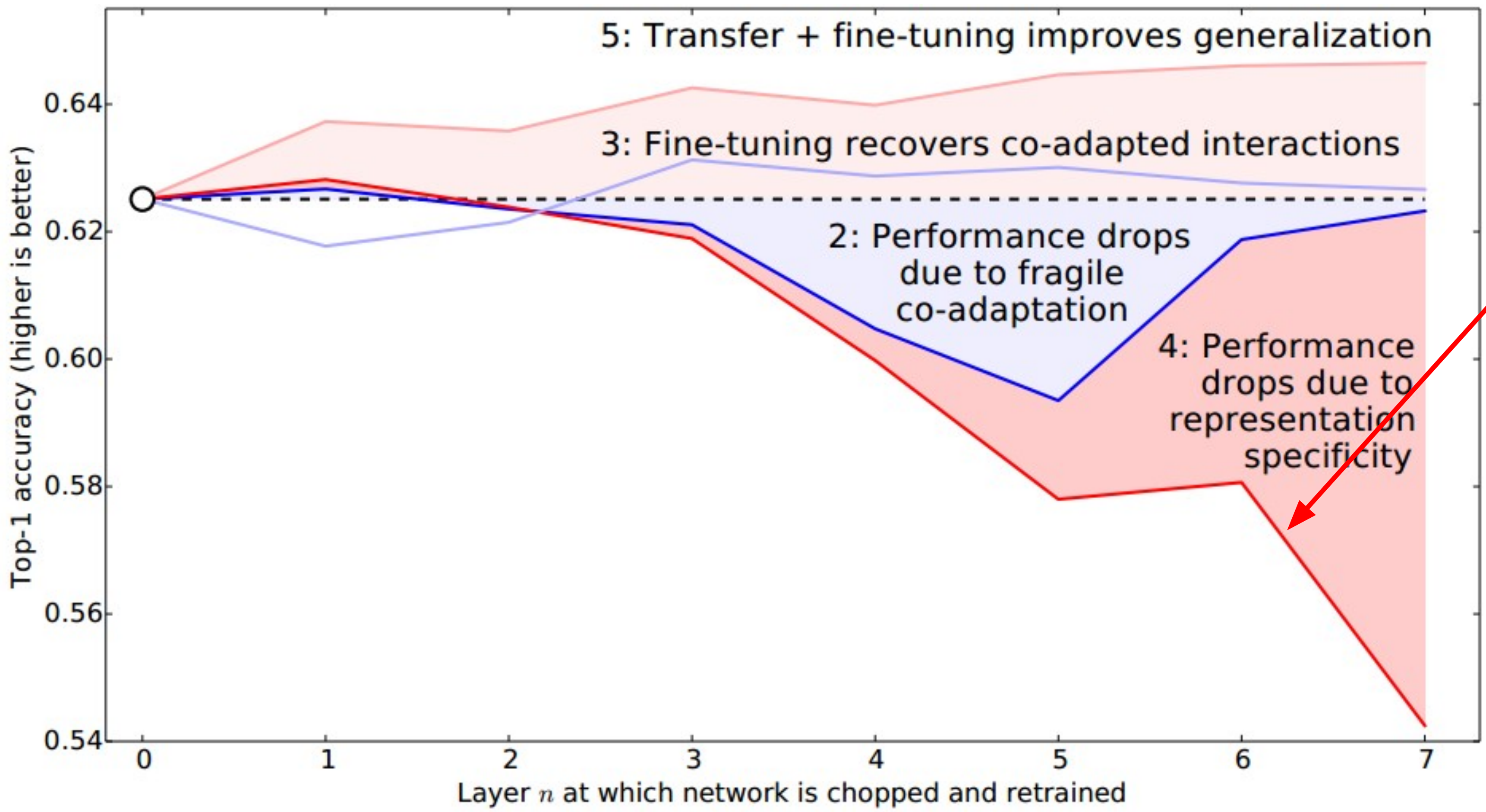
CNN Features off-the-shelf: an Astounding Baseline for Recognition
[Razavian et al, 2014]





Slide courtesy of Fei Fei Li and Andrej Karpathy

How transferable are features in deep neural networks?
[Yosinski et al., 2014]



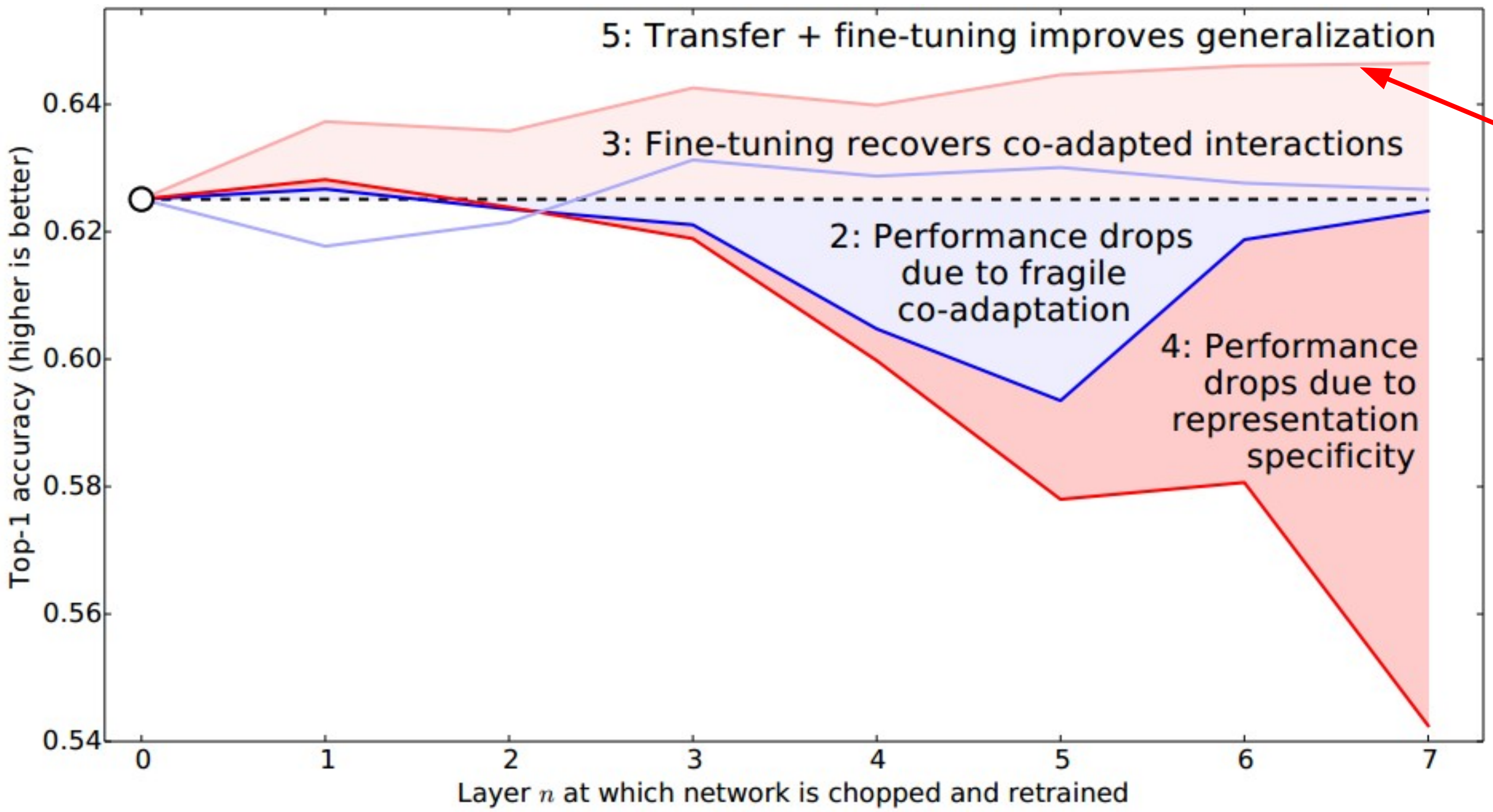
Split ImageNet classes in half to two sets: A/B.

Train on A, fix the first n layers, reinit layers n+, train on B, test on B val.

=> performance degrades because representation higher up is too A-specific

Slide courtesy of Fei Fei Li and Andrej Karpathy

How transferable are features in deep neural networks?
[Yosinski et al., 2014]

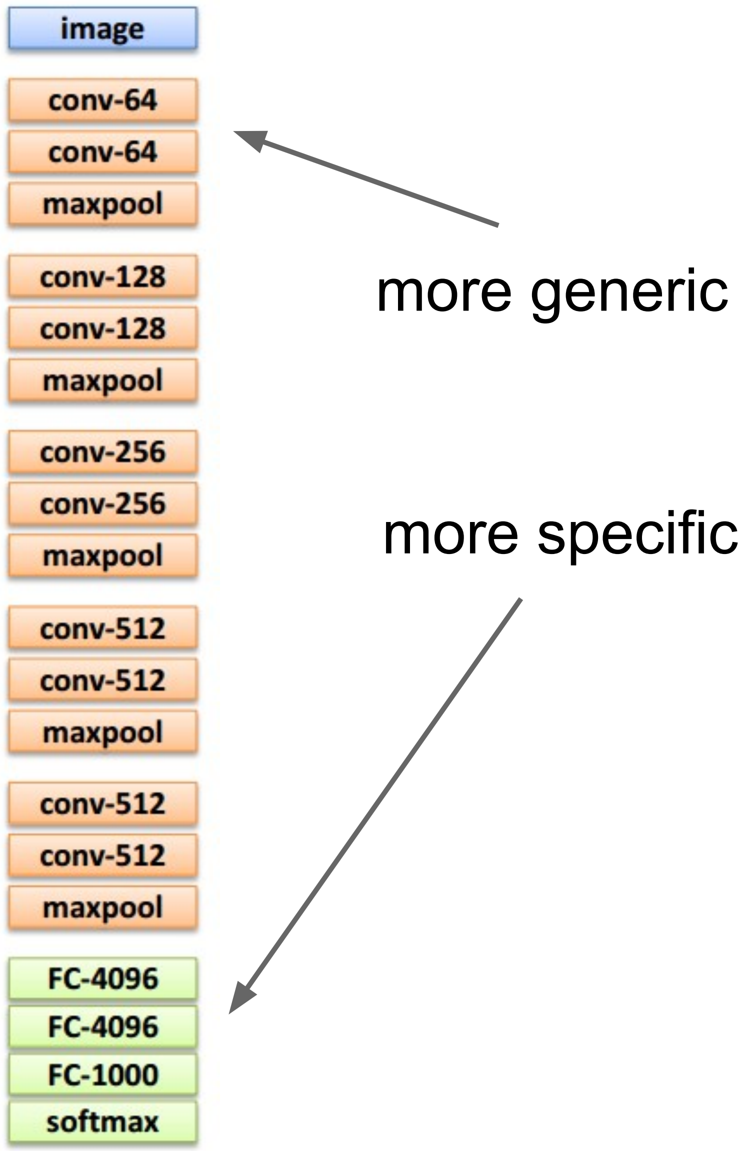


Split ImageNet classes in half to two sets: A/B.

Train on A, reinit layers n+, train on B, test on B val.

=> the information from once seeing data from A seems to linger, gives better generalization

Slide courtesy of Fei Fei Li and Andrej Karpathy



	very similar dataset	**very different dataset**
very little data	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

Slide courtesy of Fei Fei Li and Andrej Karpathy

Stacked auto encoders

- Auto encoders are deep learning networks that learn to reproduce their inputs
- The idea is to find a low-dimensional compression of the input
- They can be applied to domain adaptation and transfer learning by giving them unlabeled source and target examples as input
- Denoising stacked auto encoders are given noisy inputs and required to reproduce the noiseless version

Linear Denoising AutoEncoder

(Chen et al., ICML 2012)

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n] \in \mathcal{R}^{d \times n}$$

$$\mathcal{L}_{sq}(\mathbf{W}) = \frac{1}{2mn} \sum_{j=1}^{m} \sum_{i=1}^{n} \|\mathbf{x}_i - \mathbf{W}\tilde{\mathbf{x}}_{i,j}\|^2$$

uncorrupted input

corrupted input

$$\mathcal{L}_{sq}(\mathbf{W}) = \frac{1}{2nm} \mathrm{tr}\left[\left(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}}\right)^\top \left(\overline{\mathbf{X}} - \mathbf{W}\widetilde{\mathbf{X}}\right)\right]$$

m copies of input X

$$\mathbf{W} = \mathbf{P}\mathbf{Q}^{-1} \text{ with } \mathbf{Q} = \widetilde{\mathbf{X}}\widetilde{\mathbf{X}}^\top \text{ and } \mathbf{P} = \overline{\mathbf{X}}\widetilde{\mathbf{X}}^\top$$

Marginalized Stacked DA

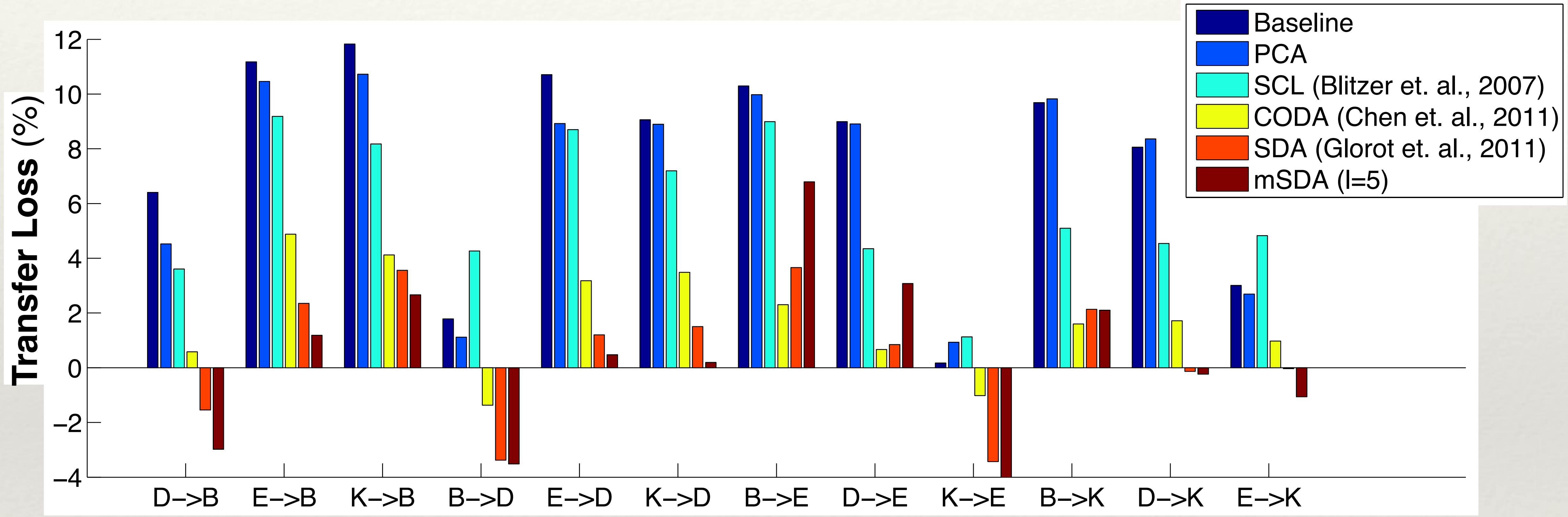
$$\mathbf{W} = E[\mathbf{P}]E[\mathbf{Q}]^{-1}.$$

$$E[\mathbf{Q}] = \sum_{i=1}^{n} E\left[\tilde{\mathbf{x}}_i \tilde{\mathbf{x}}_i^\top\right]$$

$$\mathbf{S} = \mathbf{X}\mathbf{X}^\top$$

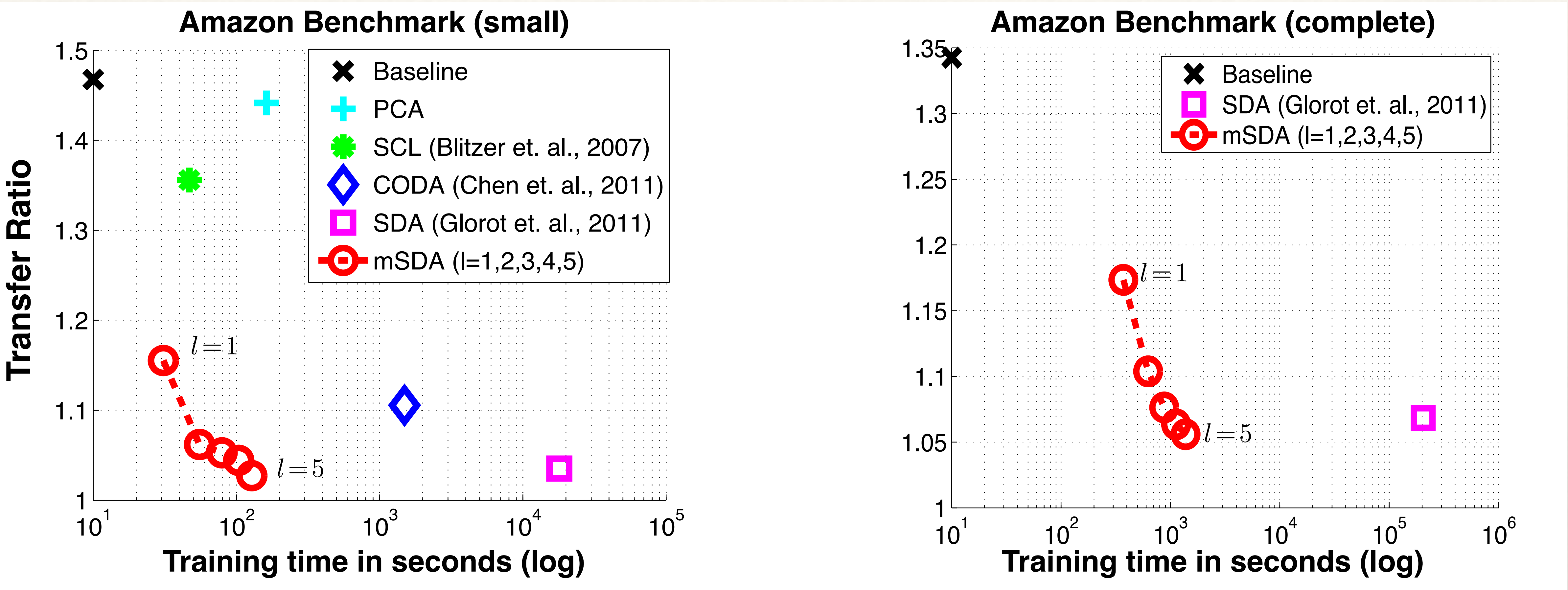
$$E[\mathbf{Q}]_{\alpha,\beta} = \begin{cases} \mathbf{S}_{\alpha\beta}\mathbf{q}_\alpha\mathbf{q}_\beta & \text{if} \quad \alpha \neq \beta \\ \mathbf{S}_{\alpha\beta}\mathbf{q}_\alpha & \text{if} \quad \alpha = \beta \end{cases}$$

MSDA Results



Amazon sentiment analysis dataset

MSDA vs. SDA

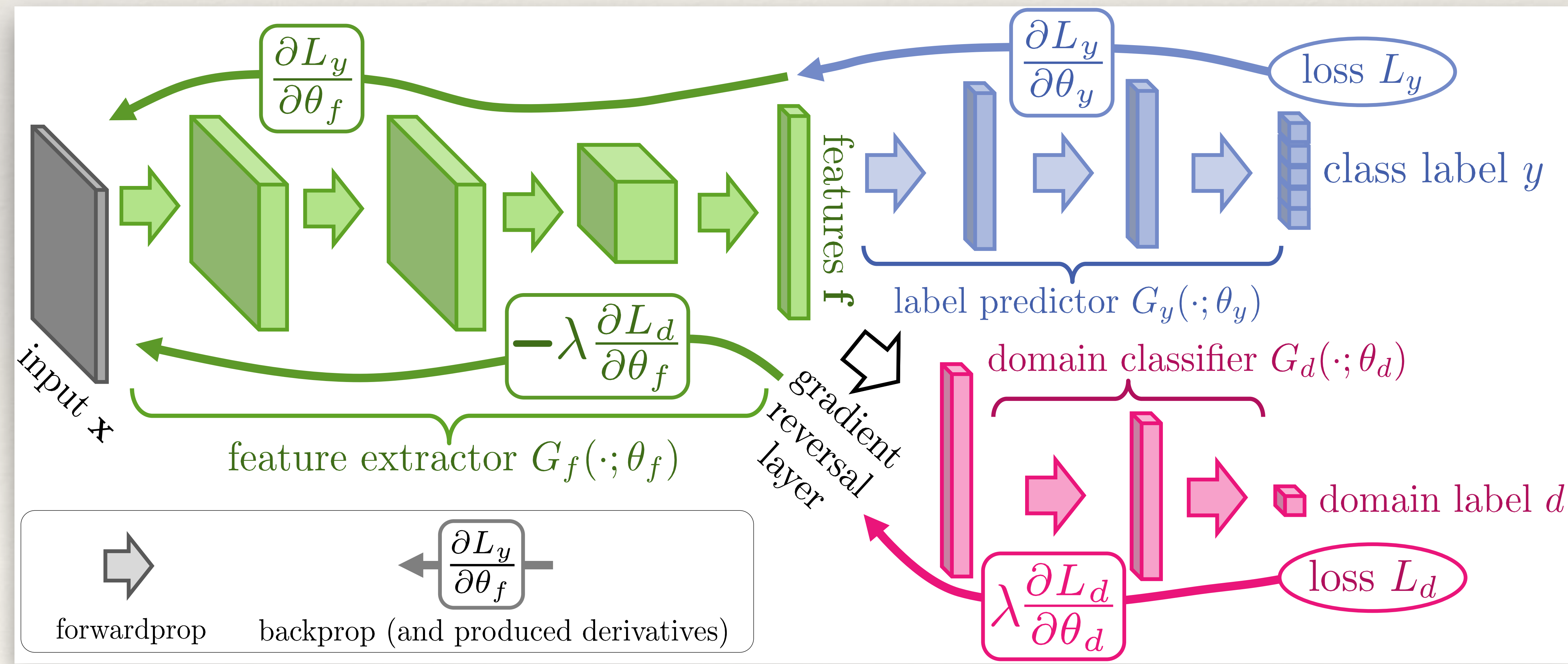


Amazon Benchmark (small)
Transfer Ratio
Training time in seconds (log)
Baseline
PCA
SCL (Blitzer et. al., 2007)
CODA (Chen et. al., 2011)
SDA (Glorot et. al., 2011)
mSDA (l=1,2,3,4,5)
l=1
l=5
Amazon Benchmark (complete)
Training time in seconds (log)
Baseline
SDA (Glorot et. al., 2011)
mSDA (l=1,2,3,4,5)
l=1
l=5

Generative Adversarial Networks

(Goodfellow et al., NIPS 2014; Ganlin, et al., JMLR 2016)

"For effective domain transfer to be achieved, predictions must be made based on features that cannot discriminate between the training (source) and test (target) domains."



GAN objective function

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

  for k steps **do**

  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

  • Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.

  • Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right].$$

  end for

  • Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.

  • Update the generator by descending its stochastic gradient:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

Results on Amazon Sentiment Analysis

		Original data			mSDA representation		
Source	Target	DANN	NN	SVM	DANN	NN	SVM
books	dvd	.784	.790	**.799**	.829	.824	**.830**
books	electronics	.733	.747	**.748**	**.804**	.770	.766
books	kitchen	**.779**	.778	.769	**.843**	.842	.821
dvd	books	.723	.720	**.743**	.825	.823	**.826**
dvd	electronics	**.754**	.732	.748	**.809**	.768	.739
dvd	kitchen	**.783**	.778	.746	.849	**.853**	.842
electronics	books	**.713**	.709	.705	**.774**	.770	.762
electronics	dvd	**.738**	.733	.726	**.781**	.759	.770
electronics	kitchen	**.854**	**.854**	.847	.881	**.863**	.847
kitchen	books	**.709**	.708	.707	.718	.721	**.769**
kitchen	dvd	**.740**	.739	.736	**.789**	**.789**	.788
kitchen	electronics	**.843**	.841	.842	.856	.850	**.861**

(a) Classification accuracy on the Amazon reviews data set

Summary

- Transfer learning is a broad topic that has been studied for many decades
- Classical approaches:
  - **Structure mapping** finds a way to transfer relationships from source to target
  - **Determination rules** provide a logical formulation for transfer learning

Summary

- Statistical Approaches:
  - **Canonical correlational analysis** (CCA) finds a lower-dimensional subspace where projected source and target vectors are maximally correlated
  - **Manifold alignment** generalizes CCA to unlabeled data and also enables its use for data that lies on a manifold

Summary

- Subspace identification:
  - **Subspace alignment** finds a linear transformation that makes the source look like the target
  - **Geodesic flow kernels** find the shortest path geodesic on the Grassmannian manifold from source subspace to target subspace
  - Correspondence o**ptimized domain-invariant projection** provides a way to choose the source and target subspaces using a small number of correspondences

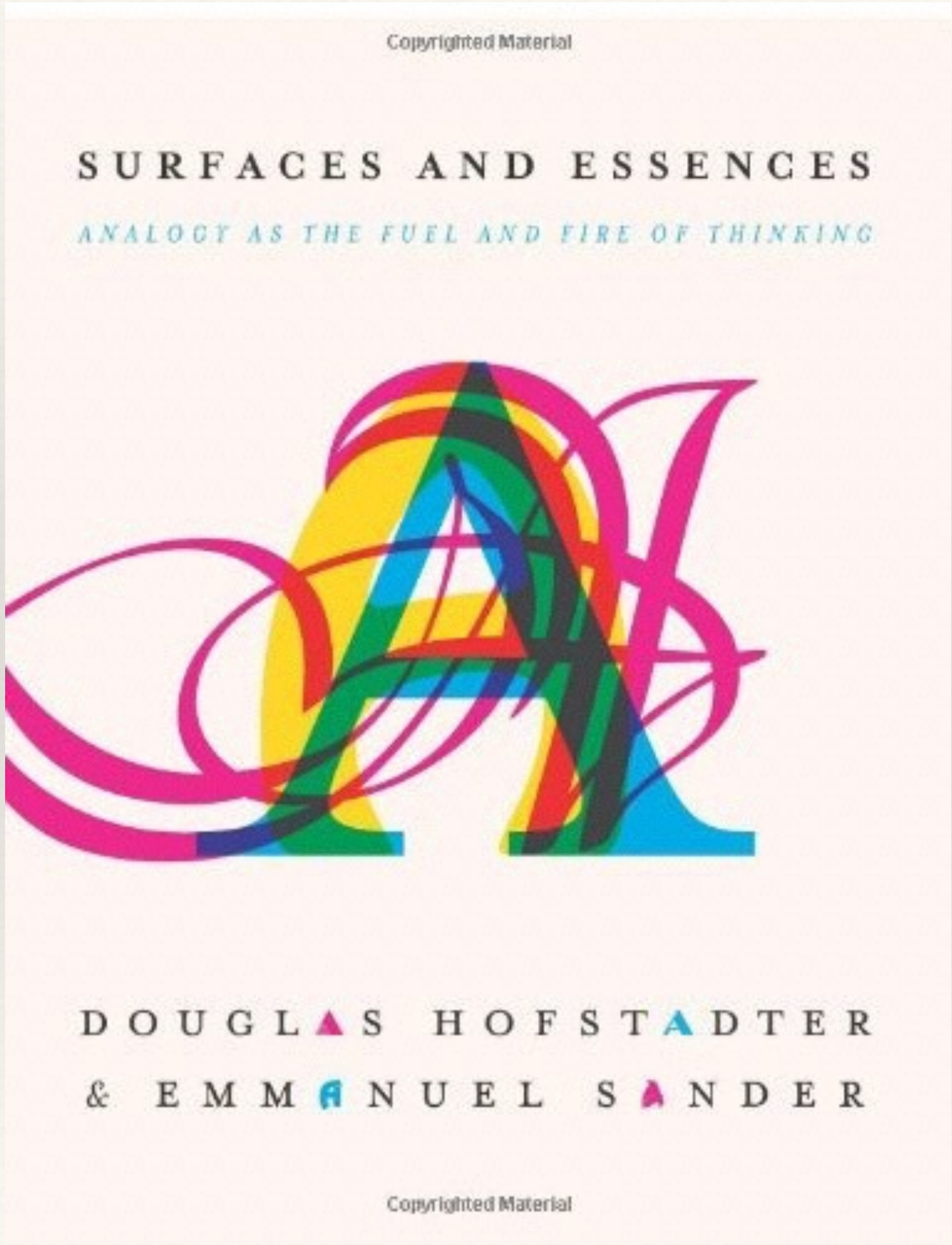
Summary

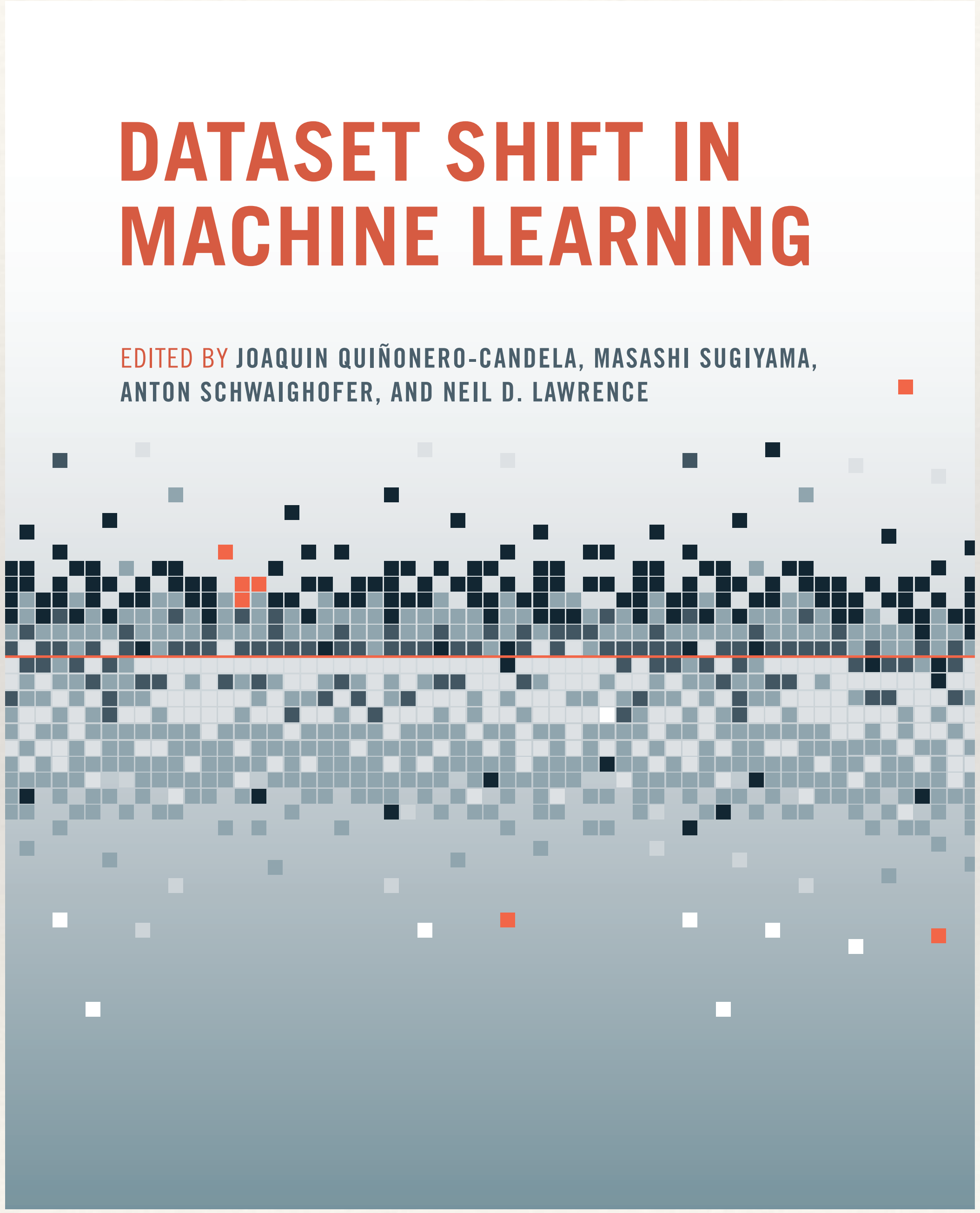
- Deep transfer learning:
  - Train a deep neural network on data (e.g., Imagenet)
  - Reuse some of the weights from the first N convolutional layers and retrain the subsequent layers
  - **Stacked denoising auto encoders** (SDA) are multi-level networks that learn to reproduce an uncorrupted version of a set of noisy input examples
  - **Marginalized SDAs** are a hybrid linear-nonlinear approach where the linear weights are trained using least-squares
  - Generative adversarial networks find a representation where source and target data look indistinguishable

Future Challenges

- Transfer learning admits a plethora of approaches, but lacks a clear unifying framework
- Two major themes:
  - Find **correlations** between source and target (CCA)
  - Find **symmetries** across source and target (CNNs)
- More sophisticated ideas from group representations can be used
  - Generalization of CNNs that extract deeper symmetries

Background Reading





Background Reading

Chang Wang and Sridhar Mahadevan, " Manifold Alignment Preserving Global Geometry ", Proceedings of the IJCAI Conference, August 3-9, 2013, Beijing, China.

Hoa Vu, CJ Carey, and Sridhar Mahadevan, " Manifold Warping: Manifold Alignment over Time ", Proceedings of the 26th Conference on Artificial Intelligence (AAAI), July 22-26, 2012, Toronto, Canada.

Chang Wang and Sridhar Mahadevan, " Manifold Alignment Preserving Global Geometry ", Technical Report, UMass Computer Science Department UM-CS-2012-031, 2012.

Chang Wang, Bo Liu, Hoa Vu, and Sridhar Mahadevan, " Sparse Manifold Alignment ", Technical Report, UMass Computer Science UM-2012-030, 2012.

Chang Wang and Sridhar Mahadevan, " Heterogeneous Domain Adaptation using Manifold Alignment ", Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), July 18-23, 2011, Barcelona, Spain.

Chang Wang and Sridhar Mahadevan, " Jointly Learning Data-Depdendent Label and Locality-Preserving Projections ", Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), July 18-23, 2011, Barcelona, Spain.

Chang Wang, Peter Krafft, and Sridhar Mahadevan, " Manifold Alignment ", appearing in Manifold Learning: Theory and Applications, Taylor and Francis CRC Press.

Chang Wang and Sridhar Mahadevan, "Multiscale Manifold Alignment" , Univ. of Massachusetts TR UM-CS-2010-049, 2010.

Chang Wang and Sridhar Mahadevan, "Learning Locality Preserving Discriminative Features" , Univ. of Massachusetts TR UM-CS-2010-048, 2010.

Sridhar Mahadevan and Prasad Tadepalli, "Quantifying Prior Determination Knowledge using the PAC Learning Model", *Machine Learning* , vol. 17, pp. 69-105, 1994