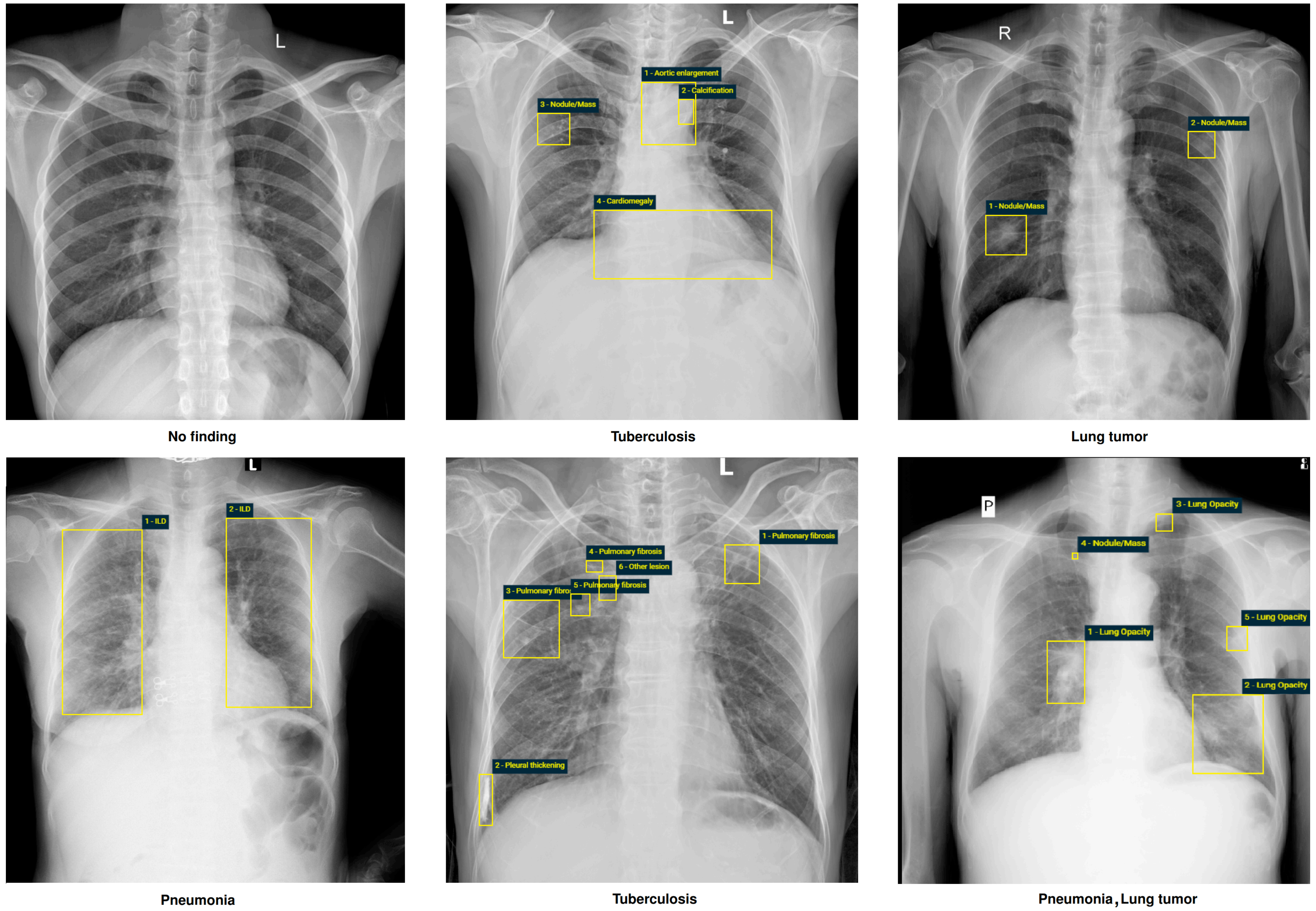
Active Learning and Weak Supervision

DSCC 251/451: Machine Learning with Limited Data

C.M. Downey

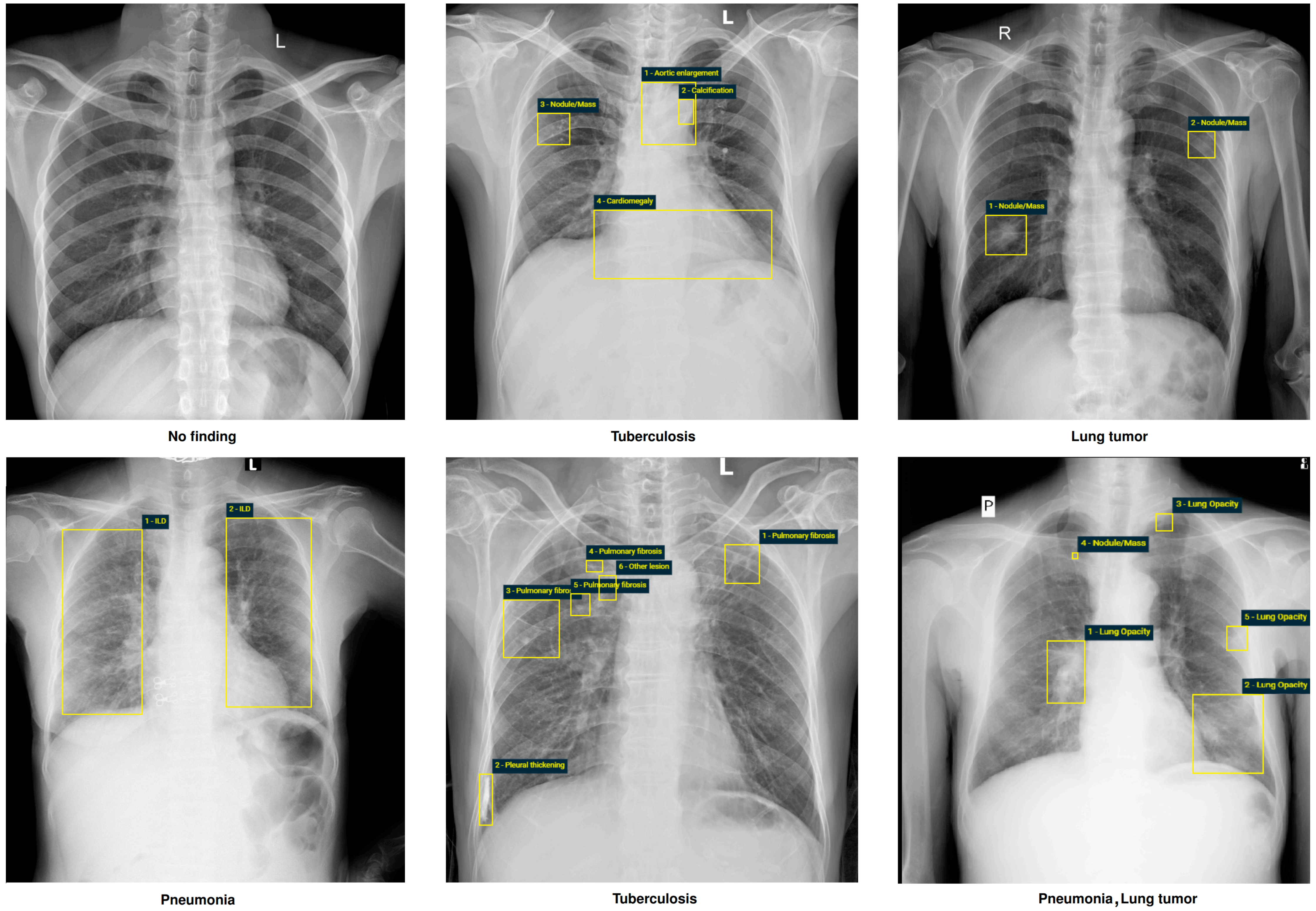
Spring 2026

Active Learning Scenario



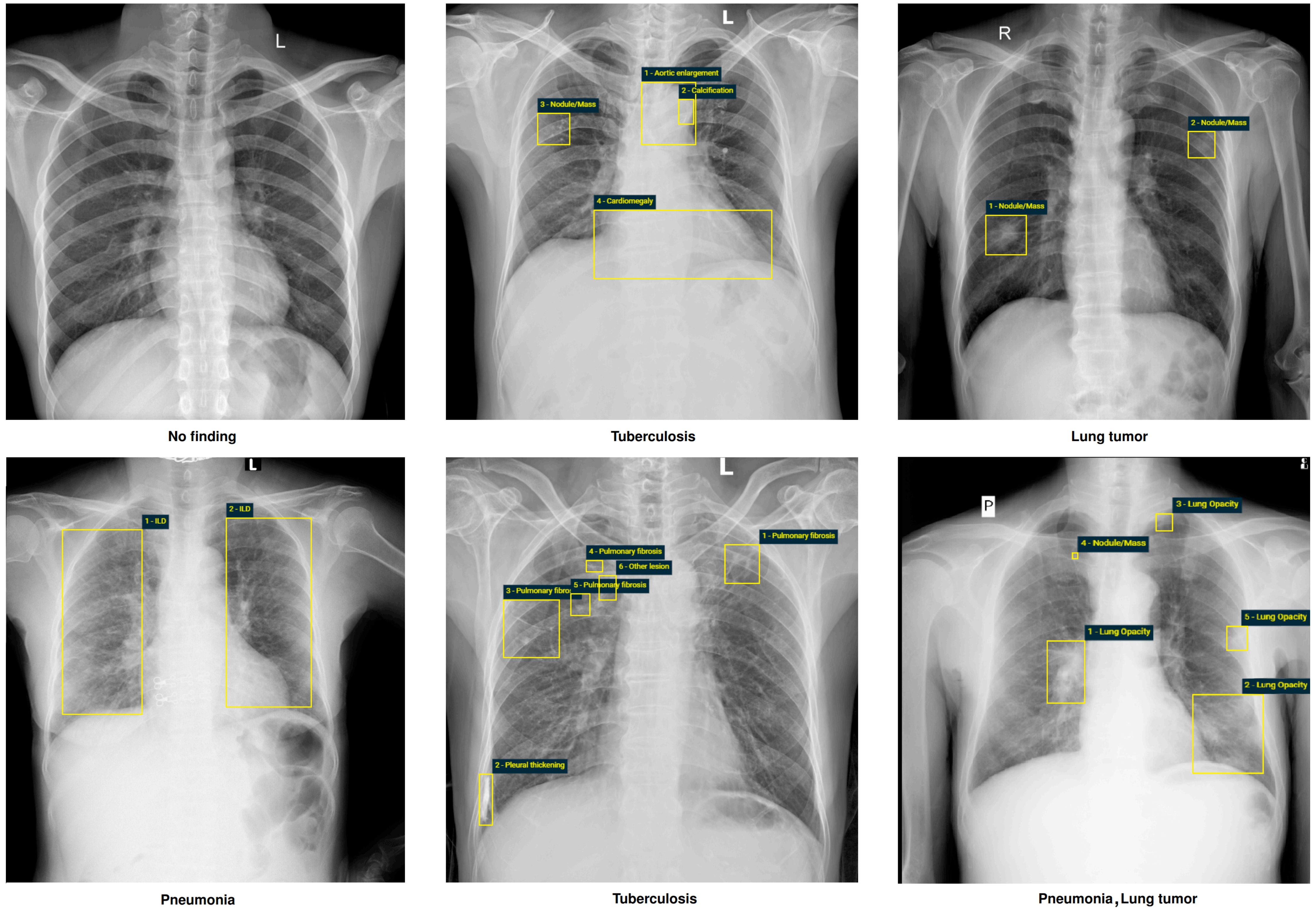
Active Learning Scenario

- Example: you have **10,000 X-ray images**, but you can only pay to have a radiologist **label 50**
  - **Which 50** do you choose?
  - Random?
  - "Hardest-looking" ones?
  - Intentionally diverse sample?



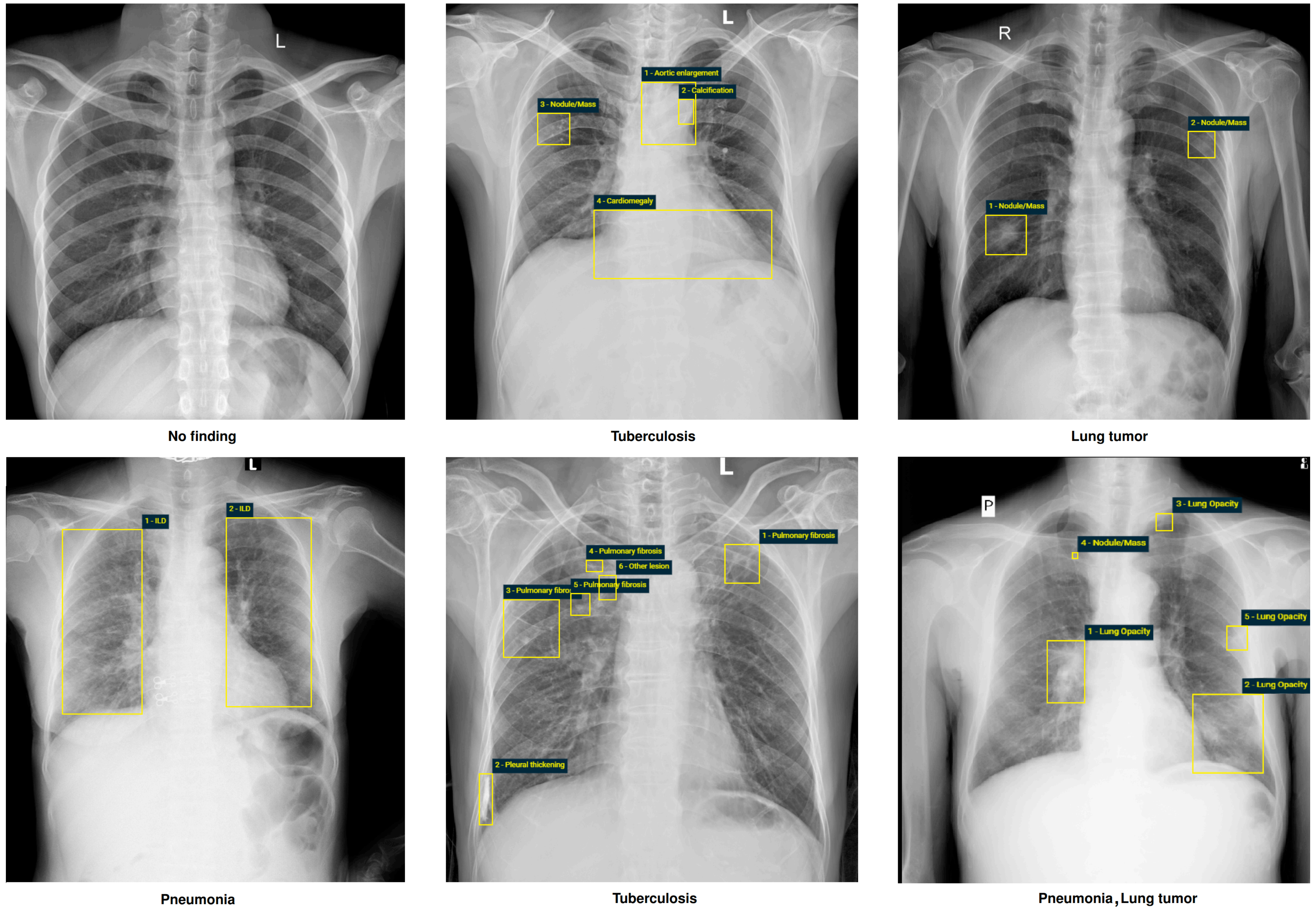
Active Learning Scenario

- Example: you have **10,000 X-ray images**, but you can only pay to have a radiologist **label 50**
  - **Which 50** do you choose?
  - Random?
  - "Hardest-looking" ones?
  - Intentionally diverse sample?
- **Labeling cost** is often a **significant bottleneck** in real-world ML



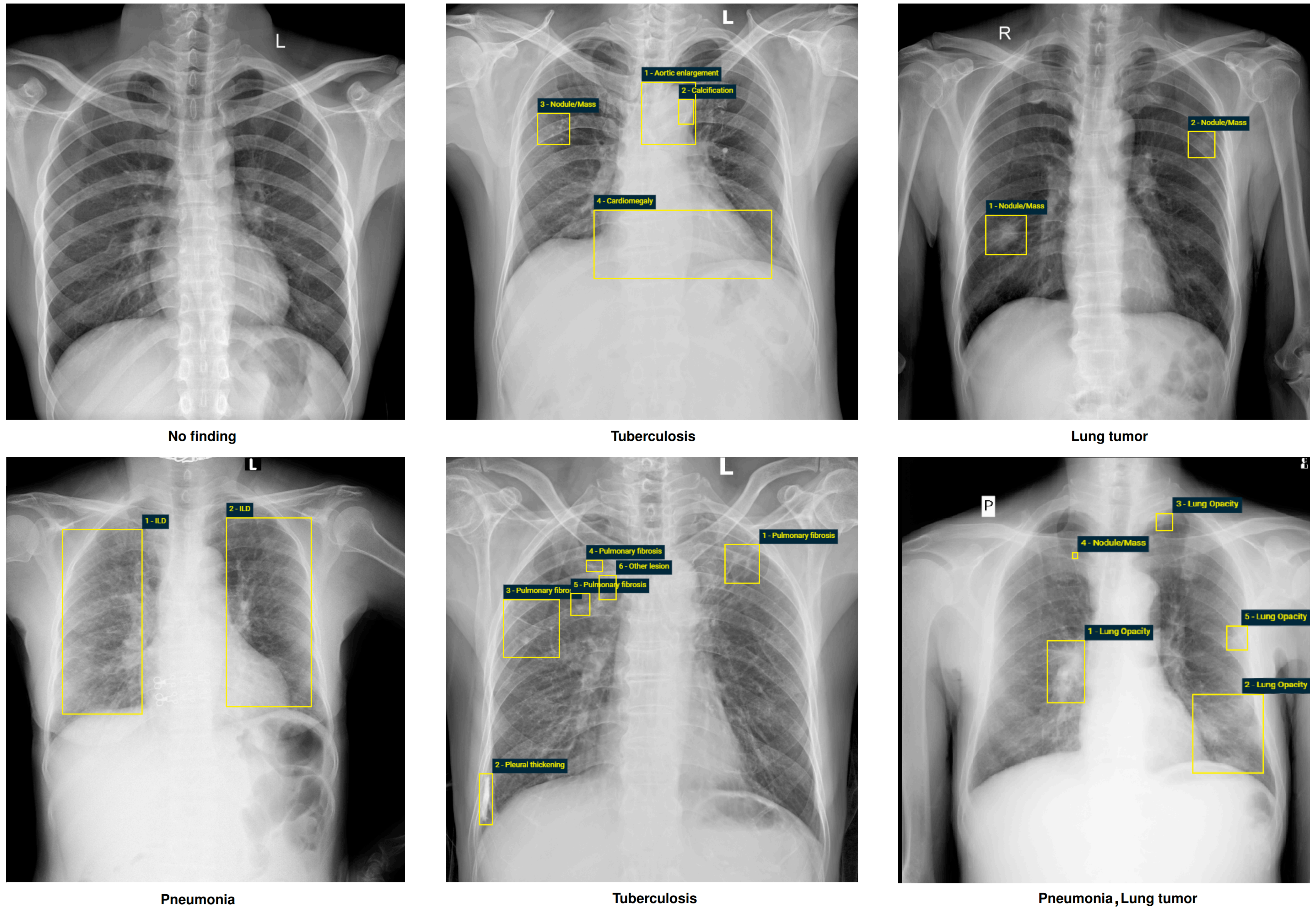
Active Learning Scenario

- Example: you have **10,000 X-ray images**, but you can only pay to have a radiologist **label 50**
  - **Which 50** do you choose?
  - Random?
  - "Hardest-looking" ones?
  - Intentionally diverse sample?
- **Labeling cost** is often a **significant bottleneck** in real-world ML
- **Not** all labels are **equally informative**



Active Learning Scenario

- Example: you have **10,000 X-ray images**, but you can only pay to have a radiologist **label 50**
  - **Which 50** do you choose?
  - Random?
  - "Hardest-looking" ones?
  - Intentionally diverse sample?
- **Labeling cost** is often a **significant bottleneck** in real-world ML
- **Not** all labels are **equally informative**
- Active Learning: **given a fixed annotation budget, which examples should be labeled?**



Active Learning, Formally

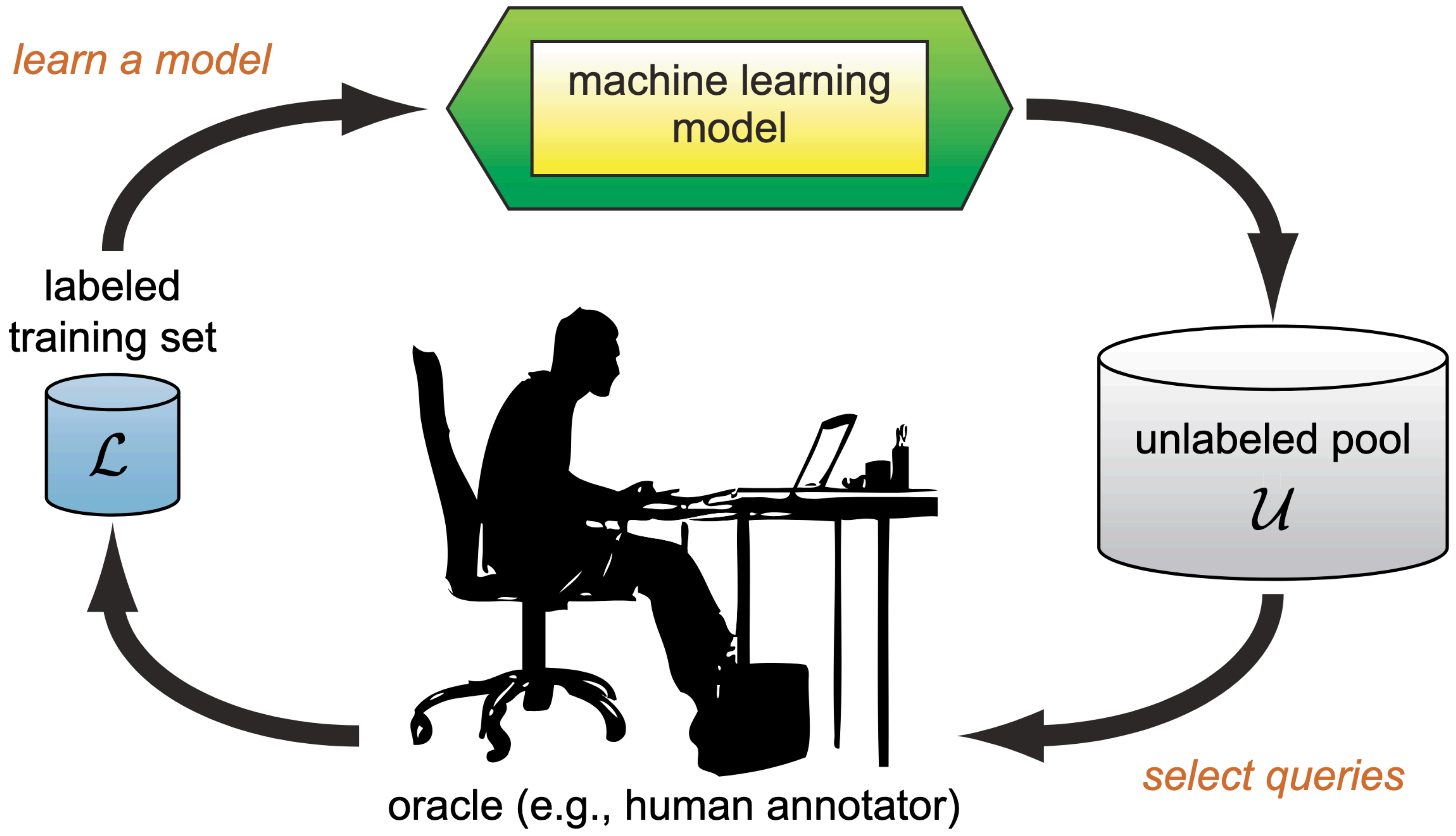


Figure 1: The pool-based active learning cycle.

[Settles, (2010)]

Active Learning, Formally

- **Pool U**: set of **unlabeled examples** (usually large)

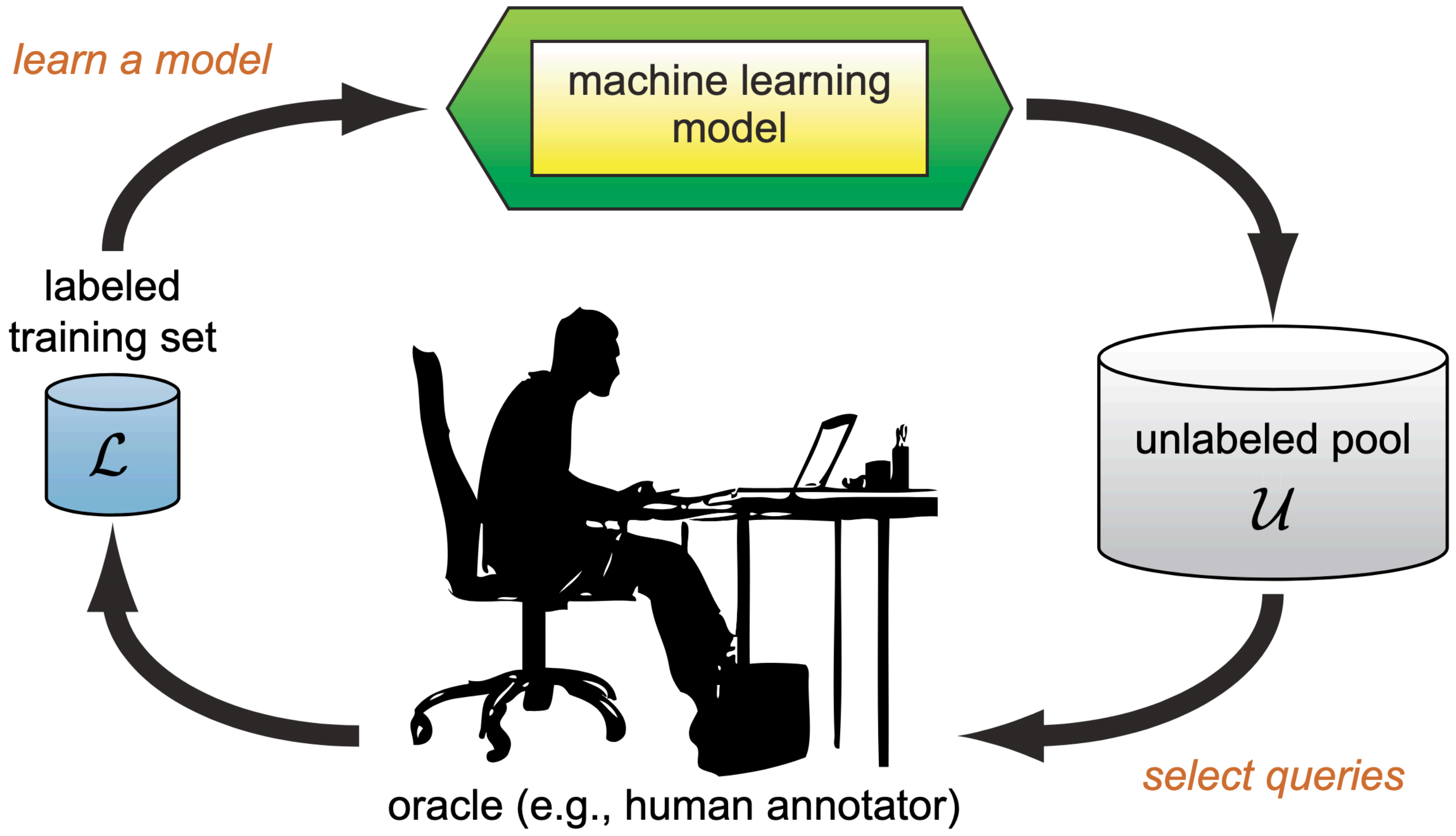


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning, Formally

- **Pool U:** set of **unlabeled examples** (usually large)
- **Labeled set L:** starts **small** or empty

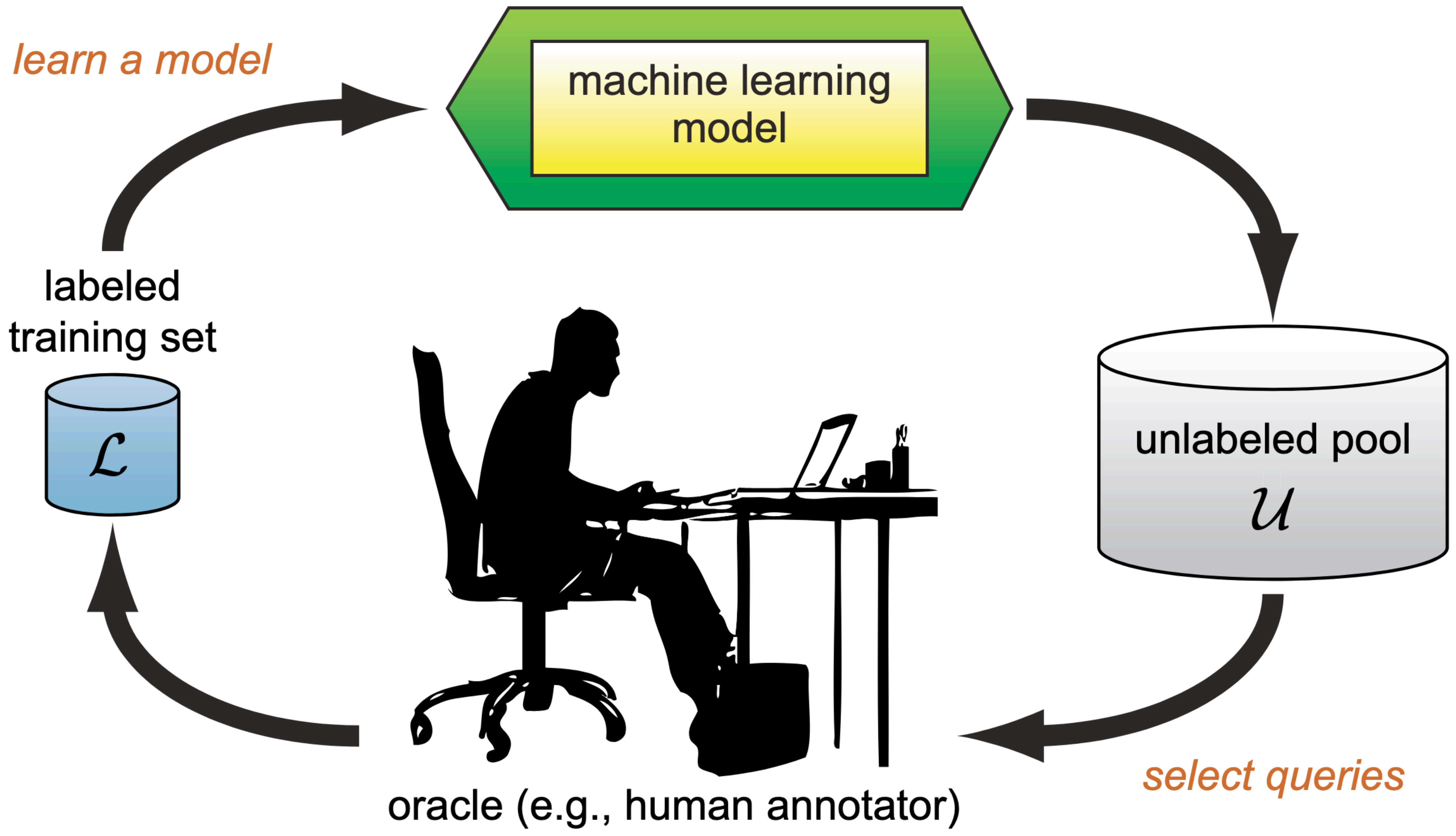


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning, Formally

- **Pool U:** set of **unlabeled examples** (usually large)
- **Labeled set L:** starts **small** or empty
- **Budget b:** how many labels you can afford to annotate ("query")

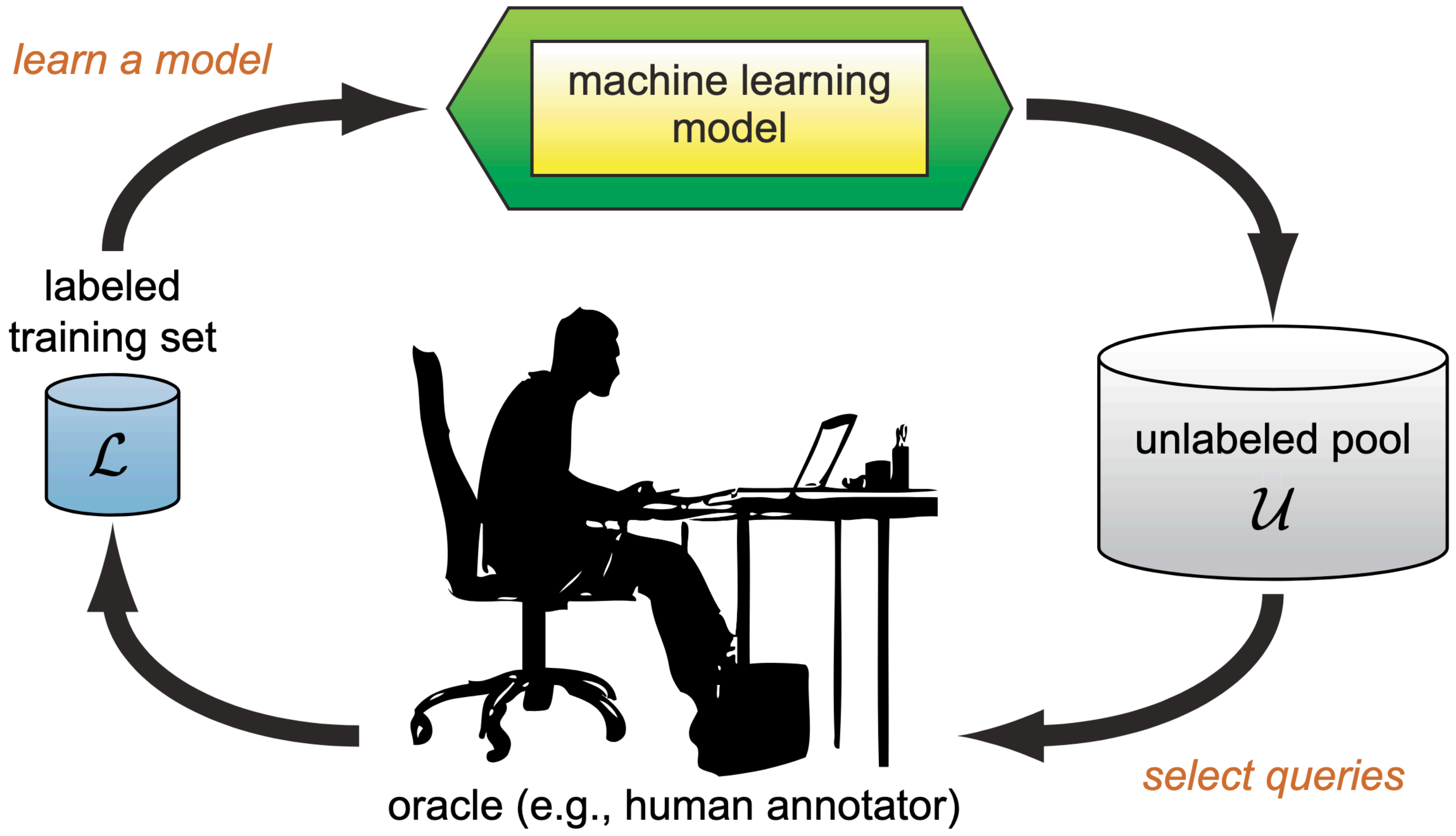


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning, Formally

- **Pool U:** set of **unlabeled examples** (usually large)
- **Labeled set L:** starts **small** or empty
- **Budget b:** how many labels you can afford to annotate ("query")
- **Oracle:** the person/thing that **provides the labels**

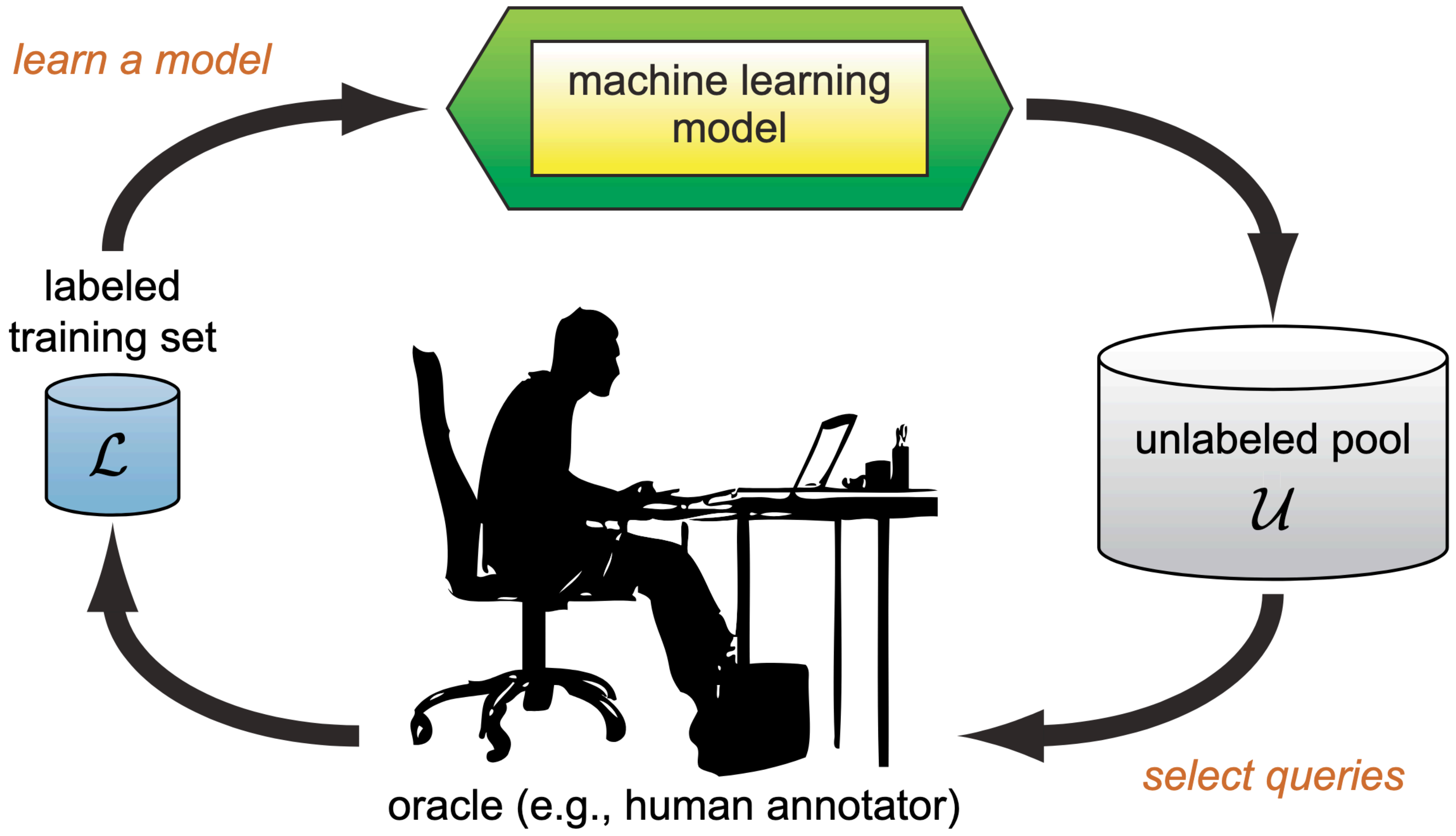


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning, Formally

- **Pool U:** set of **unlabeled examples** (usually large)
- **Labeled set L:** starts **small** or empty
- **Budget b:** how many labels you can afford to annotate ("query")
- **Oracle:** the person/thing that **provides the labels**
- **Acquisition function $a(x)$:** scores unlabeled examples by **how "useful"** a label would be

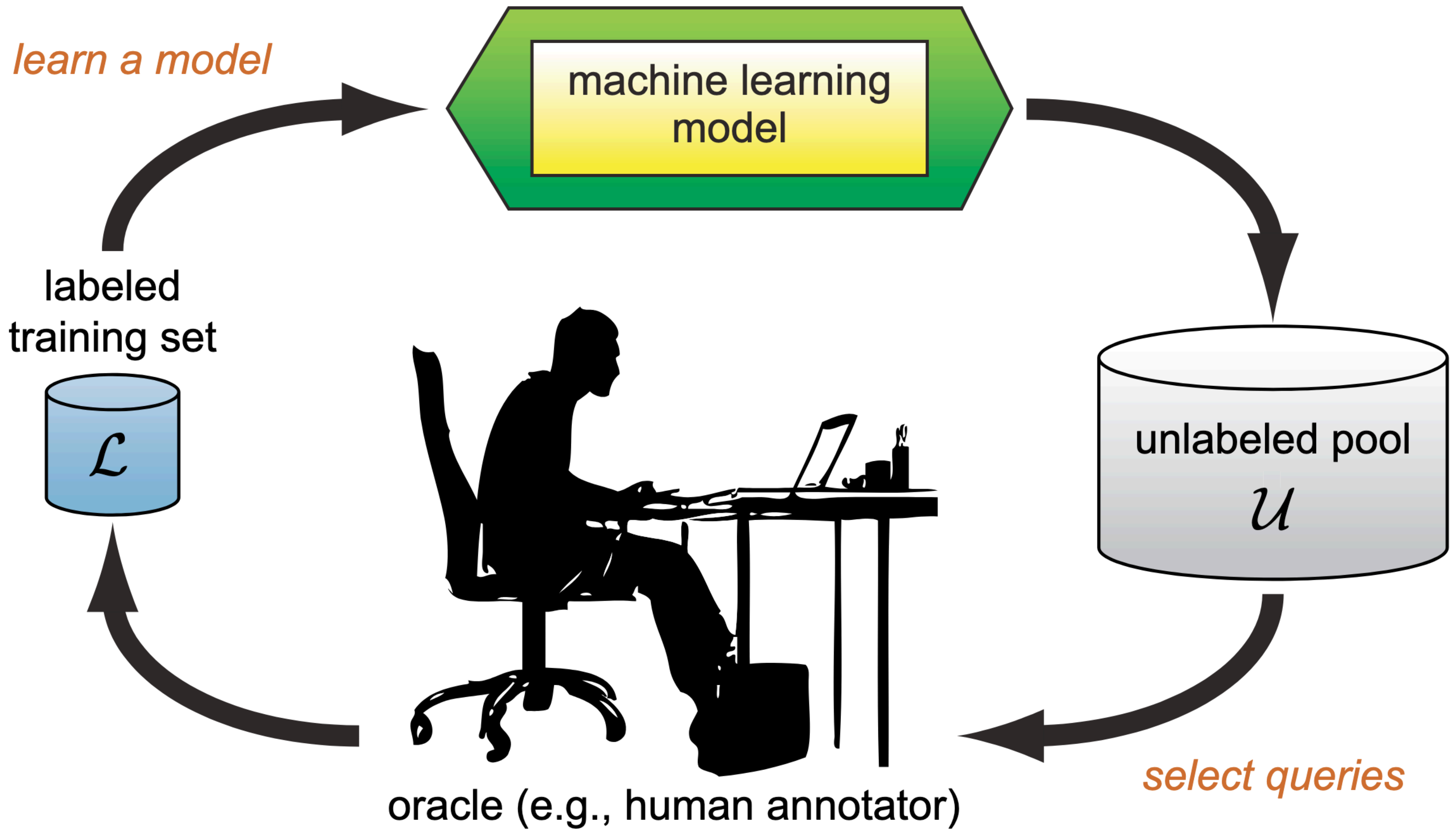


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning Loop

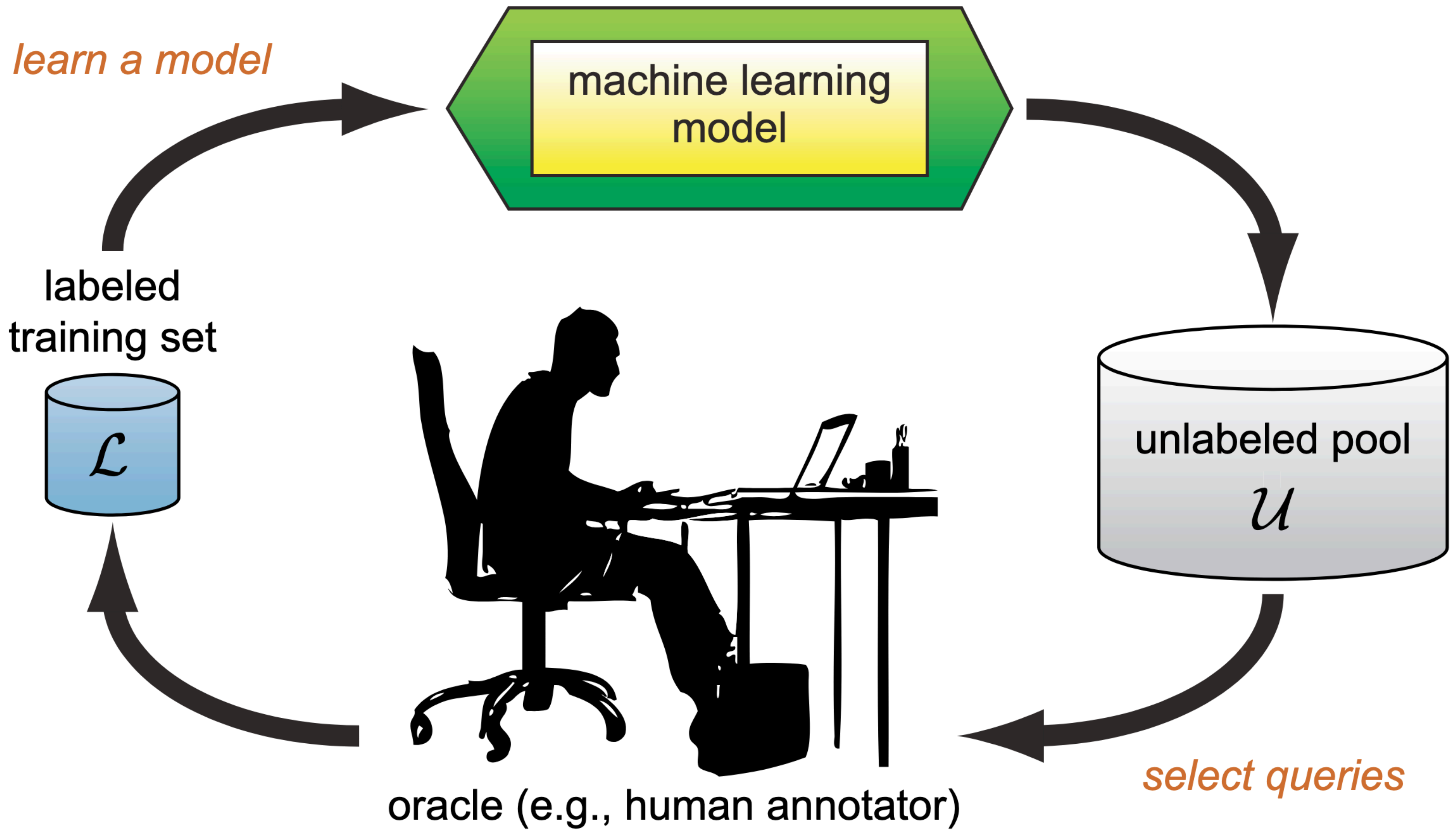


Figure 1: The pool-based active learning cycle.

[Settles, (2010)]

Active Learning Loop

1. Train on **current labeled set** L

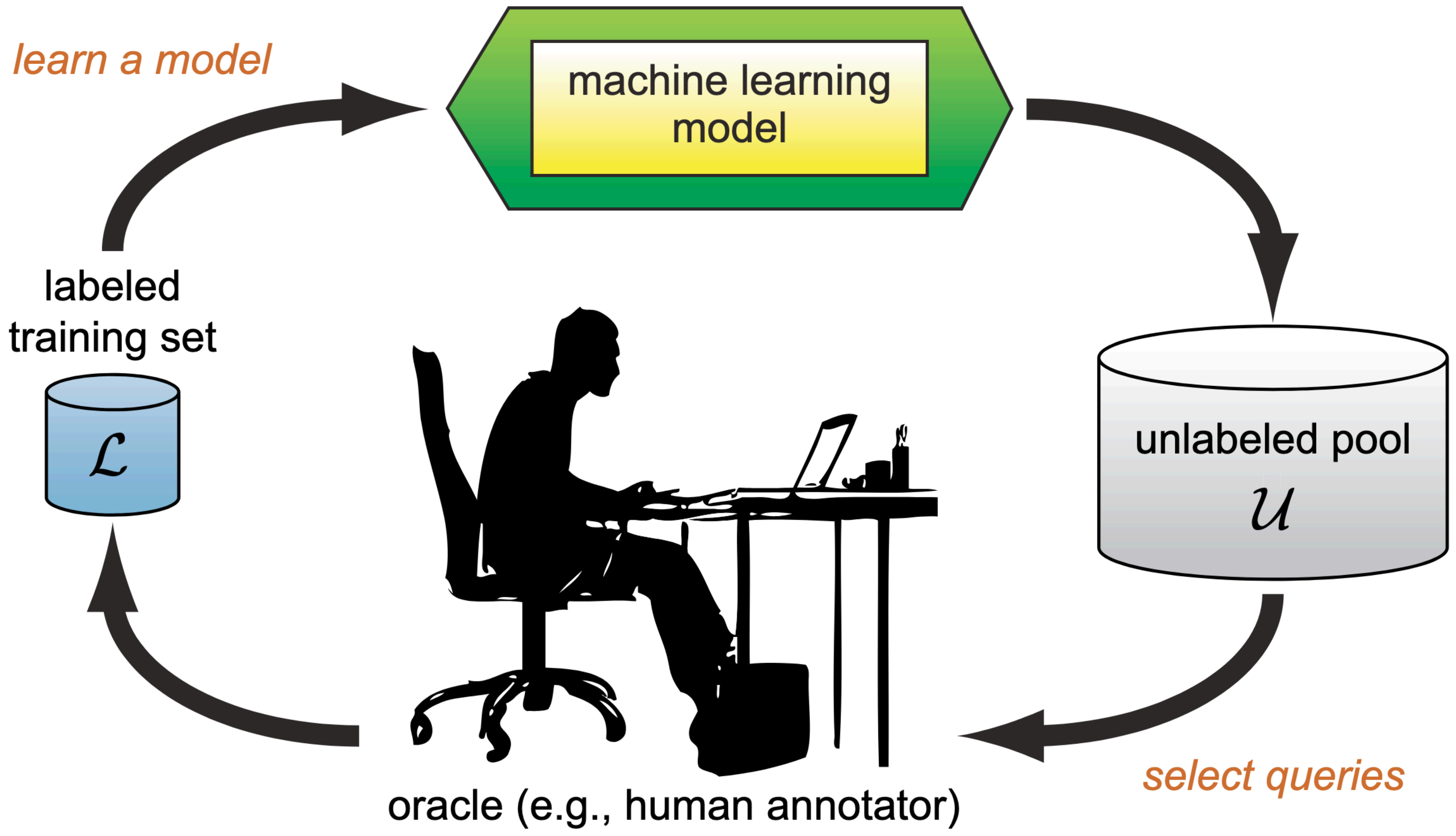


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning Loop

1. Train on **current labeled set** L
2. Apply **acquisition function** to **rank examples** in U

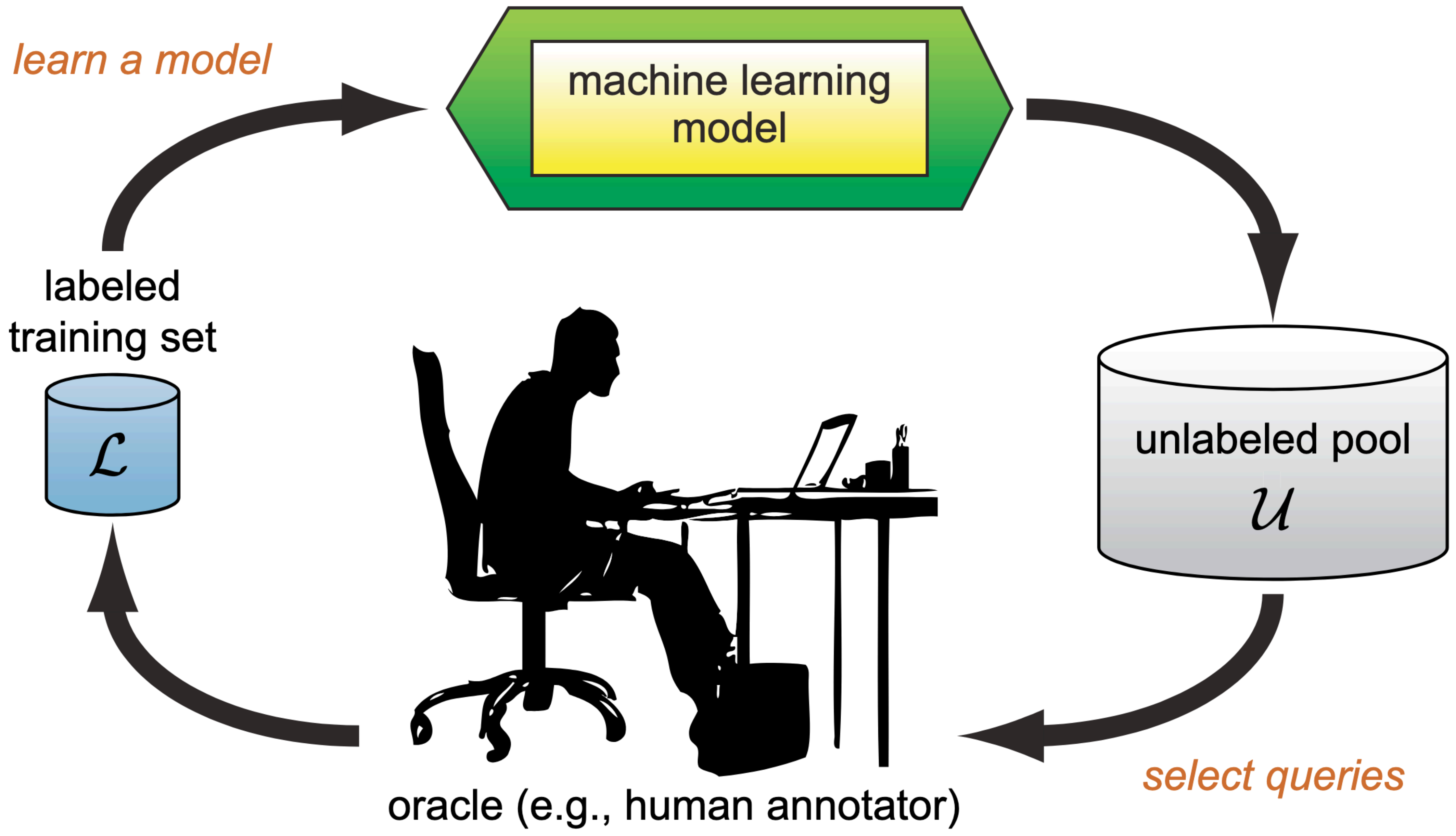


Figure 1: The pool-based active learning cycle.

[Settles, (2010)]

Active Learning Loop

1. Train on **current labeled set** L
2. Apply **acquisition function** to **rank examples** in U
3. Select **top-k** examples and **query the oracle** for labels

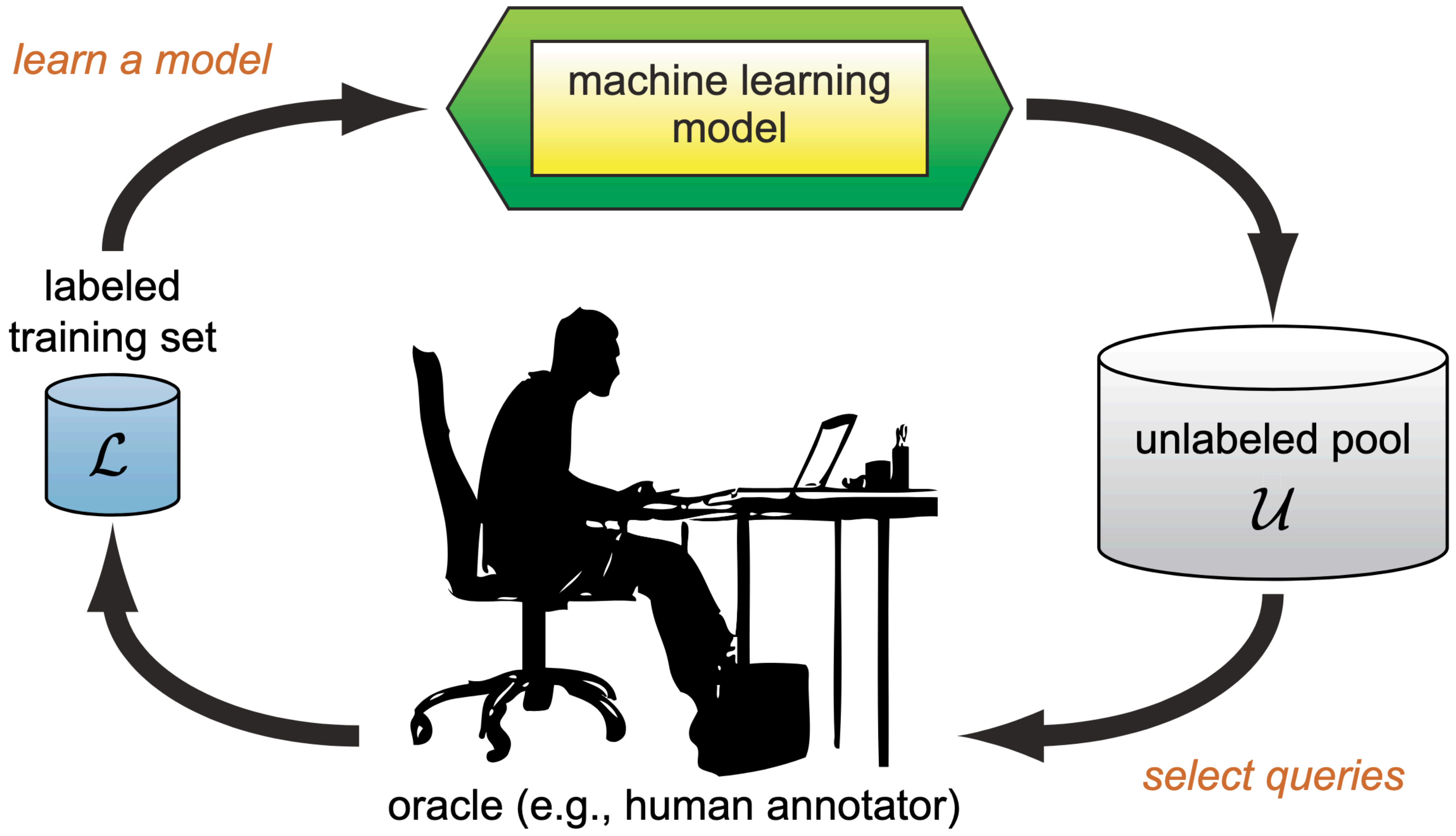


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning Loop

1. Train on **current labeled set** L
2. Apply **acquisition function** to **rank examples** in U
3. Select **top-k** examples and **query the oracle** for labels
4. Add **newly-labeled examples** to L

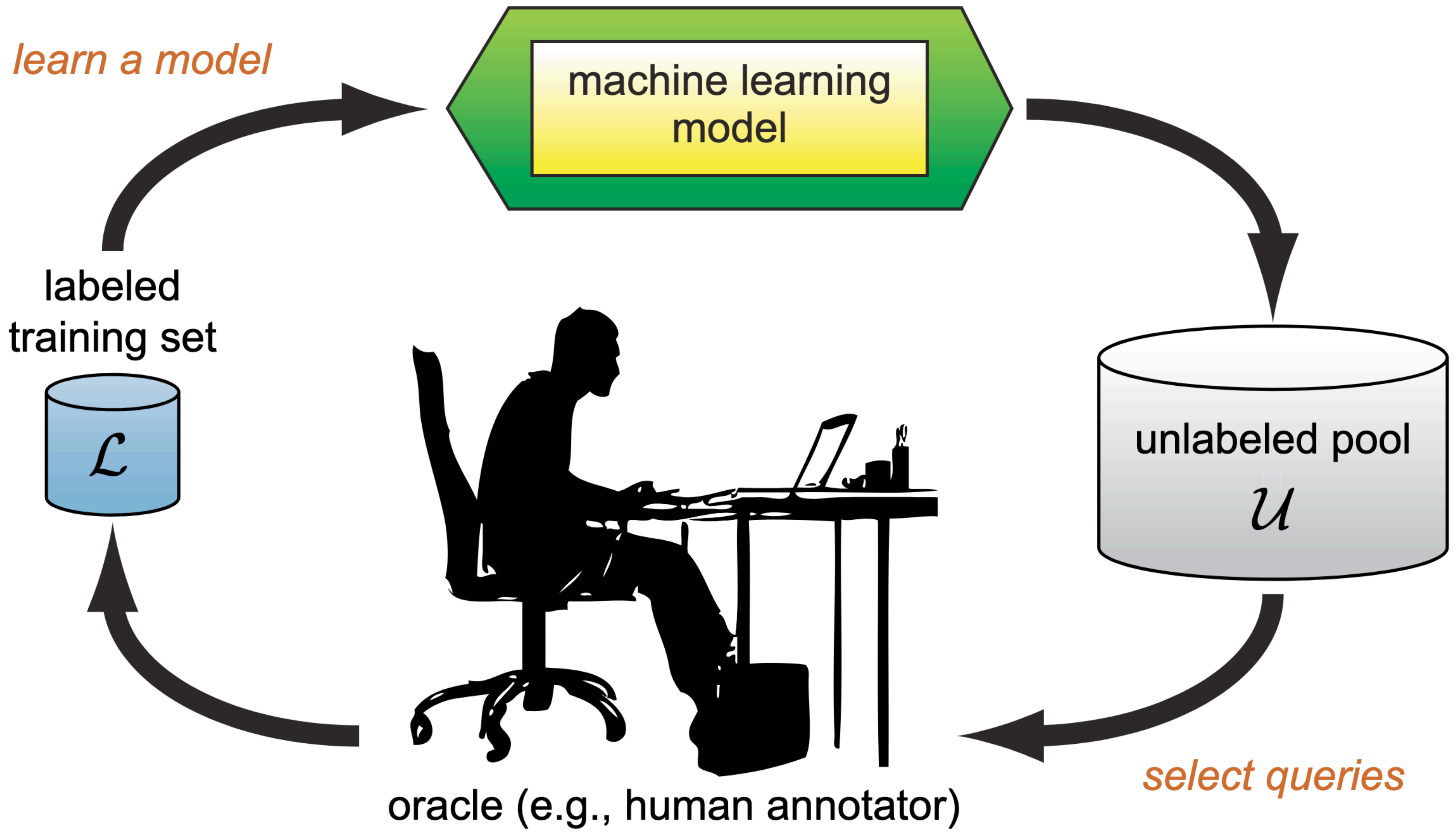


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Active Learning Loop

1. Train on **current labeled set** L
2. Apply **acquisition function** to **rank examples** in U
3. Select **top-k** examples and **query the oracle** for labels
4. Add **newly-labeled examples** to L
5. Repeat until **budget exhausted** (or performance saturated)

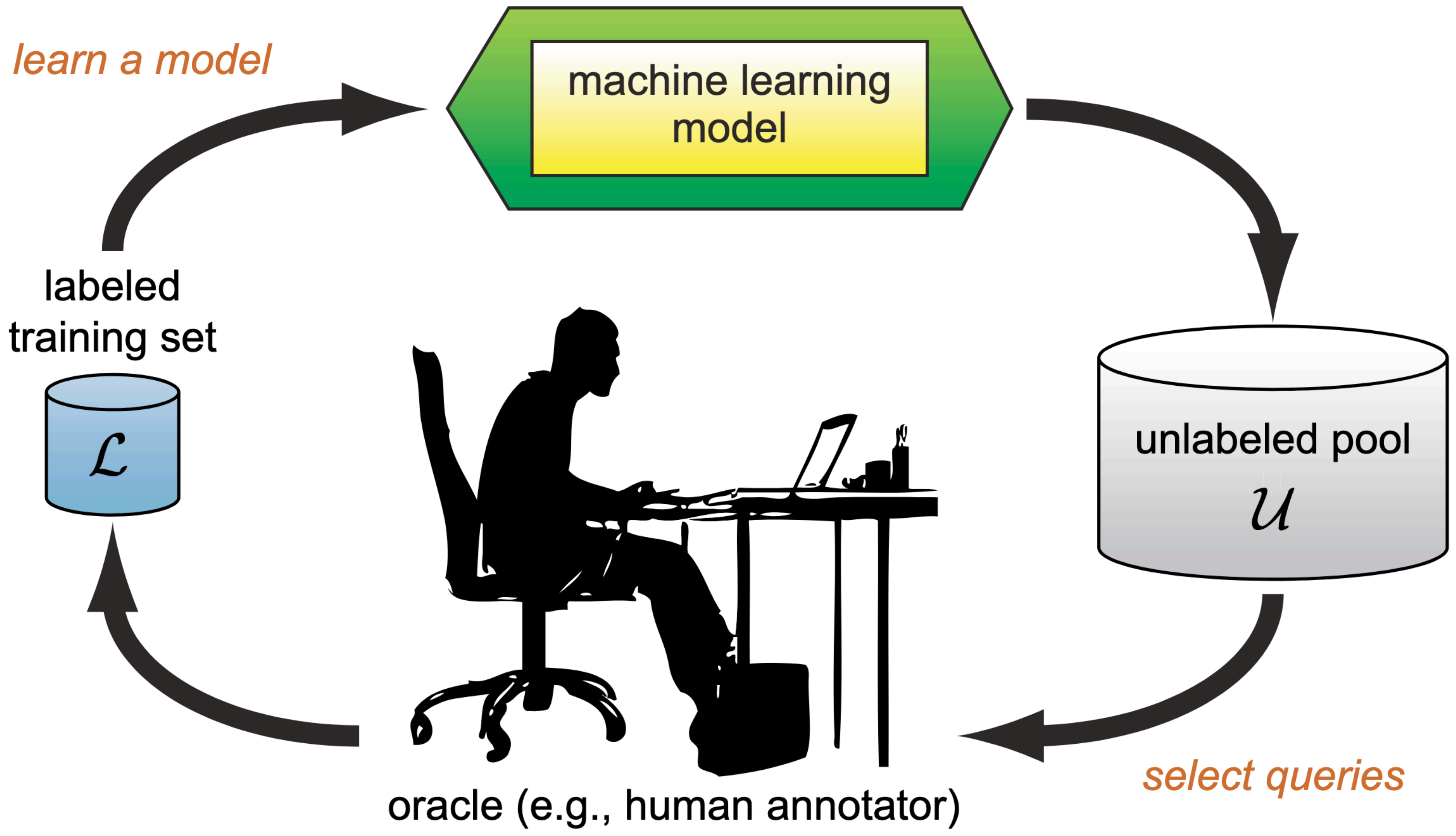


Figure 1: The pool-based active learning cycle.

Settles, (2010)

Connection to Semi-supervised

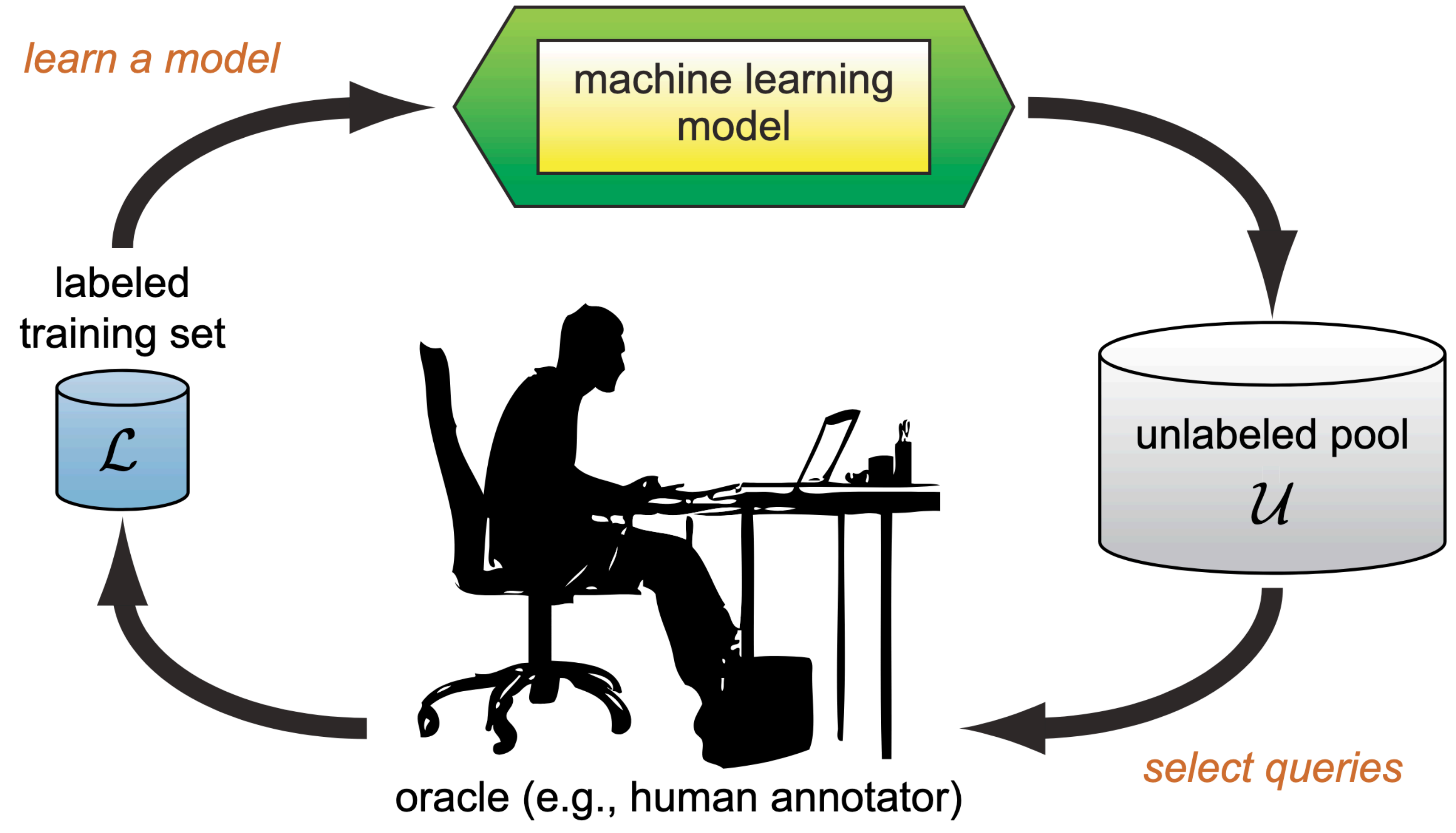
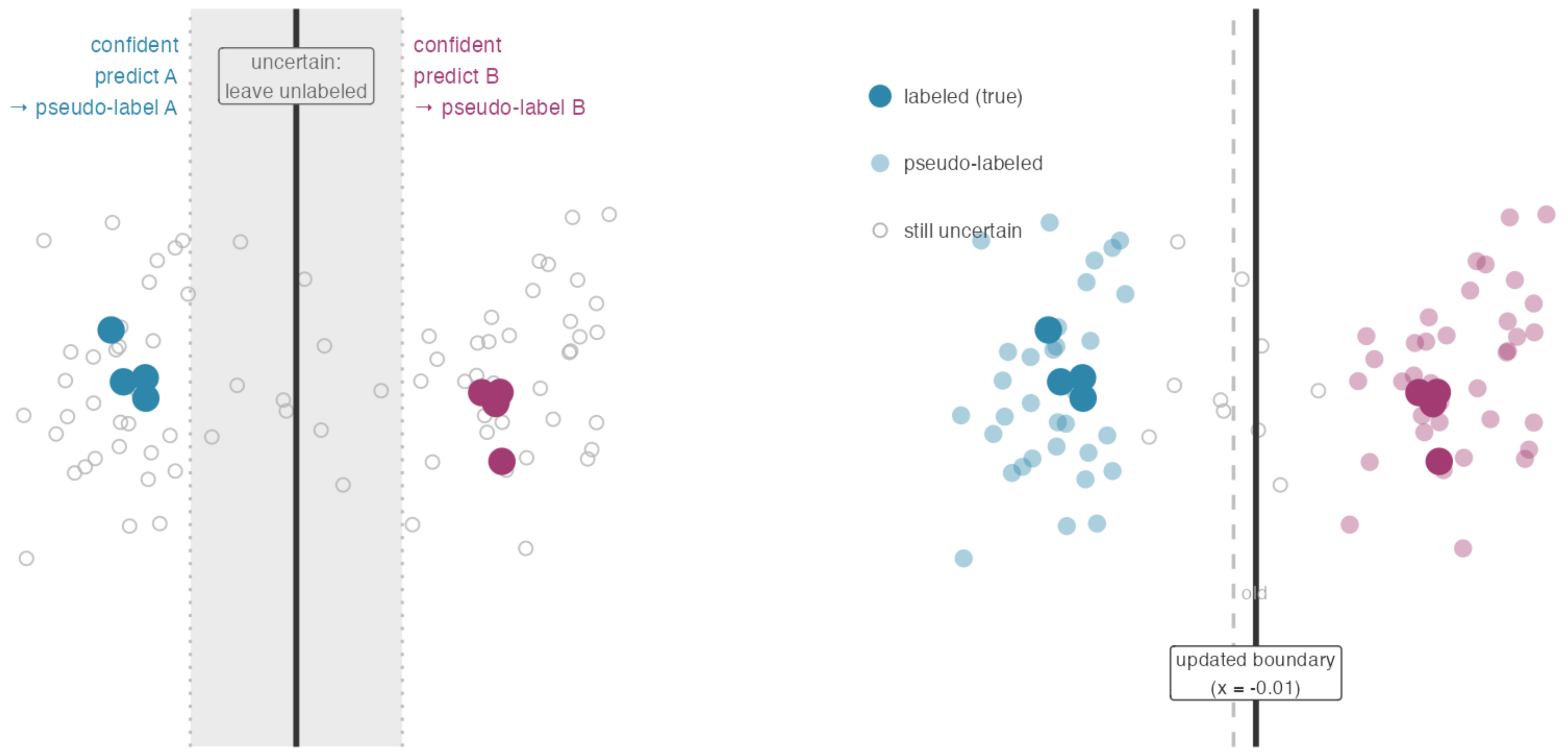


Figure 1: The pool-based active learning cycle.



Connection to Semi-supervised

- Semi-supervised: "I have some labels - how do I **use unlabeled data** to make them **go further**?"

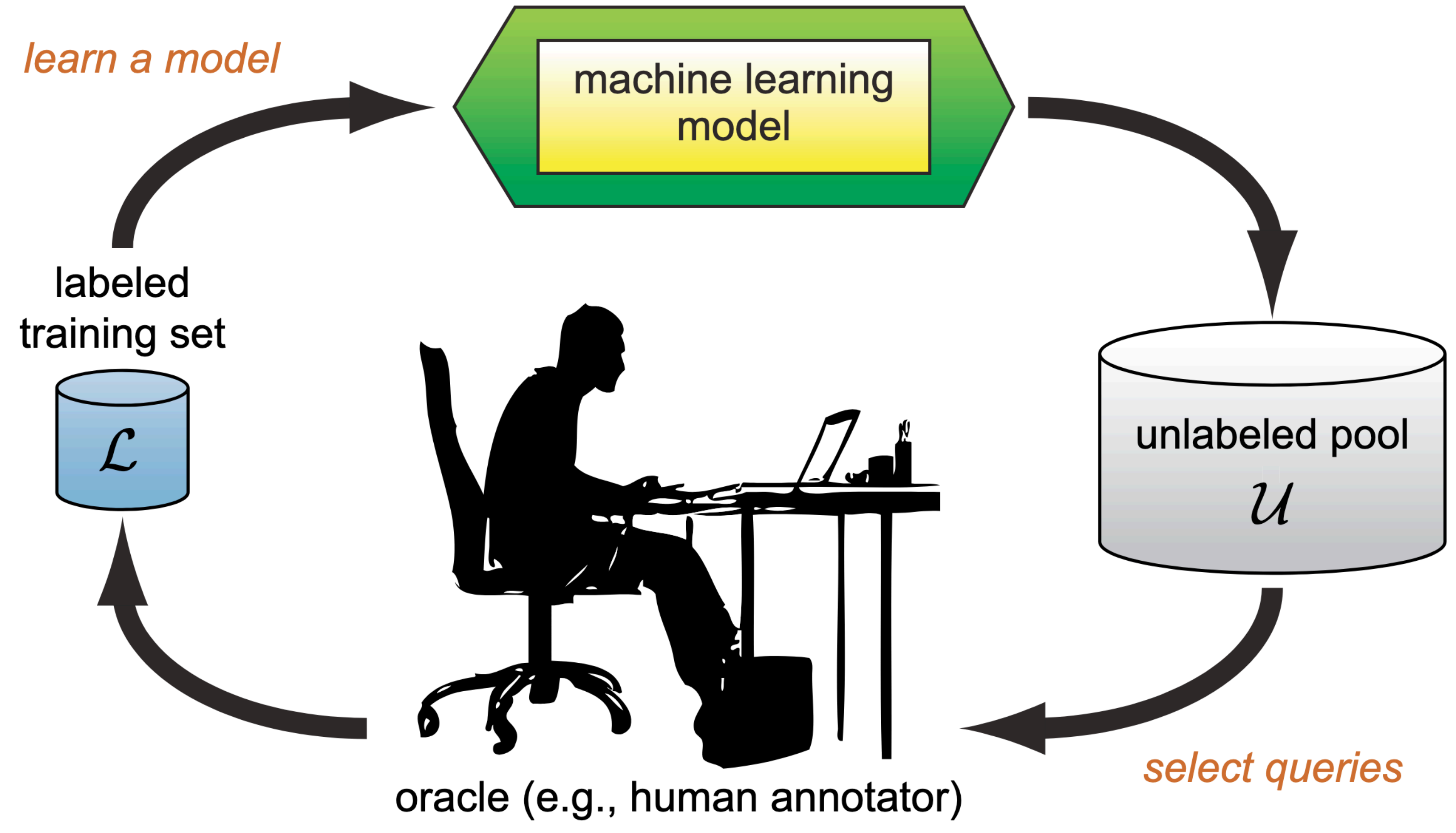
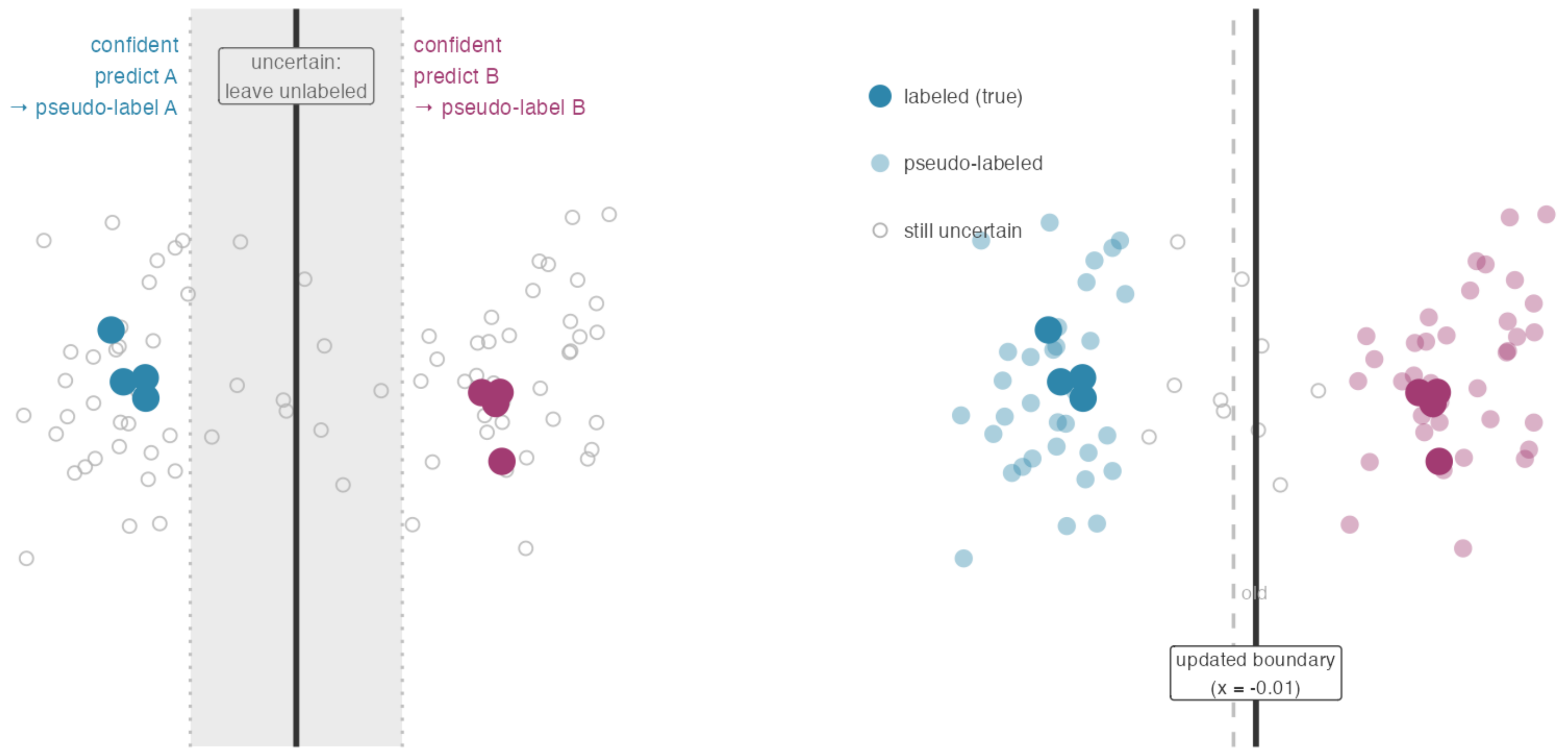


Figure 1: The pool-based active learning cycle.



Connection to Semi-supervised

- Semi-supervised: "I have some labels - how do I **use unlabeled data** to make them **go further**?"
- Active Learning: "How do I **pick the most useful labels** to acquire?"

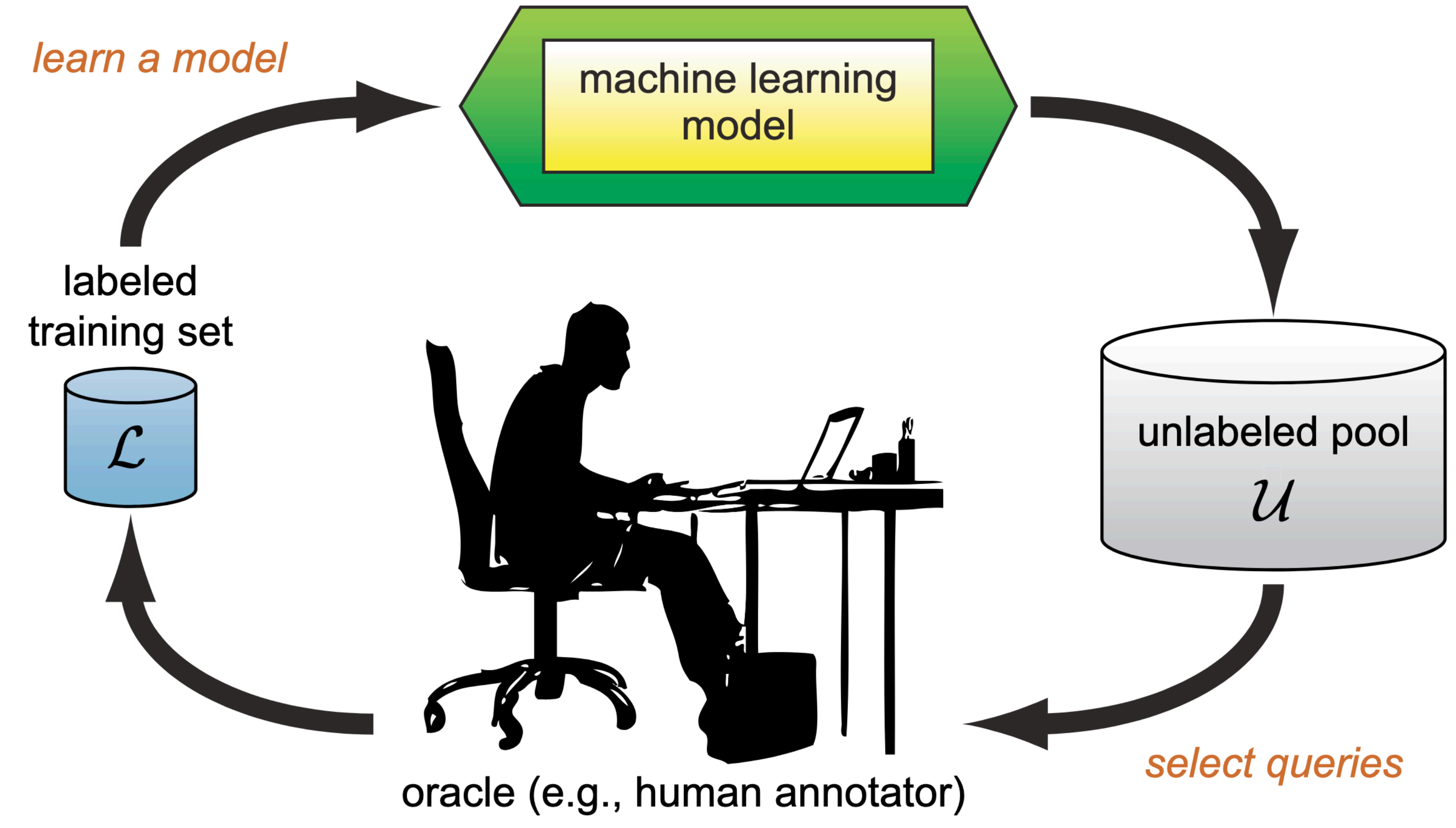
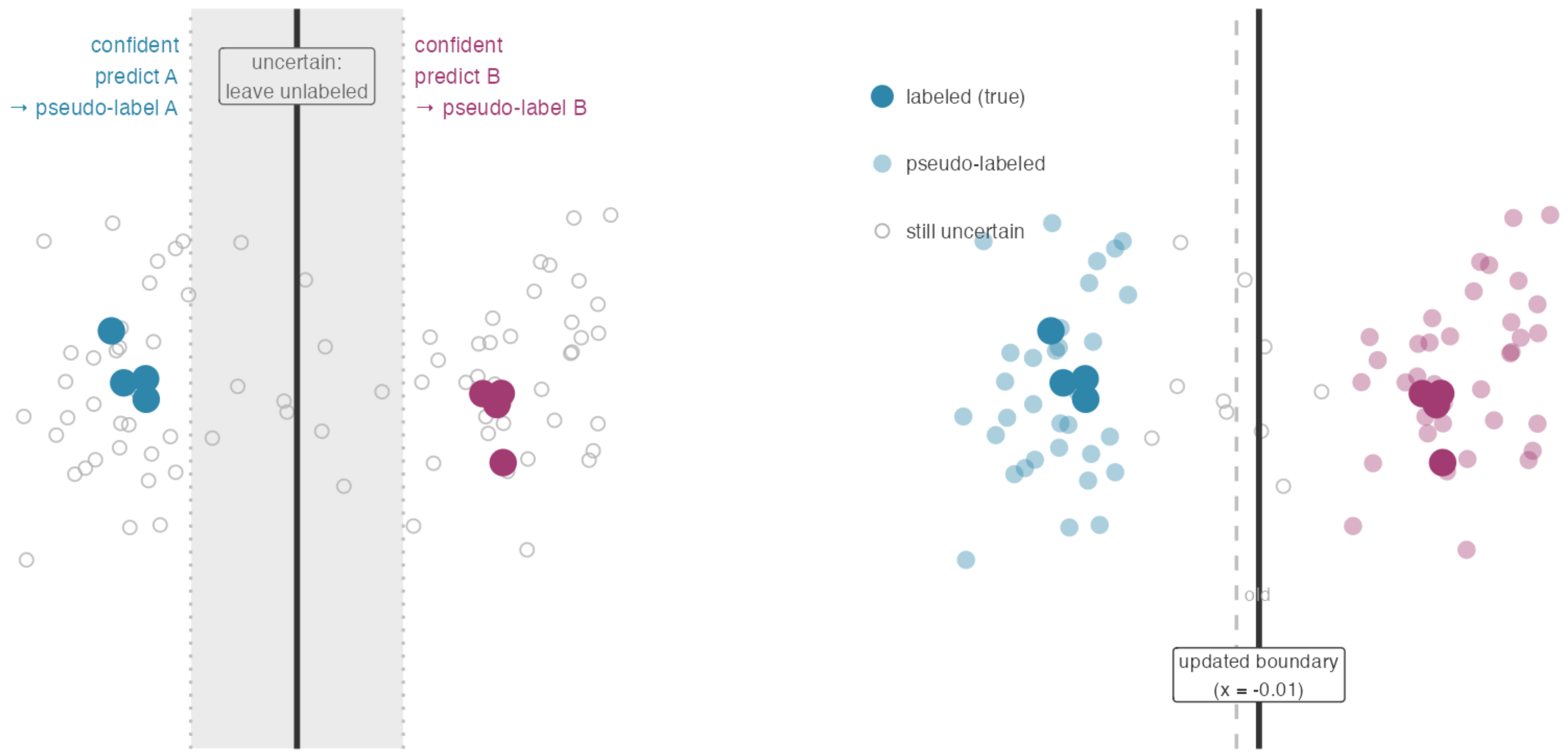


Figure 1: The pool-based active learning cycle.



Connection to Semi-supervised

- Semi-supervised: "I have some labels - how do I **use unlabeled data** to make them **go further**?"
- Active Learning: "How do I **pick the most useful labels** to acquire?"
- These are **completely complementary** - they can be combined easily

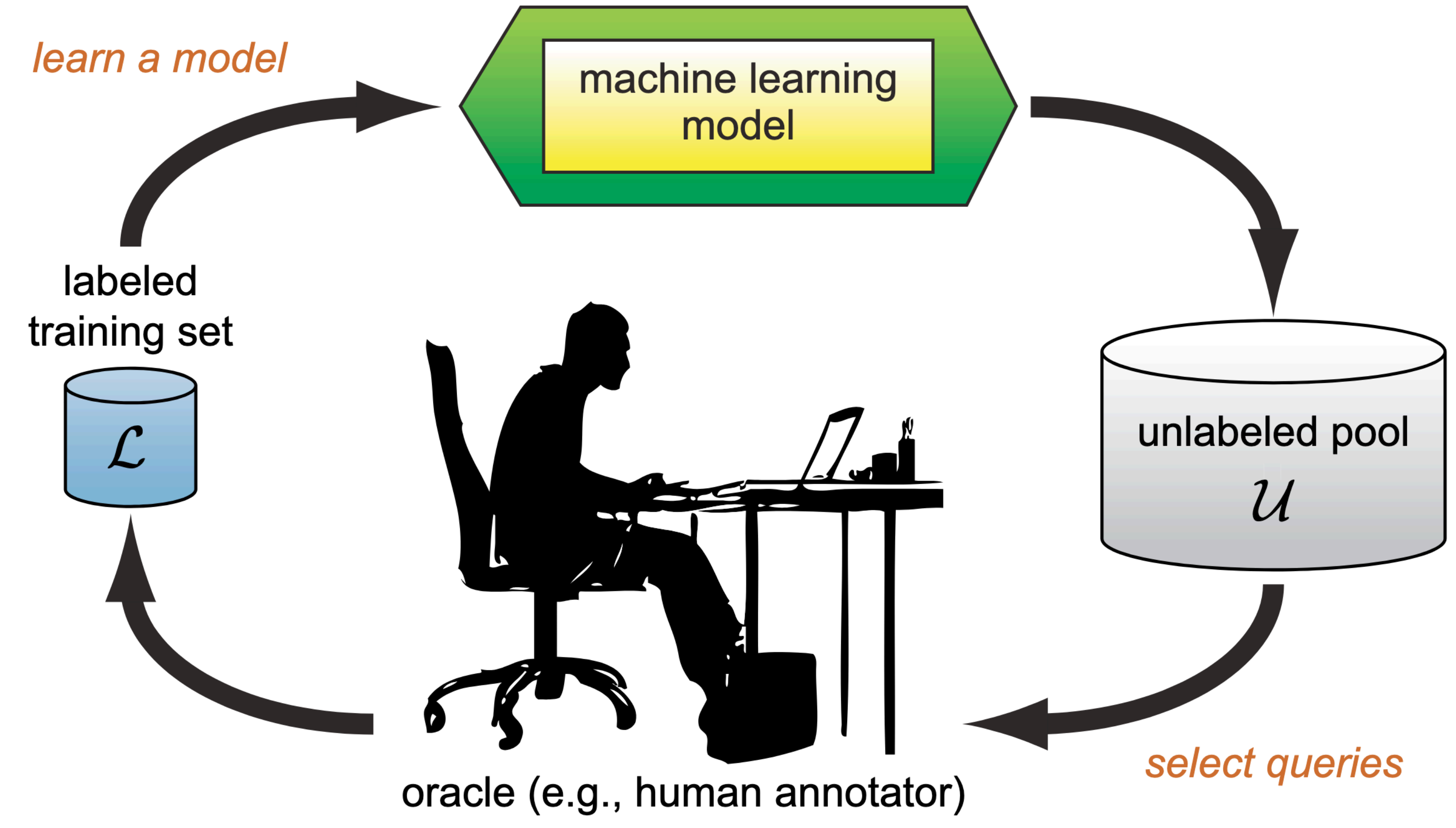
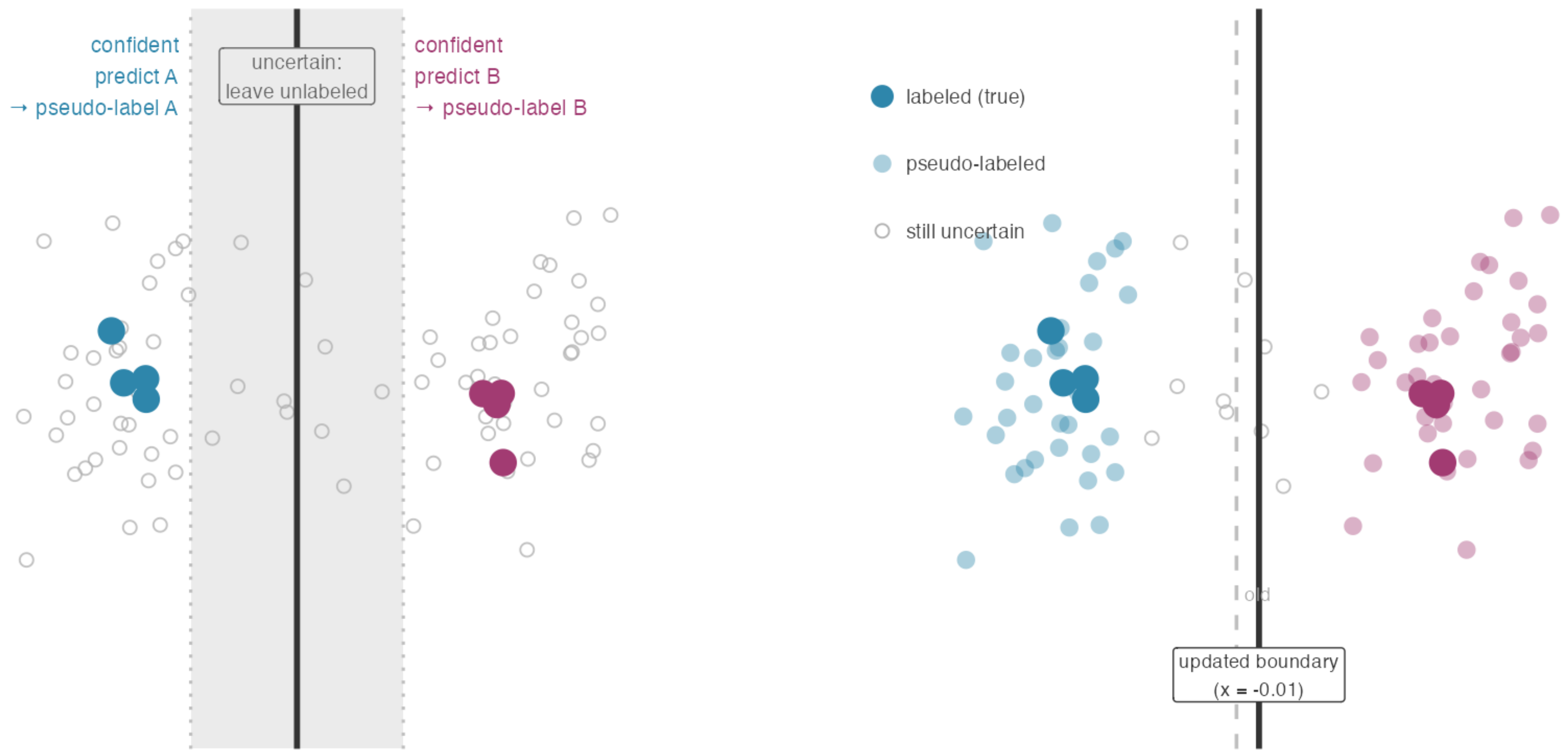


Figure 1: The pool-based active learning cycle.



Paradigms

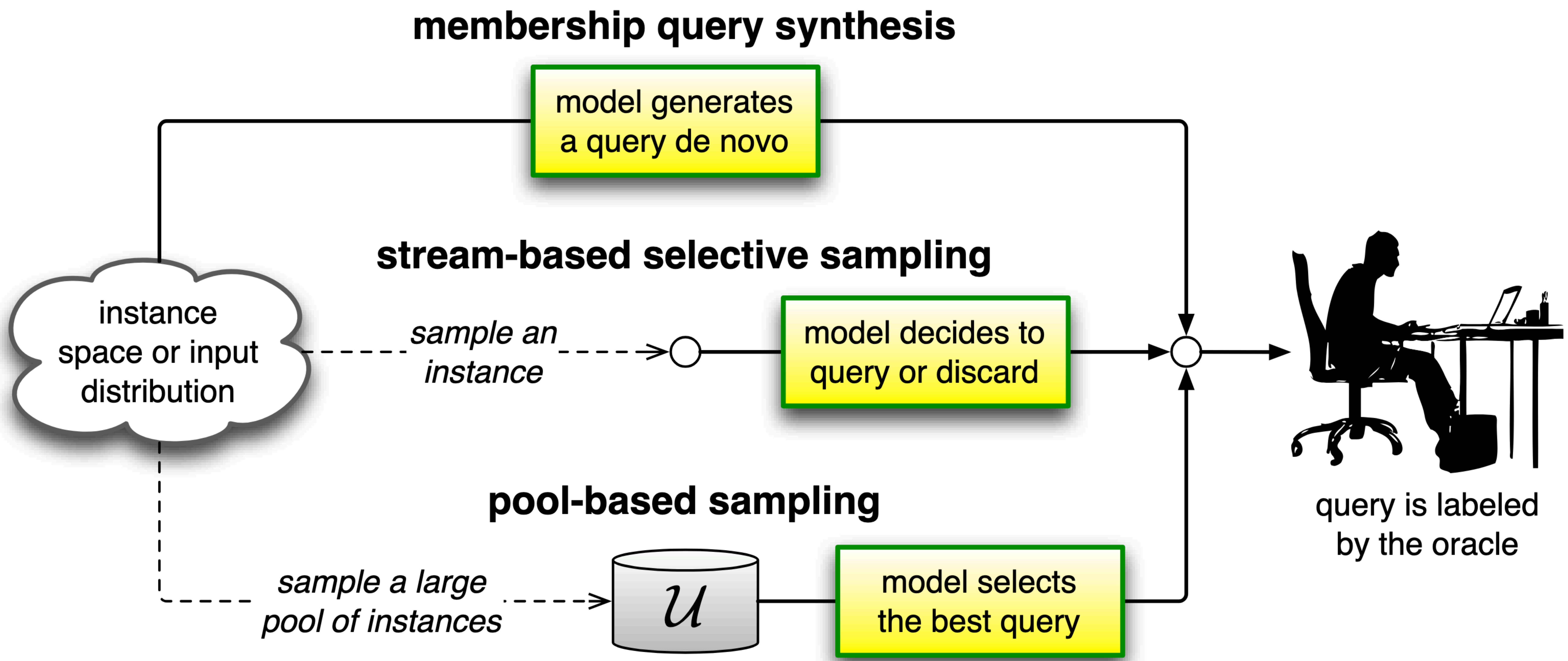
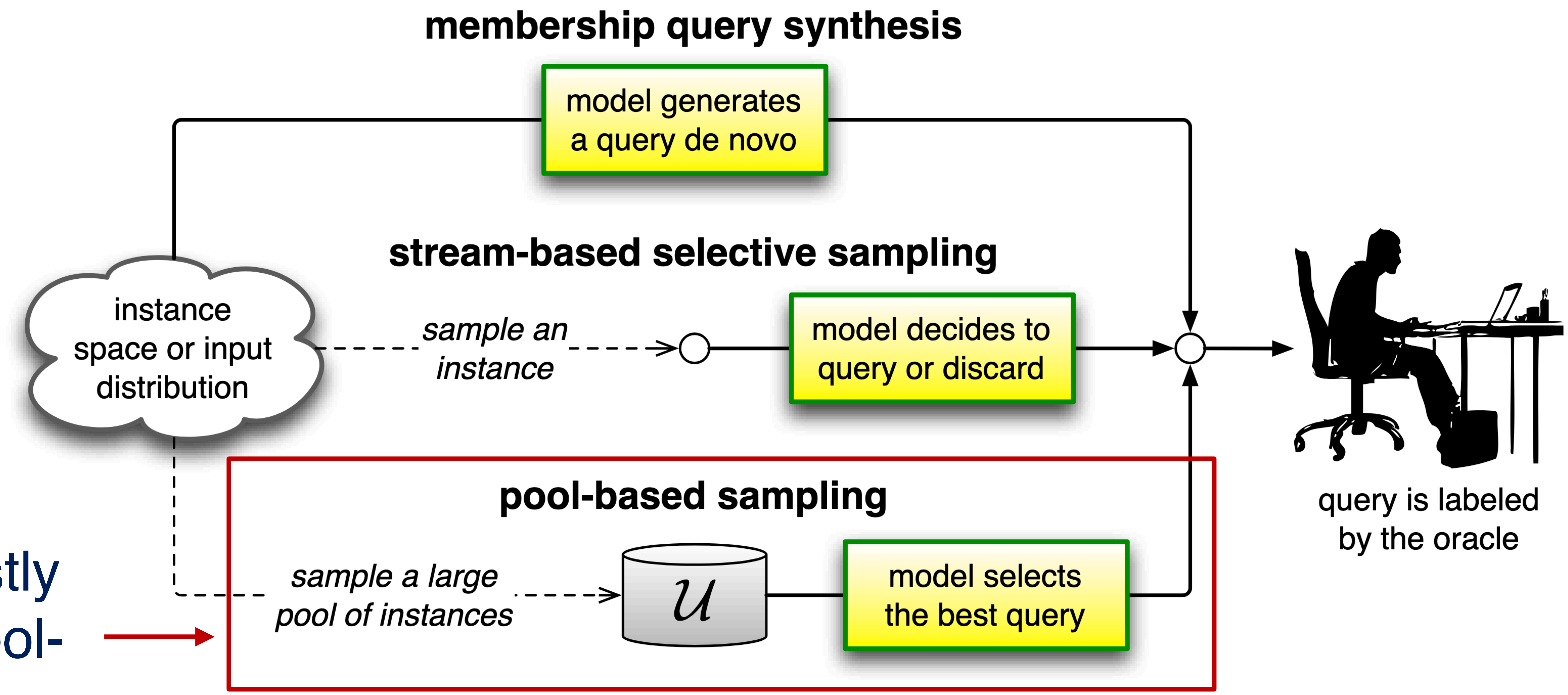


Figure 4: Diagram illustrating the three main active learning scenarios.

Settles, (2010)

Paradigms



We'll mostly look at pool-based!

Figure 4: Diagram illustrating the three main active learning scenarios.

Settles, (2010)

Uncertainty Sampling

Uncertainty Sampling: Main Idea

Uncertainty Sampling: Main Idea

- Label examples the model is **most uncertain** about

Uncertainty Sampling: Main Idea

- Label examples the model is **most uncertain** about
- If the model is **already confident**, there is **little else to learn**

Uncertainty Sampling: Main Idea

- Label examples the model is **most uncertain** about
- If the model is **already confident**, there is **little else to learn**
- Measuring uncertainty

Uncertainty Sampling: Main Idea

- Label examples the model is **most uncertain** about
- If the model is **already confident**, there is **little else to learn**
- Measuring uncertainty
  - **Least Confidence:** how unsure is the **best classification guess?**

$$a_{\mathrm{LC}}(x) = 1 - P(\hat{y} \mid x)$$

Uncertainty Sampling: Main Idea

- Label examples the model is **most uncertain** about
- If the model is **already confident**, there is **little else to learn**
- Measuring uncertainty
  - **Least Confidence:** how unsure is the **best classification guess?**
  - **Margin:** how close are the **top two predictions?** Small margin → the model is **torn between two classes**

$$a_{\mathrm{LC}}(x) = 1 - P(\hat{y} \mid x)$$

$$a_{\mathrm{margin}}(x) = 1 - \big(P(\hat{y}_1 \mid x) - P(\hat{y}_2 \mid x)\big)$$

Uncertainty Sampling: Main Idea

- Label examples the model is **most uncertain** about
- If the model is **already confident**, there is **little else to learn**
- Measuring uncertainty
  - **Least Confidence:** how unsure is the **best classification guess?**
  - **Margin:** how close are the **top two predictions?** Small margin → the model is **torn between two classes**
  - **Entropy:** how **spread out** is the prediction distribution? (high entropy = more uncertainty)

$$a_{\mathrm{LC}}(x) = 1 - P(\hat{y} \mid x)$$

$$a_{\mathrm{margin}}(x) = 1 - \left(P(\hat{y}_1 \mid x) - P(\hat{y}_2 \mid x)\right)$$

$$a_{\mathrm{entropy}}(x) = -\sum_{y} P(y \mid x) \log P(y \mid x)$$

Uncertainty Sampling Example

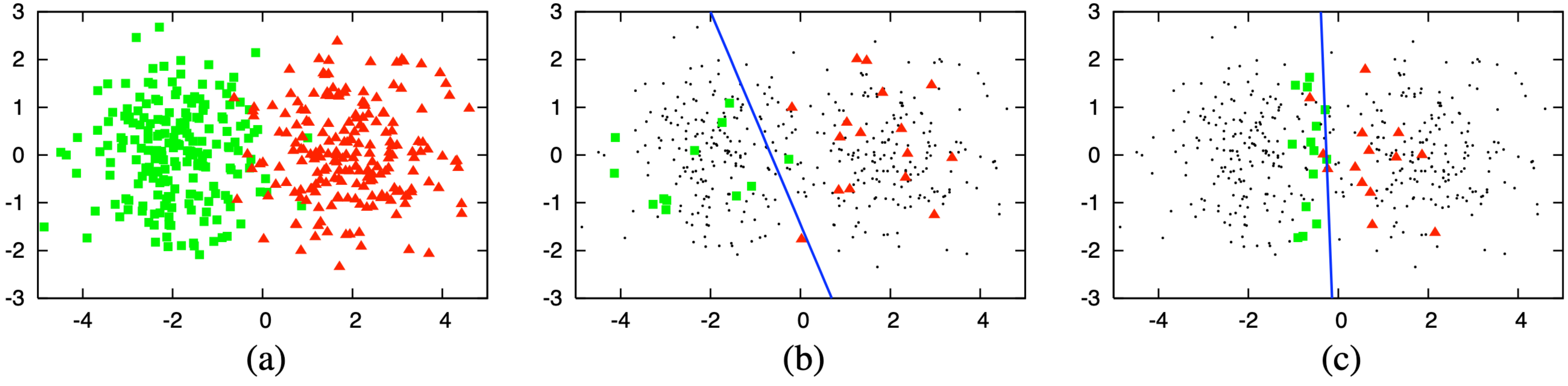
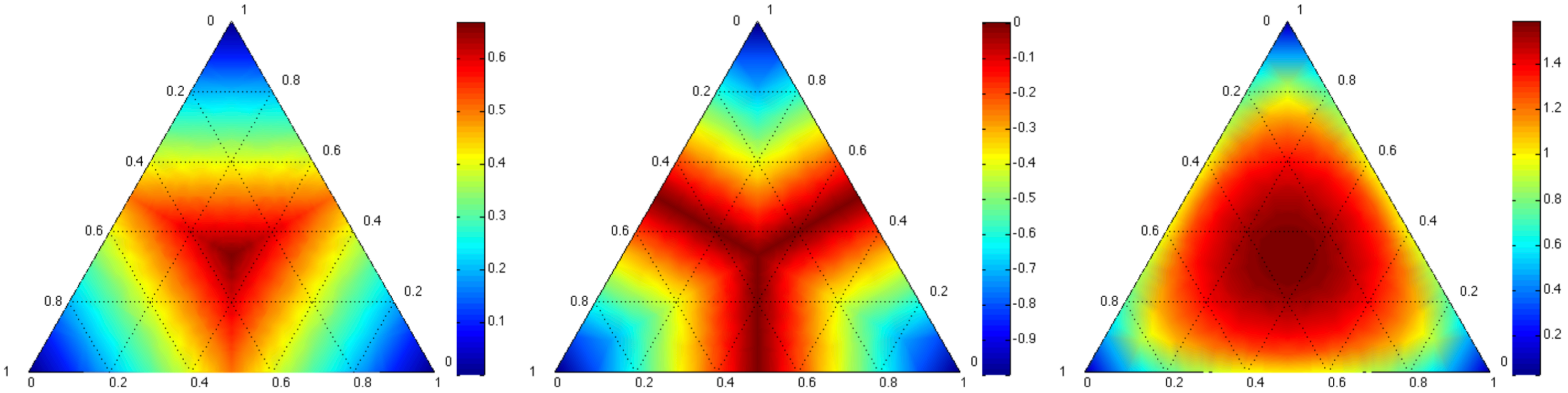


Figure 2: An illustrative example of pool-based active learning. (a) A toy data set of 400 instances, evenly sampled from two class Gaussians. The instances are represented as points in a 2D feature space. (b) A logistic regression model trained with 30 labeled instances randomly drawn from the problem domain. The line represents the decision boundary of the classifier (70% accuracy). (c) A logistic regression model trained with 30 actively queried instances using uncertainty sampling (90%).

Settles, (2010)

Uncertainty Metrics Visualized



(a) least confident (b) margin (c) entropy

Figure 5: Heatmaps illustrating the query behavior of common uncertainty measures in a three-label classification problem. Simplex corners indicate where one label has very high probability, with the opposite edge showing the probability range for the *other* two classes when that label has very low probability. Simplex centers represent a uniform posterior distribution. The most informative query region for each strategy is shown in dark red, radiating from the centers.

Settles, (2010)

Uncertainty Sampling: Failure Modes

Uncertainty Sampling: Failure Modes

- Brainstorm: **what could go wrong** when sampling by uncertainty?

Uncertainty Sampling: Failure Modes

- Brainstorm: **what could go wrong** when sampling by uncertainty?
- Failure mode 1: **Outlier Attraction**
  - Outliers and **noisy examples** tend to have **high uncertainty**
  - May sample **genuinely bad examples** (e.g. ambiguous, distorted)
  - Ex: paying radiologist to annotate a blurry or botched X-ray

Uncertainty Sampling: Failure Modes

- Brainstorm: **what could go wrong** when sampling by uncertainty?
- Failure mode 1: **Outlier Attraction**
  - Outliers and **noisy examples** tend to have **high uncertainty**
  - May sample **genuinely bad examples** (e.g. ambiguous, distorted)
  - Ex: paying radiologist to annotate a blurry or botched X-ray
- Failure mode 2: **Sampling Bias**
  - Uncertainty tends to sample near the **current decision boundary**
  - ...but that boundary might be in the **wrong place** (especially early on)
  - Sampling here can **reinforce a bad boundary**, and **discourage exploration**

Uncertainty Sampling: Failure Modes

Uncertainty Sampling: Failure Modes

- Failure mode 3: **Redundancy**
  - Many examples might be **uncertain in the same way**
  - Could lead to sampling of **nearly identical points** with **no new information**

Uncertainty Sampling: Failure Modes

- Failure mode 3: **Redundancy**
  - Many examples might be **uncertain in the same way**
  - Could lead to sampling of **nearly identical points** with **no new information**
- Highlights a common tradeoff in ML: **Exploitation vs. Exploration**
  - Uncertainty yields points that **seem informative/beneficial** (exploitation)
  - But this doesn't guarantee a **representative sample** (exploration)
  - Methods usually try to **balance the tradeoff**

Strategy Landscape

Query-by-Committee

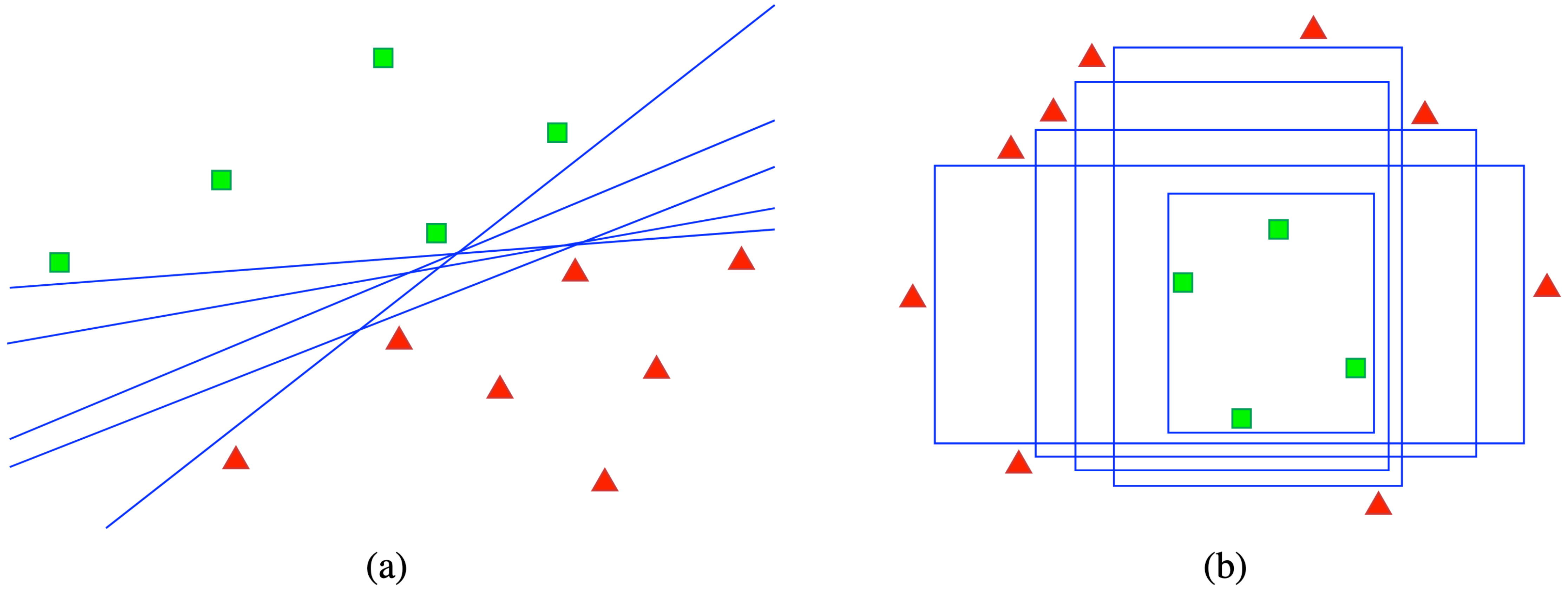


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

Settles, (2010)

Query-by-Committee

- Train an model **ensemble** ("committee"), find examples where models **disagree** the most
  - Disagreement ≈ uncertainty

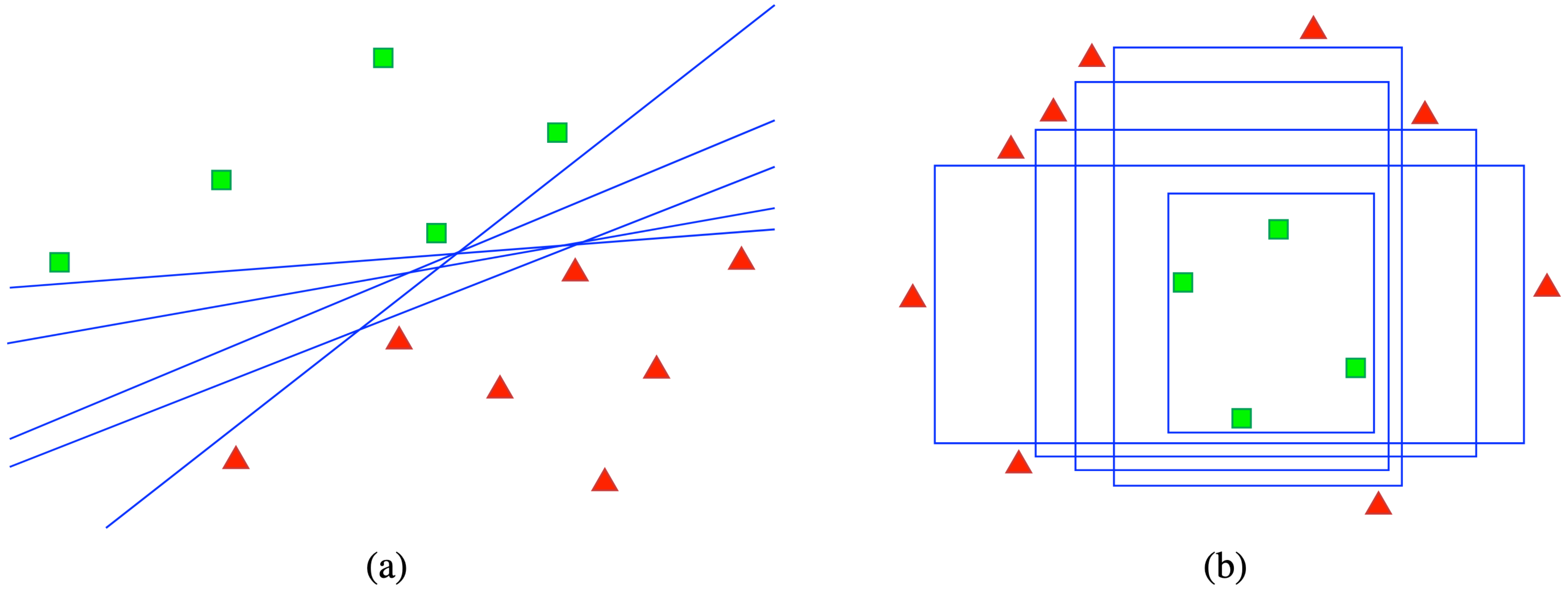


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

Settles, (2010)

Query-by-Committee

- Train an model **ensemble** ("committee"), find examples where models **disagree** the most
  - Disagreement ≈ uncertainty
- Can be measured with **vote entropy** or **KL divergence**
  - KL: how much do the models' **prediction distributions diverge?**
  - Nuance: KL is **not symmetric**. Divergence measured from consensus

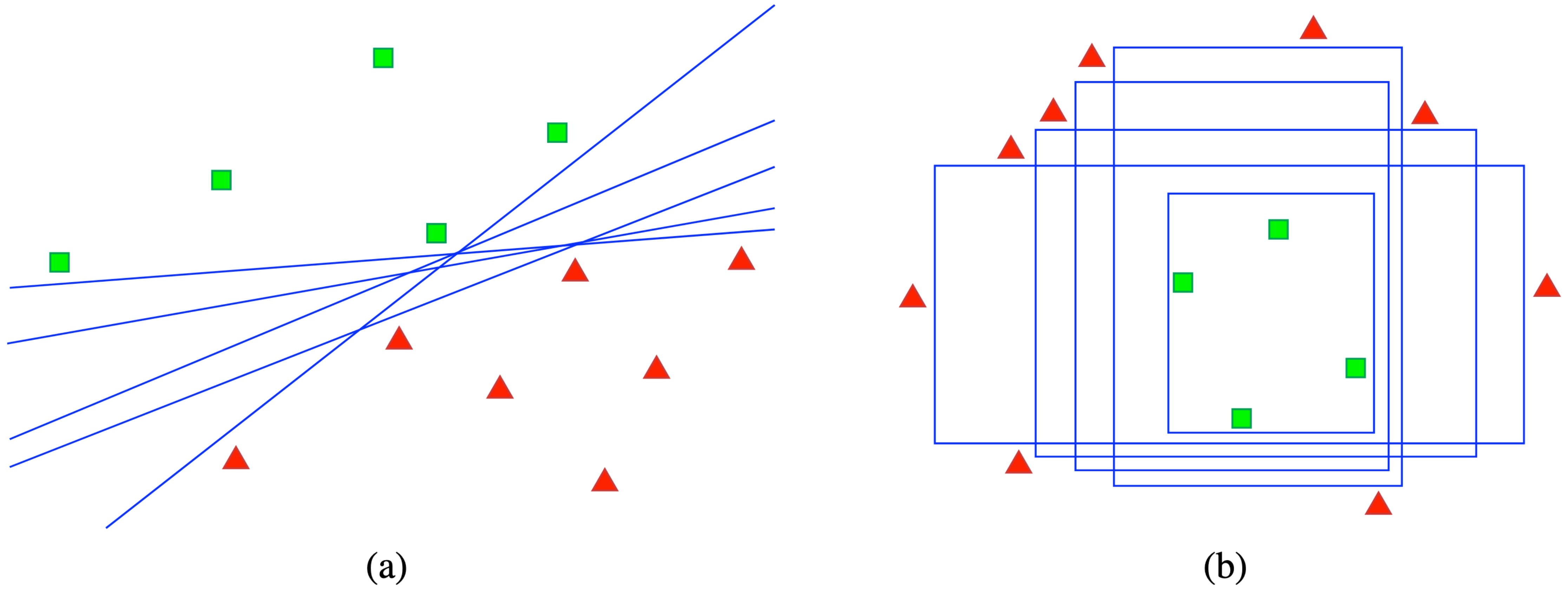


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

Settles, (2010)

Query-by-Committee

- Train an model **ensemble** ("committee"), find examples where models **disagree** the most
  - Disagreement ≈ uncertainty
- Can be measured with **vote entropy** or **KL divergence**
  - KL: how much do the models' **prediction distributions diverge?**
  - Nuance: KL is **not symmetric**. Divergence measured from consensus
- Intuition: a single model might be **confidently wrong.** Multiple predictions is **more robust**

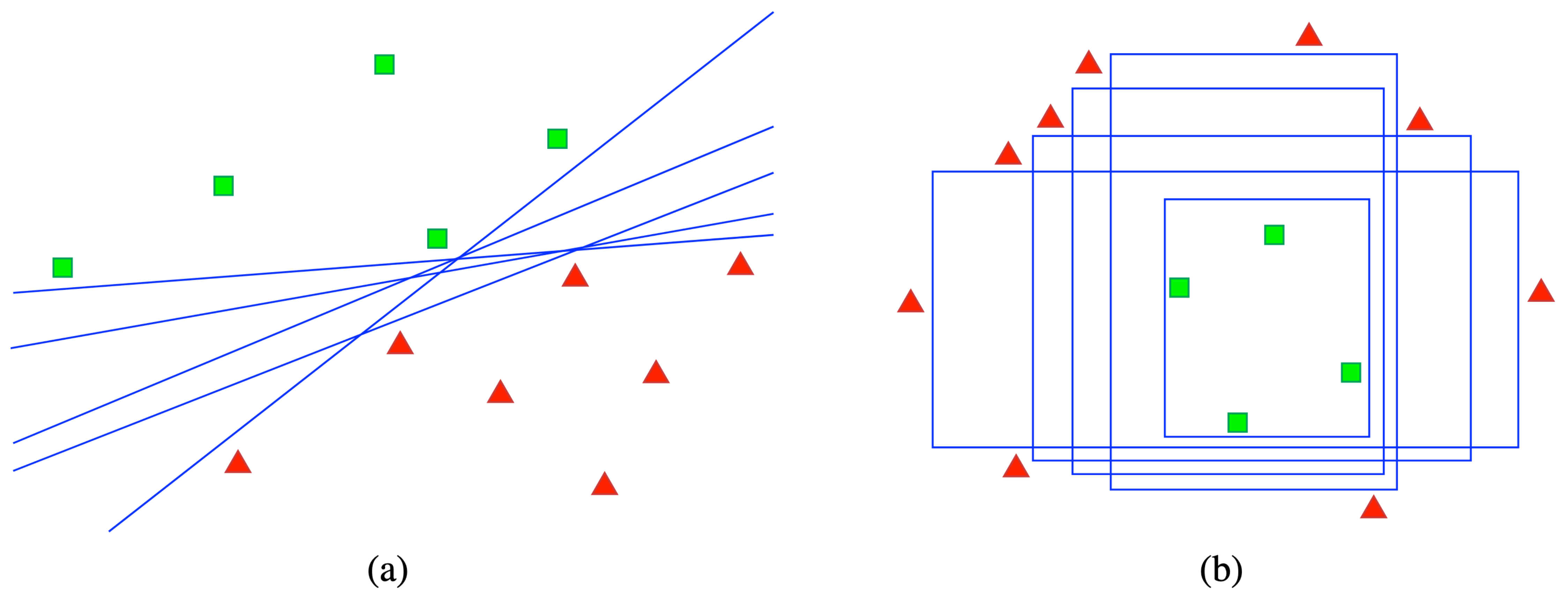


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

Settles, (2010)

Query-by-Committee

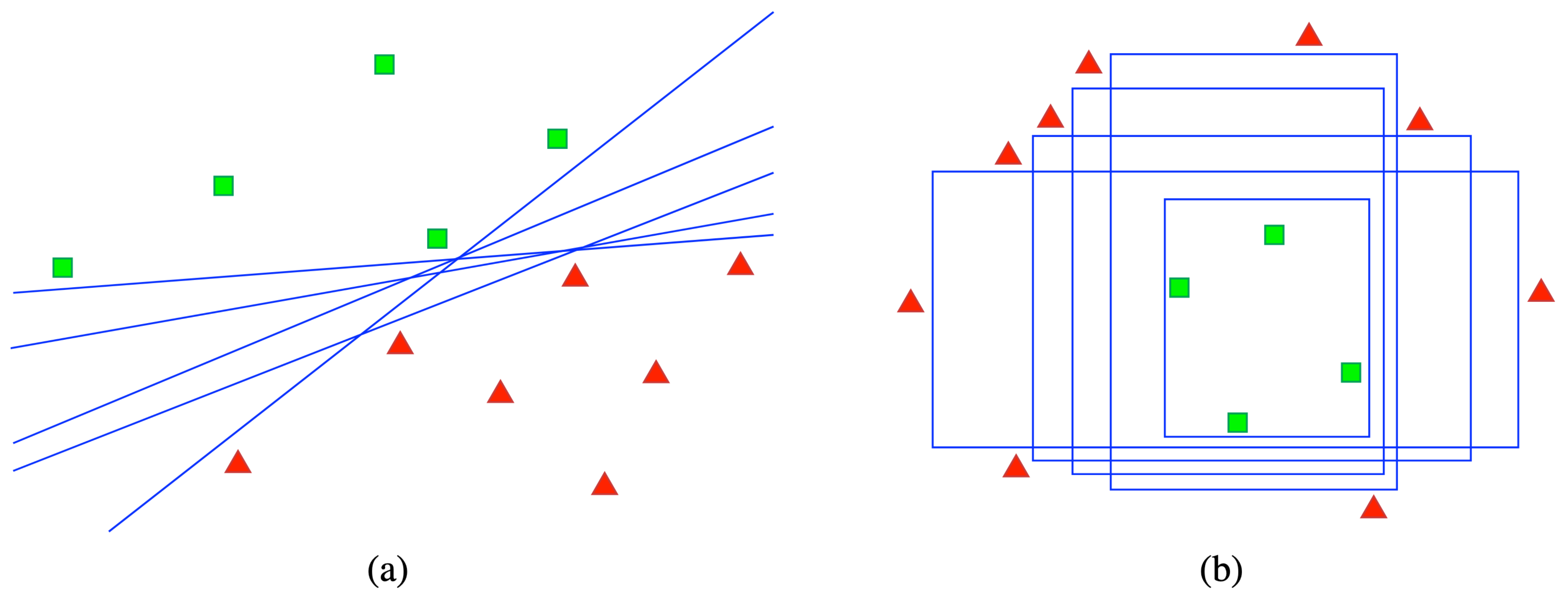


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

[Settles, (2010)]

Query-by-Committee

- Downside: training multiple models is **expensive** (especially NNs)

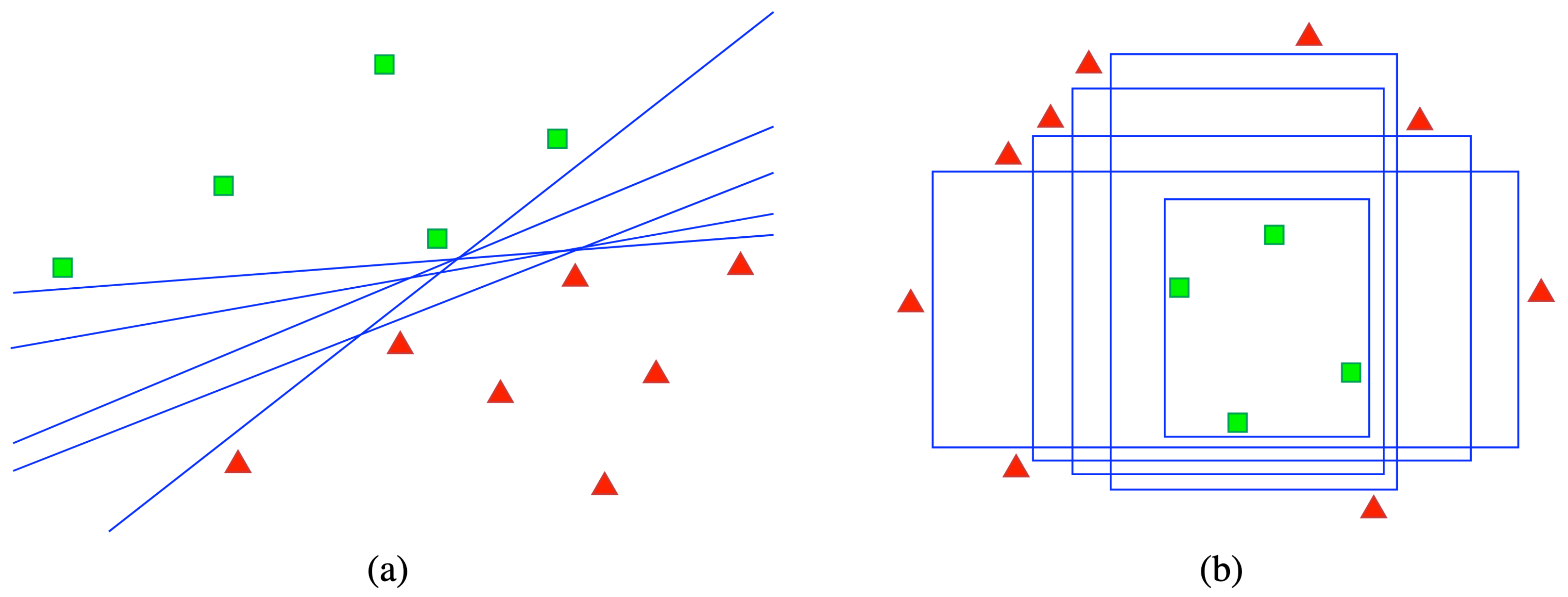


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

Settles, (2010)

Query-by-Committee

- Downside: training multiple models is **expensive** (especially NNs)
- Ways exist to **approximate** an ensemble with a single model

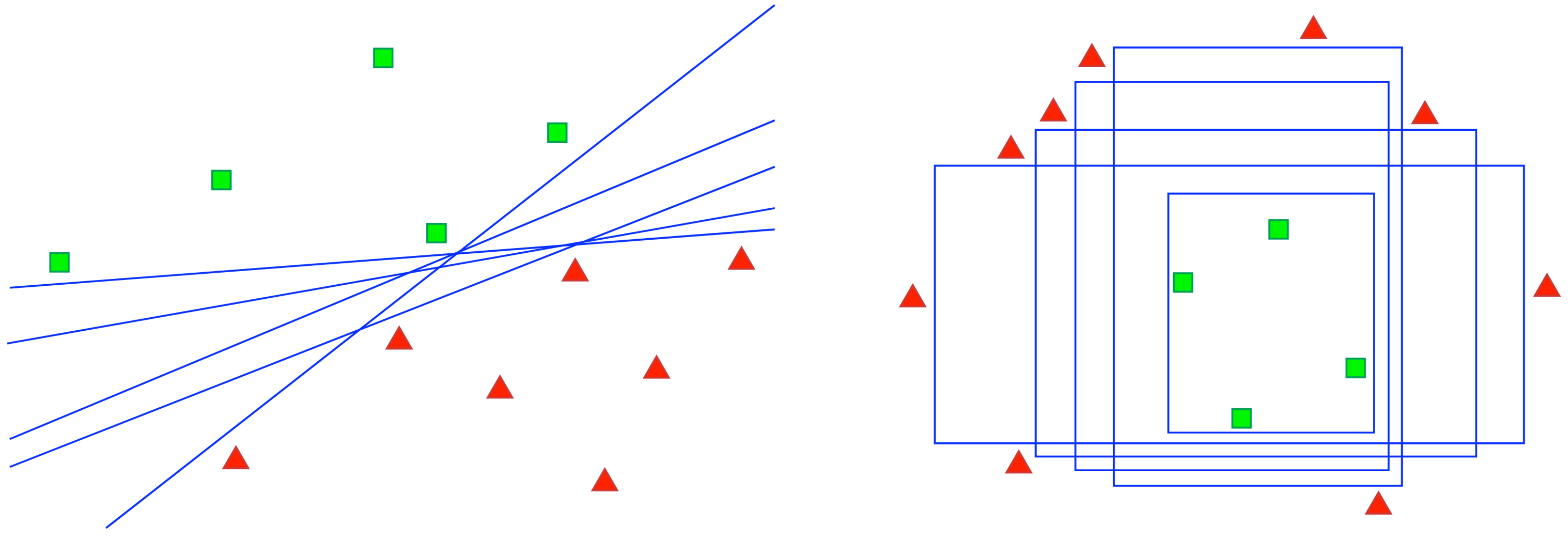


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

[Settles, (2010)]

Query-by-Committee

- Downside: training multiple models is **expensive** (especially NNs)
- Ways exist to **approximate** an ensemble with a single model
  - **MC Dropout:** multiple forward-passes with **different dropout**

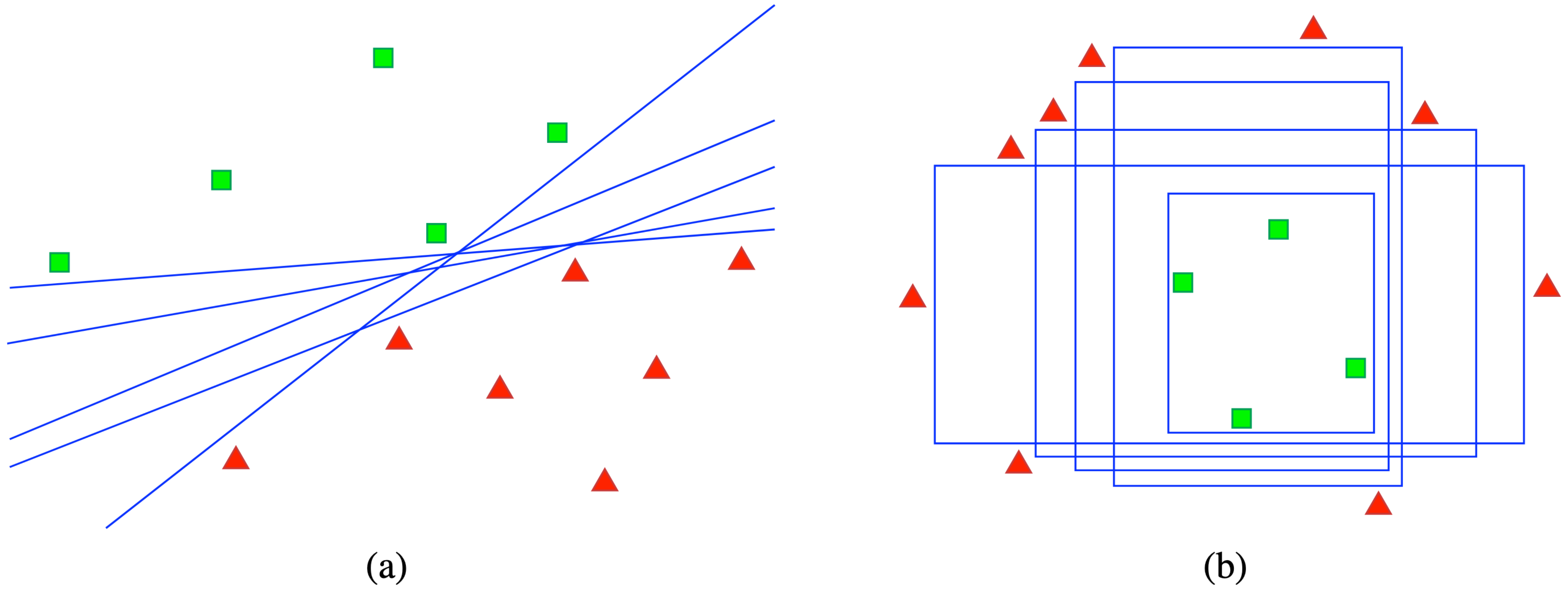


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

[Settles, (2010)]

Query-by-Committee

- Downside: training multiple models is **expensive** (especially NNs)
- Ways exist to **approximate** an ensemble with a single model
  - **MC Dropout:** multiple forward-passes with **different dropout**
  - **Snapshot Ensembles:** use same model at **different local minima**

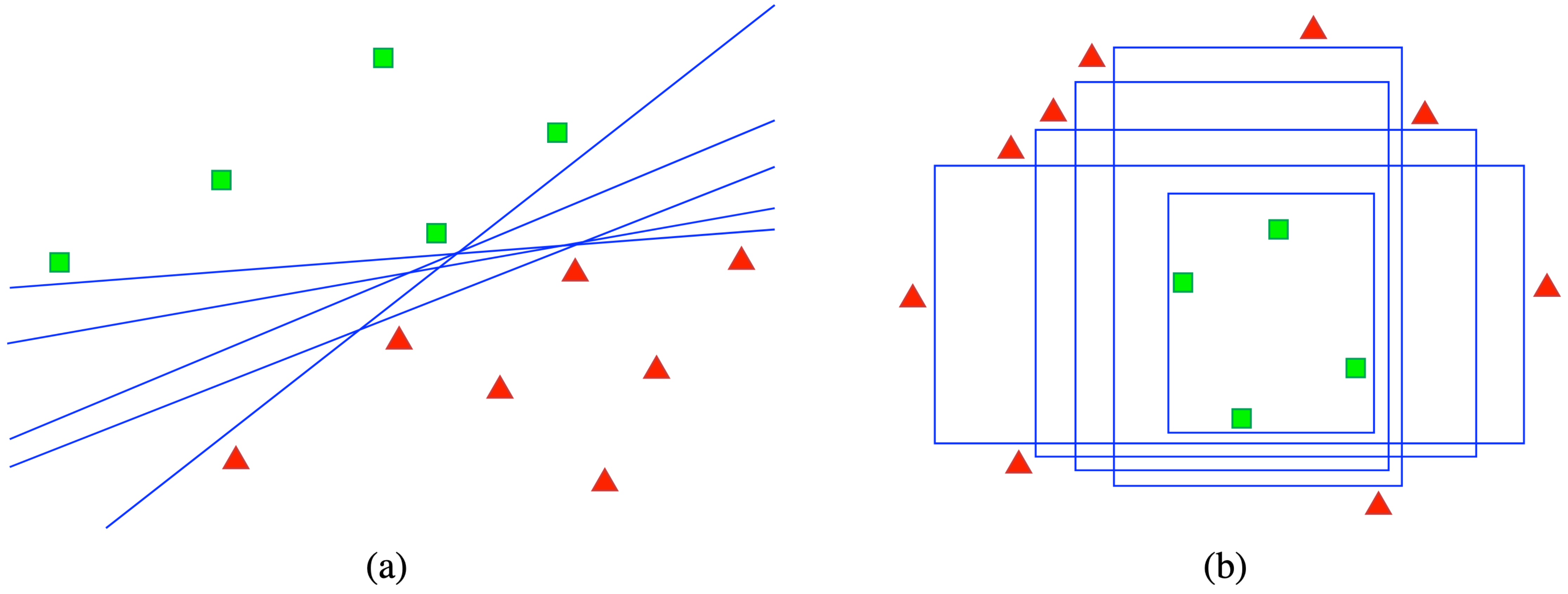
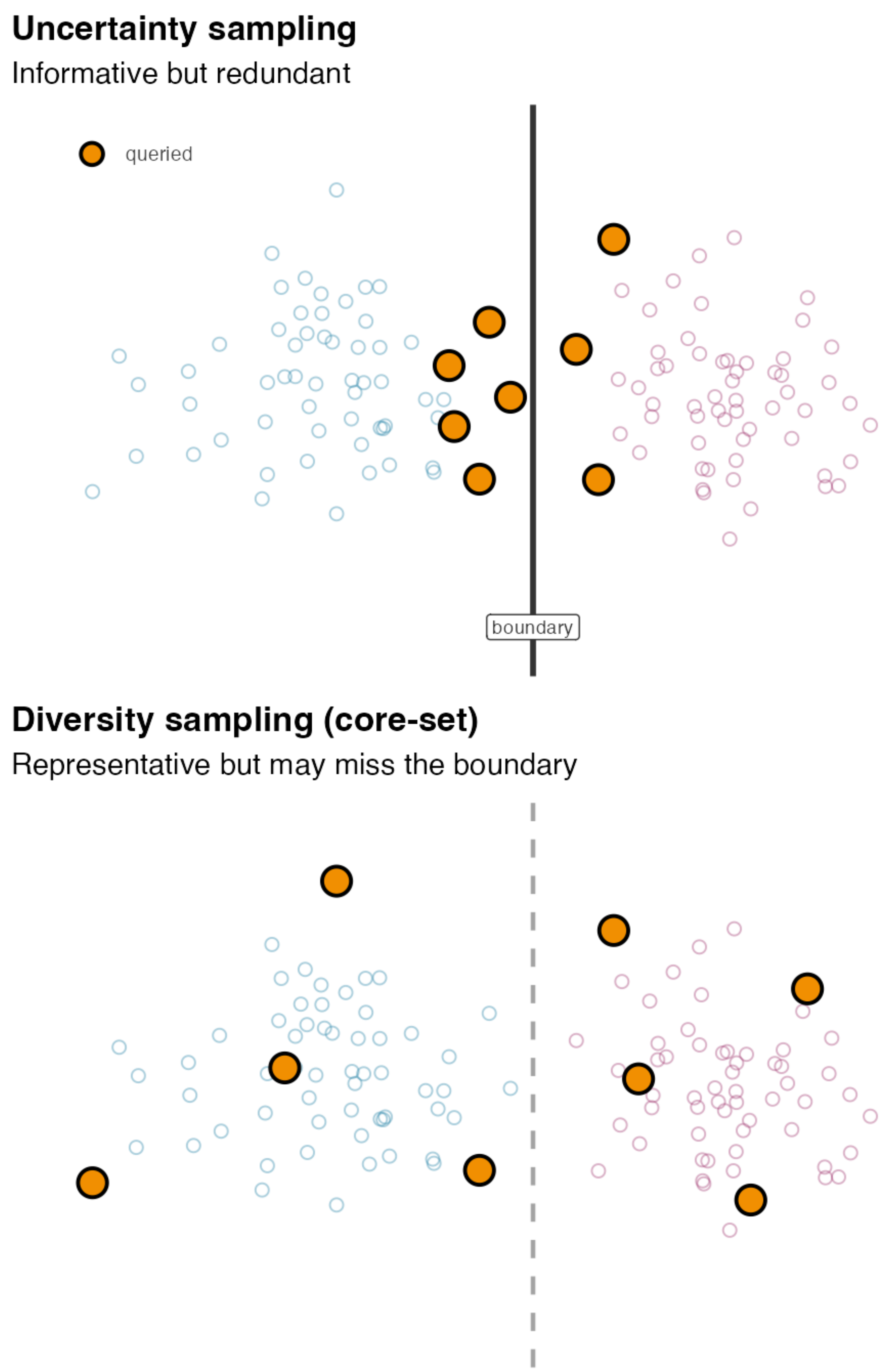


Figure 6: Version space examples for (a) linear and (b) axis-parallel box classifiers. All hypotheses are consistent with the labeled training data in $\mathcal{L}$ (as indicated by shaded polygons), but each represents a different model in the version space.

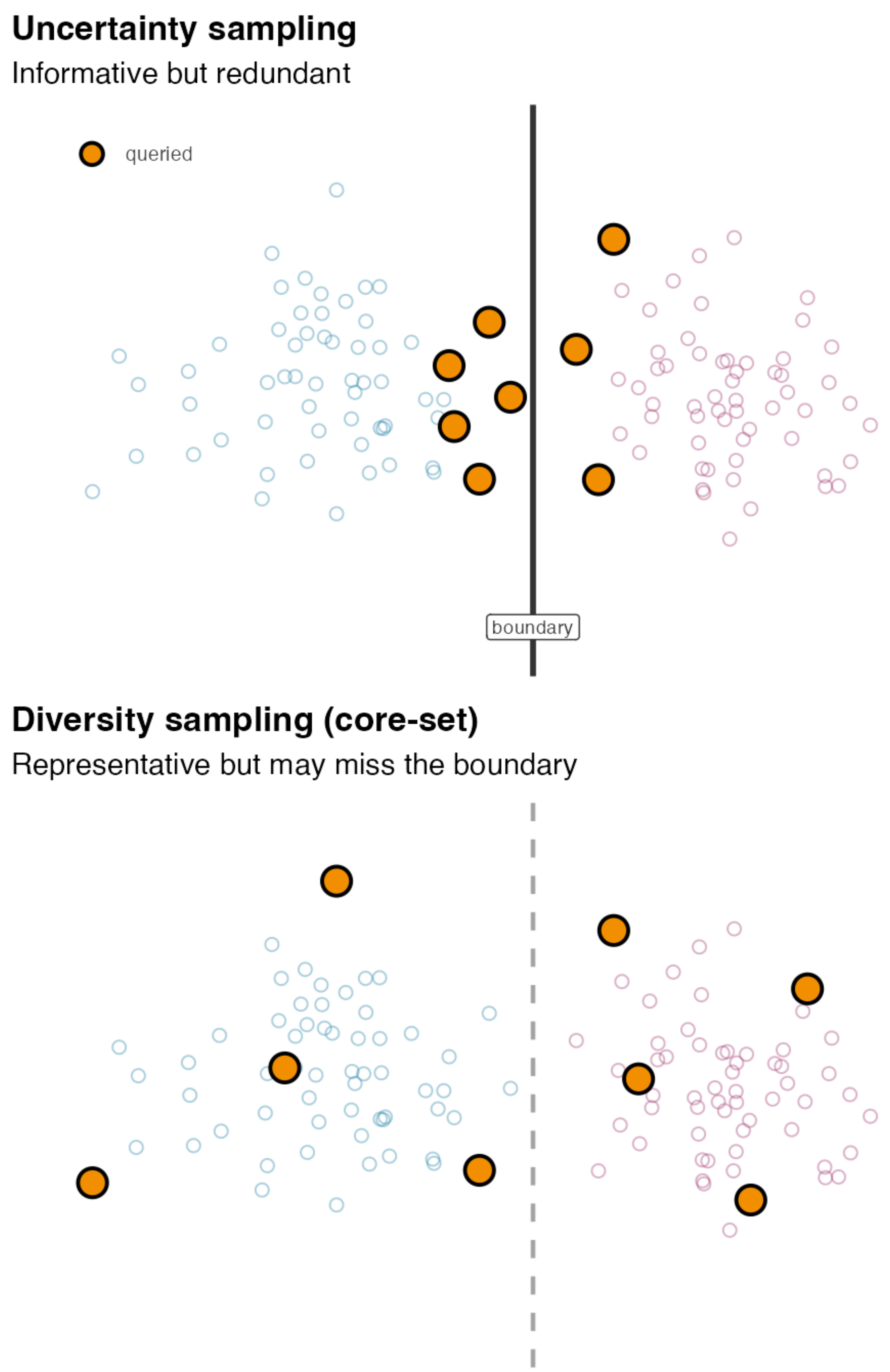
[Settles, (2010)]

Diversity/Core-Set



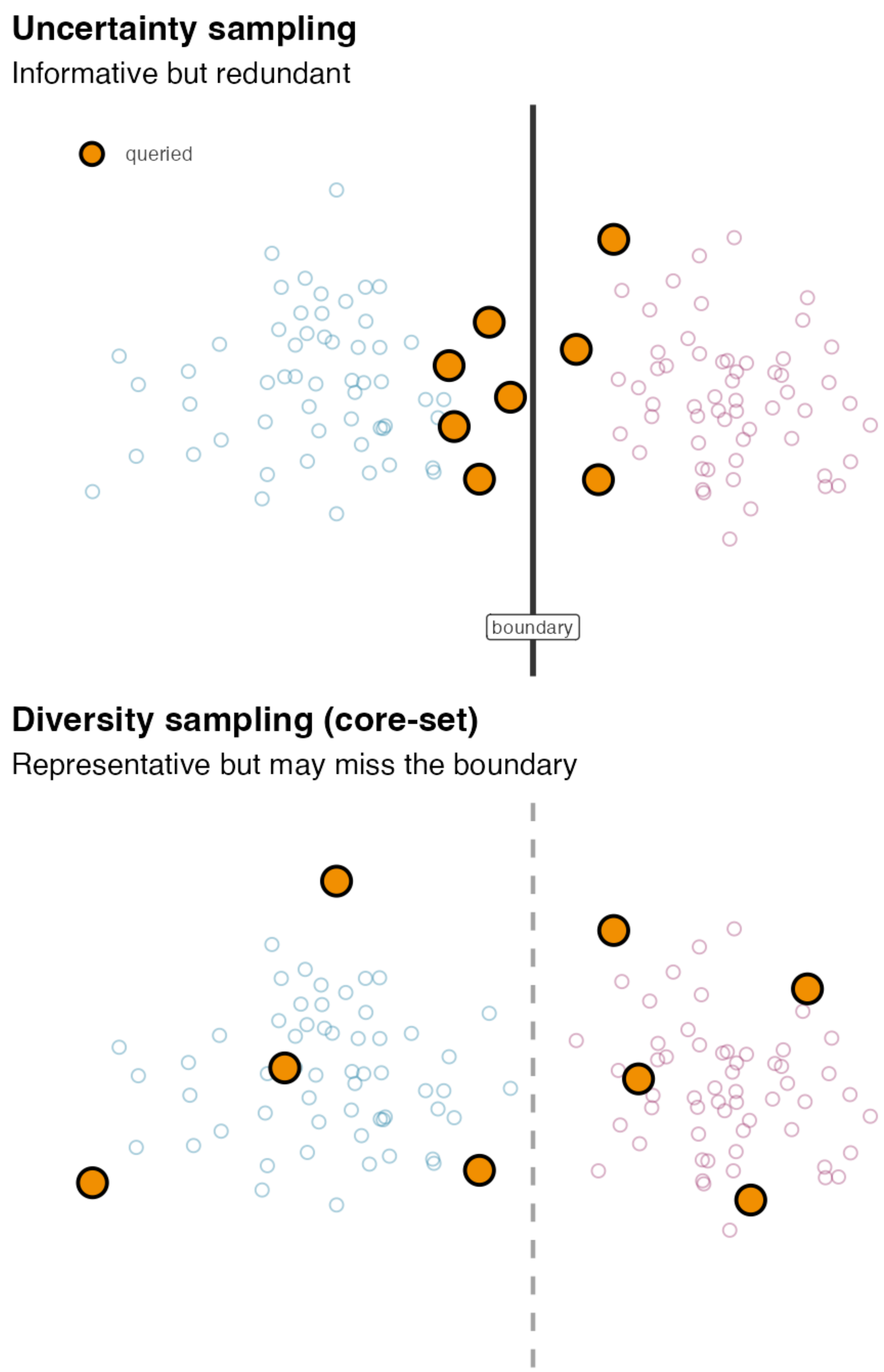
Diversity/Core-Set

- What if we focused on **exploration** of the unlabeled space rather than uncertainty?



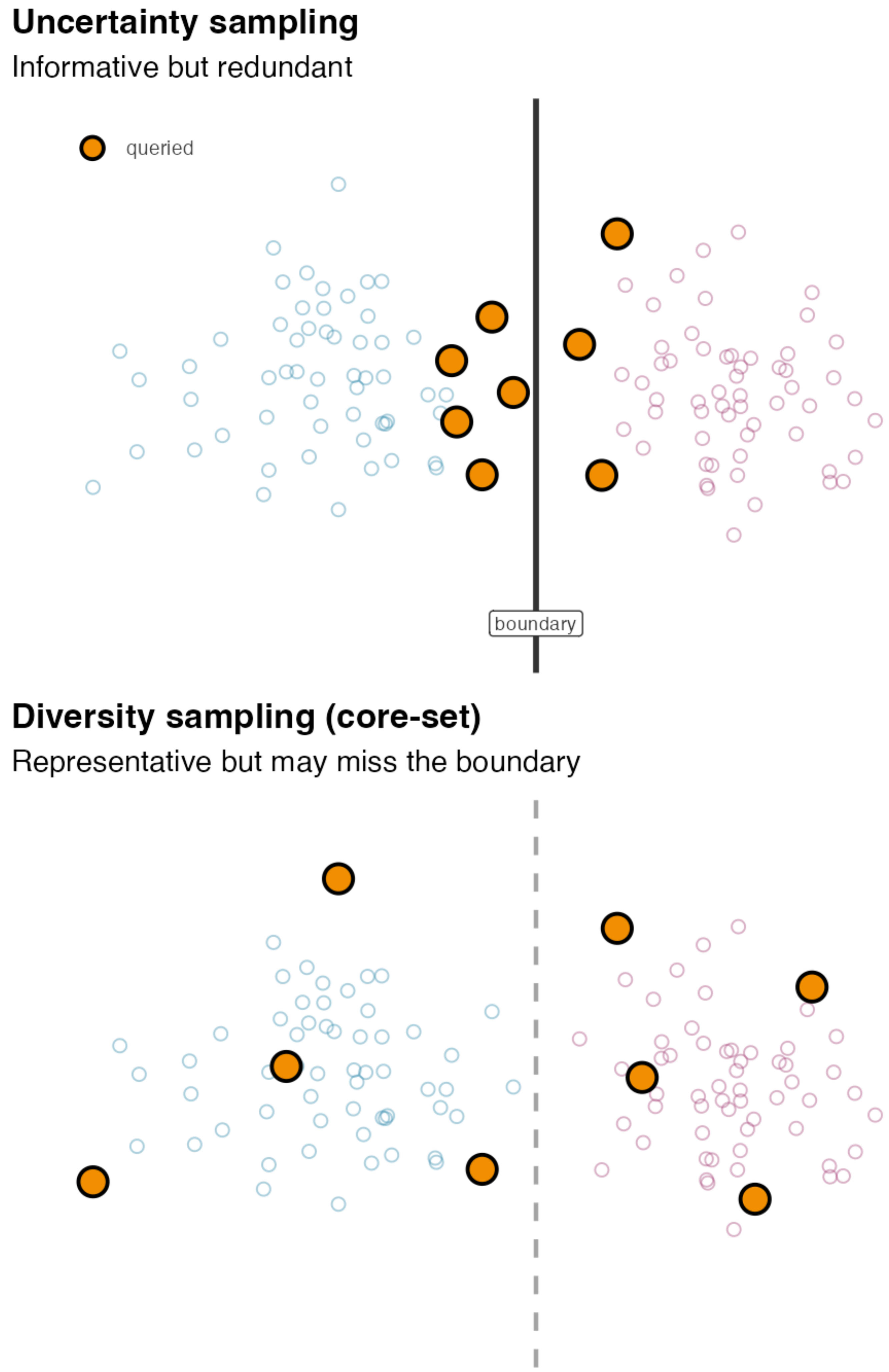
Diversity/Core-Set

- What if we focused on **exploration** of the unlabeled space rather than uncertainty?
- **Core-Set:** choose a **subset of** U that has **maximum "coverage"**



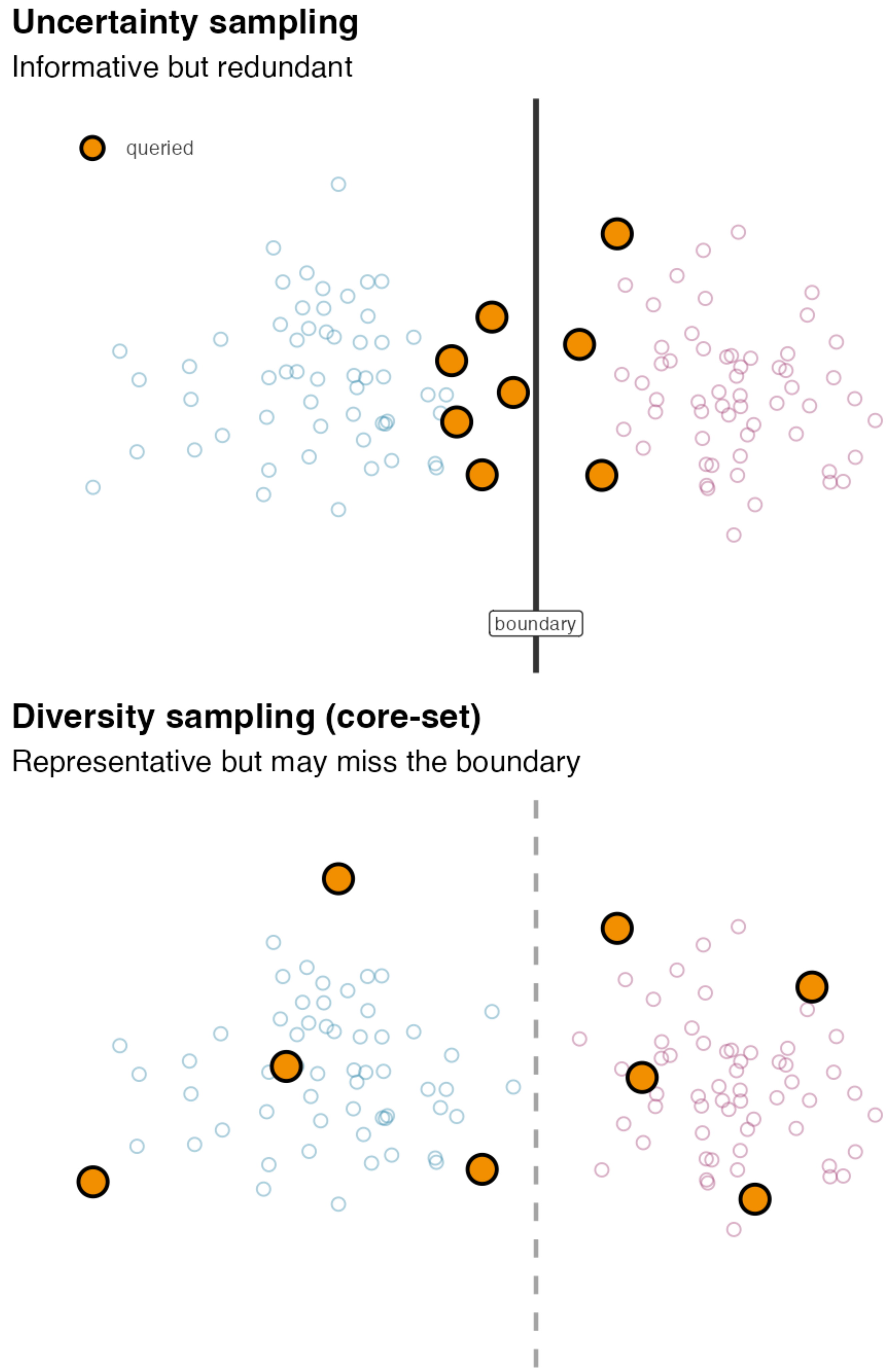
Diversity/Core-Set

- What if we focused on **exploration** of the unlabeled space rather than uncertainty?
- **Core-Set:** choose a **subset of** U that has **maximum "coverage"**
  - Formally: minimize the **maximum distance** from **any unlabeled point** to one in the **Core Set**



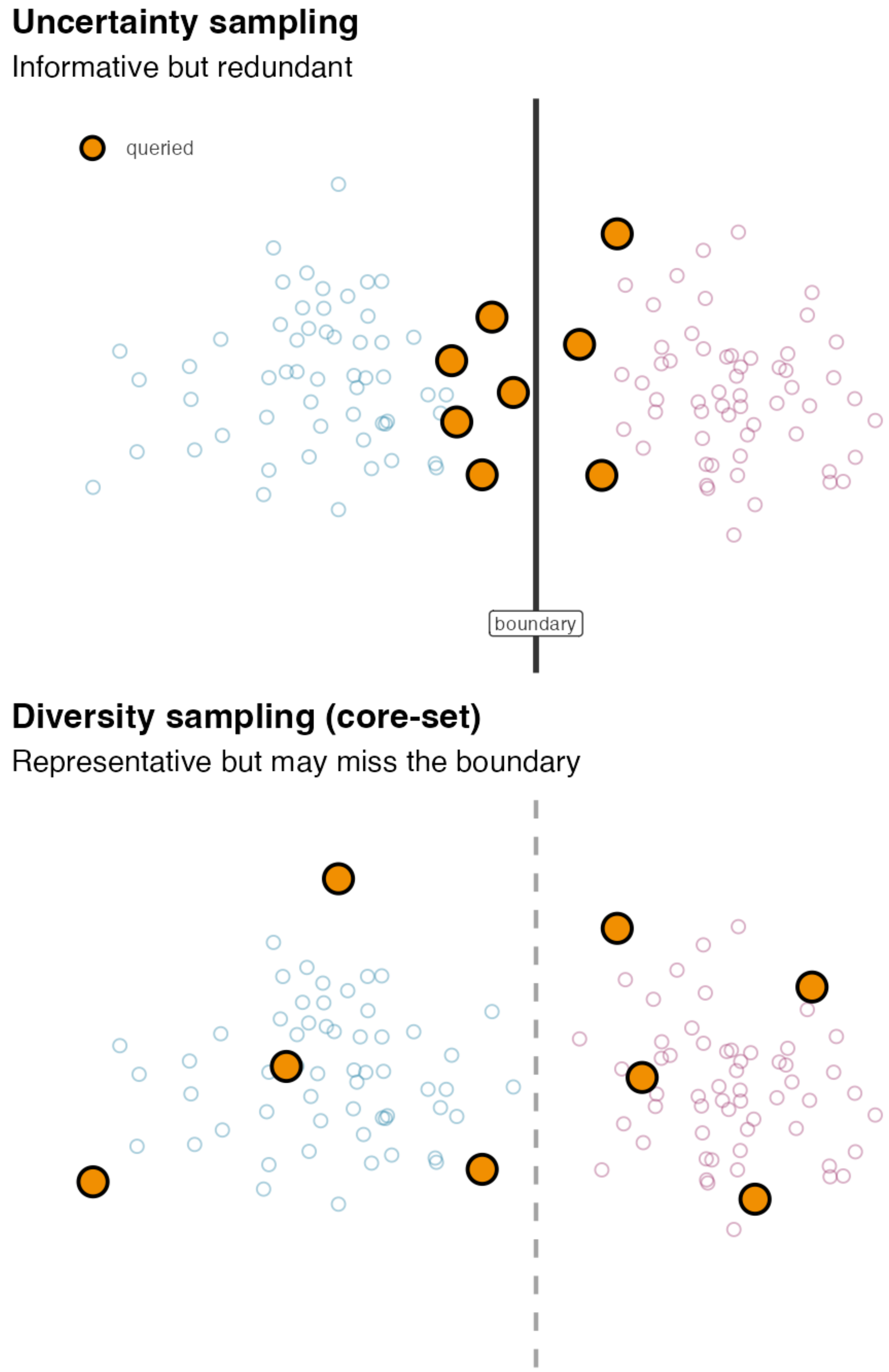
Diversity/Core-Set

- What if we focused on **exploration** of the unlabeled space rather than uncertainty?
- **Core-Set:** choose a **subset of** U that has **maximum "coverage"**
  - Formally: minimize the **maximum distance** from **any unlabeled point** to one in the **Core Set**
  - Similar to k-means, how do you determine the **right number of points?** (think for a second)

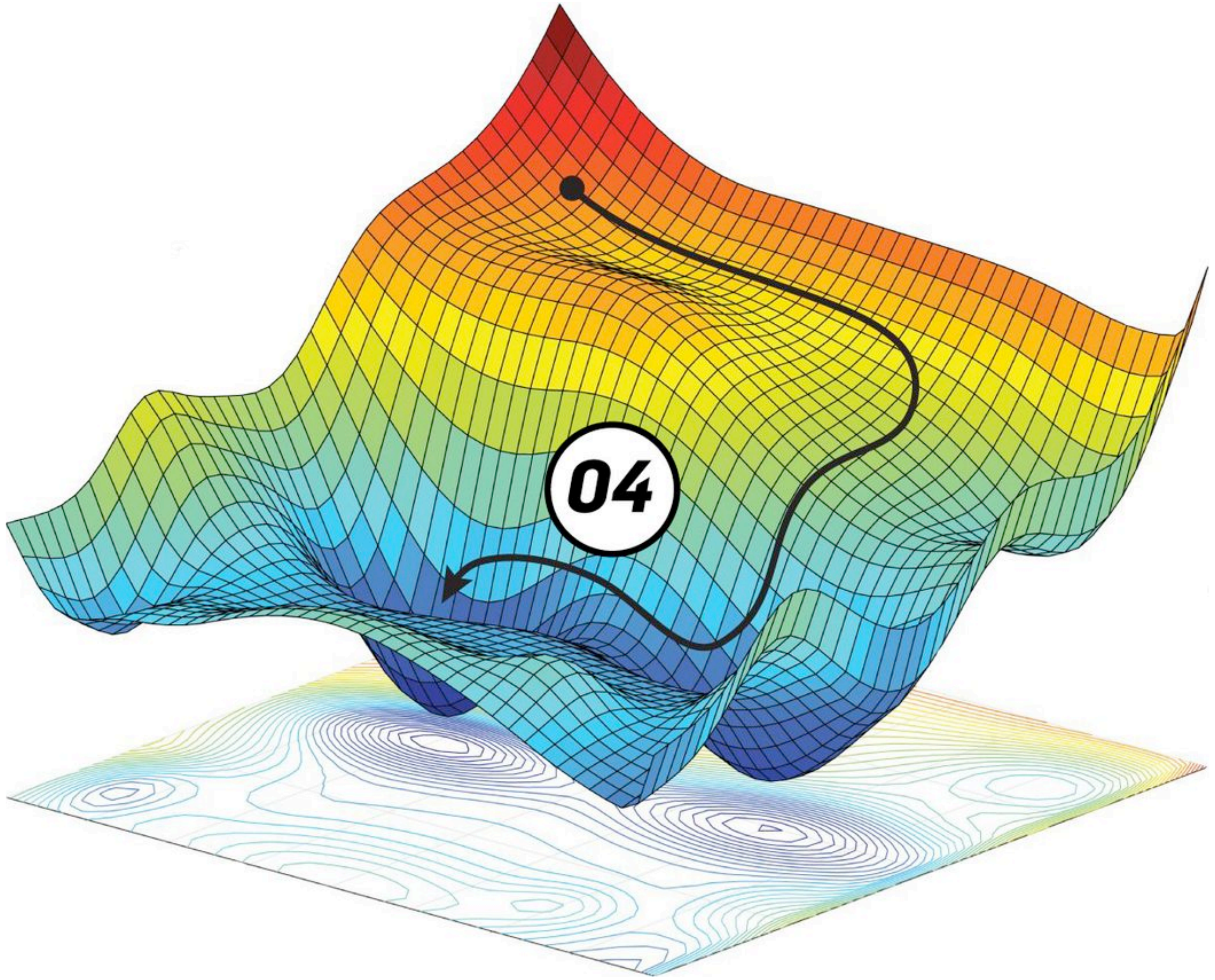


Diversity/Core-Set

- What if we focused on **exploration** of the unlabeled space rather than uncertainty?
- **Core-Set:** choose a **subset of** U that has **maximum "coverage"**
  - Formally: minimize the **maximum distance** from **any unlabeled point** to one in the **Core Set**
  - Similar to k-means, how do you determine the **right number of points?** (think for a second)
  - Answer: your **budget** or a fraction of it

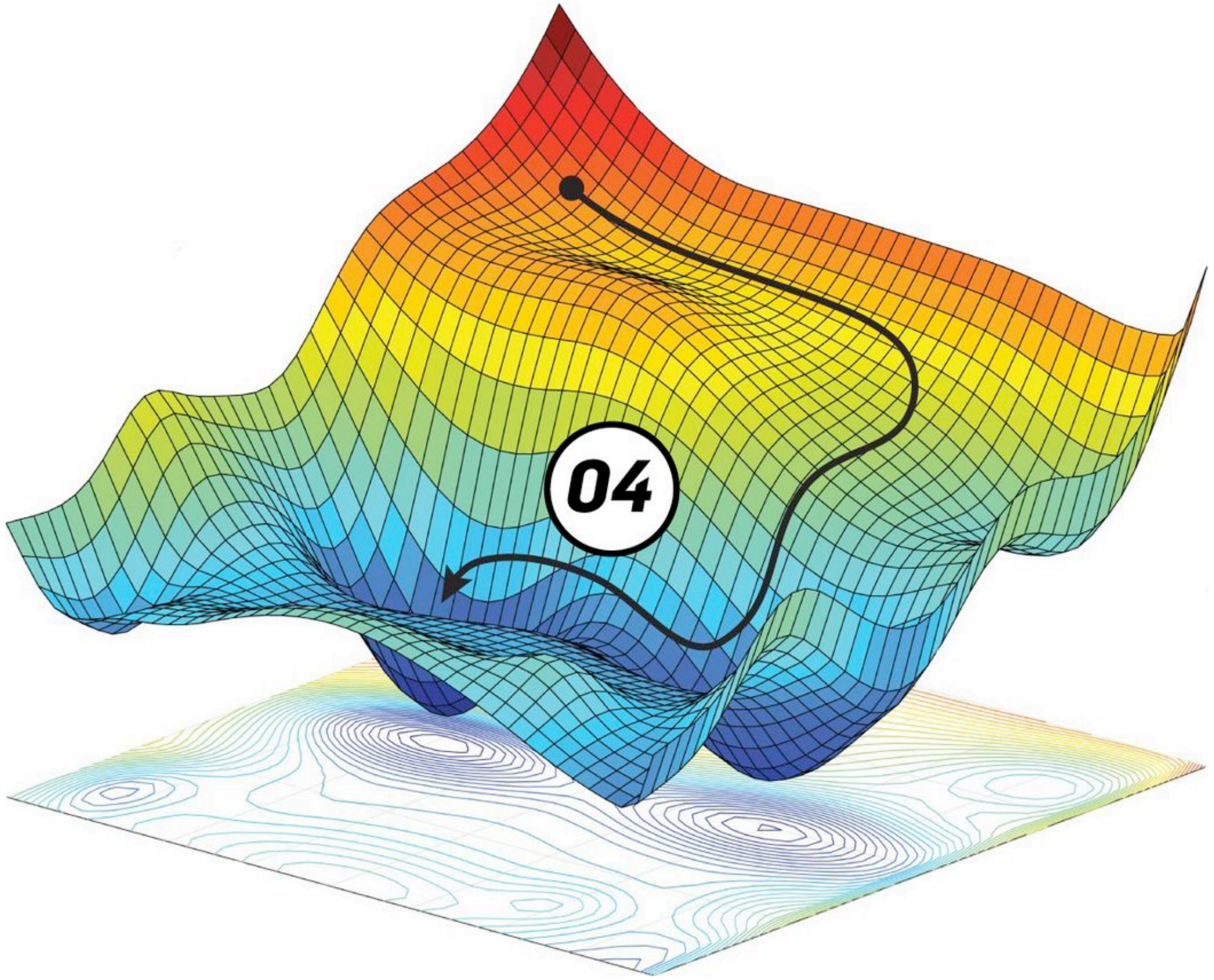


Gradient-Based Methods (BADGE)



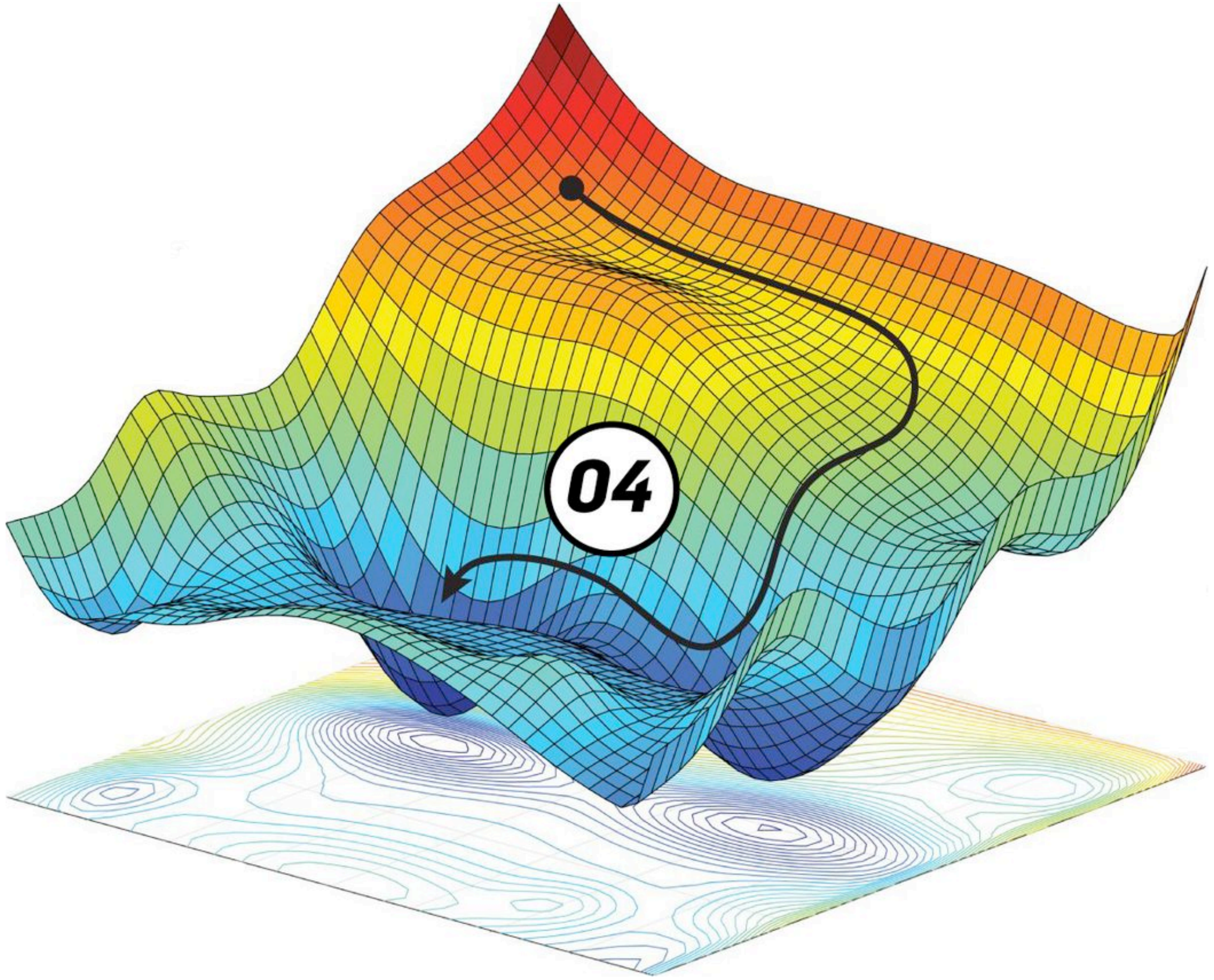
Gradient-Based Methods (BADGE)

- Use the **gradient** of the loss for each example as an **informativeness representation**



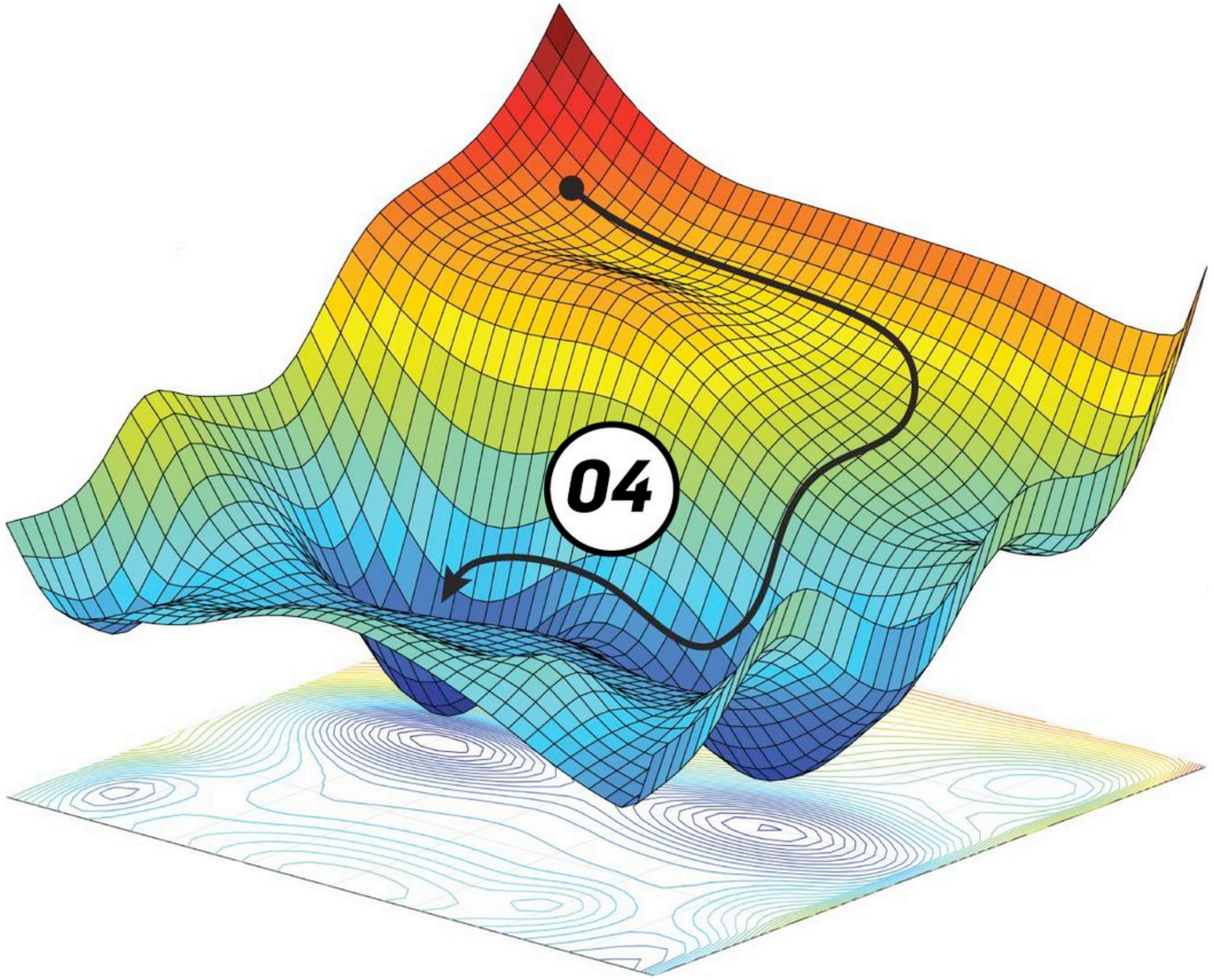
Gradient-Based Methods (BADGE)

- Use the **gradient** of the loss for each example as an **informativeness representation**
- Large gradient → large **model update** due to the example



Gradient-Based Methods (BADGE)

- Use the **gradient** of the loss for each example as an **informativeness representation**
- Large gradient → large **model update** due to the example
- Gradient **direction** also captures **"what the example teaches"** (similar direction ≈ similar information)
    - Direction can help detect **redundant examples**



BADGE Algorithm

Algorithm 1 BADGE: Batch Active learning by Diverse Gradient Embeddings

Require: Neural network $f(x;\theta)$, unlabeled pool of examples U, initial number of examples M, number of iterations T, number of examples in a batch B.

1: Labeled dataset $S \leftarrow M$ examples drawn uniformly at random from U together with queried labels.
2: Train an initial model θ_1 on S by minimizing $\mathbb{E}_S[\ell_{\mathrm{CE}}(f(x;\theta), y)]$.
3: **for** $t = 1, 2, \ldots, T$: **do**
4: For all examples x in $U \setminus S$:
  1. Compute its hypothetical label $\hat{y}(x) = h_{\theta_t}(x)$.
  2. Compute gradient embedding $g_x = \frac{\partial}{\partial \theta_{\mathrm{out}}} \ell_{\mathrm{CE}}(f(x;\theta), \hat{y}(x))|_{\theta=\theta_t}$, where θ_{out} refers to parameters of the final (output) layer.

5: Compute S_t, a random subset of $U \setminus S$, using the k-MEANS++ seeding algorithm on $\{g_x : x \in U \setminus S\}$ and query for their labels.
6: $S \leftarrow S \cup S_t$.
7: Train a model θ_{t+1} on S by minimizing $\mathbb{E}_S[\ell_{\mathrm{CE}}(f(x;\theta), y)]$.
8: **end for**
9: **return** Final model θ_{T+1}.

Ash et al., (2020)

Practical Considerations

Chicken-and-Egg Problem

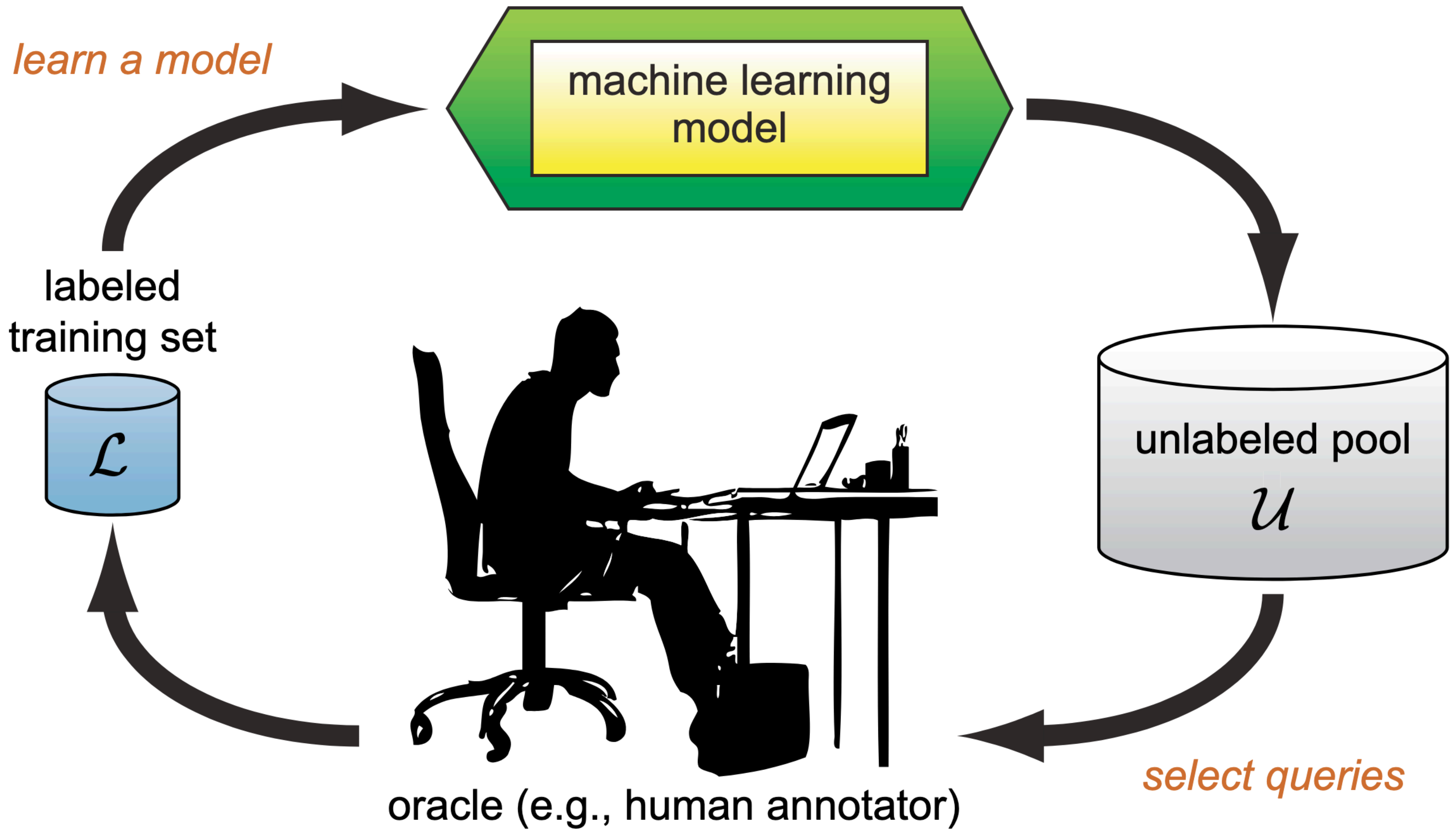


Figure 1: The pool-based active learning cycle.

Chicken-and-Egg Problem

- Sampling usually requires a **model to score examples**

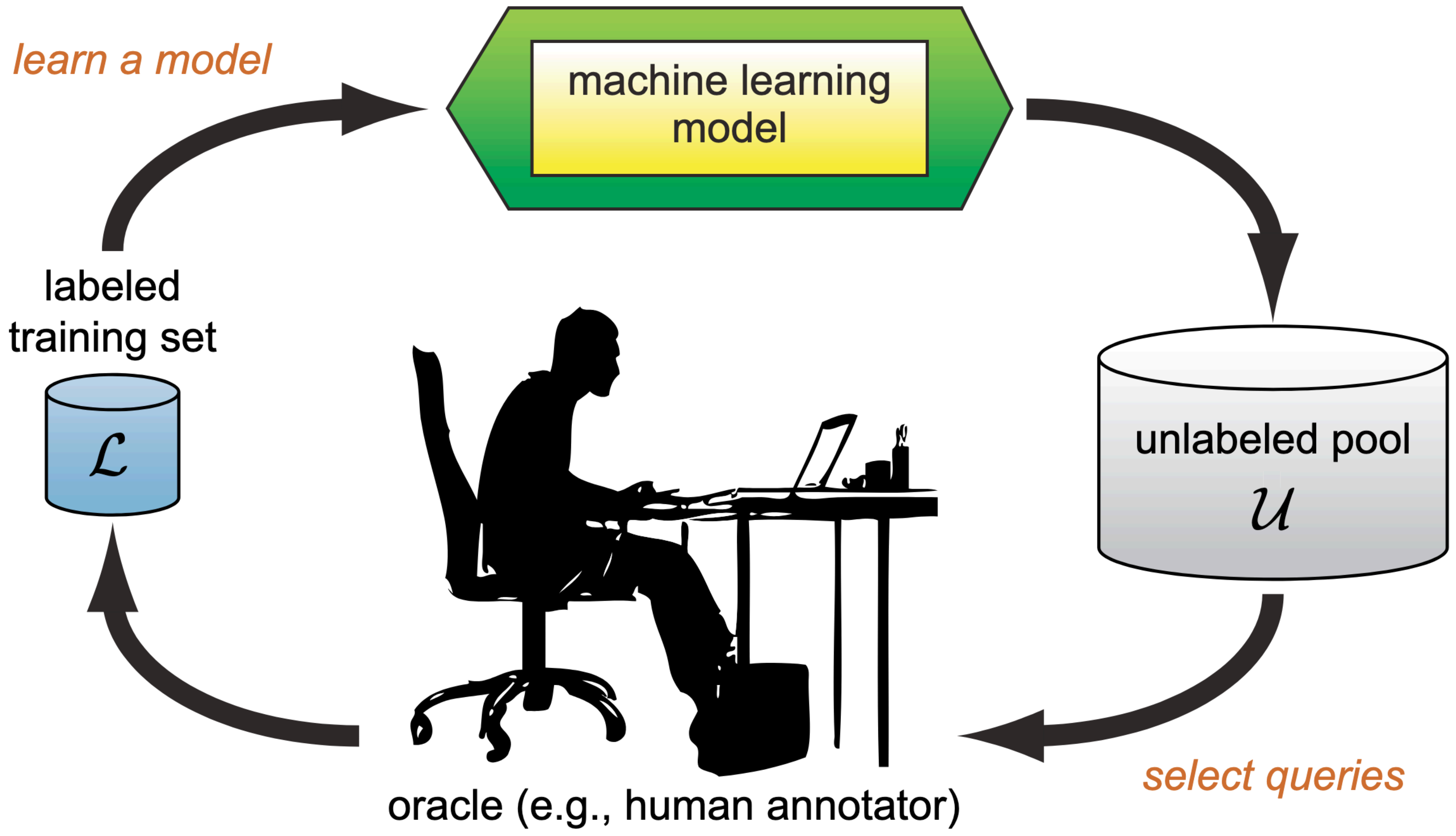


Figure 1: The pool-based active learning cycle.

Chicken-and-Egg Problem

- Sampling usually requires a **model to score examples**
- Model training needs **labeled data**

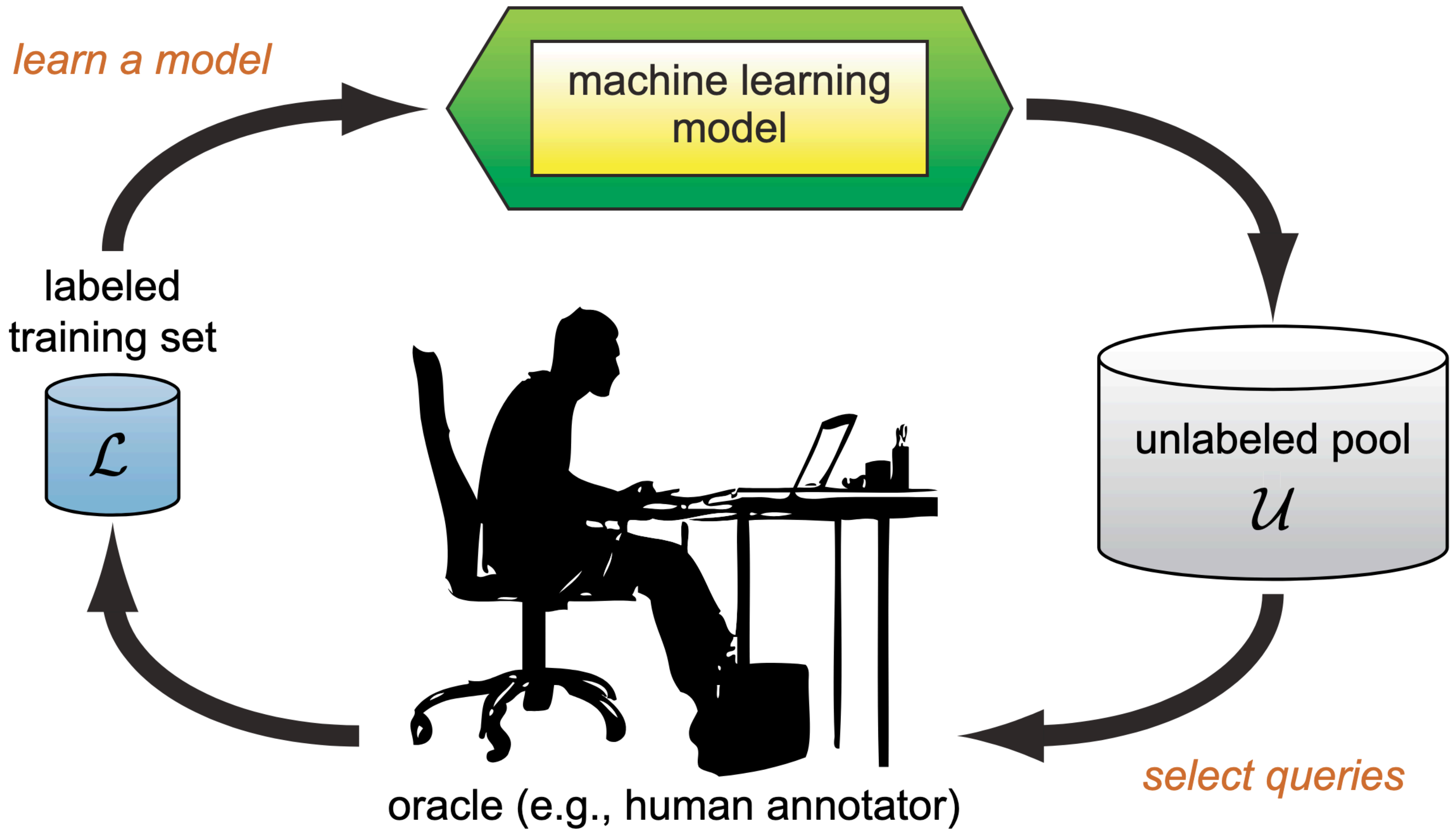


Figure 1: The pool-based active learning cycle.

Chicken-and-Egg Problem

- Sampling usually requires a **model to score examples**
- Model training needs **labeled data**
- How do you get one without the other? Solutions:
  - Start with a **random subset to label**
  - Start with a **diverse subset**
  - Use **pre-trained representations**

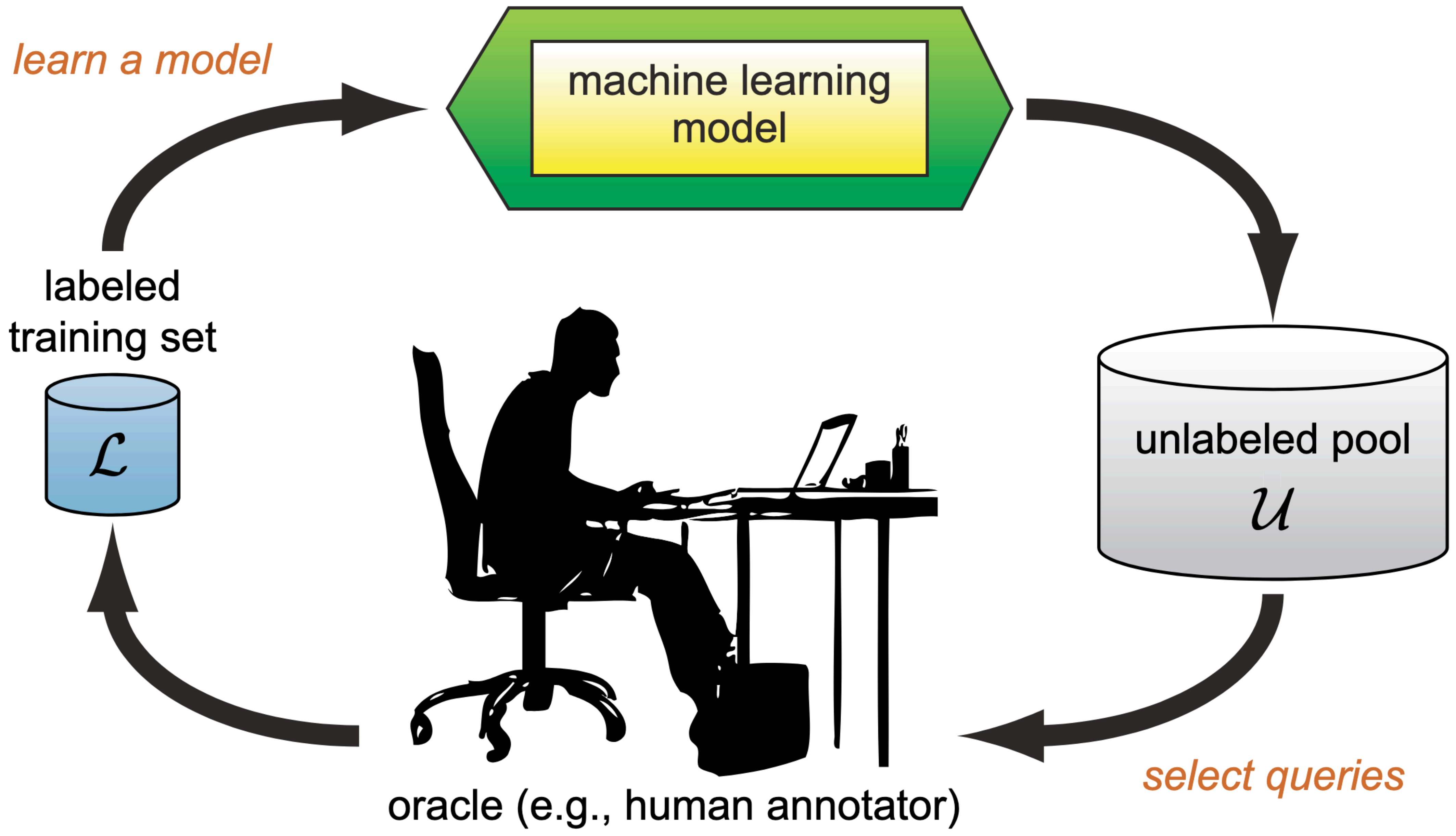
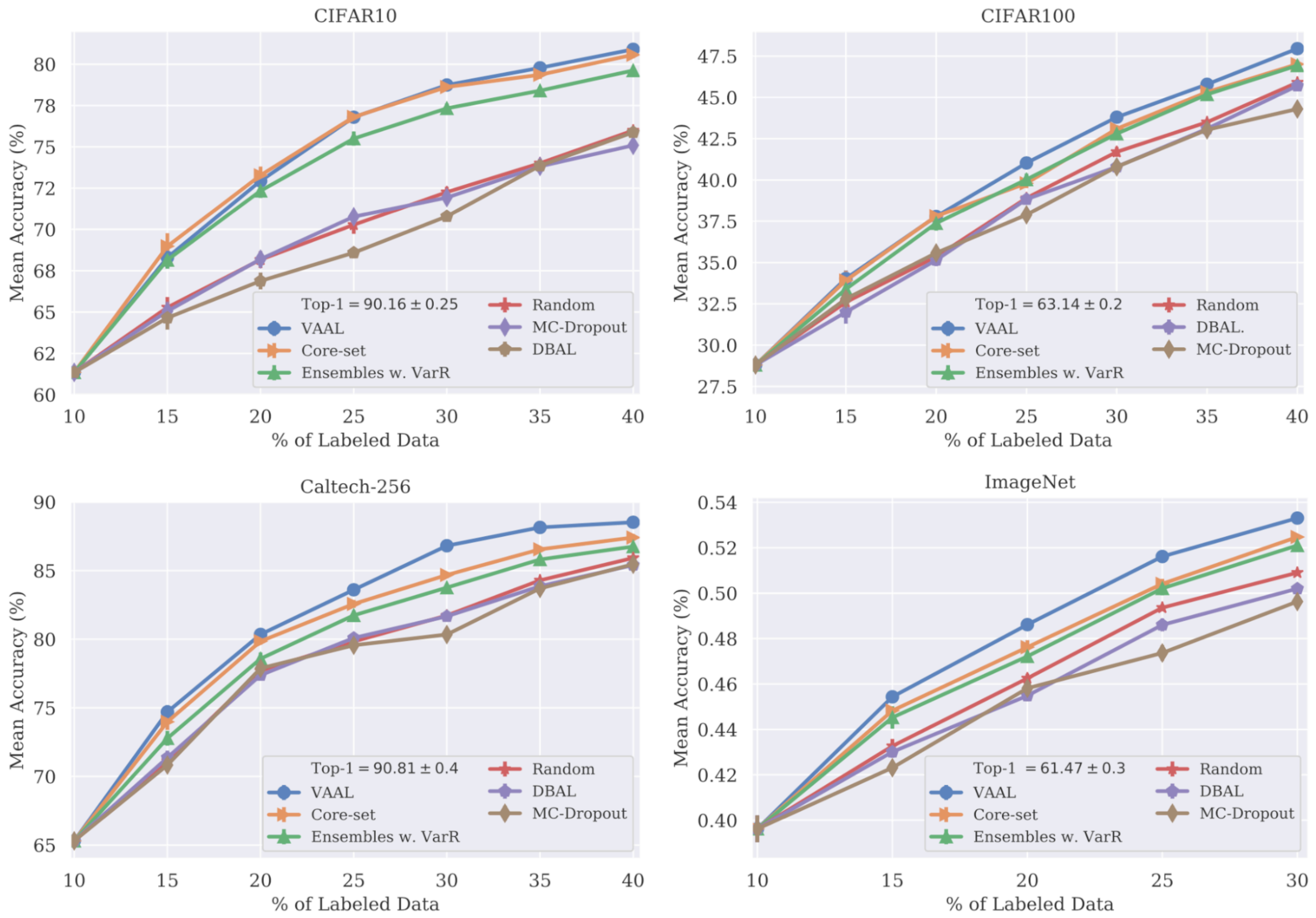


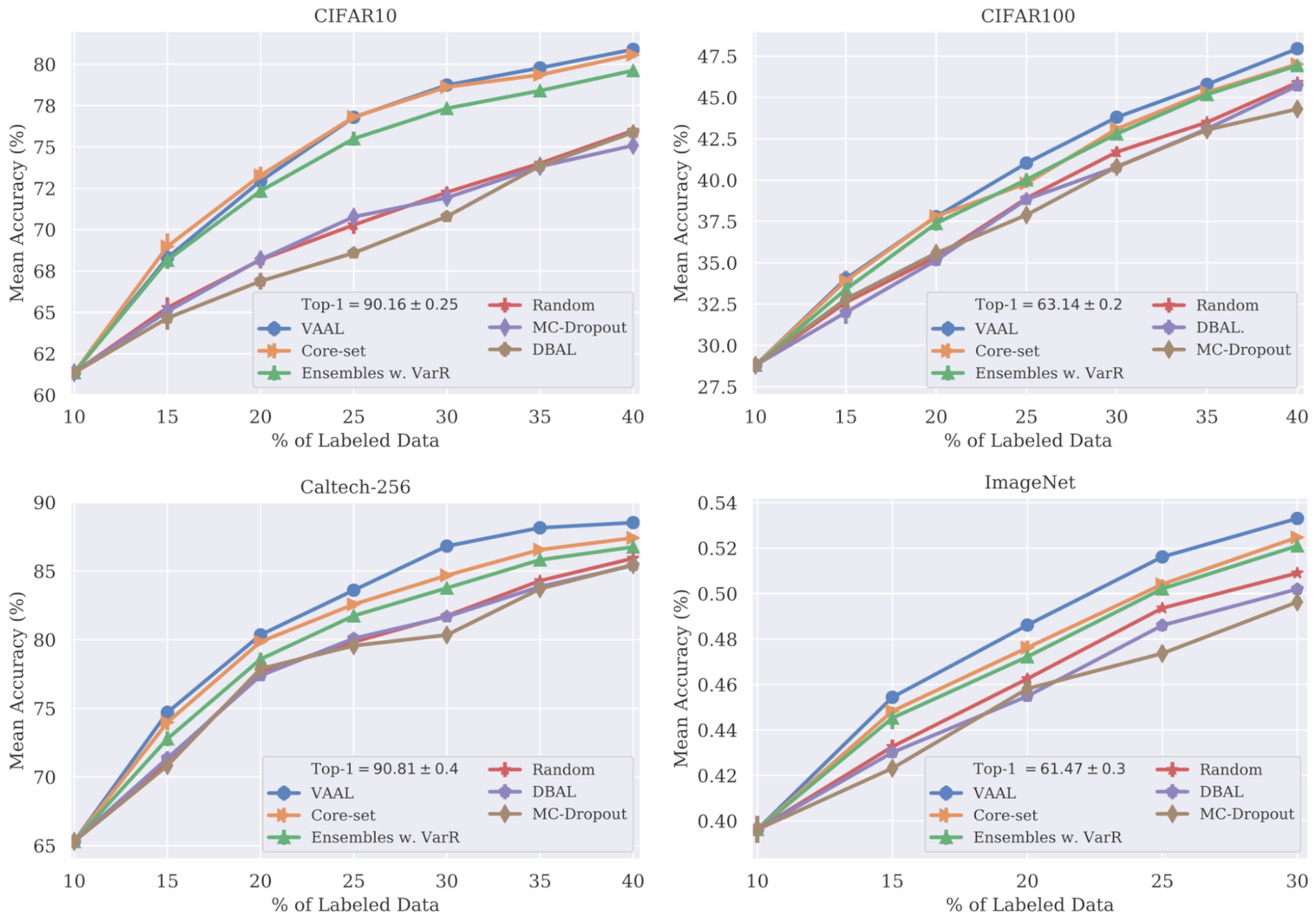
Figure 1: The pool-based active learning cycle.

When is Active Learning Worth It?



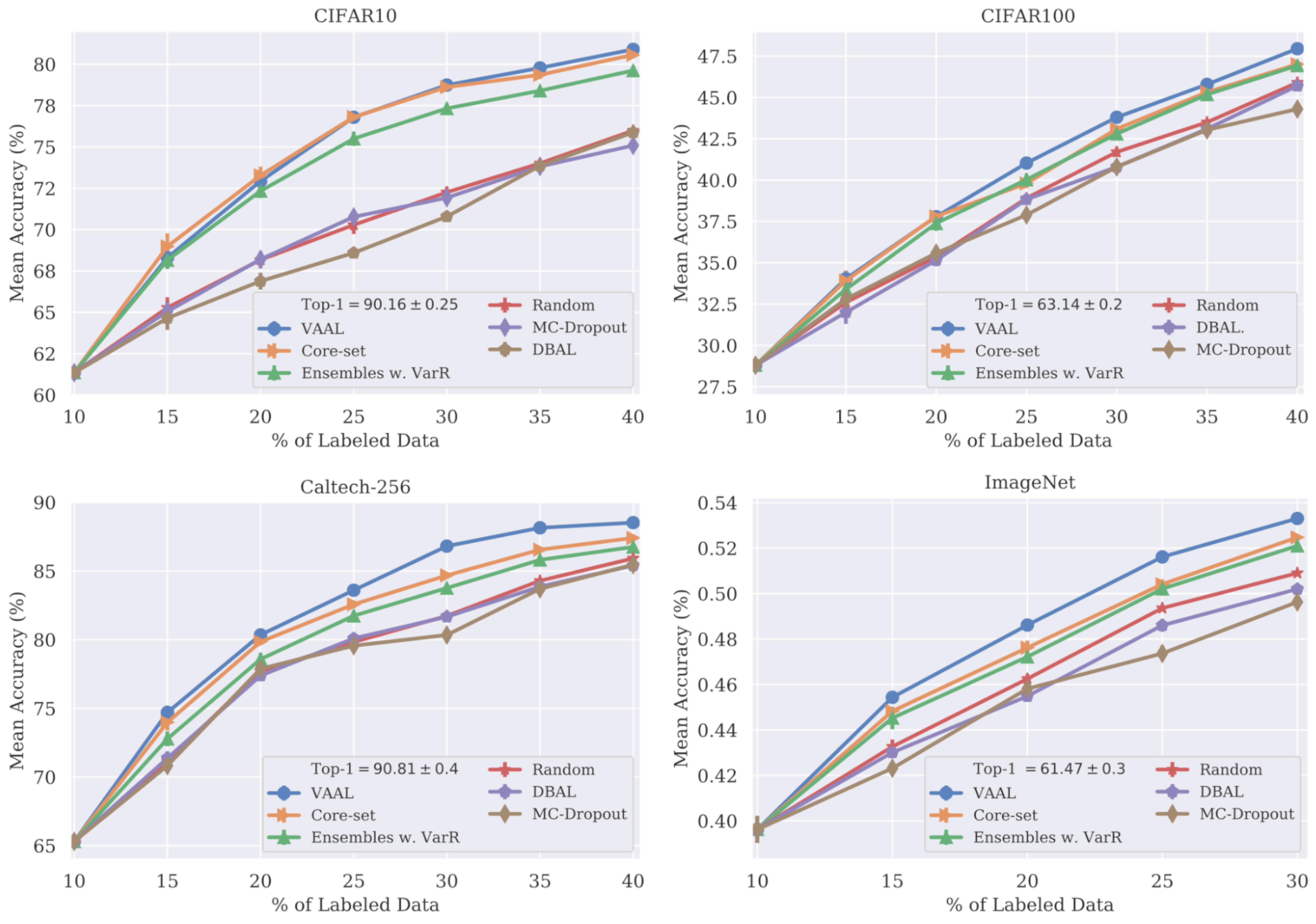
When is Active Learning Worth It?

- AL adds **engineering complexity** on top of the ML pipeline



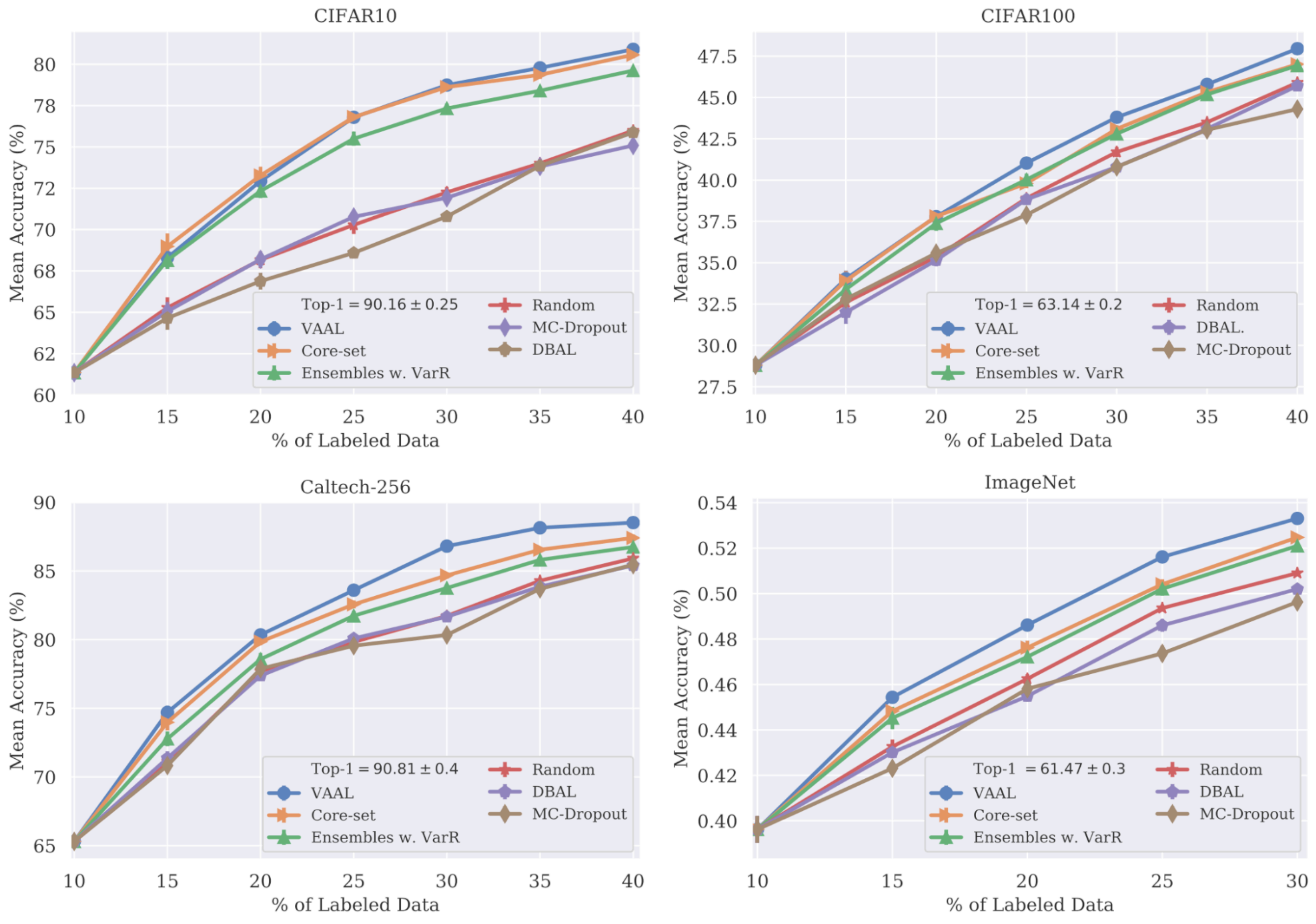
When is Active Learning Worth It?

- AL adds **engineering complexity** on top of the ML pipeline
- Think: is labeling ***that expensive** compared to your training pipeline?
  - X-rays: maybe yes
  - Movie reviews: probably not



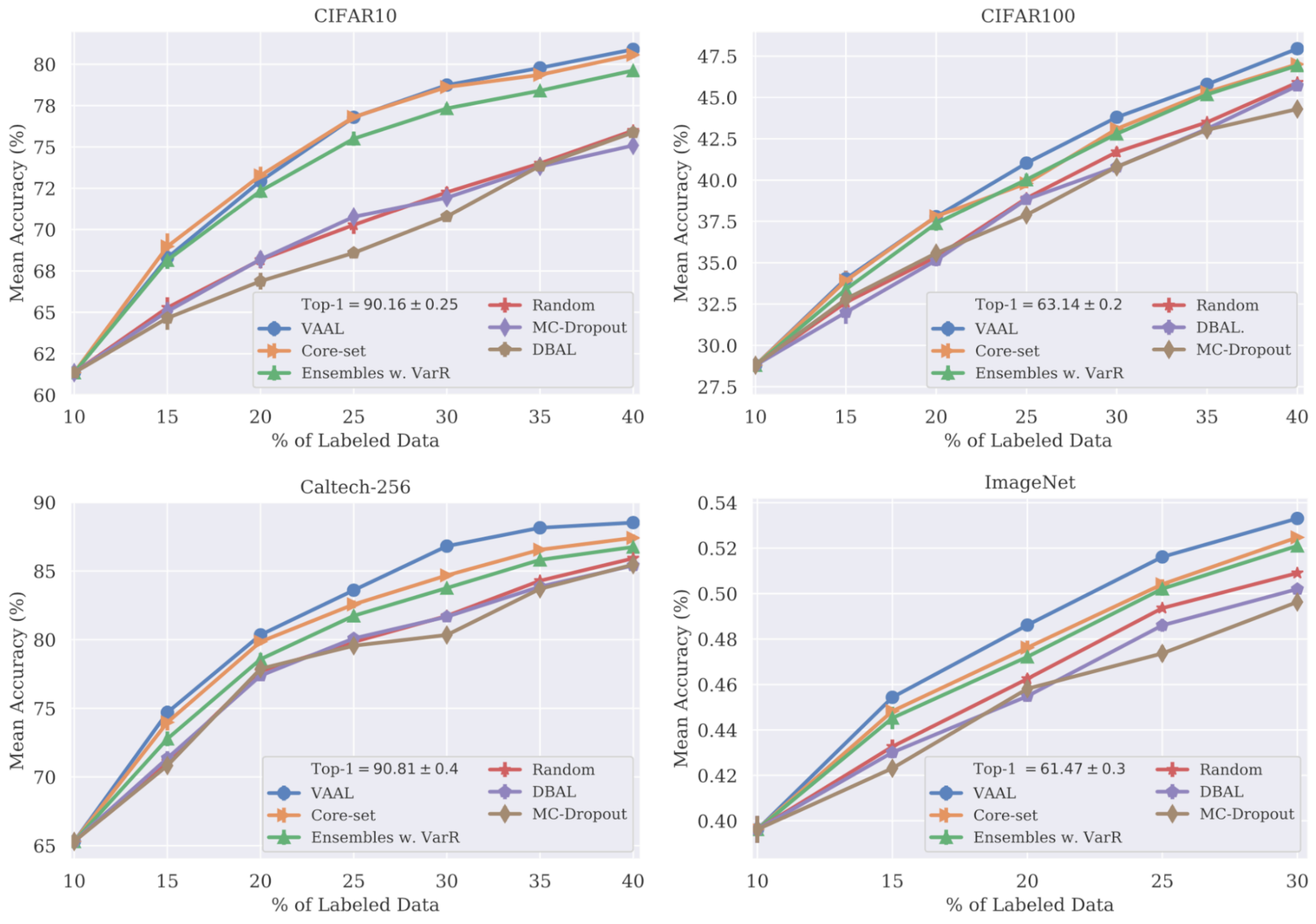
When is Active Learning Worth It?

- AL adds **engineering complexity** on top of the ML pipeline
- Think: is labeling ***that expensive** compared to your training pipeline?
  - X-rays: maybe yes
  - Movie reviews: probably not
- If the task is **intrinsically easy**, it will converge fast no matter what



When is Active Learning Worth It?

- AL adds **engineering complexity** on top of the ML pipeline
- Think: is labeling ***that expensive** compared to your training pipeline?
  - X-rays: maybe yes
  - Movie reviews: probably not
- If the task is **intrinsically easy**, it will converge fast no matter what
- AL evaluated by **performance vs. labels**
  - When the **gap between curves** is large and **labeling cost is high**, AL may be worth it



Weak Supervision

Skipping the Oracle

```python
def lf_keyword_sports(article):
    if any(w in article for w in ["touchdown", "goalkeeper", "innings"]):
        return "sports"
    return ABSTAIN

def lf_source_nyt_politics(article):
    if article.source == "NYT" and article.section == "Politics":
        return "politics"
    return ABSTAIN

def lf_stock_ticker(article):
    if re.search(r'\$[A-Z]{1,5}\b', article.text):
        return "business"
    return ABSTAIN
```

Skipping the Oracle

- Weak Supervision: **skip expensive labeling** and use **cheap heuristics** to generate **noisy labels at scale**

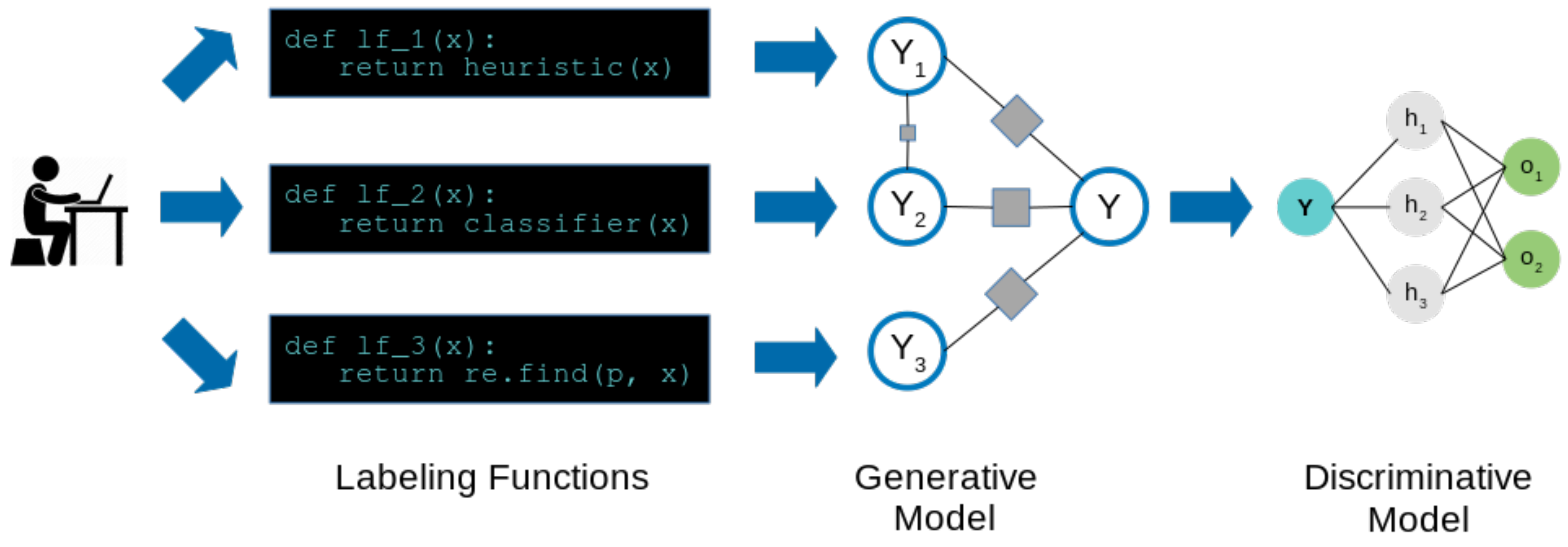
```
def lf_keyword_sports(article):
    if any(w in article for w in ["touchdown", "goalkeeper", "innings"]):
        return "sports"
    return ABSTAIN

def lf_source_nyt_politics(article):
    if article.source == "NYT" and article.section == "Politics":
        return "politics"
    return ABSTAIN

def lf_stock_ticker(article):
    if re.search(r'\$[A-Z]{1,5}\b', article.text):
        return "business"
    return ABSTAIN
```

Skipping the Oracle

- Weak Supervision: **skip expensive labeling** and use **cheap heuristics** to generate **noisy labels at scale**
- Snorkel (Ratner et al., 2017):
  - Use **Labeling Functions** (LFs) to programmatically create labels
  - May result in **absent** or **conflicting labels**, but can be applied **fast**
  - Use **statistical models** to learn **correlations, conflicts** between labels and **assign probabilities**

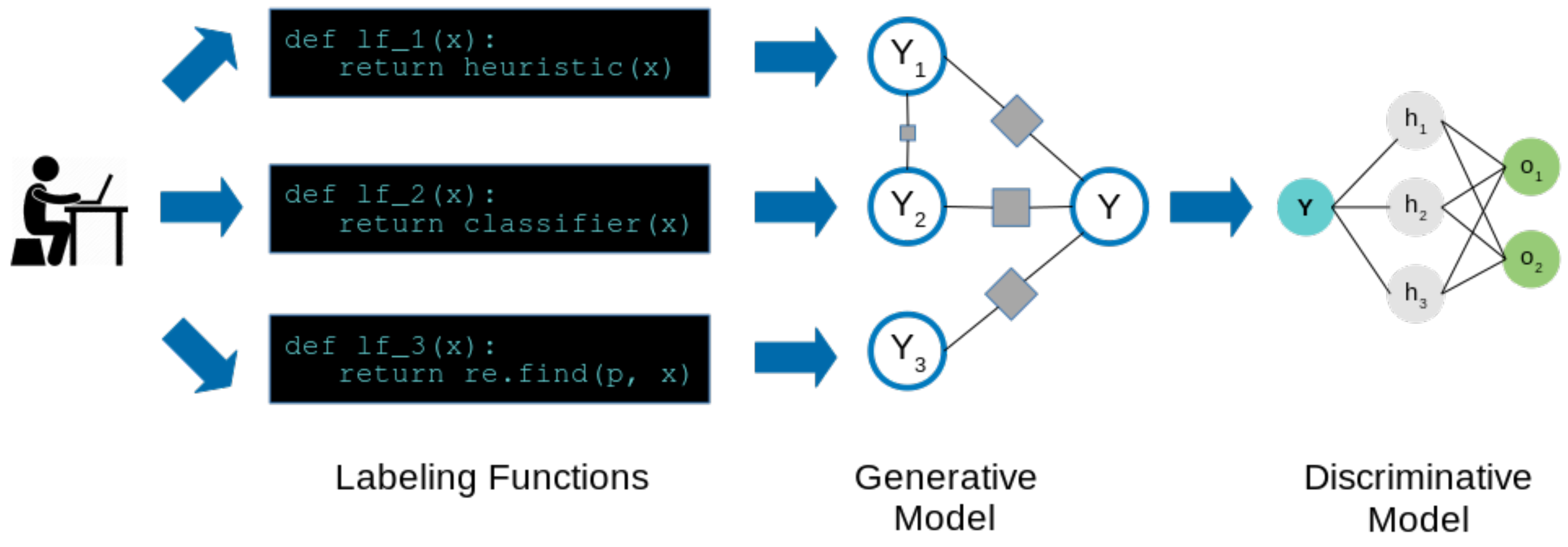
```python
def lf_keyword_sports(article):
    if any(w in article for w in ["touchdown", "goalkeeper", "innings"]):
        return "sports"
    return ABSTAIN

def lf_source_nyt_politics(article):
    if article.source == "NYT" and article.section == "Politics":
        return "politics"
    return ABSTAIN

def lf_stock_ticker(article):
    if re.search(r'\$[A-Z]{1,5}\b', article.text):
        return "business"
    return ABSTAIN
```

Why LFs Work


def lf_1(x):
return heuristic(x)
def lf_2(x):
return classifier(x)
def lf_3(x):
return re.find(p, x)
Y1
Y2
Y3
Y
Y
h1
h2
h3
o1
o2
Labeling Functions
Generative Model
Discriminative Model

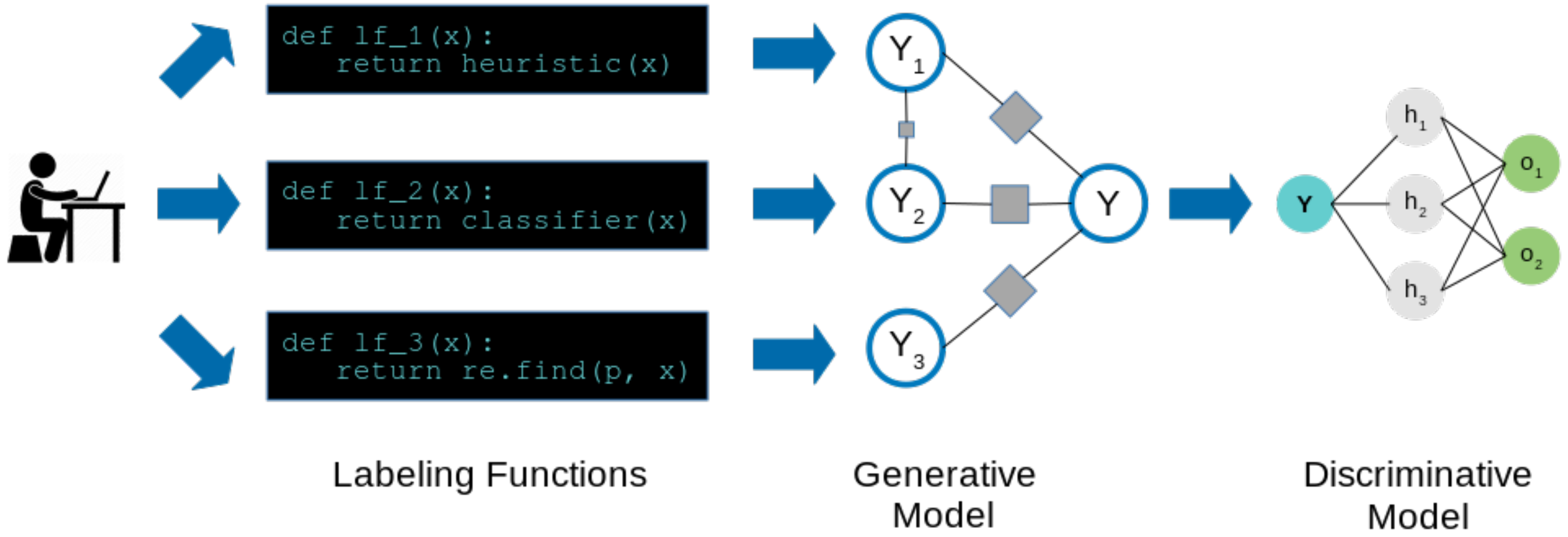
Why LFs Work

- Writing a Labeling Function takes **minutes** as opposed to **hours** of labeling



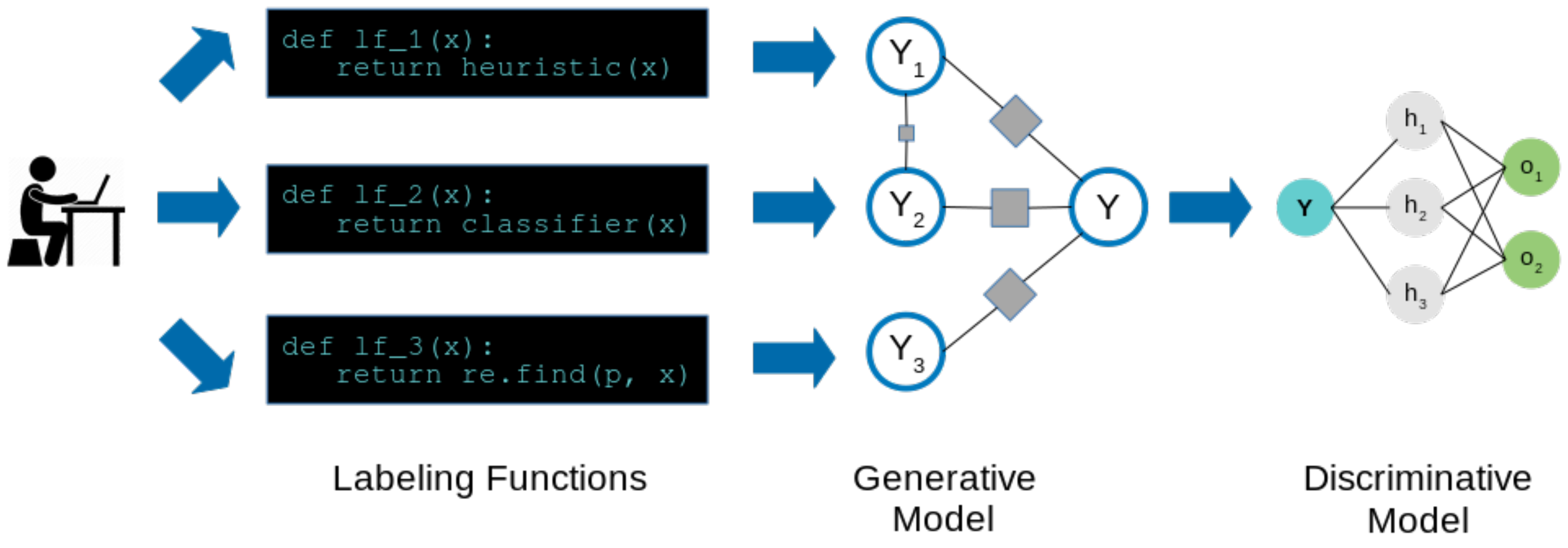
Why LFs Work

- Writing a Labeling Function takes **minutes** as opposed to **hours** of labeling
- LFs themselves can be **written by an expert**
  - i.e. a doctor can provide **expert-curated heuristic rules** from clinic notes



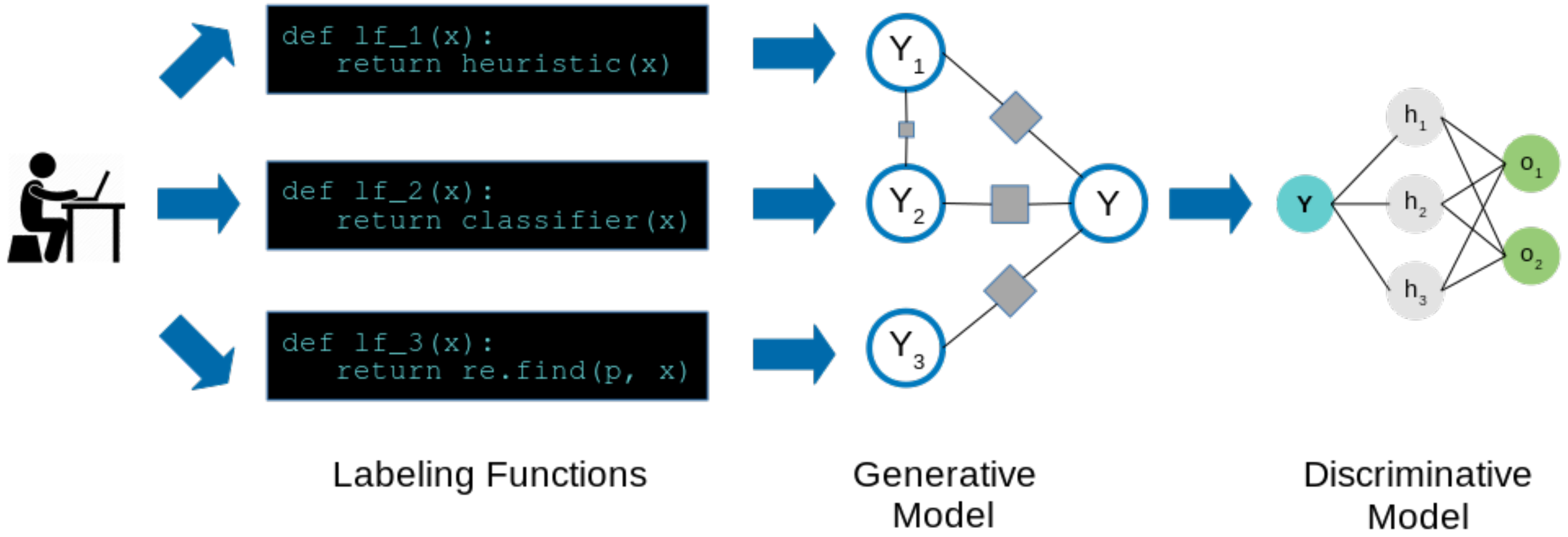
Why LFs Work

- Writing a Labeling Function takes **minutes** as opposed to **hours** of labeling
- LFs themselves can be **written by an expert**
  - i.e. a doctor can provide **expert-curated heuristic rules** from clinic notes
- LFs can be **iteratively improved**, also quickly



Why LFs Work

- Writing a Labeling Function takes **minutes** as opposed to **hours** of labeling
- LFs themselves can be **written by an expert**
  - i.e. a doctor can provide **expert-curated heuristic rules** from clinic notes
- LFs can be **iteratively improved**, also quickly
- The downstream-trained models **generalizes beyond these heuristics**



Other Weak Supervision

Other Weak Supervision

- **Distant Supervision:** use an **existing knowledge base** to generate labels
  - Ex: wikipedia links define a **knowledge graph** that links concepts

Other Weak Supervision

- **Distant Supervision:** use an **existing knowledge base** to generate labels
  - Ex: wikipedia links define a **knowledge graph** that links concepts
- **Crowdsourcing:** get labels from **non-expert annotators**
  - Very common in modern ML! E.g. with **Amazon Mechanical Turk**
  - Crowdworkers tend to spend **low effort**, so annotations are dubious
  - Annotators often **disagree with each other**
  - Working with noisy labels is a whole subfield

Putting it Together

Acquiring Labels (comparison)

	Semi-supervised	Active Learning	Weak Supervision
Label Source	Given (small, fixed)	Oracle	Heuristics
Unlabeled Data Role	Propagate/constrain labels	Pool for selection	Inputs to LFs
Human Involvement	Upfront labeling	Oracle	Heuristics/LFs author
Assumptions	Smoothness, cluster, manifold	Some examples more informative	LFs better than random
Cost	Labels sunk cost	Per-query oracle	Per-LF authoring
When it works	Assumptions met, few labels	Expensive labels	Heuristics work, scale needed

Methods so far

Lecture	Label Usage	Core Idea
L5-6 (Unsupervised)	Leverage unlabeled data	Discover latent structure
L7 (Self-supervised)	Create labels from raw data	"Free" supervision from pretext task
L8 (Semi-supervised)	Propagate few labels	Use unlabeled structure
L9 (Active Learning)	**Choose labels wisely**	**Maximize information-per-label**
L9 (Weak Supervision)	**Generate cheap labels**	**Use heuristics at scale**