TBSI 清华–伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

Using Maximum Correlation for Transferability Estimation and Multi-Modal Learning

Yang Li

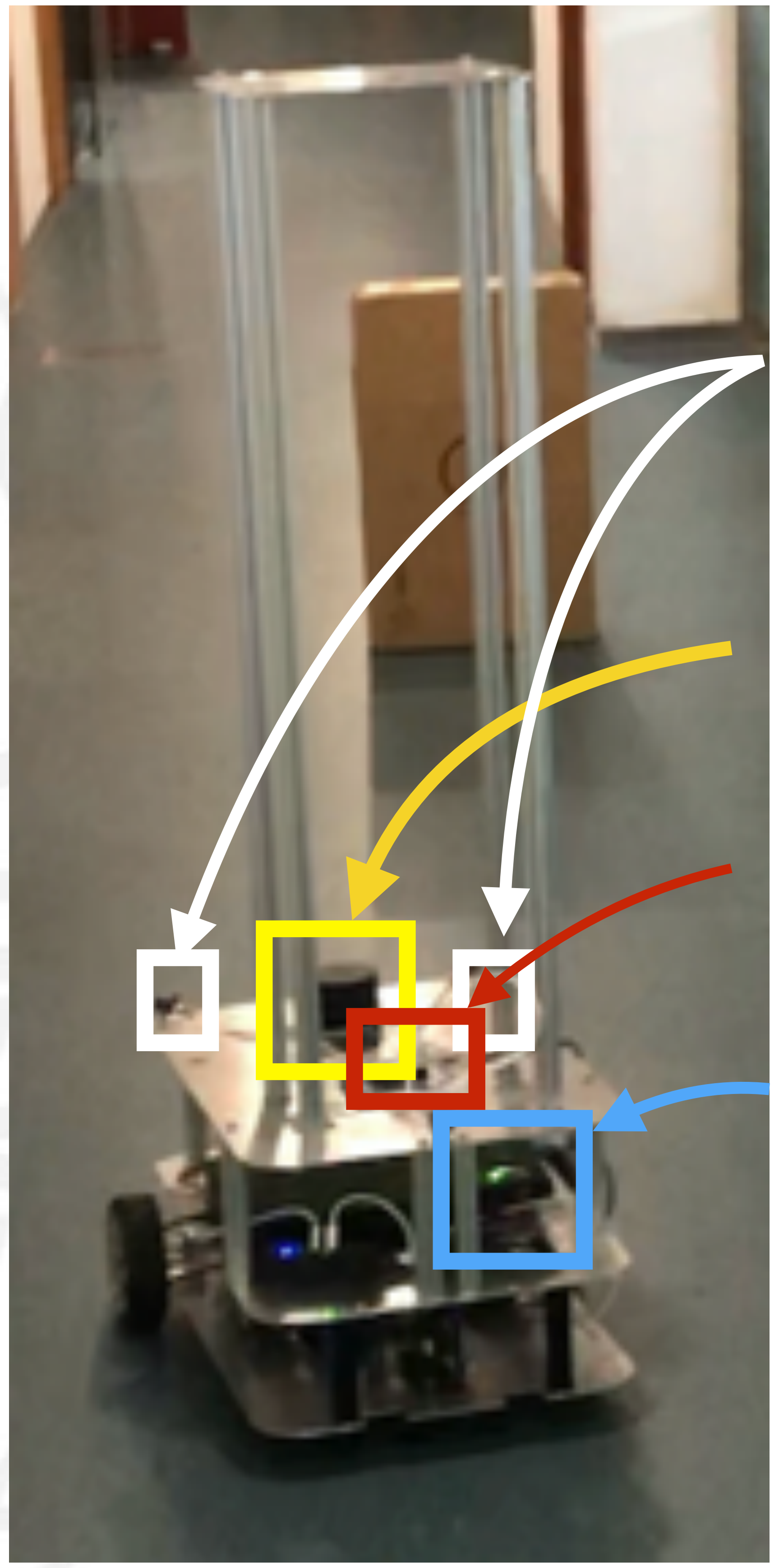
Center of Data Science and Information Technology

Tsinghua-Berkeley Shenzhen Institute

June 19, 2019, Texas A&M University

Machine Learning in the Wild

Example: A robotic tour guide



Sonar

Lidar

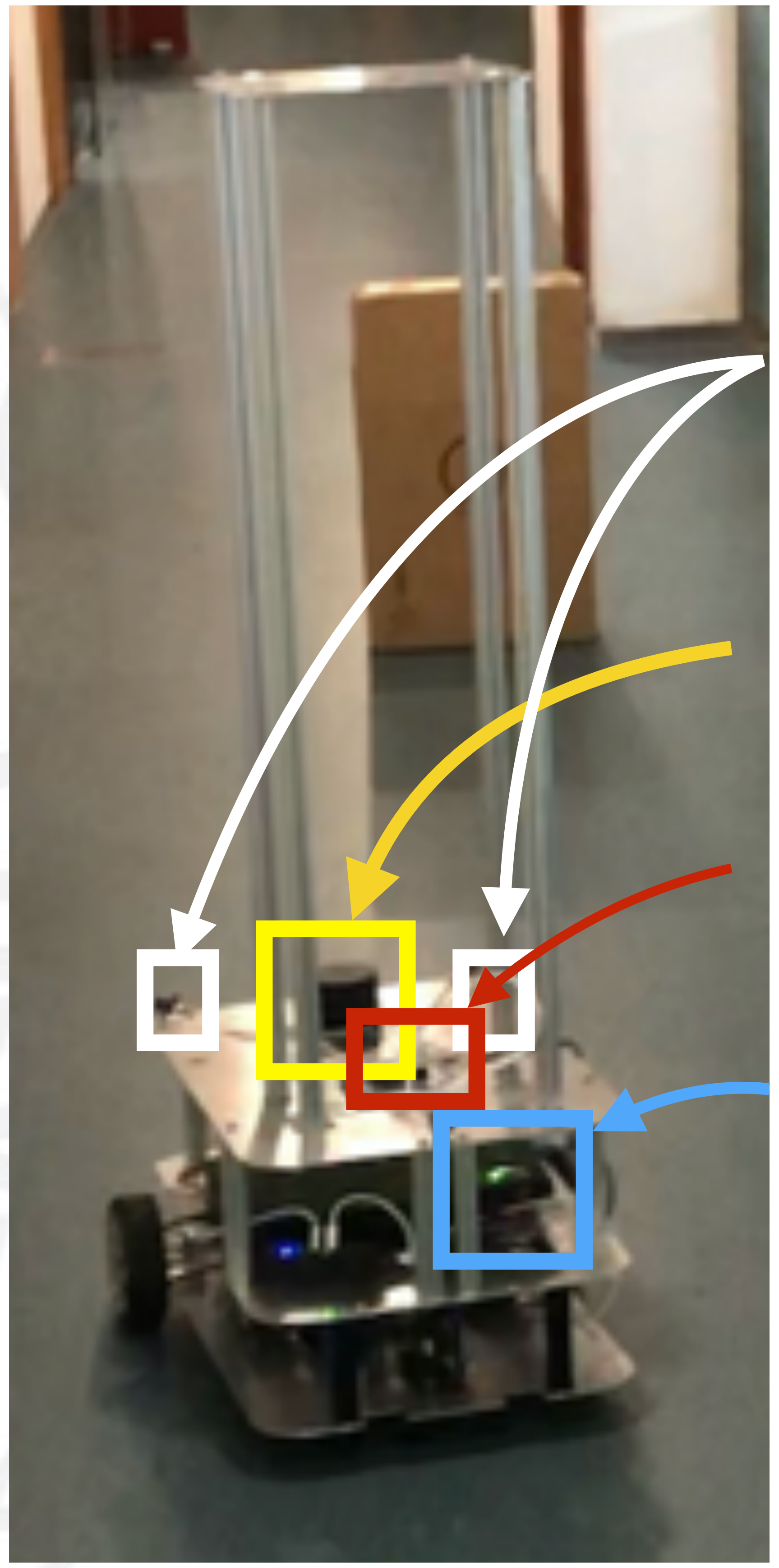
Microphone Array

Camera

Machine Learning in the Wild

Example: A robotic tour guide

- Need to solve many learning tasks

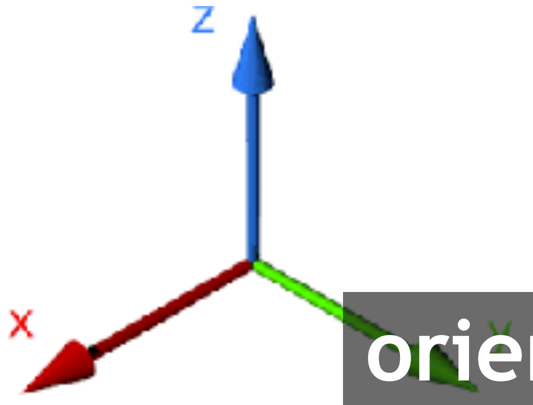


Sonar

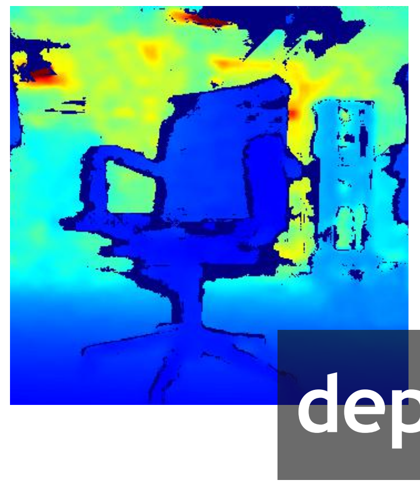
Lidar

Microphone Array

Camera



orientation estimation



depth estimation

identification

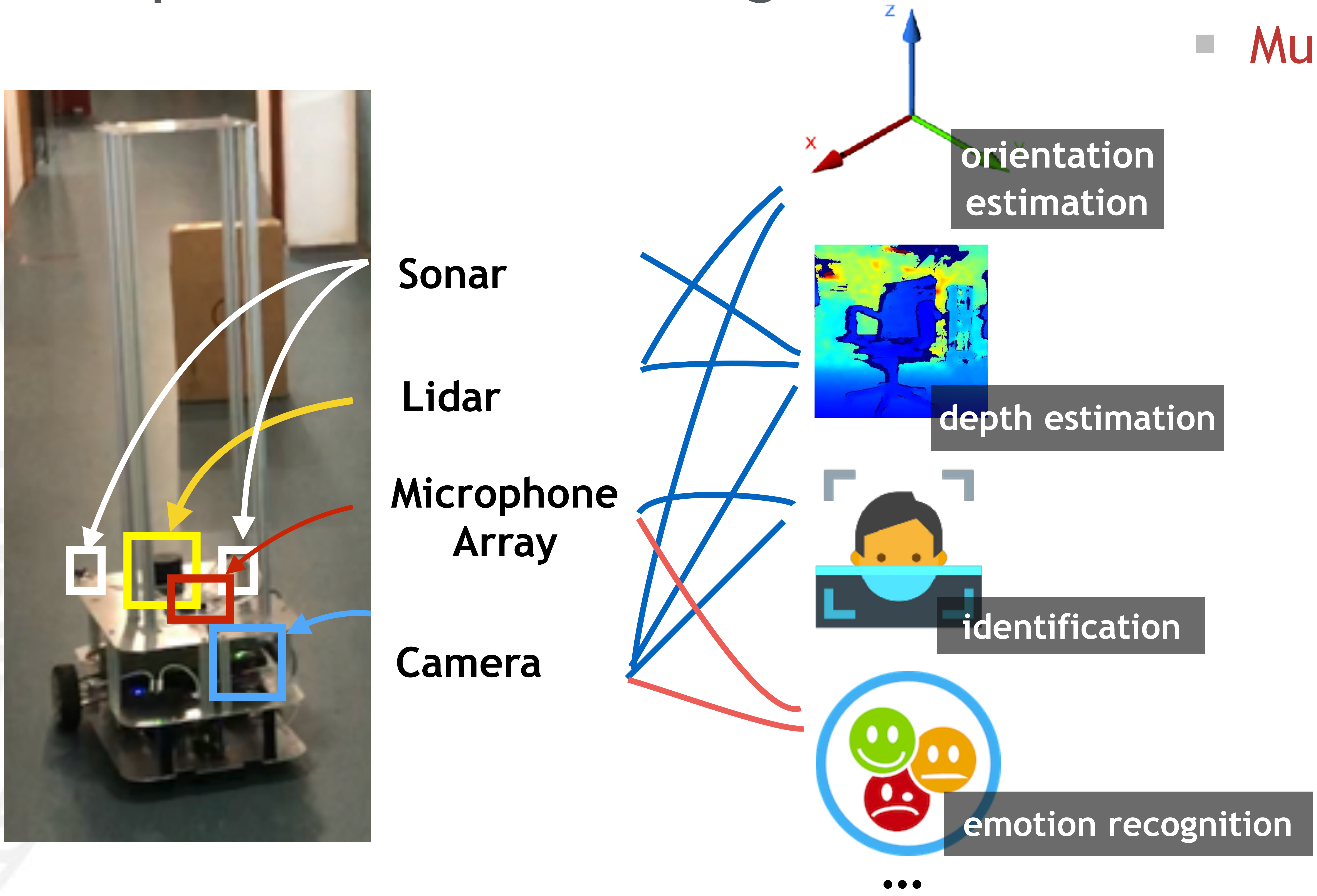
emotion recognition

…

Machine Learning in the Wild

Example: A robotic tour guide

- Need to solve many learning tasks
- Multiple data sources



Sonar

Lidar

Microphone Array

Camera

orientation estimation

depth estimation

identification

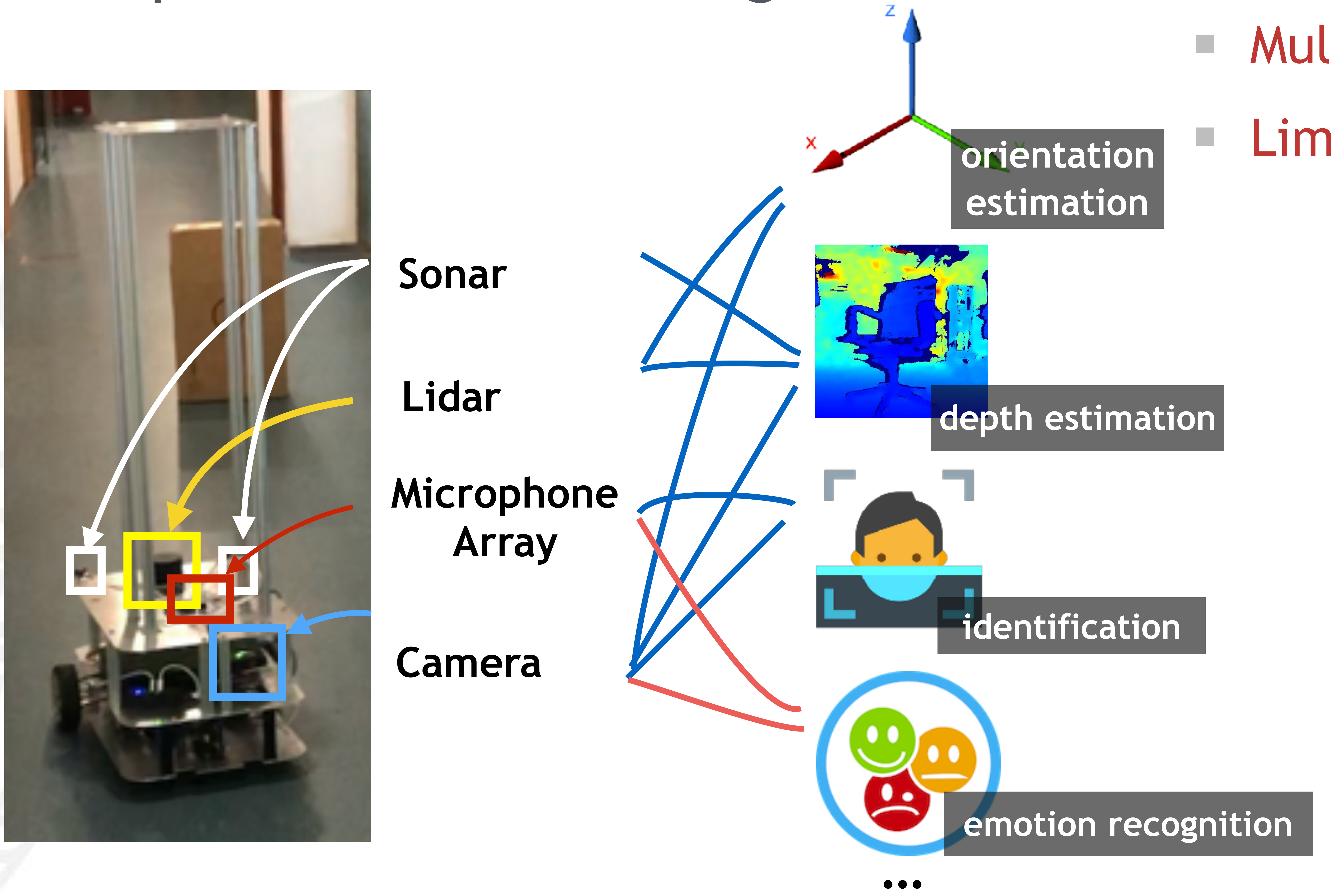
emotion recognition

…

Machine Learning in the Wild

Example: A robotic tour guide

- Need to solve many learning tasks
- Multiple data sources
- Limited training data



Sonar

Lidar

Microphone Array

Camera

orientation estimation

depth estimation

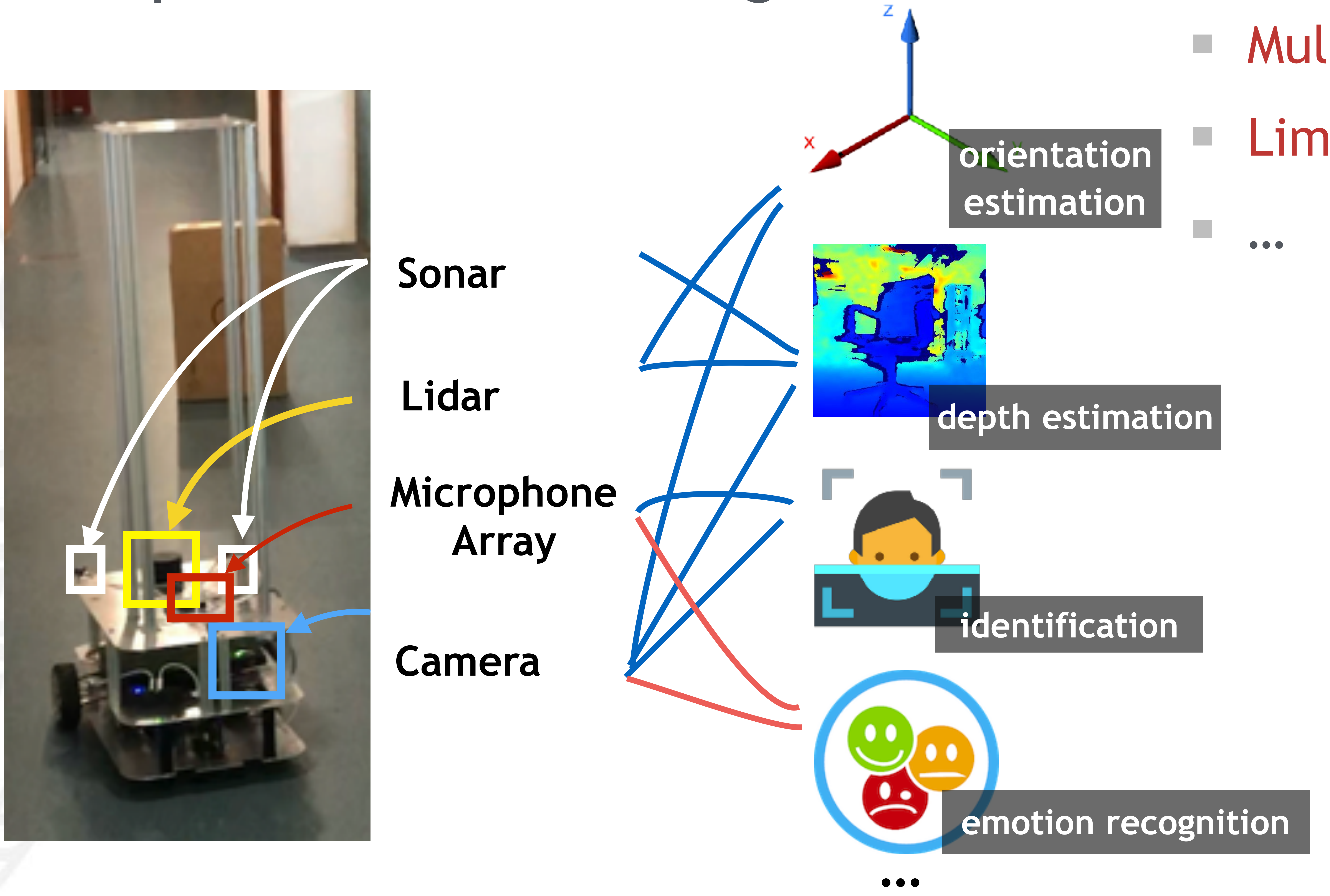
identification

emotion recognition

…

Machine Learning in the Wild

Example: A robotic tour guide



Sonar

Lidar

Microphone Array

Camera

orientation estimation

depth estimation

identification

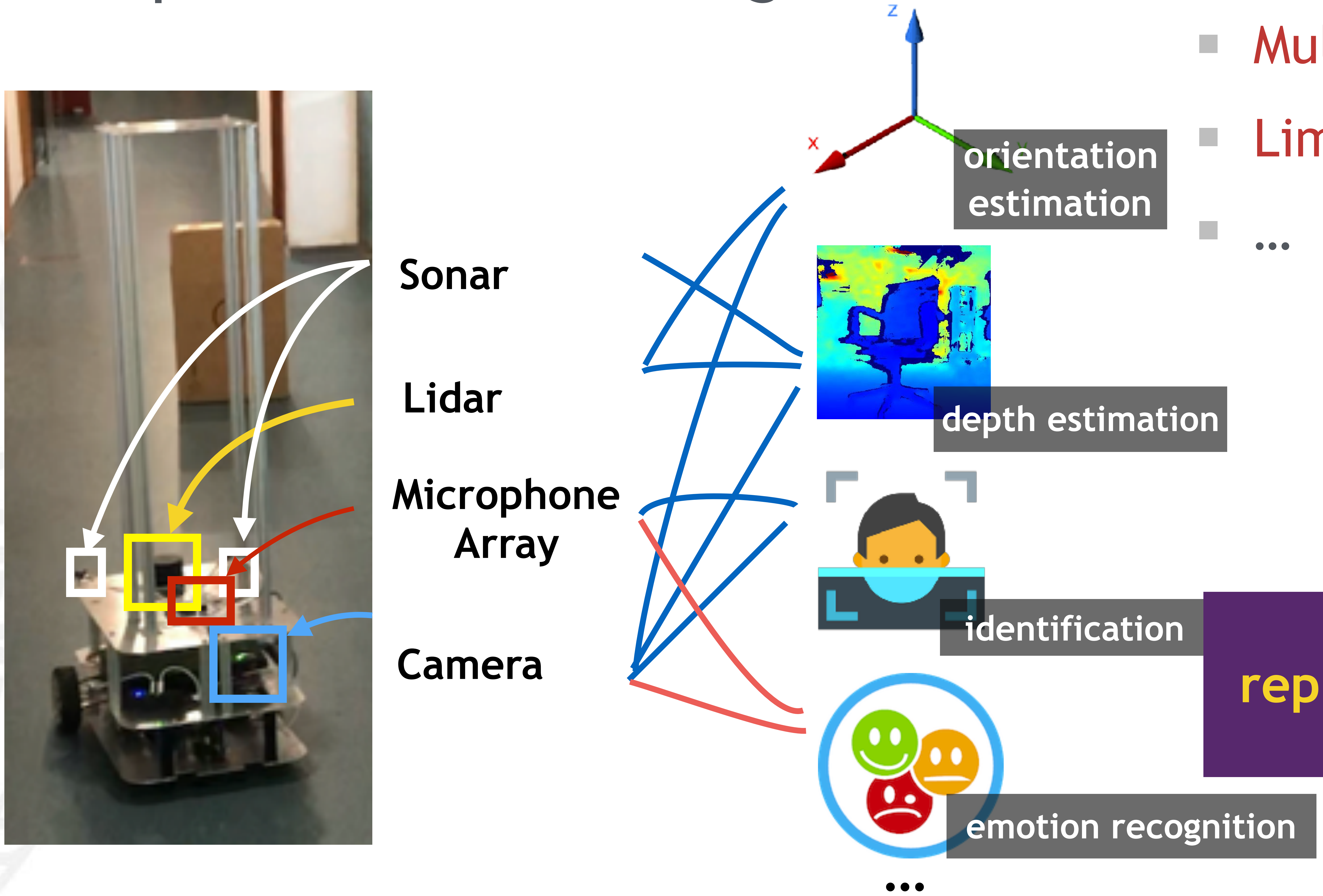
emotion recognition

…

- Need to solve many learning tasks
- Multiple data sources
- Limited training data
- …

Machine Learning in the Wild

Example: A robotic tour guide



- Need to solve many learning tasks
- Multiple data sources
- Limited training data
- …

Need to exploit shared representations in the complex data and tasks

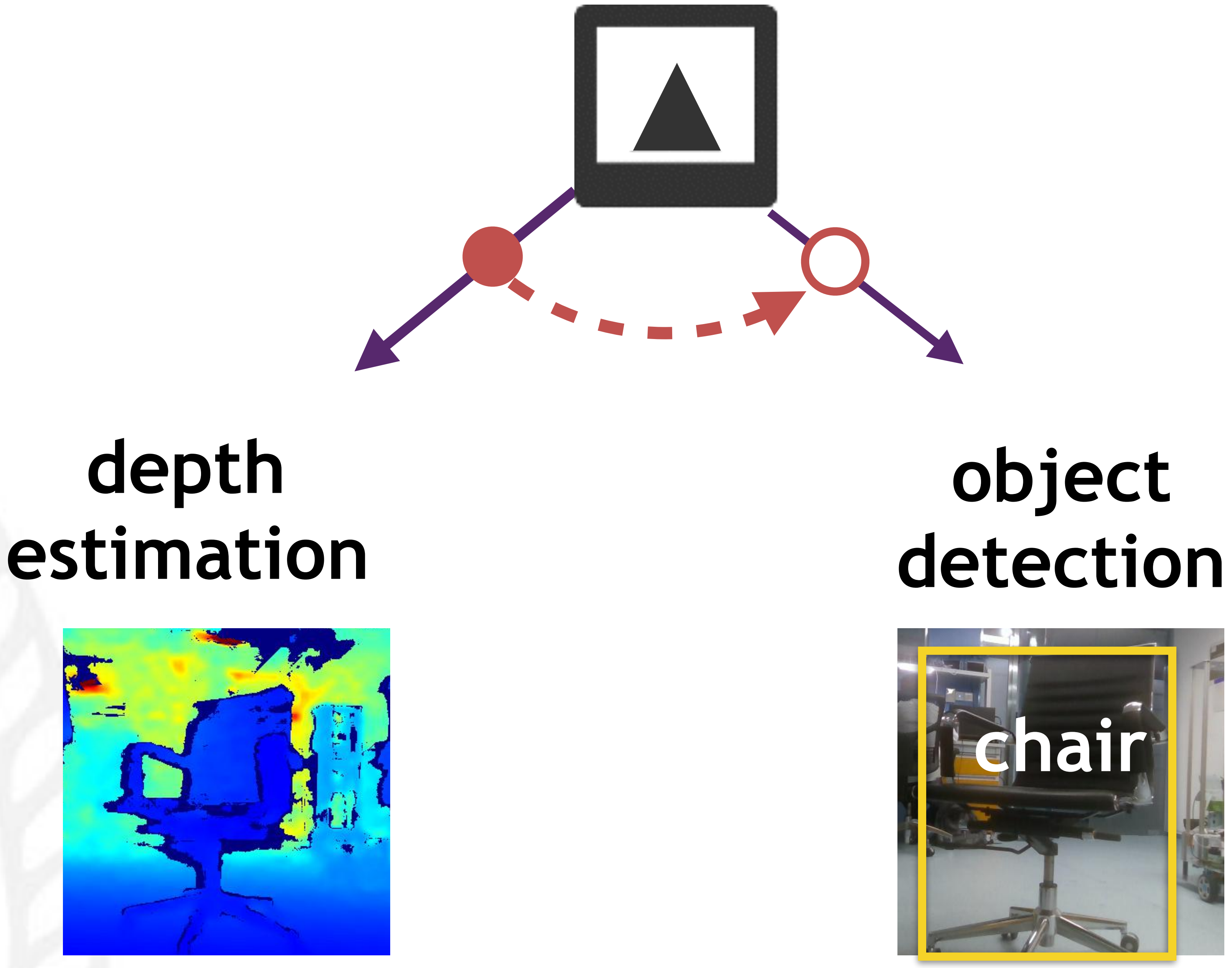
Exploiting Shared Representation

- … among different tasks

- … among different views (input sources, feature sets)

Exploiting Shared Representation

- … among different tasks

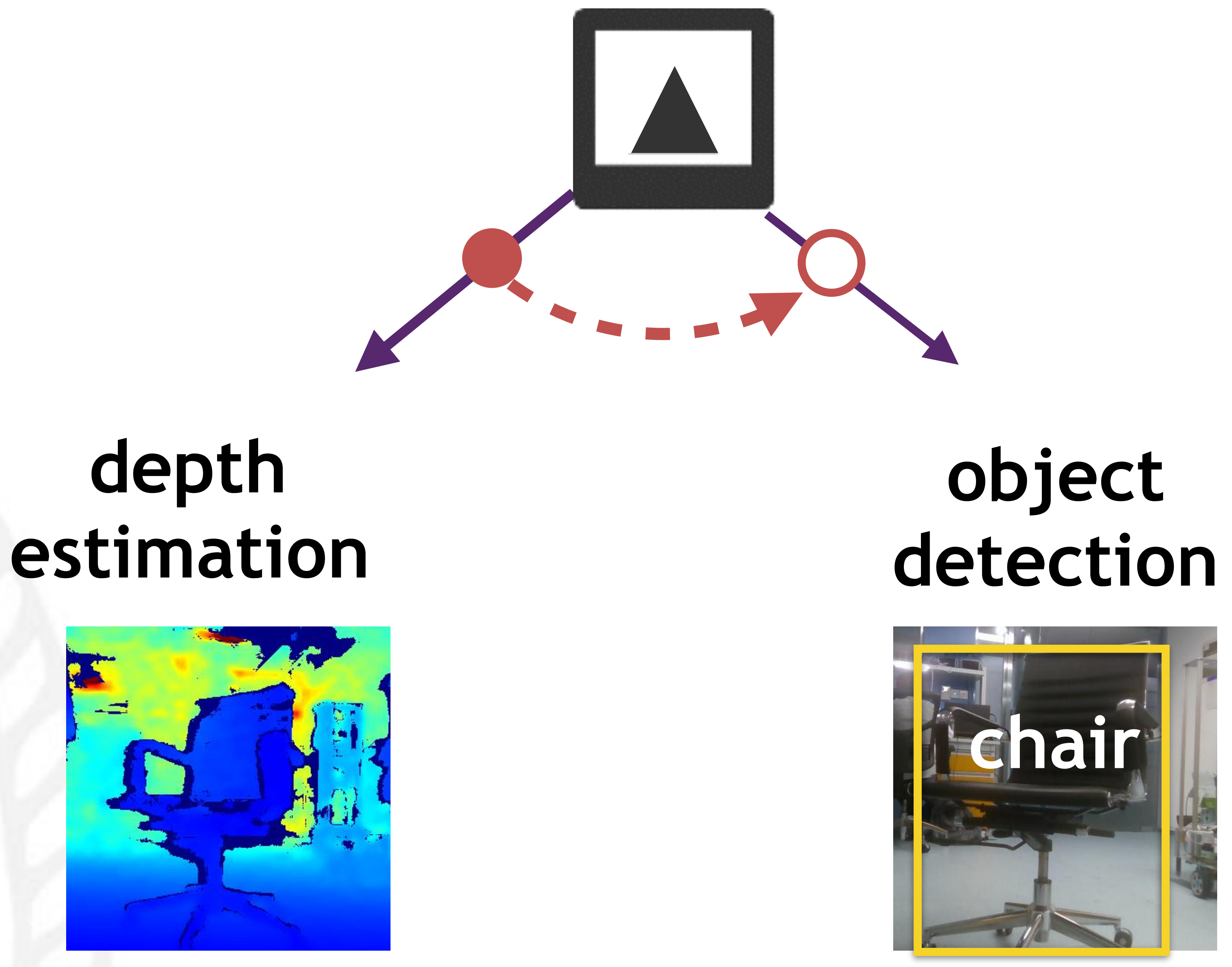


Task transfer learning: reuse the representation of task A for task B

- … among different views (input sources, feature sets)

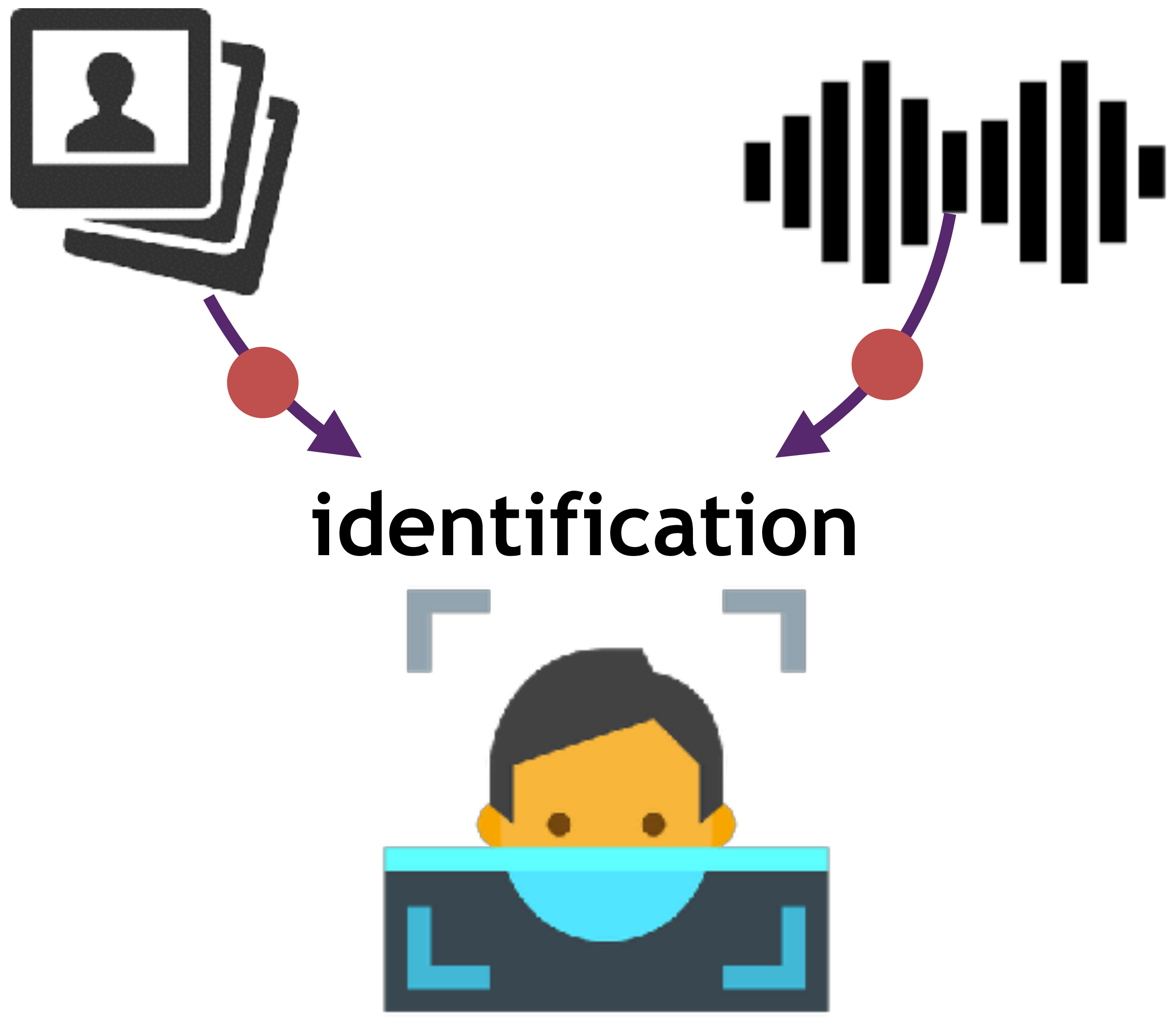
Exploiting Shared Representation

- … among different tasks



Task transfer learning: reuse the representation of task A for task B

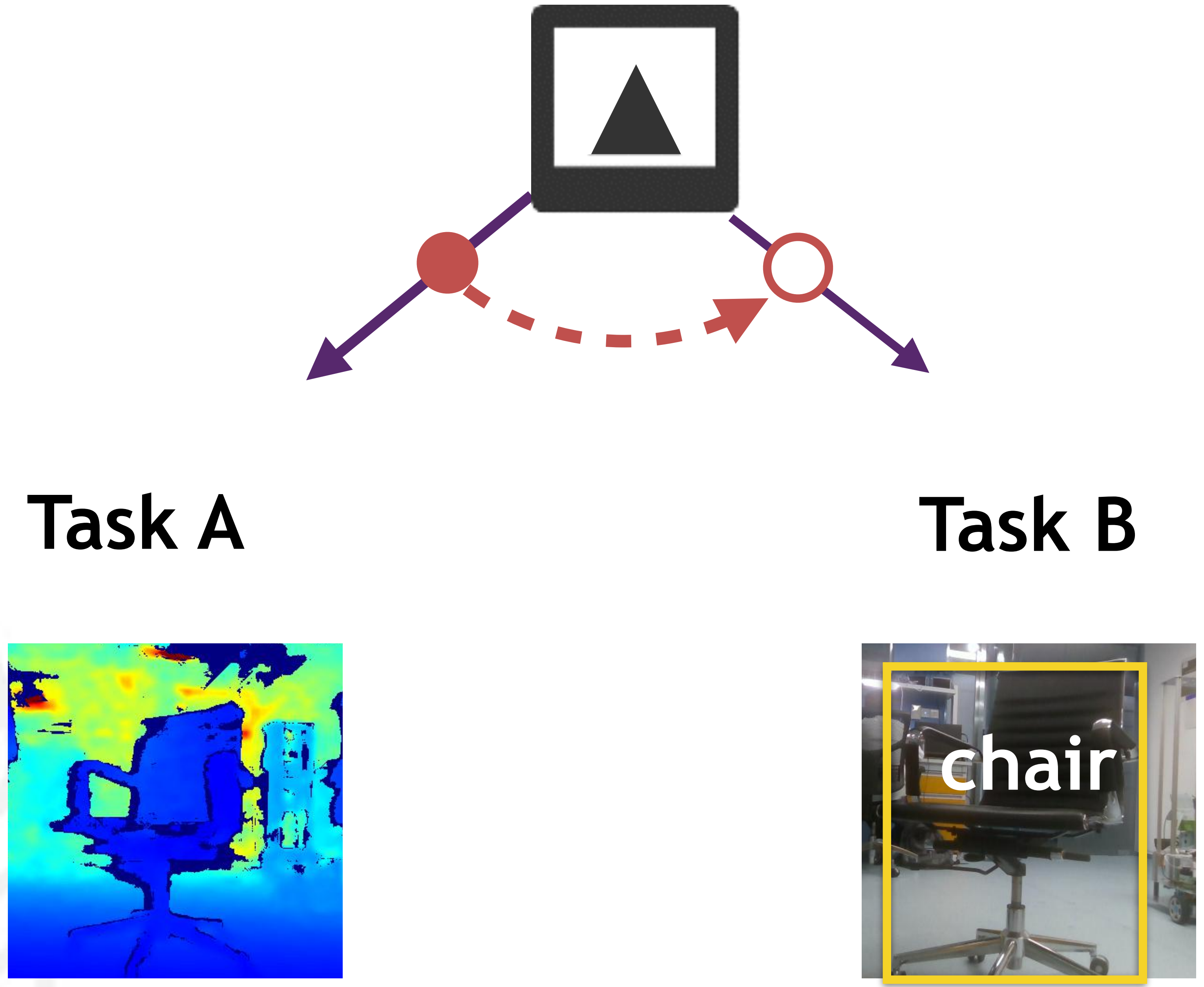
- … among different views (input sources, feature sets)



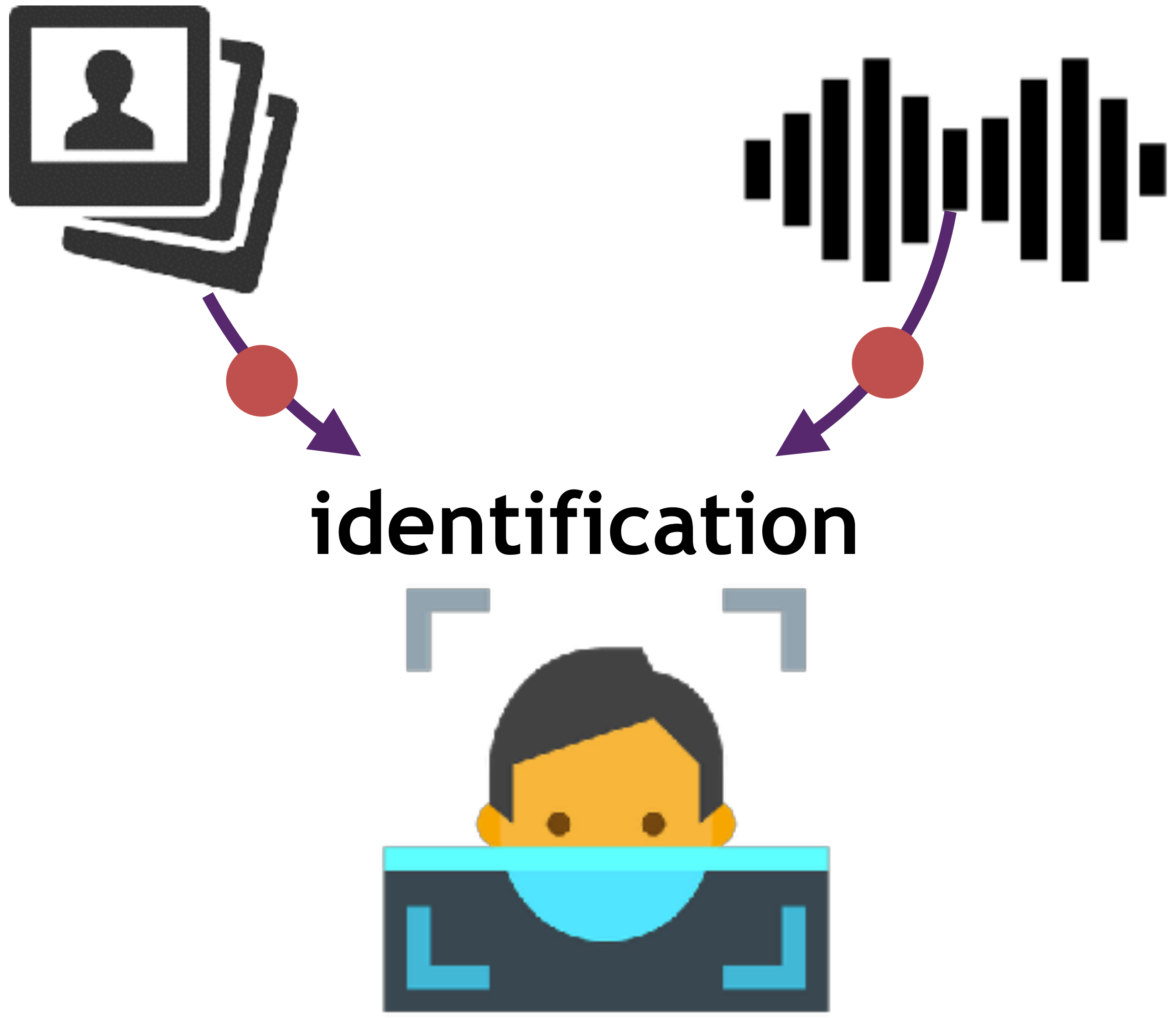
Multiview learning: learn from multi-view representations

Exploiting Shared Representation

- Task transfer learning

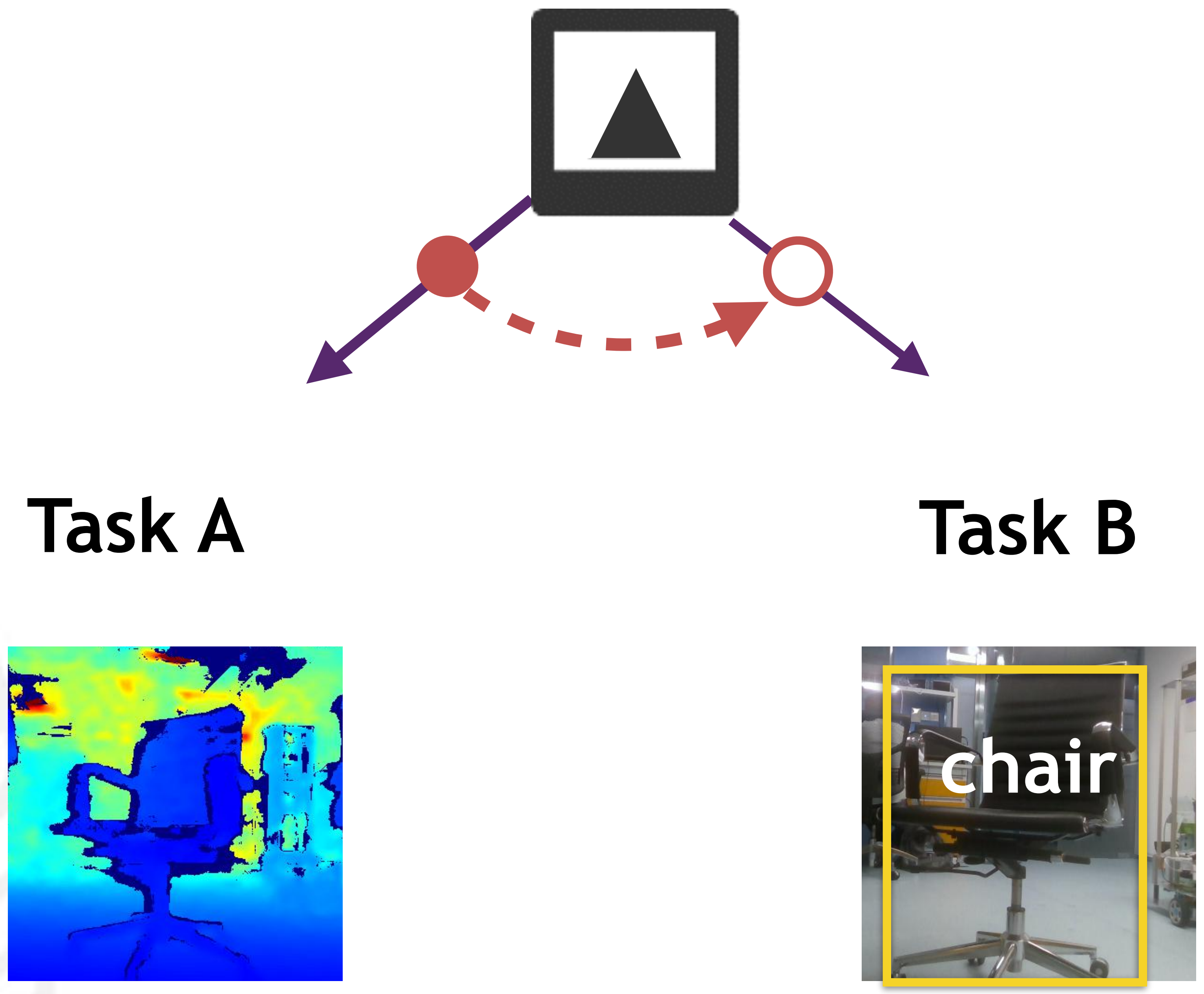


- Multi-view learning



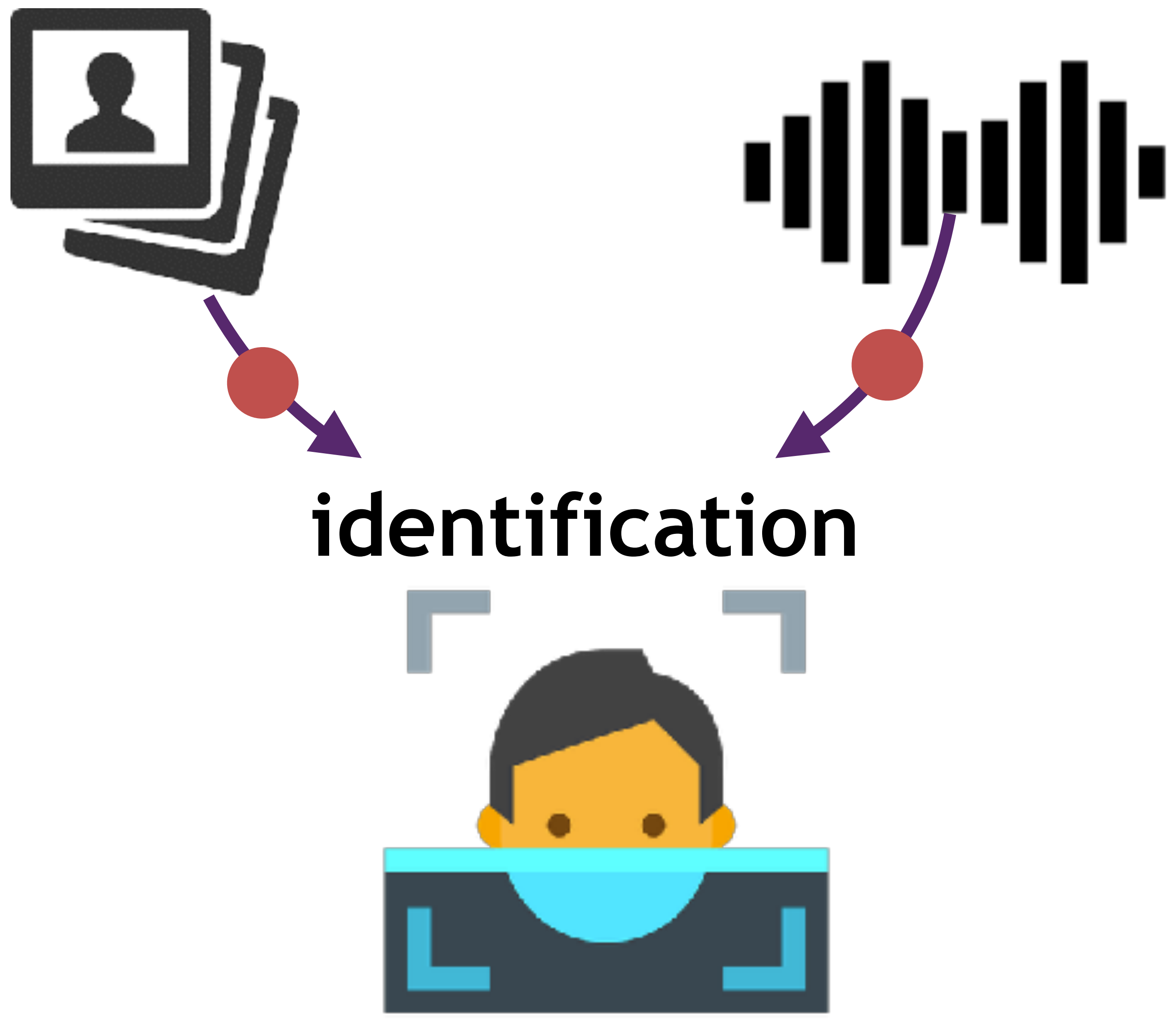
Exploiting Shared Representation

- Task transfer learning



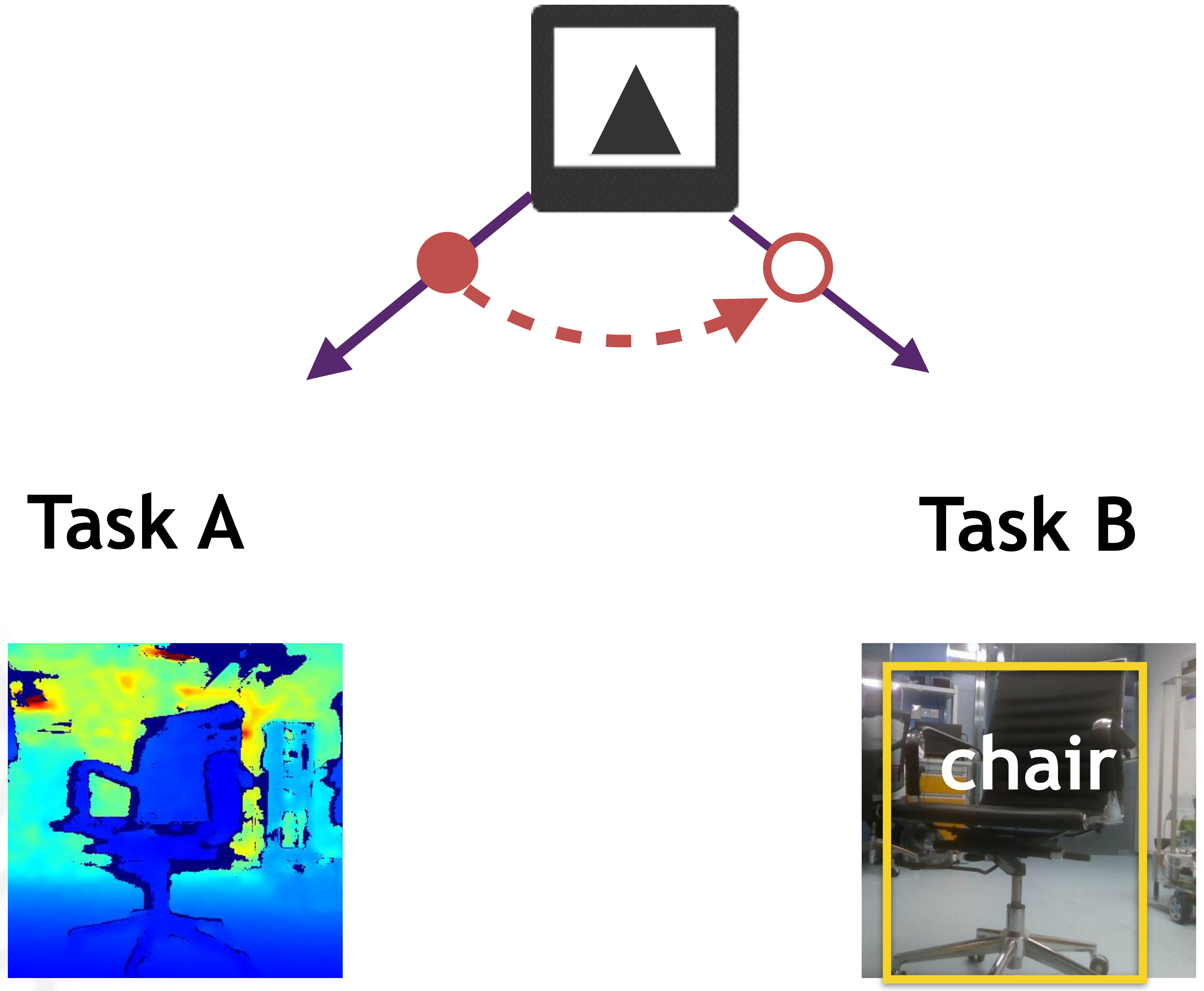
Estimate to what extent representation of task A can help task B?

- Multi-view learning



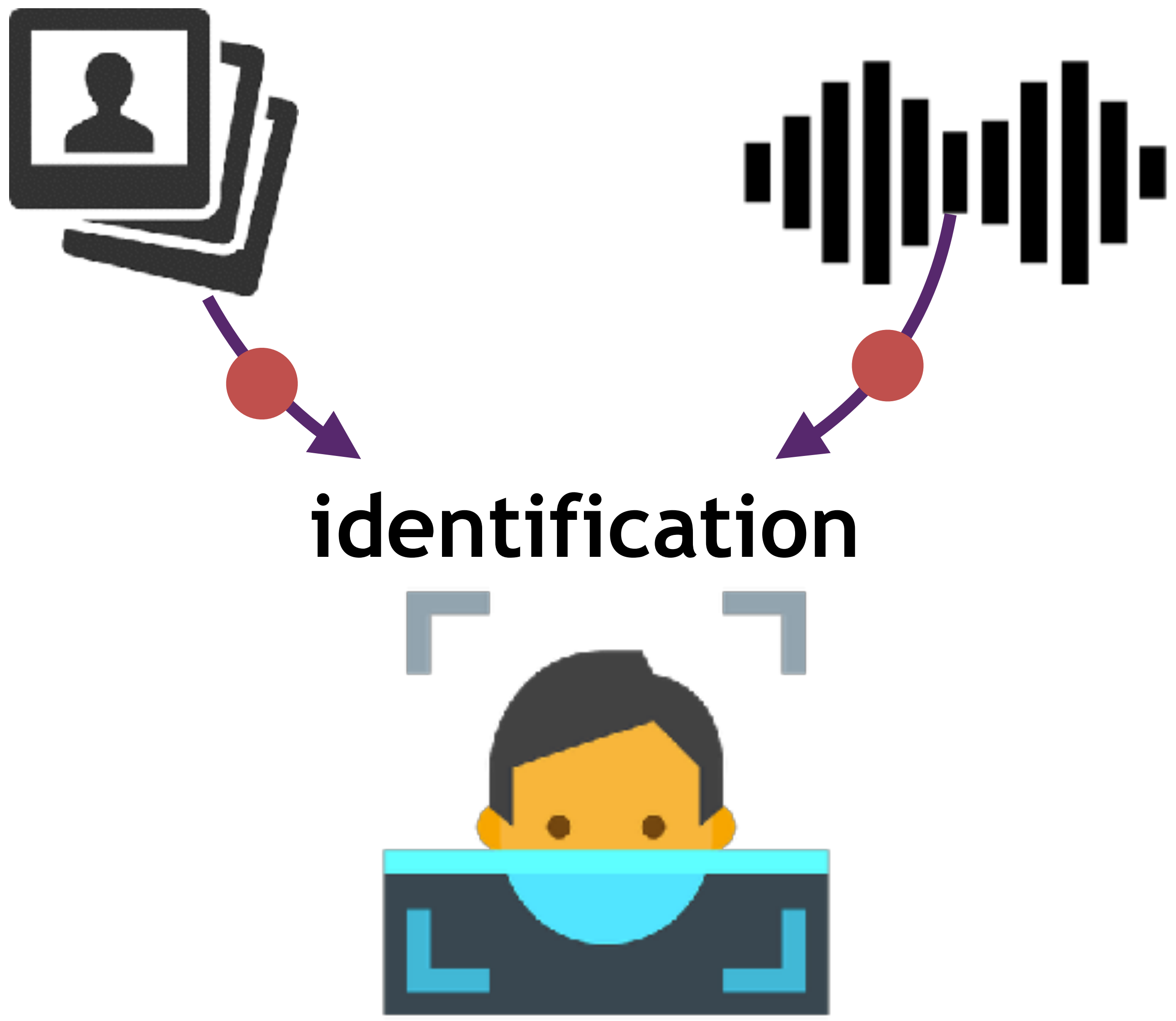
Exploiting Shared Representation

- Task transfer learning



Estimate to what extent representation of task A can help task B?

- Multi-view learning



How to effectively extract shared information?

Representation Learning based on Correlation

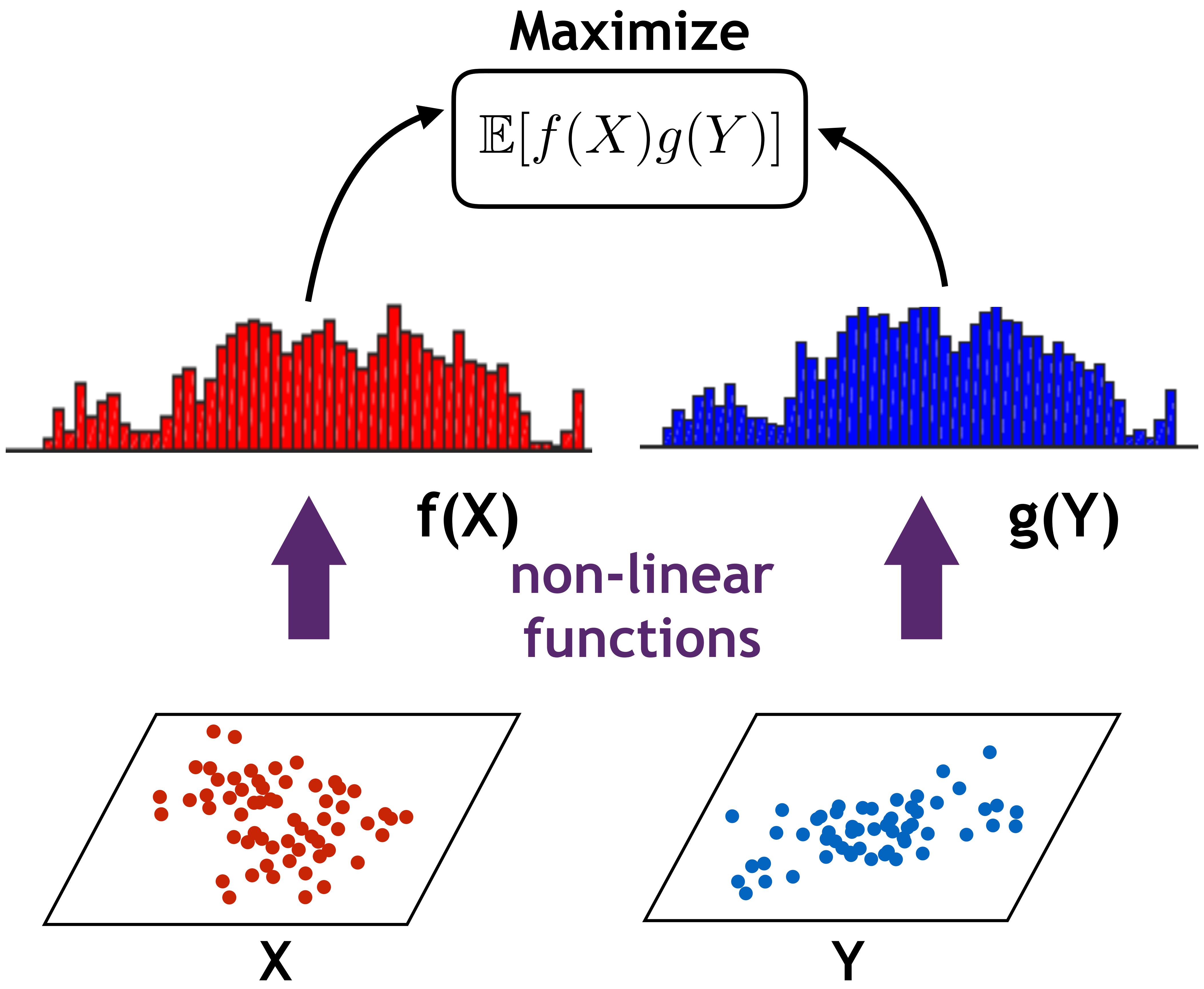
- corr(X,Y) measures the **statistical dependence** between X and Y
  - e.g. Pearson's correlation coefficient $corr_P(X,Y) = \dfrac{\mathbb{E}[(X-\bar{X})^T(Y-\bar{Y})]}{\sigma_X\sigma_Y}$
- Example: **Canonical Correlation Analysis (CCA)**

$$a^*, b^* = \text{argmax}_{a,b} corr\ (a^T X, b^T Y)$$

  - Finds a pair of vectors (a, b) that maximizes correlation between attributes
  - subsequent features are mutually orthogonal
  - limited to linear dependence

Maximal HGR Correlation

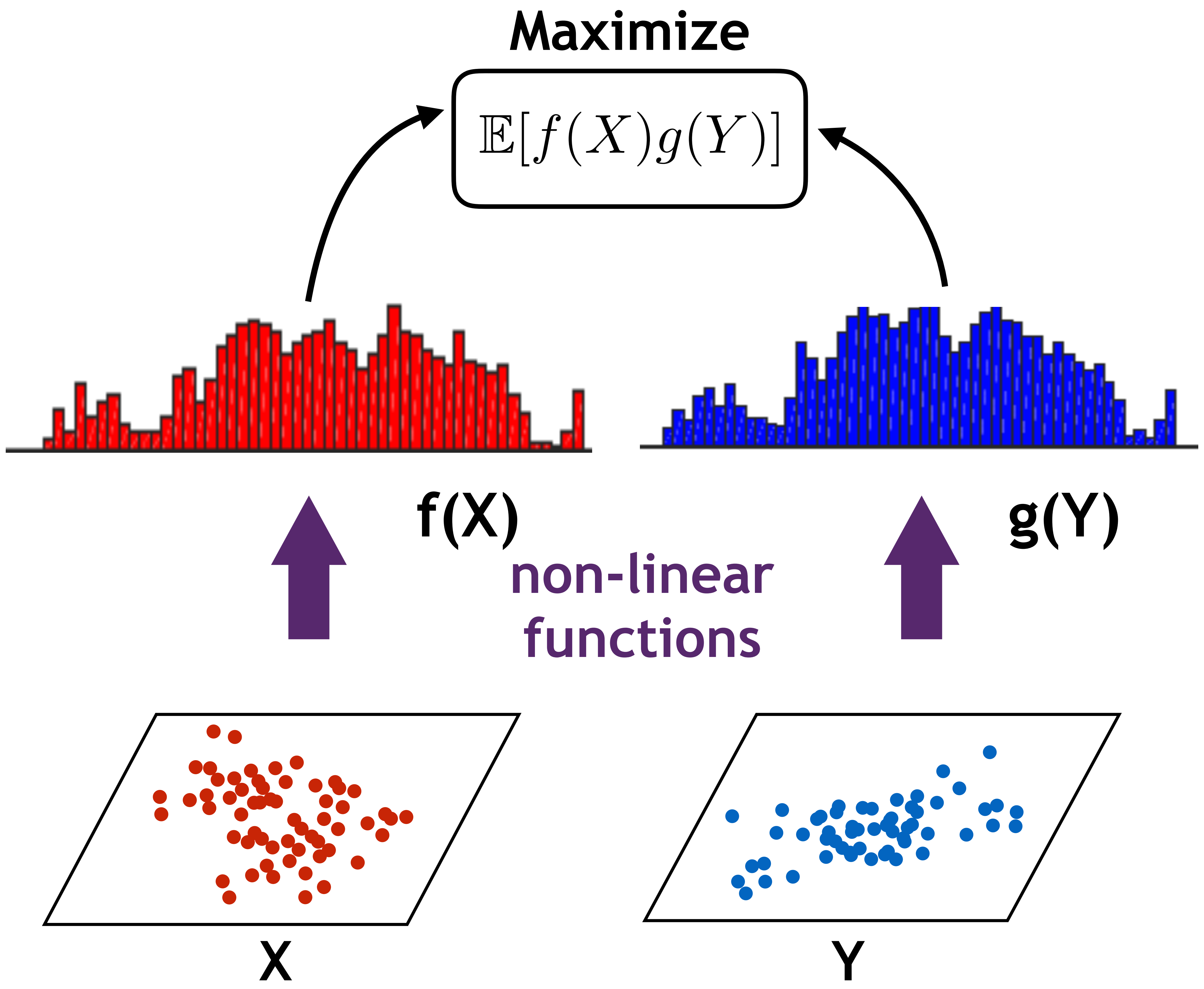
Given random variables X, Y, the Maximal Hirschfeld-Gebelein-Renyi (HGR) correlation [Renyi 1959] is:



$$\sup_{f,g} \mathbb{E}[f(X)g(Y)]$$
$$s.t.\ \mathbb{E}[f(X)] = \mathbb{E}[g(Y)] = 0$$
$$\mathbb{E}[f(X)^2] = \mathbb{E}[g(Y)^2] = 1$$

Maximal HGR Correlation

Given random variables X, Y, the Maximal Hirschfeld-Gebelein-Renyi (HGR) correlation [Renyi 1959] is:



$$\sup_{f,g} \mathbb{E}[f(X)g(Y)]$$

$$s.t.\ \mathbb{E}[f(X)] = \mathbb{E}[g(Y)] = 0$$

$$\mathbb{E}[f(X)^2] = \mathbb{E}[g(Y)^2] = 1$$

- Alternating Conditional Expectation (ACE) algorithm [Breiman 1985]

Recent Information-Theoretic Development

[Huang et al. 2017]

High dimensional cases: $f: \mathcal{X} \to \mathbb{R}^k \quad g: \mathcal{Y} \to \mathbb{R}^k$

$$\begin{aligned}
&\max_{f,g} \mathbb{E}[f(X)^T g(Y)] \\
&\text{s.t. } \mathbb{E}[f(X)] = \mathbb{E}[g(Y)] = 0 \\
&\qquad \operatorname{Cov}[f(X)] = \operatorname{Cov}(g(Y)) = I
\end{aligned}$$

Recent Information-Theoretic Development

[Huang et al. 2017]

High dimensional cases: $f : \mathscr{X} \to \mathbb{R}^k \quad g : \mathscr{Y} \to \mathbb{R}^k$

$$\begin{aligned} \max_{f,g} \ & \mathbb{E}[f(X)^T g(Y)] \\ \text{s.t.} \ & \mathbb{E}[f(X)] = \mathbb{E}[g(Y)] = 0 \\ & \mathrm{Cov}[f(X)] = \mathrm{Cov}(g(Y)) = I \end{aligned}$$

Effective and robust information decompostion

Recent Information-Theoretic Development

High dimensional cases: $f : \mathcal{X} \rightarrow \mathbb{R}^k \quad g : \mathcal{Y} \rightarrow \mathbb{R}^k$

[Huang et al. 2017]

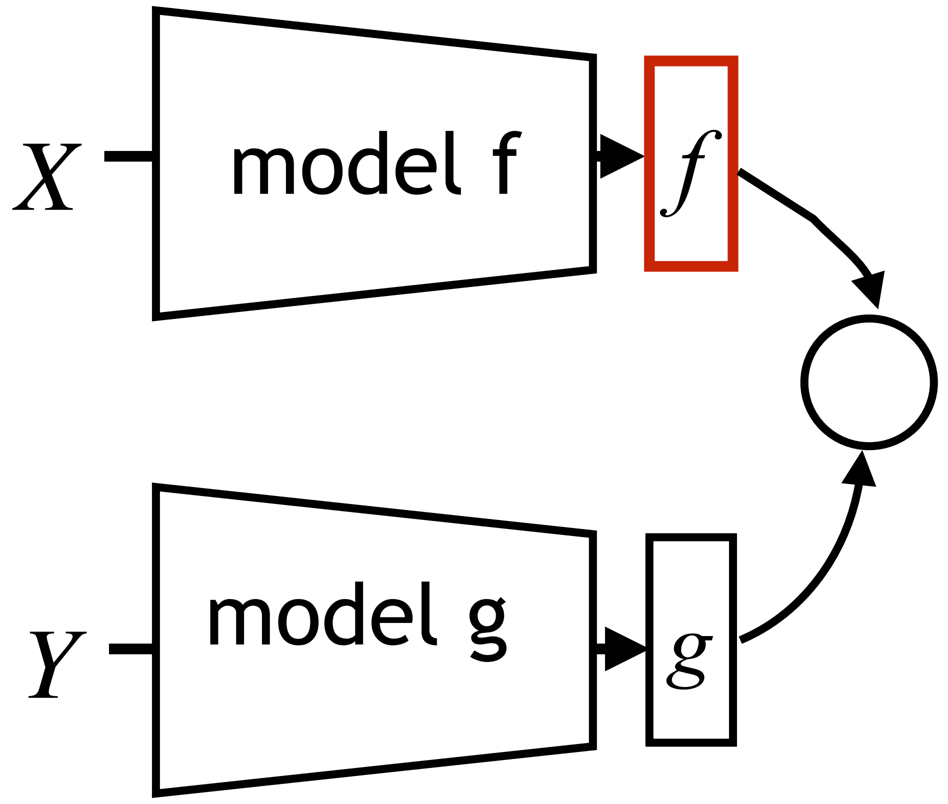
$$\max_{f,g} \mathbb{E}[f(X)^T g(Y)]$$

$$\text{s.t. } \mathbb{E}[f(X)] = \mathbb{E}[g(Y)] = 0$$

$$\text{Cov}[f(X)] = \text{Cov}(g(Y)) = I$$

Effective and robust information decompostion

Soft-HGR Loss [Wang et al. 2018]:



$$L = -2\mathbb{E}[f(X)^T g(Y)] + \text{tr}(\text{cov}(f(X))\text{cov}(g(Y)))$$

Recent Information-Theoretic Development

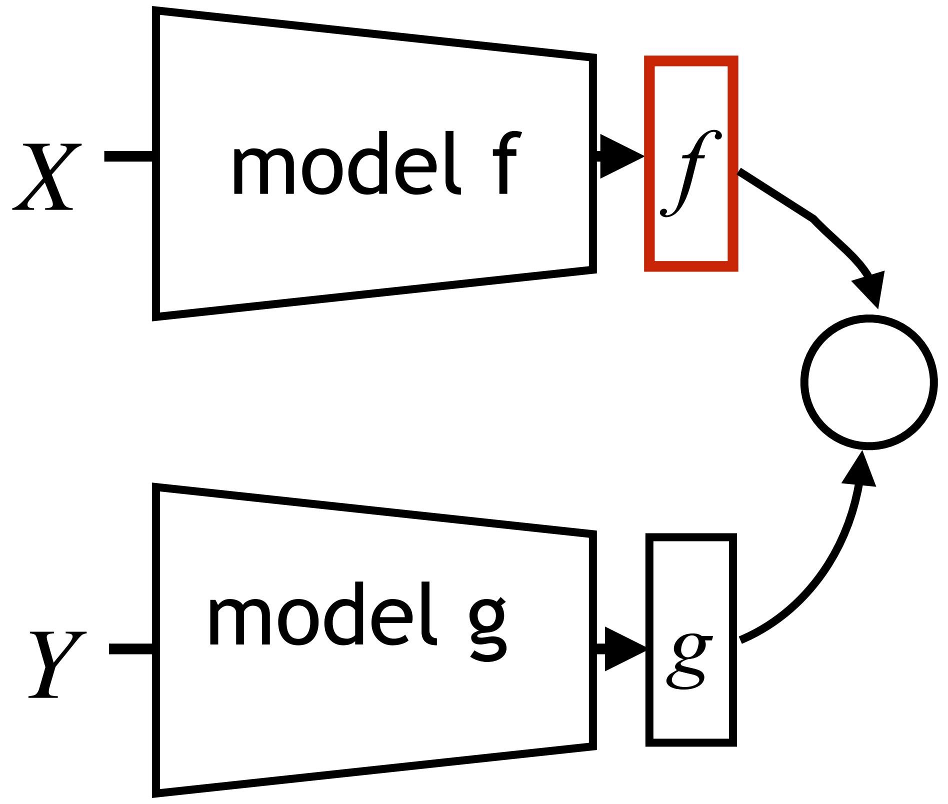
High dimensional cases: $f : \mathcal{X} \to \mathbb{R}^k \quad g : \mathcal{Y} \to \mathbb{R}^k$

[Huang et al. 2017]

$$\max_{f,g} \mathbb{E}[f(X)^T g(Y)]$$
$$\text{s.t. } \mathbb{E}[f(X)] = \mathbb{E}[g(Y)] = 0$$
$$\text{Cov}[f(X)] = \text{Cov}(g(Y)) = I$$

Effective and robust information decompostion

Soft-HGR Loss [Wang et al. 2018]:



$$L = -2\mathbb{E}[f(X)^T g(Y)] + \text{tr}(\text{cov}(f(X))\text{cov}(g(Y)))$$

Eliminate the whitening constraint

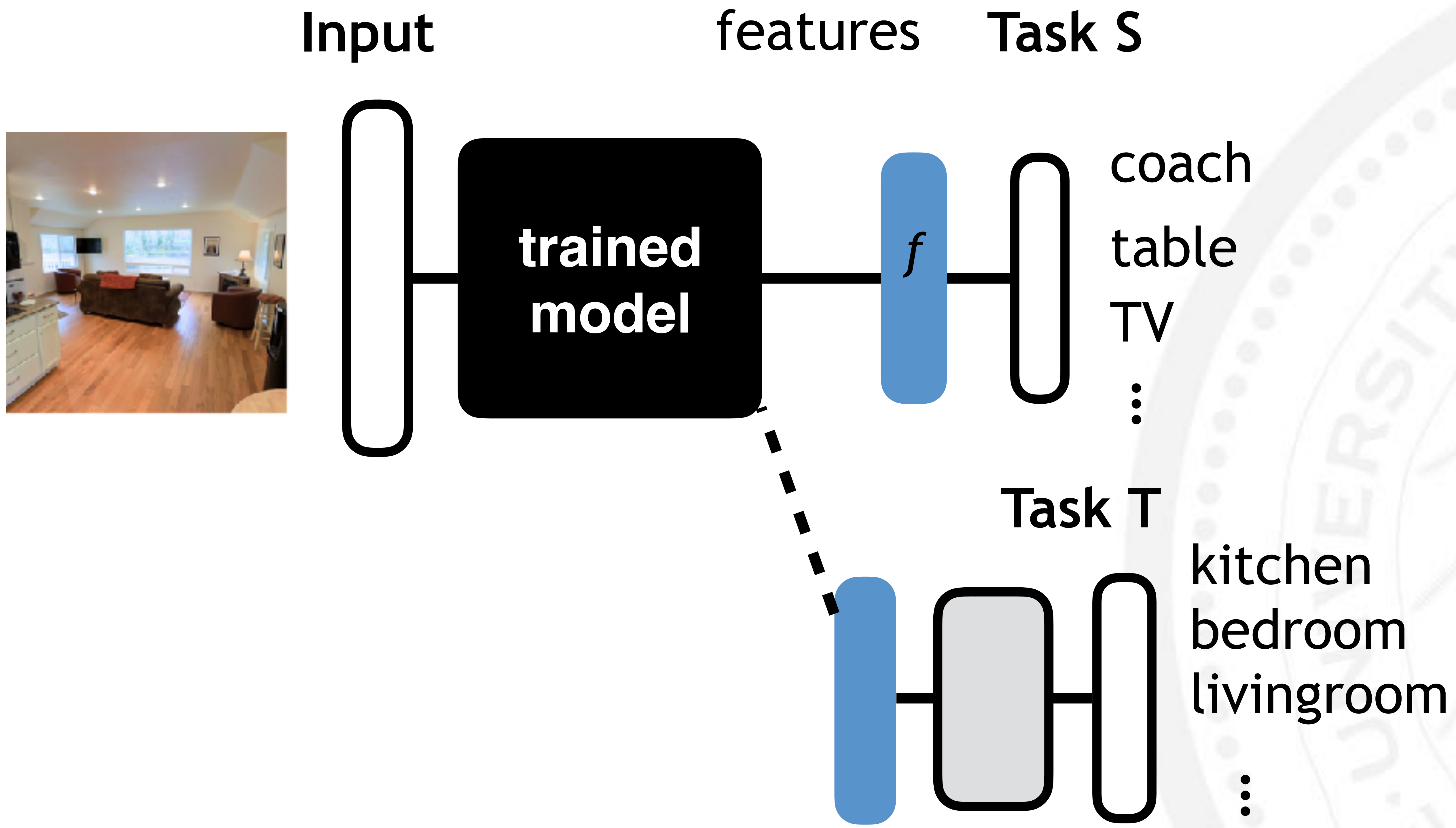
Outline

- Intro: Shared Representation & Maximal Correlation
- Estimating Task Transferability in Task Transfer Learning
- Multi-view learning
- Conclusion

Task Transfer Learning

“ Discriminability-Based Transfer between Neural Networks” (Pratt 1993):

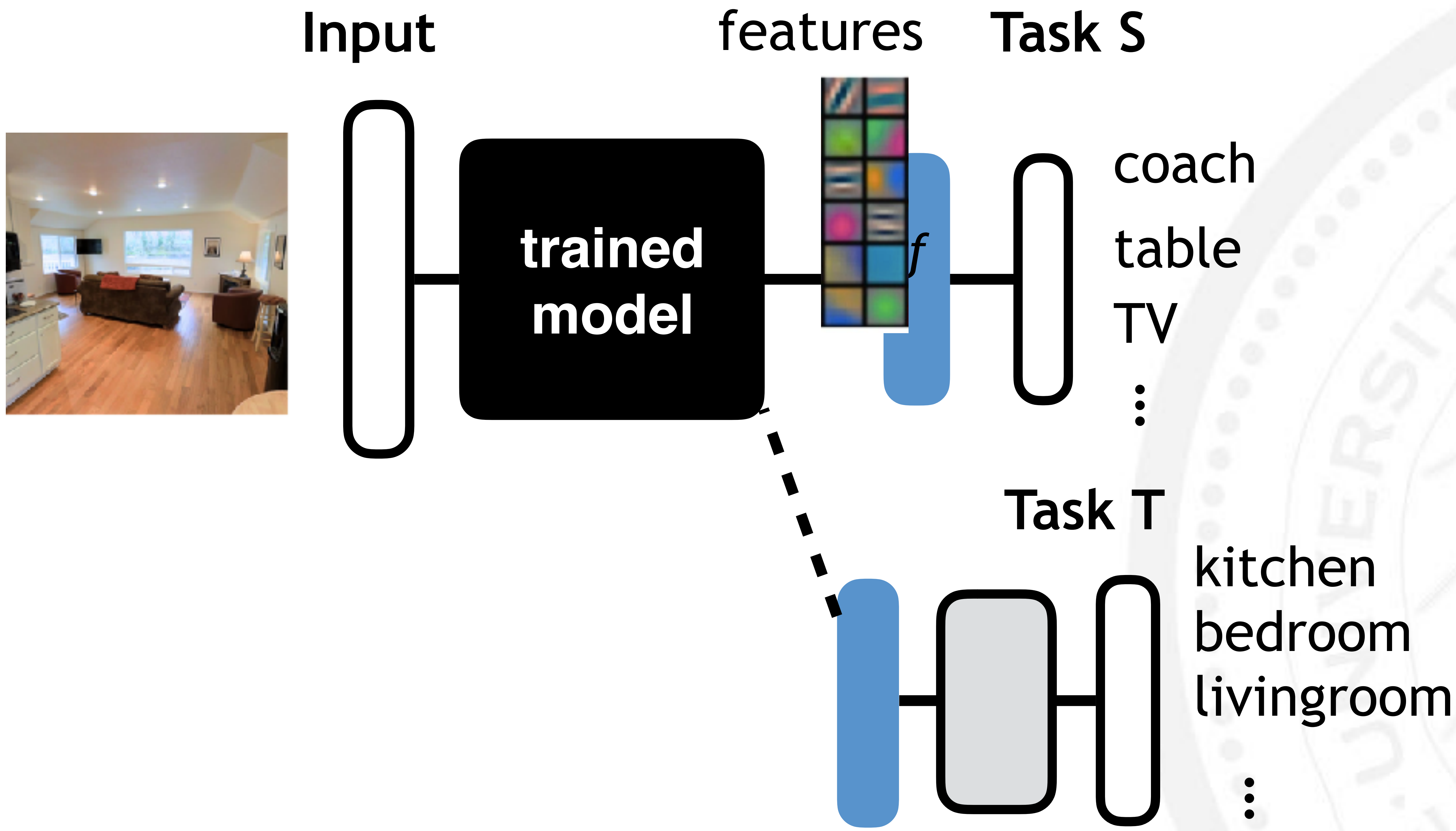
- Input: training data for task S and T, and a pre-trained source model
- Goal: train task T



Task Transfer Learning

“ Discriminability-Based Transfer between Neural Networks” (Pratt 1993):

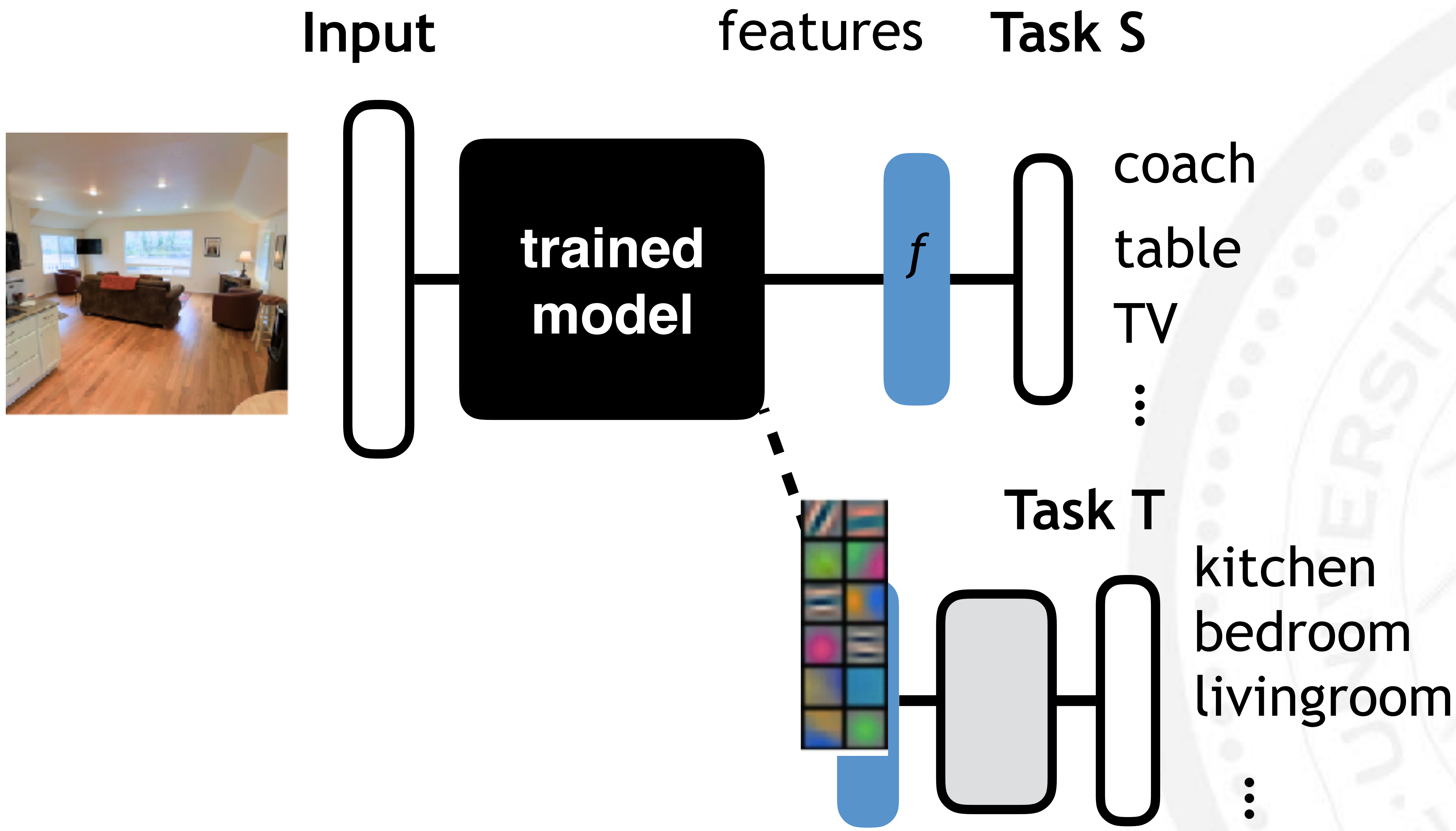
- Input: training data for task S and T, and a pre-trained source model
- Goal: train task T



Task Transfer Learning

“ Discriminability-Based Transfer between Neural Networks” (Pratt 1993):

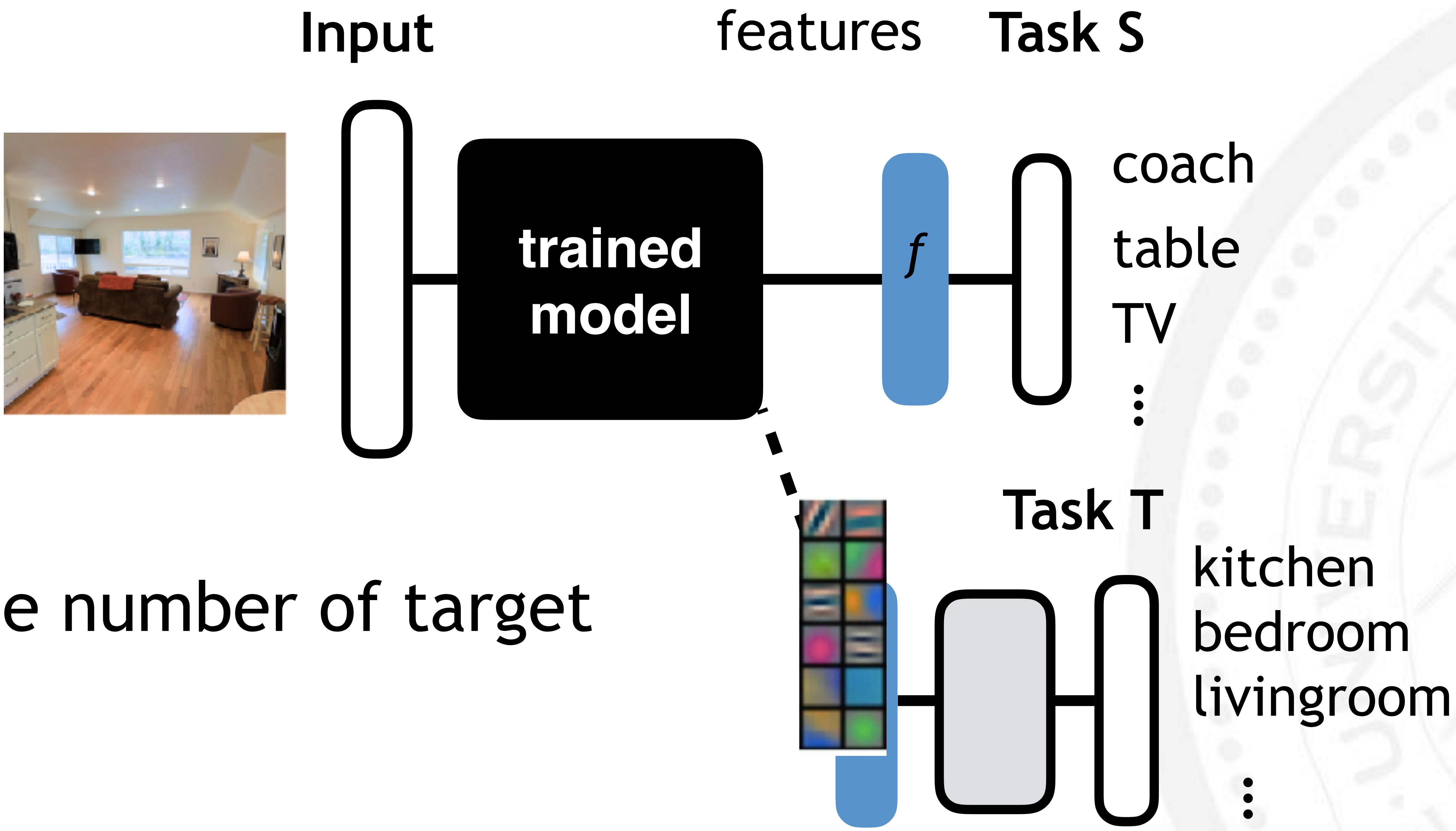
- Input: training data for task S and T, and a pre-trained source model
- Goal: train task T



Task Transfer Learning

“ Discriminability-Based Transfer between Neural Networks” (Pratt 1993):

- Input: training data for task S and T, and a pre-trained **source model**
- Goal: train task T



Improve target training efficiency, reduce number of target labeled data needed

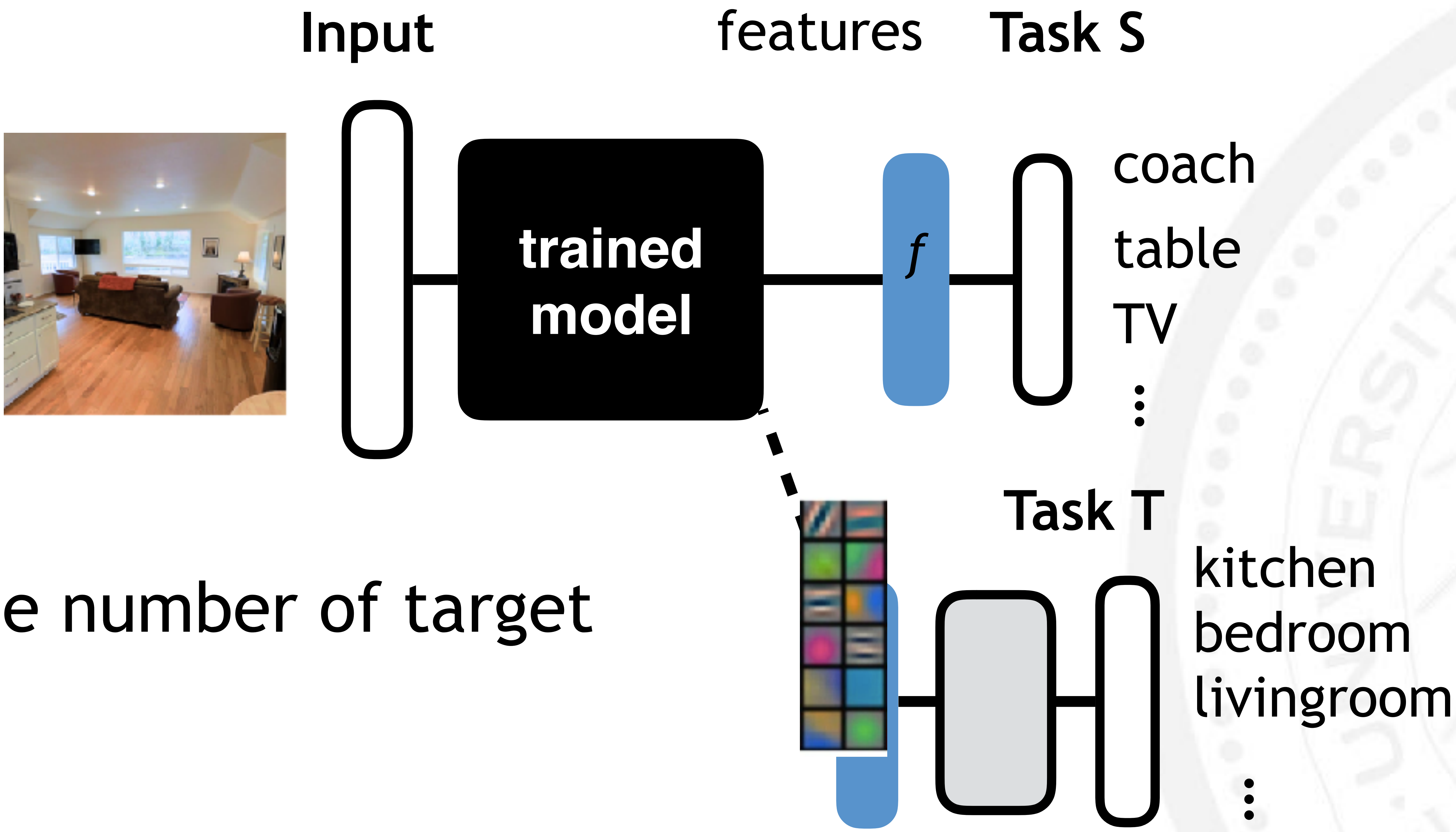
Task Transfer Learning

“ Discriminability-Based Transfer between Neural Networks” (Pratt 1993):

- Input: training data for task S and T, and a pre-trained source model
- Goal: train task T

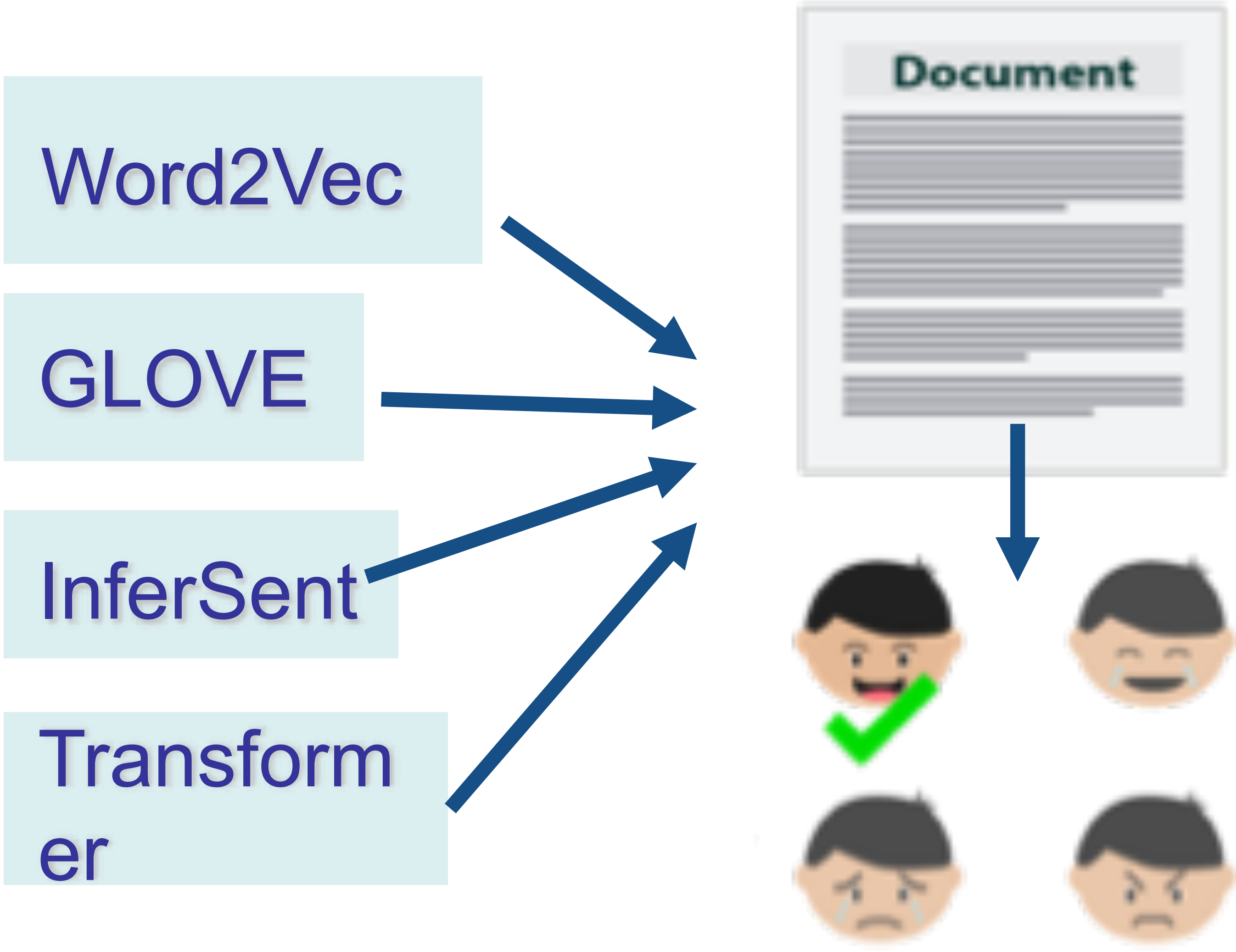
Improve target training efficiency, reduce number of target labeled data needed

Assumes represenation of S is *transferable* to T



Why Task Transferability is Important?

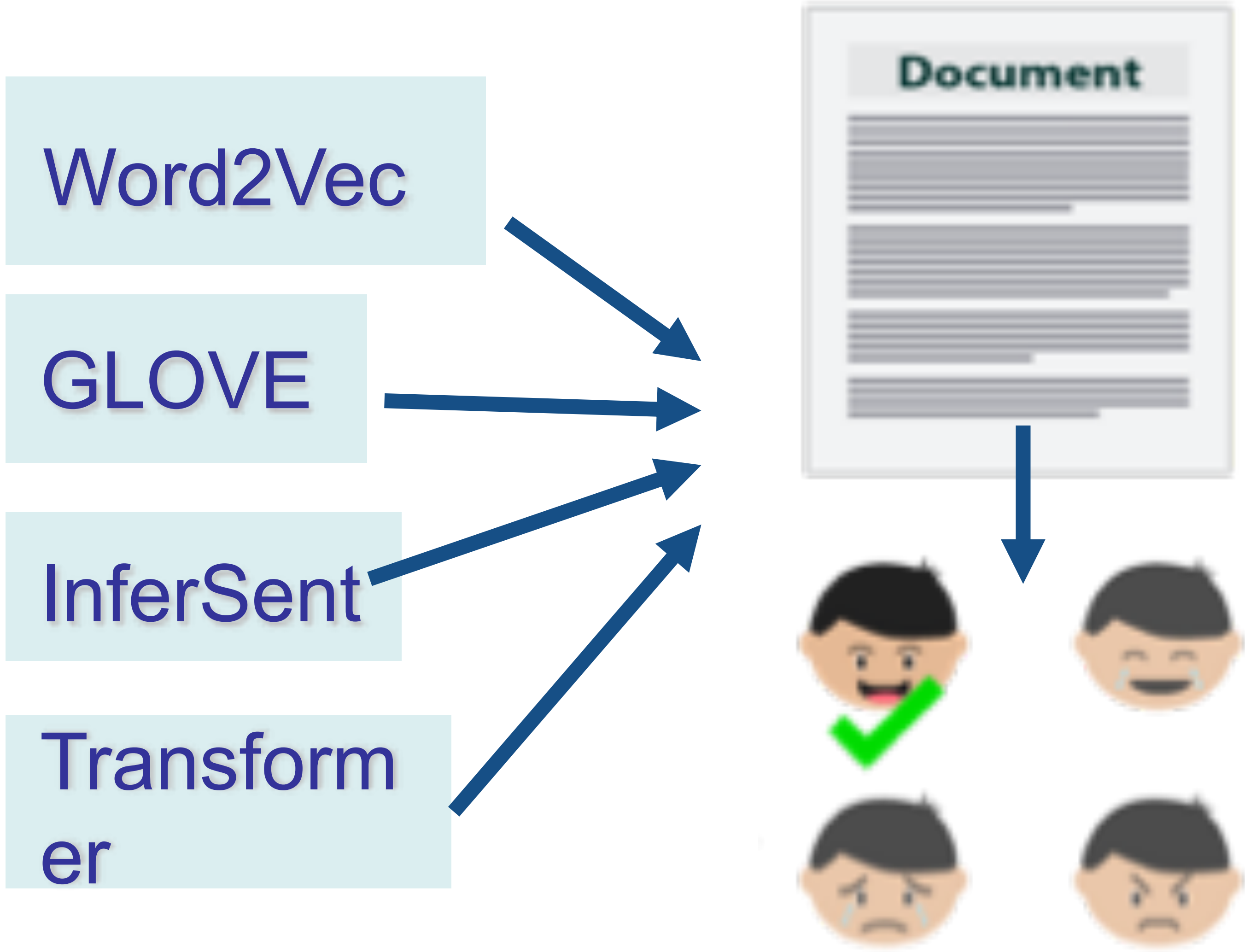
- Model selection



e.g. Select the best word/sentence encoder for NLP tasks

Why Task Transferability is Important?

- Model selection
- Task transfer policy learning



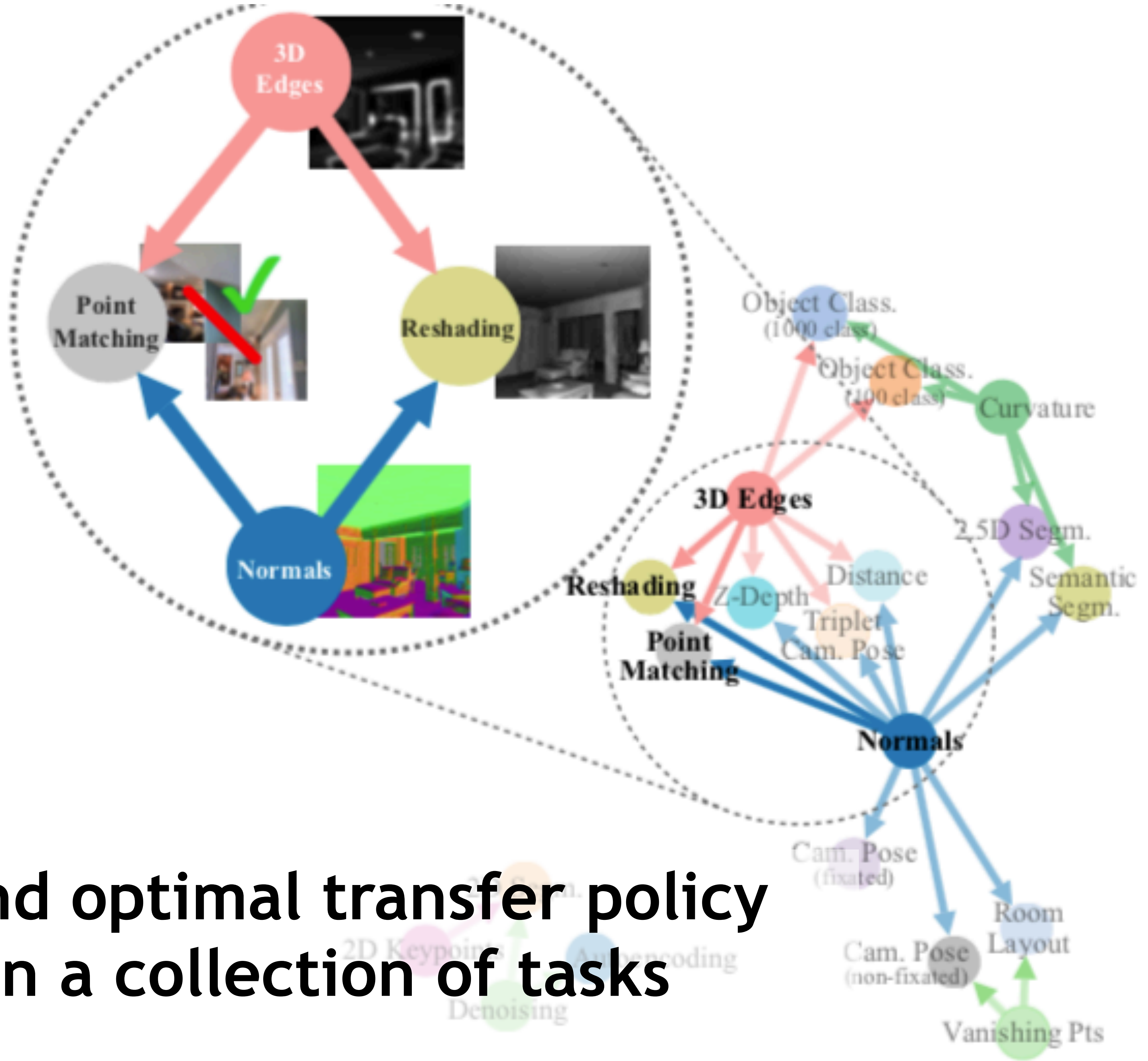
Word2Vec

GLOVE

InferSent

Transform er

e.g. Select the best word/sentence encoder for NLP tasks

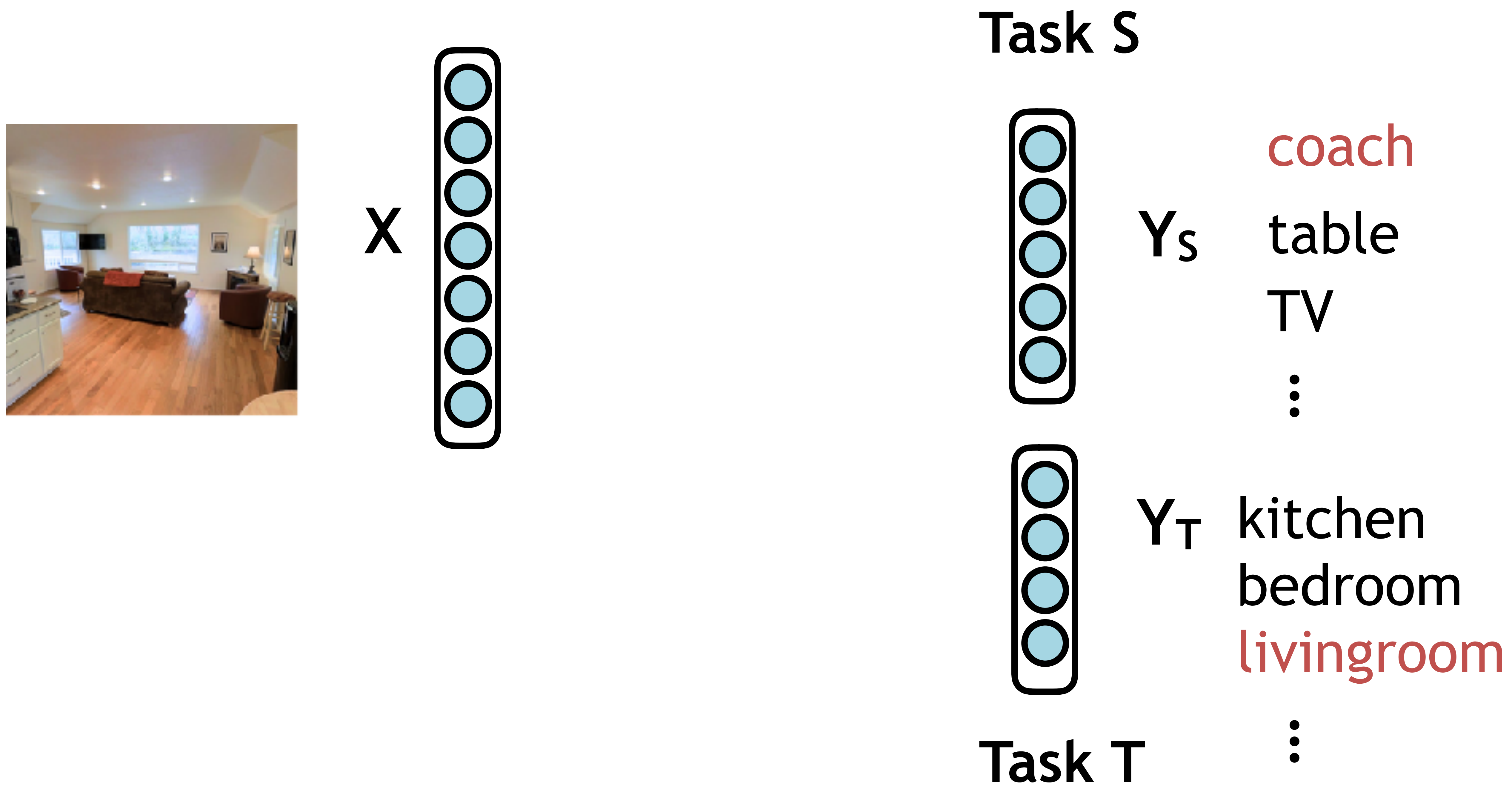


e.g. Find optimal transfer policy given a collection of tasks

The Task Transferability Problem

Given:

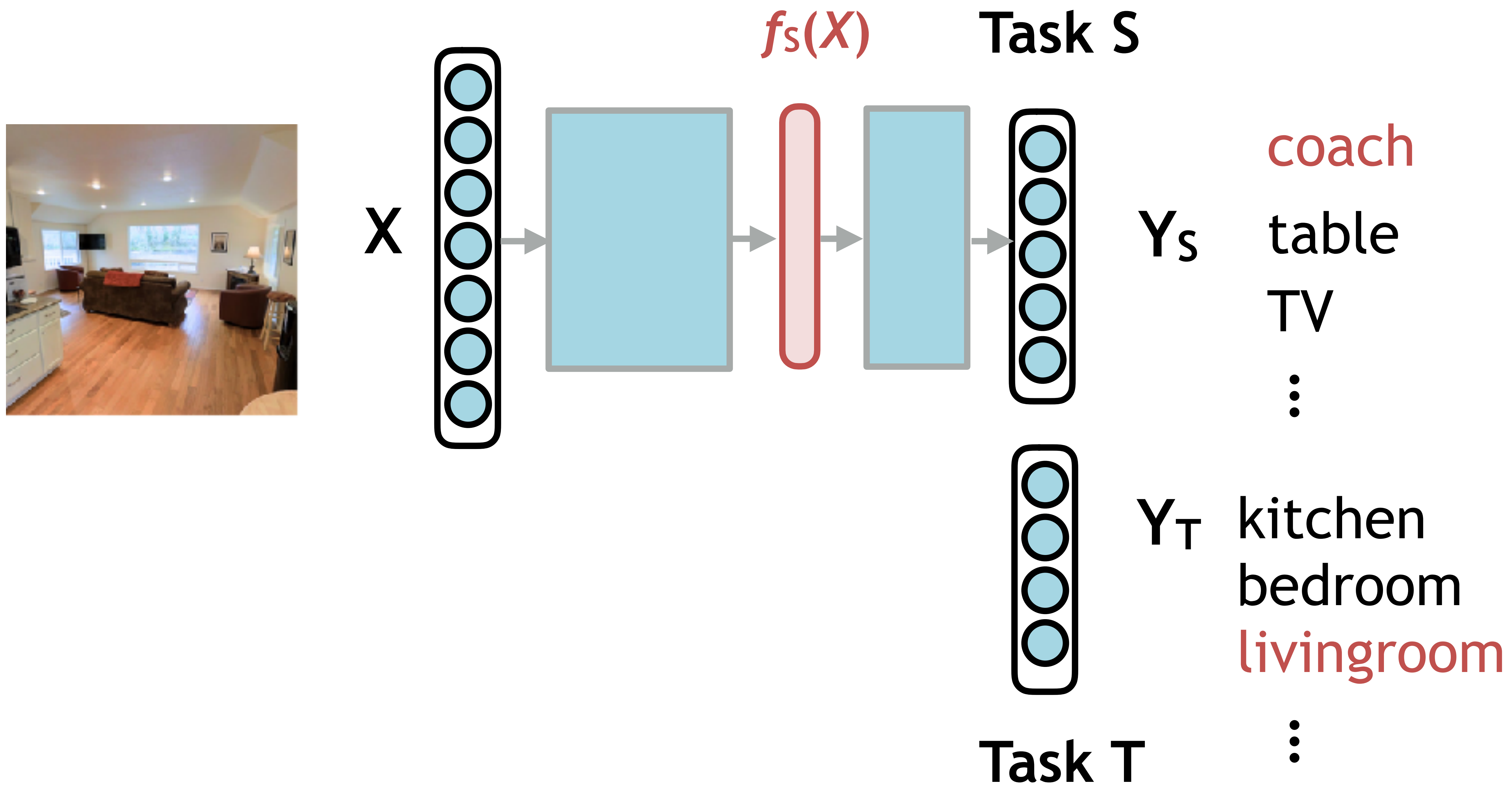
- Input X, source task label Y_S, target task label Y_T



The Task Transferability Problem

Given:

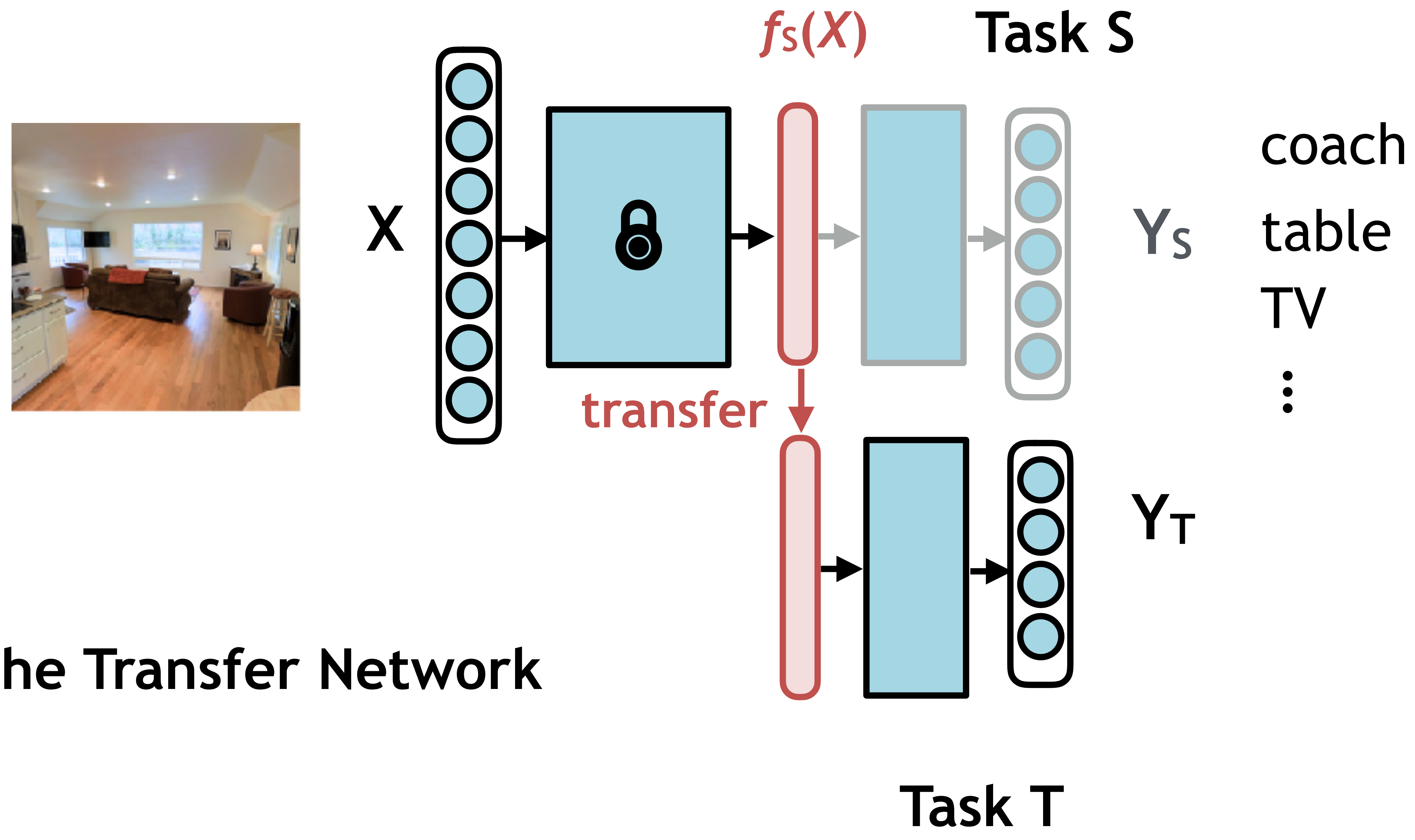
- Input X, source task label Y_S, target task label Y_T
- Trained source model with optimal feature $f_S(X)$



The Task Transferability Problem

Given:

- Input X, source task label Y_S, target task label Y_T
- Trained source model with optimal feature $f_S(X)$

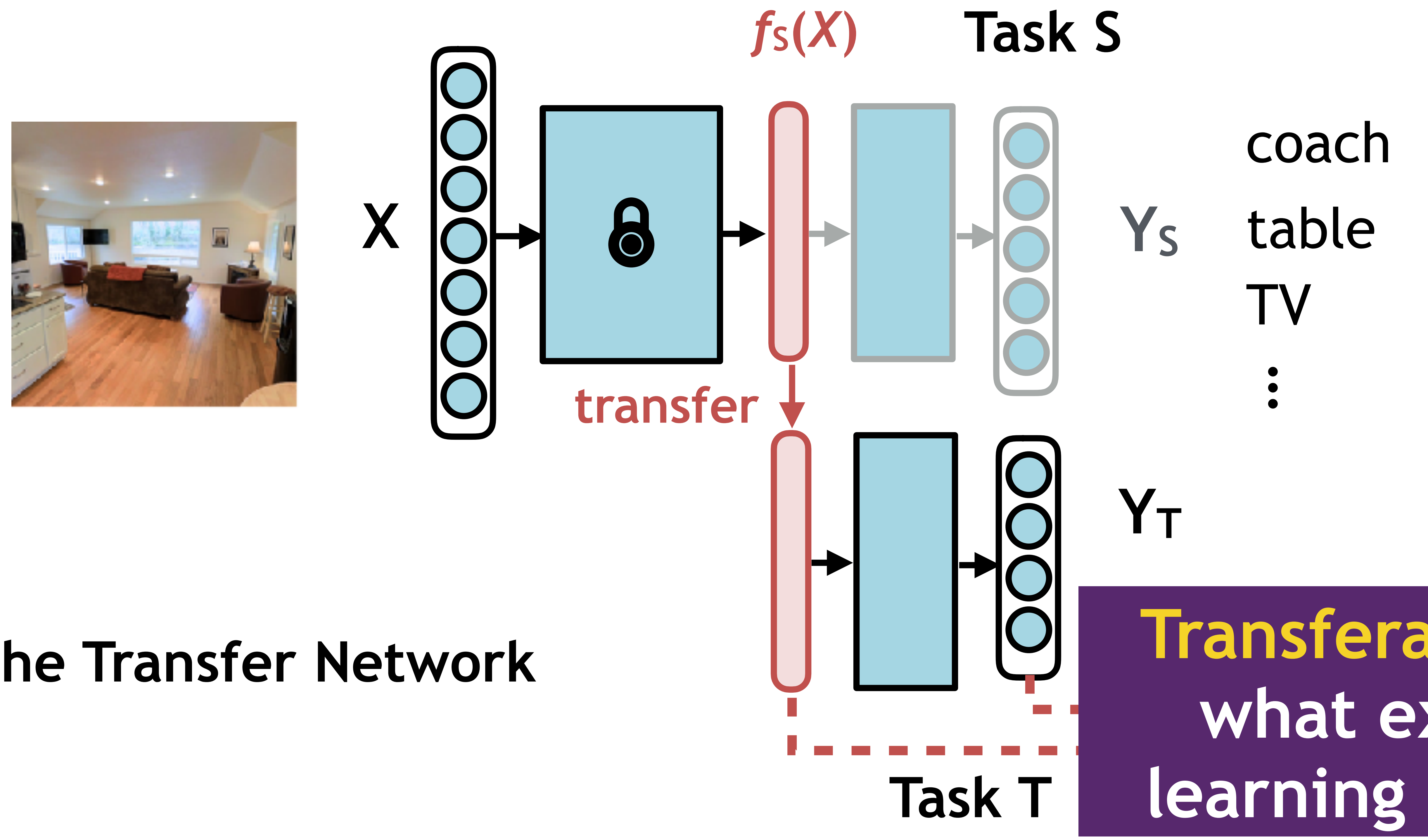


The Transfer Network

The Task Transferability Problem

Given:

- Input X, source task label Y_S, target task label Y_T
- Trained source model with optimal feature $f_S(X)$



The Transfer Network

Transferability of S→T : to what extent can f_S help learning target task (X,Y_T)?

Related Works — Theoretical Results

Why does transfer learning work?

- Inductive bias learning (Baxter 2000): Learning with multiple related tasks generalize better to novel tasks
- Transfer bounds for linear feature learning (Maurer 2009)

Related Works — Theoretical Results

Why does transfer learning work?

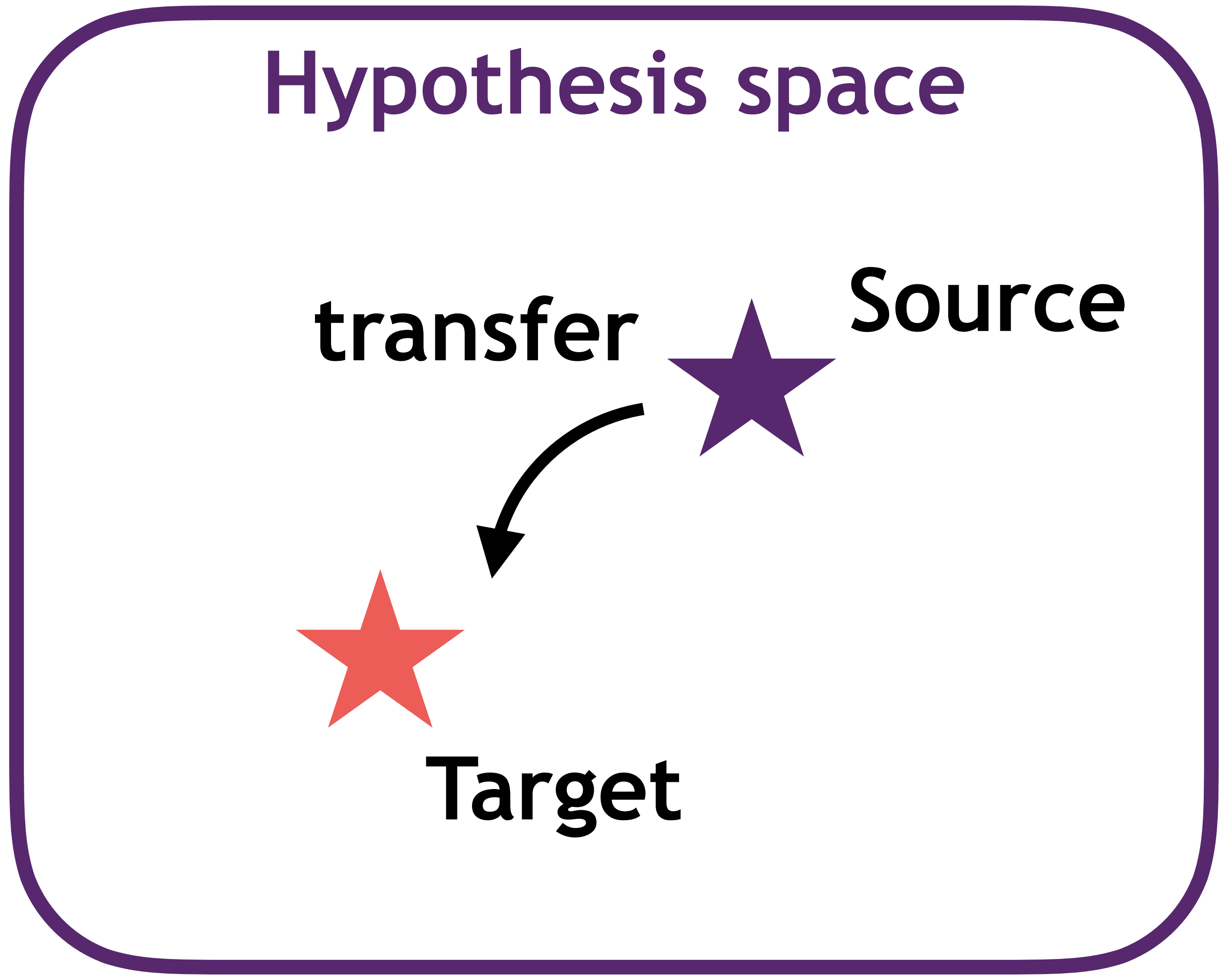
- Inductive bias learning (Baxter 2000): Learning with multiple related tasks generalize better to novel tasks

- Transfer bounds for linear feature learning (Maurer 2009)

Limitation

- Assumes hypotheses of all tasks are within an *environment* of related tasks

- Can not be computed directly from data



Related Works — Empirical Transferability

Empirical Approach: measure transfer results based on model loss / accuracy

- e.g. Feature transferability in Neural Network (Yosinski 2014), Taskonomy (Zamir et. al 2018), Shape Inductive Biases (Feinman & Lake 2018)

Related Works — Empirical Transferability

Empirical Approach: measure transfer results based on model loss / accuracy

- e.g. Feature transferability in Neural Network (Yosinski 2014), Taskonomy (Zamir et. al 2018), Shape Inductive Biases (Feinman & Lake 2018)

Limitation:

- need to train the transfer network using gradient descend
- inefficient

Related Works — Empirical Transferability

Empirical Approach: measure transfer results based on model loss / accuracy

- e.g. Feature transferability in Neural Network (Yosinski 2014), Taskonomy (Zamir et. al 2018), Shape Inductive Biases (Feinman & Lake 2018)

Limitation:

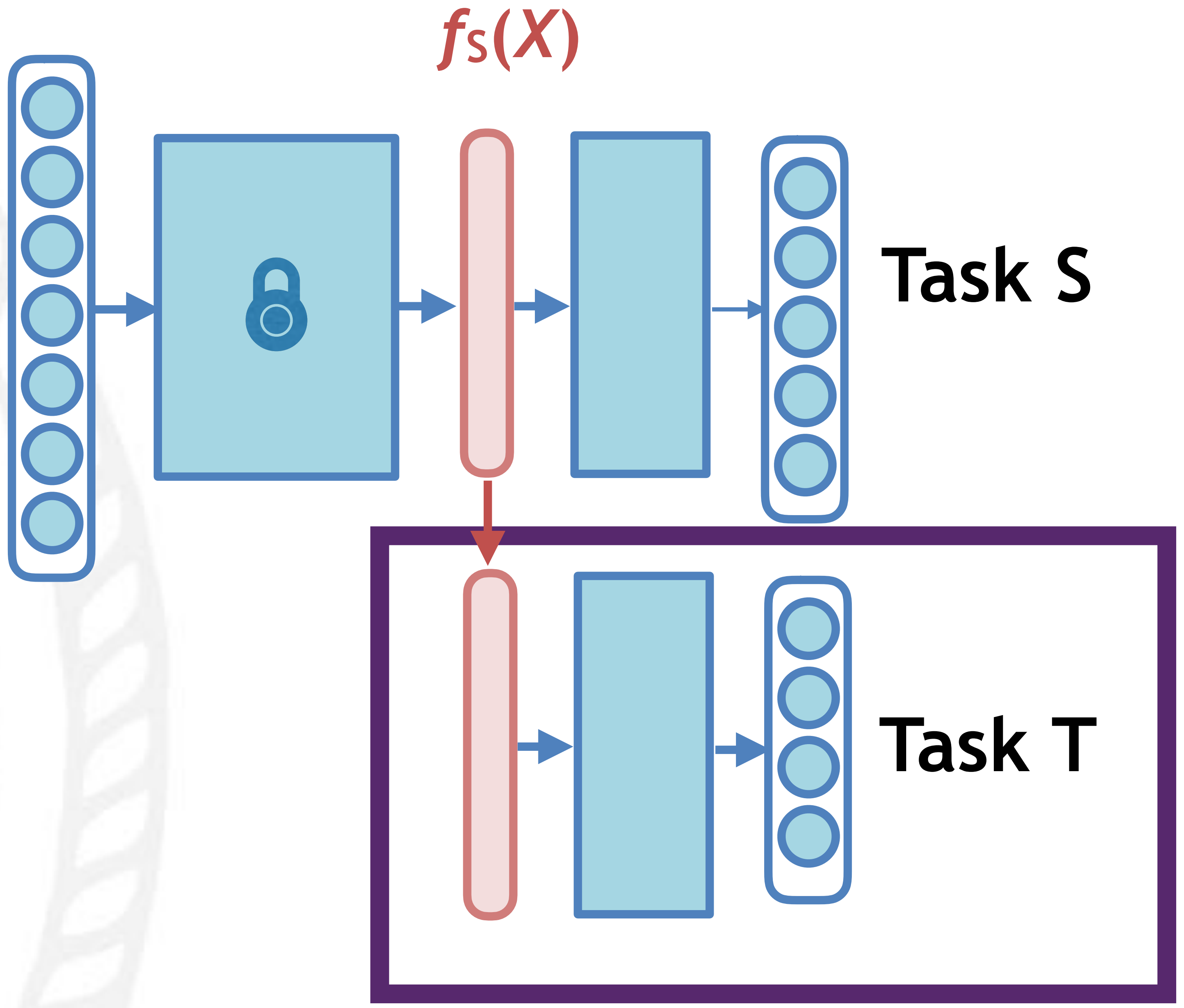
- need to train the transfer network using gradient descend
- inefficient

Can we estimate the transfer performance without any training of the target network?

Task Transferability

Transferability from Task S to Task T

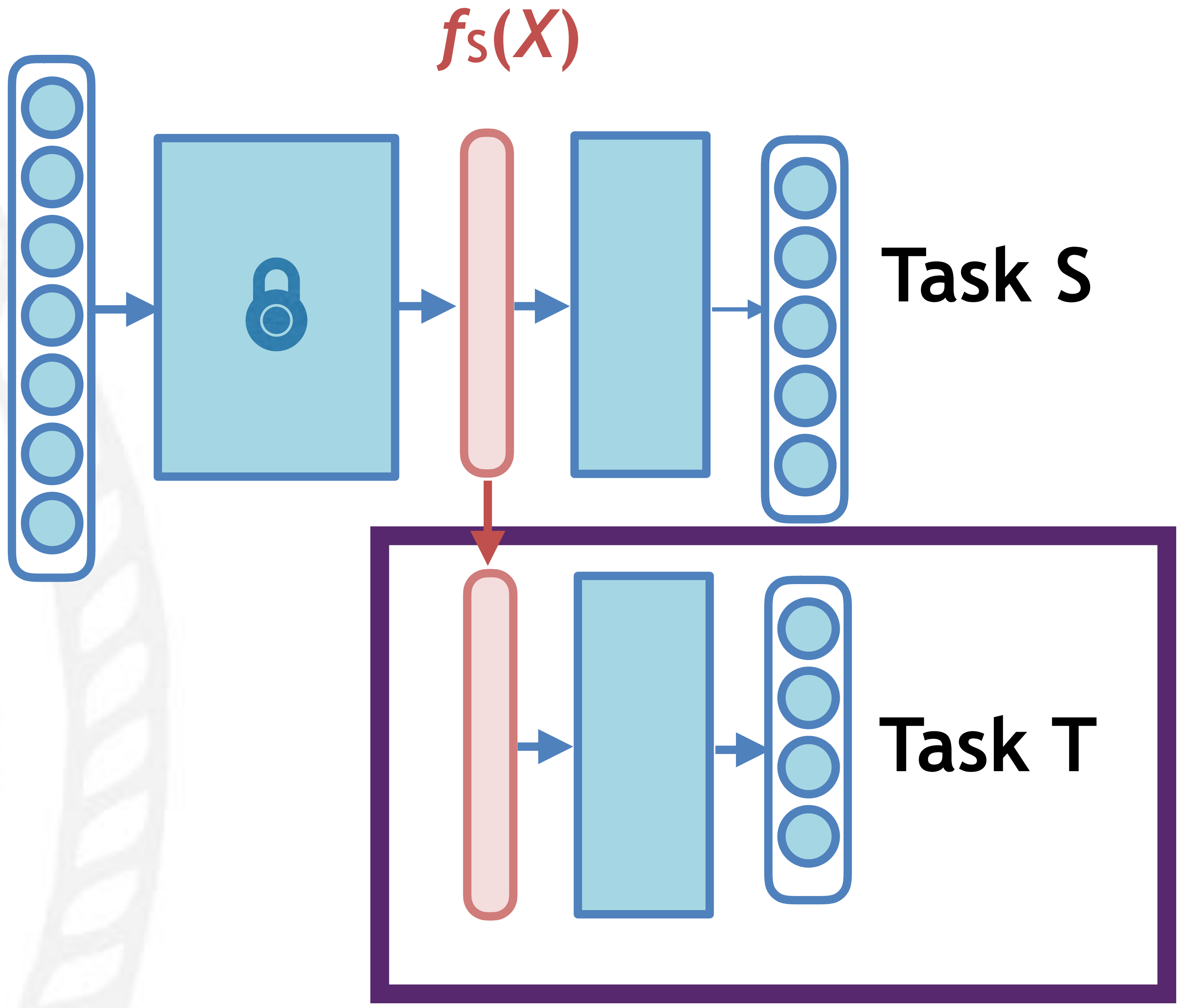
$$\mathfrak{T}(S,T) \triangleq \frac{\text{Target Performance of } f_S}{\text{Optimal Target Performance}}$$



Task Transferability

Transferability from Task S to Task T

$$\mathfrak{T}(S,T) \triangleq \frac{\text{Target Performance of } f_S}{\text{Optimal Target Performance}}$$

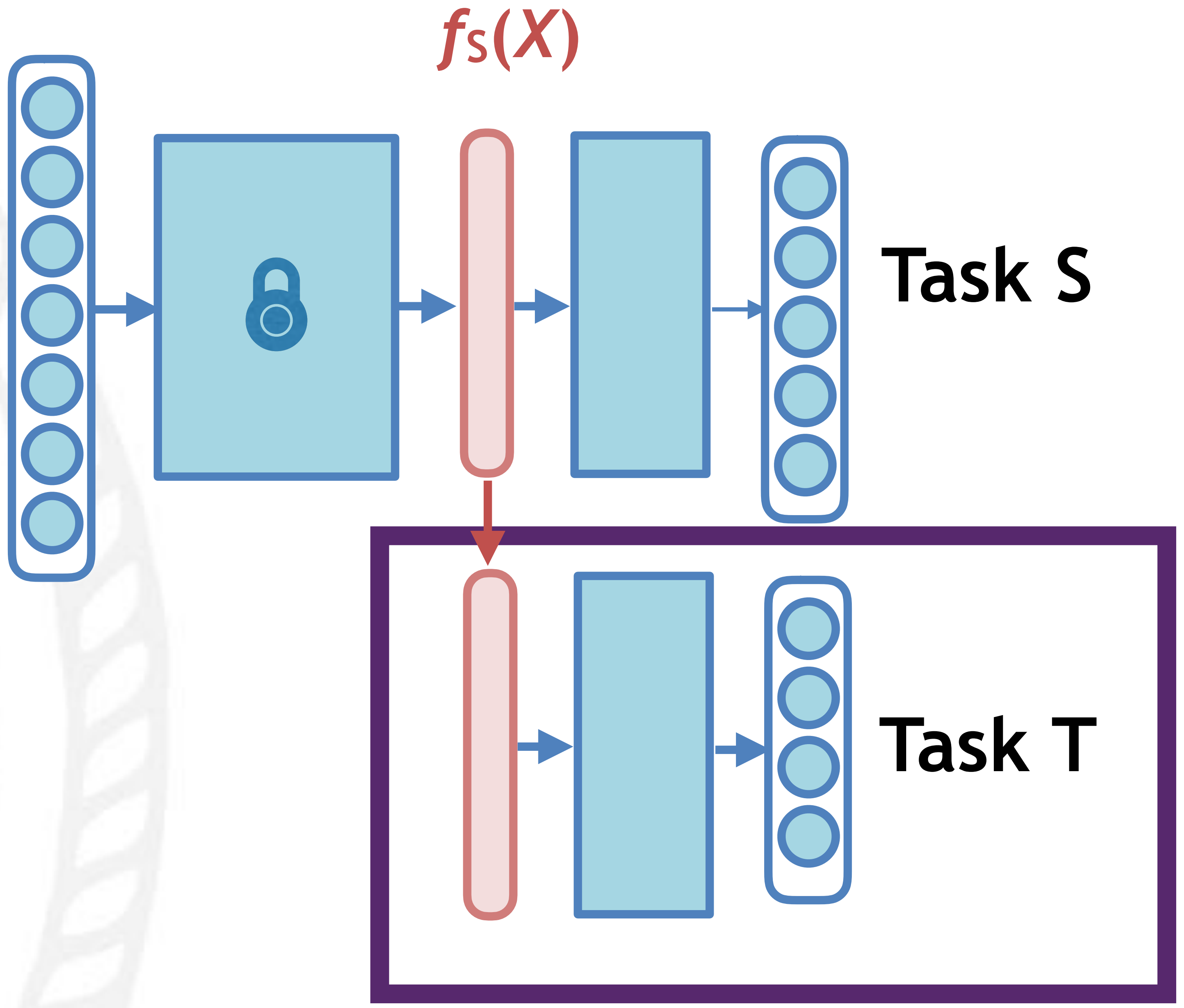


$$\begin{cases} \mathfrak{T}(S,T) = 1 & \text{☺} \\ 0 \leq \mathfrak{T}(S,T) \leq 1 & \\ \mathfrak{T}(S,T) = 0 & \text{☹} \end{cases}$$

Task Transferability

Transferability from Task S to Task T

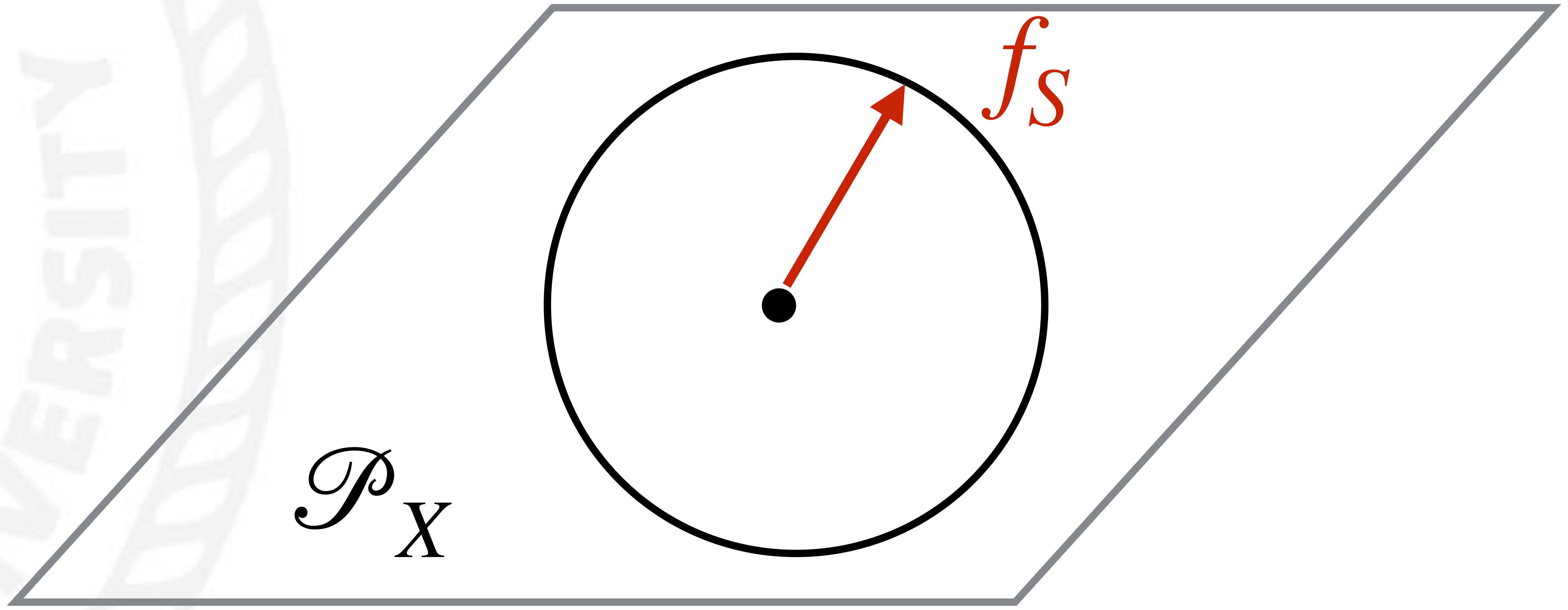
$$\mathfrak{T}(S,T) \triangleq \frac{\text{Target Performance of } f_S}{\text{Optimal Target Performance}}$$



$$\begin{cases} \mathfrak{T}(S,T) = 1 & \text{☺} \\ 0 \leq \mathfrak{T}(S,T) \leq 1 & \\ \mathfrak{T}(S,T) = 0 & \text{☹} \end{cases}$$

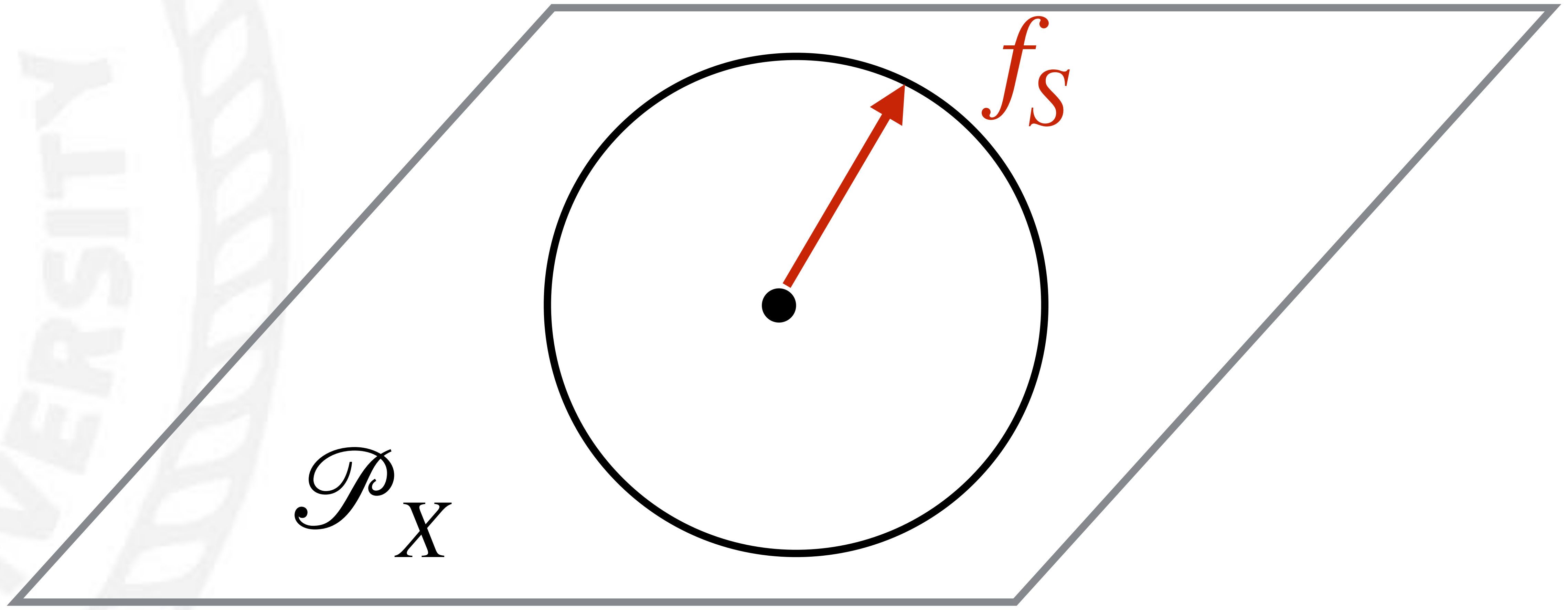
How to measure the performance of $f_S(X)$ on target task (X, Y_T) ?

Measuring Feature Performance via Information Geometry



Measuring Feature Performance via Information Geometry

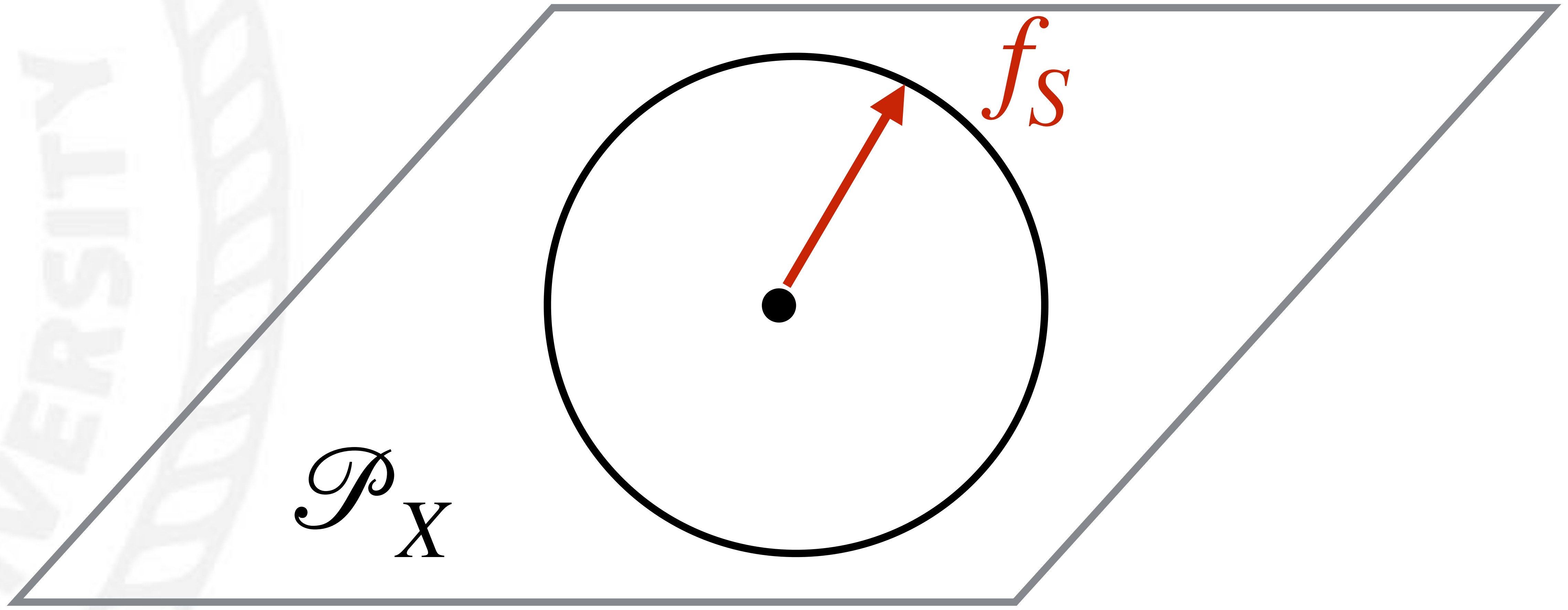
Local information geometry (Huang et al. 2017)



Measuring Feature Performance via Information Geometry

Local information geometry (Huang et al. 2017)

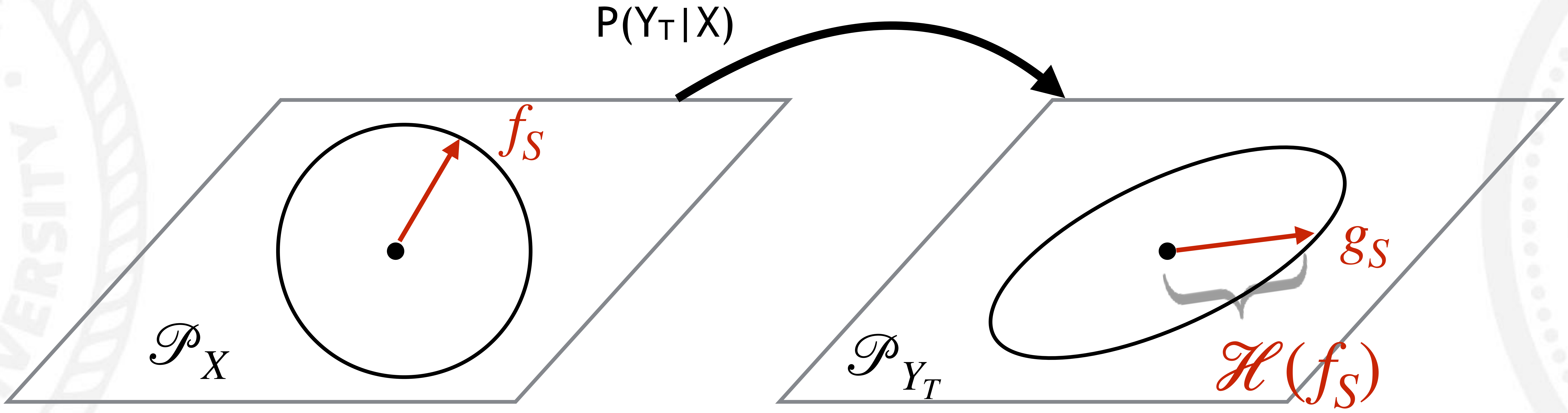
- Represent any feature f(X) as a unit vector in the distribution space of X



Measuring Feature Performance via Information Geometry

Local information geometry (Huang et al. 2017)

- Represent any feature f(X) as a unit vector in the distribution space of X
- P(Y$_T$|X) : a map between distribution spaces of X and Y$_T$

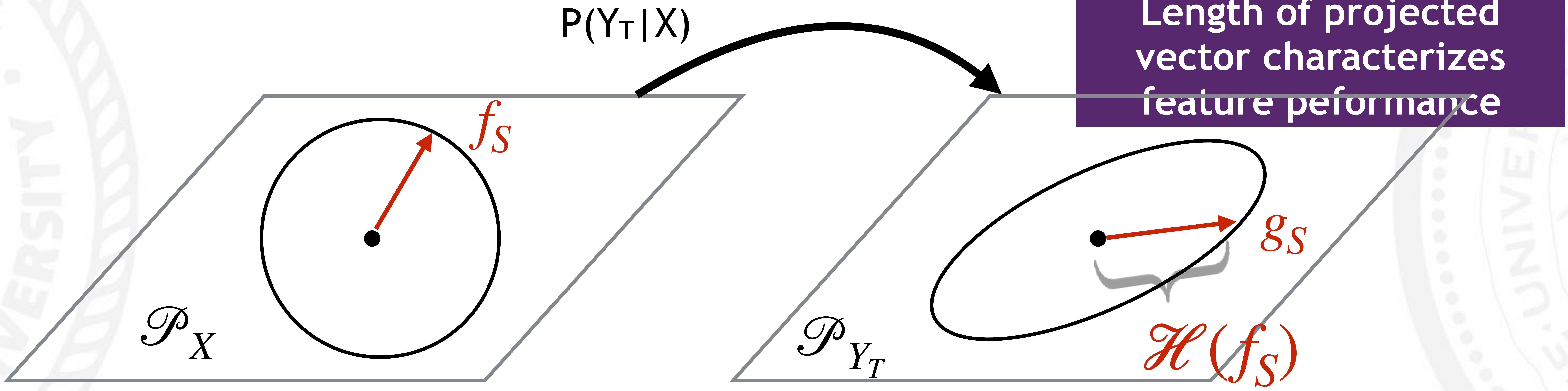


$$= \mathrm{tr}(\mathrm{cov}(f_S(X))^{-1}\mathrm{cov}(\mathbb{E}_{X|Y_T}[f_S(X)\,|\,Y_T]))$$

Measuring Feature Performance via Information Geometry

Local information geometry (Huang et al. 2017)

- Represent any feature f(X) as a unit vector in the distribution space of X
- P(Y_T|X) : a map between distribution spaces of X and Y_T



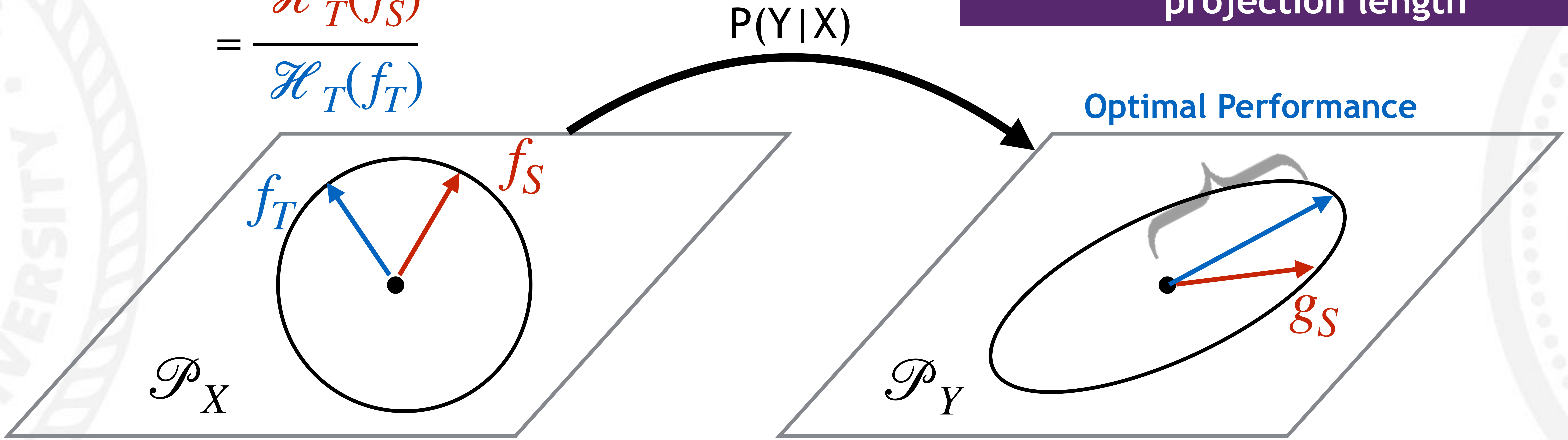
Length of projected vector characterizes feature performance

$$= \mathrm{tr}(\mathrm{cov}(f_S(X))^{-1}\mathrm{cov}(\mathbb{E}_{X|Y_T}[f_S(X)\,|\,Y_T]))$$

Measuring Feature Performance via Information Geometry

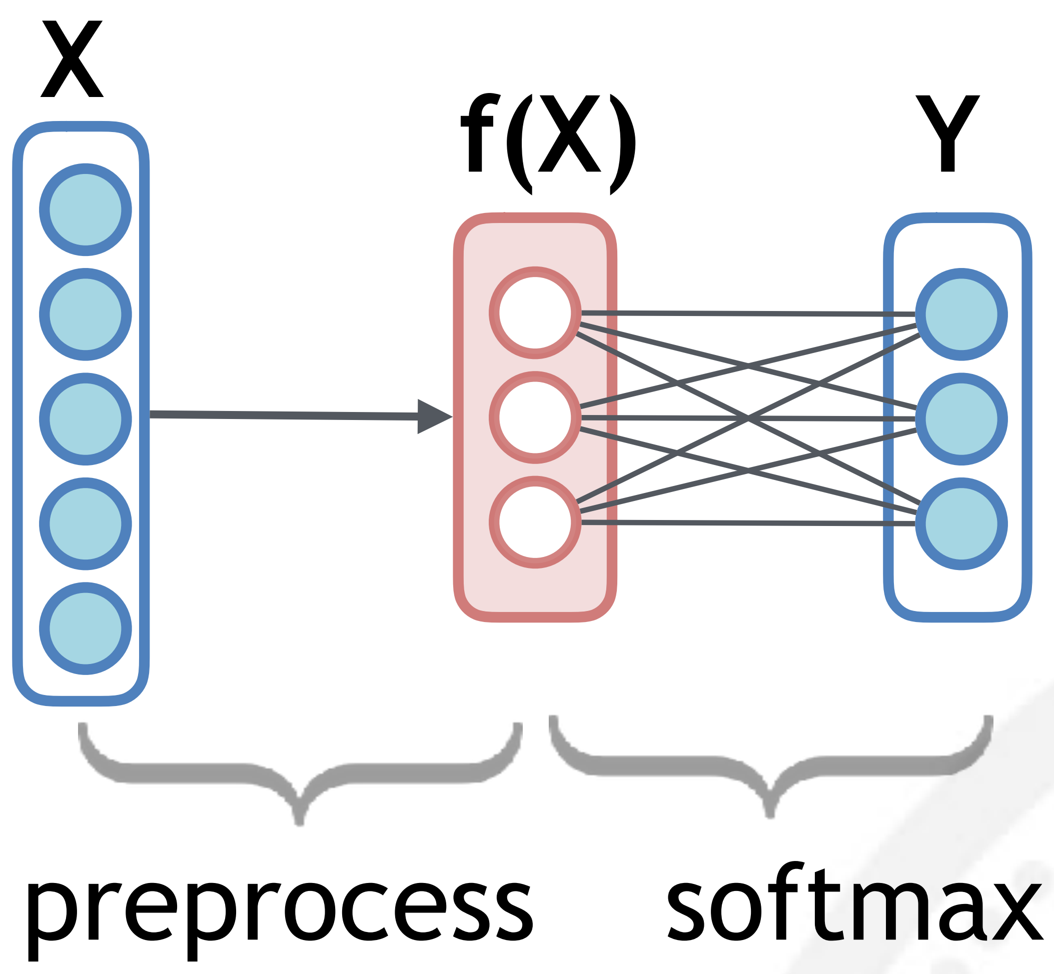
- Feature with maximum projection length: f_T
- $$\mathfrak{T}(S,T) \triangleq \frac{\text{Target Performance of } f_S}{\text{Optimal Target Performance}} = \frac{\mathscr{H}_T(f_S)}{\mathscr{H}_T(f_T)}$$

Transferability S→T: normalized projection length

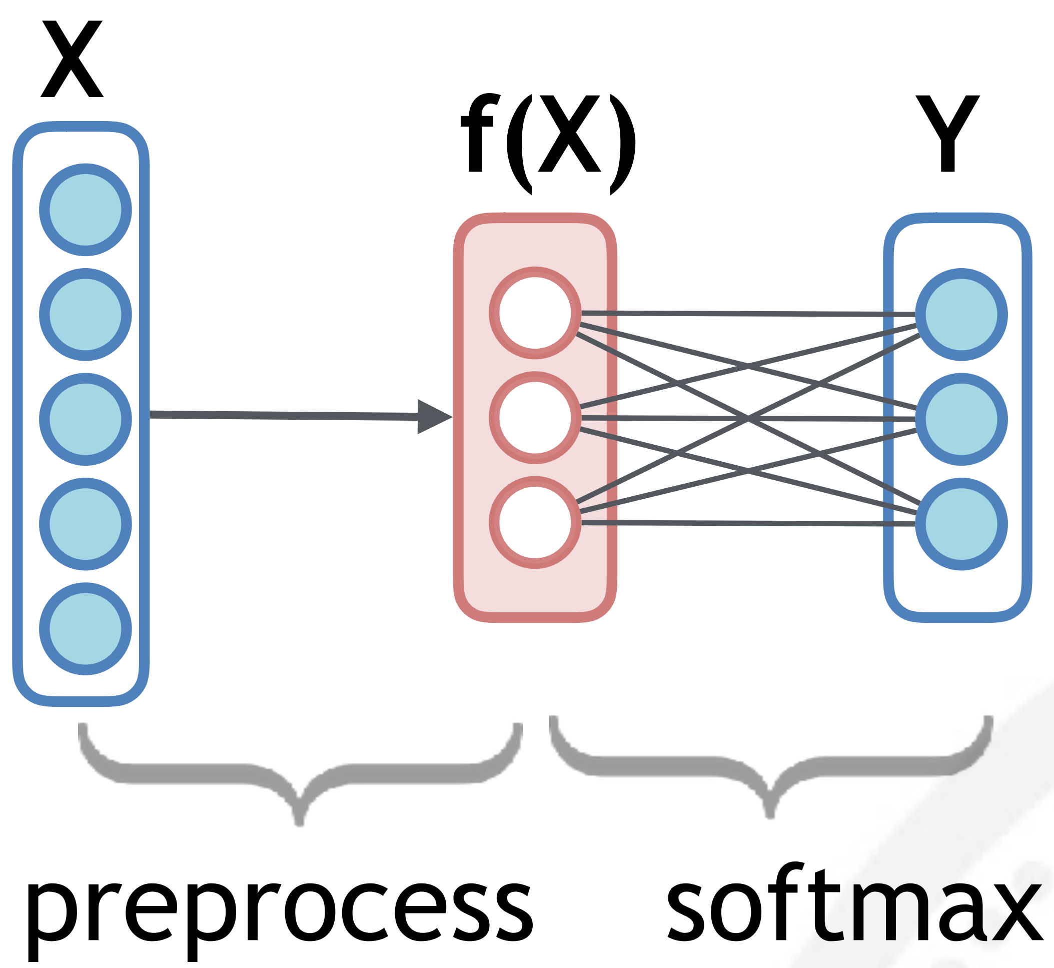


Measuring Feature Effectiveness - Neural Network Perspective

- Classification using log-loss:
  - X, Y random variables; f(X) a zero-mean feature
  - Expected log loss: $L(f;\theta) = \mathbb{E}_{X,Y}[L(f(X),Y;\theta)]$



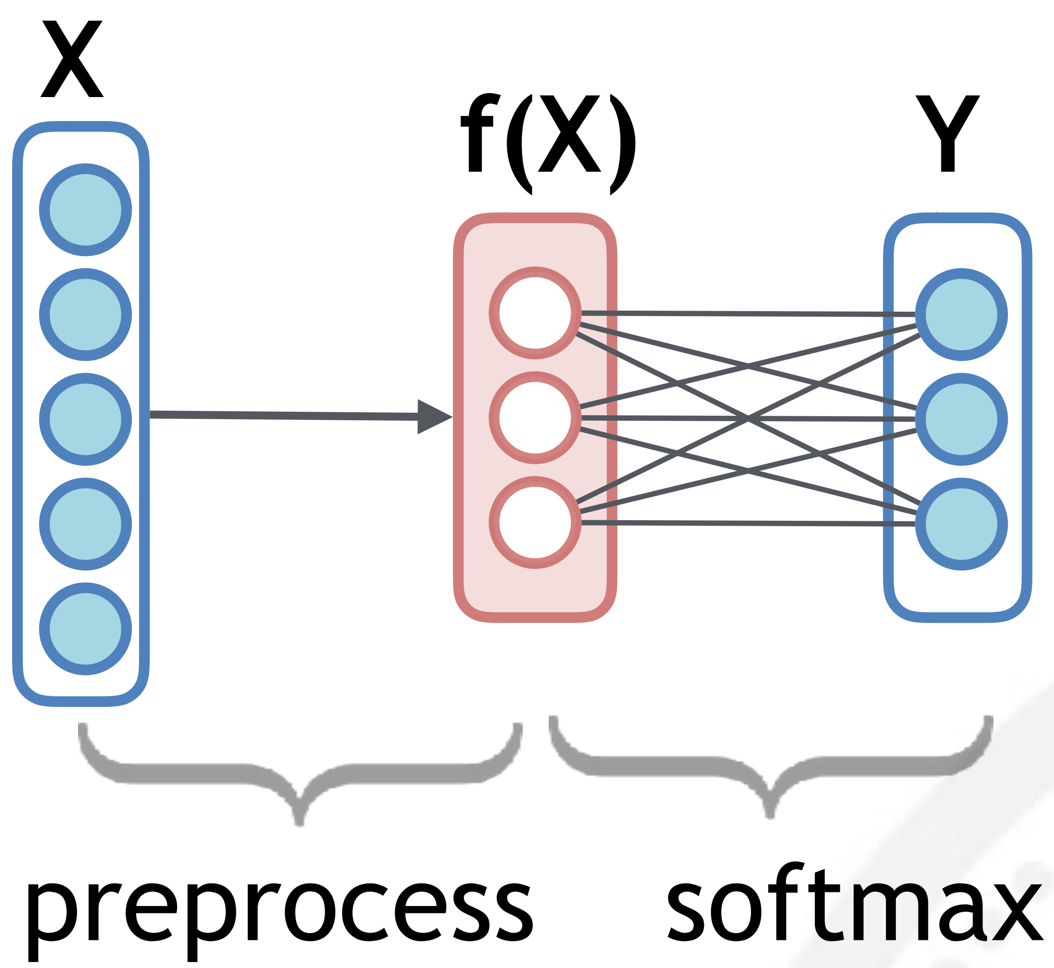
Measuring Feature Effectiveness - Neural Network Perspective



- Classification using log-loss:
  - X, Y random variables; f(X) a zero-mean feature
  - Expected log loss: $L(f;\theta) = \mathbb{E}_{X,Y}[L(f(X),Y;\theta)]$
- By Local information geometry [Huang 2018], given feature f(X), the optimal loss is

$$L(f,\theta^\star) = Const(X,Y) - H(f) + o(\epsilon^2)$$

Measuring Feature Effectiveness - Neural Network Perspective



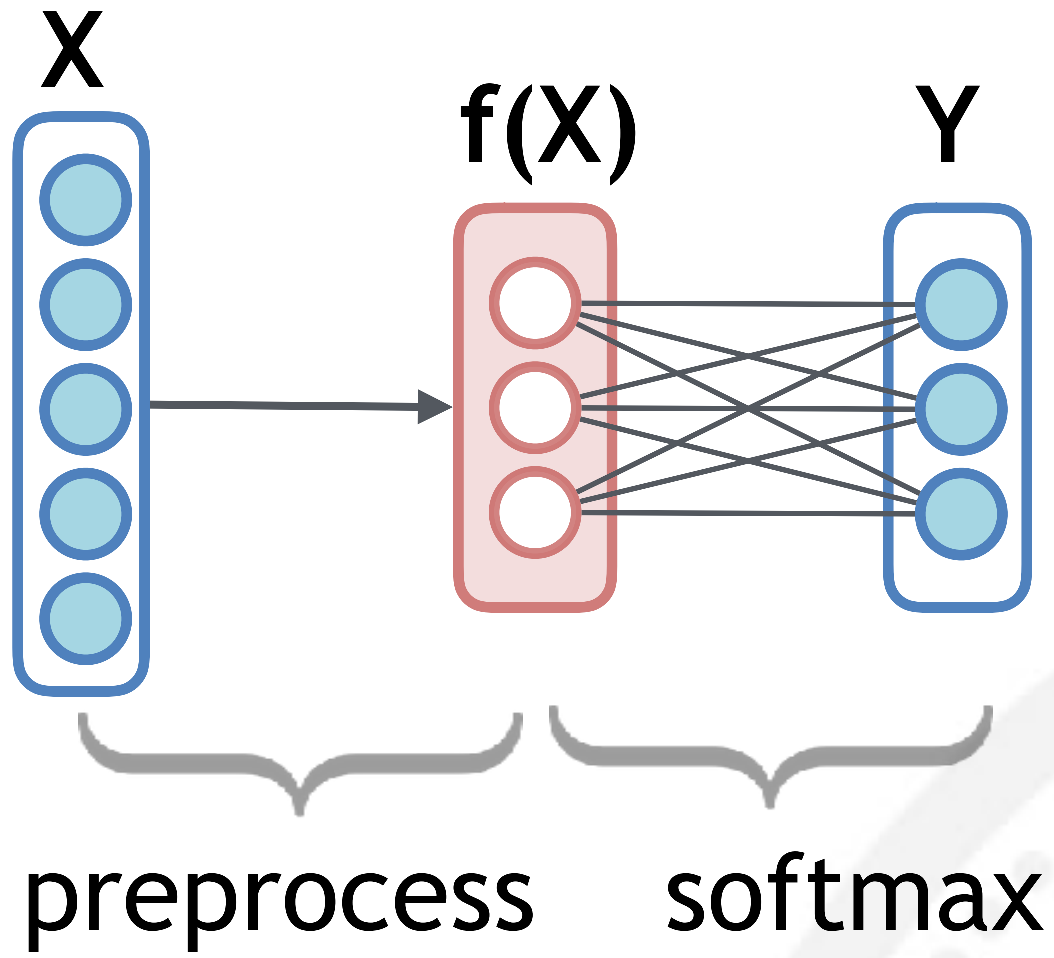
- Classification using log-loss:
  - X, Y random variables; f(X) a zero-mean feature
  - Expected log loss: $L(f;\theta) = \mathbb{E}_{X,Y}[L(f(X),Y;\theta)]$
- By Local information geometry [Huang 2018], given feature f(X), the optimal loss is

$$L(f,\theta^{\star}) = Const(X,Y) - \underbrace{H(f)}_{\text{H-score of f(X)}} + o(\epsilon^2)$$

$$\boxed{\mathcal{H}(f) = \mathrm{tr}(\mathrm{cov}(f(X))^{-1}\mathrm{cov}(\mathbb{E}_{P_{X|Y}}[f(X)|Y]))}$$

Measuring Feature Effectiveness - Neural Network Perspective

- Classification using log-loss:
  - X, Y random variables; f(X) a zero-mean feature
  - Expected log loss: $L(f;\theta) = \mathbb{E}_{X,Y}[L(f(X),Y;\theta)]$



- By Local information geometry [Huang 2018], given feature f(X), the optimal loss is

$$L(f,\theta^\star) = Const(X,Y) - \underbrace{H(f)}_{\text{H-score of f(X)}} + o(\epsilon^2)$$

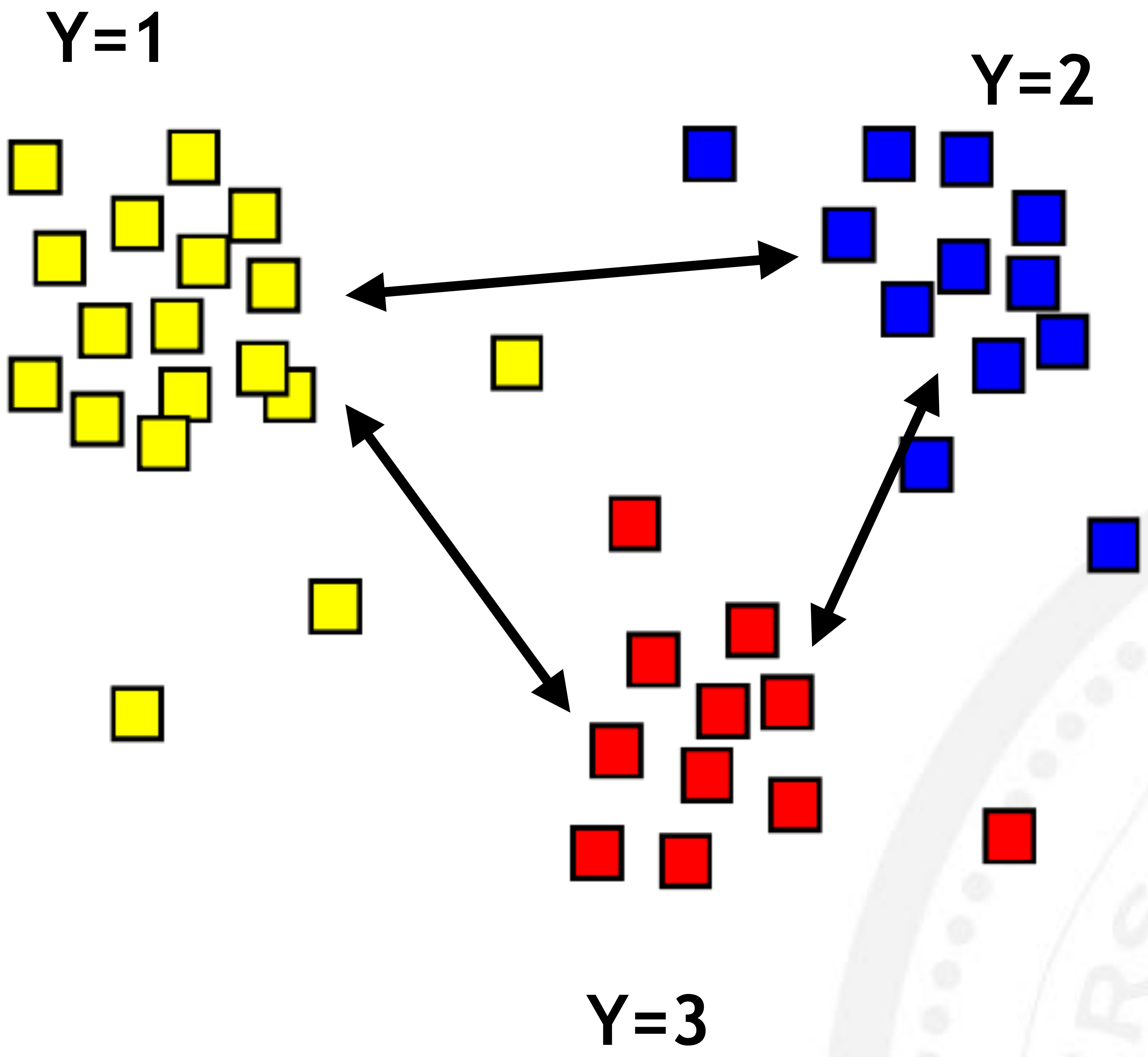
Higher H-score => Better Performance

$$\boxed{\mathcal{H}(f) = \mathrm{tr}(\mathrm{cov}(f(X))^{-1}\mathrm{cov}(\mathbb{E}_{P_{X|Y}}[f(X)|Y]))}$$

Interpretation of $\mathscr{H}(f)$

Intuition in latent space

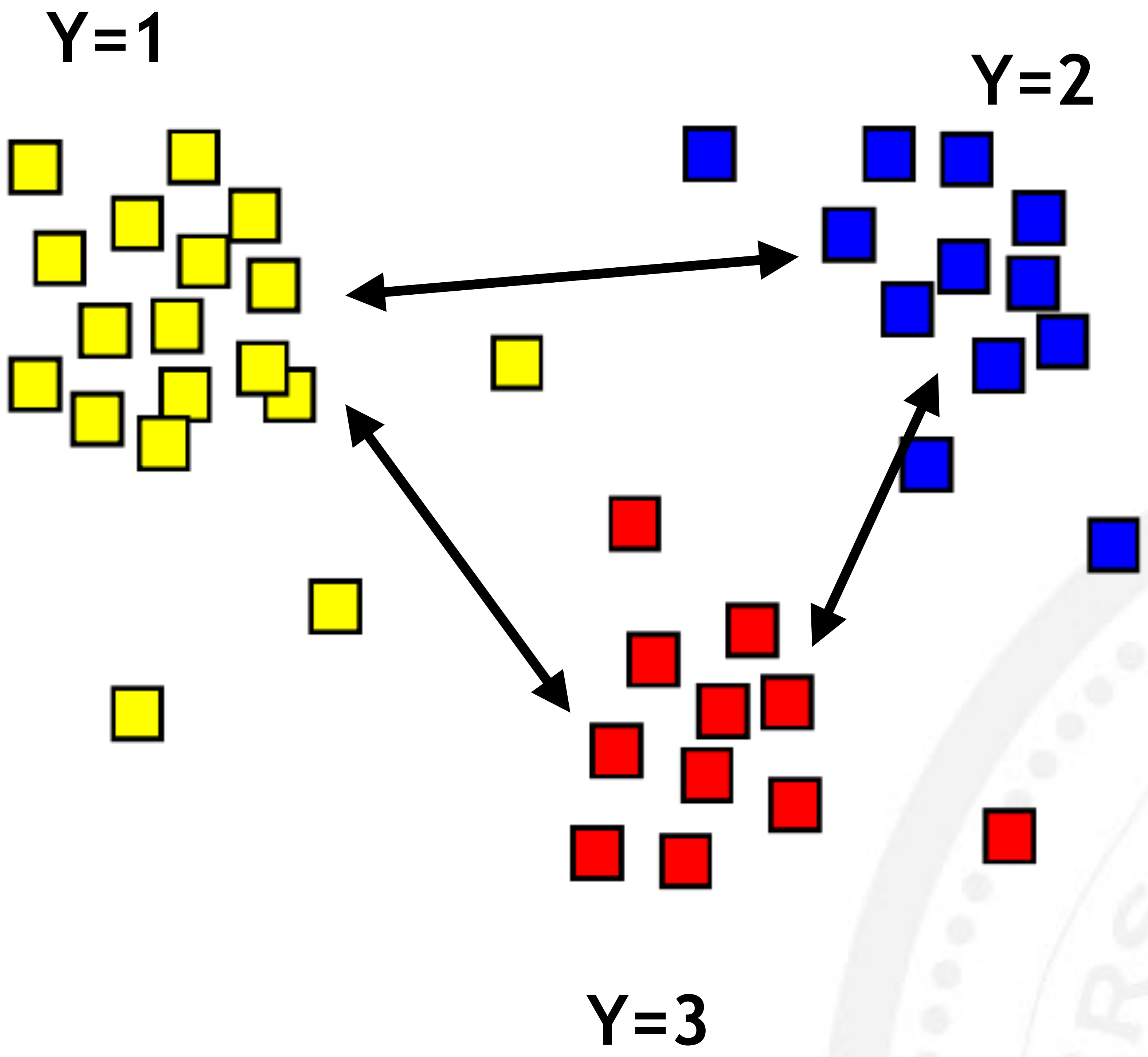
$$\mathscr{H}(f) = \mathrm{tr}(\mathrm{cov}(f(X))^{-1}\mathrm{cov}(\mathbb{E}_{X|Y}[f(X)\,|\,Y]))$$



Interpretation of $\mathscr{H}(f)$

Intuition in latent space

$$\mathscr{H}(f) = \underbrace{\boxed{\mathrm{tr}(\mathrm{cov}(f(X))}}_{\text{feature redundancy } \downarrow}{}^{-1}\mathrm{cov}(\mathbb{E}_{X|Y}[f(X) \mid Y]))$$



Interpretation of $\mathscr{H}(f)$

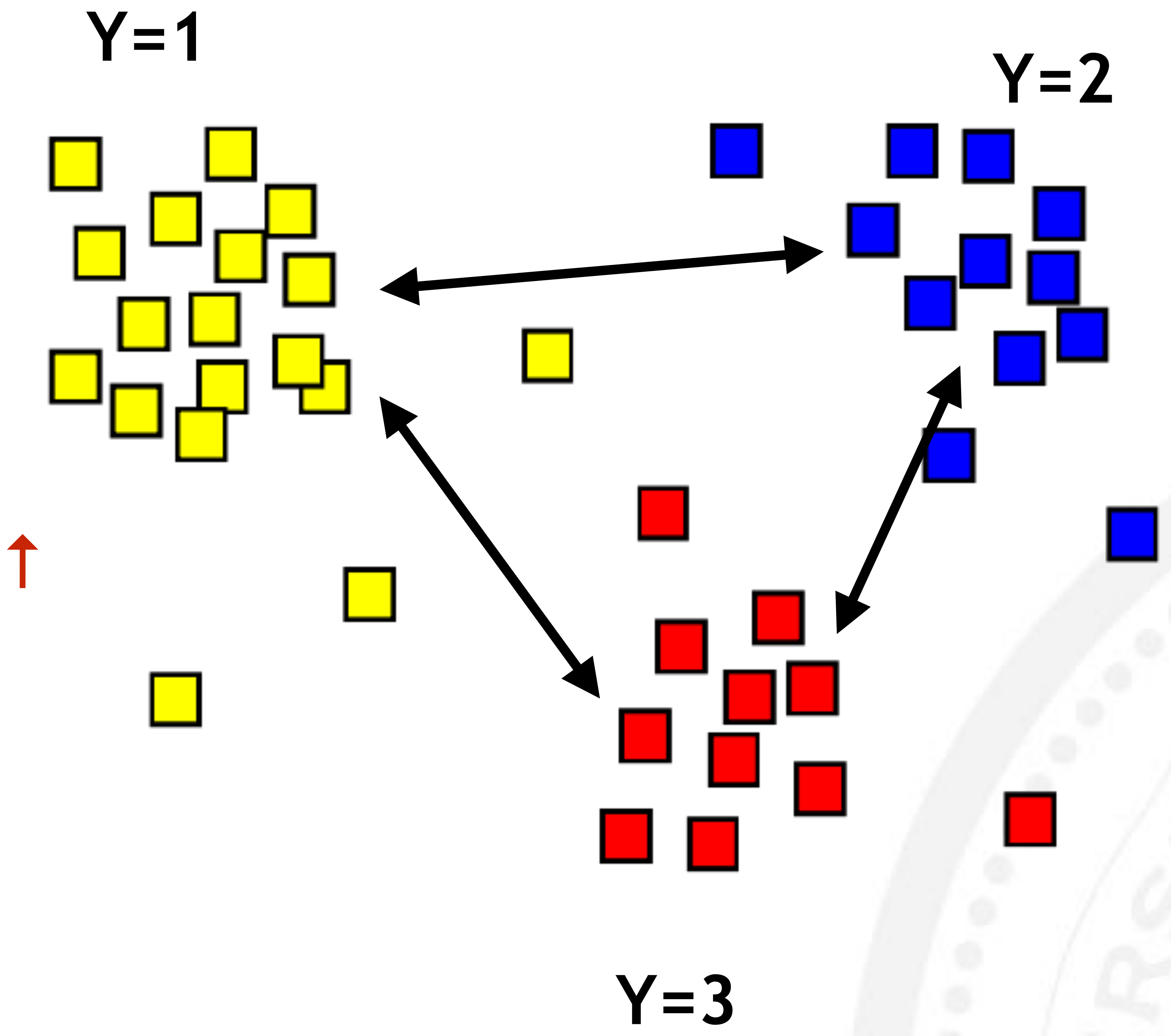
Intuition in latent space

$$\mathscr{H}(f) = \boxed{\mathrm{tr}(\mathrm{cov}(f(X))}^{-1}\boxed{\mathrm{cov}(\mathbb{E}_{X|Y}[f(X)\,|\,Y]))}$$

feature redundancy ↓

average intra-class distance ↑

$$\mathbb{E}[\|\mathbb{E}[f(X)\,|\,Y]\|^2]$$



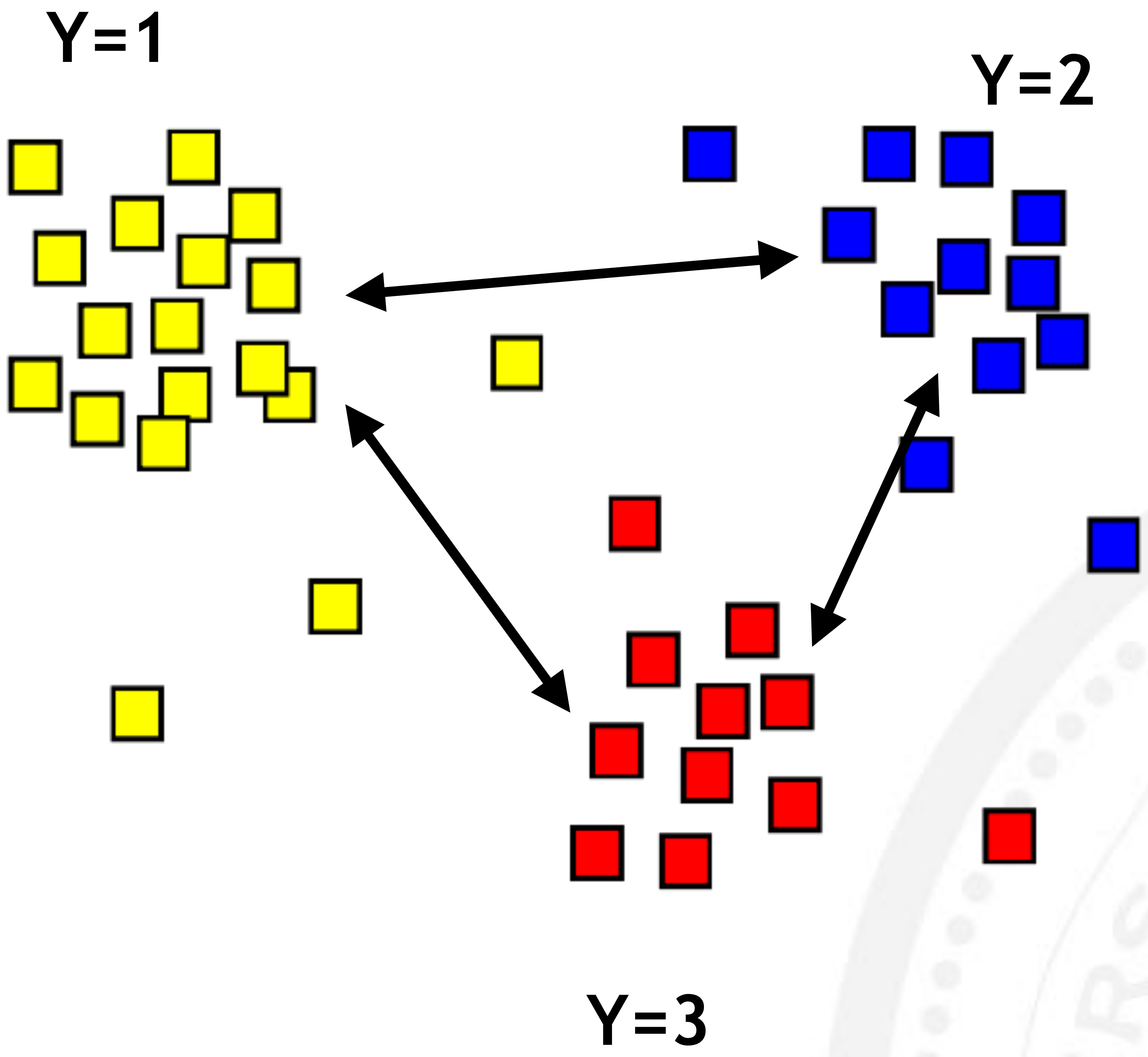
Interpretation of $\mathscr{H}(f)$

Intuition in latent space

$$\mathscr{H}(f) = \boxed{\mathrm{tr}(\mathrm{cov}(f(X))}^{-1}\boxed{\mathrm{cov}(\mathbb{E}_{X|Y}[f(X)\,|\,Y]))}$$

H-score ↑ feature redundancy ↓ average intra-class distance ↑

$$\mathbb{E}[\|\mathbb{E}[f(X)\,|\,Y]\|^2]$$



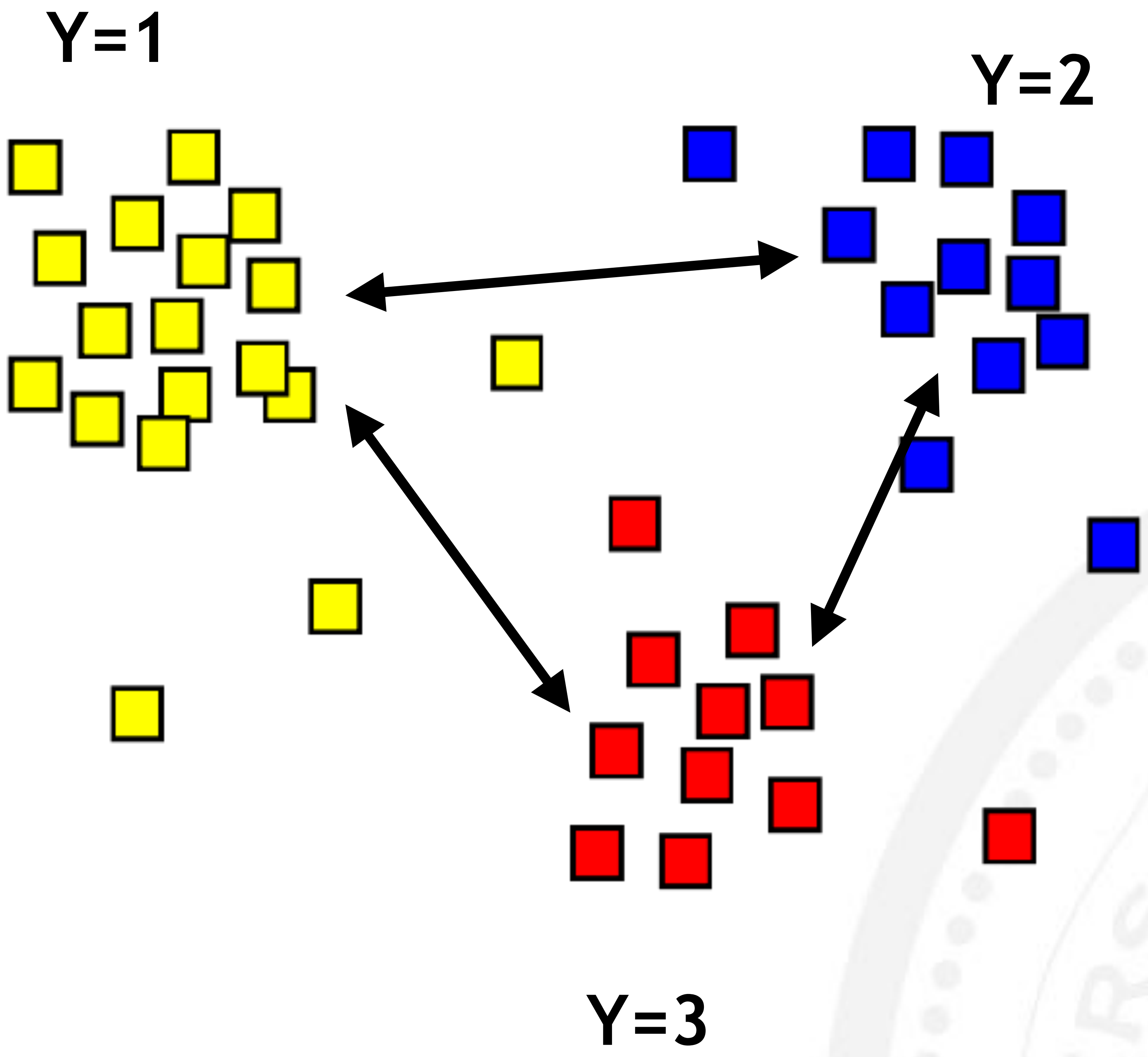
Interpretation of $\mathscr{H}(f)$

Intuition in latent space

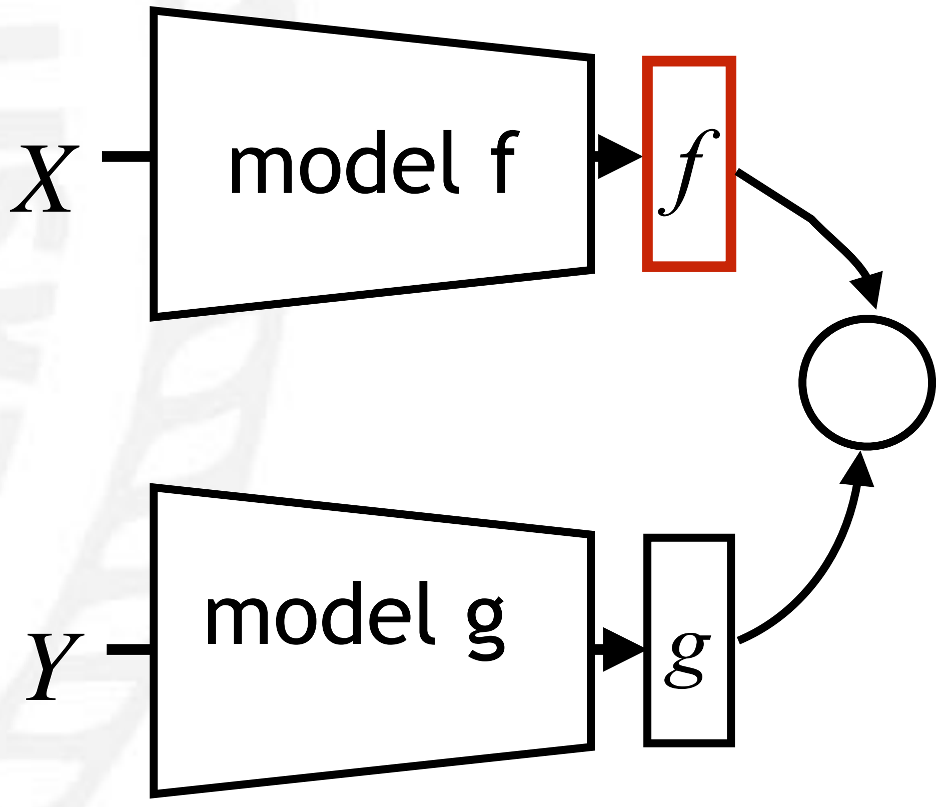
$$\mathscr{H}(f) = \boxed{\mathrm{tr}(\mathrm{cov}(f(X))}^{-1}\boxed{\mathrm{cov}(\mathbb{E}_{X|Y}[f(X)\,|\,Y]))}$$

H-score ↑ feature redundancy ↓ average intra-class distance ↑

$$\mathbb{E}[\|\mathbb{E}[f(X)\,|\,Y]\|^2]$$



Relationship with HGR maximal correlation



$$L = -2\mathbb{E}[f(X)^T g(Y)] + \mathrm{tr}(\mathrm{cov}(f(X))\mathrm{cov}(g(Y)))$$

When f is fixed, the maximum L is H-score, invariant to linear transformation

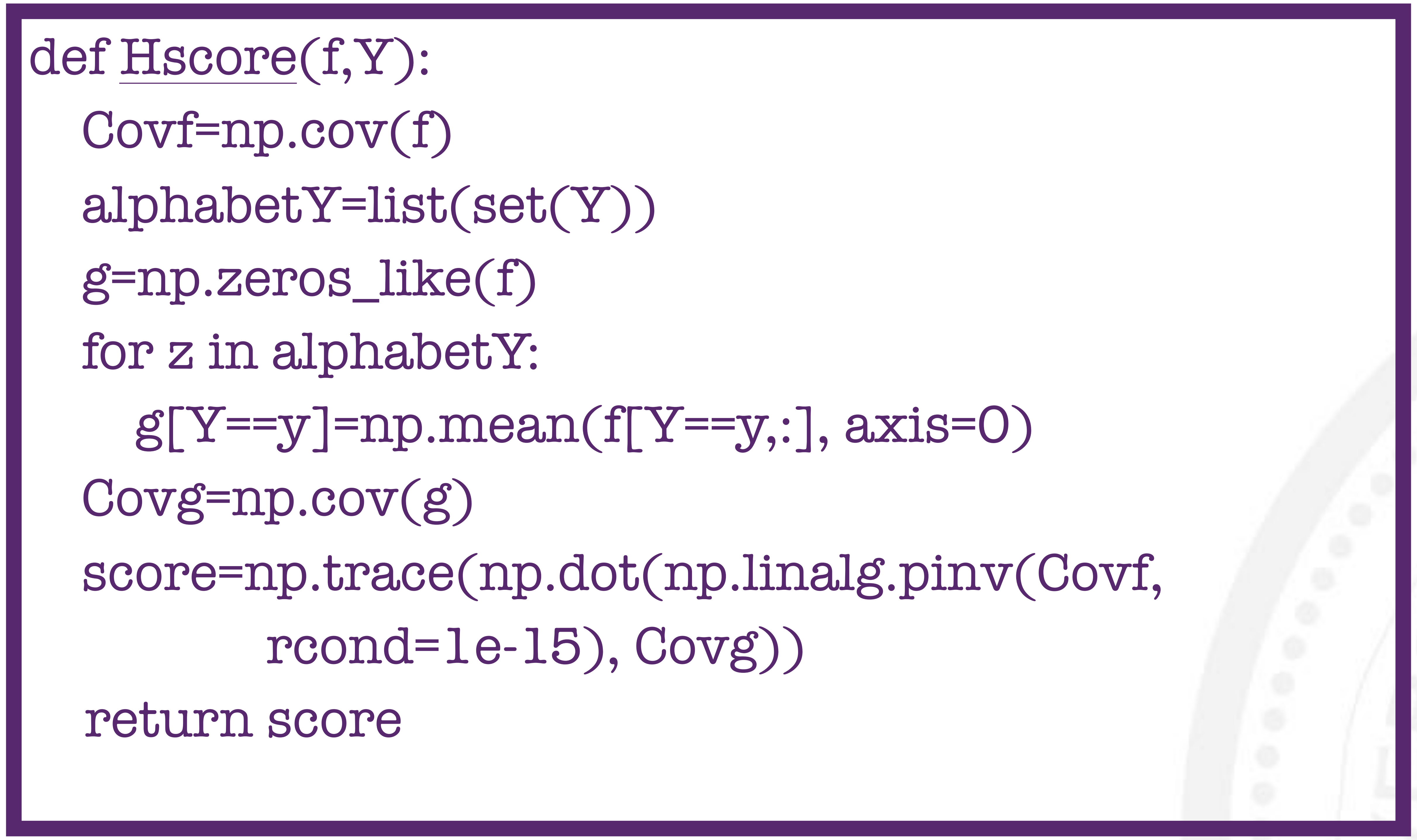
$$\mathscr{H}(f) = max_g L(f, g)$$

Computing Transferability

$$\mathfrak{T}(S, T) = \frac{\mathscr{H}_T(f_S)}{\mathscr{H}_T(f_T)}$$

- Computing H-score: $\mathscr{H}_T(f_S)$
  - Easy to compute
  - O(mk^2) time complexity

Python Code for H-Score



```
def Hscore(f,Y):
    Covf=np.cov(f)
    alphabetY=list(set(Y))
    g=np.zeros_like(f)
    for z in alphabetY:
        g[Y==y]=np.mean(f[Y==y,:], axis=0)
    Covg=np.cov(g)
    score=np.trace(np.dot(np.linalg.pinv(Covf,
                rcond=1e-15), Covg))
    return score
```

Computing Transferability

$$\mathfrak{T}(S, T) = \frac{\mathscr{H}_T(f_S)}{\mathscr{H}_T(f_T)}$$

- Computing H-score: $\mathscr{H}_T(f_S)$
  - Easy to compute
  - O(mk^2) time complexity
- Maximal H-score: $\mathscr{H}_T(f_T)$
  - Equivalent to computing the HGR maximal correlation
  - Discrete X: Alternating Conditional Expectation (ACE) algorithm (Huang et. al. 2015); Continuous X: Neural network formulation

Python Code for H-Score

```
def Hscore(f,Y):
    Covf=np.cov(f)
    alphabetY=list(set(Y))
    g=np.zeros_like(f)
    for z in alphabetY:
        g[Y==y]=np.mean(f[Y==y,:], axis=0)
    Covg=np.cov(g)
    score=np.trace(np.dot(np.linalg.pinv(Covf,
                rcond=1e-15), Covg))
    return score
```

Source Task Selection

$$\mathfrak{T}(S,T) \triangleq \frac{\textbf{Target Performance of } f_S}{\textbf{Optimal Target Performance}}$$

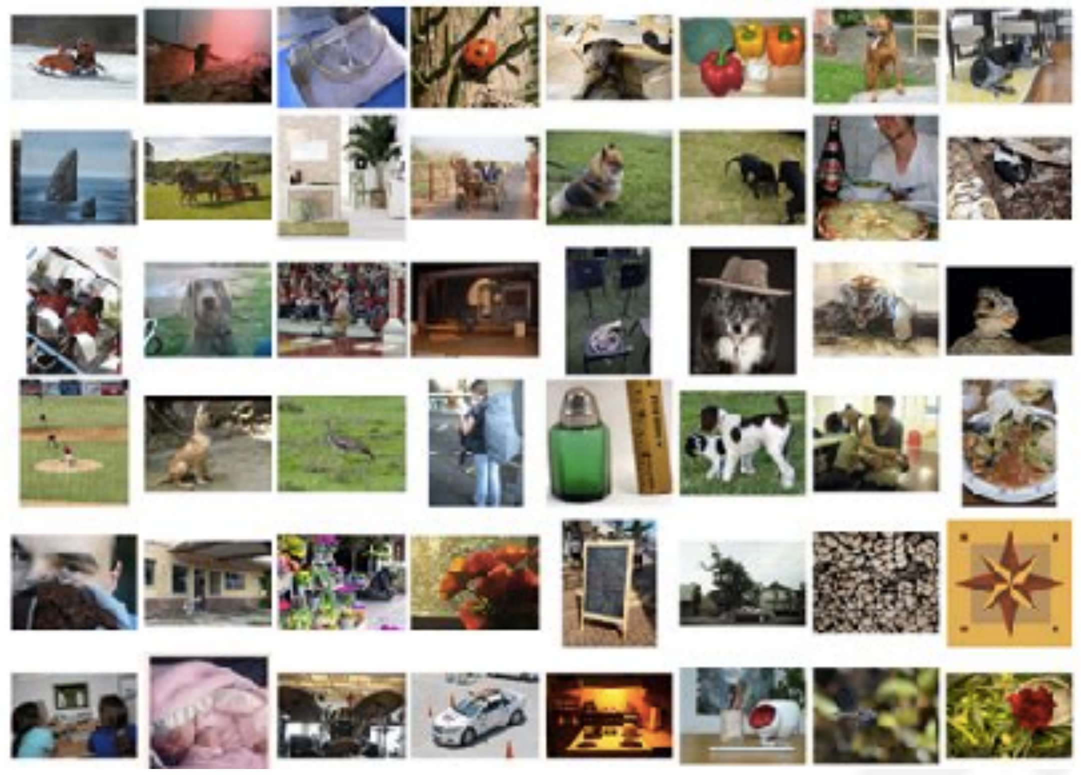
$$= \frac{\mathscr{H}_T(f_S)}{\mathscr{H}_T(f_T)}$$

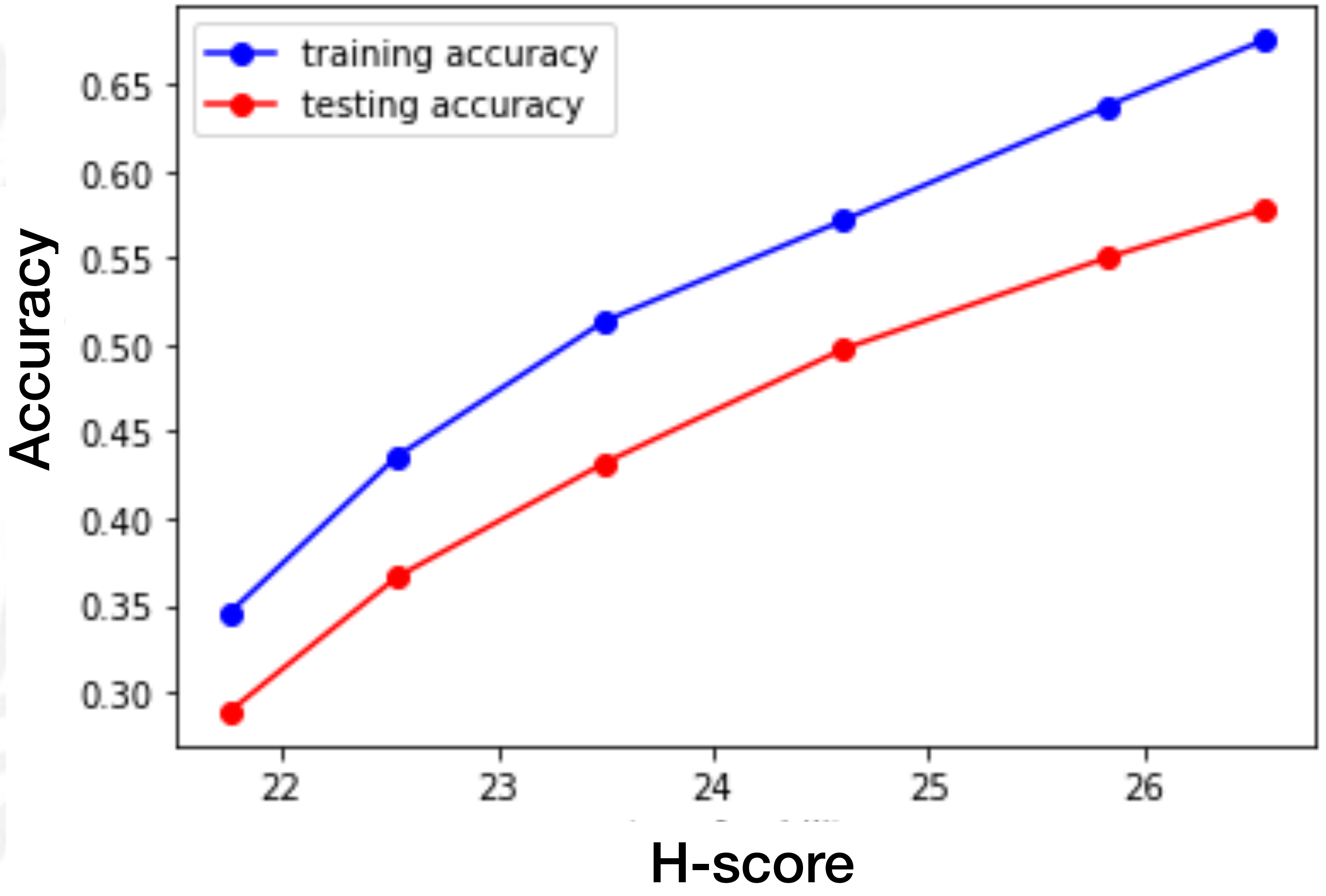
Source task selection problem: Given source tasks S_1, S_2, …, S_n. Which one is most transferable to target task T ?

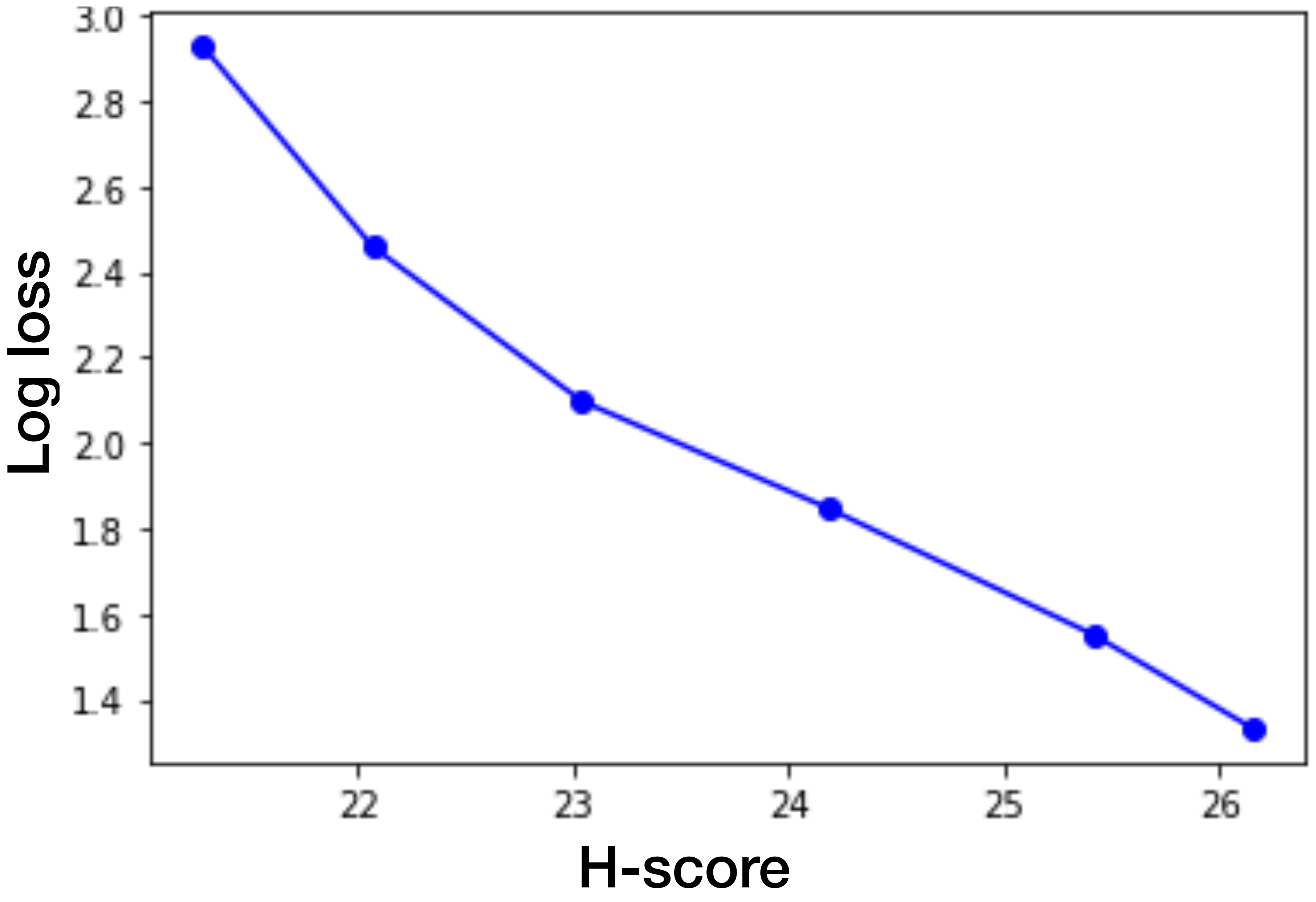
- Since T is fixed, we only need to compare $\mathscr{H}_T(f_{S_1}), \mathscr{H}_T(f_{S_2}), \ldots$

Results: Image Classification Feature Selection

- Source task: ImageNet 1000 classification (ResNet50 features from 6 layers 4a-5f)
- Target task: Cifar 100-class classification on 20,000 images

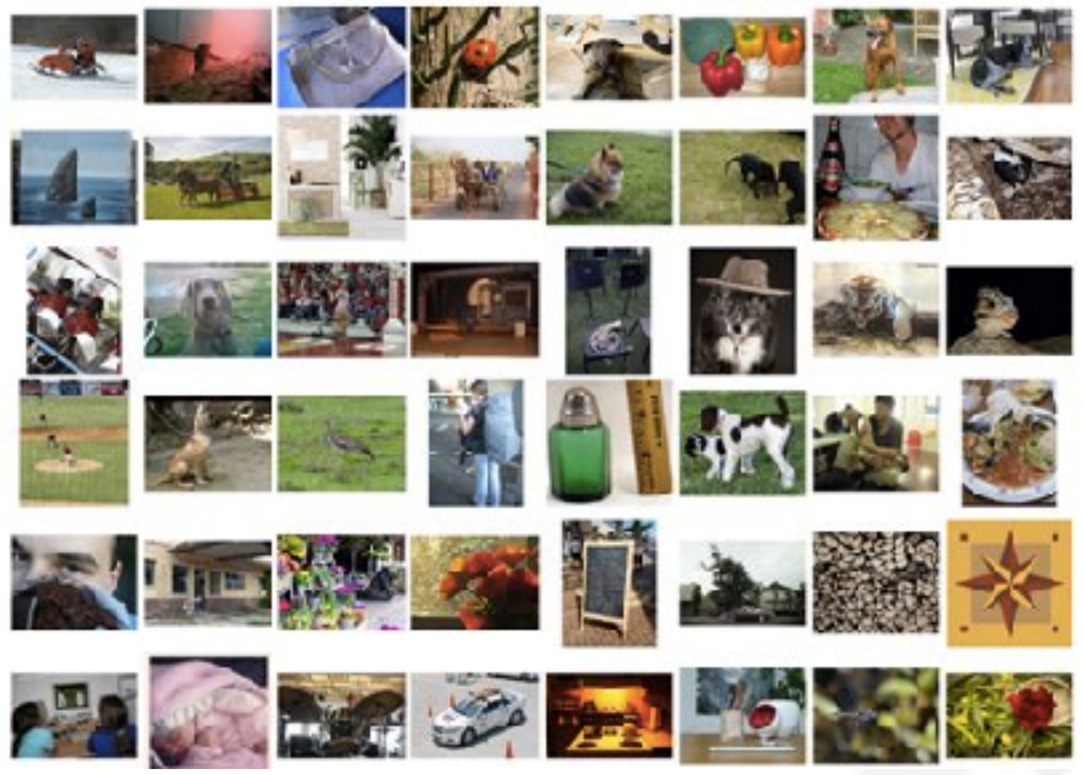


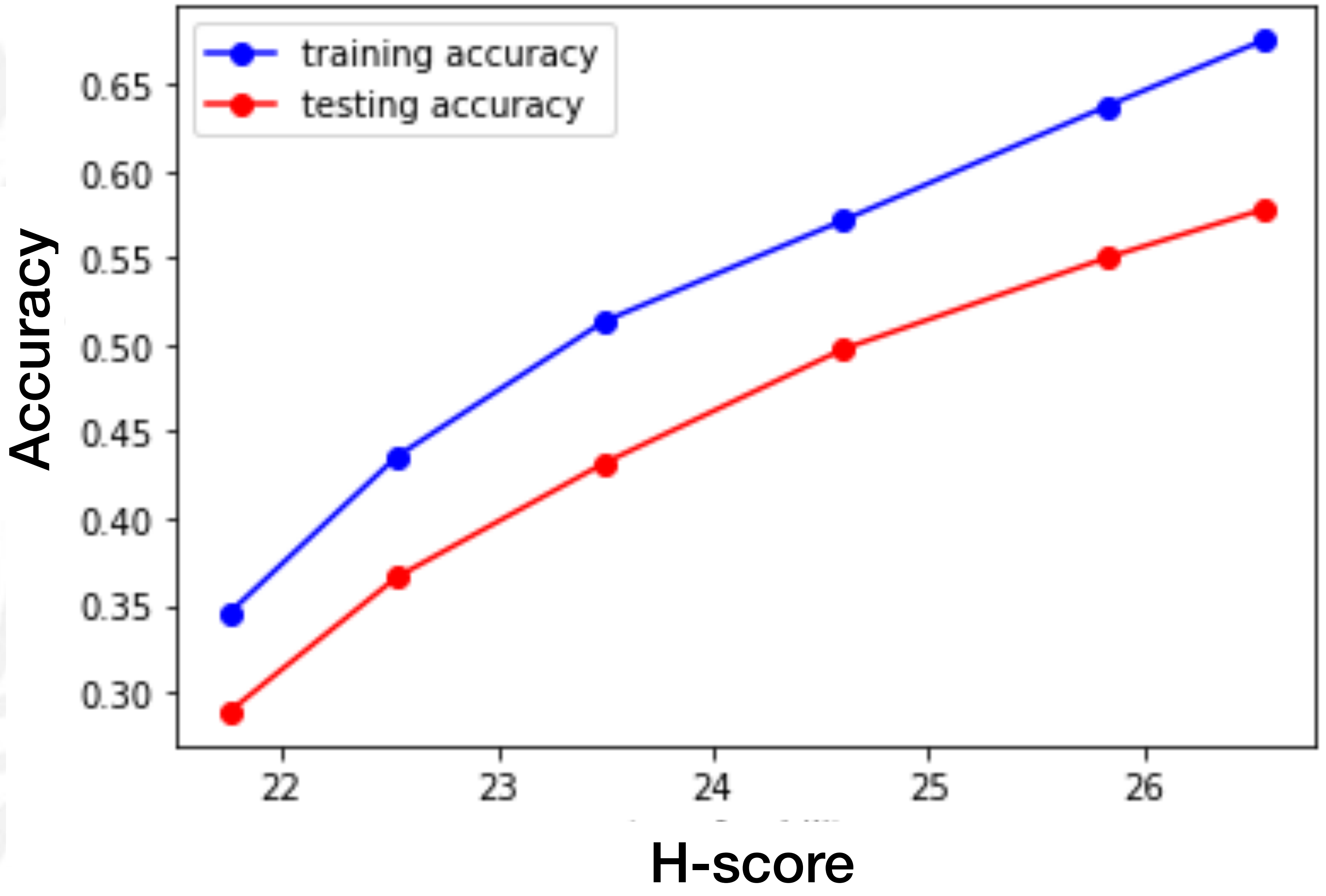


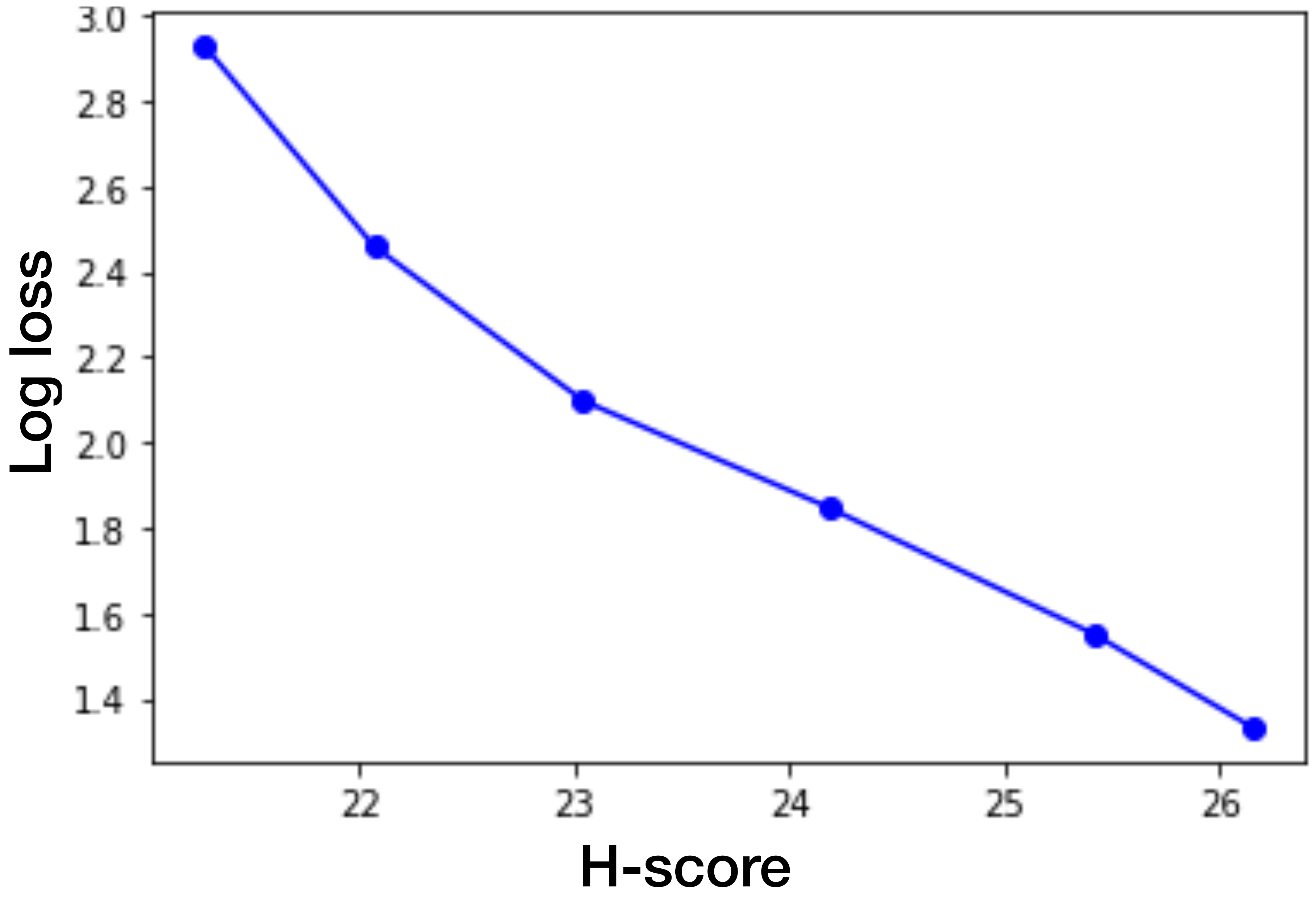


Results: Image Classification Feature Selection

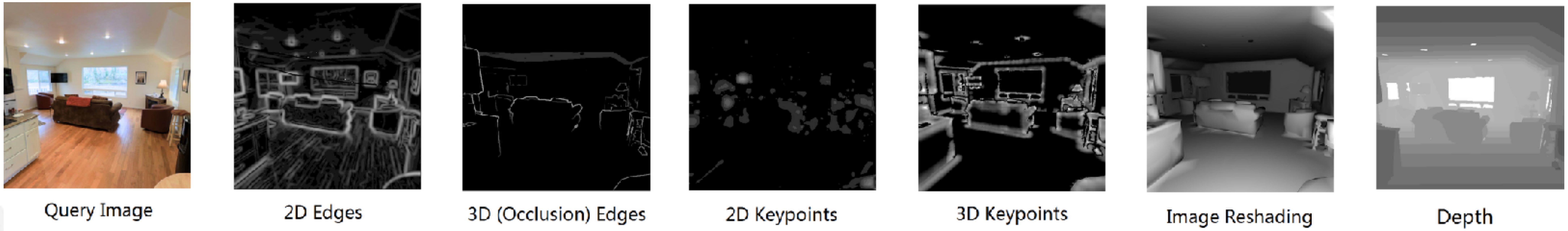
- Source task: ImageNet 1000 classification (ResNet50 features from 6 layers 4a-5f)
- Target task: Cifar 100-class classification on 20,000 images



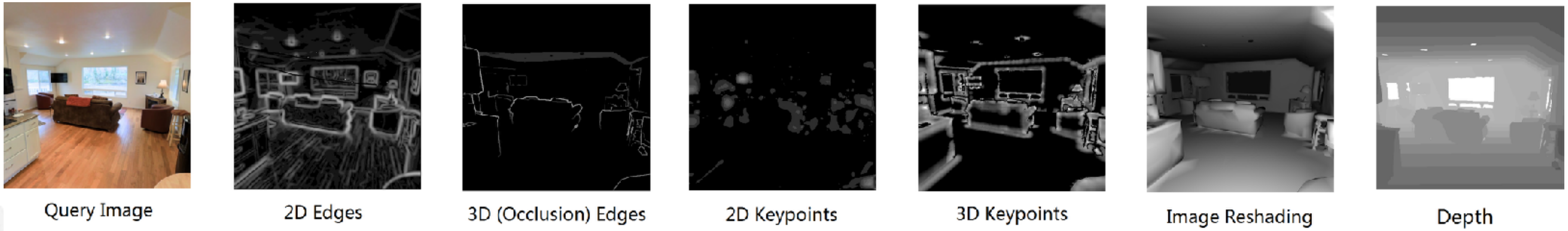




Results: Source Task Selection for 3D Scene Understanding

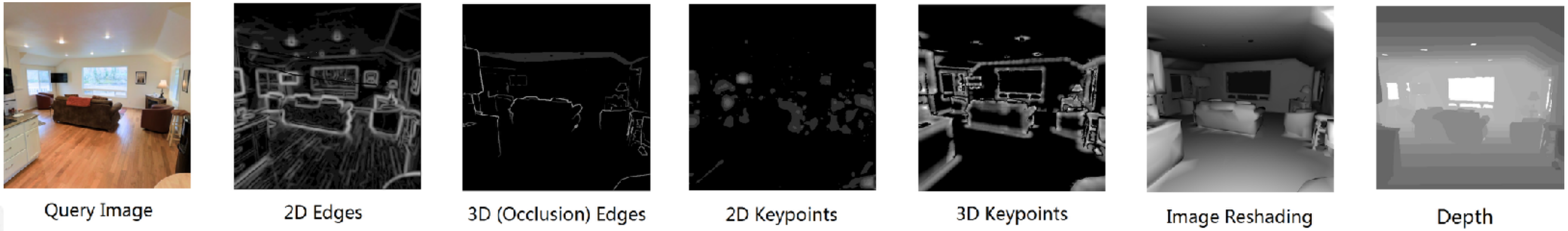


Results: Source Task Selection for 3D Scene Understanding



- 8 image-based tasks from Taskonomy dataset (Zamir et al. 2018)
  - 2 classification tasks: object-class, scene-class
  - 6 2D/3D image-to-image tasks: average H-score over all superpixels

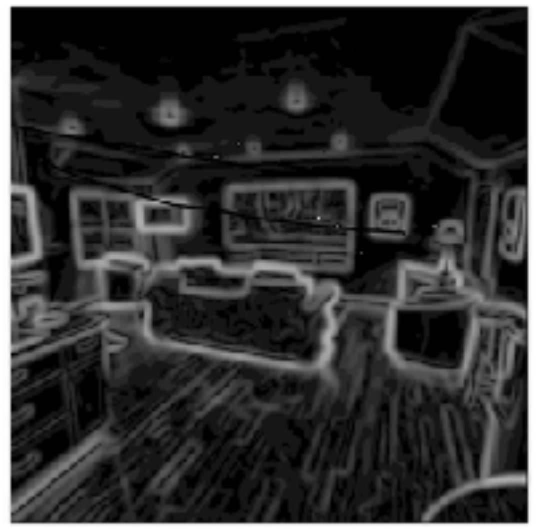
Results: Source Task Selection for 3D Scene Understanding



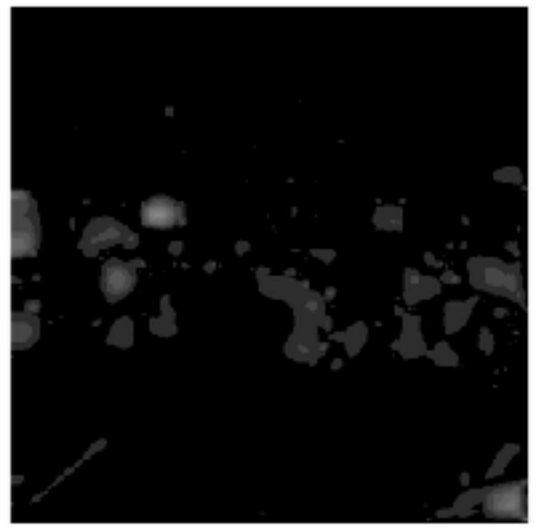
- 8 image-based tasks from Taskonomy dataset (Zamir et al. 2018)
  - 2 classification tasks: object-class, scene-class
  - 6 2D/3D image-to-image tasks: average H-score over all superpixels
- Source models: pre-trained task-specific models (4,000,000 training samples);

Results: Source Task Selection for 3D Scene Understanding

Query Image


2D Edges

3D (Occlusion) Edges


2D Keypoints

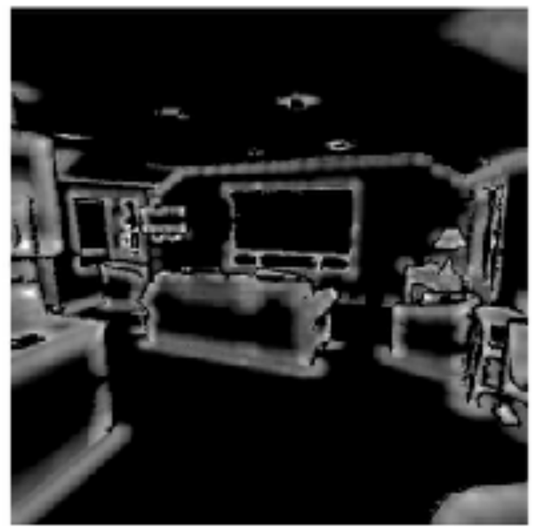
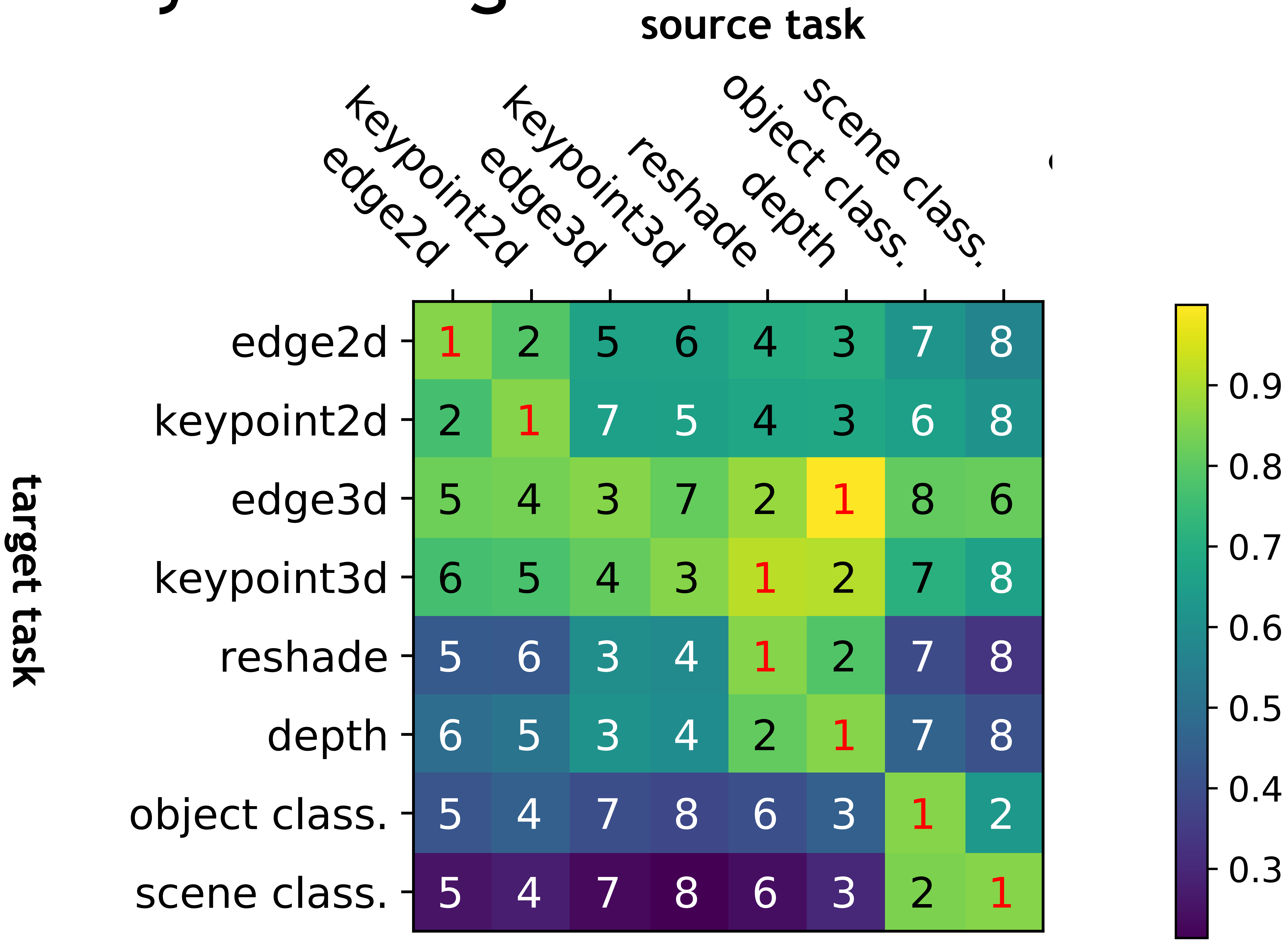

3D Keypoints

Image Reshading

Depth

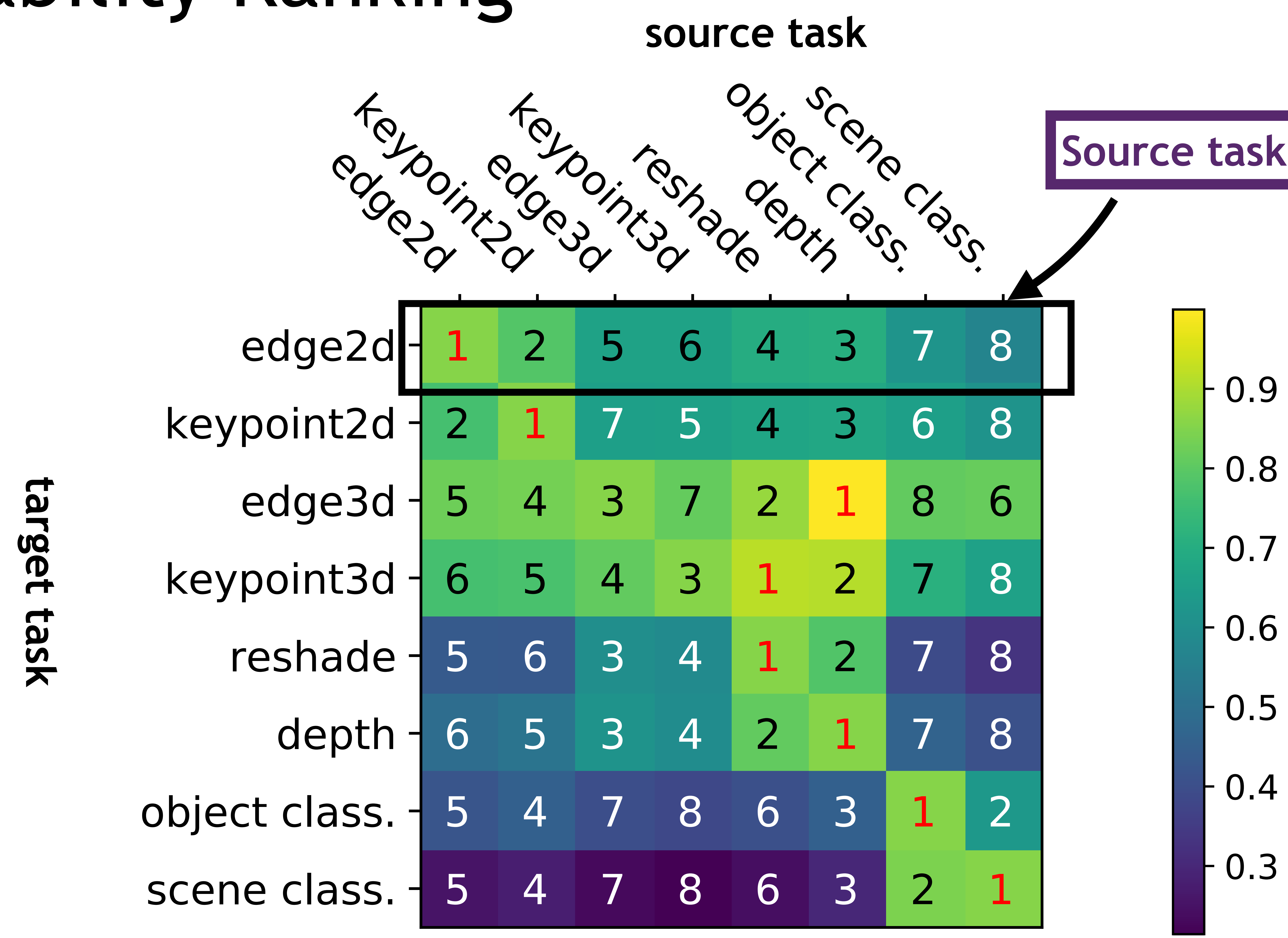
- 8 image-based tasks from Taskonomy dataset (Zamir et al. 2018)
  - 2 classification tasks: object-class, scene-class
  - 6 2D/3D image-to-image tasks: average H-score over all superpixels
- Source models: pre-trained task-specific models (4,000,000 training samples);
- Target model: linear feature transfer using 20,000 images (64 x 64)

Transferability Ranking



	edge2d	keypoint2d	edge3d	keypoint3d	reshade	depth	object class.	scene class.
edge2d	1	2	5	6	4	3	7	8
keypoint2d	2	1	7	5	4	3	6	8
edge3d	5	4	3	7	2	1	8	6
keypoint3d	6	5	4	3	1	2	7	8
reshade	5	6	3	4	1	2	7	8
depth	6	5	3	4	2	1	7	8
object class.	5	4	7	8	6	3	1	2
scene class.	5	4	7	8	6	3	2	1

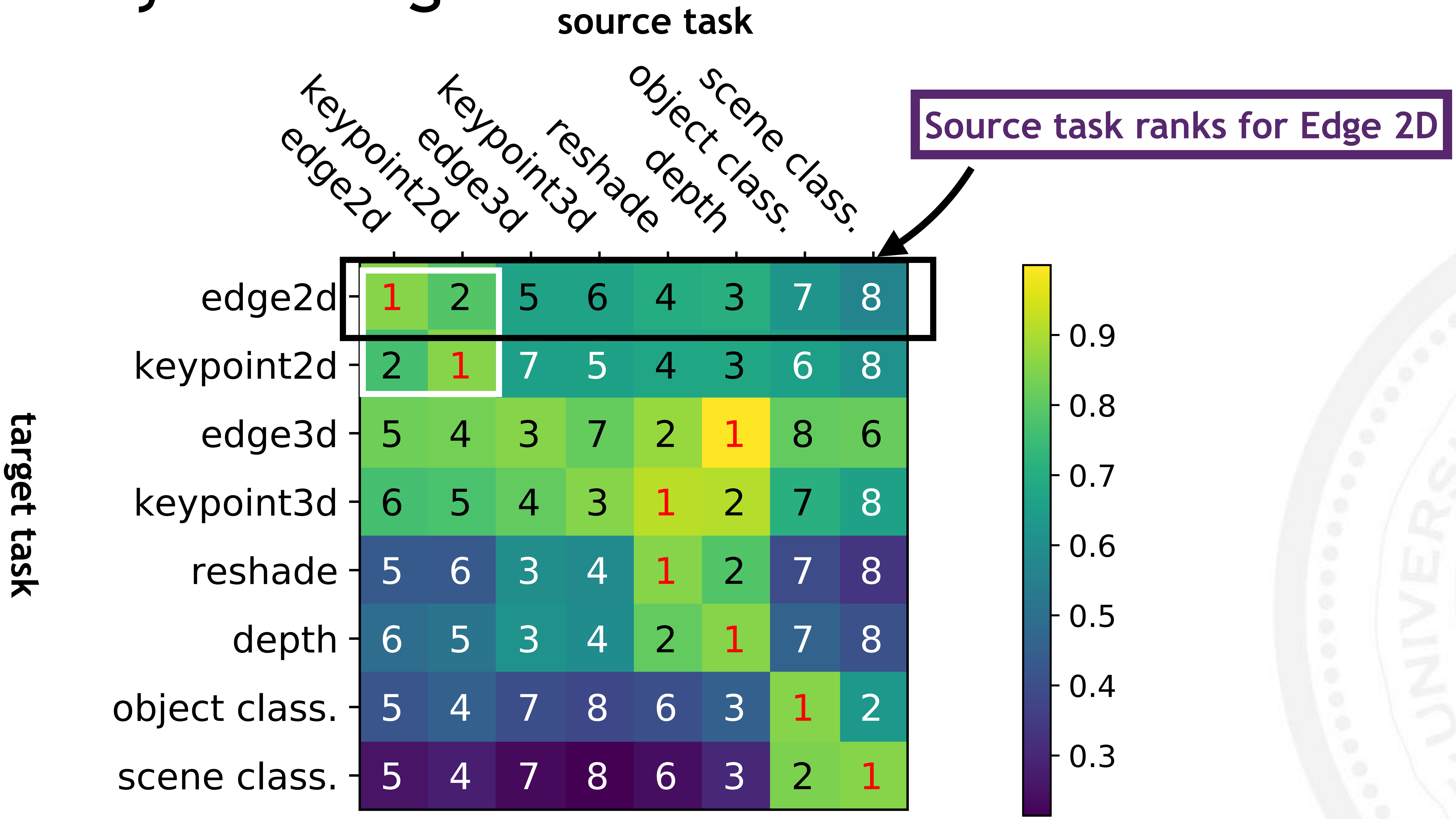
Transferability Ranking



	edge2d	keypoint2d	edge3d	keypoint3d	reshade	depth	object class.	scene class.
edge2d	1	2	5	6	4	3	7	8
keypoint2d	2	1	7	5	4	3	6	8
edge3d	5	4	3	7	2	1	8	6
keypoint3d	6	5	4	3	1	2	7	8
reshade	5	6	3	4	1	2	7	8
depth	6	5	3	4	2	1	7	8
object class.	5	4	7	8	6	3	1	2
scene class.	5	4	7	8	6	3	2	1

Source task ranks for Edge 2D

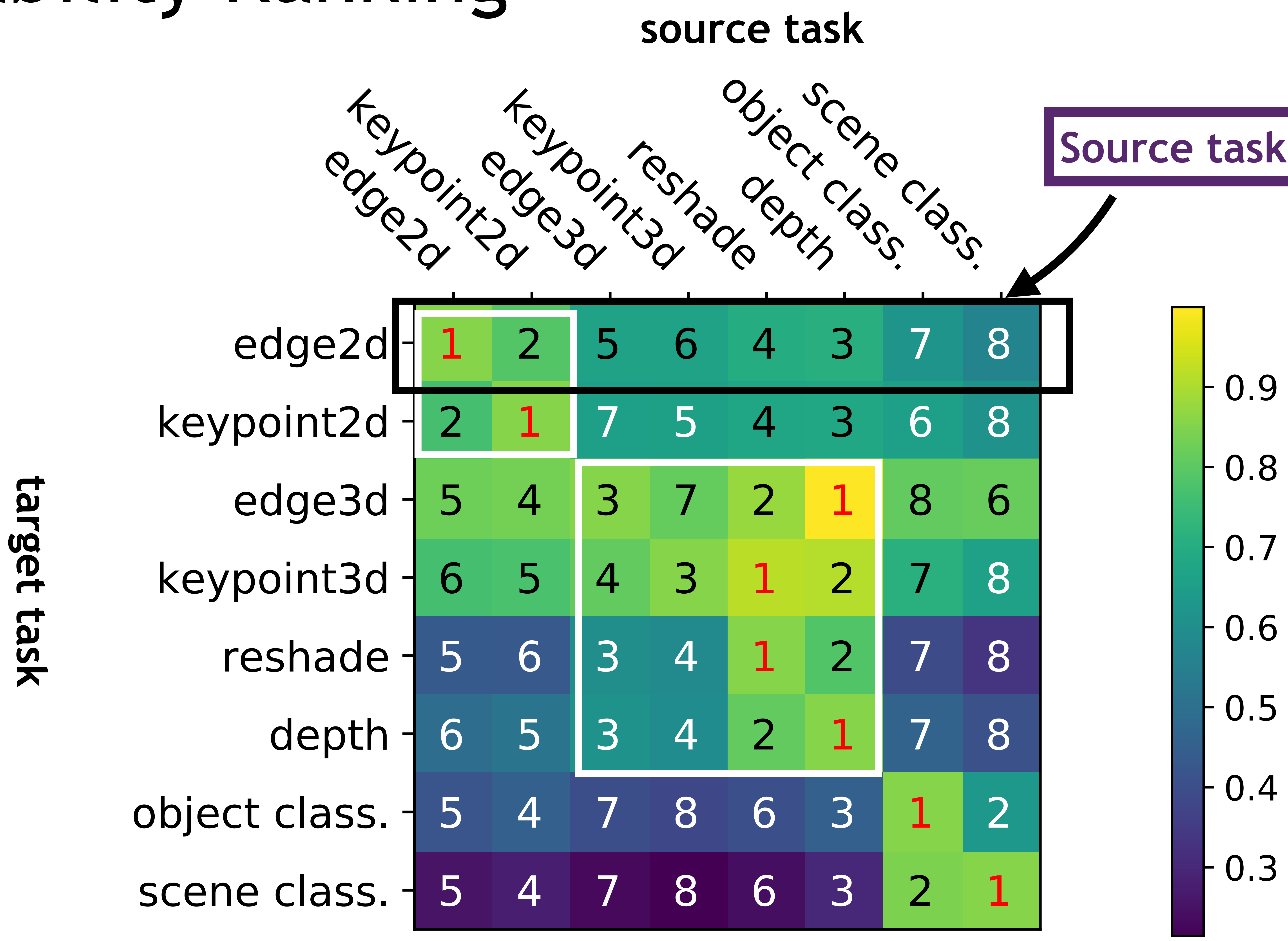
Transferability Ranking



Source task ranks for Edge 2D

	edge2d	keypoint2d	edge3d	keypoint3d	reshade	depth	object class.	scene class.
edge2d	1	2	5	6	4	3	7	8
keypoint2d	2	1	7	5	4	3	6	8
edge3d	5	4	3	7	2	1	8	6
keypoint3d	6	5	4	3	1	2	7	8
reshade	5	6	3	4	1	2	7	8
depth	6	5	3	4	2	1	7	8
object class.	5	4	7	8	6	3	1	2
scene class.	5	4	7	8	6	3	2	1

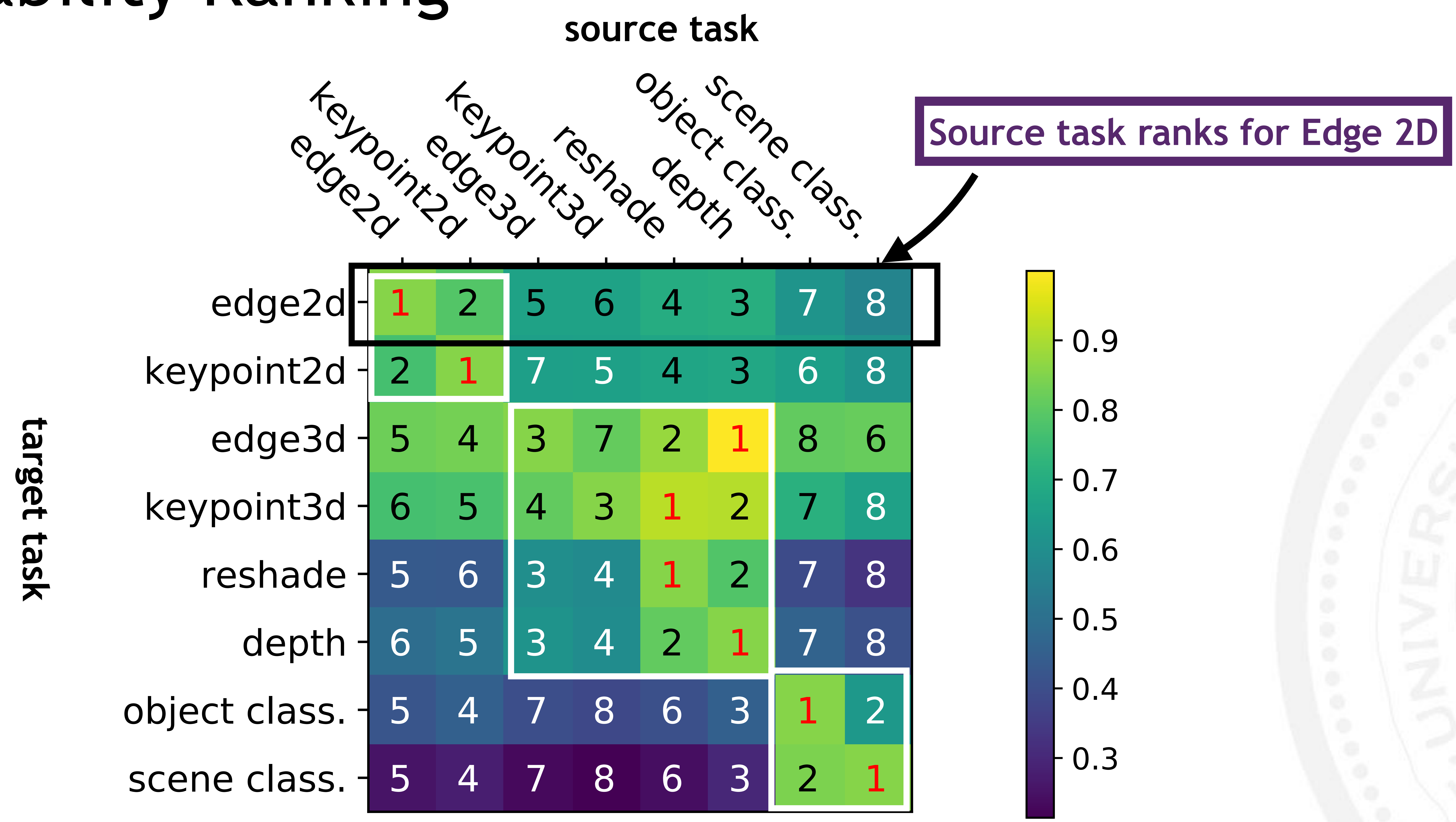
Transferability Ranking



target task \ source task	edge2d	keypoint2d	edge3d	keypoint3d	reshade	depth	object class.	scene class.
edge2d	1	2	5	6	4	3	7	8
keypoint2d	2	1	7	5	4	3	6	8
edge3d	5	4	3	7	2	1	8	6
keypoint3d	6	5	4	3	1	2	7	8
reshade	5	6	3	4	1	2	7	8
depth	6	5	3	4	2	1	7	8
object class.	5	4	7	8	6	3	1	2
scene class.	5	4	7	8	6	3	2	1

Source task ranks for Edge 2D

Transferability Ranking



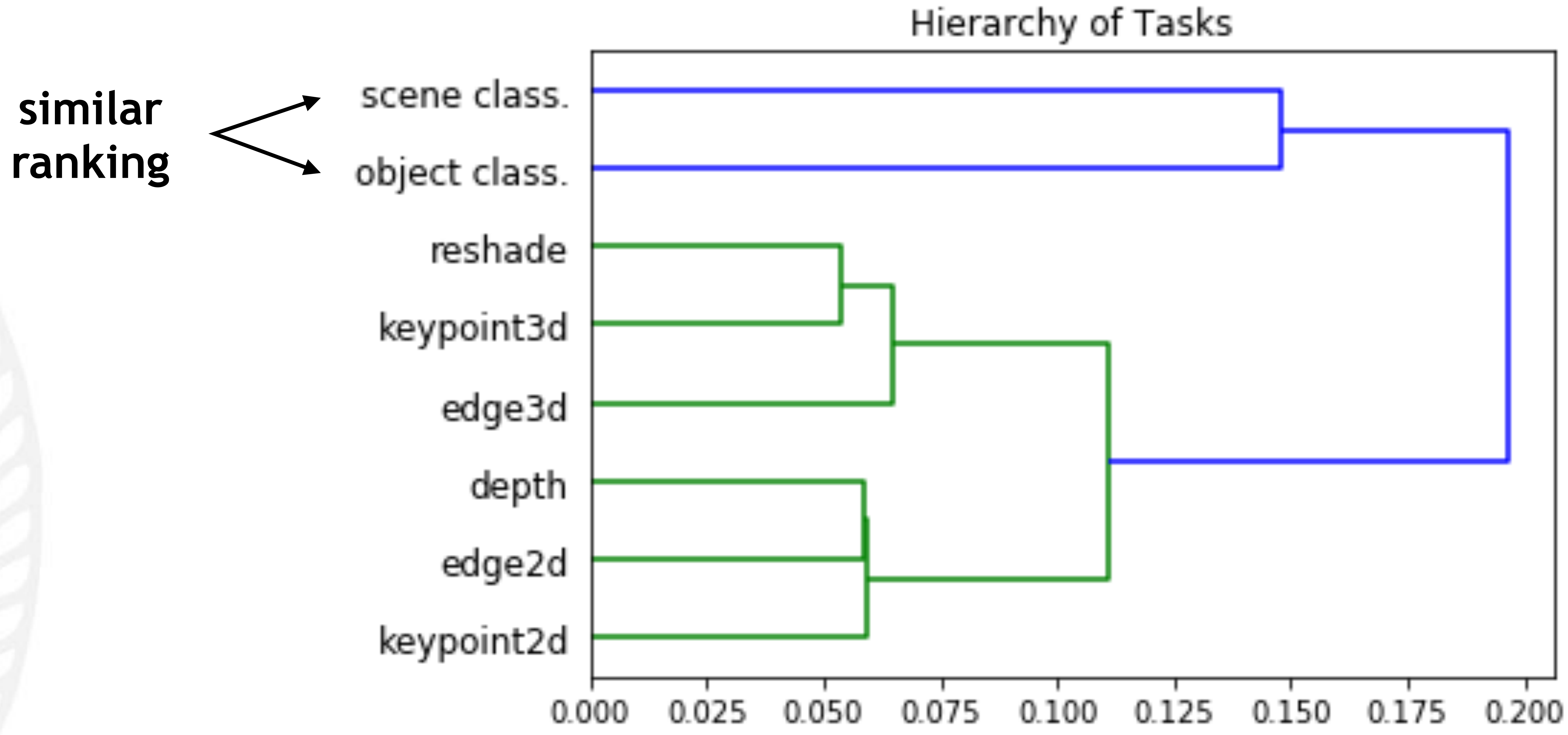
Source task ranks for Edge 2D

	edge2d	keypoint2d	edge3d	keypoint3d	reshade	depth	object class.	scene class.
edge2d	1	2	5	6	4	3	7	8
keypoint2d	2	1	7	5	4	3	6	8
edge3d	5	4	3	7	2	1	8	6
keypoint3d	6	5	4	3	1	2	7	8
reshade	5	6	3	4	1	2	7	8
depth	6	5	3	4	2	1	7	8
object class.	5	4	7	8	6	3	1	2
scene class.	5	4	7	8	6	3	2	1

Task Relationships

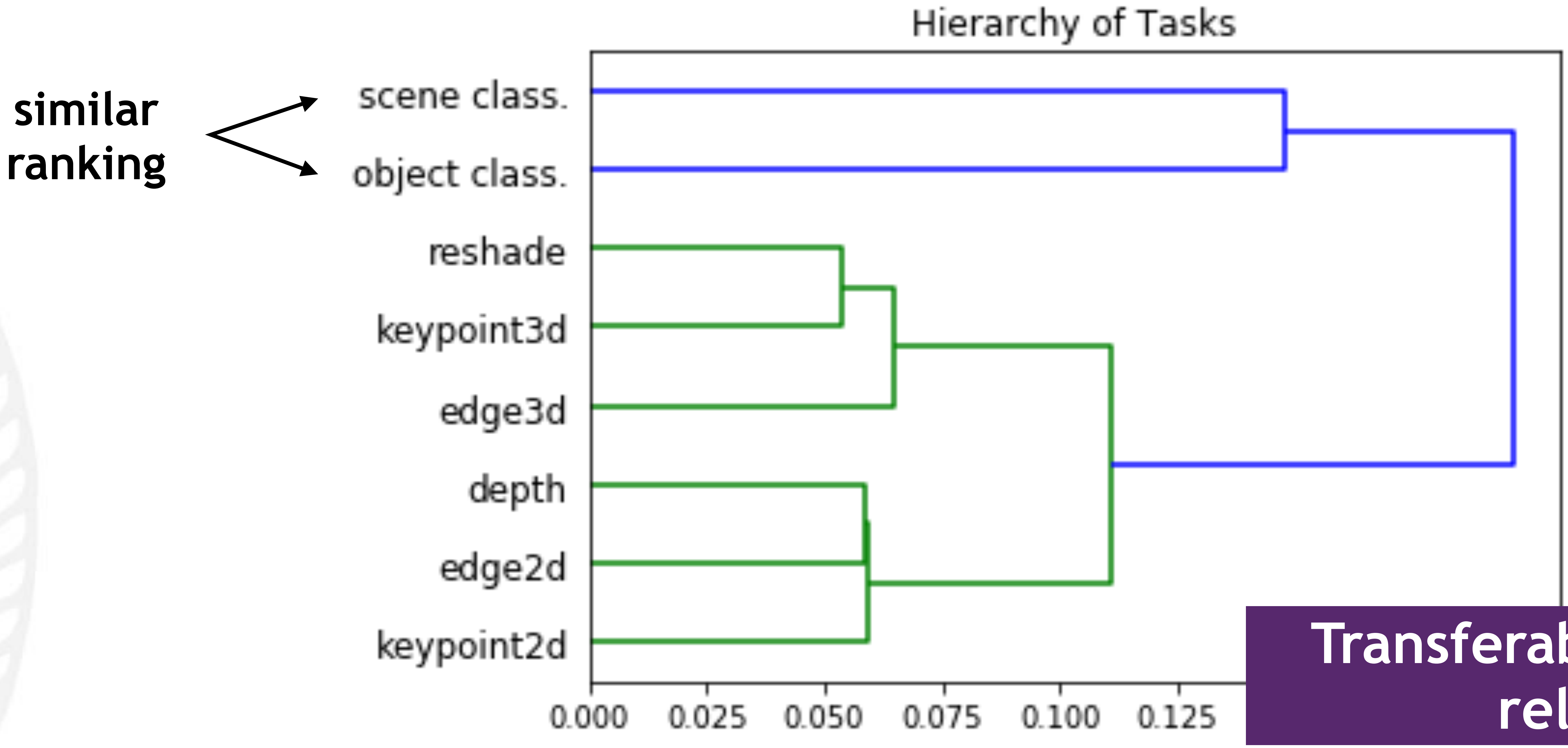
Cluster the source task transferability scores for each target task.

-



Task Relationships

Cluster the source task transferability scores for each target task.

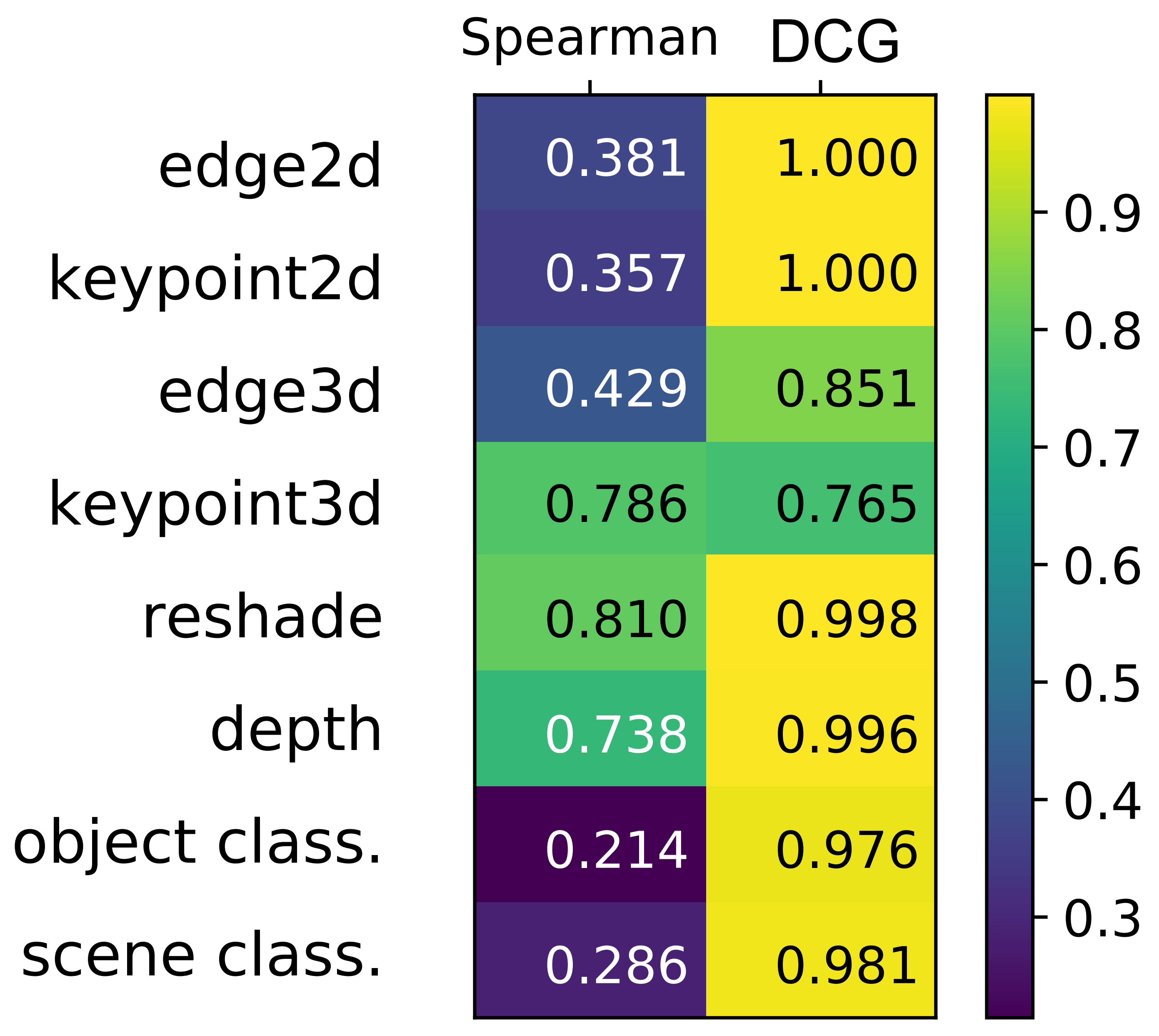


Transferability reveals task relationships

Comparison with Task Affinity

Reference metric: task affinity, an empirical transferability score (Amir et al. 2018)

- **Ranking results** agrees mostly on the top three rankings for each task



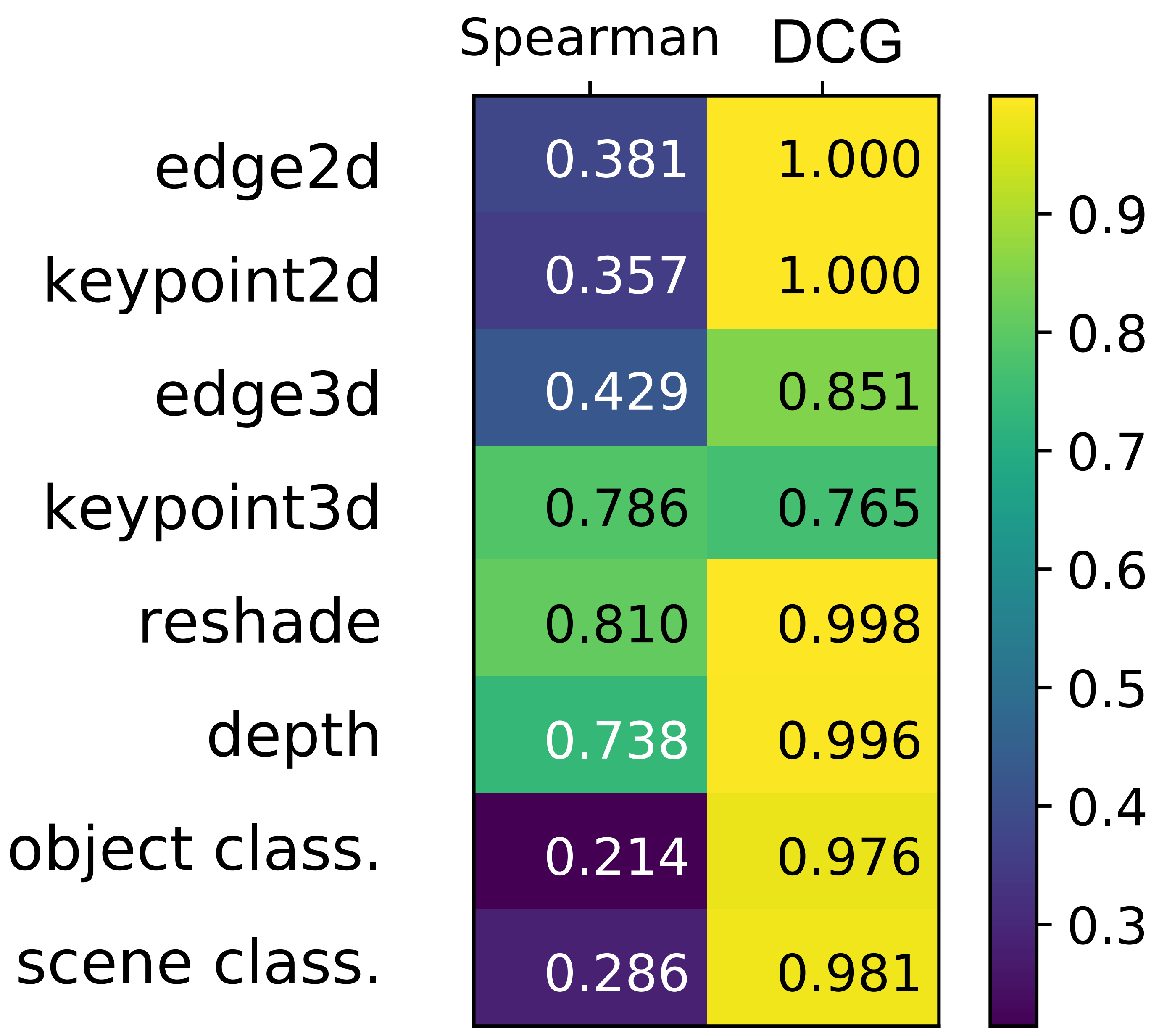
	Spearman	DCG
edge2d	0.381	1.000
keypoint2d	0.357	1.000
edge3d	0.429	0.851
keypoint3d	0.786	0.765
reshade	0.810	0.998
depth	0.738	0.996
object class.	0.214	0.976
scene class.	0.286	0.981

Rank Comparison

Comparison with Task Affinity

Reference metric: task affinity, an empirical transferability score (Amir et al. 2018)

- Ranking results agrees mostly on the top three rankings for each task



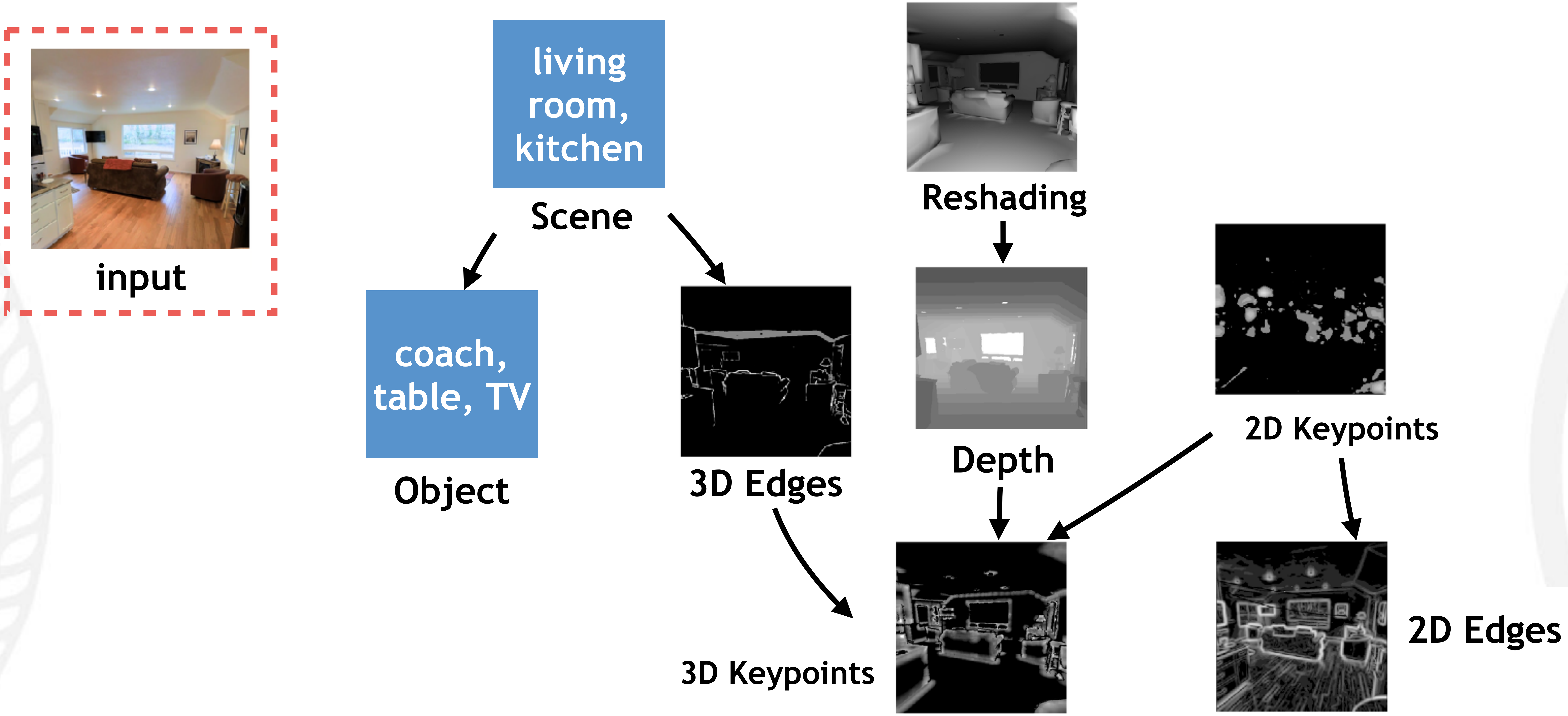
Rank Comparison

Advantage of our approach:

- Efficiency: five times more efficient than Affinity
- Clear operational meaning based on statistics & information theory

Application: Task Curriculum based on Transferability

- A minimum-spanning tree approach to design transfer curriculum

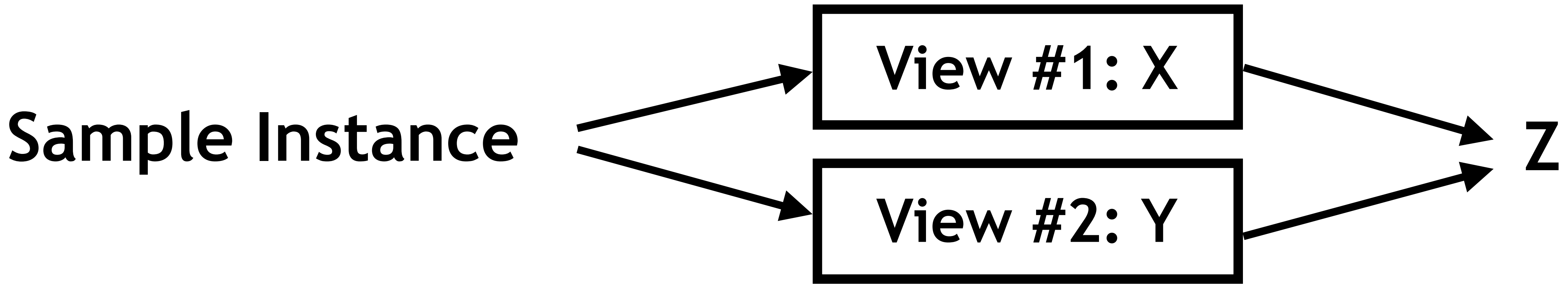


Outline

- Intro: Shared Representation & Maximal Correlation
- Estimating Task Transferability in Task Transfer Learning
- Multi-view learning
- Conclusion

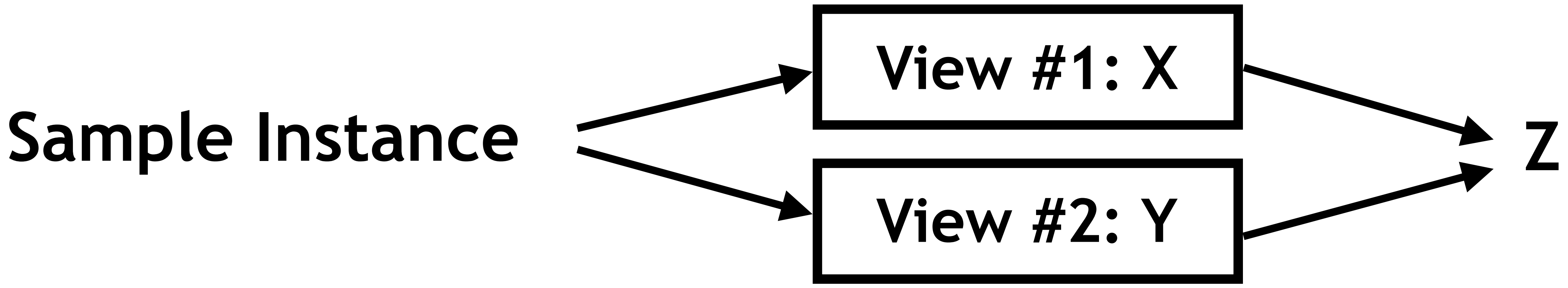
Multi-View Learning

- Exploits shared knowledge among different data sources or different feature subsets



Multi-View Learning

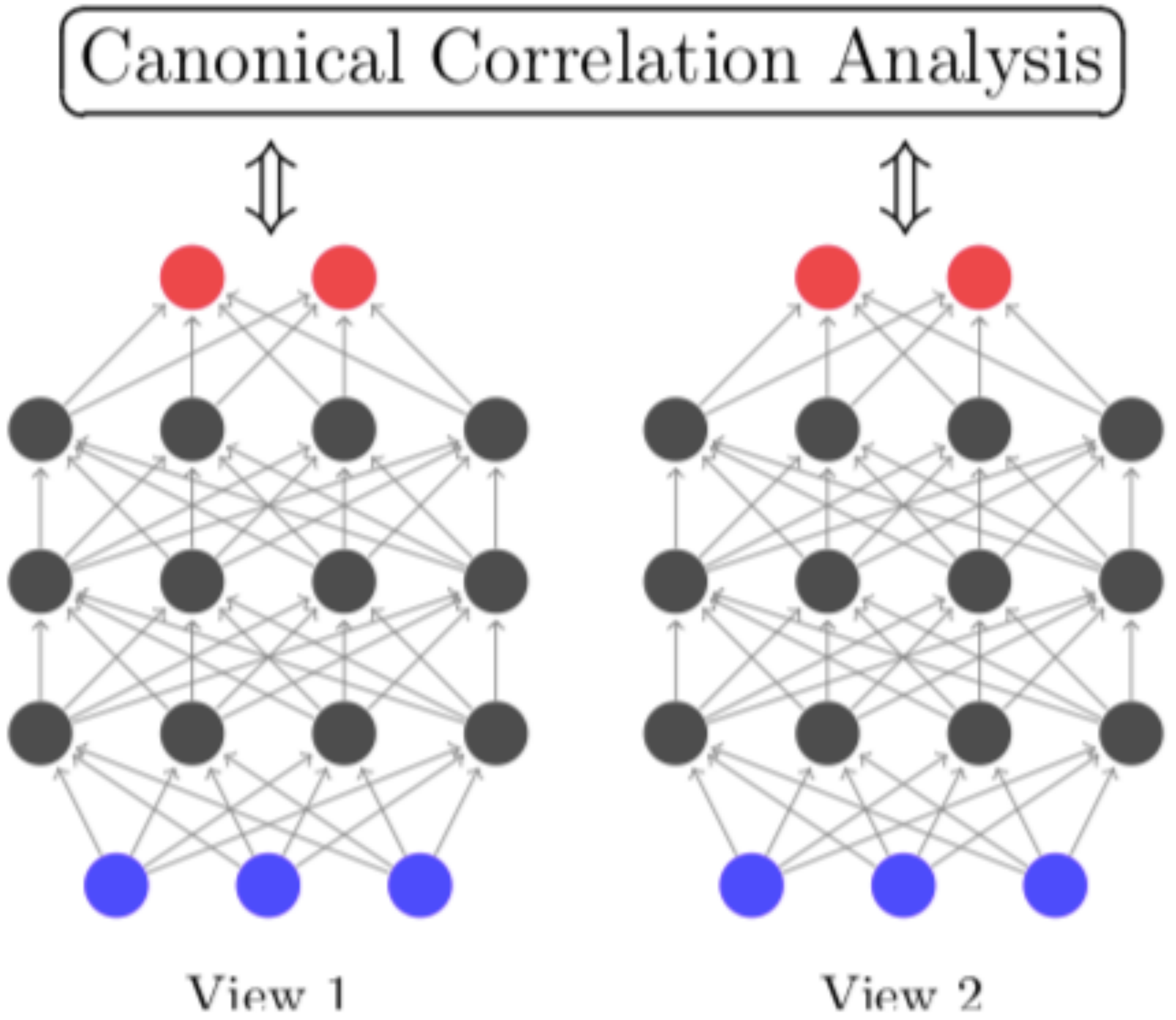
- Exploits shared knowledge among different data sources or different feature subsets



- Correlation-based approaches: a natural way to capture the shared information between views

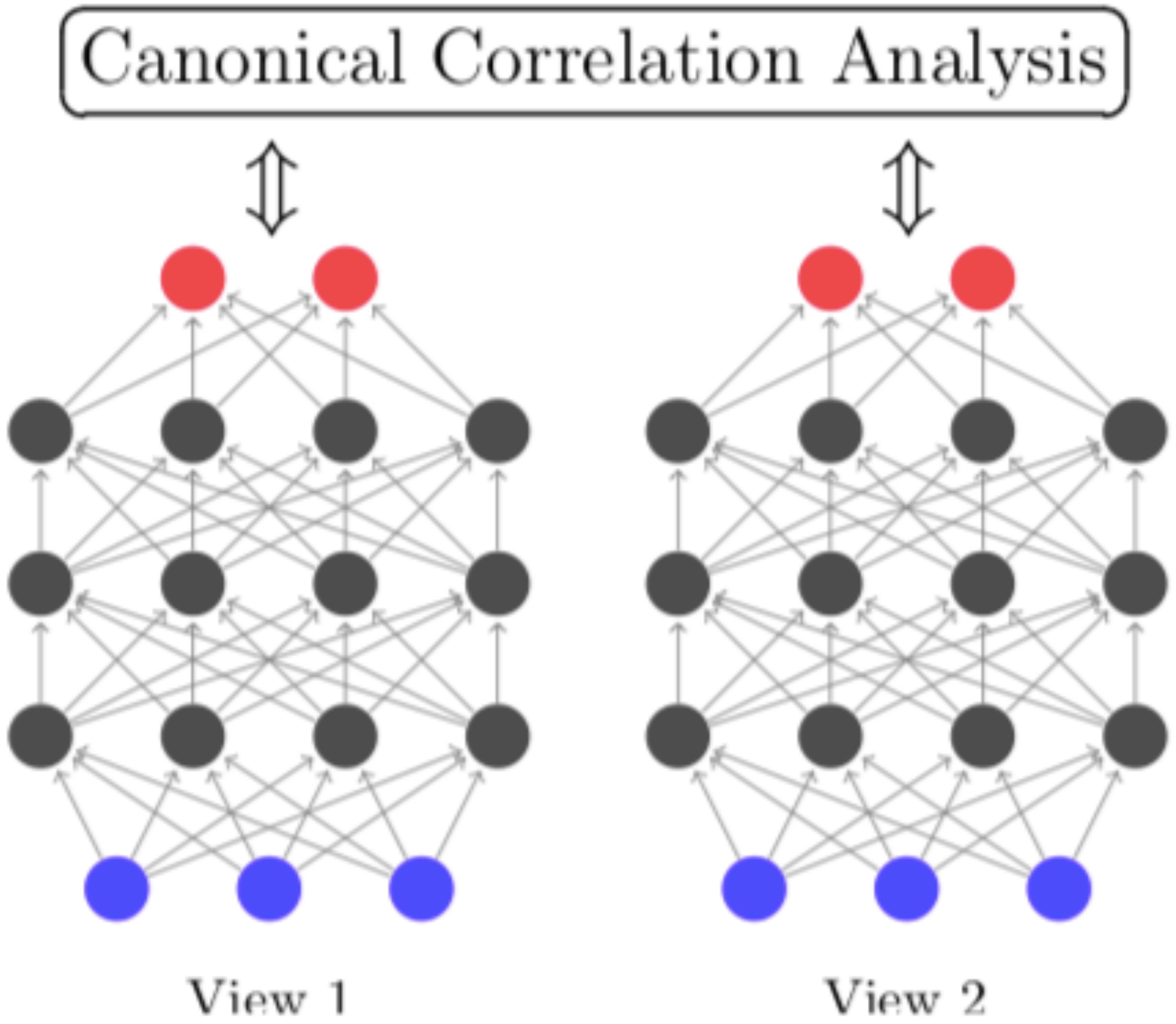
Correlation-Based Approaches

- CCA and Kernalized CCA: shallow modes
- Deep CCA (DCCA) [Andrew et. al. 2013]
- Deep CCA Auto Encoder (DCCAE) [Wang et. al 2016]



Correlation-Based Approaches

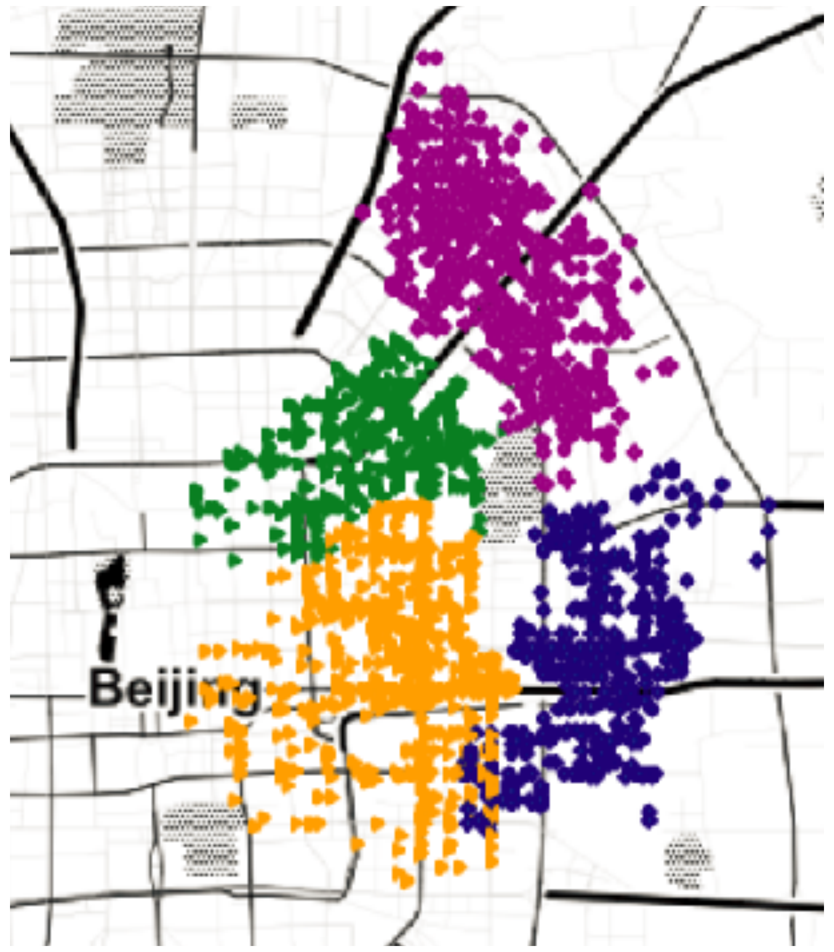
- CCA and Kernalized CCA: shallow modes
- Deep CCA (DCCA) [Andrew et. al. 2013]
- Deep CCA Auto Encoder (DCCAE) [Wang et. al 2016]



- Limitations:
  - Numerical issues (whitening based on matrix inverse)
  - Feature dimension is limited

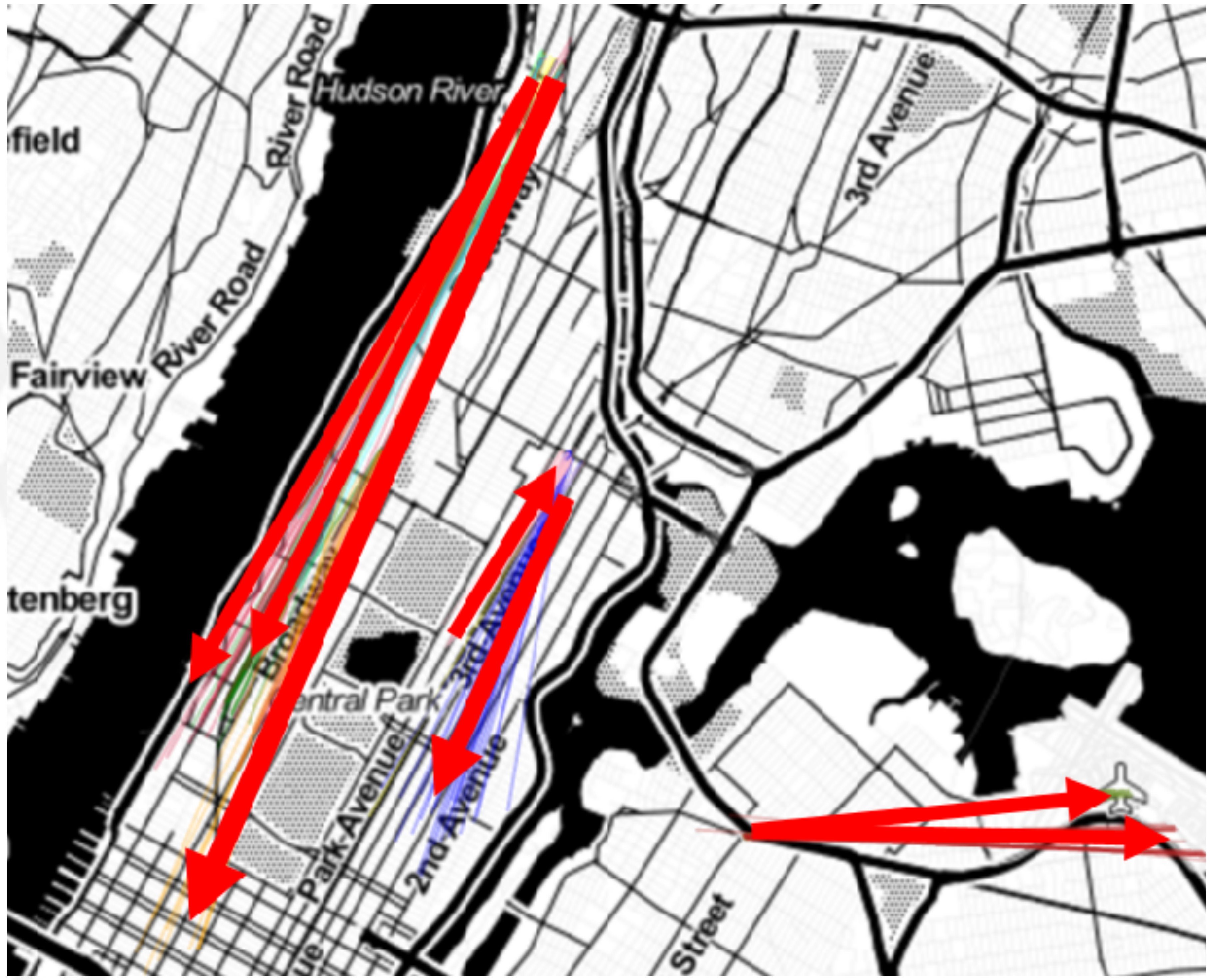
Multi-View Learning using Maximal HGR Correlation

- Unsupervised task:
  - multi-view mobility pattern extraction



- Supervised task:
  - mutli-modal emotion recognition

Mobility Pattern Mining



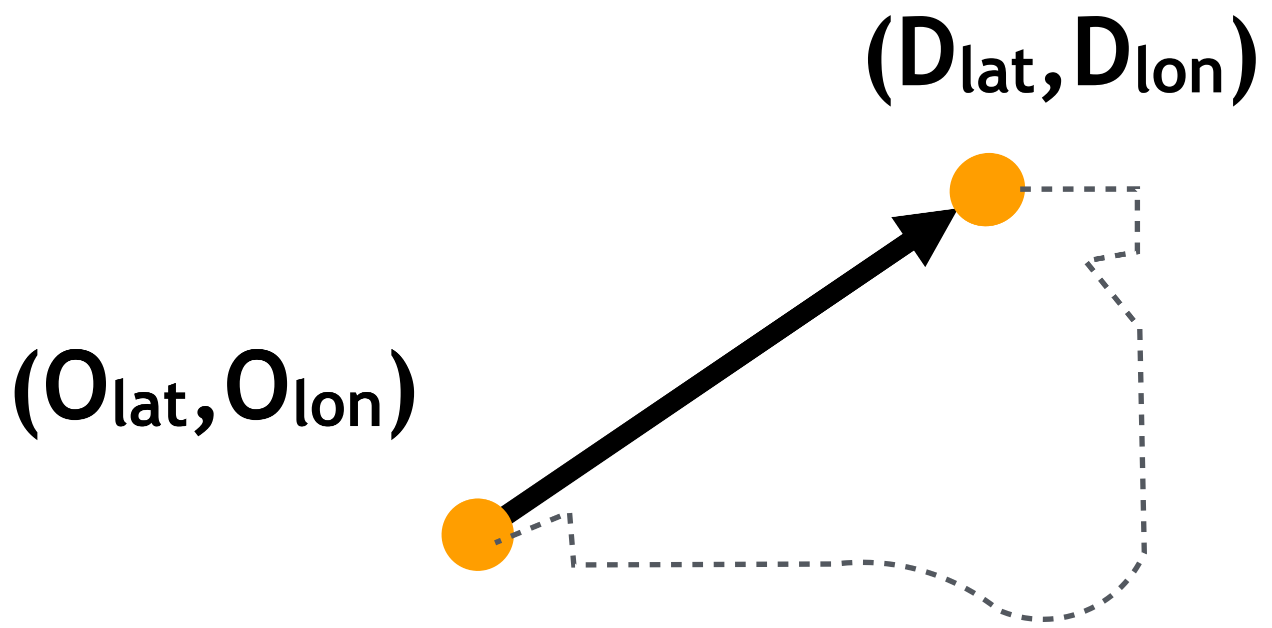
New York Taxi Trip Records, 17:00 – 18:00, 2015 May 11th – May 15th

- Mobility pattern: Common Repeated Travel Demand among a Population

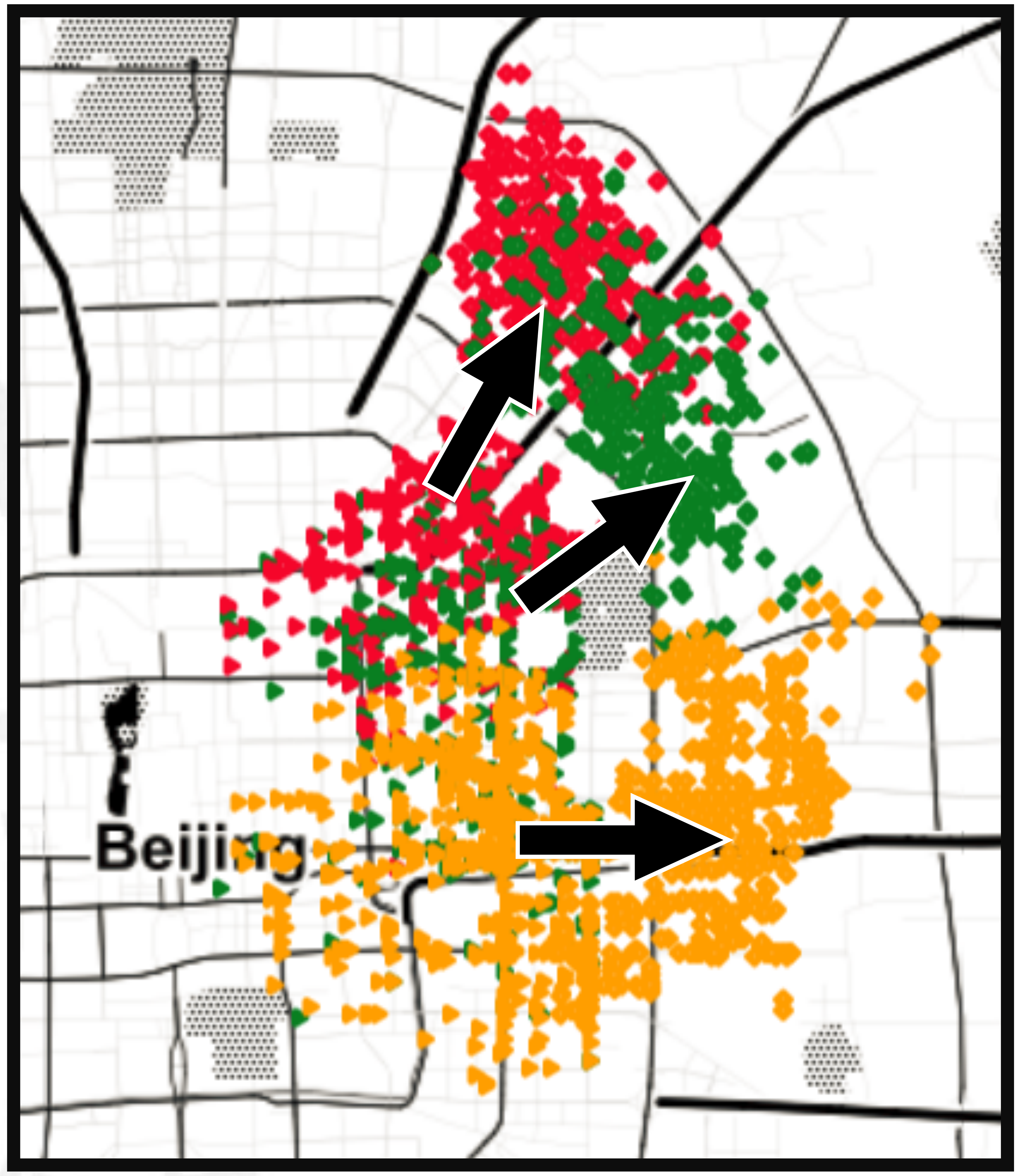
Public Transport

Location-Based Service

- Learn from trip (origin, destination) data

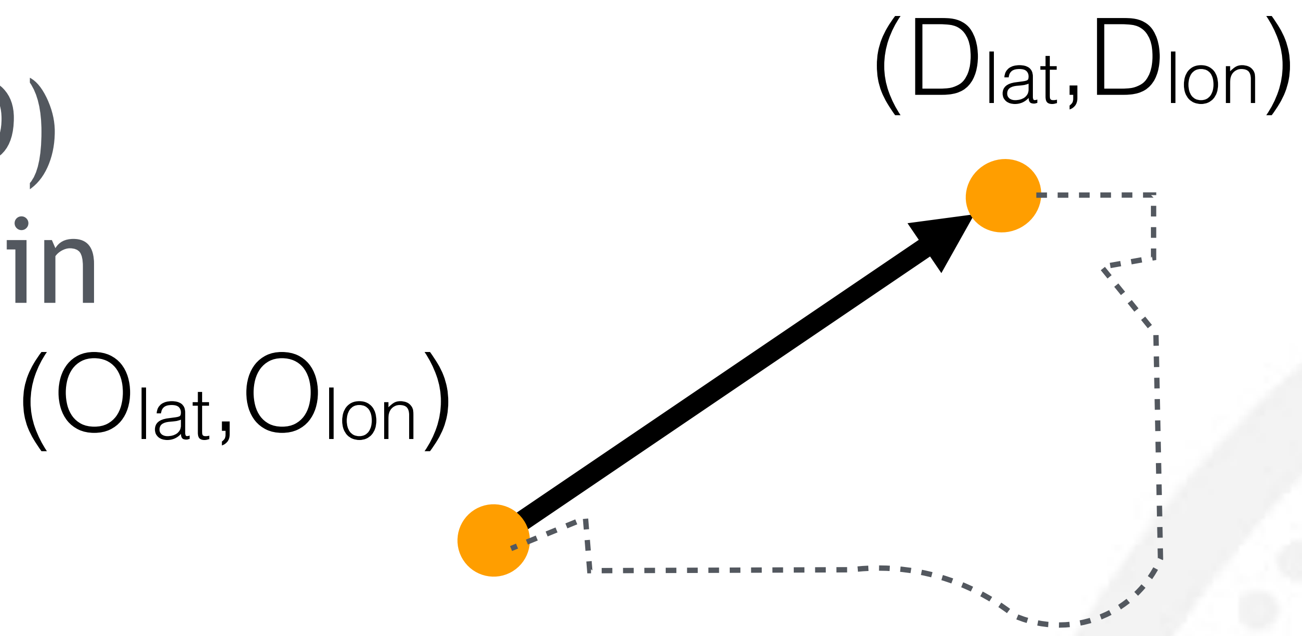


Single-View Approaches

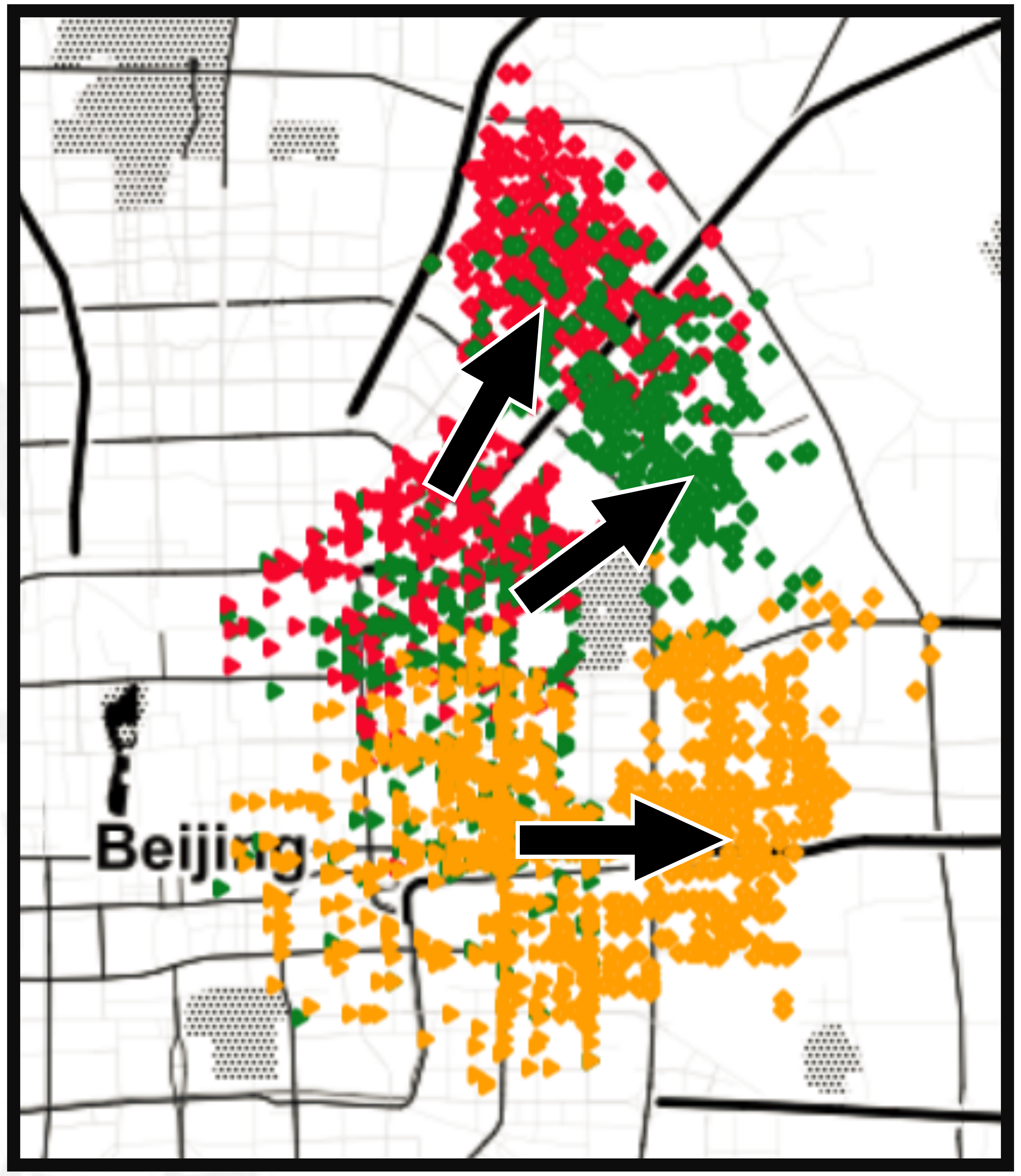


K-Means & DBSCAN

- Cluster Orgin-Destination (OD) trips (Olat, Olon, Dlat, Dlon) in 4D space
- Projecting to 2D space causes spatial overlap

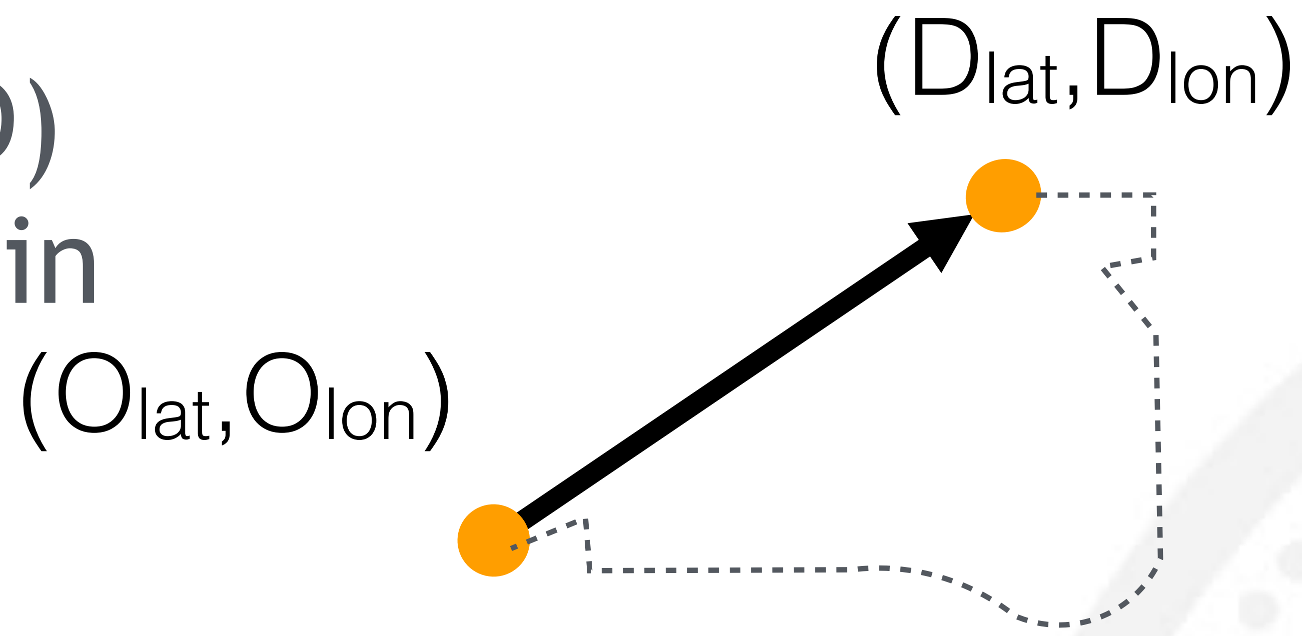


Single-View Approaches

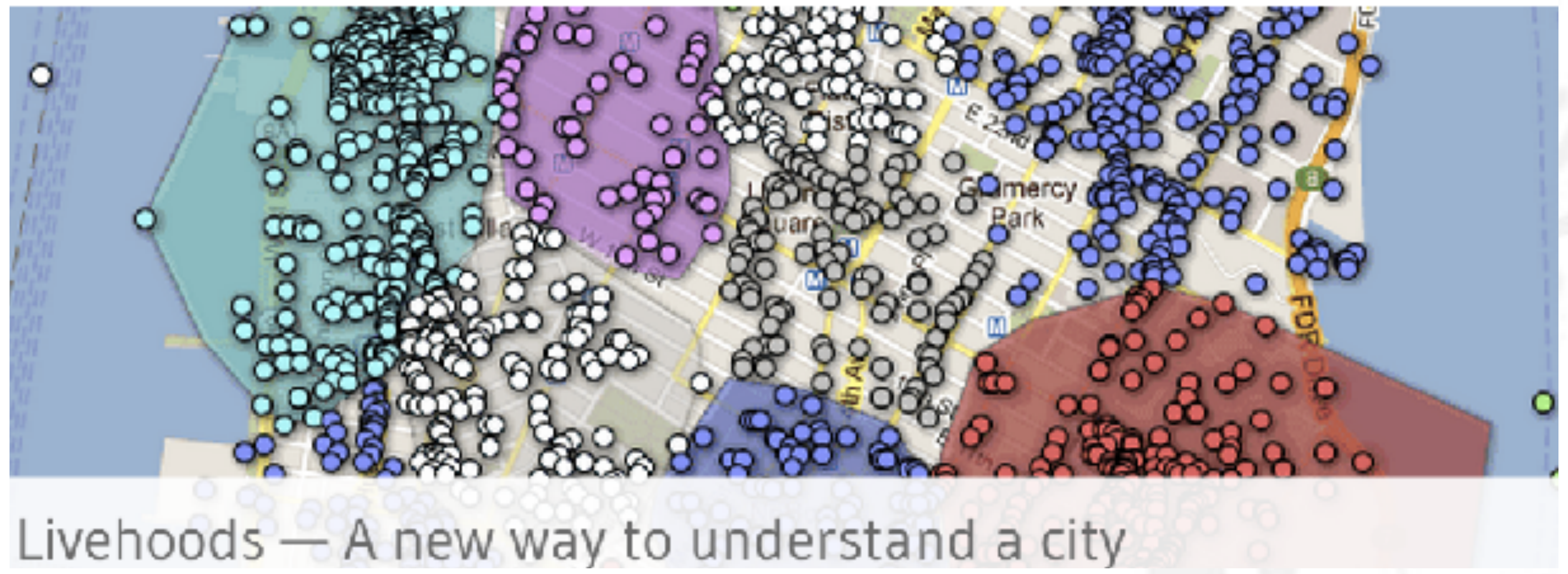


K-Means & DBSCAN

- Cluster Orgin-Destination (OD) trips (Olat, Olon, Dlat, Dlon) in 4D space



- Projecting to 2D space causes spatial overlap

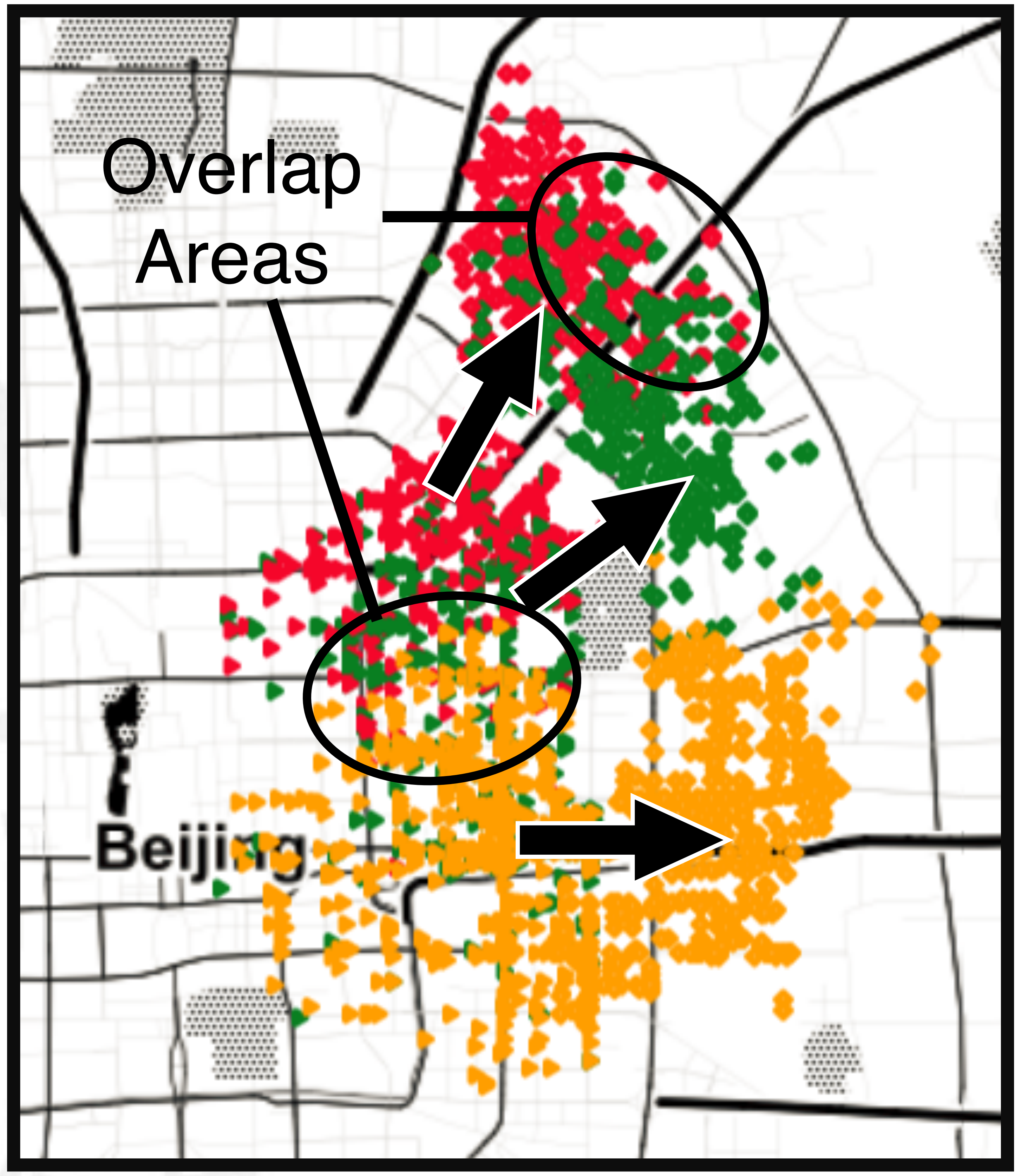


Livehoods — A new way to understand a city

City & Traffic Planning

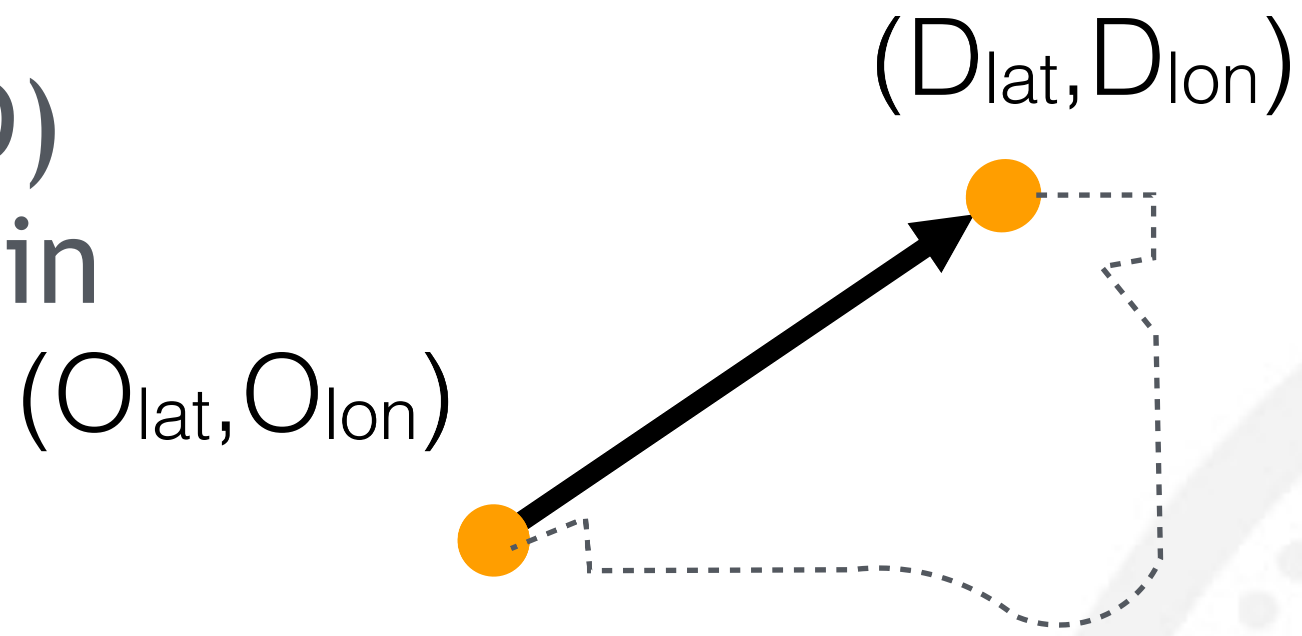
- Define traffic dynamic by regions
- Ambiguities for overlapped regions

Single-View Approaches

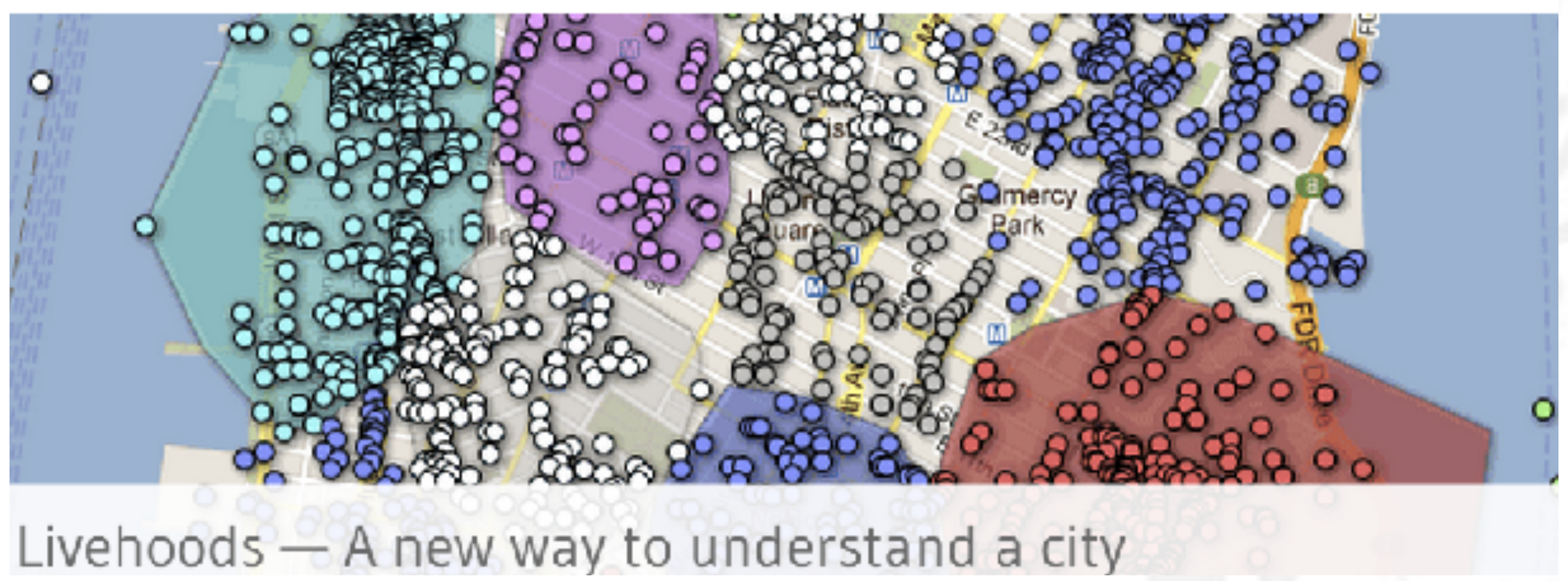


K-Means & DBSCAN

- Cluster Orgin-Destination (OD) trips (Olat, Olon, Dlat, Dlon) in 4D space



- Projecting to 2D space causes spatial overlap


Livehoods — A new way to understand a city

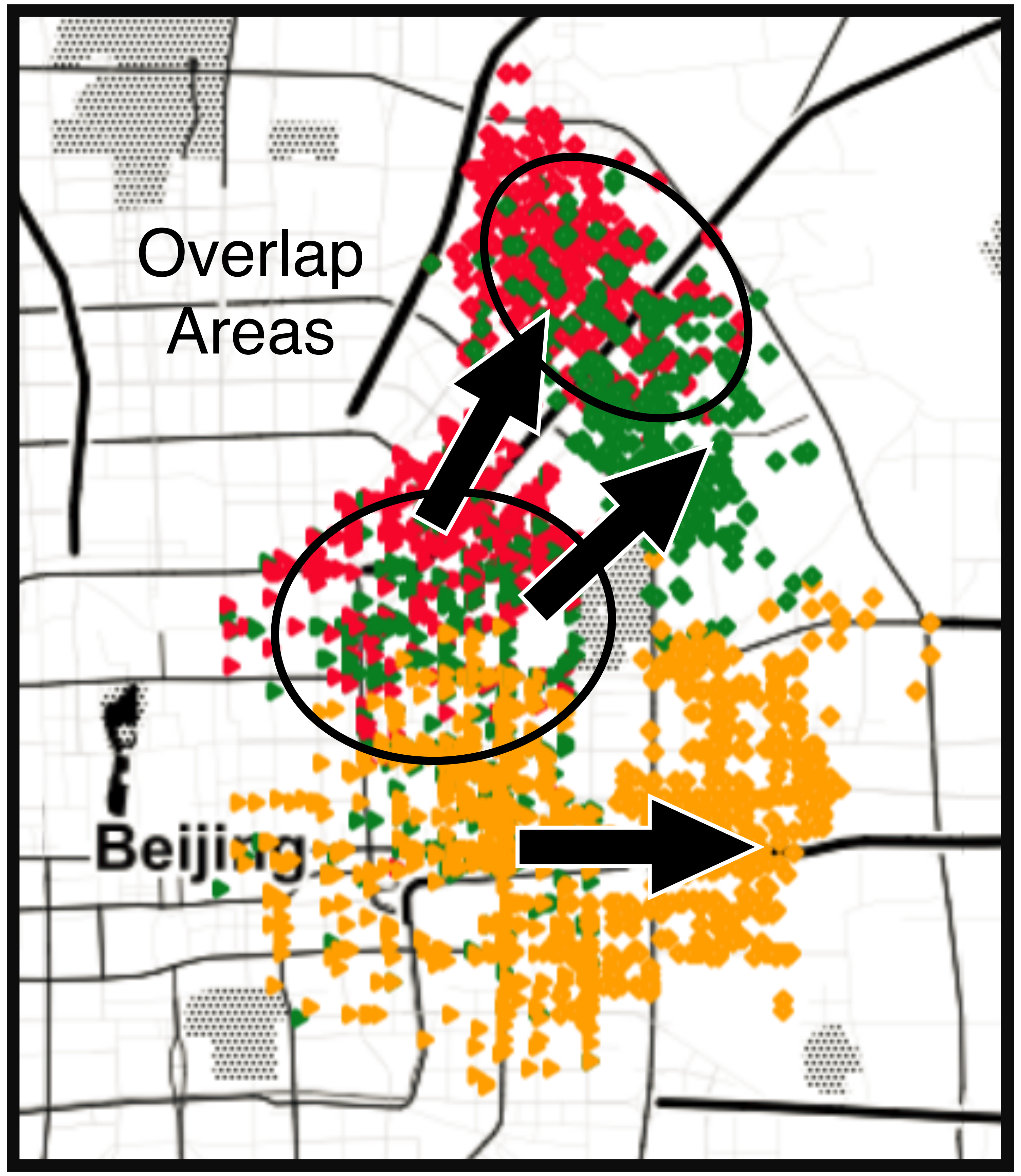
City & Traffic Planning

- Define traffic dynamic by regions
- Ambiguities for overlapped regions

Multi-view learning of mobility features

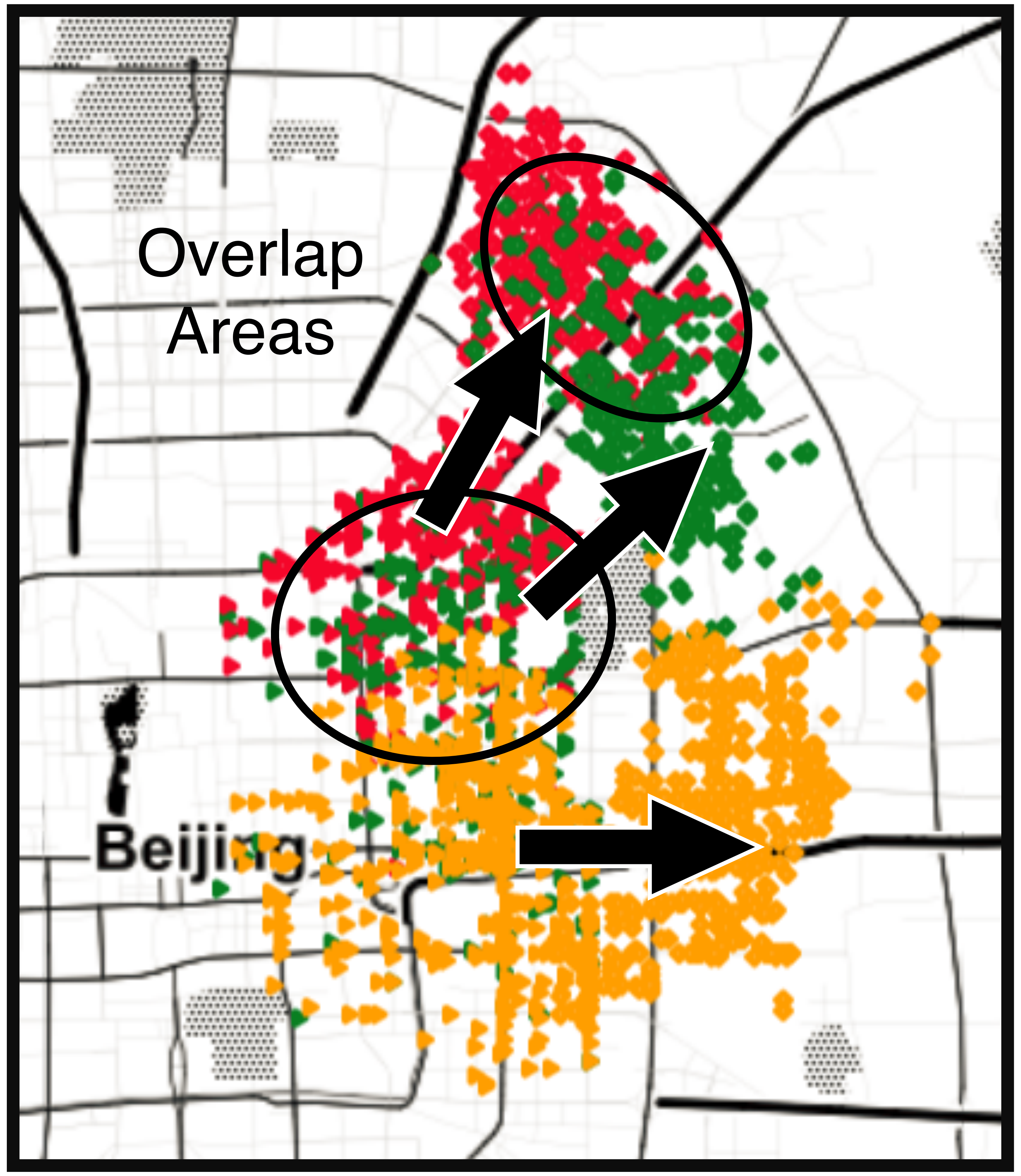
Traditional Approach

[Lian et. al. 2019]

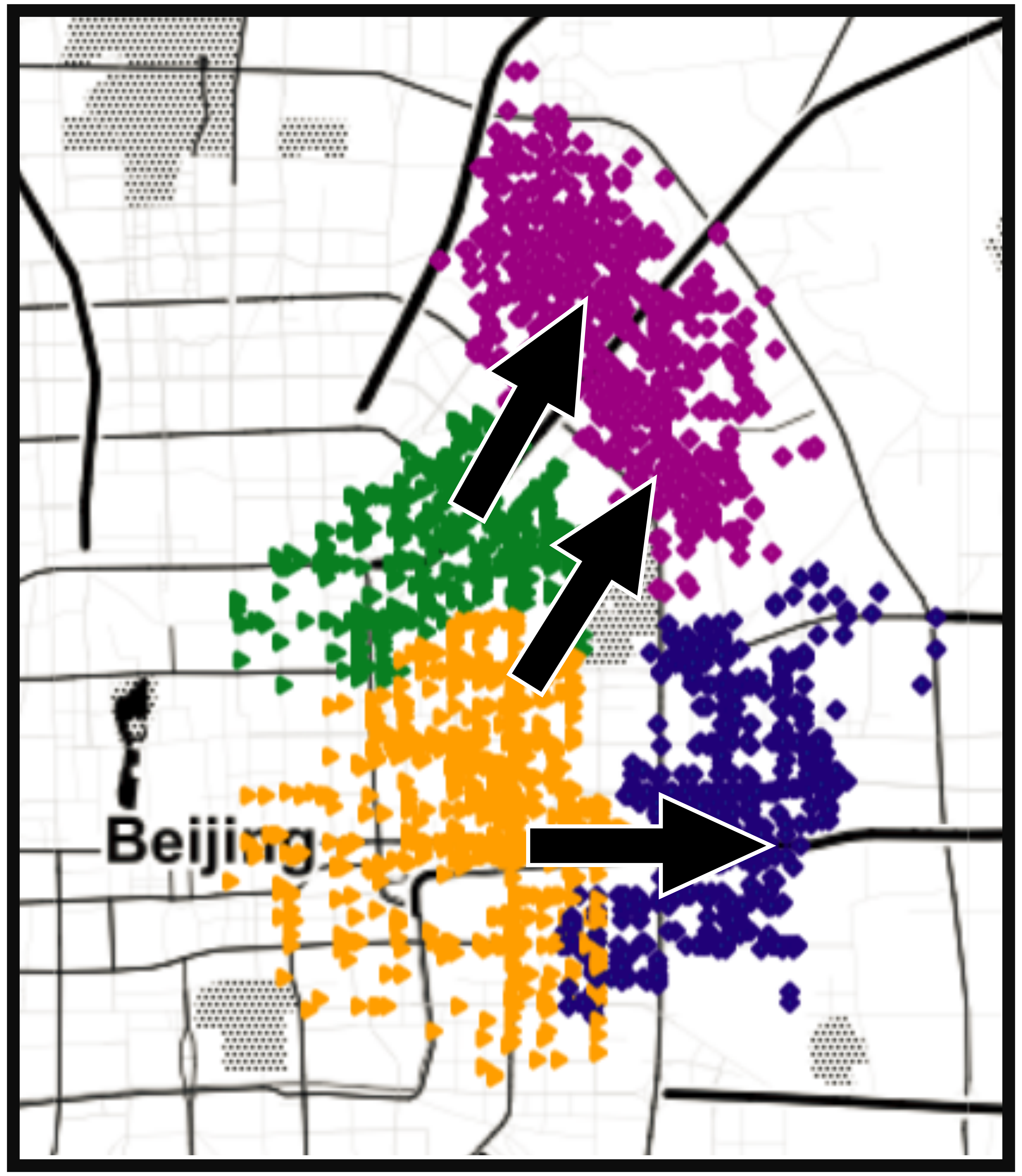


Multi-view learning of mobility features

Traditional Approach



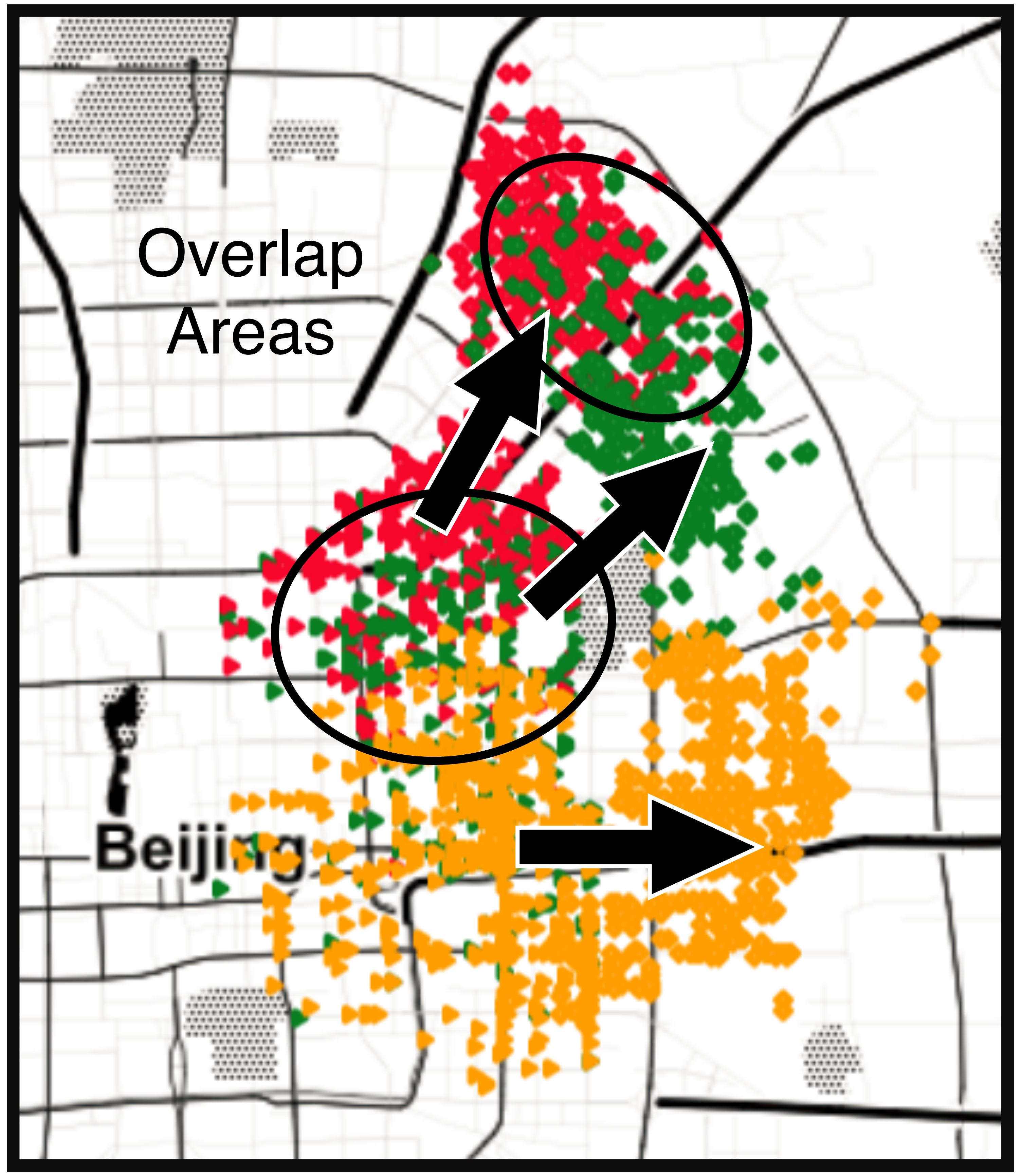
Our Approach **[Lian et. al. 2019]**



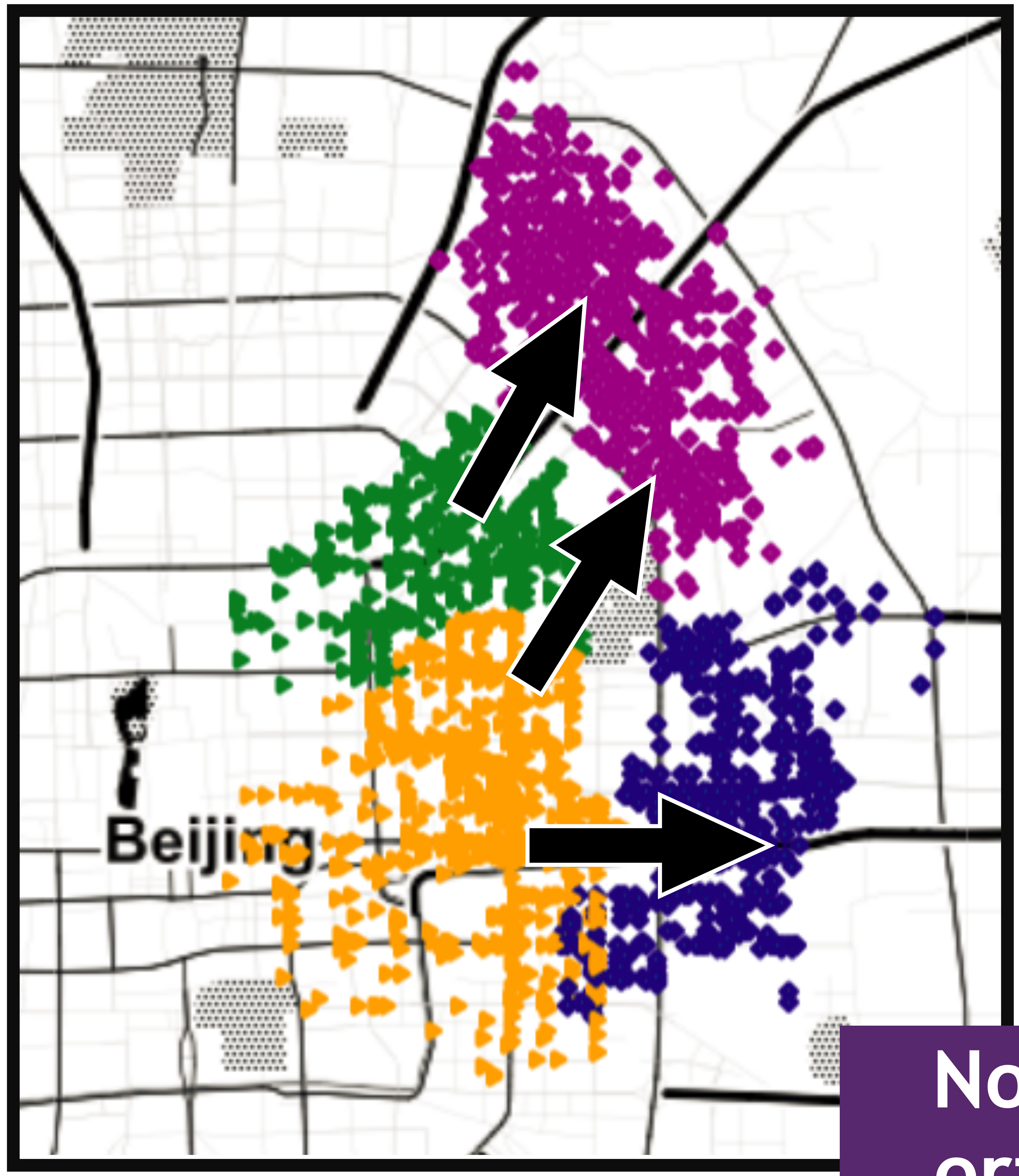
- Learn features for Origin view and Destination view

Multi-view learning of mobility features

Traditional Approach



Our Approach **[Lian et. al. 2019]**

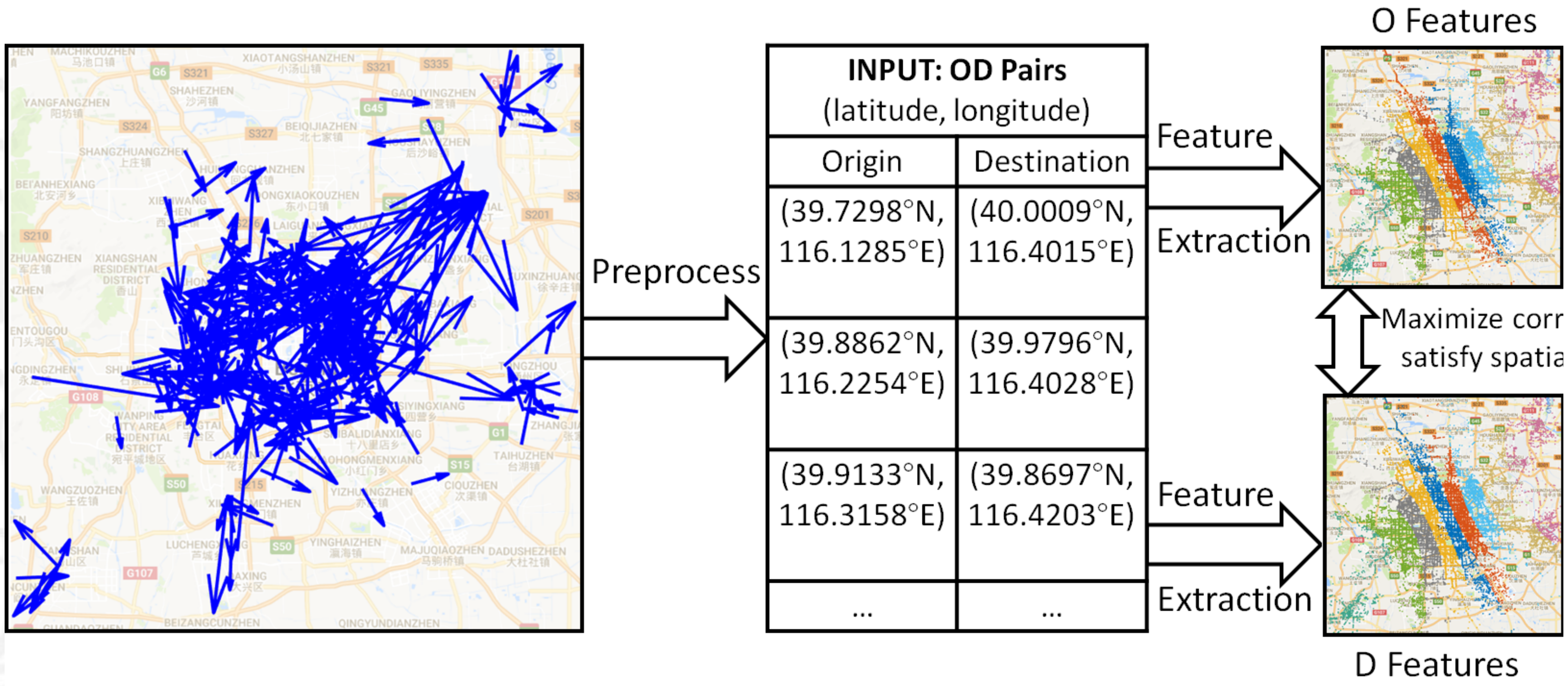


No overlap among origin/destination regions

- Learn features for Origin view and Destination view

System Architecture: KACE

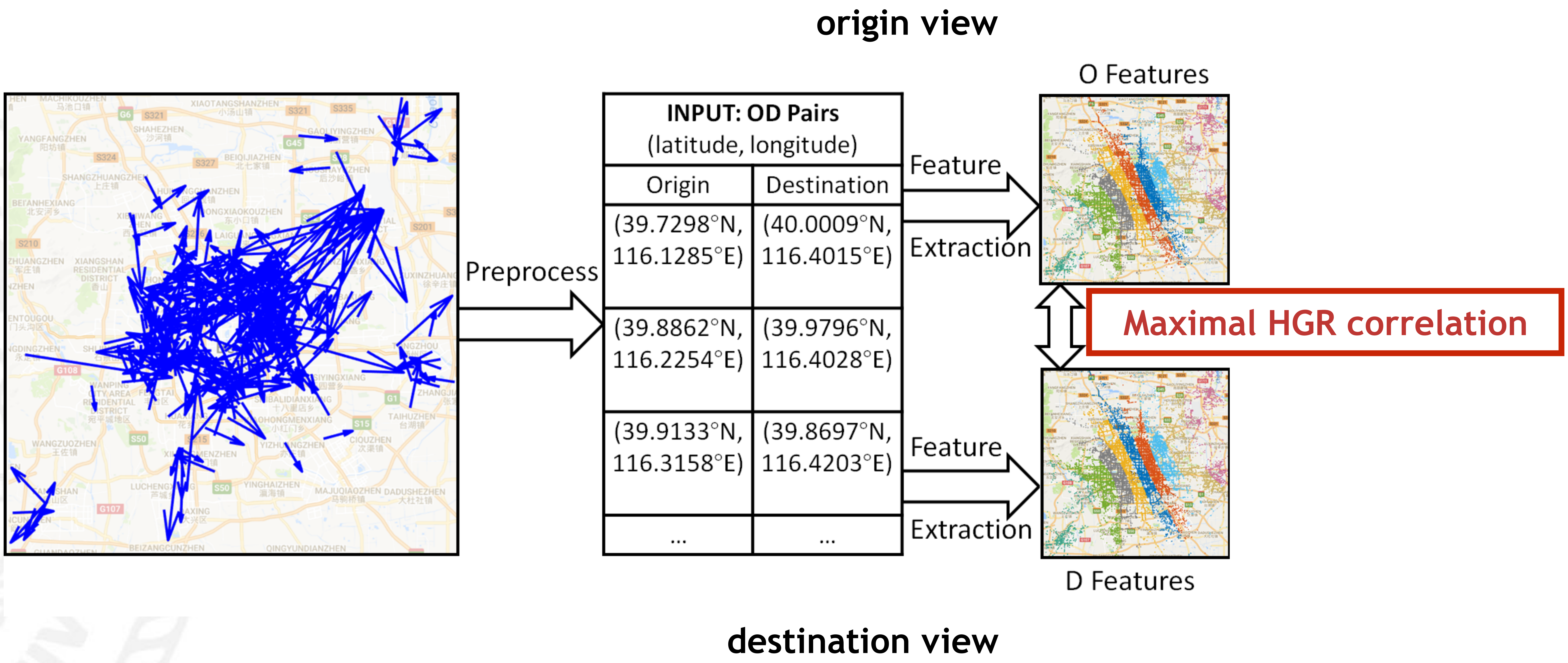
origin view



INPUT: OD Pairs (latitude, longitude)	
Origin	Destination
(39.7298°N, 116.1285°E)	(40.0009°N, 116.4015°E)
(39.8862°N, 116.2254°E)	(39.9796°N, 116.4028°E)
(39.9133°N, 116.3158°E)	(39.8697°N, 116.4203°E)
…	…

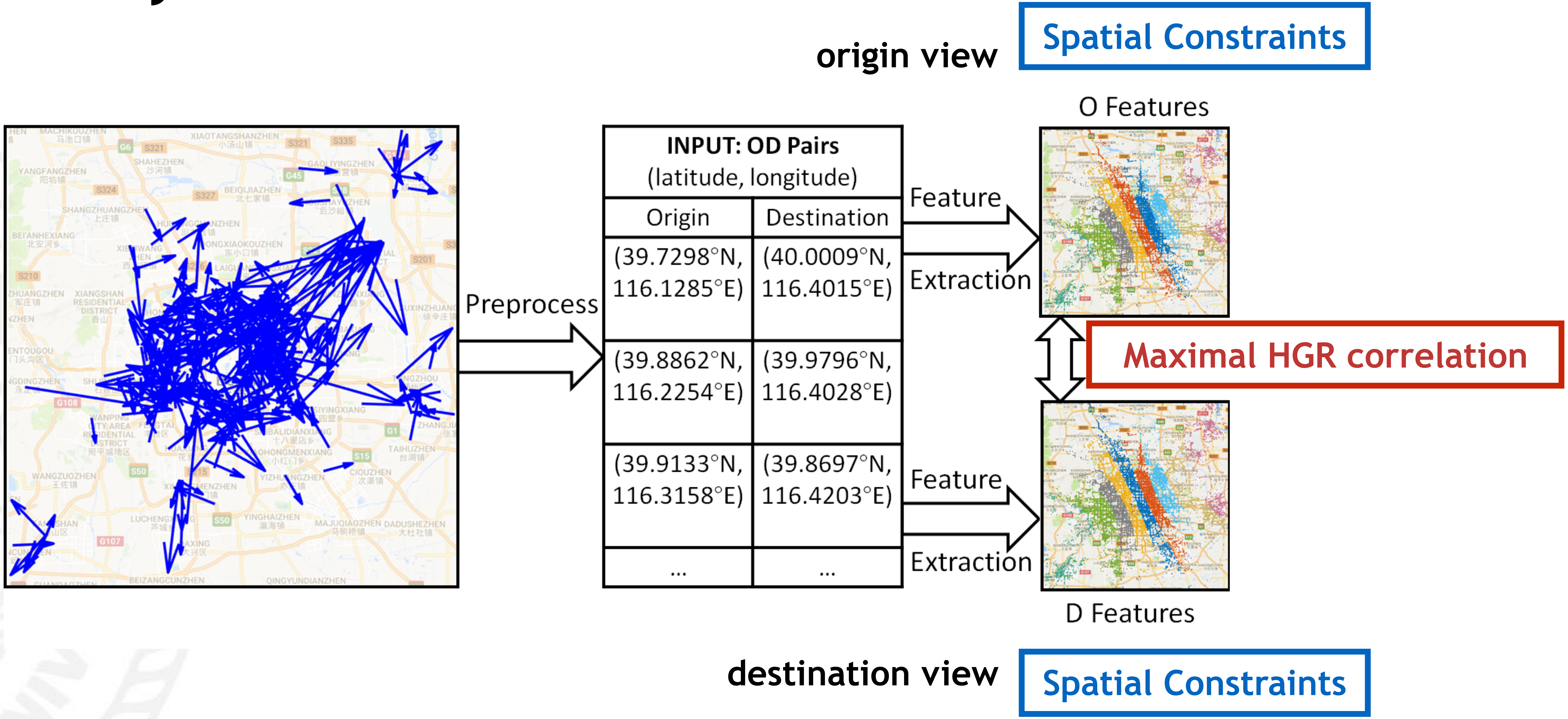
destination view

System Architecture: KACE



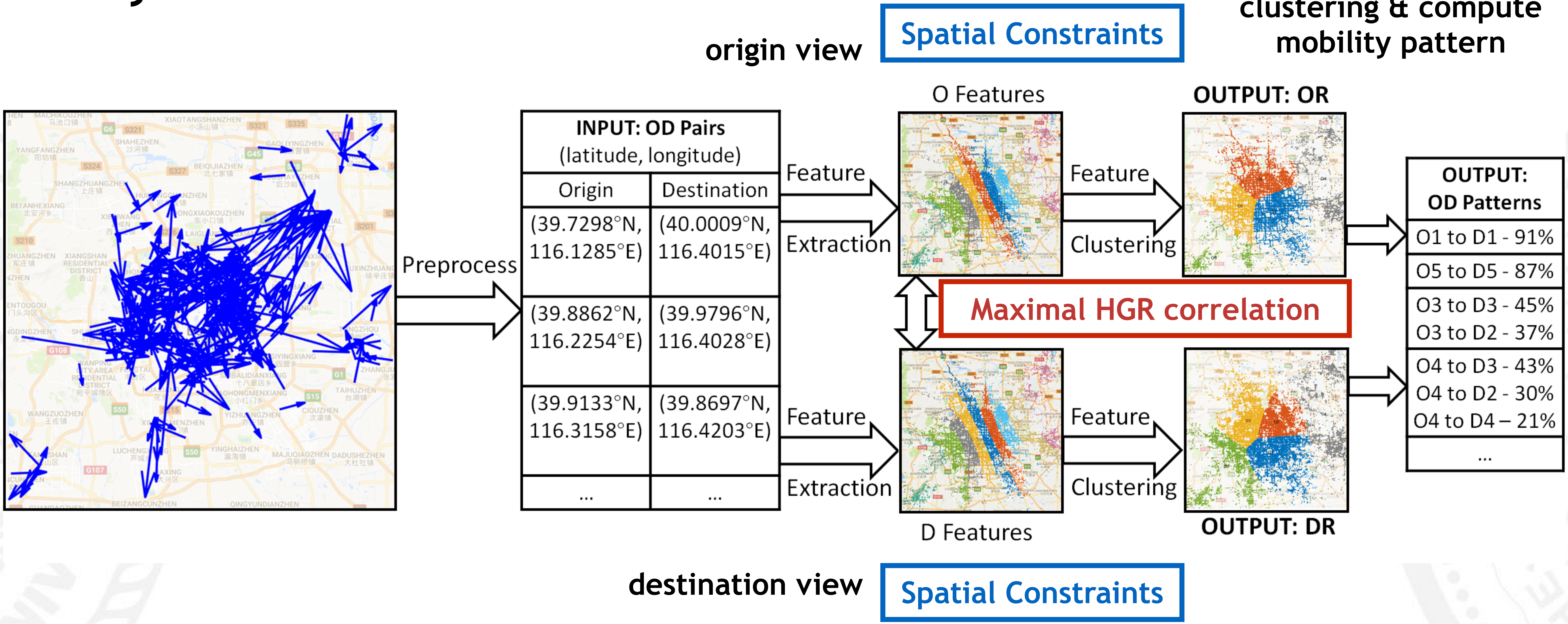
INPUT: OD Pairs (latitude, longitude)	
Origin	Destination
(39.7298°N, 116.1285°E)	(40.0009°N, 116.4015°E)
(39.8862°N, 116.2254°E)	(39.9796°N, 116.4028°E)
(39.9133°N, 116.3158°E)	(39.8697°N, 116.4203°E)
…	…

System Architecture: KACE



INPUT: OD Pairs (latitude, longitude)	
Origin	Destination
(39.7298°N, 116.1285°E)	(40.0009°N, 116.4015°E)
(39.8862°N, 116.2254°E)	(39.9796°N, 116.4028°E)
(39.9133°N, 116.3158°E)	(39.8697°N, 116.4203°E)
...	...

System Architecture: KACE



origin view

Spatial Constraints

clustering & compute mobility pattern

Preprocess

INPUT: OD Pairs (latitude, longitude)	
Origin	Destination
(39.7298°N, 116.1285°E)	(40.0009°N, 116.4015°E)
(39.8862°N, 116.2254°E)	(39.9796°N, 116.4028°E)
(39.9133°N, 116.3158°E)	(39.8697°N, 116.4203°E)
…	…

Feature Extraction

O Features

Feature Clustering

OUTPUT: OR

Maximal HGR correlation

Feature Extraction

D Features

Feature Clustering

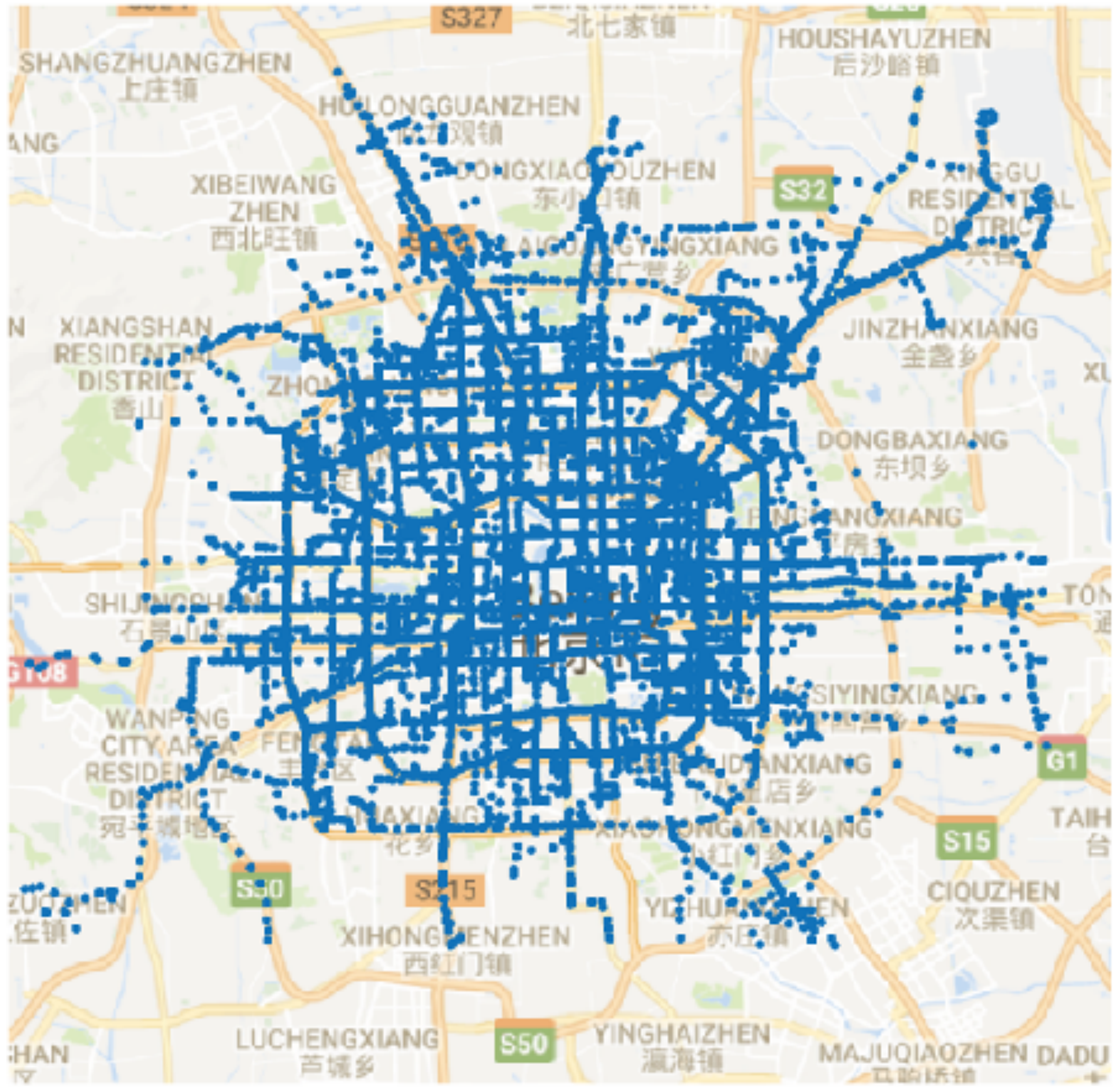
OUTPUT: DR

OUTPUT: OD Patterns
O1 to D1 - 91%
O5 to D5 - 87%
O3 to D3 - 45% O3 to D2 - 37%
O4 to D3 - 43% O4 to D2 - 30% O4 to D4 – 21%
…

destination view

Spatial Constraints

Experiment Data

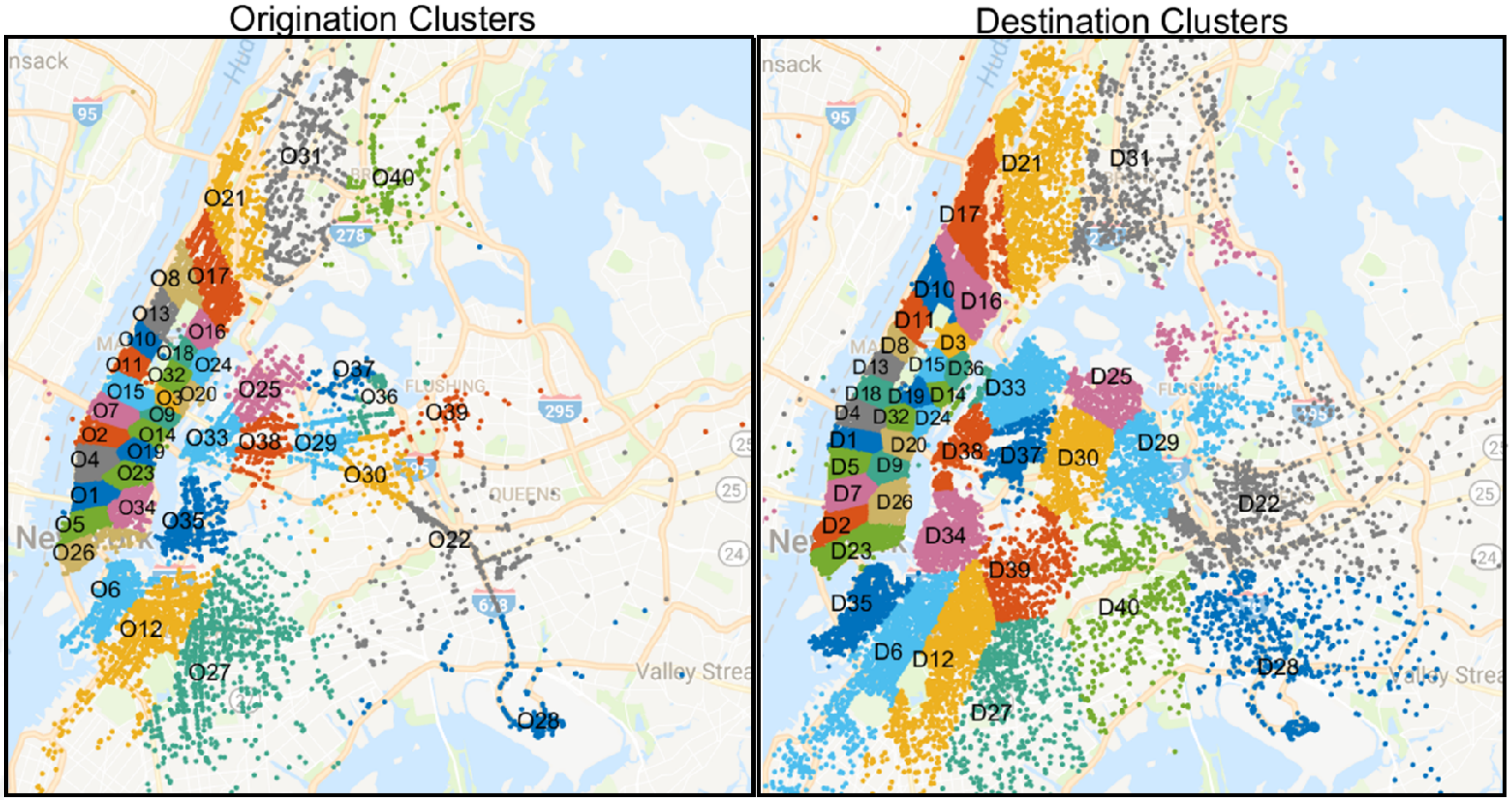


- Weekdays’ data in Nov. 2015
- Beijing: Extract OD pairs from taxi trajectories
- NYC: Open data published by NYC TLC

		Beijing	NYC
17:00-17:59	Total Trip Number	118433	213175
	Average OD Distance (km)	3.63	2.95
	OD Filtered Trip Number	54199	127648
7:00-7:59	Total Trip Number	116817	208336
	Average OD Distance (km)	4.71	3.38
	OD Filtered Trip Number	65330	137140

NYC Results

Recovers the block city topology of Manhattan

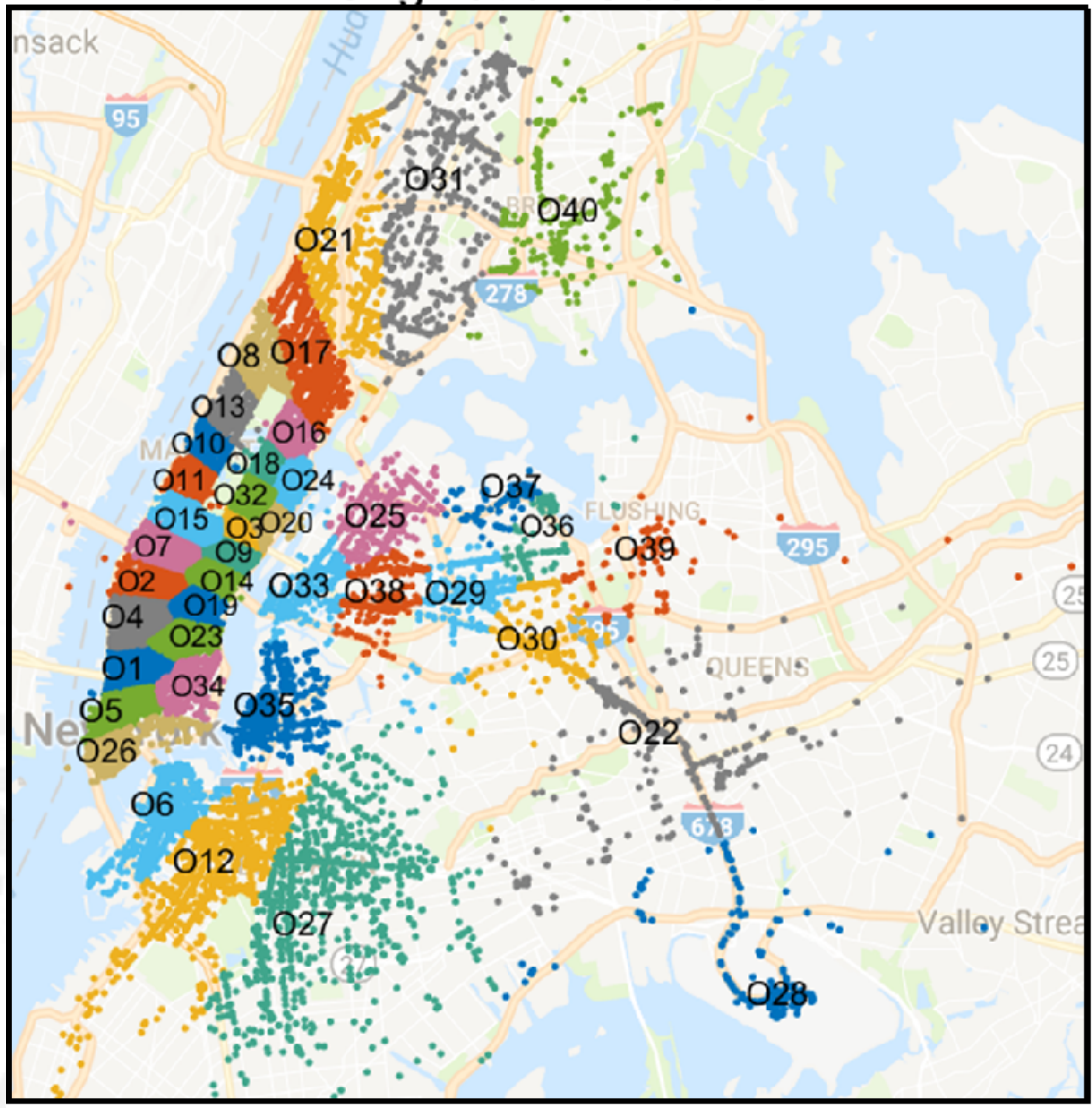


Patterns with $P_{D|O} \sim 0.5$ | ● O40 to D31-54.49%/ to D21-24.85% ● O31 to D21-48.59%/ to D31-28.08%

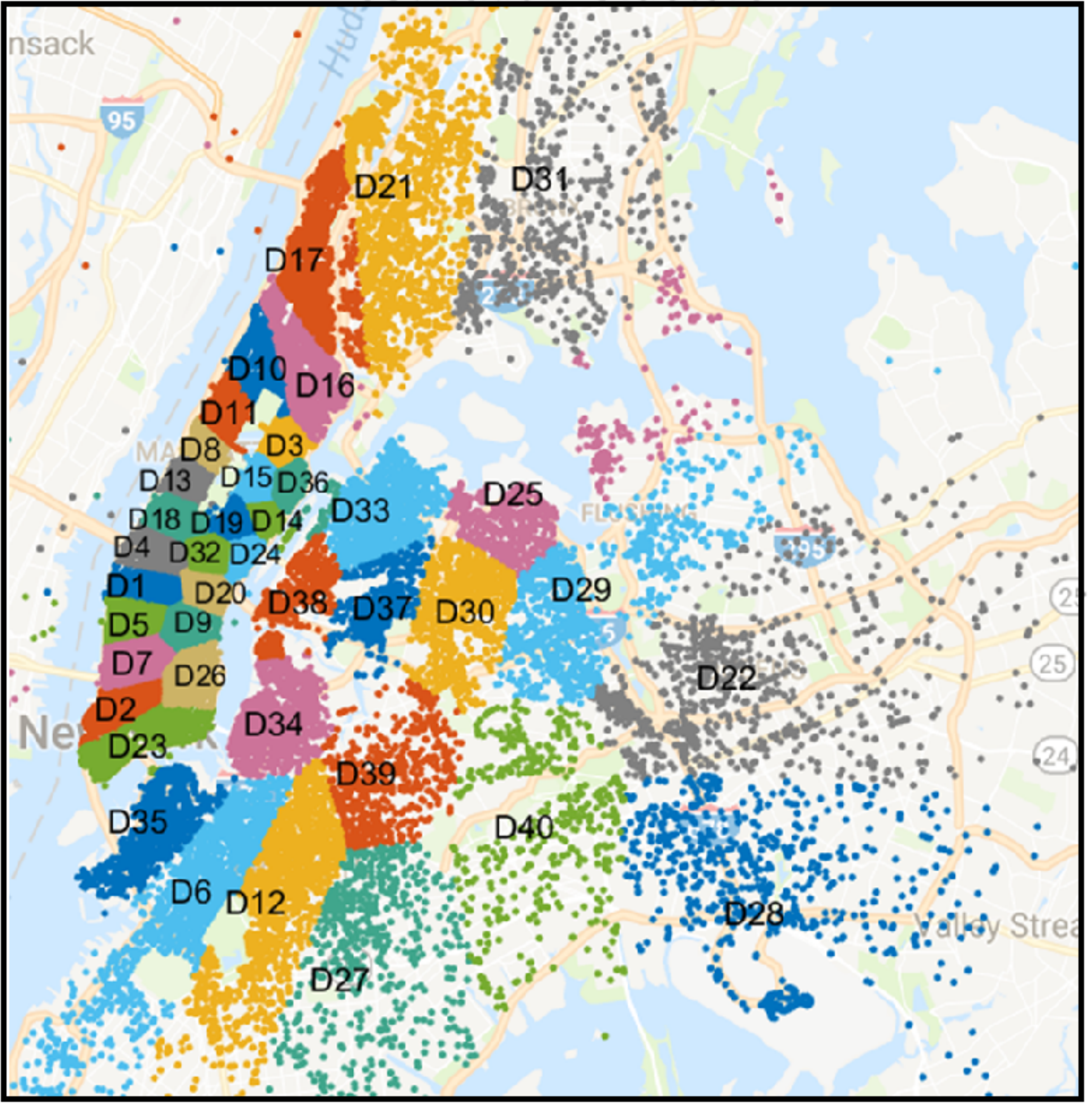
NYC Results

Recovers the block city topology of Manhattan

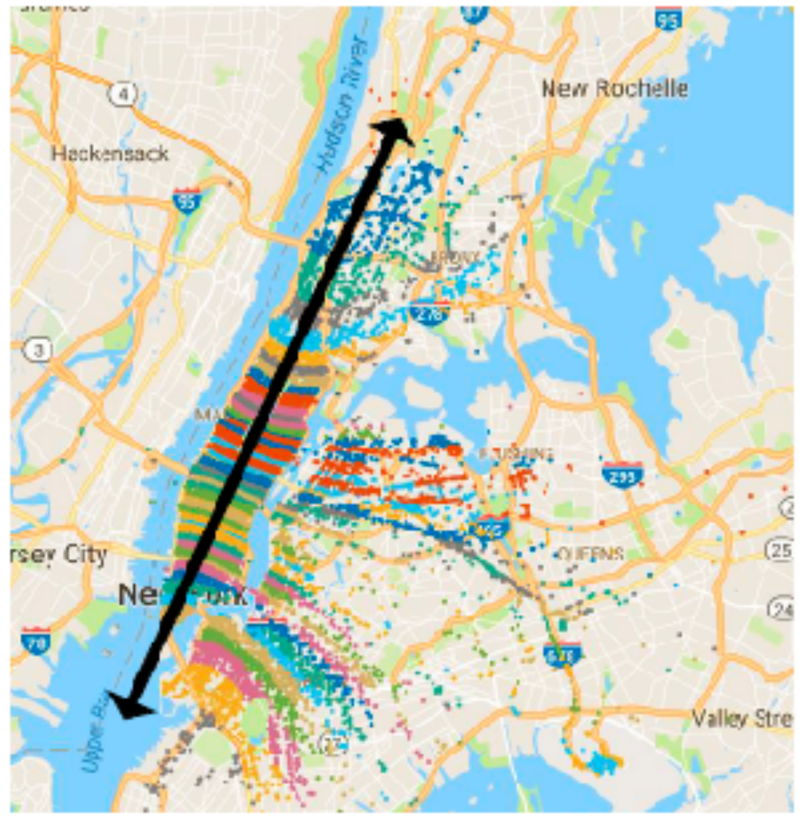
Origination Clusters



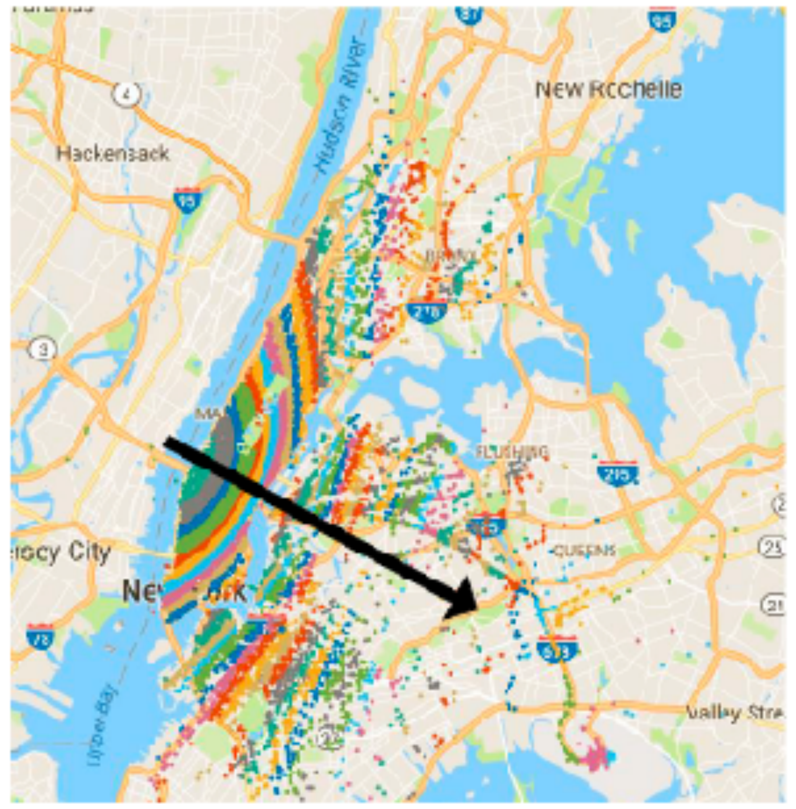
Destination Clusters



different values of f(X)



Dim=1



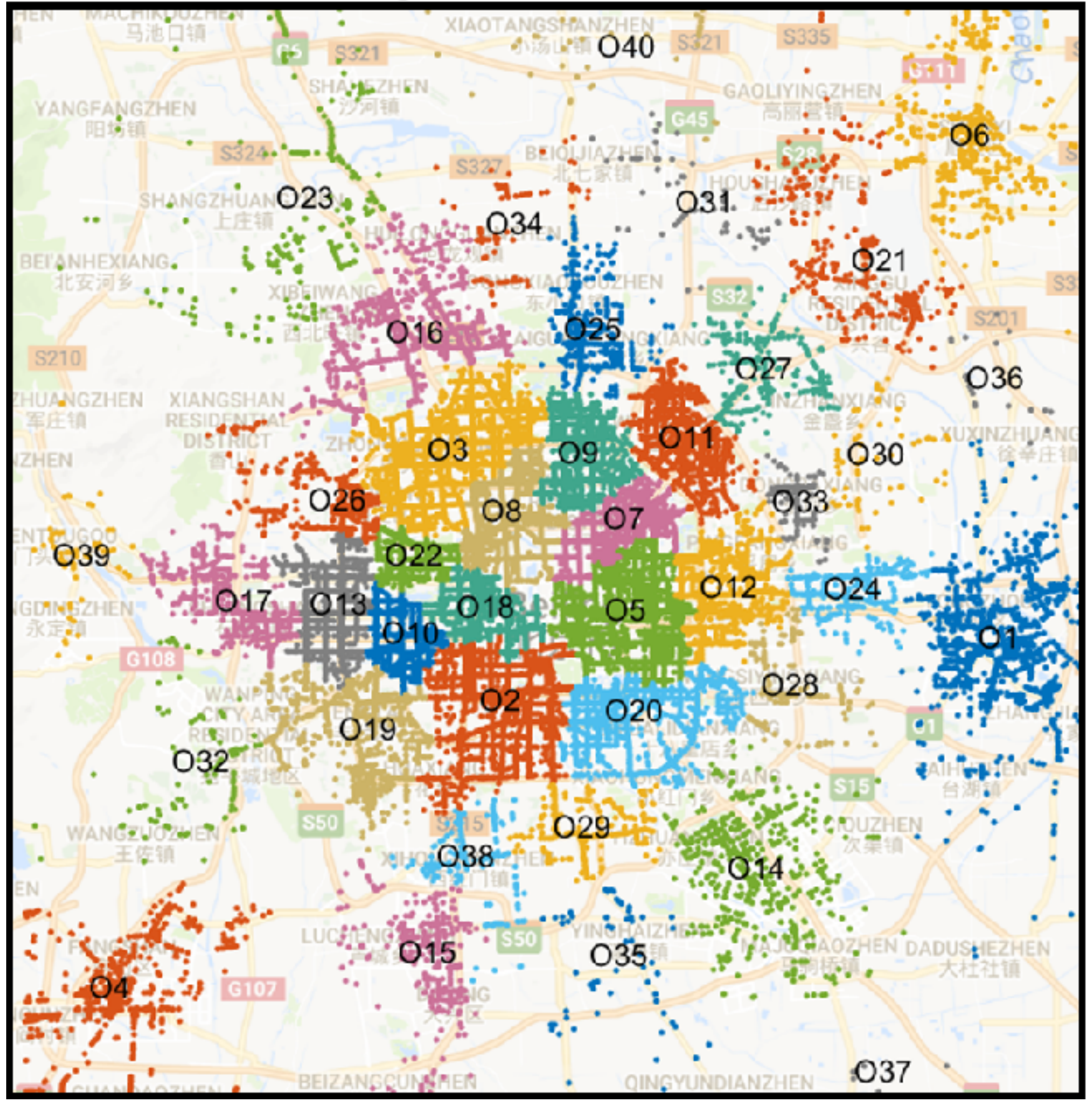
Dim=2

Patterns with $P_{D|O} \sim 0.5$ | ● O40 to D31-54.49%/ to D21-24.85% ● O31 to D21-48.59%/ to D31-28.08%

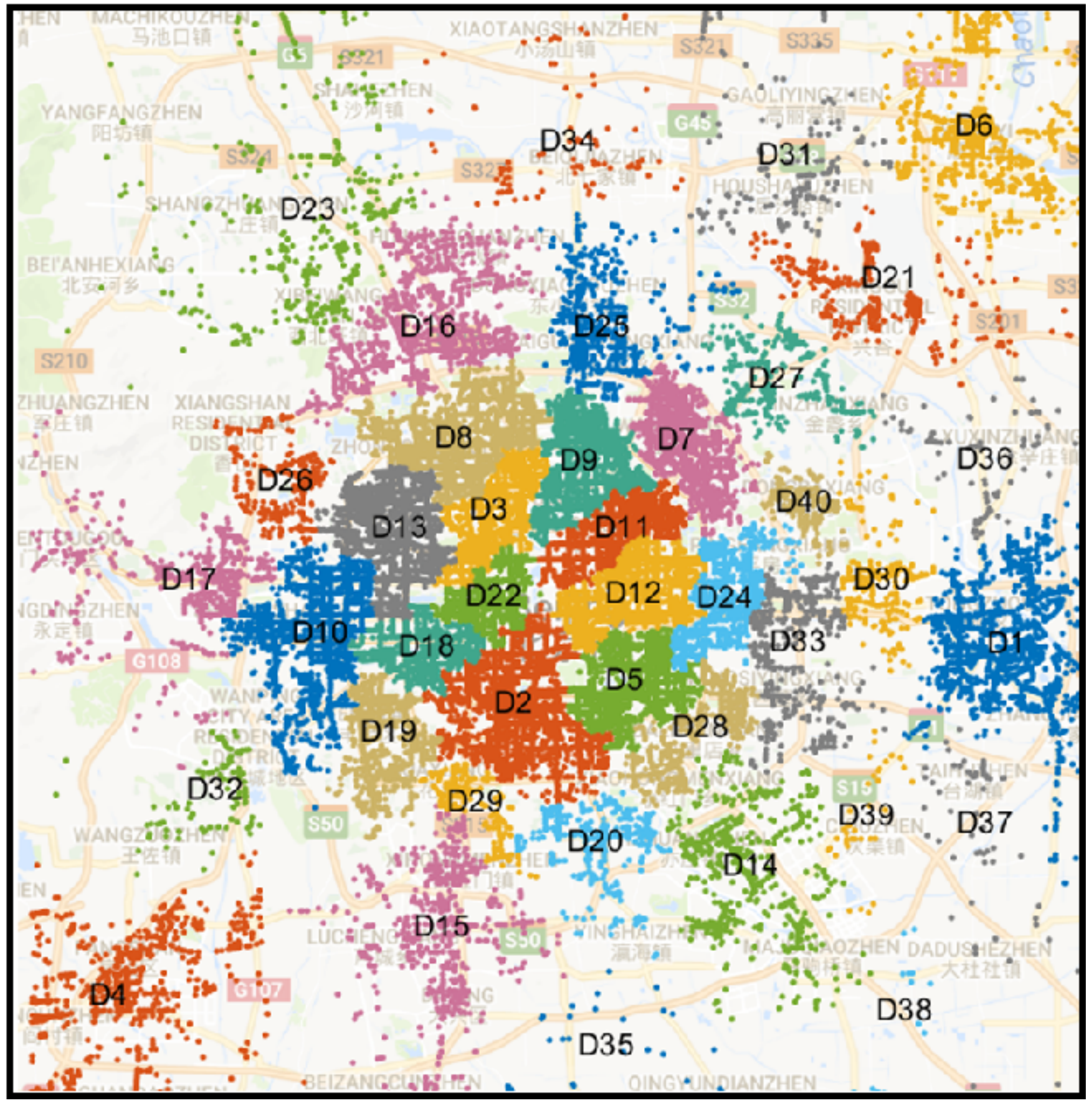
Beijing Results

- Recovers the ring-like city topology of Beijing

Origination Clusters



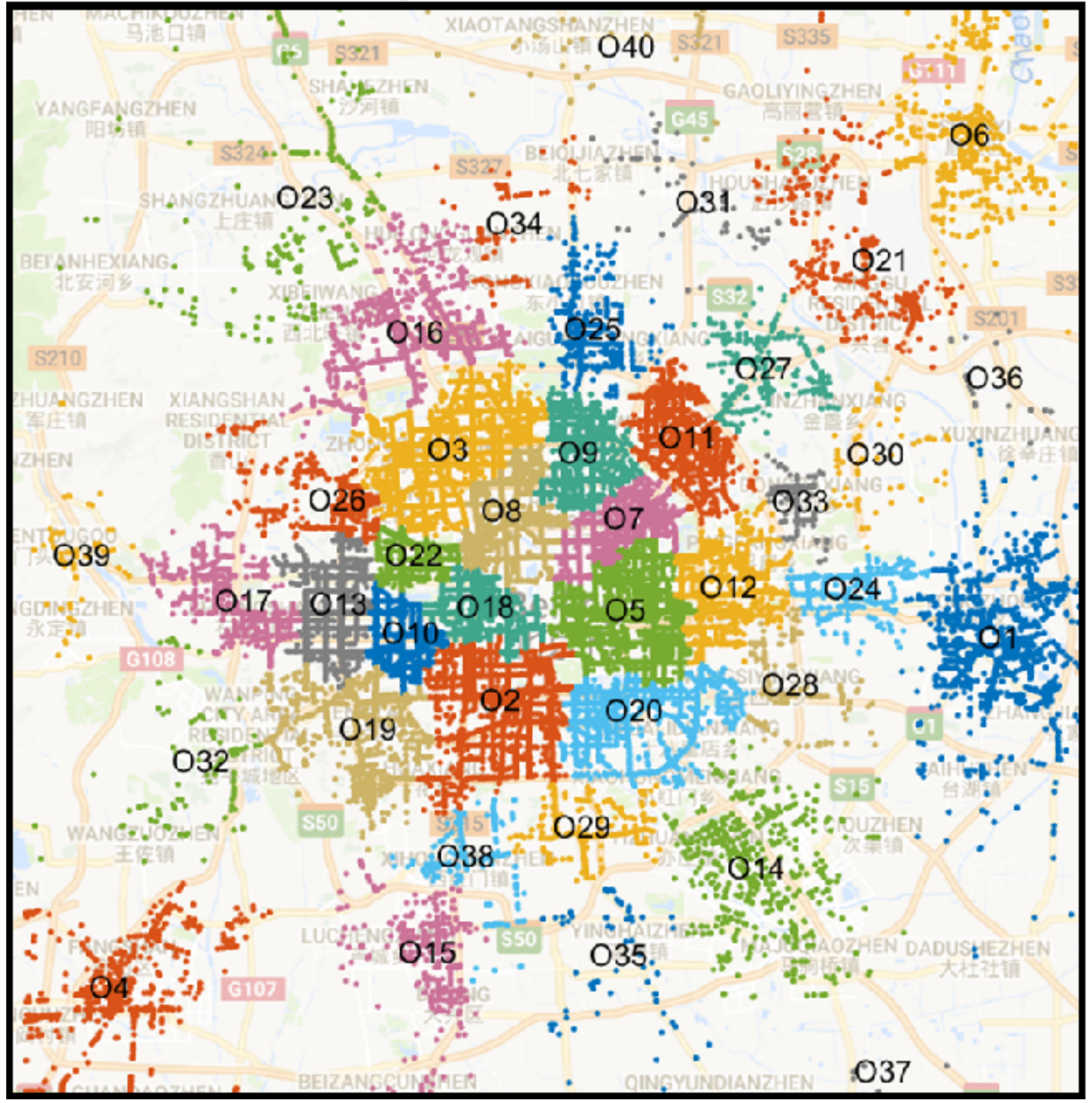
Destination Clusters



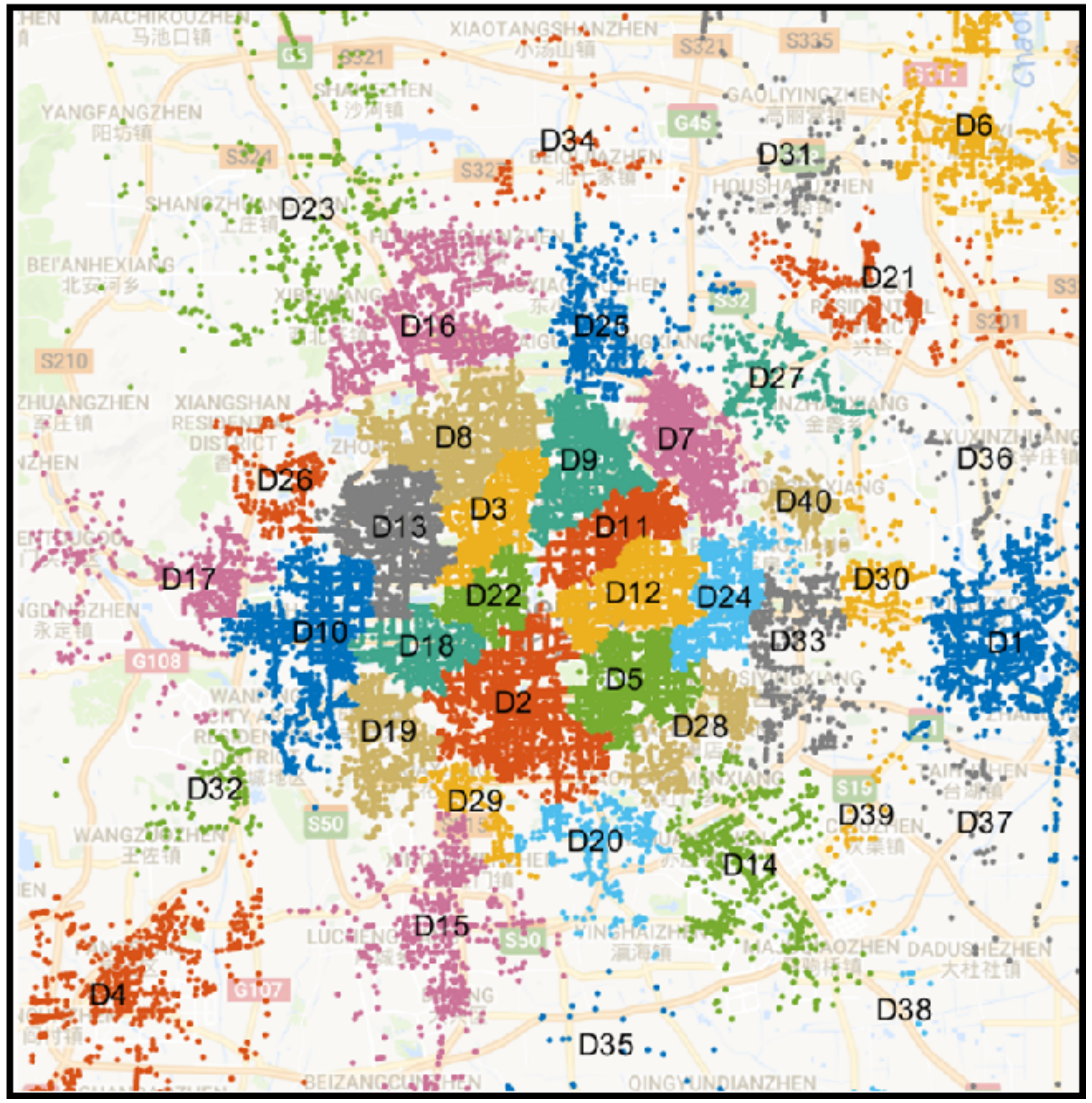
Beijing Results

- Recovers the ring-like city topology of Beijing

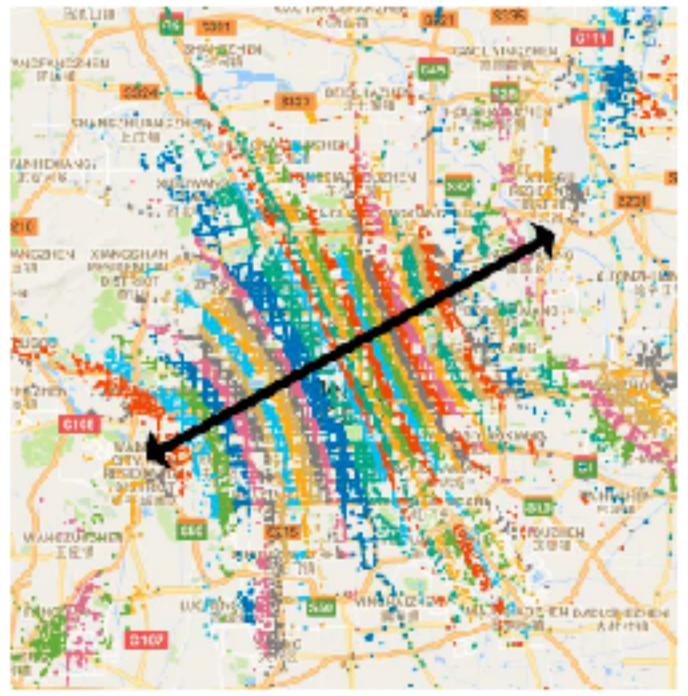
Origination Clusters



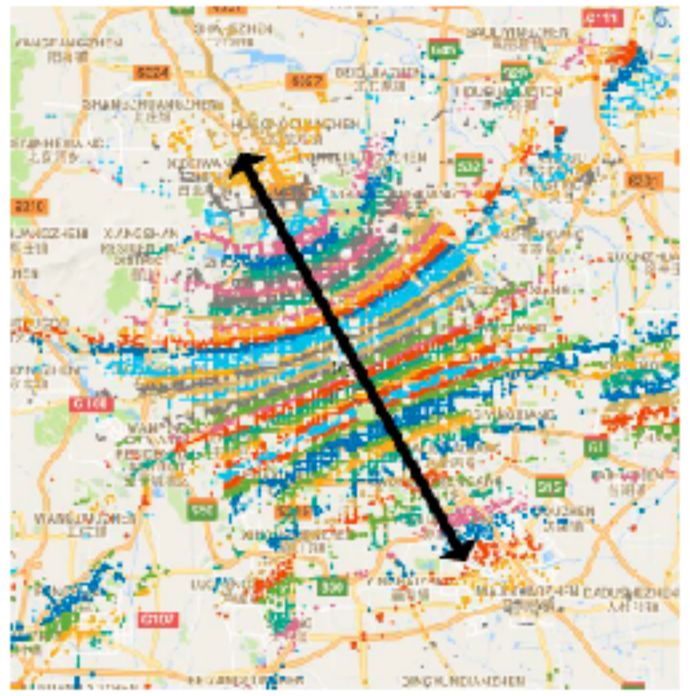
Destination Clusters



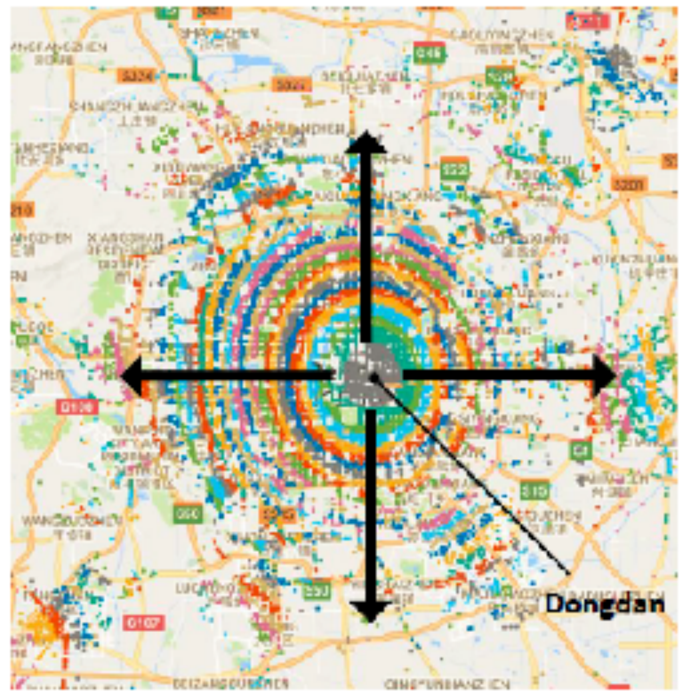
different values of f(X)



Dim=1

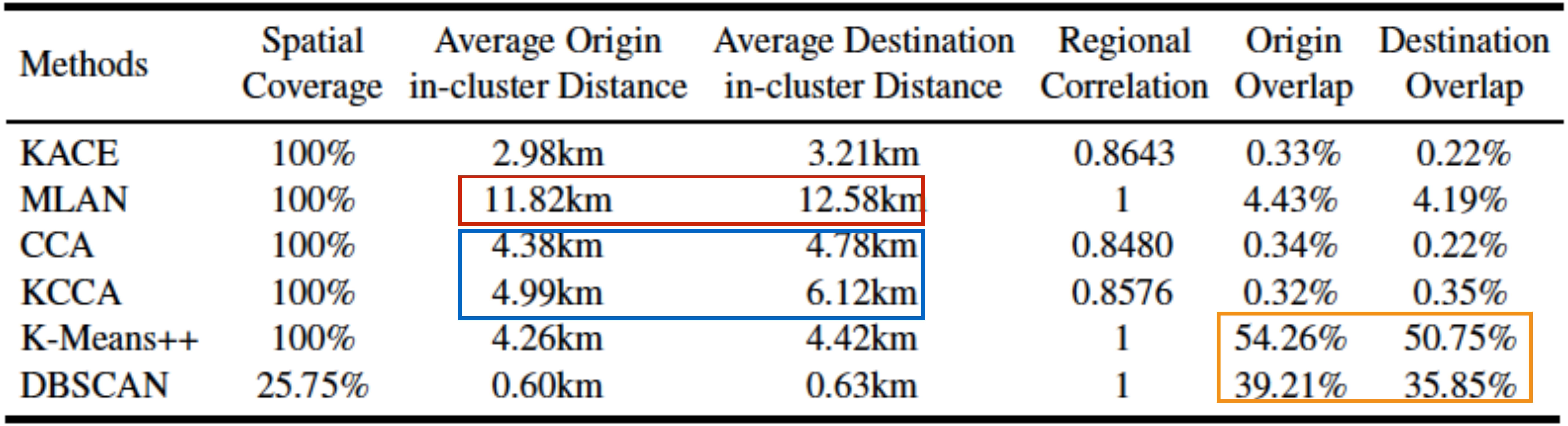


Dim=2



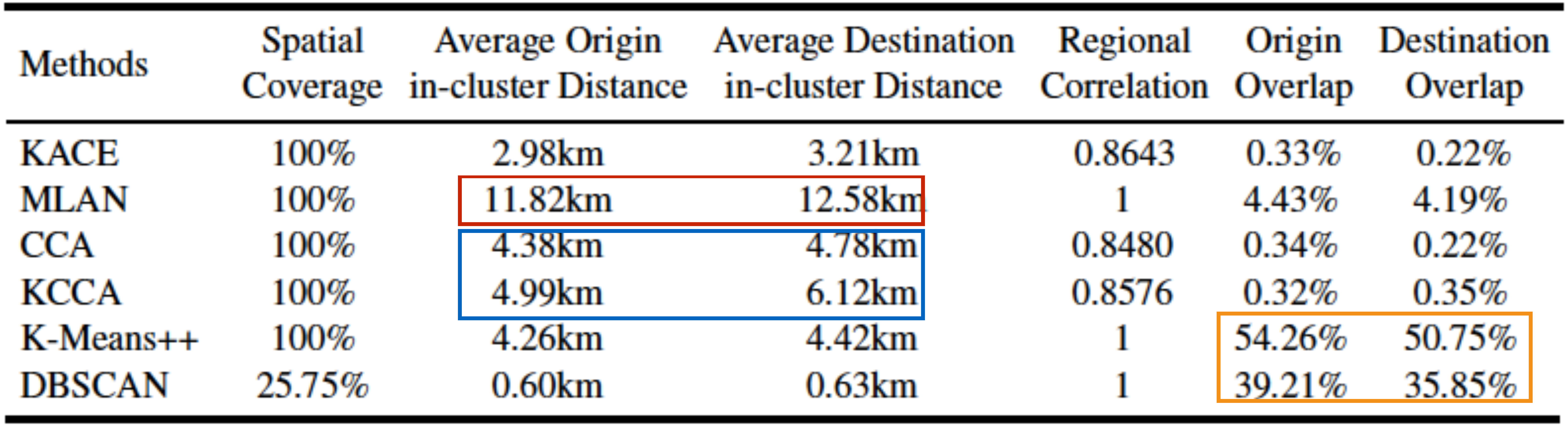
Dim=3

Comparison with Other Methods



Methods	Spatial Coverage	Average Origin in-cluster Distance	Average Destination in-cluster Distance	Regional Correlation	Origin Overlap	Destination Overlap
KACE	100%	2.98km	3.21km	0.8643	0.33%	0.22%
MLAN	100%	11.82km	12.58km	1	4.43%	4.19%
CCA	100%	4.38km	4.78km	0.8480	0.34%	0.22%
KCCA	100%	4.99km	6.12km	0.8576	0.32%	0.35%
K-Means++	100%	4.26km	4.42km	1	54.26%	50.75%
DBSCAN	25.75%	0.60km	0.63km	1	39.21%	35.85%

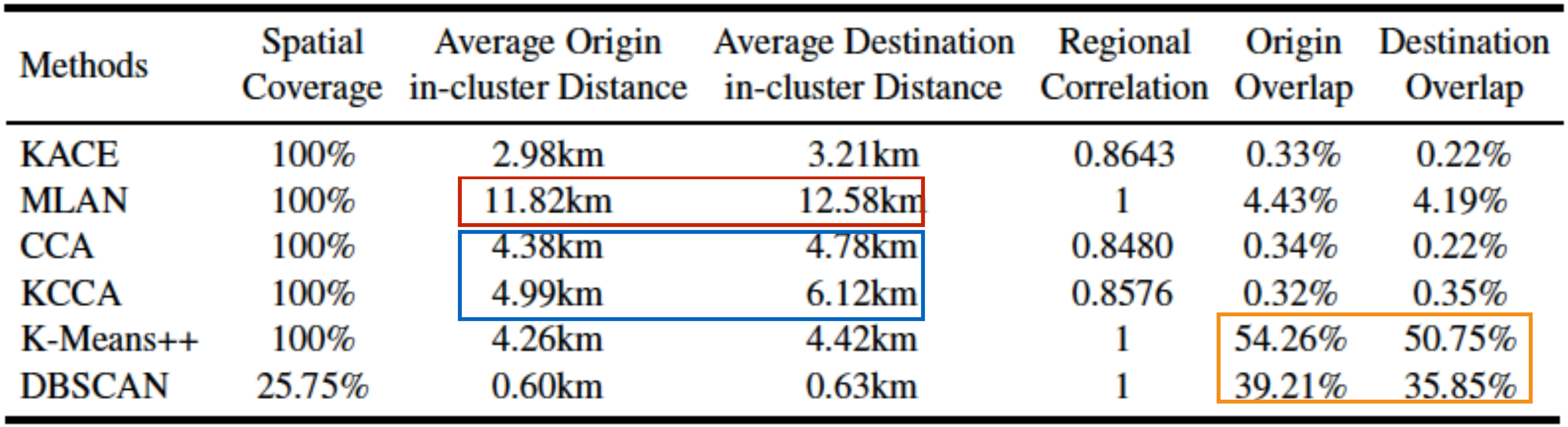
Comparison with Other Methods



Methods	Spatial Coverage	Average Origin in-cluster Distance	Average Destination in-cluster Distance	Regional Correlation	Origin Overlap	Destination Overlap
KACE	100%	2.98km	3.21km	0.8643	0.33%	0.22%
MLAN	100%	11.82km	12.58km	1	4.43%	4.19%
CCA	100%	4.38km	4.78km	0.8480	0.34%	0.22%
KCCA	100%	4.99km	6.12km	0.8576	0.32%	0.35%
K-Means++	100%	4.26km	4.42km	1	54.26%	50.75%
DBSCAN	25.75%	0.60km	0.63km	1	39.21%	35.85%

- Traditional methods by clustering trips, has big overlap.

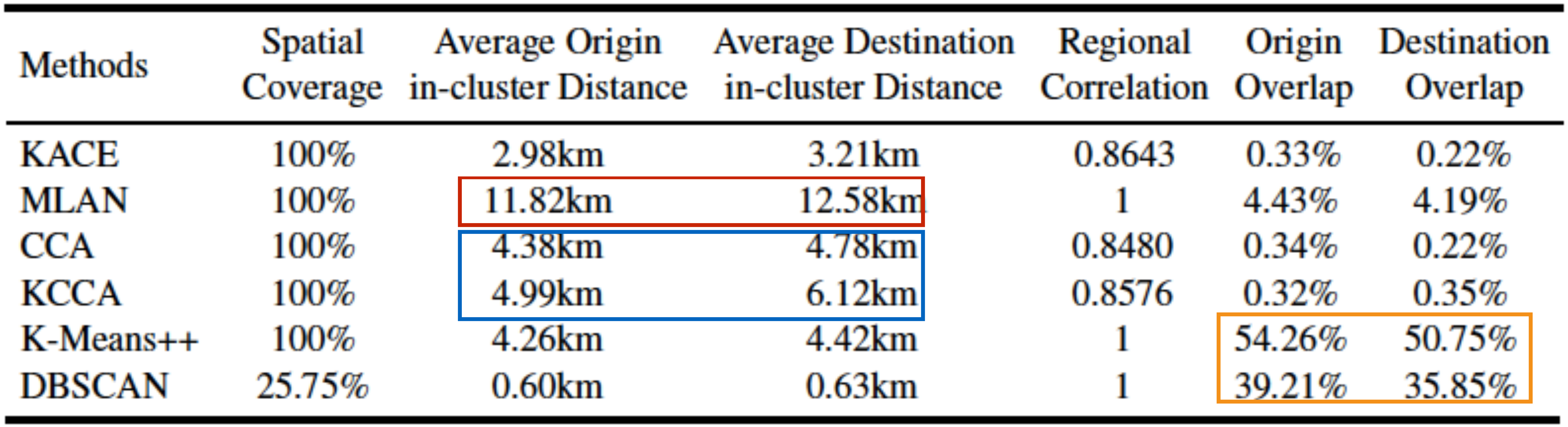
Comparison with Other Methods



Methods	Spatial Coverage	Average Origin in-cluster Distance	Average Destination in-cluster Distance	Regional Correlation	Origin Overlap	Destination Overlap
KACE	100%	2.98km	3.21km	0.8643	0.33%	0.22%
MLAN	100%	11.82km	12.58km	1	4.43%	4.19%
CCA	100%	4.38km	4.78km	0.8480	0.34%	0.22%
KCCA	100%	4.99km	6.12km	0.8576	0.32%	0.35%
K-Means++	100%	4.26km	4.42km	1	54.26%	50.75%
DBSCAN	25.75%	0.60km	0.63km	1	39.21%	35.85%

- Traditional methods by clustering trips, has big overlap.
- Multi-view clustering MLAN , CCA-based methods results in less compact clusters

Comparison with Other Methods

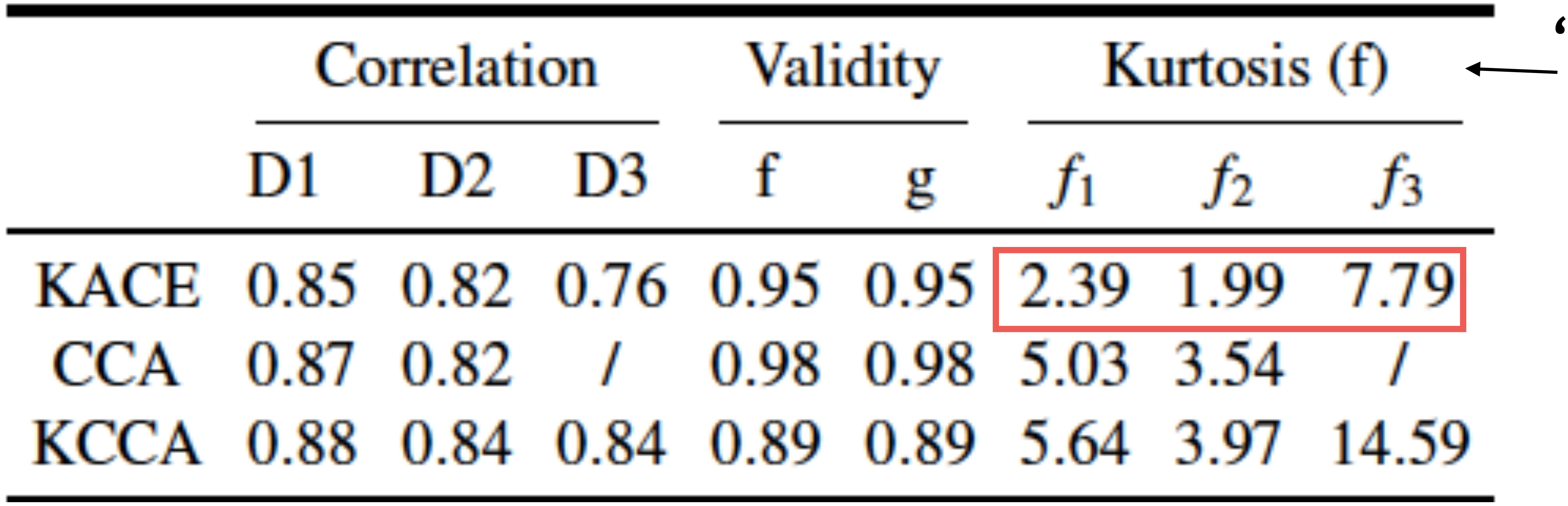


Methods	Spatial Coverage	Average Origin in-cluster Distance	Average Destination in-cluster Distance	Regional Correlation	Origin Overlap	Destination Overlap
KACE	100%	2.98km	3.21km	0.8643	0.33%	0.22%
MLAN	100%	11.82km	12.58km	1	4.43%	4.19%
CCA	100%	4.38km	4.78km	0.8480	0.34%	0.22%
KCCA	100%	4.99km	6.12km	0.8576	0.32%	0.35%
K-Means++	100%	4.26km	4.42km	1	54.26%	50.75%
DBSCAN	25.75%	0.60km	0.63km	1	39.21%	35.85%

- Traditional methods by clustering trips, has big overlap.
- Multi-view clustering MLAN , CCA-based methods results in less compact clusters
- Our method, KACE has the best overall performance

Comparison with Canonical Correlation

- Evaluate extracted features:

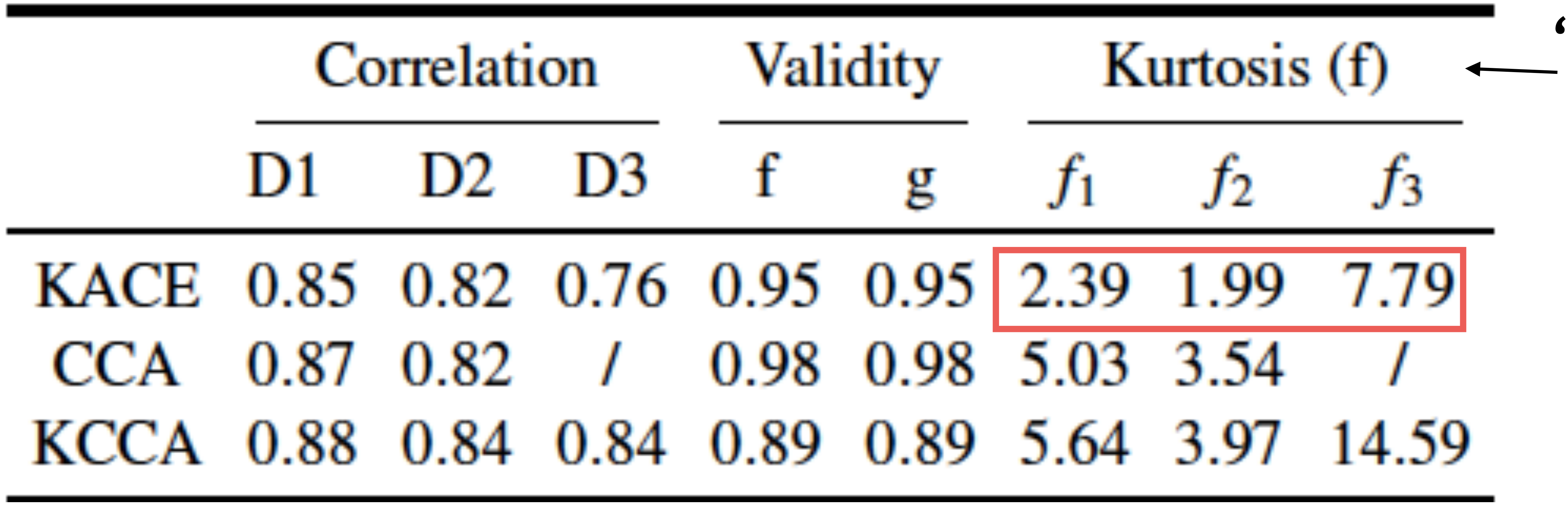


	Correlation			Validity		Kurtosis (f)		
	D1	D2	D3	f	g	f_1	f_2	f_3
KACE	0.85	0.82	0.76	0.95	0.95	2.39	1.99	7.79
CCA	0.87	0.82	/	0.98	0.98	5.03	3.54	/
KCCA	0.88	0.84	0.84	0.89	0.89	5.64	3.97	14.59

“tailedness" of a distribution

Comparison with Canonical Correlation

- Evaluate extracted features:



	Correlation			Validity		Kurtosis (f)		
	D1	D2	D3	f	g	f_1	f_2	f_3
KACE	0.85	0.82	0.76	0.95	0.95	2.39	1.99	7.79
CCA	0.87	0.82	/	0.98	0.98	5.03	3.54	/
KCCA	0.88	0.84	0.84	0.89	0.89	5.64	3.97	14.59

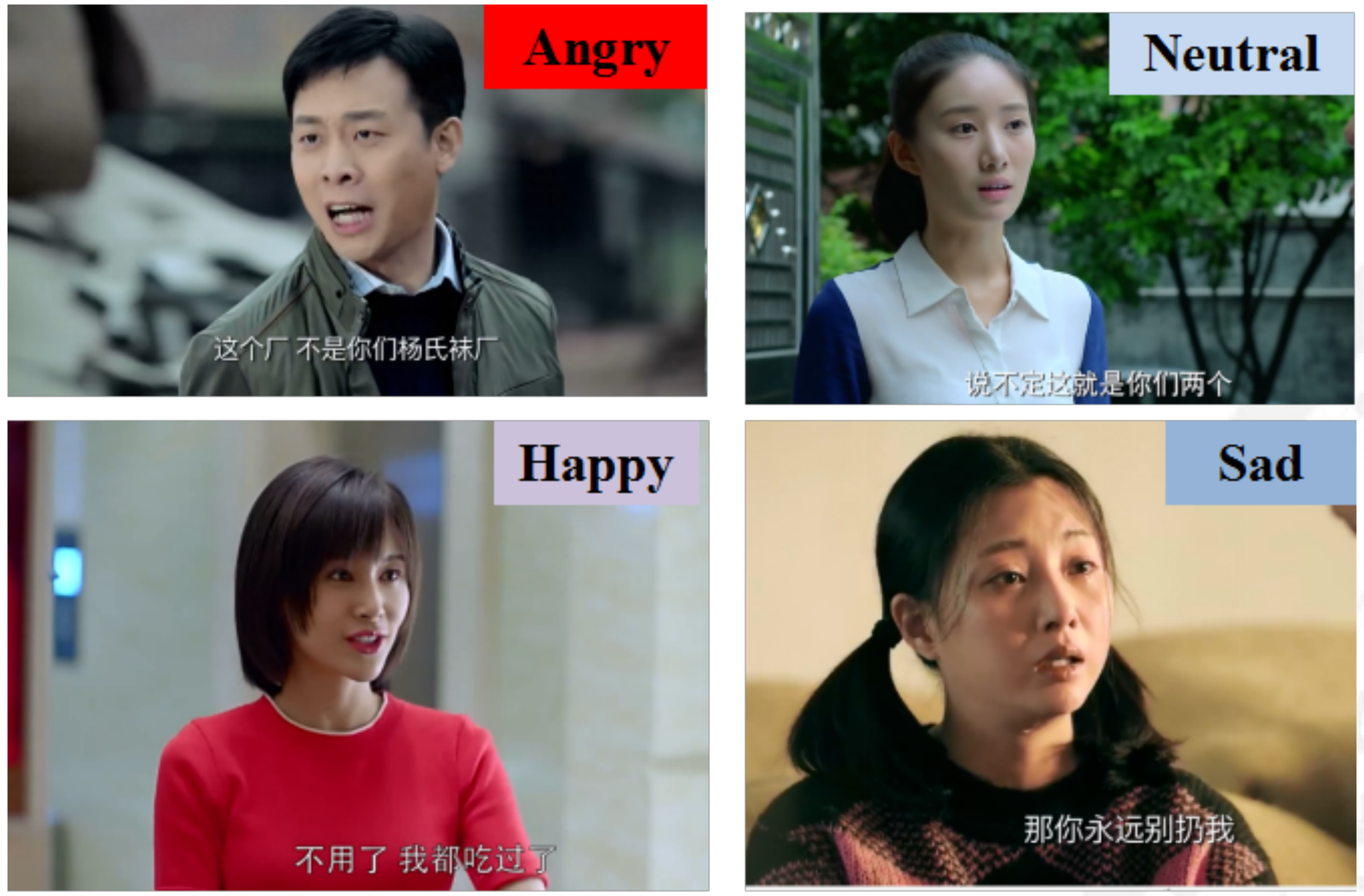
“tailedness" of a distribution

KACE features have much smaller Kurtosis than CCA/KCCA

Application II: Multi-Modal Emotion Recognition

[Ma et. al. 2019]

- Goal: classify emotion from audio and visual data
  - important for machine-based understanding

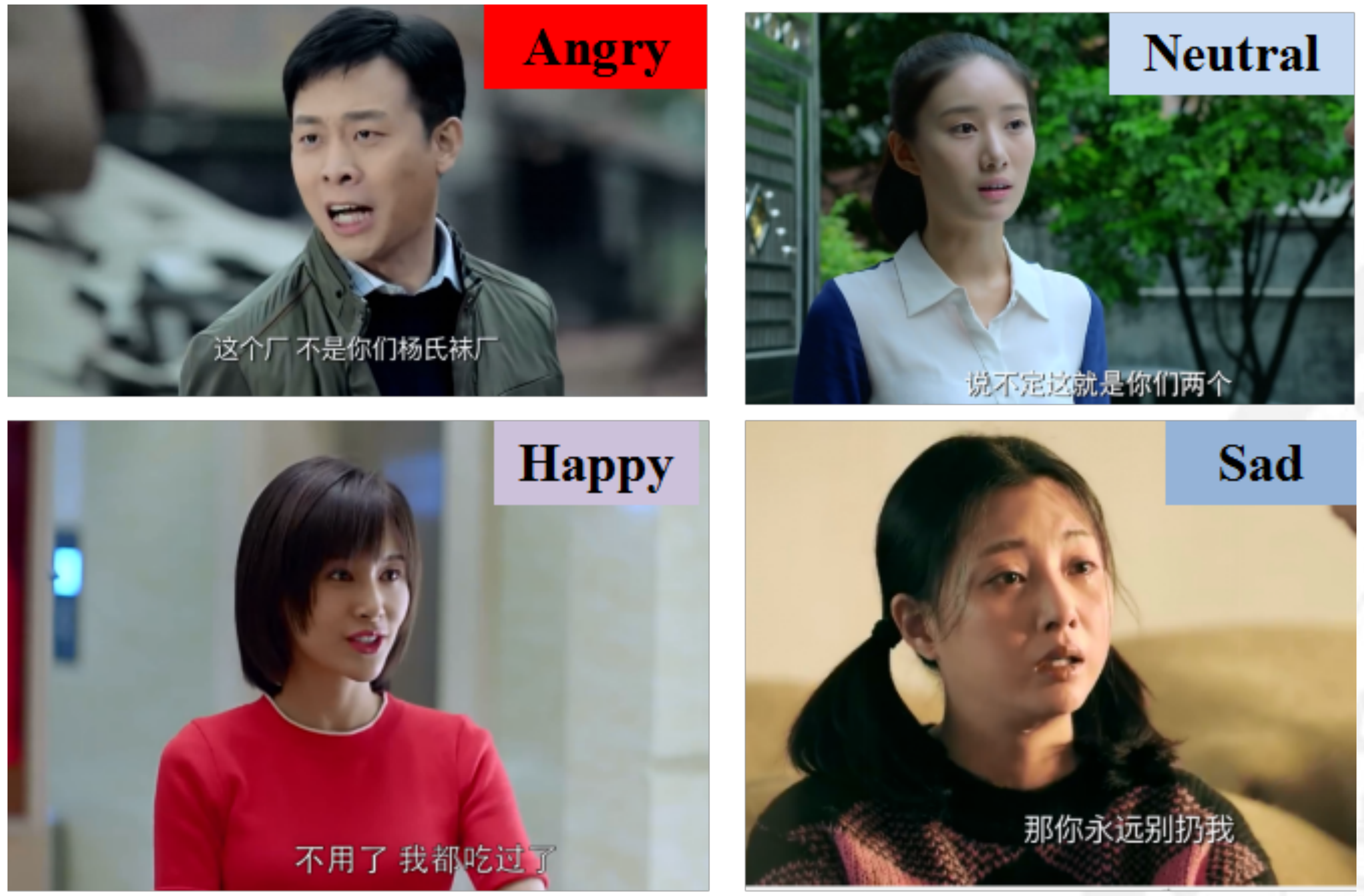


example of movie annotation

Application II: Multi-Modal Emotion Recognition

[Ma et. al. 2019]

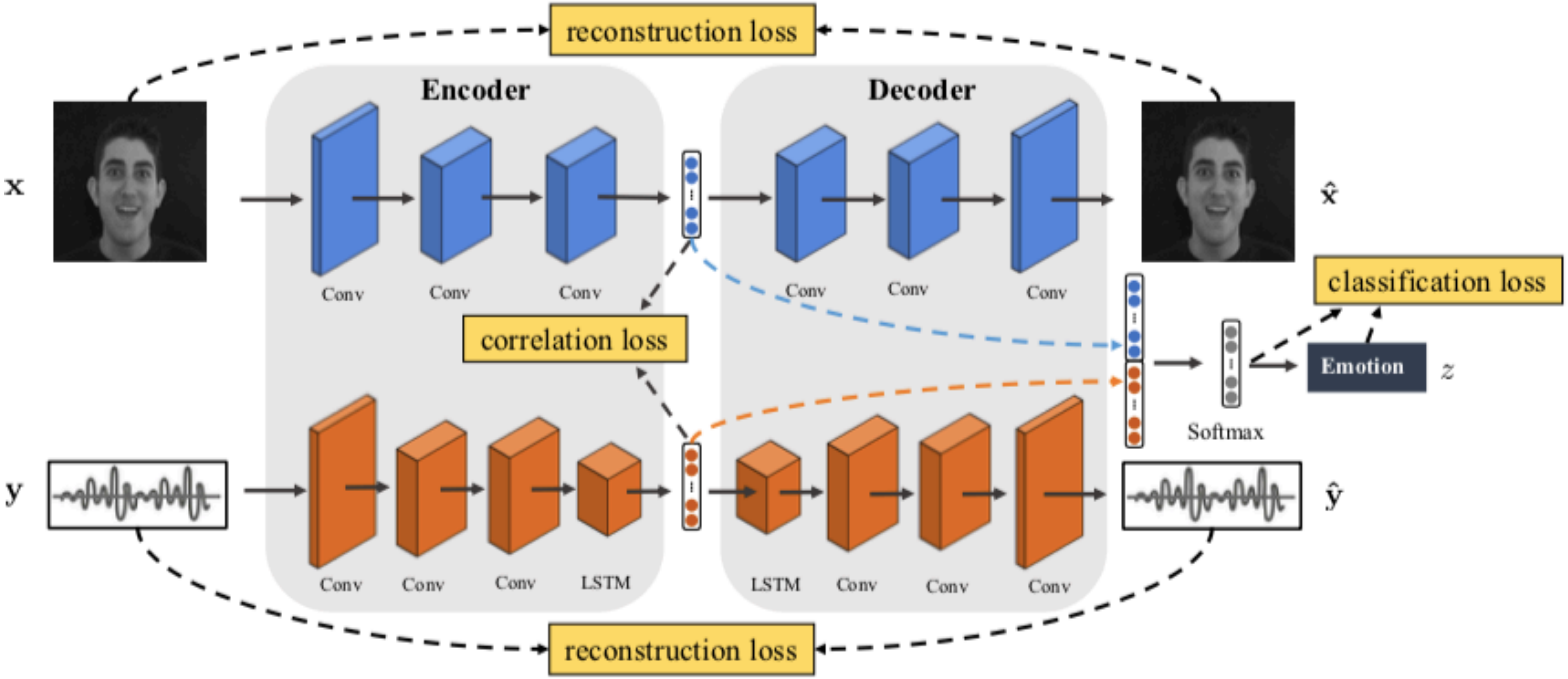
- Goal: classify emotion from audio and visual data
  - important for machine-based understanding



example of movie annotation

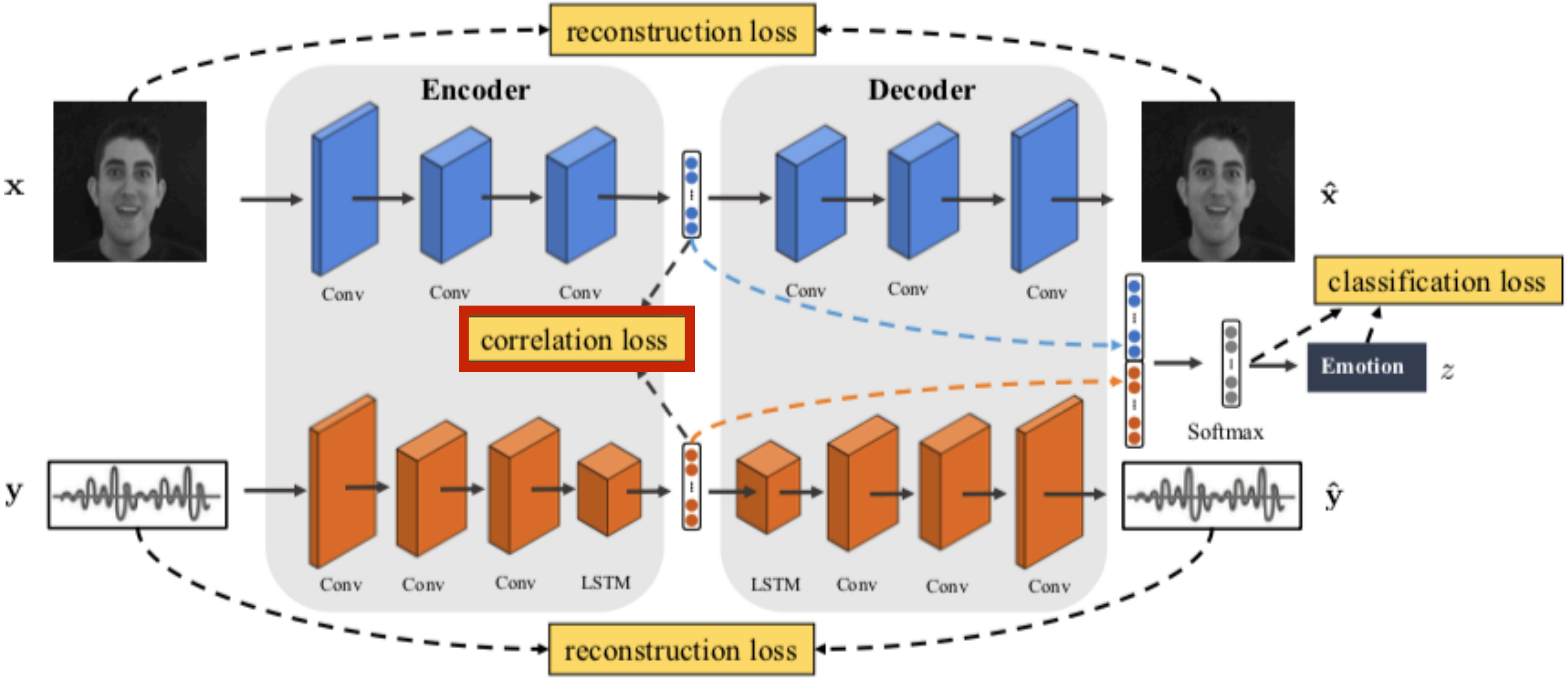
- Challenge: disentangling private and public information

System Architecture



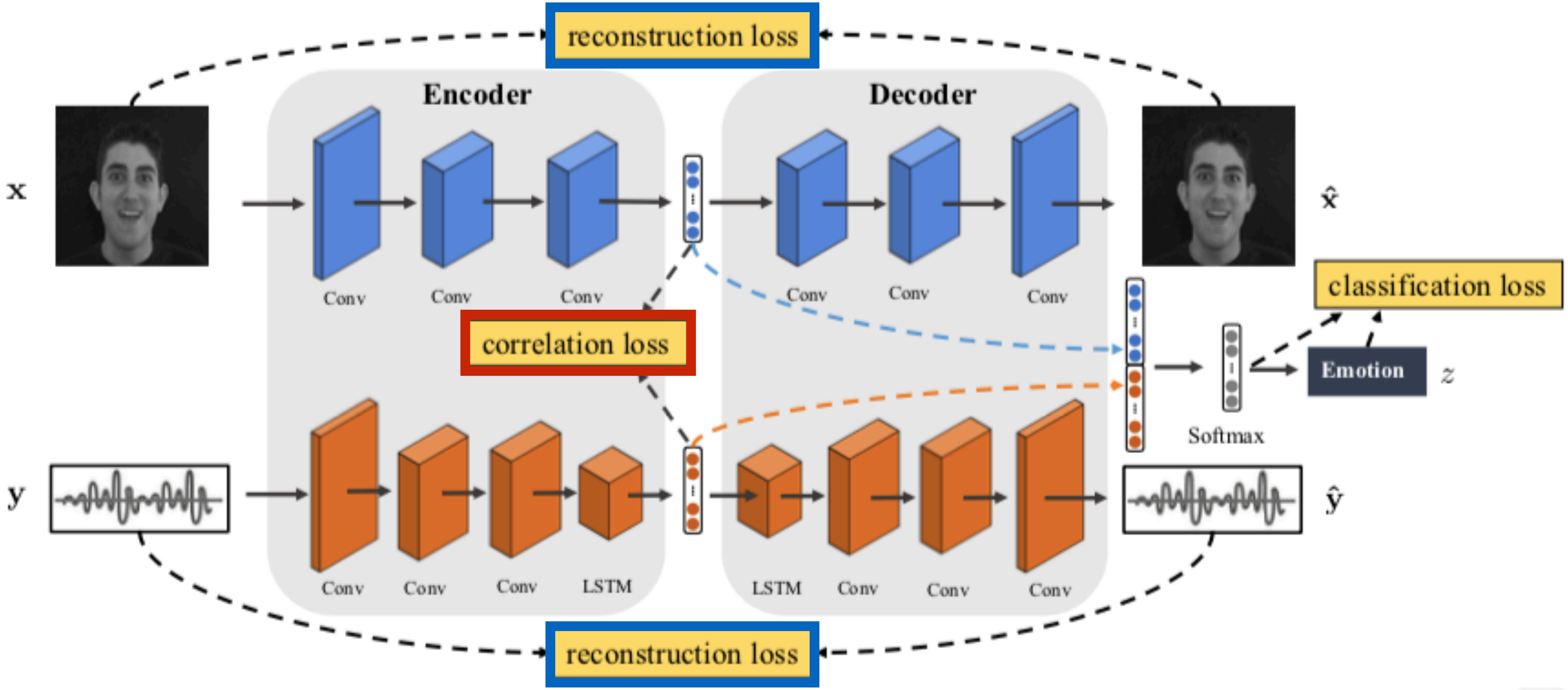
System Architecture

- Public information: maximize correlation between modalities



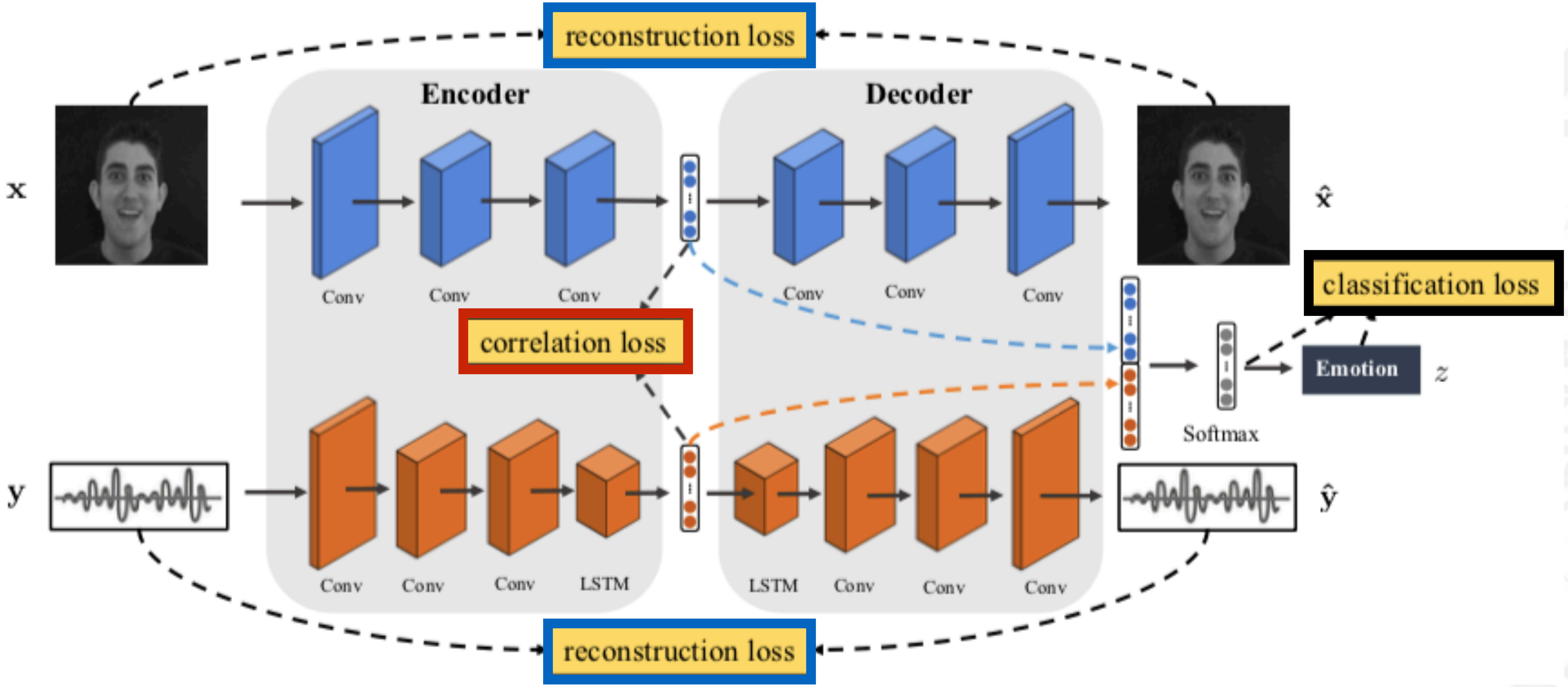
System Architecture

- Public information: maximize correlation between modalities
- Private information: preserving structure of each modality



System Architecture

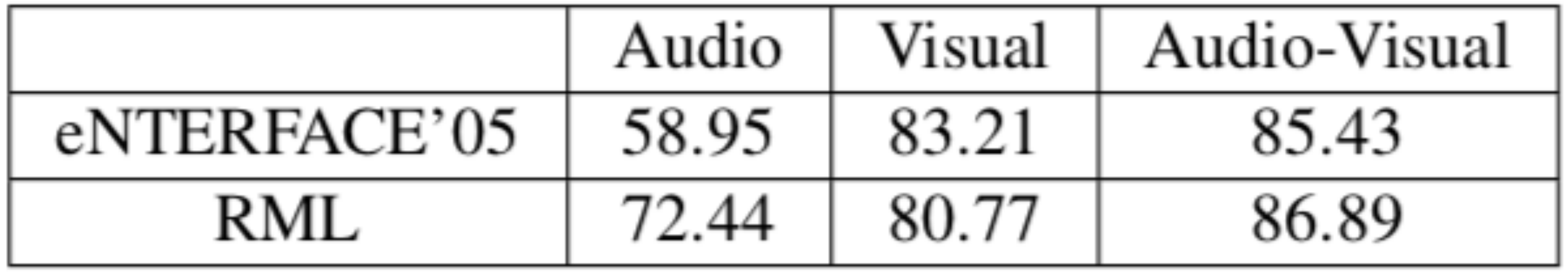
- Public information: maximize correlation between modalities
- Private information: preserving structure of each modality
- Utility: classification using fused features



Evaluation

- tested on two video-audio emotion databases: eNTERFACE’05 and RML

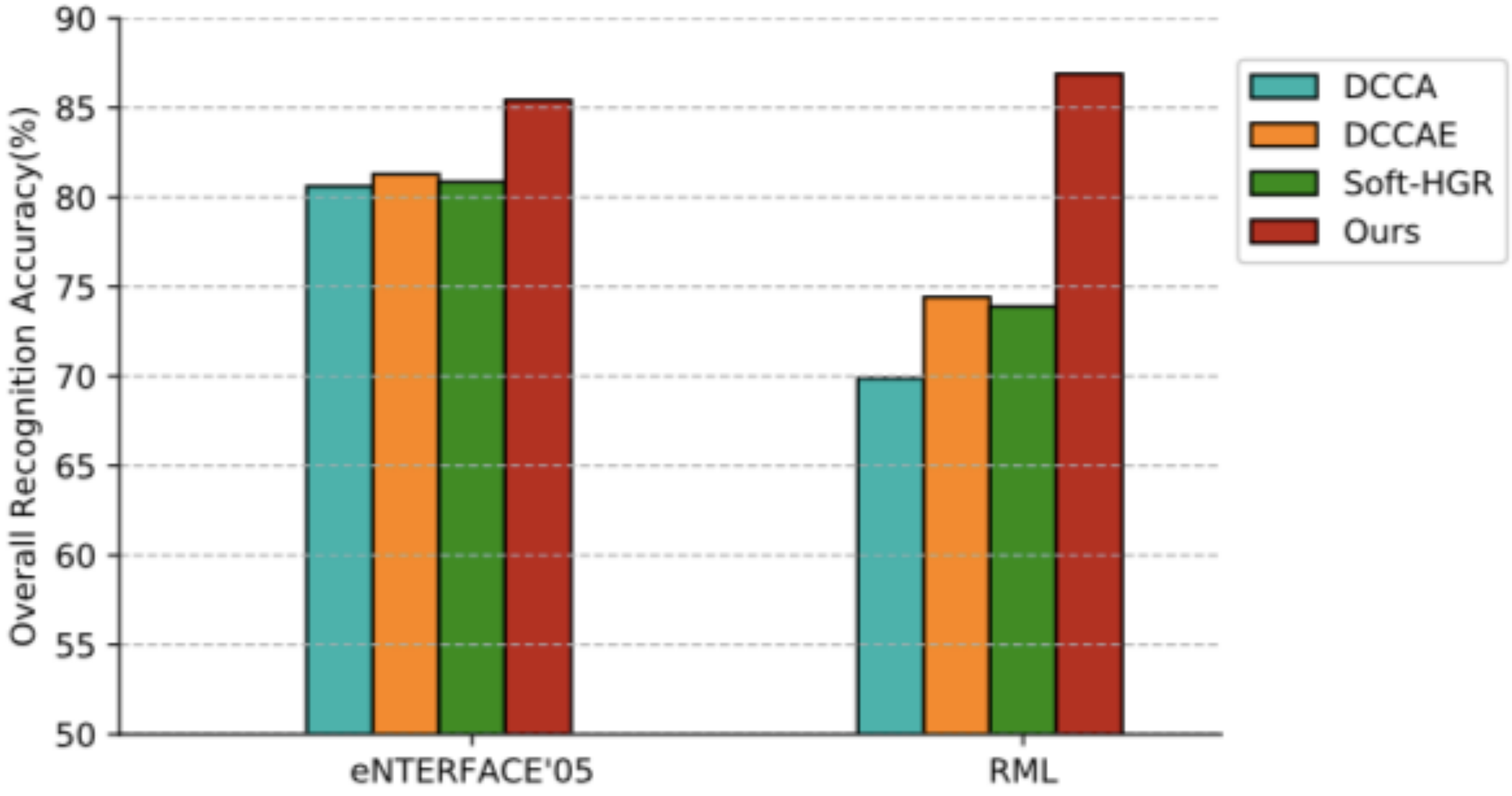
Table 1: Recognition performance of our method.



	Audio	Visual	Audio-Visual
eNTERFACE’05	58.95	83.21	85.43
RML	72.44	80.77	86.89

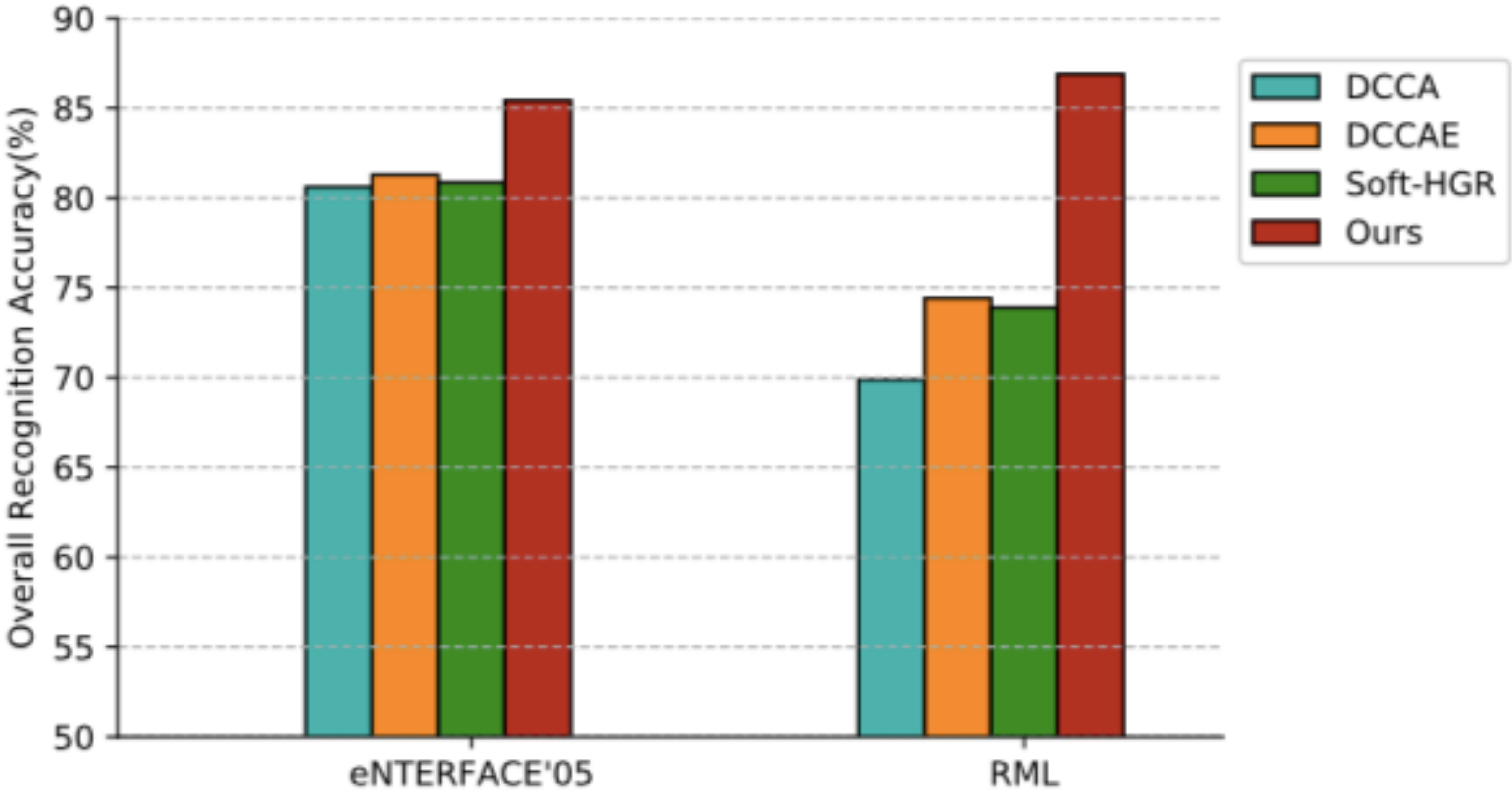
Evaluation

- comparison with CCA-based methods

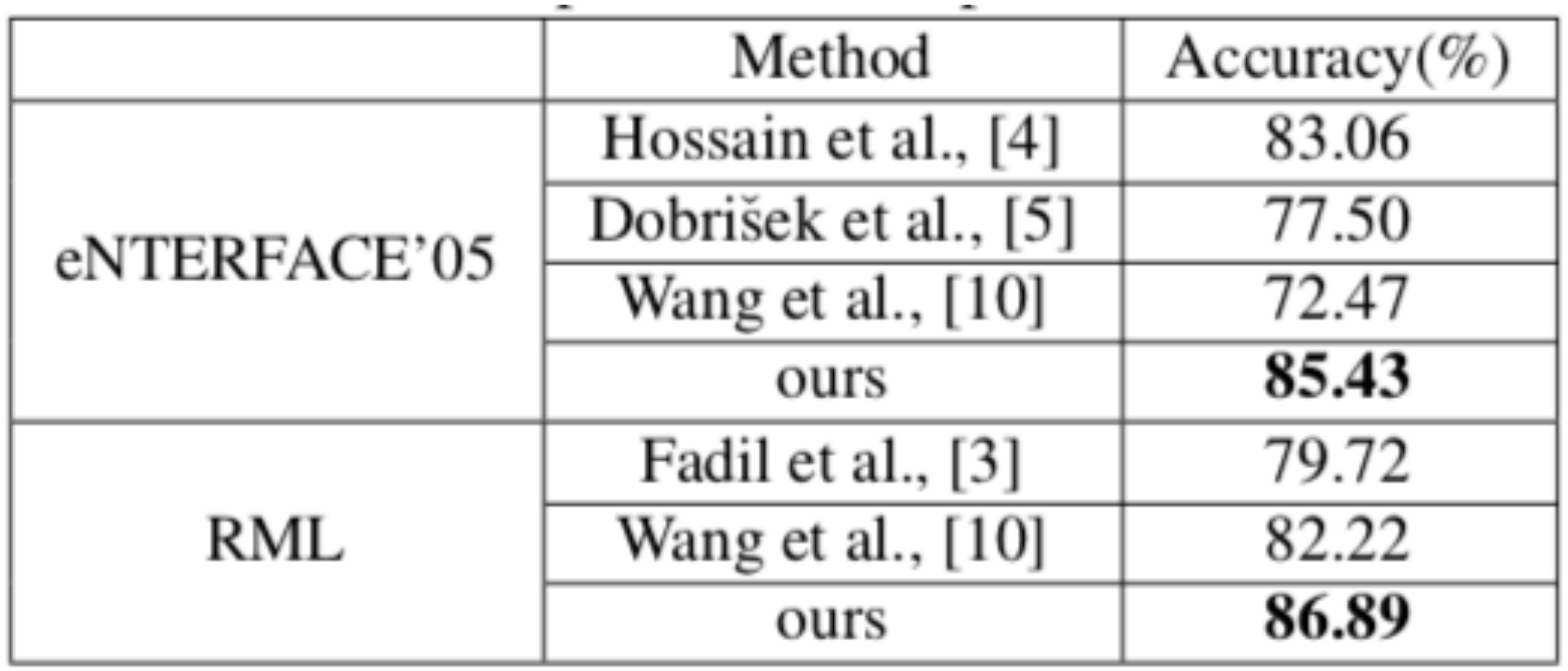


Evaluation

- comparison with CCA-based methods



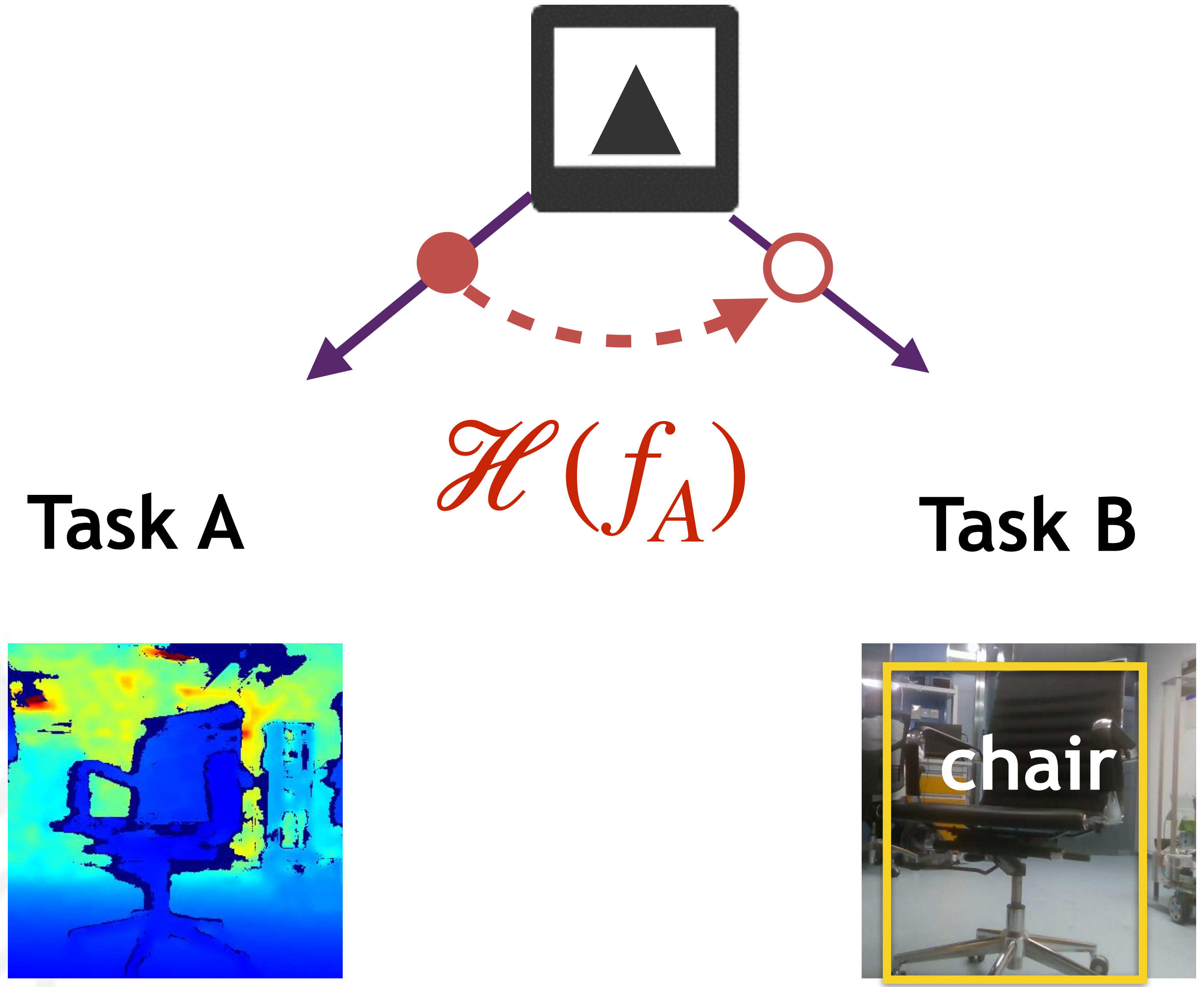
- comparison with existing benchmark results



	Method	Accuracy(%)
eNTERFACE'05	Hossain et al., [4]	83.06
	Dobrišek et al., [5]	77.50
	Wang et al., [10]	72.47
	ours	**85.43**
RML	Fadil et al., [3]	79.72
	Wang et al., [10]	82.22
	ours	**86.89**

Summary

- Estimate task transferability

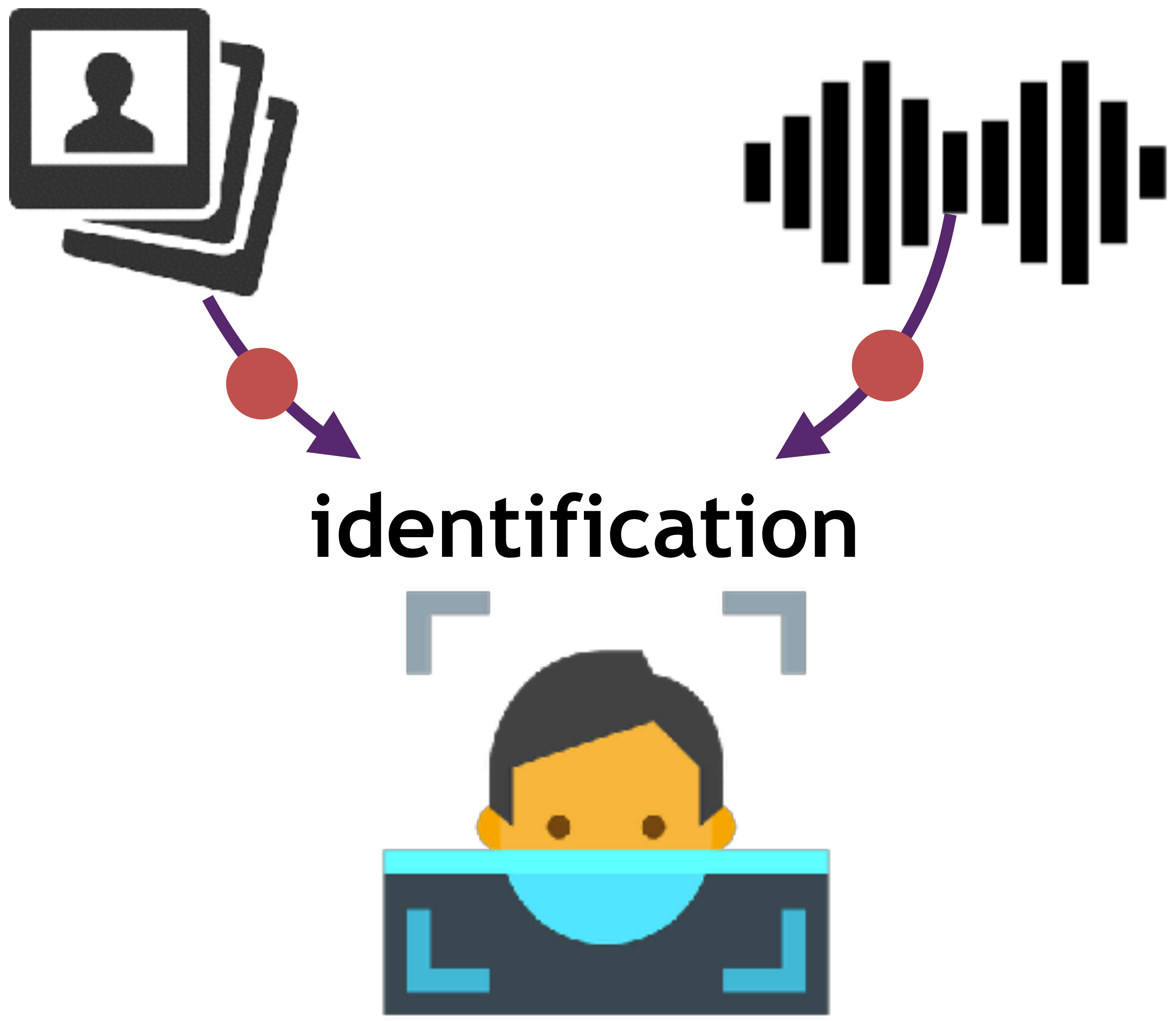


Task A

$\mathcal{H}(f_A)$

Task B

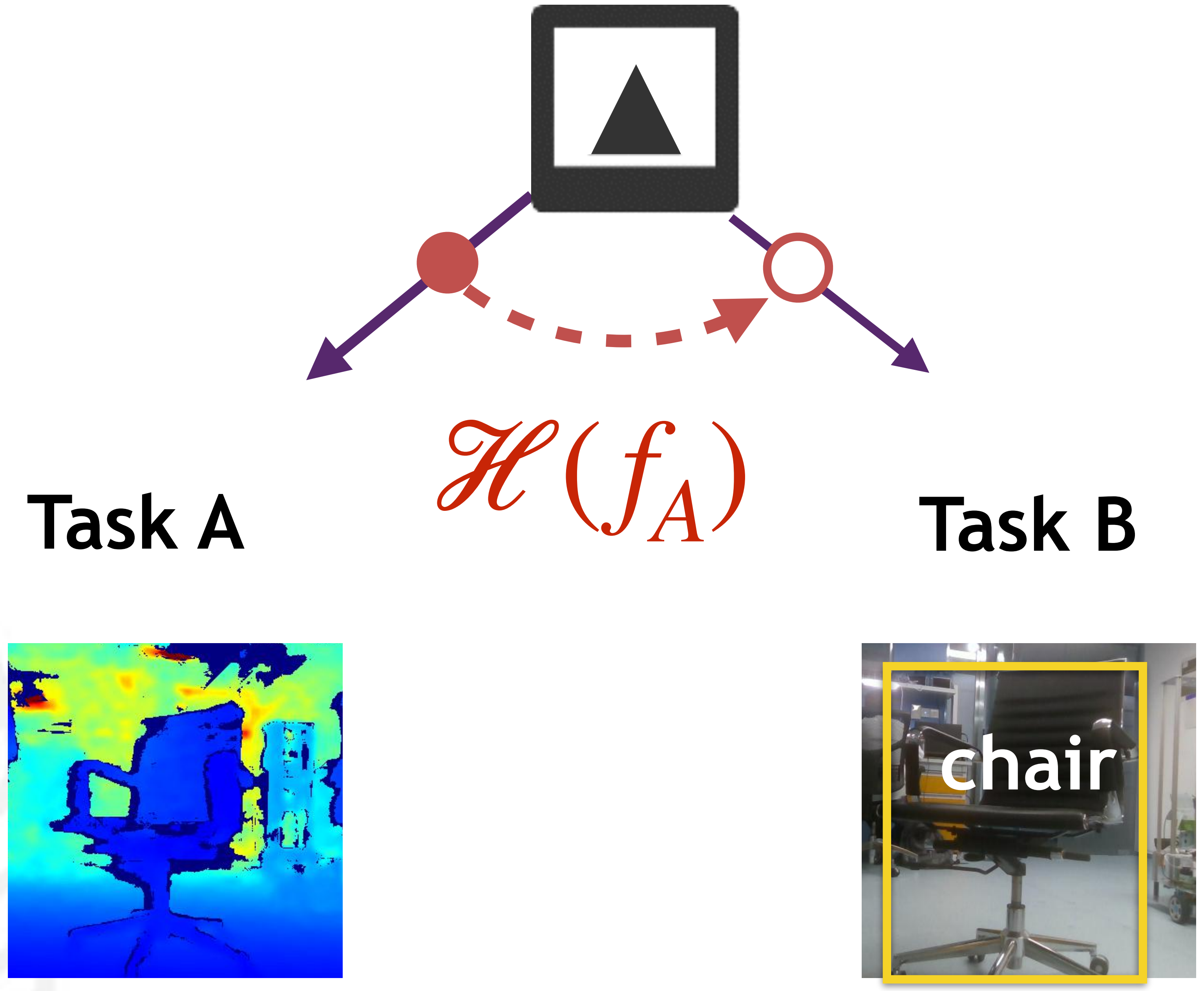
- Multi-view learning



identification

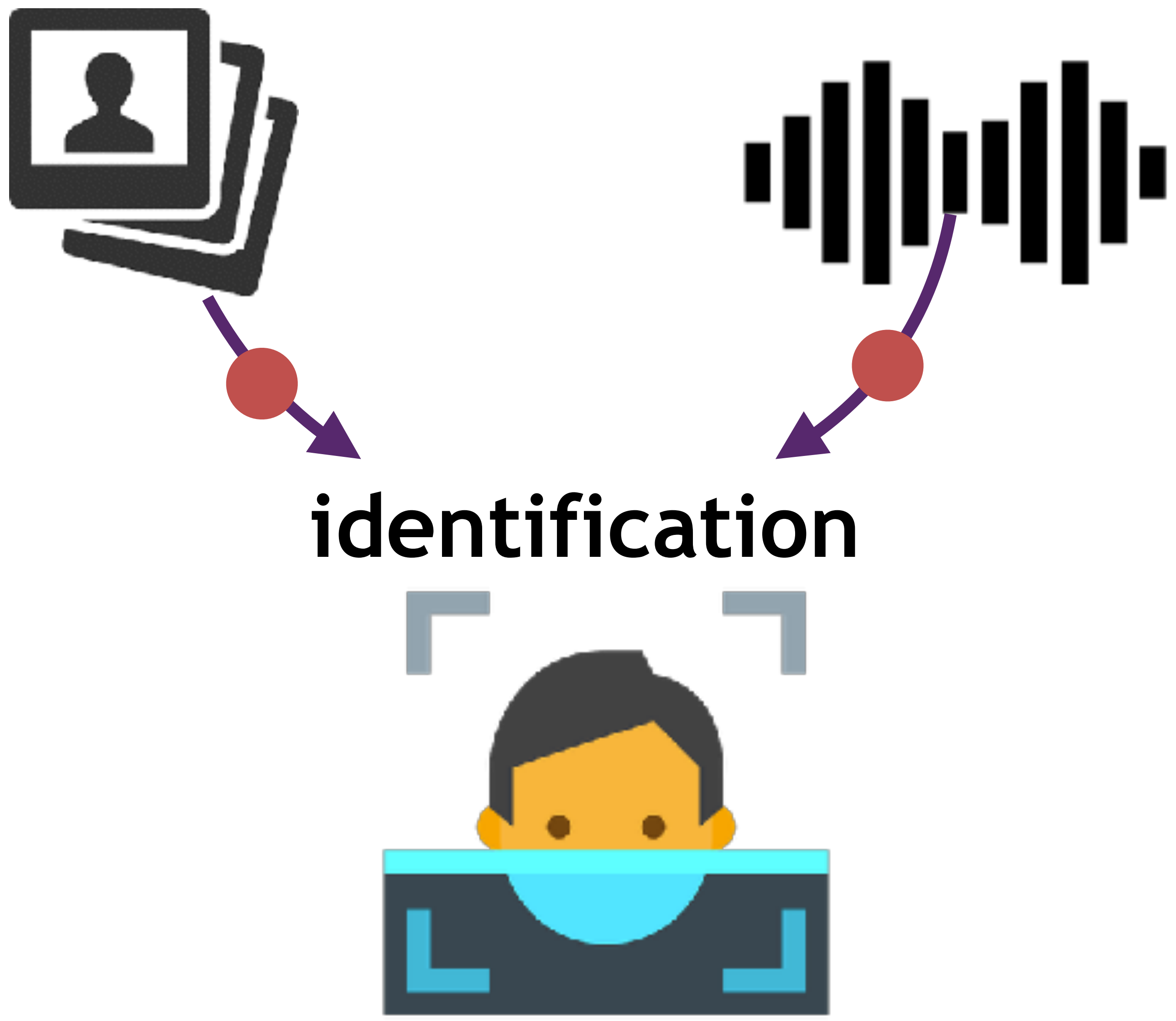
Summary

- Estimate task transferability



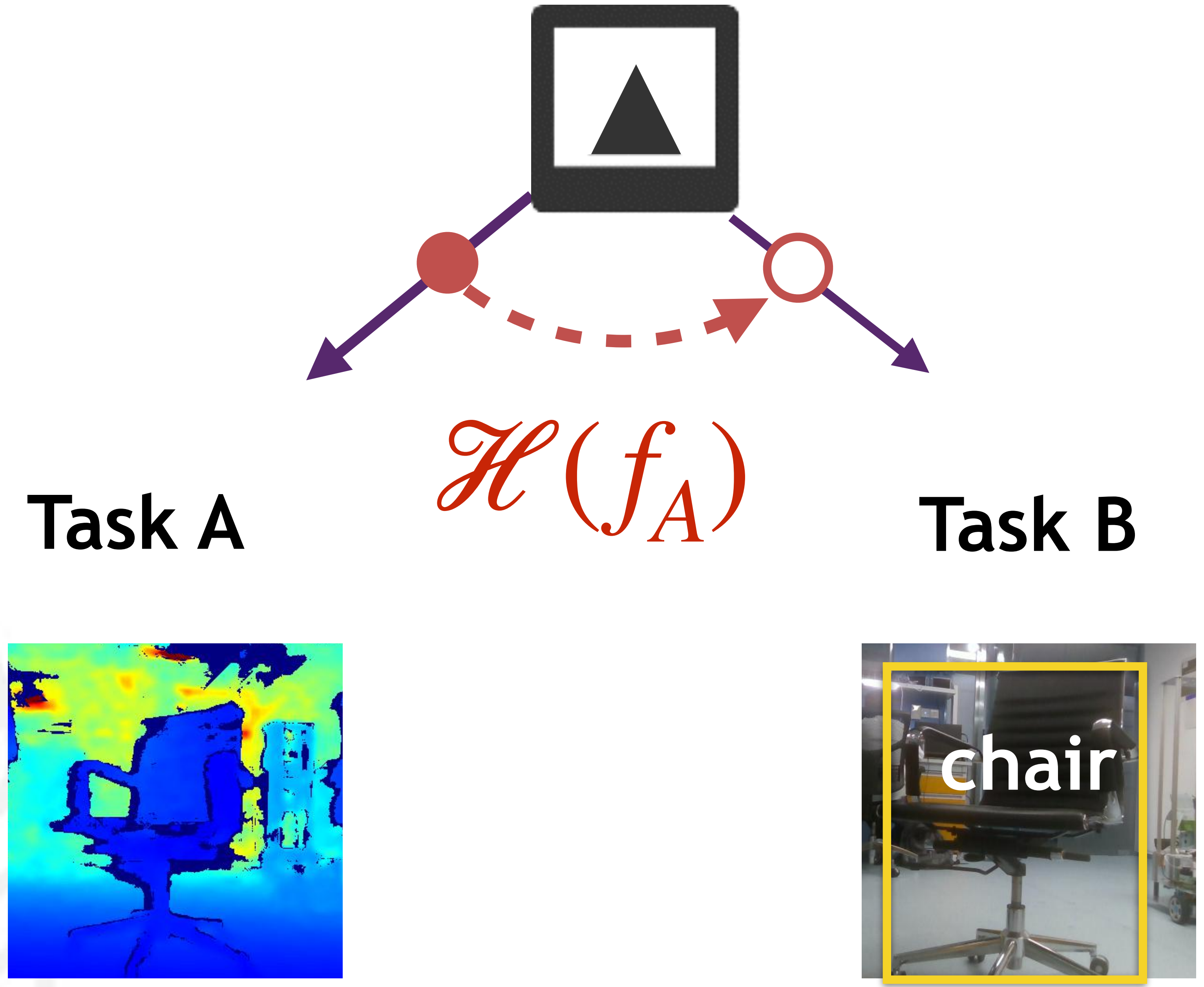
equivalent to HGR maximal correlation with fixed f(X)

- Multi-view learning



Summary

- Estimate task transferability



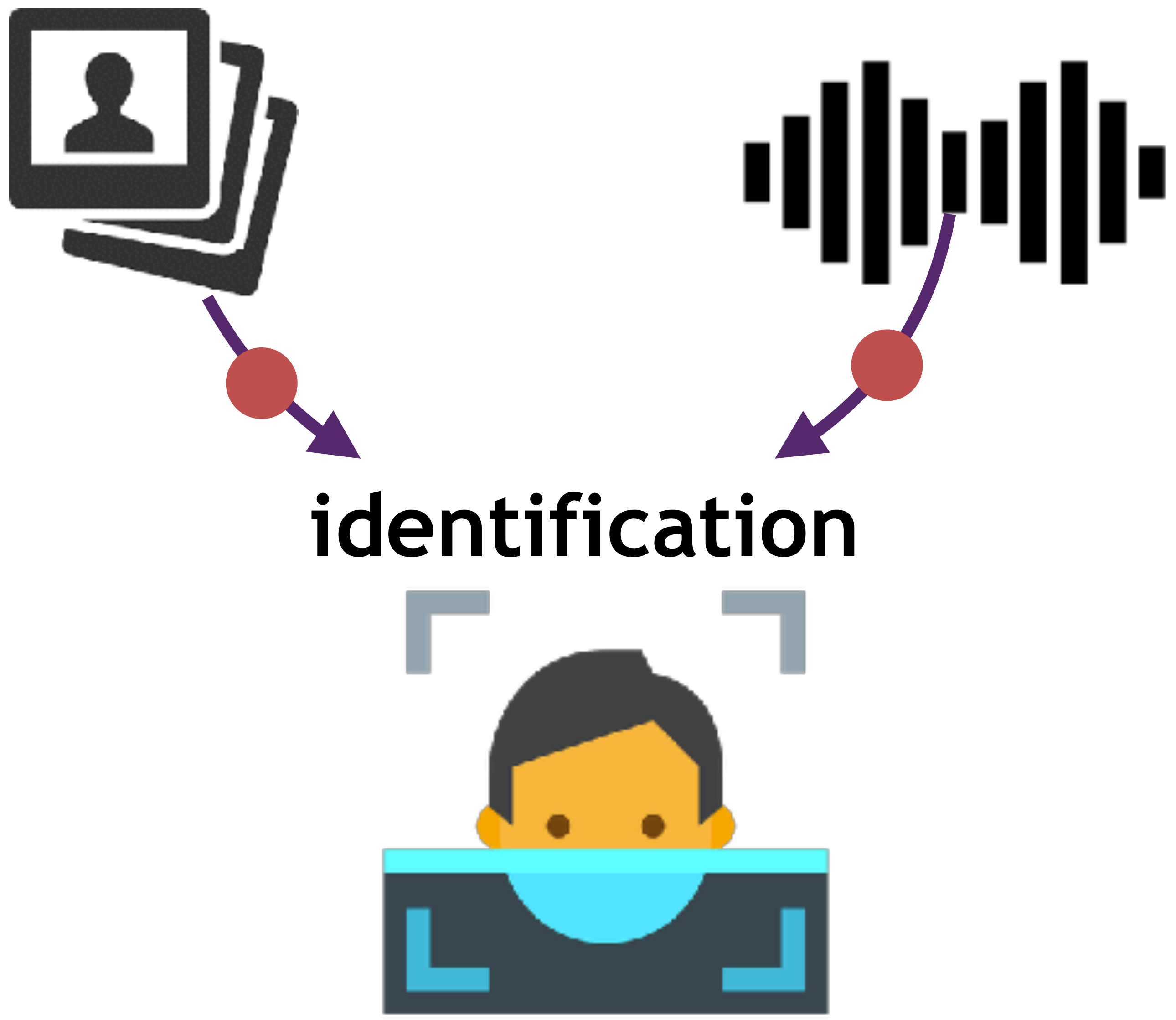
Task A

$\mathscr{H}(f_A)$

Task B

equivalent to HGR maximal correlation with fixed f(X)

- Multi-view learning



identification

learn maximal correlation representations

Conclusion

Conclusion

- Exploiting shared representation between tasks and between multi-view data is important for complex AI applications

Conclusion

- Exploiting shared representation between tasks and between multi-view data is important for complex AI applications

- HGR Maximal correlation is a useful tool to measure and extract shared information

Related Publications

Yajie Bao*, **Yang Li***, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir and Leonidas Guibas, An Information-Theoretic Metric of Transferability for Task Transfer Learning, ICIP 2019 (joint first author)

Jing Lian*, **Yang Li**, Weixi Gu, Shao-Lun, Huang, Lin Zhang, Joint mobility pattern mining with urban region partitions, Mobiquitous 2018 (Best Paper)

Fei Ma*, Wei Zhang, **Yang Li**, Shao-Lun Huang, and Lir Zhang, An end-to-end Learning Approach for Multimodal Emotion Recognition: Extracting Common and Private Information, ICME 2019

Thank you!

Acknowledgement

My awesome collaborators:

- Prof. Shao-Lun Huang
- Prof. Leonidas Guibas
- Prof. Lizhong Zheng
- Prof. Lin Zhang

- Yajie Bao
- Changjin Liu
- Jing Lian
- Lu Li
- Fei Ma
- Xiangxiang Xu

TBSI 清华-伯克利深圳学院 Tsinghua-Berkeley Shenzhen Institute

Massachusetts Institute of Technology

Stanford University