# Transfer Reinforcement Learning

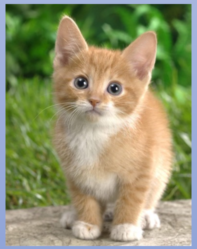
王志　南京大学

2025年04月28日

# Content

# 1. Transfer learning

Dog/Cat
Classifier



cat

dog

Data ***<u>not directly related to</u>*** the task considered

elephant

tiger

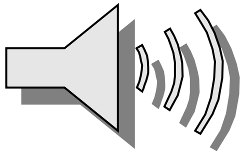
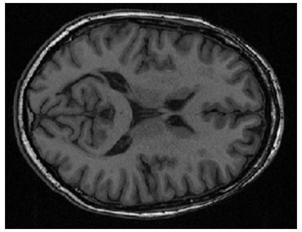
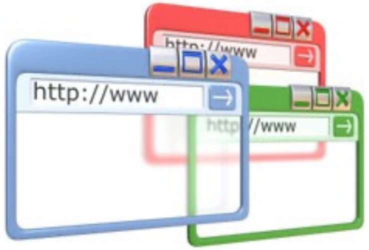
Similar domain, different tasks

dog

cat

Different domains, same task

# 1. Transfer learning

| Task Considered | | Data not directly related | |
|---|---|---|---|
| Speech Recognition |  Chinese | YouTube | English<br>Korean<br>...... |
| Image Recognition |  Medical Images |  | |
| Text Analysis |  Specific domain |  | Webpages |

## Transfer Learning - Overview

| | | Source Data (not directly related to the task): labelled | Source Data (not directly related to the task): unlabeled |
|---|---|---|---|
| Target Data | labelled | Model Fine-tuning<br>Multitask Learning | Self-taught learning<br>Rajat Raina , Alexis Battle , Honglak Lee , Benjamin Packer , Andrew Y. Ng, Self-taught learning: transfer learning from unlabeled data, ICML, 2007 |
| Target Data | unlabeled | Domain-adversarial training<br>Zero-shot learning | Self-taught Clustering<br>Wenyuan Dai, Qiang Yang, Gui-Rong Xue, Yong Yu, "Self-taught clustering", ICML 2008 |

Different from semi-supervised learning

# Model Fine-tuning

Task description

Source data: $(x^s, y^s)$ ⬅ A large amount

Target data: $(x^t, y^t)$ ⬅ Very little

One-shot learning: only a few examples in target domain

Example: (supervised) speaker adaption

Source data: audio data and transcriptions from many speakers

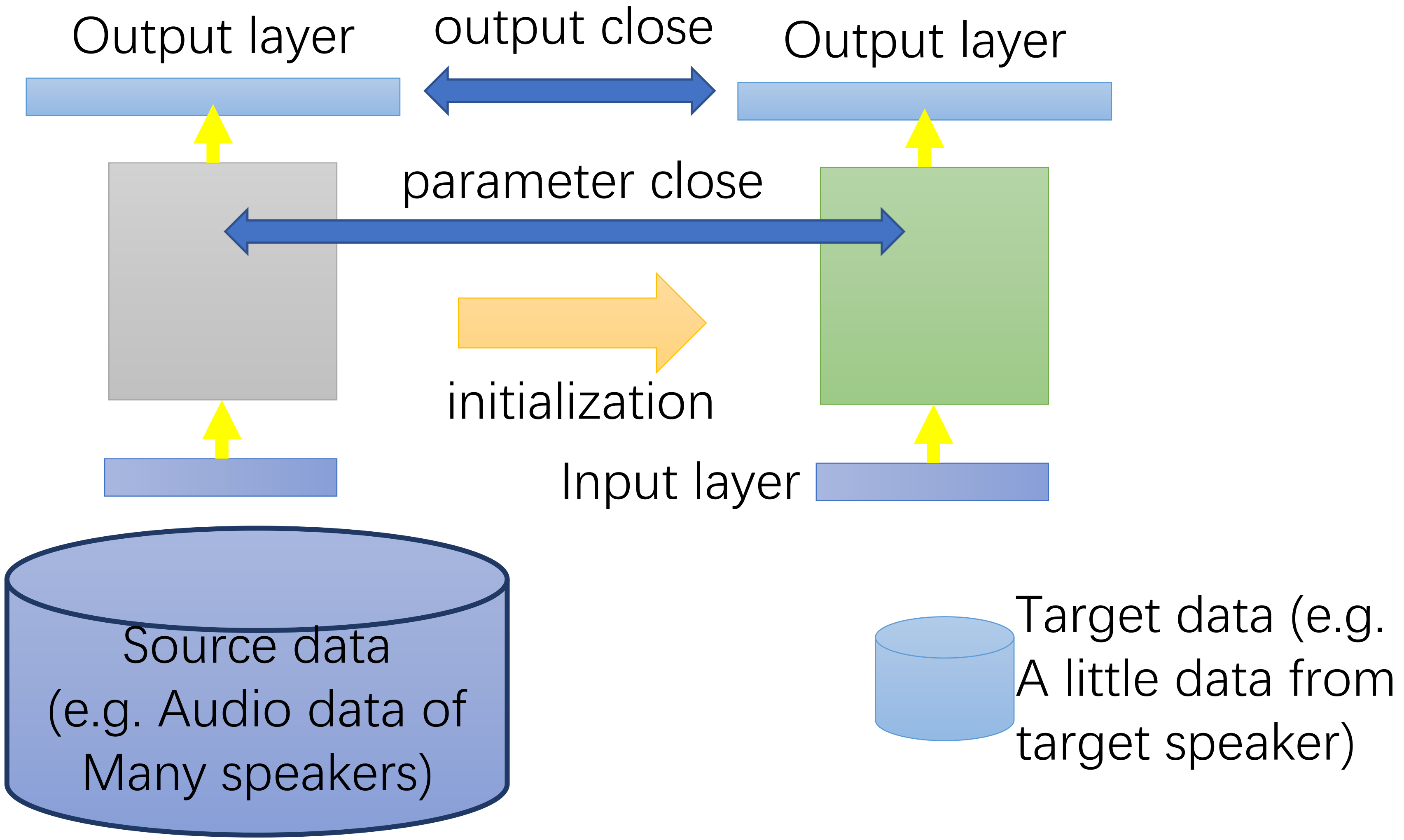
Target data: audio data and its transcriptions of specific user

Idea: training a model by source data, then fine-tune the model by target data

Challenge: only limited target data, so be careful about overfitting

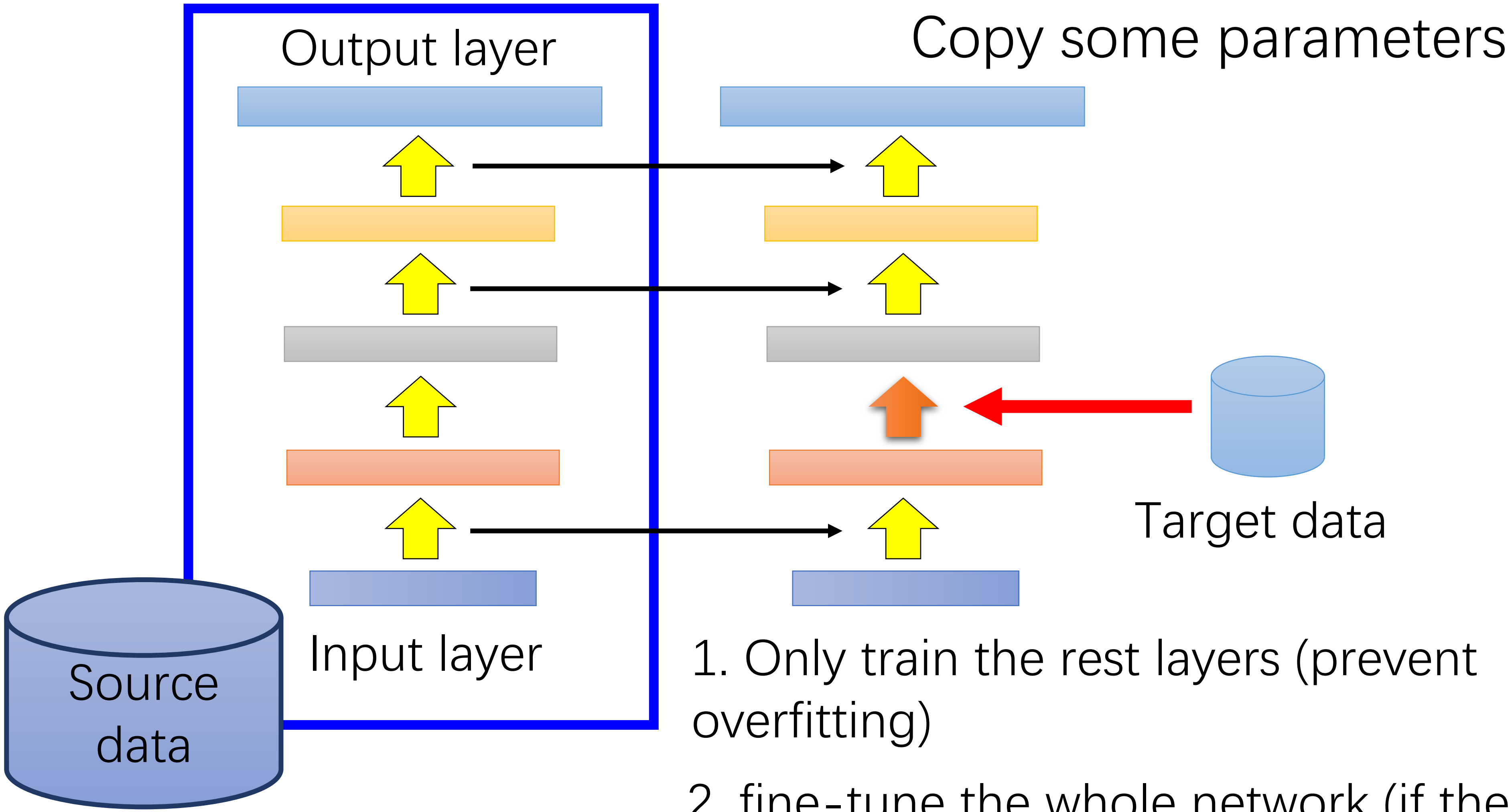
# Model Fine-tuning

## Conservative Training



# Model Fine-tuning

## Layer Transfer

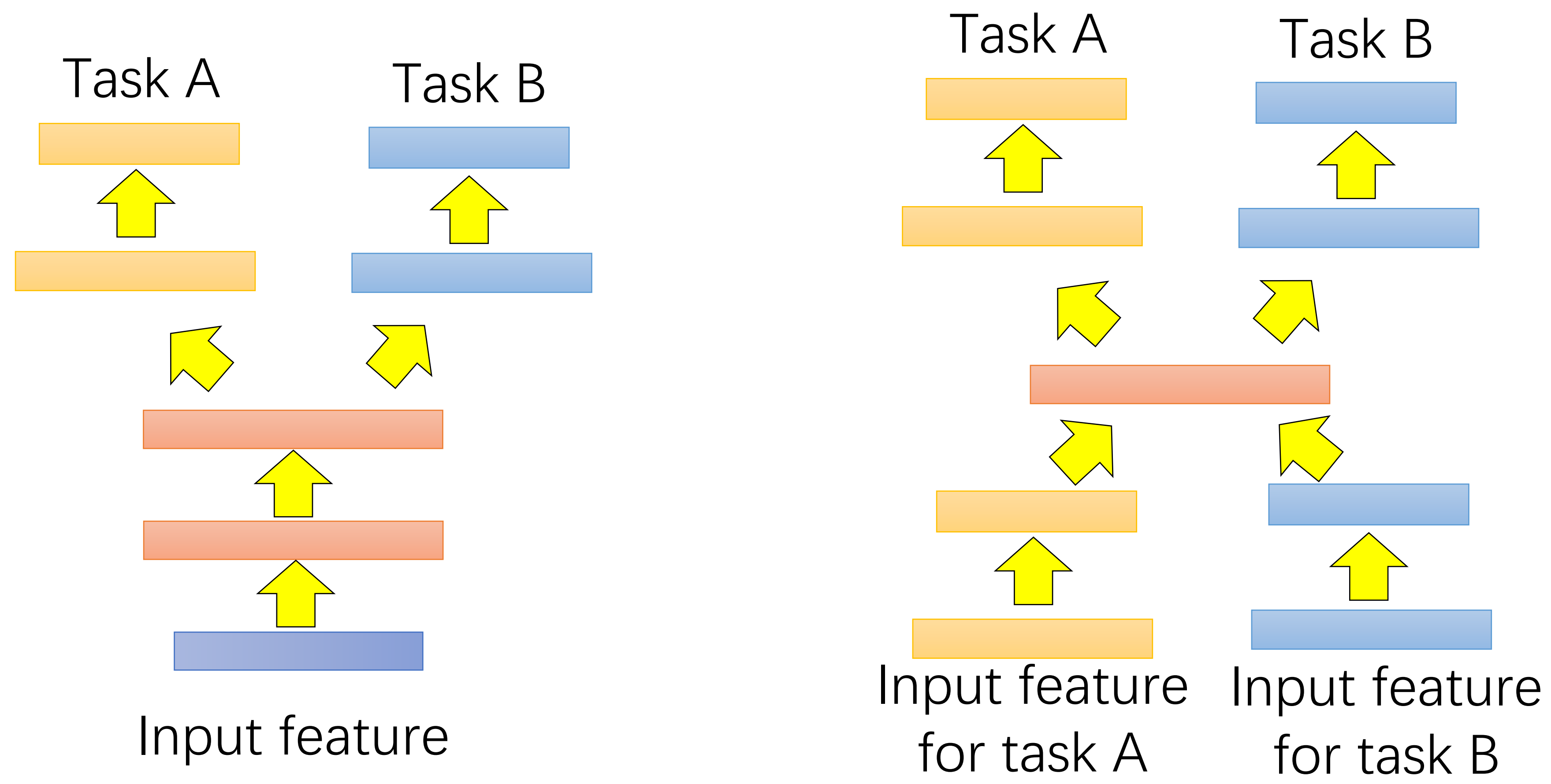



1. Only train the rest layers (prevent overfitting)

2. fine-tune the whole network (if there is sufficient data)

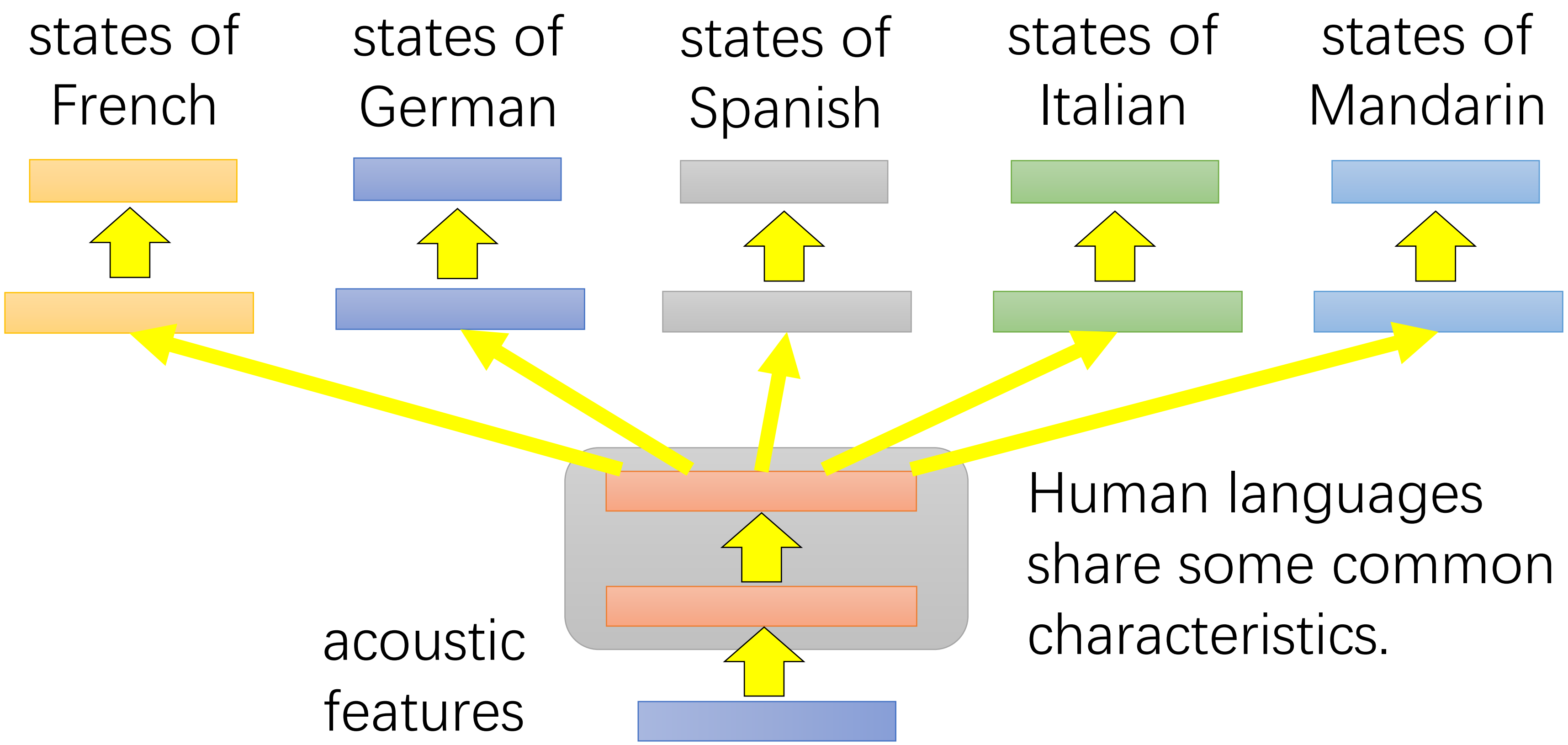
# Multitask Learning

The multi-layer structure makes NN suitable for multitask learning



# Multitask Learning

## Multitask Learning
## - Multilingual Speech Recognition




Human languages share some common characteristics.

***Similar idea in translation***: Daxiang Dong, Hua Wu, Wei He, Dianhai Yu and Haifeng Wang, "Multi-task learning for multiple language translation.", ACL 2015

# Domain-adversarial training

## Task description

Source data: $(x^s, y^s)$ → Training data
Target data: $(x^t)$ → Testing data

Same task, mismatch

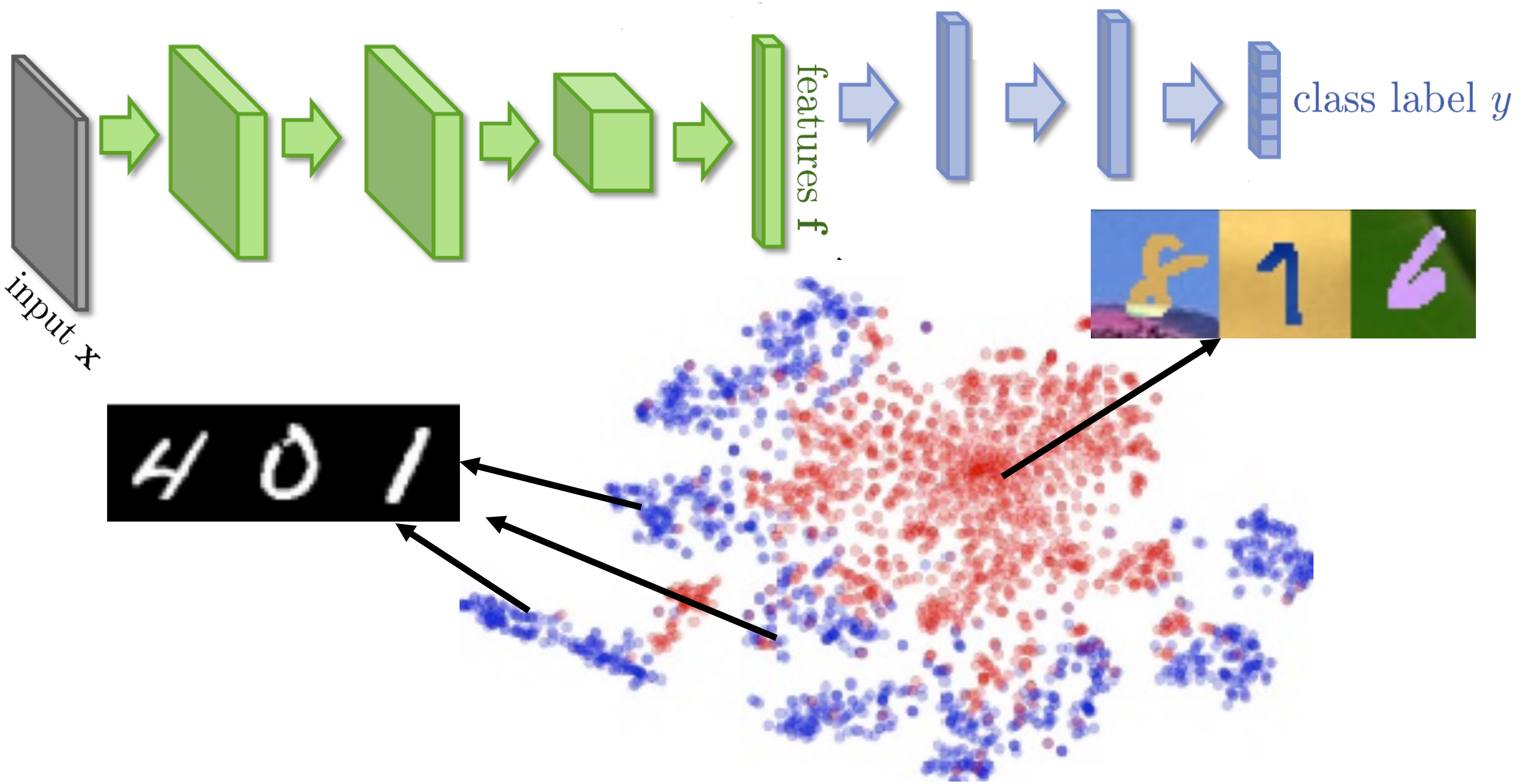
MNIST

SOURCE — with label

TARGET — without label

MNIST-M

# Domain-adversarial training

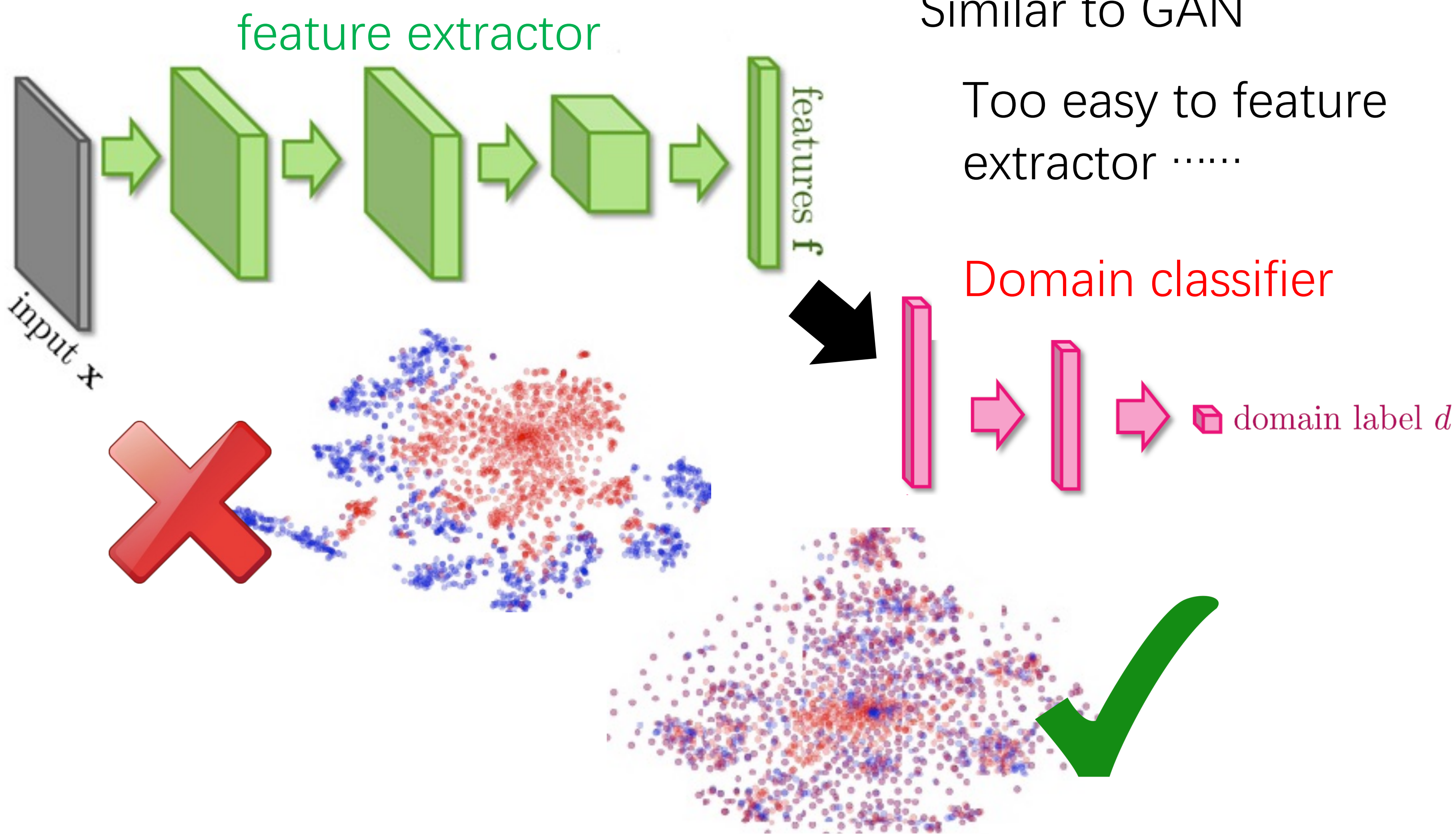


# Domain-adversarial training

## Domain-adversarial training

Similar to GAN

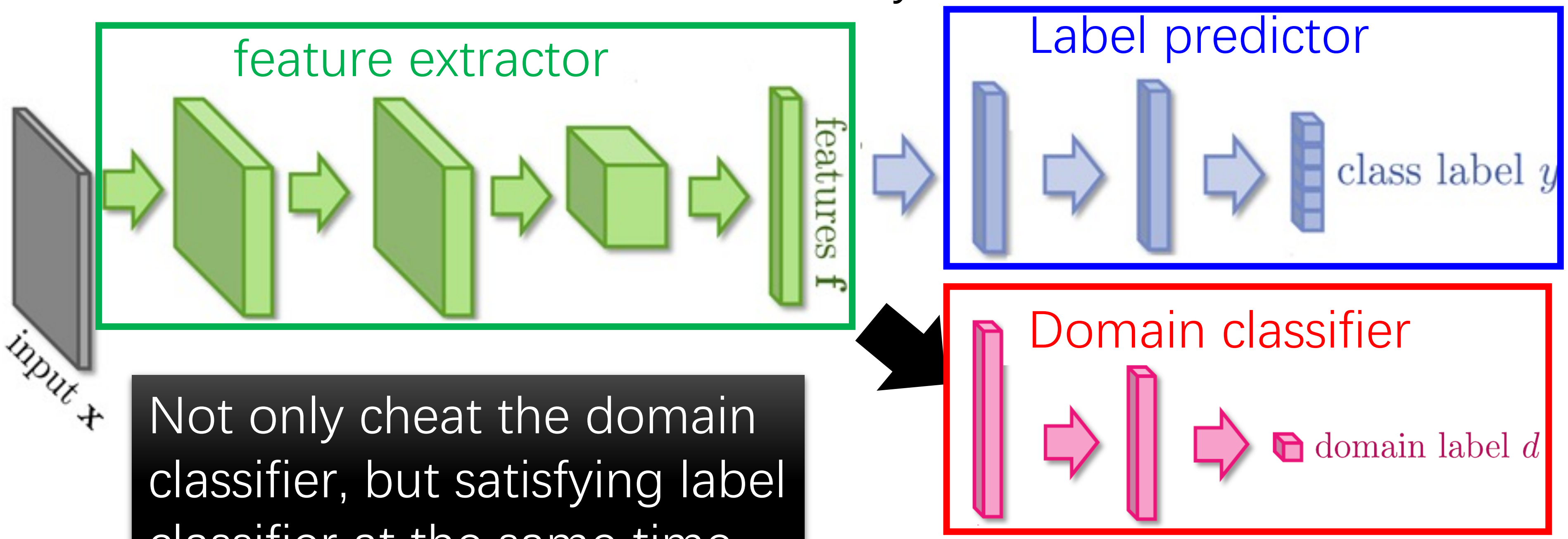
Too easy to feature extractor ……



# Domain-adversarial training

Maximize label classification accuracy + minimize domain classification accuracy

Maximize label classification accuracy




Not only cheat the domain classifier, but satisfying label classifier at the same time

Minimize domain classification accuracy

This is a big network, but different parts have different goals.

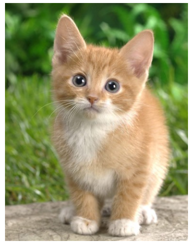
# Zero-shot Learning

Source data: $(x^s, y^s)$ → Training data

Target data: $(x^t)$ → Testing data

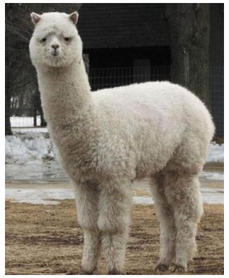
Different tasks

$x^s$:



……

$x^t$:



$y^s$: cat dog ……

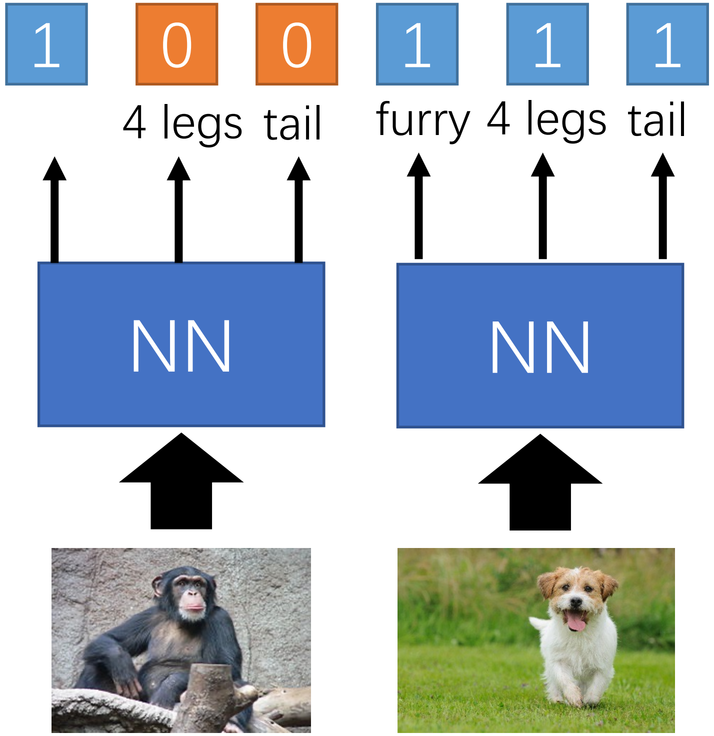
In image classification, we can not have all possible class in the source (training) data.

How we solve this problem in image classification?

# Zero-shot Learning

## Representing each class by its attributes

***Training***




Database

attributes

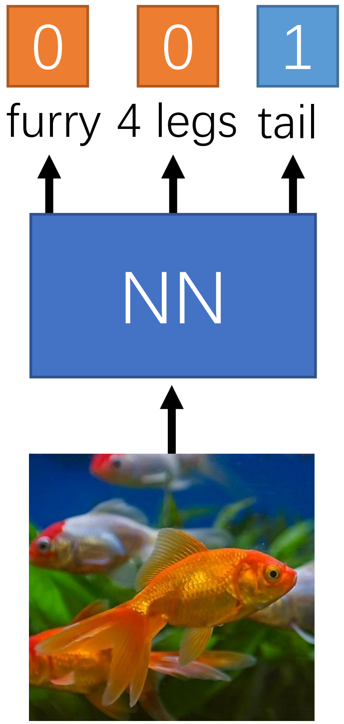
class

| | furry | 4 legs | tail | … |
|---|---|---|---|---|
| Dog | O | O | O | |
| Fish | X | X | O | |
| Chimp | O | X | X | |
| … | | | | |

sufficient attributes for one to one mapping

# Representing each class by its attributes

***Testing***




Find the class with the most similar attributes

class

attributes

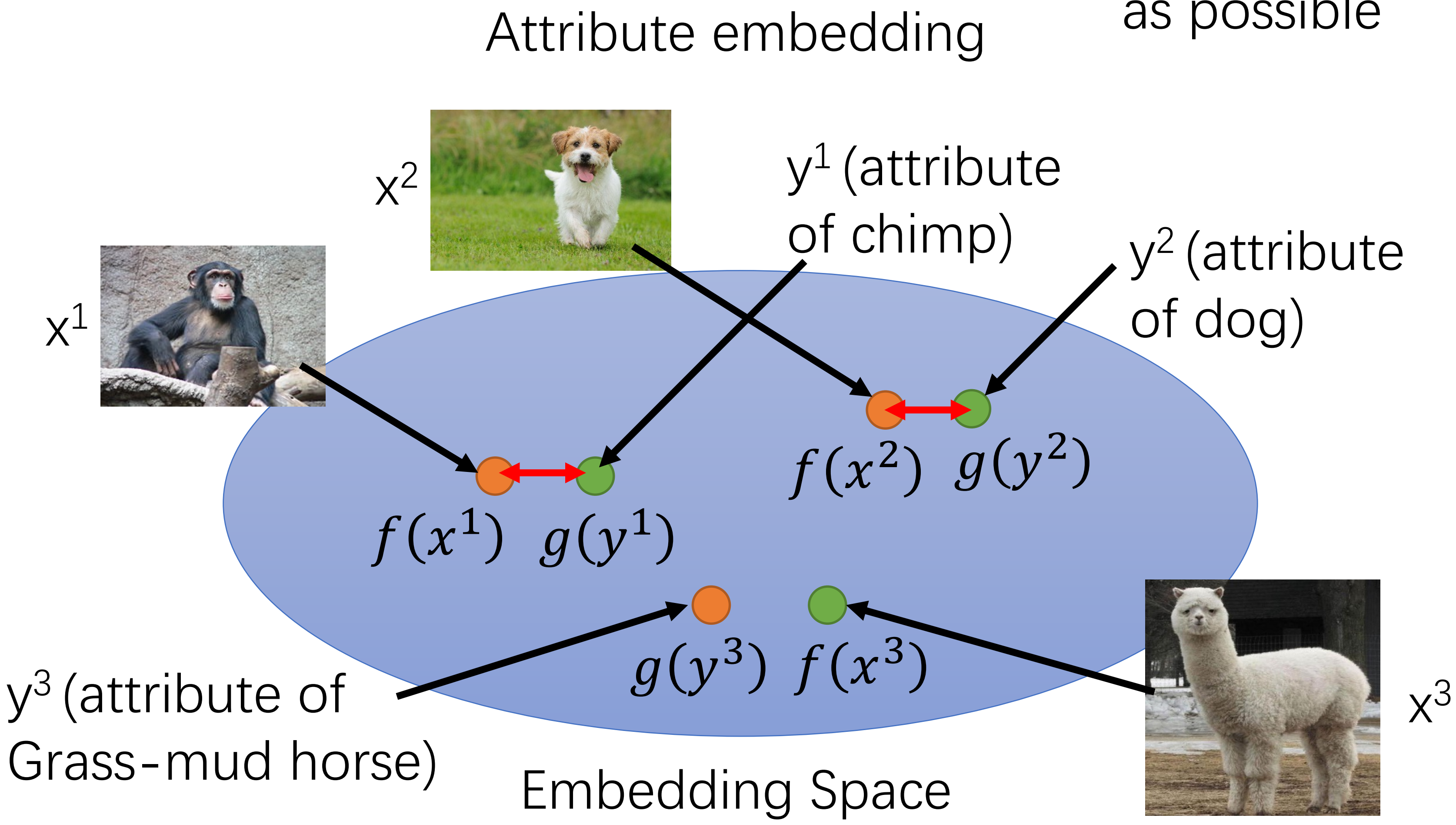
| | furry | 4 legs | tail | … |
|---|---|---|---|---|
| Dog | O | O | O | |
| Fish | X | X | O | |
| Chimp | O | X | X | |
| … | | | | |

sufficient attributes for one to one mapping

# Zero-shot Learning

$f(*)$ and $g(*)$ can be NN.

Training target:
$f(x^n)$ and $g(y^n)$ as close as possible



# Self-taught learning

Learning to extract better representation from the source data (unsupervised approach)

Extracting better representation for target data

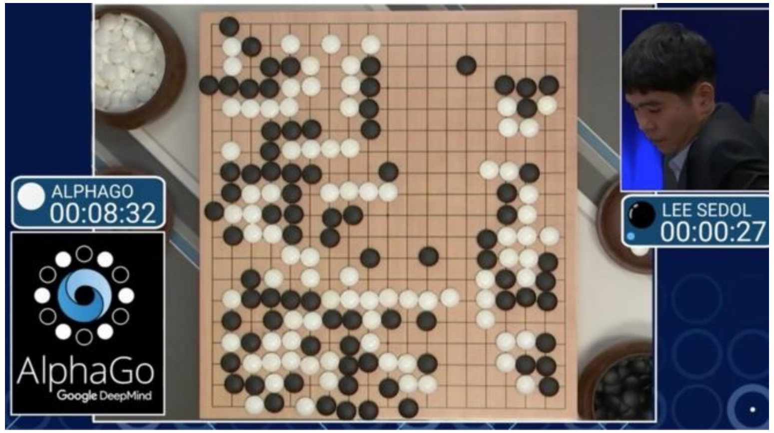
| Domain | Unlabeled data | Labeled data | Classes | Raw features |
|---|---|---|---|---|
| Image classification | 10 images of outdoor scenes | Caltech101 image classification dataset | 101 | Intensities in 14x14 pixel patch |
| Handwritten character recognition | Handwritten digits ("0"–"9") | Handwritten English characters ("a"–"z") | 26 | Intensities in 28x28 pixel character/digit image |
| Font character recognition | Handwritten English characters ("a"–"z") | Font characters ("a"/"A" – "z"/"Z") | 26 | Intensities in 28x28 pixel character image |
| Song genre classification | Song snippets from 10 genres | Song snippets from 7 *different* genres | 7 | Log-frequency spectrogram over 50ms time windows |
| Webpage classification | 100,000 news articles (Reuters newswire) | Categorized webpages (from DMOZ hierarchy) | 2 | Bag-of-words with 500 word vocabulary |
| UseNet article classification | 100,000 news articles (Reuters newswire) | Categorized UseNet posts (from "SRAA" dataset) | 2 | Bag-of-words with 377 word vocabulary |

# Content

1. Transfer learning
2. Transfer Reinforcement Learning
3. Concluding Remarks

# Transfer Reinforcement Learning： why ?

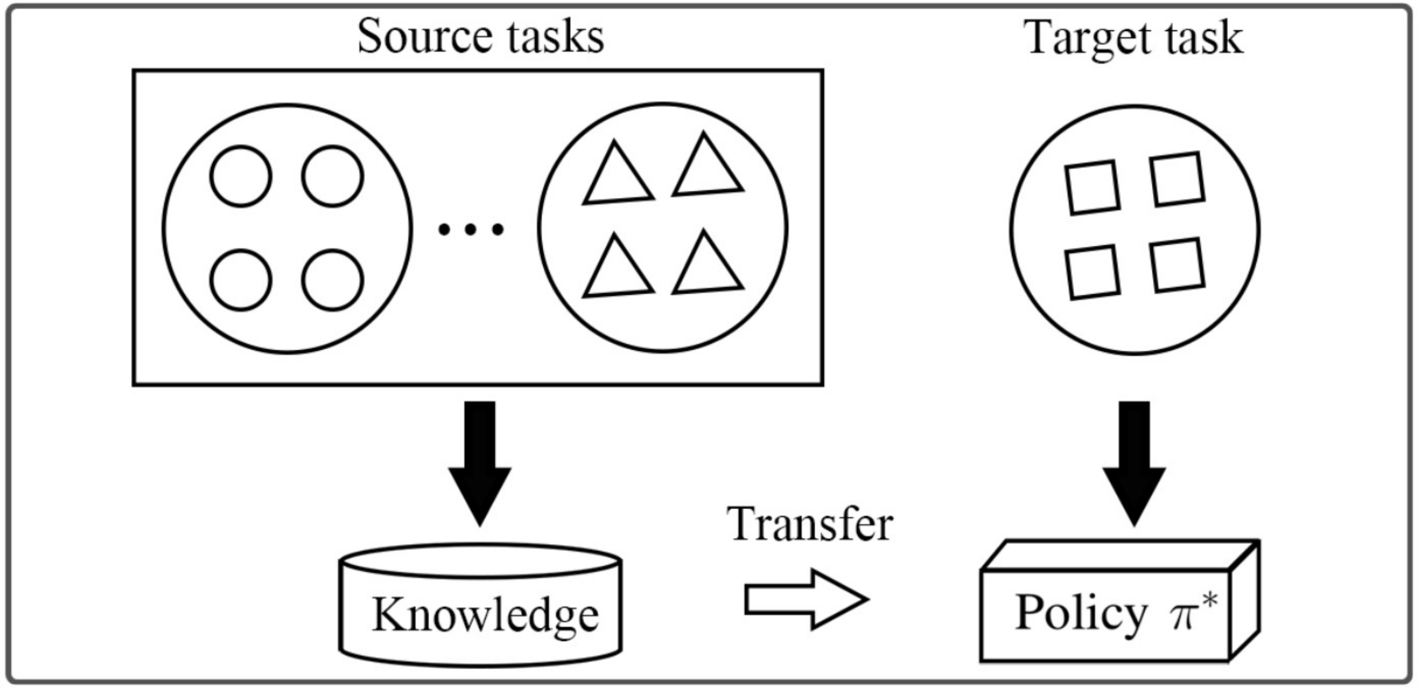
- Deep Reinforcement Learning(DRL)
  - sensitive to the hyper-parameters
  - require numerous samples




# Transfer Reinforcement Learning

Given a set of source domains and a target domain

Transfer Learning aims to learn an optimal policy $\pi^*$ for the target domain, by leveraging knowledge from source tasks

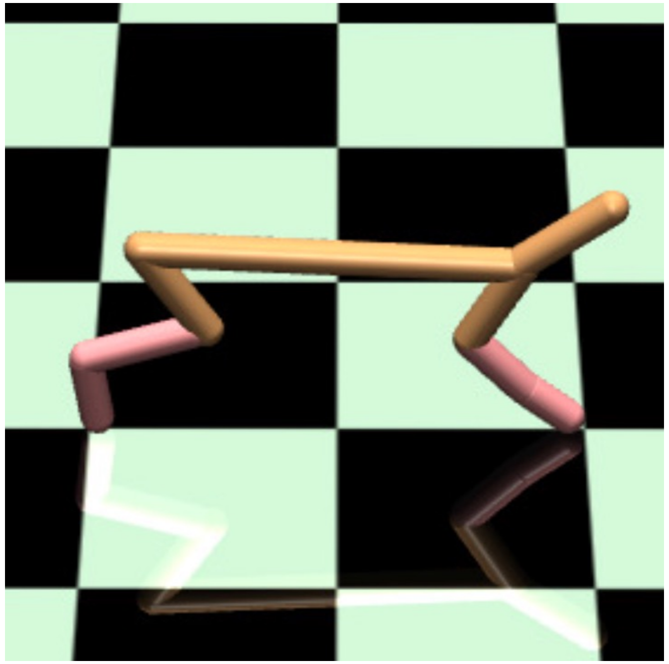


# Transfer Reinforcement Learning

- Categorization of Transfer Learning Approaches
  - What knowledge is transferred?
  - What RL frameworks are compatible with the transfer learning approach?
  - What is the difference between the source and the target domain?
  - What information is available in the target domain?
  - How sample-efficient the transfer learning approach is ?
  - What are the goals of transfer learning ?

# Case Analysis of Transfer Learning: HalfCheetah

- Potential Domain Differences:
  - S (State-space)
  - A (Action-space)
  - R (Reward function)
  - T (Transition dynamics)
  - $\mu_0$ (Initial states)
  - $\tau$ (Trajectories)



# Case Analysis of Transfer Learning: HalfCheetah

- Transferrable Knowledge:
  - Demonstrated trajectories
  - Model dynamics
  - Teacher policies
  - Teacher value functions

# Case Analysis of Transfer Learning: HalfCheetah

- Evaluation metrics:
  - Jumpstart performance( jp)
  - Asymptotic performance (ap)
  - Accumulated rewards(ar)
  - Transfer ratio(tr)
  - Time to threshold(tt)
  - Performance with fixed training epochs(pe)
  - Performance sensitivity (ps)

# TRANSFER LEARNING APPROACHES

- Reward Shaping(RS):

  - RS learns a reward-shaping function $\mathcal{F} : S \times S \times A \rightarrow \mathbb{R}$

  - agent will learn its policy using the newly shaped rewards $\mathcal{R}' = \mathcal{R} + \mathcal{F}$

  - RS has altered the target domain with a different reward function:

$$\mathrm{M} = (S, A, \mathrm{T}, \gamma, \mathcal{R}) \rightarrow \mathrm{M}' = (S, A, \mathrm{T}, \gamma, \mathcal{R}')$$

# TRANSFER LEARNING APPROACHES

- Learning from Demonstrations:

In general, learning from demonstrations (LfD) is a technique to assist RL by utilizing provided demonstrations for more efficient exploration.

Knowledge conveyed in demonstrations encourages agents to explore states which can benefit their policy learning.

Value function, Policy, Transition dynamics, Representation …

Policy: $$\mathcal{L}\left(\pi, \pi_E\right)=\frac{1}{N_E} \sum_{i=1}^{N_E} \mathbb{1}\left\{\pi_E\left(s_i\right) \neq \pi\left(s_i\right)\right\}$$

# TRANSFER LEARNING APPROACHES

## ➢ Policy Transfer:

policy transfer, where the external knowledge takes the form of pretrained policies from one or multiple source domains $\{\pi_{E_i}\}_{i=1}^{K}$ .

Transfer Learning via Policy Distillation

$$\min_{\theta} \mathbb{E}_{\tau\sim\pi_E}\left[\sum_{t=1}^{|\tau|} \nabla_\theta \mathcal{H}^{\times}\left(\pi_E\left(\tau_t\right) \mid \pi_\theta\left(\tau_t\right)\right)\right]$$

Transfer Learning via Policy Reuse

$$P\left(\pi_{E_i}\right)=\frac{\exp_1^1\left(tW_i\right)^1}{\sum_{j=0}^{K}\exp\left(tW_j\right)},$$

# TRANSFER LEARNING APPROACHES

➢ Inter-Task Mapping：

utilize mapping functions between the source and the target domains to assist knowledge transfer. (1) which domain does the mapping function apply to, and (2) how is the mapped representation utilized.

One-to-one mappings exist between the source domain Ms and the target domain Mt.

$$X_S\left(\mathcal{S}^t\right) \rightarrow \mathcal{S}^s, X_A\left(\mathcal{A}^t\right) \rightarrow \mathcal{A}^s$$

tackles the inter-task mapping problem by automatically learning a mapping function

$$r'(s, \cdot) = \alpha \left\| f\left(s_{agent}^s; \theta_f\right) - g\left(s_{agent}^t; \theta_g\right) \right\|$$

# TRANSFER LEARNING APPROACHES

- Representation Transfer :

transfer knowledge are feature representations, such as representations learned for the value function or Q-function.

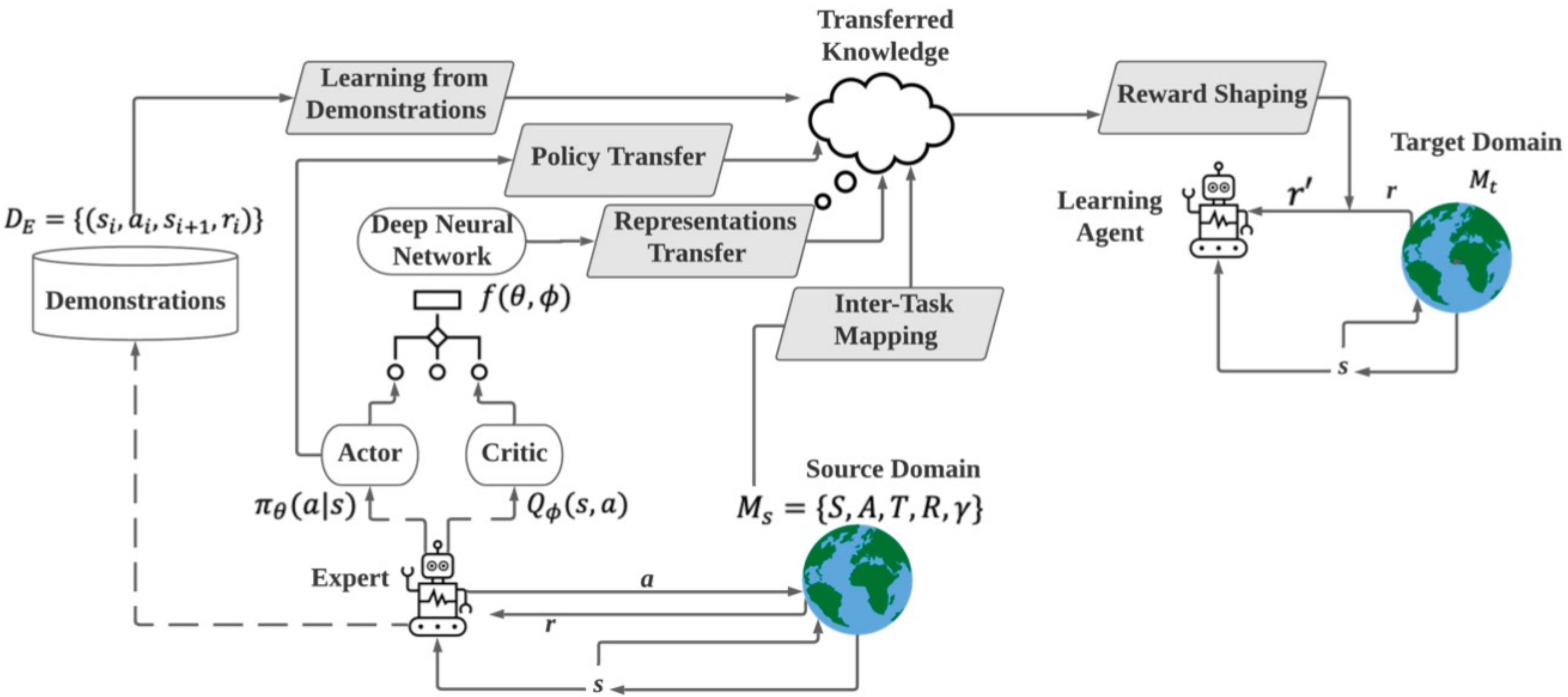
Reusing Representations: progressive network, PathNet, modular networks …

$$h_i^{(k)} = f\left(W_i^{(k)} h_{i-1}^{(k)} + \sum_{j<k} U_i^{(k:j)} h_{i-1}^{(j)}\right) \qquad \pi(s) := \phi\left(s_{env}, s_{\text{agent}}\right) = f_r\left(g_k\left(s_{env}\right), s_{\text{agent}}\right)$$

Disentangling Representations : Successor Representations (SR) …

$$V^\pi(s) = \sum_{s'} \psi\left(s, s'\right) \mathbf{w}\left(s'\right)$$

# TRANSFER LEARNING APPROACHES



# Content

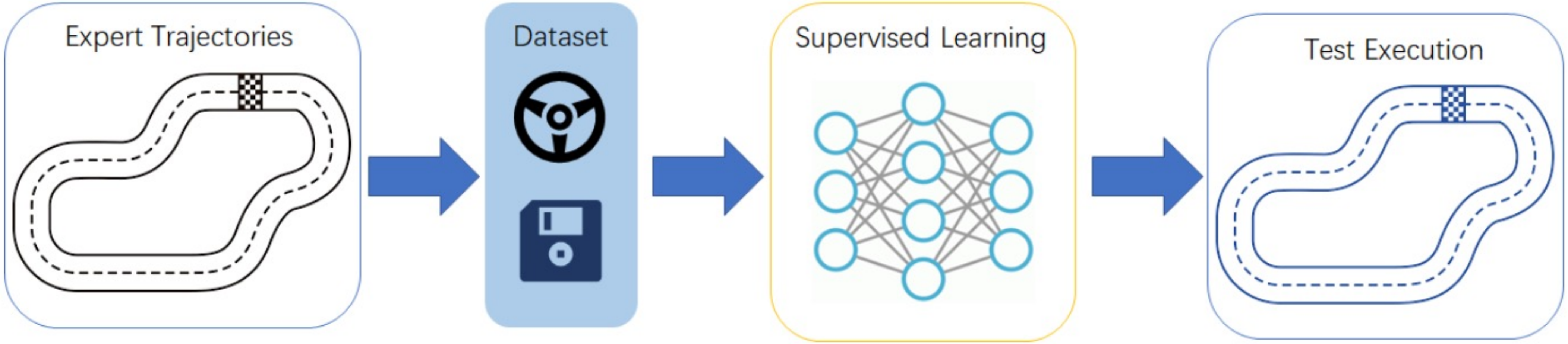
1. Transfer learning
2. Transfer Reinforcement Learning
3. Concluding Remarks

# Imitation Learning

Imitation Learning aims to train a policy to mimic the behavior of an expert policy.

imitation learning ⇔ Demonstrations (LfD)

LfD still interacts with the domain to access reward signals



# Lifelong Learning

Lifelong Learning, or Continual Learning, refers to the ability to learn multiple tasks that are temporally or spatially related.

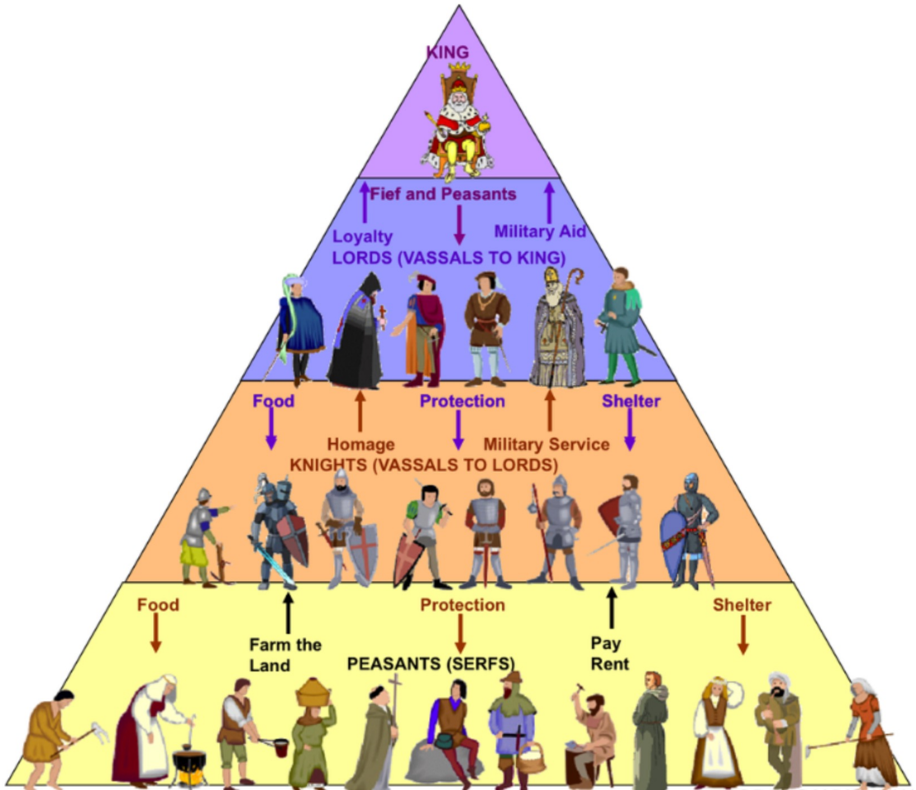
tradeoff between obtaining new information over time and retaining the previously learned knowledge across new tasks.

the ability of automatic task detection also be a requirement for Lifelong Learning

# Hierarchical RL

the action space is grouped into different granularities to form higher-level macro actions

Given the higher-level abstraction on tasks, actions, and state spaces, hierarchical RL can facilitate knowledge transfer across similar domains.



# Multi-Agent RL

multi-agent RL considers an MDP with multiple agents acting simultaneously in the environment.

Approaches of knowledge transfer for Multi-agent RL fall into two classes: inter-agent transfer and intra-agent transfer.