Chapter 15
Representation Learning

郑华滨

Content

- 1. Introduction
- 2. Unsupervised Representation Learning
- 3. Supervised Representation Learning
- 4. What is a Good Representation?

Introduction

- AI tasks can be very difficult / easy depending on how data is represented

- e.g. Roman numerals / Arabic numerals

- e.g. coffee bean / ground coffee

- Definition: a good representation is one that makes a subsequent learning task easier

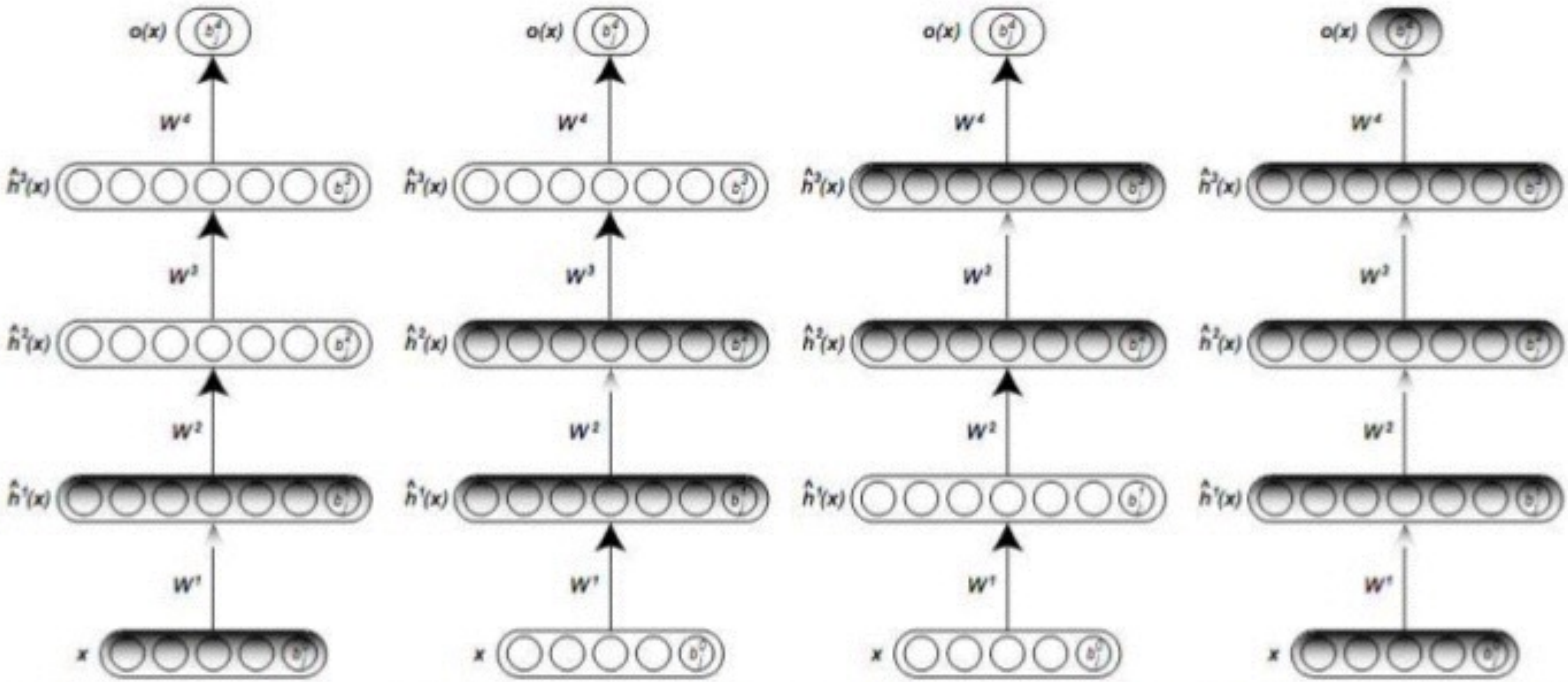
- Deep neural networks overwhelm shallow models by implicitly learn a hierarchical representation
  - Layer N: linear classifier
  - Layer N - 1: linear-separable representation (ideally)
  - Layer N - 2: not-so linear-separable representation
  - …
  - Layer 0: raw representation

- Both labeled / unlabeled data can be used
- Labeled data: supervised representation learning
- Unlabeled data: unsupervised representation learning

- 2. Unsupervised Representation Learning (Sec.15.1)
    - ancient approach
    - middle-ages approach
    - when and why does pre-training work?
    - modern approach

Ancient Approach

- Greedy layer-wise unsupervised pre-training
- Proposed because training DNN is difficult
- Largely abandoned today



(a) First hidden layer pre-training (b) Second hidden layer pre-training (c) Third hidden layer pre-training (d) Fine-tuning of whole network

each layer can be RBM, auto-encoder, …

Mid-Ages Approach

- greedy layer-wise training —> end-to-end training
  - with the rise of ReLU, Batch Normalization, …
- pre-train —> jointly-train
  - makes the unsupervised part aware of the supervised object

When & Why Works

- Unsupervised pre-training is sometimes helpful but sometimes harmful, why?

- If we know why it works, we can guess when it will & won’t work

- Focus on pre-training approach, comparing with jointly-training approach

When & Why Works

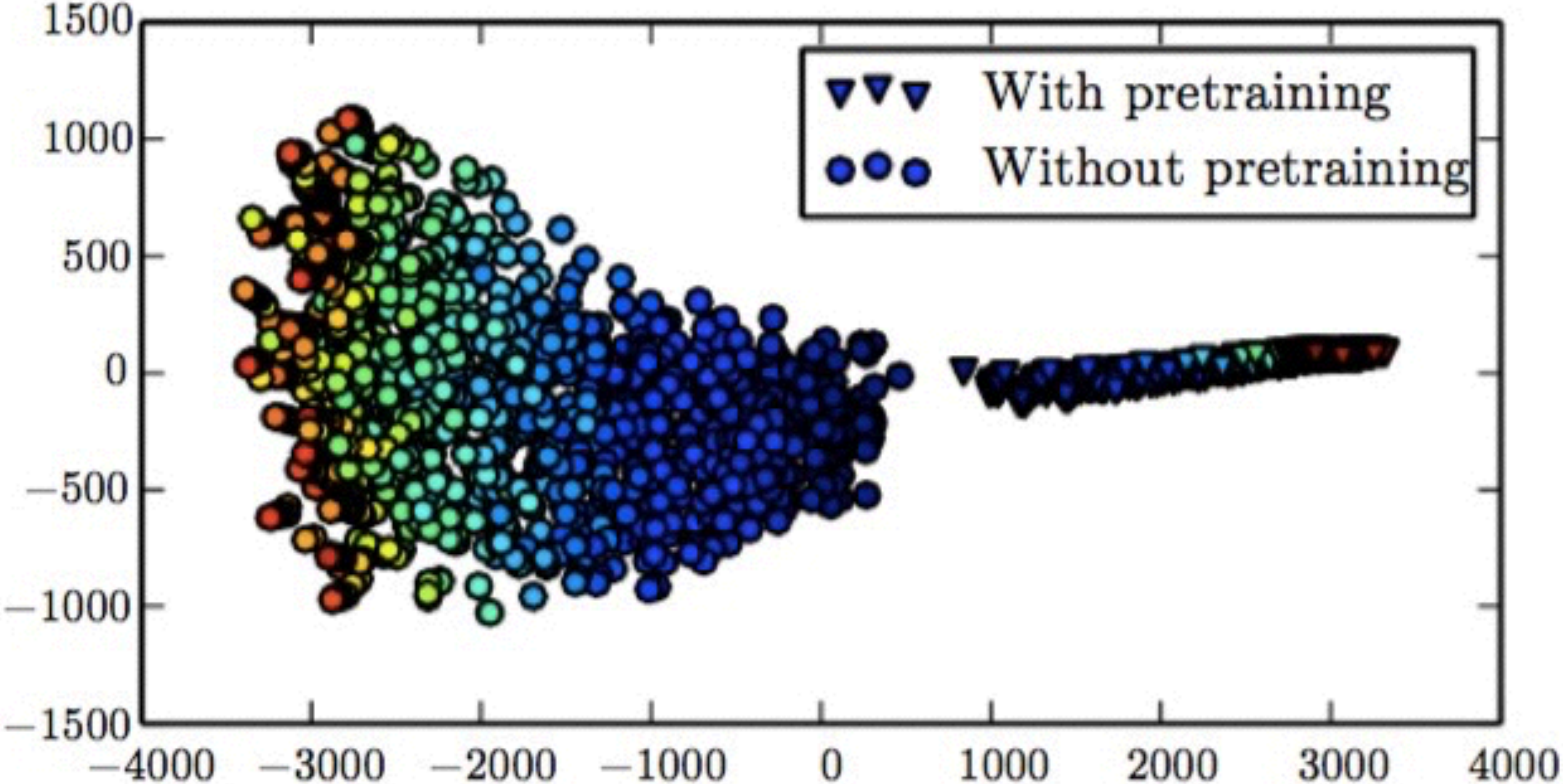
- 1. Learn a good representation (key idea)
- 2. As a regularization
- 3. Better parameter initialization

When & Why Works

- 1. Learn a good representation
  - **Why**: features useful for unsupervised object may also be useful for supervised object (e.g. low level feature of image)
  - **When**: raw representation is bad **and** not enough labeled data to learn a good representation (e.g. word vector)
  - The "why" above may not be true, that's one reason to prefer jointly-training, in which unsupervised object has more chance to learn features useful for supervised object

When & Why Works

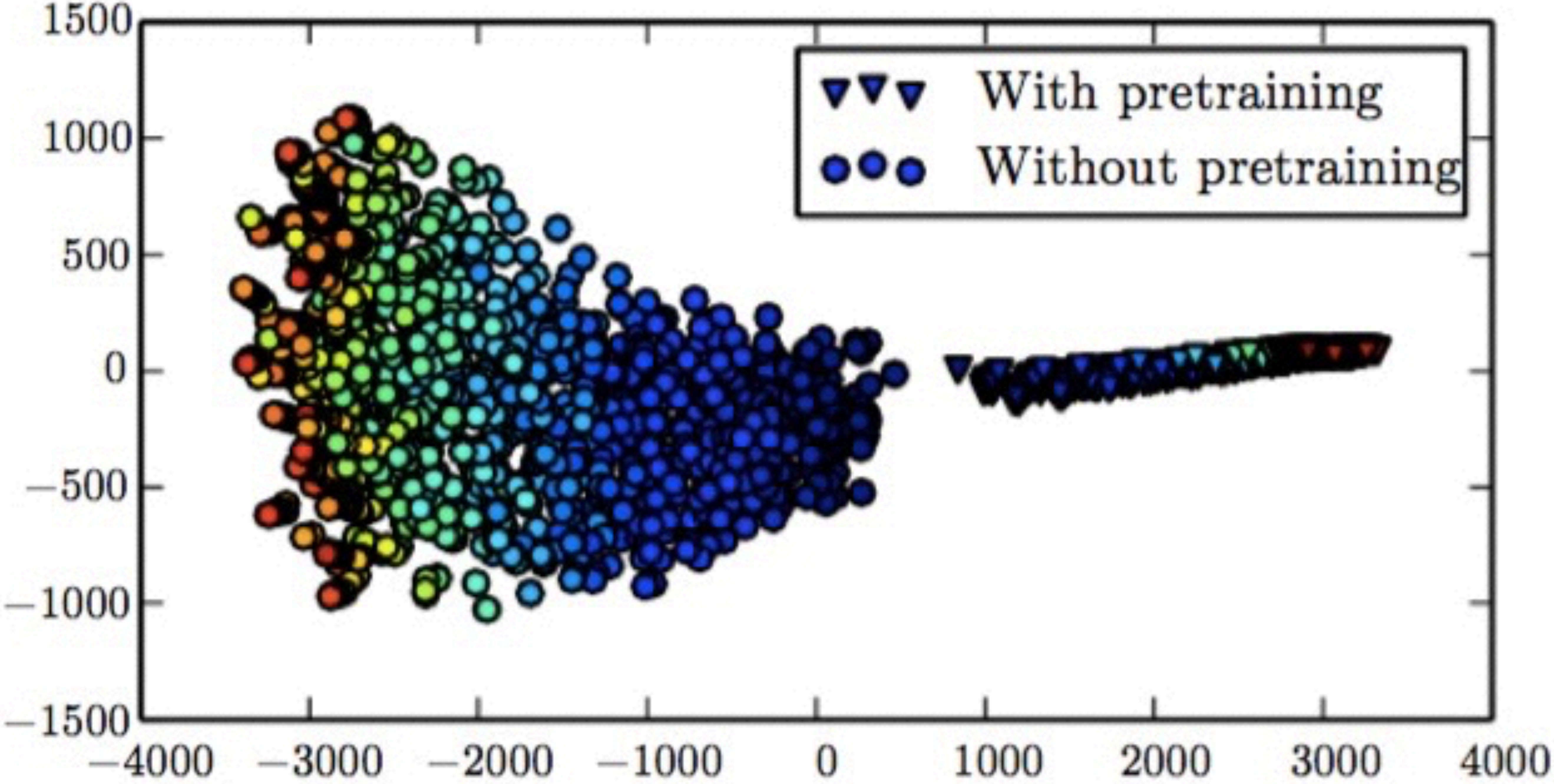
- 2. As a regularizer
  - **Why**: bias part of the model towards one that can achieve the unsupervised object, reduce the number of possible models
  - Different with weight decay (L1/L2 norm), which bias the model towards a simple one
  - **When**: not enough labeled data so a regularizer is needed **but** the model to be learn is so complicated that weight decay doesn't make sense



With pretraining
Without pretraining
1500
1000
500
0
−500
−1000
−1500
−4000
−3000
−2000
−1000
0
1000
2000
3000
4000

When & Why Works

- 3. Better parameter initialization
    - **Why**: make it possible to reach the region in parameter space that is impossible to reach given only supervised object
    - Not well understood yet, can't say much about this idea

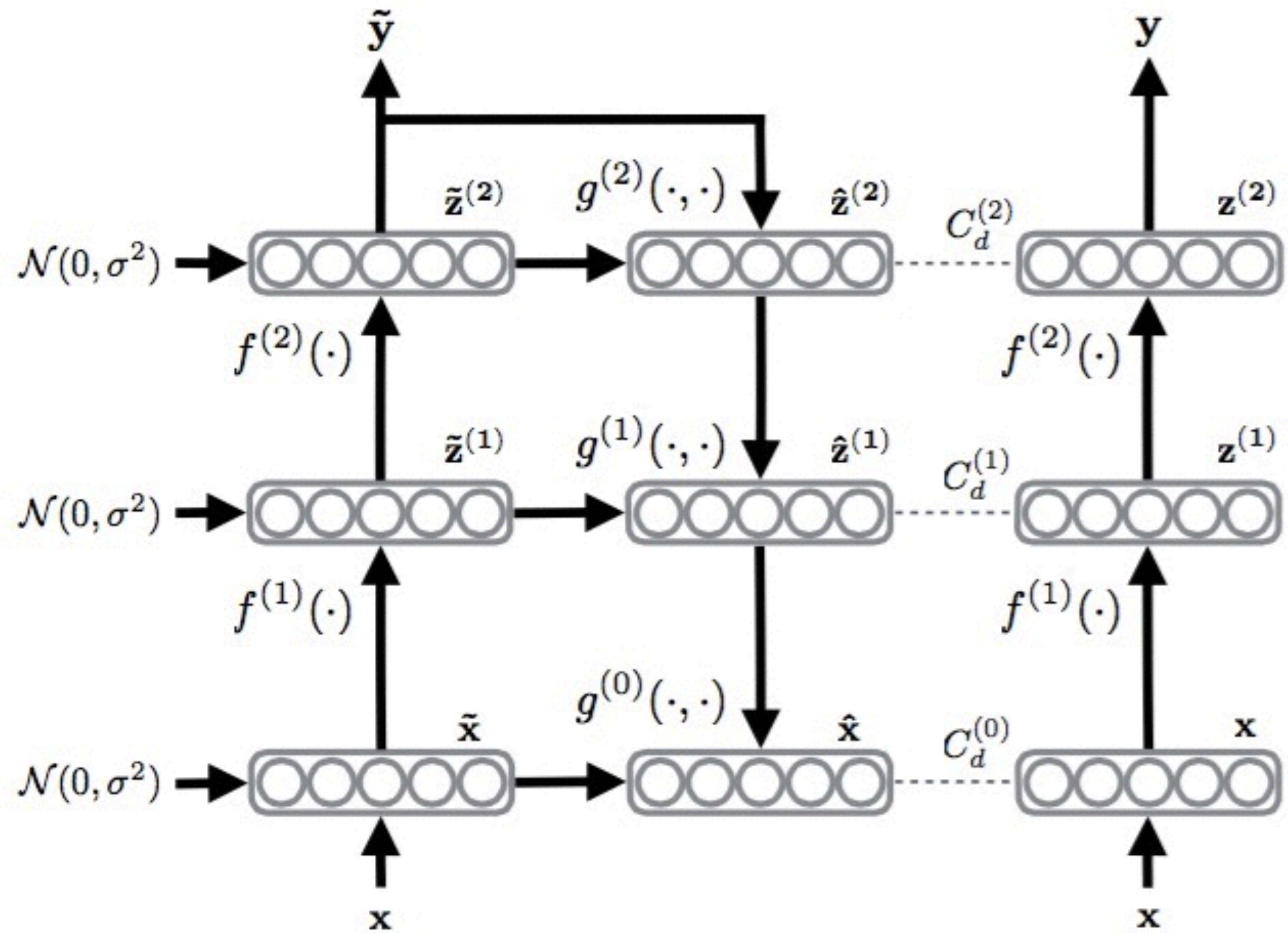

1500
1000
500
0
−500
−1000
−1500
−4000
−3000
−2000
−1000
0
1000
2000
3000
4000
With pretraining
Without pretraining

Modern Approach

- Problem: auto-encoder’s reconstruction object may not be fully compatible with supervised task

- Solution: release the burden of reconstruction

Modern Approach

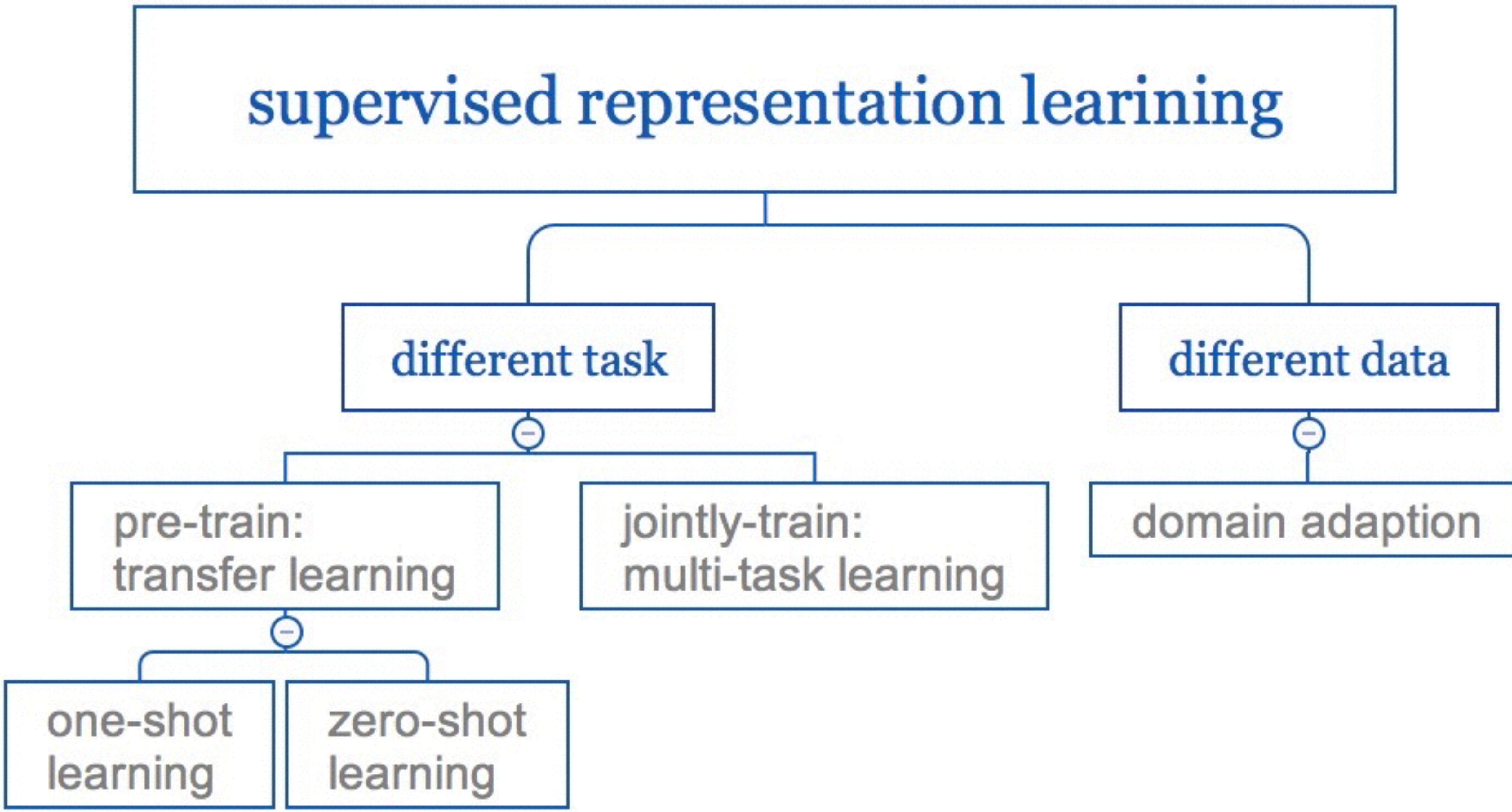


Semi-Supervised Learning with Ladder Networks, 2015

Modern Approach

Model	Number of incorrectly predicted test examples for a given number of labeled samples			
	20	50	100	200
DGN [21]			333 ± 14	
Virtual Adversarial [22]			212	
CatGAN [14]			191 ± 10	
Skip Deep Generative Model [23]			132 ± 7	
Ladder network [24]			106 ± 37	
Auxiliary Deep Generative Model [23]			96 ± 2	
Our model	1677 ± 452	221 ± 136	93 ± 6.5	90 ± 4.2
Ensemble of 10 of our models	1134 ± 445	142 ± 96	86 ± 5.6	81 ± 4.3

- 3. Supervised Representation Learning (Sec.15.2)
    - transfer learning
    - multi-task learning
    - domain adaption
    - one-shot learning
    - zero-shot learning



supervised representation learining
different task
different data
pre-train:
transfer learning
jointly-train:
multi-task learning
domain adaption
one-shot
learning
zero-shot
learning

Transfer Learning

- **Assumption**: learning task A helps to learn task B
- **Assumption**: representation learned from task A helps task B
- **Practice**:
  - pre-train model A on task A
  - use part of the model A's parameters to initialize model B
  - fine-tune model B on task B
- **Example**:
  - ImageNet classification —> other vision task
  - why: low & middle level feature of image

Multi-task Learning

- **Assumption**: several tasks share some common features / representation
- **Example**: jointly train POS + NER + sentence classification, share word vectors
- Useful when labeled data is not enough in each task

Domain Adaption

- Focus on different data distribution, rather than different task
- Example: sentiment analysis of customer reviews on
  - media, books, …
  - social networks, web forums, …

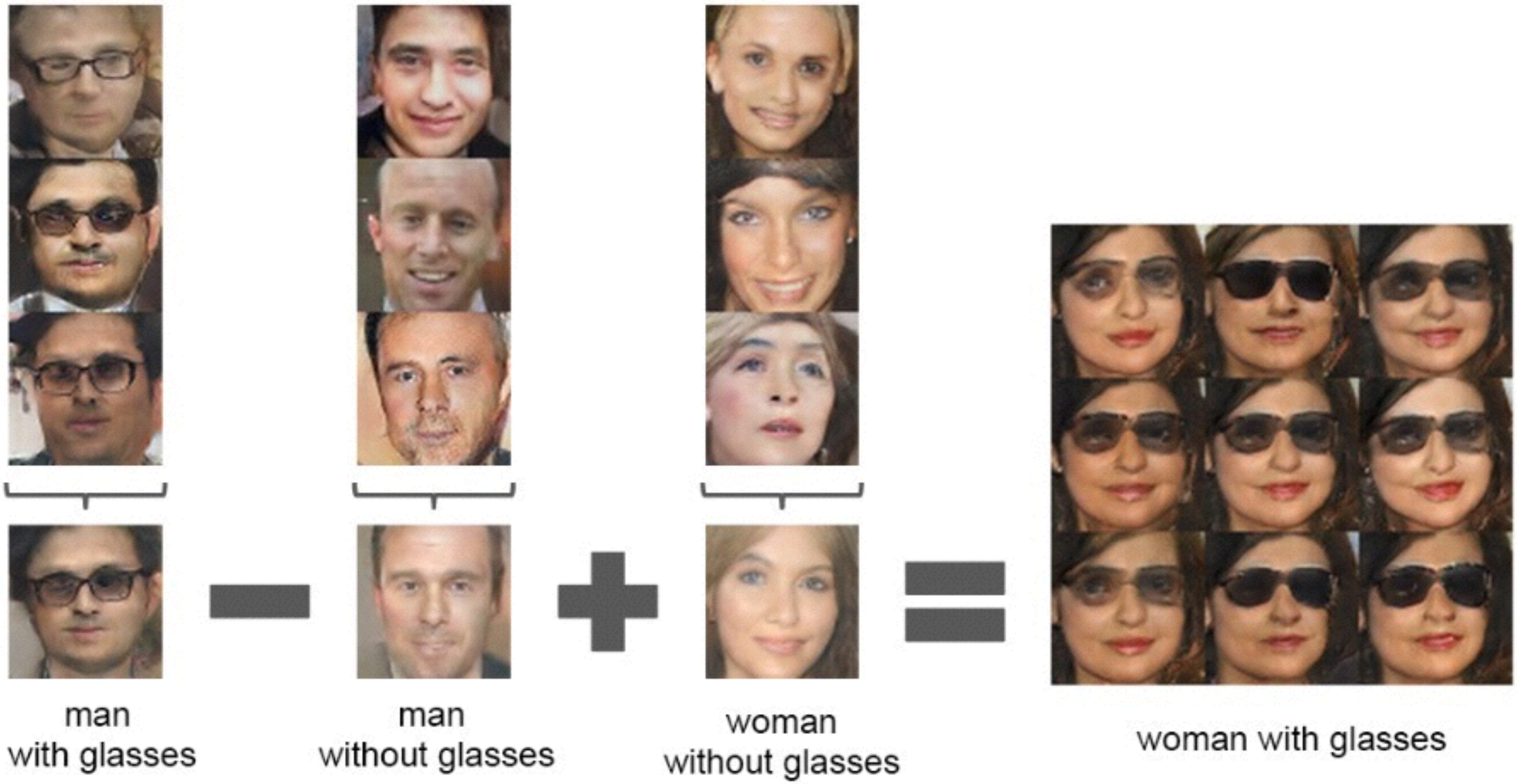
High Level v.s. Low Level

- **same data, different task**: share low level representation
  - e.g. same low level feature of image, different high level semantics
- **same task, different data**: share high level representation
  - e.g. same high level semantics of different language, different low level word embeddings

One-shot Learning

- Proposed by Fei-Fei Li in 2006
- An extreme form of transfer learning
- Only 1-5 labeled examples for each class

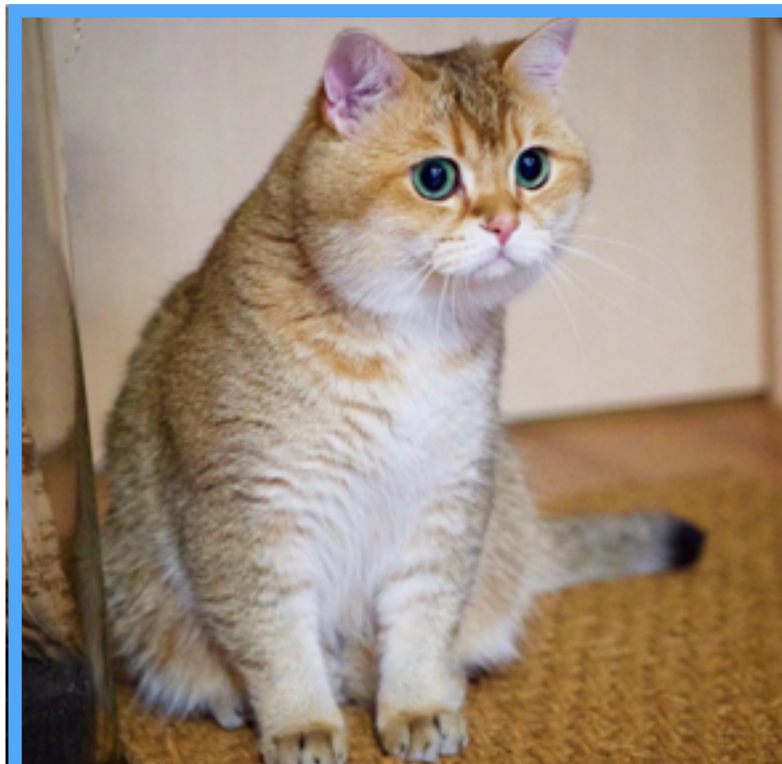
Example


man
with glasses
man
without glasses
woman
without glasses
woman with glasses

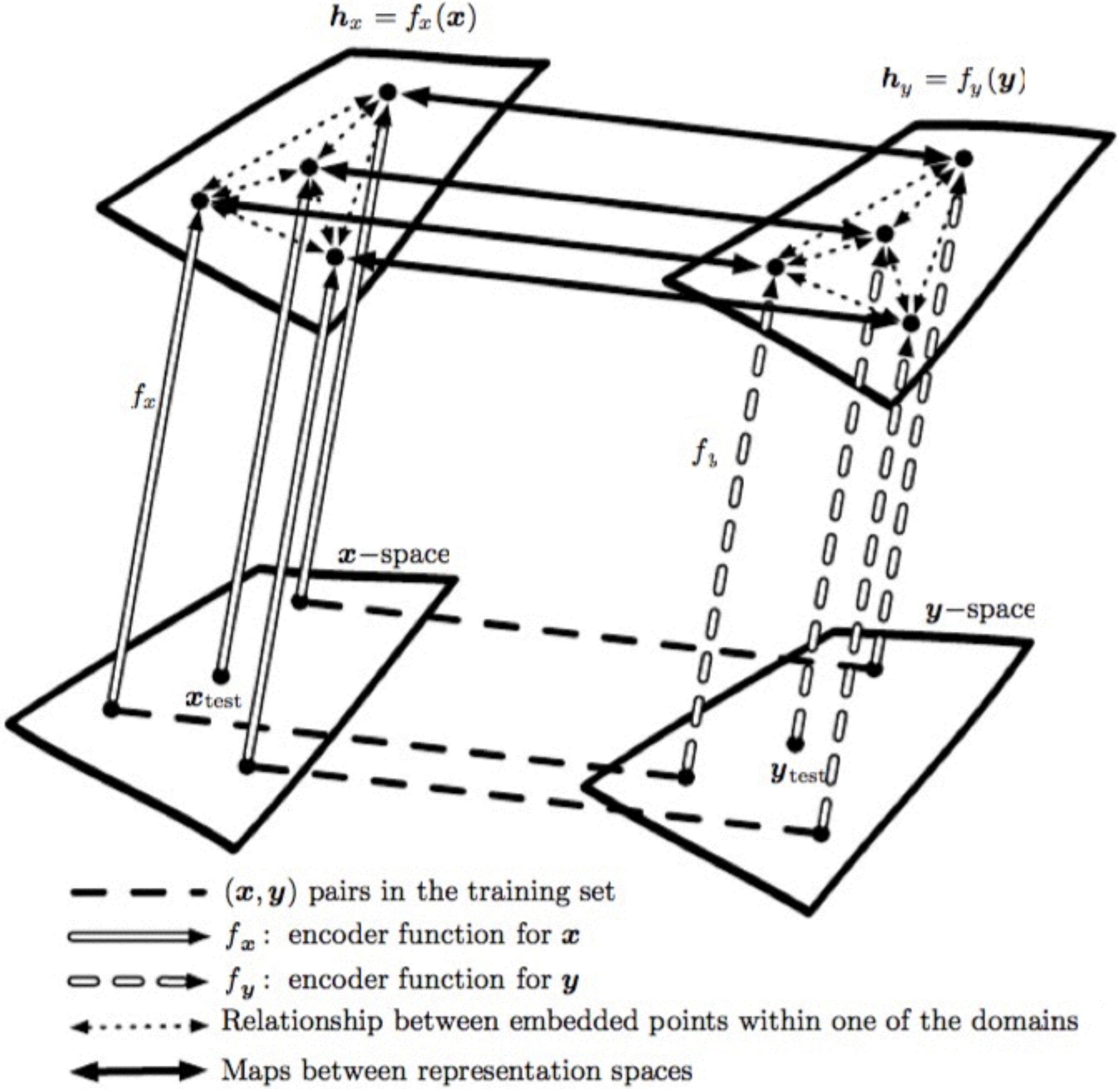
Zero-shot Learning

- **Example**: description text —> image

training: "four legs and pointy ears" —> [cat]

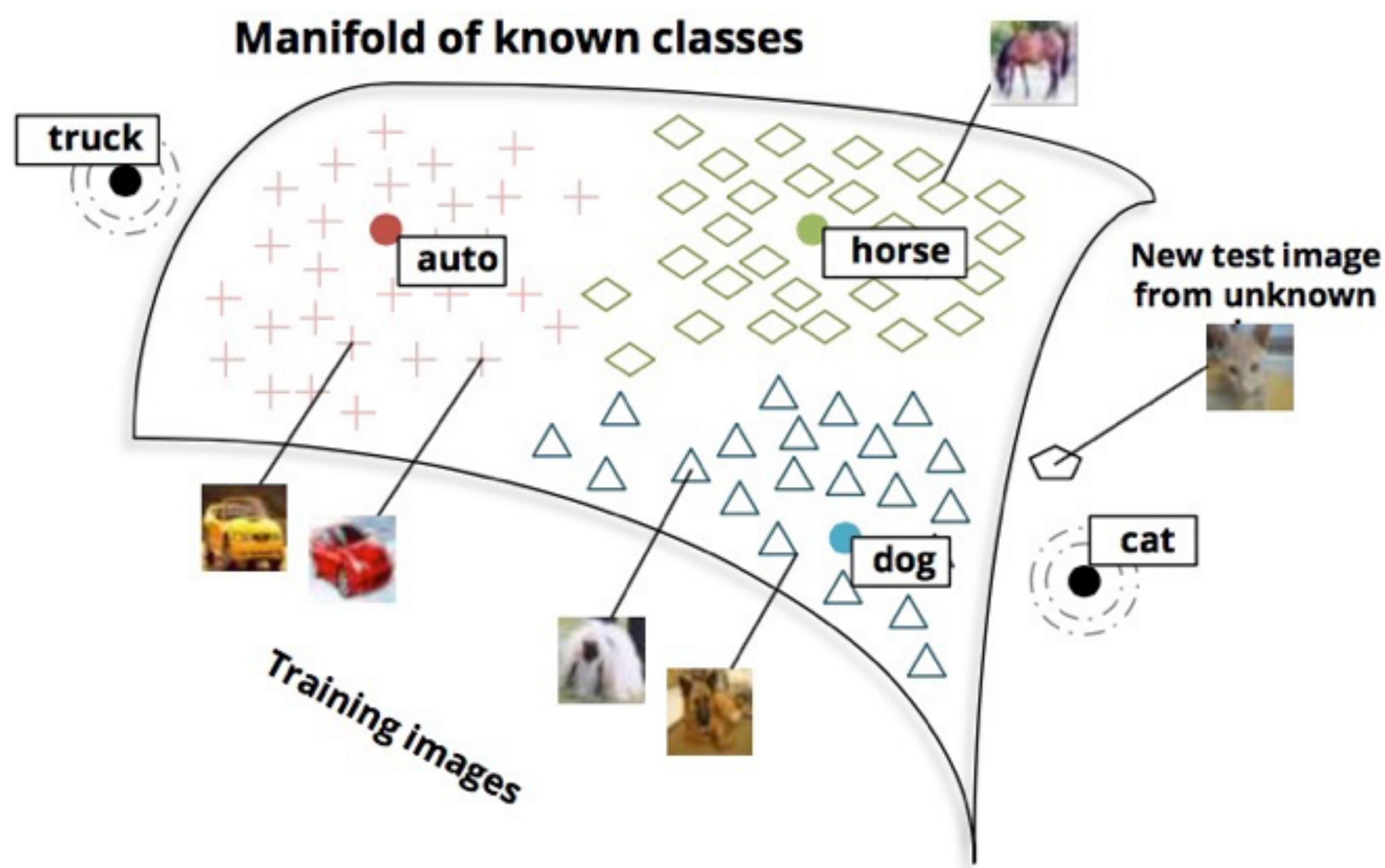
testing:  ————> [cat]

- How can zero-shot learning be possible?
- One approach: alignment in representation space


$\boldsymbol{h}_x = f_x(\boldsymbol{x})$
$\boldsymbol{h}_y = f_y(\boldsymbol{y})$
f_x
f_y
$\boldsymbol{x}$−space
$\boldsymbol{y}$−space
$\boldsymbol{x}_{\text{test}}$
$\boldsymbol{y}_{\text{test}}$
$(\boldsymbol{x}, \boldsymbol{y})$ pairs in the training set
f_x: encoder function for $\boldsymbol{x}$
f_y: encoder function for $\boldsymbol{y}$
Relationship between embedded points within one of the domains
Maps between representation spaces

Example

- *Zero-Shot Learning Through Cross-Modal Transfer*, Richard Socher, 2013



Example

- **Task**: Word translation on two languages
- **Possible approach**: jointly train word embeddings in both language, as well as their alignment in embedding space
- **At testing time**: translate unseen words in bilingual corpus by adding an offset vector in embedding space

- 4. What is a Good Representation?
    - disentangling causes / distributed (Sec.15.3, 15.4)
    - others (Sec. 15.5, 15.6)

Compare Two Assumptions

- **Disentangling causes**: good representation —> features within the representation correspond to the underlying causes of the observed data
- **Distributed**: good representation —> composed of elements that can be set separately from each other
- Different?
- e.g. some dimensions in word embeddings have particular semantics (*Evaluation of Word Vector Representations by Subspace Alignment*, 2015*)*

Compare Two Assumptions

- Low level:
  - features (e.g. pixels)
- High level:
  - causes, latent factors, variance, semantics, …
  - just the same thing?
- If yes, then these two assumptions are equivalent

Opposite Side

- Entangled representation
  - no single independent elements / direction
  - e.g. raw pixels
- Symbolic / one-hot representation

Why Good? (1)

- **Condition 1**: if the representation successfully disentangle causes of the data

- **Condition 2**: and the label we want to predict is closely associated with one of the causes

- **Result**: then this representation will make predicting much easier

Why Good? (2)

- Share attributes makes generalization easier
- Symbolic
    - given: [cat, mouse, …] can be pets
    - given: [A, B, …] can’t be pets
    - infer: [dog] ???
- Distributed
    - given: [cat, mouse, …] {cute == True, legs = 4} can be pets
    - given: [A, B, …] {cute == False, legs == 100 } can’t be pets
    - infer: [dog] {cute == True, legs = 4}, maybe can also be pets

Why Good? (3)

- Richer similarity space
- Symbolic
  - 2 similarity: is / isn’t
- N binary attributes
  - N+1 similarity: share 0, 1, … N same attributes
- N continous attributes
  - continuous similarity

Why Good? (4)

- Exponential representation power v.s. dimension disaster

- If D dimension binary features can be learn **separately**, we need O(2D) samples to generalize

- But the same generalization power can only be achieved by O(2^D) samples

Redefinition of Salient Causes

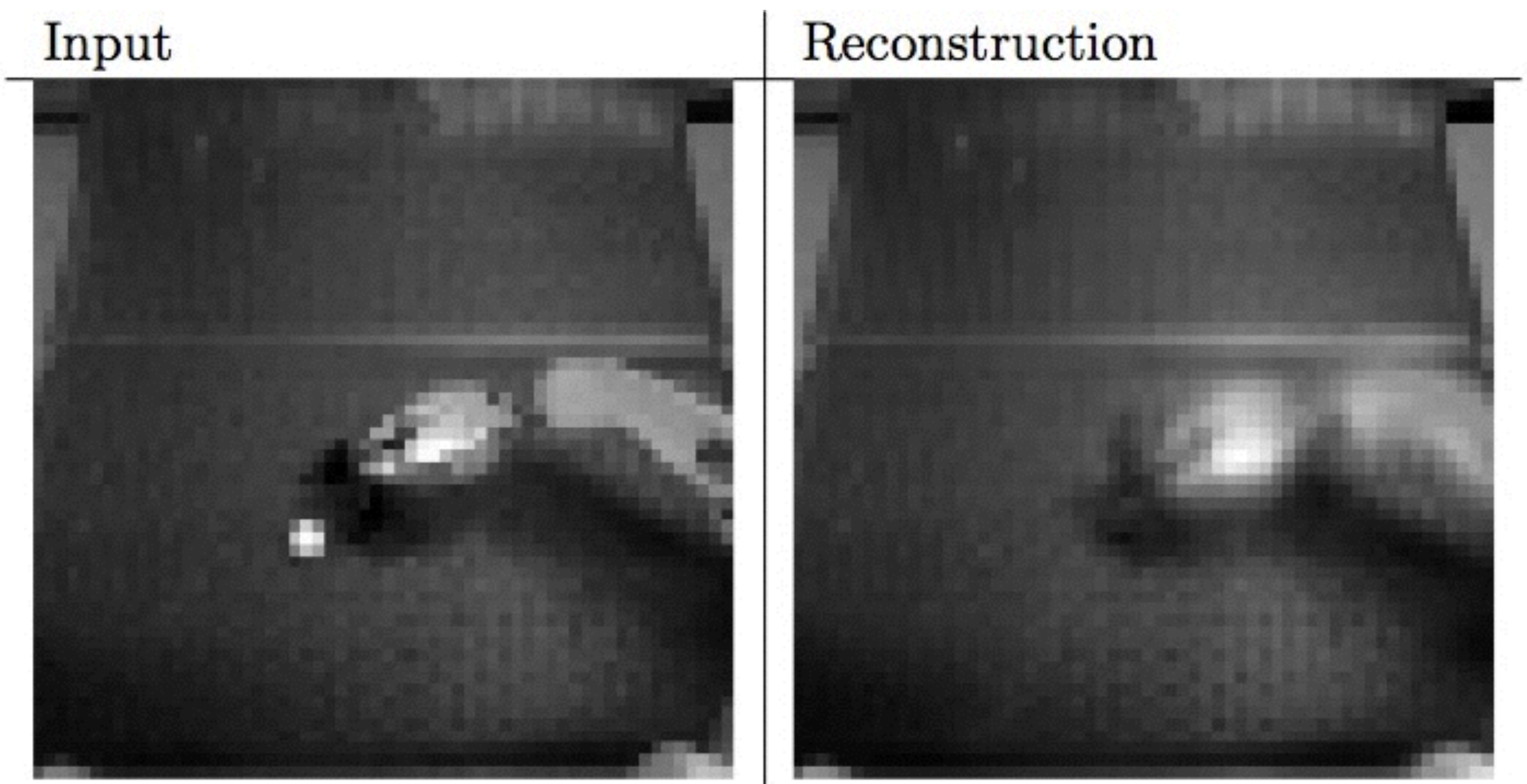
- Problem 1: not all causes / semantics / factors can be capture, the most salient ones will be capture first
- Problem 2: the auxiliary task and main task may disagree on what is salient

Redefinition of Salient Causes

- Solution 1: let the main task to be learn together with the auxiliary task
- Solution 2: redefine what is salient

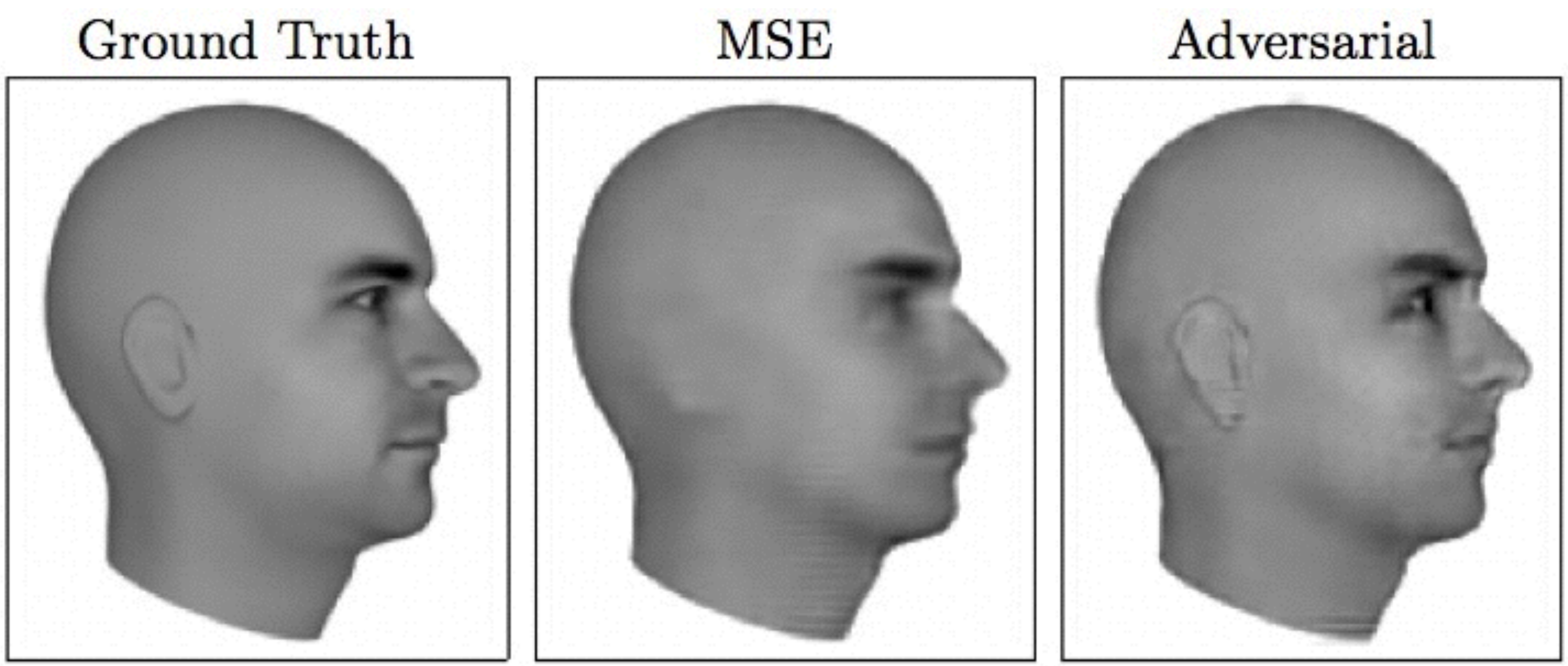
Redefinition of Salient Causes

- Traditional definition of salient: reconstruction error
- Problematic if some objects of interest is small



Redefinition of Salient Causes

- GAN is a brilliant way to redefine what is salient
- Discriminator will always capture the most salient features first to classify real and fake samples, no matter these features is “big” or “small”



Others: Depth

- Exponential gains

Others: Smoothness

- Notice that distributed representation can be smooth or non-smooth

Others: Manifolds

- Probability mass concentrates
- Locally connected
- Models like auto-encoders explicitly try to learn a manifold

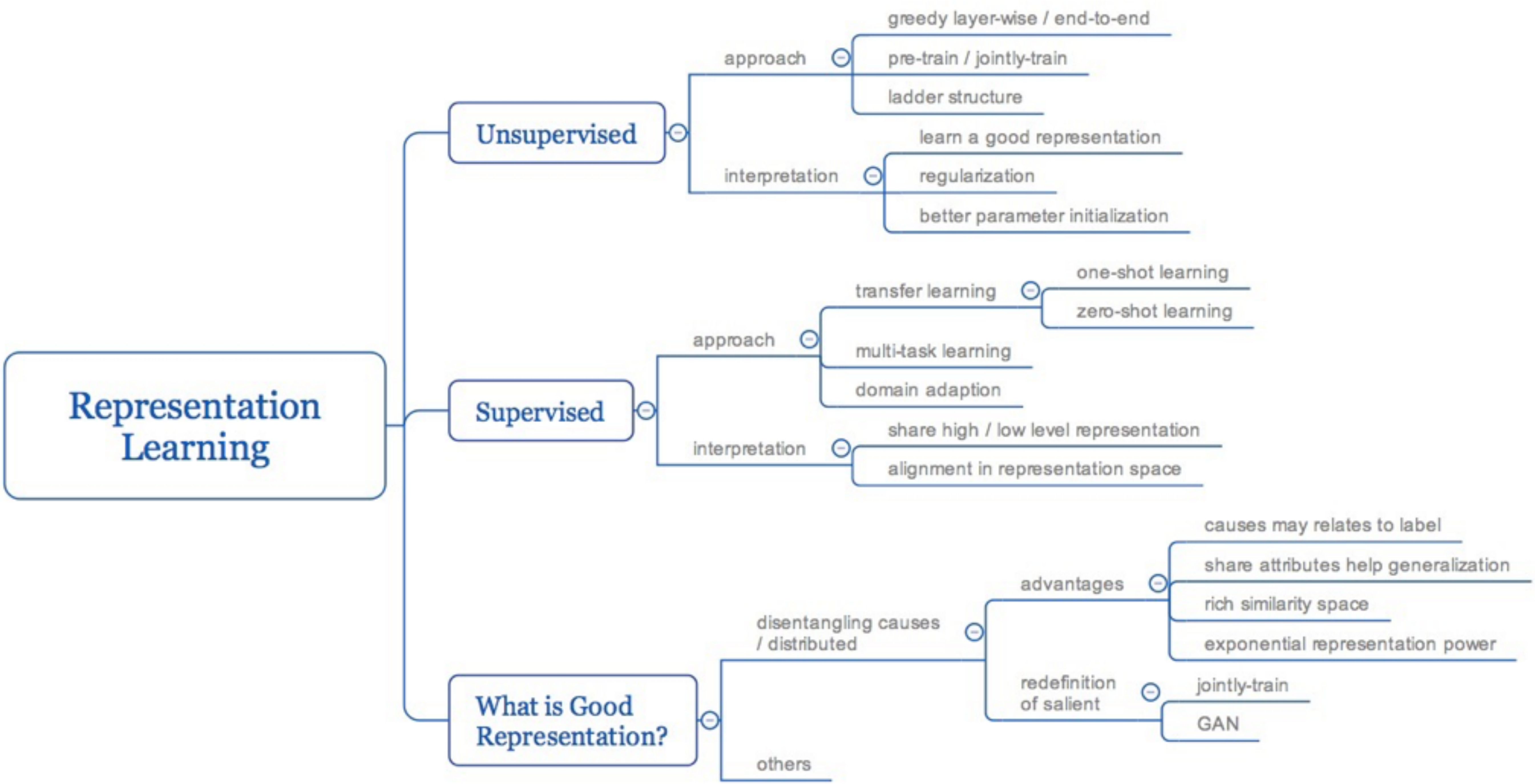
Others: Sparsity

- Spatial sparsity: dependent, non-distributed
- Temporal sparsity: can be independent, distributed

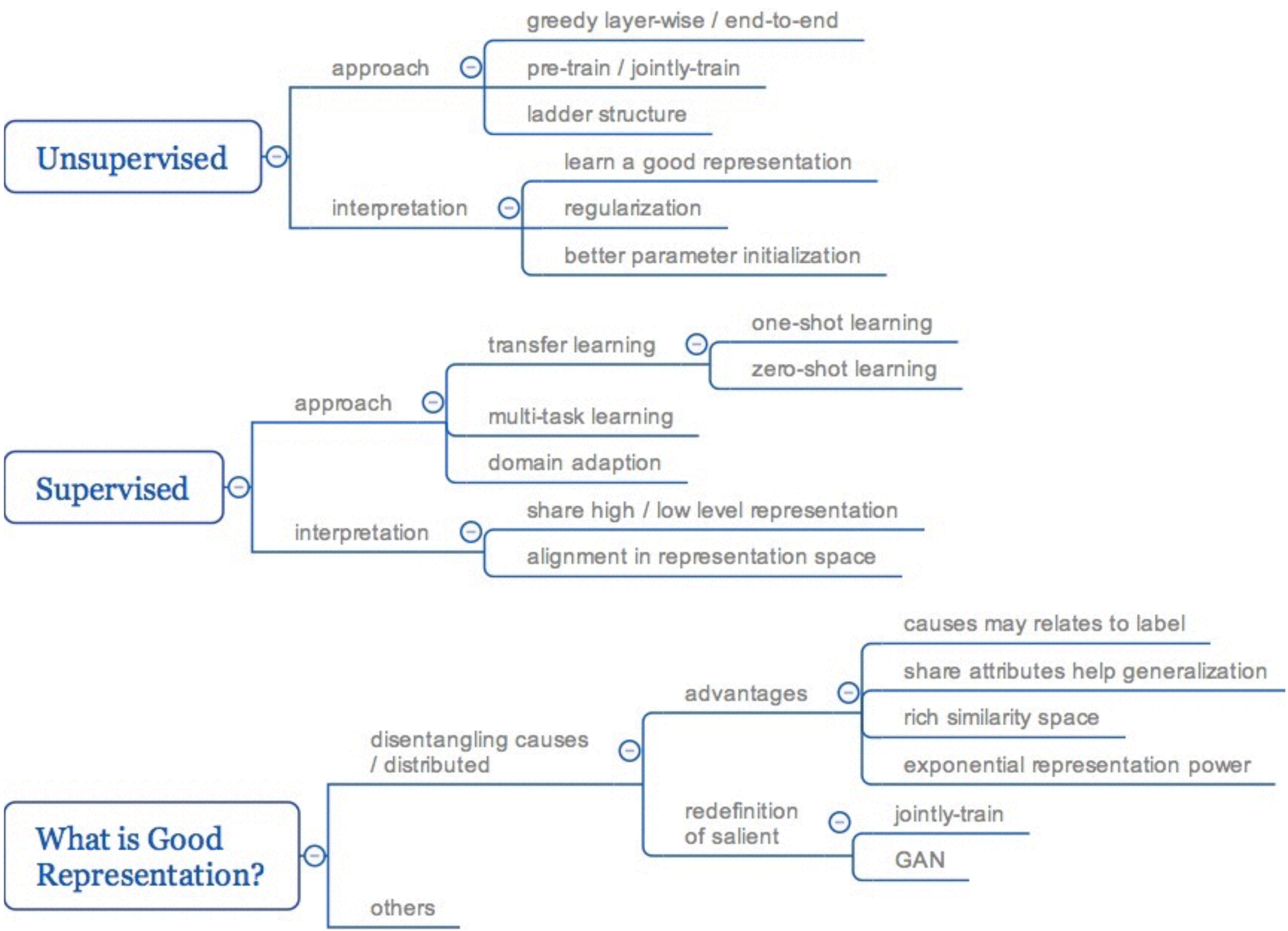
Others: Simplicity of Factor Dependencies

- Looser requirement than disentangling / distributed
- In the simplest possible form: $P(\boldsymbol{h}) = \sum_i P(h_i)$

Summary



Representation Learning
Unsupervised
approach
greedy layer-wise / end-to-end
pre-train / jointly-train
ladder structure
interpretation
learn a good representation
regularization
better parameter initialization
Supervised
approach
transfer learning
one-shot learning
zero-shot learning
multi-task learning
domain adaption
interpretation
share high / low level representation
alignment in representation space
What is Good Representation?
disentangling causes / distributed
advantages
causes may relates to label
share attributes help generalization
rich similarity space
exponential representation power
redefinition of salient
jointly-train
GAN
others


Unsupervised
approach
greedy layer-wise / end-to-end
pre-train / jointly-train
ladder structure
interpretation
learn a good representation
regularization
better parameter initialization
Supervised
approach
transfer learning
one-shot learning
zero-shot learning
multi-task learning
domain adaption
interpretation
share high / low level representation
alignment in representation space
What is Good Representation?
disentangling causes / distributed
advantages
causes may relates to label
share attributes help generalization
rich similarity space
exponential representation power
redefinition of salient
jointly-train
GAN
others