Active Fine-Tuning
of Large Neural Networks

Jonas Hübotter

with Bhavya Sukhija, Lenart Treven, Yarden As, Andreas Krause

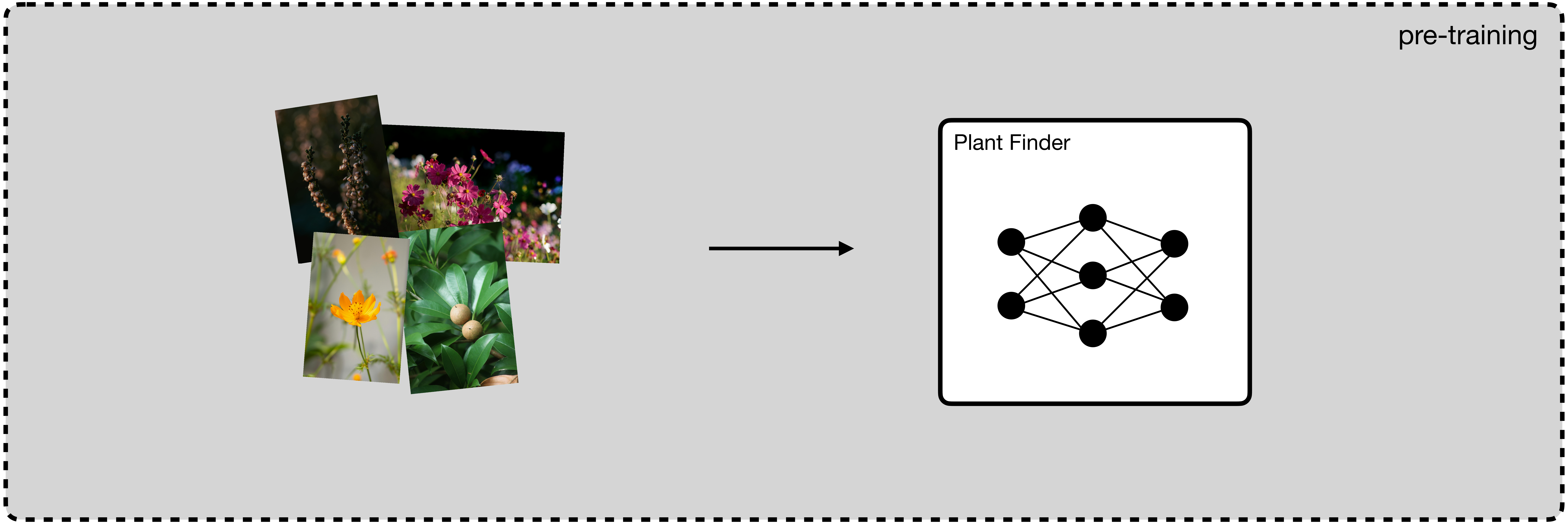



What I will *not* talk about: How to use a small dataset for fine-tuning

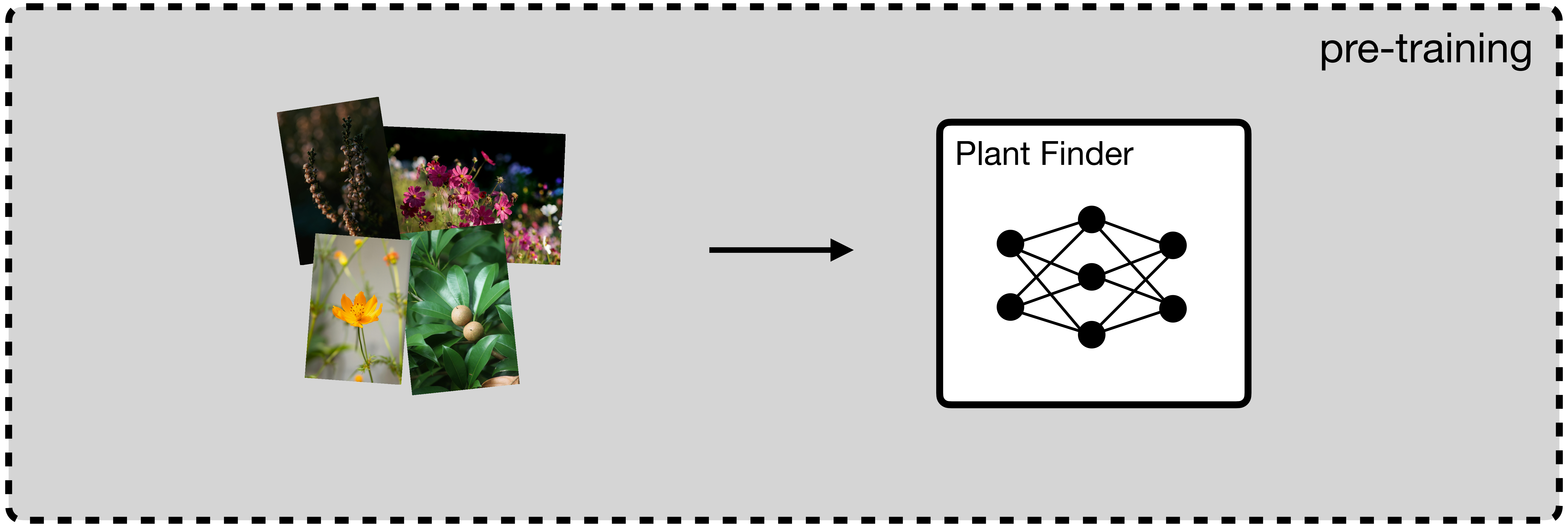
What I will *not* talk about: How to use a small dataset for fine-tuning

What I aim to convince you of:
Retrieving the **right** examples for fine-tuning can lead to substantial performance gains

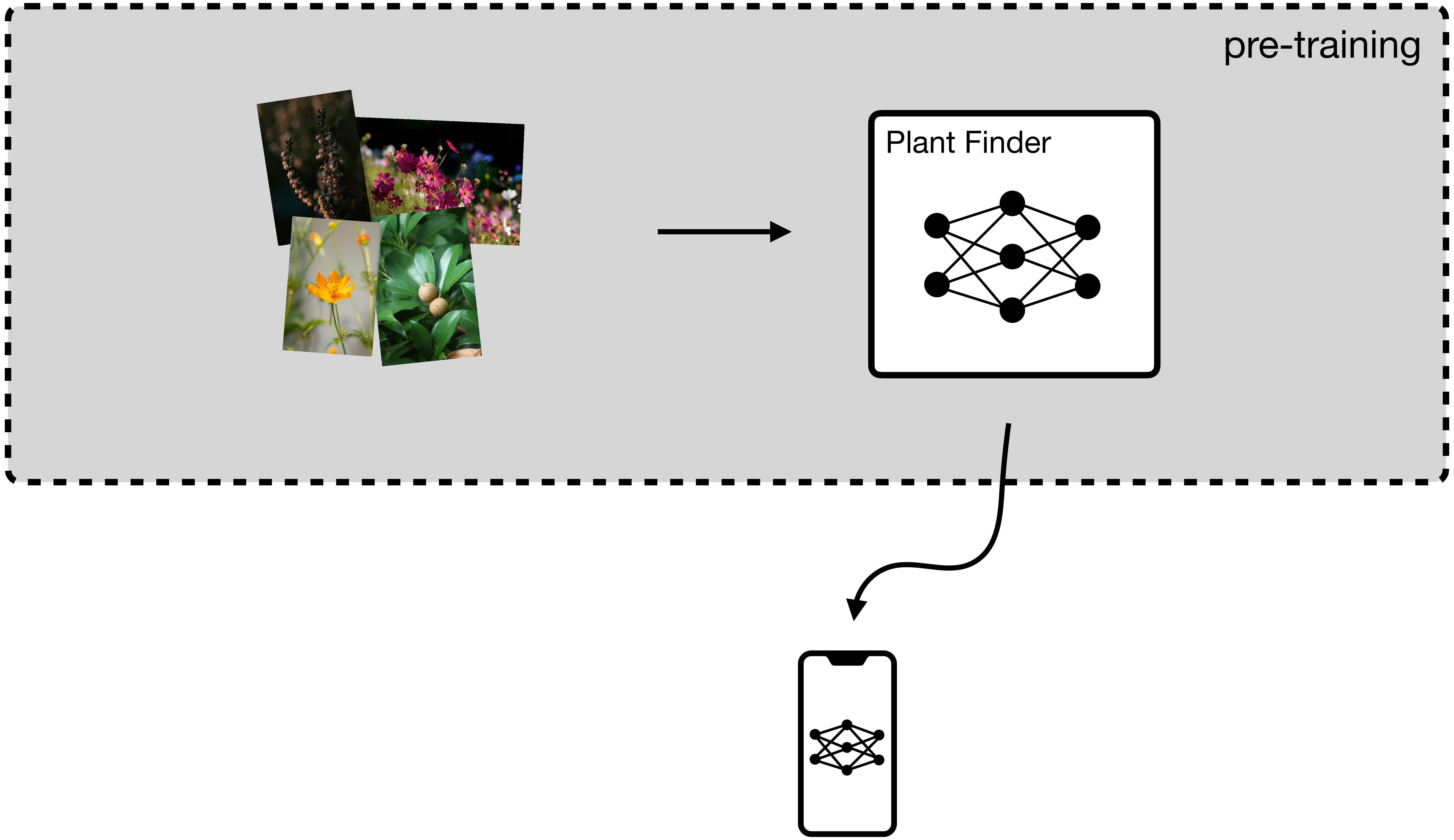
A Motivating Example



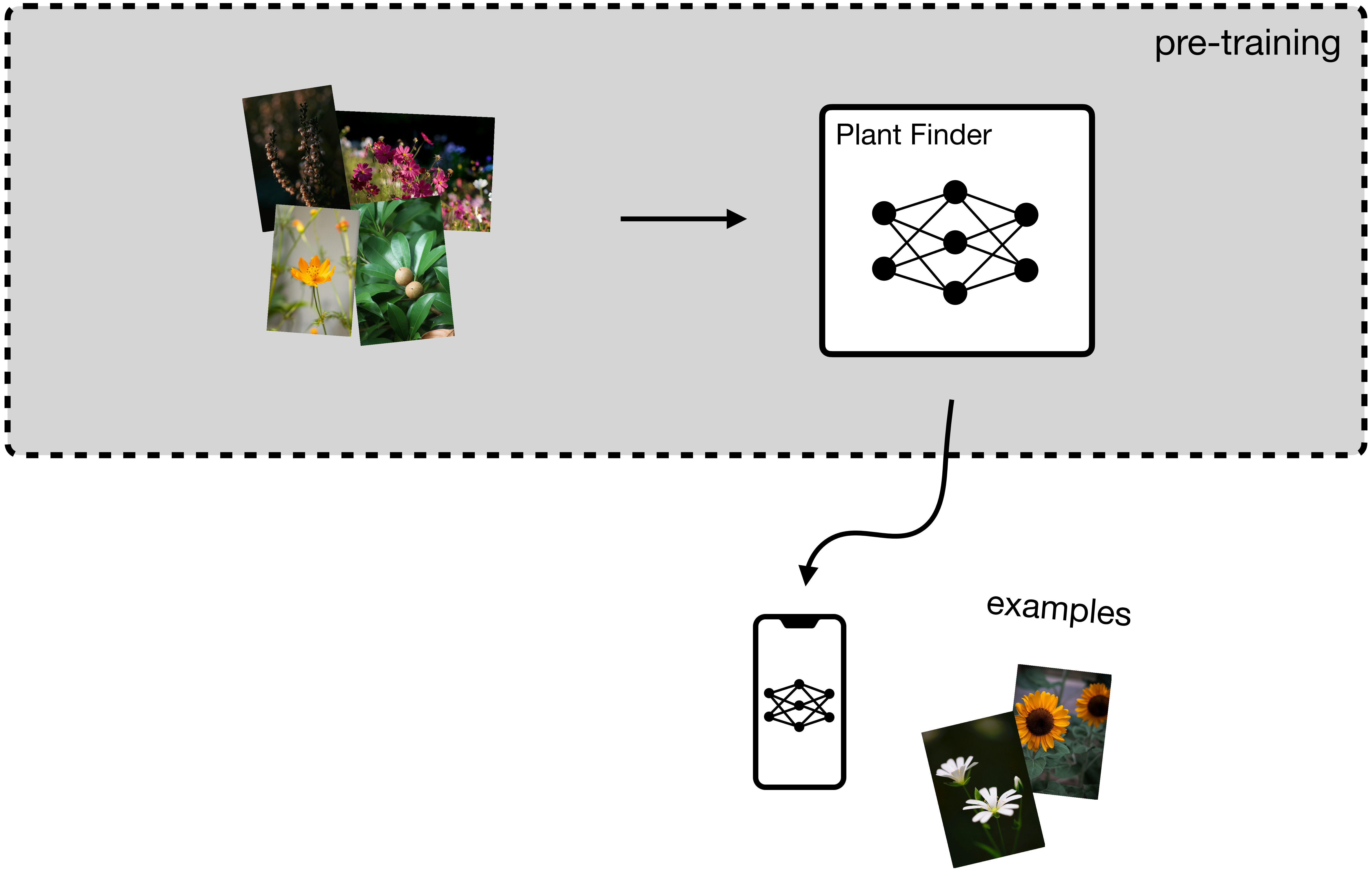
A Motivating Example



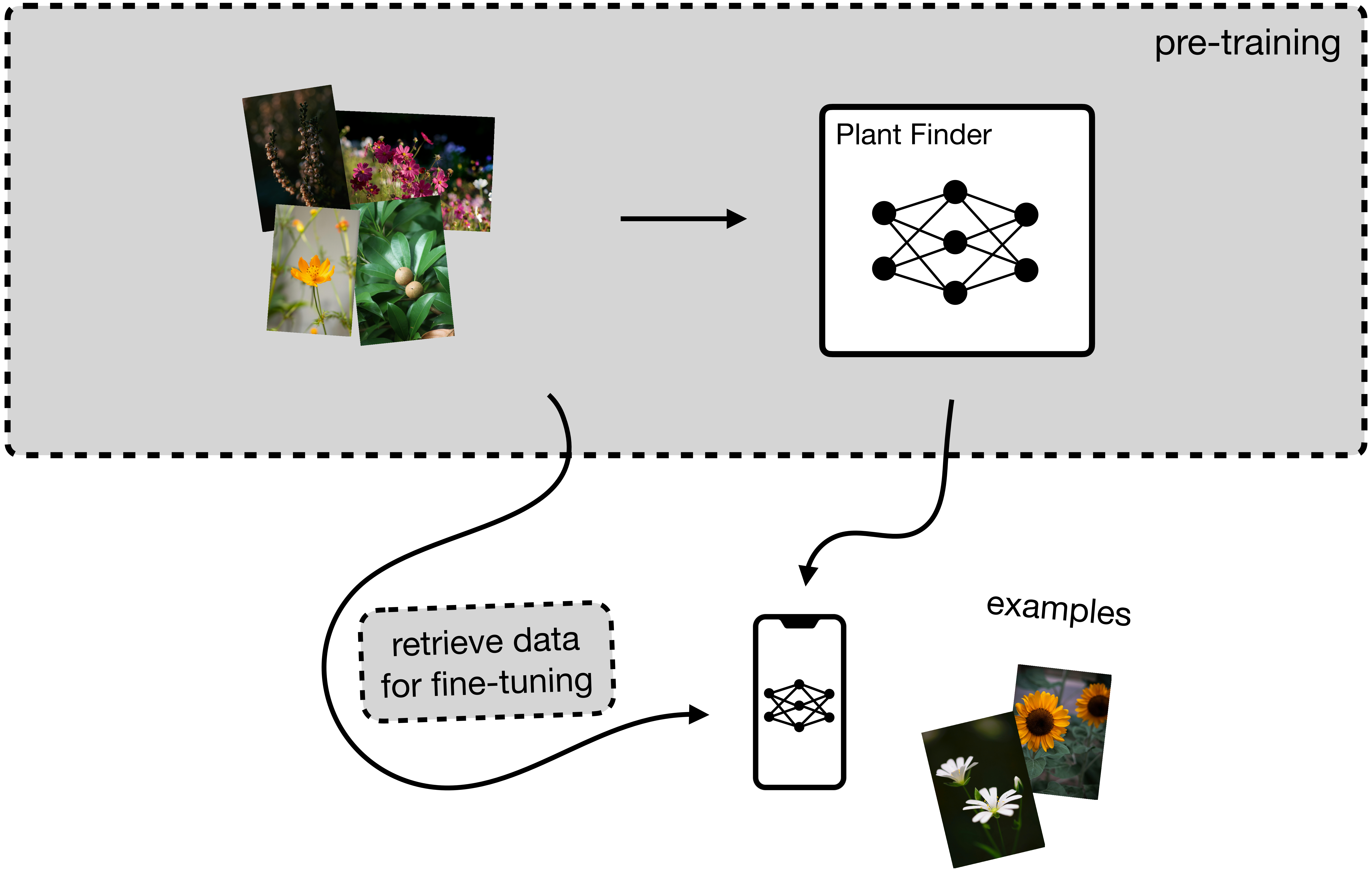
A Motivating Example



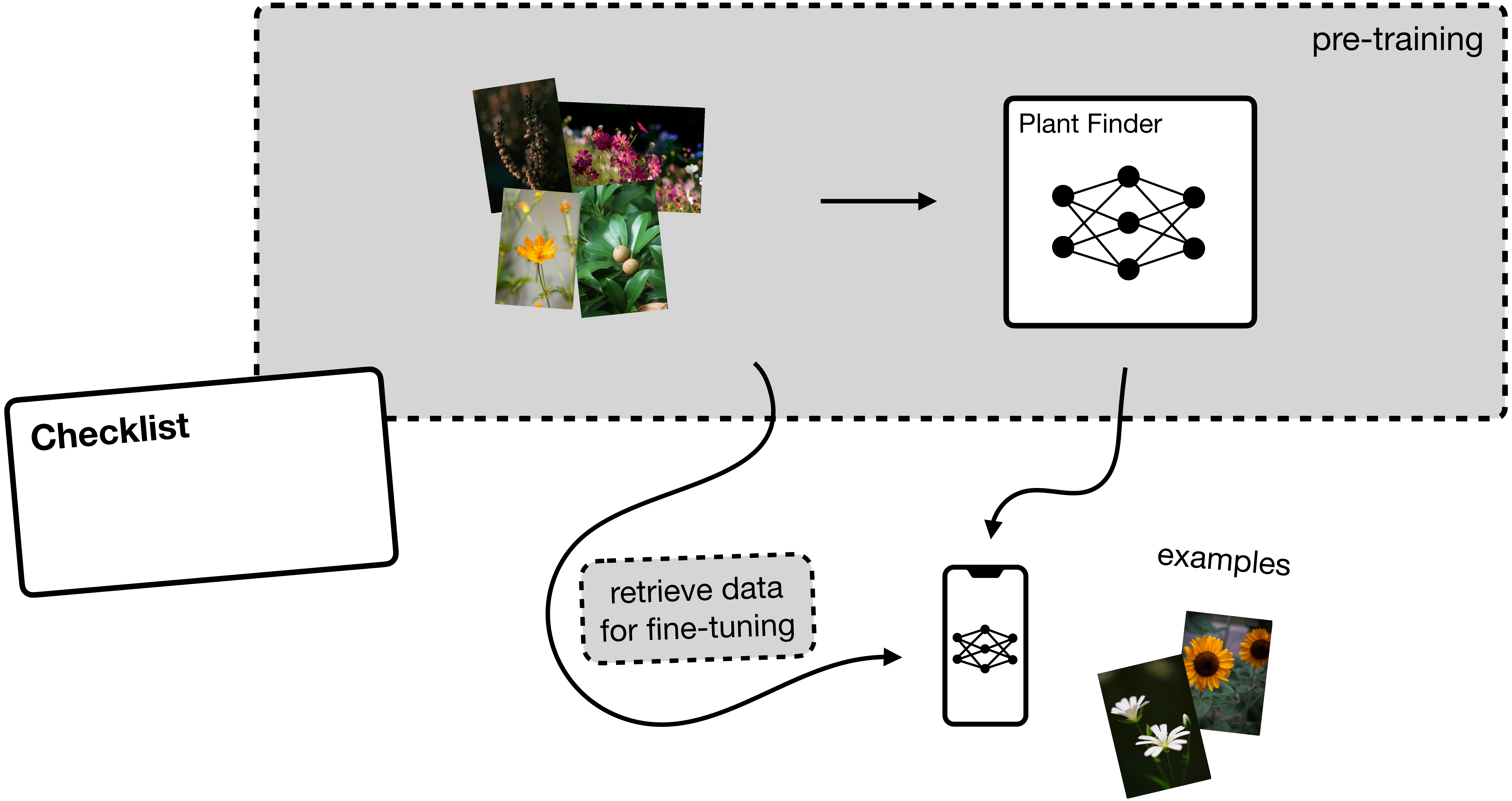
A Motivating Example



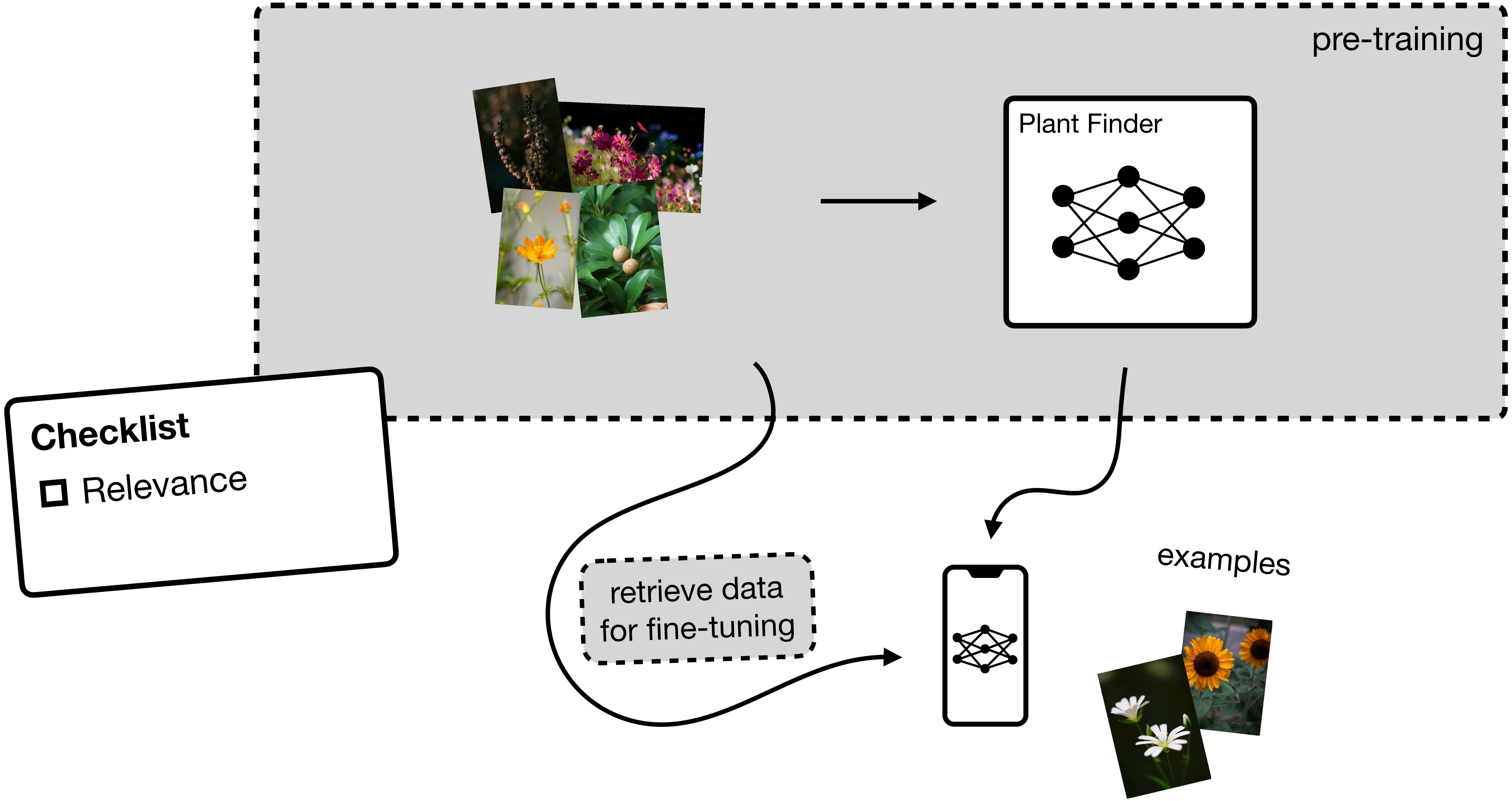
A Motivating Example



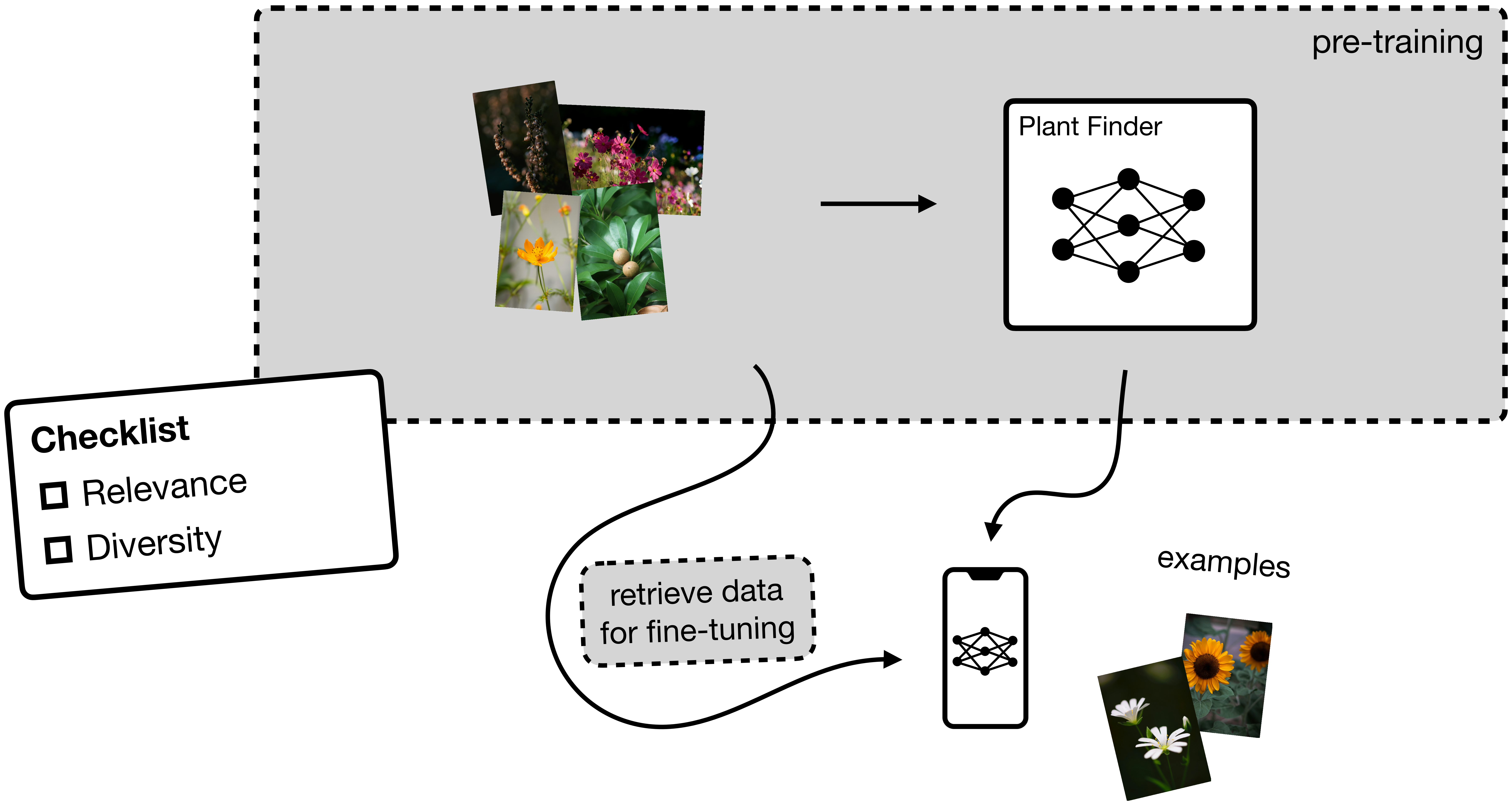
A Motivating Example



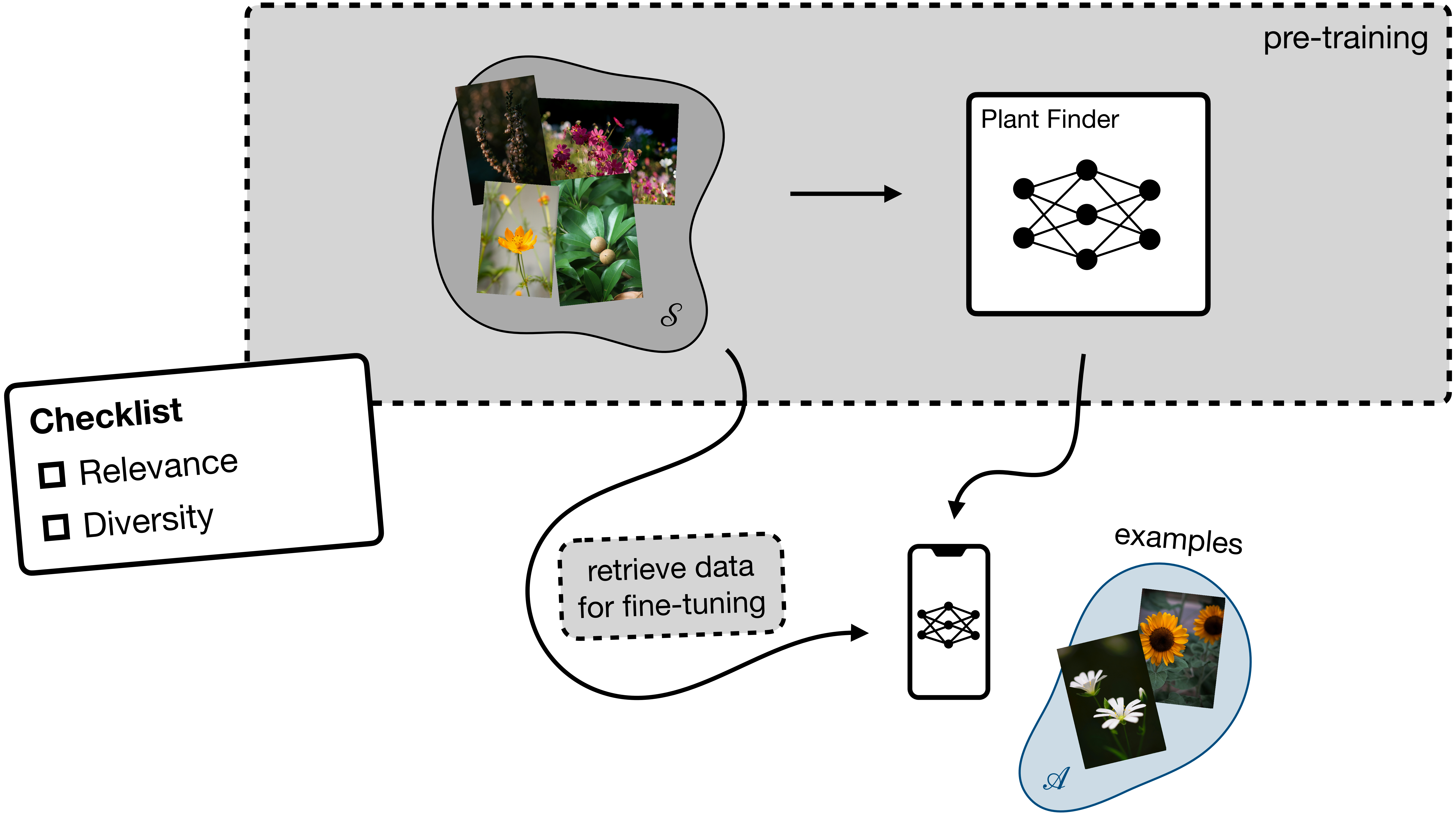
A Motivating Example



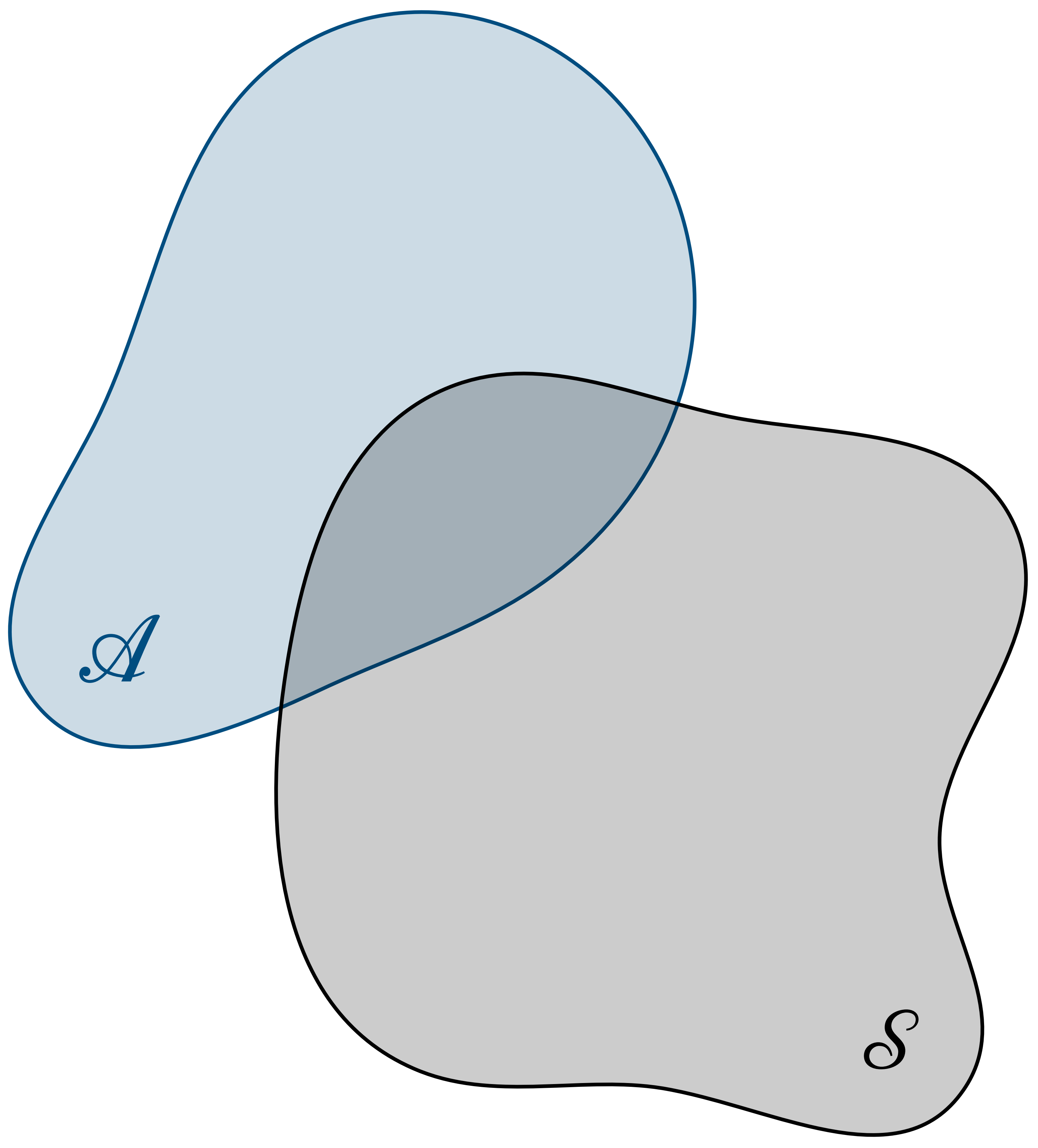
A Motivating Example



A Motivating Example

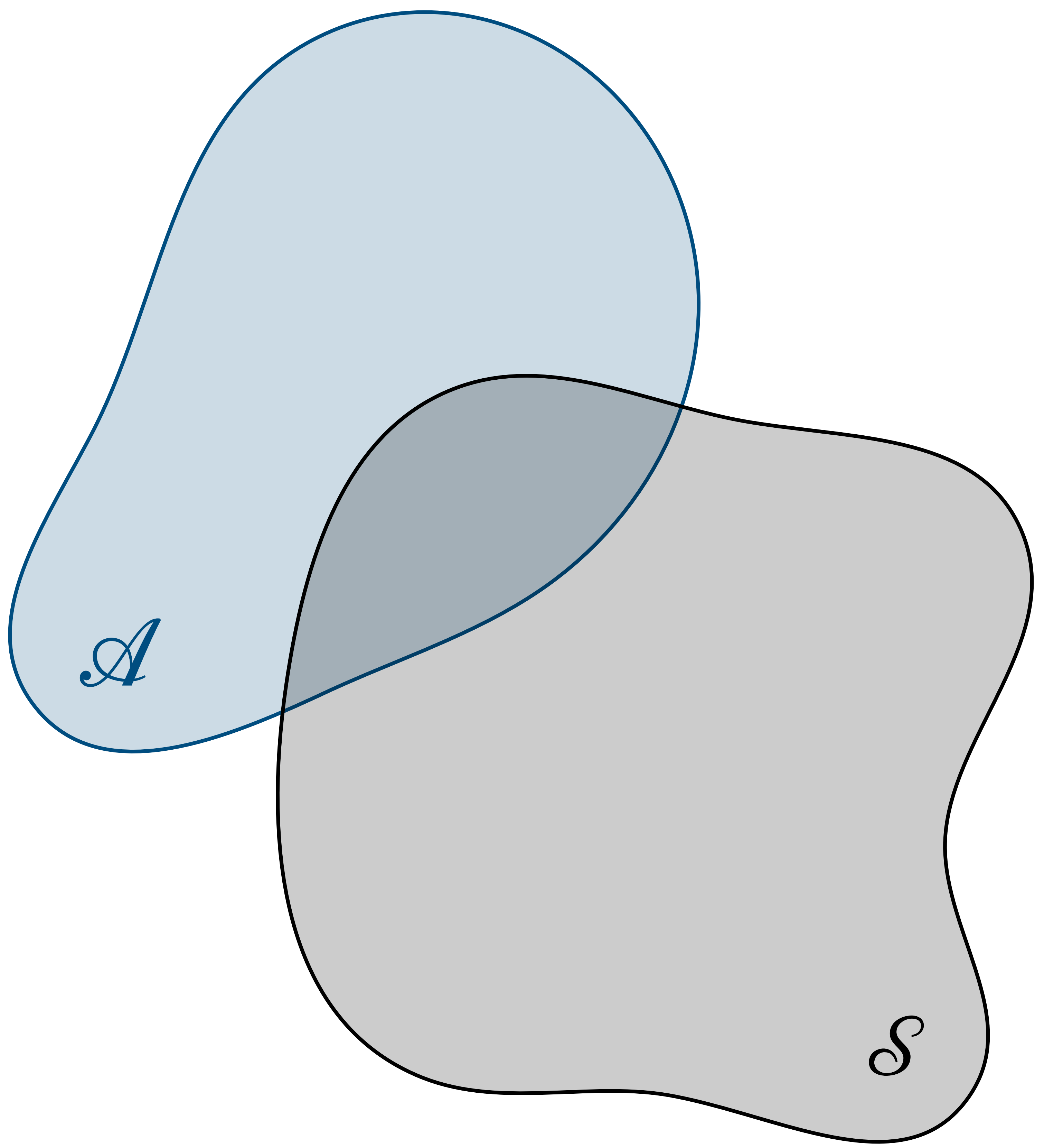


Setting



Setting

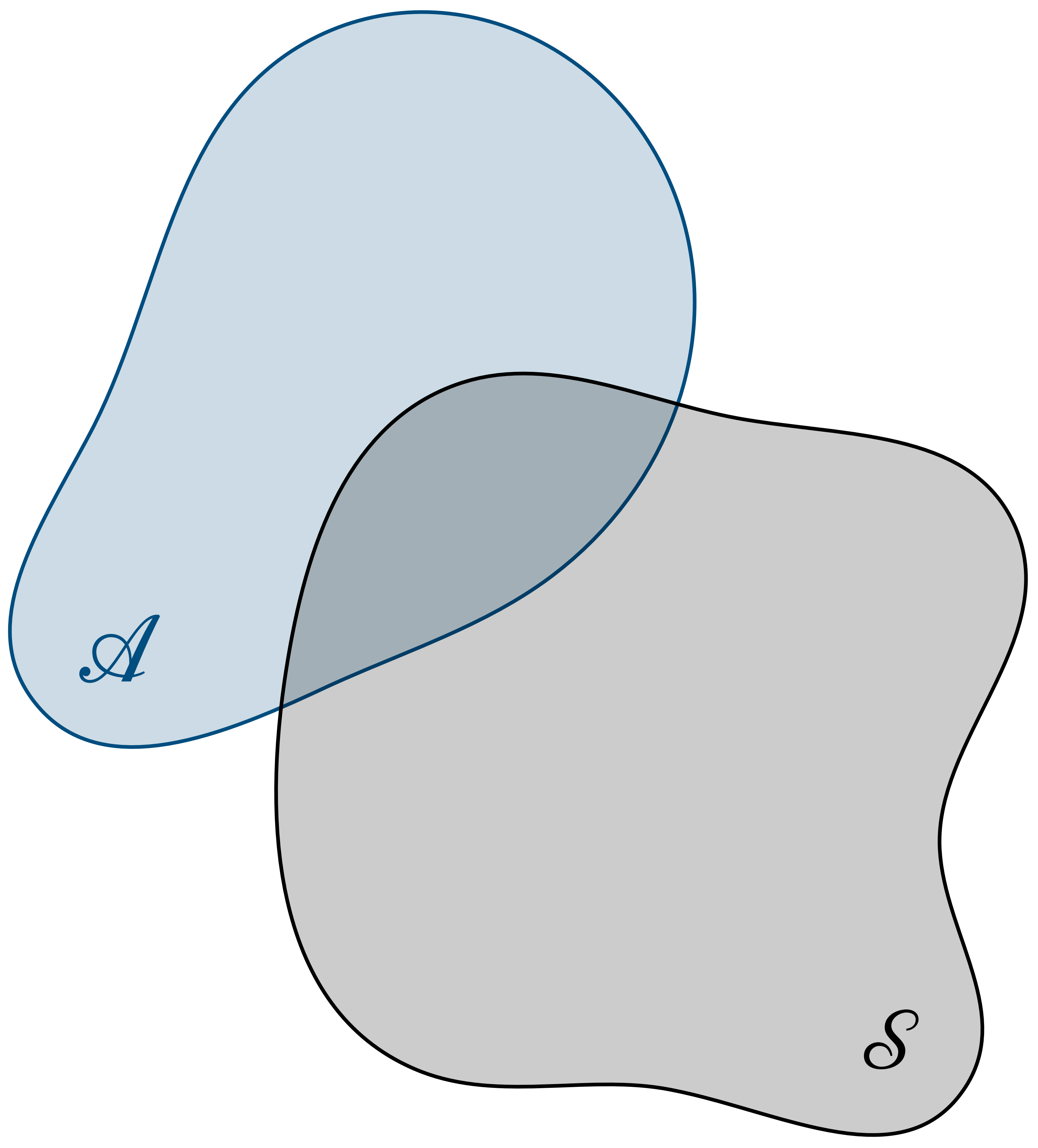
- $\mathscr{S} \subseteq \mathscr{X}$ - sample space
- $\mathscr{A} \subseteq \mathscr{X}$ - target space
- Unknown function f over $\mathscr{X}$



Setting

- $\mathscr{S} \subseteq \mathscr{X}$ - sample space
- $\mathscr{A} \subseteq \mathscr{X}$ - target space
- Unknown function f over $\mathscr{X}$

Goal: Learn f within $\mathscr{A}$ by sampling from $\mathscr{S}$

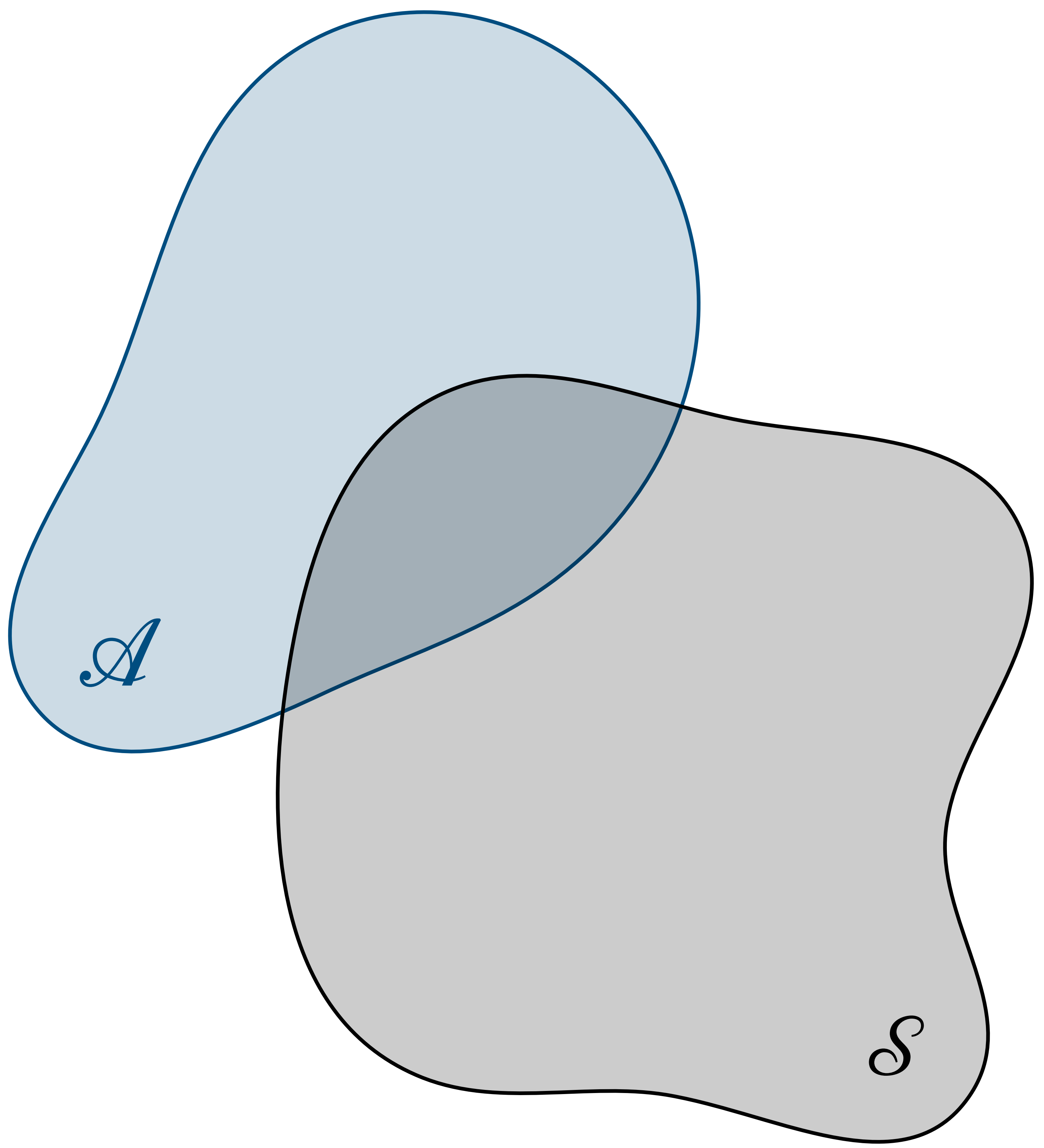


Setting

- $\mathcal{S} \subseteq \mathcal{X}$ - sample space
- $\mathcal{A} \subseteq \mathcal{X}$ - target space
- Unknown function f over $\mathcal{X}$

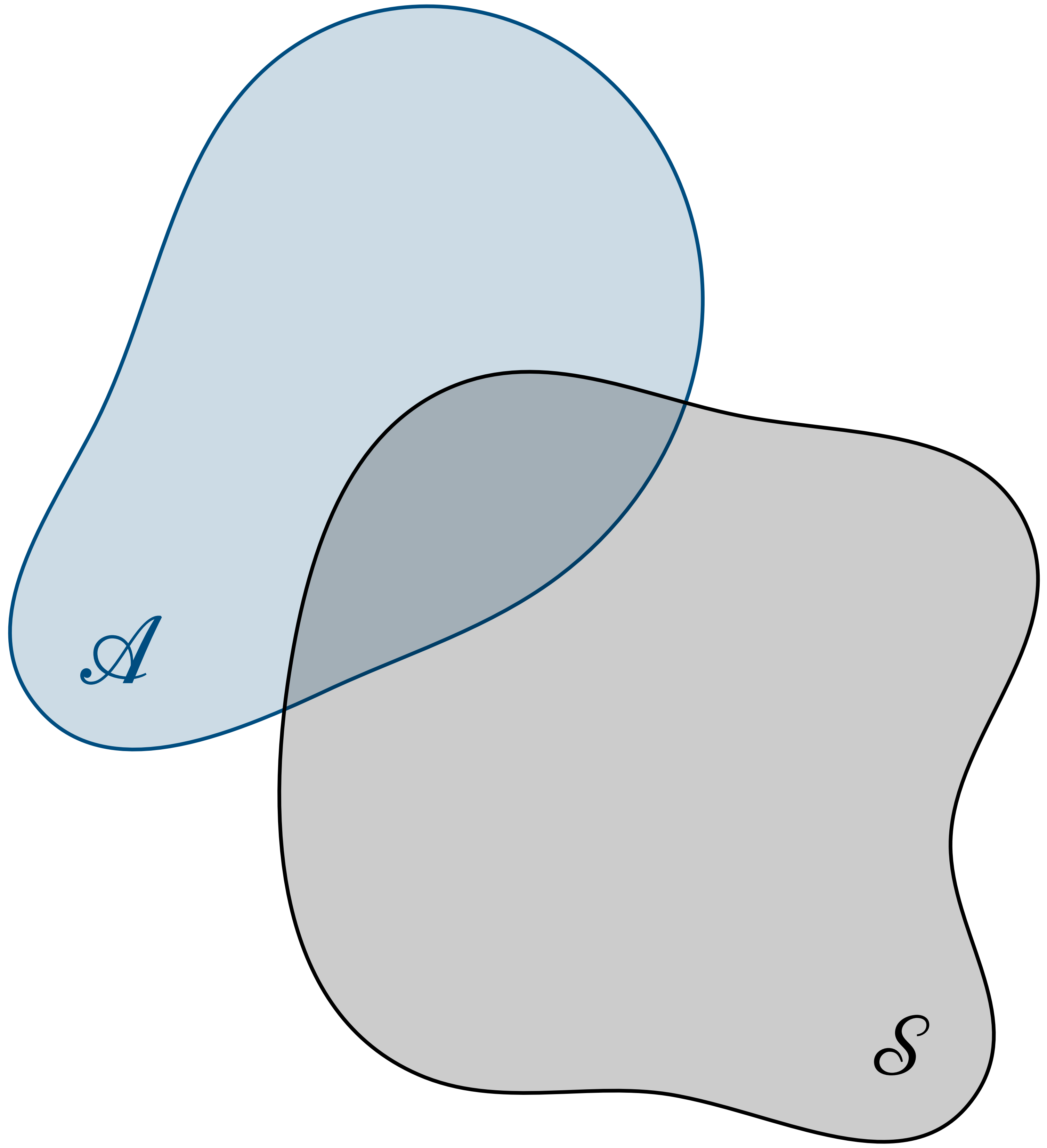
Goal: Learn f within $\mathcal{A}$ by sampling from $\mathcal{S}$

We call this **Transductive Active Learning**, generalizing classical Active Learning



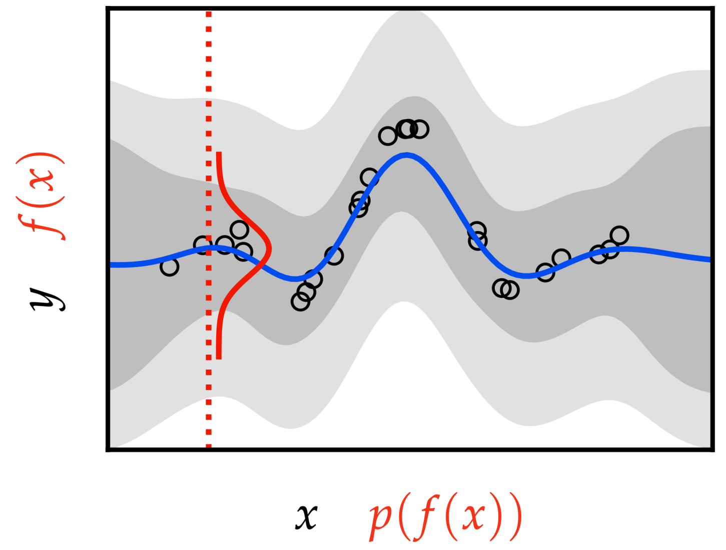
Setting

- $\mathscr{S} \subseteq \mathscr{X}$ - sample space
- $\mathscr{A} \subseteq \mathscr{X}$ - target space
- Unknown function f over $\mathscr{X}$

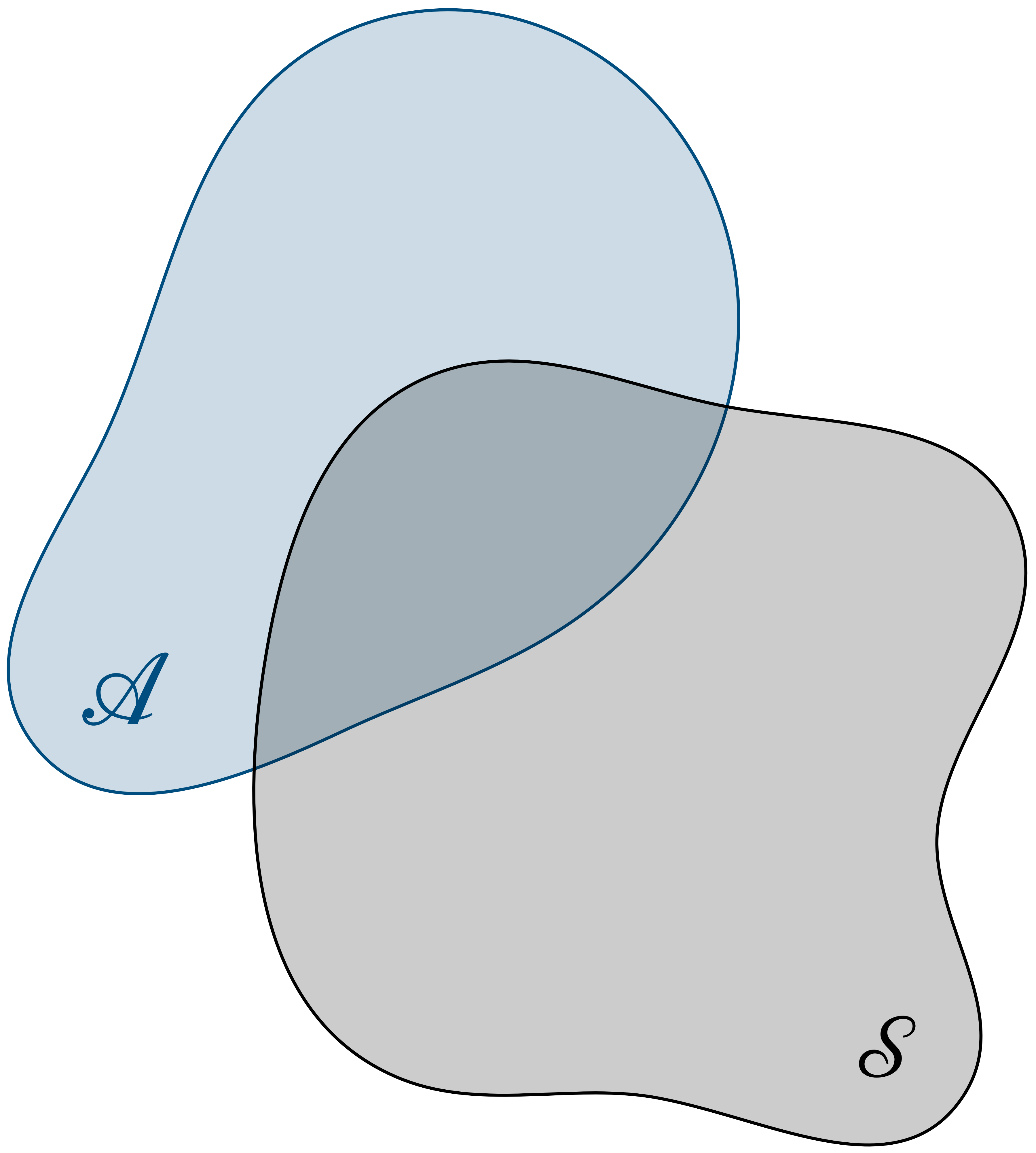


Goal: Learn f within $\mathscr{A}$ by sampling from $\mathscr{S}$

We call this **Transductive Active Learning**, generalizing classical Active Learning

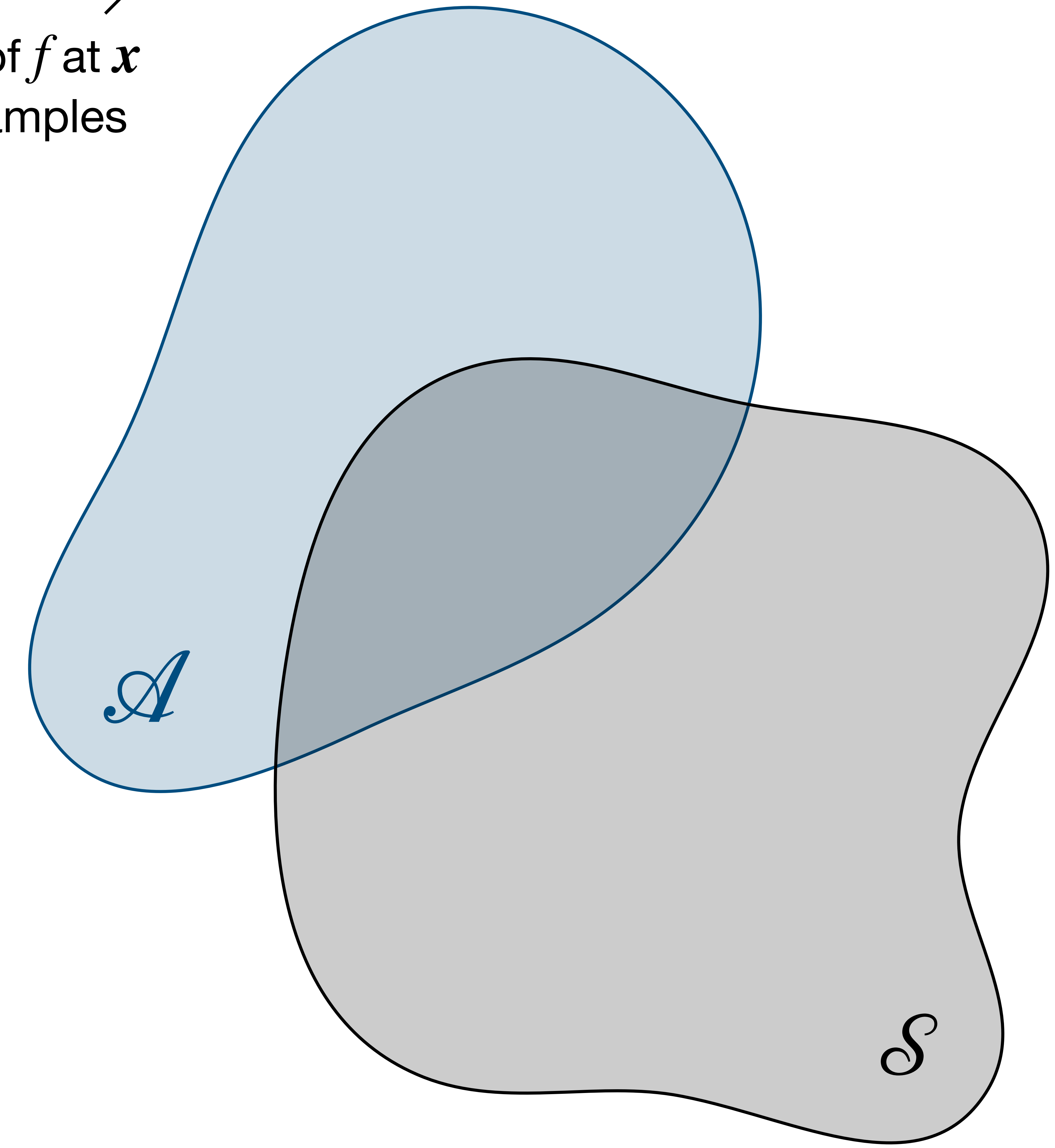


Assume for us: $\mathscr{S}, \mathscr{A}$ finite, and NN f is approximated by a Gaussian process with kernel $k(\boldsymbol{x}, \boldsymbol{x}') = \boldsymbol{\phi}(\boldsymbol{x})^\top \boldsymbol{\phi}(\boldsymbol{x}')$ where $\boldsymbol{\phi}(\cdot)$ are embeddings generated by the NN

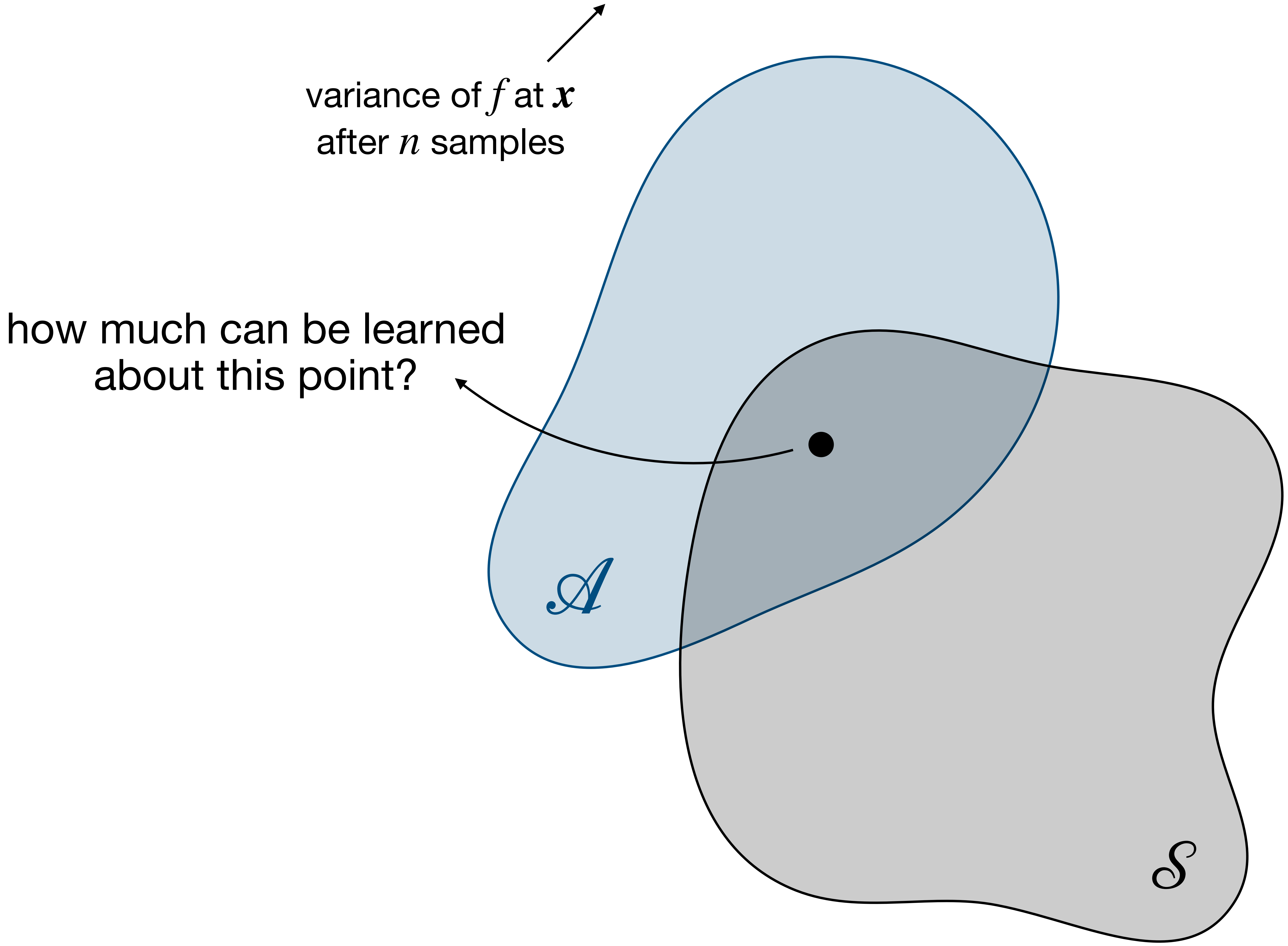

$\mathscr{A}$
$\mathscr{S}$

Goal: Reduce uncertainty $\sigma_n^2(\boldsymbol{x})$ at $\boldsymbol{x} \in \mathscr{A}$

variance of f at $\boldsymbol{x}$
after n samples



Goal: Reduce uncertainty $\sigma_n^2(\boldsymbol{x})$ at $\boldsymbol{x} \in \mathscr{A}$

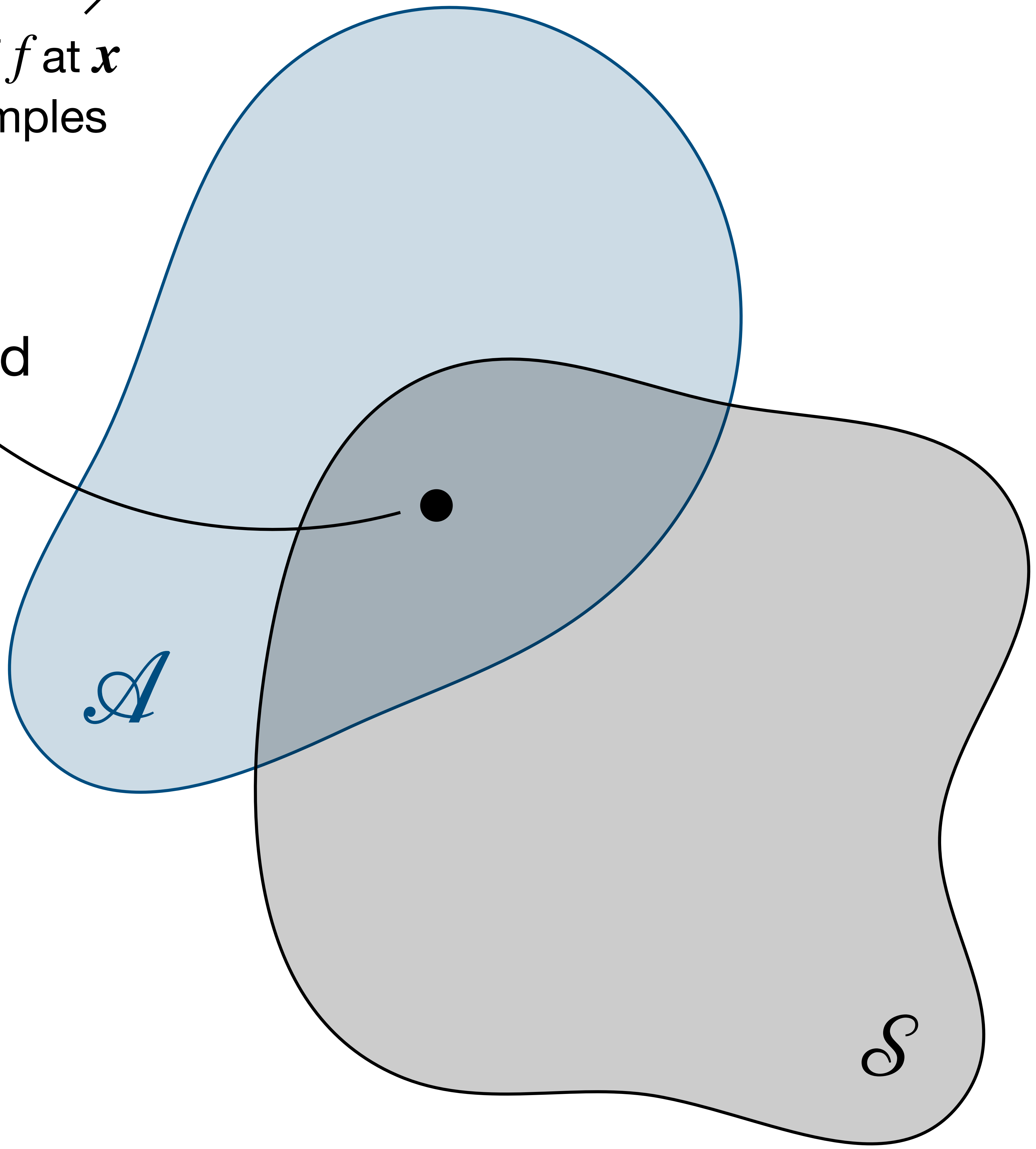


Goal: Reduce uncertainty $\sigma_n^2(\boldsymbol{x})$ at $\boldsymbol{x} \in \mathscr{A}$

variance of f at $\boldsymbol{x}$
after n samples

how much can be learned
about this point?

$\sigma_n^2(\boldsymbol{x}) \to 0$ as $n \to \infty$

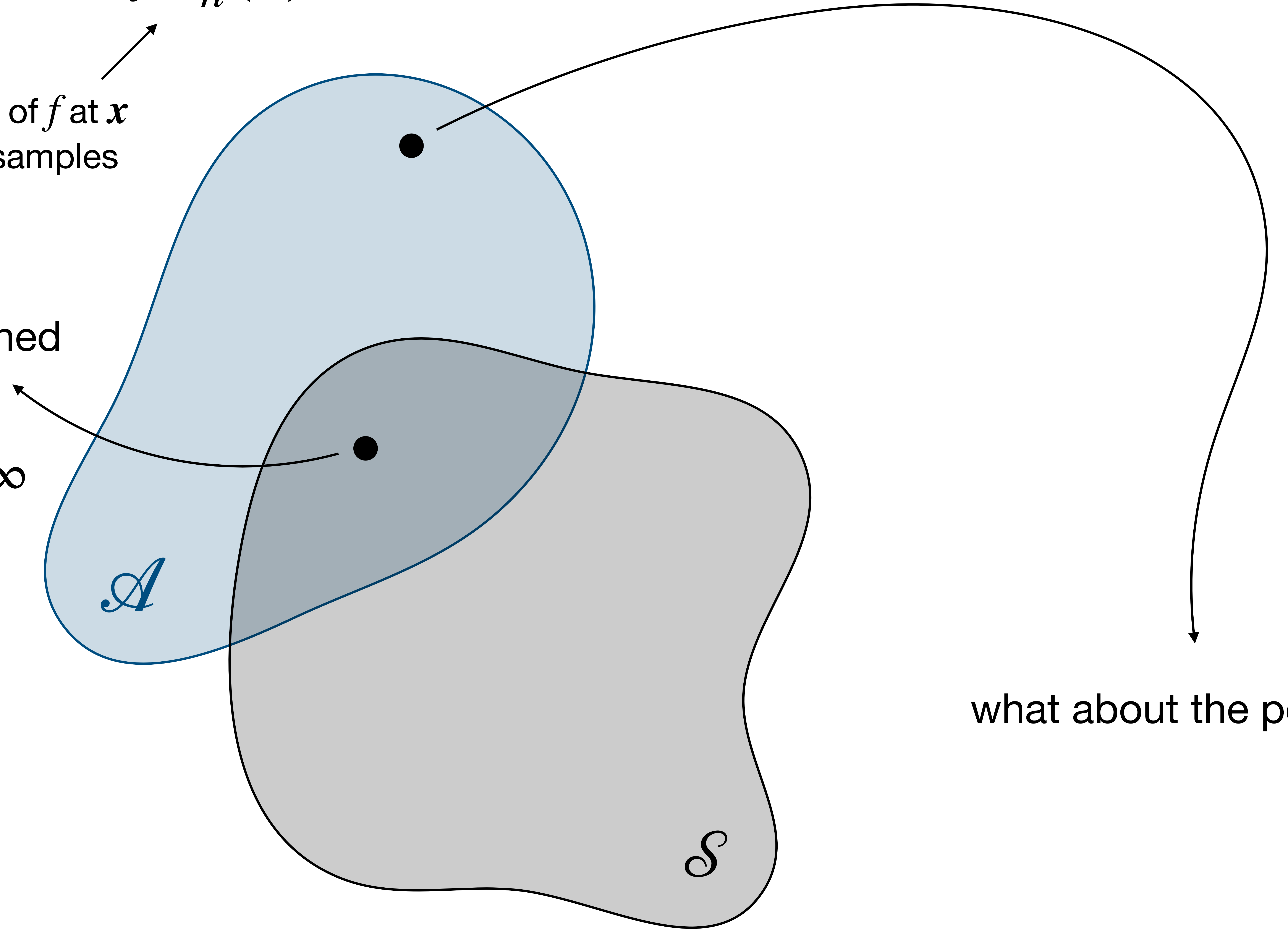


Goal: Reduce uncertainty $\sigma_n^2(\boldsymbol{x})$ at $\boldsymbol{x} \in \mathcal{A}$

variance of f at $\boldsymbol{x}$
after n samples

how much can be learned
about this point?

$\sigma_n^2(\boldsymbol{x}) \to 0$ as $n \to \infty$



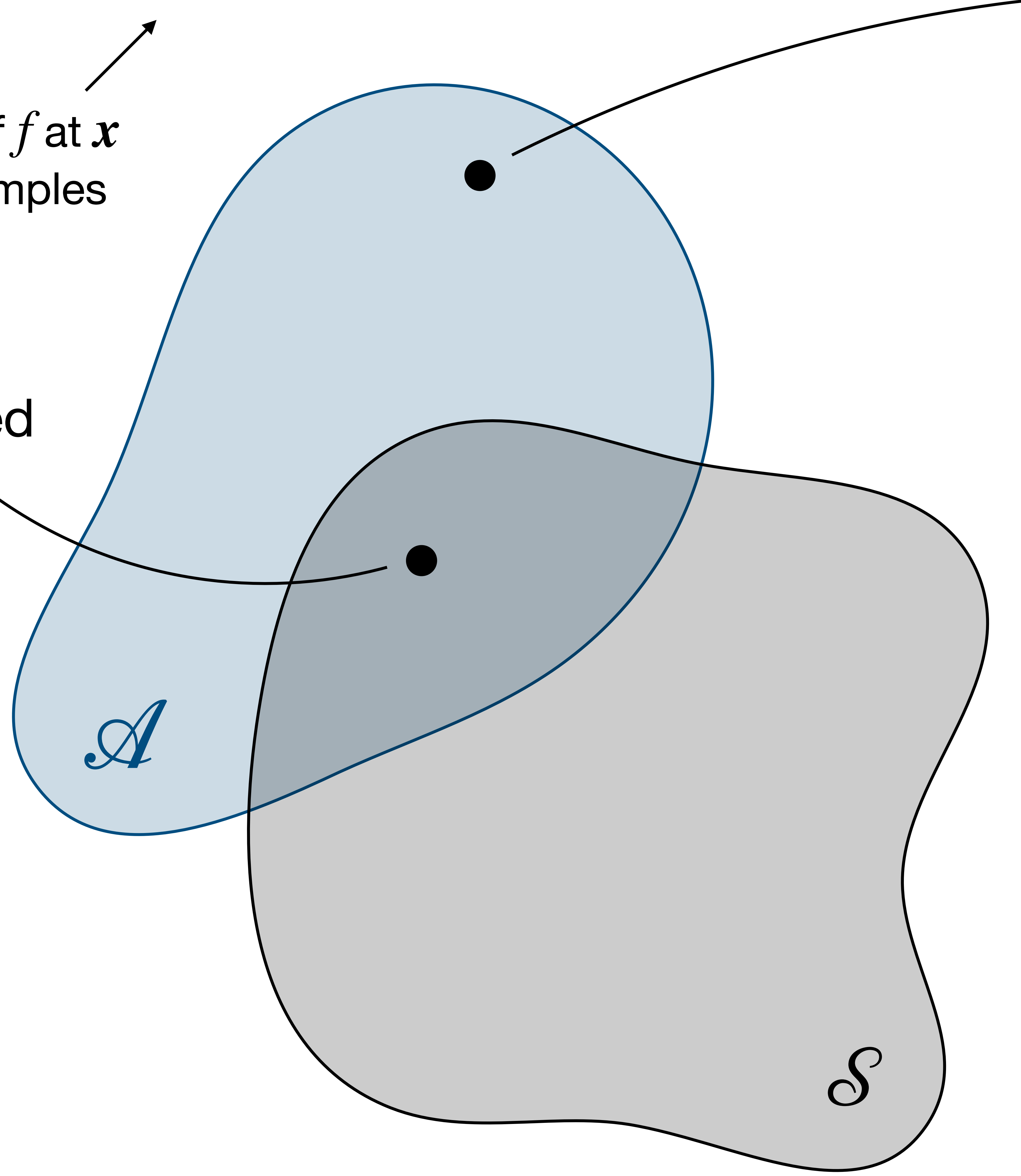
what about the point $\boldsymbol{x}'$?

Goal: Reduce uncertainty $\sigma_n^2(\boldsymbol{x})$ at $\boldsymbol{x} \in \mathscr{A}$

variance of f at $\boldsymbol{x}$
after n samples

how much can be learned
about this point?

$\sigma_n^2(\boldsymbol{x}) \to 0$ as $n \to \infty$



what about the point $\boldsymbol{x}'$?

$\sigma_n^2(\boldsymbol{x}') \overset{?}{\to} \eta_{\mathscr{S}}^2(\boldsymbol{x}')$ as $n \to \infty$

where $\eta_{\mathscr{S}}^2(\boldsymbol{x}') = \mathrm{Var}[f(\boldsymbol{x}') \mid \boldsymbol{f}(\mathscr{S})]$ is
the **irreducible uncertainty**

ITL: Information-directed Transductive Learning

Proposal: select the next sample to minimize *posterior* uncertainty within $\mathscr{A}$

ITL: Information-directed Transductive Learning

Proposal: select the next sample to minimize *posterior* uncertainty within $\mathcal{A}$

$$\underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\max}\ \mathrm{I}(\boldsymbol{f}(\mathcal{A}); y_n \mid D_{n-1}) = \underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\min}\ \mathrm{H}[\boldsymbol{f}(\mathcal{A}) \mid D_{n-1}, (\boldsymbol{x}_n, y_n)]$$

ITL: Information-directed Transductive Learning

Proposal: select the next sample to minimize *posterior* uncertainty within $\mathscr{A}$

$$\underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\max}\ \mathrm{I}(\boldsymbol{f}(\mathscr{A}); y_n \mid D_{n-1}) = \underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\min}\ \mathrm{H}[\boldsymbol{f}(\mathscr{A}) \mid D_{n-1}, (\boldsymbol{x}_n, y_n)]$$

MacKay, 1992

ITL: Information-directed Transductive Learning

Proposal: select the next sample to minimize *posterior* uncertainty within $\mathscr{A}$

$$\underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\max}\ \mathrm{I}(\boldsymbol{f}(\mathscr{A}); y_n \mid D_{n-1}) = \underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\min}\ \mathrm{H}[\boldsymbol{f}(\mathscr{A}) \mid D_{n-1}, (\boldsymbol{x}_n, y_n)]$$

Generalization bound for ITL (informal). $\forall \boldsymbol{x}' \in \mathscr{A}$:

$$\sigma_n^2(\boldsymbol{x}') \leq \eta_{\mathcal{S}}^2(\boldsymbol{x}') + C \log n / \sqrt{n}$$ (C is a constant)

ITL: Information-directed Transductive Learning

Proposal: select the next sample to minimize *posterior* uncertainty within $\mathcal{A}$

$$\underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\max}\ \mathrm{I}(\boldsymbol{f}(\mathcal{A}); y_n \mid D_{n-1}) = \underset{\boldsymbol{x}_n \in \mathcal{S}}{\arg\min}\ \mathrm{H}[\boldsymbol{f}(\mathcal{A}) \mid D_{n-1}, (\boldsymbol{x}_n, y_n)]$$

Generalization bound for ITL (informal). $\forall \boldsymbol{x}' \in \mathcal{A}$:

$$\sigma_n^2(\boldsymbol{x}') \leq \underbrace{\eta_{\mathcal{S}}^2(\boldsymbol{x}')}_{\text{irreducible}} + \underbrace{C \log n / \sqrt{n}}_{\text{reducible}}$$

(C is a constant)

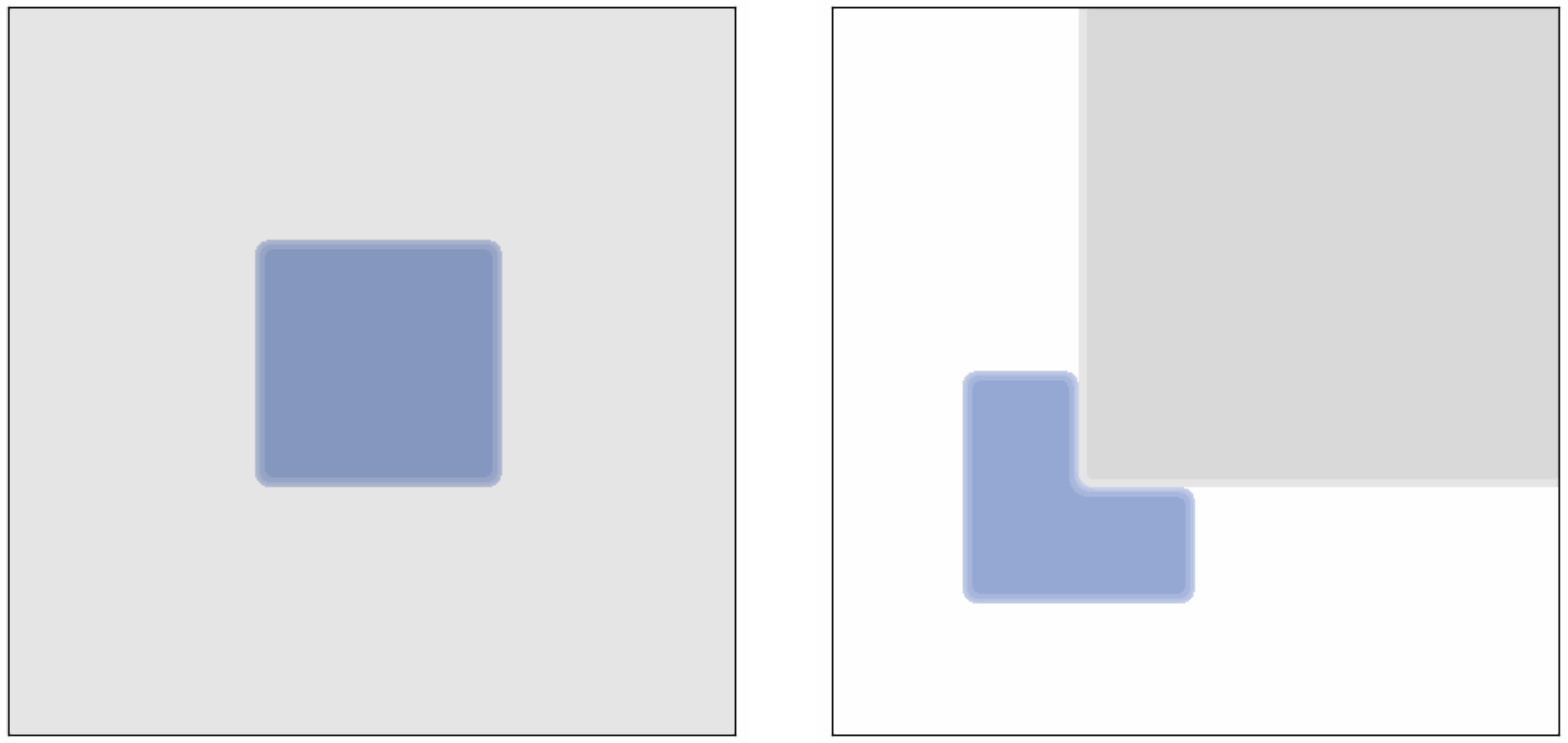


Illustration of ITL on a Gaussian process with Gaussian kernel

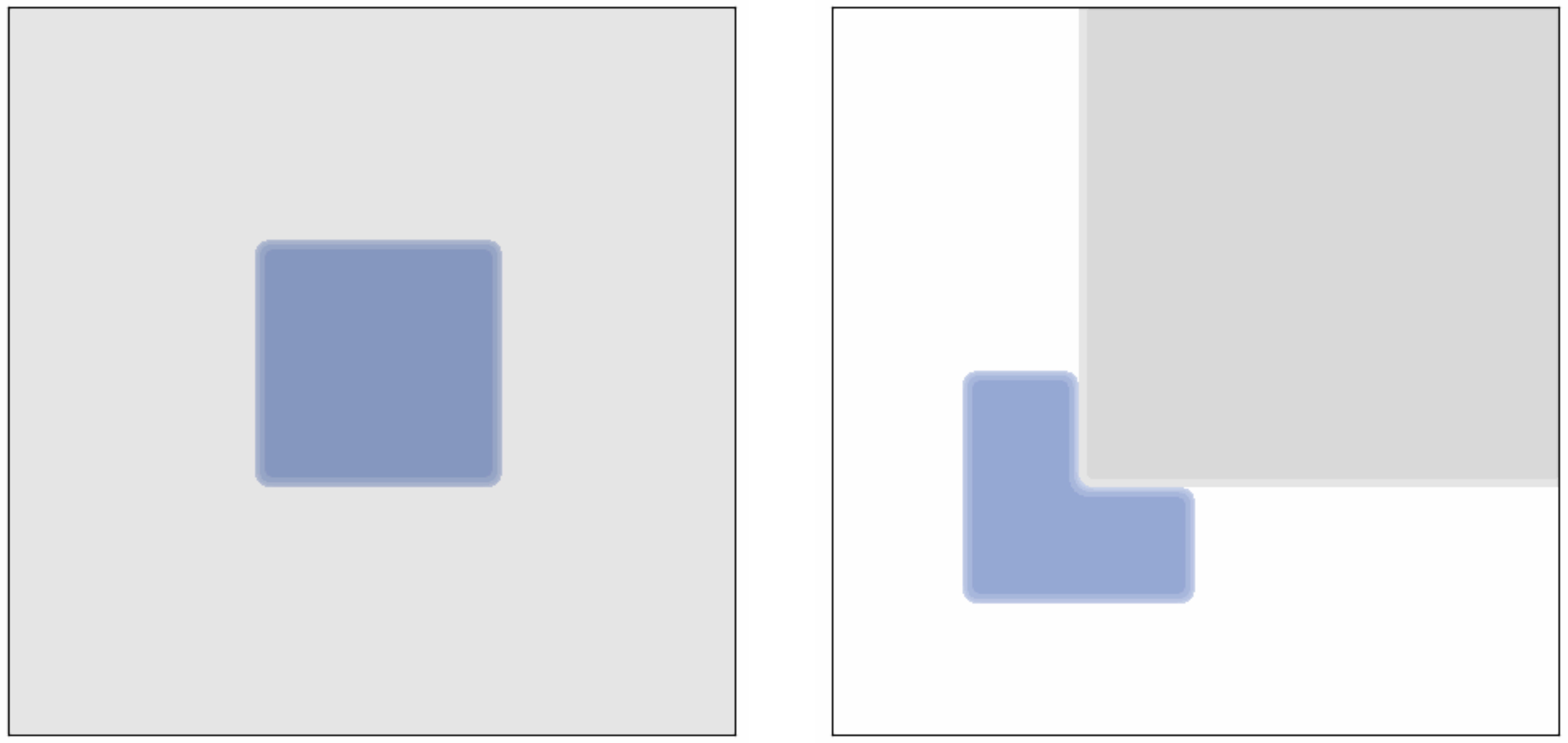


Illustration of ITL on a Gaussian process with Gaussian kernel

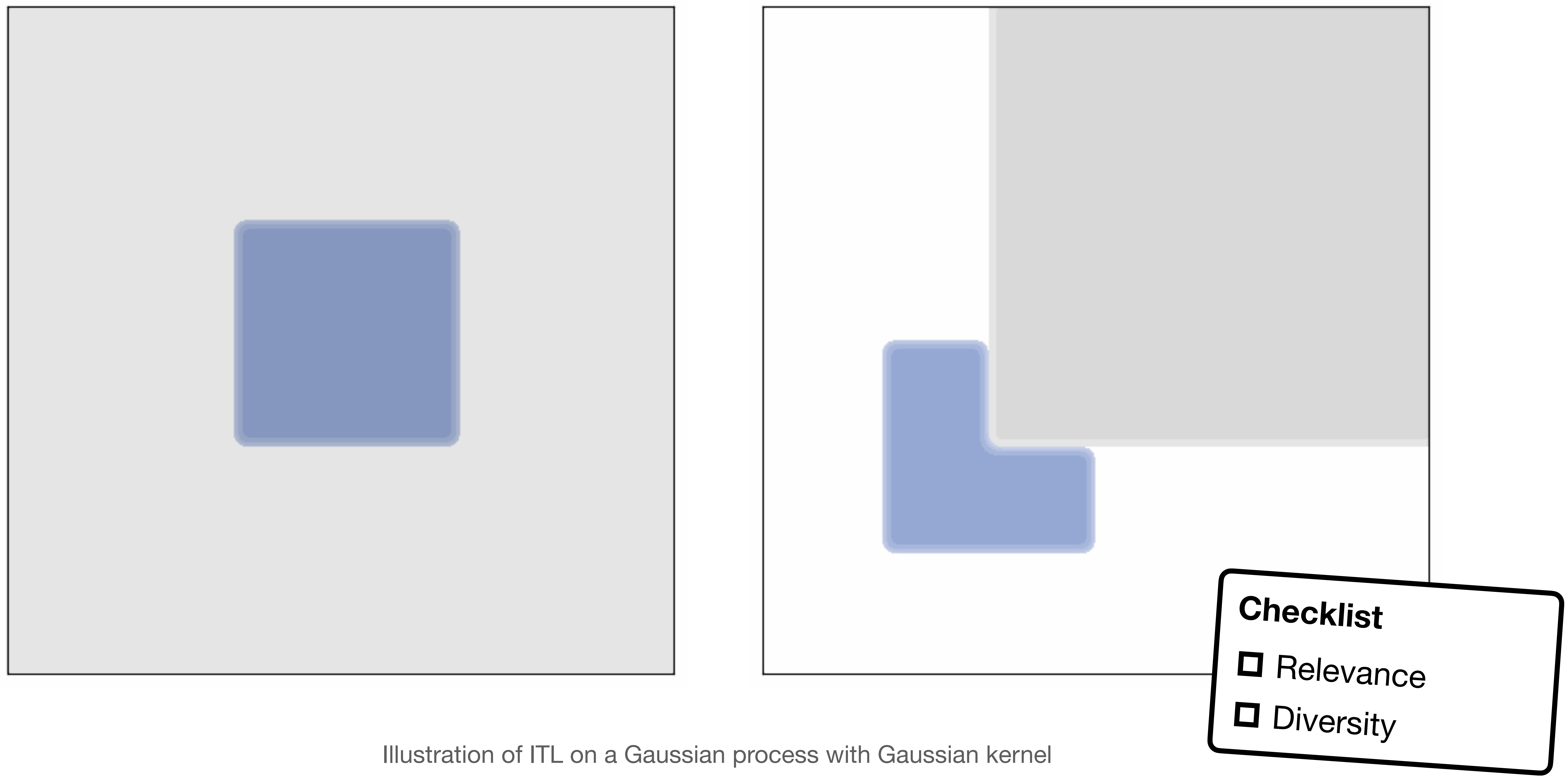


Illustration of ITL on a Gaussian process with Gaussian kernel

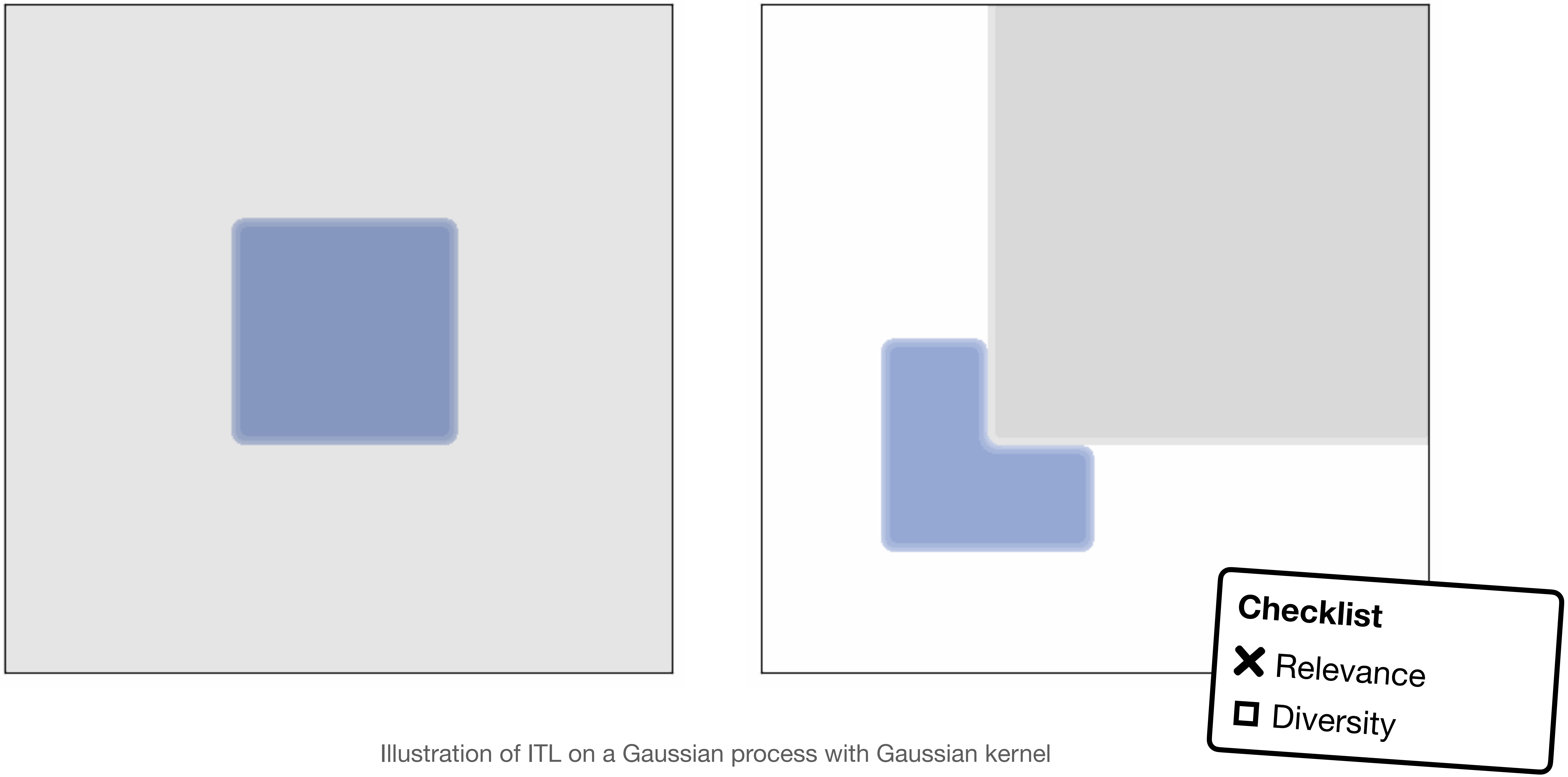


Illustration of ITL on a Gaussian process with Gaussian kernel

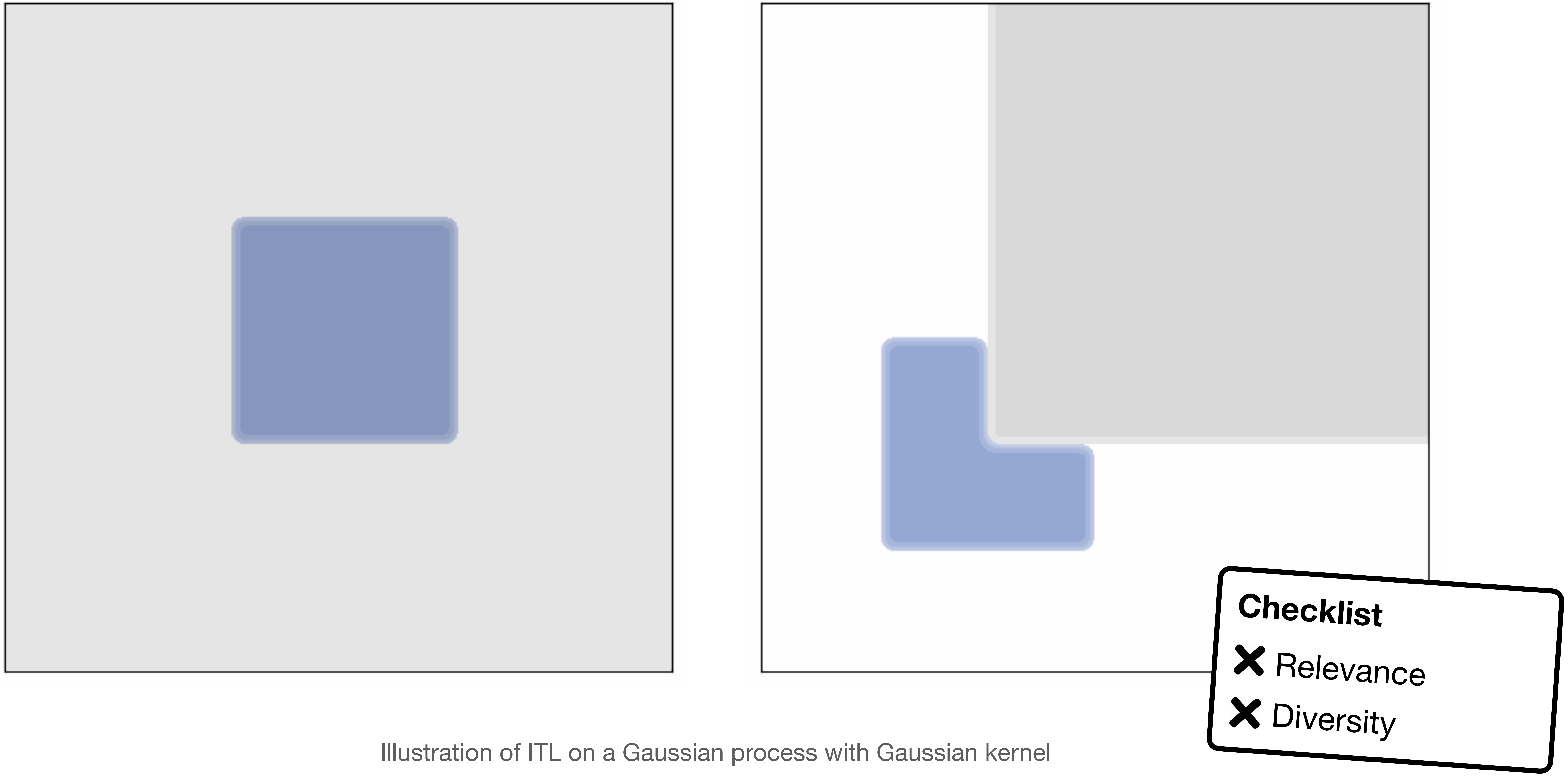
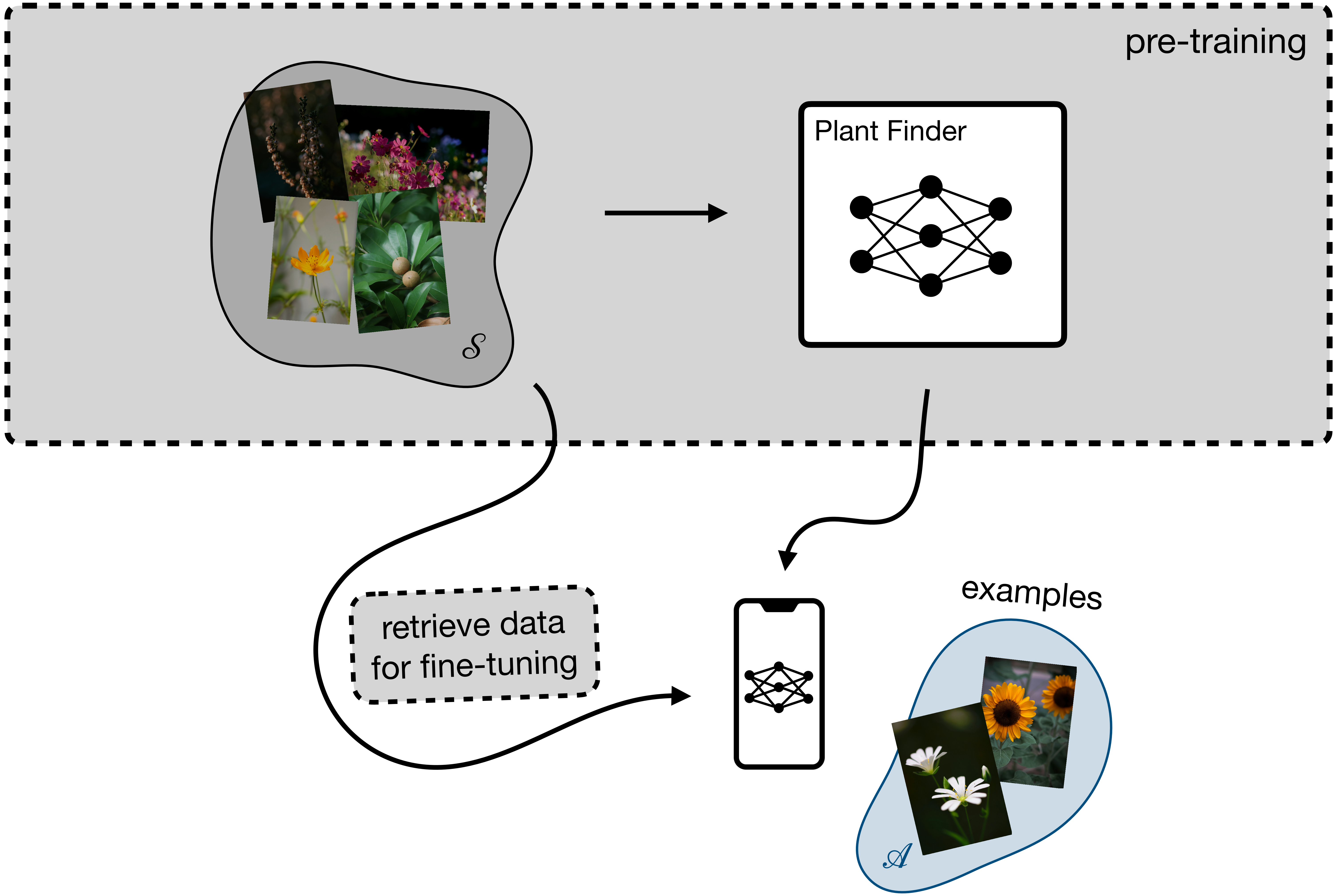
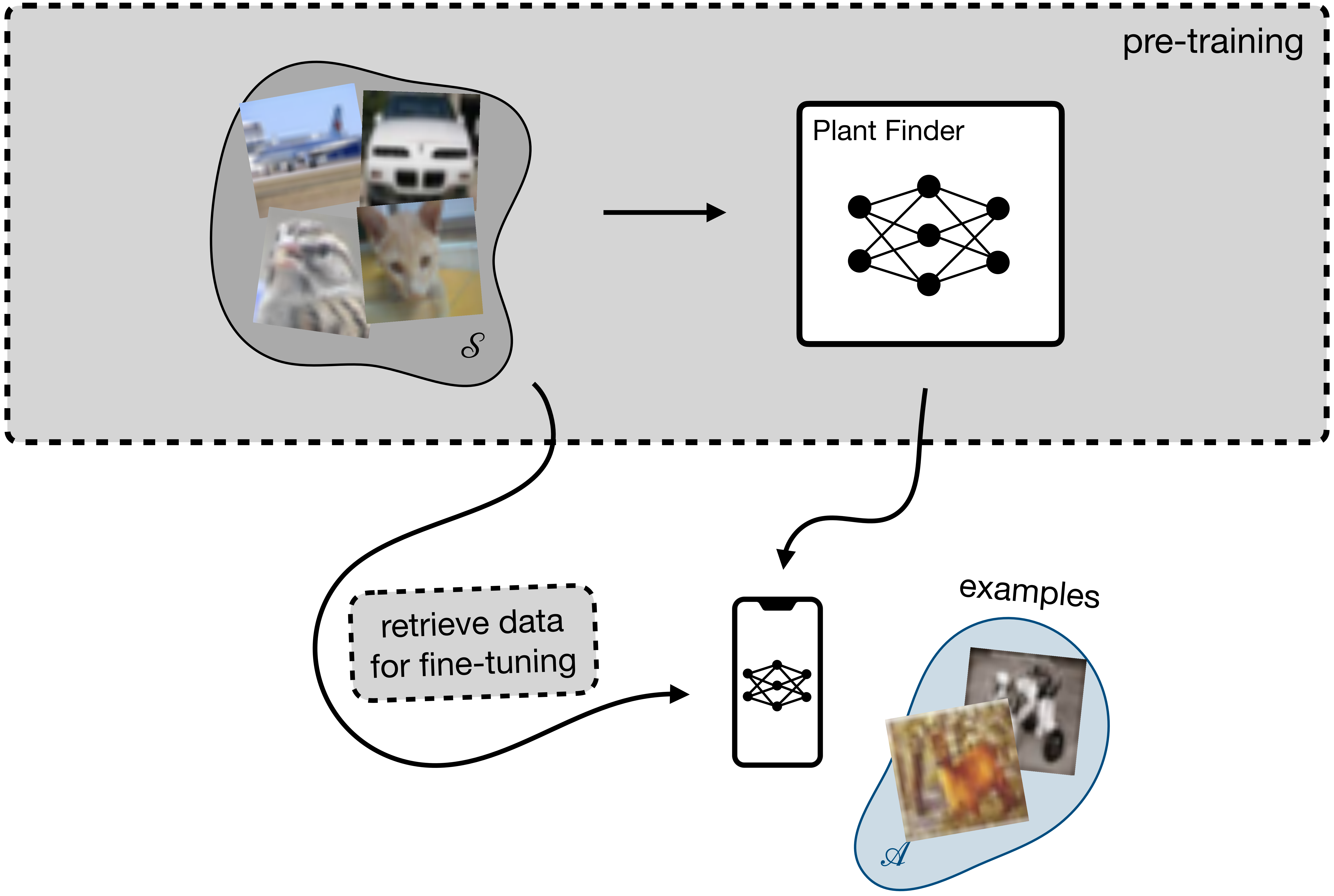


Illustration of ITL on a Gaussian process with Gaussian kernel

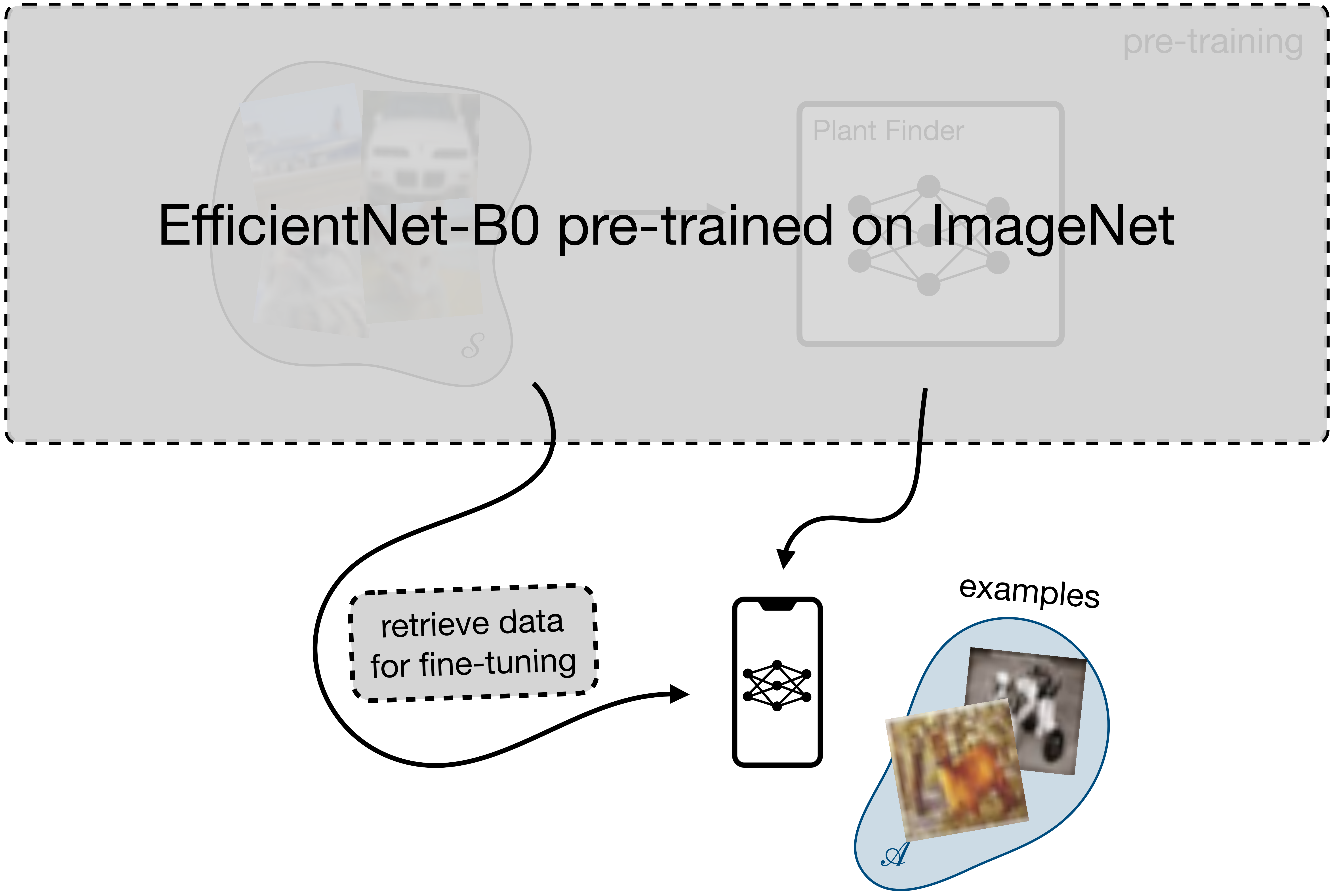
Fine-Tuning


pre-training
Plant Finder
$\mathcal{S}$
retrieve data
for fine-tuning
examples
$\mathcal{A}$

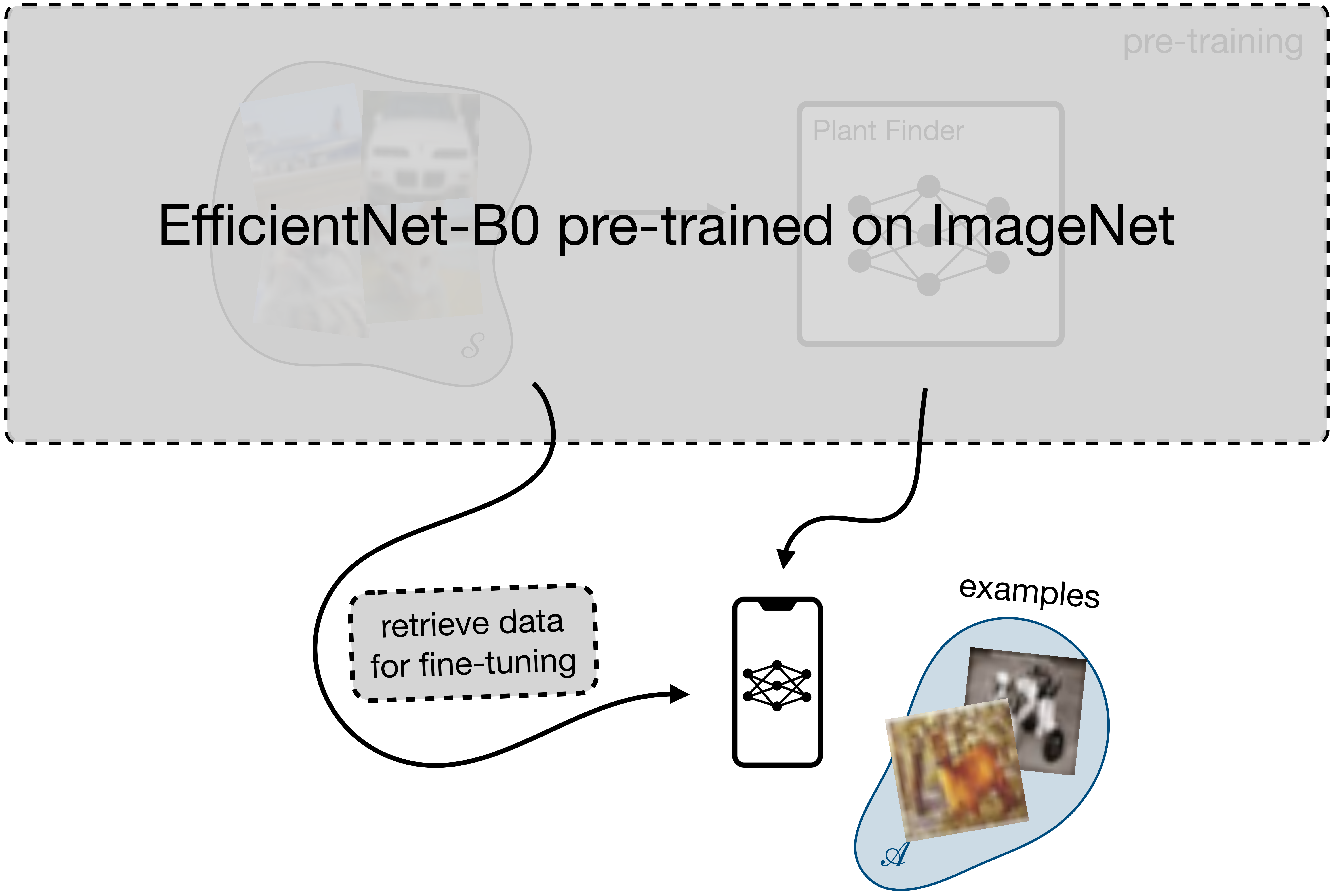
Fine-Tuning on CIFAR-100



Fine-Tuning on CIFAR-100

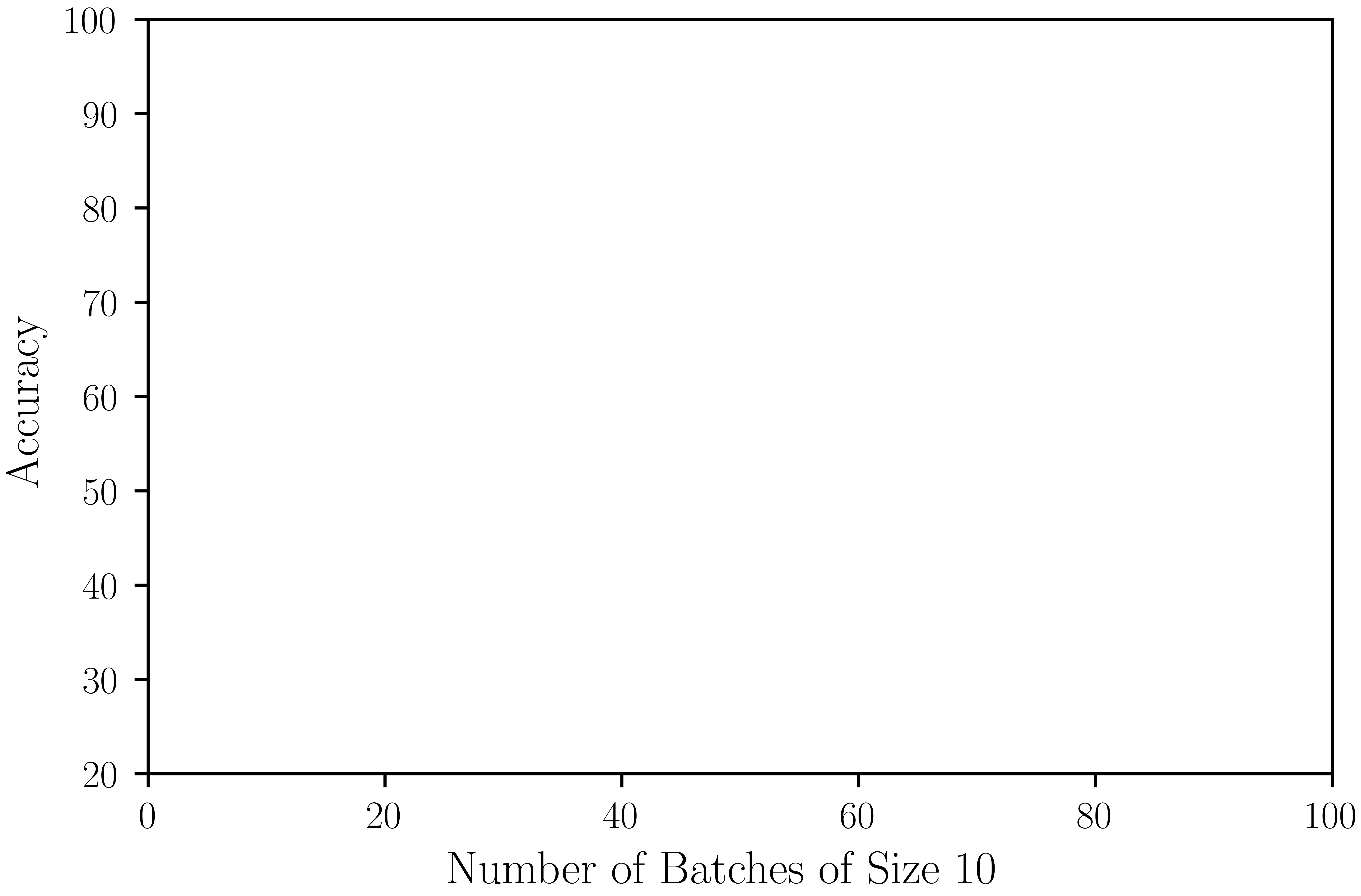


Fine-Tuning on CIFAR-100

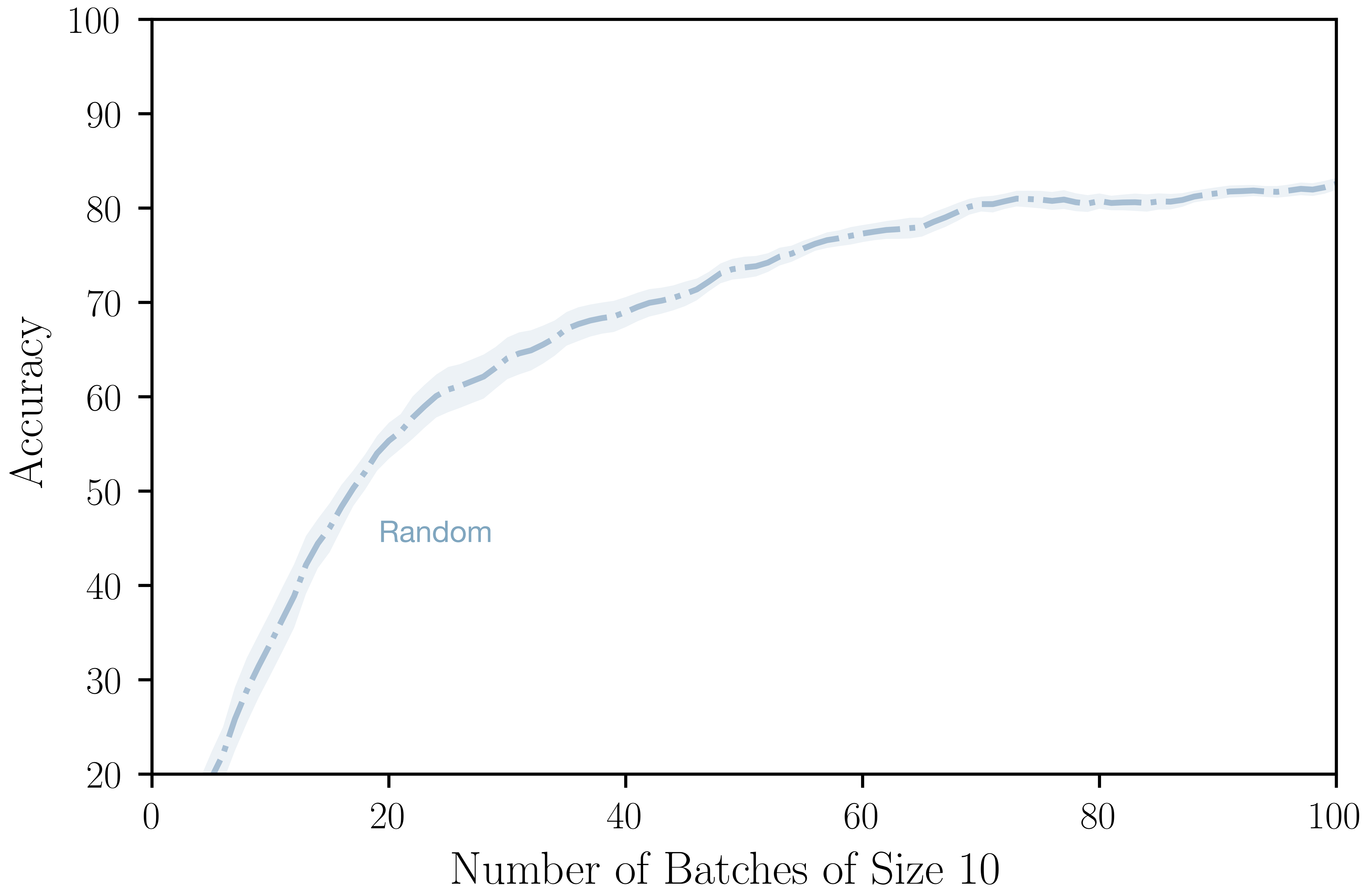


Goal: high accuracy on fresh examples from $\mathscr{A}$

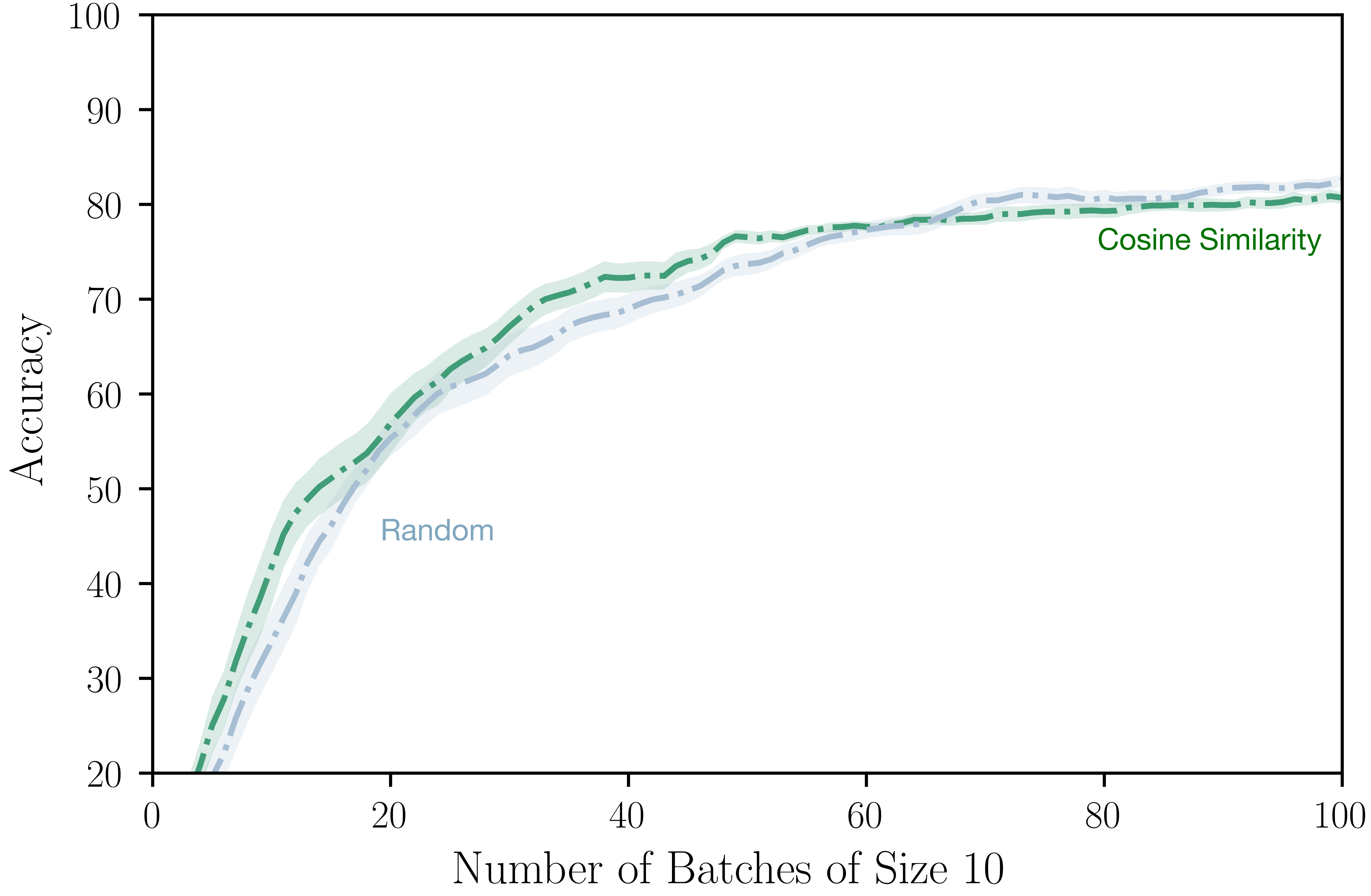
Fine-Tuning on CIFAR-100



Fine-Tuning on CIFAR-100



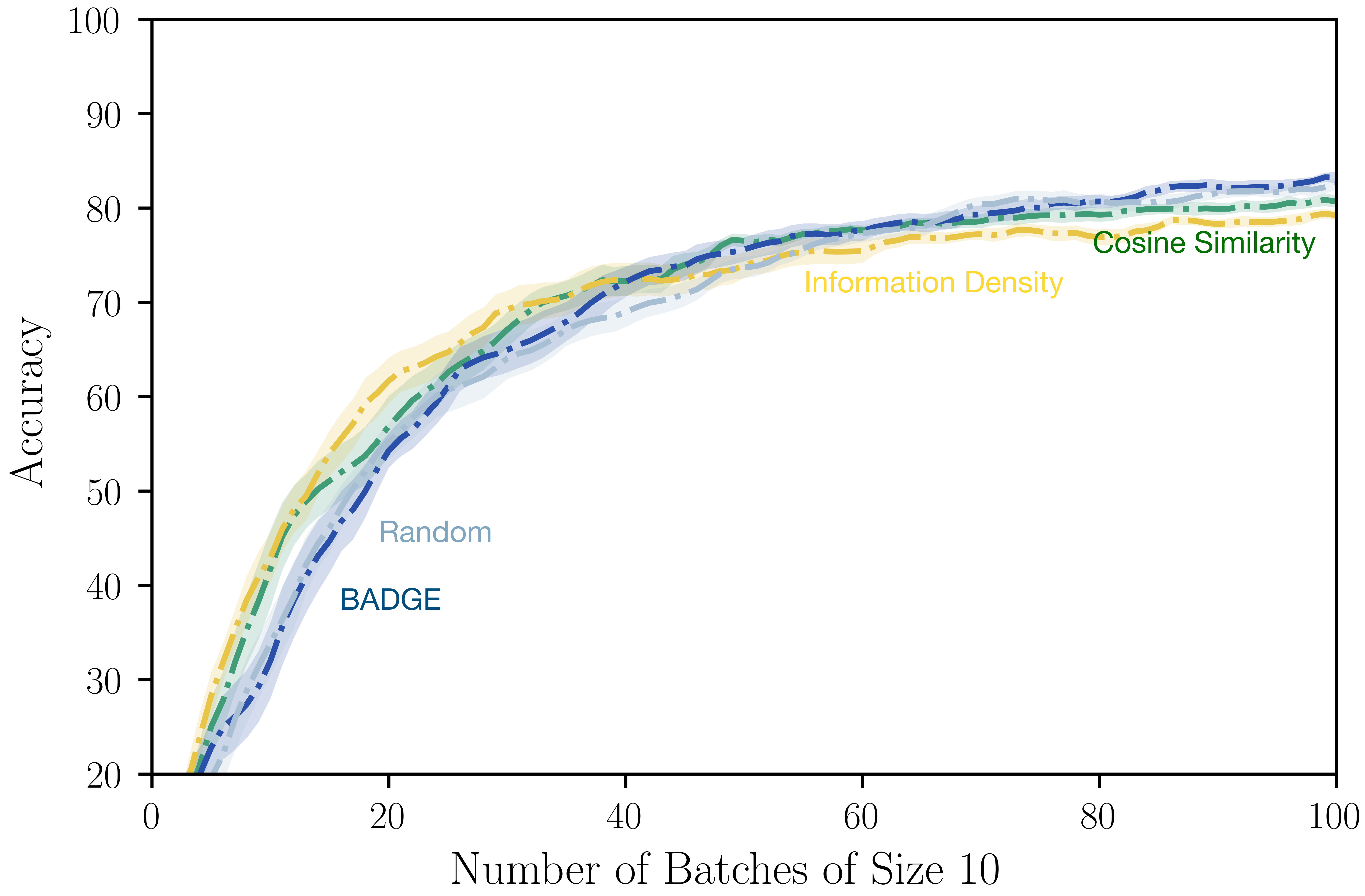
Fine-Tuning on CIFAR-100



Cosine Similarity: only relevance

$$\underset{x \in \mathcal{S}}{\arg\max} \frac{1}{|\mathcal{A}|} \sum_{x' \in \mathcal{A}} \underbrace{\measuredangle(\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}'))}_{\mathrm{Cor}[f(\boldsymbol{x}), f(\boldsymbol{x}') | D_{n-1}]}$$

Fine-Tuning on CIFAR-100



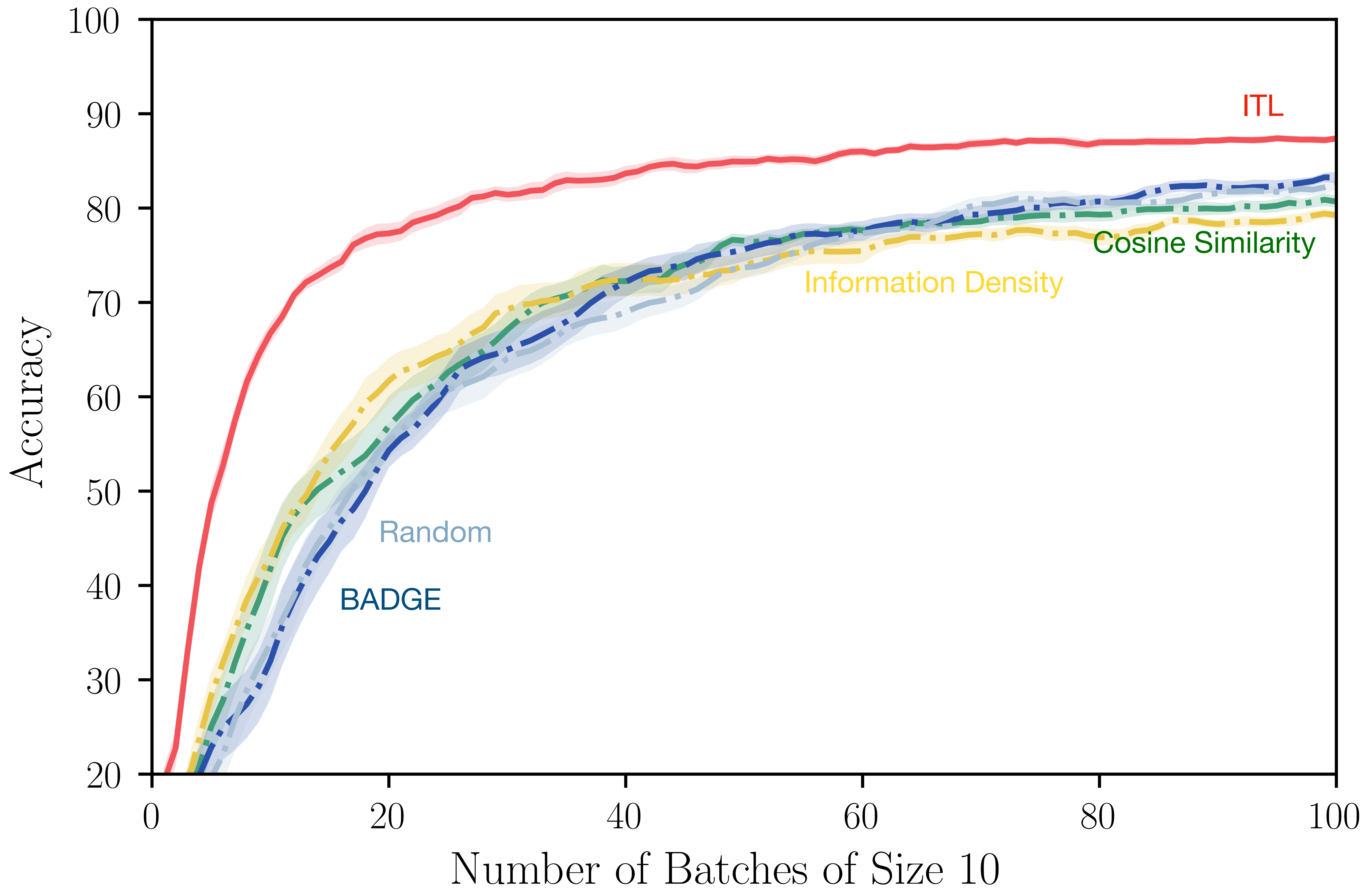
Cosine Similarity: only relevance

$$\arg\max_{x \in \mathcal{S}} \frac{1}{|\mathcal{A}|} \sum_{x' \in \mathcal{A}} \underbrace{\measuredangle(\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}'))}_{\mathrm{Cor}[f(\boldsymbol{x}), f(\boldsymbol{x}') | D_{n-1}]}$$

Information Density:
only relevance

BADGE:
only diversity

Fine-Tuning on CIFAR-100



Cosine Similarity: only relevance

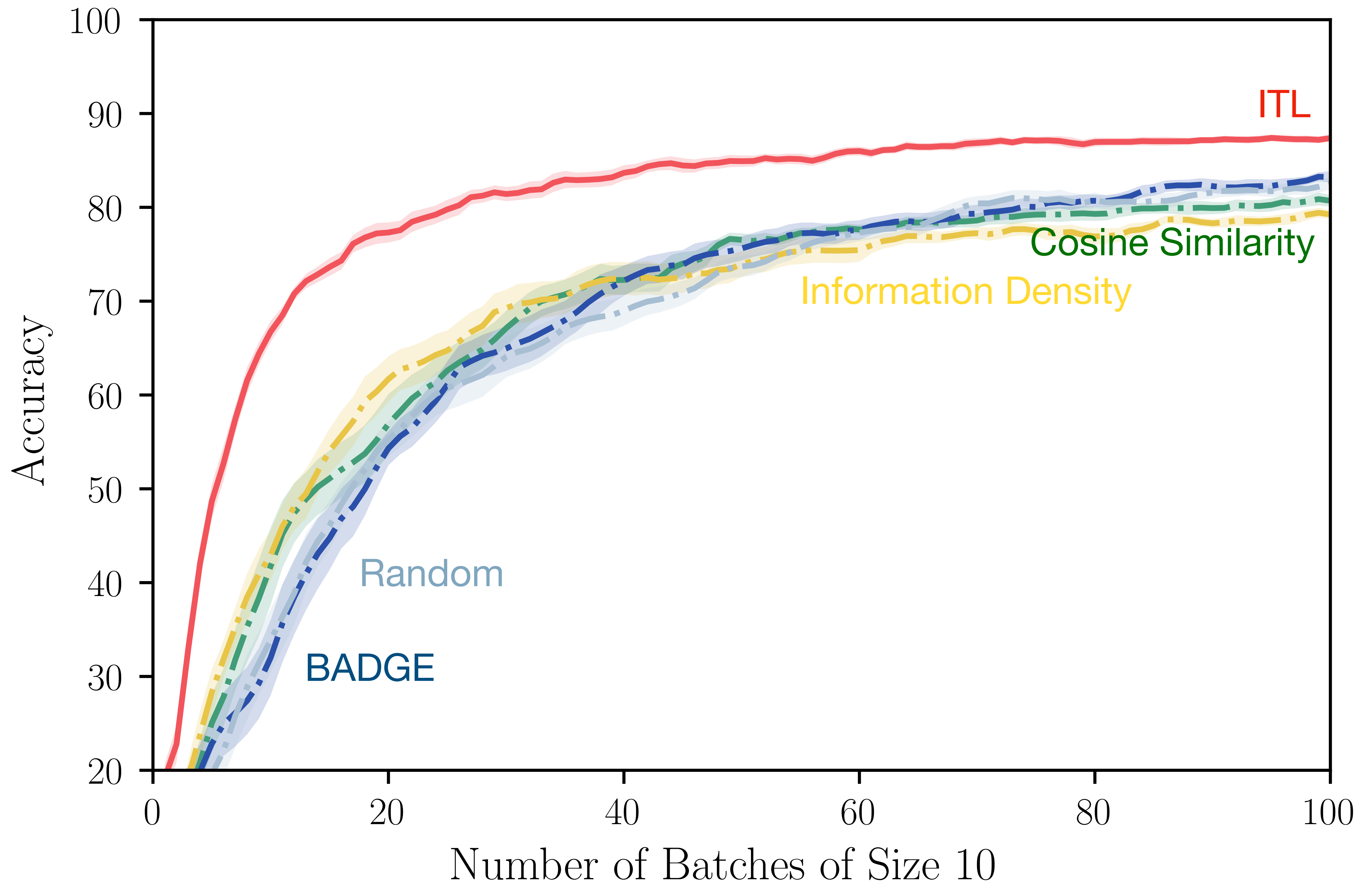
$$\arg\max_{x \in \mathcal{S}} \frac{1}{|\mathcal{A}|} \sum_{x' \in \mathcal{A}} \underbrace{\measuredangle(\boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{\phi}(\boldsymbol{x}'))}_{\mathrm{Cor}[f(\boldsymbol{x}), f(\boldsymbol{x}') | D_{n-1}]}$$

Information Density:
only relevance

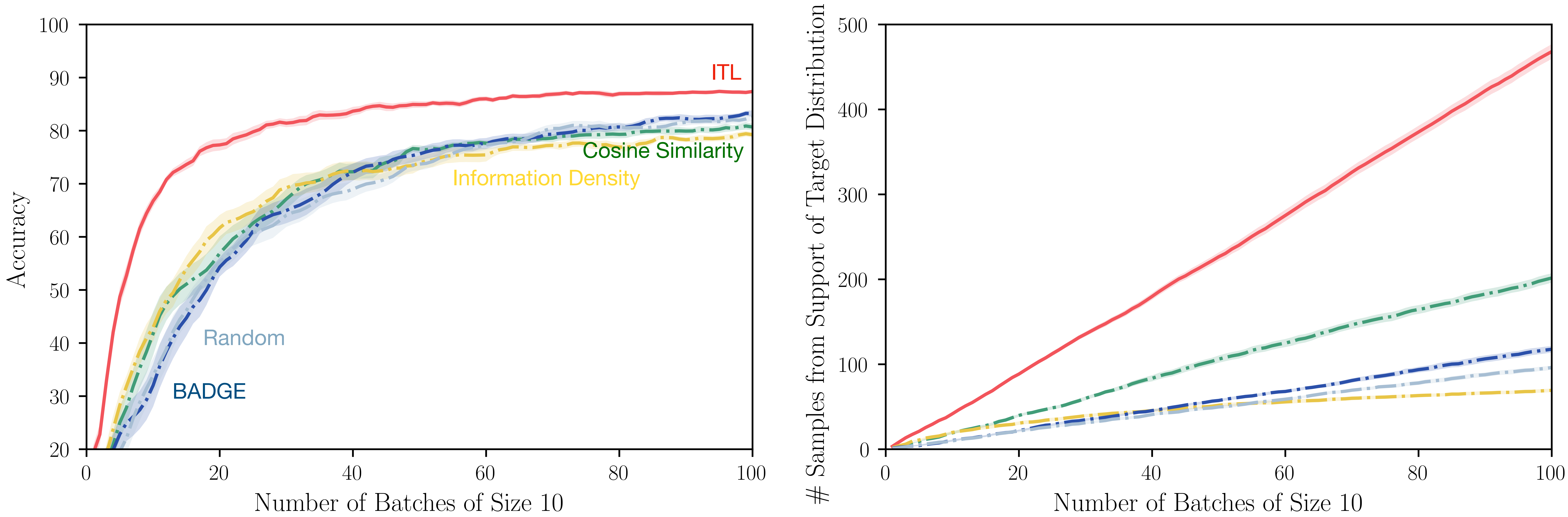
BADGE:
only diversity

ITL:
relevance + diversity

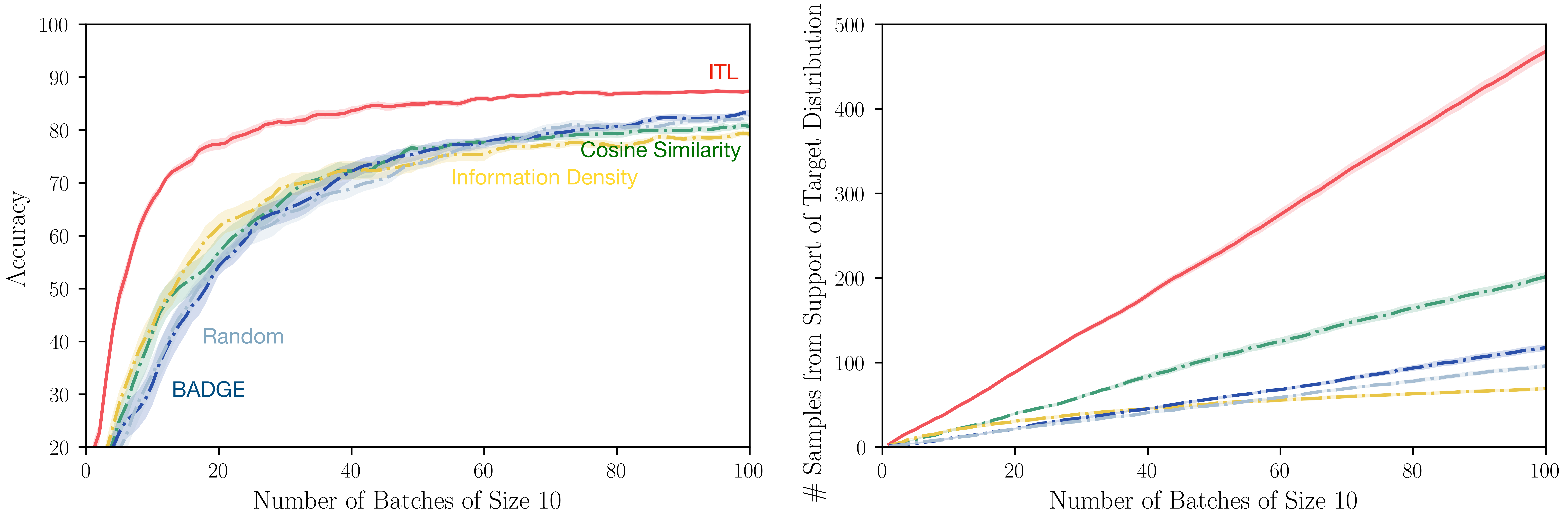
Fine-Tuning on CIFAR-100



Fine-Tuning on CIFAR-100



Fine-Tuning on CIFAR-100



ITL generalizes Cosine Similarity to query & batch sizes larger than 1!

Outlook

```python
from afsl import ActiveDataLoader

train_loader = ActiveDataLoader.initialize(dataset, target, batch_size=32)

while not converged:
    batch = dataset[train_loader.next(model)]
    model.step(batch)
```

Outlook

```
from afsl import ActiveDataLoader

train_loader = ActiveDataLoader.initialize(dataset, target, batch_size=32)

while not converged:
    batch = dataset[train_loader.next(model)]
    model.step(batch)
```

- Fine-tuning in domains other than image classification on standard datasets
- Connection between learning and retrieval (in-context learning)
- Analyzing submodularity of retrieval / ITL
- Other applications of Transductive Active Learning (Safe BO, …happy to chat!)

Bibliography

1. MacKay, D. J. Information-based objective functions for active data selection. Neural computation, 4(4), 1992.

2. Settles, B. and Craven, M. An analysis of active learning strategies for sequence labeling tasks. In EMNLP, 2008.

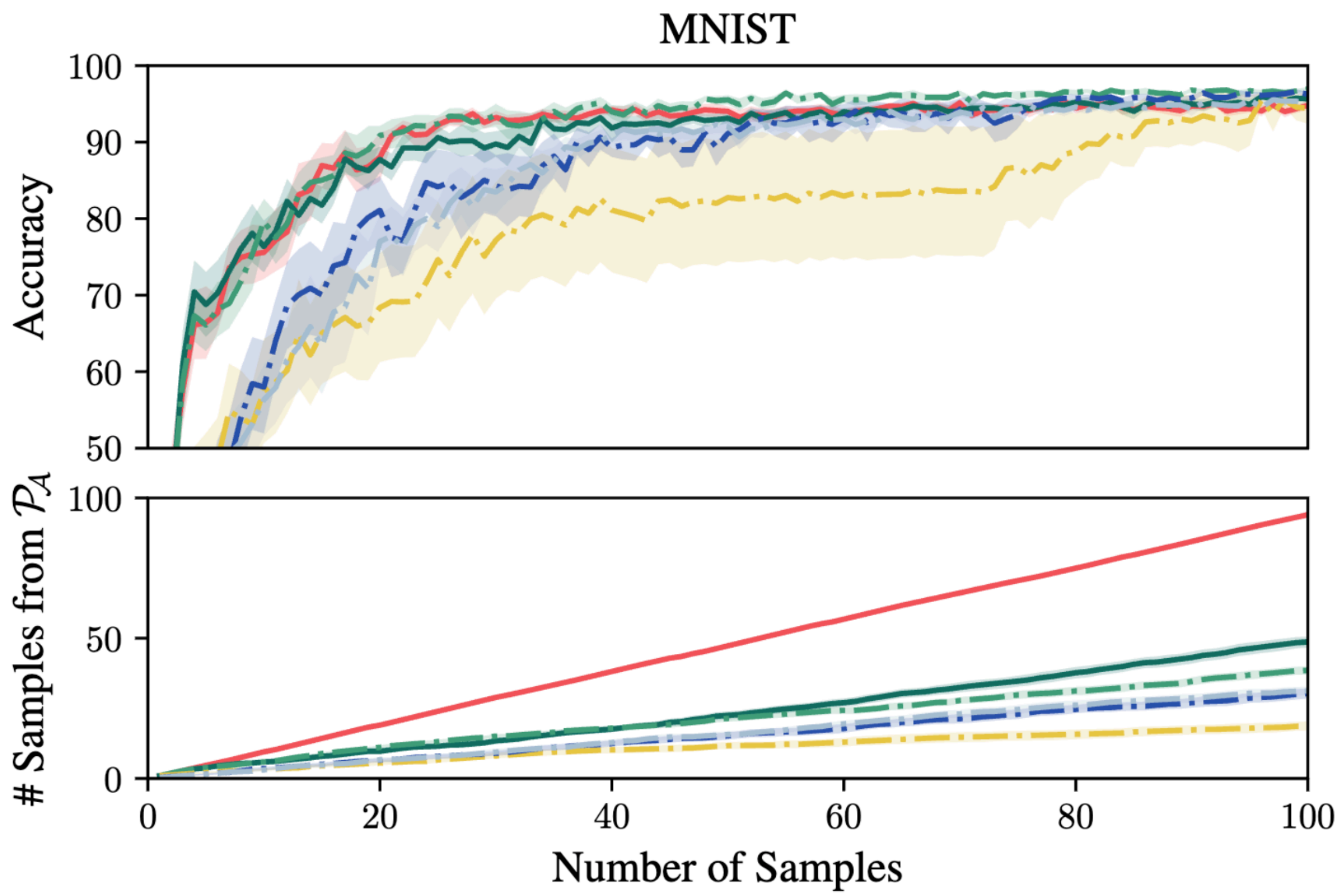
3. Ash, J. T., Zhang, C., Krishnamurthy, A., Langford, J., and Agarwal, A. Deep batch active learning by diverse, uncertain gradient lower bounds. ICLR, 2020.

Appendix

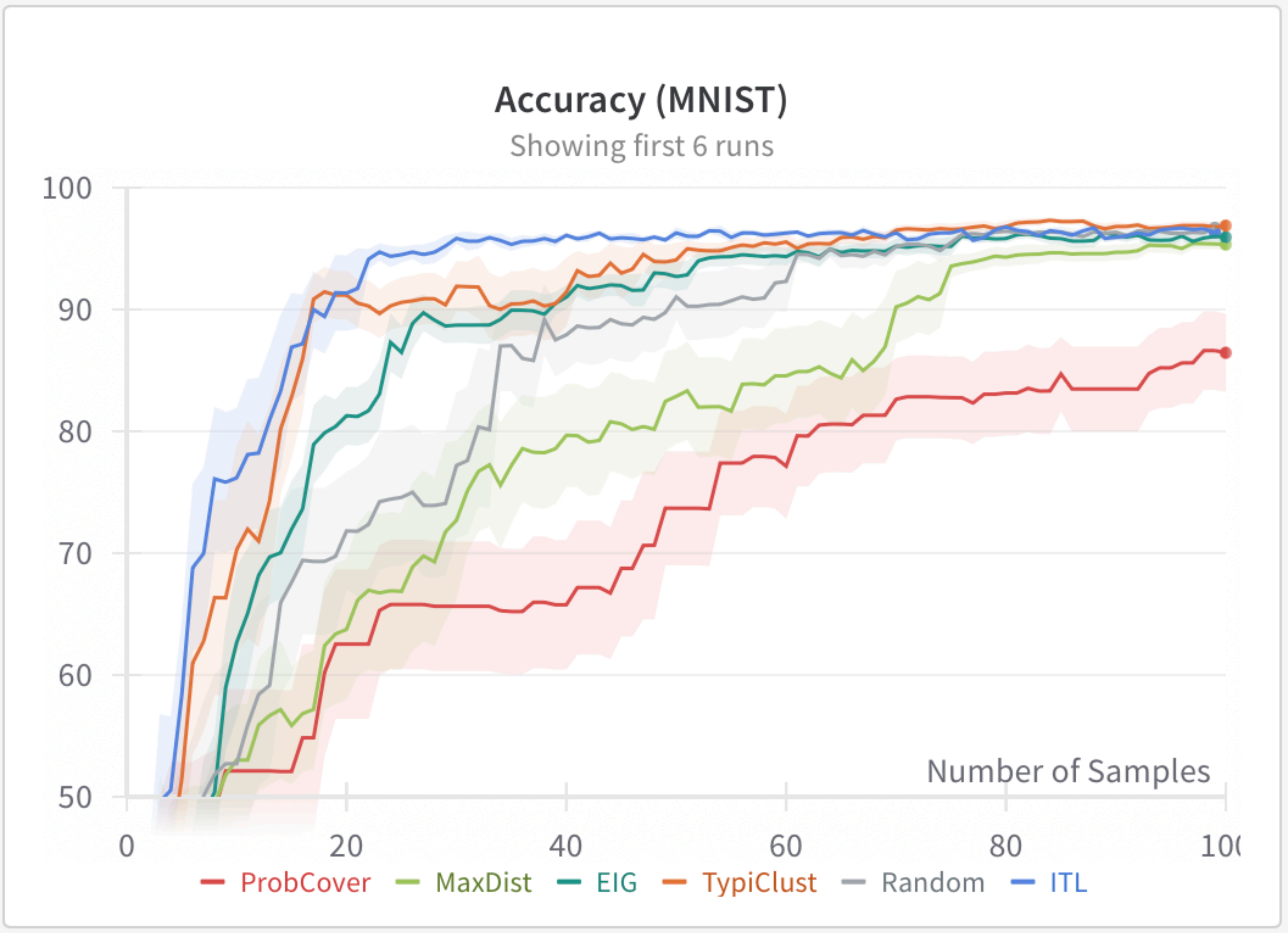
Embeddings

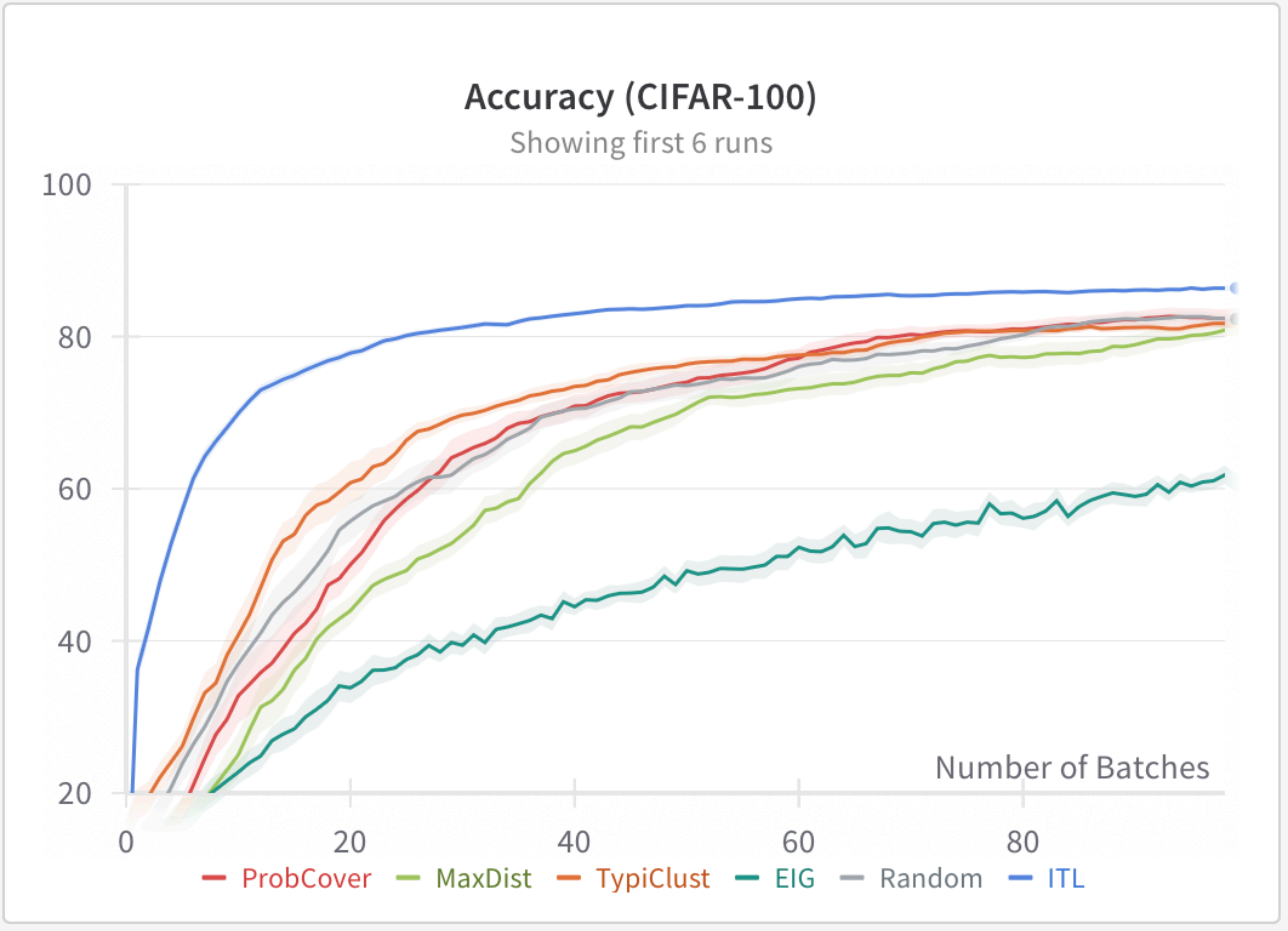
$$\boldsymbol{\phi}(\boldsymbol{x}) = \nabla_{\boldsymbol{\theta}} \ell\left(\boldsymbol{f}(\boldsymbol{x};\boldsymbol{\theta}), \hat{y}(\boldsymbol{x})\right)\Big|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}}$$

Without Pre-Training

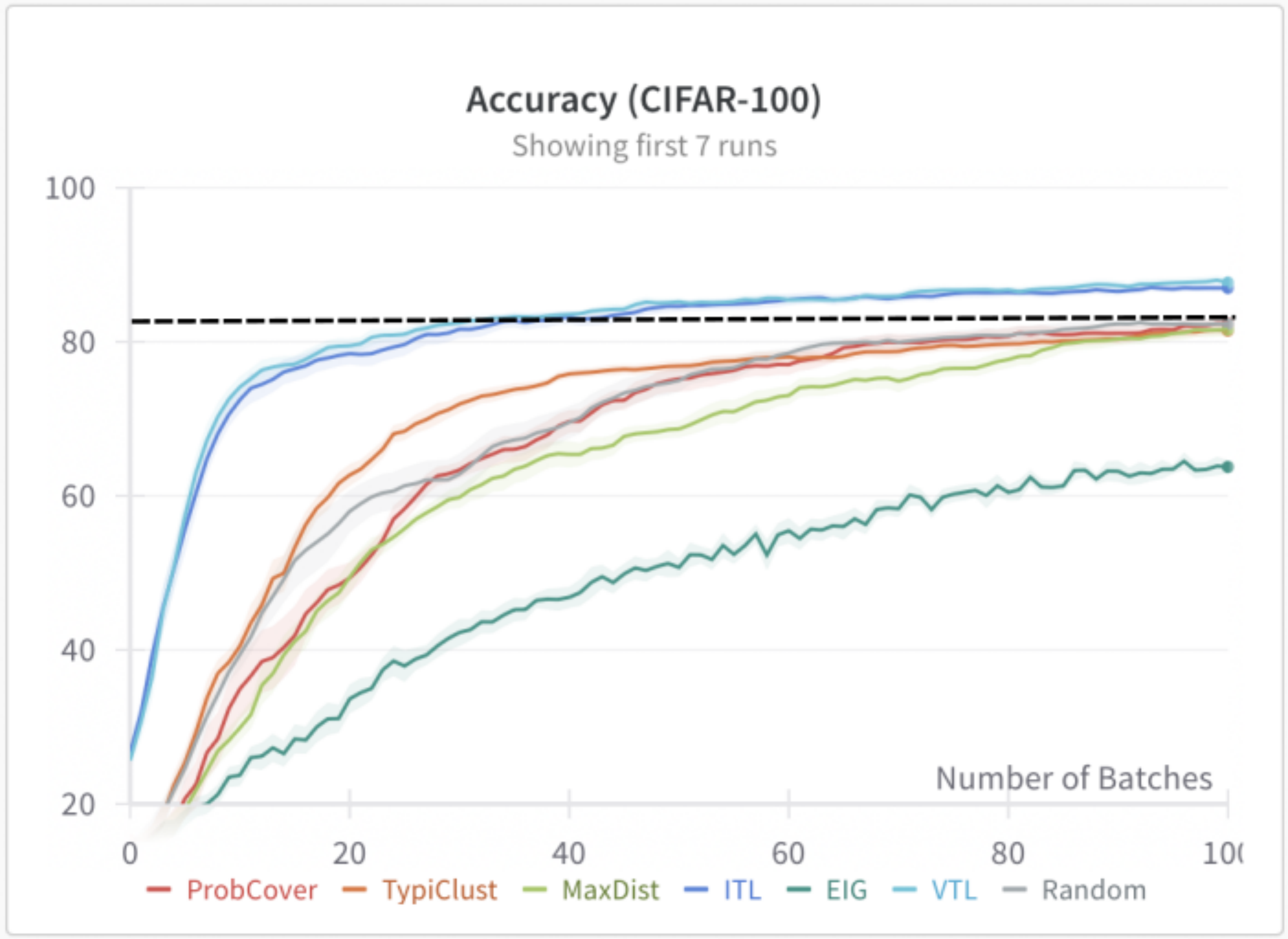


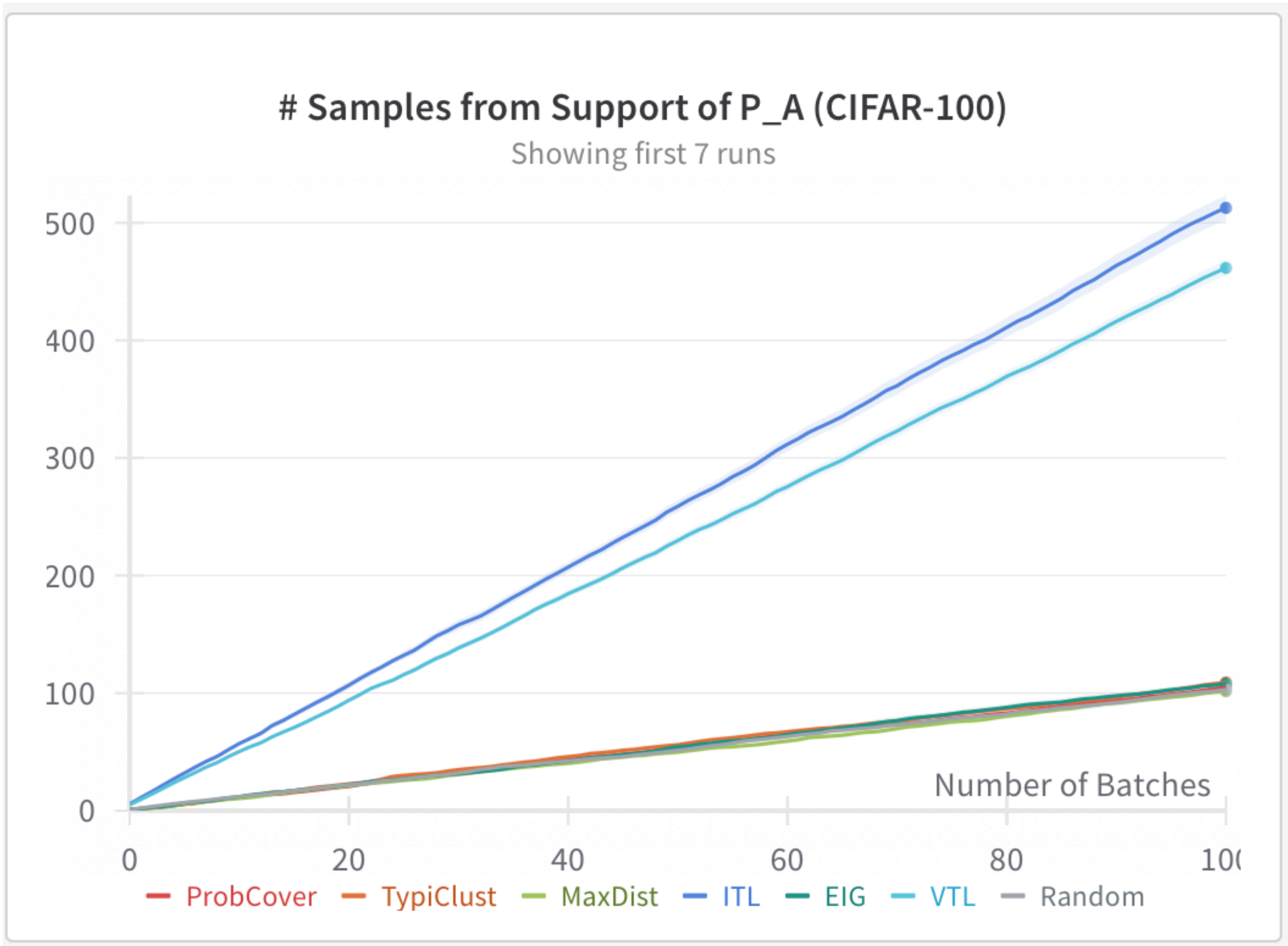
Additional Baselines





Allow Sampling from Target Set: $\mathscr{A} \subseteq \mathscr{S}$





Setting where the sample space is $\mathscr{S} \cup \mathscr{A}$, i.e., includes the target space. The dashed black line is the accuracy after training on $\mathscr{A}$ only where $|\mathscr{A}| = 100$.

Batch Selection via Conditional Embeddings

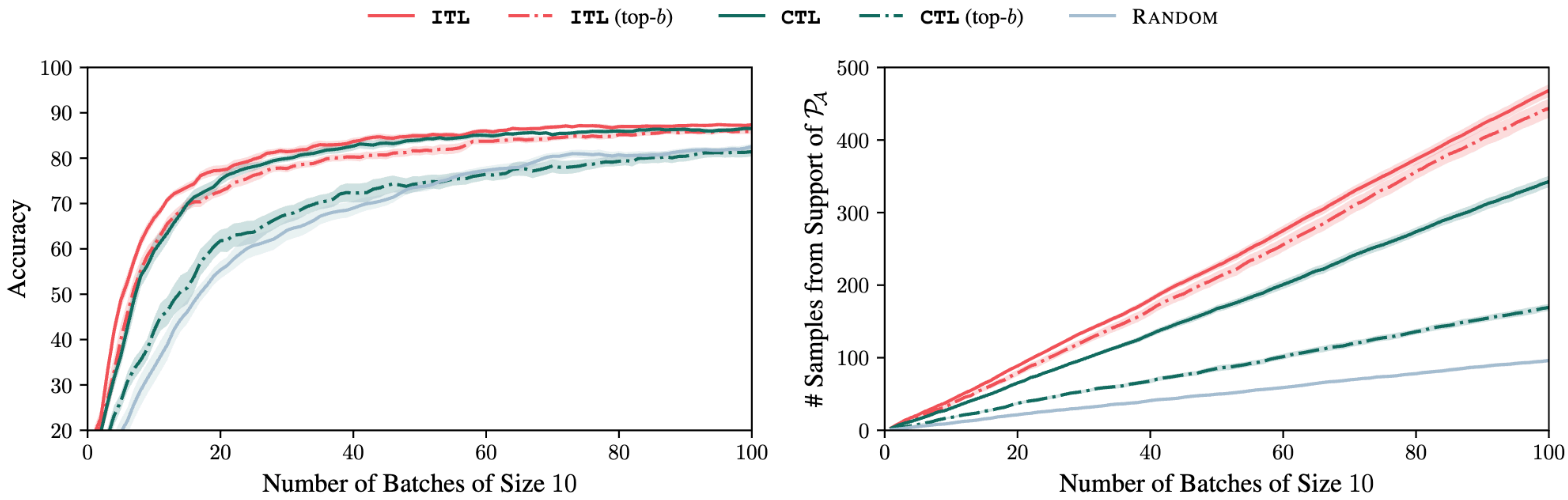


Figure 8. Advantage of batch selection via conditional embeddings over top-b selection in the CIFAR-100 experiment.