# Continual Machine Learning
## Summer 2023

**Teacher**

Dr. Martin Mundt,

hessian.AI-DEPTH junior research group leader on Open World Lifelong Learning (OWLL)
& researcher in the Artificial Intelligence and Machine Learning (AIML) group at TU Darmstadt

**Time**

Every Friday 14:25 - 16:05 CEST

**Course Homepage**

http://owll-lab.com/teaching/cl_lecture_23

https://www.youtube.com/playlist?list=PLm6QXeaB-XkA5-lVBB-h7XeYzFzgSh6sk

OWLL ContinualAI hessian.AI TECHNISCHE UNIVERSITÄT DARMSTADT

# Week 9: Ordering, Curricula & Difficulty

# Recall: from closed world …

**What if we don't know the boundary & aren't constrained on our testing examples?**

What if future or unrelated data is in the test set?

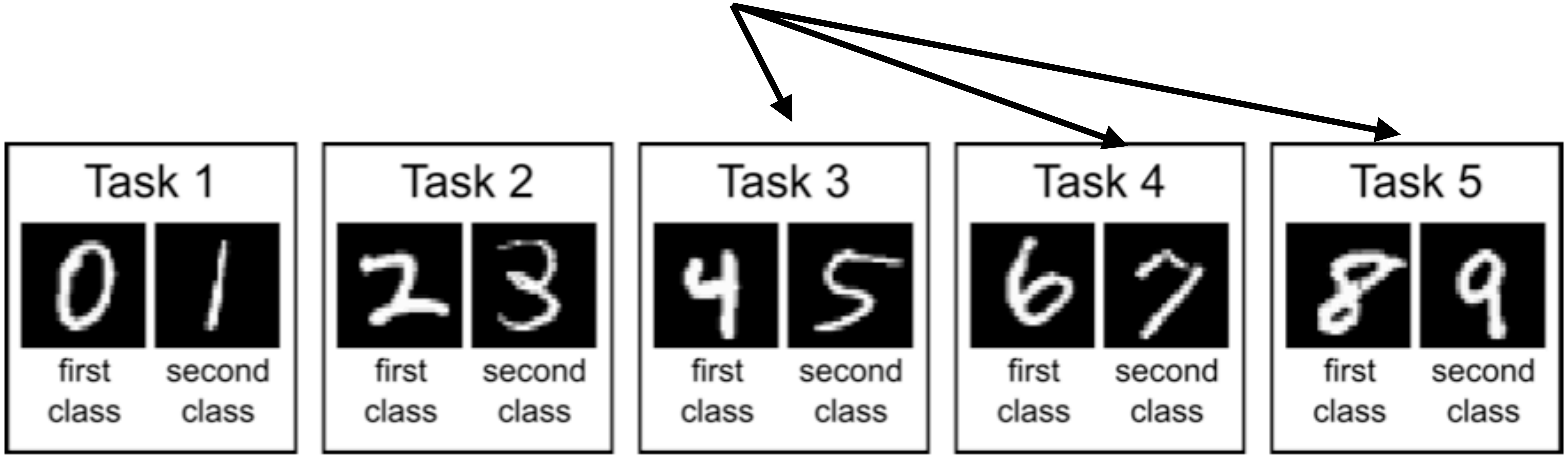



Figure 1: Schematic of split MNIST task protocol.

van de Ven & Tolias, "Three scenarios for continual learning", arXiv:1904.07734, 2019

# Recall: … to open world

**In retrospect: although there have been increments, the types of continual learning we have seen so far were indeed in a closed world**

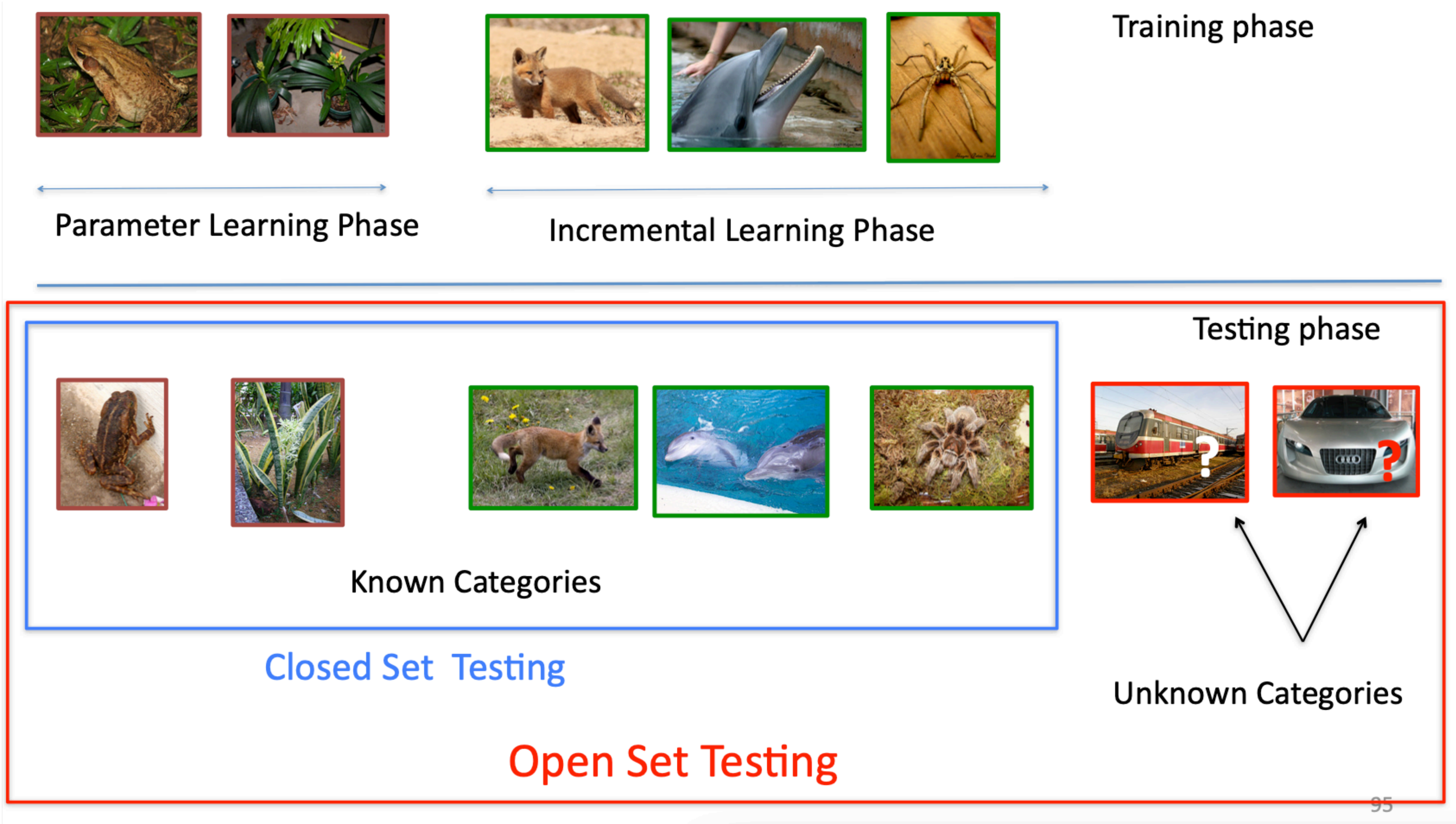
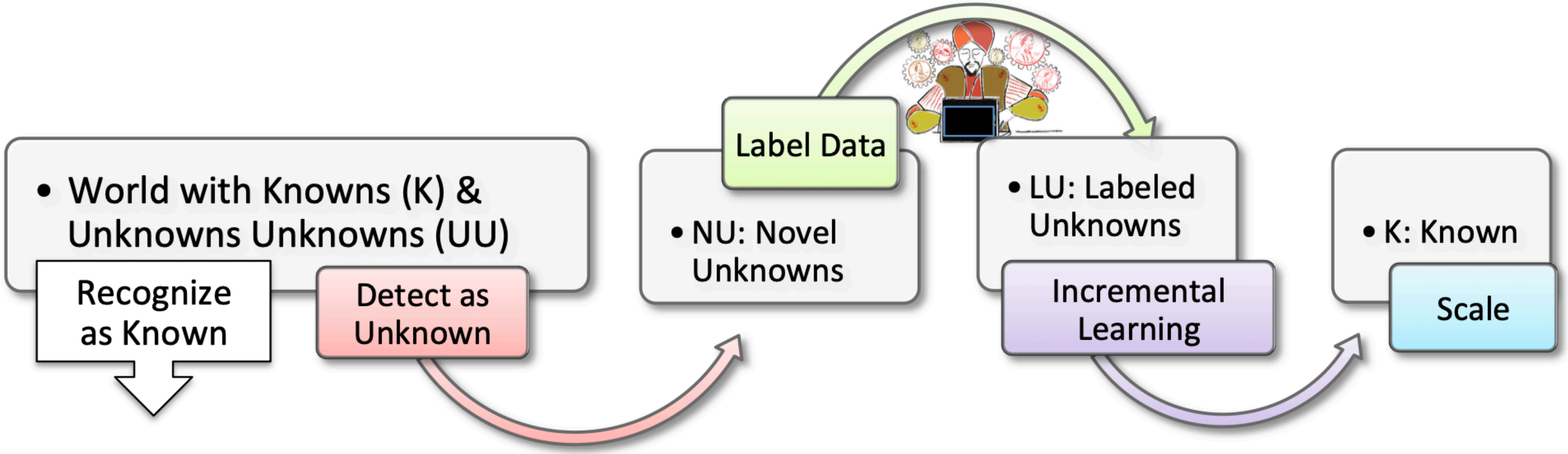



Figure from CVPR16 “Statistical Methods for Open Set Recognition” by Scheirer & Boult, https://www.wjscheirer.com/misc/openset/cvpr2016-open-set-part3.pdf

# Recall: open world learning

**Open world learning tries to "puzzle together" the pieces we have seen so far**

"An effective open world recognition system must efficiently perform four tasks: detect unknown, choose which points to label for addition to the model, label the points, and update the model" (Boult et al, "Learning and the Unknown", AAAI 2019)




Bendale & Boult ,"Towards Open World Recognition", CVPR 2015

# What about concept/task order?

**What about the order in which we learn? If we have the choice, which (identified unknown) data should we start with/include next?**

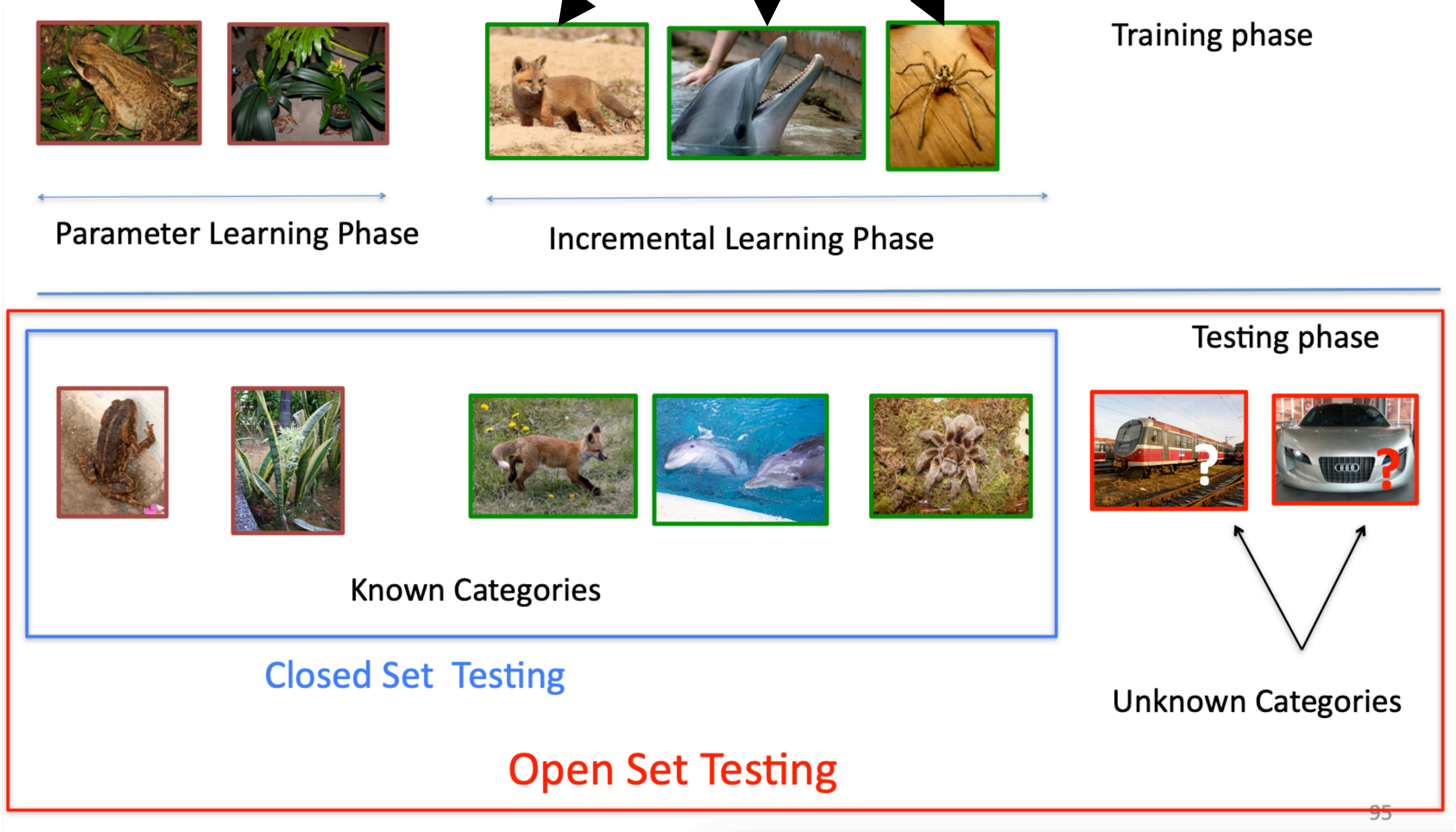



Figure from CVPR16 “Statistical Methods for Open Set Recognition” by Scheirer & Boult, https://www.wjscheirer.com/misc/openset/cvpr2016-open-set-part3.pdf

# Curriculum learning

# Curriculum Learning

## Curriculum learning: a definition

**Definition 1: Original Curriculum Learning [6].** A curriculum is a sequence of training criteria over $T$ training steps: $\mathcal{C} = \langle Q_1, \ldots, Q_t, \ldots, Q_T \rangle$. Each criterion $Q_t$ is a reweighting of the target training distribution $P(z)$:

$$Q_t(z) \propto W_t(z)P(z) \quad \forall \text{example } z \in \text{training set } D, \tag{1}$$

such that the following three conditions are satisfied:

1) The entropy of distributions gradually increases, i.e., $H(Q_t) < H(Q_{t+1})$.
2) The weight for any example increases, i.e., $W_t(z) \leq W_{t+1}(z) \quad \forall z \in D$.
3) $Q_T(z) = P(z)$.

From Wang et al, “A Survey on Curriculum Learning”, TPAMI 2021, based on original definition by Bengio et al, “Curriculum Learning”, ICML 2009

# Curriculum Learning

## Curriculum learning: the more intuitive definition (with a little bit of a tautology)

**Definition 3: Generalized Curriculum Learning.** Discarding the definition of $Q_t$ (Eq. 1) and its three conditions in Definition 1, a curriculum is a sequence of training criteria over $T$ training steps. Each criterion $Q_t$ includes the design for all the elements in training a machine learning model, e.g., data/tasks, model capacity, learning objective, etc. Curriculum learning is the strategy that trains a model with such a curriculum.

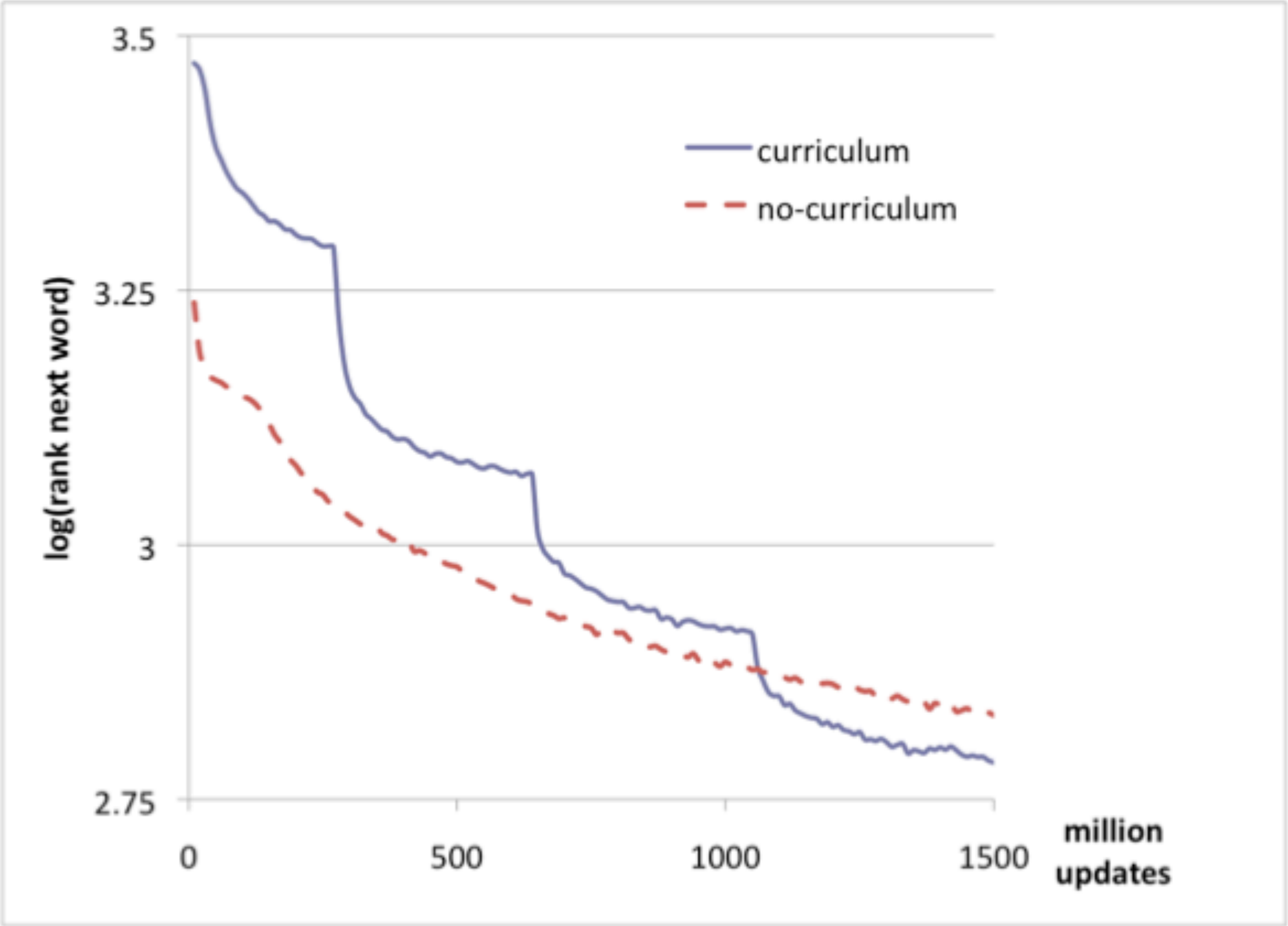
From Wang et al, "A Survey on Curriculum Learning", TPAMI 2021, based on original definition by Bengio et al, "Curriculum Learning", ICML 2009

# Recall L1: a motivating example

Example: Ranking language model trained with vs without curriculum on Wikipedia

"Error" is log of the rank of the next word (within 20k-word vocabulary).

1. The curriculum-trained model skips examples with words outside of 5k most frequent words
2. Then skips examples outside 10k most frequent words and so on




Bengio et al, "Curriculum Learning", ICML 2009

# What are central questions in curriculum learning?

# Curriculum learning

**Two key challenges**:

**Scoring function (difficulty measurer):**

Any function that provides us with an estimate of the difficulty of the instances in our dataset(s).

**Pacing function (training scheduler):**

(sometimes also called competence, as we'll see later)

The function that tells us how to interleave samples into the training process over time.

# Curriculum learning

**Two key challenges**:

**Scoring function (difficulty measurer):**

Any function that provides us with an estimate of the difficulty of the instances in our dataset(s).

**Pacing function (training scheduler):**

(sometimes also called competence, as we'll see later)

The function that tells us how to interleave samples into the training process over time.

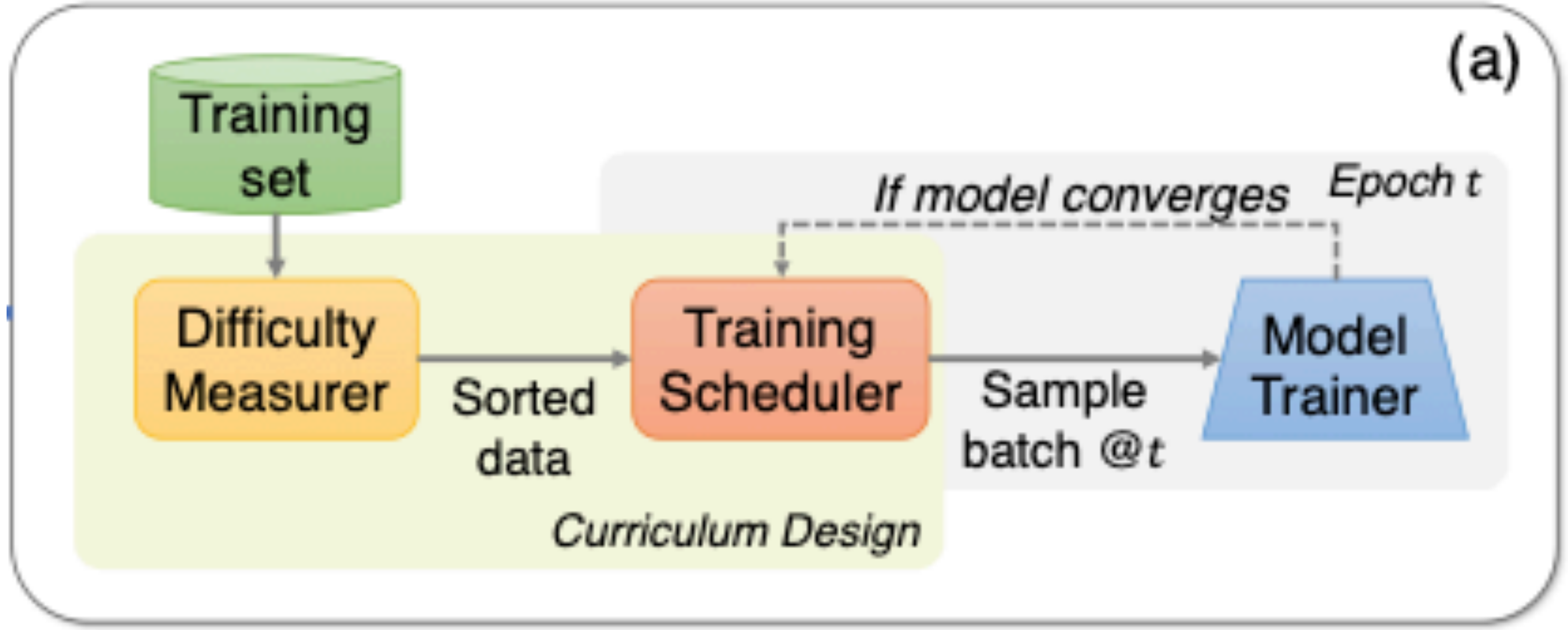
**Algorithm 1** Curriculum learning method

**Input:** *pacing function* $g_\vartheta$, *scoring function* $f$, data $\mathbb{X}$.
**Output:** sequence of mini-batches $\left[\mathbb{B}'_1, ..., \mathbb{B}'_M\right]$.
sort $\mathbb{X}$ according to $f$, in ascending order
$result \leftarrow []$
**for all** $i = 1, ..., M$ **do**
  $size \leftarrow g_\vartheta(i)$
  $\mathbb{X}'_i \leftarrow \mathbb{X}[1, ..., size]$
  uniformly sample $\mathbb{B}'_i$ from $\mathbb{X}'$
  append $\mathbb{B}'_i$ to $result$
**end for**
**return** $result$

Algorithm from Hacohen & Weinshall, "On the power of curriculum learning in deep networks", ICML 2019

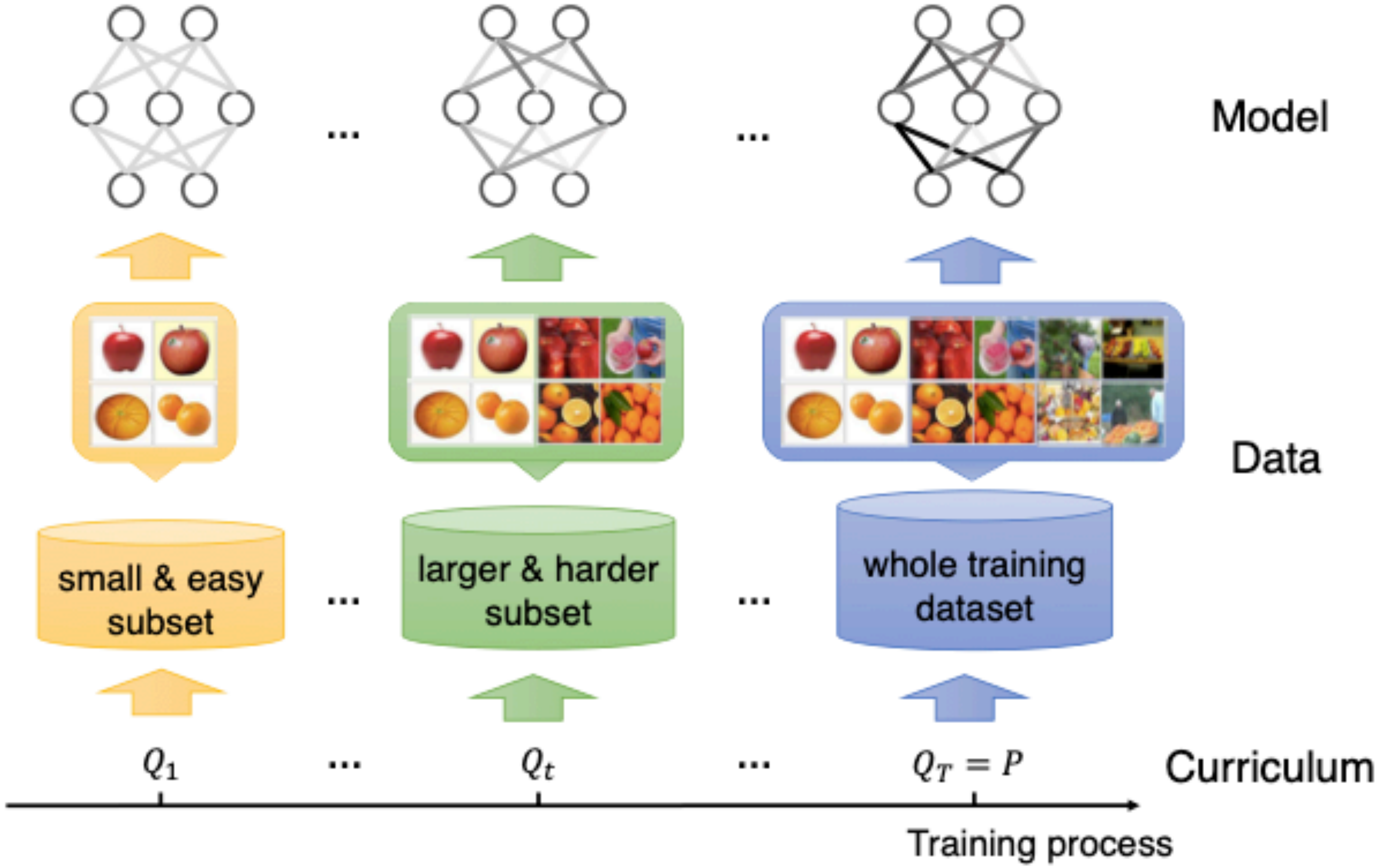
# Curriculum learning

**Let's start by considering a pre-defined curriculum, inspired by learning from "textbook style" content**




Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

# Defining difficulty




If we want to define the curriculum up-front, according to prior knowledge, then:

what is an “**easy**” & what is a “**harder**” subset/dataset?

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

**Can you think of ways to define “difficulty”?**

# How to define difficulty

TABLE 2
Common types of predefined Difficulty Measurer. The "+" in ∝Easy means the higher the measured value, the easier the data example, and the "-" has the opposite meaning.

| Difficulty Measurer* | Angle | Data Type | ∝Easy |
|---|---|---|---|
| Sentence length [86], [107] | Complexity | Text | - |
| Number of objects [122] | Complexity | Images | - |
| # conj. [50], #phrases [113] | Complexity | Text | - |
| Parse tree depth [113] | Complexity | Text | - |
| Nesting of operations [131] | Complexity | Programs | - |
| Shape variability [6] | Diversity | Images | - |
| Word rarity [50], [86] | Diversity | Text | - |
| POS entropy [113] | Diversity | Text | - |
| Mahalanobis distance [14] | Diversity | Tabular | - |
| Cluster density [11], [31] | Noise | Images | + |
| Data source [10] | Noise | Images | / |
| SNR / SND [7], [89] | Noise | Audio | - |
| Grammaticality [66] | Domain | Text | + |
| Prototypicality [113] | Domain | Text | + |
| Medical based [44] | Domain | X-ray film | / |
| Retrieval based [18], [82] | Domain | Retrieval | / |
| Intensity [30] / Severity [111] | Intensity | Images | + |
| Image difficulty score [106], [114] | Annotation | Images | - |
| Norm of word vector [68] | Multiple | Text | - |

Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

# Is difficulty task & model specific?

# How to define difficulty

**We have already seen that specific tasks allow for specific definitions of difficulty**

Example: natural language translation (sentence length)

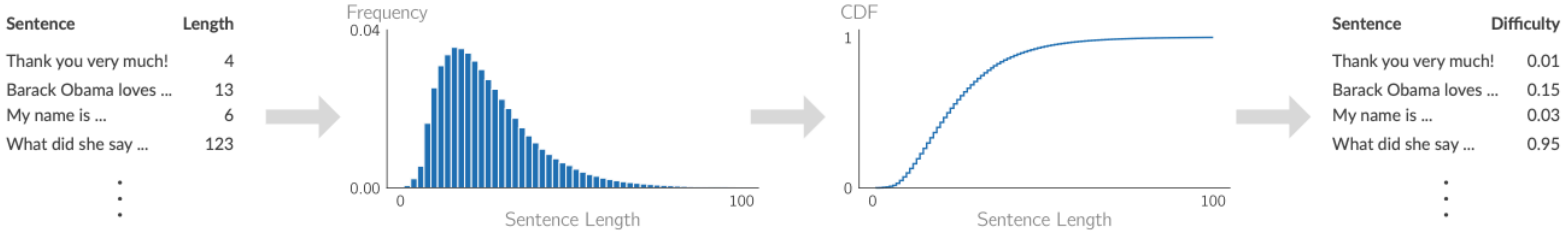



Figure 2: Example visualization of the preprocessing sequence used in the proposed algorithm. The histogram shown is that of sentence lengths from the WMT-16 En→De dataset used in our experiments. Here sentence lengths represent an example difficulty scoring function, $d$. “CDF” stands for the empirical “cumulative density function” obtained from the histogram on the left plot.

Platanios et al, “Competence based curriculum learning for neural machine translation”, NAACL-HIT 2019

# What is difficulty for a task?

**We have already seen that specific tasks allow for specific definitions of difficulty**

Example: image segmentation (entropy/clutter)

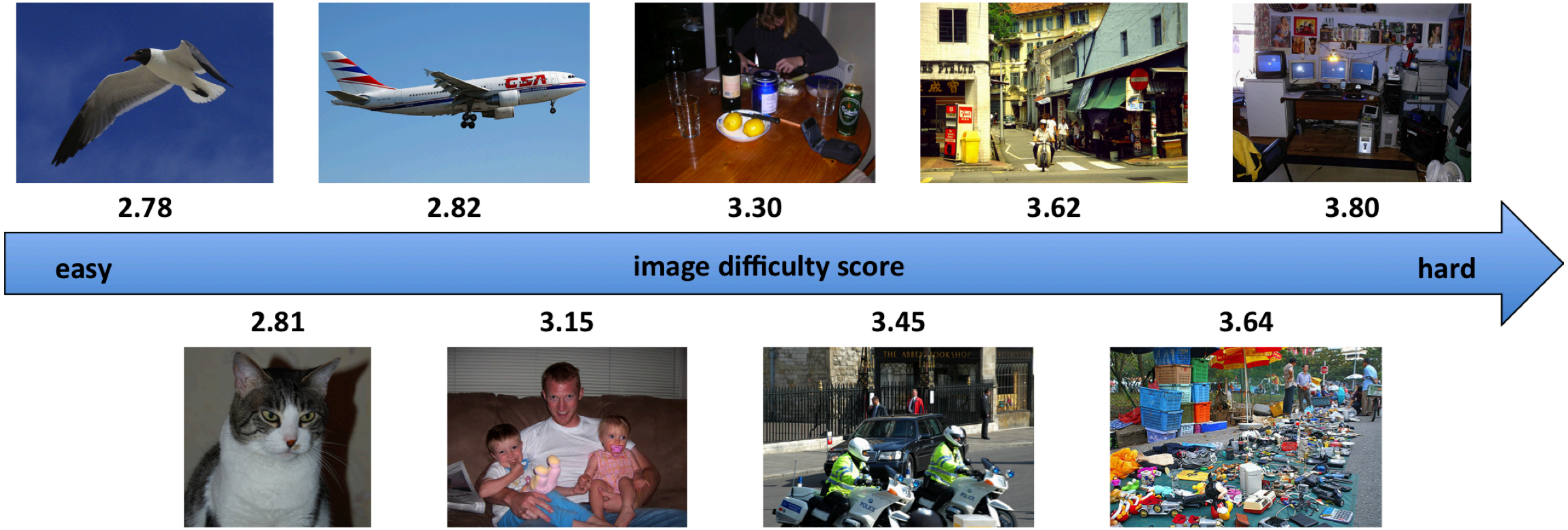



Figure 1. Images with difficulty scores predicted by our system in increasing order of their difficulty.

Ionescu et al, “How hard can it be? Estimating the difficulty of visual search in an image”, CVPR 2016

# What is difficulty for a task?

**There are various dimensions to difficulty, not just (basic) data statistics.**

Especially if we think about factors that relate to what humans may find difficult

## Compositional factors:

Size

"A sail boat on the ocean."

Location

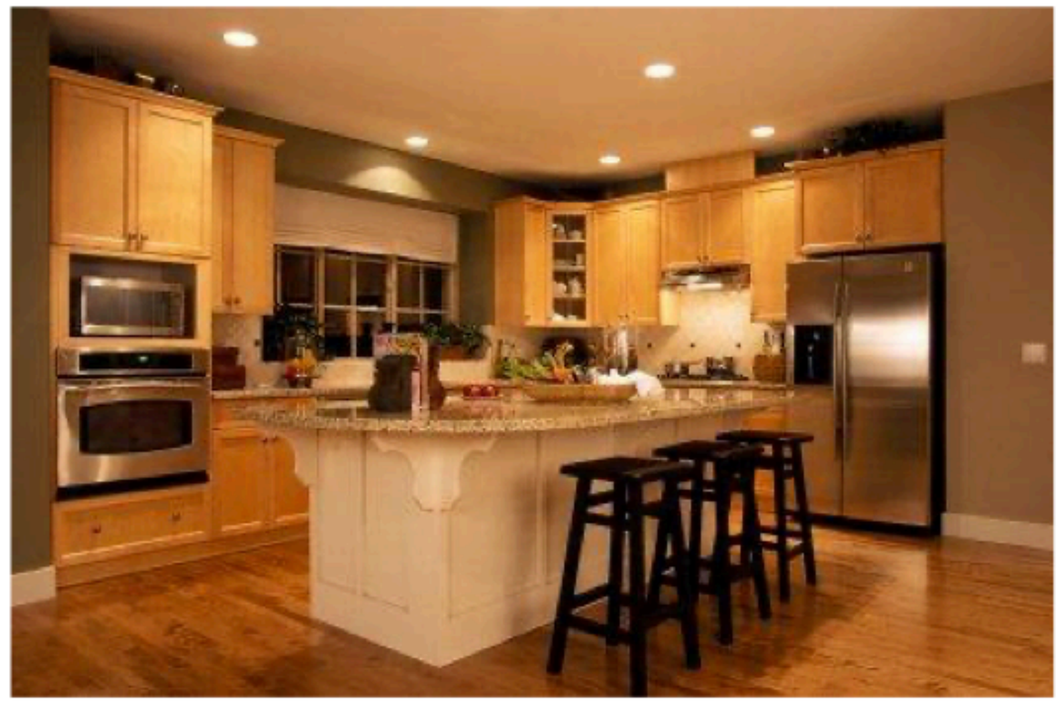
"Two men standing on beach."

## Semantic factors:

Object Type

"Girl in the street"

Scene Type & Depiction Strength



"kitchen in house"

## Context factors:

Unusual object-scene Pair

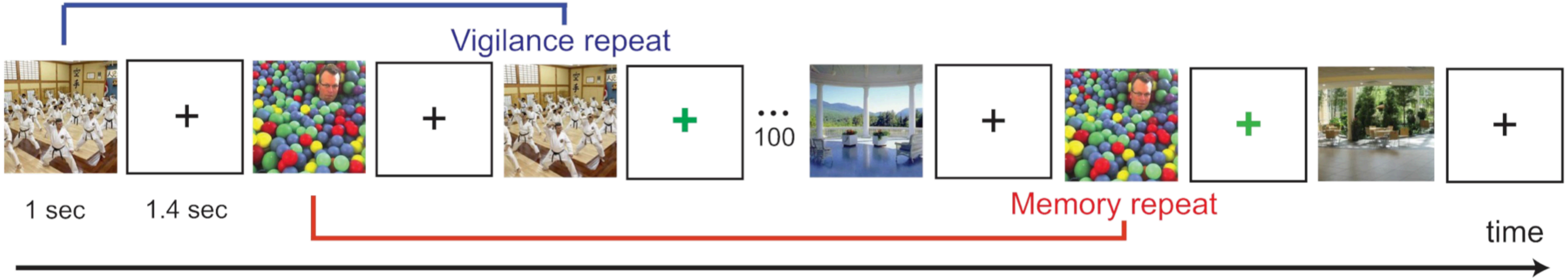
"A tree in water and a boy with a beard"

Berg et al, "Understanding and predicting importance in images", CVPR 2012

# What is difficult for ML models?

## But what is difficult for ML models & is this related to human perception?

Example: human memorability & image statistics




| | Object Counts | Object Areas | Multiscale Object Areas | Object Label Presences | Labeled Object Counts | Labeled Object Areas | Labeled Multiscale Object Areas | Scene Category | Objects and Scenes | Other Humans |
|---|---|---|---|---|---|---|---|---|---|---|
| Top 20 | 68% | 67% | 73% | 84% | 82% | 84% | 84% | 81% | 85% | 86% |
| Top 100 | 68% | 68% | 73% | 79% | 79% | 82% | 82% | 78% | 82% | 84% |
| Bottom 100 | 67% | 64% | 64% | 57% | 57% | 56% | 56% | 57% | 55% | 47% |
| Bottom 20 | 67% | 63% | 65% | 55% | 54% | 53% | 52% | 55% | 53% | 40% |
| $\rho$ | 0.05 | 0.05 | 0.20 | 0.43 | 0.44 | 0.47 | 0.48 | 0.37 | 0.50 | 0.75 |

Table 1. Comparison of predicted versus measured memorabilities.

Isola et al, “What makes an image memorable”, CVPR 2011

# What is difficult for ML models?

**But what is difficult for ML models & is this related to human perception?**

Example: human response times

**Collecting response times.** We collected ground-truth difficulty annotations by human evaluators using the following protocol: (i) we ask each annotator a question of the type "Is there an {*object class*} in the next image?", where {*object class*} is one of the 20 classes included in the PASCAL VOC 2012; (ii) we show the image to the annotator; (iii) we record the time spent by the annotator to answer the question by "Yes" or "No". Finally, we use this response time to estimate the visual search difficulty.

Ionescu et al, "How hard can it be? Estimating the difficulty of visual search in an image", CVPR 2016

# What is difficult for ML models?

**But what is difficult for ML models & is this related to human perception?**

Example: human response times

**Collecting response times.** We collected ground-truth difficulty annotations by human evaluators using the following protocol: (i) we ask each annotator a question of the type "Is there an {*object class*} in the next image?", where {*object class*} is one of the 20 classes included in the PASCAL VOC 2012; (ii) we show the image to the annotator; (iii) we record the time spent by the annotator to answer the question by "Yes" or "No". Finally, we use this response time to estimate the visual search difficulty.

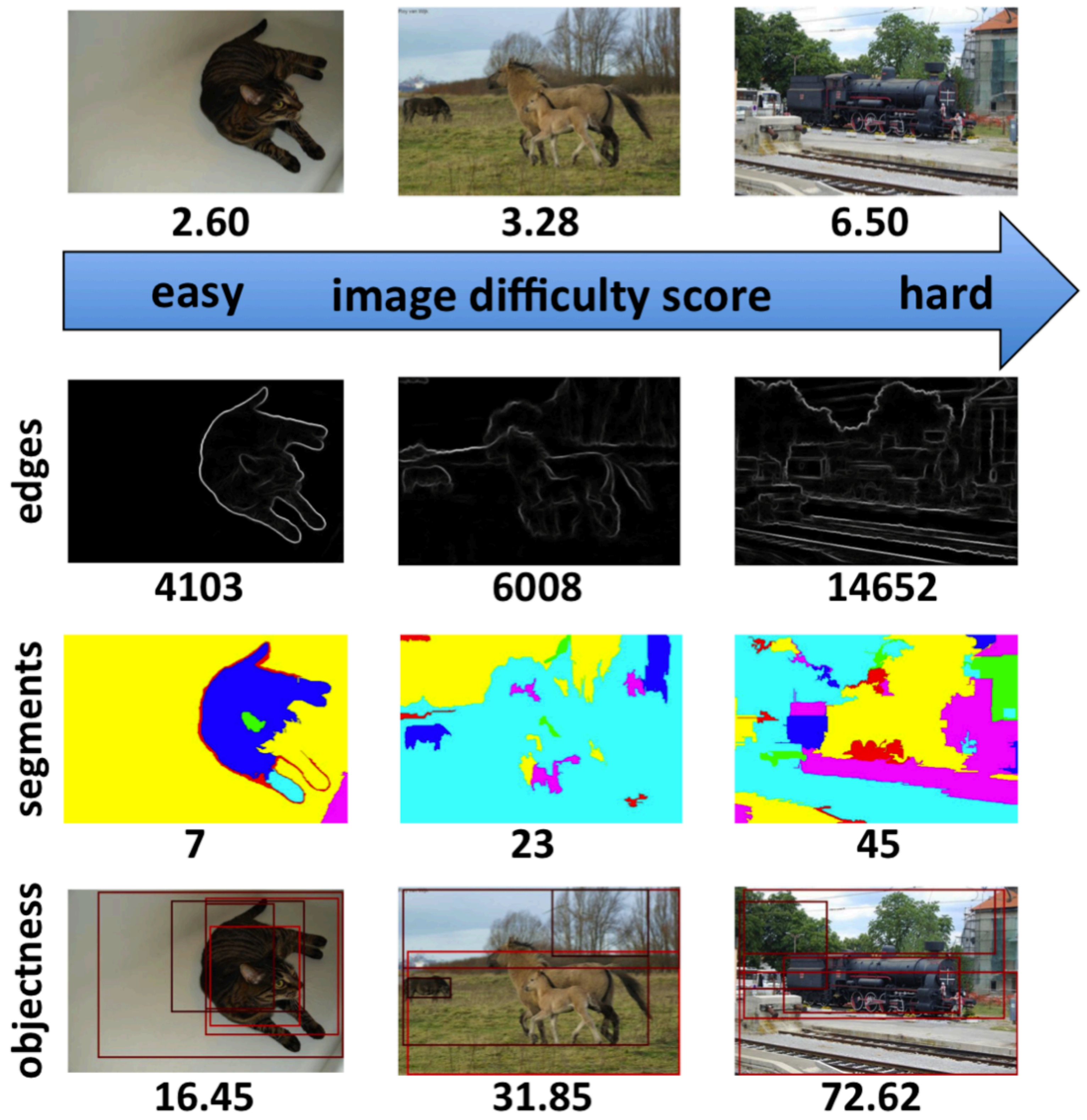
| | Image property | Kendall $\tau$ |
|---|---|---|
| (i) | number of objects | 0.32 |
| (ii) | mean area covered by objects | $-0.28$ |
| (iii) | non-centeredness | 0.29 |
| (iv) | number of different classes | 0.33 |
| (v) | number of truncated objects | 0.22 |
| (vi) | number of occluded objects | 0.26 |
| (vii) | number of difficult objects | 0.20 |

Ionescu et al, "How hard can it be? Estimating the difficulty of visual search in an image", CVPR 2016

# What is difficult for ML models?

**Human difficulty & model difficulty are not necessarily the same**

Various factors come into play in ML models: a regression example




| Model | MSE | Kendall $\tau$ |
|---|---|---|
| Random scores | 0.458 | 0.002 |
| Image area | - | 0.052 |
| Image file size | - | 0.106 |
| Objectness [1, 2] | - | 0.238 |
| Edge strengths [13] | - | 0.240 |
| Number of segments [16] | - | 0.271 |
| Combination with $\nu$-SVR | 0.264 | 0.299 |
| VGG-f + KRR | 0.259 | 0.345 |
| VGG-f + $\nu$-SVR | 0.236 | 0.440 |
| VGG-f + pyramid + $\nu$-SVR | 0.234 | 0.458 |
| VGG-f + pyramid + flip + $\nu$-SVR | 0.233 | 0.459 |
| VGG-vd + $\nu$-SVR | 0.235 | 0.442 |
| VGG-vd + pyramid + $\nu$-SVR | 0.232 | 0.467 |
| VGG-vd + pyramid + flip + $\nu$-SVR | 0.231 | 0.468 |
| VGG-f + VGG-vd + pyramid + flip + $\nu$-SVR | **0.231** | **0.472** |

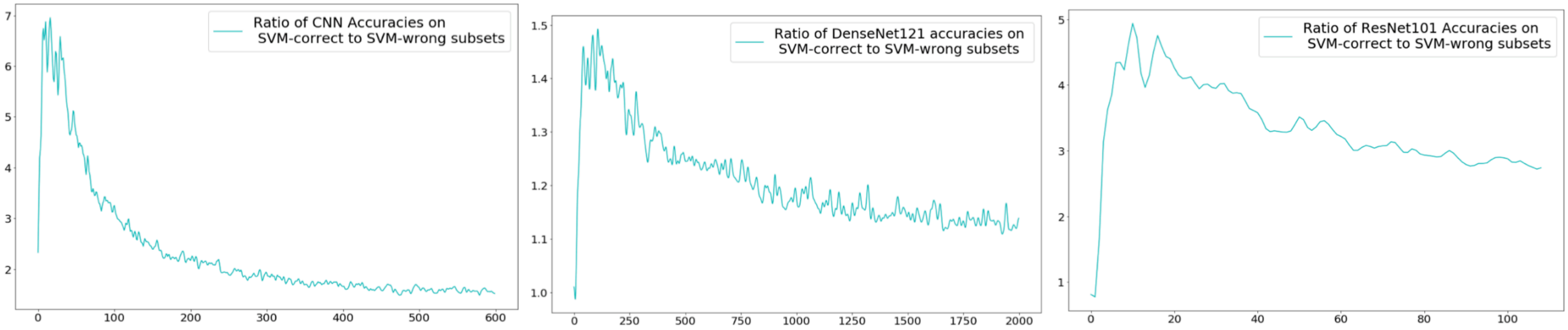
Ionescu et al, “How hard can it be? Estimating the difficulty of visual search in an image”, CVPR 2016

# What is difficult for ML models?

**Various factors come into play in ML models**

Example: shallow embeddable examples seem to be learned first

A deep network in comparison to a SVM (random forest also in the paper)



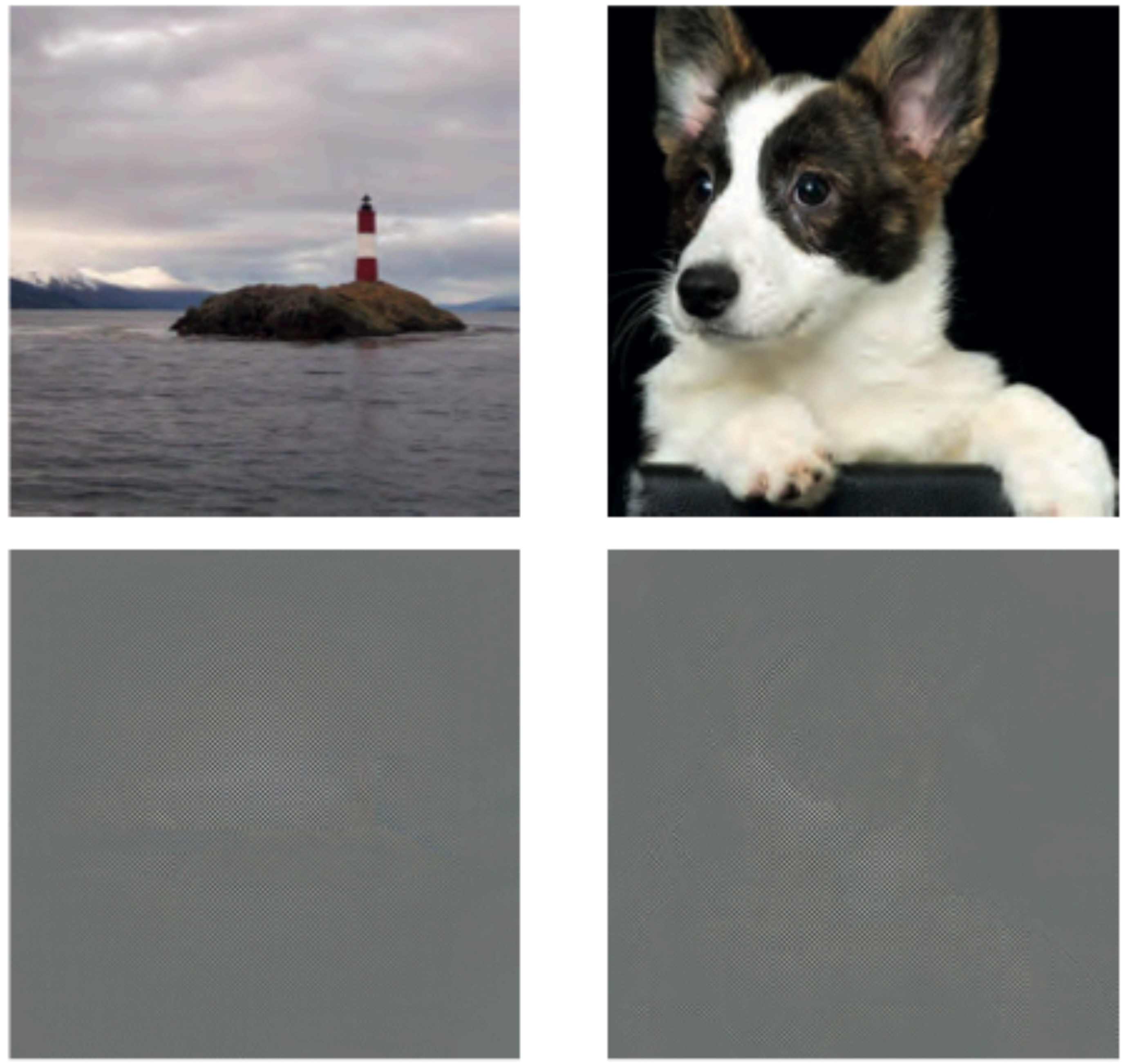

Ratio of Accuracies $\mathtt{R}^{\mathtt{i}}$ plotted against $i$ with $\mathcal{M}$ being a Support Vector Machine.

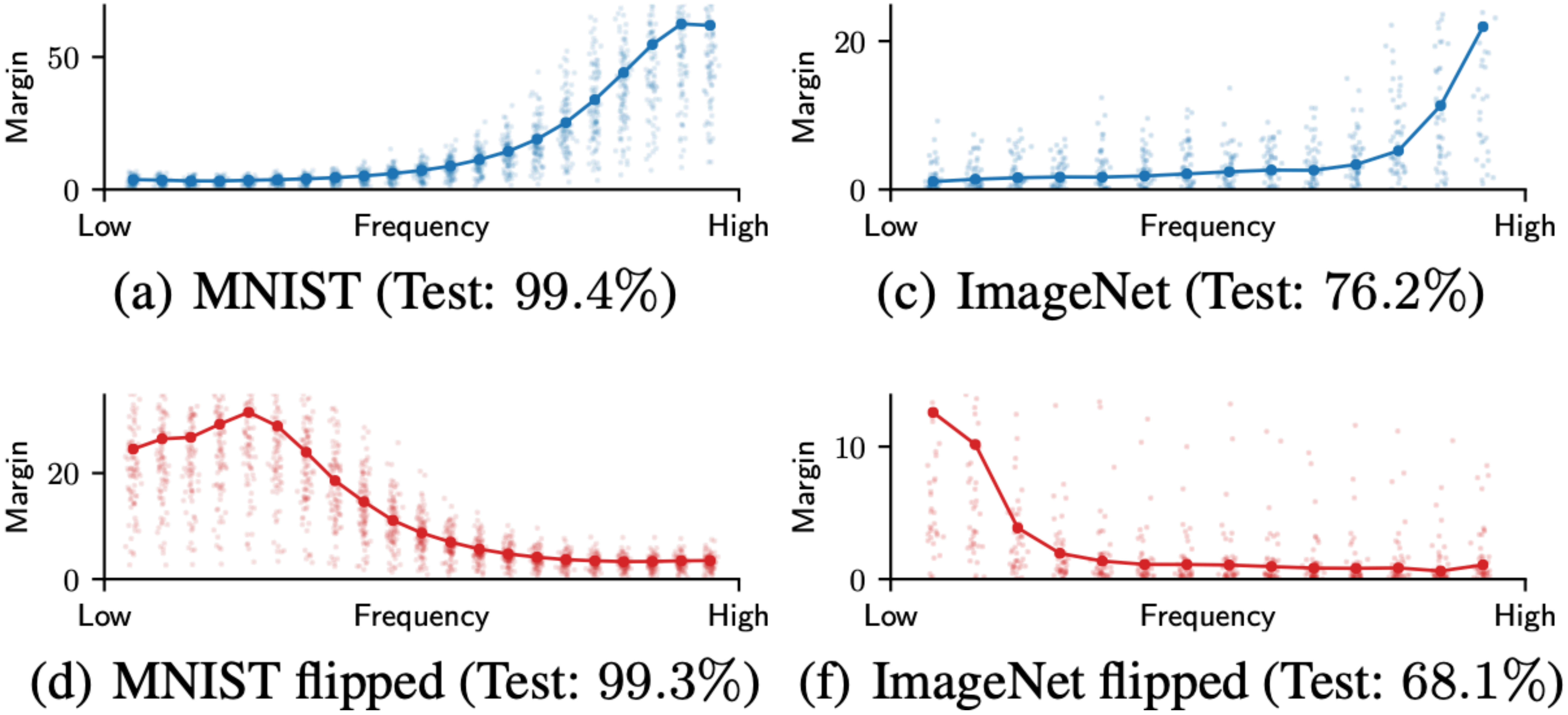
Mangalam & Prabhu, "Do deep neural networks learn shallow learnable examples first?", ICML 2019 workshop on identifying and understanding deep learning phenomena

# What is difficult for ML models?

**Various factors come into play in ML models**

Example: invariance to certain discriminative factors (e.g. frequencies) may exist



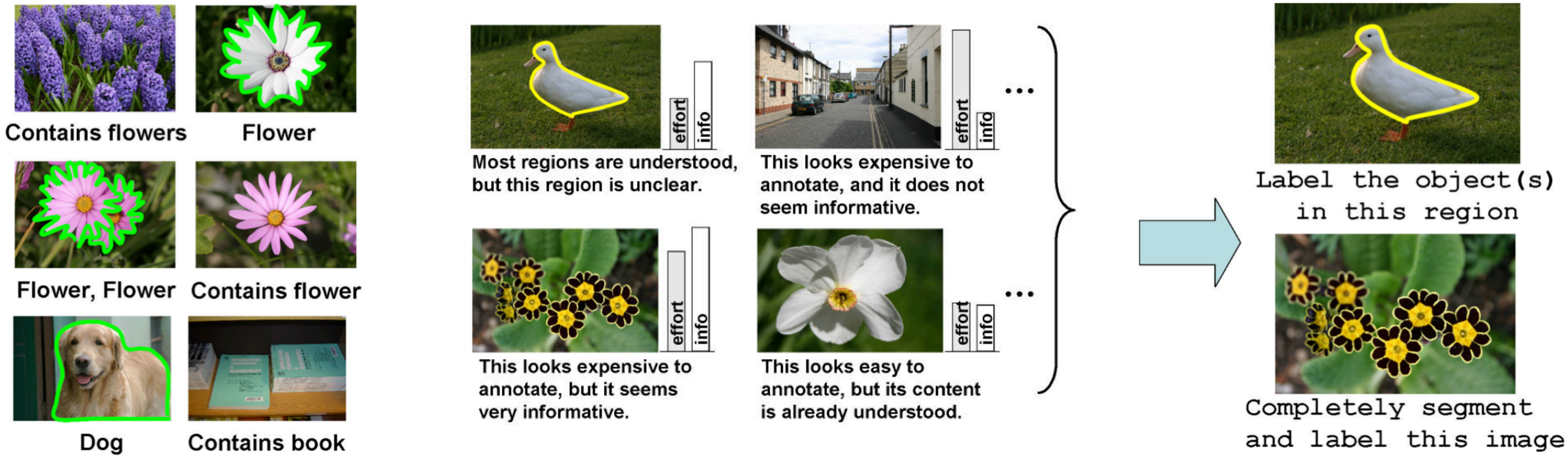



Ortiz-Jimenéz et al, “Hold me tight! Influence of discriminative features on deep network boundaries”, NeurIPS 2020

# Beyond curriculum learning

**Assessing difficulty of data instances is interesting beyond curriculum learning**

Example: estimating the difficulty with respect to annotation cost




(a) Labeled (and partially labeled) examples to build models

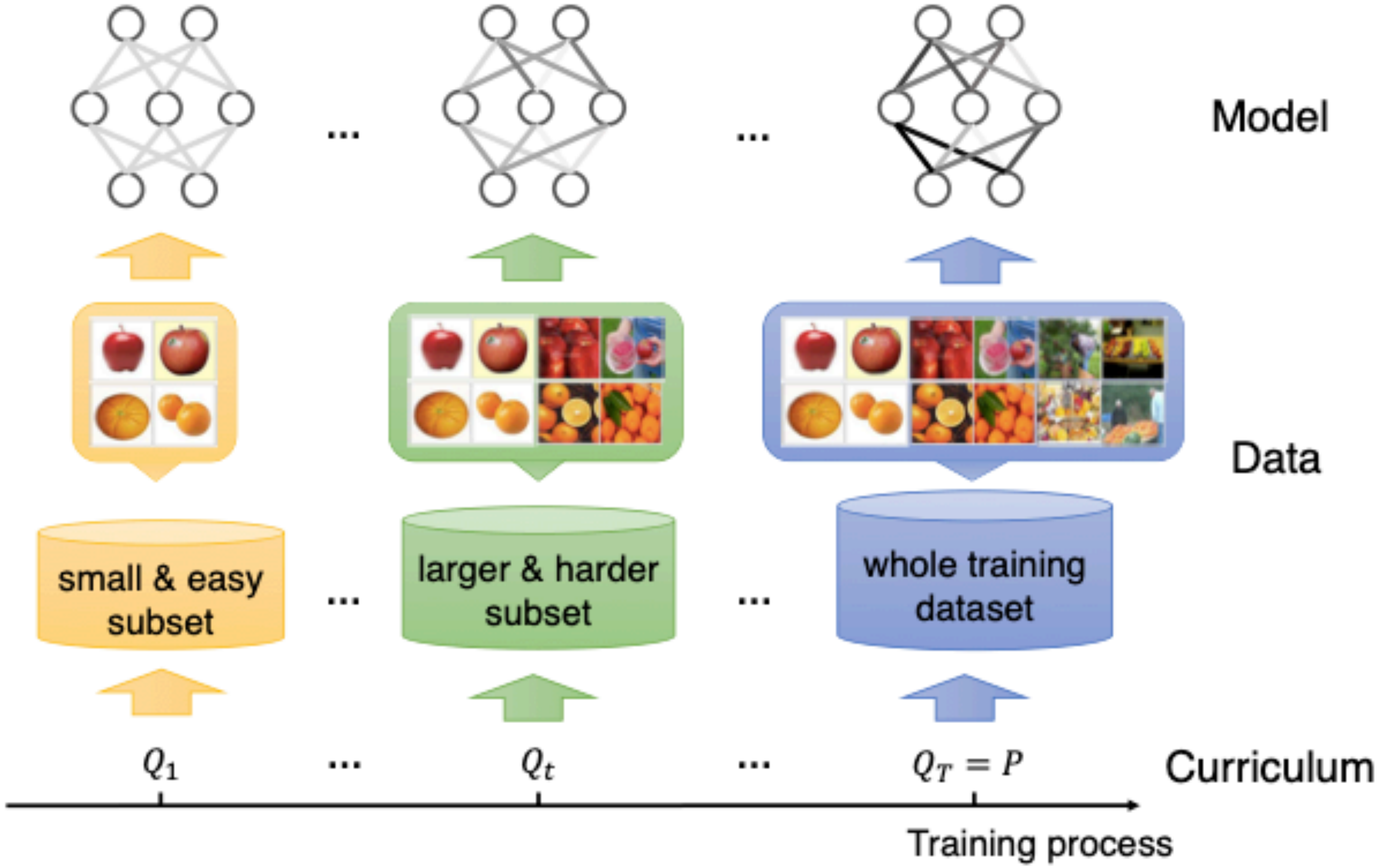
(b) Unlabeled and partially labeled examples to survey

(c) Actively chosen queries sent to annotators

Vijayanarasimhan & Grauman, "What's It Going to Cost You?: Predicting Effort vs. Informativeness for Multi-Label Image Annotations", CVPR 2009

# Pacing: how to schedule the training

# Scheduling training




Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

If we want to define the curriculum up-front, according to prior knowledge, then:

**When do we introduce** more difficult examples?

# Pacing functions

## Various options & heuristics are conceivable

**Algorithm 1** One-Pass Curriculum

**procedure** OP-CURRICULUM($M$,$\mathcal{D}$, $\mathcal{C}$)
    $\mathcal{D}' = \text{sort}(\mathcal{D}, \mathcal{C})$
    $\{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^k\} = \mathcal{D}'$ where $\mathcal{C}(d_a) < \mathcal{C}(d_b)$ $d_a \in D^i$ , $d_b \in D^j, \forall i < j$
    **for** $s = 1...k$ **do**
        **while** not converged for $p$ epochs **do**
            train($M$, $\mathcal{D}^s$)
        **end while**
    **end for**
**end procedure**

Algorithm from Cirik et al, “Visualizing and understanding curriculum learning for long short-term memory networks”, arXiv, 2016
Based on the procedure described in Bengio et al, “Curriculum Learning”, ICML 2009

# Pacing functions

## Various options & heuristics are conceivable

**Algorithm 1** One-Pass Curriculum

**procedure** OP-CURRICULUM($M$,$\mathcal{D}$, $\mathcal{C}$)
    $\mathcal{D}' = \text{sort}(\mathcal{D}, \mathcal{C})$
    $\{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^k\} = \mathcal{D}'$ where $\mathcal{C}(d_a) < \mathcal{C}(d_b)$ $d_a \in D^i$ , $d_b \in D^j, \forall i < j$
    **for** $s = 1...k$ **do**
        **while** not converged for $p$ epochs **do**
            $\text{train}(M, \mathcal{D}^s)$
        **end while**
    **end for**
**end procedure**

Algorithm from Cirik et al, “Visualizing and understanding curriculum learning for long short-term memory networks”, arXiv, 2016
Based on the procedure described in Bengio et al, “Curriculum Learning”, ICML 2009

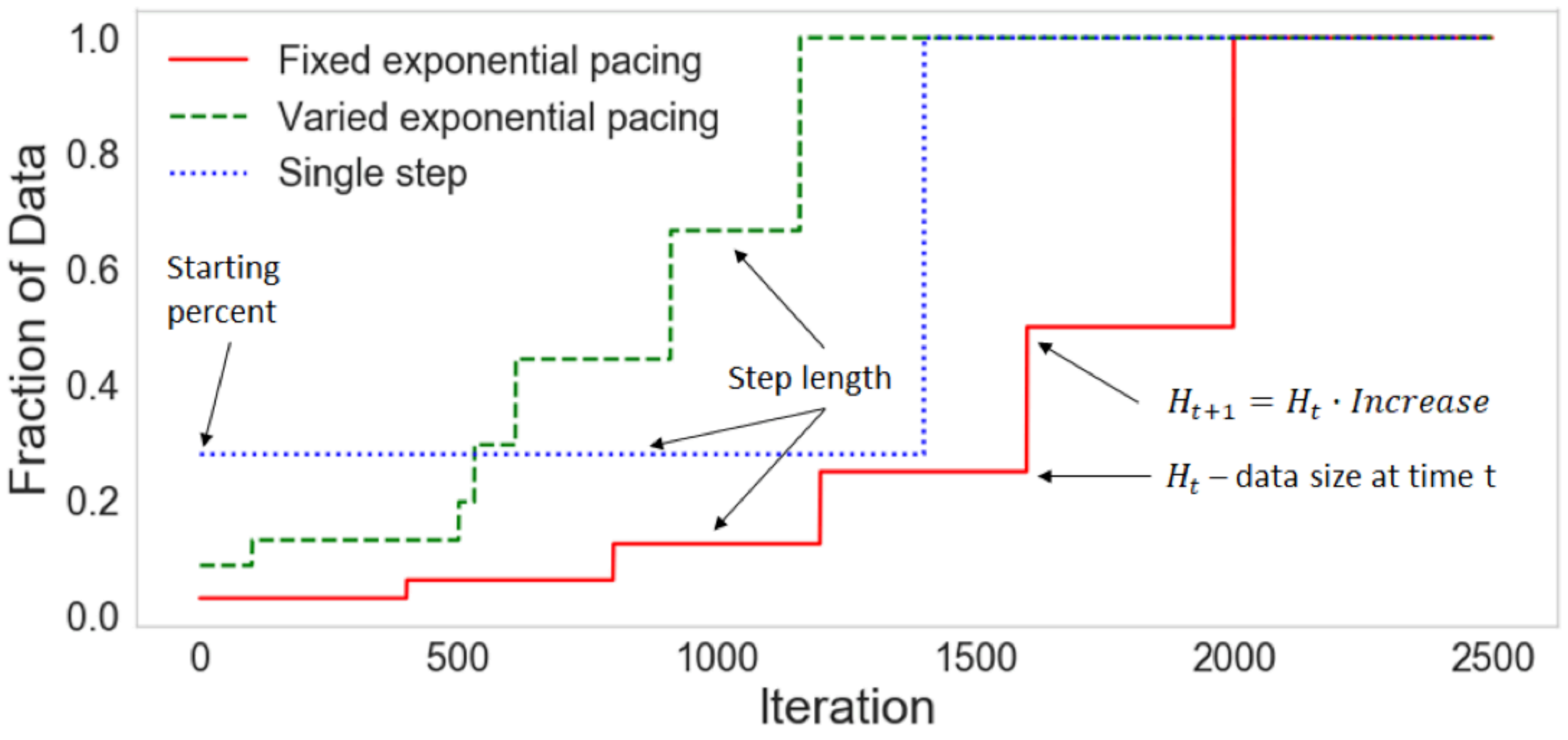
**Algorithm 2** Baby Steps Curriculum

**procedure** BS-CURRICULUM($M$,$\mathcal{D}$, $\mathcal{C}$)
    $\mathcal{D}' = \text{sort}(\mathcal{D}, \mathcal{C})$
    $\{\mathcal{D}^1, \mathcal{D}^2, ..., \mathcal{D}^k\} = \mathcal{D}'$ where $\mathcal{C}(d_a) < \mathcal{C}(d_b)$ $d_a \in D^i$ , $d_b \in D^j, \forall i < j$
    $\mathcal{D}^{train} = \emptyset$
    **for** $s = 1...k$ **do**
        $\mathcal{D}^{train} = \mathcal{D}^{train} \cup \mathcal{D}^s$
        **while** not converged for $p$ epochs **do**
            $\text{train}(M, \mathcal{D}^{train})$
        **end while**
    **end for**
**end procedure**

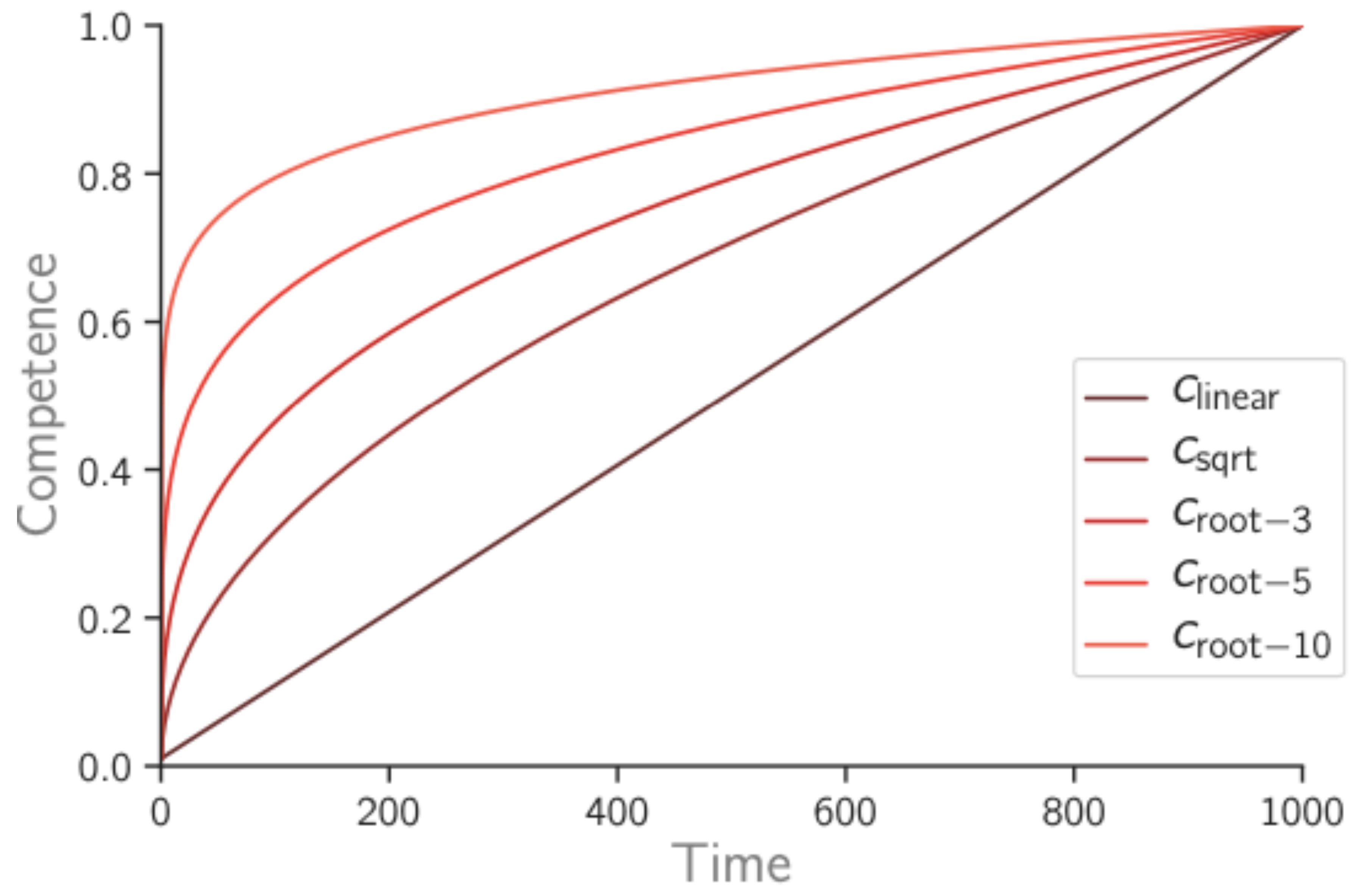
Algorithm from Cirik et al, “Visualizing and understanding curriculum learning for long short-term memory networks”, arXiv, 2016
Based on the procedure described in Spitkovsky et al, “From baby steps to leapfrogs: how less is more in unsupervised dependency parsing”, NAACL-HLT, 2010

# Pacing functions

## Various options & heuristics are conceivable



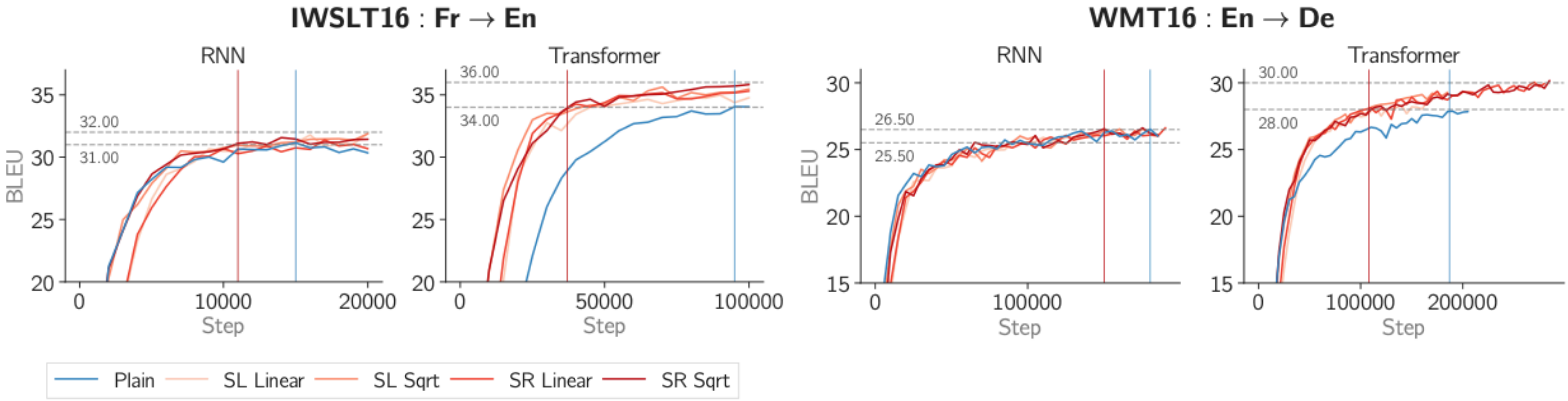

Hacohen & Weinshall, “On the power of curriculum learning in deep networks”, ICML 2019




Platanios et al, “Competence based curriculum learning for neural machine translation”, NAACL-HLT 2019

# Pacing functions

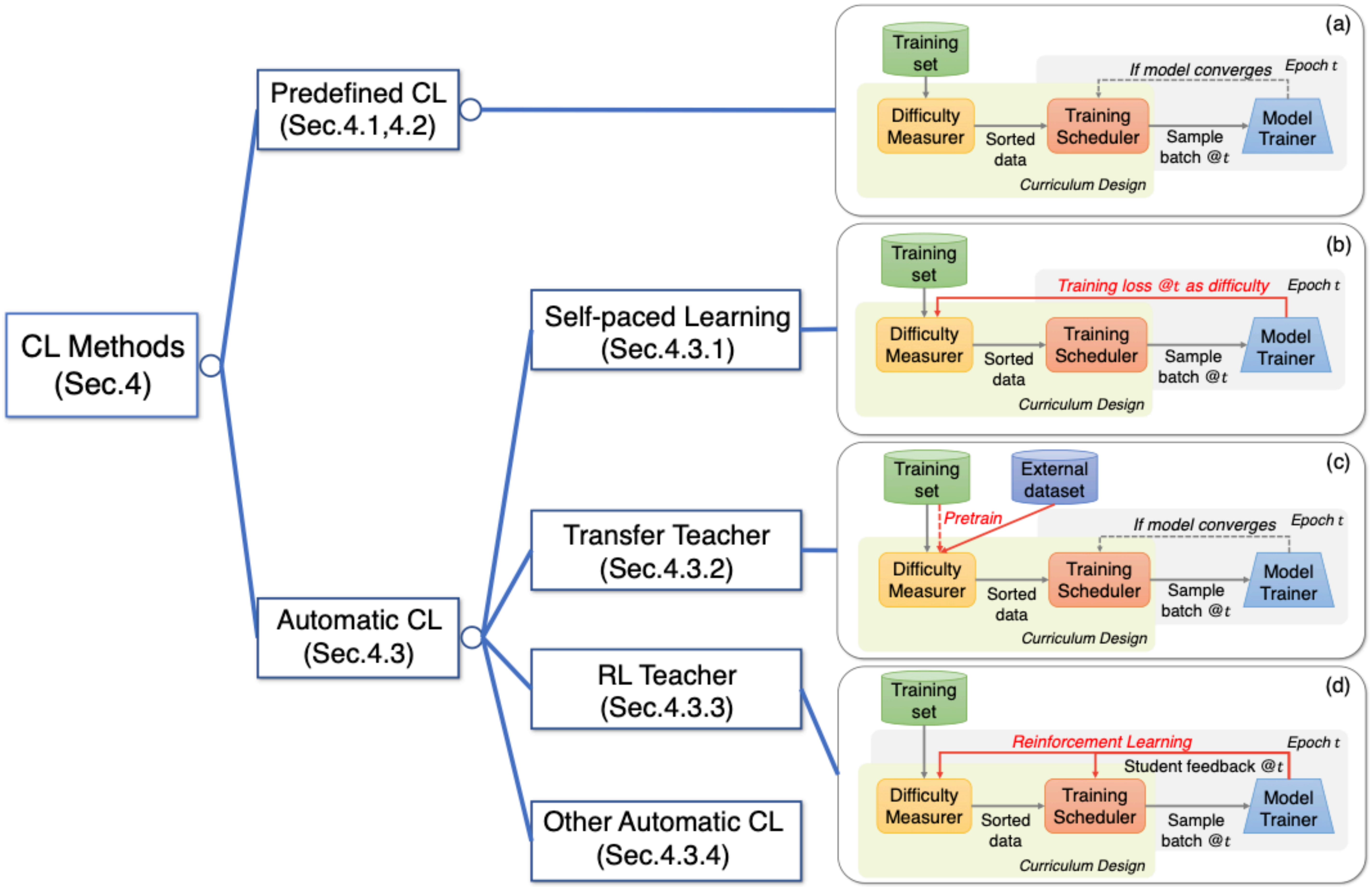
**It's not straightforward to choose, especially due to model/task dependency**




Platanios et al, "Competence based curriculum learning for neural machine translation", NAACL-HIT 2019

# Moving away from a pre-defined curriculum

# Beyond pre-defined curricula



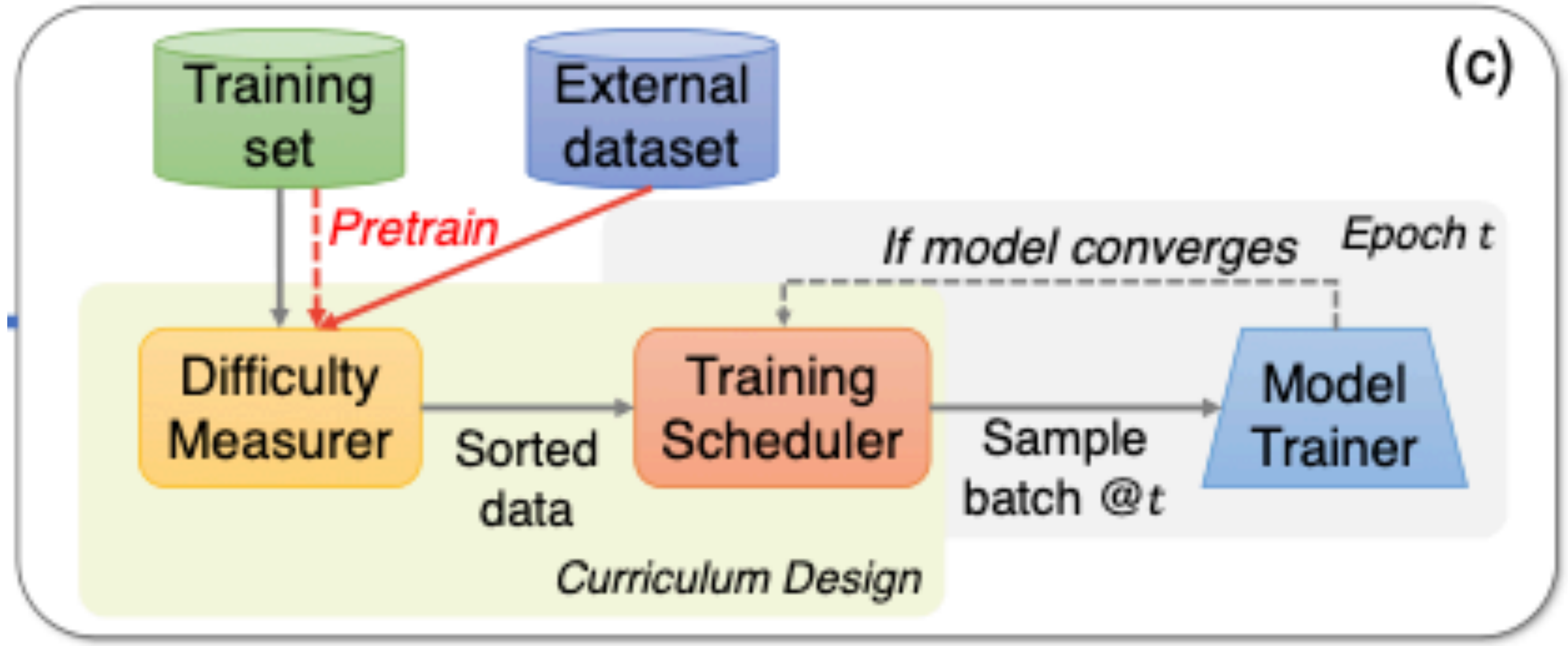

**What we just talked about**

**What we'll consider next**

Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

# Transfer-teacher curricula

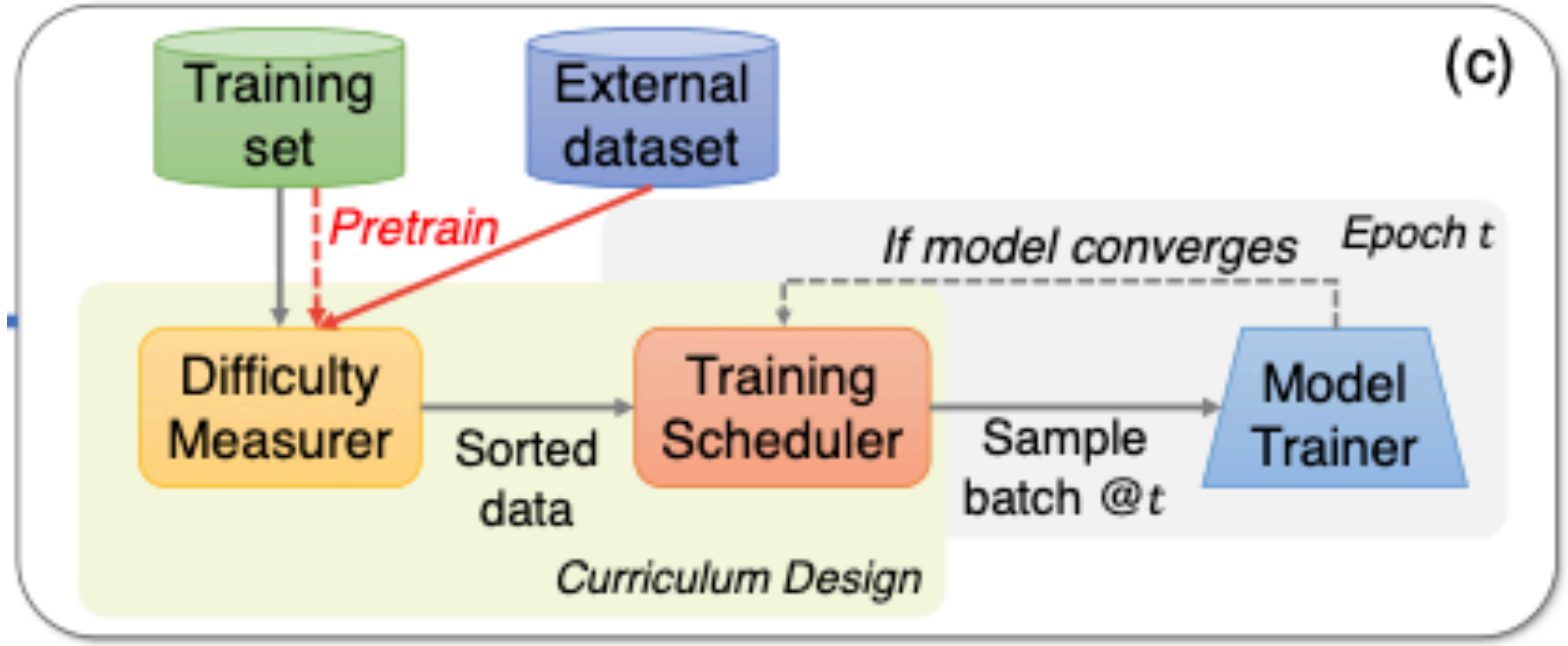
**Instead of defining the curriculum ourselves, we could use a pre-trained teacher model (based on a different related dataset) based difficulty measure**




Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

# Transfer-teacher curricula

**Instead of defining the curriculum ourselves, we could use a pre-trained teacher model (based on a different related dataset) based difficulty measure**



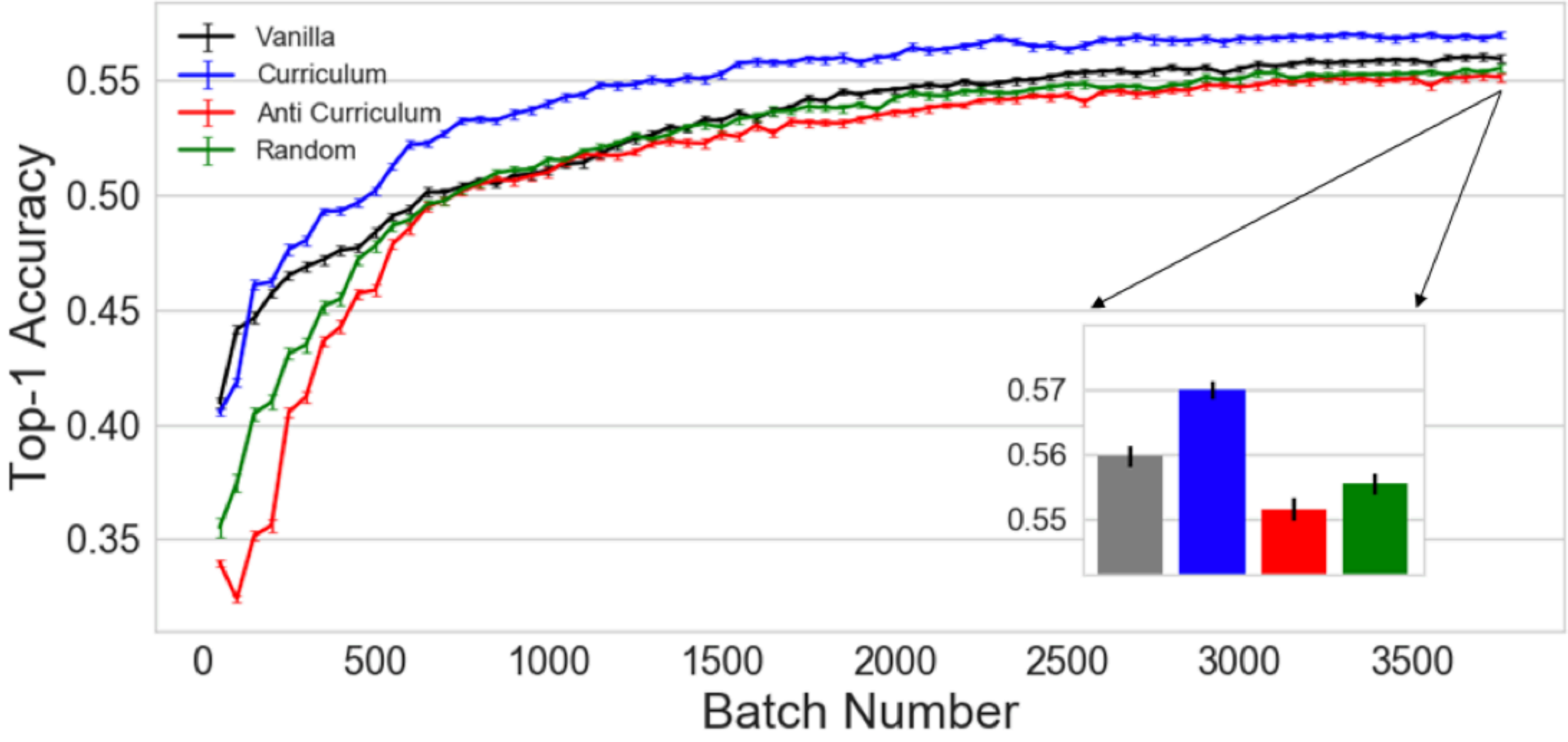




*Figure 2.* Results in **case 1**, with Inception-based *transfer scoring function* and *fixed exponential pacing function*.

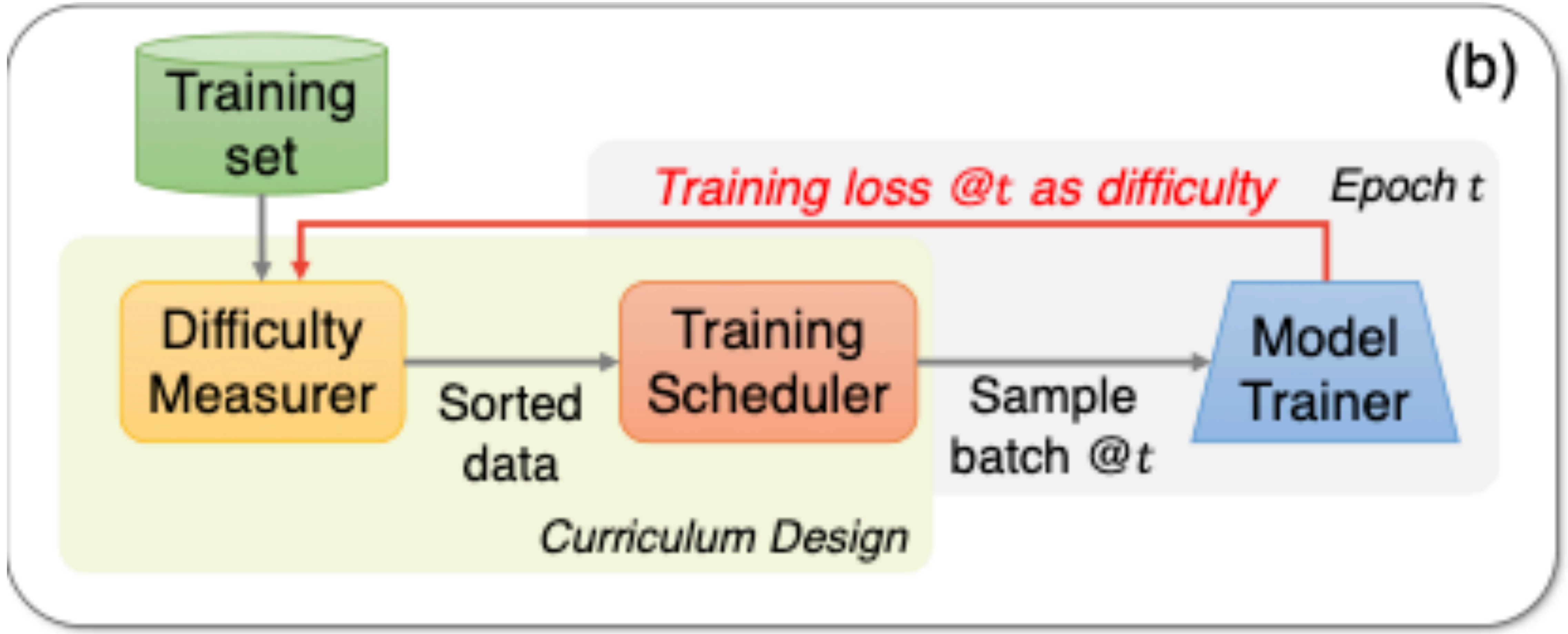
Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

Hacohen & Weinshall, "On the power of curriculum learning in deep networks", ICML 2019

# From pre-defined to self-paced

**Using a teacher is still a form of pre-defined curriculum however, what if we want to have an adaptive measure of difficulty, based on our current model?**

Moving away from a pre-defined curriculum towards model "competence"

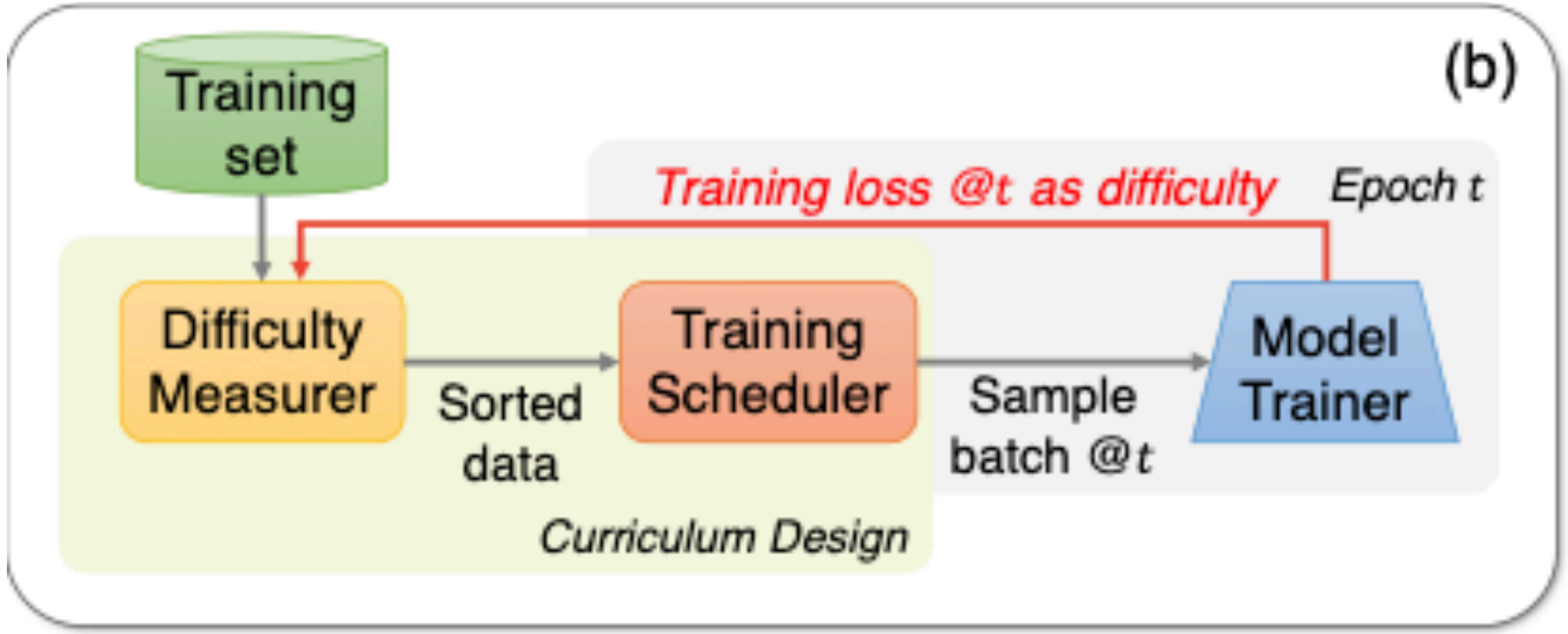



Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

# From pre-defined to self-paced

**Often this is called self-paced learning**

We now rely on the model's current hypothesis at each point in time to assign difficulty to the training instances, rather than ranking according to the target hypothesis.




Wang et al, "A Survey on Curriculum Learning", TPAMI 2021

# Self-paced & self-taught

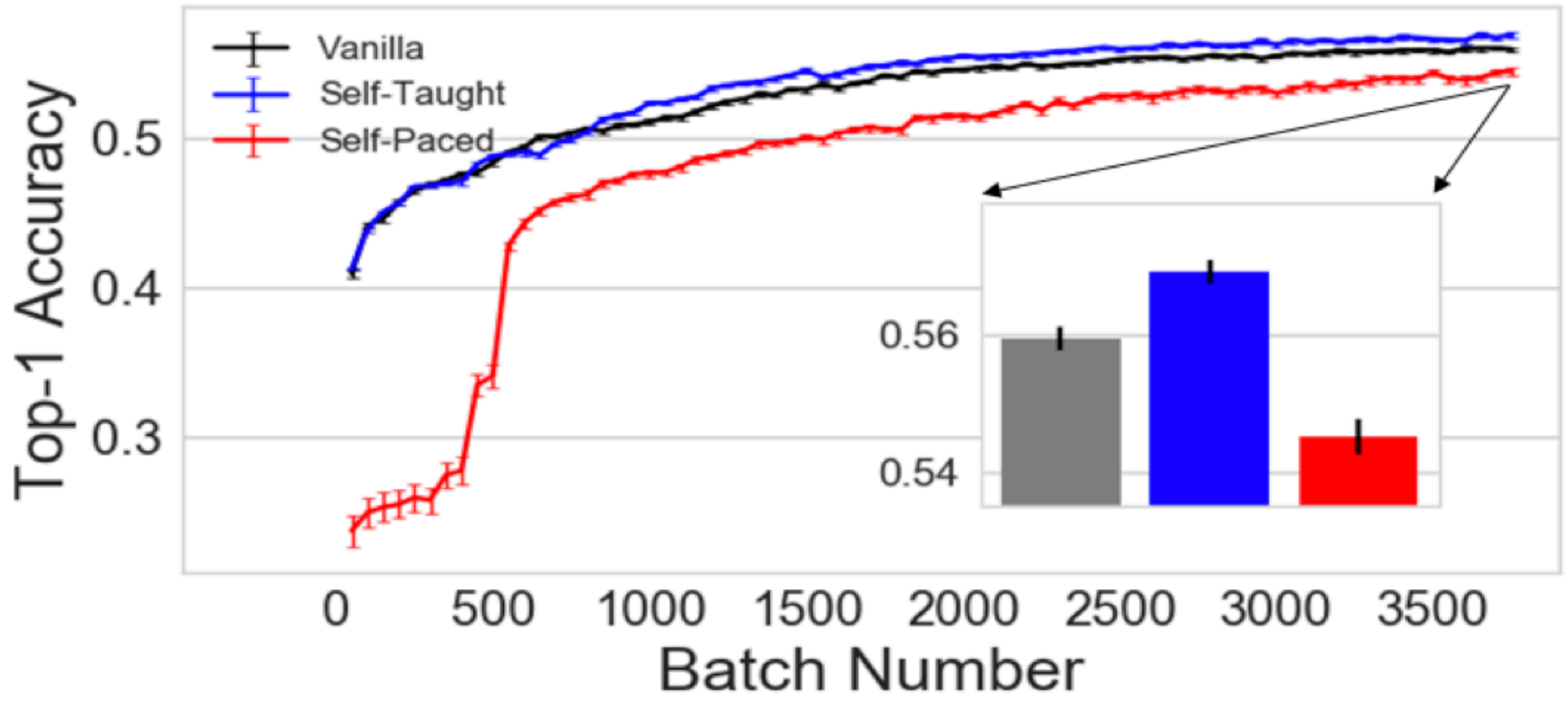
Somewhat related to what we've already seen

**Self-paced learning:**

Measure the difficulty of an instance according to current loss/predictions etc. (related to the ideas in *active learning*)

**Self-taught learning:**

Train a model fully, measure each instance according to final model, assign difficulty score and start over with curriculum -> repeat (related to the ideas in *boosting*)




Hacohen & Weinshall, "On the power of curriculum learning in deep networks", ICML 2019

# Does intrinsic ordering/pacing exist?

If we can use the loss of a model as a measure of difficulty, does this perhaps mean that models “intrinsically order” examples during regular training to some degree as well?

# Intrinsic ordering & pacing?

**An experiment: let's train multiple models & check how similar representations are**

Why is this interesting?

Recall that we typically use mini-batches + stochastic gradient descent,

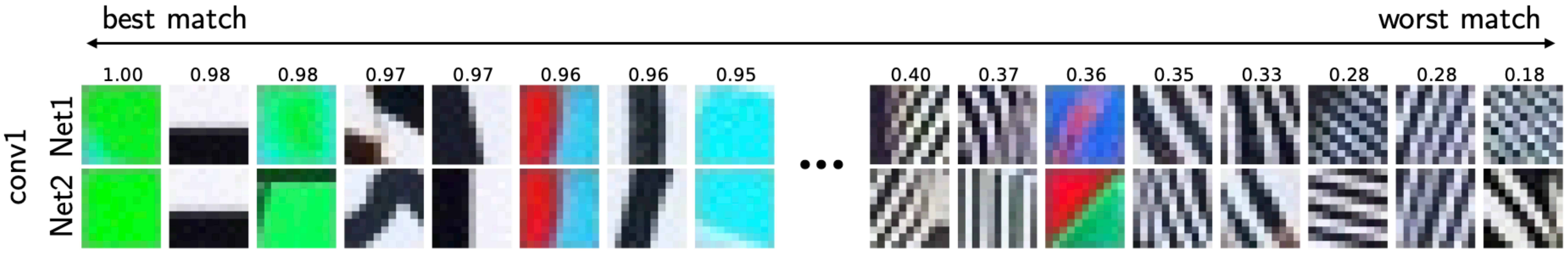
where data is shuffled differently in every "epoch"

# Intrinsic ordering & pacing?

**An experiment: let's train multiple models & check how similar representations are**

Why is this interesting?

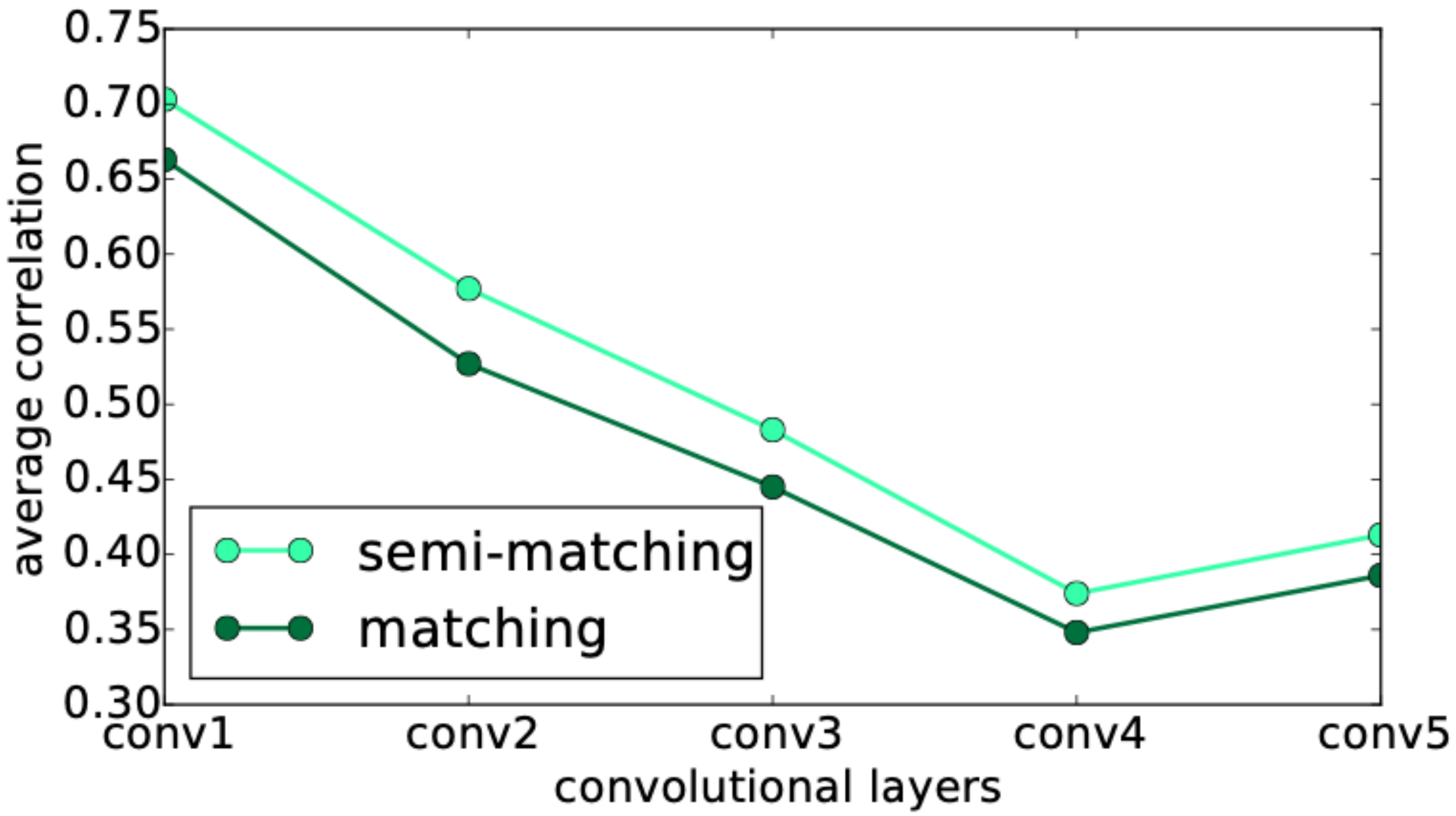
Recall that we typically use mini-batches + stochastic gradient descent, where data is shuffled differently in every "epoch"




Li & Yosinski et al, "Convergent learning: do different neural networks learn the same representations", ICLR 2016

# Intrinsic ordering & pacing?

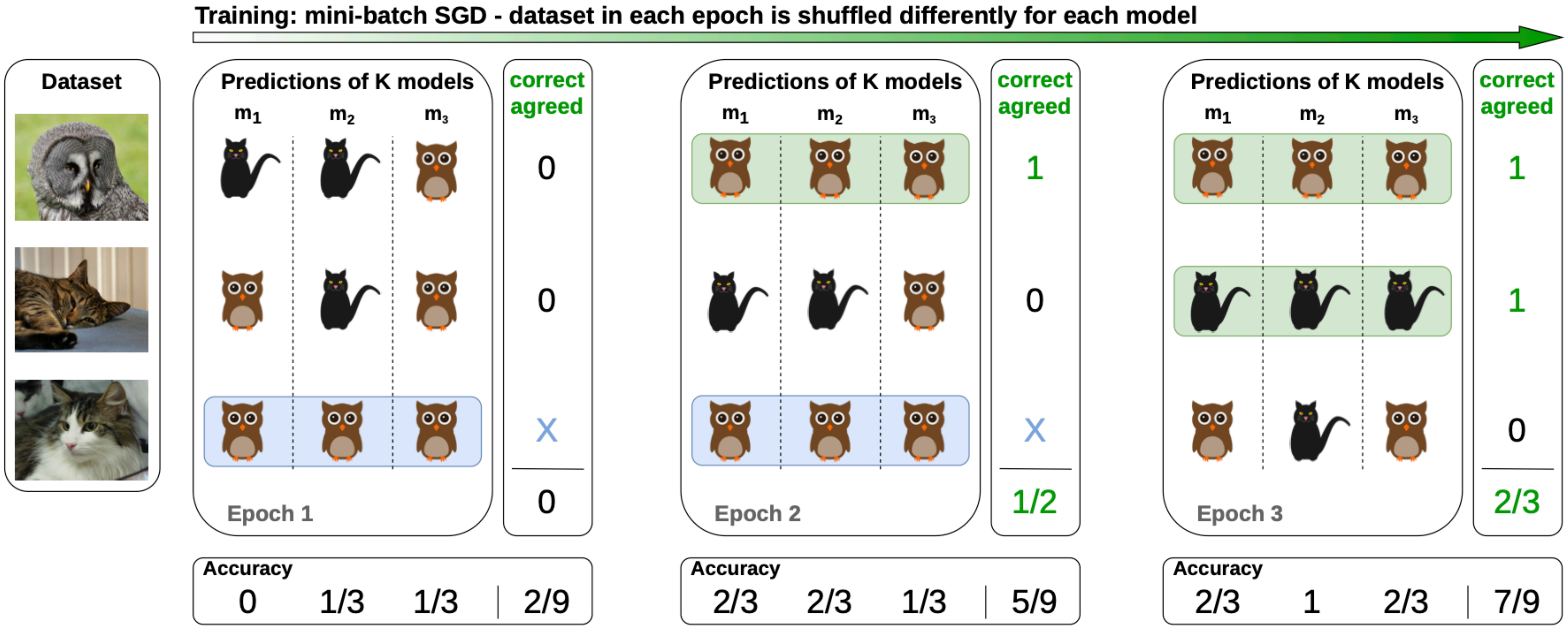
If we try to do a bi-partite matching of the representations in each neural network layer of different networks, there seem to exist strong correlations, especially in early, “generic” features




Li & Yosinski et al, “Convergent learning: do different neural networks learn the same representations”, ICLR 2016

# Intrinsic ordering & pacing?

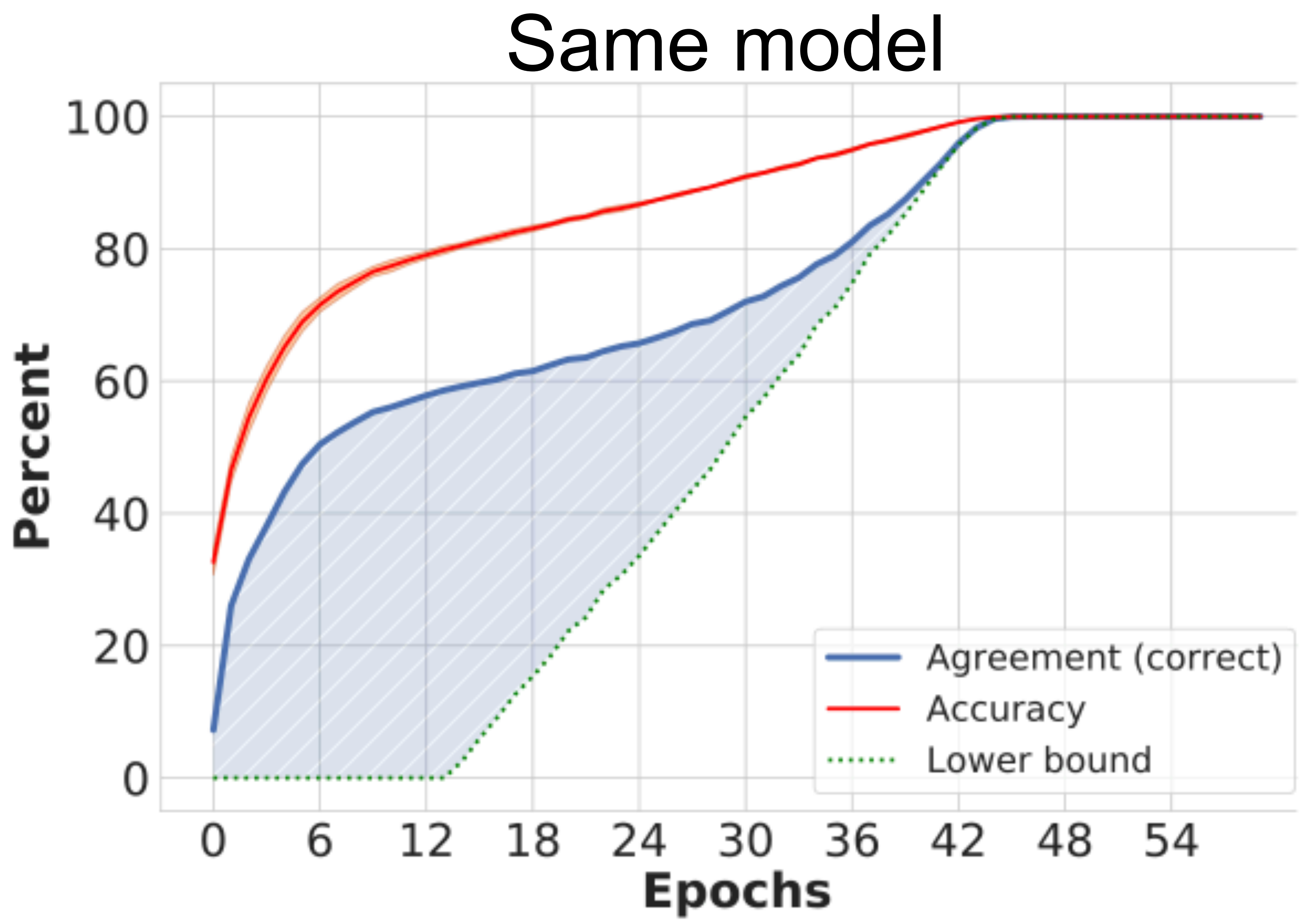
## A step further: let's train multiple models & check how much they agree on instances

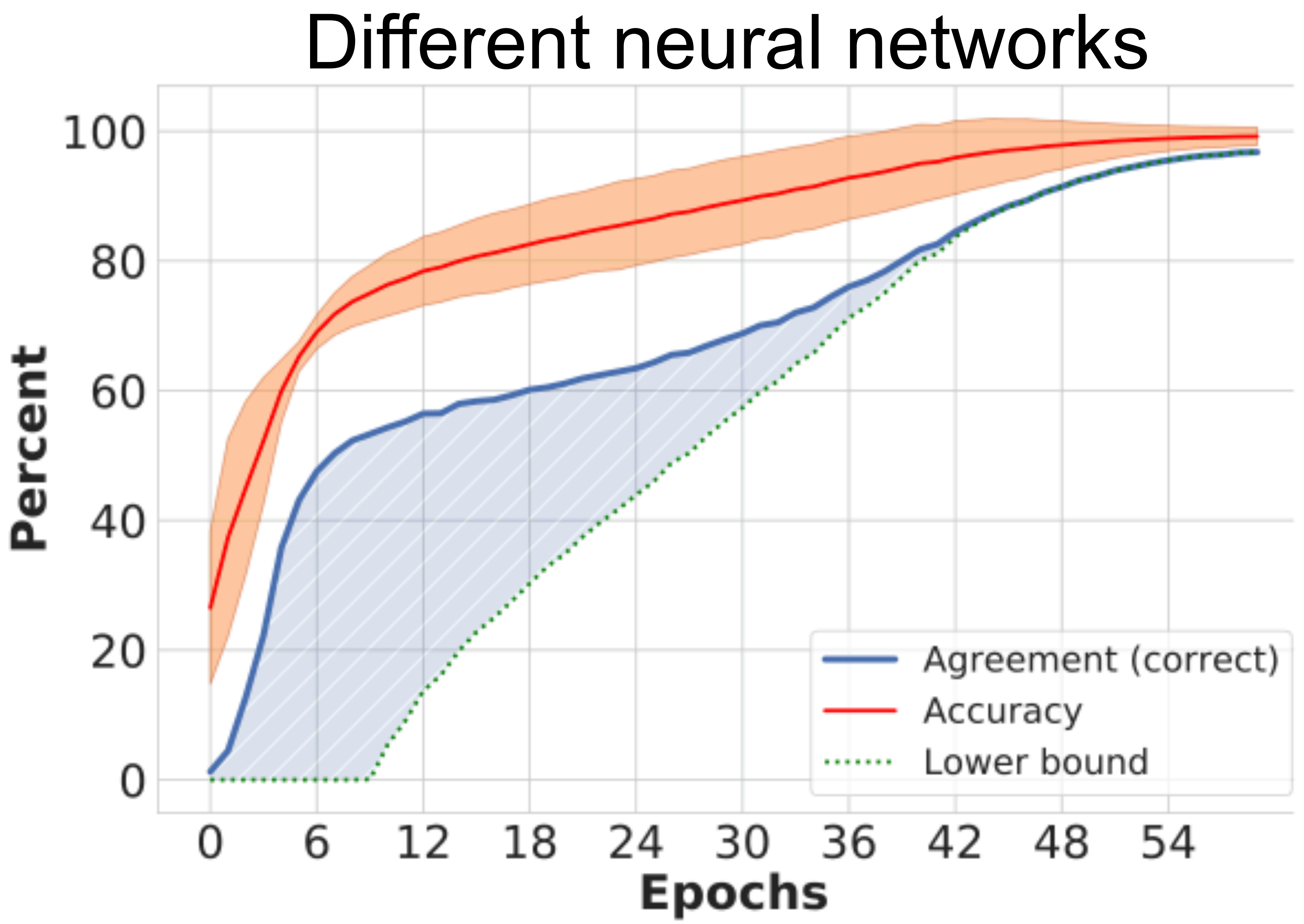



Pliushch et al, “When Deep Classifiers Agree: Analyzing correlations between learning order and image statistics”, ECCV 2022
As a reproduction & extension to the earlier Hacohen et al, “Let's agree to agree: neural networks share classification order on on real datasets”, ICML 2020

# Intrinsic ordering & pacing?

**Different neural networks seem to classify the same instances correctly at similar points in training: they "agree to agree"**



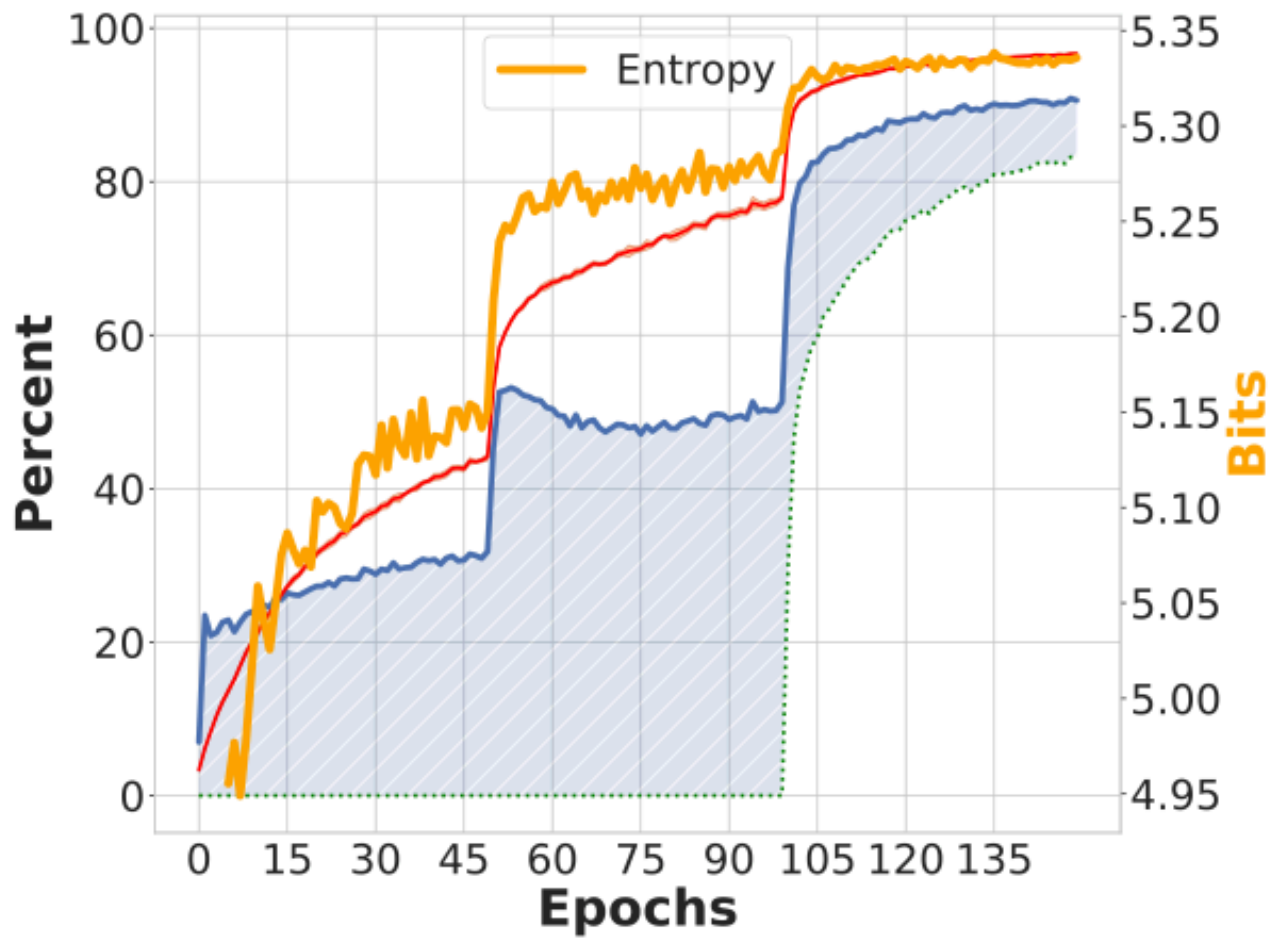




Pliushch et al, "When Deep Classifiers Agree: Analyzing correlations between learning order and image statistics", ECCV 2022
As a reproduction & extension to the earlier Hacohen et al, "Let's agree to agree: neural networks share classification order on on real datasets", ICML 2020

# An outlook to closing the circle

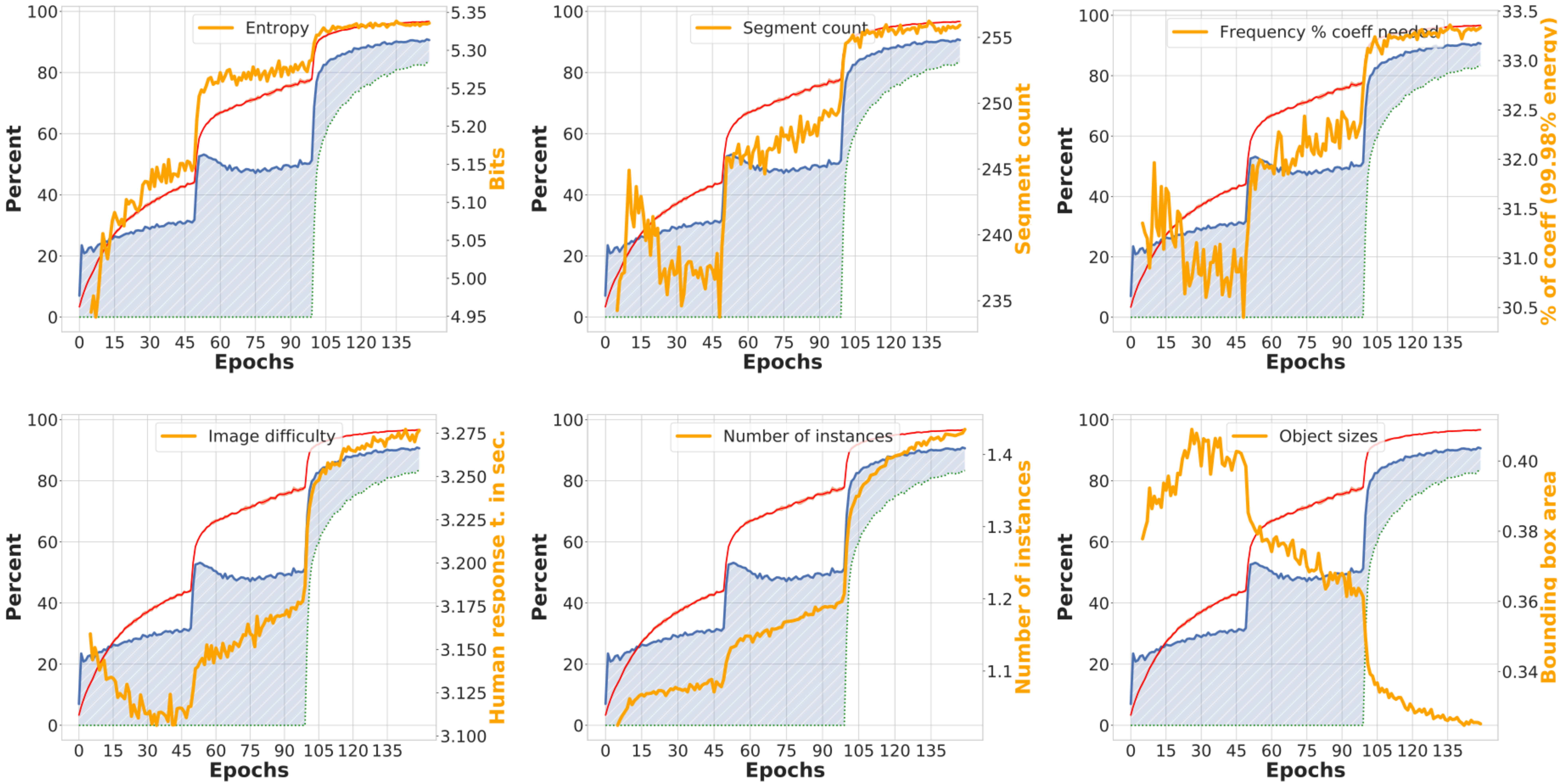
# Could such inherent agreement be related to our notions of difficulty?

# NN training, order & difficulty




Pliushch et al, “When Deep Classifiers Agree: Analyzing correlations between learning order and image statistics”, ECCV 2022

# NN training, order & difficulty

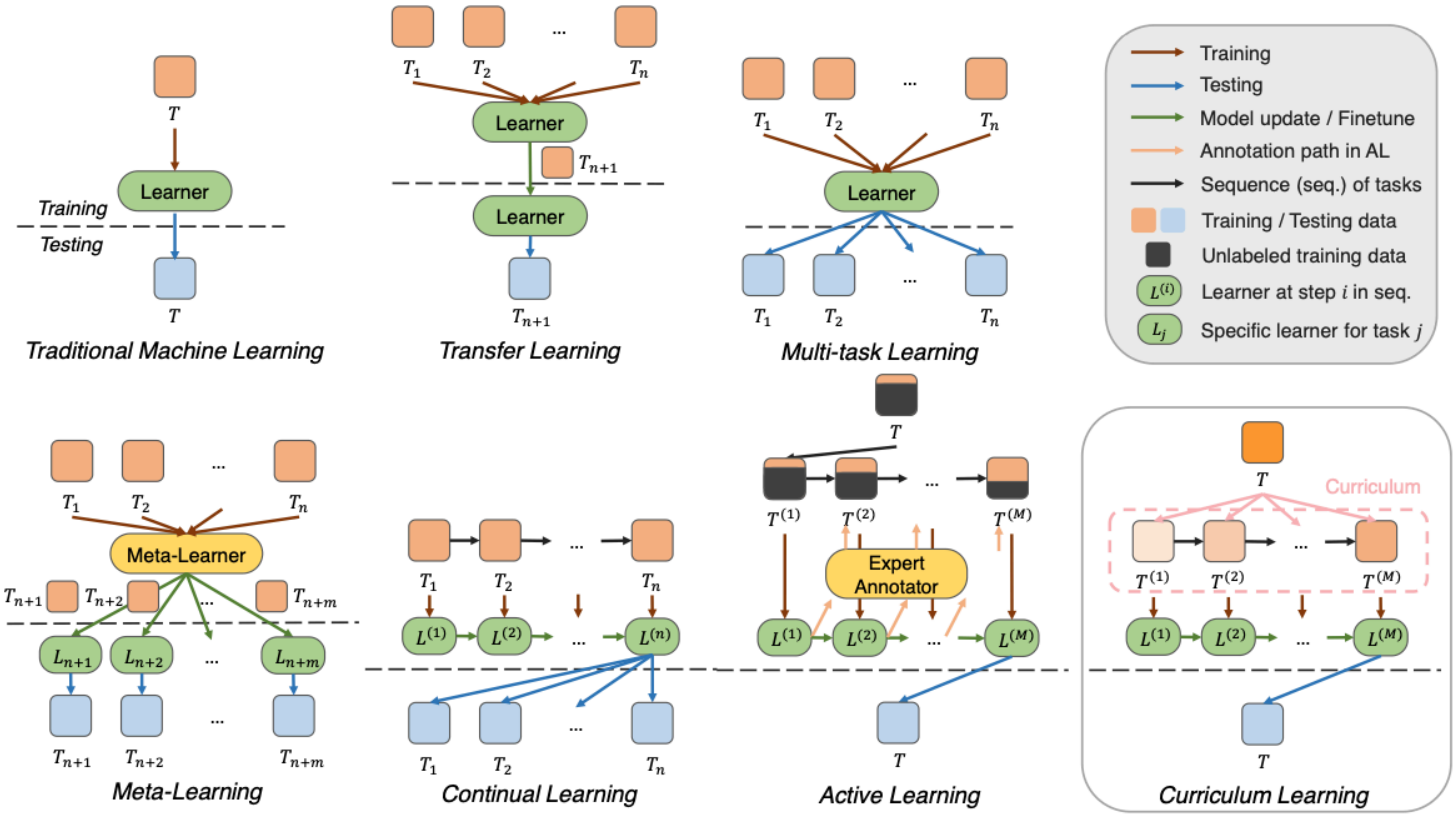



Pliushch et al, “When Deep Classifiers Agree: Analyzing correlations between learning order and image statistics”, ECCV 2022

**As always: a disclaimer**
**there’s much we don’t yet know**

**And a final note =)**

# It’s about set-up & evaluation




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