Open World Lifelong Learning

A Continual Machine Learning Course

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 Tuesday 17:30 - 19:00 CEST

Course Homepage

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

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

Recall: How to avoid forgetting?

Paradigms for Continual Learning

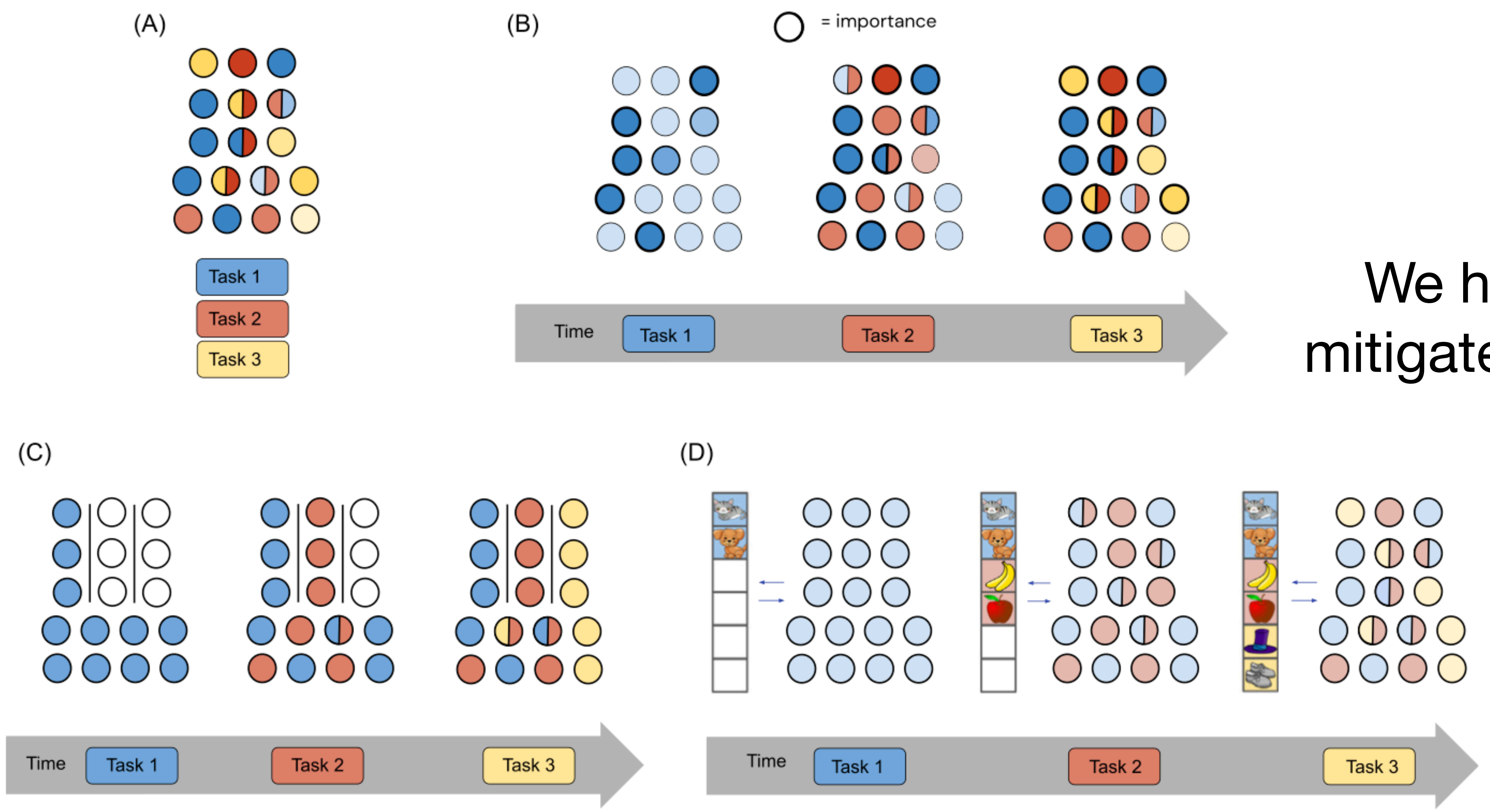


Figure 1. (A) Independent and identically distributed learning methods are standard for nonsequential, multitask learning. In this regime, tasks are learned simultaneously to avoid forgetting and instability. (B) Gradient-based approaches preserve parameters based on their importance to previously learned tasks. (C) Modularity-based methods define hard boundaries to separate task-specific parameters (often accompanied by shared parameters to allow transfer). (D) Memory-based methods write experience to memory to avoid forgetting.

We have investigated ways to mitigate **(catastrophic) forgetting**

Hadsell et al, “Embracing Change: Continual Learning in Deep Neural Networks”, Trends in Cognitive Sciences 24:12, 2020

Recall: continual experiments

But where does our data sequence actually come from?

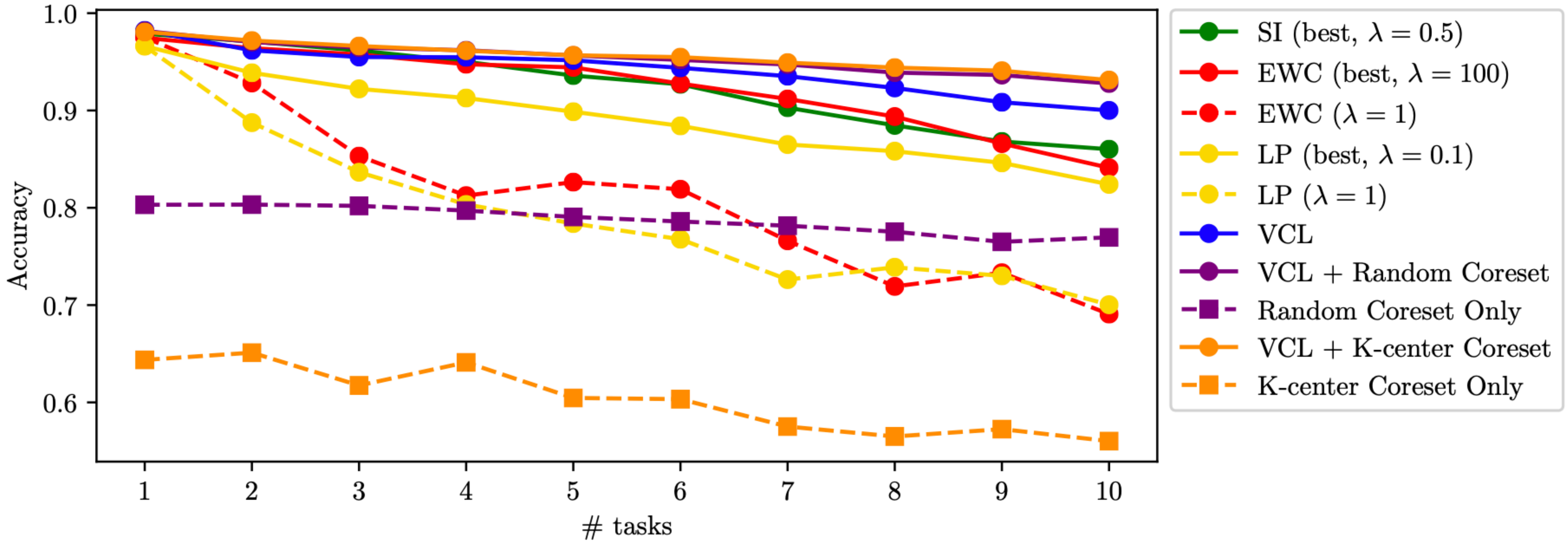


Figure 2: Average test set accuracy on all observed tasks in the Permuted MNIST experiment.

Nguyen et al, “Variational Continual Learning” ICLR 2018

Week 5: Active Learning

Active learning

In a training process on some initial data

you now want to **decide what data to include next.**

What do you think: **why should we be interested** in this question?

Active learning

Selecting upcoming data

Popular in supervised learning:

data is cheap in comparison to labels

- Also referred to as “query learning”
- Underlying mechanism for queries called “acquisition function”

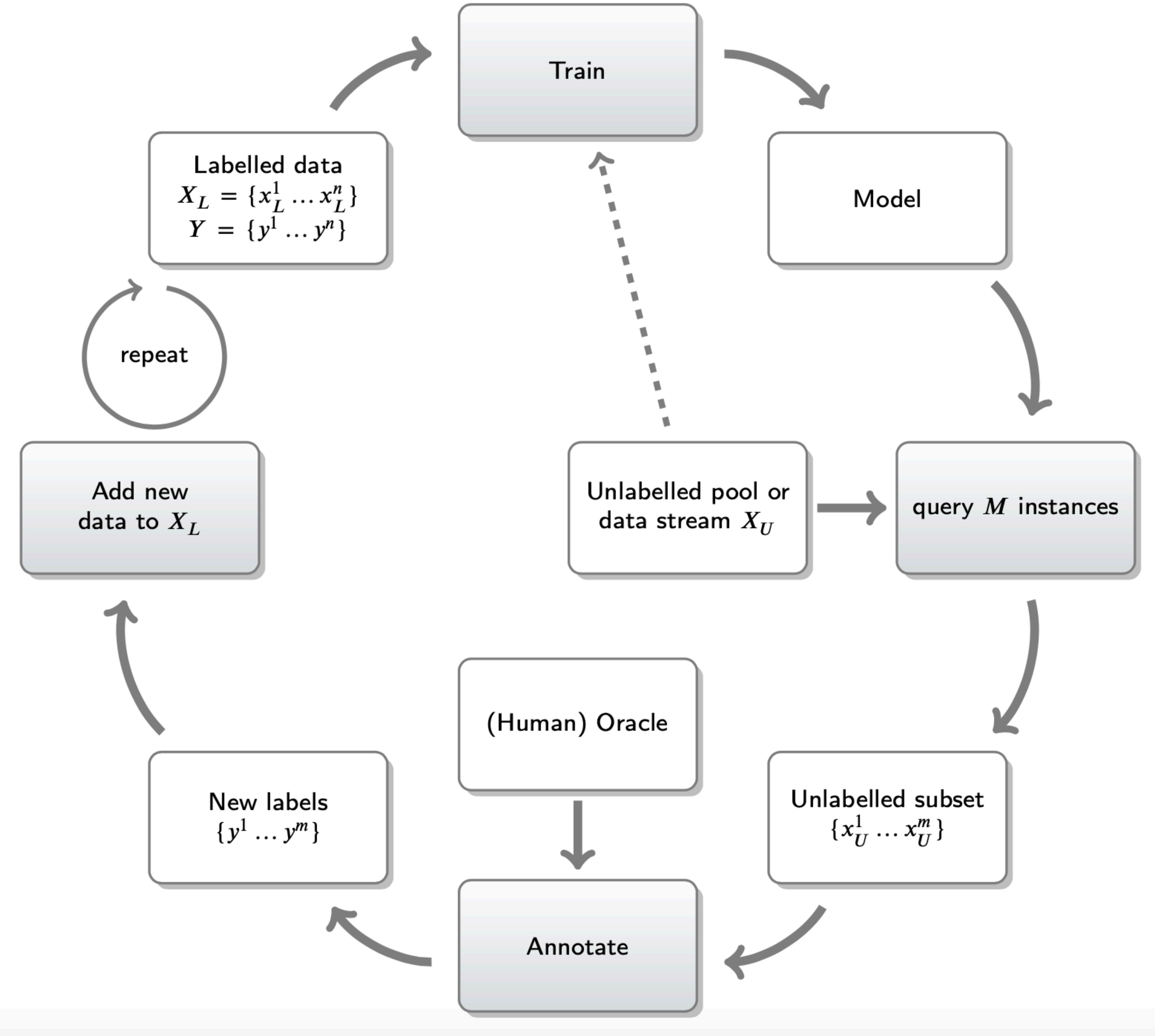

Train
Model
query M instances
Unlabelled pool or data stream X_U
Unlabelled subset $\{x_U^1 \dots x_U^m\}$
(Human) Oracle
Annotate
New labels $\{y^1 \dots y^m\}$
Add new data to X_L
repeat
Labelled data $X_L = \{x_L^1 \dots x_L^n\}$ $Y = \{y^1 \dots y^n\}$

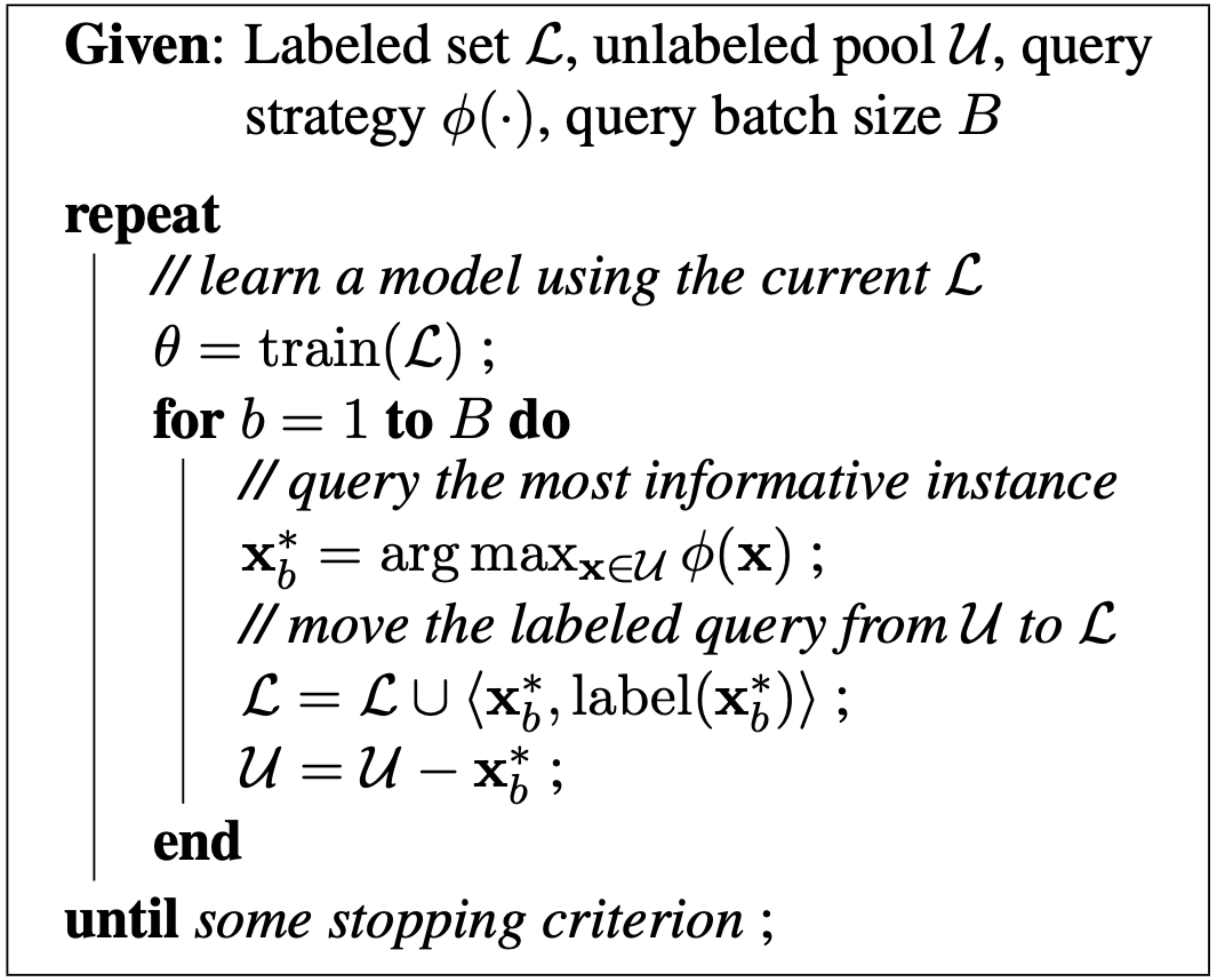
Figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Mundt et al 2020

Pool based active learning

Selecting upcoming data

(Unlabelled) data pools can be huge

- Not every data point is equally informative
- Typically referred to as “pool based” active learning
- Typically accumulate data after selection



```
Given: Labeled set $\mathcal{L}$, unlabeled pool $\mathcal{U}$, query
       strategy $\phi(\cdot)$, query batch size $B$

repeat
    // learn a model using the current $\mathcal{L}$
    $\theta = \text{train}(\mathcal{L})$ ;
    for $b = 1$ to $B$ do
        // query the most informative instance
        $\mathbf{x}_b^* = \arg\max_{\mathbf{x} \in \mathcal{U}} \phi(\mathbf{x})$ ;
        // move the labeled query from $\mathcal{U}$ to $\mathcal{L}$
        $\mathcal{L} = \mathcal{L} \cup \langle \mathbf{x}_b^*, \text{label}(\mathbf{x}_b^*) \rangle$ ;
        $\mathcal{U} = \mathcal{U} - \mathbf{x}_b^*$ ;
    end
until some stopping criterion ;
```

Algorithm 1: Pool-based active learning.

Settles & Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks”, EMNLP 2008

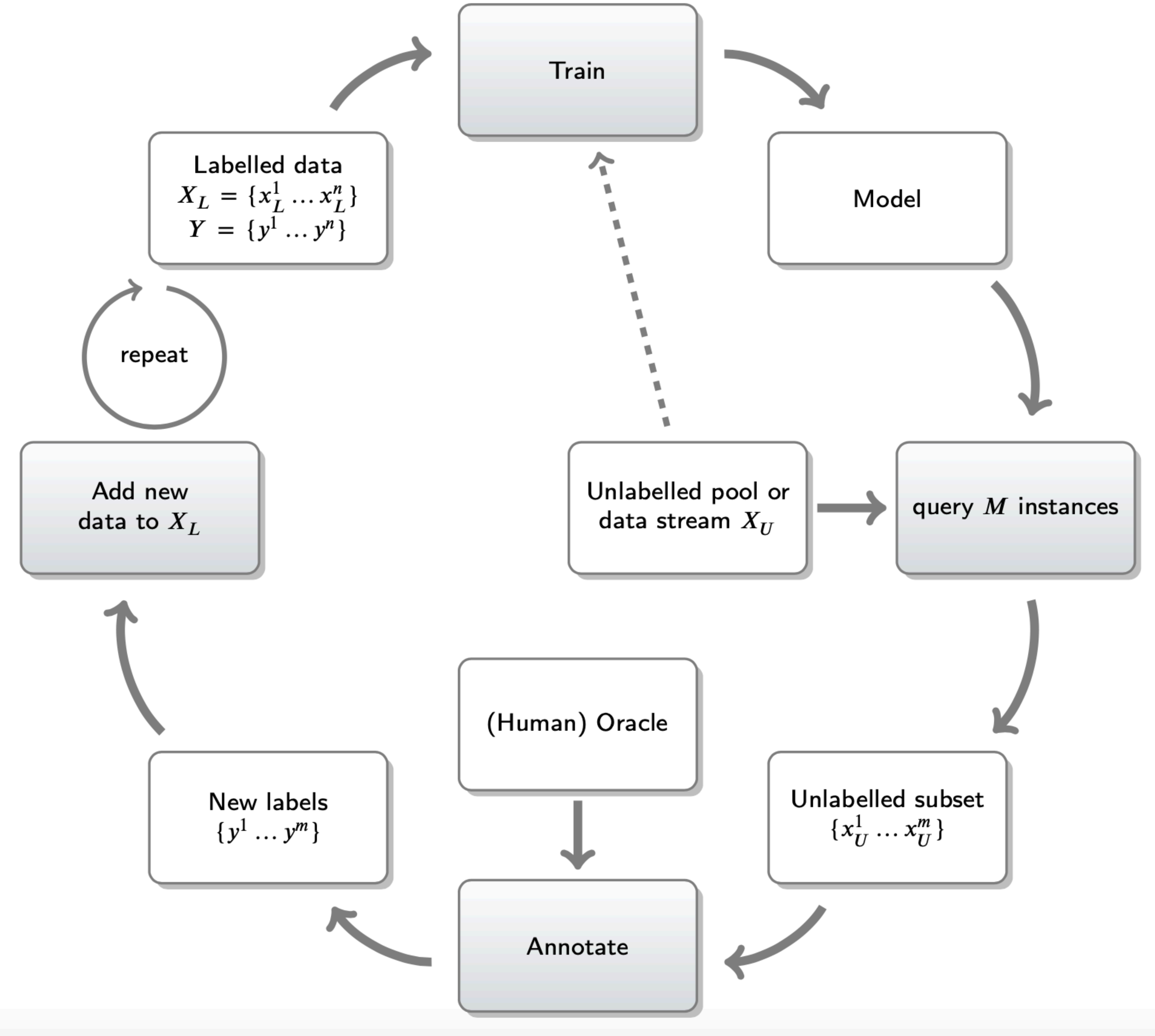
What assumptions could me make about the set-up?

(Pool based) Active learning

Many potential assumptions

(non-exhaustive)

- Pool of data entirely available upfront
- Typically accumulate data after selection
- One data element at a time vs. batches
- Queries only allowed to be based on training of already available data
- Re-train model on new dataset vs. continued training?
- Oracle: infallible versus noisy



Train
Model
Labelled data $X_L = \{x_L^1 \dots x_L^n\}$ $Y = \{y^1 \dots y^n\}$
repeat
Add new data to X_L
Unlabelled pool or data stream X_U
query M instances
(Human) Oracle
New labels $\{y^1 \dots y^m\}$
Unlabelled subset $\{x_U^1 \dots x_U^m\}$
Annotate

Figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Mundt et al 2020

Acquisition functions: what techniques can you think of?

Active learning perspectives

Version space reduction

The more formal approach: reduce the set/space of possible hypotheses $h : \mathscr{X} \rightarrow \mathscr{Y}$ by removing the ones that are inconsistent with the data

Uncertainty & heuristics

The perhaps intuitive approach: use the predictions, or maybe even better, uncertainty in the predictions for the queries

Core sets & representation learning

The distribution based approach: maximizing distribution coverage instead of reducing the possible set of hypotheses (version space) explicitly

Should we use discriminative or generative models?

Discriminative or generative

Discriminative models could allow for natural ways to assess “novelty” of a new example

-> *But caution*: overconfidence phenomena (recall lecture 1, topic in upcoming lecture)

Generative models could allow to reason about the data distribution

-> *But caution*: our parameters only reflect the distribution seen so far! (do we use the pool?)

We will see that the choice also heavily depends on the set-up assumption!

See Zhang & Oles, “A Probability Analysis on the value of Unlabeled Data for Classification Problems”, ICML 2000

Version Space

Version space (Mitchel 1978)

- Assume that there exist hypotheses consistent with the labeled data points $h : \mathscr{X} \rightarrow \mathscr{Y}$

  version space: $VS(D) = \{h \in H \,|\, \texttt{cons}(h, D)\}$

- **Specific hypotheses**: cover positive examples & as little remaining feature space as possible
- **General hypotheses**: cover positive examples & as much of the remaining feature space as possible
- **Version space:** represented as green rectangles

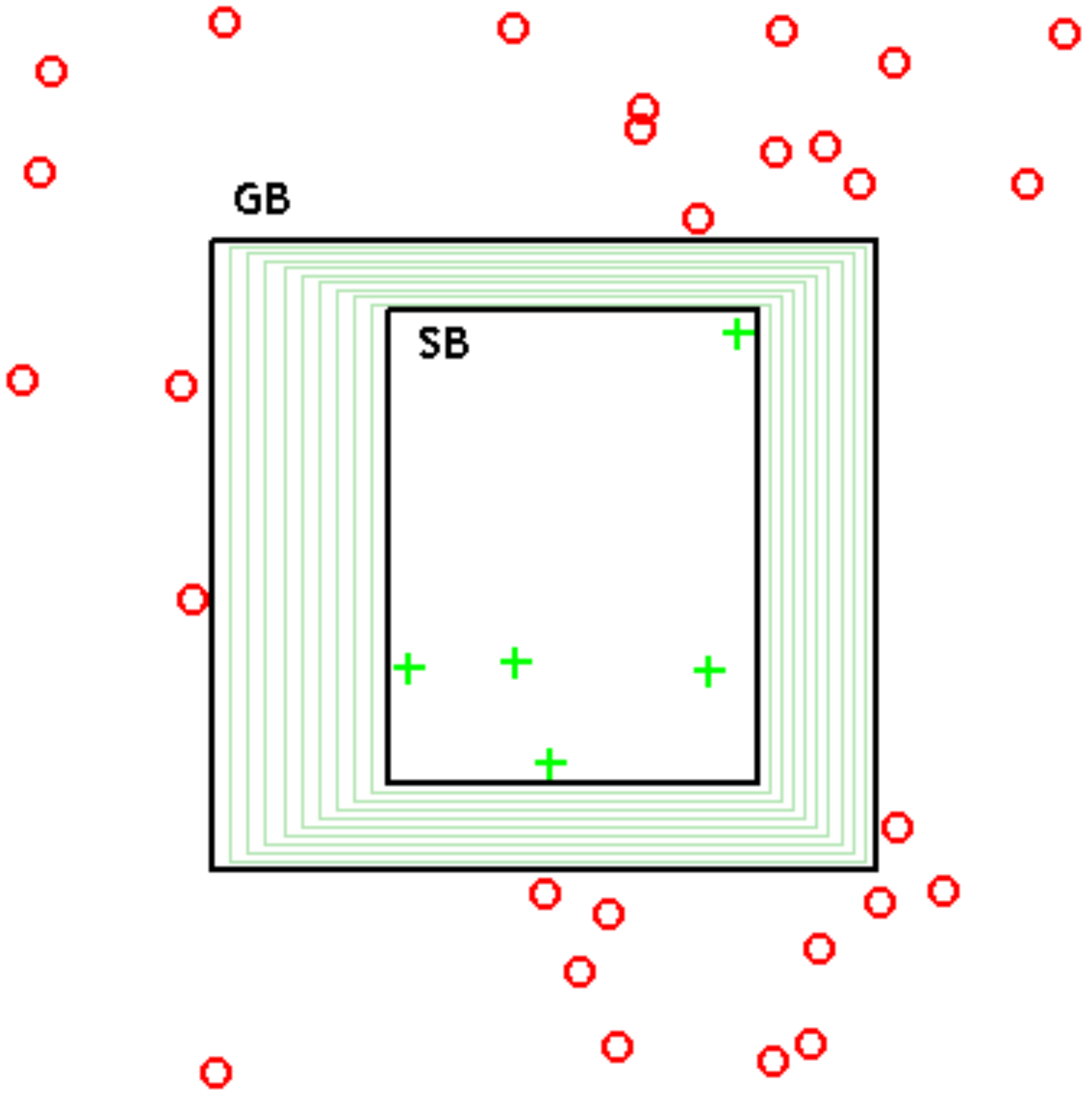


Figure from https://en.wikipedia.org/wiki/File:Version_space.png in the public domain

Version space reduction

"**Generalization as Search**", Mitchell 1982

We could query such that the **version space**:

$VS(D) = \{h \in H \,|\, \texttt{cons}(h, D)\}$,i.e. the set of consistent hypotheses, quickly **gets reduced**

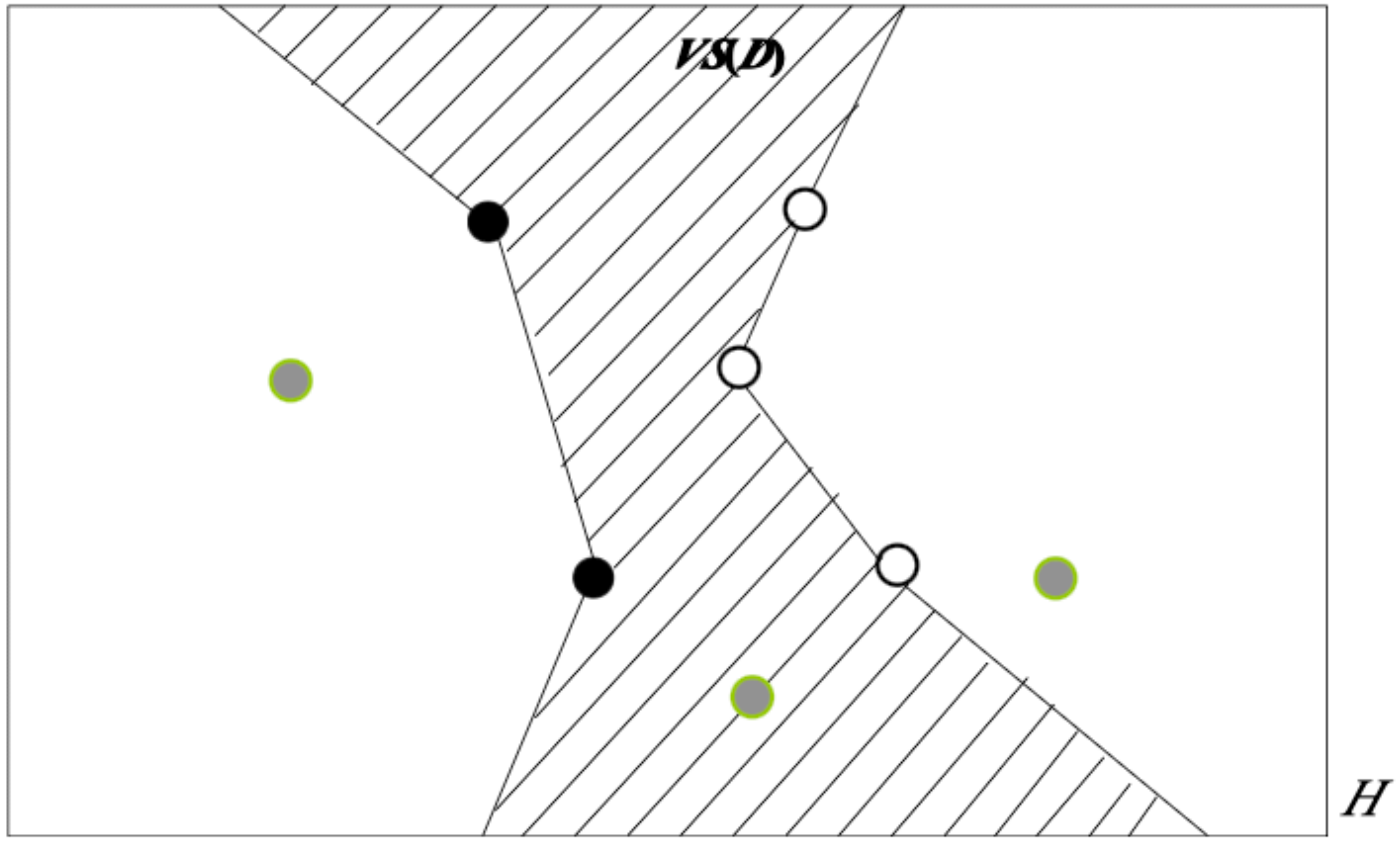


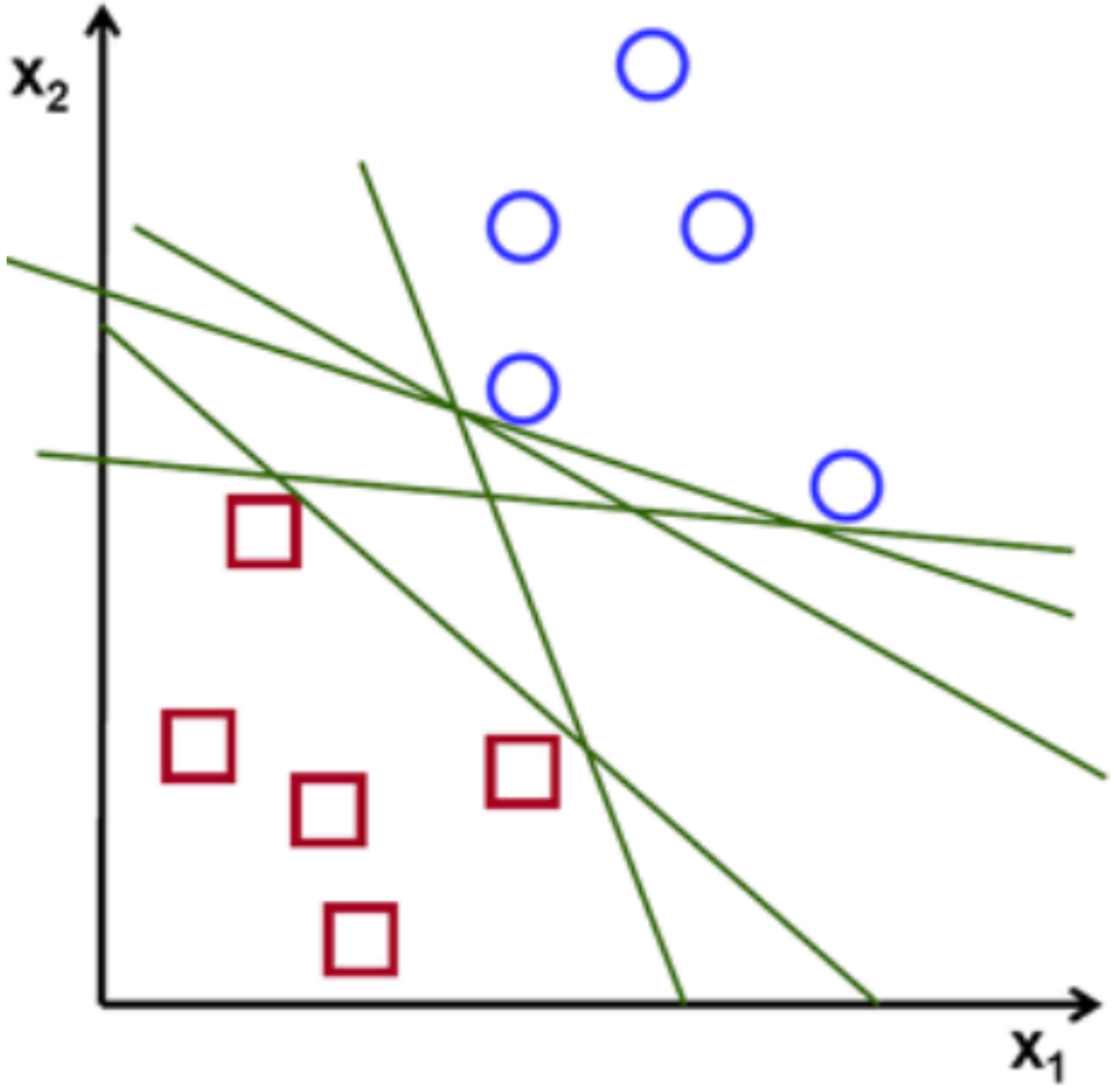
Figure from presentation of "Ensembles of Classifiers" by Evgueni Smirnov, slides available at: https://slideplayer.com/slide/10075963/

A very short excursion: support vector machines

Support vector machines (SVM)

Example: support vector machine (SVM)

- In principle, not completely different from logistic regression, neural networks etc.
- Choose **hyperplane that divides data points** into the two classes (1, -1)



https://towardsdatascience.com/support-vector-machine-vs-logistic-regression-94cc2975433f

Support vector machines (SVM)

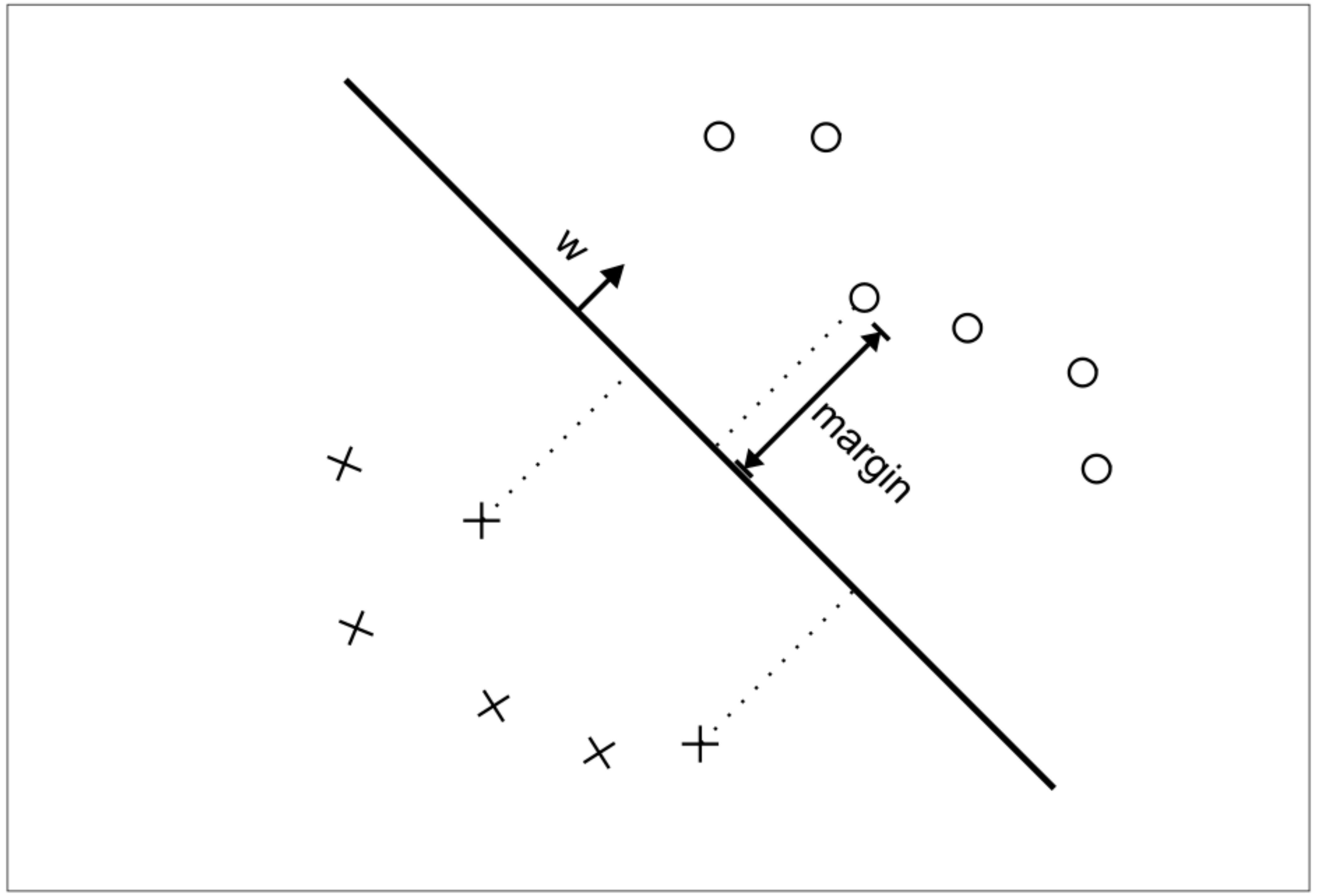
Example: support vector machine (SVM)

- Any hyperplane can be written as a set of points x satisfying:

$$w^T x - b = 0$$

where **w** is the **normal vector**

- **Margin:** $w^T x - b = 1$ & $w^T x - b = -1$



Tong & Koller, “Support Vector Machine Active Learning with Applications to Text Classification”, JMLR 2001

Support vector machines (SVM)

Example: support vector machine (SVM)

Hyperplane chosen to **maximize margin to closest instances**: the support vectors

- Rewritten $y_i(w^T x_i - b) = 0 \geq 1, \forall 1 \leq i \leq n$ (additionally, no points fall on the boundary)
- Optimization problem: **minimize ||w||** subject to: $y_i(w^T x_i - b) = 0 \geq 1, \forall 1 \leq i \leq n$

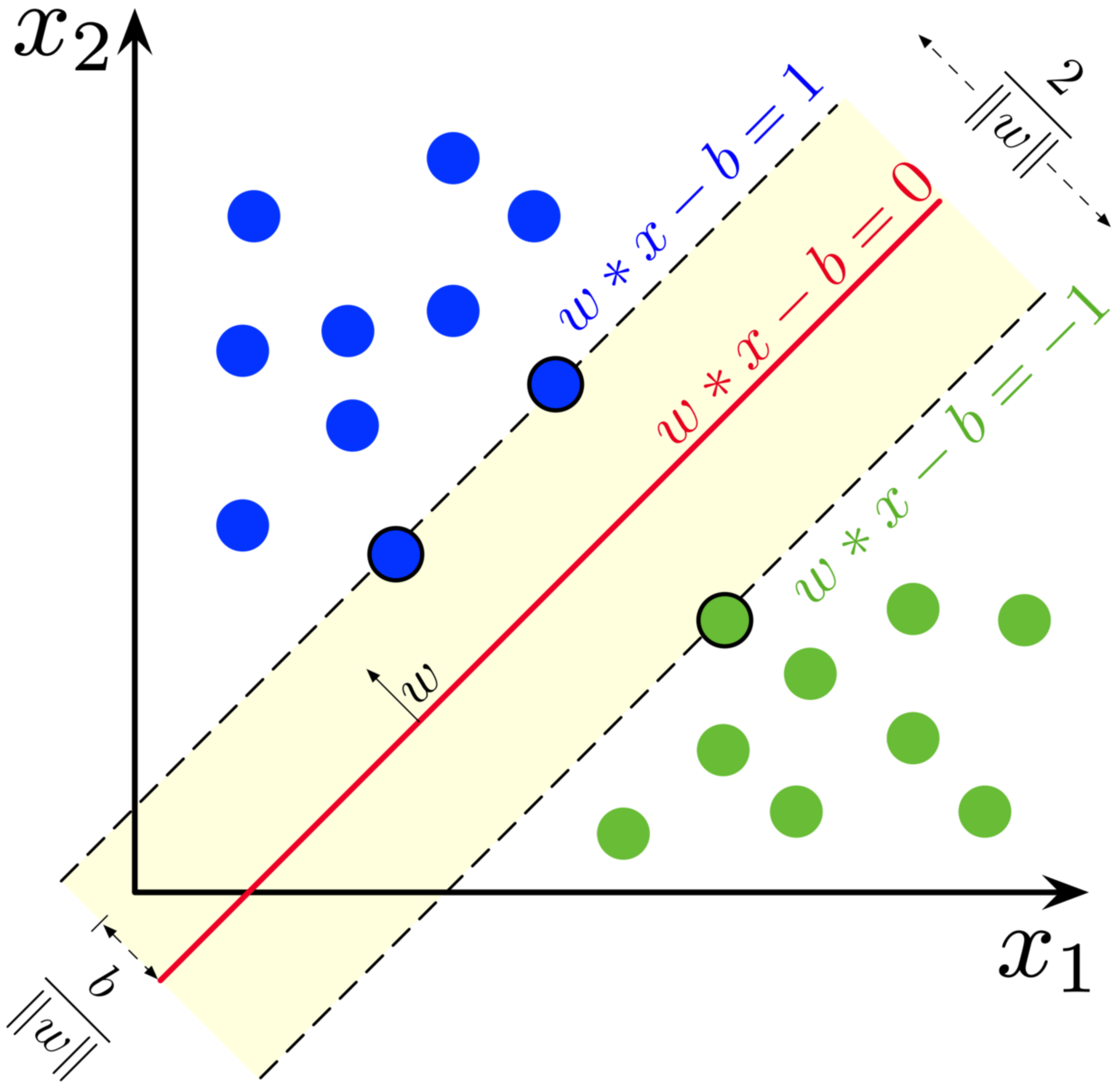


Figure from https://en.wikipedia.org/wiki/File:SVM_margin.png shared under CC 4.0 license

Support vector machines (SVM)

Example: support vector machine (SVM)

- Data is not always linearly separable
- We can **project** data to a (higher dimensional) **feature space** through kernel functions
- Example: polar coordinates

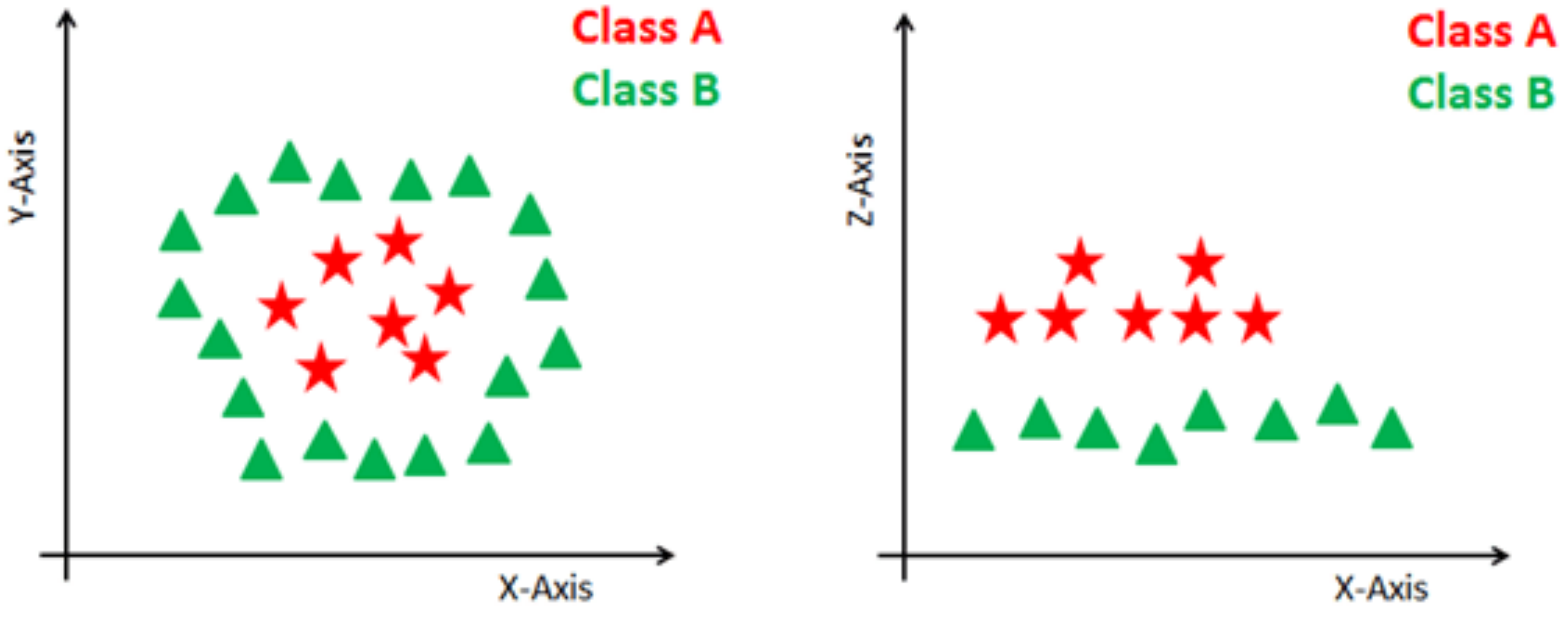
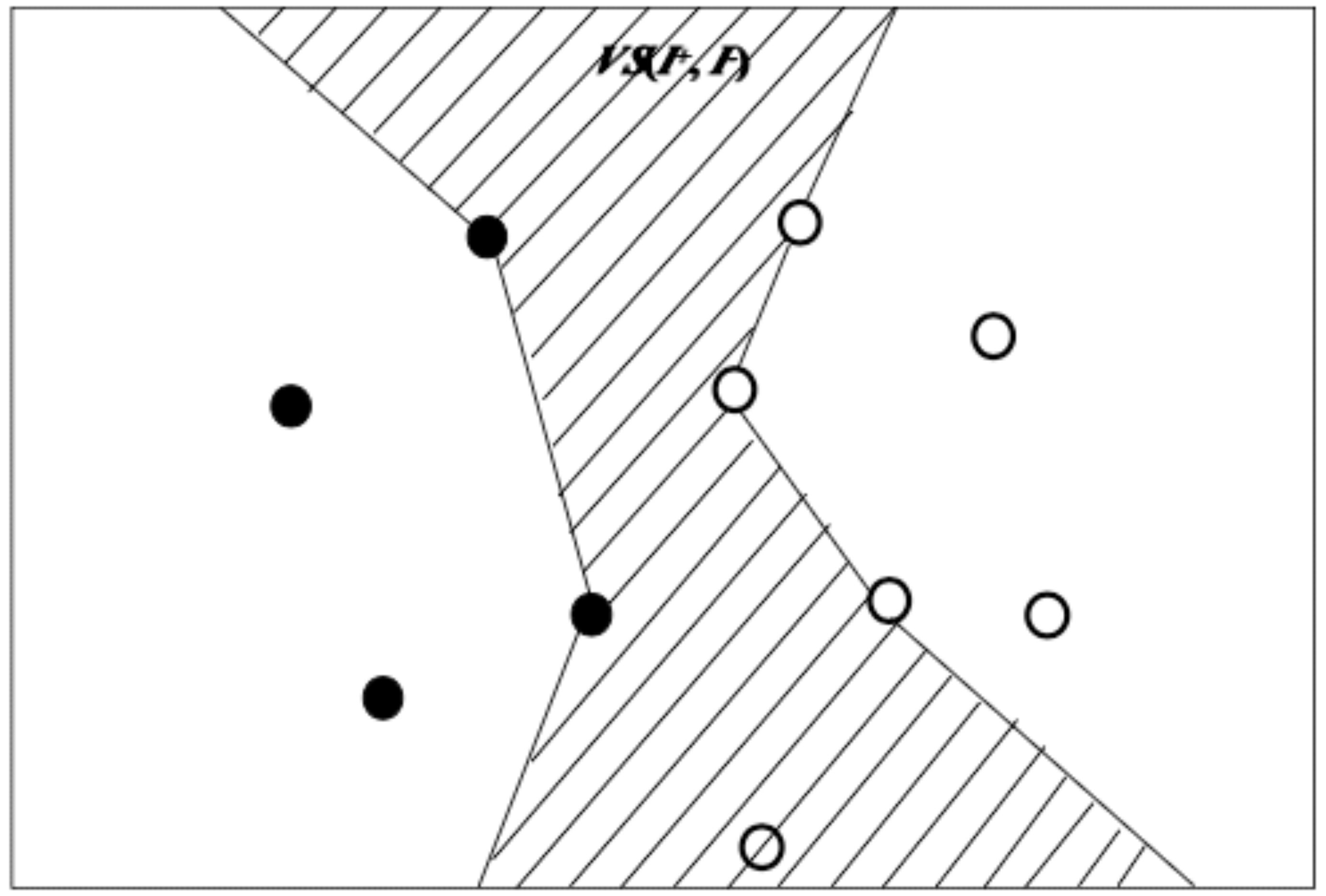
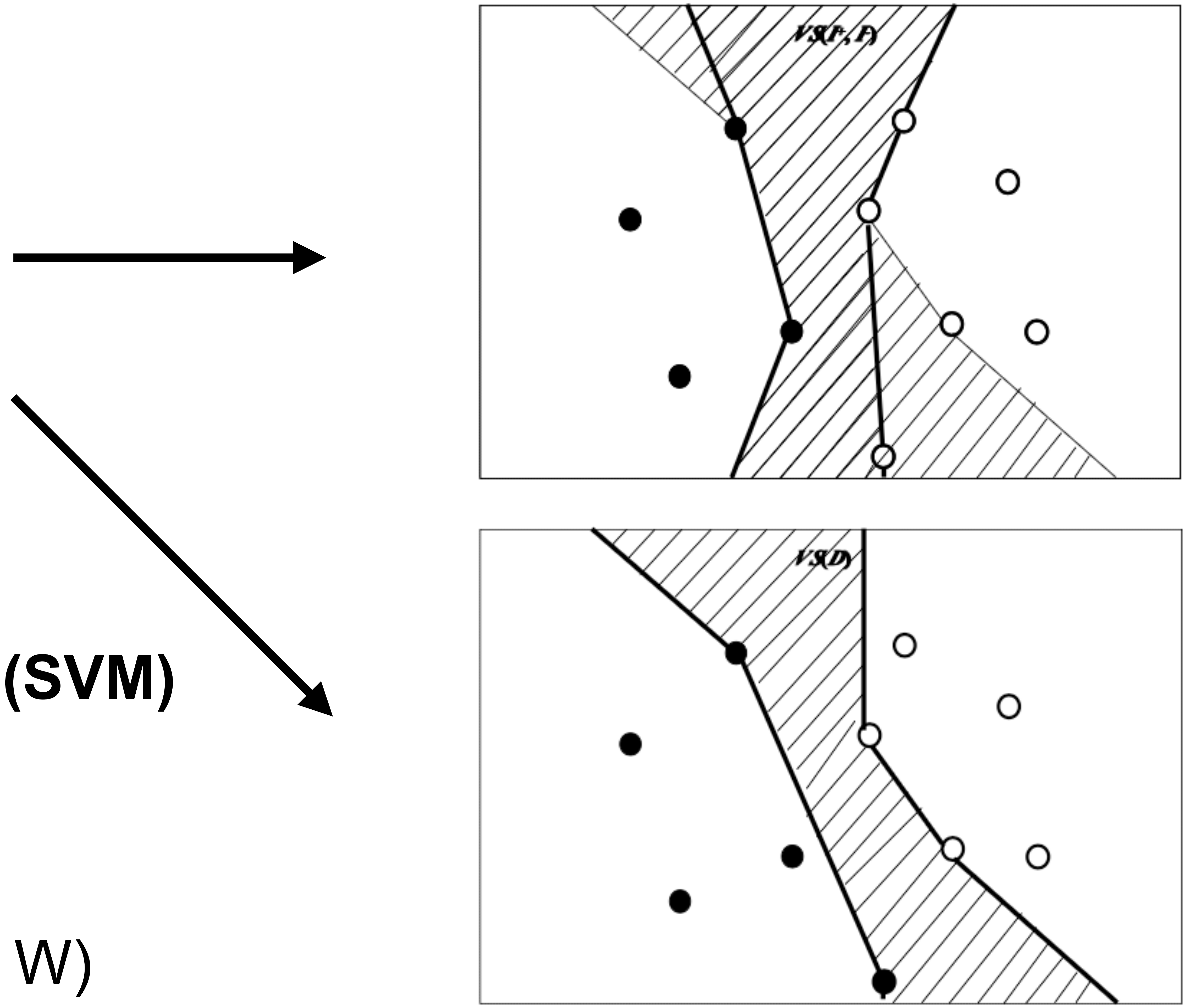


Figure from https://www.datacamp.com/tutorial/svm-classification-scikit-learn-python

Back to version spaces

Version space: SVM





An example: support vector machine (SVM)

Version space is **set of hyperplanes**

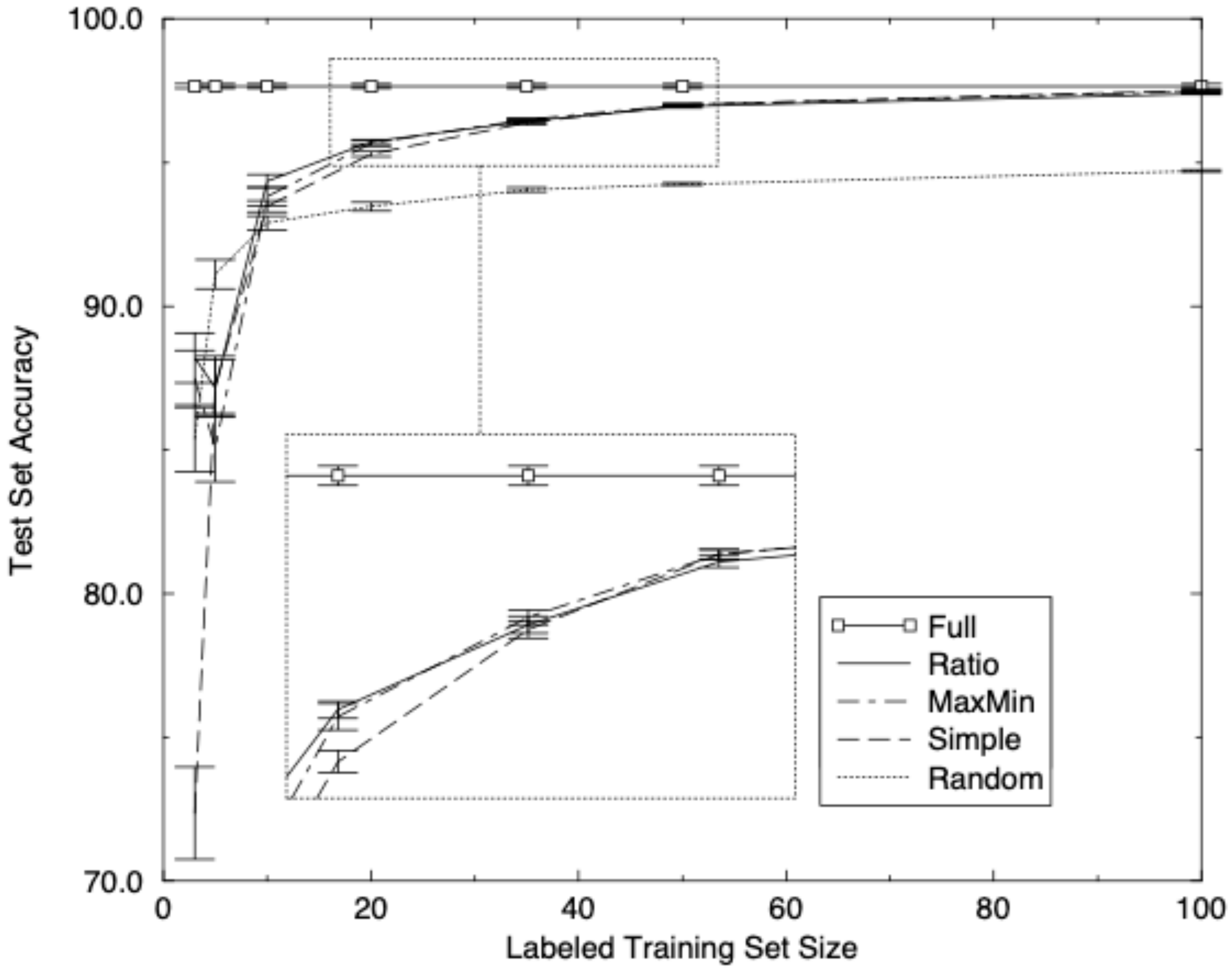
(or could be redefined through vectors W)

Figure from presentation of “Ensembles of Classifiers” by Evgueni Smirnov, slides available at: https://slideplayer.com/slide/10075963/

Version space: SVM

Active learning with SVMs

- Rapidly reduce version space
- Intuitively: choose successive queries that halve the version space
- Various approximations: is version space symmetric? Estimates of the size? etc.



Tong & Koller, "Support Vector Machine Active Learning with Applications to Text Classification", JMLR 2001

(Uncertainty &) Heuristics

An alternative to version space

Reducing the set of consistent hypotheses does not regard the **evaluation metric**.

We could also take a look at the machine learning **loss** and include points that would:

- **most reduce the expected error**
- **most change the current model**

An alternative to version space

Reducing the set of consistent hypotheses does not regard the **evaluation metric**.

We could also take a look at the machine learning **loss** and include points that would:

- **most reduce the expected error**
- **most change the current model**

"First-order Markov active learning aims to select a query $x^\star$, such that when the query is given label $y^\star$ and added to the training set, the learner trained on the resulting set $D+(x^\star,y^\star)$ has lower error than any other x"

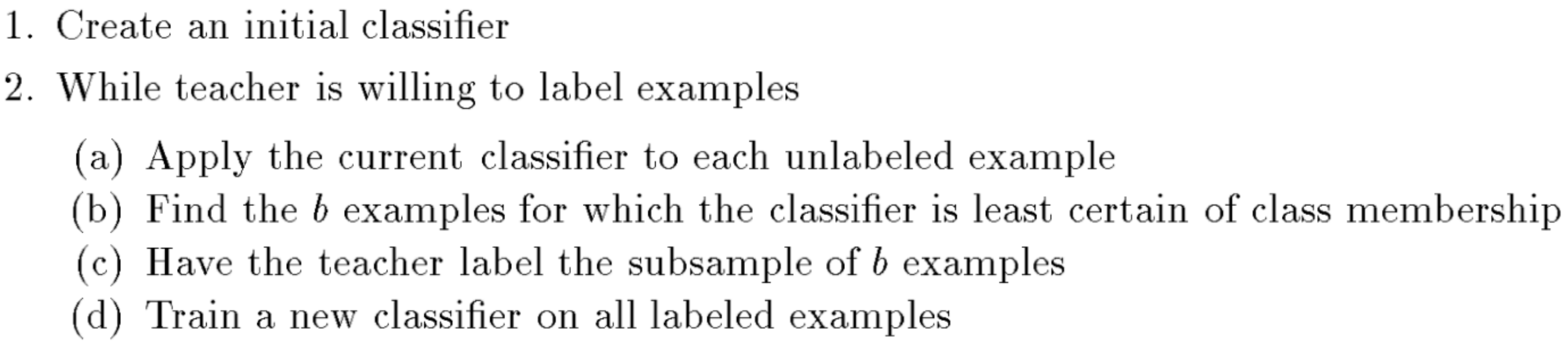
Roy & McCallum, "Toward Optimal Active Learning through Monte Carlo Estimation of Error Reduction", ICML 2001)

(See also Cohn et al, "Active learning with statistical models", JAIR 4, 1996)

The simplest (?) approach

Version spaces & expected error reduction can be complicated (& computationally heavy).

Simple heuristics are thus still popular, especially in deep learning



```
1. Create an initial classifier
2. While teacher is willing to label examples
    (a) Apply the current classifier to each unlabeled example
    (b) Find the $b$ examples for which the classifier is least certain of class membership
    (c) Have the teacher label the subsample of $b$ examples
    (d) Train a new classifier on all labeled examples
```

Figure 1. An algorithm for uncertainty sampling with a single classifier.

Lewis & Gale, "A Sequential Algorithm for Training Text Classifiers", ACM-SIGIR conference on research and development in information retrieval 1994

Information theoretic quantities

Instead of pure output confidence, we could resort to more **information theoretic** approaches

Example: maximize expected information gain by querying examples with **largest entropy** (as a measure of disorder, related to information gain)

$$H(p) = -\sum_{i}^{c} p_i \log_2(p_i)$$

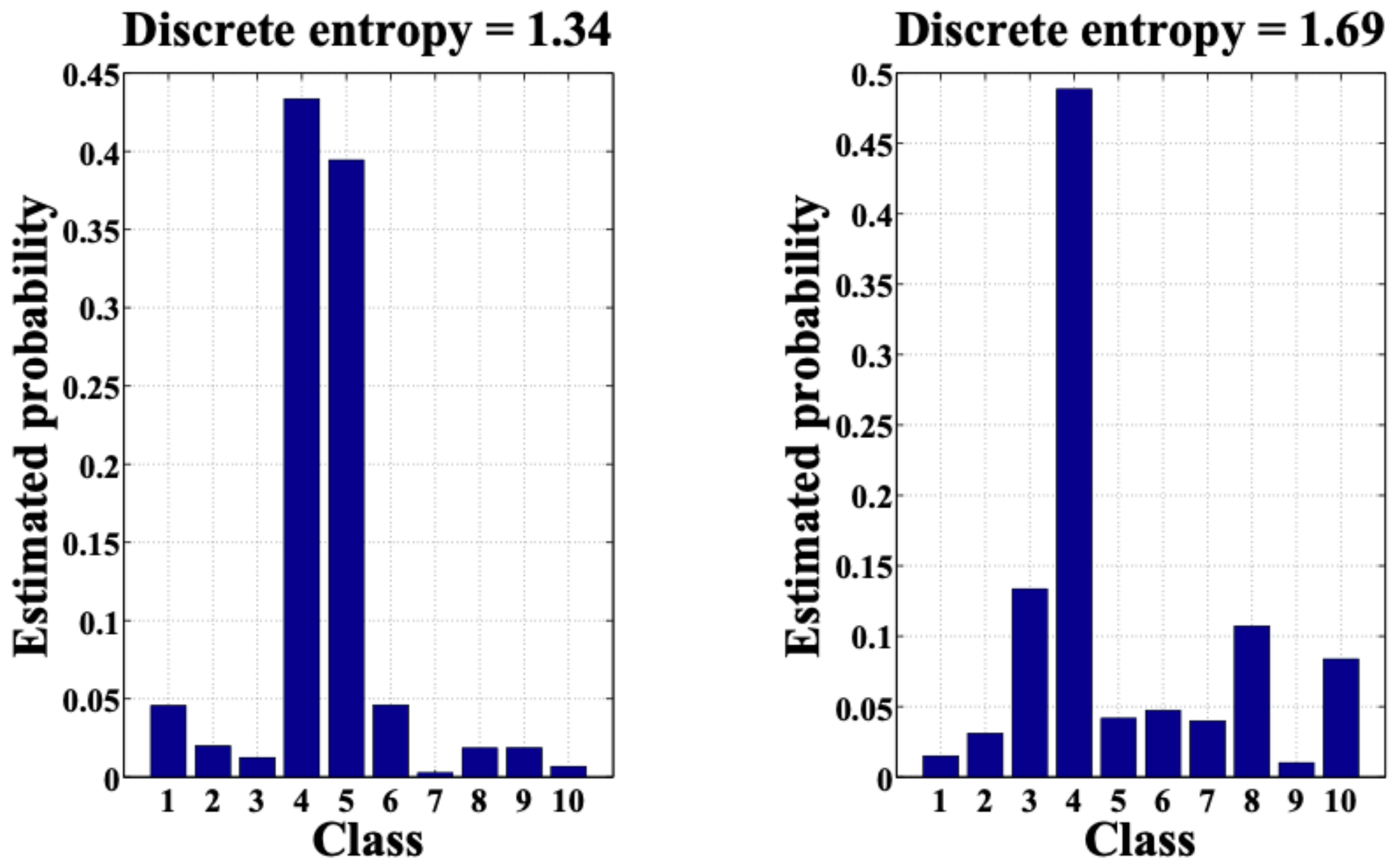
Example p(y|x):

- H[1.0, 0.0, 0.0, 0.0, 0.0] = 0
- H[0.2, 0.2, 0.2, 0.2, 0.2] = 1

See McKay, “Information-Based Objective Functions for Active Data Selection”, Neural Computation 4, 1992 based on prior works by Shannon 1948

Best versus second best

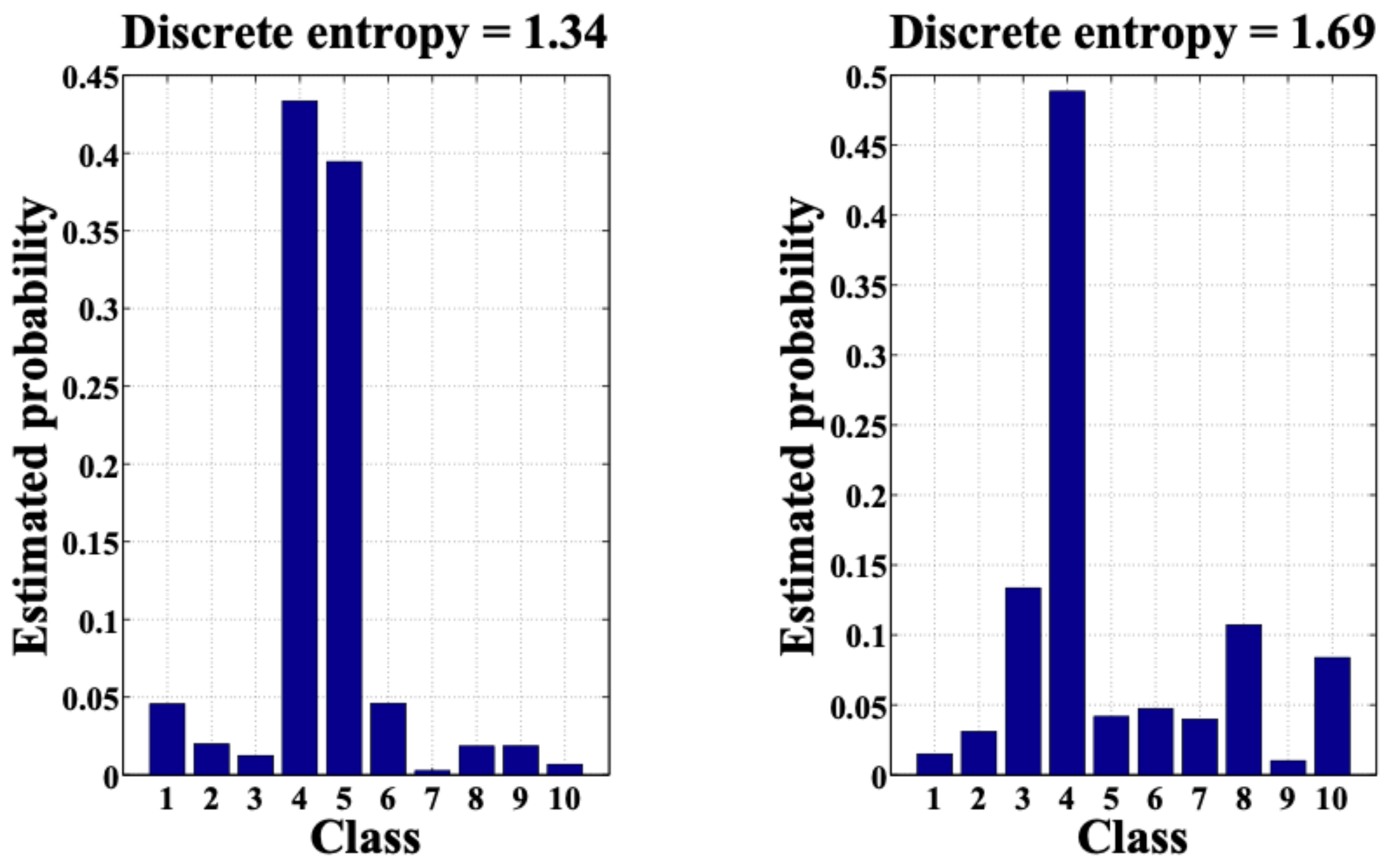
Confidence & entropy can be poor estimates when multiple classes are considered

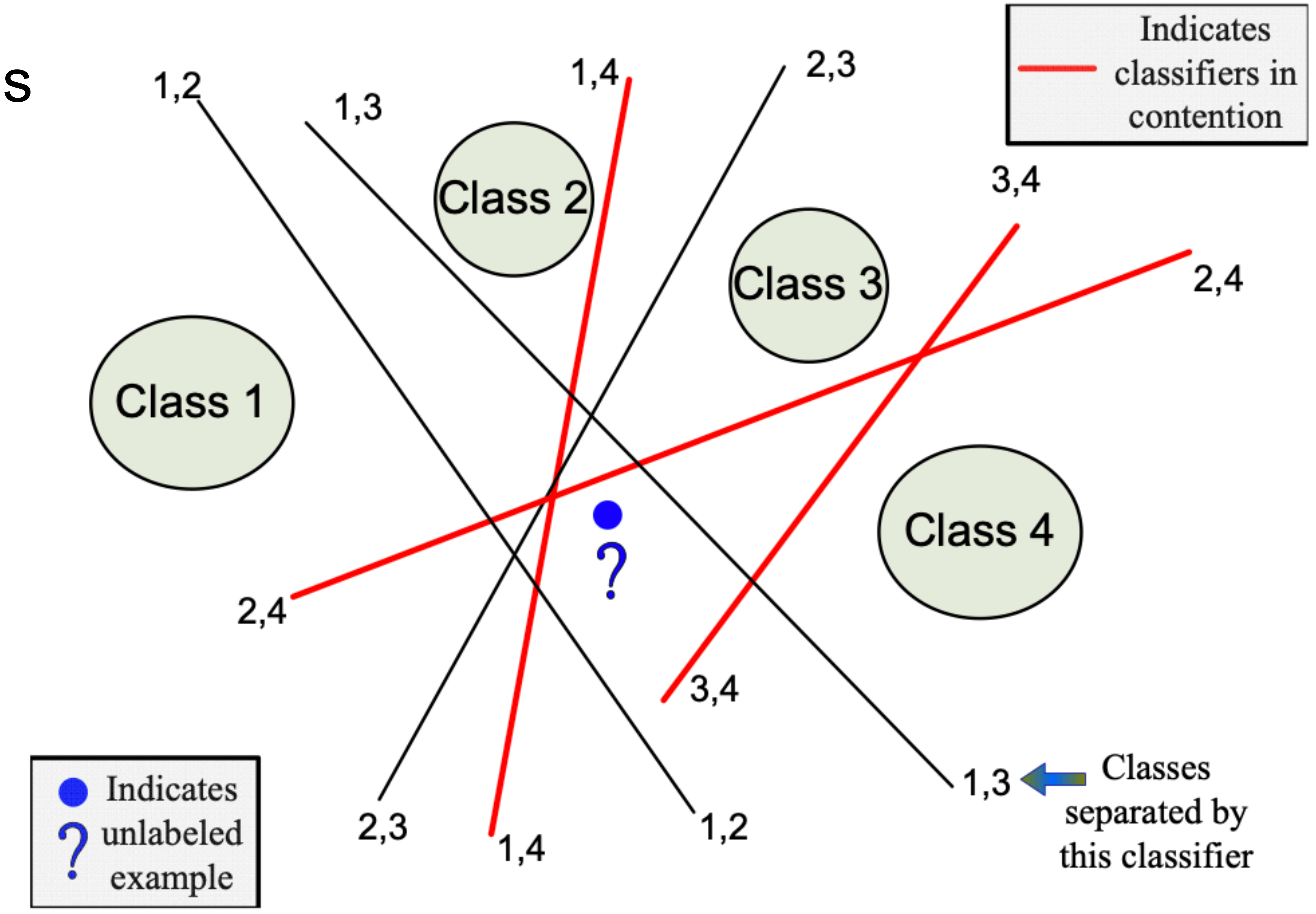


Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Best versus second best

Confidence & entropy can be poor estimates when multiple classes are considered



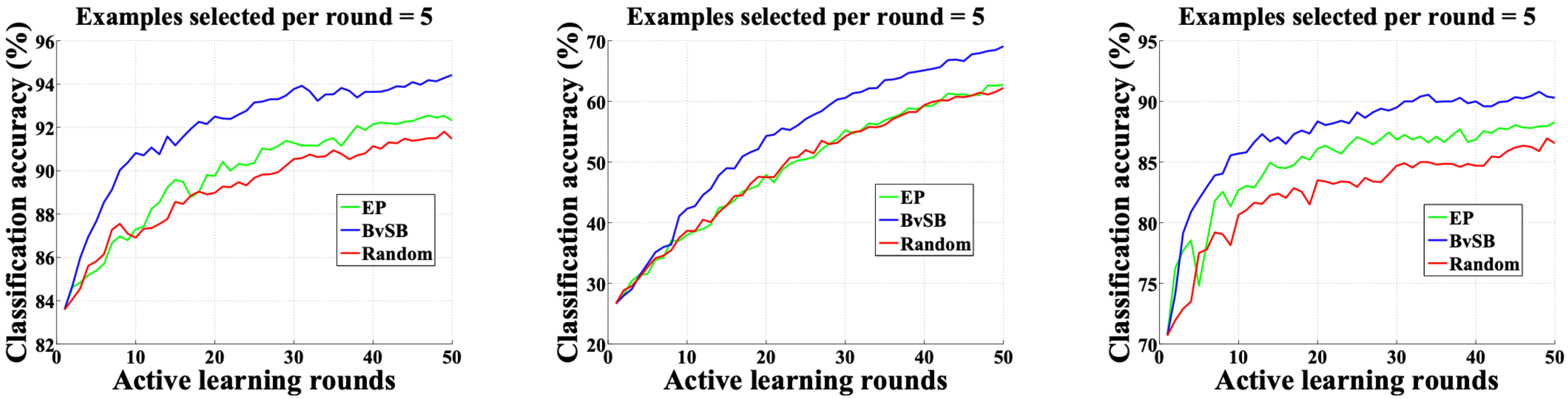


Indicates classifiers in contention
1,2
1,3
1,4
2,3
3,4
2,4
Class 1
Class 2
Class 3
Class 4
2,4
3,4
2,3
1,4
1,2
1,3
Classes separated by this classifier
Indicates unlabeled example

Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Best versus second best

Left to right: Pendigits, Letter, USPS datasets



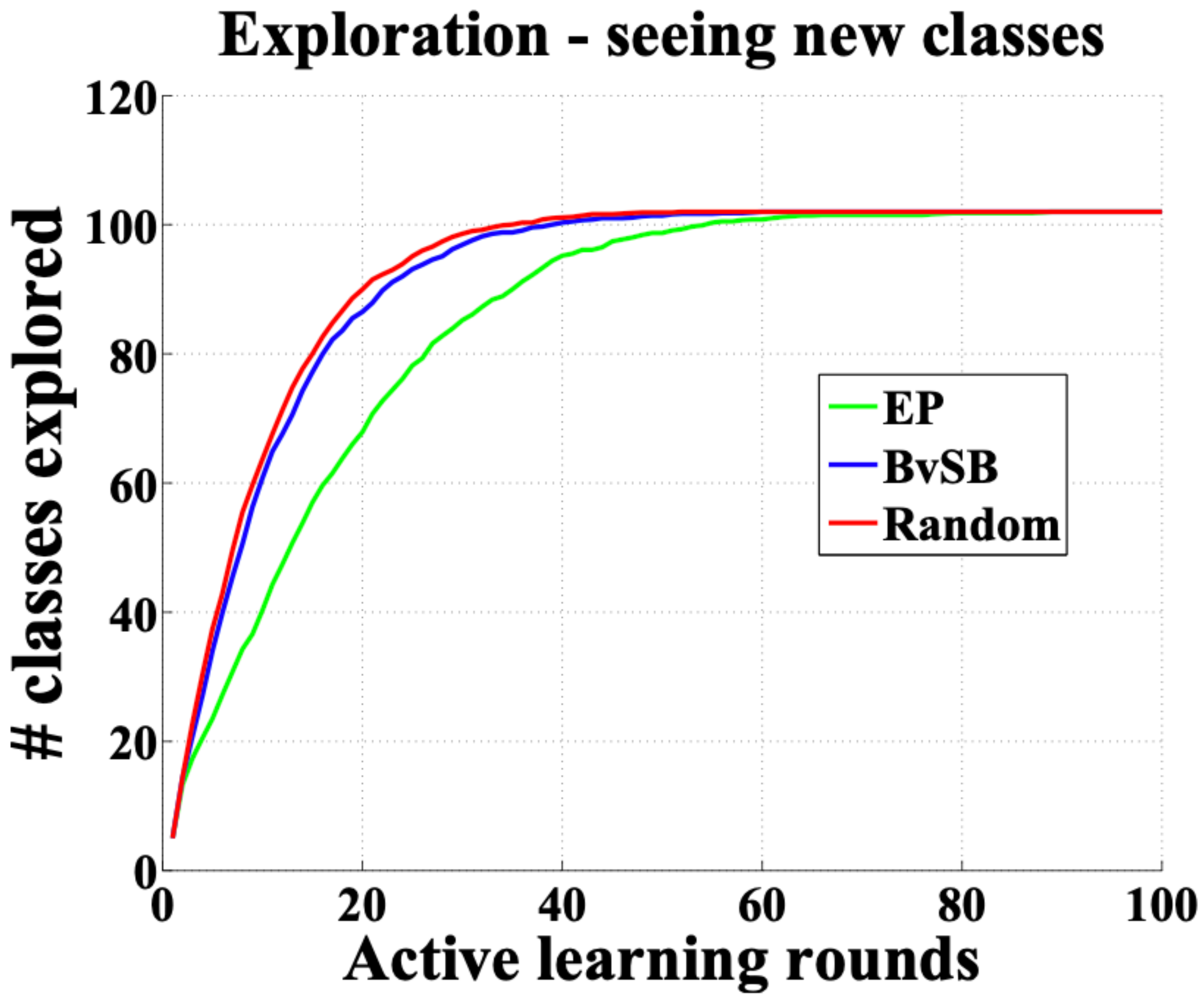
Joshi et al, “Multi-Class Active Learning for Image Classification”, CVPR 2009

Exploration vs. exploitation?

When the task isn't binary classification, we also need to care about **exploration** versus **exploitation**

How much do we explore very novel classes and how much do we extend knowledge of classes we have already seen?

Our measures often overemphasize "**novelty**"



Joshi et al, "Multi-Class Active Learning for Image Classification", CVPR 2009

Can we correct entropy alone?

We could weigh entropy with some measure of data similarity, to get "**information density**":

(Settles & Craven, An Analysis of Active Learning Strategies for Sequence Labeling Tasks, EMNLP 2008)

$$ID(x) = -\sum_{\hat{y}} p(\hat{y}\,|\,x;\theta)\,\log p(\hat{y}\,|\,x;\theta) \cdot \frac{1}{U}\left[\sum_{u} \mathtt{sim}(x, x^{(u)})\right]^{\beta}$$

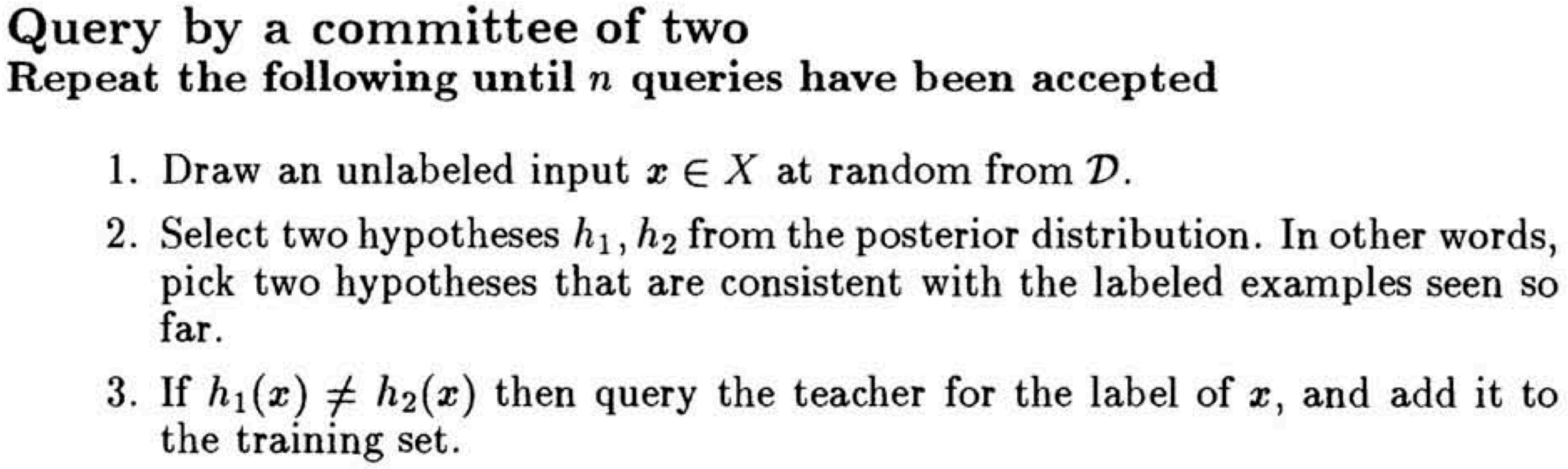
Where beta is a weighting & the similarity over all unlabelled examples U could be a distance:

$$\mathtt{sim}_{cos}(x, x^{(u)}) = \frac{\overrightarrow{x} \cdot \overrightarrow{x}^{(u)}}{||\,\overrightarrow{x}\,|| \times ||\,\overrightarrow{x}^{(u)}\,||}$$

Query by committee

We could also maximize the information gain between two/multiple models: **ensembles**

Could also be interpreted as reducing the version space across models or gauging uncertainty



Query by a committee of two

Repeat the following until n queries have been accepted

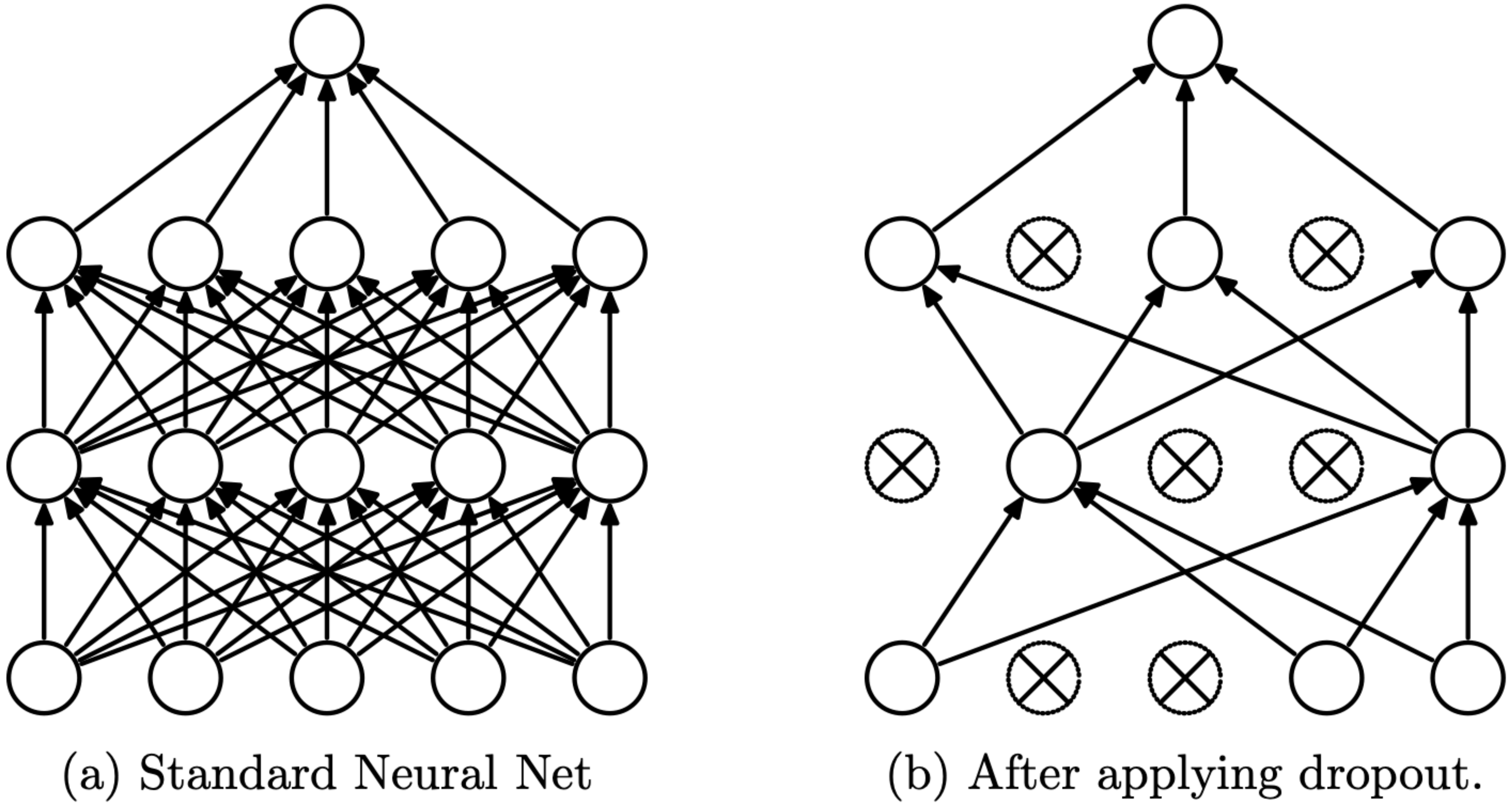
1. Draw an unlabeled input $x \in X$ at random from $\mathcal{D}$.
2. Select two hypotheses h_1, h_2 from the posterior distribution. In other words, pick two hypotheses that are consistent with the labeled examples seen so far.
3. If $h_1(x) \neq h_2(x)$ then query the teacher for the label of x, and add it to the training set.

Seung et al, “Query by Committee”, COLT 1992, and Freund, Seung et al, “Information, Prediction, and Query by Committee”, NeurIPS 1992

Monte Carlo Dropout

Monte Carlo Dropout (Gal et al, “Dropout as a Bayesian Approximation”, ICML 2016)

- Make use of dropout: randomly turning off units in a model
- Bayesian interpretation: Bernoulli distribution on the parameters
- Do stochastic forward passes to assess variation in predictions (model uncertainty)

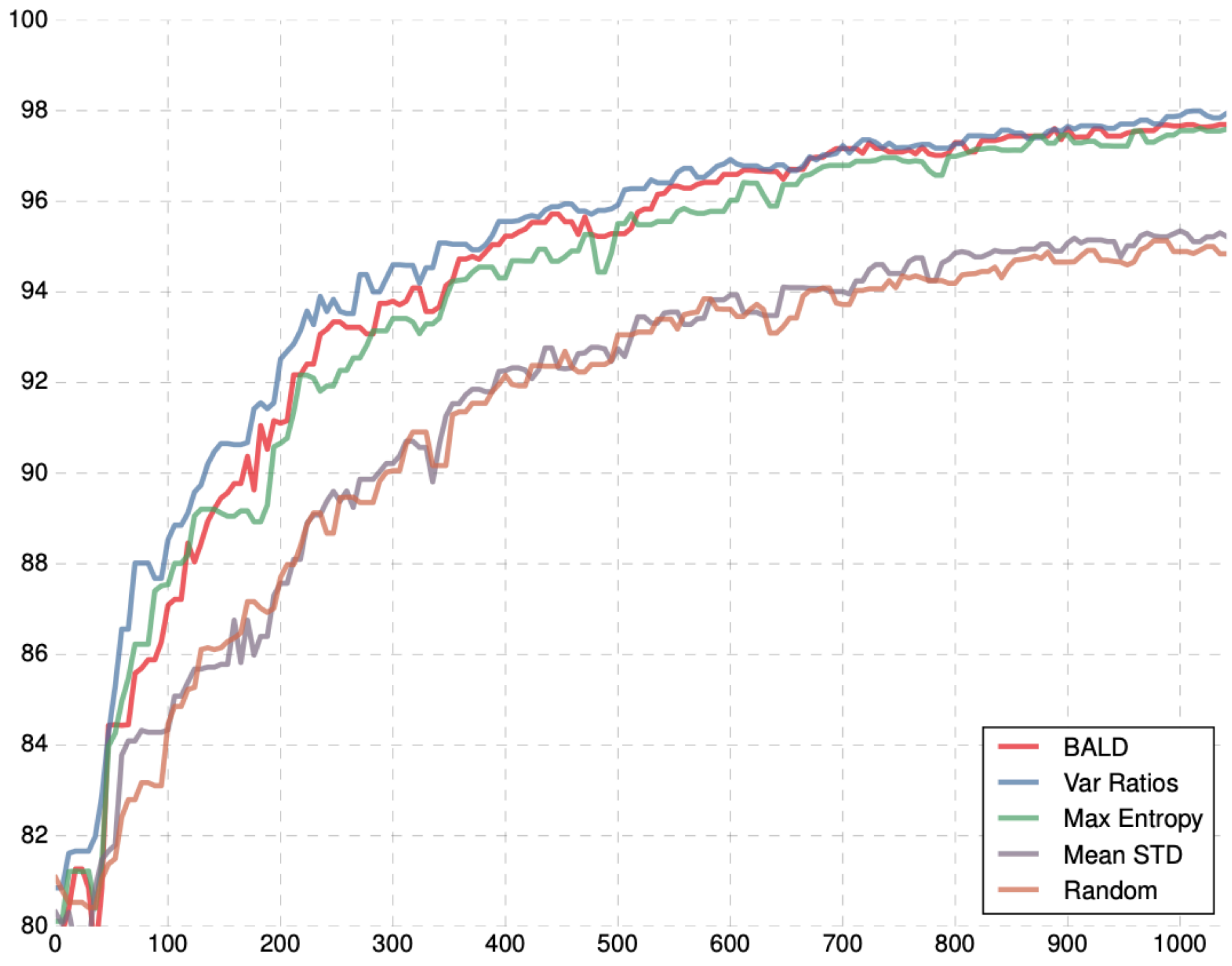


Srivastava et al, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”, JMLR 15, 2014

Monte Carlo Dropout

MCD could be useful as an approximation to using multiple model based ensembles

The acquisition function could still be entropy, standard deviation in output confidence etc.



Gal et al, “Deep Bayesian Active Learning with Image Data”, ICML 2017

Limits of uncertainty sampling

Why aren’t these approaches a lot better?

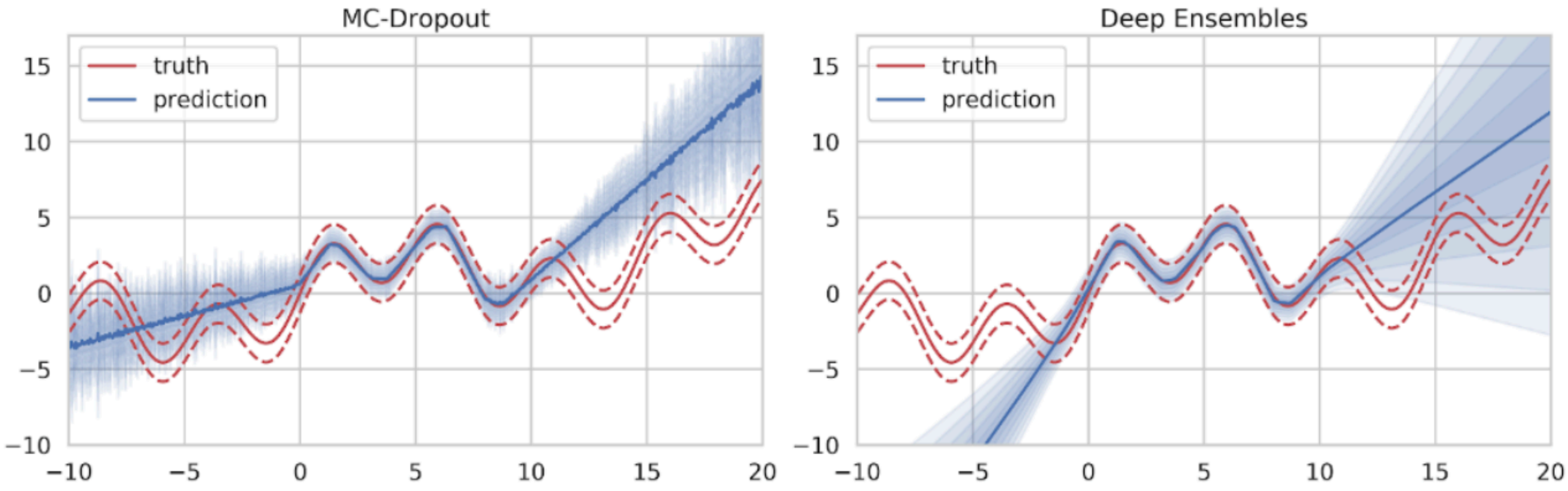


Figure from https://www.inovex.de/de/blog/uncertainty-quantification-deep-learning/

Limits of uncertainty sampling

Why aren’t these approaches a lot better?

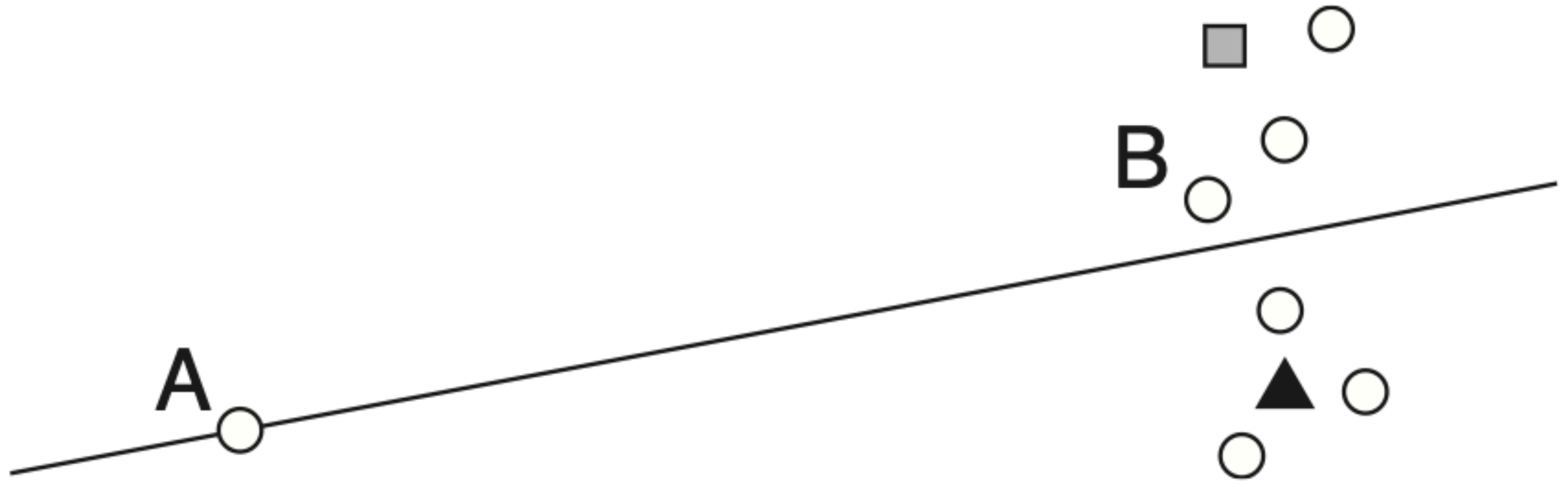


Figure 2: An illustration of when uncertainty sampling can be a poor strategy for classification. Shaded polygons represent labeled instances ($\mathcal{L}$), and circles represent unlabeled instances ($\mathcal{U}$). Since A is on the decision boundary, it will be queried as the most uncertain. However, querying B is likely to result in more information about the data as a whole.

Settles & Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks”, EMNLP 2008

Core Sets & Representation Learning

Representations & core sets

What if we allow to use and even train on the unlabelled pool: “cover the distribution”?

Assumption: a **“teacher” information source is allowed**, like a generative model

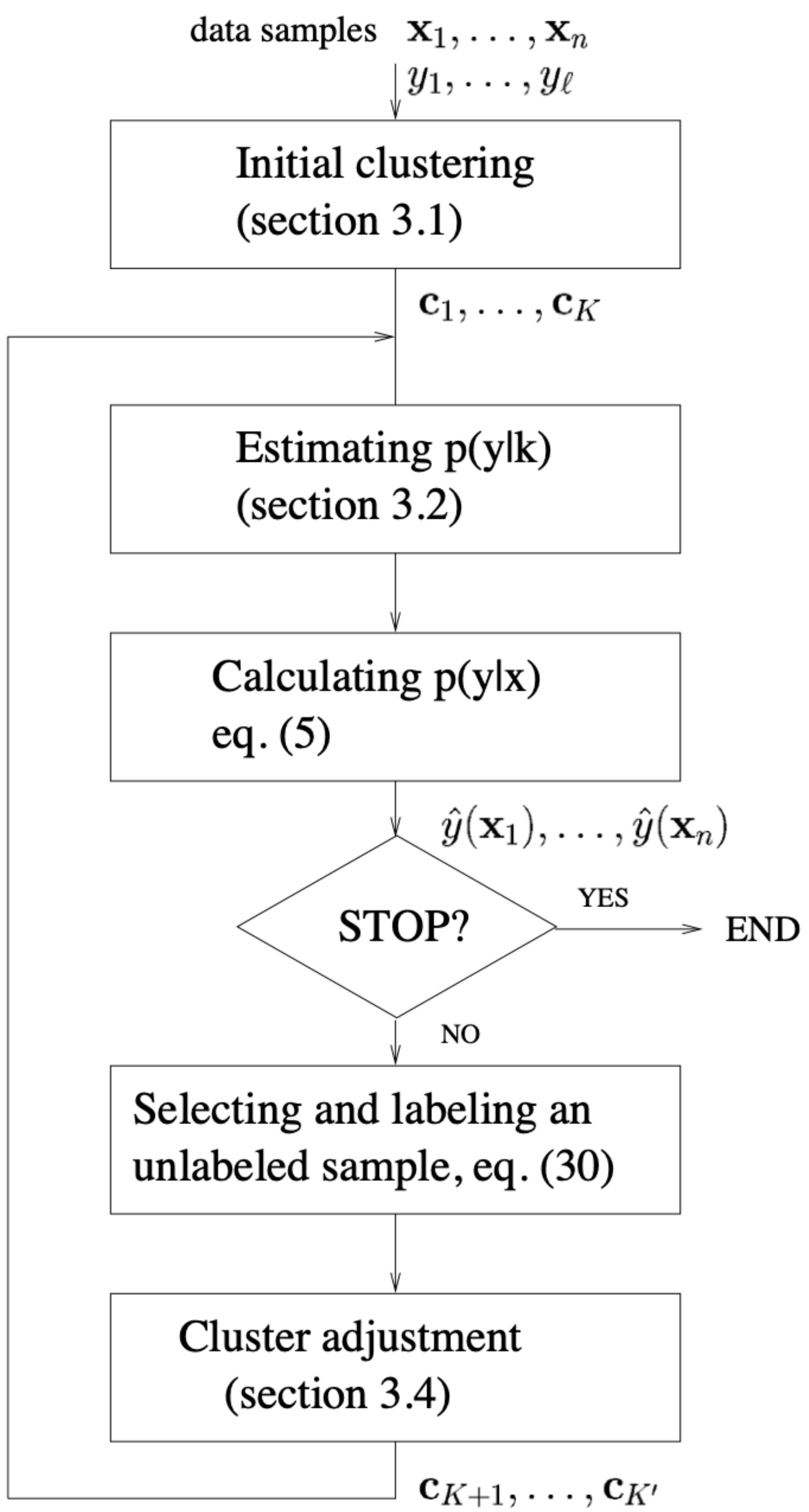
We wouldn’t necessarily get a lot of advantage of generative models in active learning, unless **we also train on the unlabelled pool**: in close relation to semi-supervised learning

We could then also **make use of core sets**, as discussed for rehearsal in the last lecture

Representations & core sets

We could now try to:

- **Pre-cluster** our unlabelled data pool
- Compute **core sets** of the unlabelled data pool
- Learn a **generative model**
  & **representations** on the unlabelled data pool



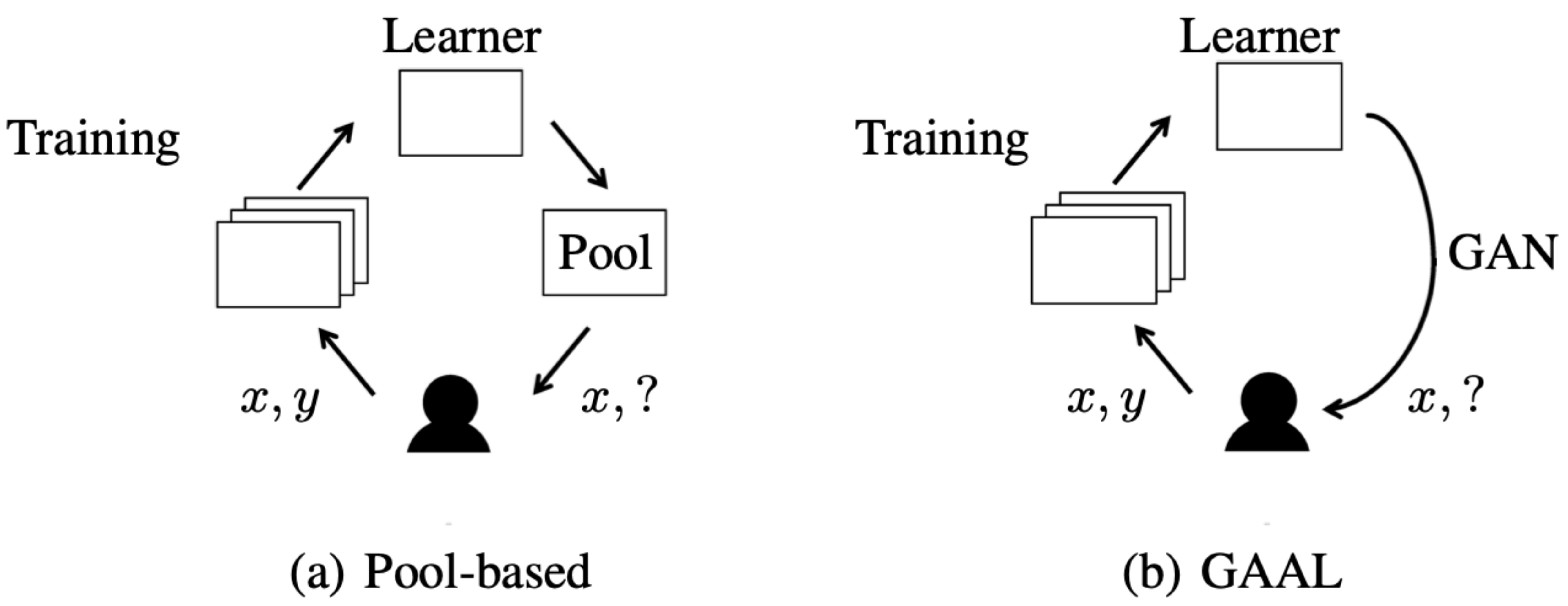
data samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$
$y_1, \ldots, y_\ell$
Initial clustering (section 3.1)
$\mathbf{c}_1, \ldots, \mathbf{c}_K$
Estimating p(y|k) (section 3.2)
Calculating p(y|x) eq. (5)
$\hat{y}(\mathbf{x}_1), \ldots, \hat{y}(\mathbf{x}_n)$
STOP?
YES
END
NO
Selecting and labeling an unlabeled sample, eq. (30)
Cluster adjustment (section 3.4)
$\mathbf{c}_{K+1}, \ldots, \mathbf{c}_{K'}$

H.T. Nguyen et al, “Active Learning Using Pre-clustering”, ICML 2004

Representations & core sets

Example: **generative adversarial active learning**

- As one example of a family of approaches of how to use a generator: “query-synthesizing”
- Let generative model interpolate/ synthesize “novel” data to label + learn actively
- Various follow-ups



(a) Pool-based

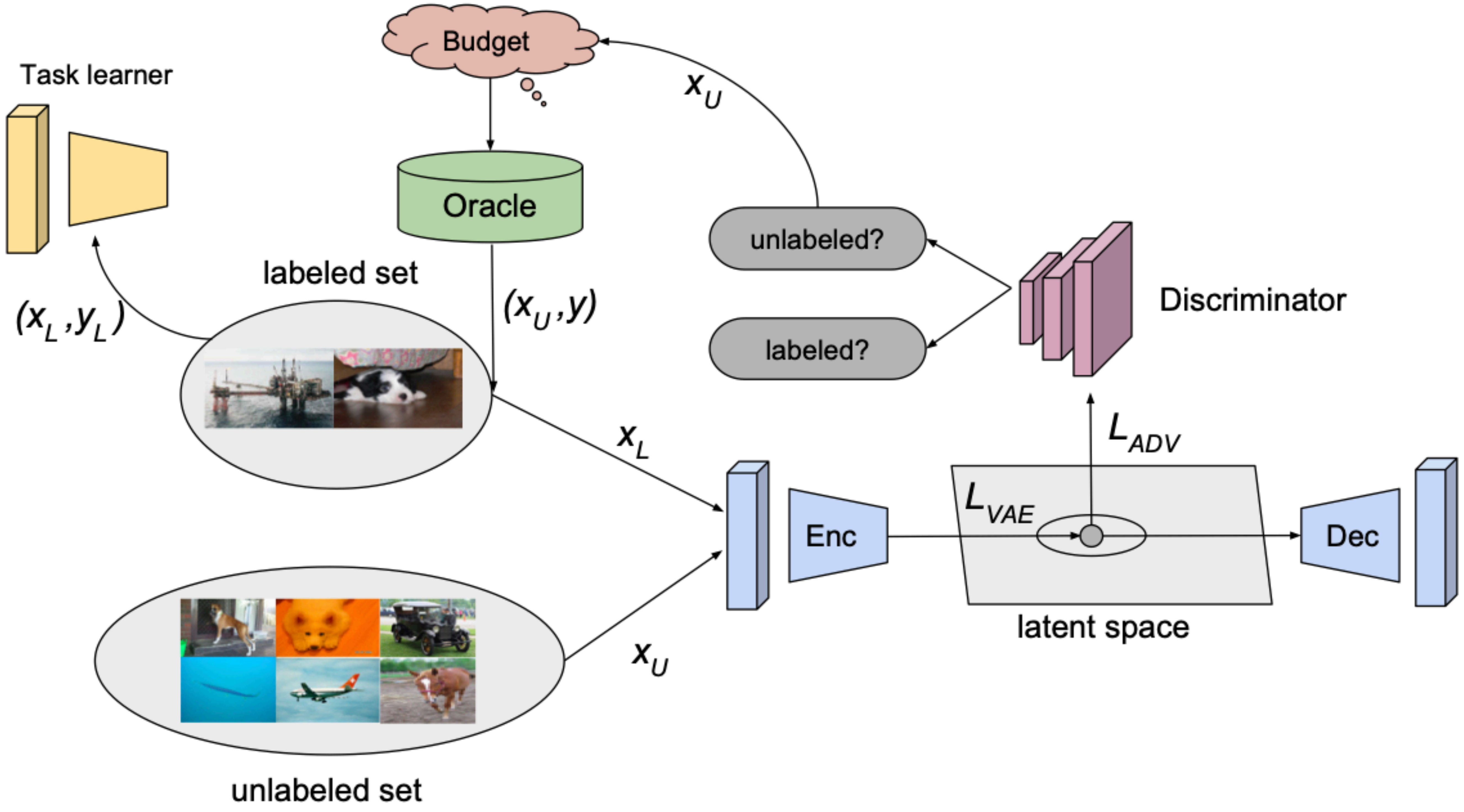
(b) GAAL

Zhu & Bento, “Generative Adversarial Active Learning”, NeurIPS workshop on Teaching Machines, Robots & Humans, 2017

Representations & core sets

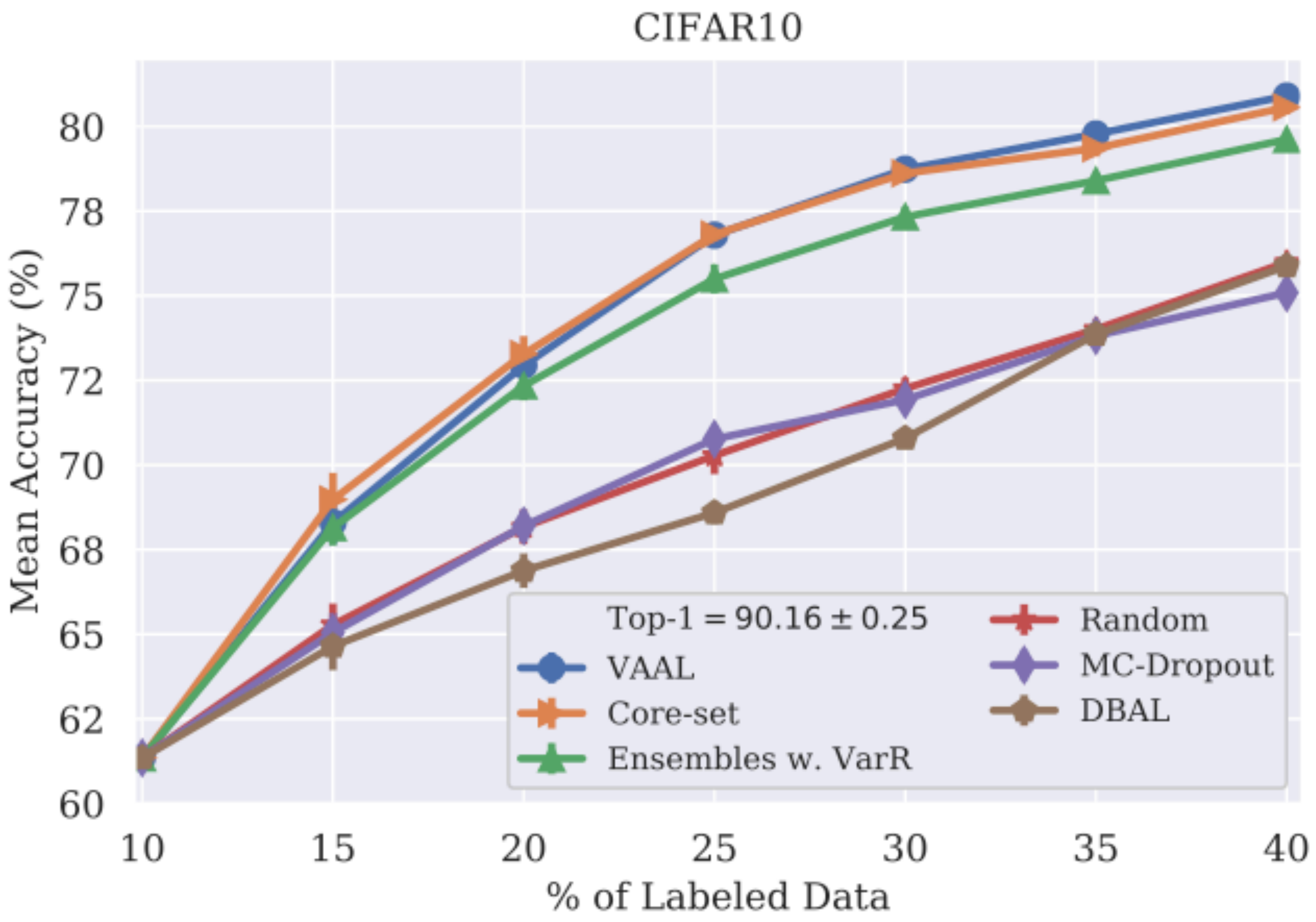
Example: **variational adversarial active learning**

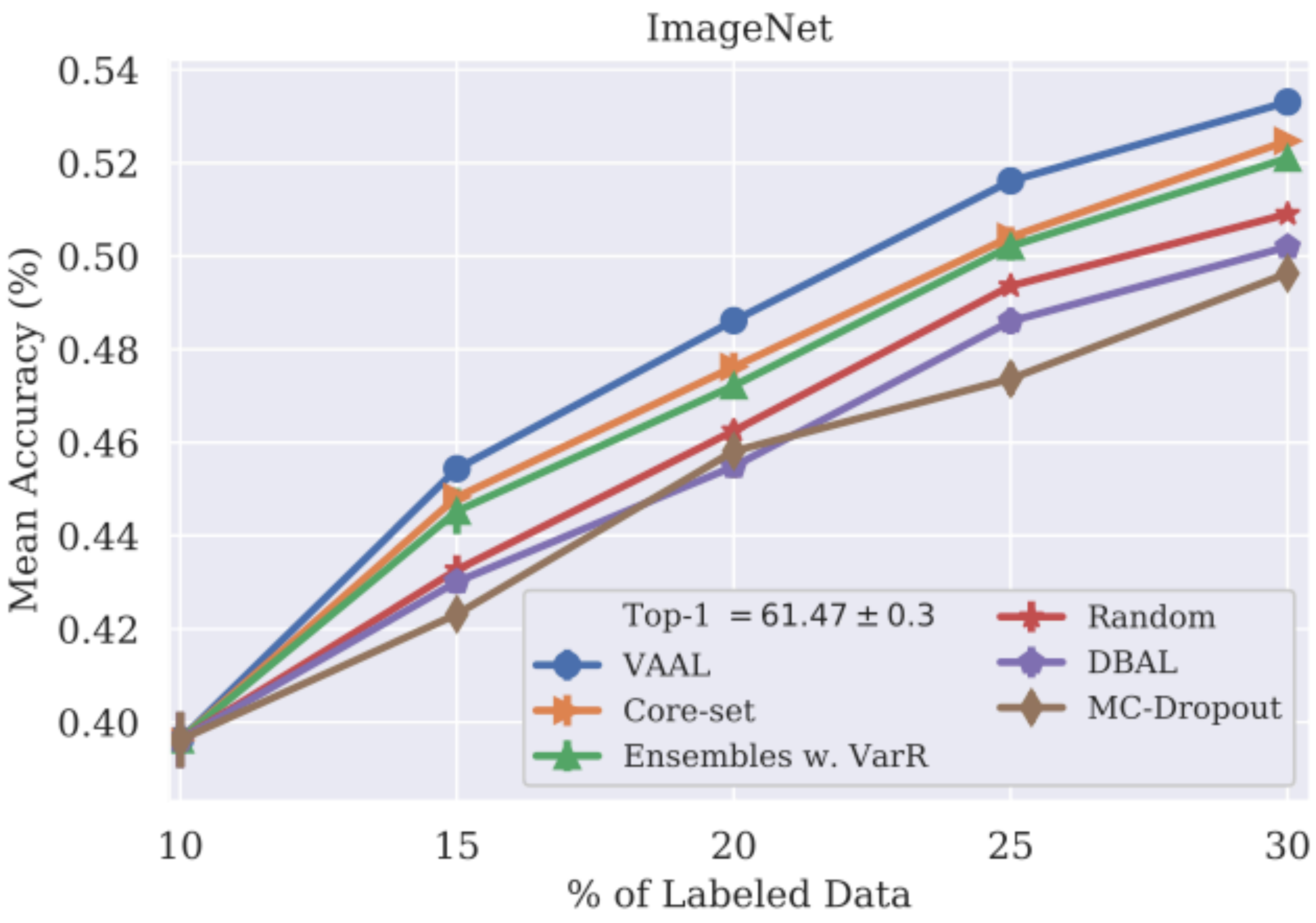
- Optimize on all data
- Learn a discriminator on latent space to distinguish labelled/ unlabelled
- Adversarial: try to fool into believing everything is labelled
- Query according to unlabelled/ labelled confidence



Sinha et al, "Variational Adversarial Active Learning", ICCV 2019

Representations & core sets





Sinha et al, “Variational Adversarial Active Learning”, ICCV 2019

Summary: Let’s keep assumptions & trade-offs in mind

Active learning perspectives

Version space reduction *(Hypotheses)*

The more formal approach: reduce the set/space of possible hypotheses $h : \mathscr{X} \rightarrow \mathscr{Y}$ by removing the ones that are inconsistent with the data

Uncertainty & heuristics *(Novelty)*

The perhaps intuitive approach: use the predictions, or maybe even better, uncertainty in the predictions for the queries

Core sets & representation learning - accessing the entire pool *(Diversity)*

The distribution based approach: maximizing distribution coverage instead of reducing the possible set of hypotheses (version space) explicitly

In summary

Techniques	& (some of) their assumptions
• Version space reduction	• Set of hypotheses is clear
• Minimum confidence • Maximum entropy • Best versus second best	• No overconfidence phenomenon and out-of-distribution/task data
• Model “uncertainty” (output variability) • Ensembles/query by committee	• Accurate uncertainty everywhere • Training of multiple models
• Representation learning on the pool • Core sets	• Upfront training on entire pool (access + computational expense)

More general assumptions

Recall our assumptions:

- *Oracle is infallible:*

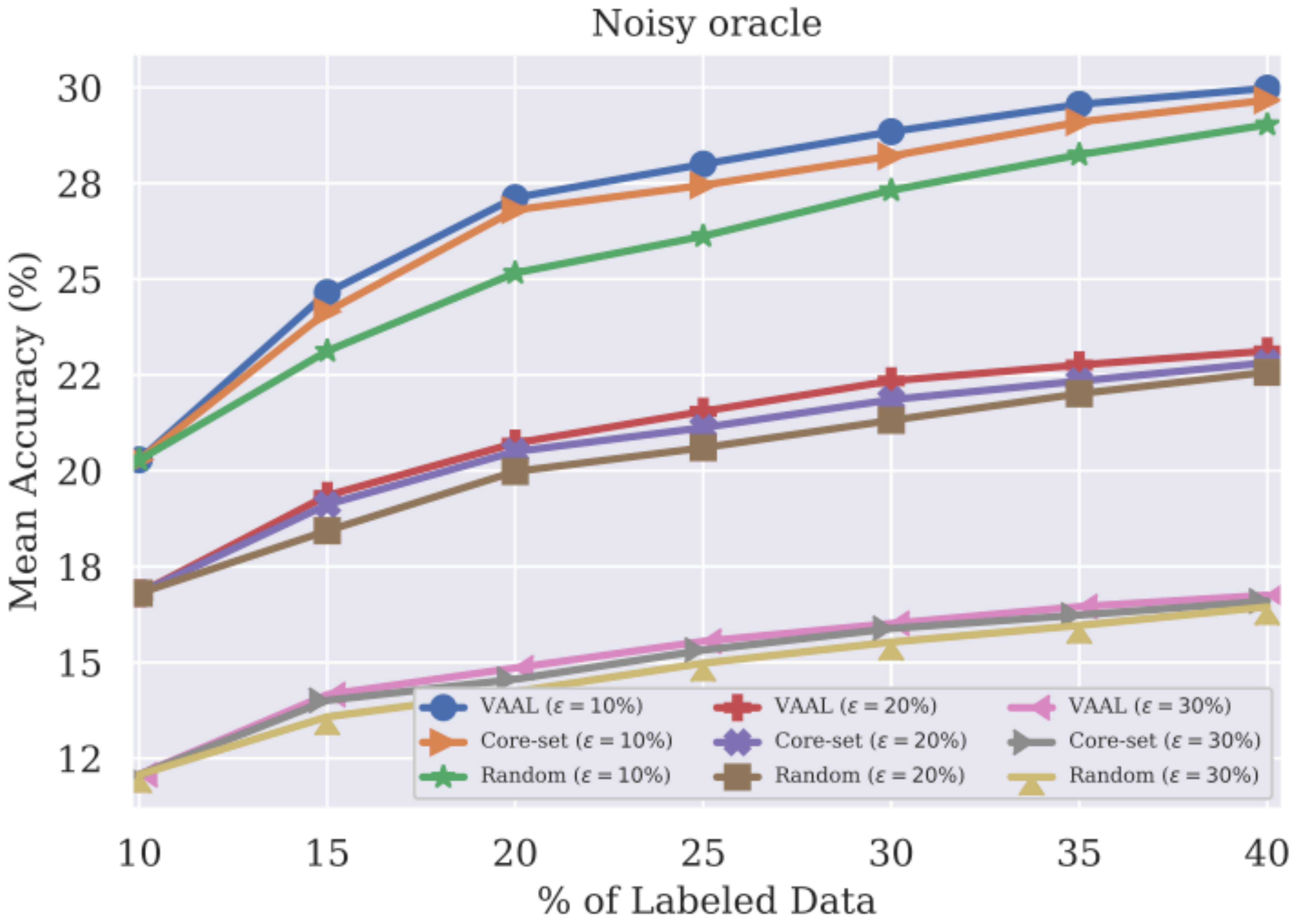
  the teacher/labeler does not make mistakes!

- *Data is accumulated*:

  no "continual active learning"

- *Pool belongs to task*:

  we will cover this in our lecture on "learning and the unknown"



Sinha et al, "Variational Adversarial Active Learning", ICCV 2019