OOD-Chameleon

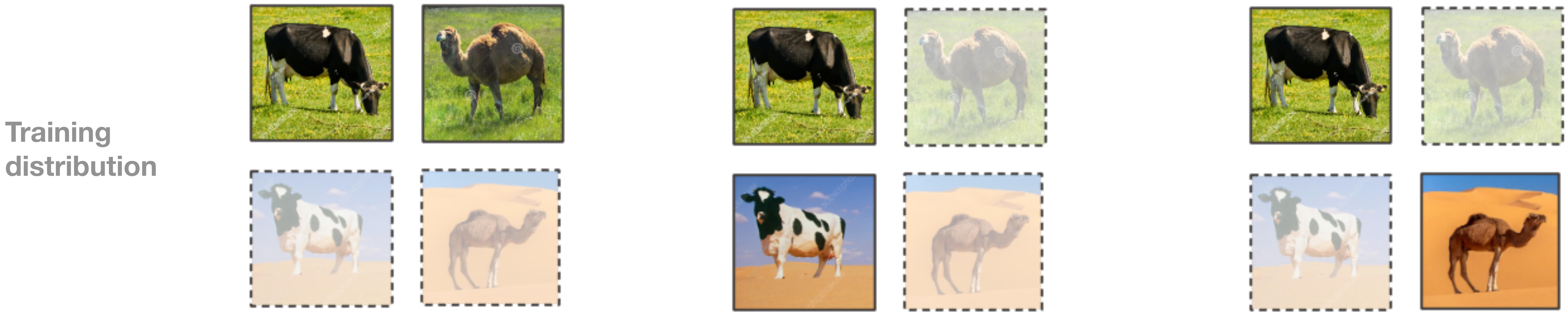
Is Algorithm Selection Learnable for OOD Generalization?

Liangze Jiang and Damien Teney

Out-of-distribution (OOD) generalization

Why is it hard?

❗ Different forms of distribution shifts *do not* permit a one-size-fits-all solution.



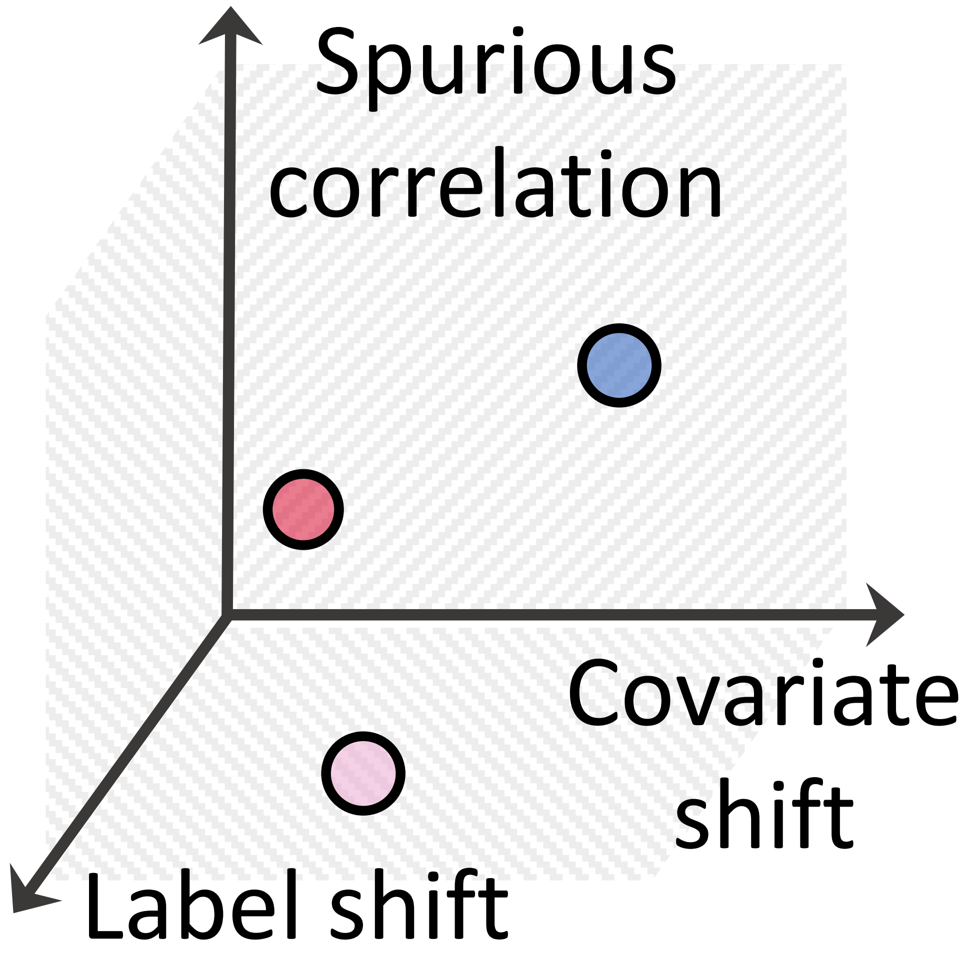
Covariate shift — $P(X)$

Label shift — $P(Y)$

Spurious correlation — $P(Y|X)$

X :Inputs Y :labels

Different algorithms excel under different conditions



Task 1	Task 2	Task 3	
ERM	GroupDRO	?	1st
ReSample	ERM	?	2nd
GroupDRO	Mixup	?	3rd
...	...	...	

Nguyen et al. Avoiding spurious correlations: Bridging theory and practice. NeurIPS 2021 Workshop on Distribution shifts

Wiles et al. A fine-grained analysis on distribution shift. ICLR 2022 (Oral)

Ye et al. OoD-Bench: Quantifying and Understanding Two Dimensions of Out-of-Distribution Generalization. CVPR 2022 (Oral)

Liang et al. MetaShift: A Dataset of Datasets for Evaluating Contextual Distribution Shifts and Training Conflicts. ICLR 2022

Yang et al. Change is hard: A closer look at subpopulation shift. ICML 2023

...

Can we learn to select the right algorithm?

A tantalizing possibility

🚨 Not clear how to select *manually* without extensive trial-and-error.

💡 We study the problem for a certain *distribution of distribution shifts*.

💭 **Key conjecture**: a learnable mapping from measurable *dataset characteristics* to the *performance* of algorithms exist.

A real-number vector

(1) Distribution shifts characteristics

Magnitudes of shifts

The strength of spurious feature

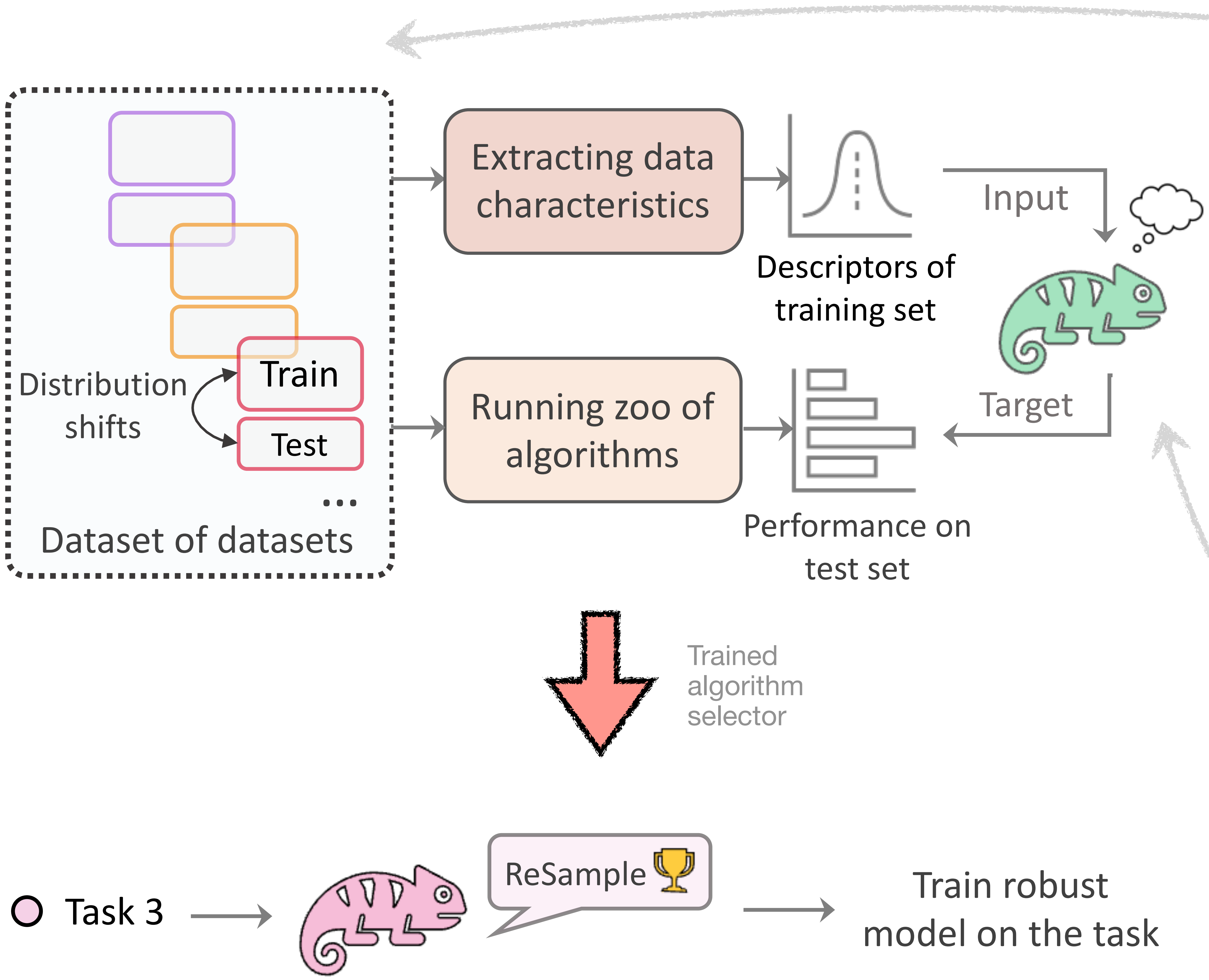
(2) Dataset complexity characteristics

Dataset size

Data dimensionality

OOD-Chameleon

Algorithm selection as a (supervised) learning task



How to obtain the **dataset**?

💡 We generate a collection of *semi-synthetic* datasets that represent diverse distribution shifts.

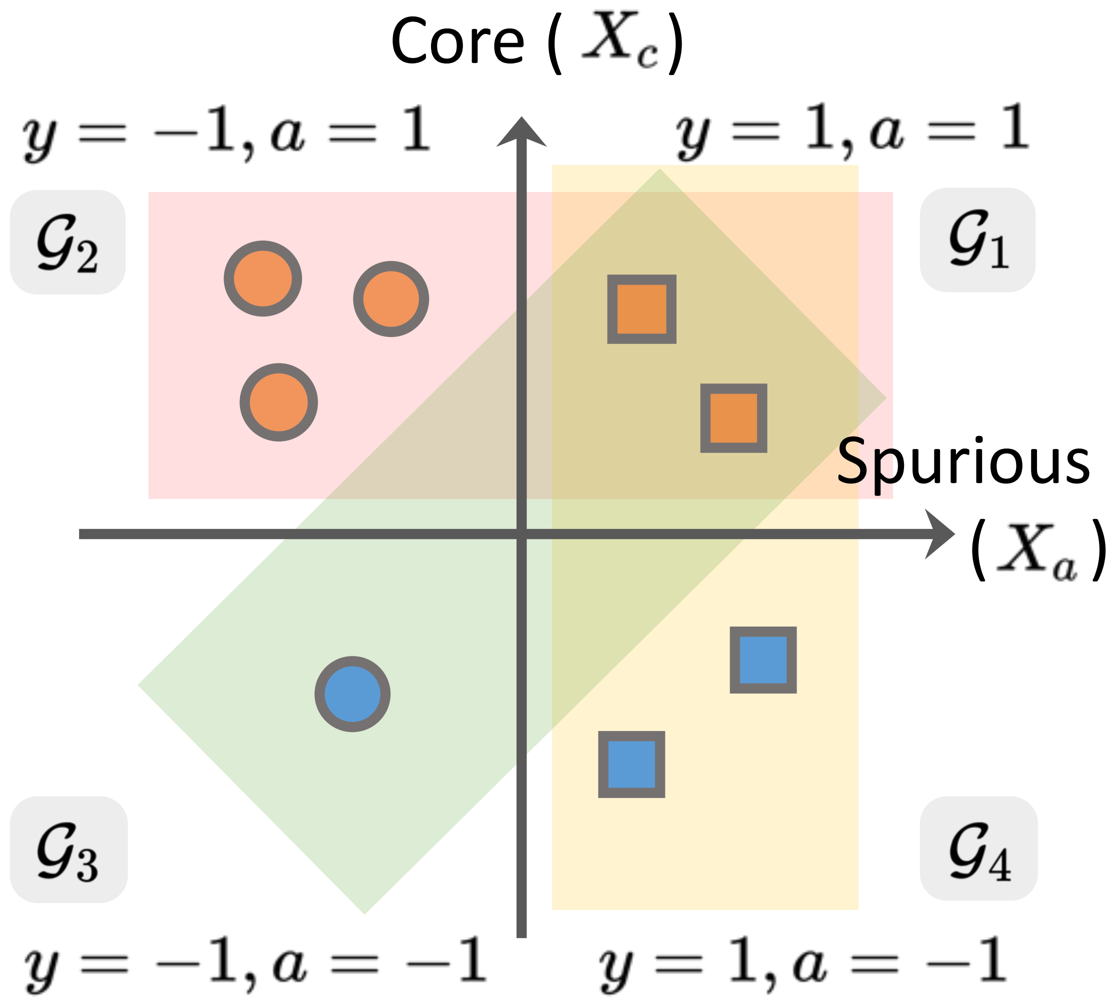
What is the **objective**?

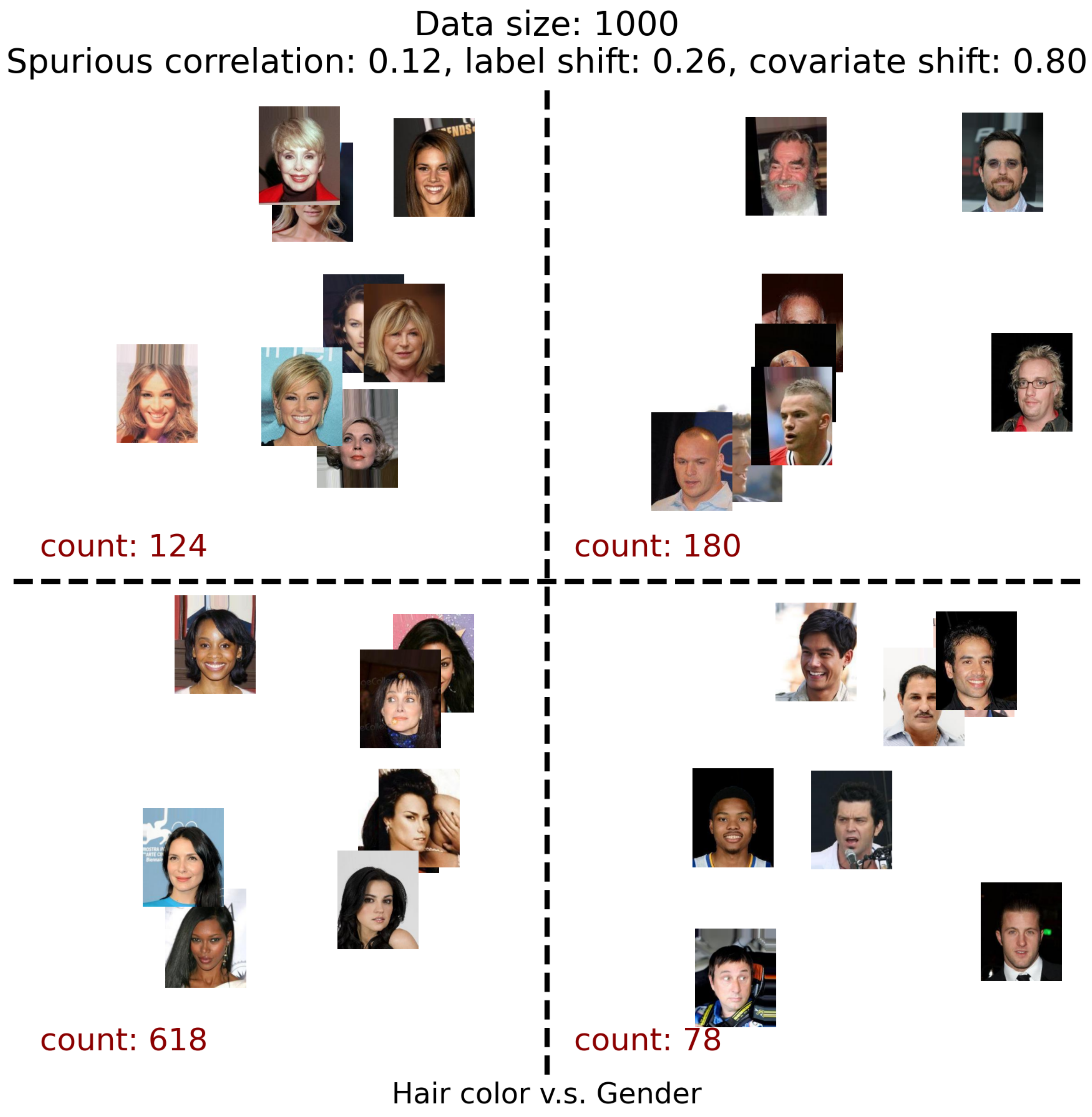
A tool to construct distribution shifts

⭐ Repurposing existing datasets (e.g. CelebA) with constrained subsampling.

Solving $|\mathcal{G}_i|$

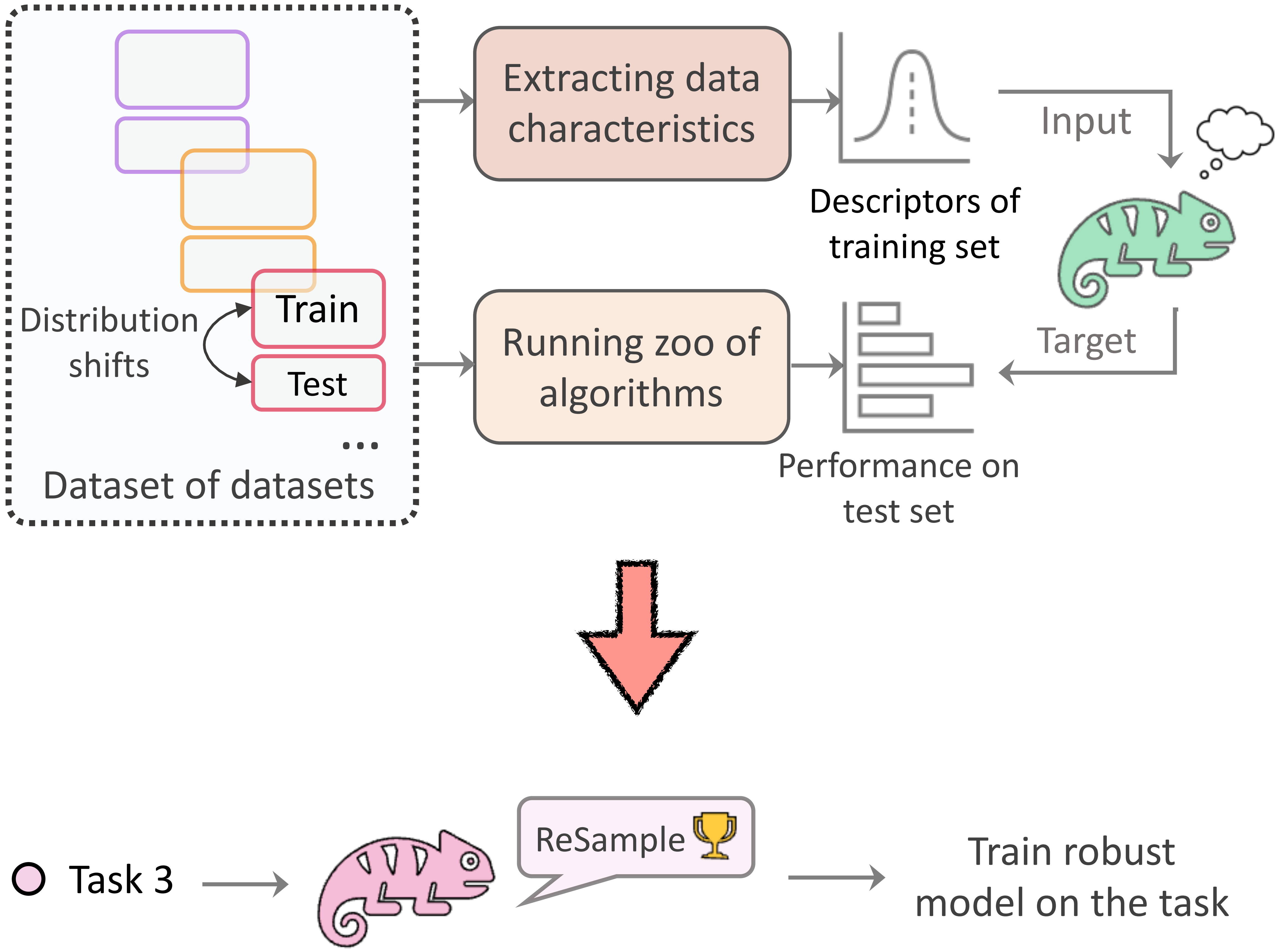
$$\left\{ \begin{aligned} d_{\text{sc}} &= \frac{|\mathcal{G}_1| + |\mathcal{G}_3|}{\sum_i |\mathcal{G}_i|} \\ d_{\text{ls}} &= \frac{|\mathcal{G}_1| + |\mathcal{G}_4|}{\sum_i |\mathcal{G}_i|} \\ d_{\text{cs}} &= \frac{|\mathcal{G}_1| + |\mathcal{G}_2|}{\sum_i |\mathcal{G}_i|} \end{aligned} \right.$$





OOD-Chameleon

Algorithm selection as a (supervised) learning task



How to obtain the **dataset**? ✅

💡 We generate a collection of *semi-synthetic* datasets that represent diverse distribution shifts.

What is the **objective**?

💡 We formulate the algorithm selection as a (binary) *multi-label classification.*

Algorithm selection as a supervised learning task

regression?

Multi-label classification!

Algorithm selector

Vectorized dataset descriptor

$$\min_{w} \mathbb{E}_{\mathbb{D}} \mathcal{L}_{\text{BCE}}\left(\phi\left(w, f(D_j^{\text{tr}})\right), Y_{\mathcal{A}}\right)$$

Dataset of datasets

Binary cross-entropy

Algorithms' suitability on dataset j (multi-hot binary vector)

Algorithm m's accuracy on dataset j

$$Y_{\mathcal{A}} = \begin{cases} 1 & P_{jm} - \min_m P_{jm} \leq \epsilon \\ 0 & \text{else} \end{cases}$$

💡 Learn the mapping from measurable dataset descriptor to the suitability (relative performance) of the algorithm.

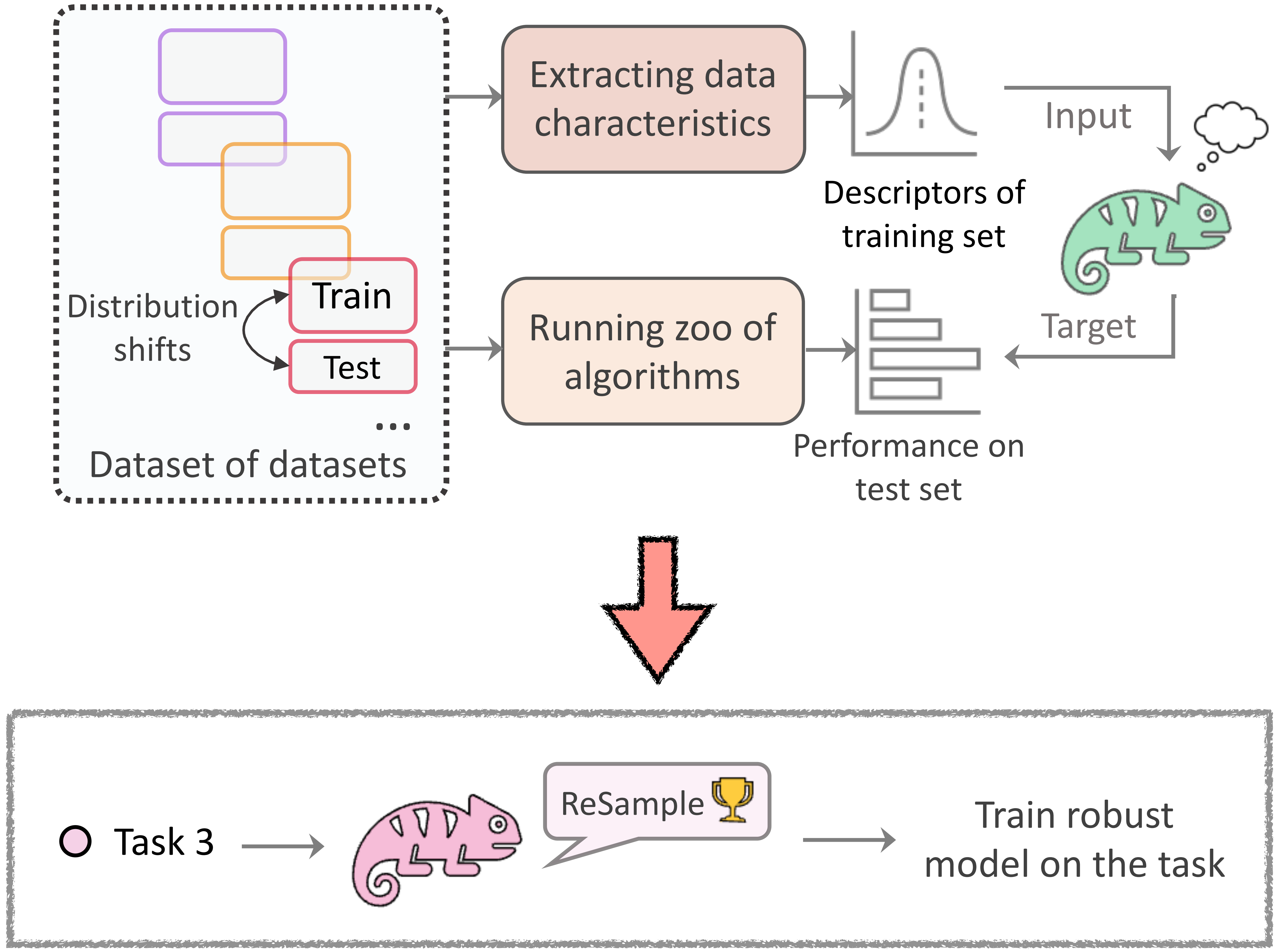
Classification is generally easier than regression*…

Different algorithms do not compete with each other, hence 'multi-label'…

* Devroye et al. A probabilistic theory of pattern recognition. 2013

OOD-Chameleon

Algorithm selection as a (supervised) learning task



How to obtain the dataset? ✅

💡 We generate a collection of *semi-synthetic* datasets that represent diverse distribution shifts.

What is the objective? ✅

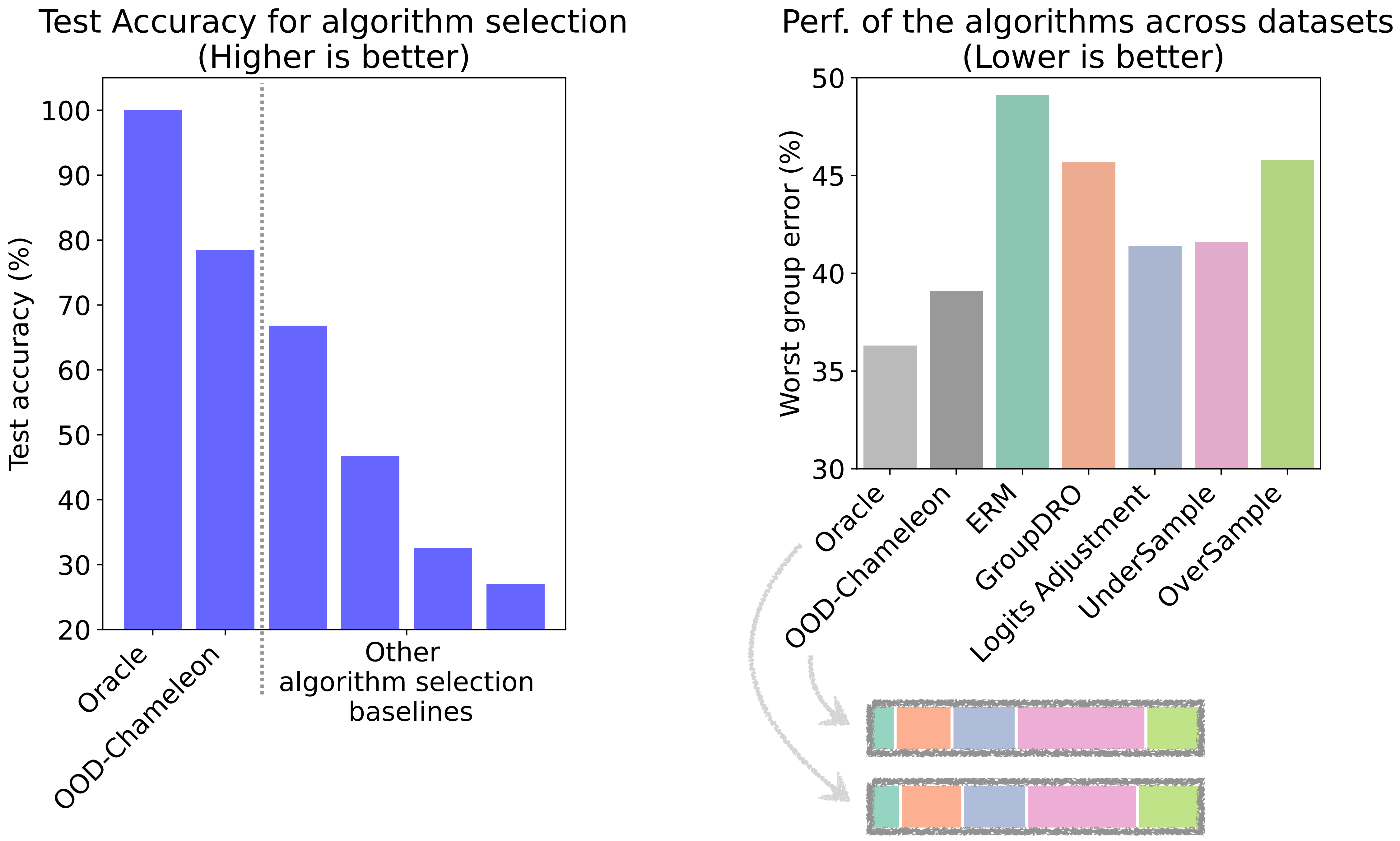
💡 We formulate the algorithm selection as a *multi-label classification.*

See our paper for many more results & details!

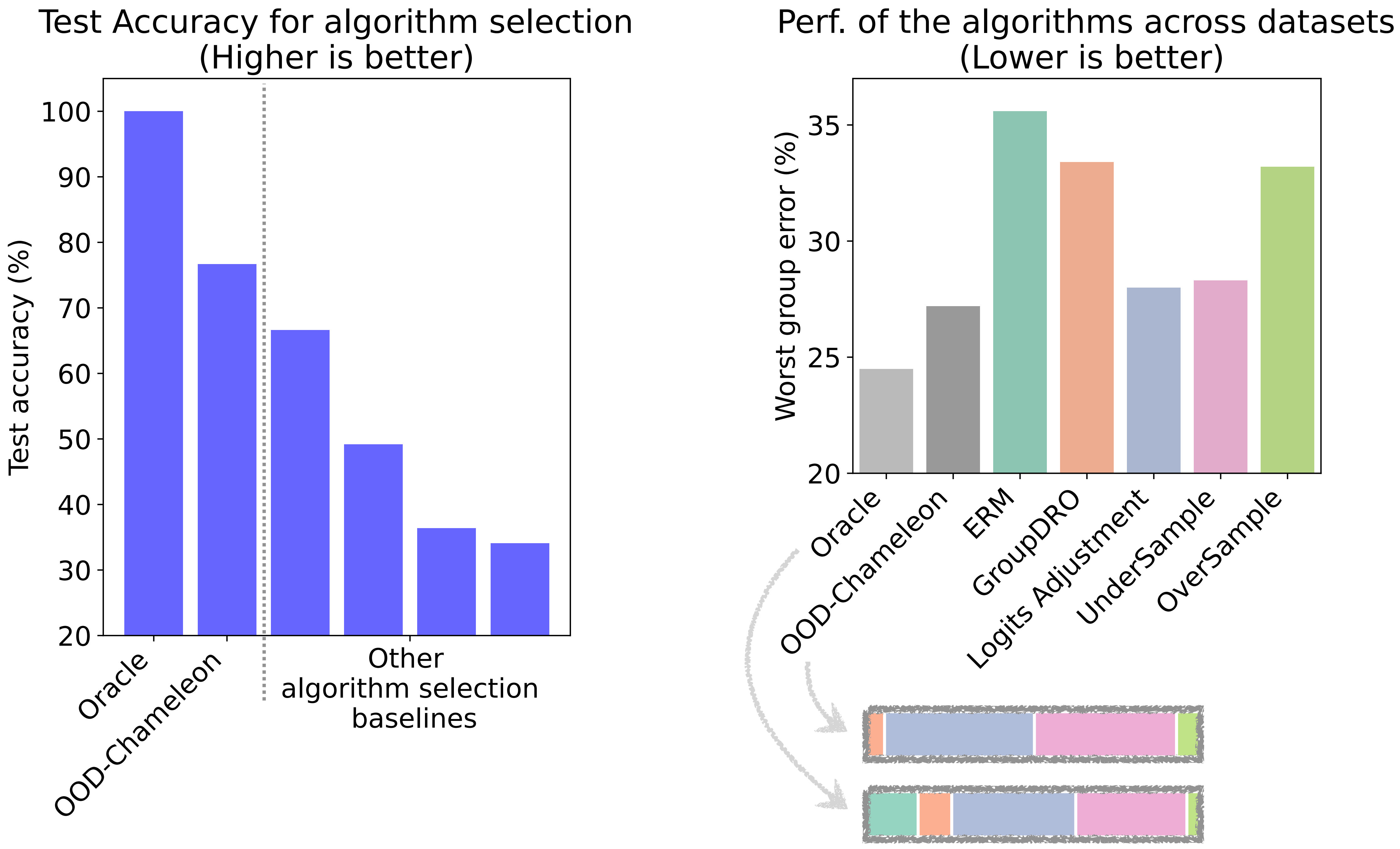
Results

Vision datasets

CelebA



MetaShift



⚙️ Training is on a dataset of ~600 datasets from CelebA, and evaluation is on ~150 and ~130 unseen datasets from CelebA and MetaShift.

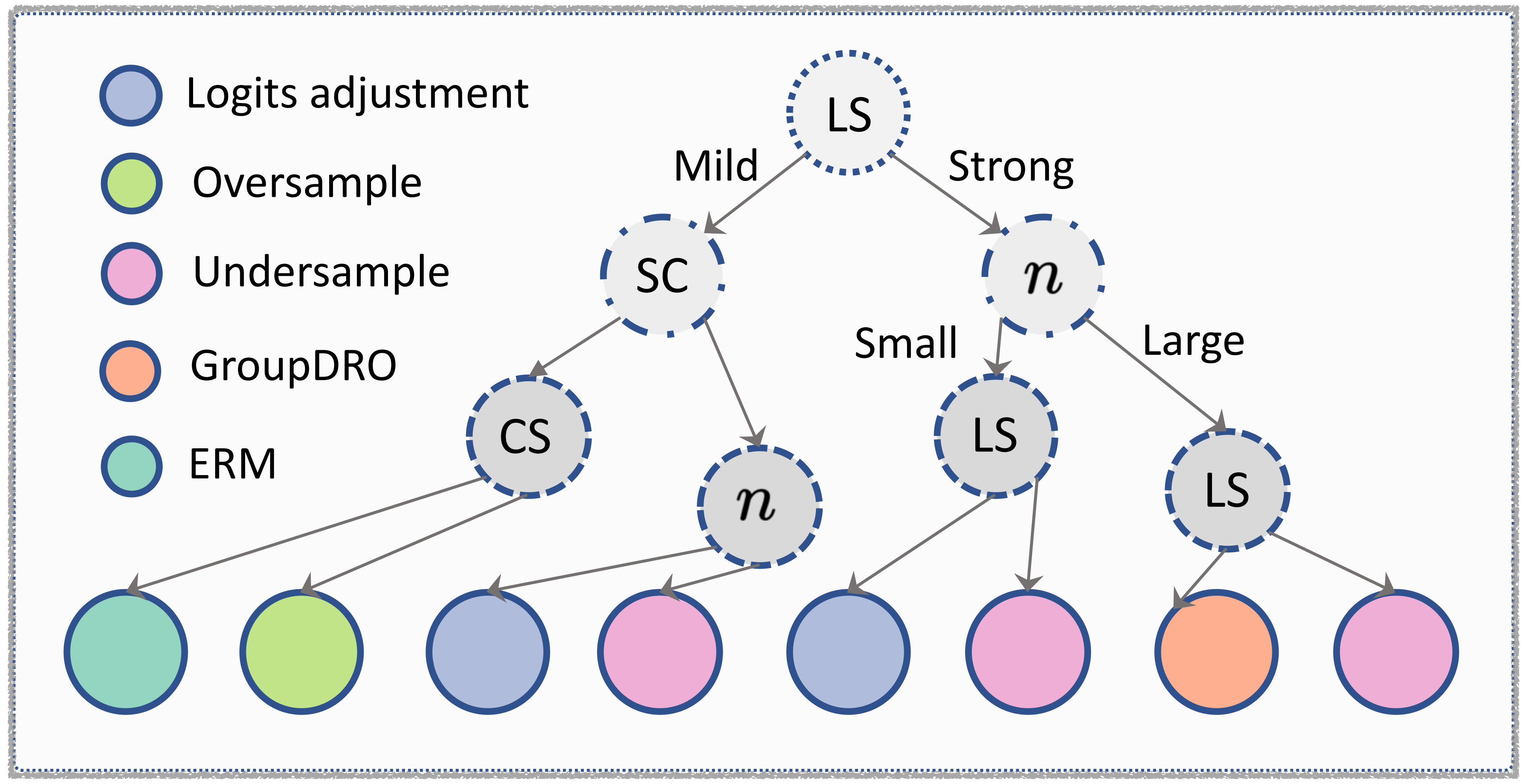
What does OOD-Chameleon learn?

(c.f. Linear)

💡 Learned decision rules are *non-linear*, and cannot be explained by *memorization* of a large number of samples.

(c.f. k-NN)

	ResNet18		CLIP (ViT-B/32)	
Implementation	0–1 ACC. ↑	WG error ↓	0–1 ACC. ↑	WG error ↓
Linear	63.4 $\pm$1.0	49.6 $\pm$0.3	67.6 $\pm$1.8	42.1 $\pm$0.3
k-NN	38.6 $\pm$0.7	49.3 $\pm$0.2	50.0 $\pm$1.3	41.5 $\pm$0.4
Decision tree	73.1 $\pm$0.8	48.1 $\pm$0.3	74.3 $\pm$0.7	39.4 $\pm$0.5
MLP	**75.0** $\pm$1.3	**47.7** $\pm$0.2	**78.5** $\pm$0.8	**39.1** $\pm$0.2



Takeaways

🔎 Algorithm selection is an often-overlooked key factor in OOD generalization.

📍 Non-trivial (and learnable) mapping from measurable data properties to algorithm suitability exists.

It is non-linear, and transferrable to unseen datasets…

⭐ These findings suggest possibilities for better leveraging and understanding existing OOD generalization algorithms.

E.g. understanding the inductive bias of the algorithms…

🛠️ We release the tool to construct diverse distribution shifts for future research.