

# Continual Machine Learning

## Summer 2024

### Teacher

Dr. Martin Mundt,  
Research Group on Open World Lifelong Learning

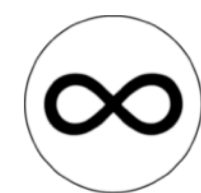
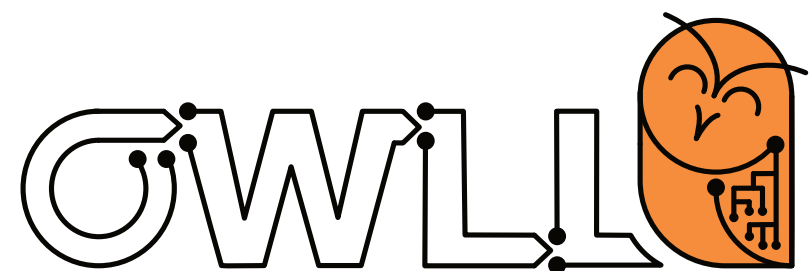
### Time

Every Friday 14:25 - 16:05 CEST

### Course Homepage

[http://owll-lab.com/teaching/cl\\_lecture\\_24](http://owll-lab.com/teaching/cl_lecture_24)

<https://www.youtube.com/playlist?list=PLm6QXeaB-XkA5-IVBB-h7XeYzFzgSh6sk>



Continual **AI**



**hessian.AI**



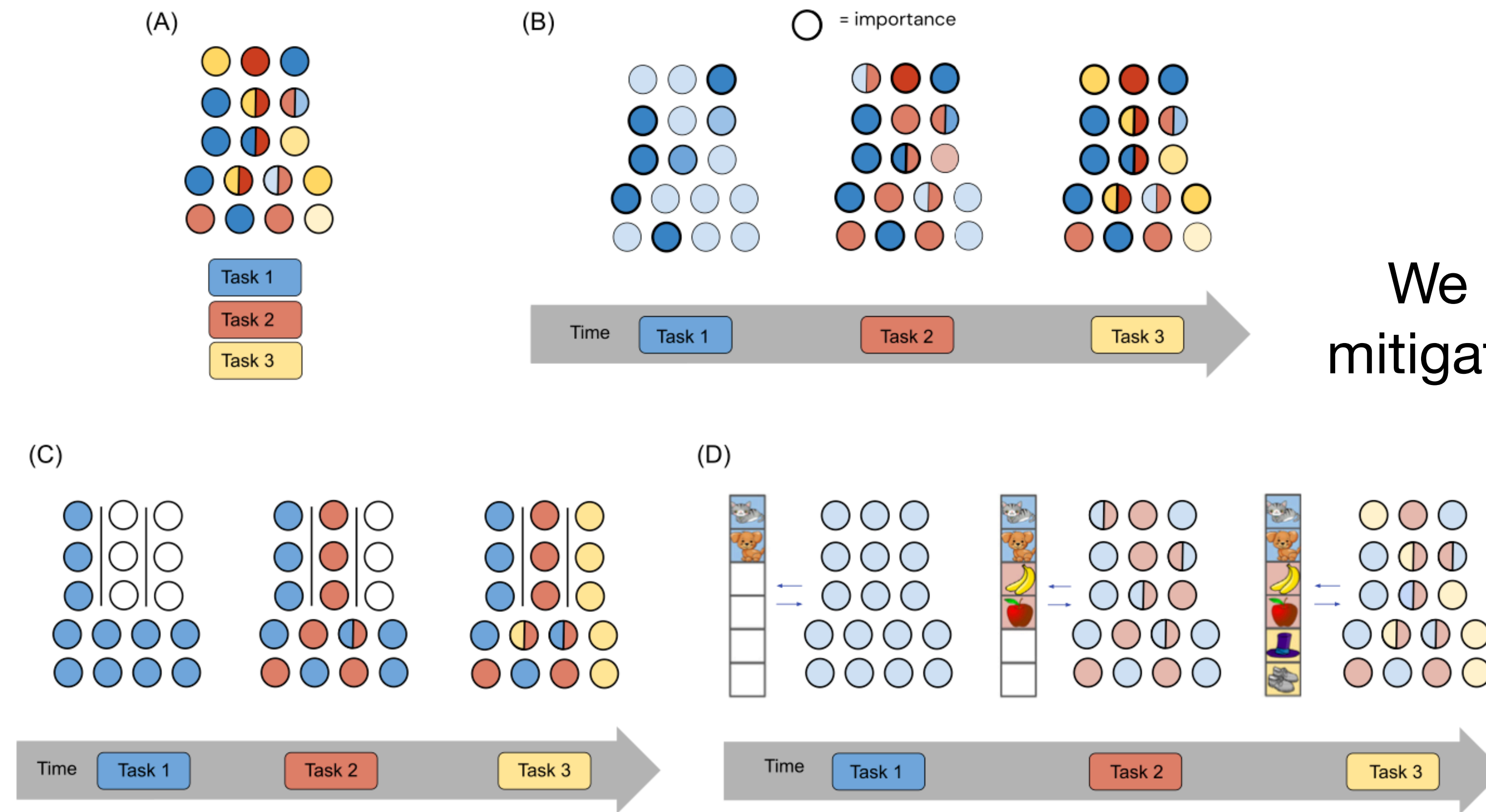
TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

# Recall: How to avoid forgetting?



## Paradigms for Continual Learning

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



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

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.

# Recall: continual experiments



But where does our data sequence actually come from?

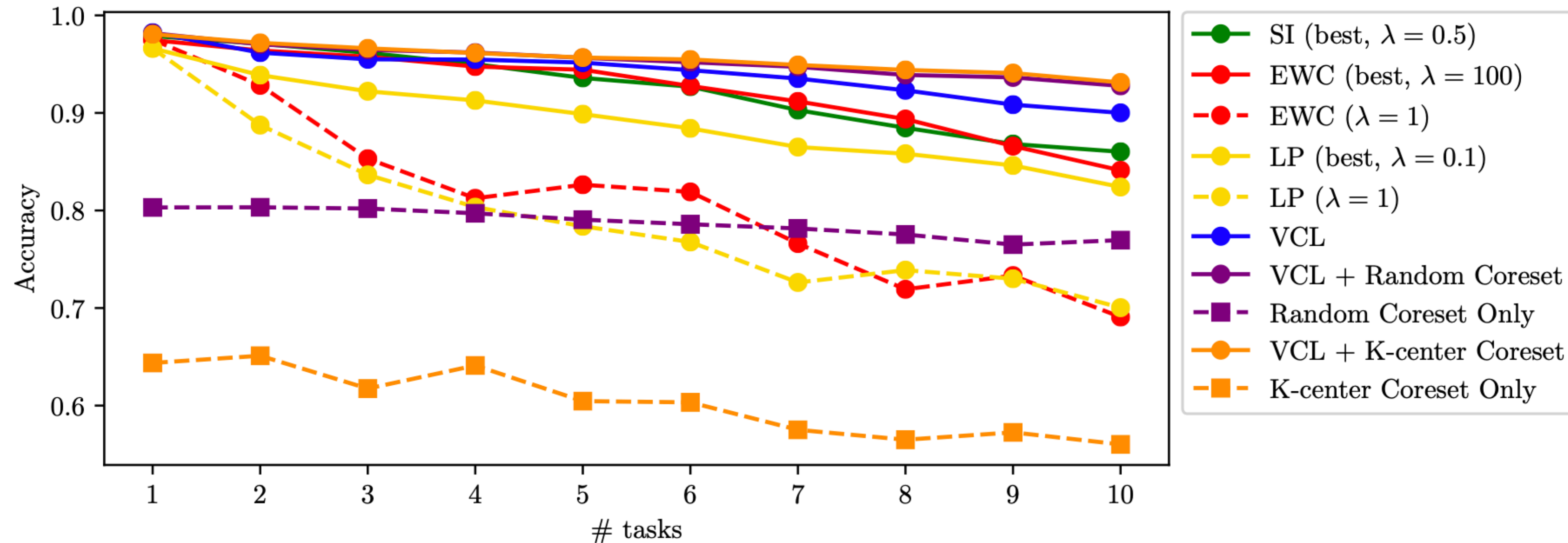
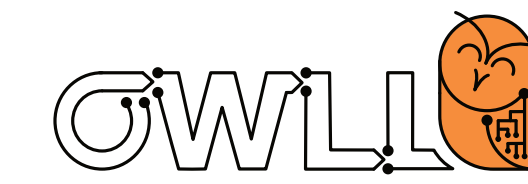


Figure 2: Average test set accuracy on all observed tasks in the Permuted MNIST experiment.



# Week 5: Active Learning

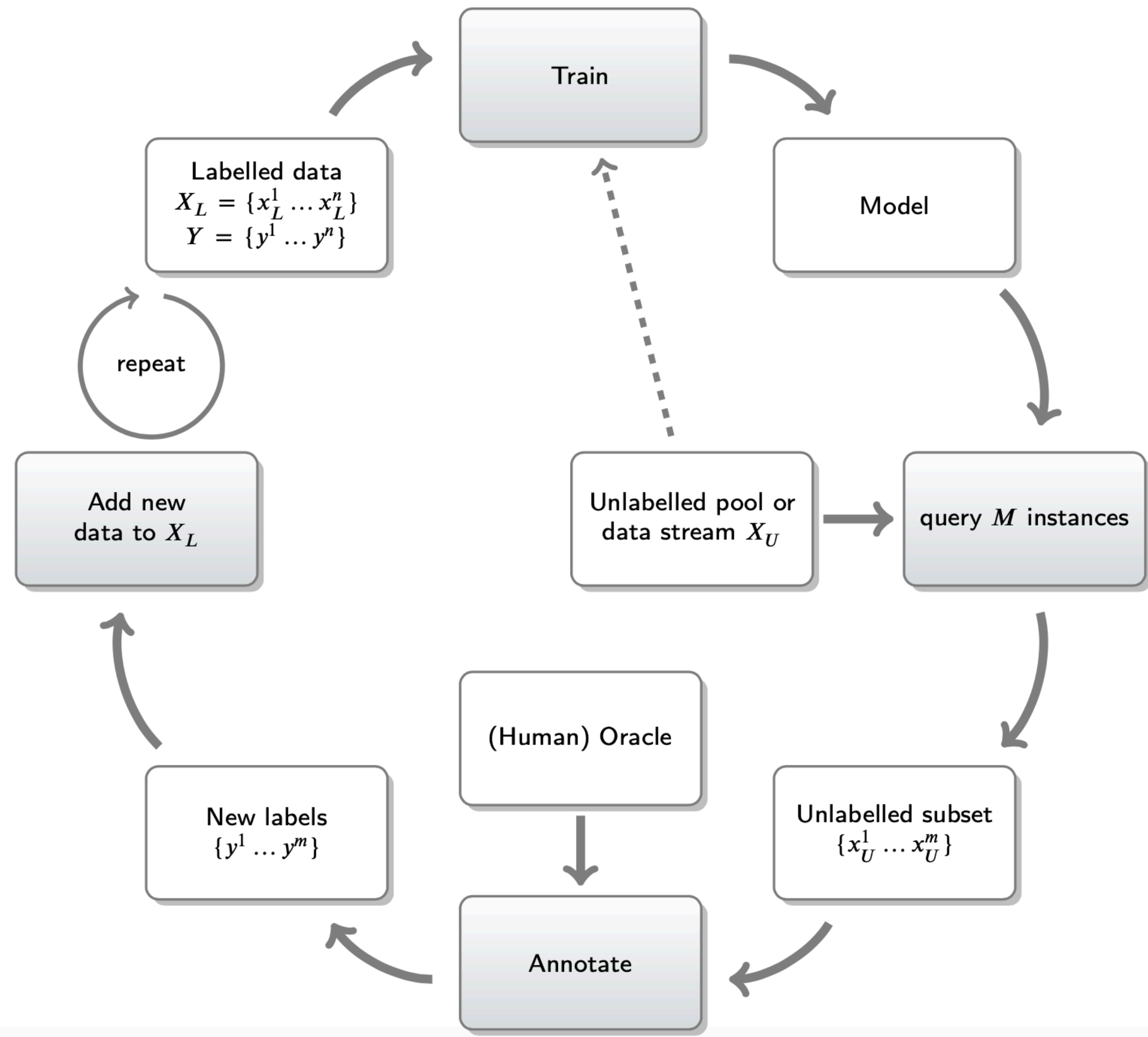
# Active learning



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”

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



**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* ;

*(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

**Algorithm 1:** Pool-based active learning.



**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

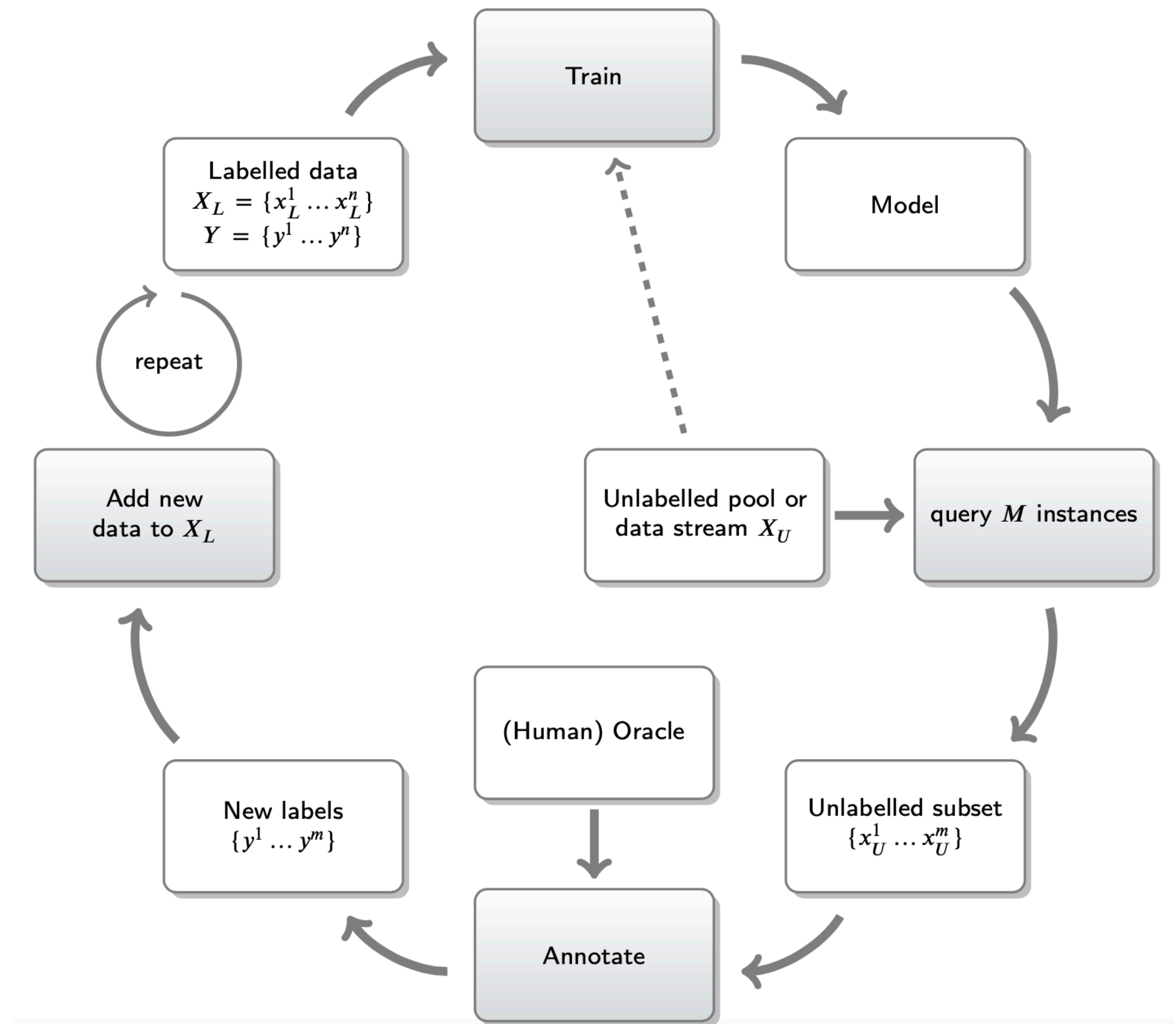


Figure from “A Wholistic View of Deep Neural Networks: Forgotten Lessons and the Bridge to Active and Open World Learning”, Mundt et al 2020



**Acquisition functions: what techniques can you think of?**

# Active learning perspectives



## Version space reduction

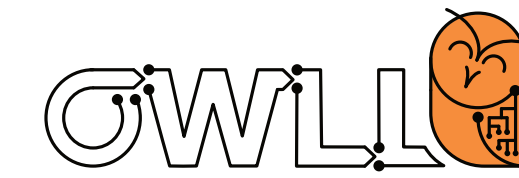
*The more formal approach:* reduce the set/space of possible hypotheses  $h : \mathcal{X} \rightarrow \mathcal{Y}$  by removing the ones that are inconsistent with the data

## 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



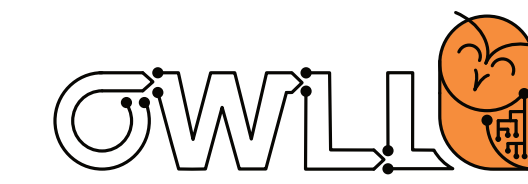
**Discriminative** models could allow for natural ways to assess “novelty” of a new example

-> *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?)

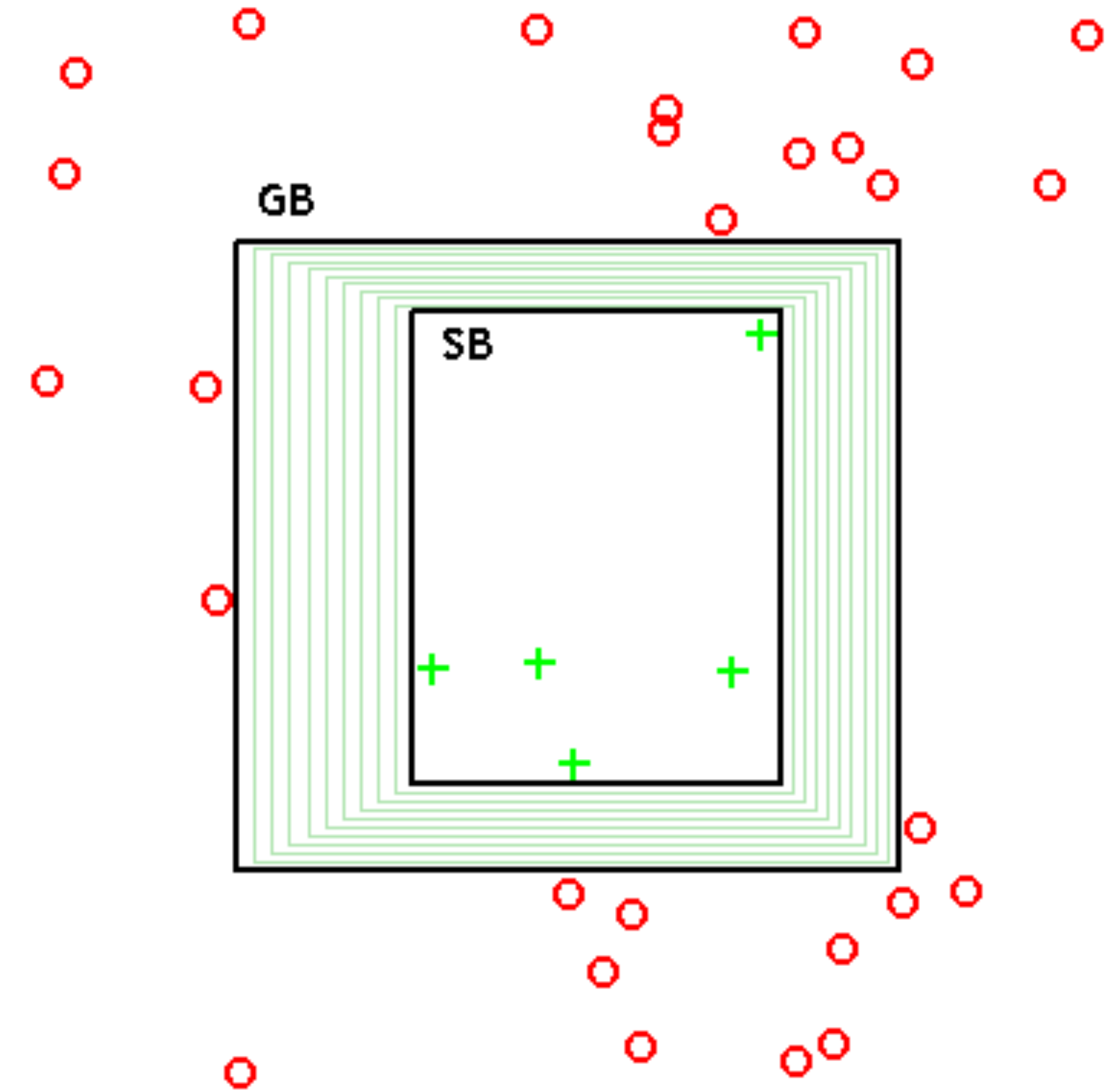
We will see that the choice also heavily depends on the set-up assumption!



# Version Space

# Version space (Mitchel 1978)

- Assume that there exist hypotheses consistent with the labeled data points  $h : \mathcal{X} \rightarrow \mathcal{Y}$   
**version space:**  $VS(D) = \{h \in H \mid \text{cons}(h, D)\}$
- **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

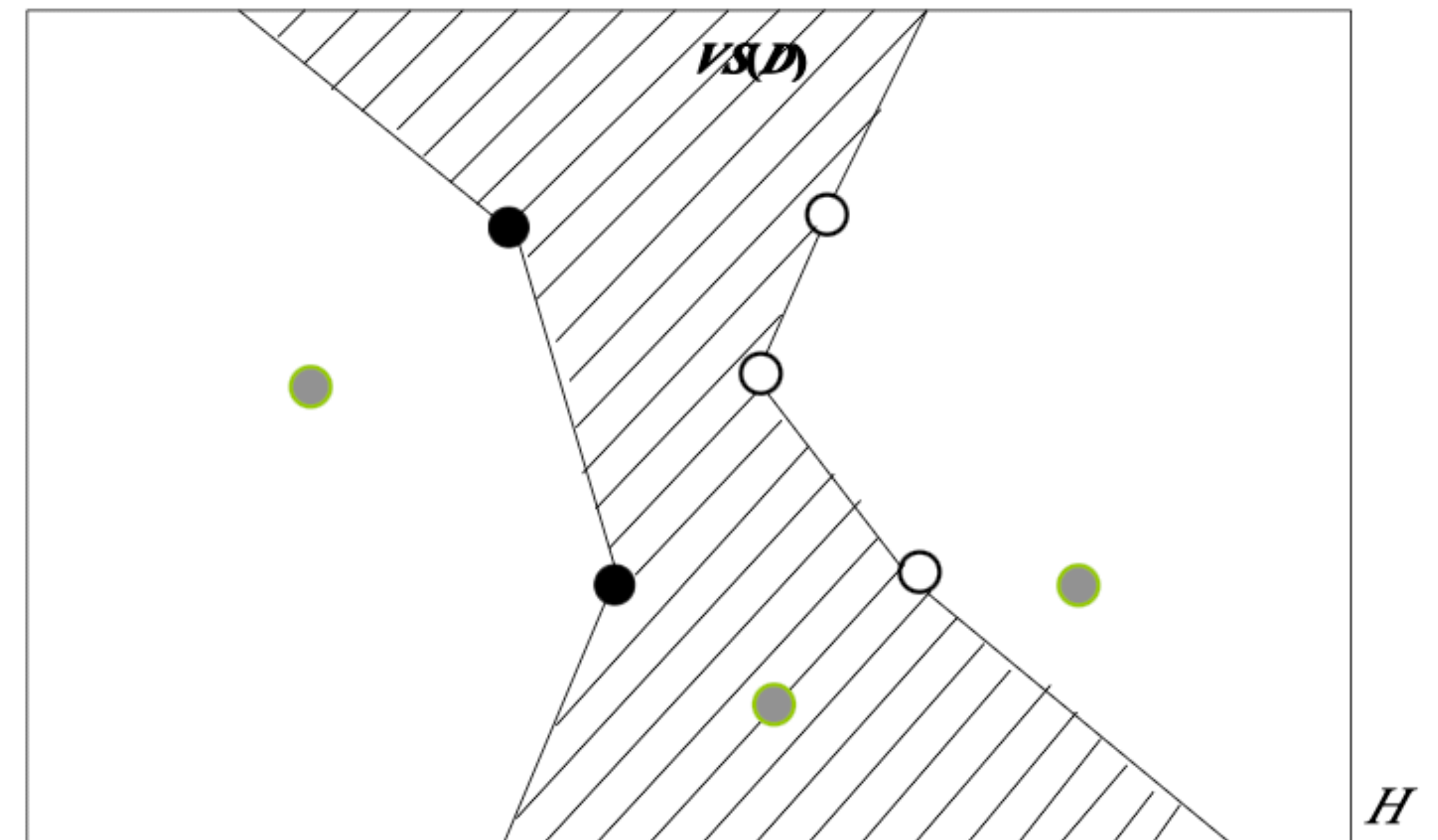


# Version space reduction

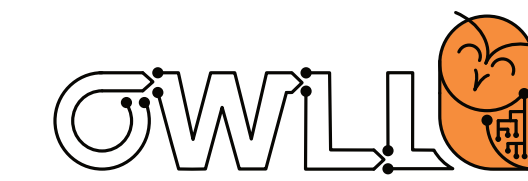
“Generalization as Search”, Mitchell 1982

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

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







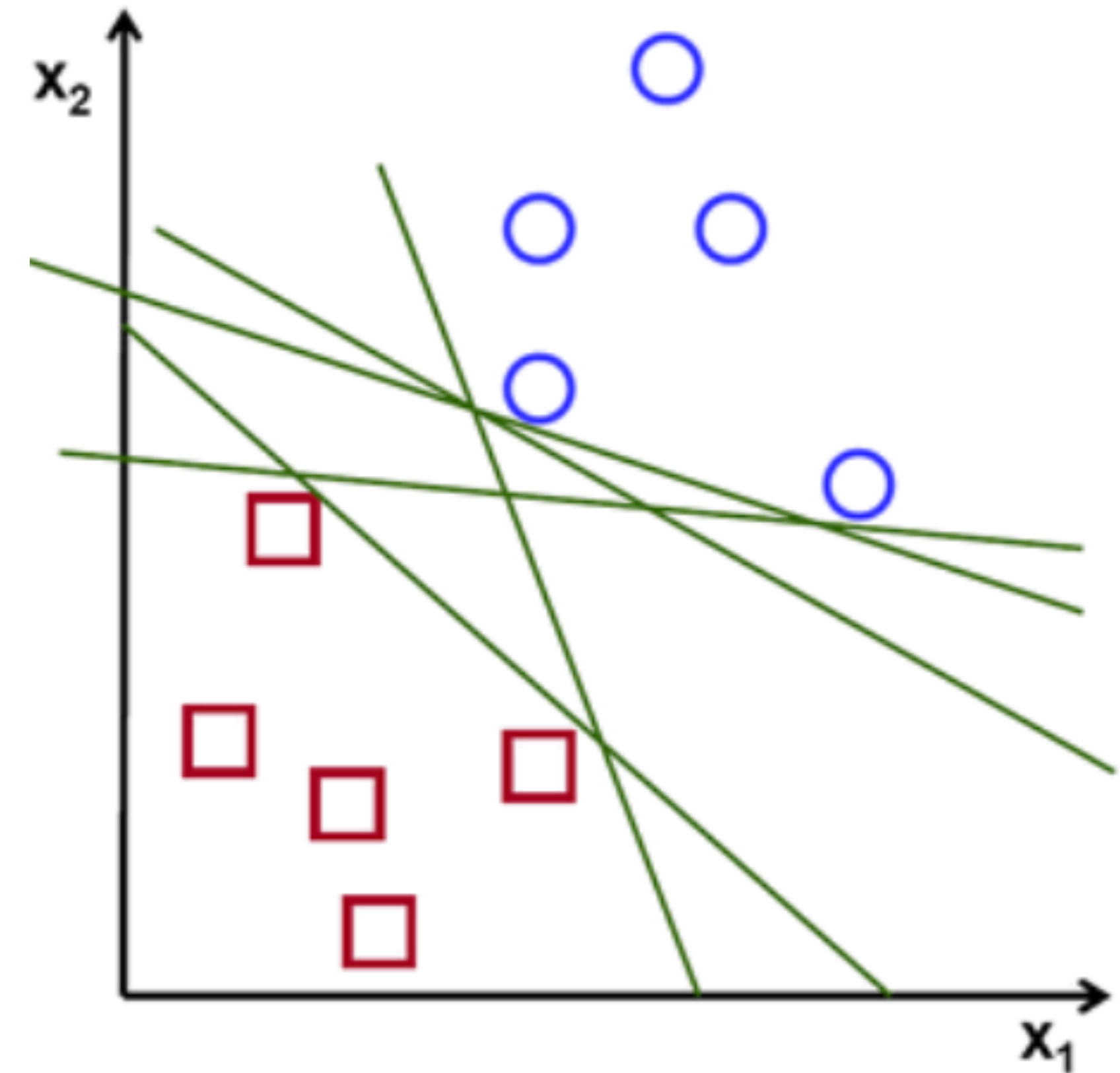
# **A very short excursion: support vector machines**

# 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)



# Support vector machines (SVM)



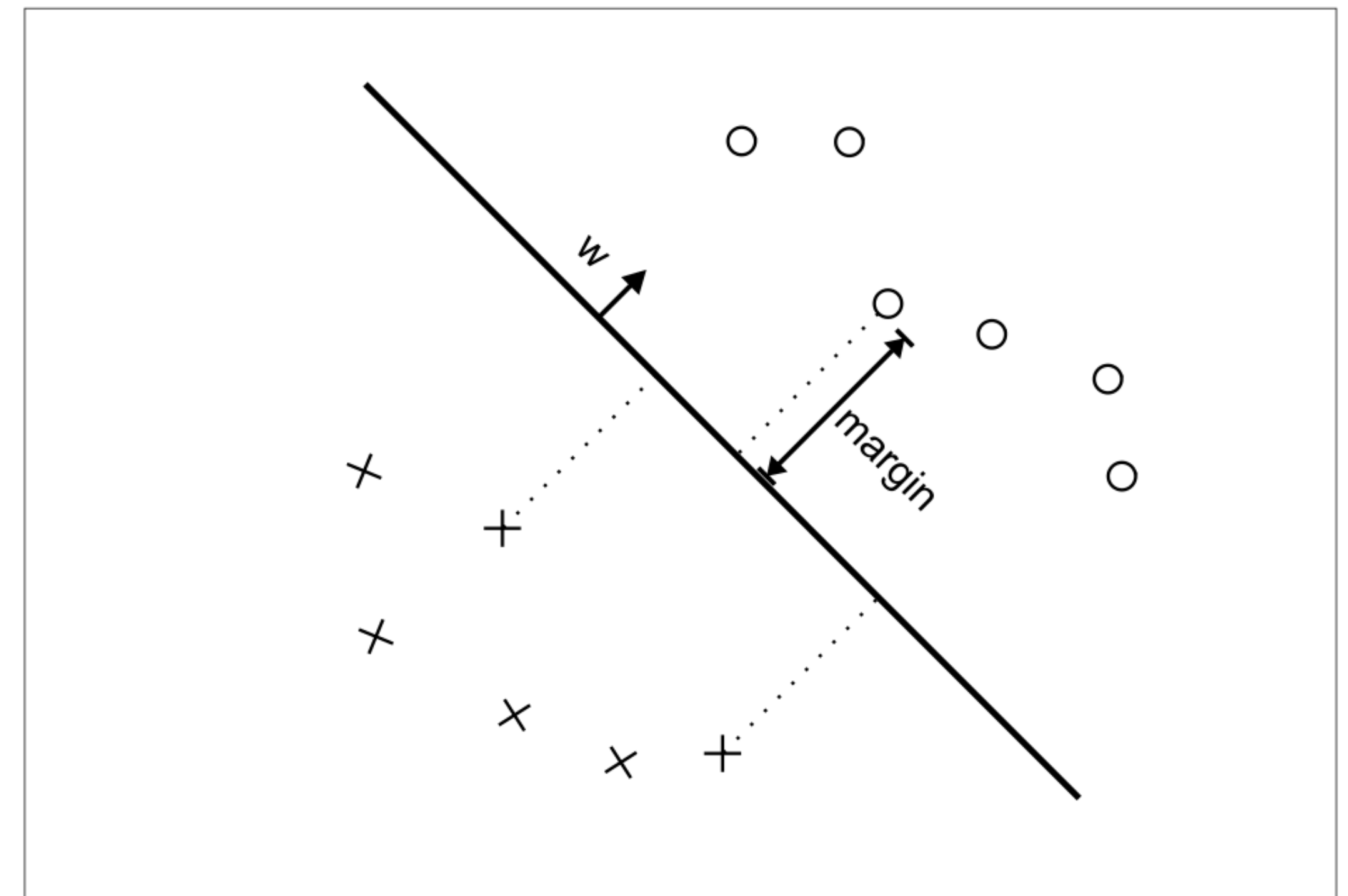
## Example: support vector machine (SVM)

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

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

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

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



# 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:  $y_i(w^T x_i - b) = 0 \geq 1, \forall 1 \leq i \leq n$

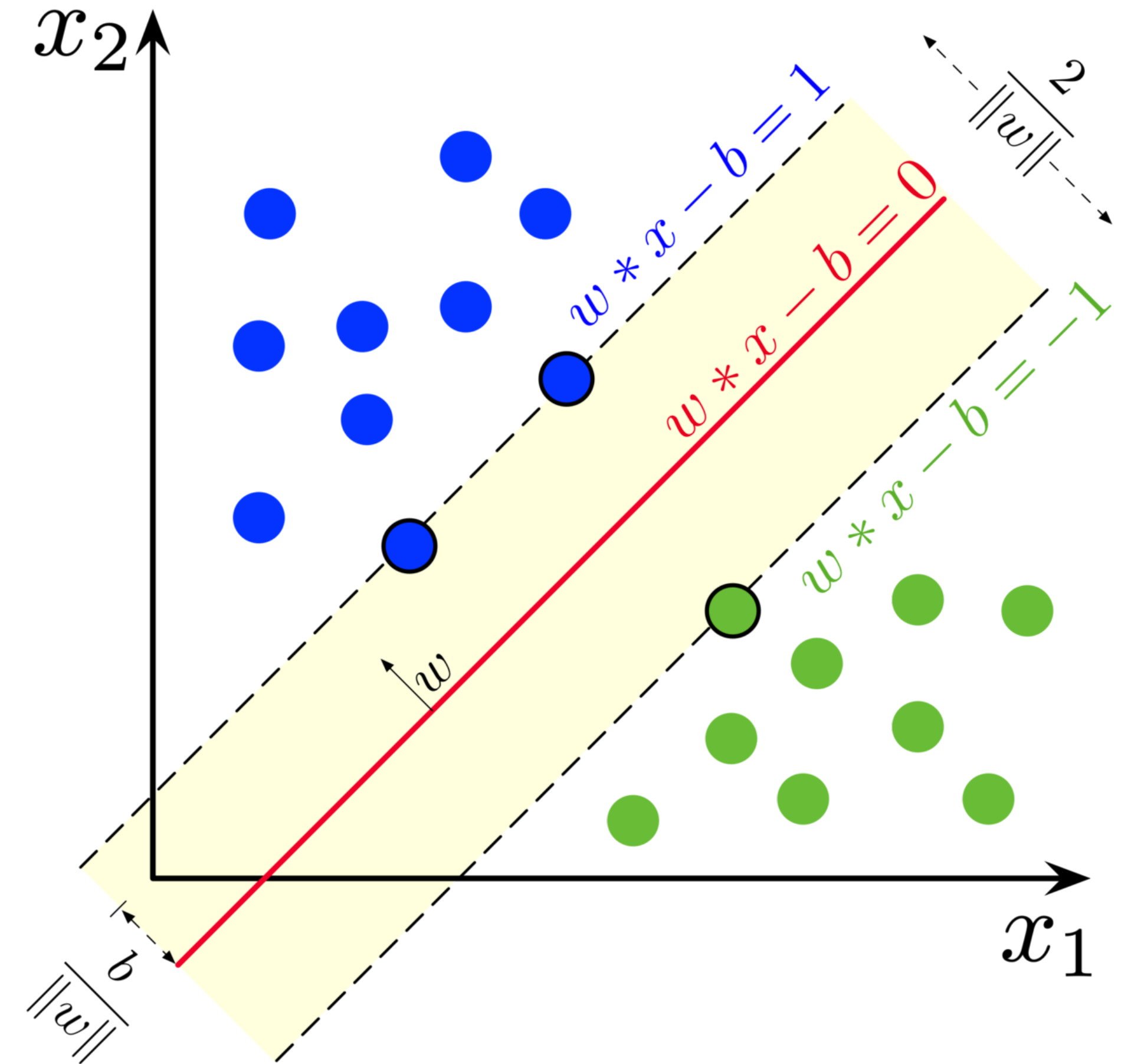


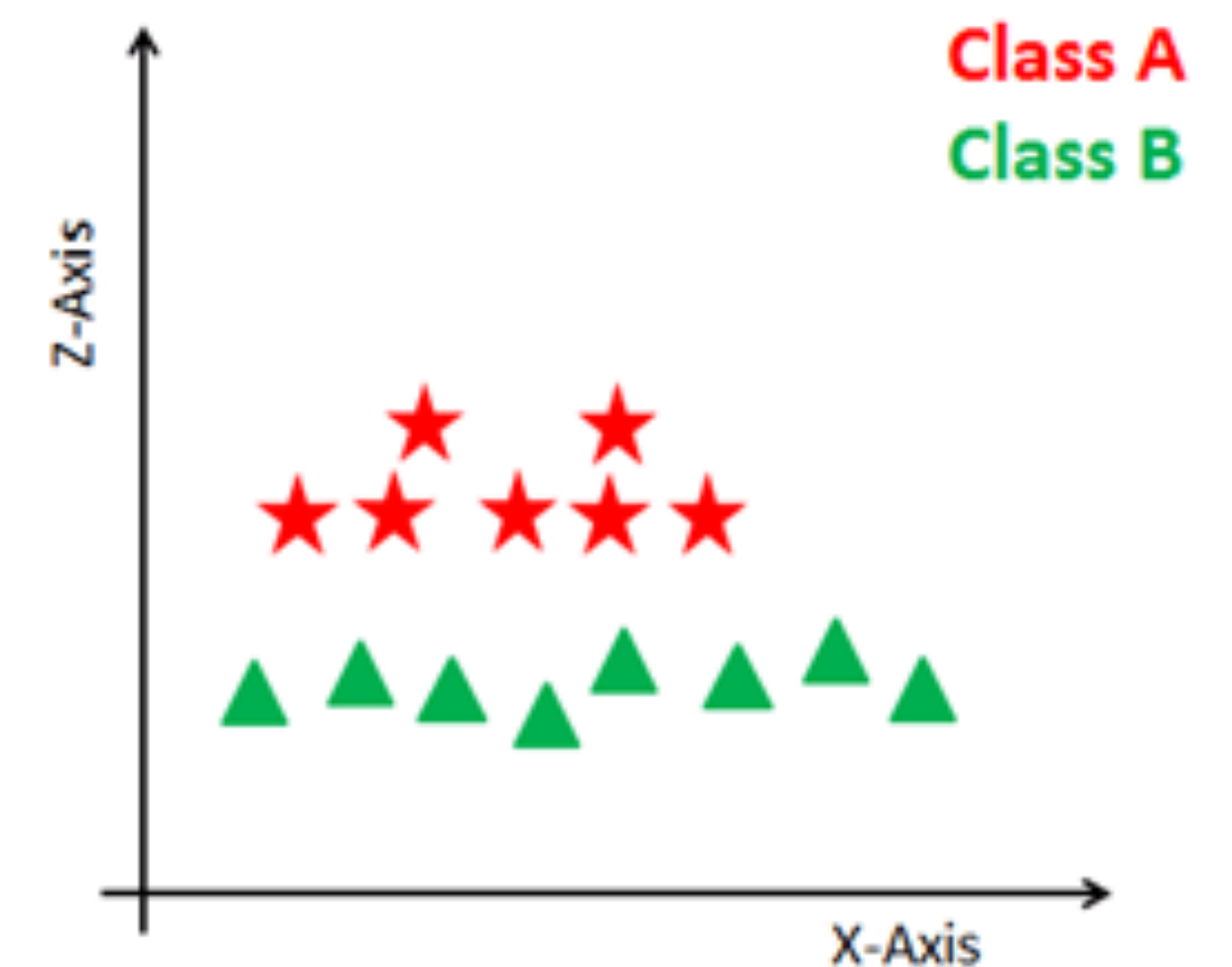
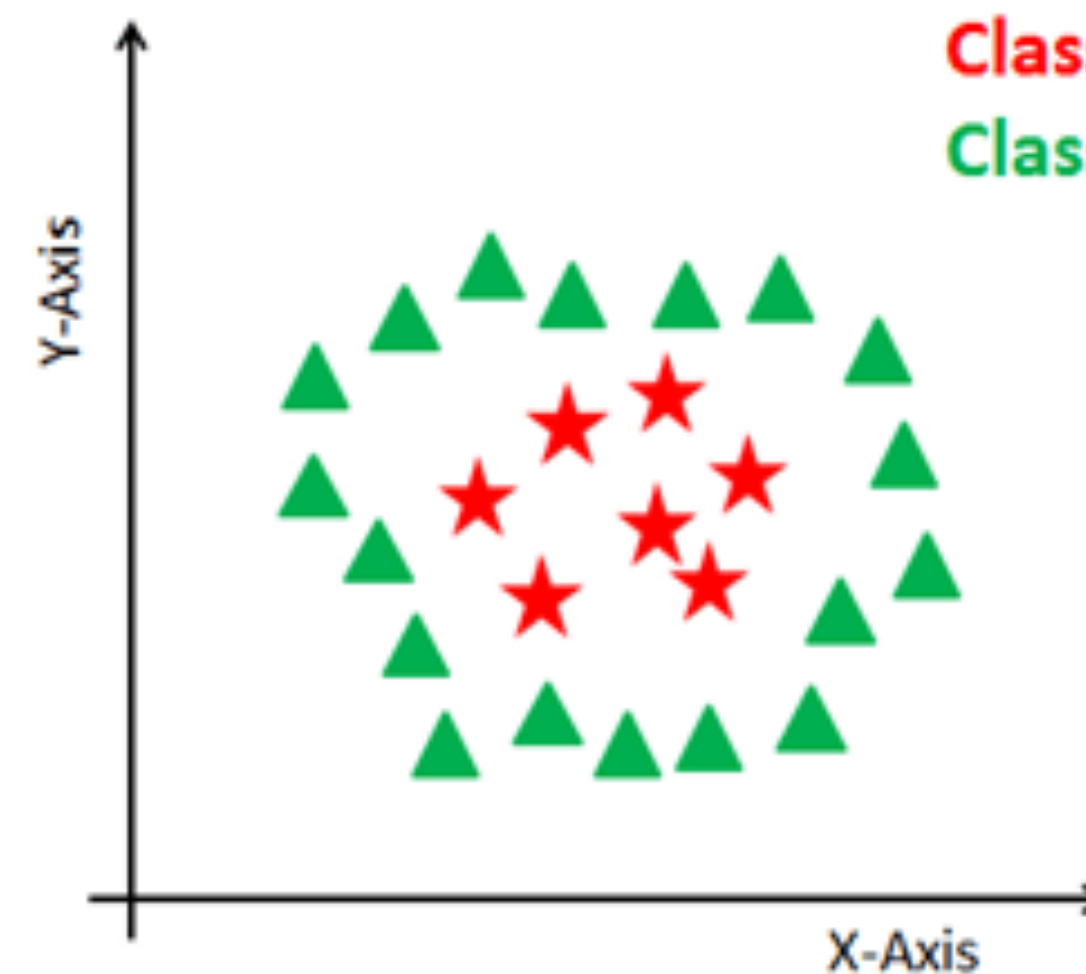
Figure from [https://en.wikipedia.org/wiki/File:SVM\\_margin.png](https://en.wikipedia.org/wiki/File:SVM_margin.png)  
shared under CC 4.0 license

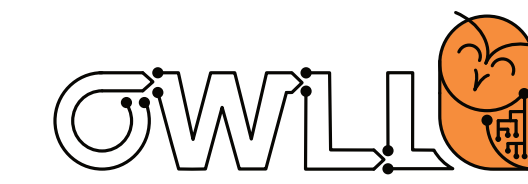
# Support vector machines (SVM)



## Example: support vector machine (SVM)

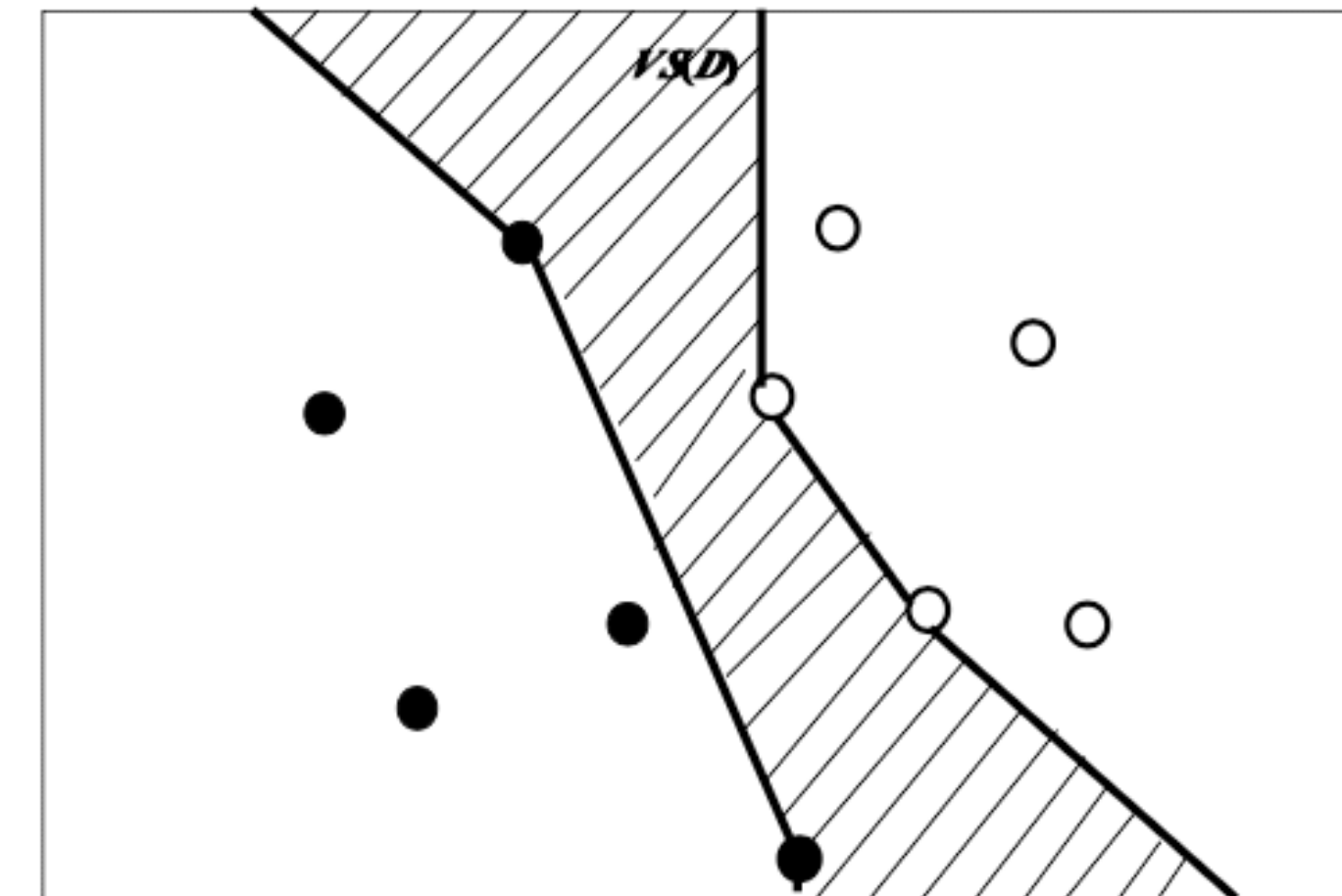
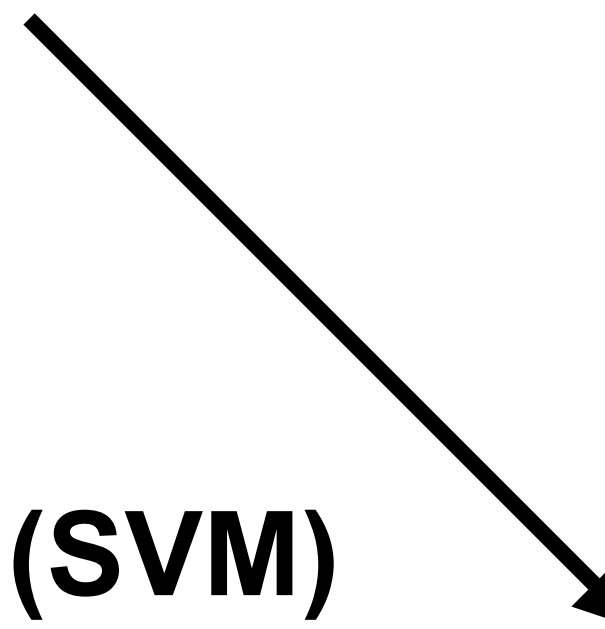
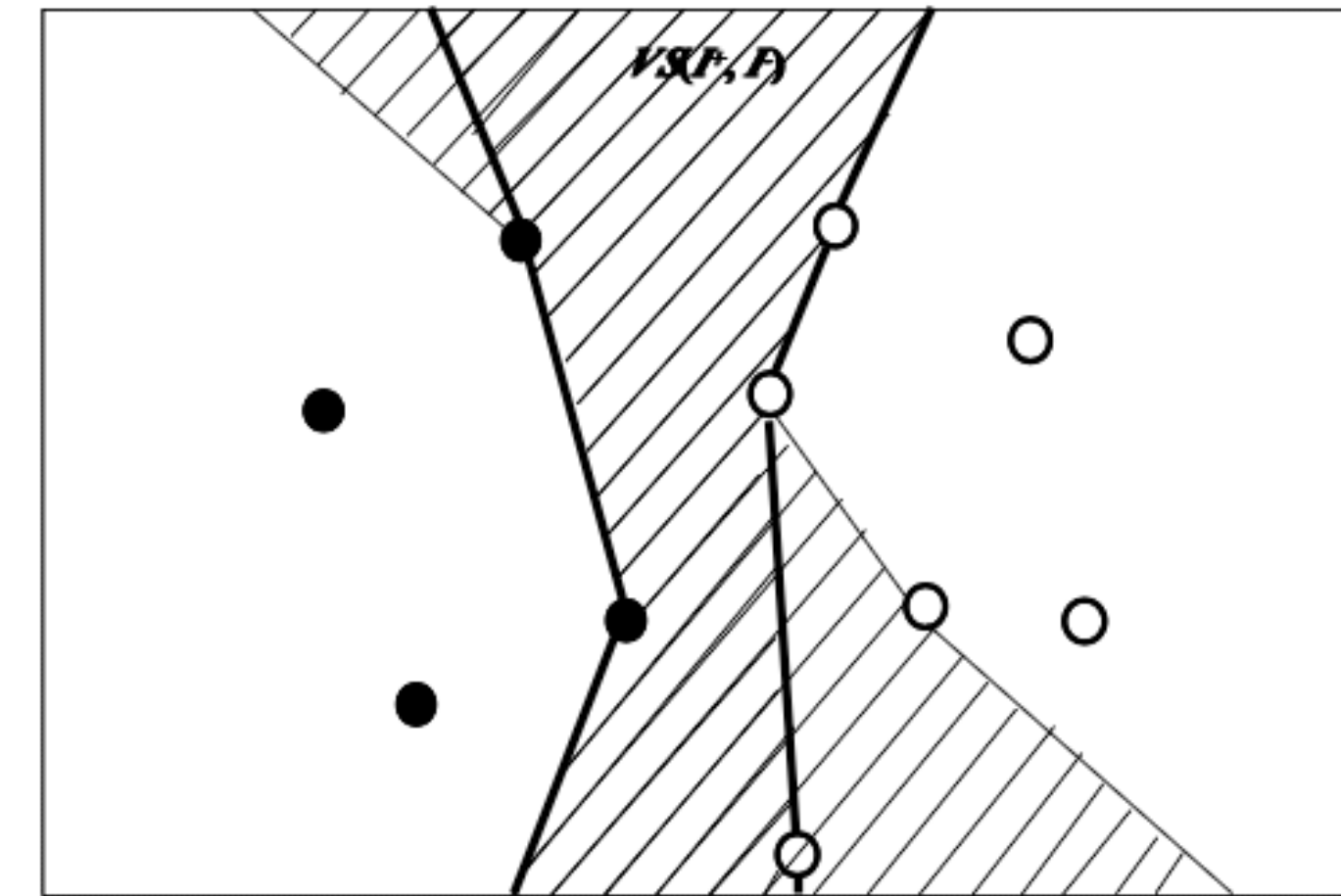
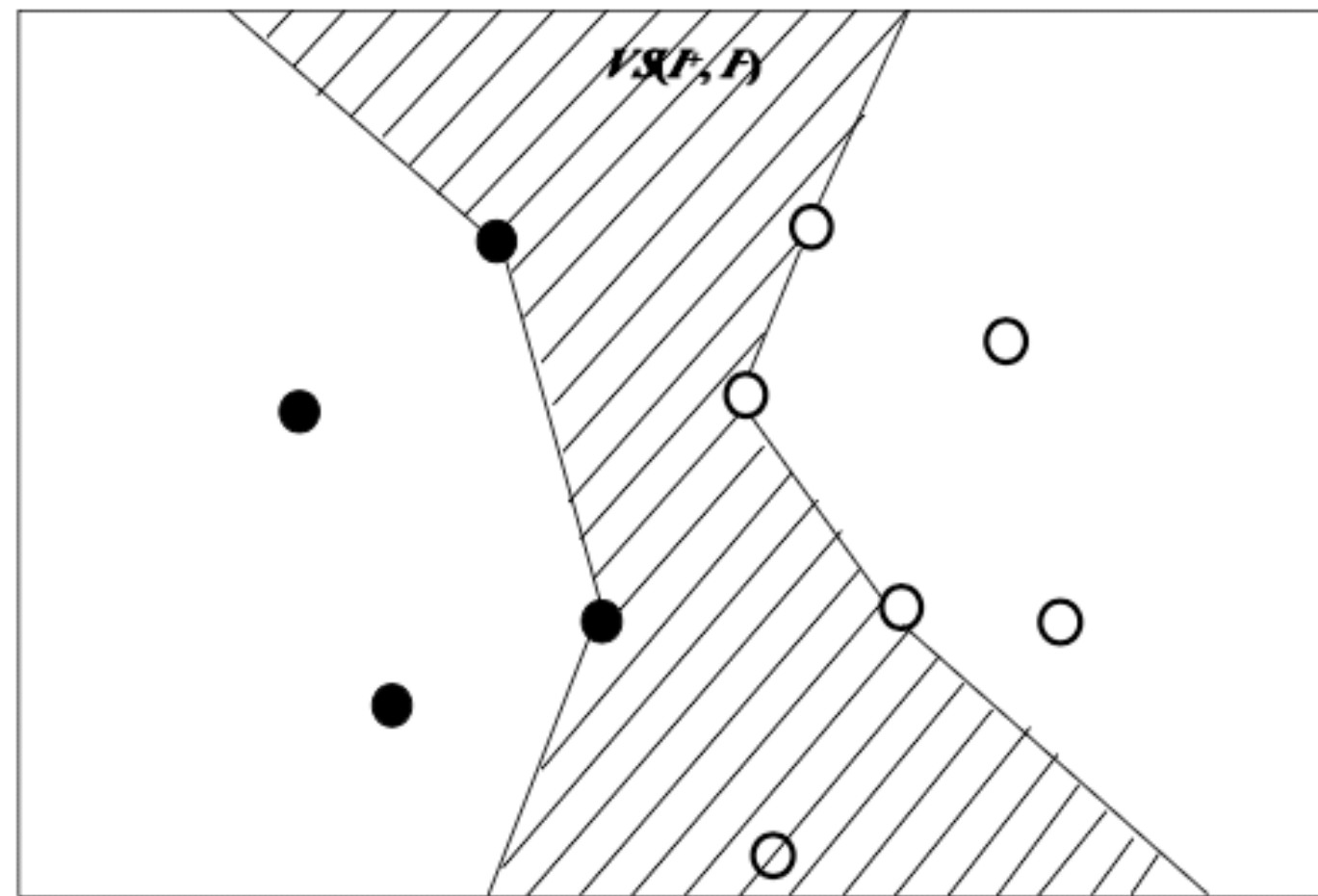
- Data is not always linearly separable
- We can **project** data to a (higher dimensional) **feature space** through kernel functions
- Example: polar coordinates





# Back to version spaces

# Version space: SVM



An example: support vector machine (SVM)

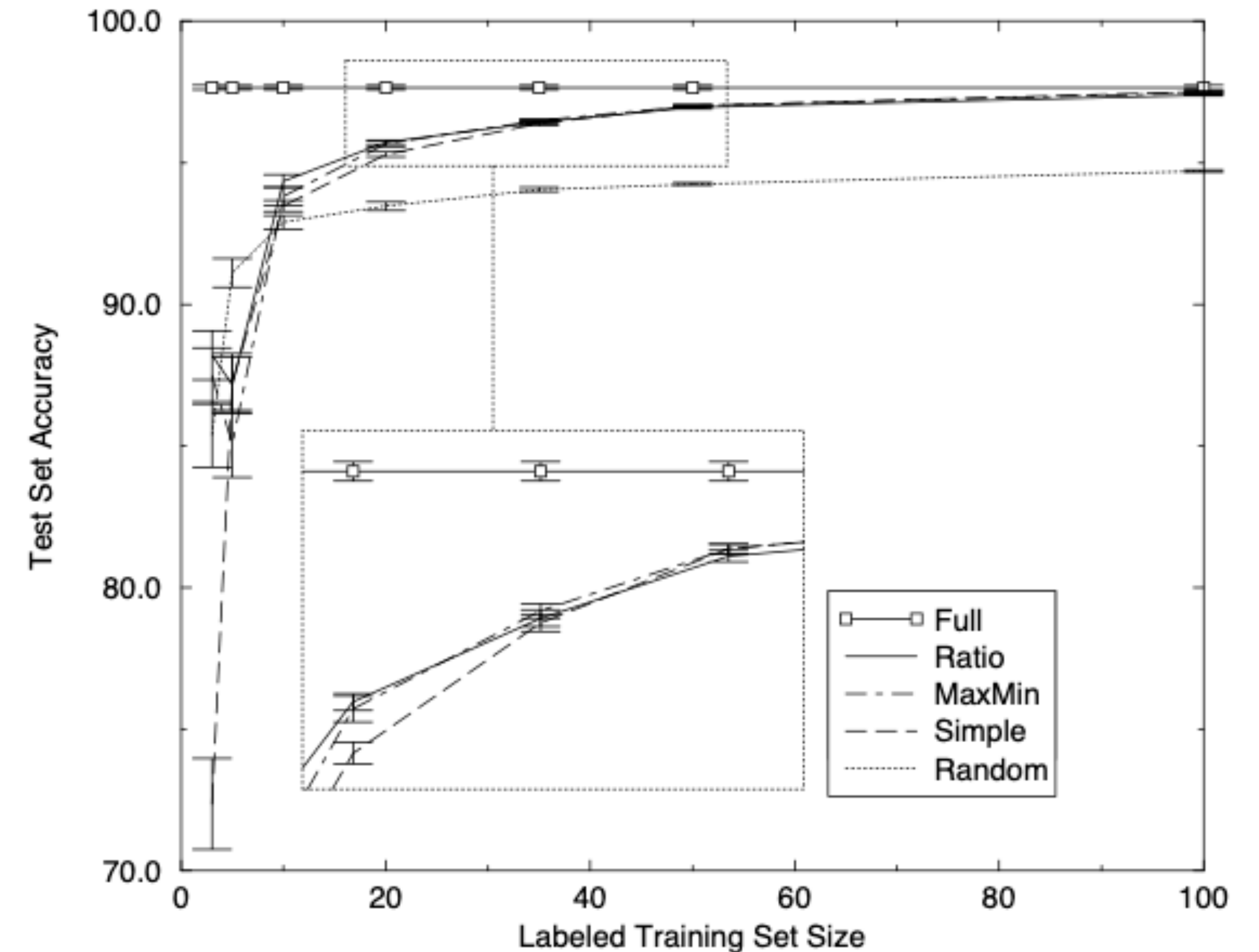
Version space is **set of hyperplanes**  
(or could be redefined through vectors  $W$ )

# Version space: SVM

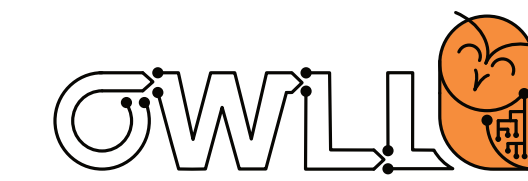


## Active learning with SVMs

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







# (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  $x^*$ , such that when the query is given label  $y^*$  and added to the training set, the learner trained on the resulting set  $D+(x^*, y^*)$  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.

# Information theoretic quantities



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

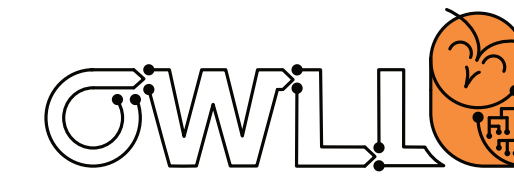
Example: maximize expected information gain by querying examples with **largest entropy**  
(as a measure of disorder, related to information gain)

$$H(p) = - \sum_i^c p_i \log_2(p_i)$$

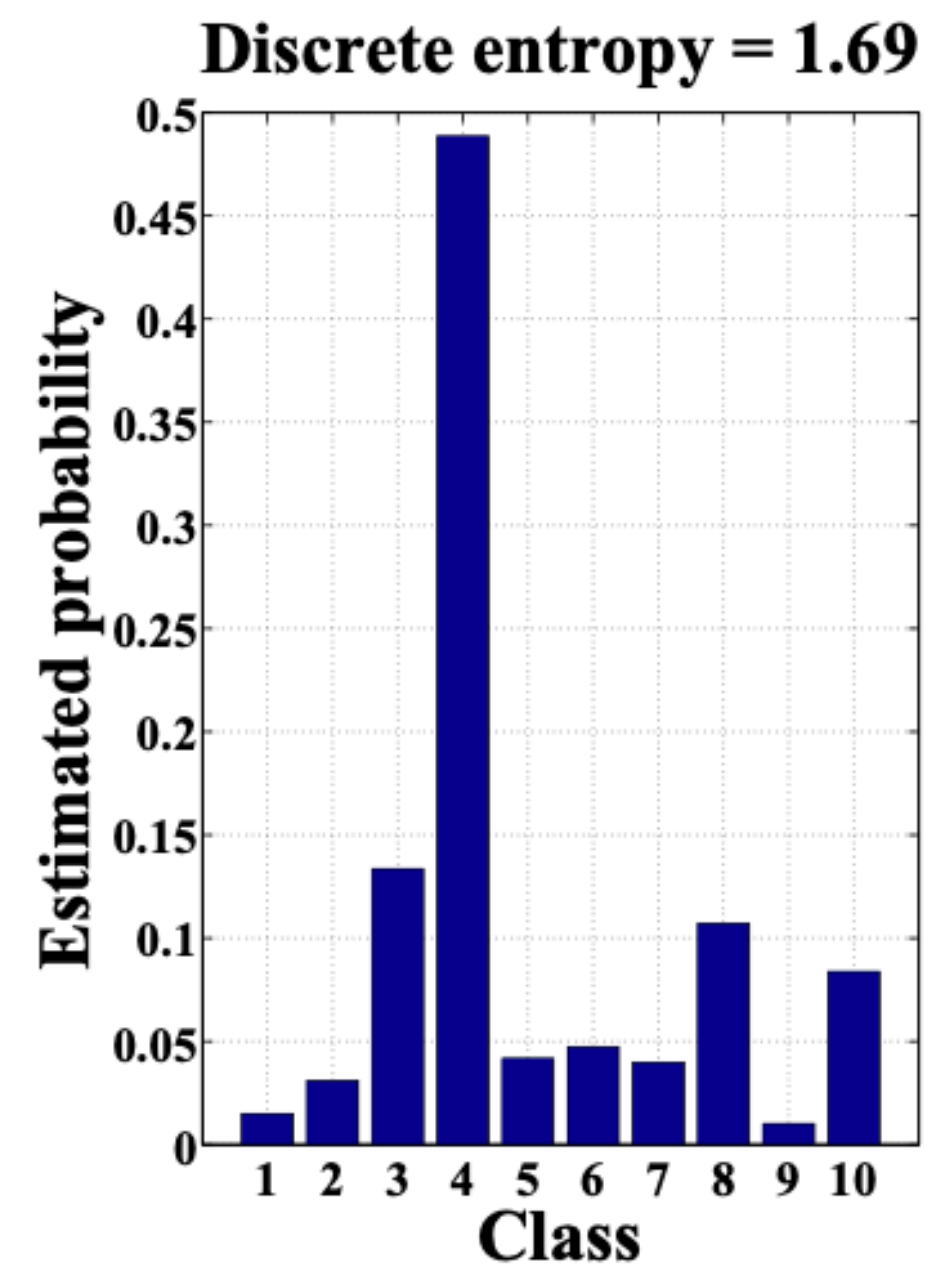
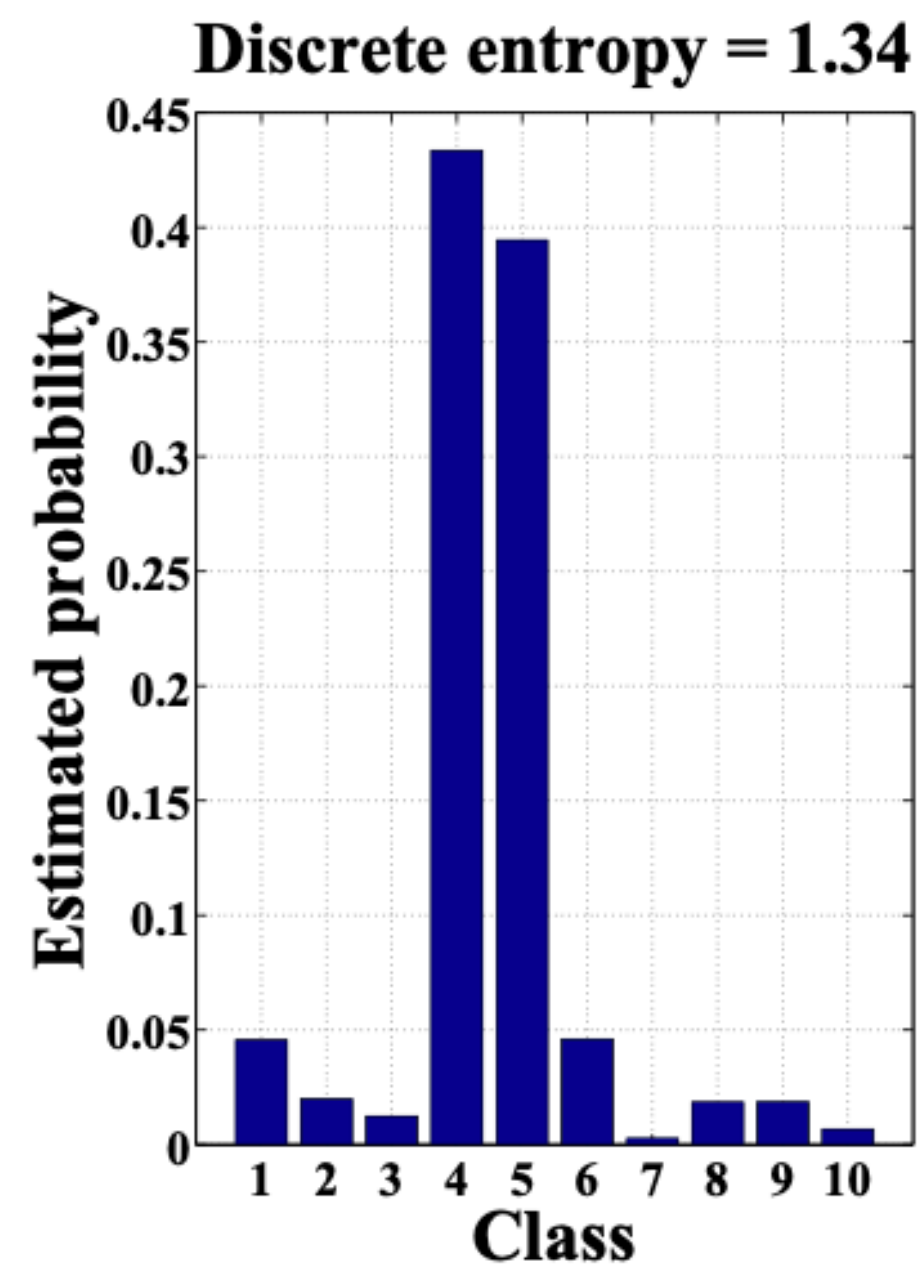
Example  $p(y|x)$ :

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

# Best versus second best

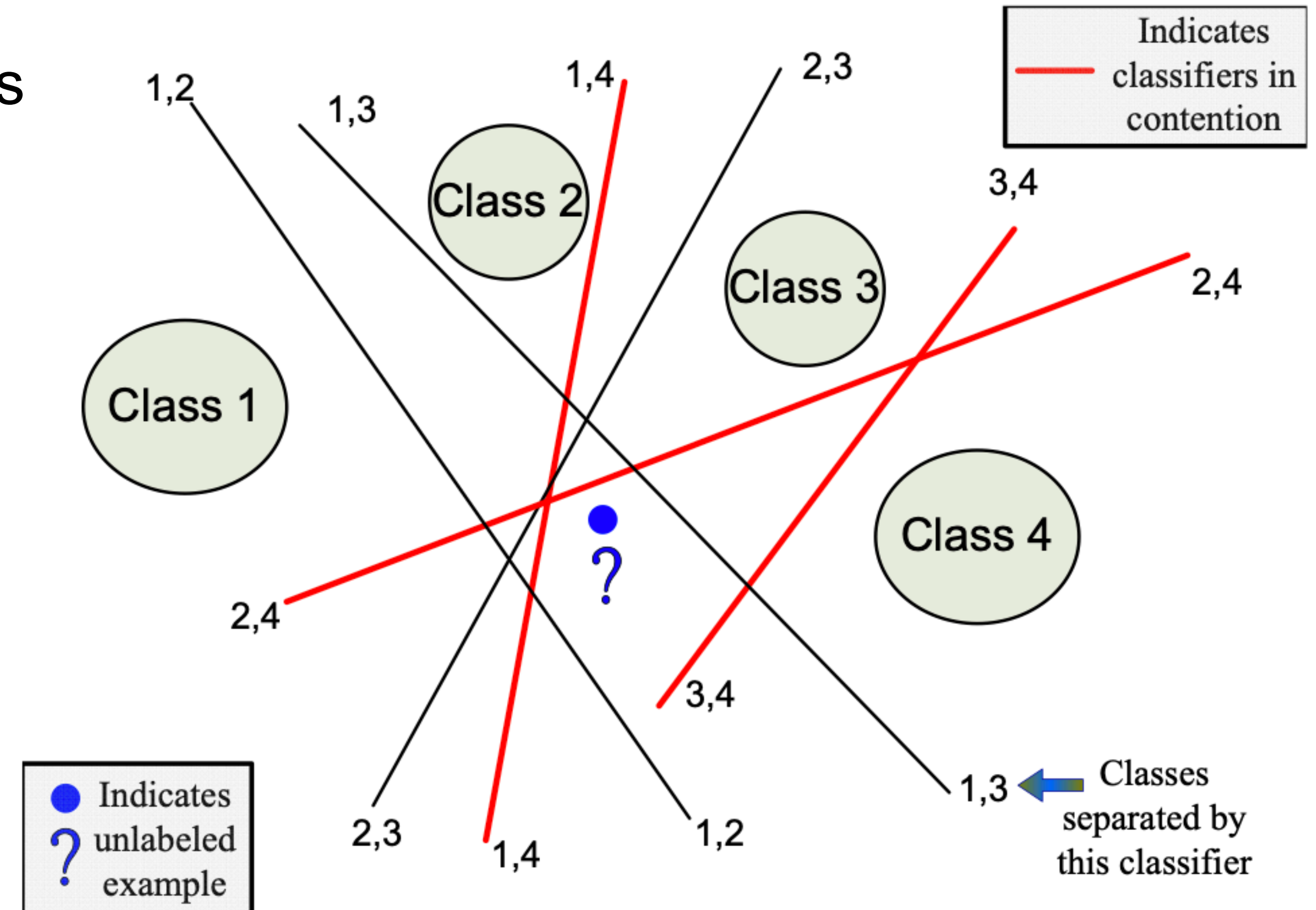
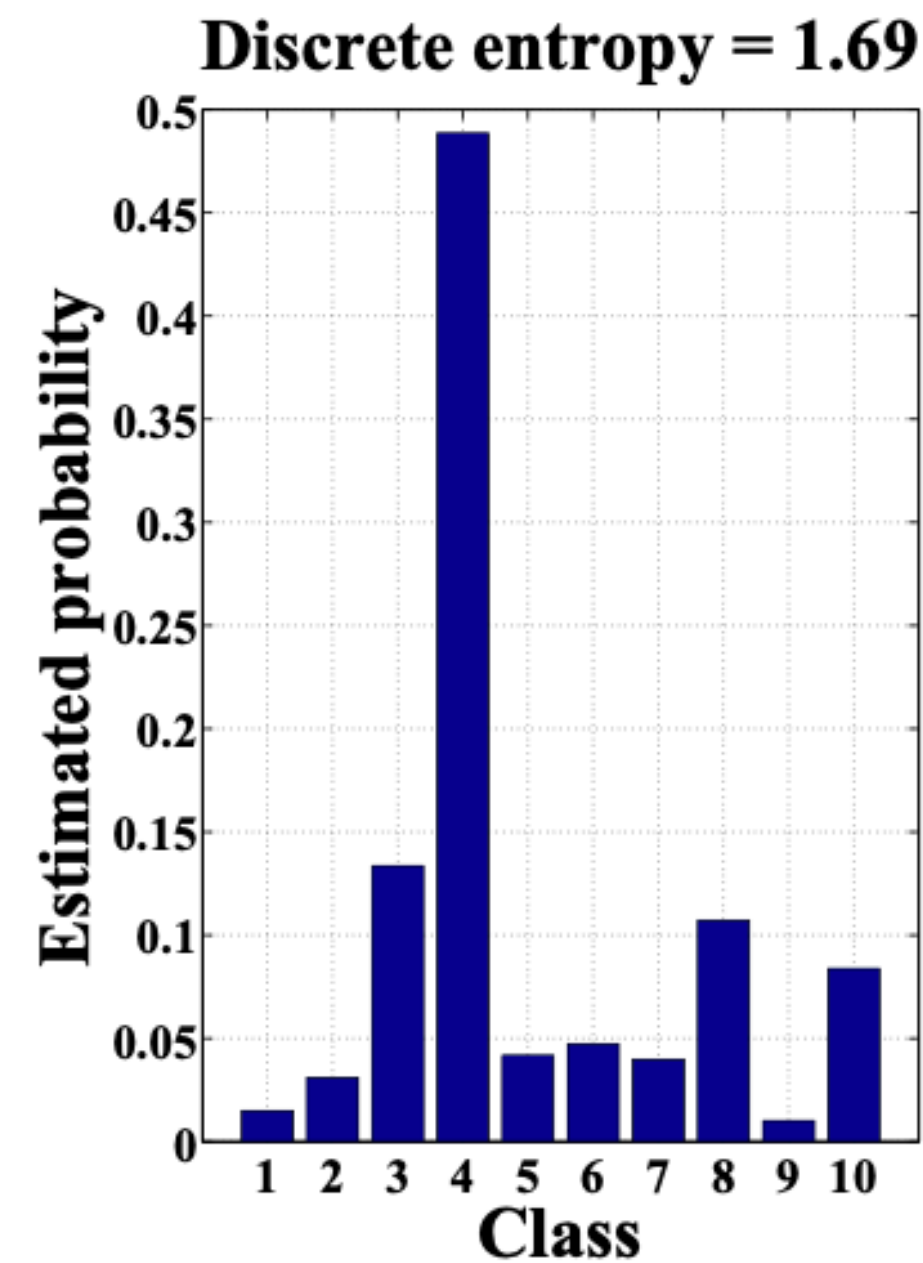
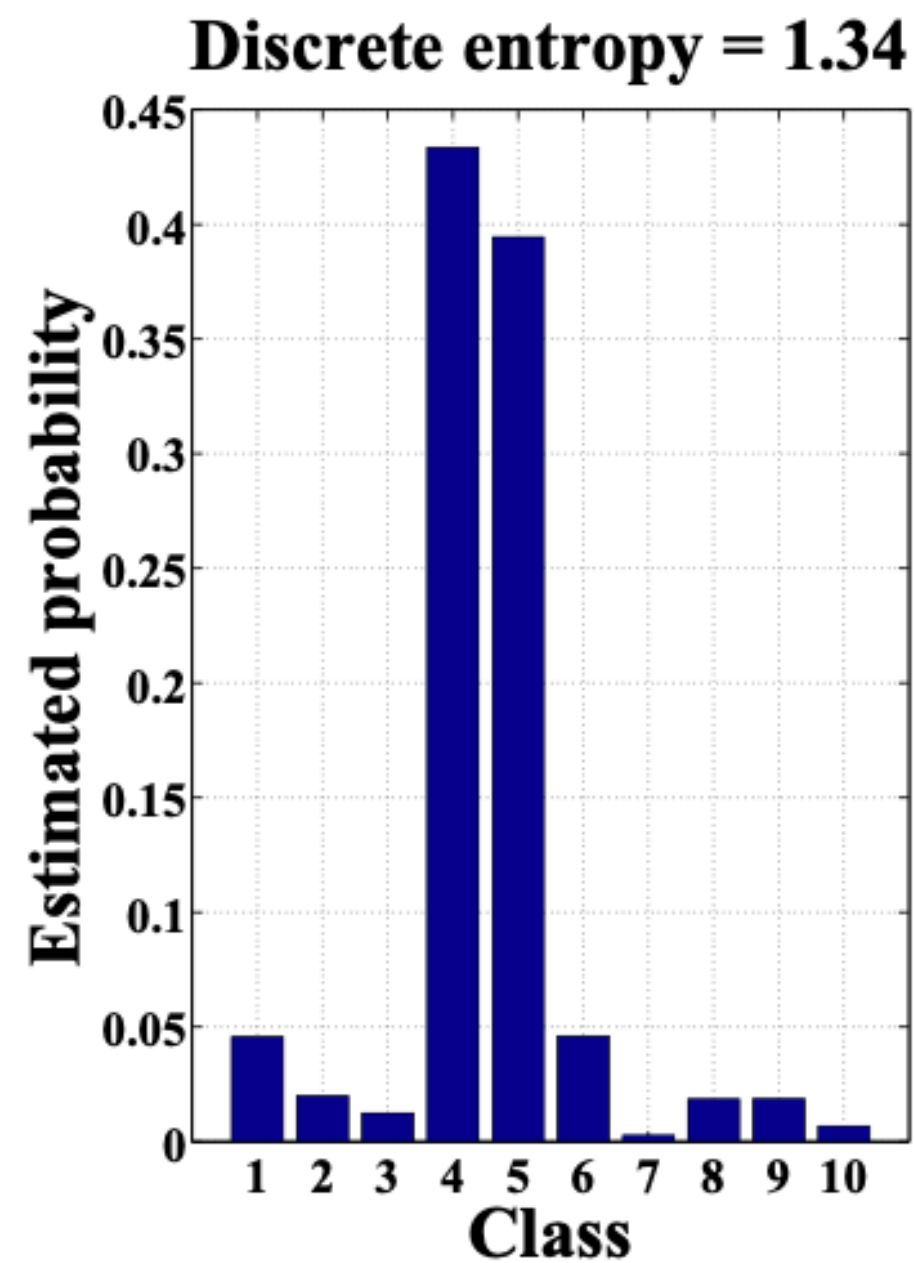


Confidence & entropy can be poor estimates when multiple classes are considered

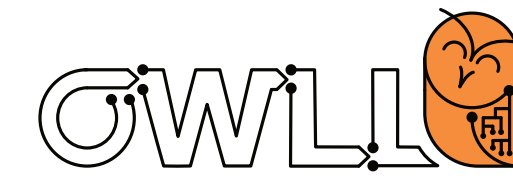


# Best versus second best

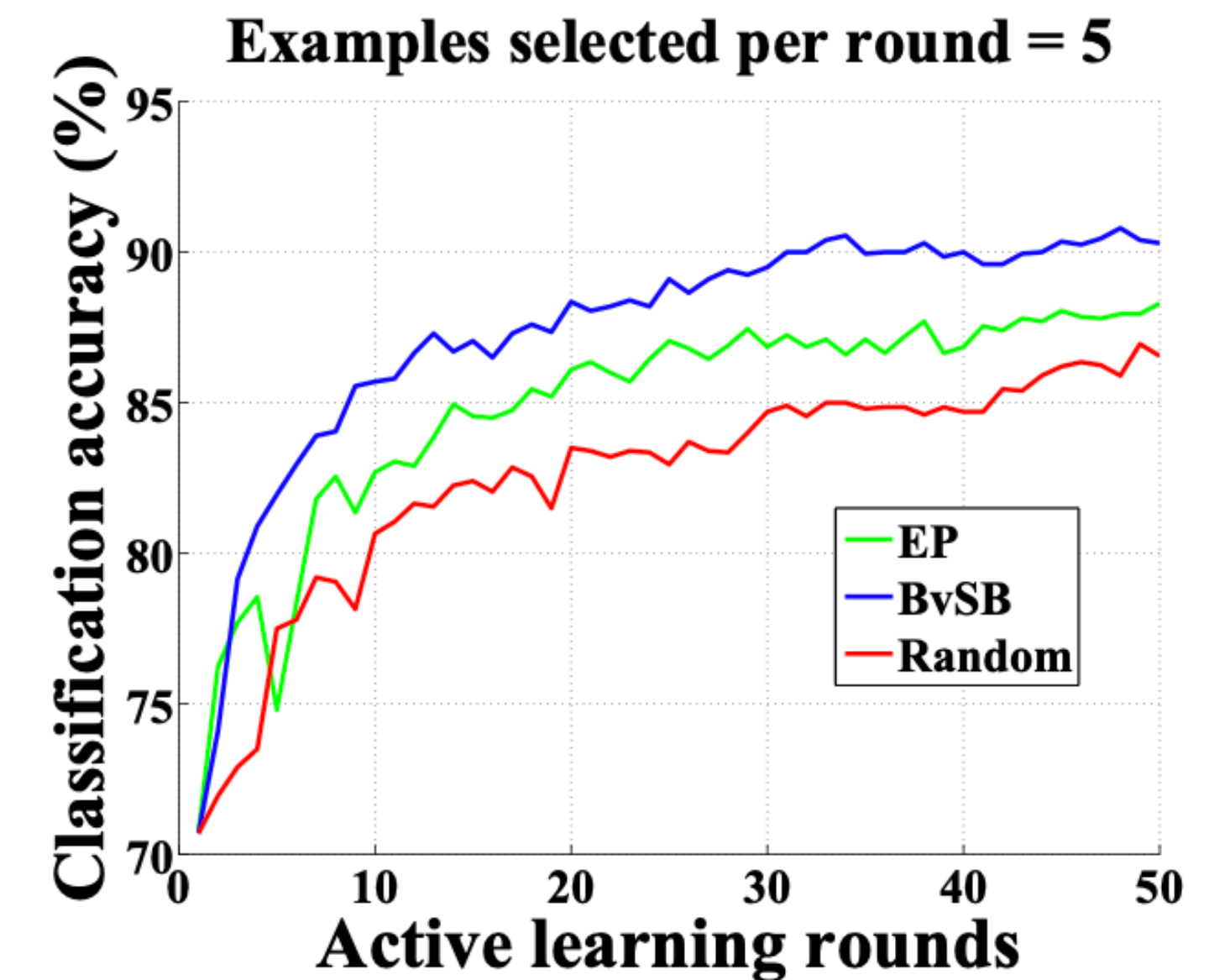
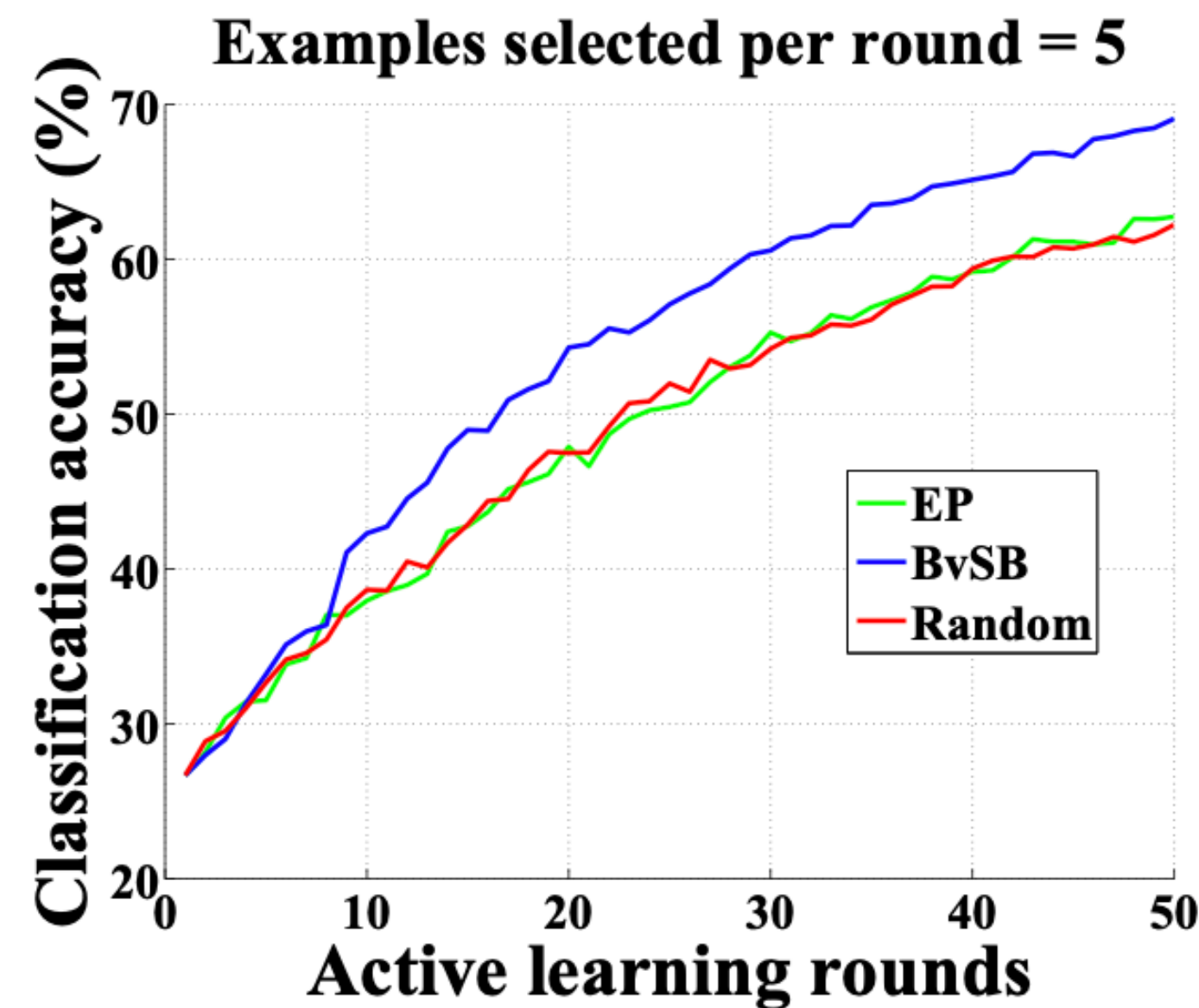
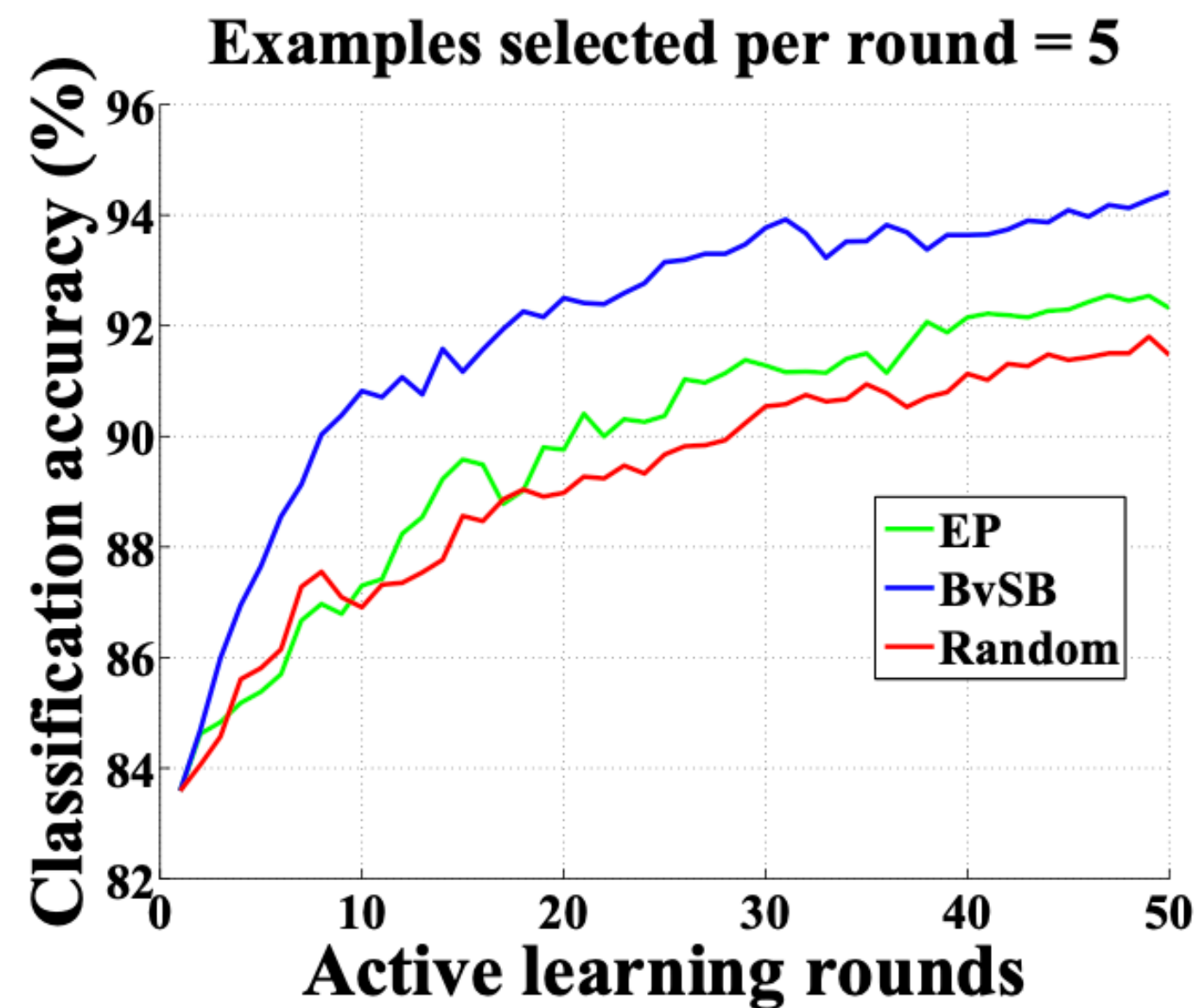
Confidence & entropy can be poor estimates when multiple classes are considered



# Best versus second best



Left to right: Pendigits, Letter, USPS datasets





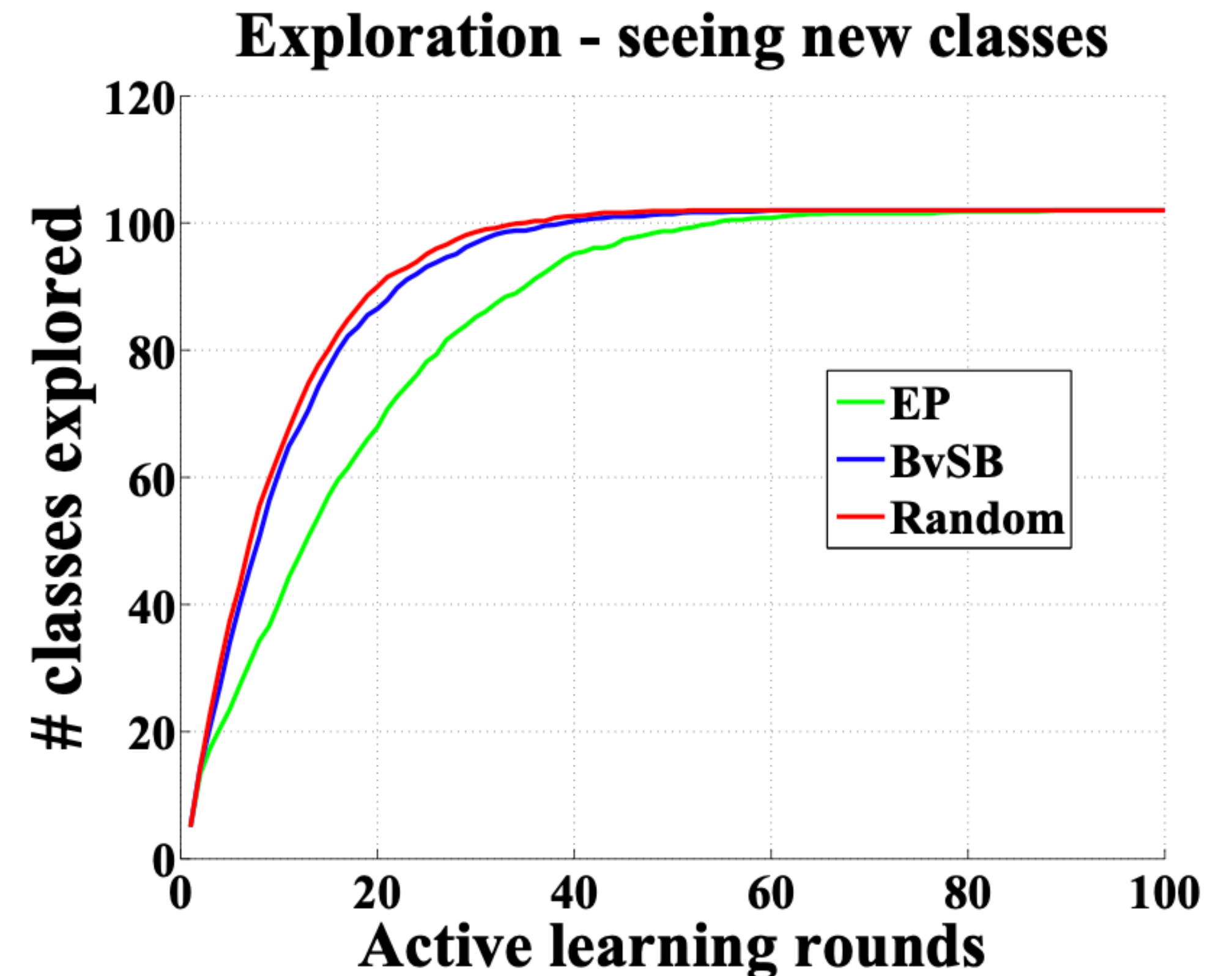
# Exploration vs. exploitation?



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**”



# 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)

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

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

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

# 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

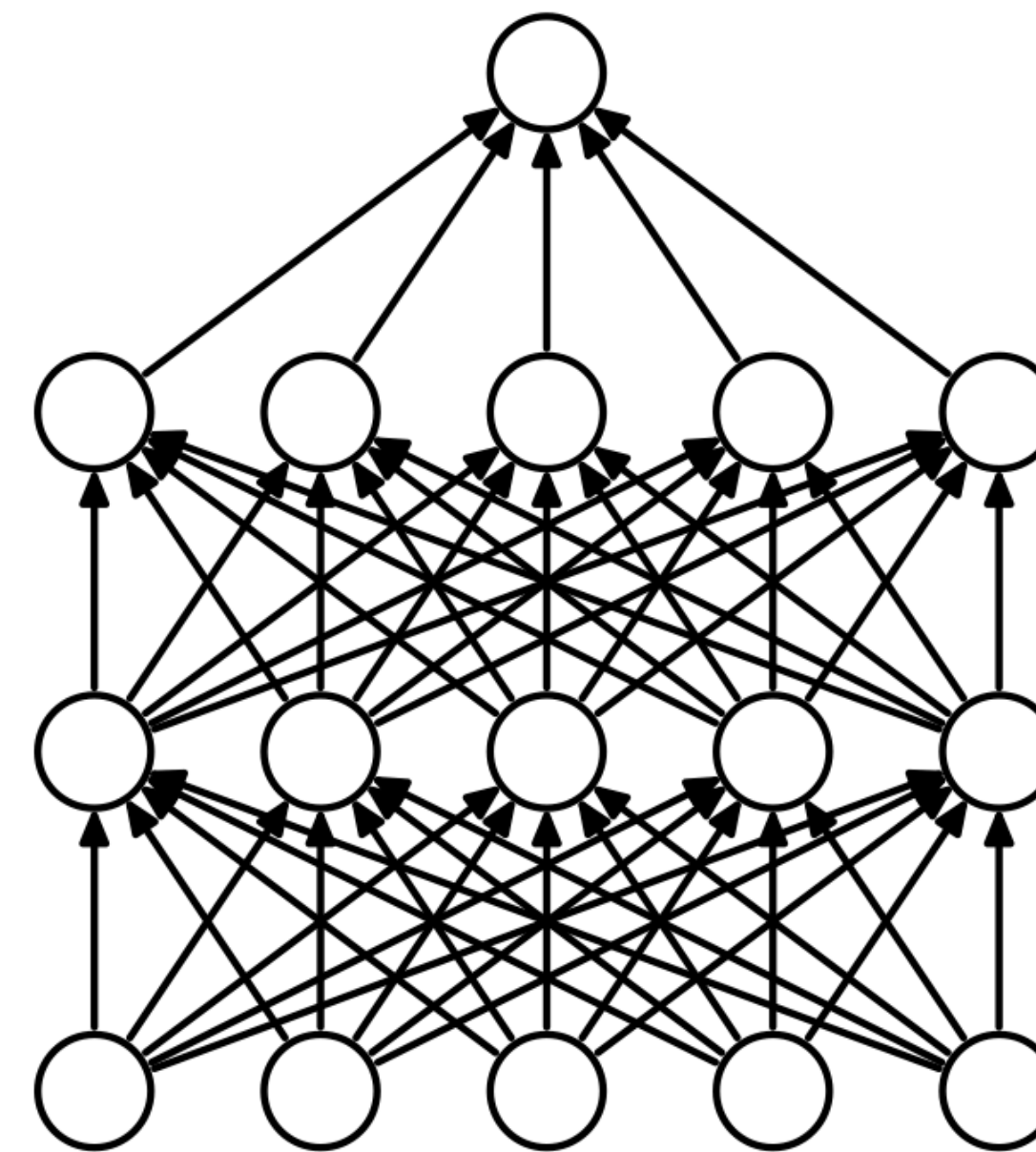
Repeat the following until  $n$  queries have been accepted

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.

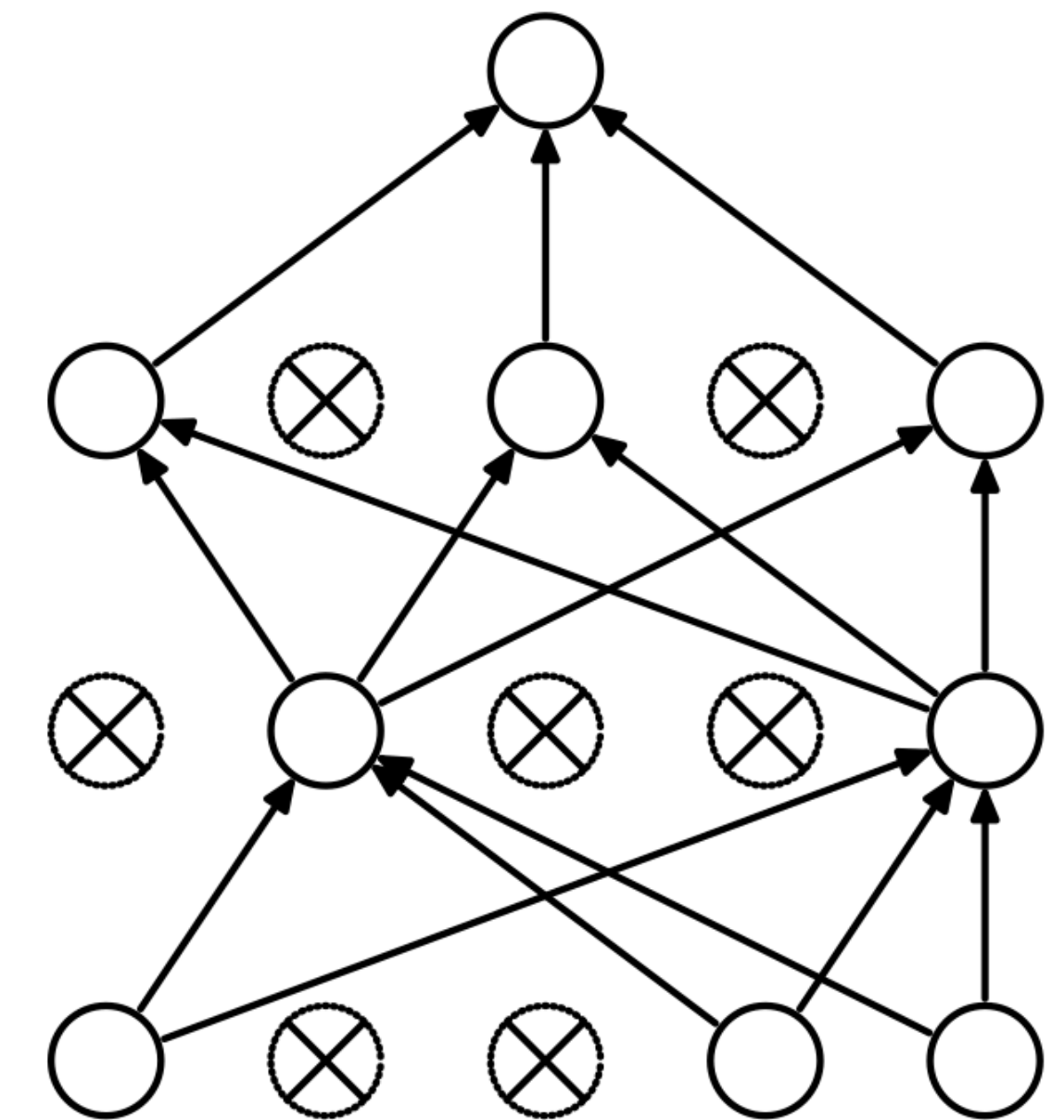
# Monte Carlo Dropout

**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)

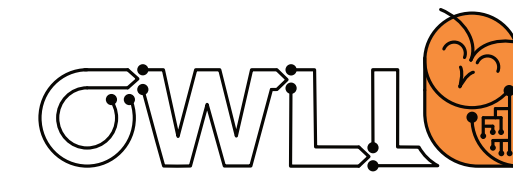


(a) Standard Neural Net



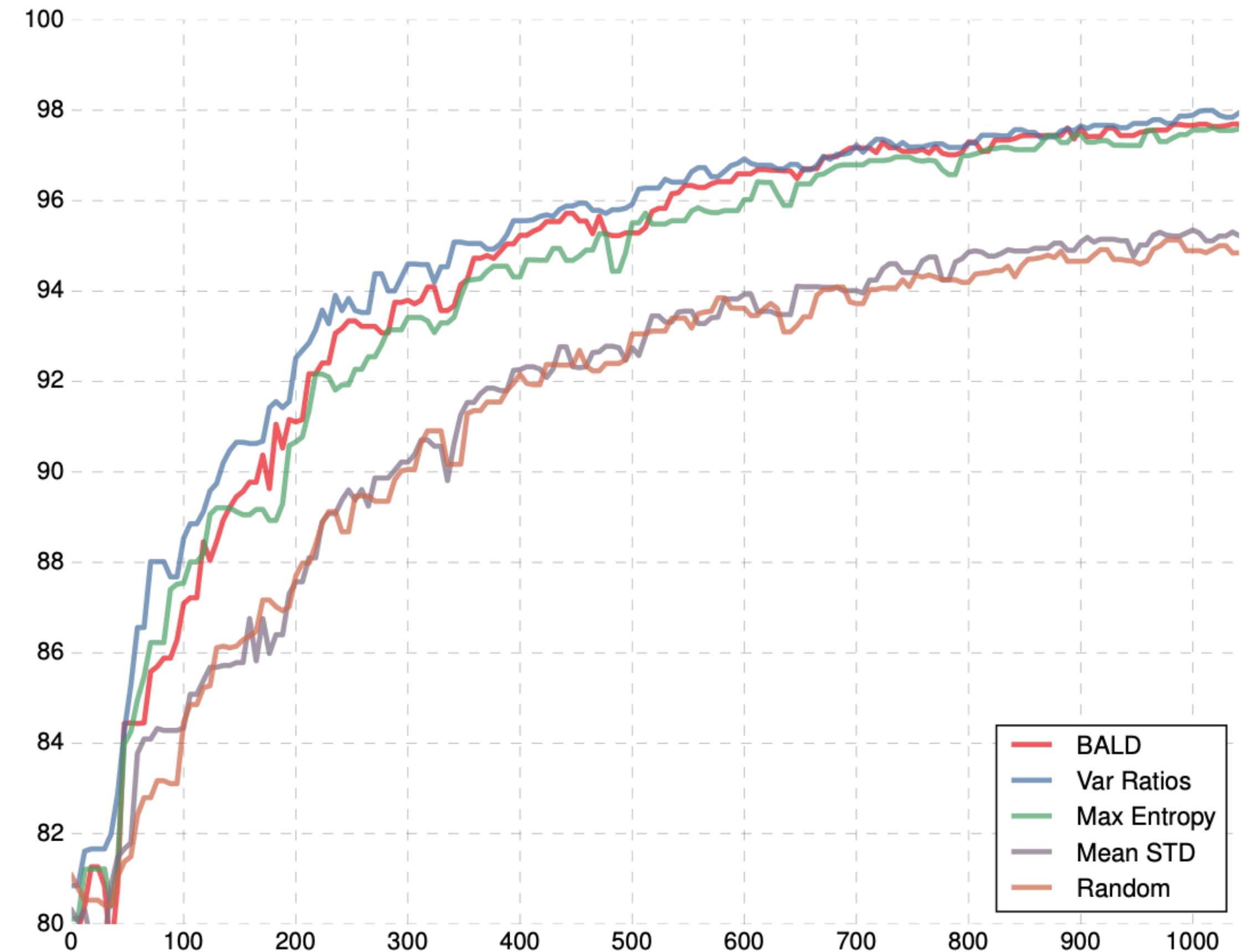
(b) After applying dropout.

# Monte Carlo Dropout



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

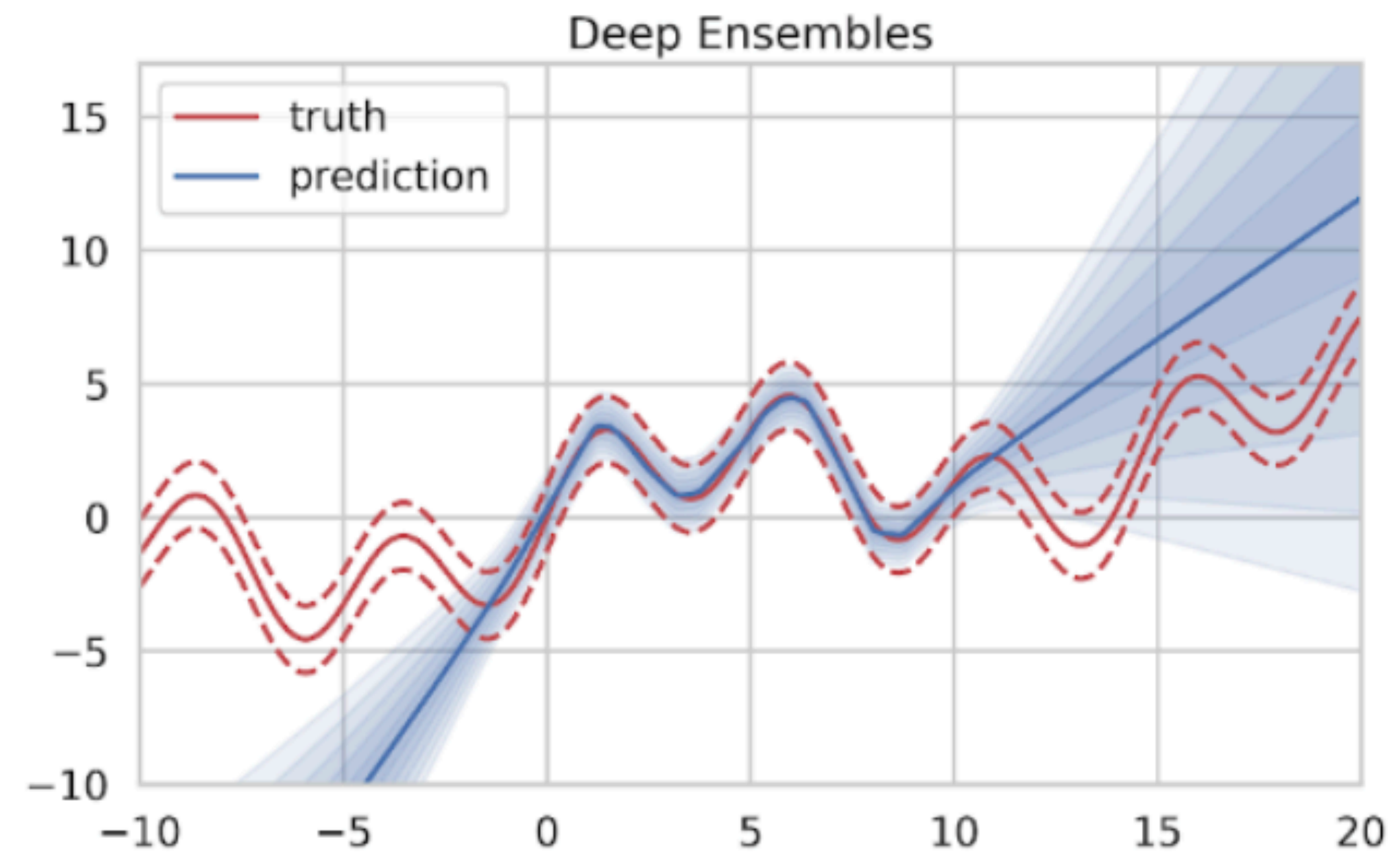
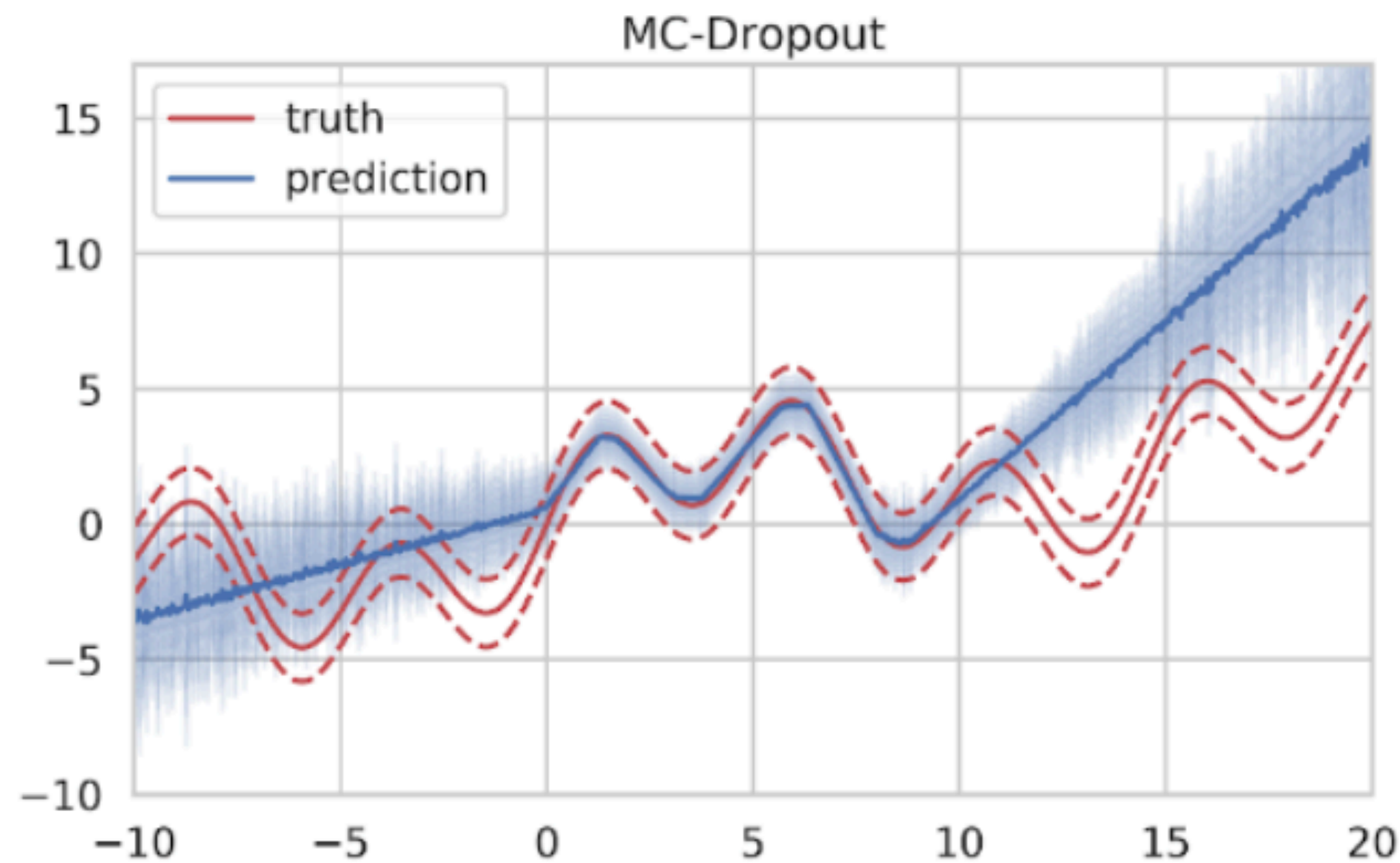
The acquisition function could still be entropy, standard deviation in output confidence etc.



# Limits of uncertainty sampling



Why aren't these approaches a lot better?



# Limits of uncertainty sampling



Why aren't these approaches a lot better?

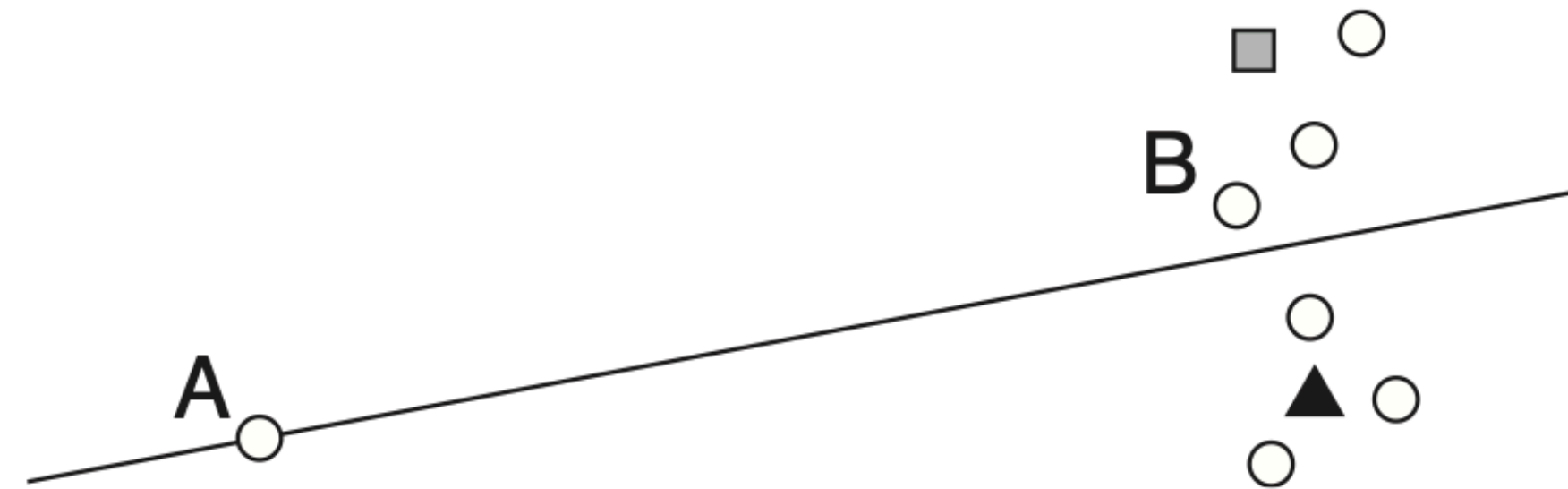
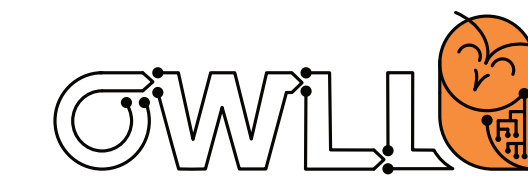


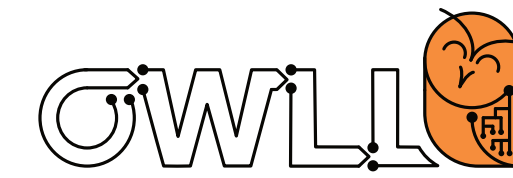
Figure 2: An illustration of when uncertainty sampling can be a poor strategy for classification. Shaded polygons represent labeled instances ( $\mathcal{L}$ ), and circles represent unlabeled instances ( $\mathcal{U}$ ). Since  $A$  is on the decision boundary, it will be queried as the most uncertain.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



# 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

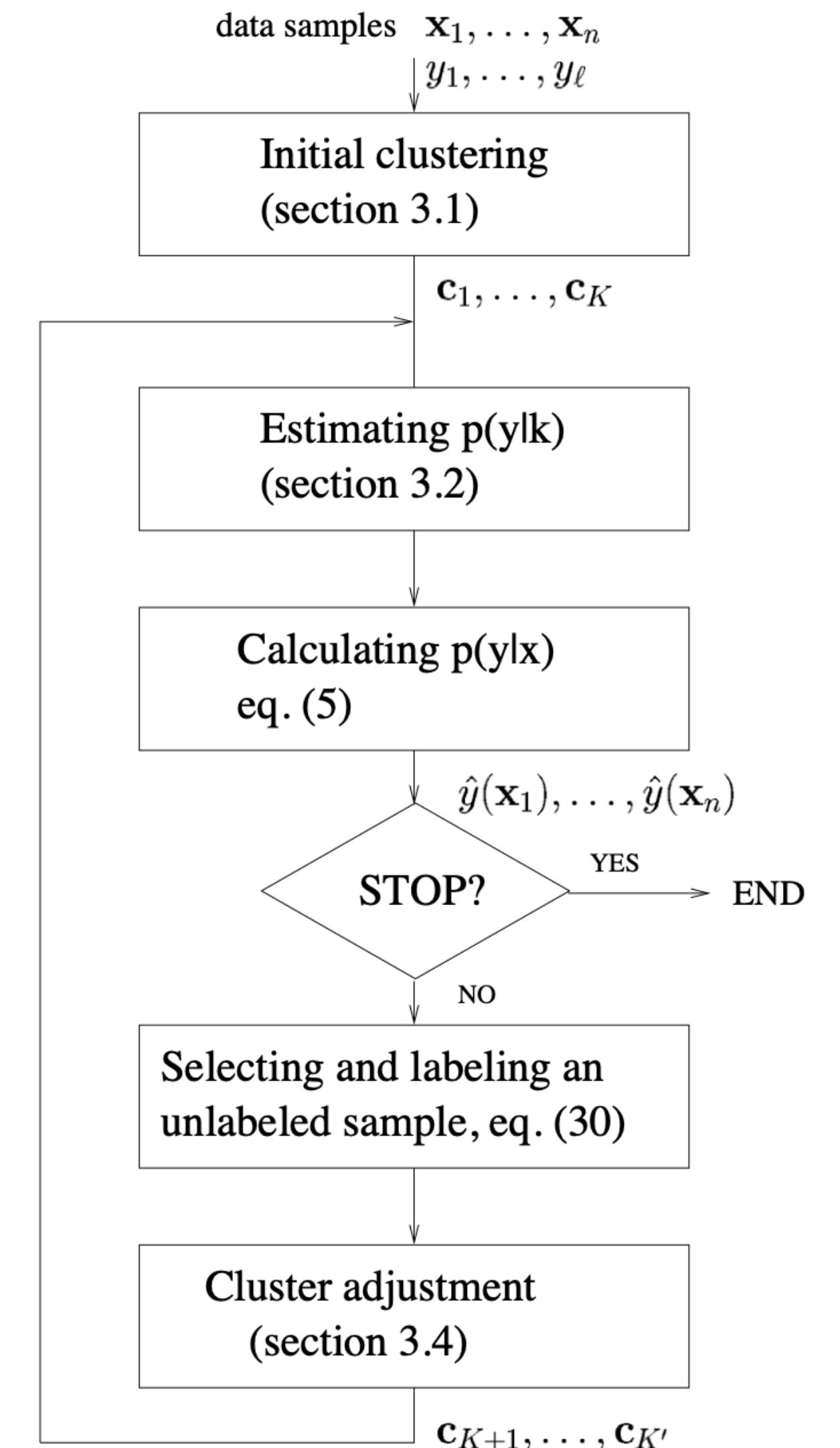
We could then also **make use of core sets**, as discussed for rehearsal in the last lecture

# 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

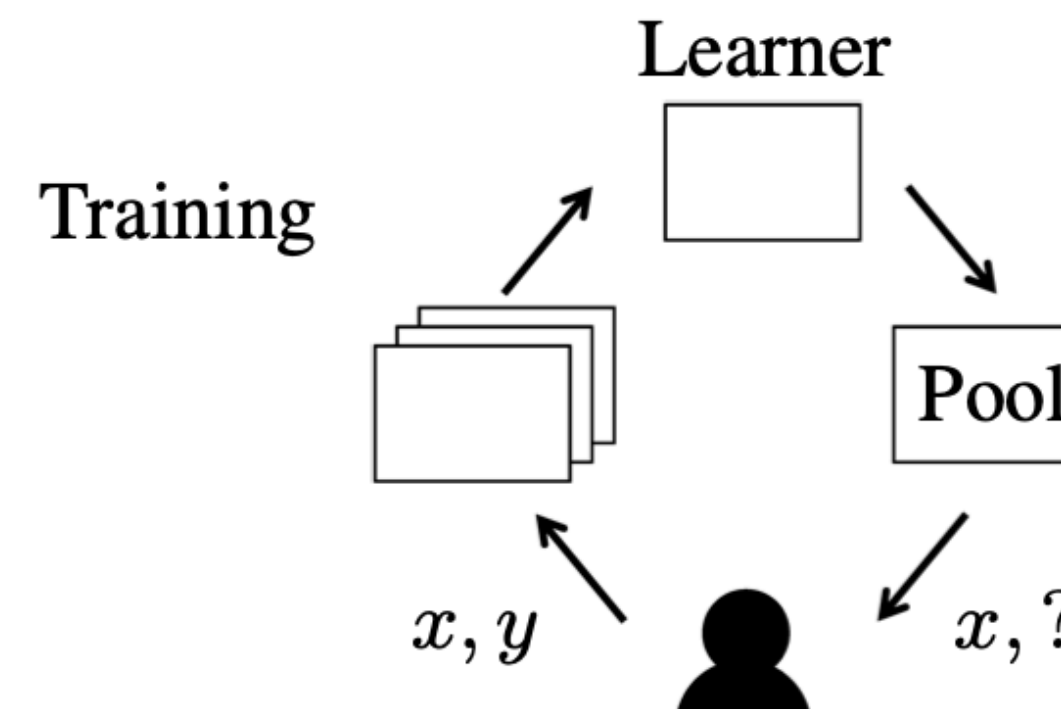


# Representations & core sets

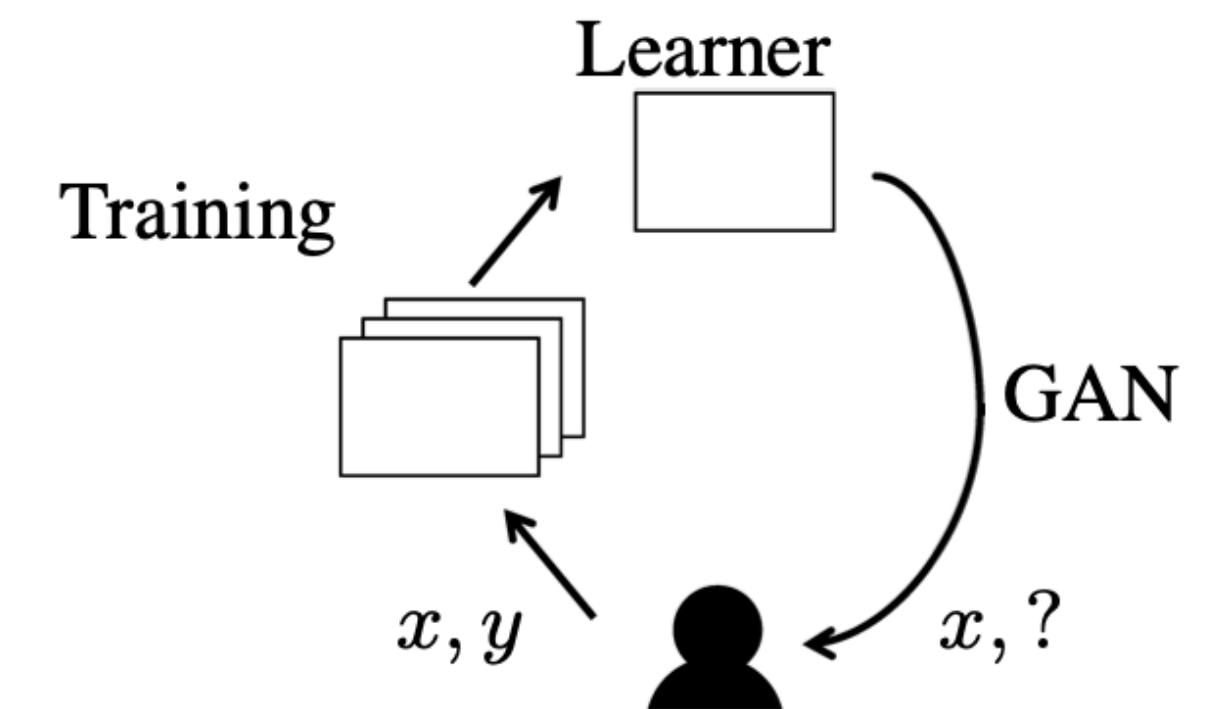


Example: **generative adversarial active learning**

- 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

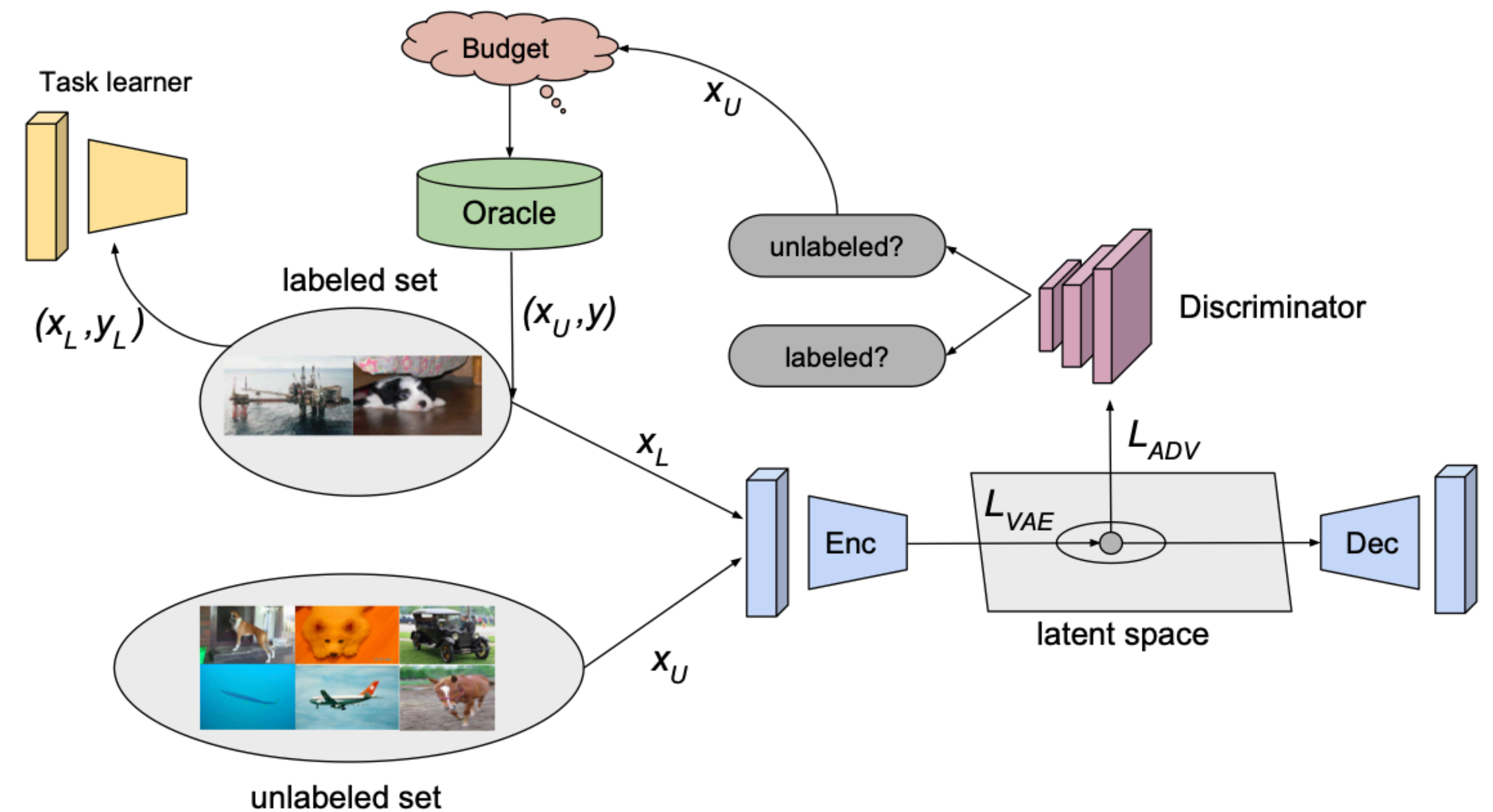


(b) GAAL

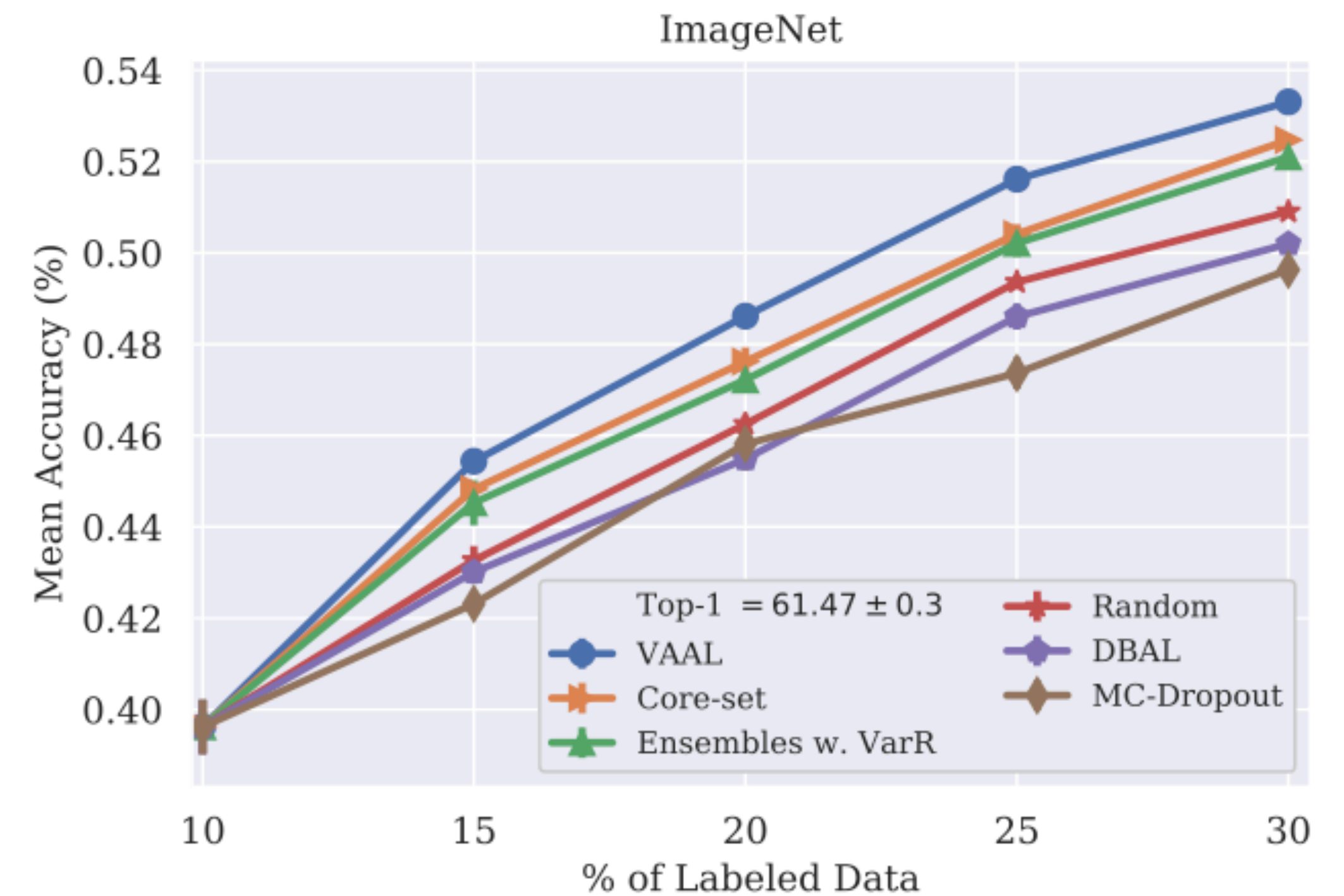
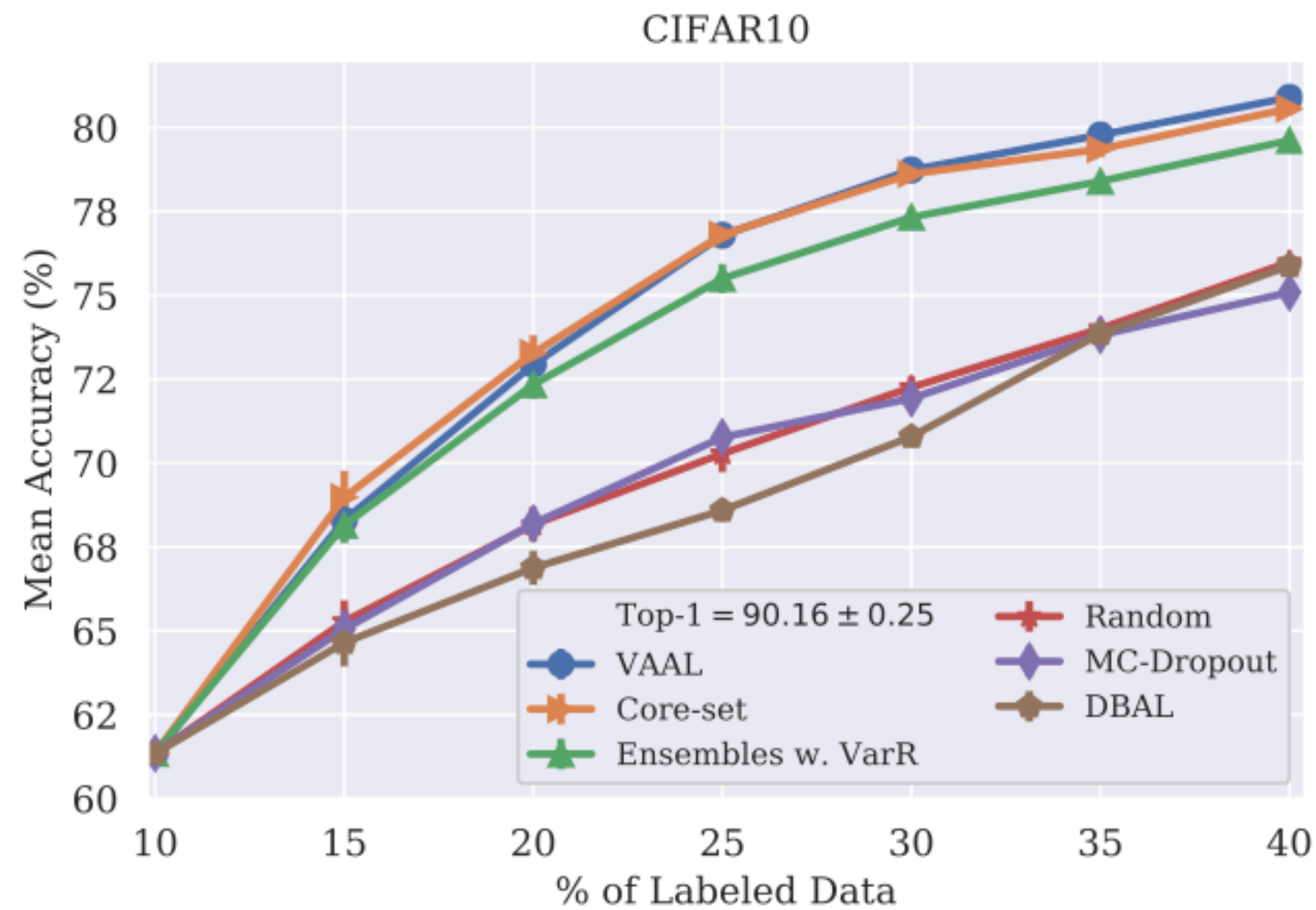
# Representations & core sets

## 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



# Representations & core sets





**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  $h : \mathcal{X} \rightarrow \mathcal{Y}$  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

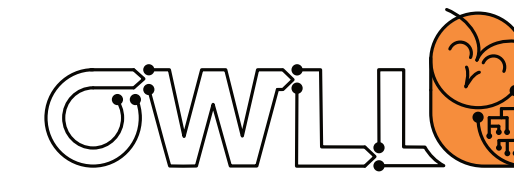
- Version space reduction
- Minimum confidence
- Maximum entropy
- Best versus second best
- Model “uncertainty” (output variability)
- Ensembles/query by committee
- Representation learning on the pool
- Core sets

## & (some of) their assumptions

- Set of hypotheses is clear
- No overconfidence phenomenon and out-of-distribution/task data
- Accurate uncertainty everywhere
- Training of multiple models
- 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”

