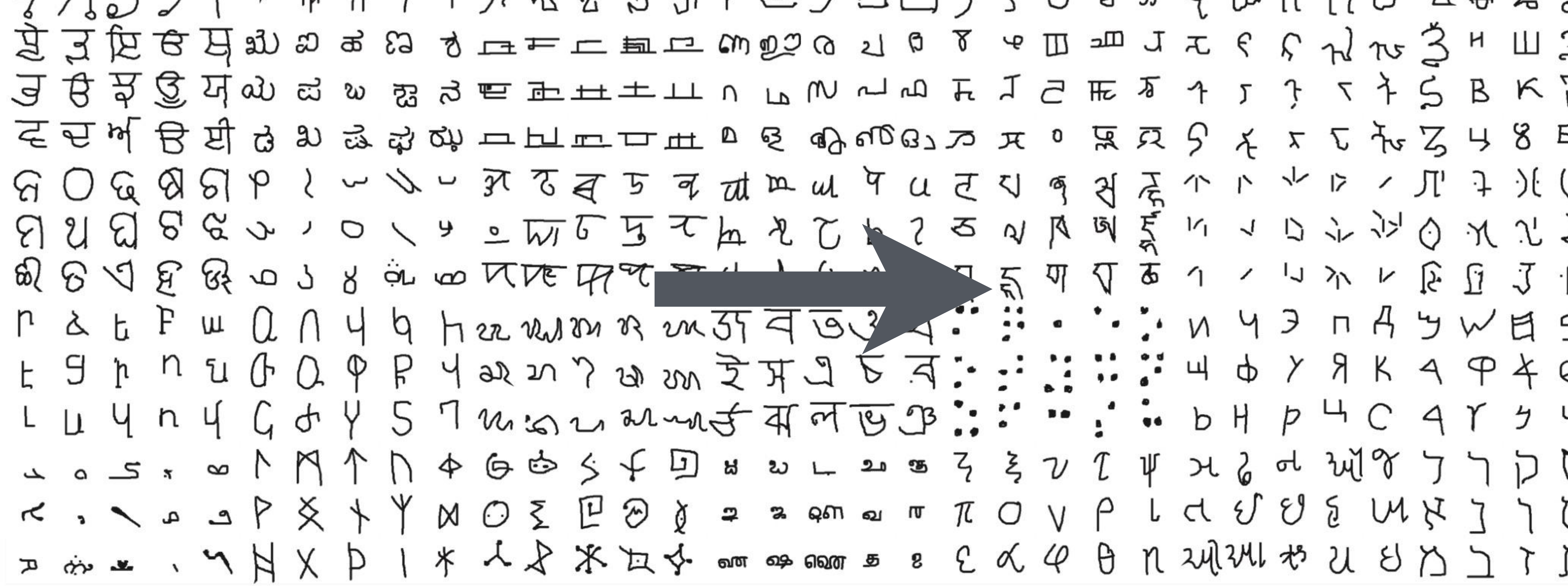
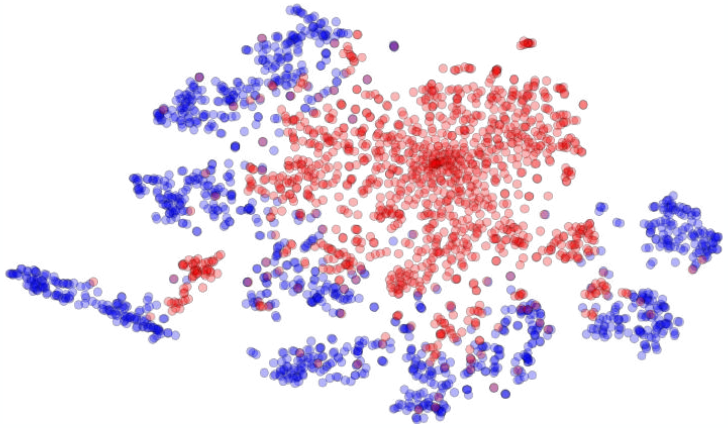


A Tutorial on Transfer Learning

Yang Li
2019/11/26

Today's Talk

- What's Transfer Learning
- Transfer Learning Techniques
  - Task transfer learning
  - Domain adaptation
  - Transfer bound on domain adaptation



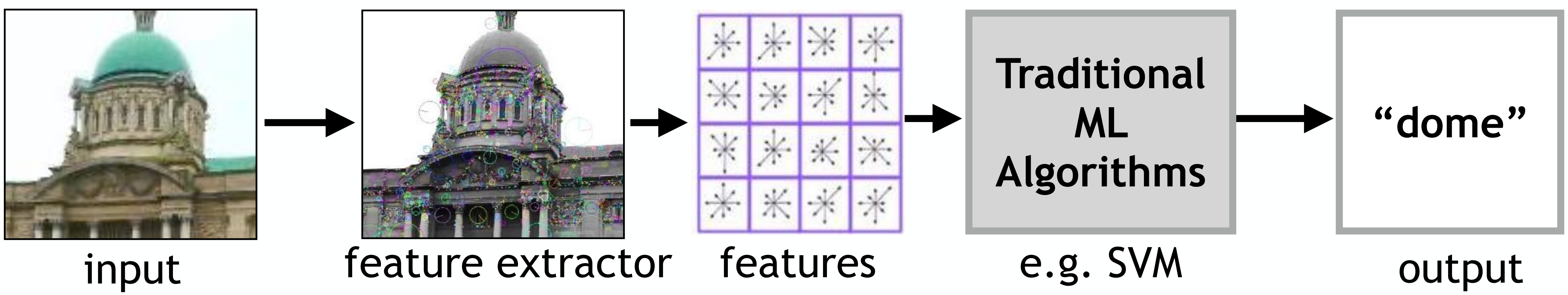
- How to avoid negative transfer?
  - Case study on feature transferability
  - Task transferability: empirical and theoretical methods
- Discussions and Q&A

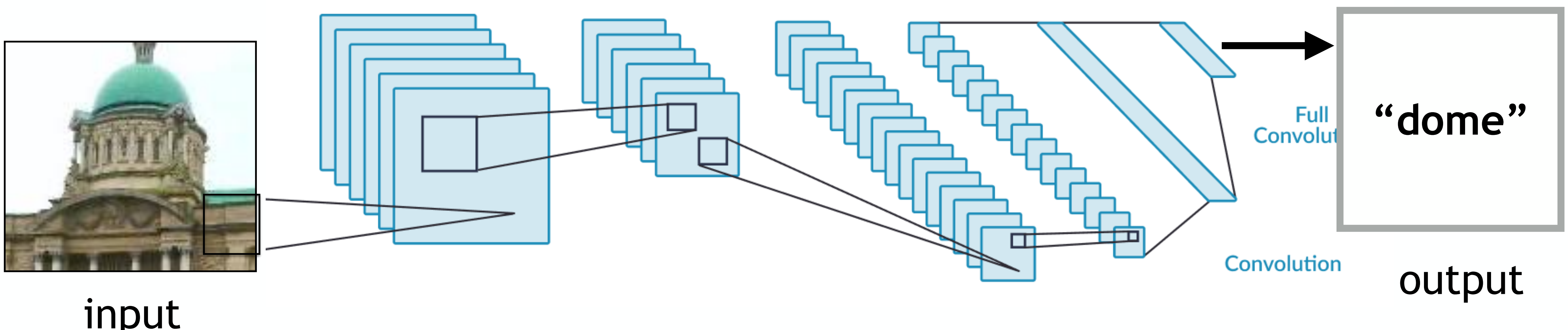
Single-Task Machine Learning

Designed for solving a single task, trained from scratch

Example: image-based recognition task

Traditional machine learning flow





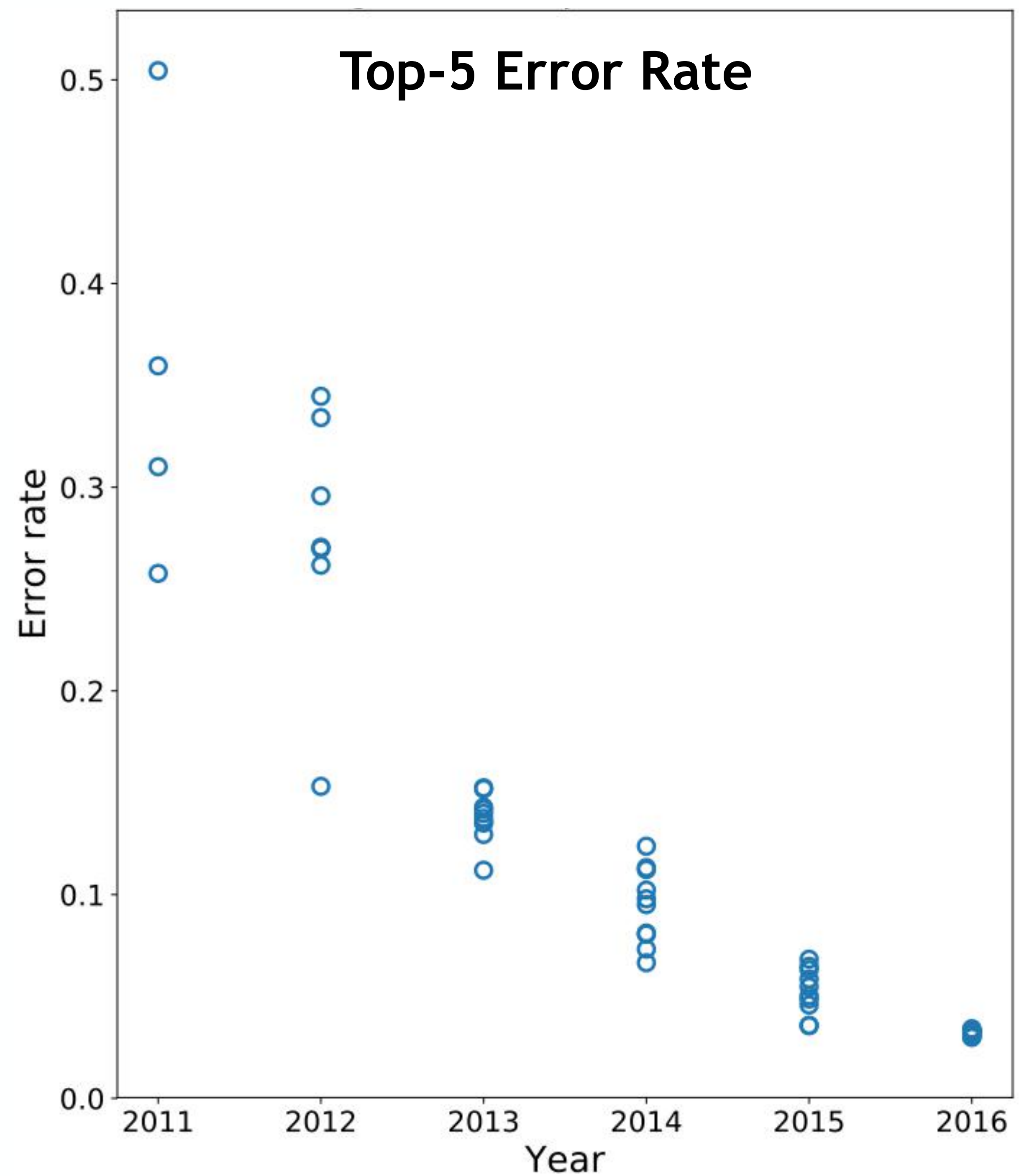
End-to-end learning with deep neural netes

Single-Task Machine Learning

- ImageNet competition results over the years

IMAGE Net (2009)

1,034,908
labeled images



Single-Task Machine Learning

Most models are large and costly to train!

ImageNet classification models

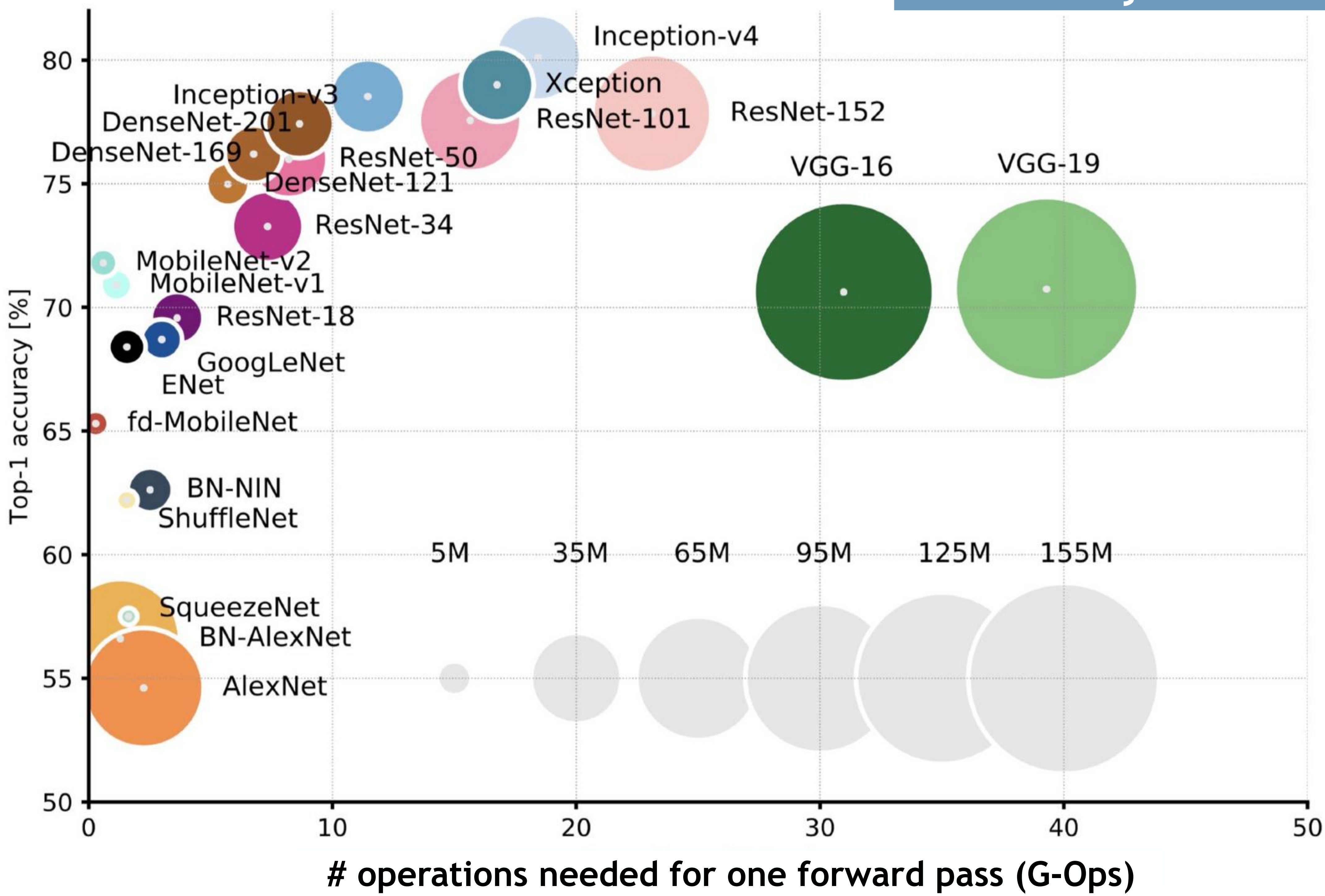
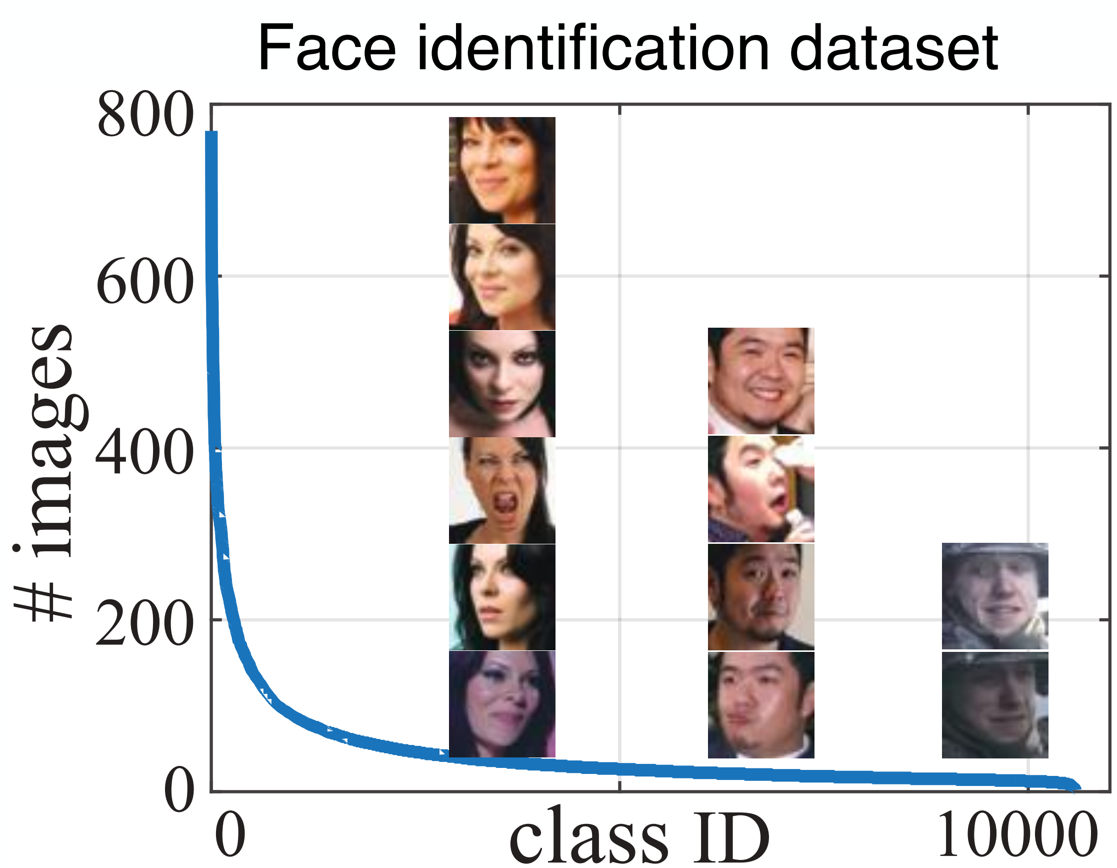
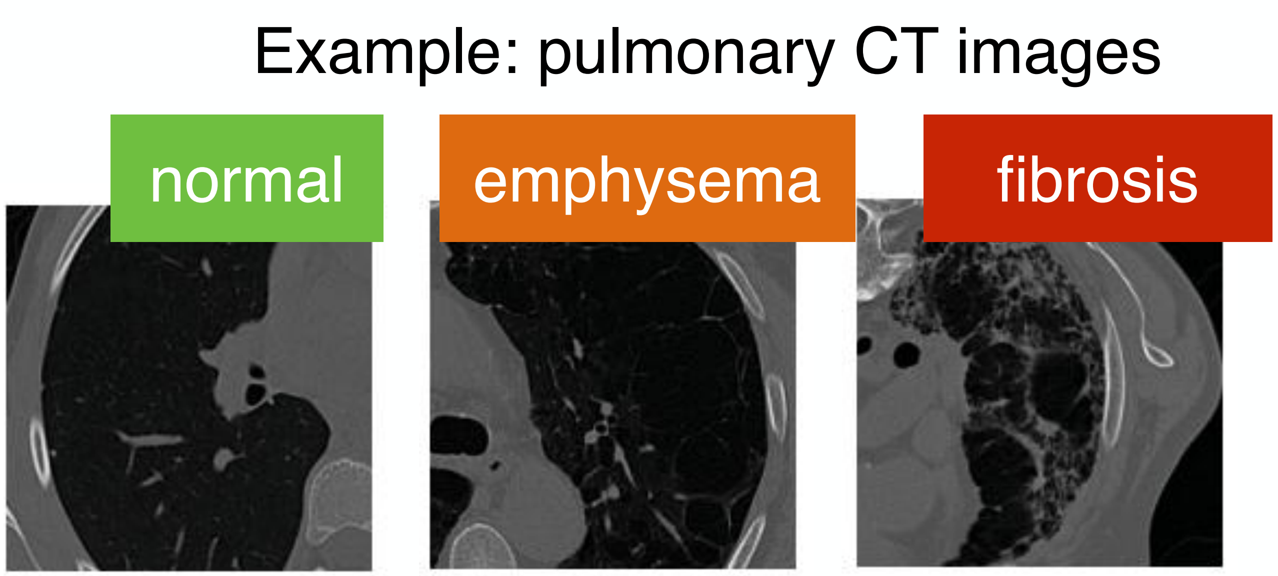


image credit: https://arxiv.org/abs/1605.07678

Issues with Single-Task Learning

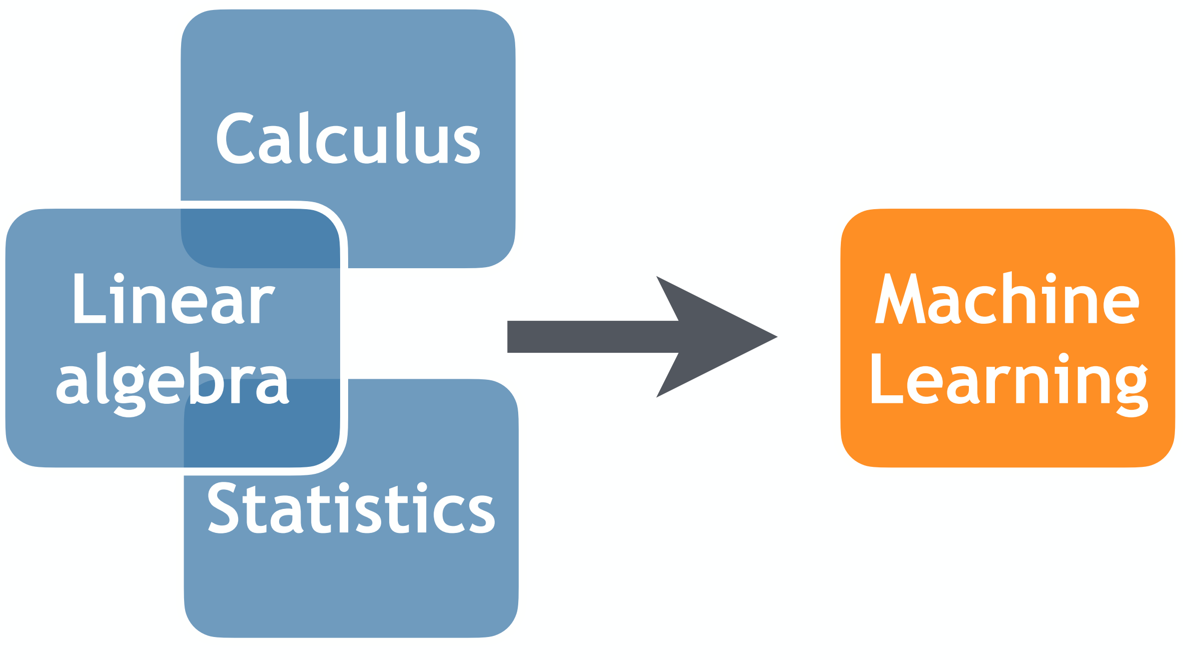
- **Learn something new quickly**: can't train from scratch every time
- Most classes/tasks have very few data samples
- Training labels may be expensive to obtain





Transfer learning

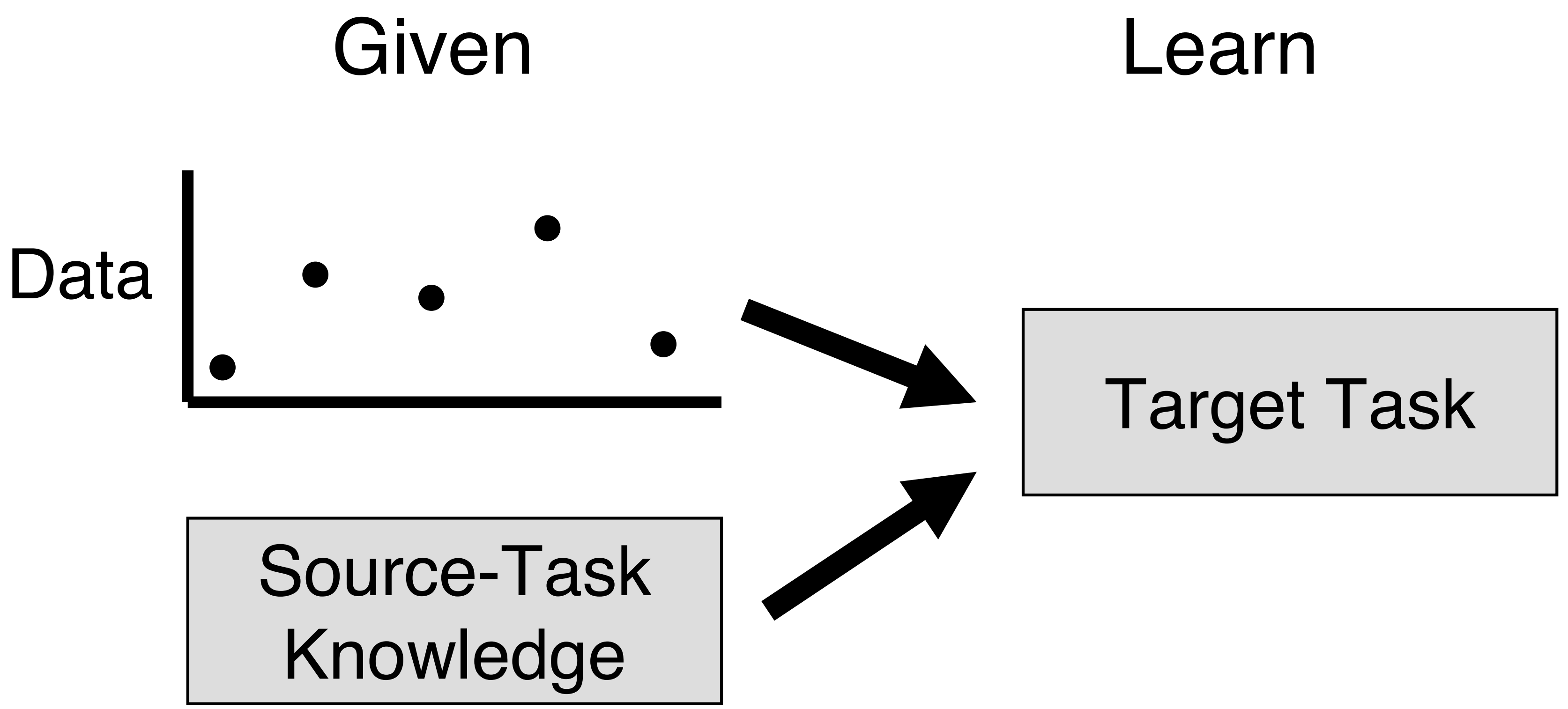
- Human learners can inherently transfer knowledge between tasks



How can machines recognize and apply relavent knowledge from previous learning experience?

Transfer Learning at 1000 feet

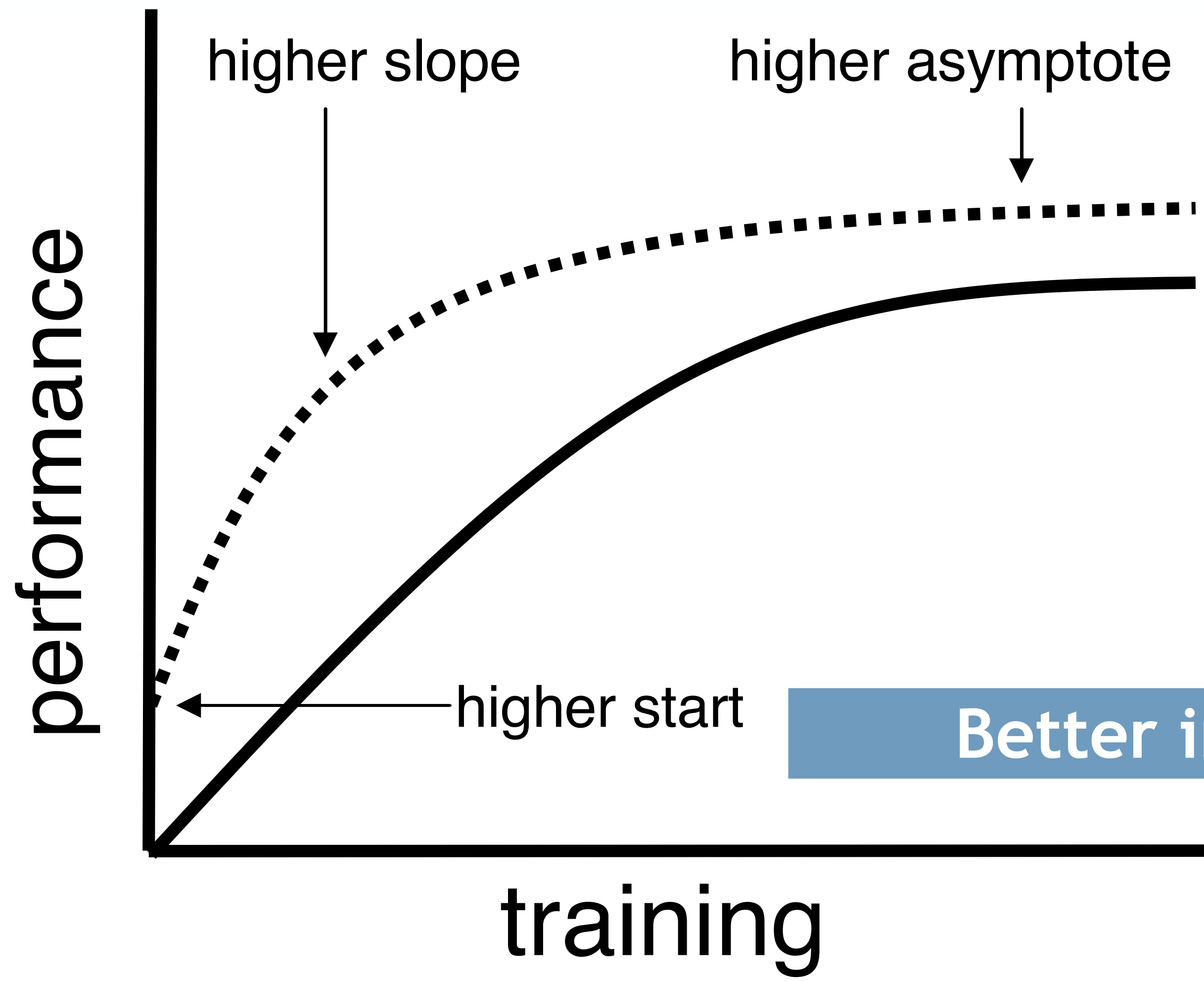
- Transfer knowledge from one or more source domains/tasks to a target domain/task.



How transfer might improve target learnring

Less time to fully learn the target

Better final (asymptotic) performance



with transfer
without transfer

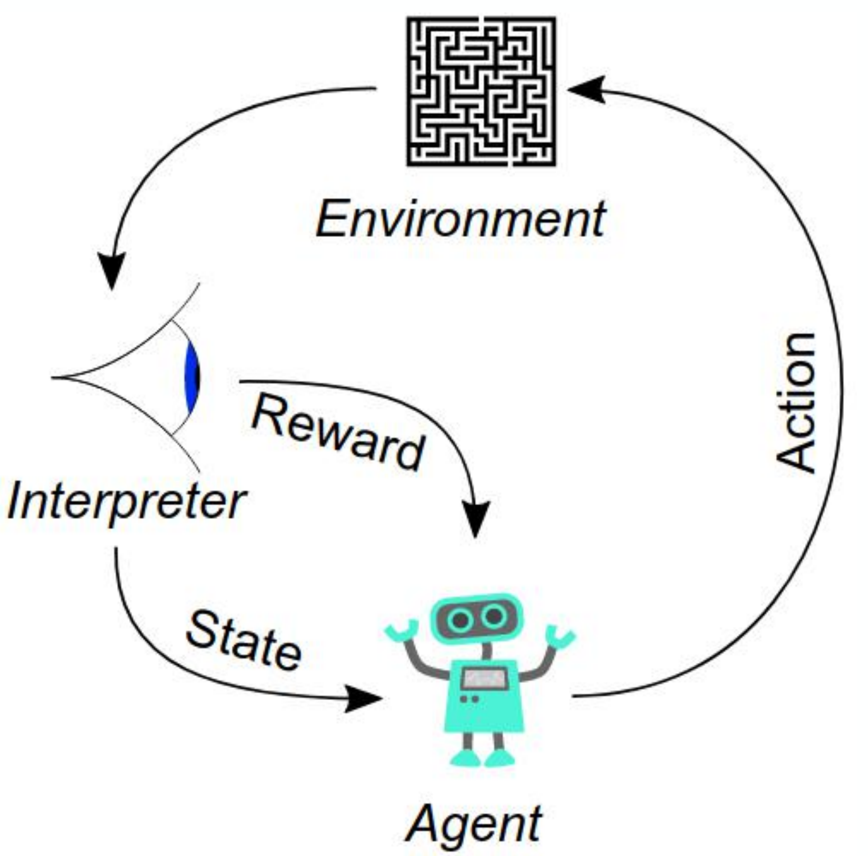
Better initial performance

Transfering might reduce target learning performance (negative transfer)

Two Branches of Transfer Learning Paradigms

Inductive Learning: Learn decision function f from training data, test on unseen data

Reinforcement Learning: sequential decision making problems



Inductive Transfer Learning Examples

- Domain-specific computer vision tasks
- Common to transfer pre-trained features from ImageNet

ImageNet 1000-class classification task

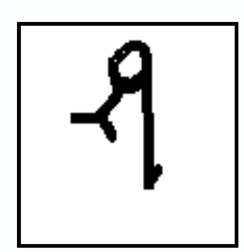
Structural Damage Detection

Yuqing Zhao et. al. Deep Transfer Learning for Image-Based Structural Damage Recognition

Learning with Small Samples: K-Shot Learning

- When the training set of a task only has k samples
- e.g. one-shot alphabet classification:

Which letter ?

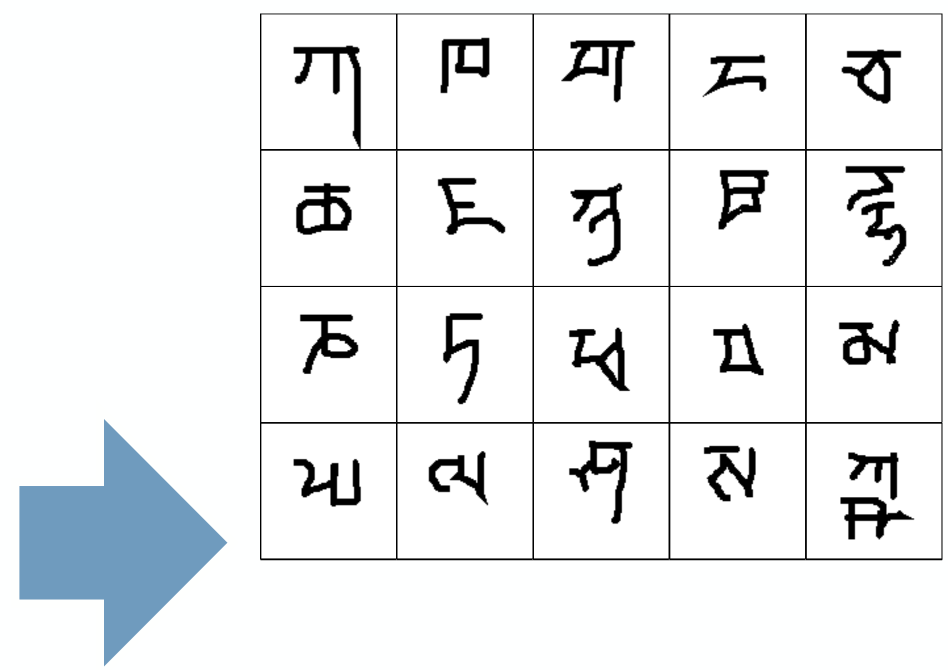


OMIGLOT dataset

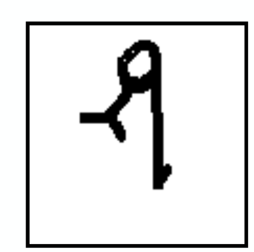
K-Shot Learning

- **Transfer latent knowledge of handwritten characters from other tasks**

50 classification tasks in different alphabets



Which letter ?



K-Shot Learning

- One-shot person re-identification from video

VIPeR PRID2011 CUHK01

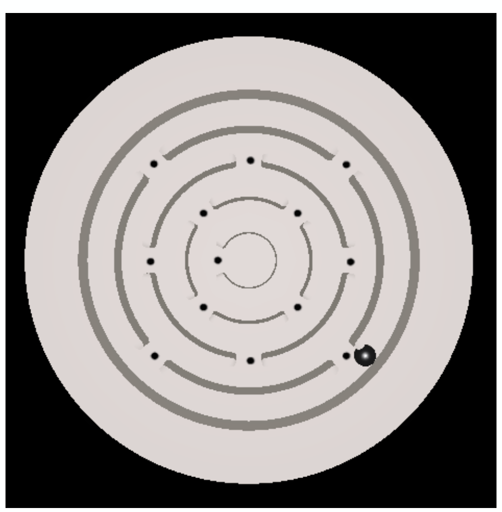
who is this person?

Key Idea: Transfer knowledge from multiple domains (datasets)

Bak et. al. (2017) One-Shot Metric Learning for Person Re-identification

Reinforcement Transfer Learning Examples

- **Reinforcement learning for robotic control, e.g**
  - SIMtoReal：transfer knowledge from simulated robot to physical robot



Simulated marble maze game

Real maze on robotic arm

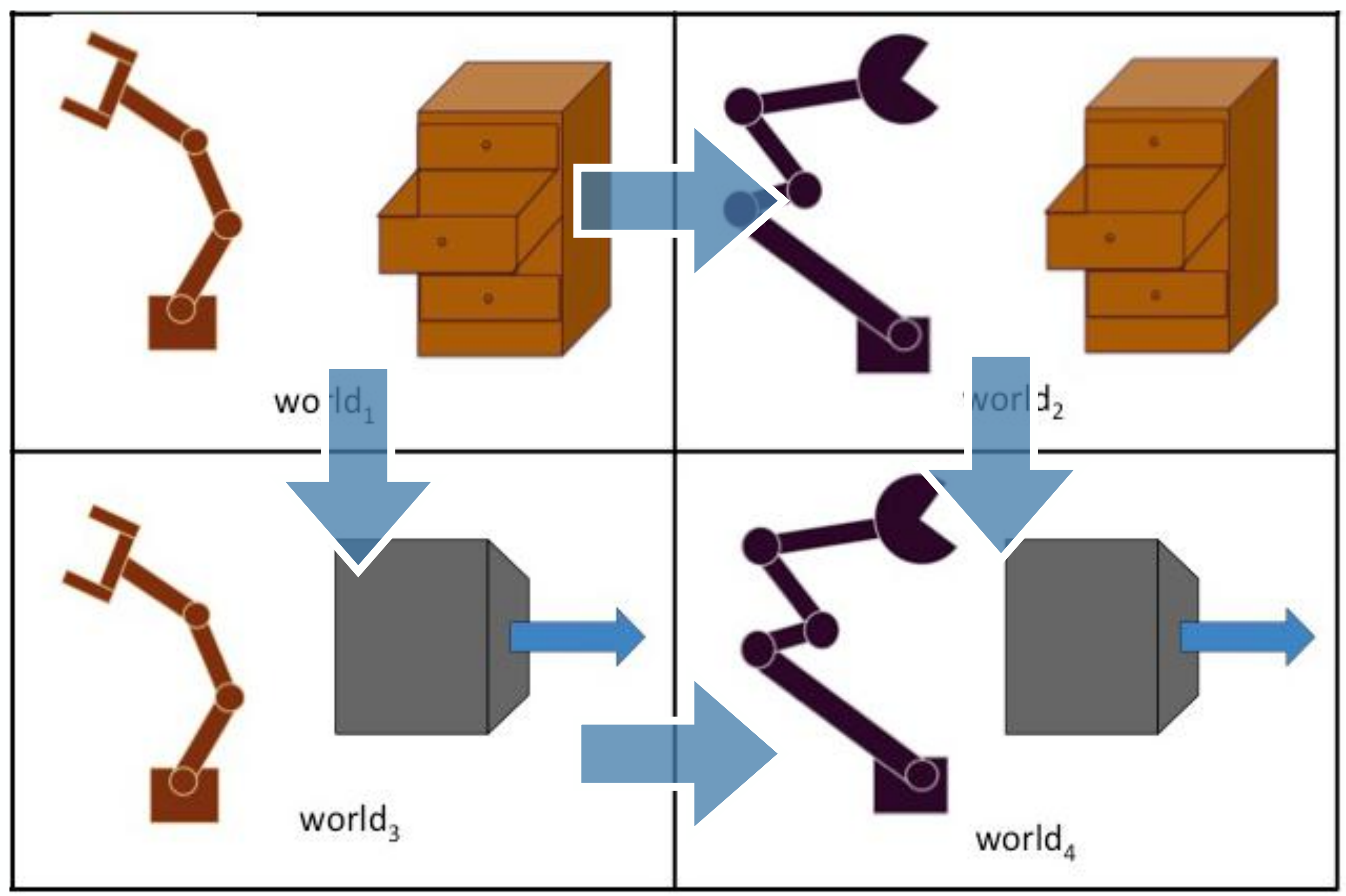
Applications of Transfer Learning

- **Reinforcement learning for robotic control, e.g**
  - Transfer between robots and between tasks

Robot 1 (3-link) **Robot 2 (4-link)**

Open drawer

Push box

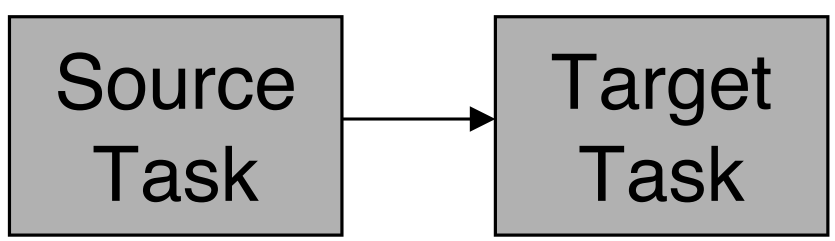


Devin (2016) Learning Modular Neural Network Policies for Multi-Task Multi-Robot Transfer

Transfer Learning vs Multi-Task Learning

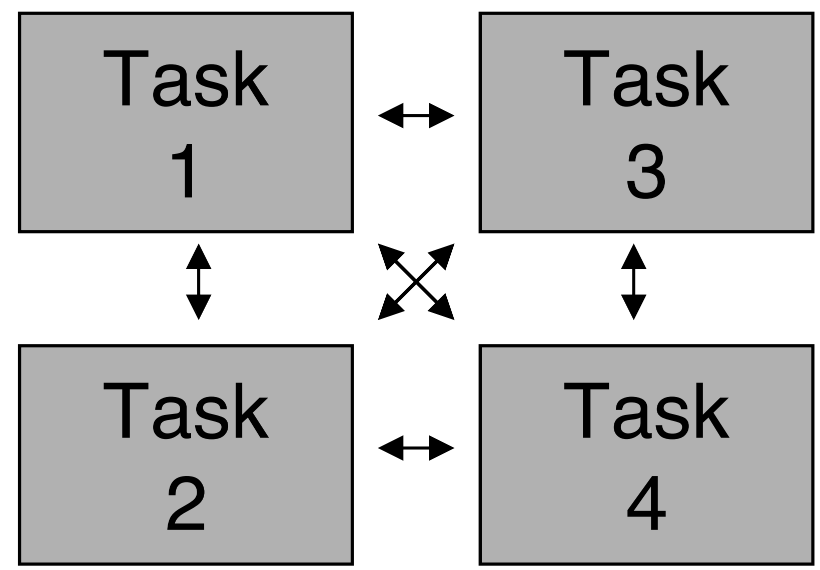
TL is more likely to encounter in real world than MTL

Transfer Learning



Sequential learning:
focus on target task

Multi-task Learning



Joint learning:
focus on all tasks

Today’s Talk

- What’s Transfer Learning
- Transfer Learning Techniques
  - Task transfer learning
  - Domain adaptation
  - Transfer bound on domain adaptation
- How to avoid negative transfer?
  - Case study on feature transferability
  - Task transferability: empirical and theoretical methods
- Discussions and Q&A

Transfer Learning Definition

Terminologies

input features

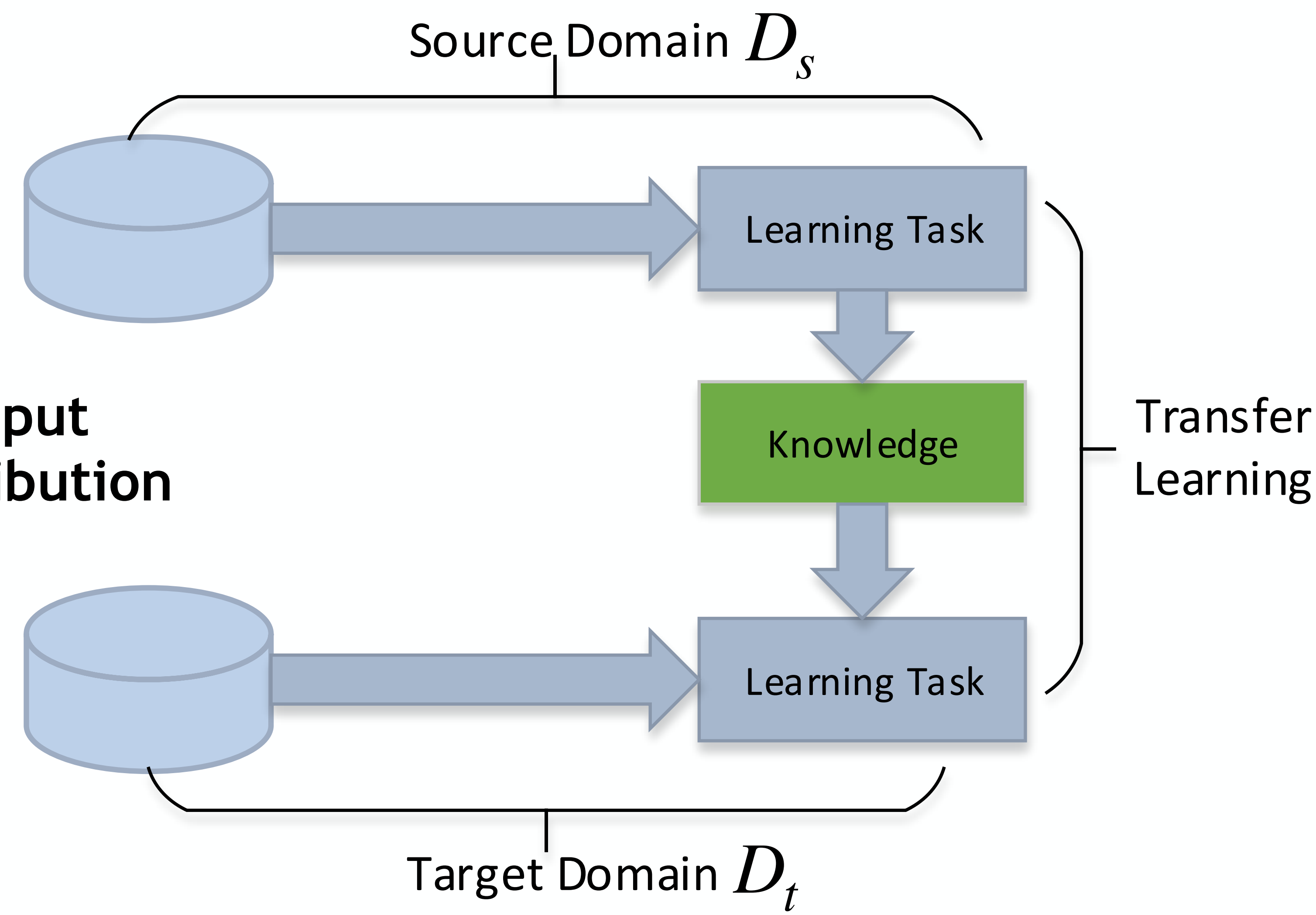
- Domain: $D = \{X, P_X\}$

input distribution

- Task: $T = \{Y, f\}$

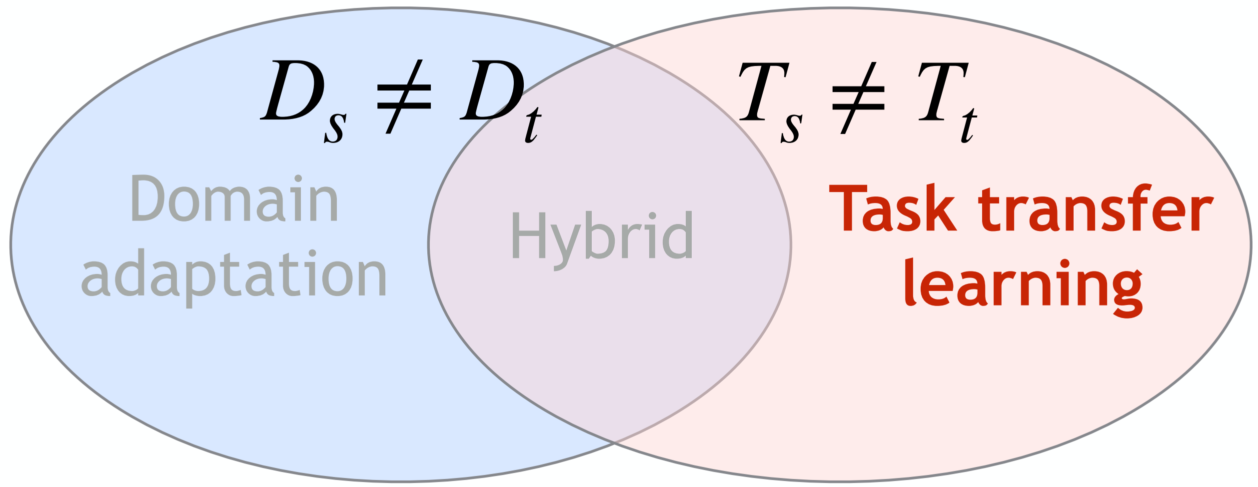
labels

predictive function $P_{Y|X}$



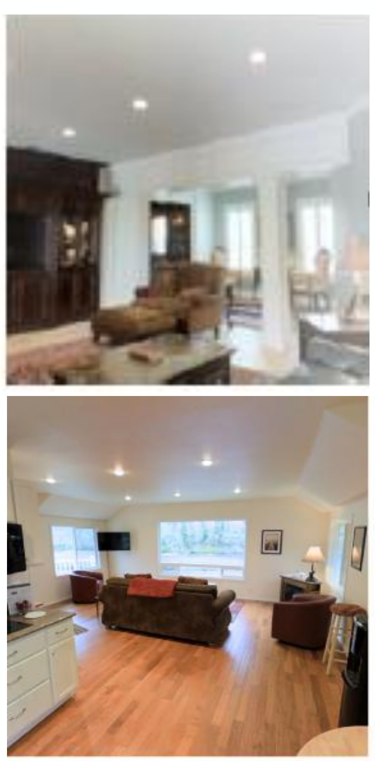
Transfer learning: improve the performance of predictive function f_t for T_t by discover and transfer latent knowledge from (D_s, T_s) , where $D_s \neq D_t$ and/or $T_s \neq T_t$

Transfer Learning



Task Transfer Learning: adapt source hypothesis or feature to target task

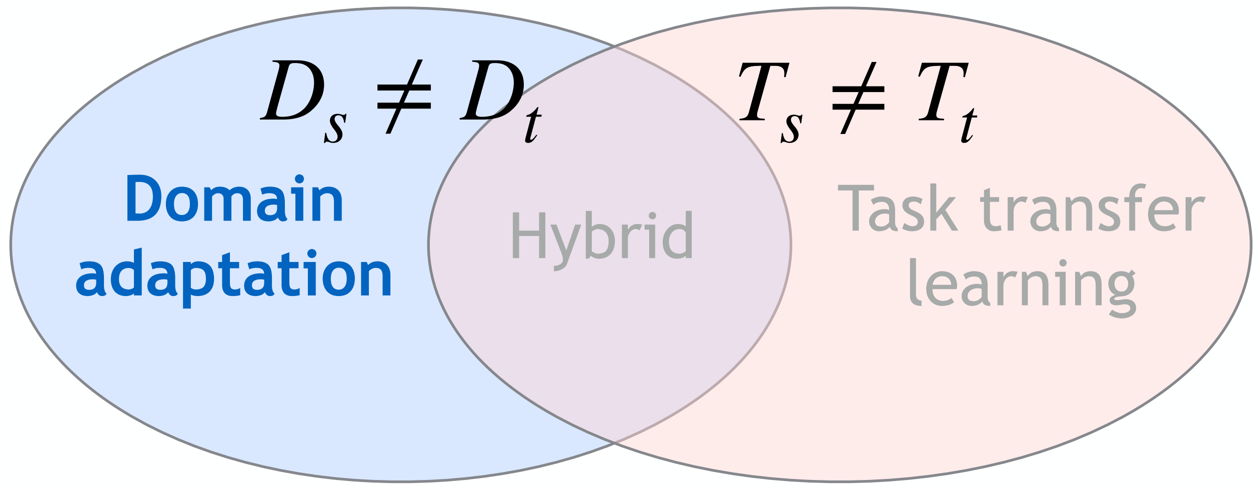
D_s/D_t: Indoor Scene



T_s: scene classification → living room

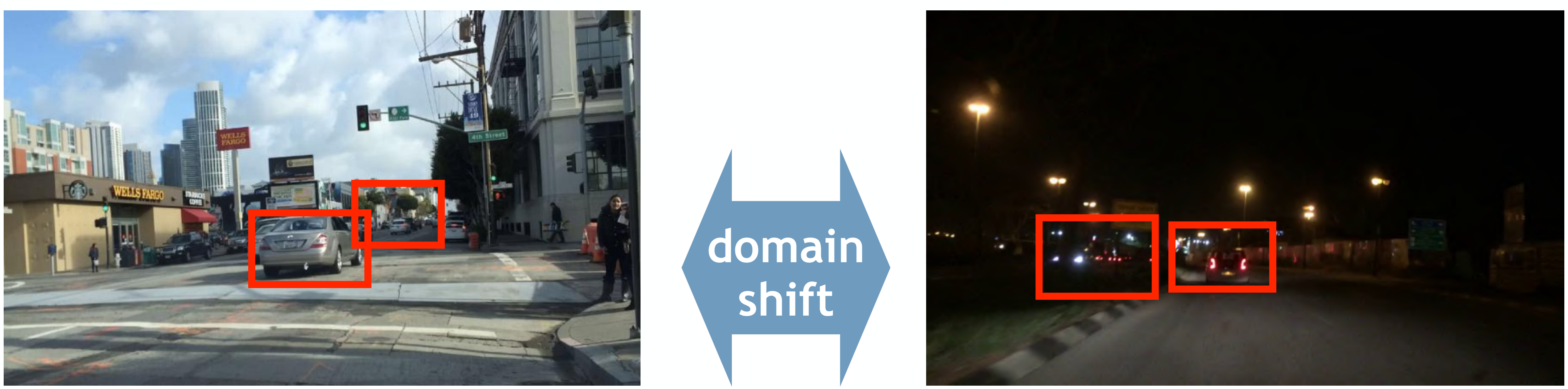
T_t: object detection → sofa, table, lamp, …

Transfer Learning



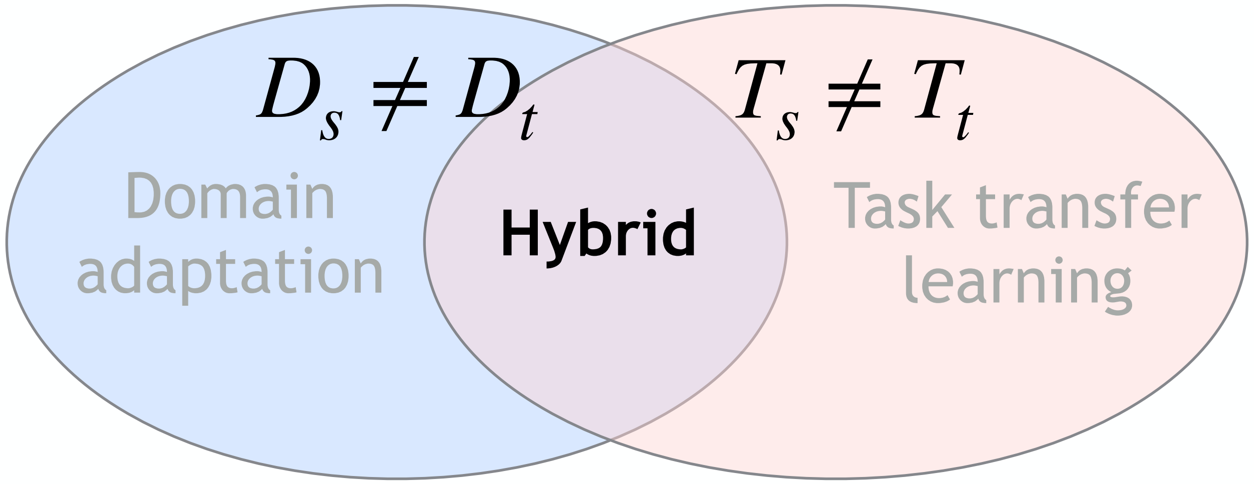
Domain adaptation: Learn domain agnostic representations

T_s/T_t: Vehicle Detection



D_s (day) **D_t (night)**

Transfer Learning



Task Transfer Learning: adapt source hypothesis or feature to target task

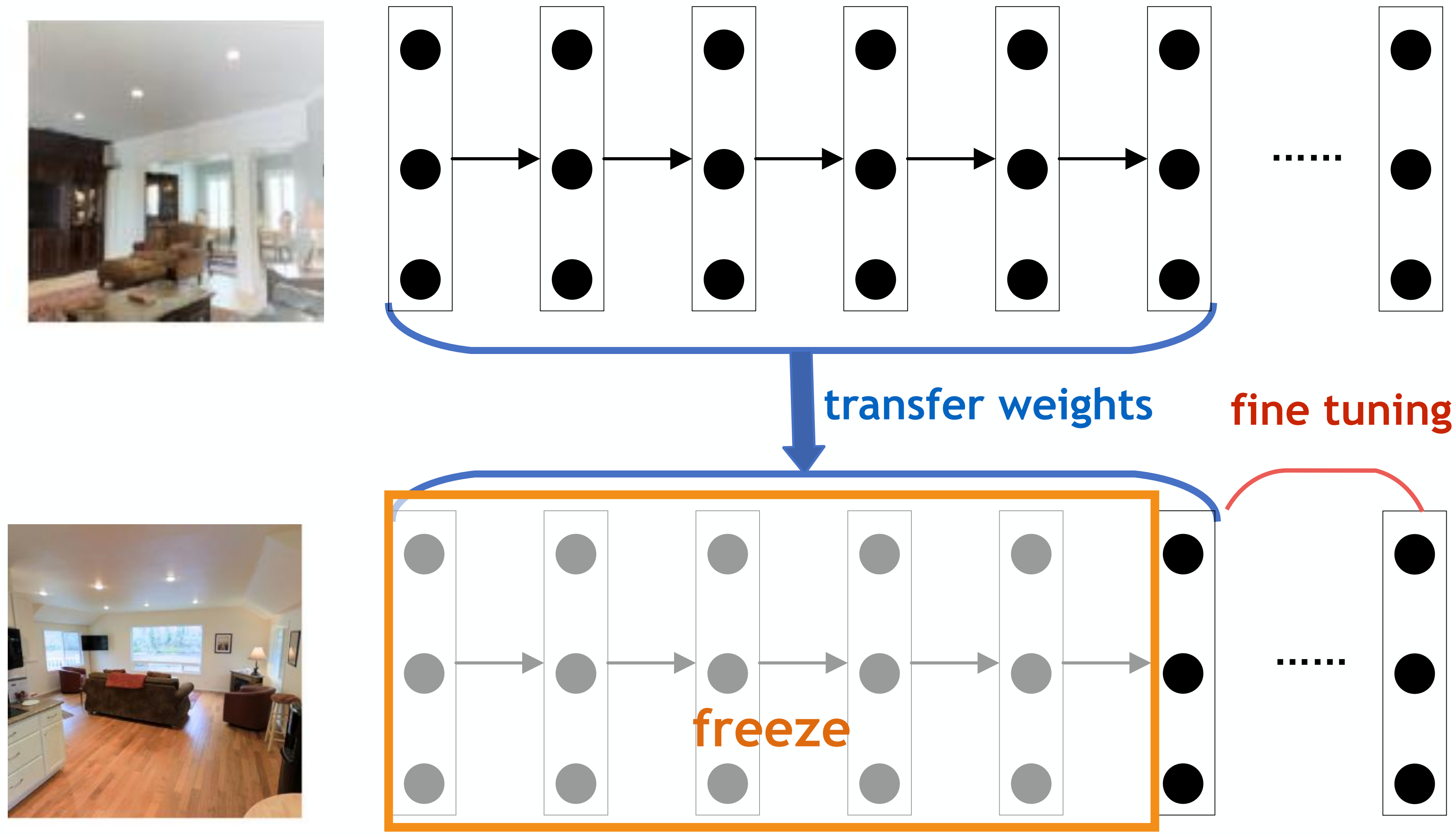
Domain adaptation: Learn domain agnostic representations

Most transfer learning problems in practice are hybrid!

Task Transfer Learning

- Pretrained Model + Fine Tuning

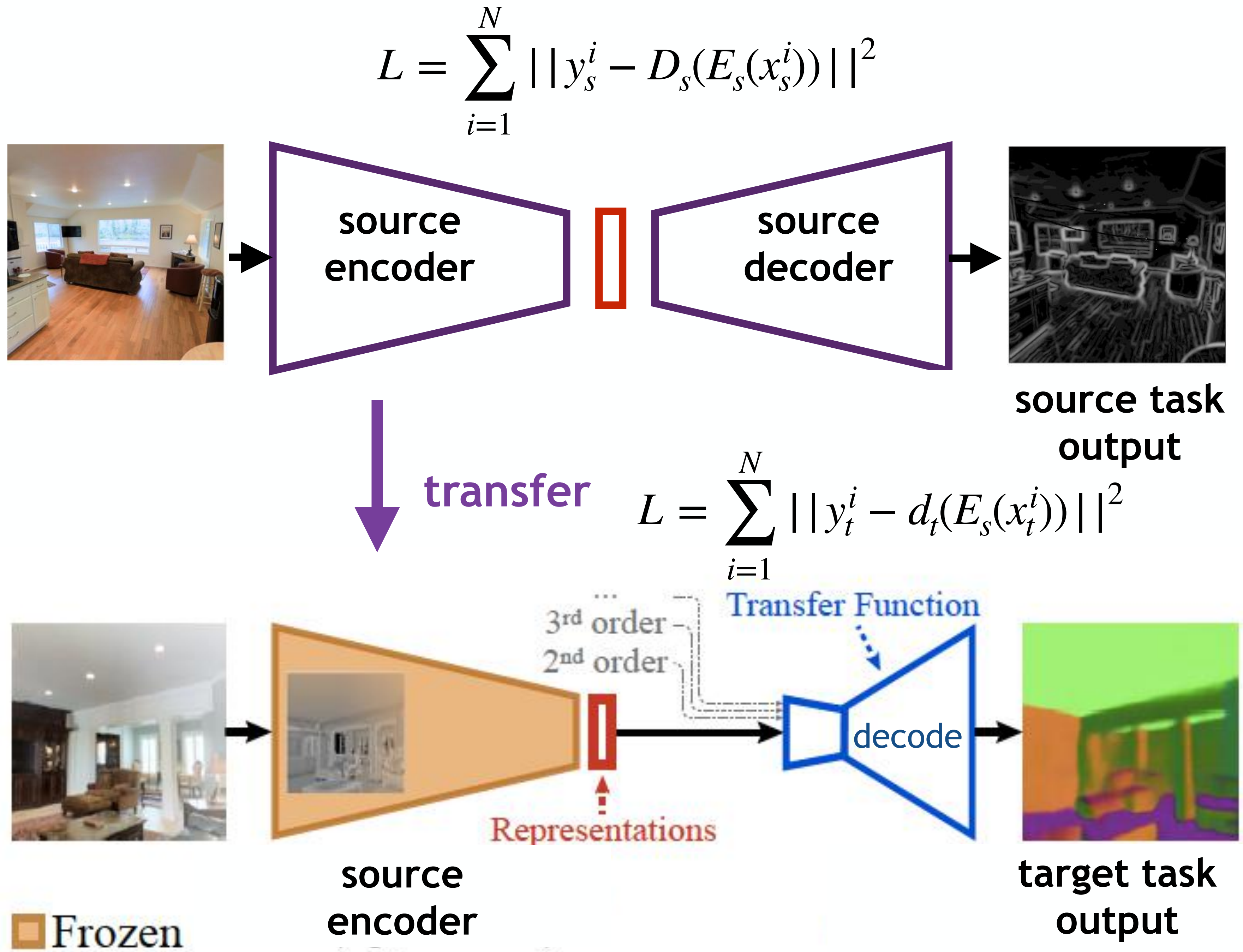
e.g object classification -> scene classification



intuition: low level features are shared across most vision tasks

Heterogeneous Task Transfer Learning

- Heterogeneous task transfer learning using encoder-decoder network

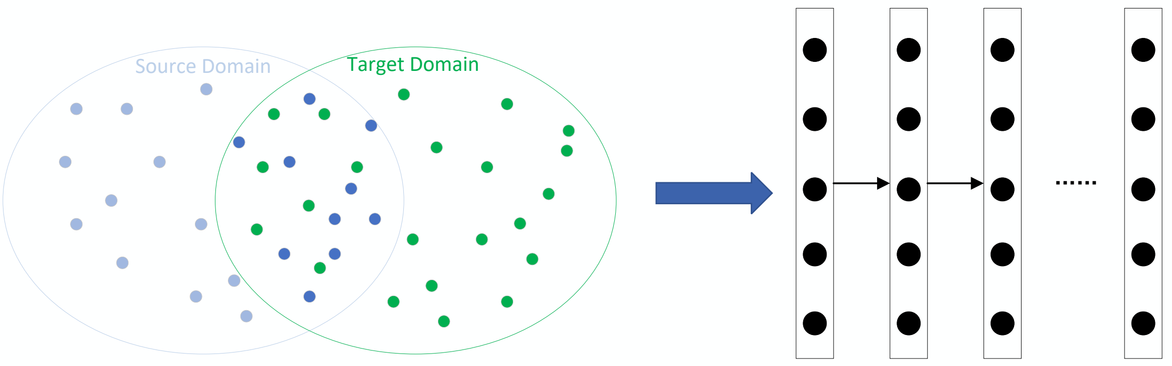


Today's Talk

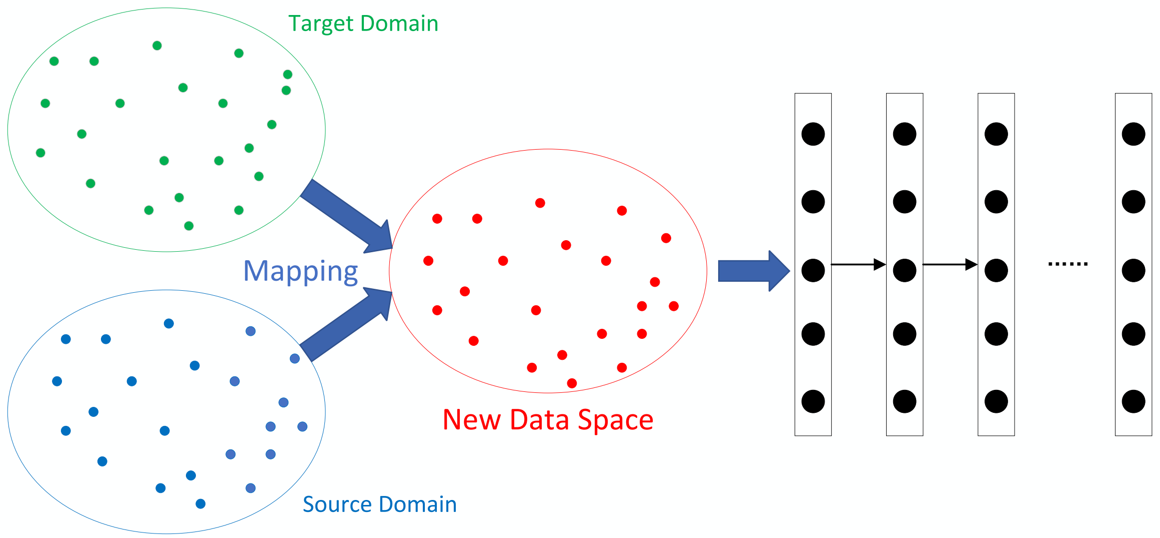
- What's Transfer Learning
- Transfer Learning Techniques
  - Task transfer learning
  - Domain adaptation
  - Transfer bound on domain adaptation
- How to avoid negative transfer?
  - Case study on feature transferability
  - Task transferability: empirical and theoretical methods
- Discussions and Q&A

Domain Adaptation Techniques

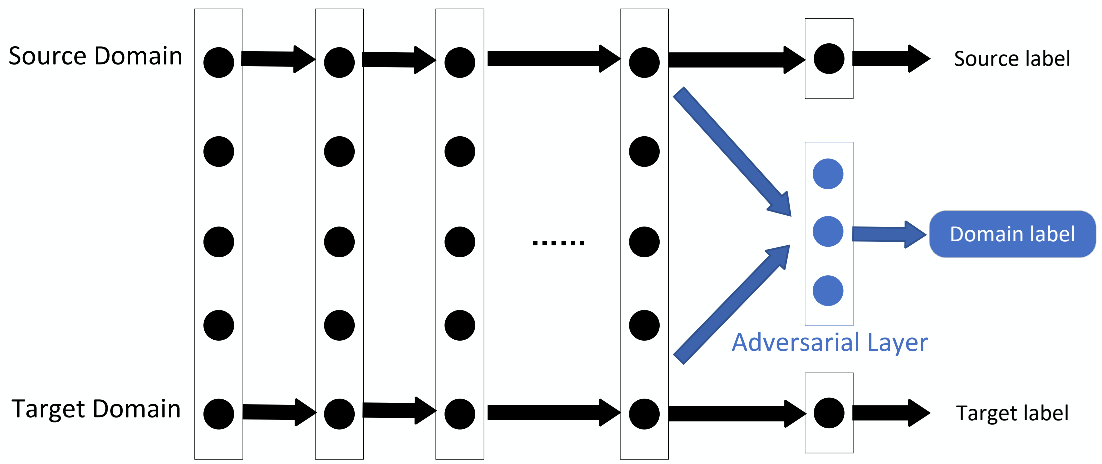
- Instance-based approach



- Mapping-based approach

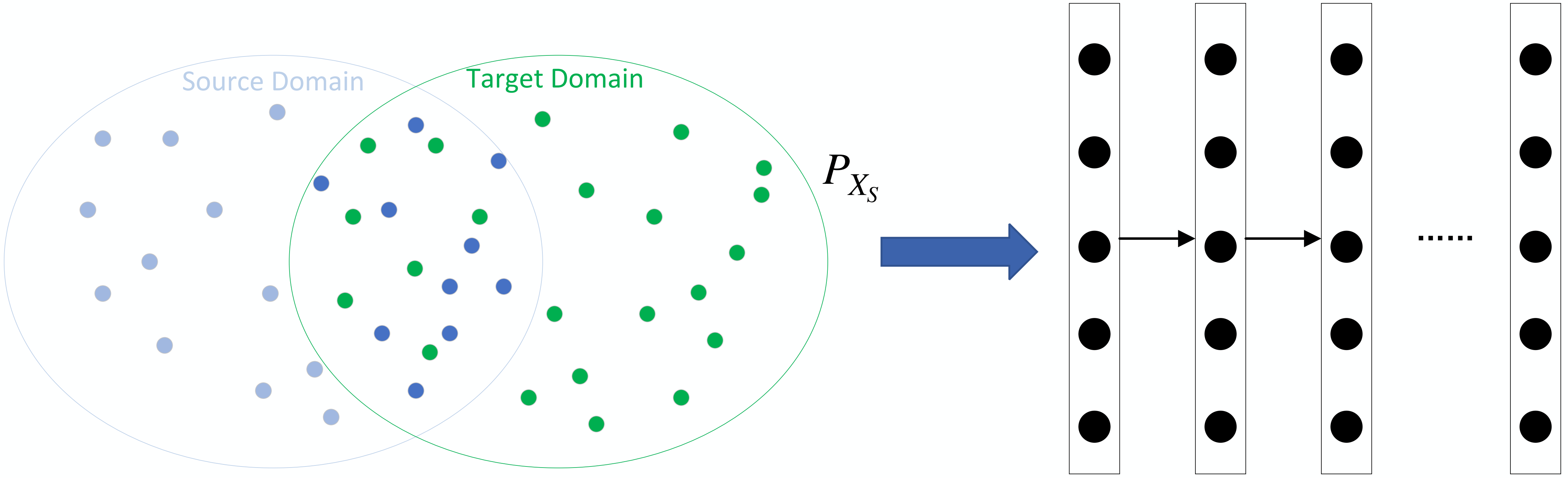


- Adversarial-based approach



Instance-based approaches

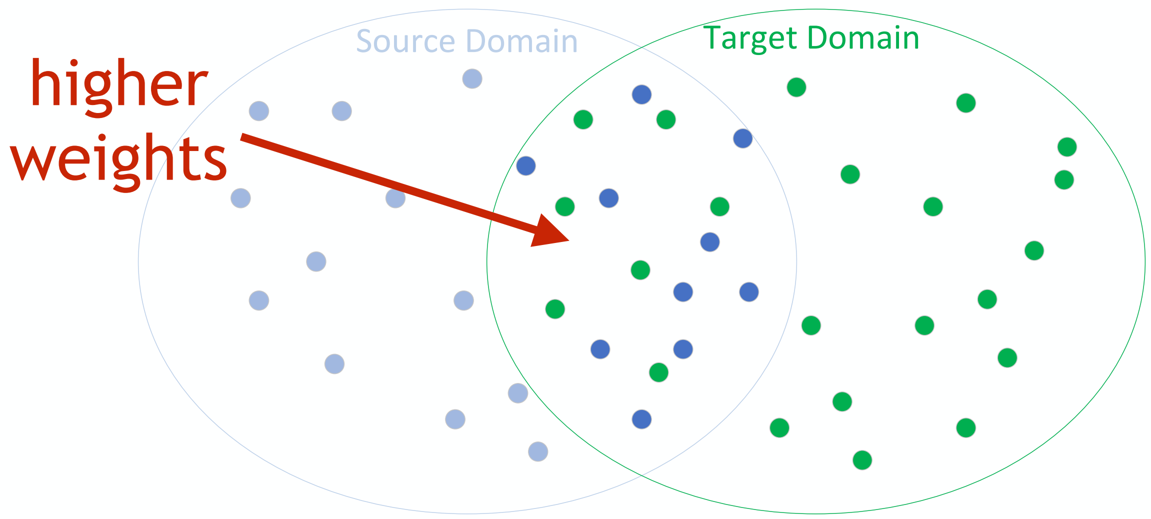
- select **partial instances** from the source domain as supplements to the training set in the target domain



Partial instances in the source domain can be utilized by the target domain with appropriate weights

Boosting for instance-based transfer

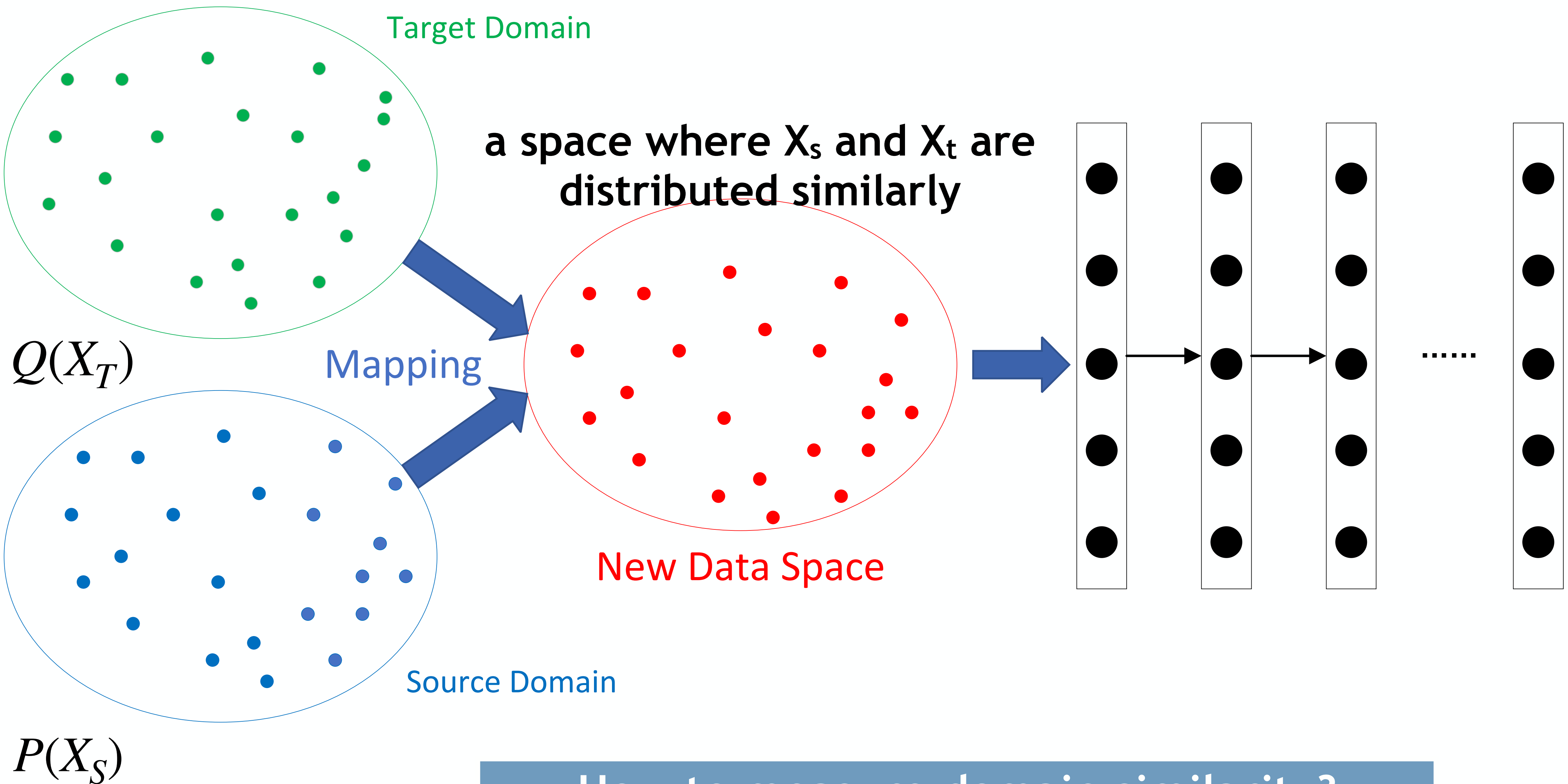
- TrAdaBoost (Dai 2007)
  - Use AdaBoost to filter out source domain instances that are dissimilar to target domain
  - Reweight source domain instances to resemble target domain distribution
  - Train model with reweighted source + target domain instances



- TaskTrAdaBoost (2010): a boosting technique for transferring from multiple sources

Mapping-based approach

- Mapping instances from the source domain and target domain into a new data space



How to measure domain similarity?

Maximal Mean Discrepency (MMD)

- Maximal Mean Discrepency : a kernel-based 2 sample test for the null hypothesis P=Q (Fortet and Mourier, 1953)

$$D_{MMD}[P,Q] \triangleq \sup_{\phi \in \mathscr{F}} (\mathbb{E}_P[\phi(X)] - \mathbb{E}_Q[\phi(Y)])$$

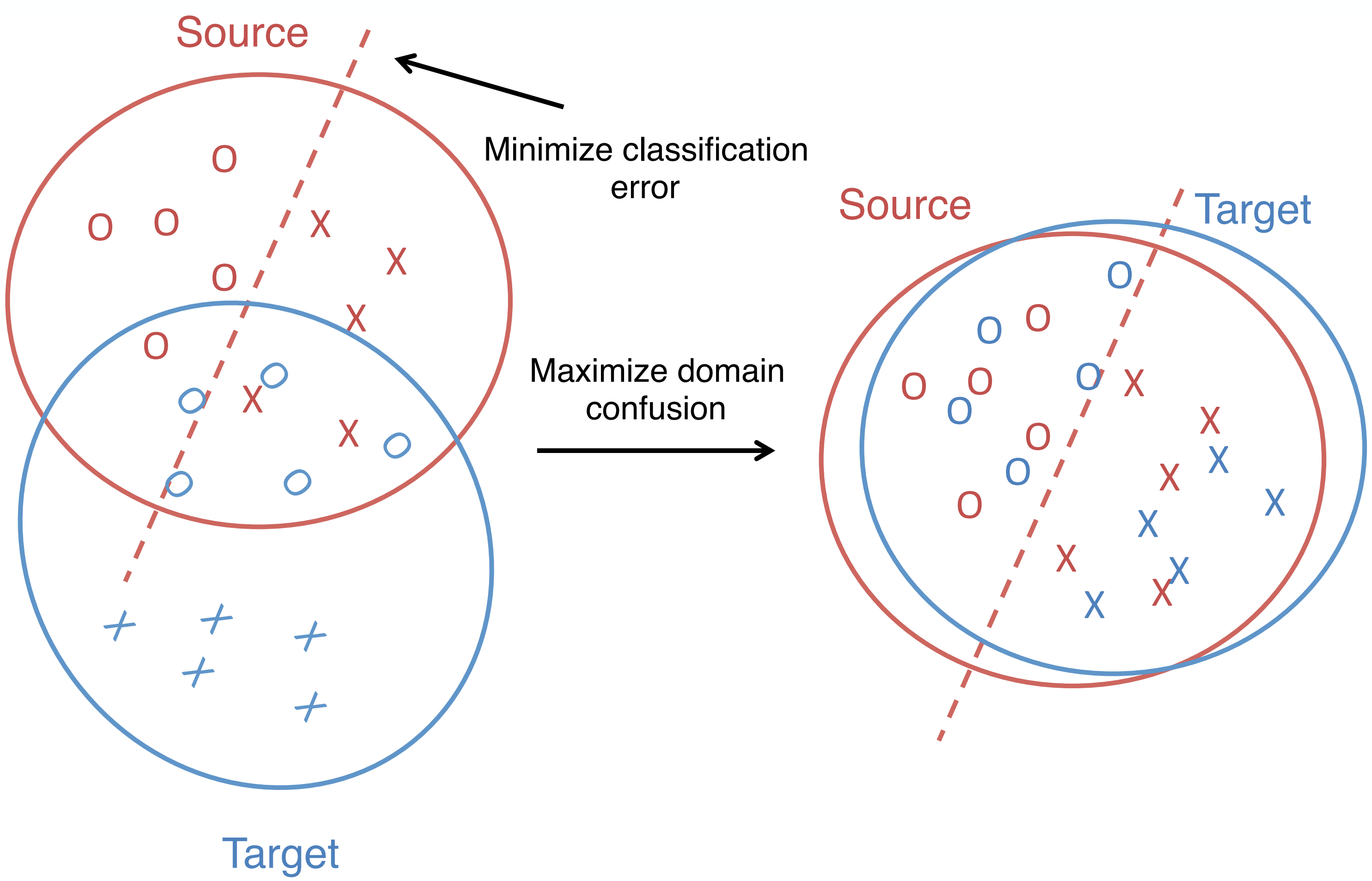
  - where $X \sim P, Y \sim Q$
  - feature map $\phi(\cdot)$
- Used in Transfer Component Analysis (TCA) (Yang, 2018) to correct domain shift

$$D_{MMD}(X_S, X_T) = \left\| \frac{1}{N_S} \sum_{x_s \in X_S} \phi(x_s) - \frac{1}{N_T} \sum_{x_t \in X_T} \phi(x_t) \right\|_{\mathscr{H}}$$

Use MMD as a Domain Regularization Term

- Given pre-trained source model, train an adpation network that minimizes classification error and domain MMD

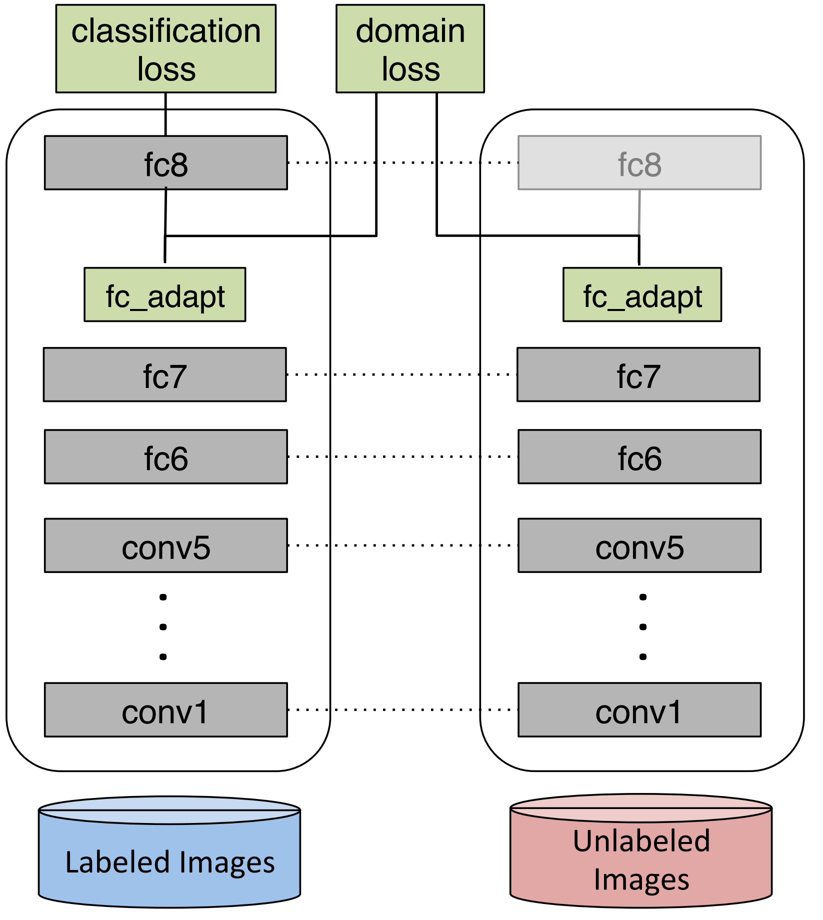
$$L = L_C(X_L, y) + \lambda D^2_{MMD}(X_S, X_T)$$



Tzeng et. al. Deep Domain Confusion: Maximizing for Domain Invariance

Use MMD as a Domain Regularization Term

- Training step:
  - 1. Select the layer to transfer from using MMD metric
  - 2. Train an adaptation layer f_a on source and target data using MMD as a regularizer
- Testing step:
  - Transform target input by $f_a(X_T)$

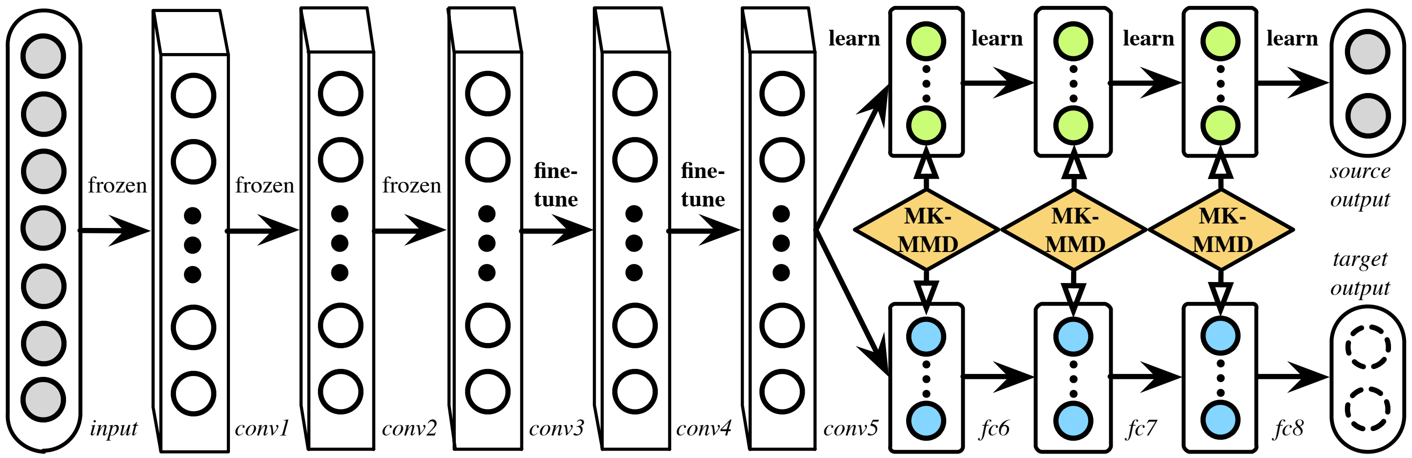


$$L = L_C(X_L, y) + \lambda D^2_{MMD}(X_S, X_T)$$

Tzeng et. al. Deep Domain Confusion: Maximizing for Domain Invariance

Variations with MMD-based domain adaptation

- Deep Adaptation Network (Long et.al. 2015):
  - Use multi-kernel MMD (MK-MMD)
    $$D_{MMD}[P, Q, K] \triangleq \|(\mathbb{E}_P[\phi(X)] - \mathbb{E}_Q[\phi(Y)])\|_{\mathscr{H}_K}$$
  - Fine-tune source task jointly with MMD constraints on multiple layers



- Joint Adaptation (2018): adapt joint distributions instead of $P(X_s)$, $Q(X_t)$

Comparisons of MMD-based domain adaptation methods

- Office+Caltech Benchmark

-

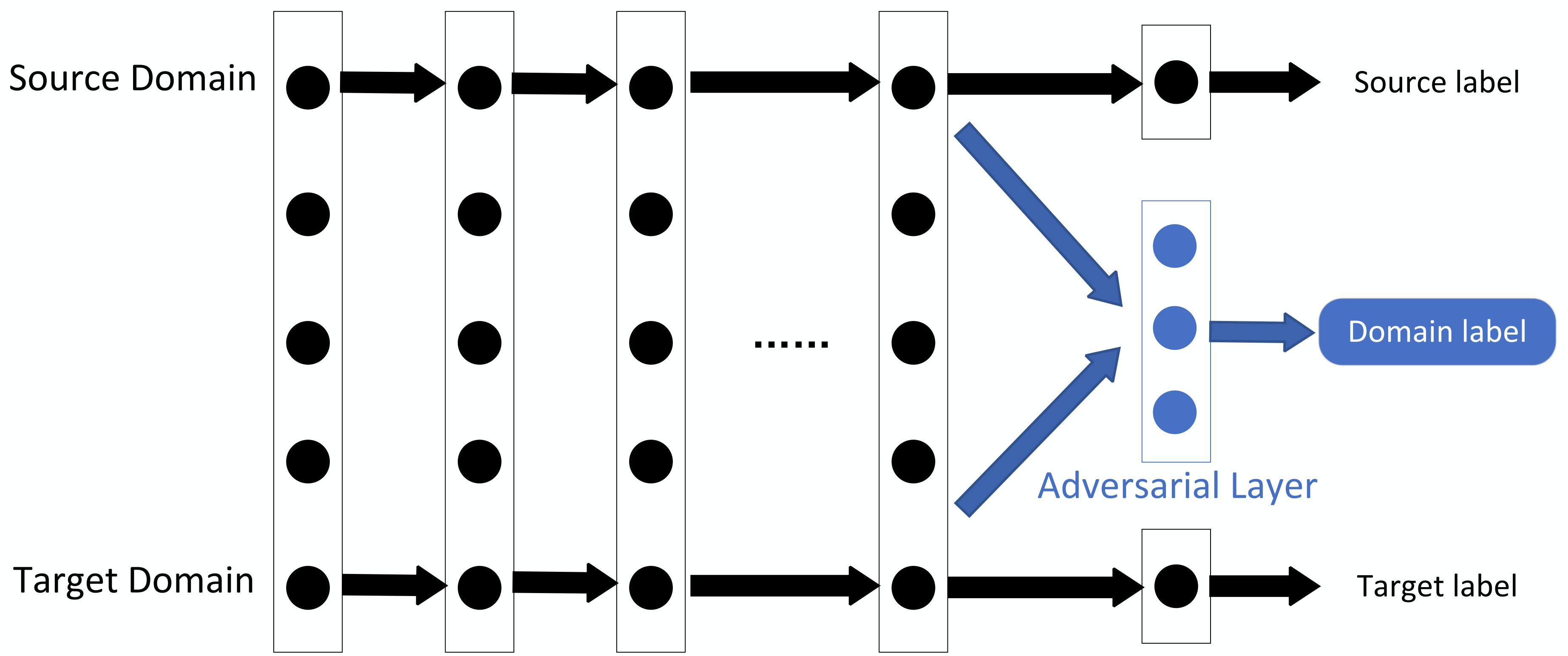
Table 1. Classification accuracy (%) on *Office-31* dataset for unsupervised domain adaptation (AlexNet and ResNet)

Method	A → W	D → W	W → D	A → D	D → A	W → A	Avg
AlexNet (Krizhevsky et al., 2012)	61.6±0.5	95.4±0.3	99.0±0.2	63.8±0.5	51.1±0.6	49.8±0.4	70.1
TCA (Pan et al., 2011)	61.0±0.0	93.2±0.0	95.2±0.0	60.8±0.0	51.6±0.0	50.9±0.0	68.8
GFK (Gong et al., 2012)	60.4±0.0	95.6±0.0	95.0±0.0	60.6±0.0	52.4±0.0	48.1±0.0	68.7
DDC (Tzeng et al., 2014)	61.8±0.4	95.0±0.5	98.5±0.4	64.4±0.3	52.1±0.6	52.2±0.4	70.6
DAN (Long et al., 2015)	68.5±0.5	96.0±0.3	99.0±0.3	67.0±0.4	54.0±0.5	53.1±0.5	72.9
RTN (Long et al., 2016)	73.3±0.3	**96.8**±0.2	**99.6**±0.1	71.0±0.2	50.5±0.3	51.0±0.1	73.7
RevGrad (Ganin & Lempitsky, 2015)	73.0±0.5	96.4±0.3	99.2±0.3	72.3±0.3	53.4±0.4	51.2±0.5	74.3
JAN (ours)	74.9±0.3	96.6±0.2	99.5±0.2	71.8±0.2	**58.3**±0.3	55.0±0.4	76.0
JAN-A (ours)	**75.2**±0.4	96.6±0.2	**99.6**±0.1	**72.8**±0.3	57.5±0.2	**56.3**±0.2	**76.3**
ResNet (He et al., 2016)	68.4±0.2	96.7±0.1	99.3±0.1	68.9±0.2	62.5±0.3	60.7±0.3	76.1
TCA (Pan et al., 2011)	72.7±0.0	96.7±0.0	99.6±0.0	74.1±0.0	61.7±0.0	60.9±0.0	77.6
GFK (Gong et al., 2012)	72.8±0.0	95.0±0.0	98.2±0.0	74.5±0.0	63.4±0.0	61.0±0.0	77.5
DDC (Tzeng et al., 2014)	75.6±0.2	96.0±0.2	98.2±0.1	76.5±0.3	62.2±0.4	61.5±0.5	78.3
DAN (Long et al., 2015)	80.5±0.4	97.1±0.2	99.6±0.1	78.6±0.2	63.6±0.3	62.8±0.2	80.4
RTN (Long et al., 2016)	84.5±0.2	96.8±0.1	99.4±0.1	77.5±0.3	66.2±0.2	64.8±0.3	81.6
RevGrad (Ganin & Lempitsky, 2015)	82.0±0.4	96.9±0.2	99.1±0.1	79.7±0.4	68.2±0.4	67.4±0.5	82.2
JAN (ours)	85.4±0.3	**97.4**±0.2	**99.8**±0.2	84.7±0.3	68.6±0.3	70.0±0.4	84.3
JAN-A (ours)	**86.0**±0.4	96.7±0.3	99.7±0.1	**85.1**±0.4	**69.2**±0.4	**70.7**±0.5	**84.6**

Long et. al. (2017). Deep Transfer Learning with Joint Adaptation Networks.

Adversarial-based approach

- Adopt adversarial training in learning transferable representation.

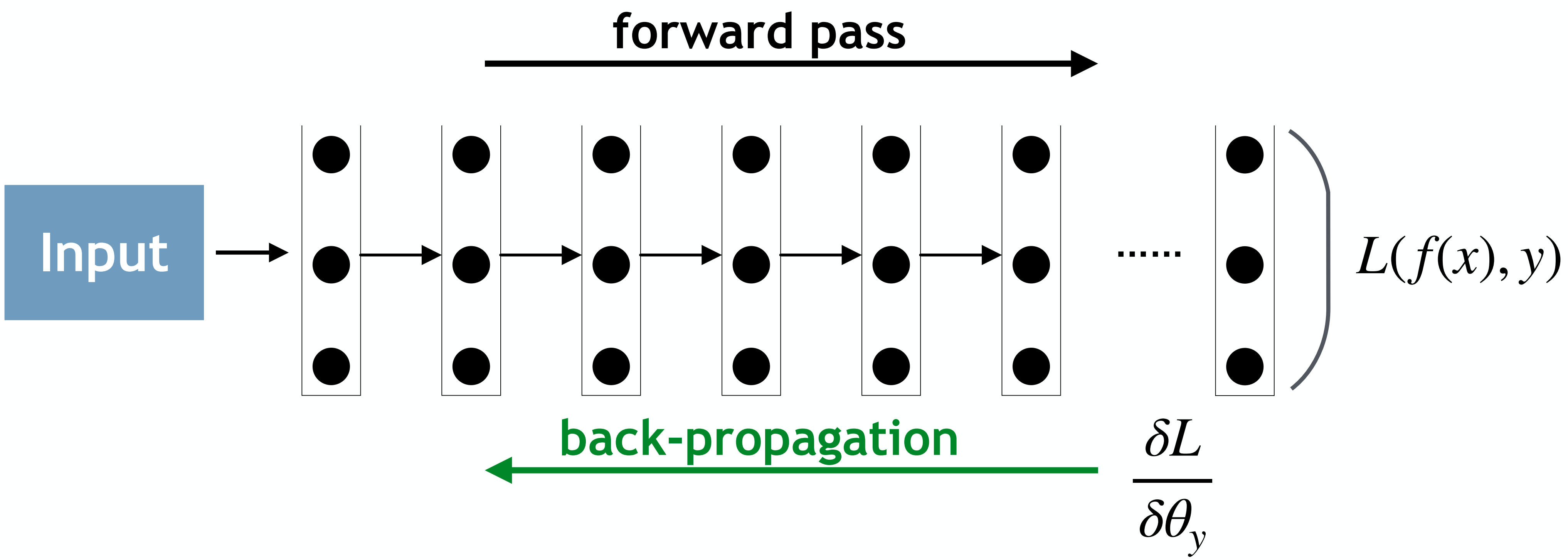


Effective features should be discriminative for the main learning task and indiscriminative between the source domain and target domain.

Adversarial-based approach

Ajakan et al. (2014) Domain-adversarial neural networks.

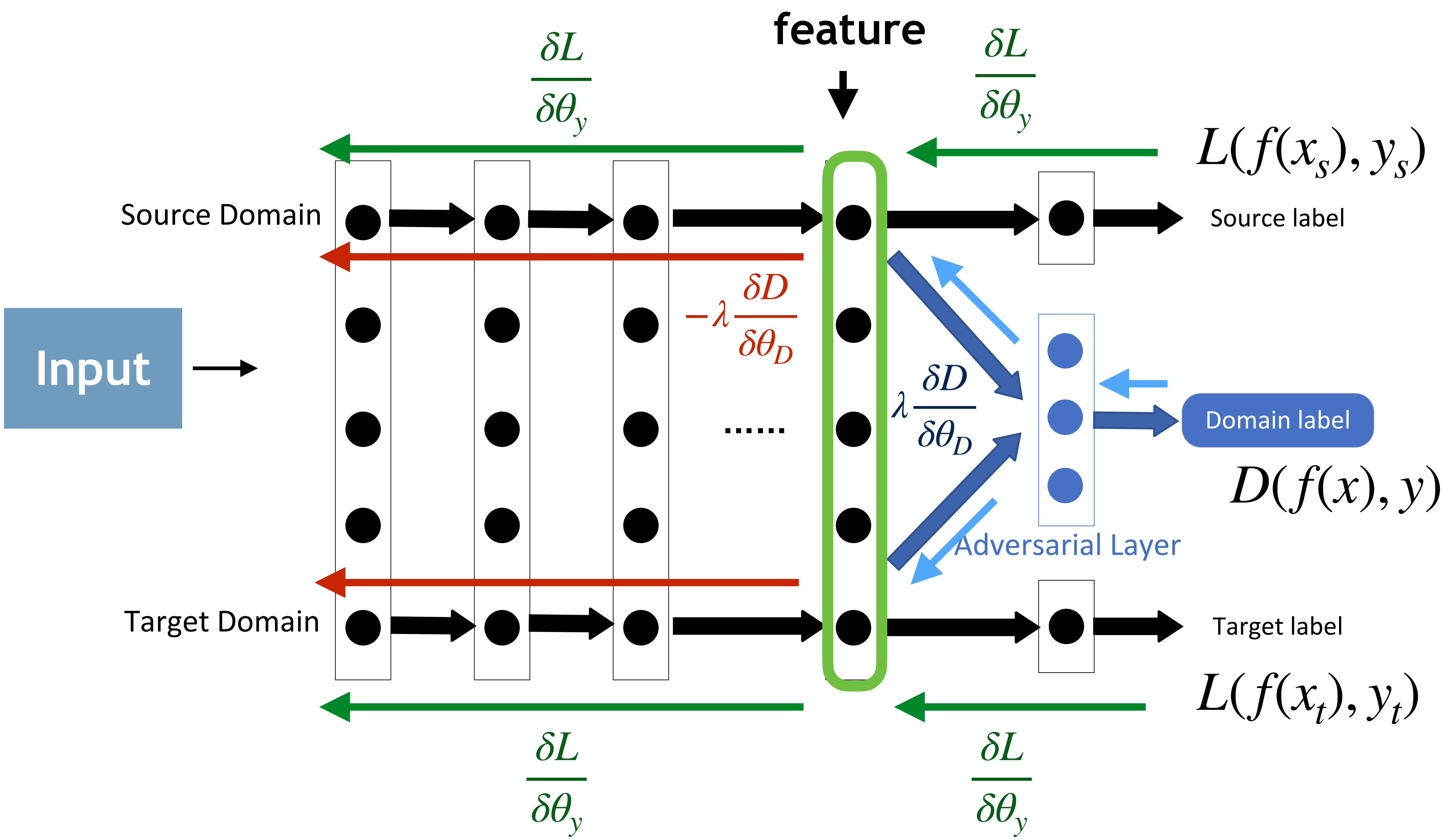
- Standard deep neural network training



Domain Adversarial Neural Networks

Ajakan et al. (2014) Domain-adversarial neural networks.

- Gradient Reversal



Domain Adversarial Neural Networks (DANN)

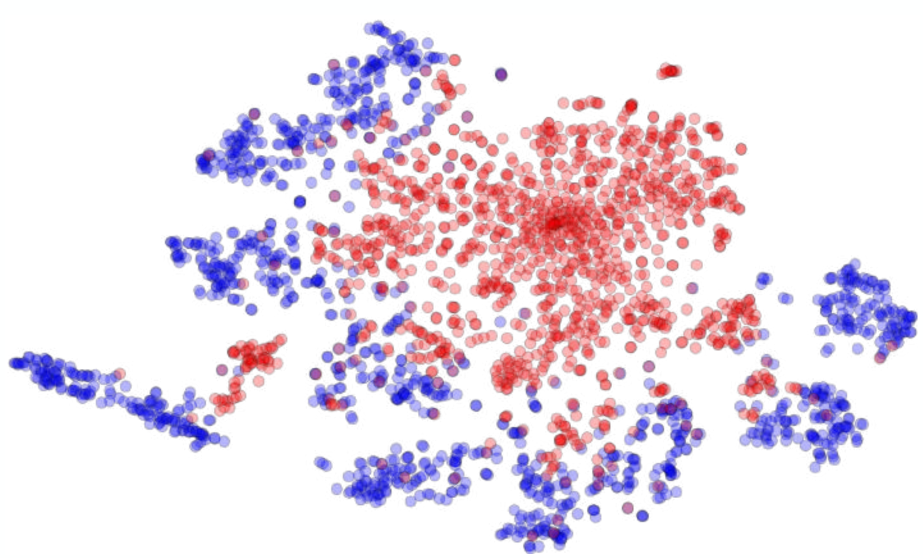
Ajakan et al. (2014) Domain-adversarial neural networks.

- DNN adapted feature distribution

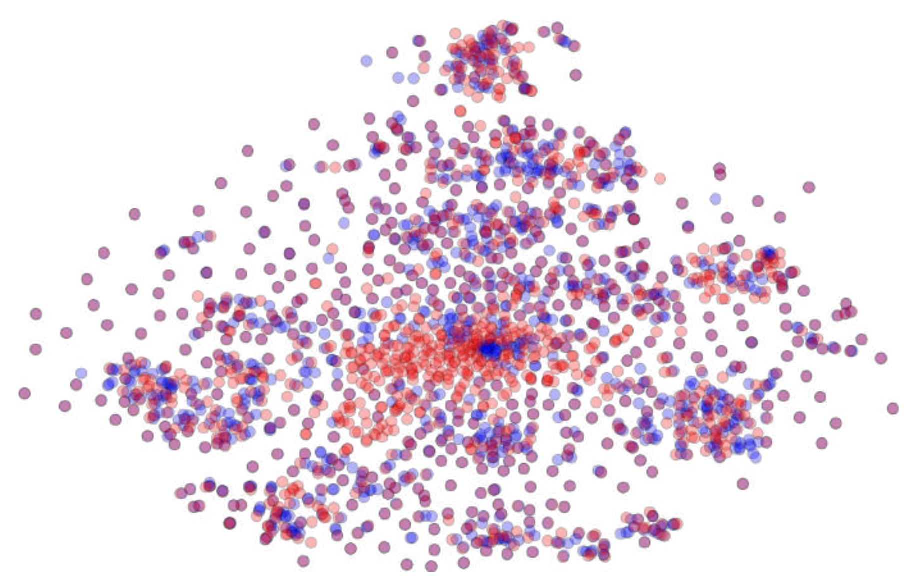
source domain (MINIST)

target domain (MNIST-M)

TSNE visualization of CNN extracted features



Non-Adapted



Adapted

Domain Adaptation Discussion

- Instance-based approach: select and reweight instances in the source domain to be similar to the target distribution

  easy to implement, work with any base classifiers

- Mapping-based approach: map source and target data to latent space where source and target domains are similar

  easy to incorporate to neural network training

- Adversarial-based approach: find features that are indiscriminative between source and target domains

  good performance in computer vision

Why does such methods work?

A detour to learning theory

Transfer Bounds for Domain Adaptation

- Given input $x \sim D$ with discrete alphabet $\mathscr{X}$ and label $y \in \{0,1\}$
- A hypothesis is a function $h : \mathscr{X} \rightarrow \{0,1\}$
- Error (risk) of hypothesis h :

$$\epsilon(h) = \mathbb{E}_{x \sim D}[\,|\,h(x) - y\,|\,]$$

- Empirical risk of hypothesis h given N samples (x_i, y_i) drawn i.i.d. from D:

$$\hat{\epsilon}(h) = \frac{1}{N}\sum_{i=1}^{N} |\,h(x_i) - y_i\,|$$

- Source risk: $\epsilon_S(h) = \mathbb{E}_{x_S \sim P}[\,|\,h(x_S) - y_S\,|\,]$
- Target risk: $\epsilon_T(h) = \mathbb{E}_{x_T \sim Q}[\,|\,h(x_T) - y_T\,|\,]$

Transfer Bounds for Domain Adaptation

Ben-David et.al. (2010). A theory of learning from different domains

Theorem. Let $h \in \mathcal{H}$ be a hypothesis, $\epsilon_S(h)$ and $\epsilon_T(h)$ be risks of source and target respectively, then

$$\epsilon_T(h) \leq \epsilon_S(h) + \boxed{d_{\mathcal{H}}(P, Q)} + C_0$$

C$_0$: a constant for the complexity of $\mathcal{H}$

where

$$d_{\mathcal{H}}(P, Q) \triangleq 2 \sup_{\eta \in \mathcal{H}} \left| \Pr_P[\eta(x_S) = 1] - \Pr_Q[\eta(x_T) = 1] \right|$$

is the **H-divergence** between P and Q.

Lemma. The H-divergence can be bounded by the empirical estimate:

$$d_{\mathcal{H}}(P, Q) \leq \boxed{\hat{d}_{\mathcal{H}}(P, Q)} + C_1$$

Make P and Q as indistinguishable as possible

e.g. minimize MMD, MK-MMD, domain discriminative loss, etc

Decrease the upper bound on target risk !

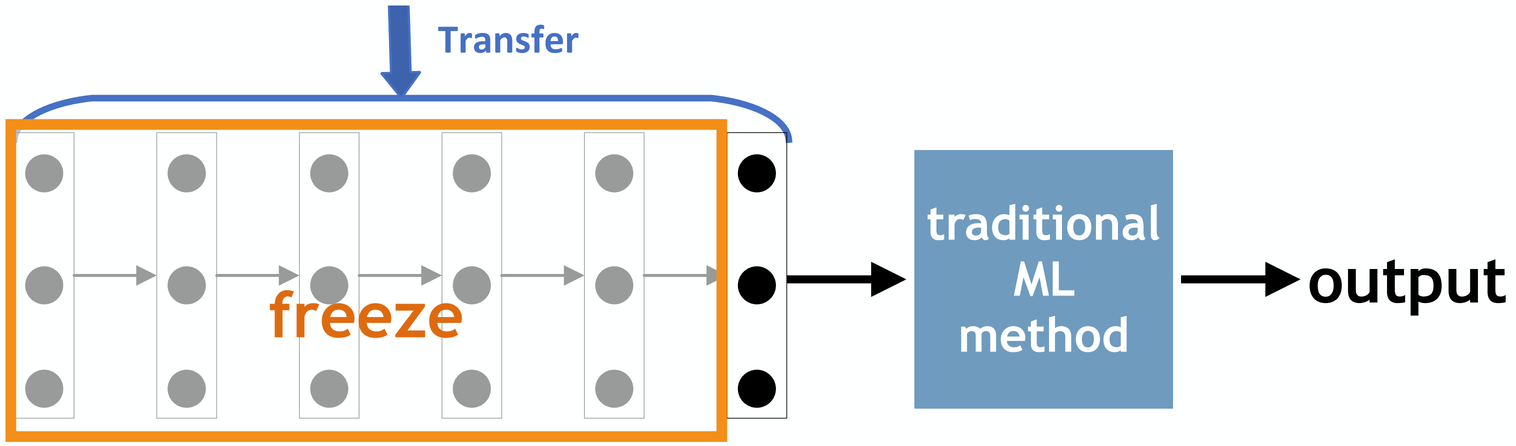
Today’s Talk

- What’s Transfer Learning
- Transfer Learning Techniques
  - Task transfer learning
  - Domain adaptation
  - Transfer bound on domain adaptation
- How to avoid negative transfer?
  - Case study on feature transferability in vision
  - Task transferability: empirical and theoretical methods
- Discussions and Q&A

Where to start fine-tuning?

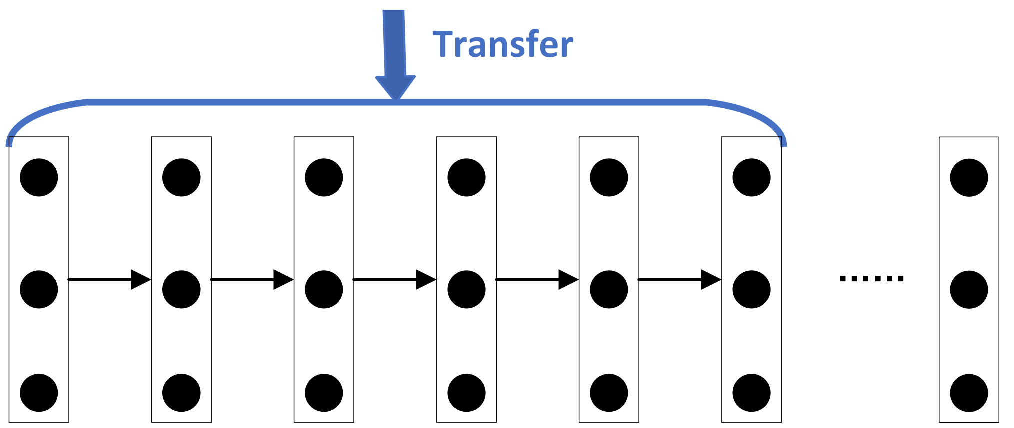
- Use pre-trained model as a fixed feature extractor

  most efficient, but with limited performance



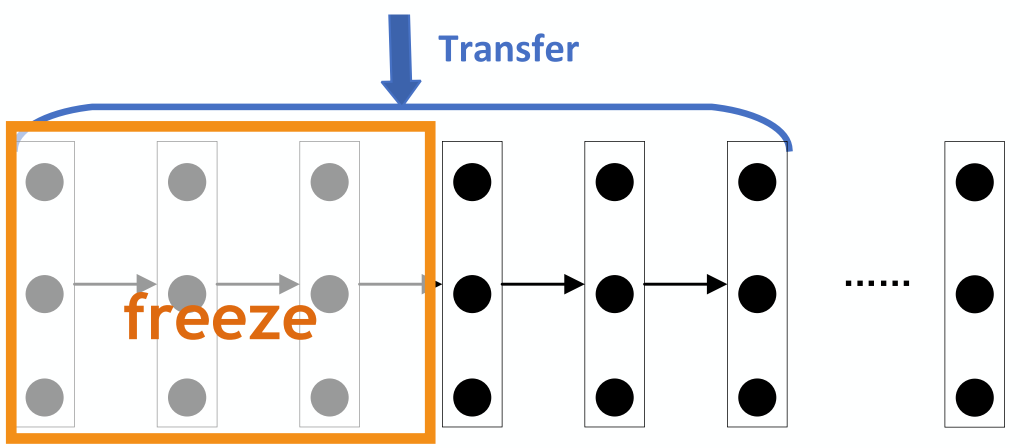
- Fine-tune all the way

  slow, easy to overfit when target labels are few



- Fine-tune first k layers

  How to choose k?



Which layers to transfer?

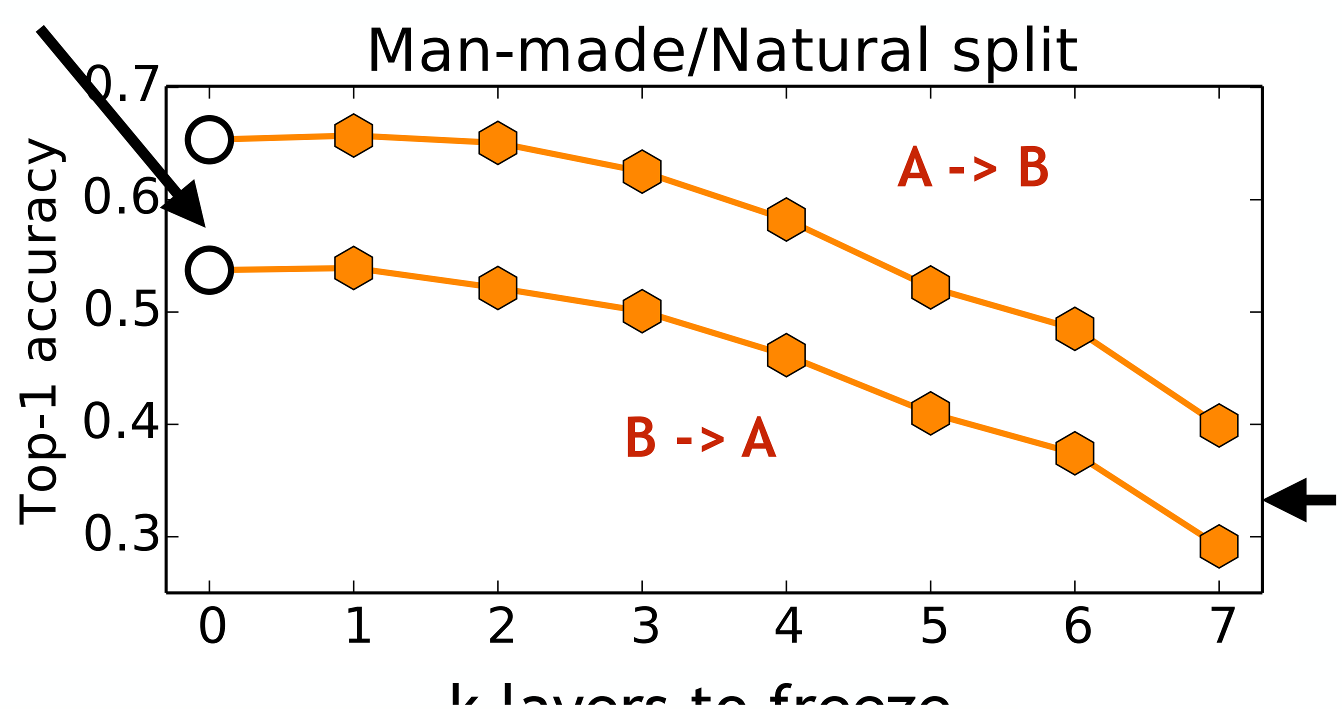
Yosinski et.al. (2014) How transferable are features in deep neural networks?

A case study using ImageNet classification tasks (trained on 7 CNN layers + output layer)

Dissimilar tasks

- Task A: Man-made object classification
- Task B: Natural object classification

baseline (train from scratch)



softmax classification (no fine-tuning)

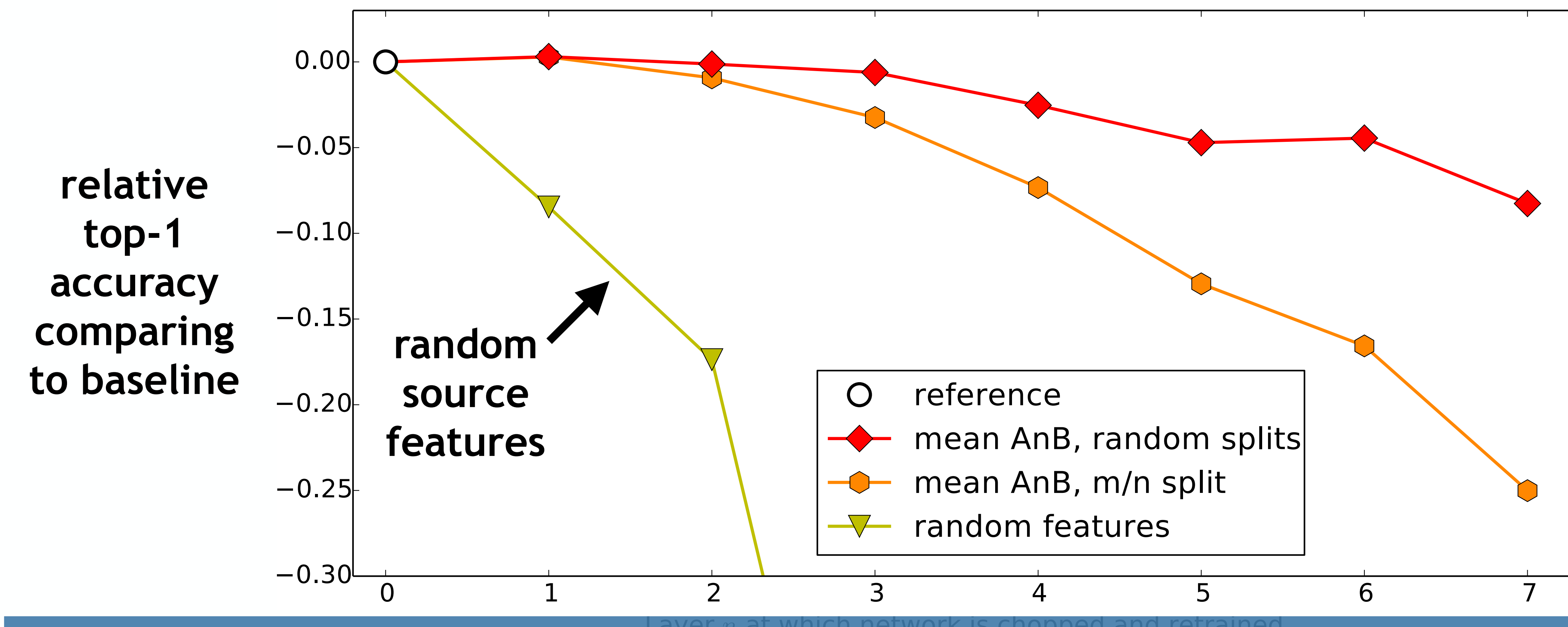
When target supervision is sufficient, performance degrades when more layers are frozen

Which layers to transfer?

Yosinski et.al. (2014) How transferable are features in deep neural networks?

A case study using ImageNet classification tasks (trained on 7 CNN layers + output layer)

- **Similar tasks:** Random A/B split (500 classes in each task)
- **Dissimilar tasks:** Man-made (A) -> Natural (B)

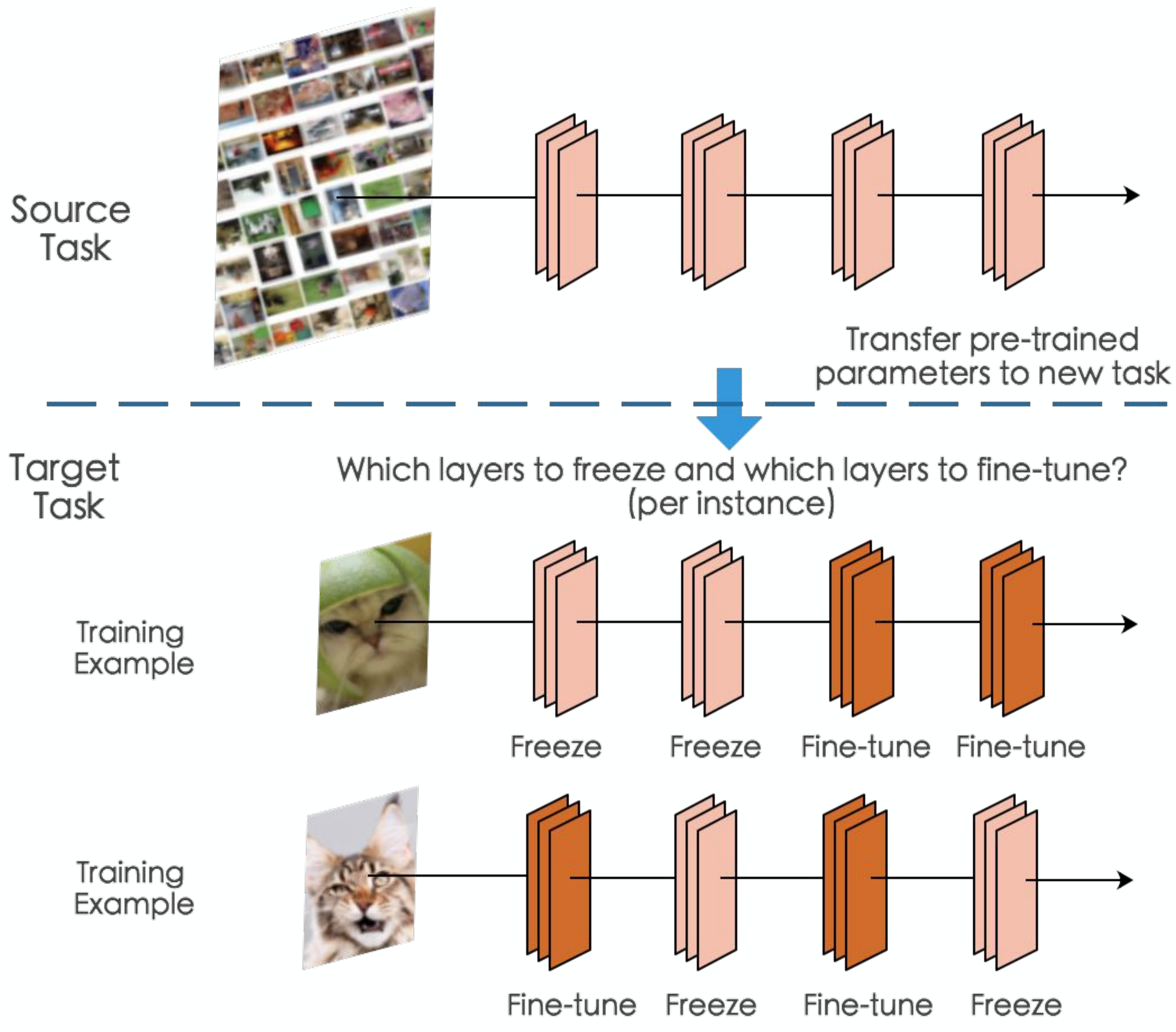


- **transferability gap grows as the distance between tasks increases**
- **features transferred from distant tasks are better than random features!**

Fine-Tune Selected Layers

Guo et.al. (2019) SpotTune: Transfer Learning through Adaptive Fine-tuning

- for each training instance, adaptively decide which sets of layers to fine tune



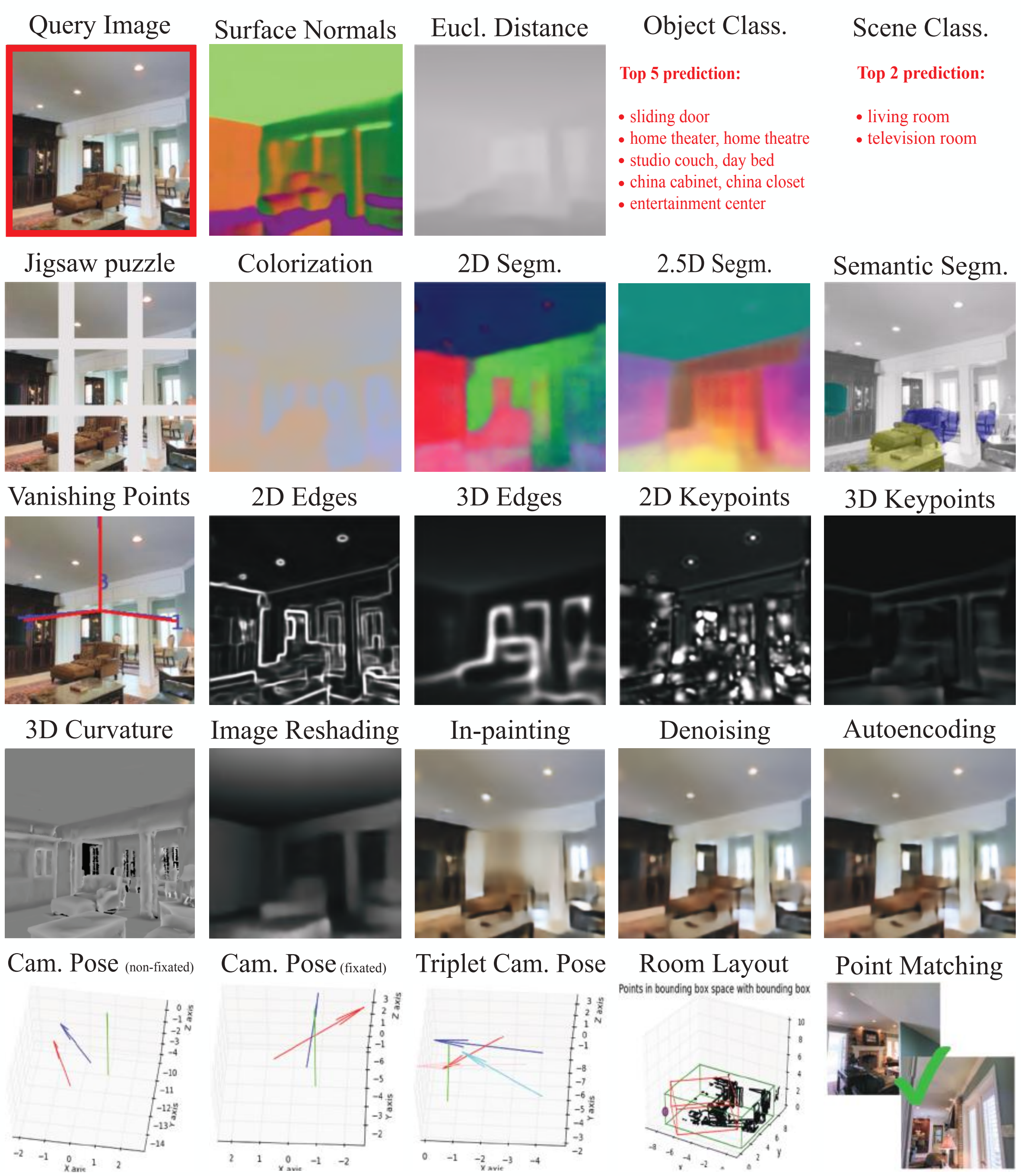
How to Measure Task Transferability?

Zamir et.al. (2018) Taskonomy: Disentangling Task Transfer Learning

investigated the transferability among 26 image-based indoor scene understanding tasks on low-data scenario

Main steps:

1. train task-specific networks (source models) on all data
2. For each S-T task pair, train a transfer network on a small validation dataset (20,000 images)

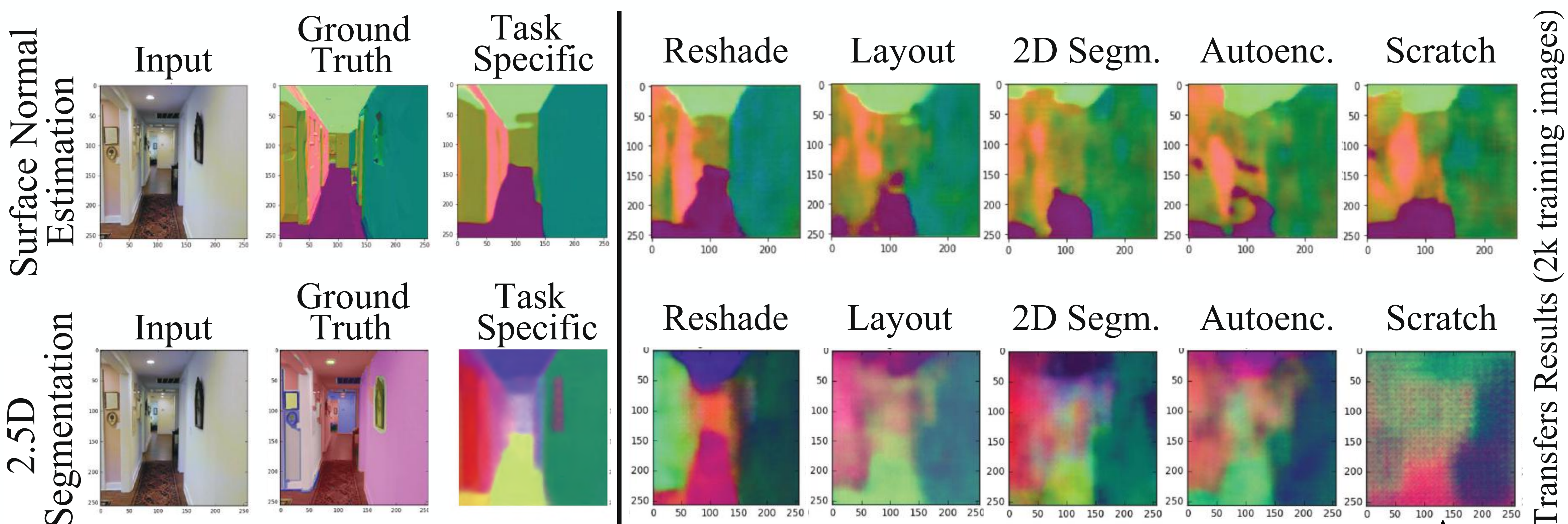


How to Measure Task Transferability?

Zamir et.al. (2018) Taskonomy: Disentangling Task Transfer Learning

- Visual transferability results

On Validation Dataset (1/60 training data)



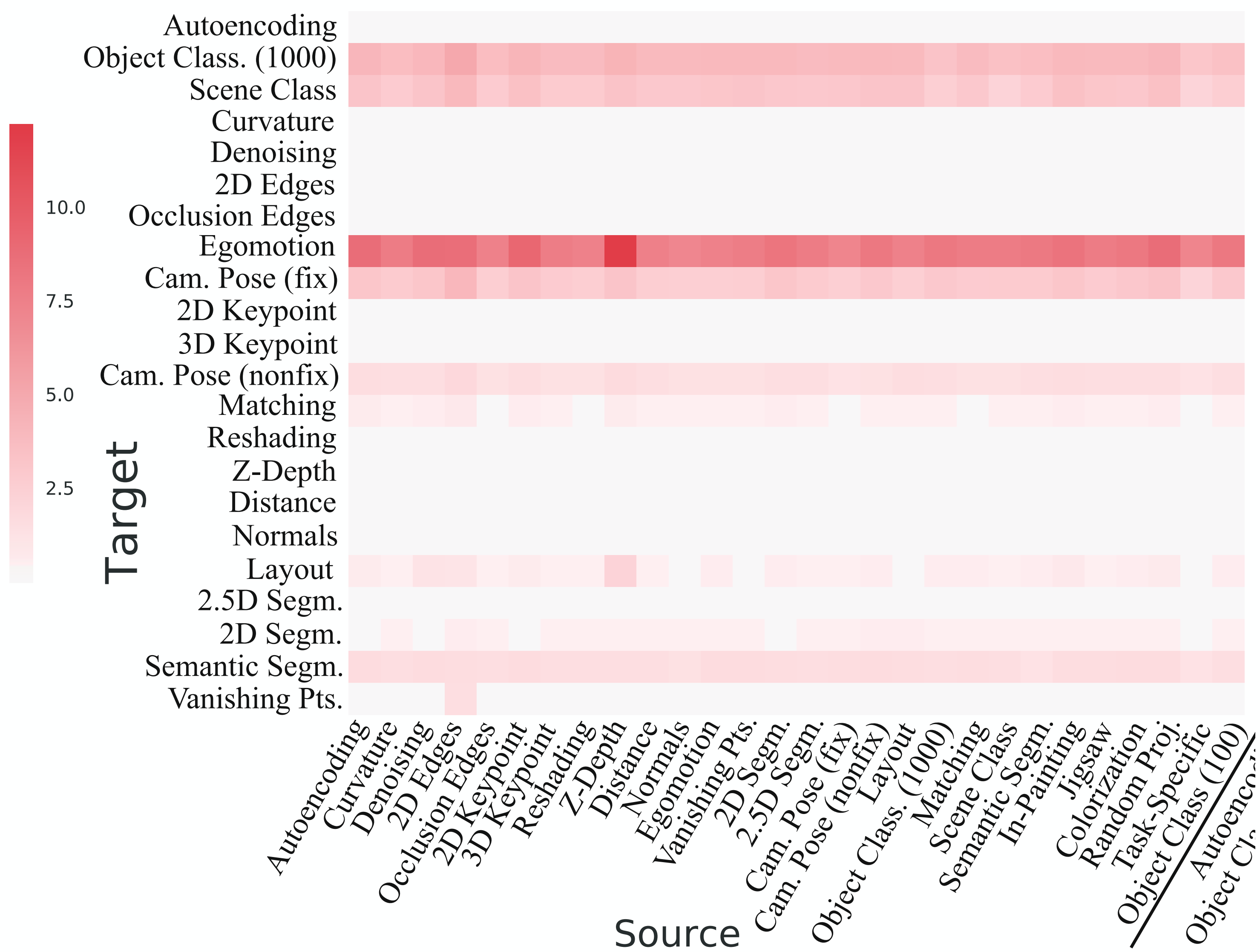
best performance

worst

How to Measure Task Transferability?

Zamir et.al. (2018) Taskonomy: Disentangling Task Transfer Learning

- Raw losses from transfer functions have different scales



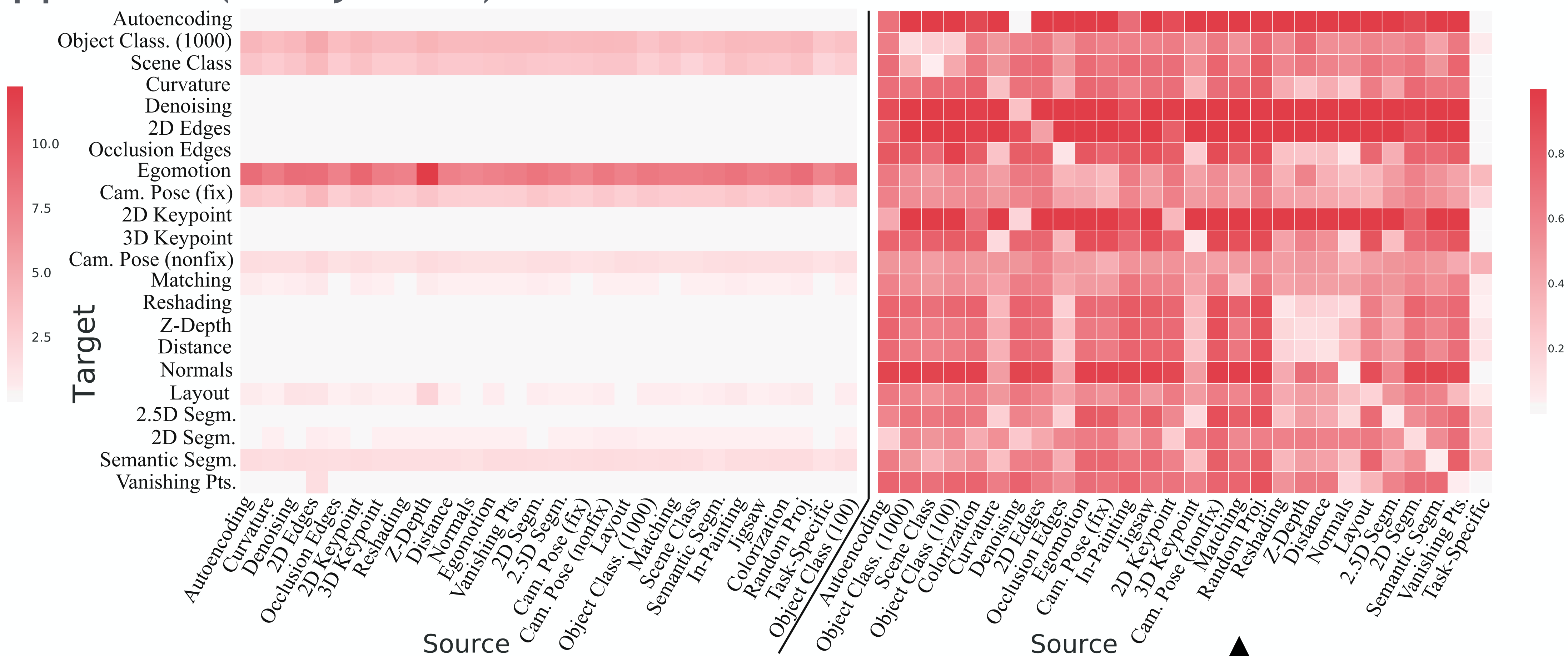
- Naive solution: linear rescale

performance increases at different speed with respective to loss !

How to Measure Task Transferability?

Zamir et.al. (2018) Taskonomy: Disentangling Task Transfer Learning

- Analytic Hierarchy Process (AHP): an ordinal normalization approach (Saaty 1987)



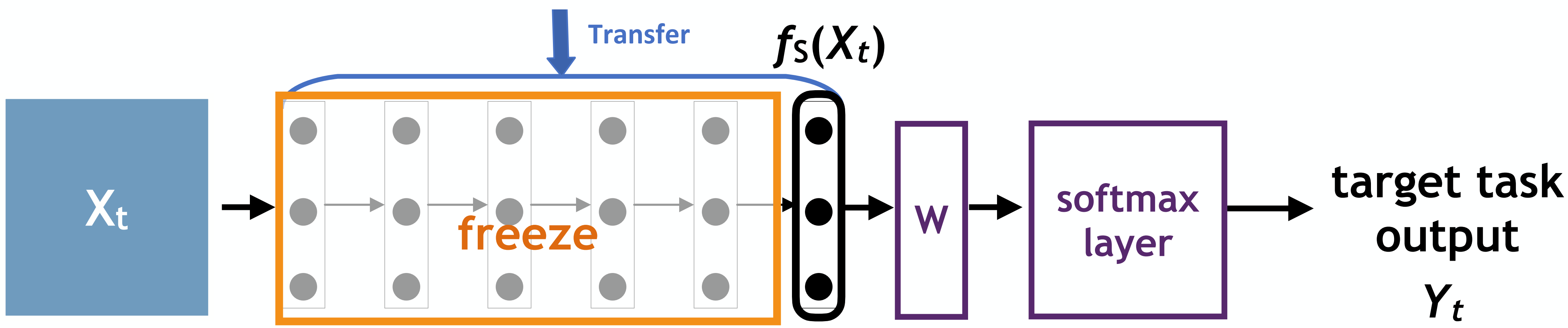
normalized results

Can we estimate transferability without relying on gradient descent?

Measure Task Transferability Analytically

Bao & Li et.al. (2019) An Information-Theoretic Metric for Task Transfera Learning

A simple task transfer learning model (with linear fine-tuning)



Transferability from Task S to Task T

$$\mathfrak{T}(S,T) \triangleq \frac{\textbf{Target Performance of } f_S}{\textbf{Optimal Target Performance}} \qquad \begin{cases} \mathfrak{T}(S,T) = 1 \\ 0 \leq \mathfrak{T}(S,T) \leq 1 \\ \mathfrak{T}(S,T) = 0 \end{cases}$$

How to measure feature performance ?

Feature performance via local information geometry

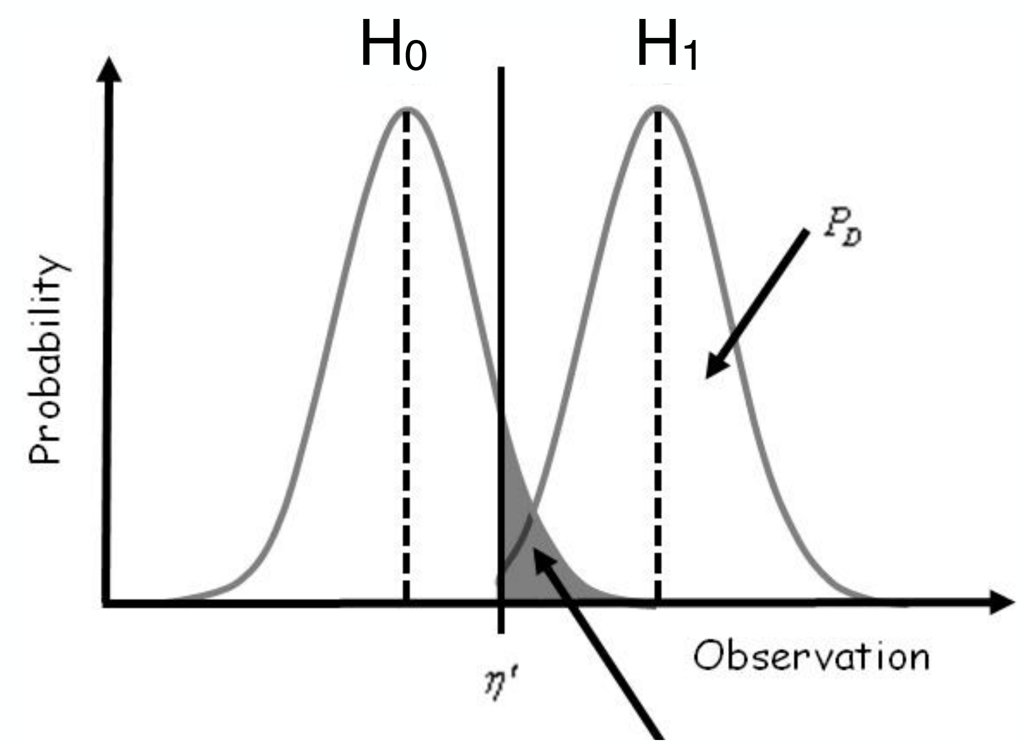
— a statistical view of binary classification

- Binary hypothesis testing of m observations of x:

$$H_0 : x \sim P_{X|Y=0}, \quad H_1 : x \sim P_{X|Y=1}$$

- Error exponent E_f: the asymptotic rate at which the error probability of f(x) decays as m increases

$$\lim_{m\to\infty} -\frac{1}{m}\log(P_e) = E$$



Theorem. (Huang et al. 2015) When $P_{X|Y=0}$, $P_{X|Y=1}$, and P_X are locally distributed, for some constant $c > 0$

$$E_f = c\underbrace{\mathcal{H}(f)}_{\text{H-score of f(X)}}$$

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

An Information-Theoretic Metric for Transferability

$$\mathfrak{T}(S,T) \triangleq \frac{\textbf{Target Performance of } f_S}{\textbf{Optimal Target Performance}} = \frac{\mathscr{H}_T(f_S)}{\mathscr{H}_T(f_T^*)}$$

H-score of source feature $\mathscr{H}_T(f_S)$

- 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
```

Maximal H-score: $\mathscr{H}_T(f_T^*)$

- Discrete X: Alternating Conditional Expectation (ACE) algorithm

  Makur et. al. (2015) An Efficient algorithm for information decomposition and extraction

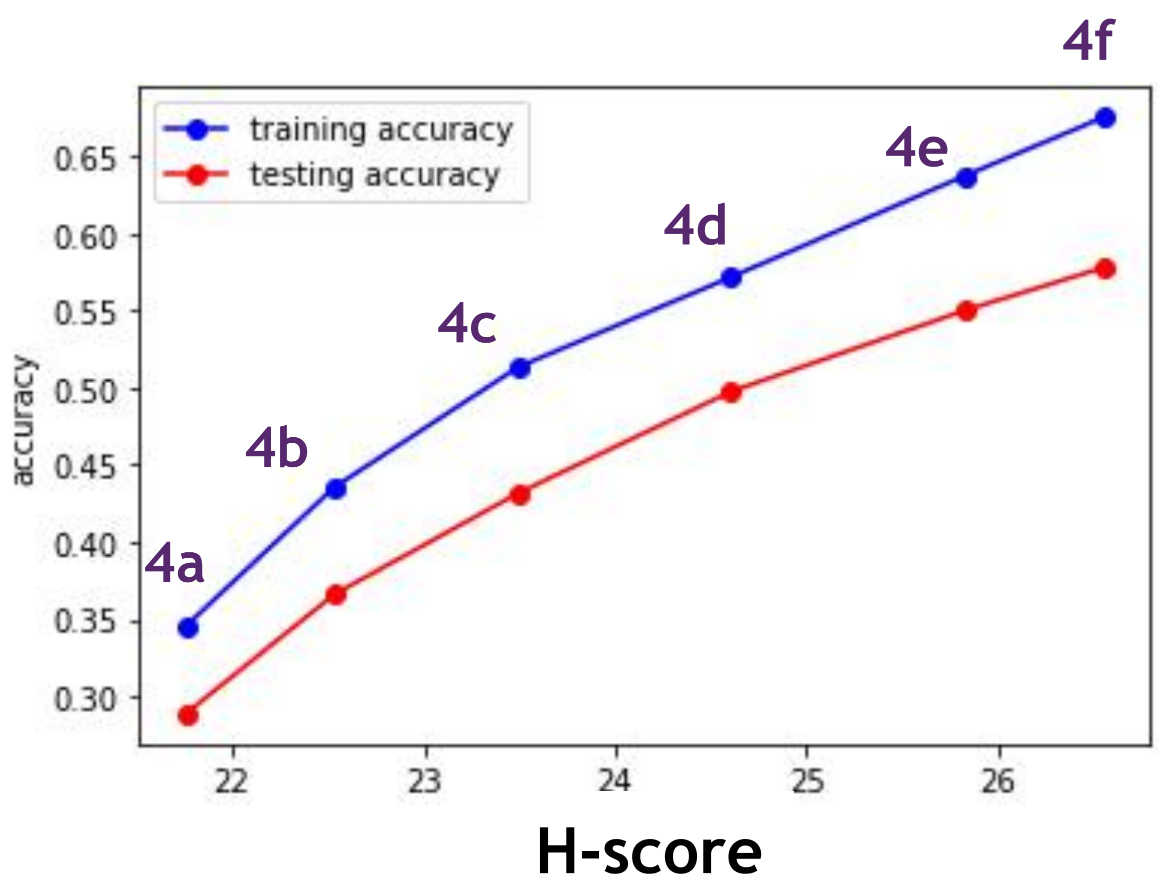
- Continuous X: Neural network formulation

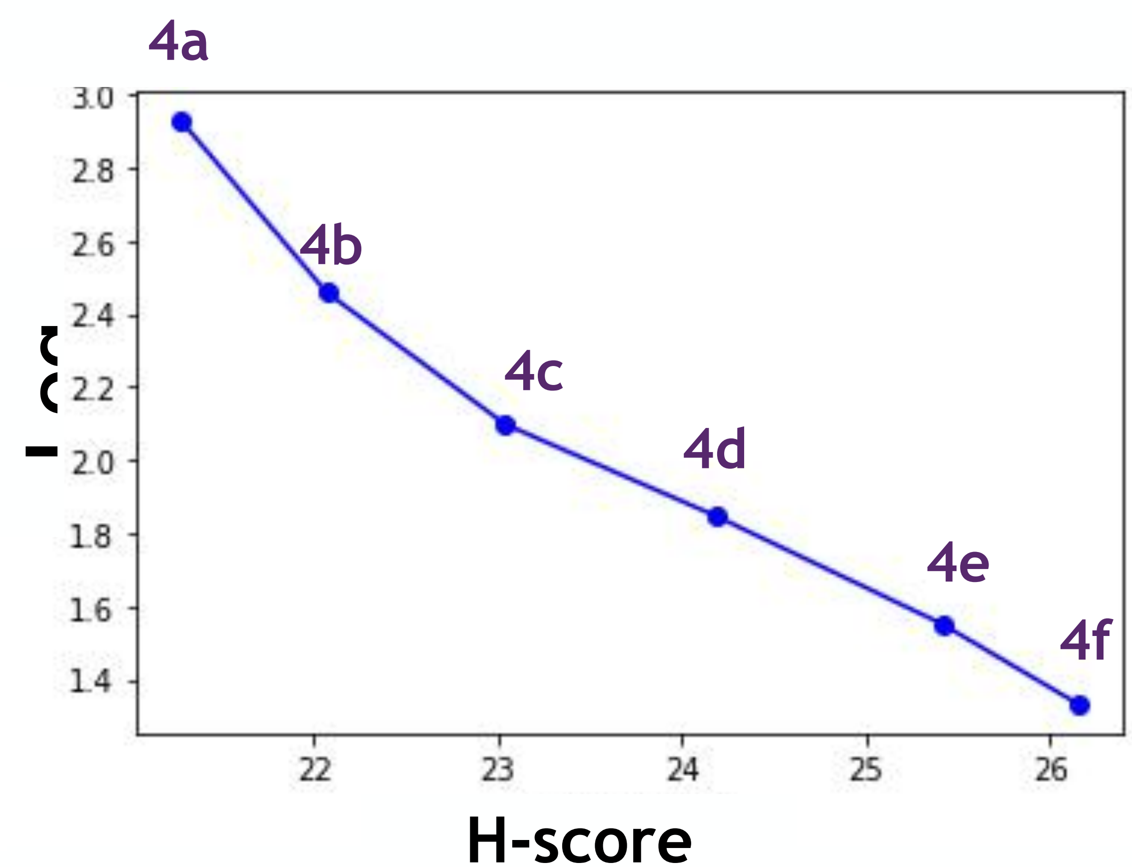
  Wang et. al. (2018) An Efficient Approach to Informative Feature Extraction from Multimodal Data

In source feature task selection problems, only need to compute $\mathcal{H}_T(f_S)$!

An Information-Theoretic Metric for Transferability

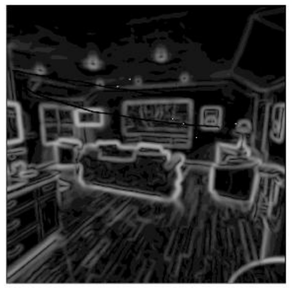
- Source task: ImageNet 1000 classification (ResNet50 features from 6 layers 4a-4f)
- Target task: Cifar 100-class classification on 20,000 images





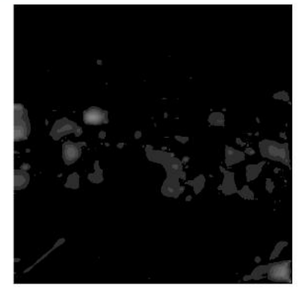
An Information-Theoretic Metric for Transferability

Query Image

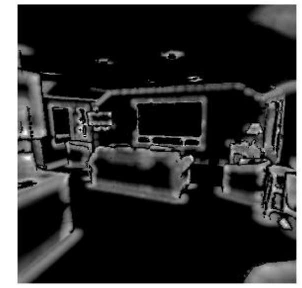


2D Edges

3D (Occlusion) Edges

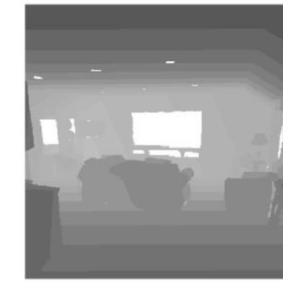


2D Keypoints



3D Keypoints

Image Reshading

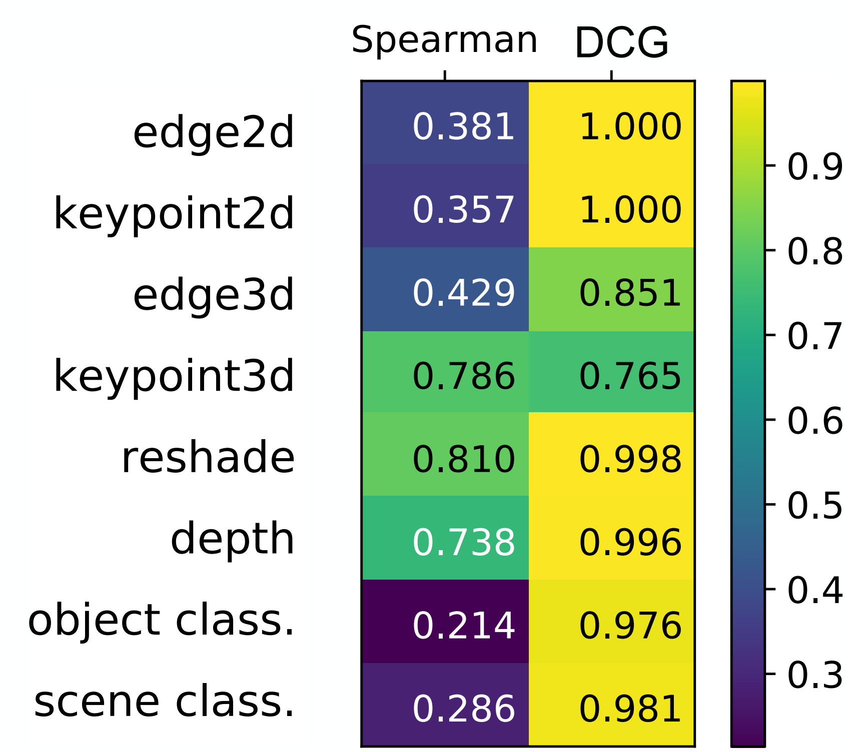


Depth

Comparison with Task Affinity Score on 8 vision tasks.

- > 6 times faster
- top three most transferable source tasks are consistent with Task Affinity on most target tasks

easy-to-compute, efficient transferability metric with strong operational meaning!



Rank Comparison

Other Analytical Transferability Metrics

- Transferability metrics for different transfer settings

Algorithm	Different Tasks $P(Y_S \mid X_S) \neq P(Y_T \mid X_T)$	Different Instance $X_S \neq X_T$	Different Domain $P(X_S) \neq P(X_T)$
NCE (Tran et al. 2019) *	✓	×	×
H-Score (Bao et al. 2019)	✓	✓	×
LEEP (Nguyen et al. 2020)**	✓	✓	×
OTCE (Y. Tan et al. *under review*)	✓	✓	✓

* Anh T Tran, Cuong V Nguyen, and Tal Hassner. Transfer- ability and hardness of supervised classification tasks. ICCV, 2019.

** Cuong V Nguyen, Tal Hassner, Cedric Archambeau, and 952 Matthias Seeger. Leep: A new measure to evaluate trans- 953 ferability of learned representations.ICML, 2020.

Today's Talk

- What's Transfer Learning
- Transfer Learning Techniques
  - Task transfer learning
  - Domain adaptation
  - Transfer bound on domain adaptation
- How to avoid negative transfer?
  - Case study on feature transferability
  - Task transferability: empirical and theoretical methods
- Discussions and Q&A

Open Theoretical Questions

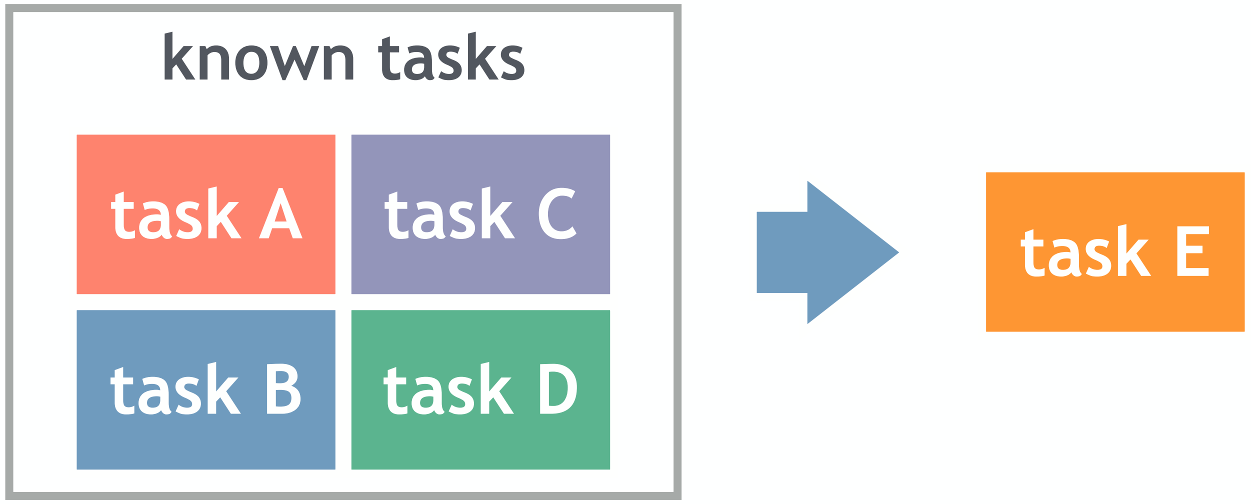
Can we find a transferability metric that …

- accounts for domain difference
- depends on target sample-size
  - Rademacher complexity for computable transfer bound (Maurer 2009)
- depends on learning algorithm
  - Kolmogorov complexity-based task relatedness (Mahmud 2007)

Beyond Transfer Learning

- **Multi-source transfer learning**: how to efficiently, adaptively combine features from multiple source tasks in transfer learning?

- **Meta learning:** given data/experience on previous tasks, learn a new task more quickly
  - transfer learning is one common approach in meta learning



challenge: efficient meta learning for heterogeneous tasks

References & Resources

Book

- Yang, Q et al. (2019). Transfer Learning. Cambridge University Press

Survey papers

- Tan, C., Sun, F., Kong, T., Zhang, W., Yang, C., & Liu, C. (2018). A survey on deep transfer learning. Lecture Notes in Computer Science, 11141 LNCS, 270–279.
- Lisa Torrey and Jude Shavlik (2009). Transfer learning. Handbook of Research on Machine Learning Applications
- Pan, S.J., Yang, Q. (2010). A survey on transfer learning. IEEE Transactions on knowledge and data engineering 22(10), 1345–1359

Related web links:

- An Information-Theoretic Metric for Task Transfer Learning: http://yangli-feasibility.com/home/ttl.html
- Disentangling Task Transfer Learning: http://taskonomy.stanford.edu/

Thank You!