Machine Learning Beyond Static Datasets

ESSAI 2023

Dr. Martin Mundt,

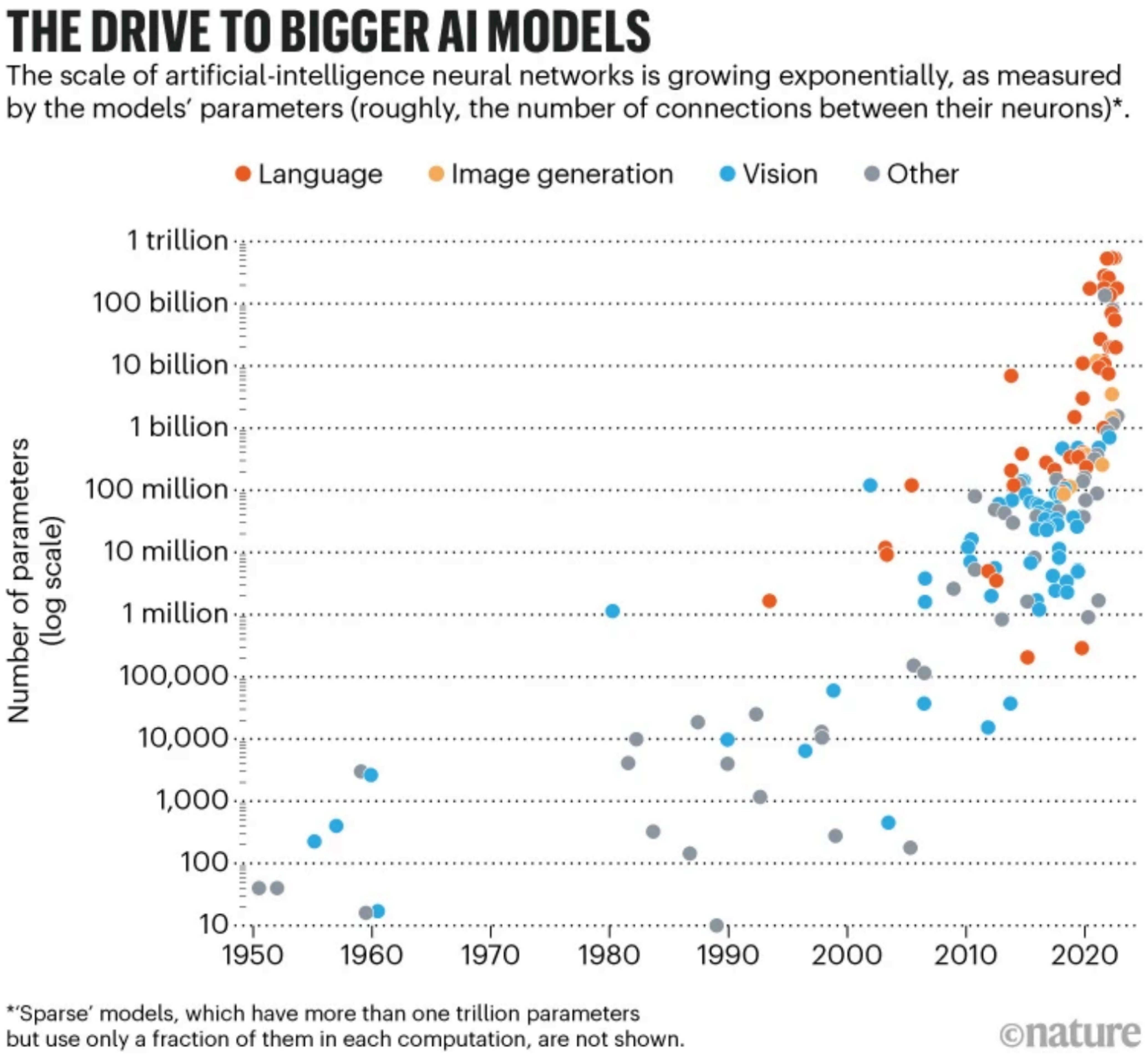
Research Group Leader, TU Darmstadt & hessian.AI

Board Member of Directors, ContinualAI

Course: http://owll-lab.com/teaching/essai-23

Day 1 - The Present:
Static Datasets & Re-use

Is scale all we need?!



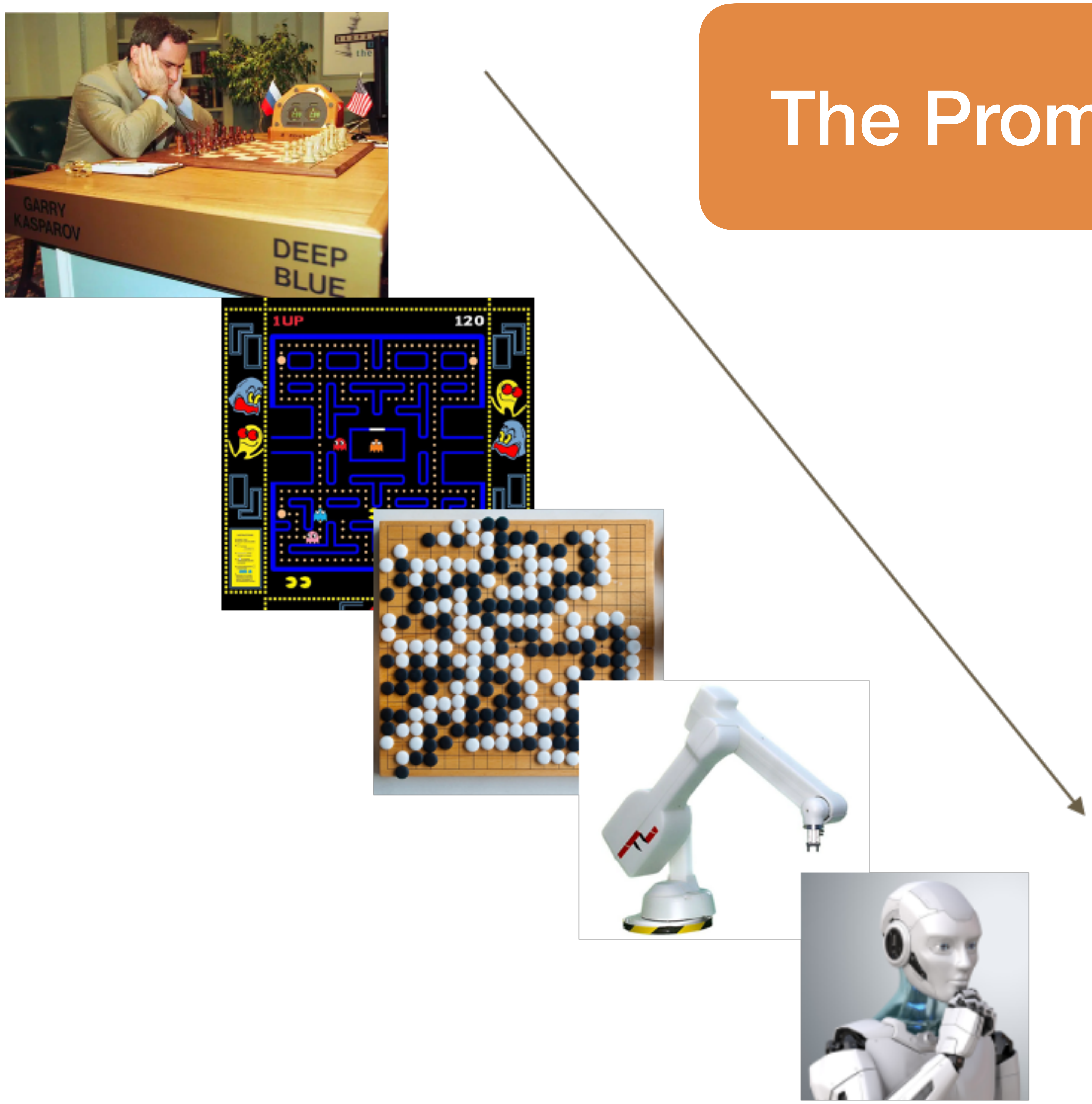
Source: Adapted from Our World in Data, and from J. Sevilla *et al*. Preprint at https://arxiv.org/abs/2202.05924 (2022).

Humans learn continually! Why shouldn't ML models?

The Promise

At the least, lifelong learning may be one pathway to more human-like intelligence

At the most, its one pathway towards strong, more general artificial intelligence



"Intelligence is the ability to adapt to change."
- Stephen Hawking

Slide adapted from our AAAI-23 Continual Causality Tutorial, Cooper & Mundt

Despite many great achievements of current systems, few, if any, truly can learn & predict over time

The Premise

"It's about making the models bigger, safer, compute efficient, faster at sampling…"

But narrow models aren't robust, suffer from incomplete & biased datasets, don't adapt to novel situations

Can we really capture everything upfront?

The Problems!
Why are we not there & what to do - Course Overview

Day 1: The Present
Static Datasets & Re-use

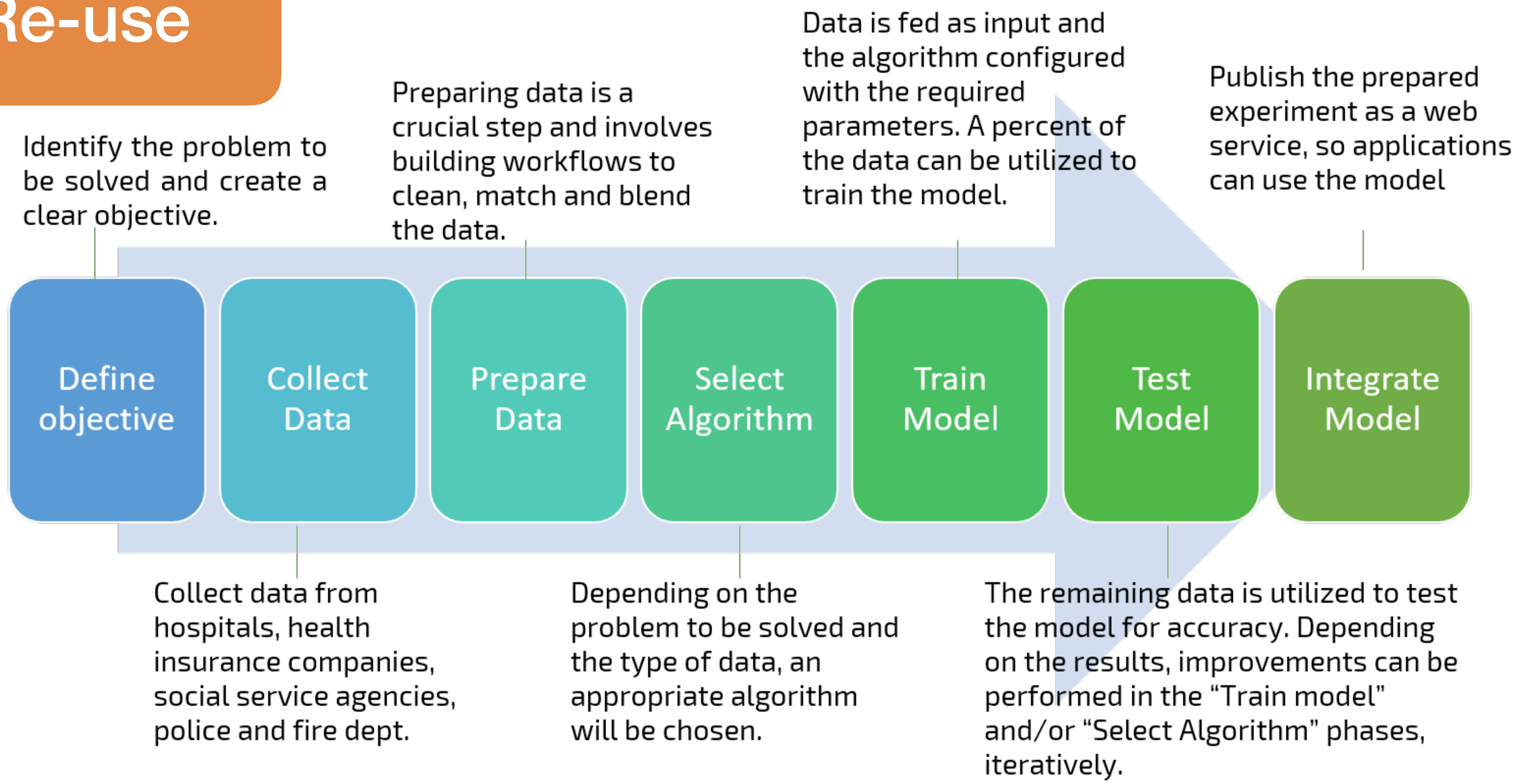
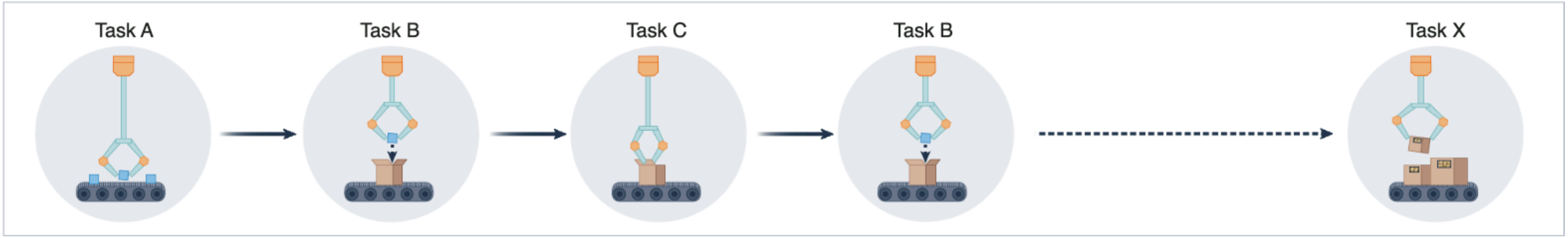


Figure from https://www.congrelate.com/get-workflow-machine-learning-images/

The Problems!
Why are we not there & what to do - Course Overview

Day 1: The Present
Static Datasets & Re-use



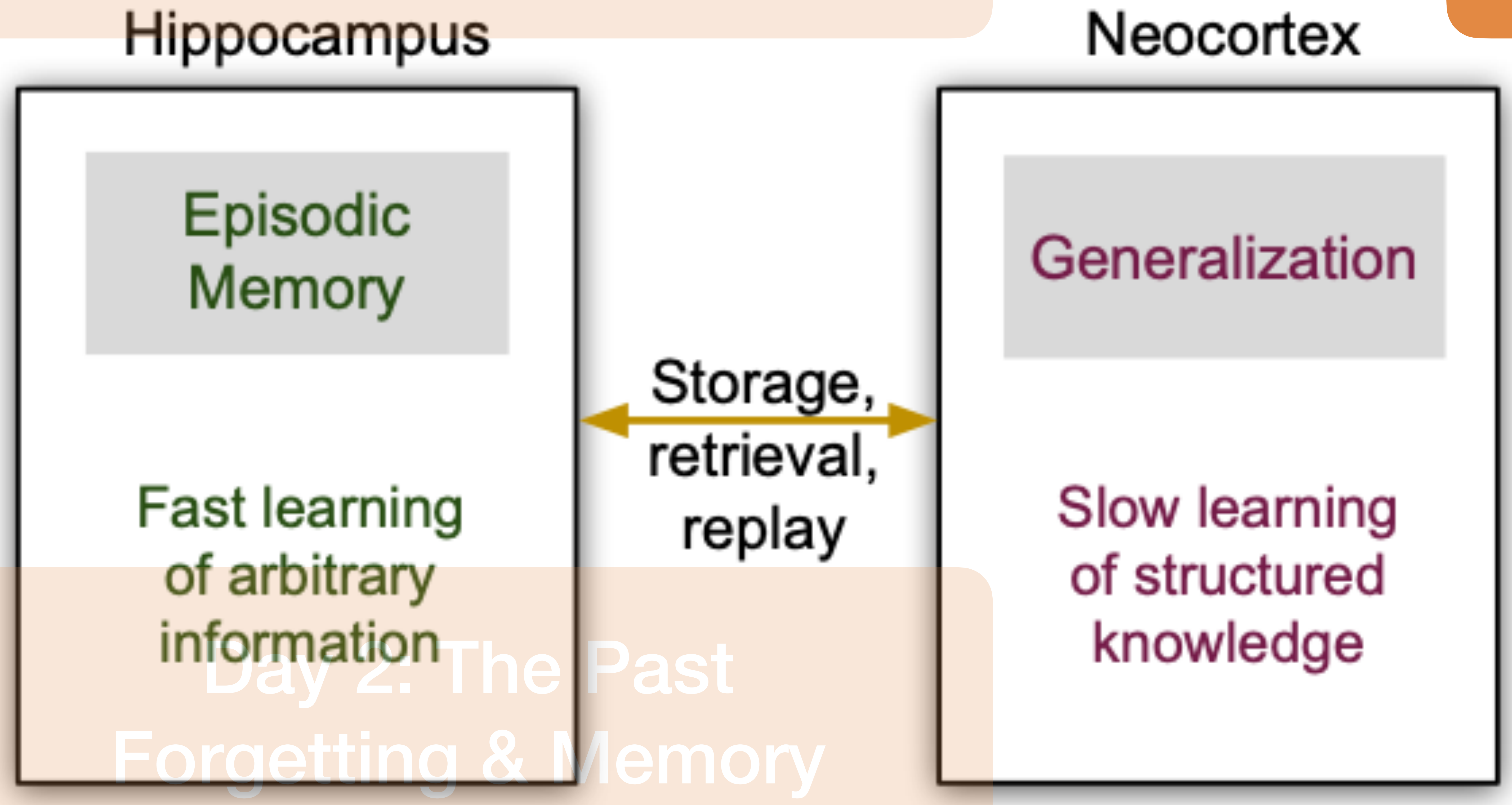
Day 2: The Past
Forgetting & Memory

Figure from Kudithipudi et al, “Biological underpinnings for lifelong learning machines”, Nature Machine Intelligence (4), 2022

The Problems!
Why are we not there & what to do - Course Overview

Day 1: The Present
Static Datasets & Re-use

Day 3: From Past to Future
Memory & Growth



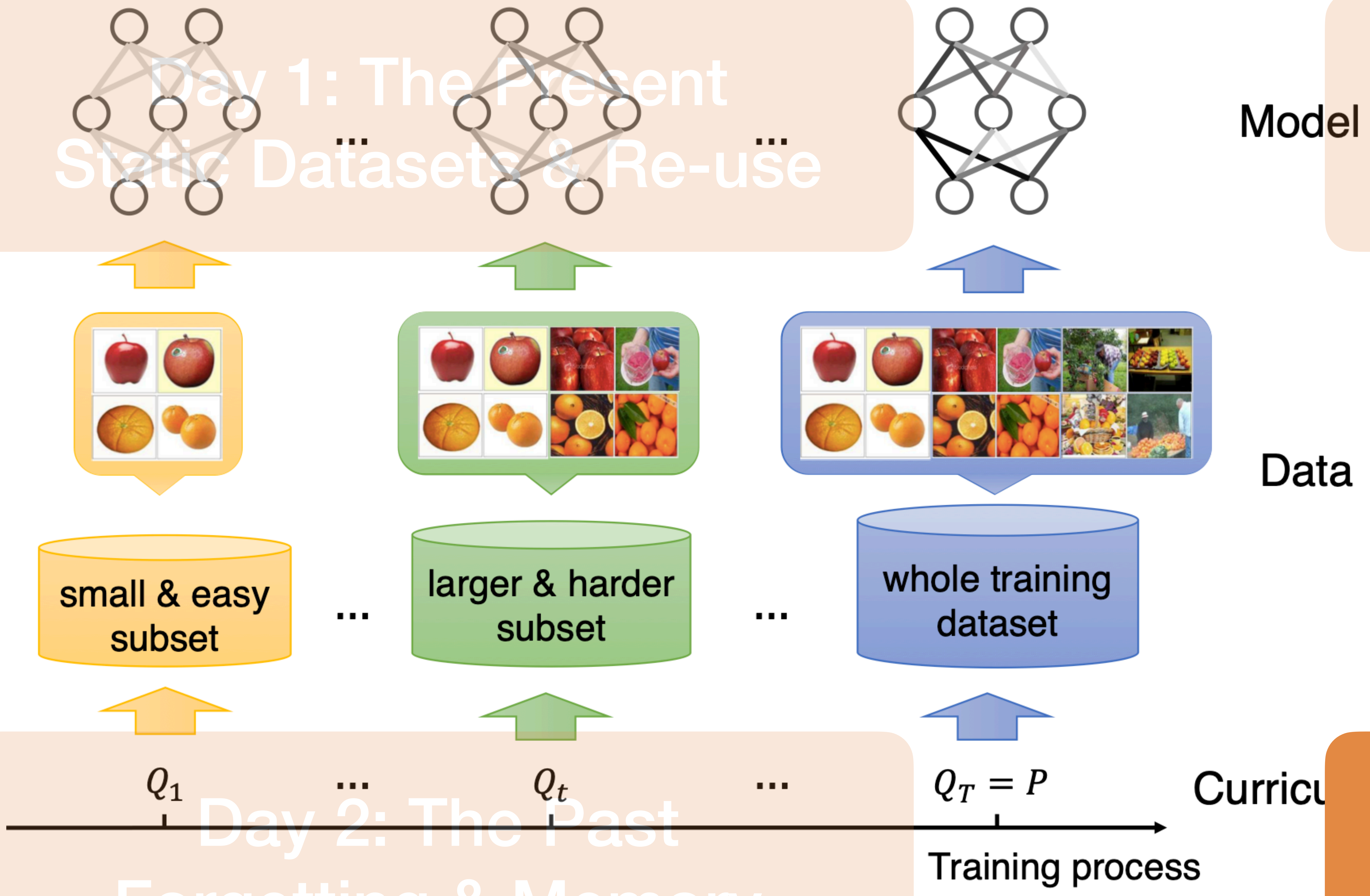
Day 2: The Past
Forgetting & Memory

Figure from Parisi et al, “Continual Lifelong Learning with Neural Networks: A Review”, Neural Networks 113, 2019

The Problems!
Why are we not there & what to do - Course Overview

Day 1: The Present
Static Datasets & Re-use

Day 3: From Past to Future
Memory & Growth



Day 2: The Past
Forgetting & Memory

Day 4: The Future
Data Selection &
Learning Curricula

Figure from Wang et al, “A Survey on Curriculum Learning”, TPAMI 2021

The Problems!
Why are we not there & what to do - Course Overview

Day 1: The Present
Static Datasets & Re-use

Day 3: From Past to Future
Memory & Growth

Day 5: The Unknown
Open World Learning &
Evaluation

Day 2: The Past
Forgetting & Memory

Day 4: The Future
Data Selection &
Learning Curricula

Motivation: A step back - what is machine learning?

The static ML workflow

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E”.

Machine Learning, T. M. Mitchell, McGraw-Hill,1997

ML recap: train - test splits

*“The result of running the machine learning algorithm can be expressed as a **function**. The precise form of the function is determined during the **training** phase, also known as the **learning** phase, on the basis of the **training data.***

*Once the model is trained it can then determine the identity of new instances, which are said to comprise a **test set**. The ability to categorize correctly new examples that differ from those used for training is known as **generalization**”.*

Pattern Recognition and Machine Learning, C. M. Bishop, Springer 2006, page 2

ML recap: error/loss & learning

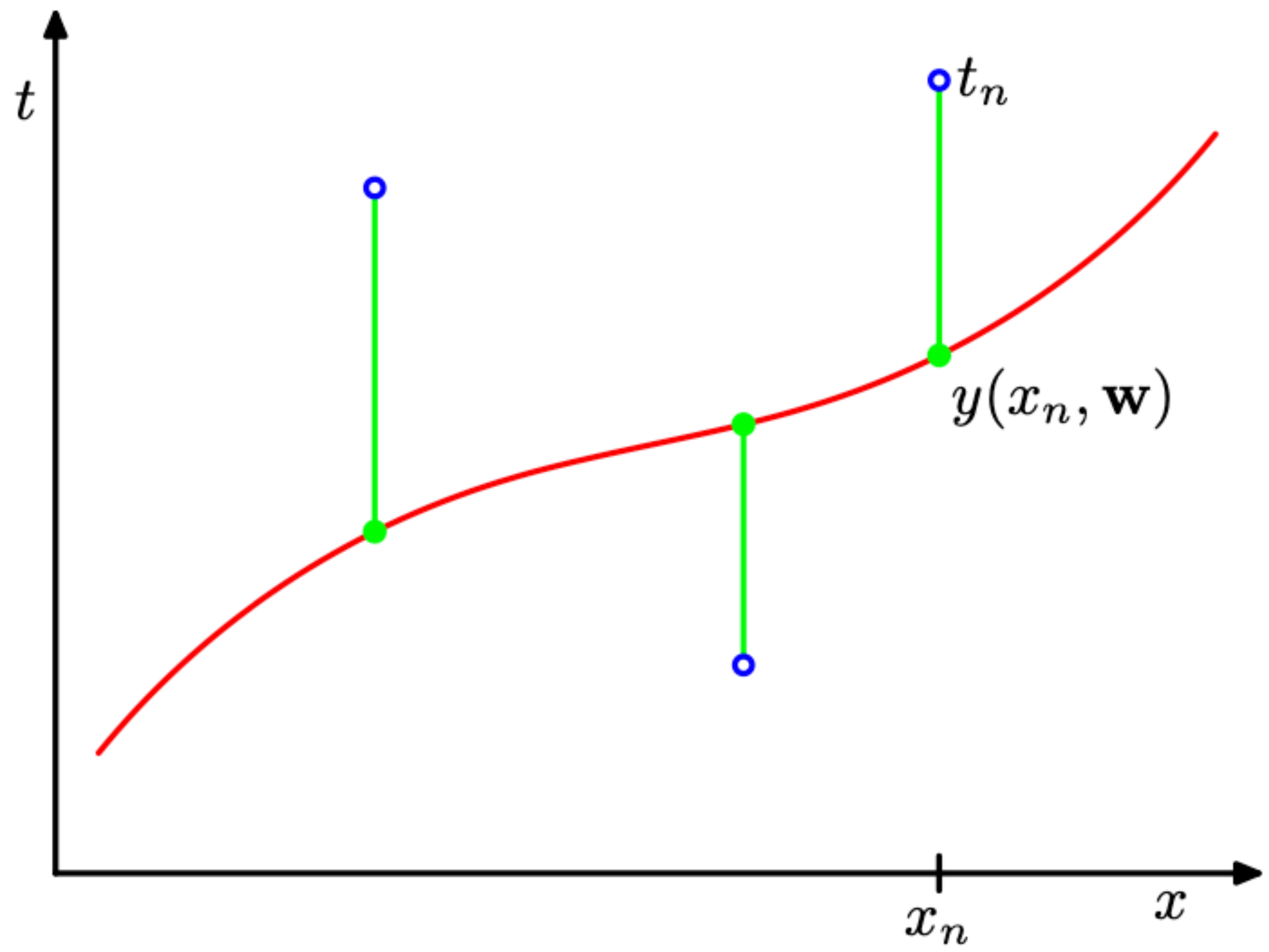


Figure 1.3 The error function (1.2) corresponds to (one half of) the sum of the squares of the displacements (shown by the vertical green bars) of each data point from the function $y(x, \mathbf{w})$.

Pattern Recognition and Machine Learning, C. M. Bishop,

Springer 2006, example on polynomial curve fitting: intro page 6

ML recap: under & overfitting

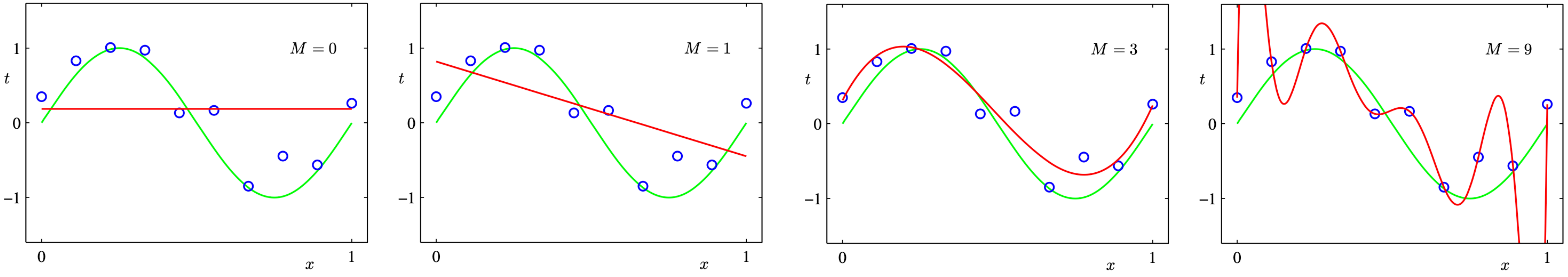


Figure 1.4 Plots of polynomials having various orders M, shown as red curves, fitted to the data set shown in Figure 1.2.

Pattern Recognition and Machine Learning, C. M. Bishop, Springer 2006,
example on polynomial curve fitting: page 7

ML recap: under & overfitting

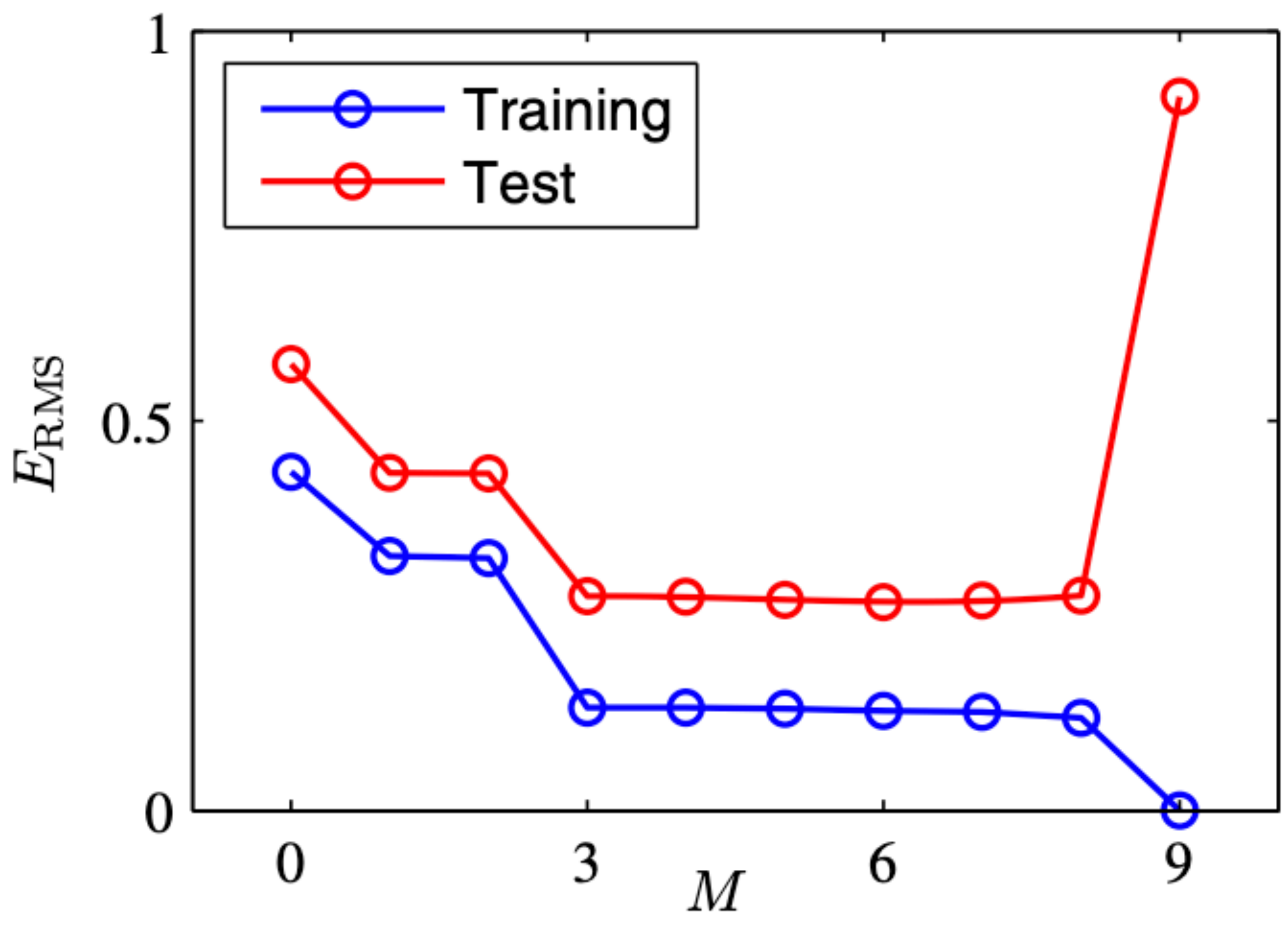


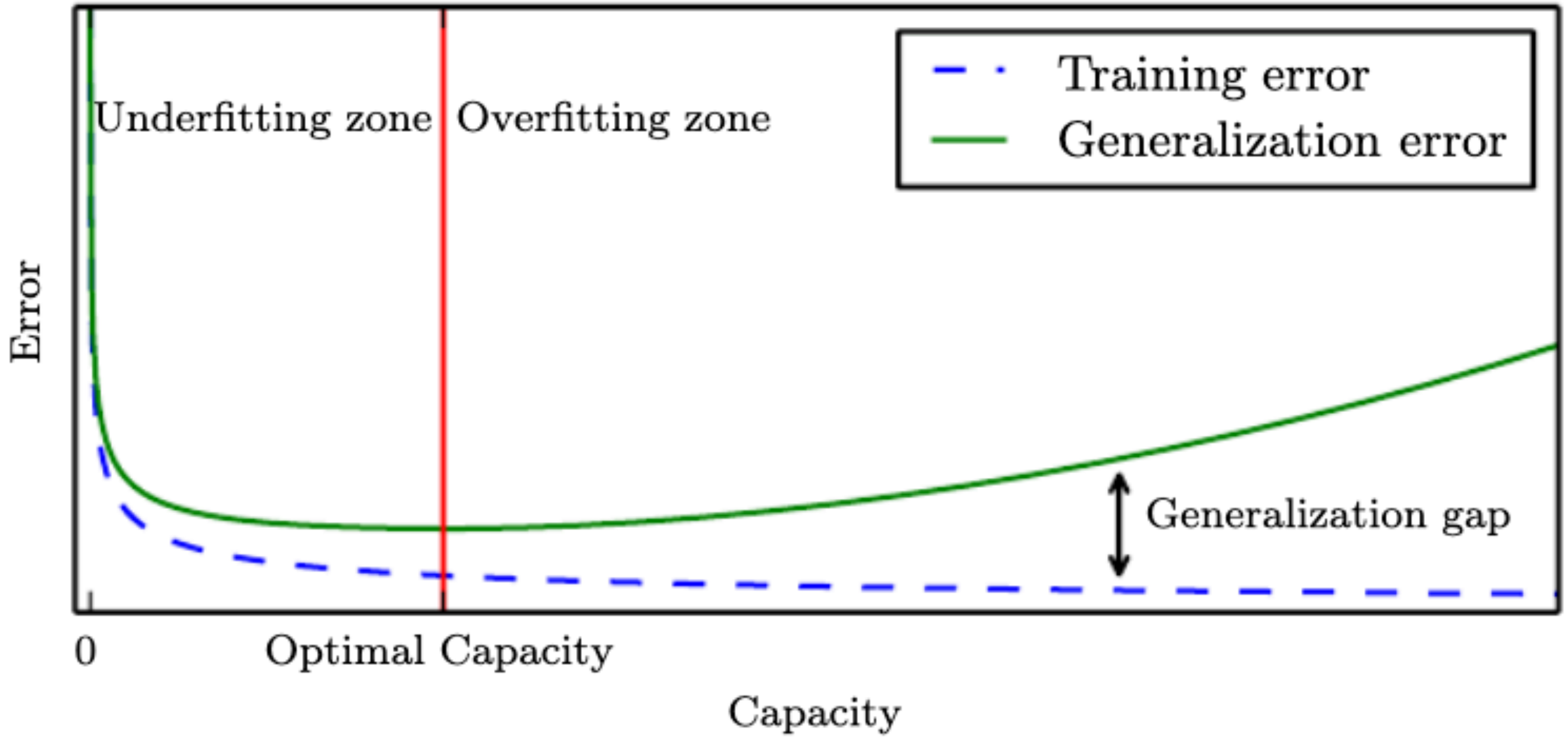
Figure 1.5 Graphs of the root-mean-square error, defined by (1.3), evaluated on the training set and on an independent test set for various values of M.

"Intuitively, what is happening is that the more flexible polynomials with larger values of M are becoming increasingly tuned to the random noise on the target values".

Pattern Recognition and Machine Learning, C. M. Bishop, Springer 2006, example on polynomial curve (over-)fitting in the introduction on page 8

ML recap: under & overfitting

This picture is still very much the same in the “deep learning era”



Deep Learning, Goodfellow, Bengio, Courville, MIT Press 2016,

Machine Learning Basics chapter, page 112.

What do you think are the goals of ML?

The static ML workflow: goals

*“Of course, when we use a machine learning algorithm, we **do not fix the parameters ahead of time**, then sample both datasets. We **sample the training set**, **then** use it to **choose the parameters** to reduce training set error, **then sample the test set**.*

The factors determining how well a ML algorithm will perform are its ability to:

1. Make the training error small.

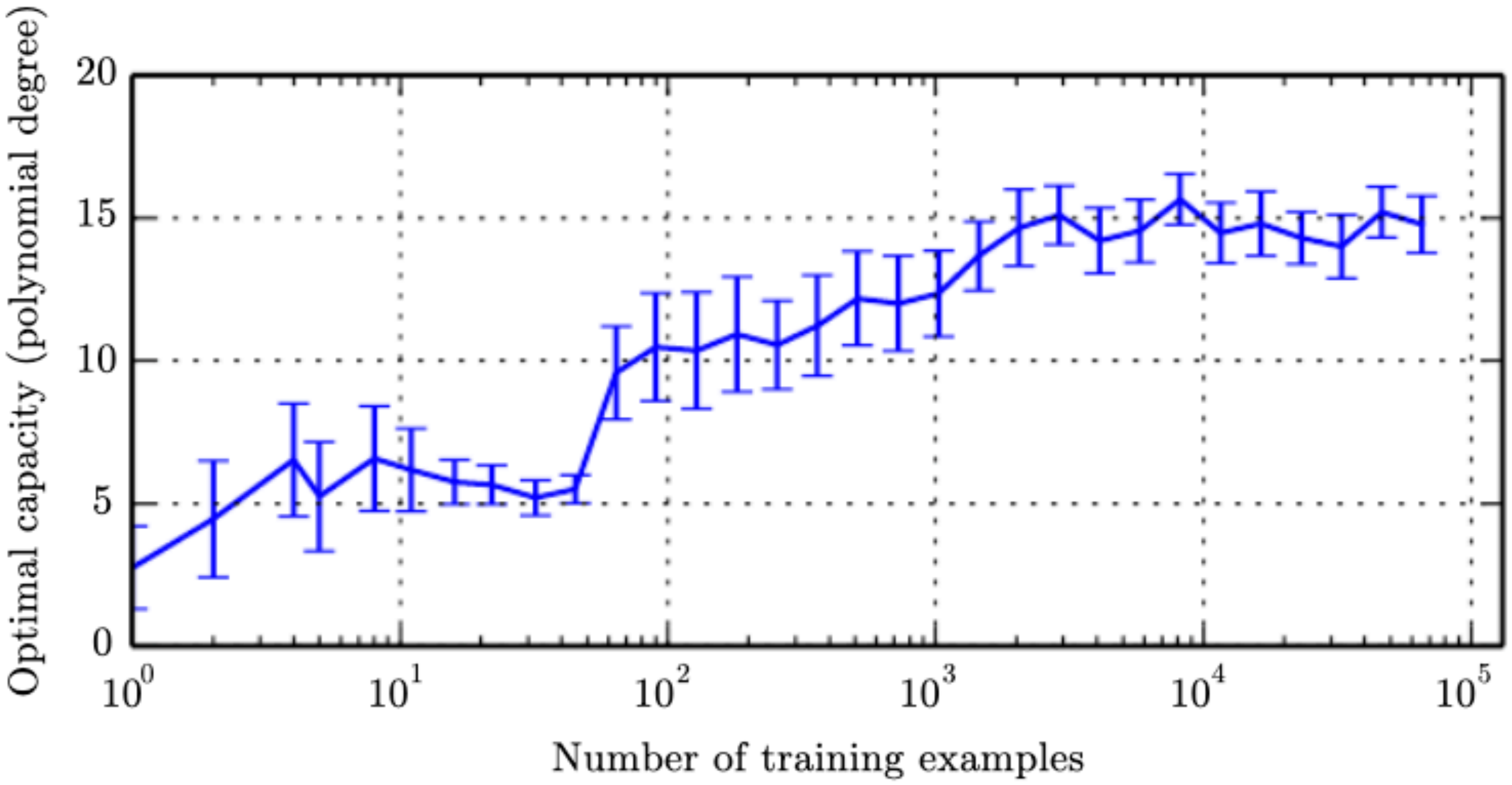
2. Make the gap between training and test error small”.

Deep Learning, Goodfellow, Bengio, Courville, MIT Press 2016,

Machine Learning Basics chapter, page 108.

The static ML workflow: goals

So is ML all about finding a large dataset & a right capacity model?



Deep Learning, Goodfellow, Bengio, Courville, MIT Press 2016,

Machine Learning Basics chapter, page 114.

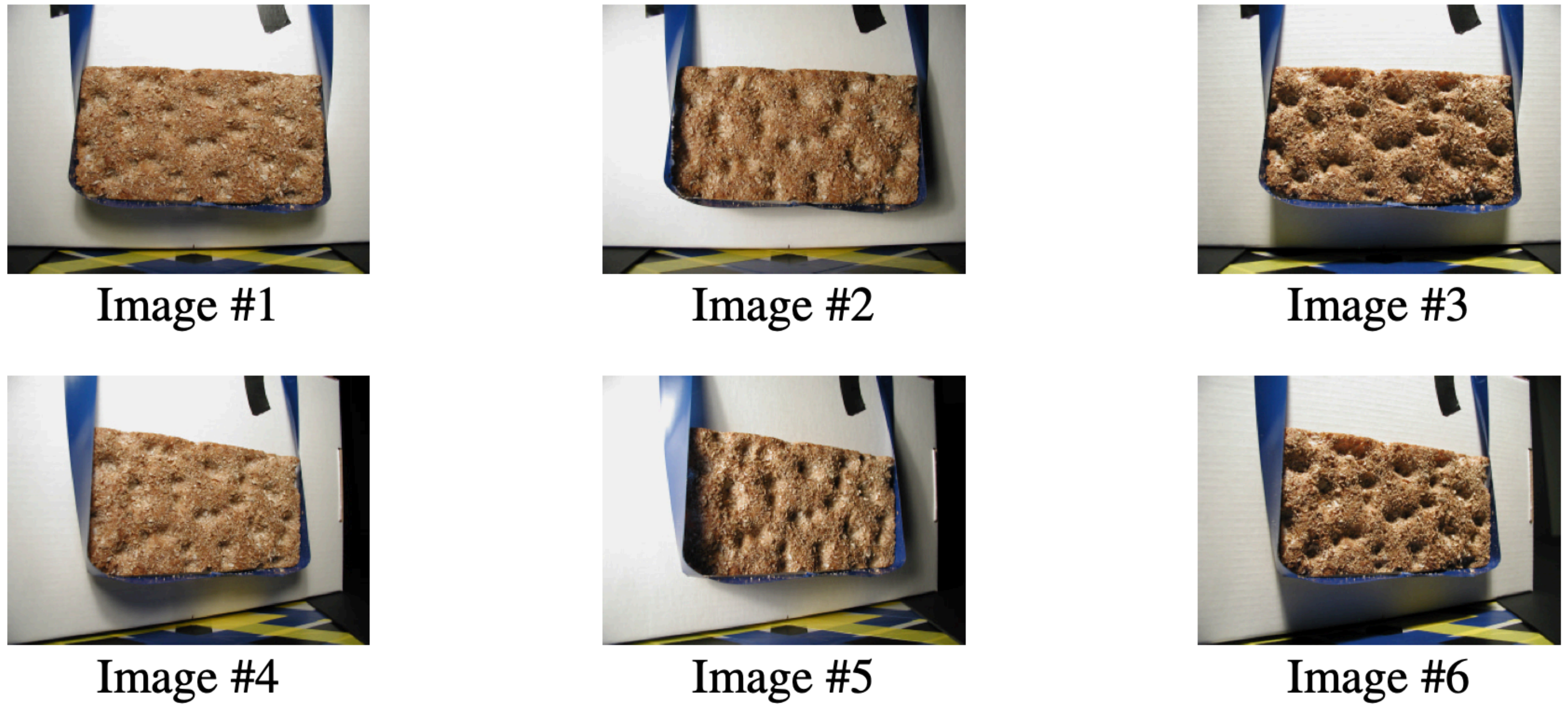
How do you think datasets should be acquired?

Static datasets: controlled

Small scale, but (some) controlled acquisition parameters

Image number	Object pose			Illumination direction		
	Frontal	22.5 ° right	22.5 ° left	Frontal	≈ 45 ° from top	≈ 45 ° from side
1	x			x		
2	x				x	
3	x					x
4		x		x		
5		x			x	
6		x				x
7			x	x		
8			x		x	
9			x			x

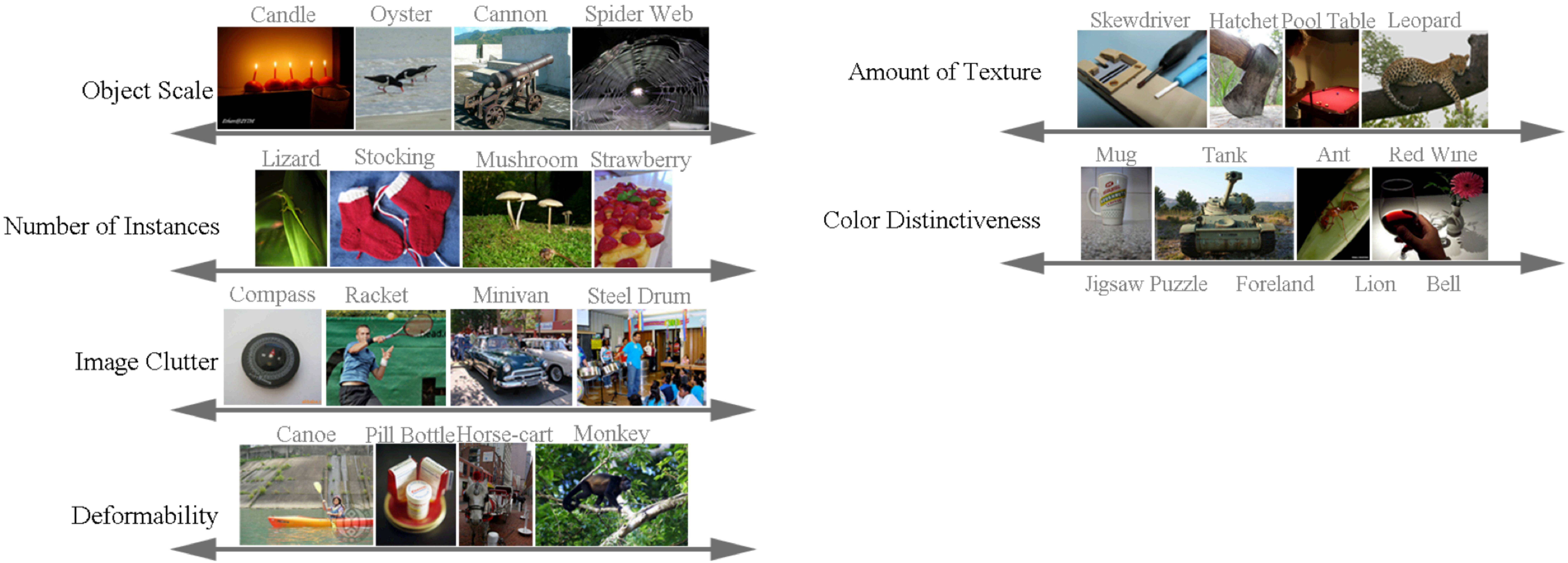
Table 3: The labeling of images within each scale in the KTH-TIPS database.



Hayman et al, "On the significance of real-world conditions for material classification", ECCV 2004
& Fritz, Hayman et al, "The KTH-TIPS database", technical report 2004

Static datasets: large scale

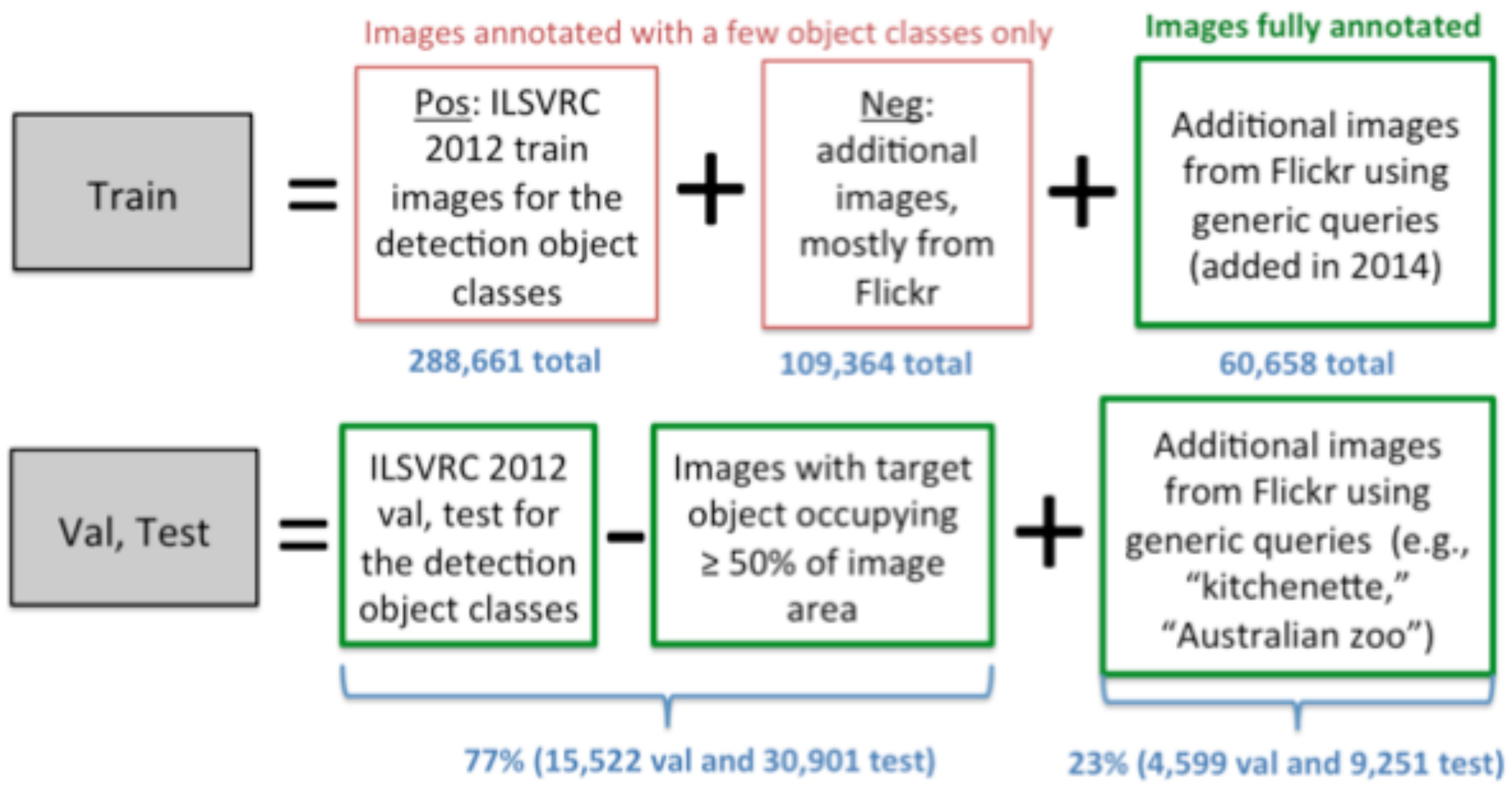
A big focus of modern dataset has been on large scale & diversity



Russakovsky & Deng et al, “ImageNet Large Scale Visual Recognition Challenge, IJCV 2015, (challenges since 2010)

Static datasets: large scale

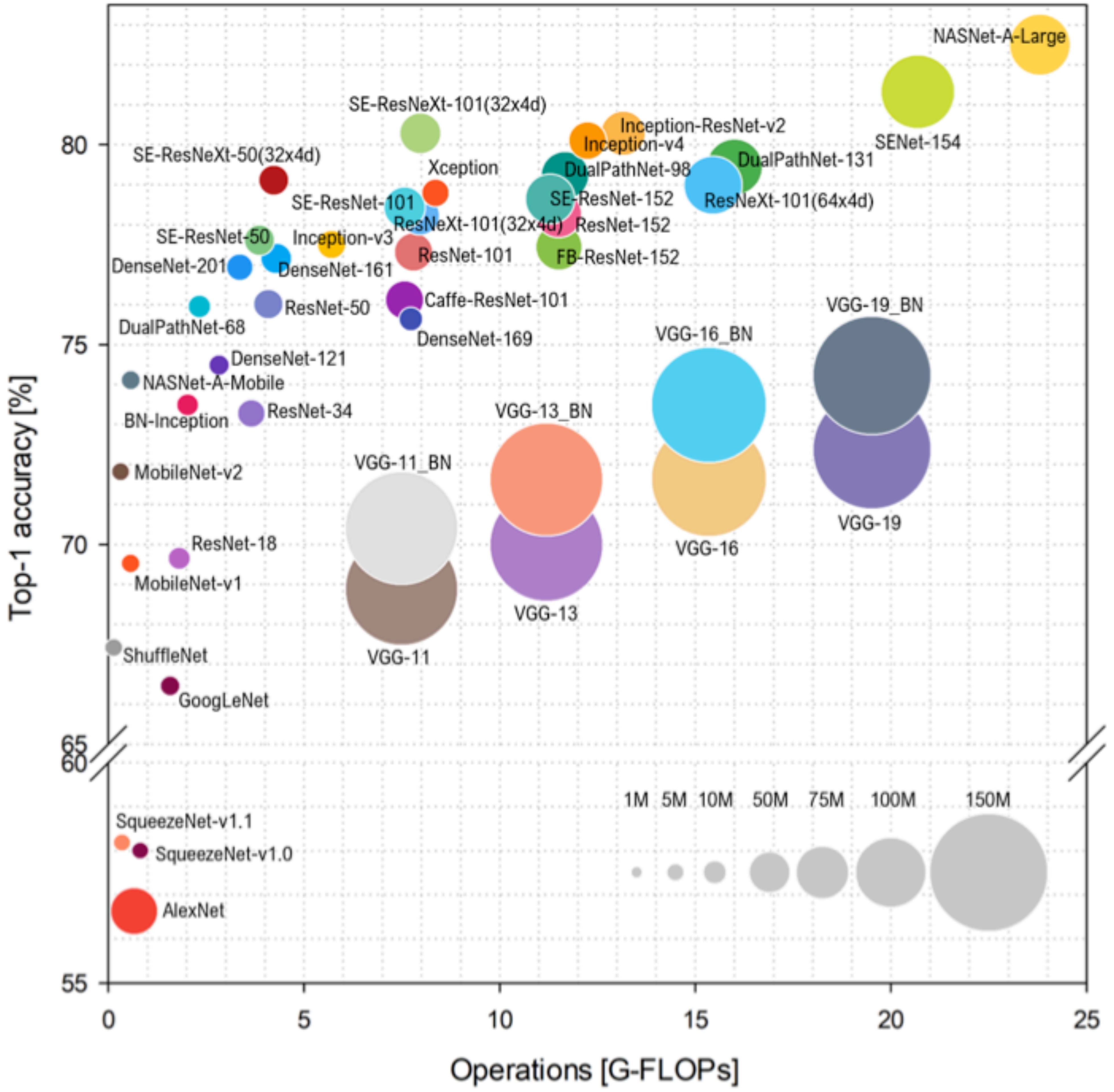
And trying to ensure reasonable train, validation, test splits through complex collection processes



Russakovsky & Deng et al, "ImageNet Large Scale Visual Recognition Challenge, IJCV 2015, (challenges since 2010)

What do you think:
should our primary goal be the solution to such benchmarks?

Static models

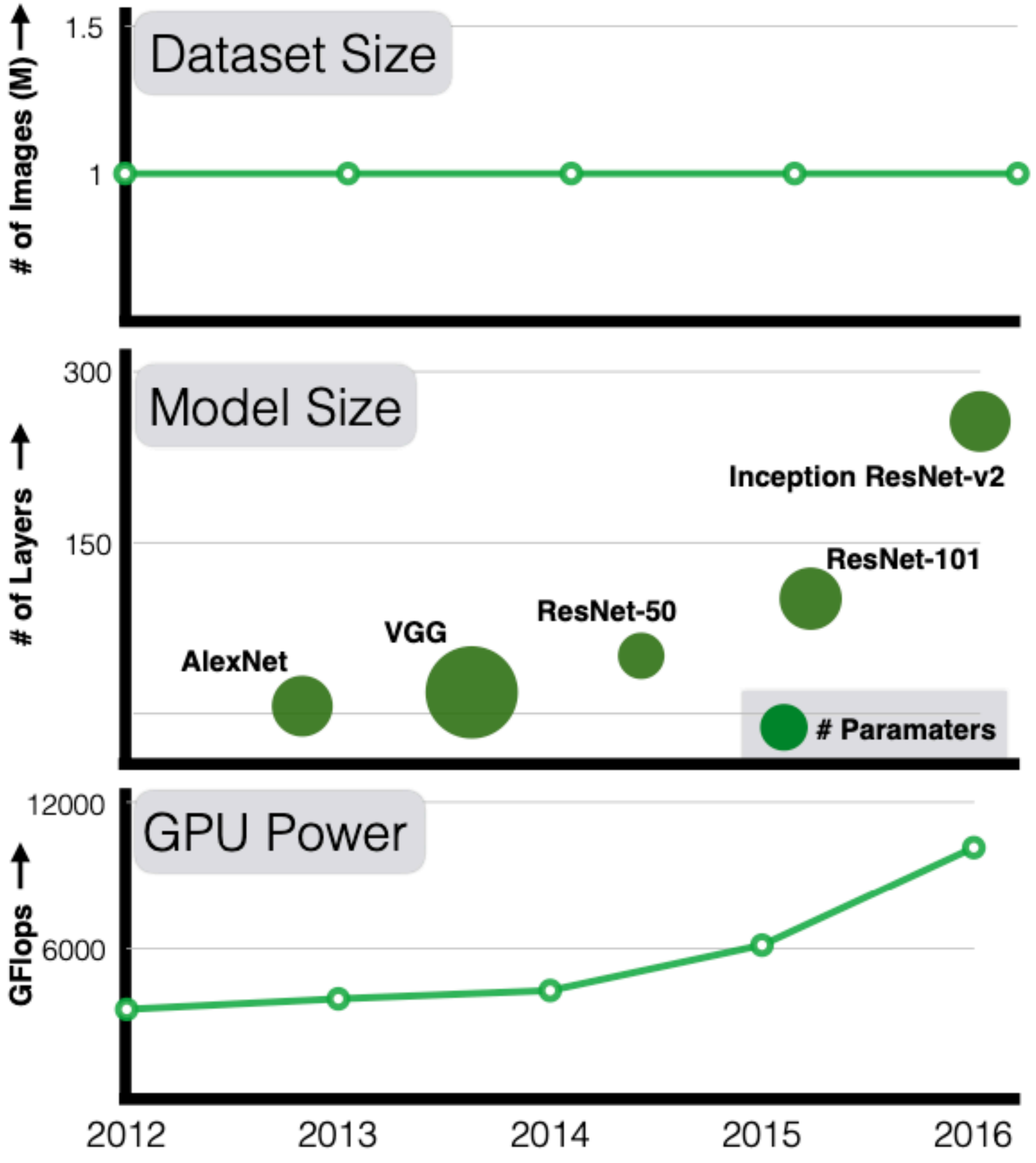


A very big emphasis has then been on “solving” such benchmarks

ImageNet is a prime example, where models & compute got bigger and more accurate over time

Bianco et al, “Benchmark Analysis of Representative Deep Neural Network Architectures”, IEEE Access, 2018

Data and model centrism



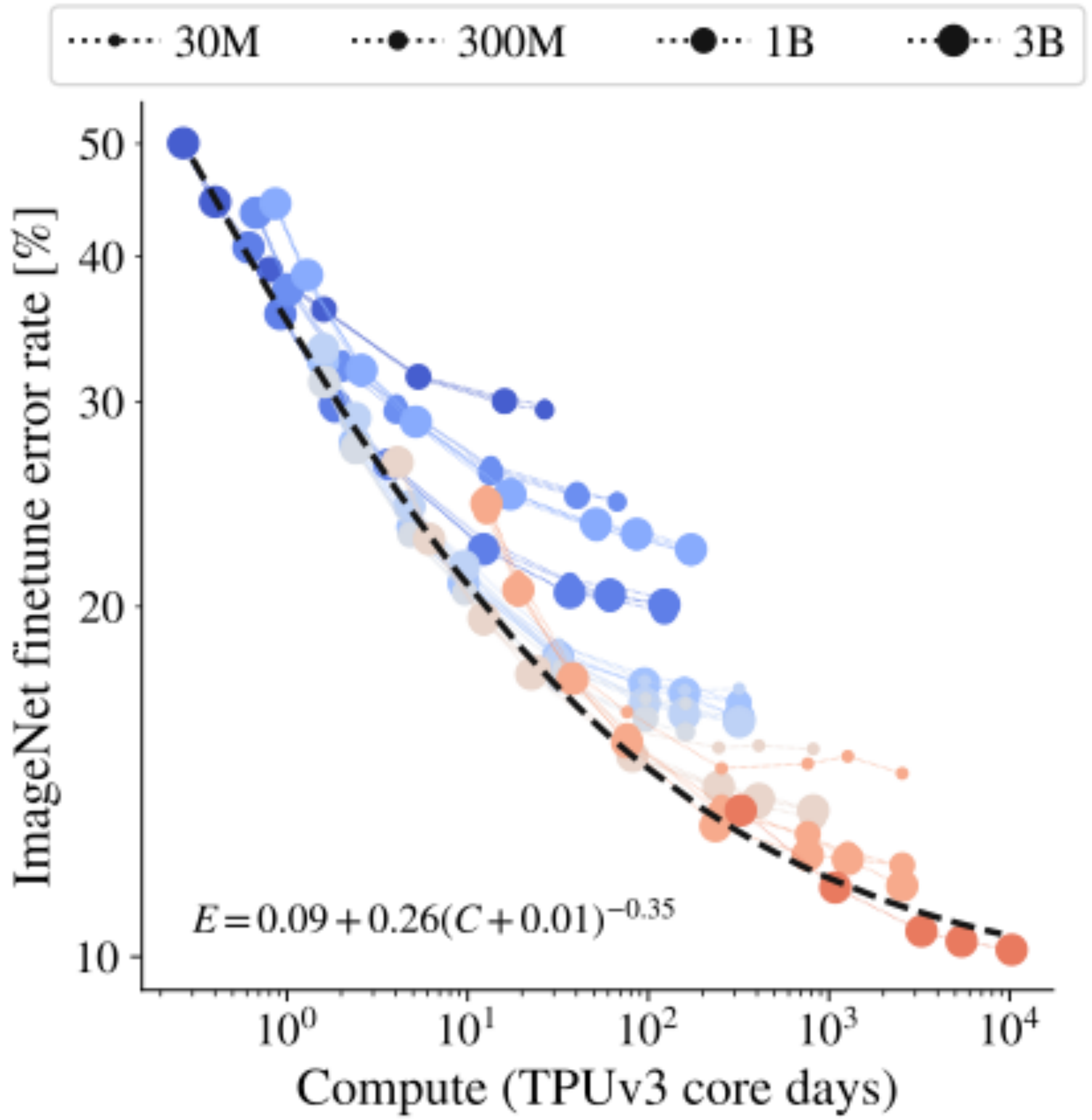
At the same time, it's often "either" models or data

For example, ImageNet has remained largely static* over time

* (excluding some concerns over fair representation)

Sun et al, "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era", ICCV 2017

Data and model centrism



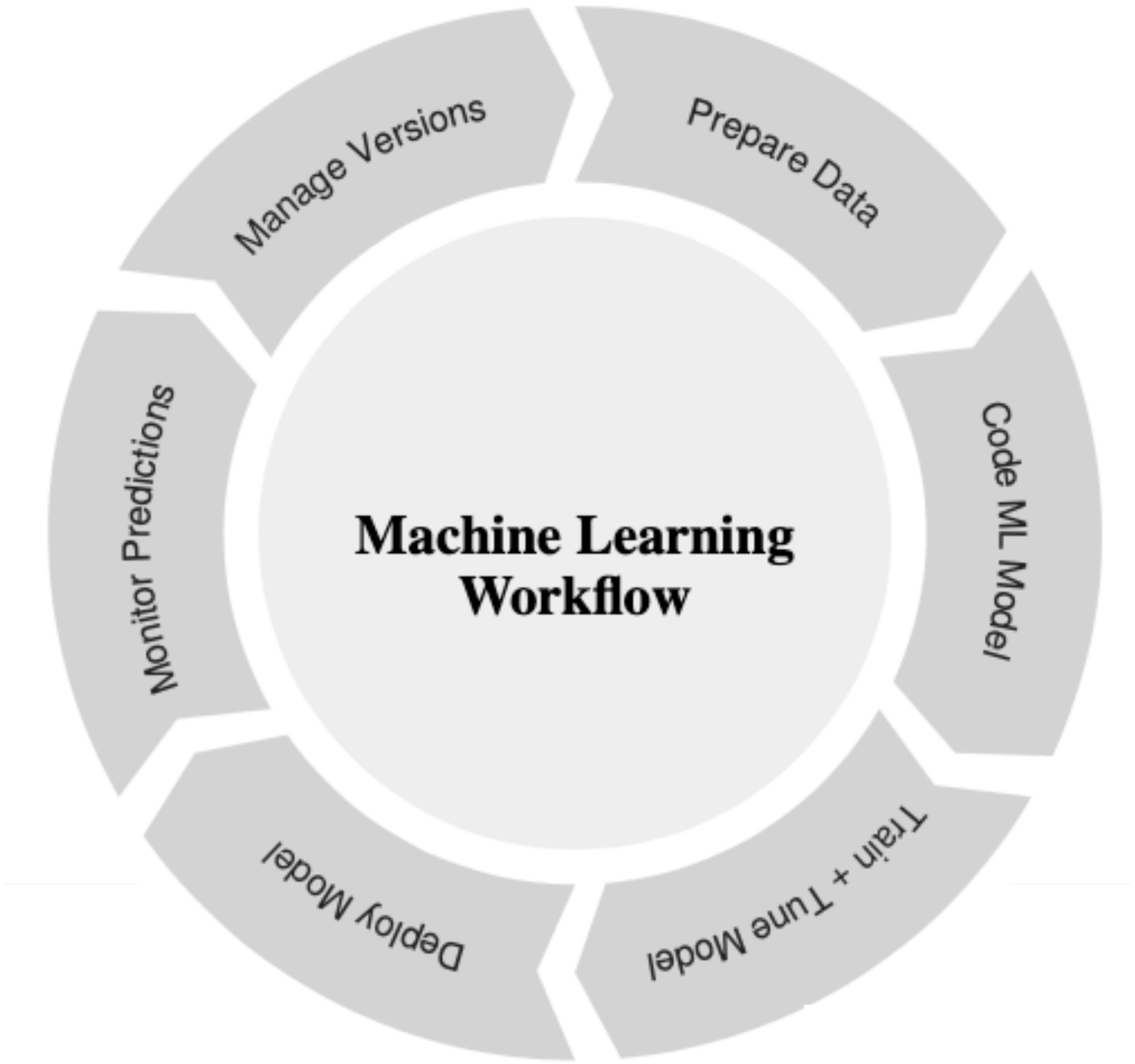
Zhao et al, "Scaling Vision Transformers", preprint 2021

Or conversely, a model is picked (here a transformer) and datasets are extended

Example from ImageNet to the (non-public) JFT 300M & JFT-3B

Let’s start moving beyond static datasets + models

Can we just iterate?



Turns out that this will be much **harder** than you perhaps expect now!

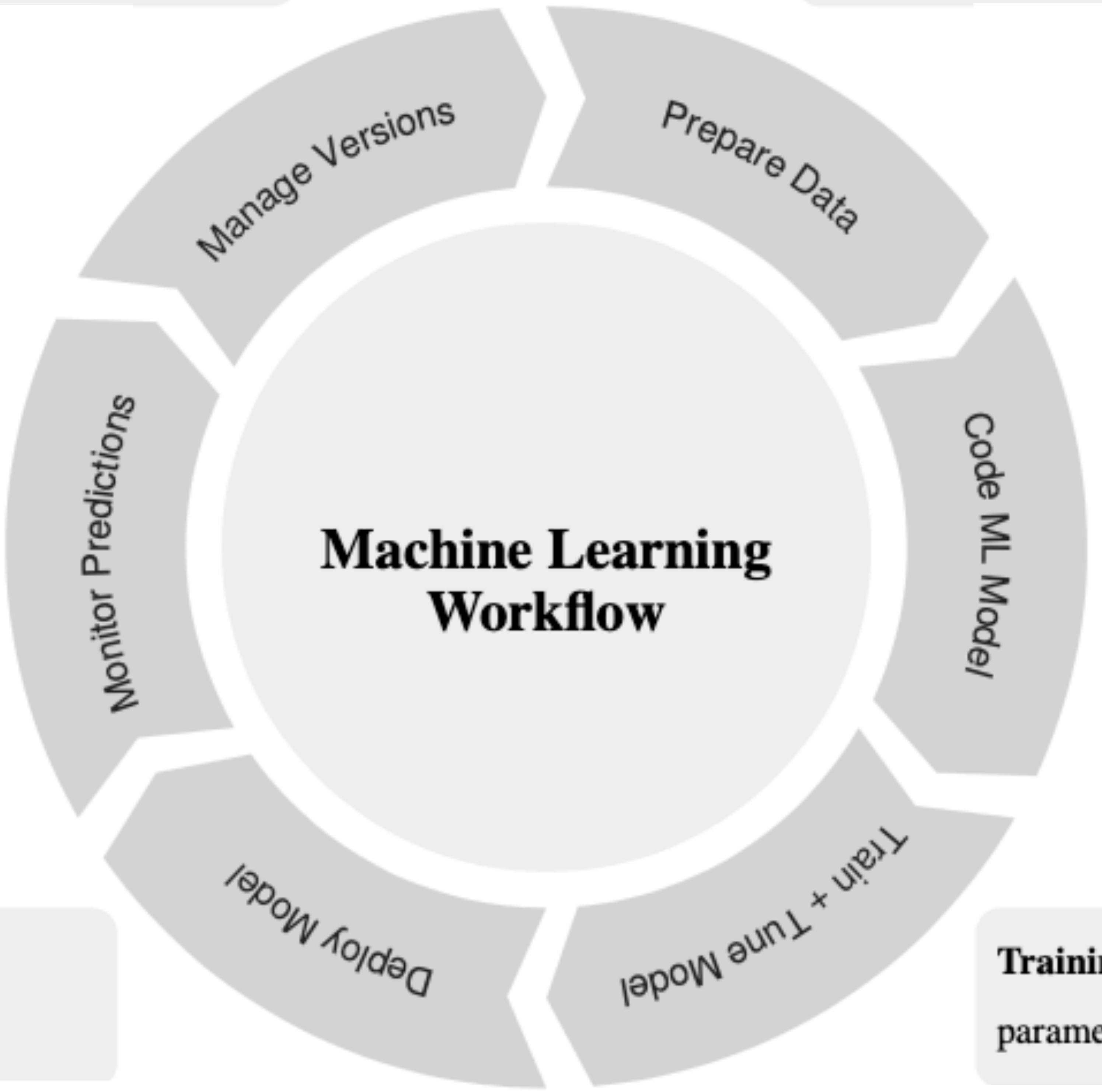
Mundt et al, “CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability”, ICLR 2022

Why? From static ML workflow ...

Versioning: stage versions according to prediction evaluation and deployment

Data: amount, redundancy vs. diversity, cleaning, preprocessing

Individual questions



Prediction: test set evaluation, failure modes and robustness

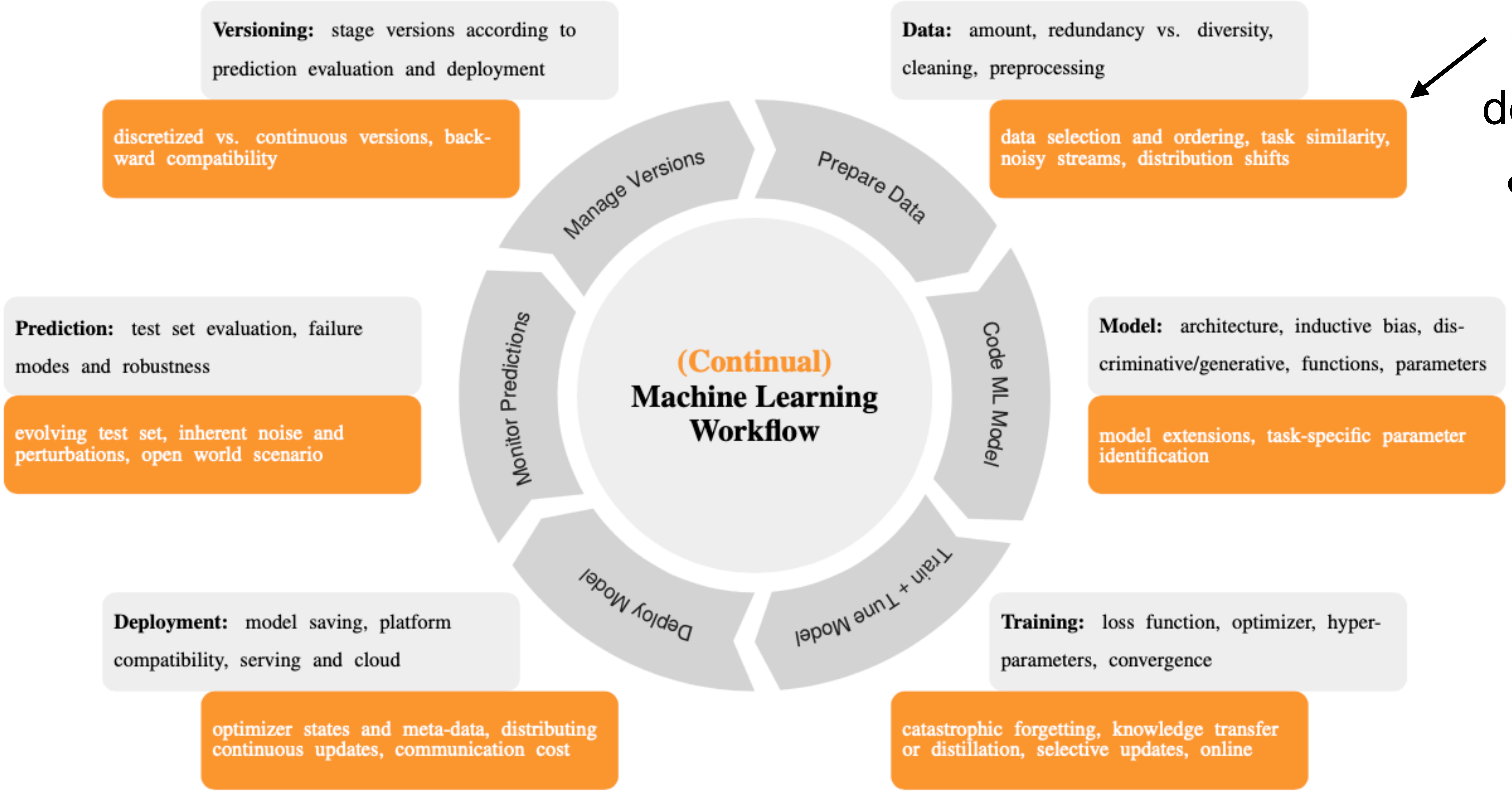
Model: architecture, inductive bias, discriminative/generative, functions, parameters

Deployment: model saving, platform compatibility, serving and cloud

Training: loss function, optimizer, hyper-parameters, convergence

Mundt et al, "CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability", ICLR 2022

… to continual/lifelong ML …



Continual dependencies & synergies

Mundt et al, “CLEVA-Compass: A Continual Learning Evaluation Assessment Compass to Promote Research Transparency and Comparability”, ICLR 2022

The first in a chain of questions:
can we transfer our models?

Early definition: lifelong ML

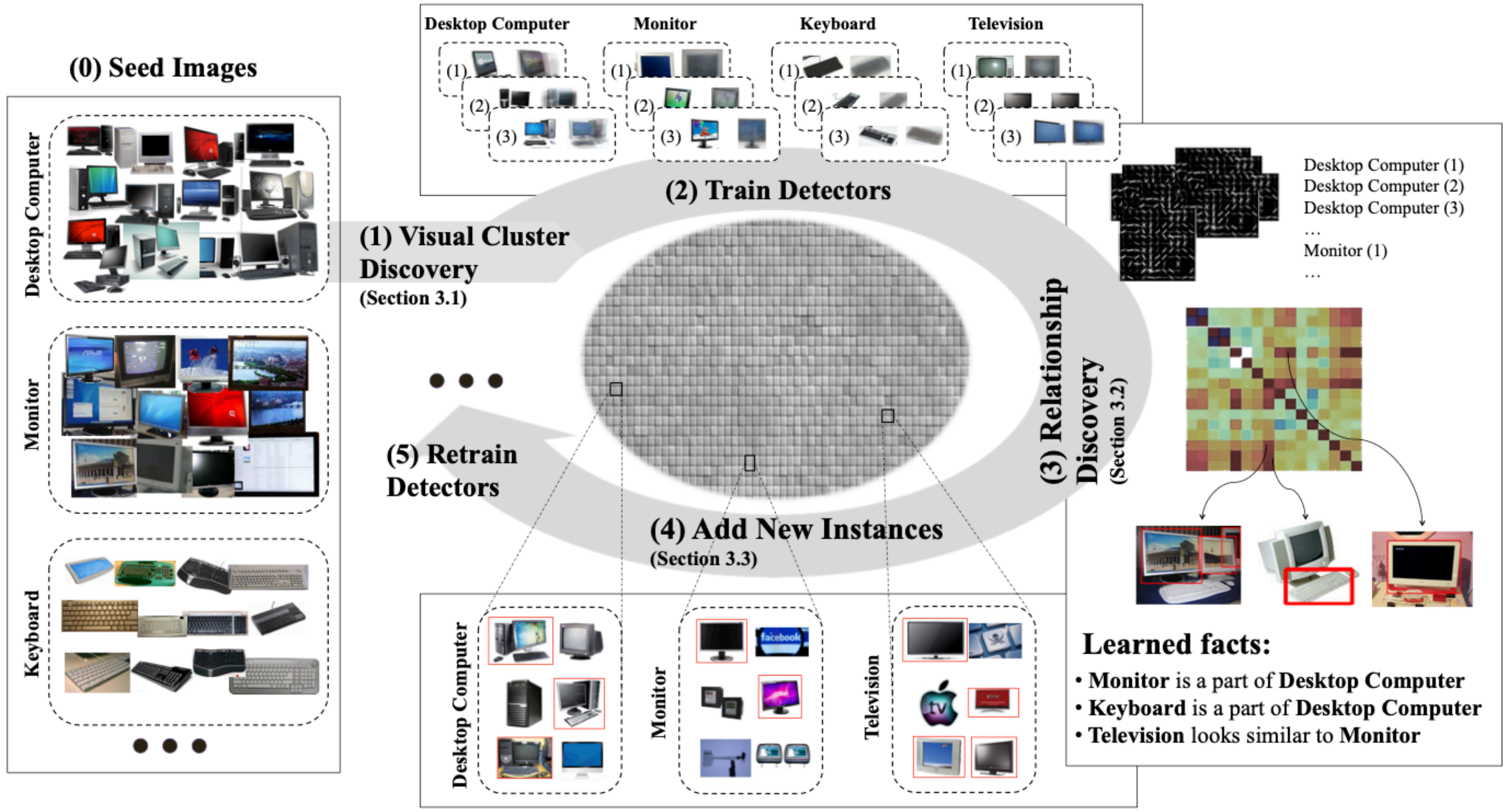
Definition - **Lifelong Machine Learning** - Thrun 1996:

“The system has performed N tasks. When faced with the (N+1)th task, it uses the knowledge gained from the N tasks to help the (N+1)th task.”

“Is Learning The n-th Thing Any Easier Than Learning the First?” (NeurIPS 1996) & “Explanation based Neural Network Learning A Lifelong Learning Approach”, Springer US, 1996

What is *knowledge* in a machine learning system?

Never-ending (language/image) learner



Knowledge is more than params

- (NELL) Ran 24/7 from 2010-2018
- Accumulated over 50 million candidate "beliefs" by reading the web
- Relational database
- Facts: barley is a grain
- Beliefs: sportUsesEquip (soccer, balls)

"Towards an Architecture for Never-Ending Language Learning", Carlson et al, AAAI 2010
"NEIL: Extracting Visual Knowledge form Web Data", X. Chen et al, ICCV 2013
"Never-Ending Learning", T. Mitchell et al, AAAI 2015

Early definition: lifelong ML

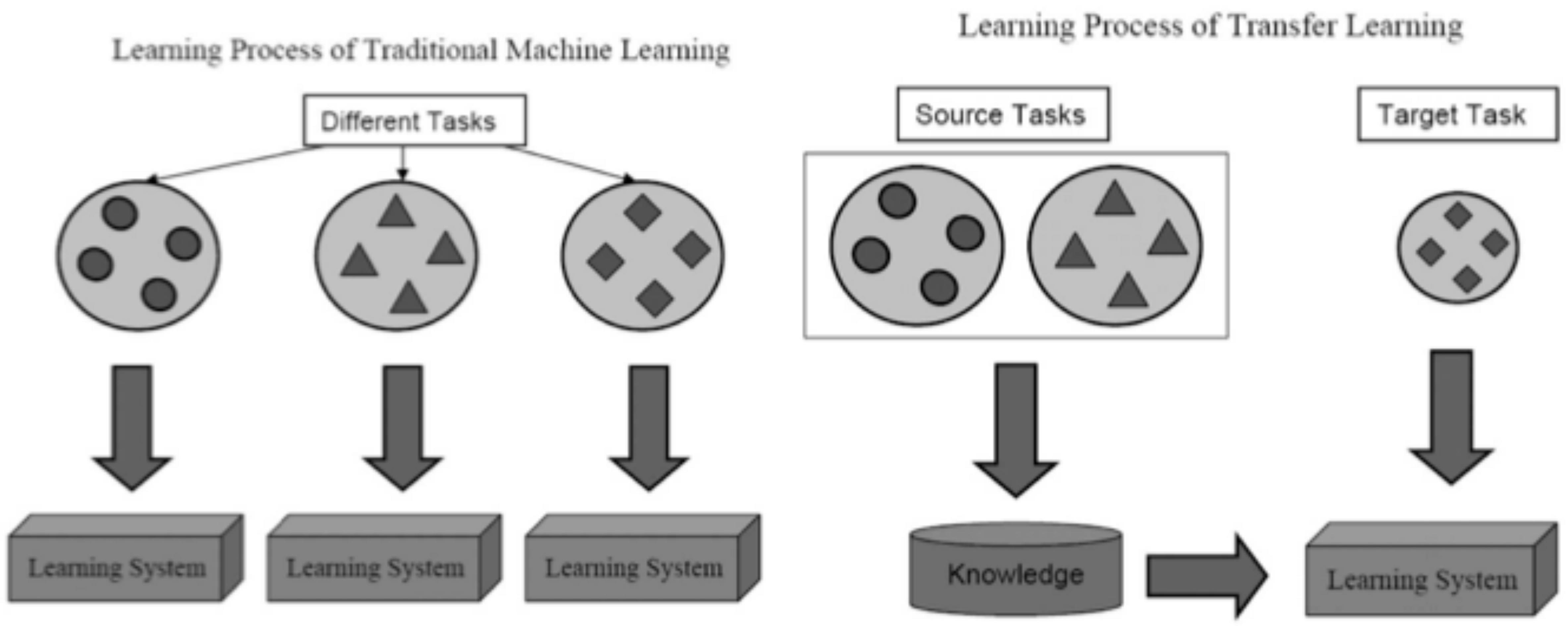
Definition - **Lifelong Machine Learning** - Thrun 1996:

“The system has performed N tasks. When faced with the (N+1)th task, it uses the knowledge gained from the N tasks to help the (N+1)th task.”

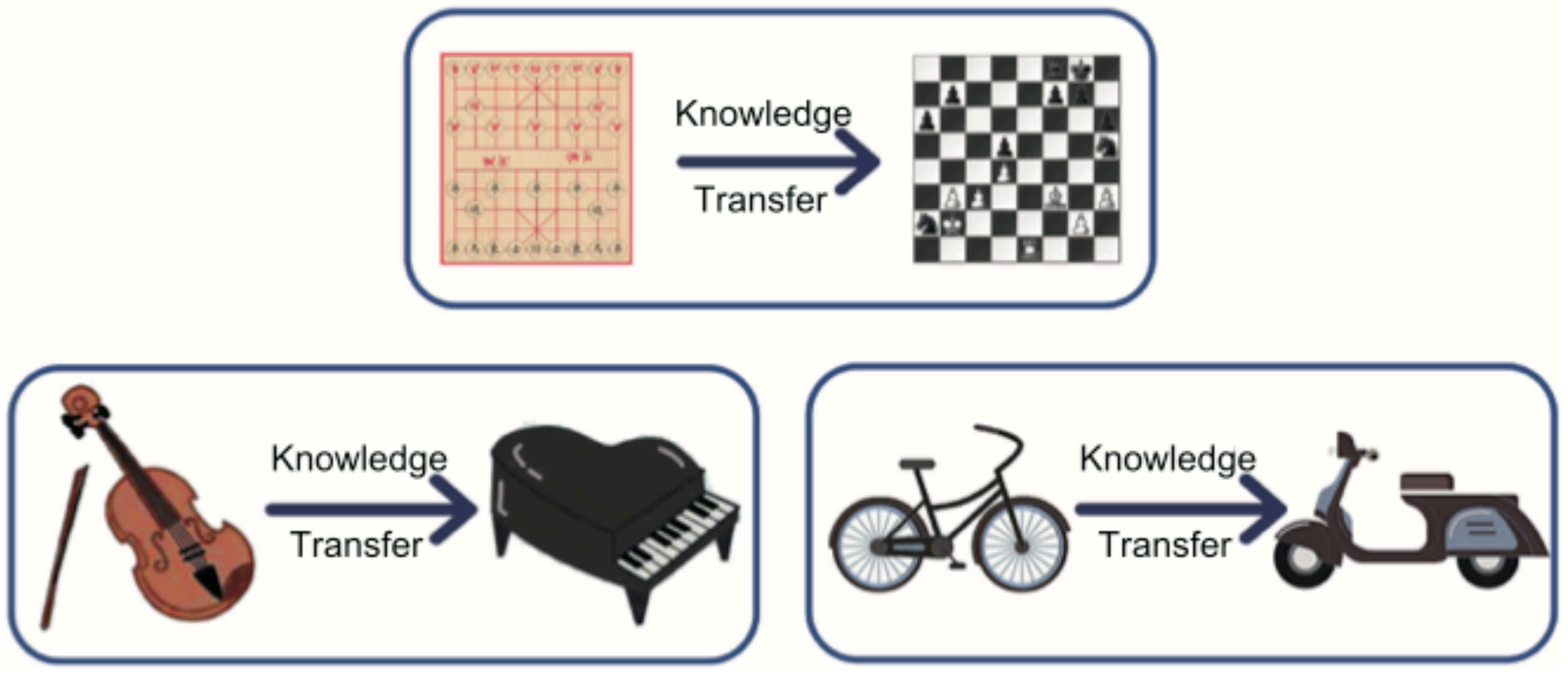
- Is data accumulated? Stored?
- What are the ways to “help” the (N+1)th task?
- What is knowledge? What is a task?
- ….

“Is Learning The n-th Thing Any Easier Than Learning the First?” (NeurIPS 1996) & “Explanation based Neural Network Learning A Lifelong Learning Approach”, Springer US, 1996

Transfer learning



“A Survey on Transfer Learning”, Pan and Yang, IEEE Transactions on Knowledge & Data Engineering, 2010

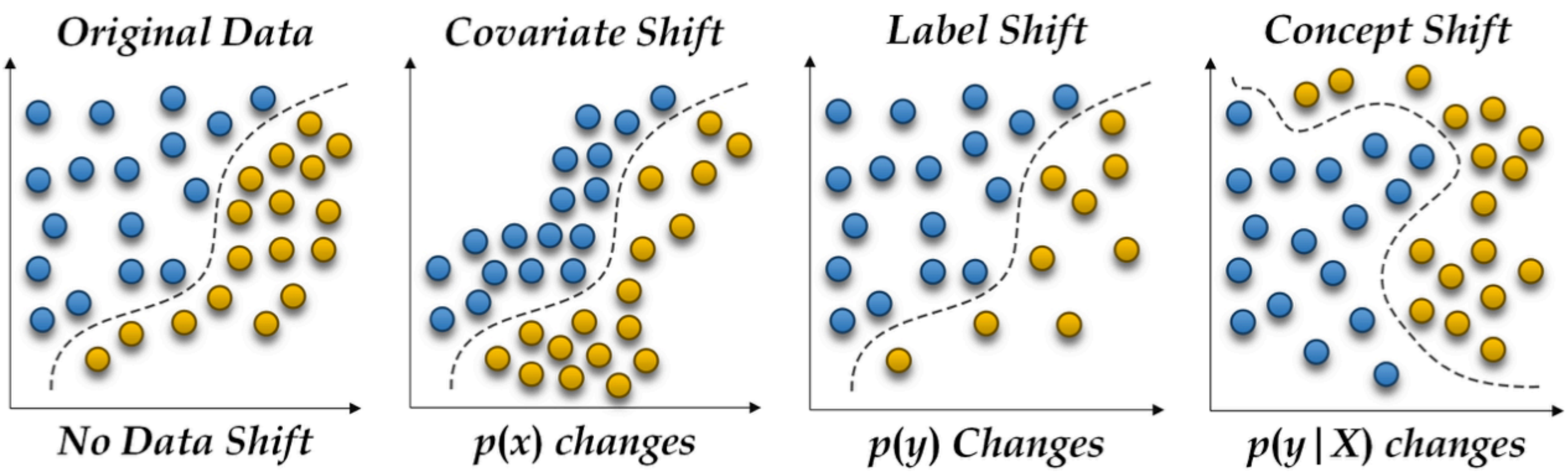


“A Comprehensive Survey on Transfer Learning”, Zhuang et al, Proceedings of IEEE, 2020

“Help the (N+1th) task!”: Assume that we already have “knowledge”/ a model based on initial task(s) -> the essence of transfer learning

What types of data *shifts* can you think of?

Dataset shifts



(a) Original data (b) Covariate shift (c) Label shift (d) Concept shift

Figure from "Understanding Dataset Shift and Potential Remedies", Vector Institute Technical Report, 2021

See also: "Dataset Shift in Machine Learning" book, MIT Press 2009

Transfer learning: definition

Definition - Transfer Learning - Pan & Yang 2009:
"*Given a source domain* D_S *and learning task* $\mathcal{T}_S$, *a target domain* D_T *and learning task* $\mathcal{T}_T$, *transfer learning aims to help improve the learning of the target predictive function* $f_T(\,.\,)$ *in* D_T *using the knowledge in* D_S *and* $\mathcal{T}_S$, *where* $D_S \neq D_T$ *or* $\mathcal{T}_s \neq \mathcal{T}_T$."

- Domain D
- Task $\mathcal{T}$
- Source S
- Target T

"A Survey on Transfer Learning", Pan & Yang, IEEE Transactions on Knowledge and Data Engineering 22(10), 2009

Transfer learning: definition

Definition - Domain & Task - Pan & Yang 2009:
"*Given a specific domain,* $D = \{\mathcal{X}, p(x)\}$*, a task consists of two components: a label space Y and an objective predictive function* $f()$ *(denoted by* $T = \{Y, f()\}$*, which is not observed but can be learned from the training data, which consist of pairs* $\{x^{(n)}, y^{(n)}\}$*, where* $x^{(n)} \in X$ and $y^{(n)} \in Y$*.*"

- Domain D: a pair of data distribution $p(x)$ and corresponding feature space $\mathcal{X}$
- Task $\mathcal{T}$: find a function f() (to map to labels in the case of supervision)
- Where generally $\mathcal{X}_S \neq \mathcal{X}_T$ or $p_S(x) \neq p_T(x)$

"A Survey on Transfer Learning", Pan & Yang, IEEE Transactions on Knowledge and Data Engineering 22(10), 2009

Transductive transfer

Definition - Transductive Transfer Learning - Pan & Yang 2009:
"*Given a source domain* D_S *and learning task* $\mathcal{T}_S$, *a target domain* D_T *and learning task* $\mathcal{T}_T$, *transductive transfer learning aims to help improve the learning of the target predictive function* $f_T(\,.\,)$ *in* D_T *using the knowledge in* D_S *and* $\mathcal{T}_S$, *where* $D_S \neq D_T$ *and* $\mathcal{T}_s = \mathcal{T}_T$."

- Feature spaces between the source and target are different $\mathcal{X}_S \neq \mathcal{X}_T$
- Feature spaces between source and target are the same, but $p_S(x) \neq p_T(x)$
- Frequently encountered as **domain adaptation** or **sample selection bias**

"A Survey on Transfer Learning", Pan & Yang, IEEE Transactions on Knowledge and Data Engineering 22(10), 2009

Inductive transfer

Definition - Inductive Transfer Learning - Pan & Yang 2009:
"*Given a source domain* D_S *and learning task* $\mathcal{T}_S$, *a target domain* D_T *and learning task* $\mathcal{T}_T$, *inductive transfer learning aims to help improve the learning of the target predictive function* $f_T(\,.\,)$ *in* D_T *using the knowledge in* D_S *and* $\mathcal{T}_S$, *where* $\mathcal{T}_s \neq \mathcal{T}_T$."

(Labeled) data points are required to "induce" the target predictive function

"A Survey on Transfer Learning", Pan & Yang, IEEE Transactions on Knowledge and Data Engineering 22(10), 2009

What do you think are the central questions & measures of success for transfer learning?

Transfer: questions & goals

(Some) **central questions**

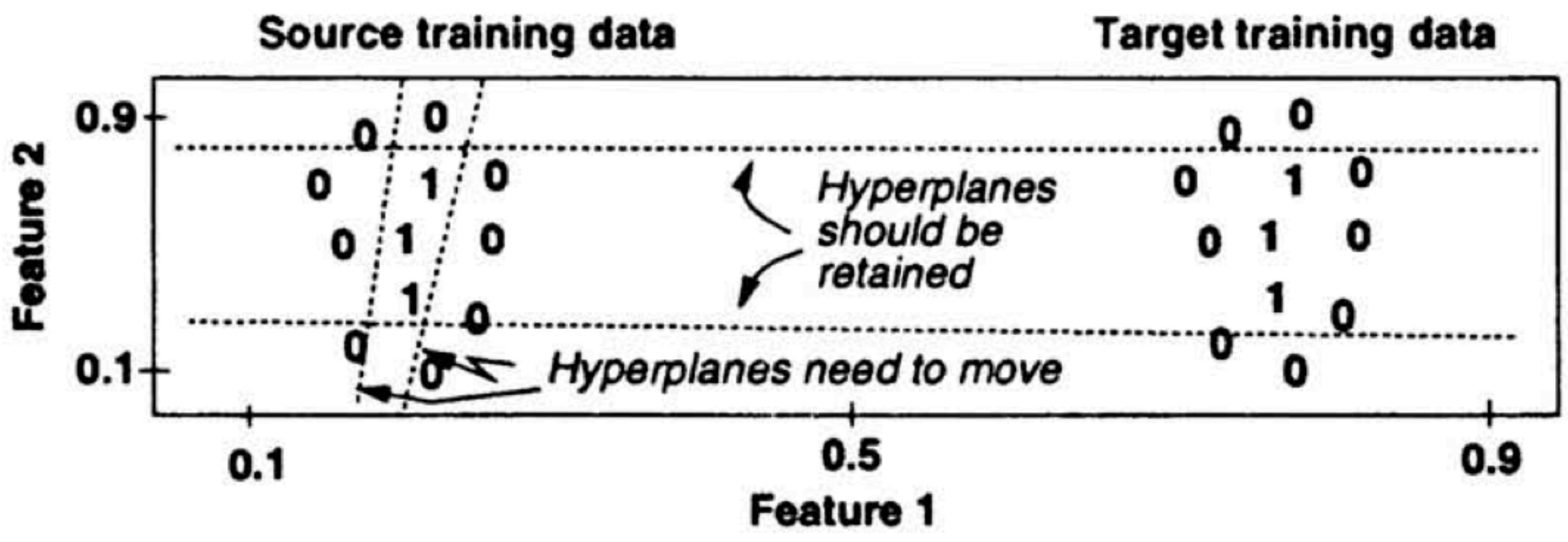
1. What to transfer: some knowledge is domain or task specific or may be more general/ transferable
2. When to transfer: when does transfer help or when does it even hurt?
3. How to transfer: algorithms to actually include, transfer/combine knowledge

(Some) **central objectives**

1. Improved loss/more accurate function in direct comparison to learning just on the target
2. Accelerate learning
3. Reduce data dependence (of target)

Examples of transfer learning approaches

Transductive transfer



Early approaches transfer by identifying the amount that a specific hyperplane helps to separate the data into different classes (& then reweighting/reinitializing).

"Discriminability-Based Transfer between Neural Networks", L. Y. Pratt, NeurIPS 1992

Transductive transfer

A domain adaptation example through feature transformation

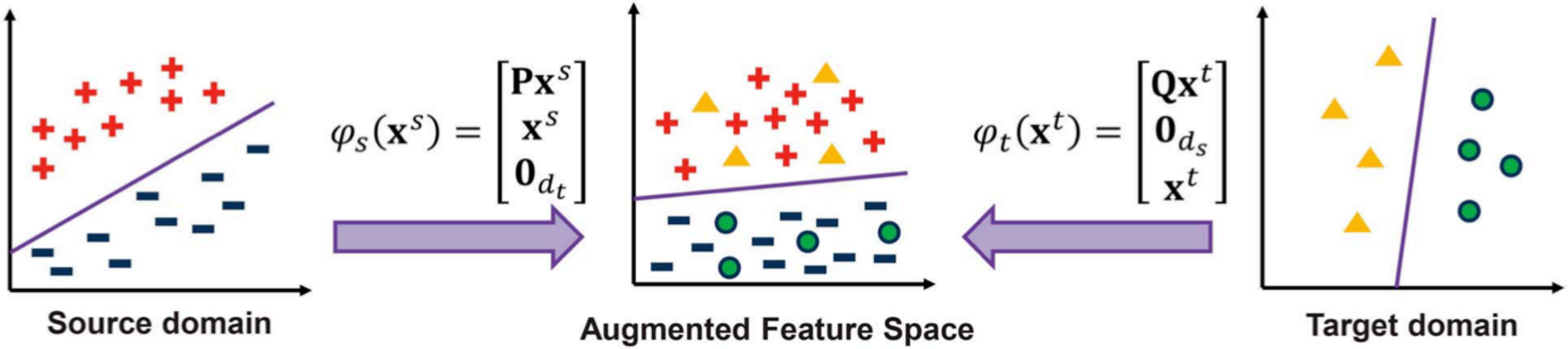
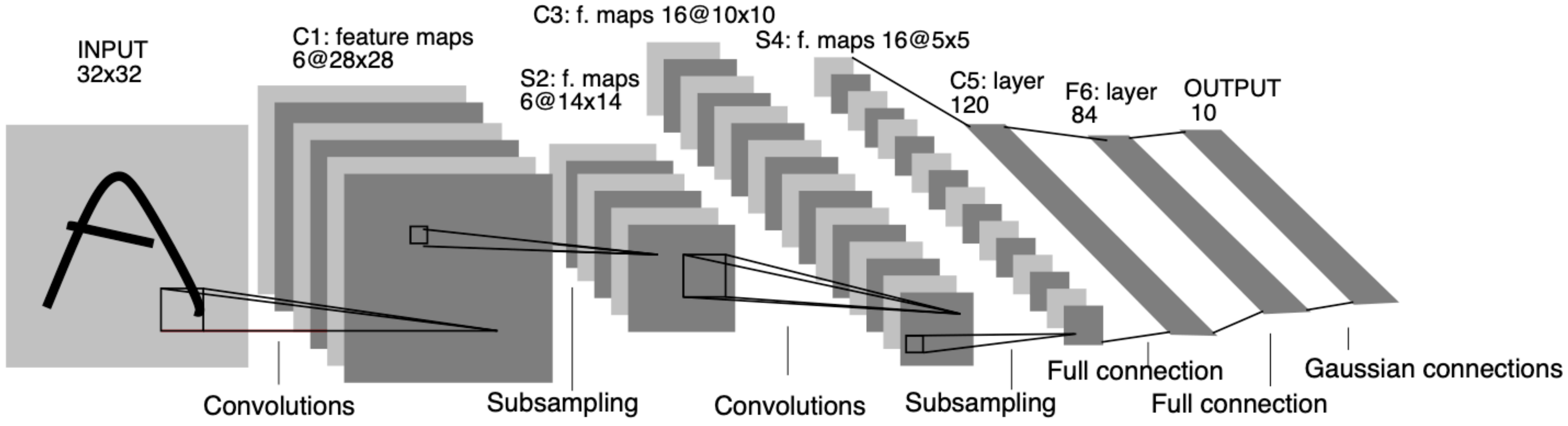


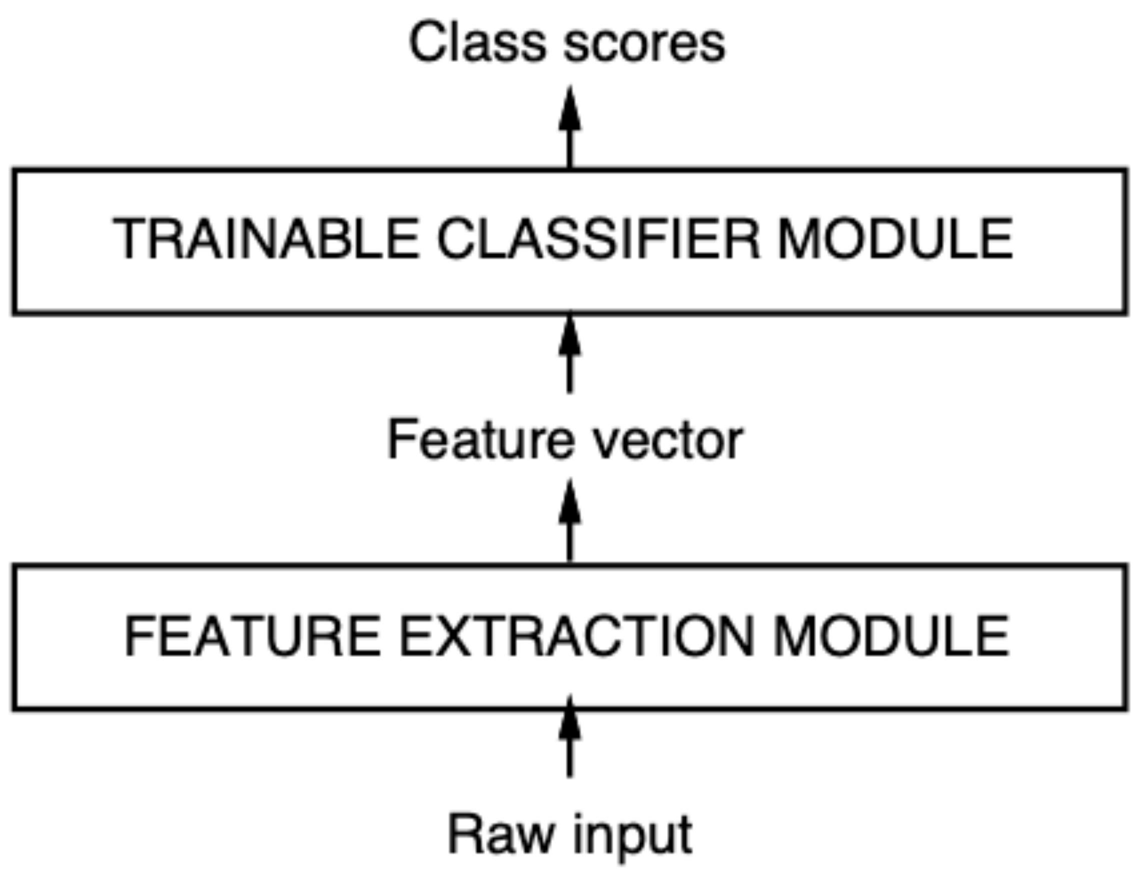
Fig. 1. Samples from different domains are represented by different features, where red crosses, blue strips, orange triangles and green circles denote source positive samples, source negative samples, target positive samples and target negative samples, respectively. By using two projection matrices **P** and **Q**, we transform the heterogenous samples from two domains into an augmented feature space.

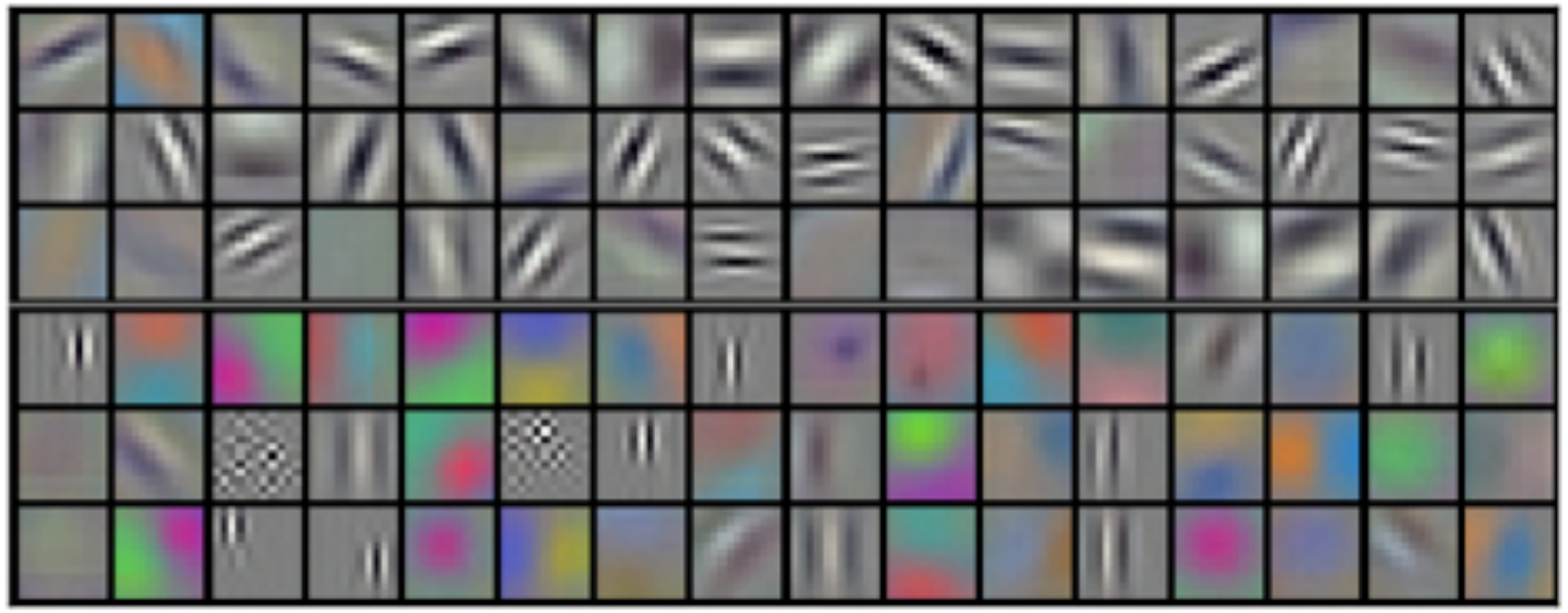
"Learning with augmented Features for Supervised and Semi-Supervised Heterogeneous Domain Adaptation", Wen Li et al, TPAMI 2014

“Gradient-Based Learning Applied to Document Recognition”, LeCun et al, Proceedings of the IEEE, 1998



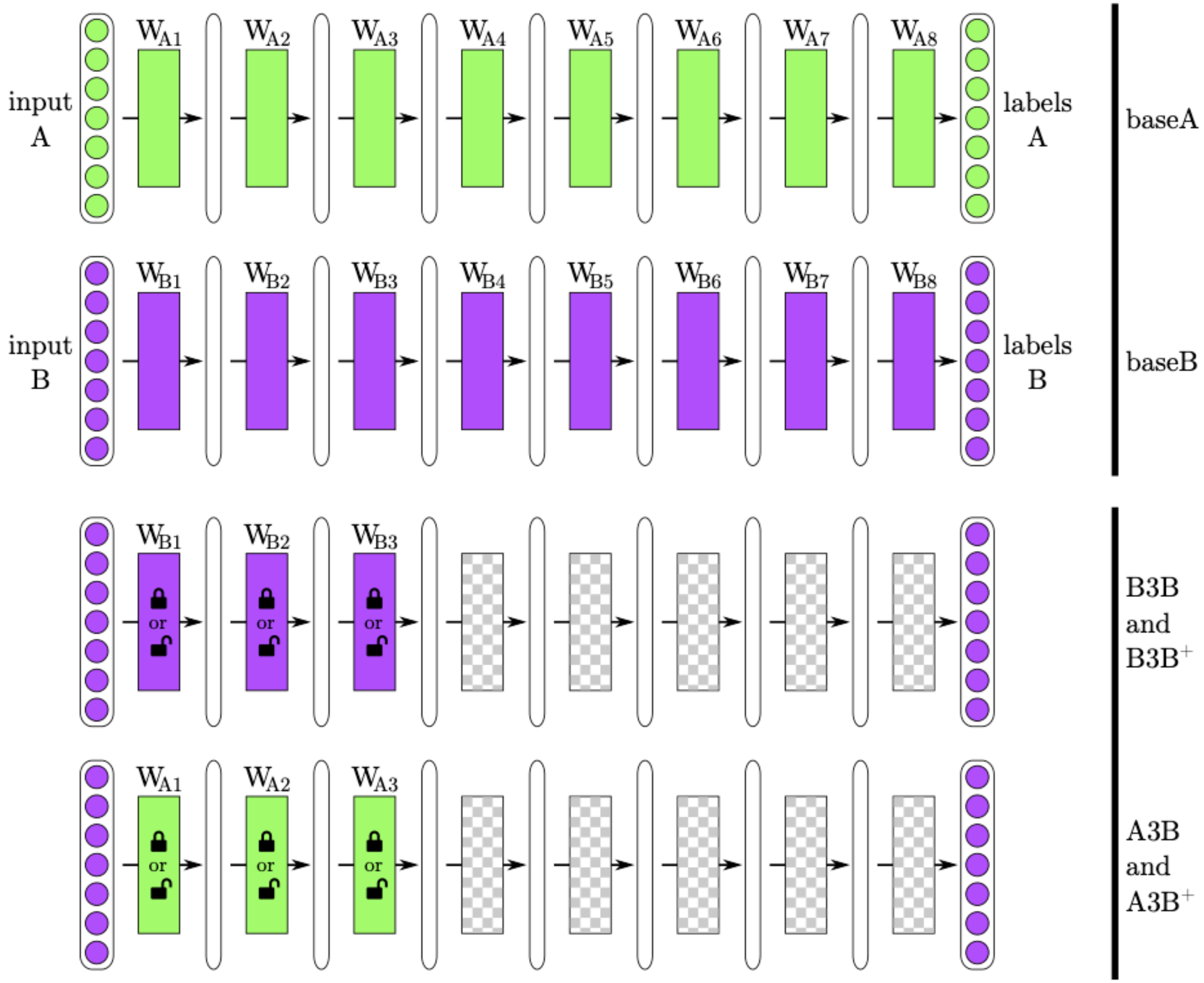
Transfer learning in deep learning





“ImageNet Classification with Deep Convolutional Neural Networks”, Krizhevsky et al, NeurIPS 2012

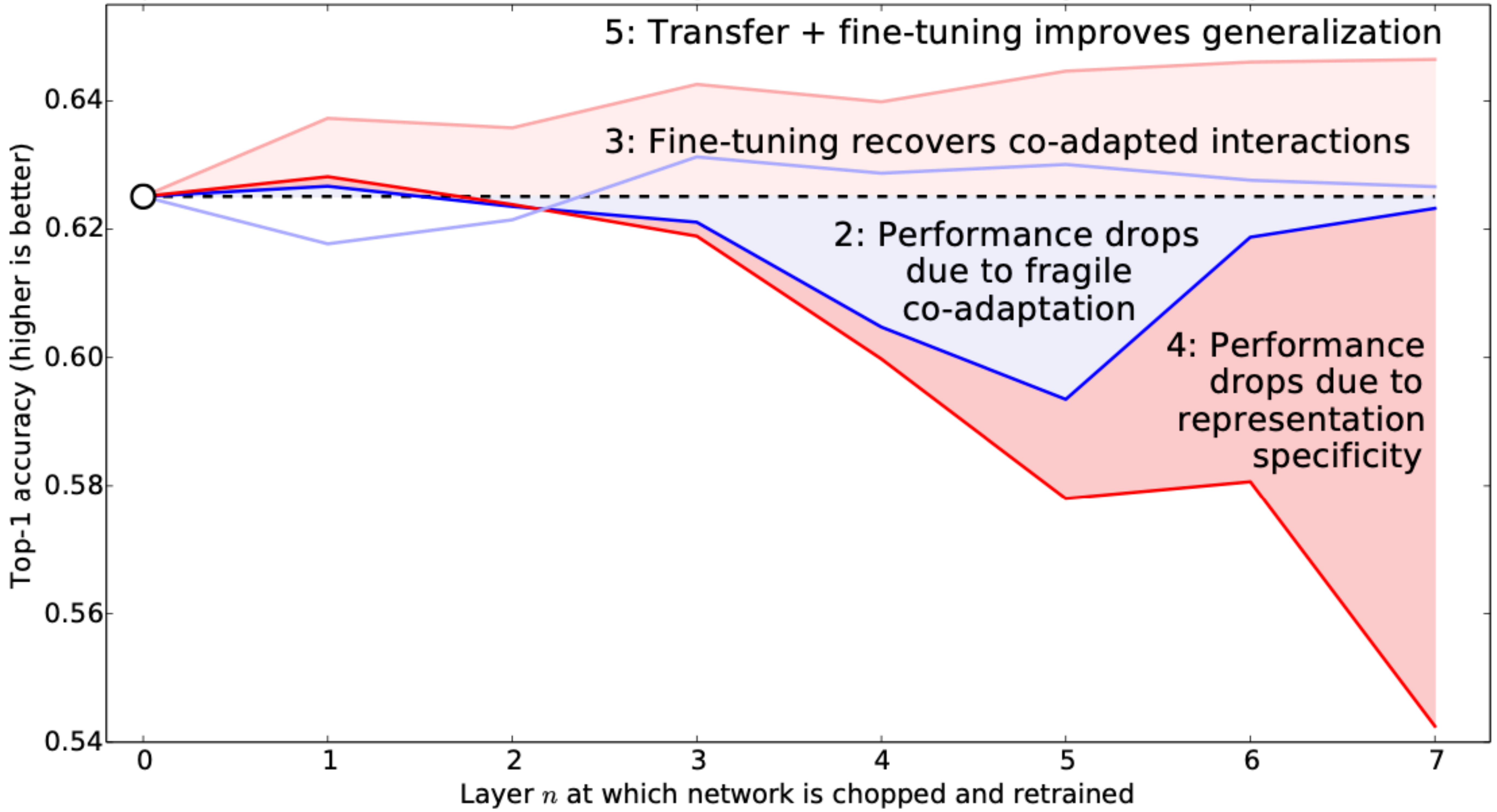
(Inductive) ImageNet transfer



- Split Imagenet into 2 sets of 500 classes: A and B
- "Lock" different sets of layers/ representations & randomly initialize upper remaining layers
- Alternatively: continue training/ fine-tuning transferred layers

"How transferable are features in deep neural networks", Yosinski et al, NeurIPS 2014

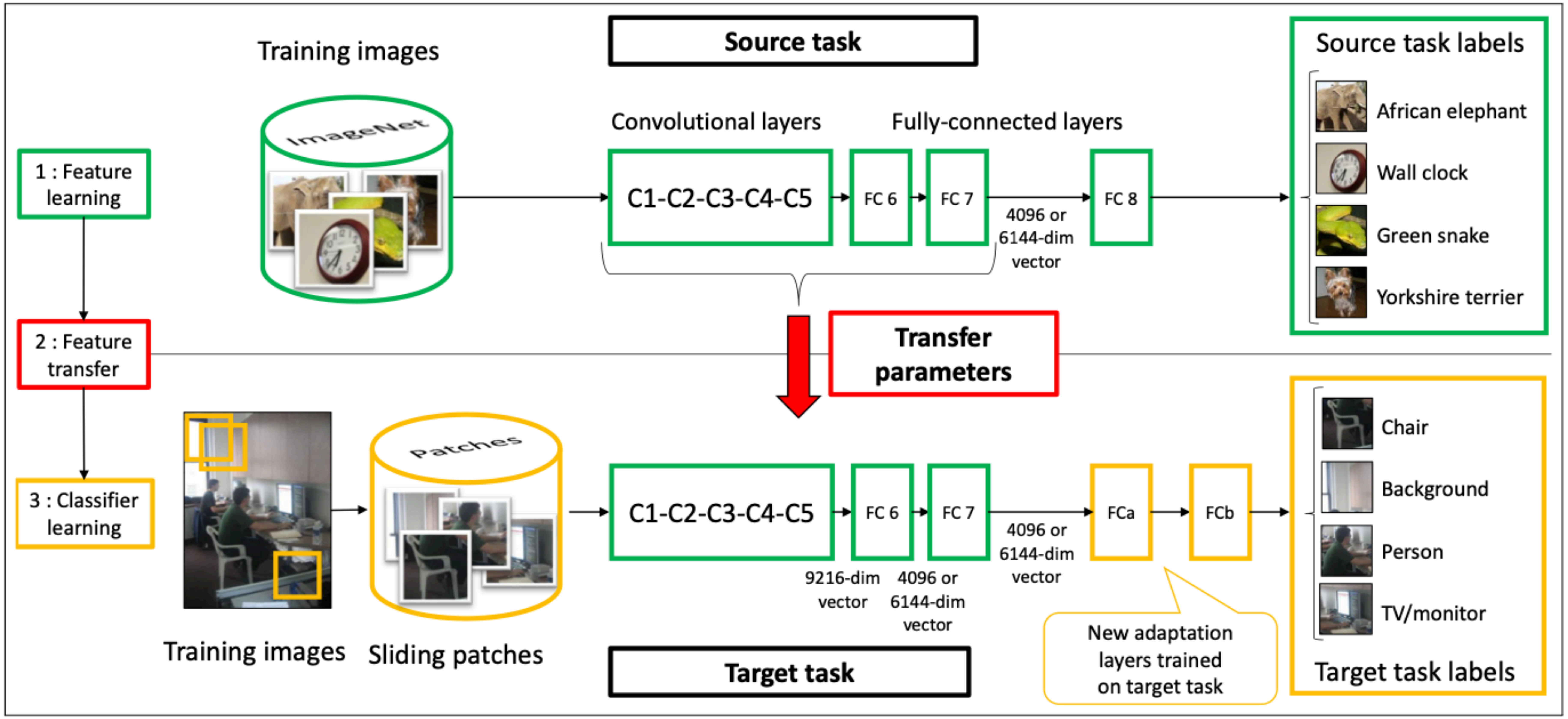
(Inductive) ImageNet transfer



2. B-B: copied from B and frozen + random rest trained on B
3. B-B+: copied features are allowed to adapt/fine-tune
4. A-B: transfer from A to B with frozen layers
5. A-B+: transferring + fine-tuning from A to B

“How transferable are features in deep neural networks”, Yosinski et al, NeurIPS 2014

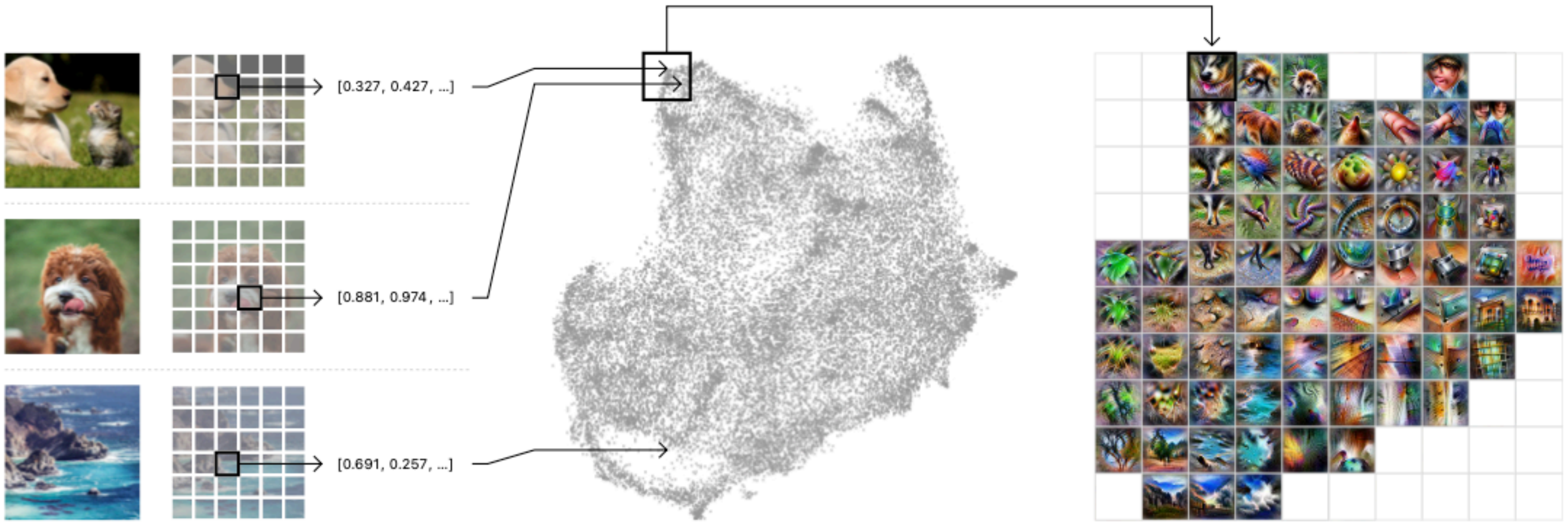
(Inductive) ImageNet transfer



"Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks", Oquab et al, CVPR 2014

The role of embeddings:
few-shot to one-shot transfer

The role of embeddings



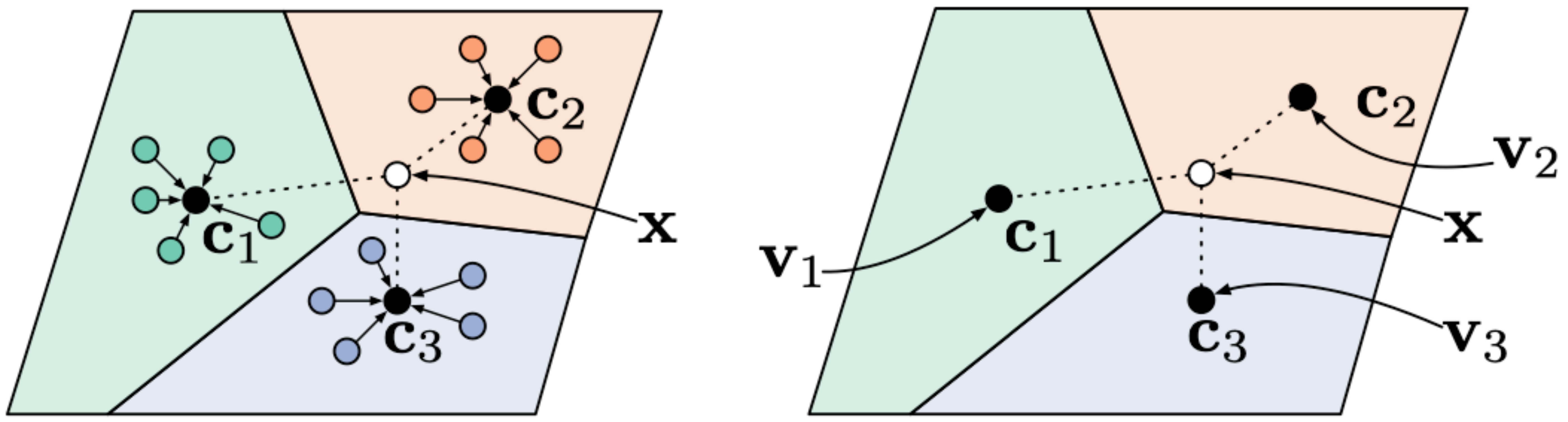
A randomized set of one million images is fed through the network, collecting one random spatial activation per image.

The activations are fed through UMAP to reduce them to two dimensions. They are then plotted, with similar activations placed near each other.

We then draw a grid and average the activations that fall within a cell and run feature inversion on the averaged activation. We also optionally size the grid cells according to the density of the number of activations that are averaged within.

“Activation Atlas”, Carter et al, Distill 2019

Special cases of transfer: few-shot learning



(a) Few-shot (b) Zero-shot

Figure 1: Prototypical networks in the few-shot and zero-shot scenarios. **Left**: Few-shot prototypes $\mathbf{c}_k$ are computed as the mean of embedded support examples for each class. **Right**: Zero-shot prototypes $\mathbf{c}_k$ are produced by embedding class meta-data $\mathbf{v}_k$. In either case, embedded query points are classified via a softmax over distances to class prototypes: $p_\phi(y = k|\mathbf{x}) \propto \exp(-d(f_\phi(\mathbf{x}), \mathbf{c}_k))$.

"Prototypical Networks for Few-shot Learning", Snell et al, NeurIPS 2017

See also "Object Classification from a Single Example Utilizing Class relevance Metrics", M. Fink, NeurIPS 2004 & "One-shot Learning of Object Categories", Fei-Fei et al, TPAMI 2006

Compute prototype c as the mean vector of each class with parametrized embedding function of a support set of labelled examples

Given a distance function d, classify according to softmax over distances to the prototypes in embedding space

Special cases of transfer: one-shot learning

"We say that a set of classes is $\gamma > 0$ separated with respect to a distance function d if for any pair of examples belonging to the same class $\{(x_1, c), (x_1', c)\}$, the distance $d(x_1, x_1')$ is smaller than the distance between any pair of examples from different classes $\{(x_2, e), (x_2', g)\}$by at least γ: $d(x_1, x_1') \leq d(x_2, x_2') - \gamma$."

1. Learn from extra sample a distance function d that achieves γ separation
2. Learn a nearest neighbor classifier, where the classifier employs d

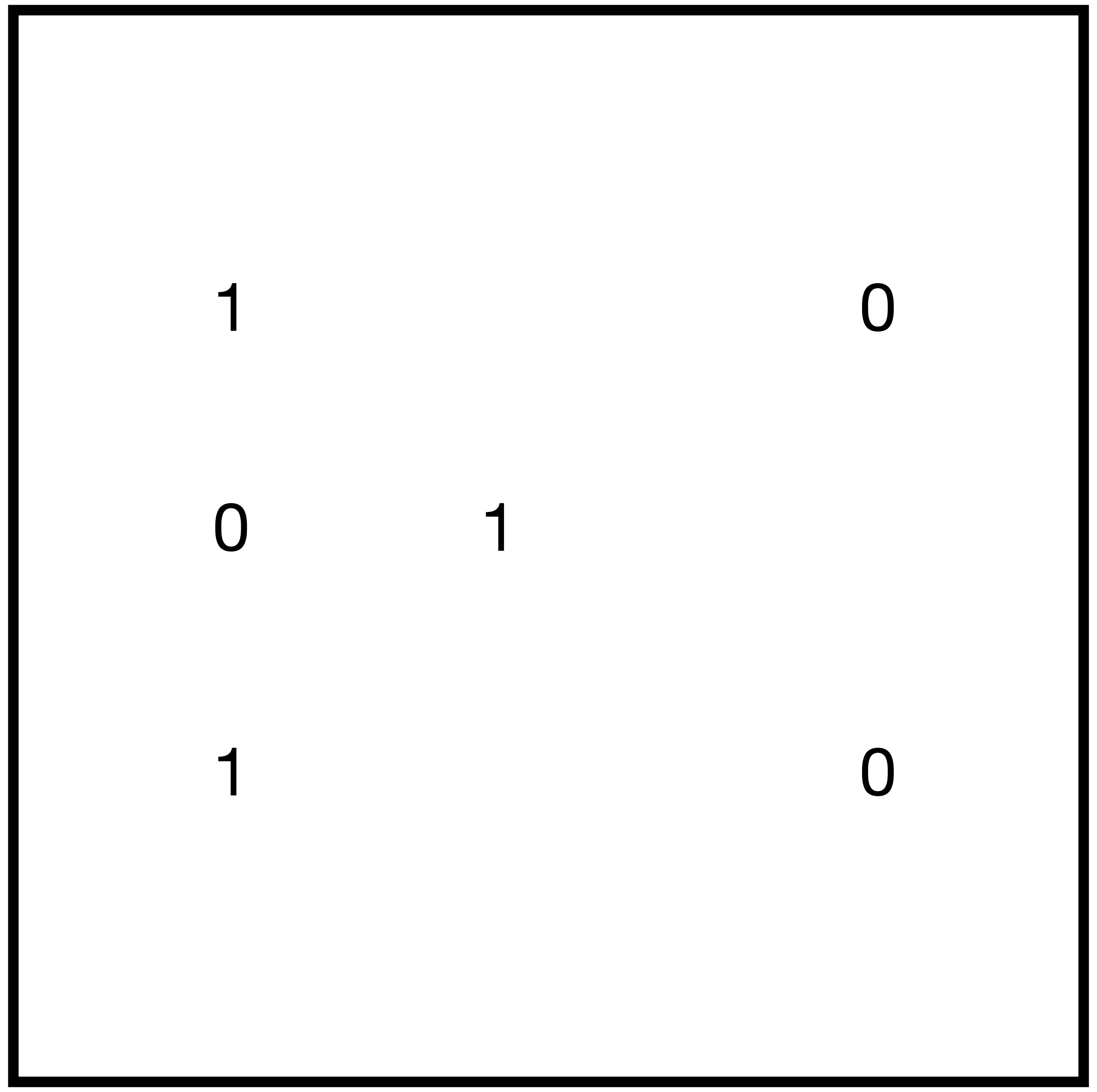
"Object Classification from a Single Example Utilizing Class relevance Metrics", M. Fink, NeurIPS 2004

See also "One-shot Learning of Object Categories", Fei-Fei et al, TPAMI 2006

Why is transfer challenging?

Transfer challenges

How would you separate this data with a set of hyperplanes? (Try 3)



Transfer challenges

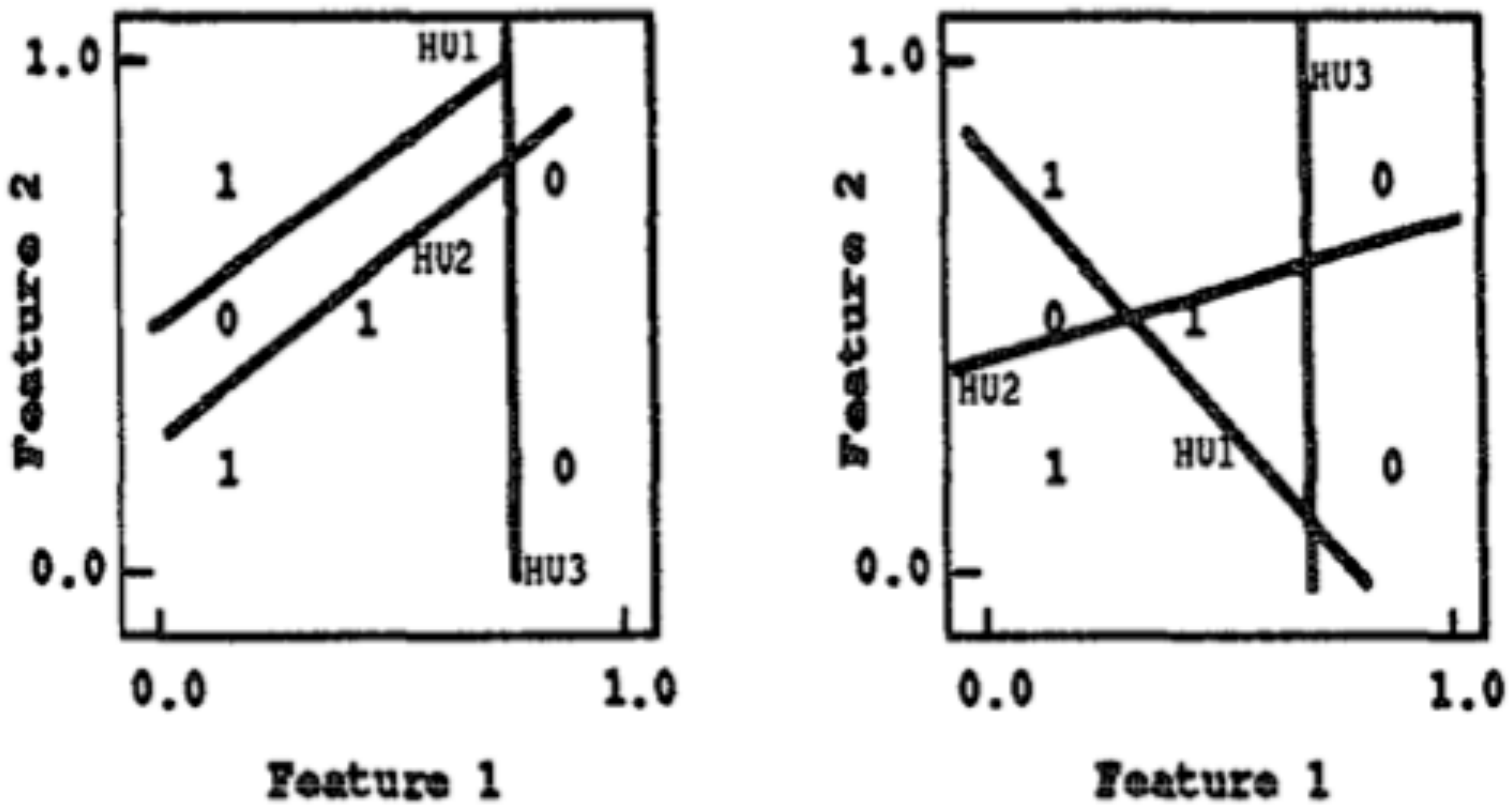
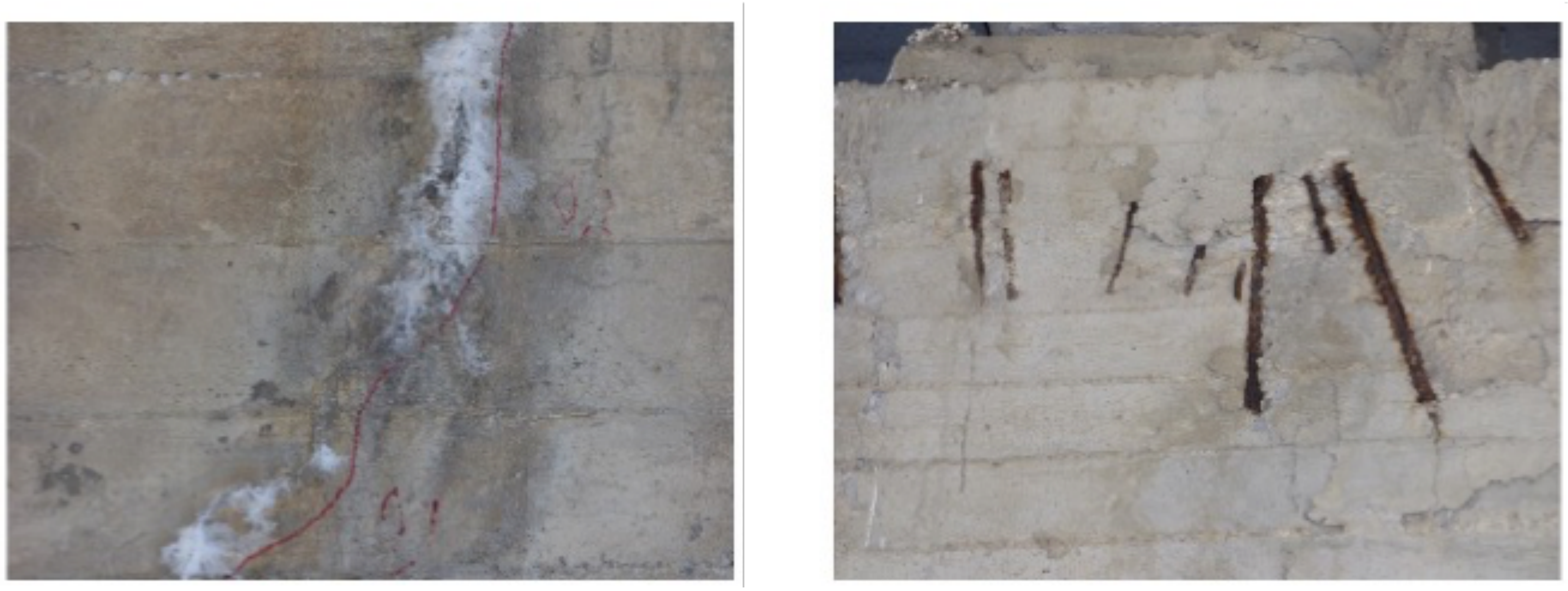


Figure 2: Two examples of hyperplane sets that separate training data in a small network.

"Direct Transfer of Learned Information Among Neural Networks" , L. Y. Pratt et al, AAAI 1991

Not intuitive if transfer works



“Meta-learning Convolutional Neural Architectures for Multi-target Concrete Defect Classification with the Concrete Defect Bridge Image Dataset”, Mundt et al, CVPR 2019

Brick Carpet Ceramic Fabric Foliage

“Material Recognition in the Wild with the Materials in Context Database, CVPR 2015”

<—Training from scratch:

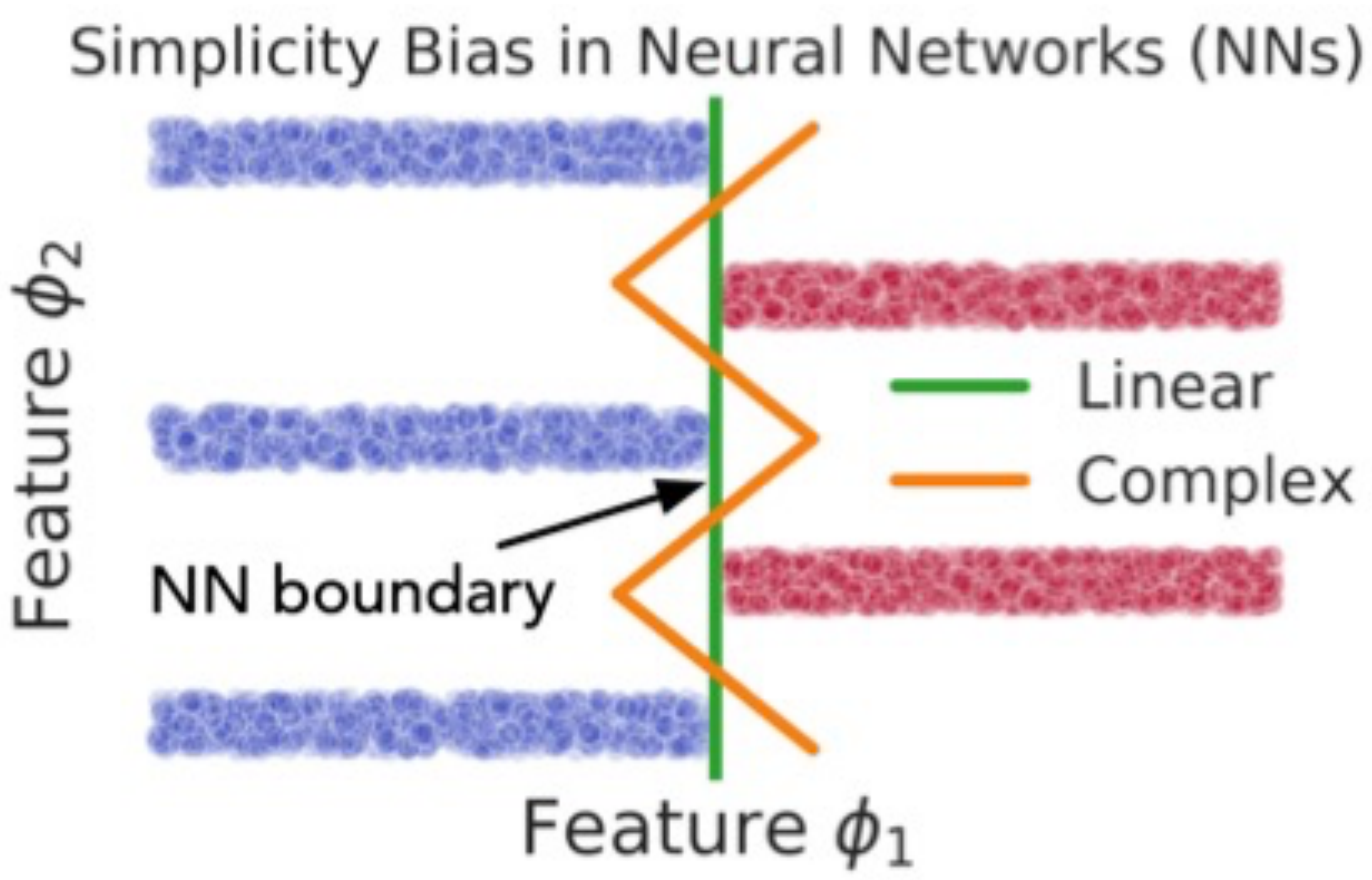
- Alexnet: 66.98 %
- VGG-A: 70.45%
- VGG-D: 70.61%

Transfer learning

Architecture	Source	Accuracy [%]
Alexnet	ImageNet	62.87
VGG-A	ImageNet	66.35
VGG-D	ImageNet	65.56
Densenet-121	ImageNet	57.66
Alexnet	MINC	66.50
VGG-D	MINC	67.14

Simplicity bias

Representations are biased in ways that we don't anticipate: **simplicity**



"The Pitfalls of Simplicity Bias in Neural Networks", Shah et al, NeurIPS 2020

Representation Bias

Representations are biased in ways that we don't anticipate: **texture bias**

(a) Texture image

81.4%	**Indian elephant**
10.3%	indri
8.2%	black swan

(b) Content image

71.1%	**tabby cat**
17.3%	grey fox
3.3%	Siamese cat

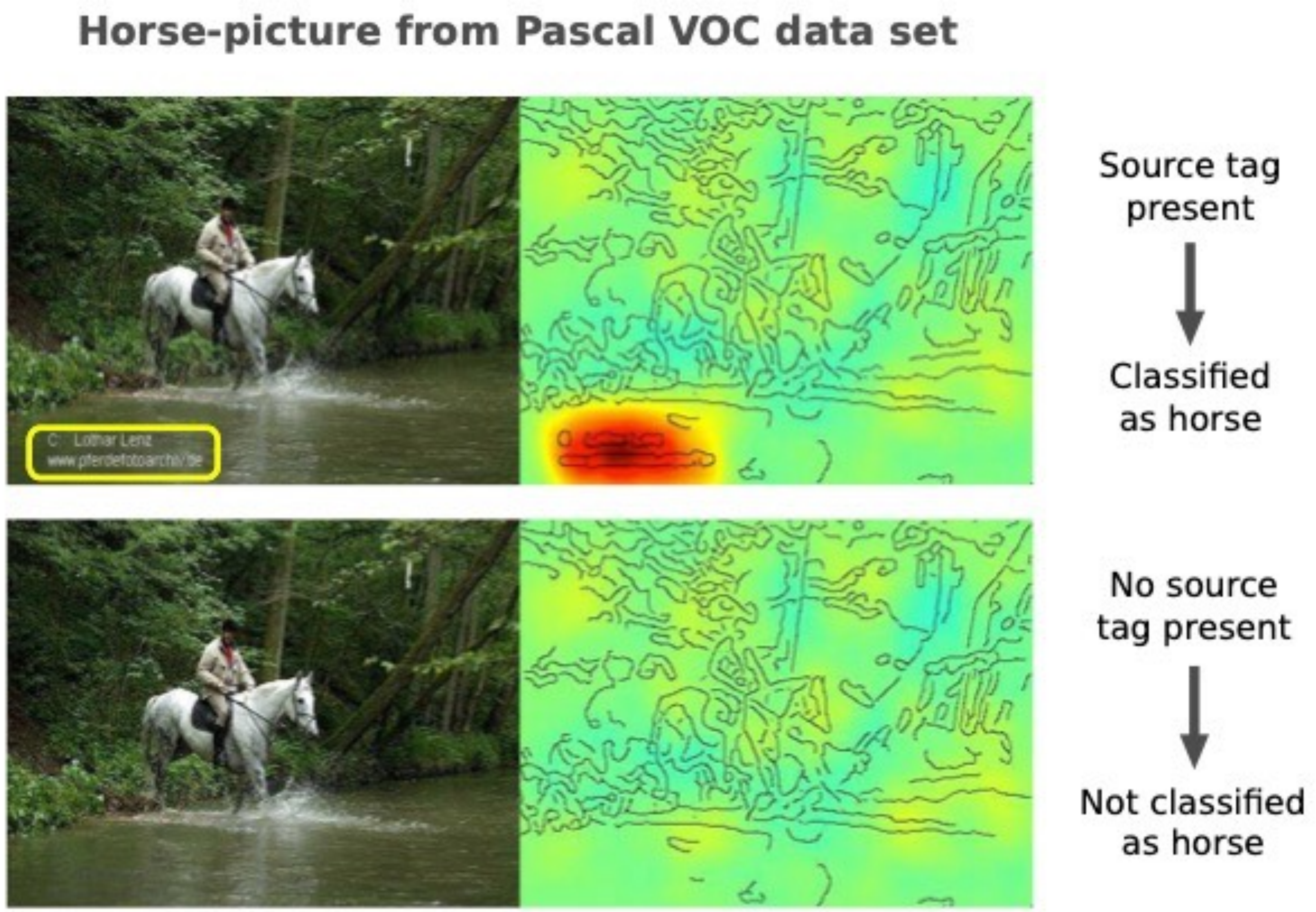
(c) Texture-shape cue conflict

63.9%	**Indian elephant**
26.4%	indri
9.6%	black swan

"ImageNet-trained CNNS are biased towards texture", Geirhos et al, ICLR 2019

Clever Hans predictors

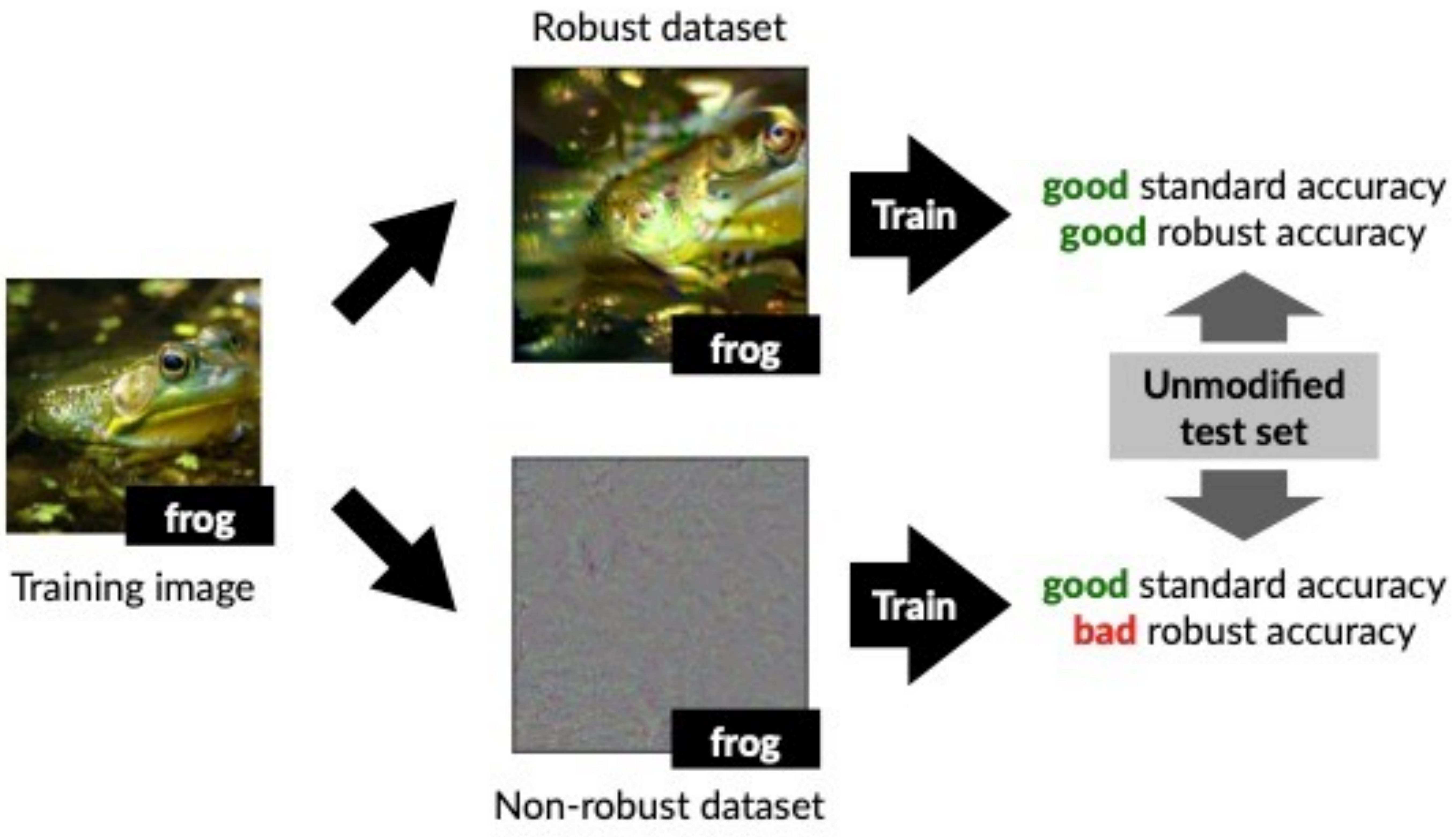
Representations are biased in ways that we don’t anticipate: **confounders**



“Unmasking Clever Hans Predictors”, Lapuschkin et al, Nature Communications 2019

Adversarial features

Representations are biased in ways that we don’t anticipate: **adversarial**



“Adversarial Examples are not Bugs, they are Features”, Ilyas et al, NeurIPS 2019

Back to the earlier definition.
It said "lifelong learning”! Not “transfer learning”

Early definition: lifelong ML

Definition - **Lifelong Machine Learning** - Thrun 1996:

“The system has performed N tasks. When faced with the (N+1)th task, it uses the knowledge gained from the N tasks to help the (N+1)th task.”

- We have looked primarily at (positive) forward transfer today
- Let us look at training & backward transfer (or forgetting) next

“Is Learning The n-th Thing Any Easier Than Learning the First?” (NeurIPS 1996) & “Explanation based Neural Network Learning A Lifelong Learning Approach”, Springer US, 1996

Later definition: lifelong ML

Definition - **Lifelong Machine Learning** - Chen & Liu 2017:

"Lifelong Machine Learning is a continuous learning process. At any time point, the learner performed a sequence of N learning tasks, $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_N$ .These tasks can be of the same type or different types and from the same domain or different domains. When faced with the (N+1)th task $\mathcal{T}_{N+1}$(which is called the new or current task) with its data D_{N+1}, the learner can leverage past knowledge in the knowledge base (KB) to help learn $\mathcal{T}_{N+1}$. The objective of LML is usually to optimize the performance on the new task $\mathcal{T}_{N+1}$, but it can optimize any task by treating the rest of the tasks as previous tasks. KB maintains the knowledge learned and accumulated from learning the previous task. After the completion of learning $\mathcal{T}_{N+1}$, KB is updated with the knowledge (e.g. intermediate as well as the final results) gained from learning $\mathcal{T}_{N+1}$. The updating can involve inconsistency checking, reasoning, and meta-mining of additional higher-level knowledge."

"Lifelong Machine Learning", Chen & Liu, Morgan Claypool, 2017