Montréal 2020

24th ACM International Systems and Software Product Line Conference

Thanks to our Sponsors!

Bronze Level

pure-systems

Supporters

acm Association for Computing Machinery

SIGSOFT

ÉTS Le génie pour l'industrie

MetaCase

ELSEVIER

Science of Computer Programming

Machine Learning and Configurable Systems: A Gentle Introduction
(Tutorial at SPLC'20)

Juliana Alves Pereira, Hugo Martin,
Paul Temple, Mathieu Acher

https://github.com/VaryVary/

DiverSE
Diversity-Centric Software Engineering
Inria
Université de Rennes 1
INSA Rennes
CNRS
PUC Rio
Université de Namur
NADI Namur Digital Institute

Tutorial

- Part 1 (theoretical)
  - Motivation and a complete overview of the progress made in this field
  - Machine Learning Background
- Part 2 (practical)
  - Specialization: the case of VaryLaTeX
- Part 3 (practical)
  - Performance and bug prediction: the case of x264
- Part 4 (theoretical)
  - Conclusion

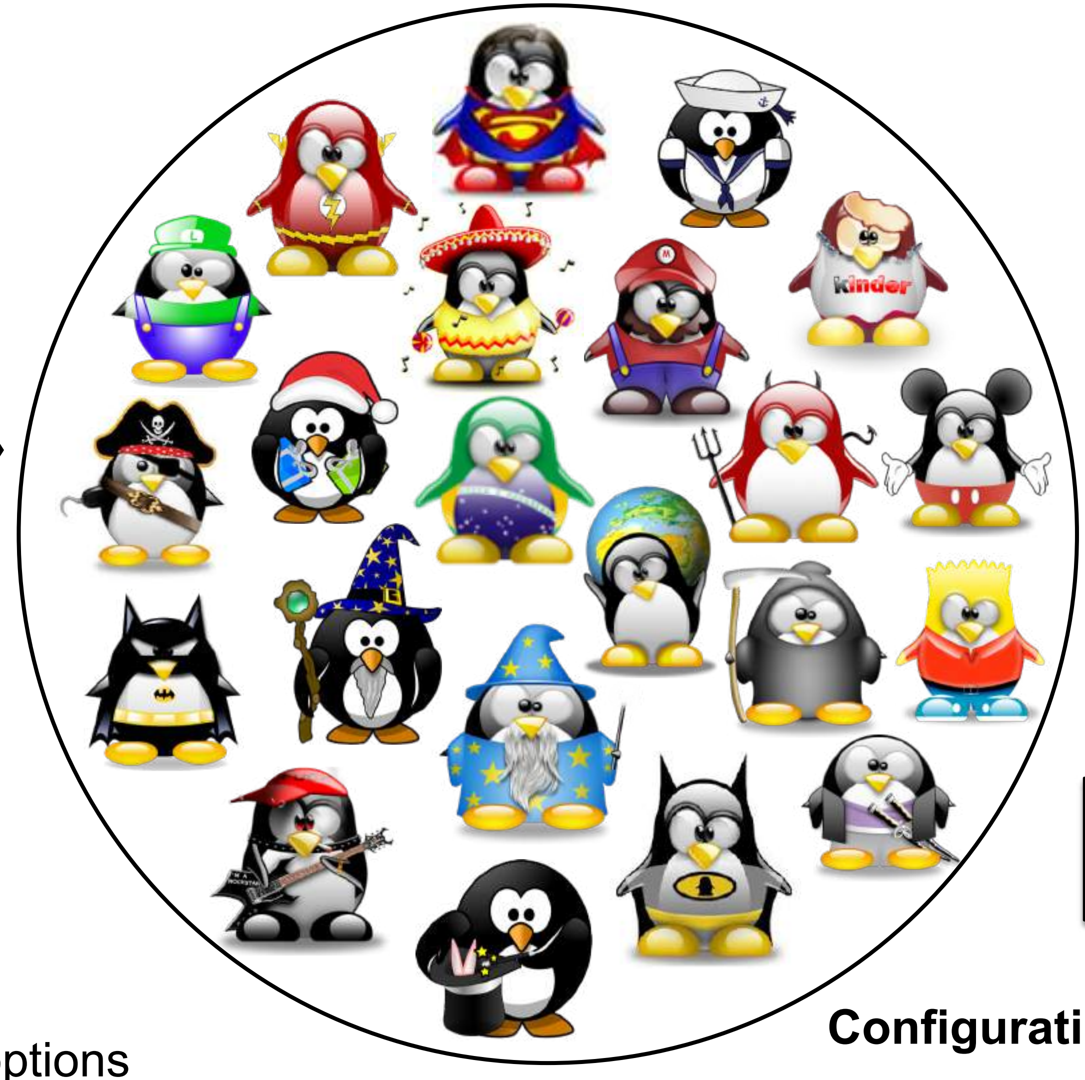
Link to the material (including slides, data, procedures):
https://github.com/VaryVary/ML-configurable-SPLCTutoria

Learning Software Configuration Spaces
(Tutorial at SPLC’20: Part 1)

Juliana Alves Pereira, Hugo Martin,
Paul Temple, Mathieu Acher
https://github.com/VaryVary/

Software Variability

Features



Configuration space

Learning

Sample

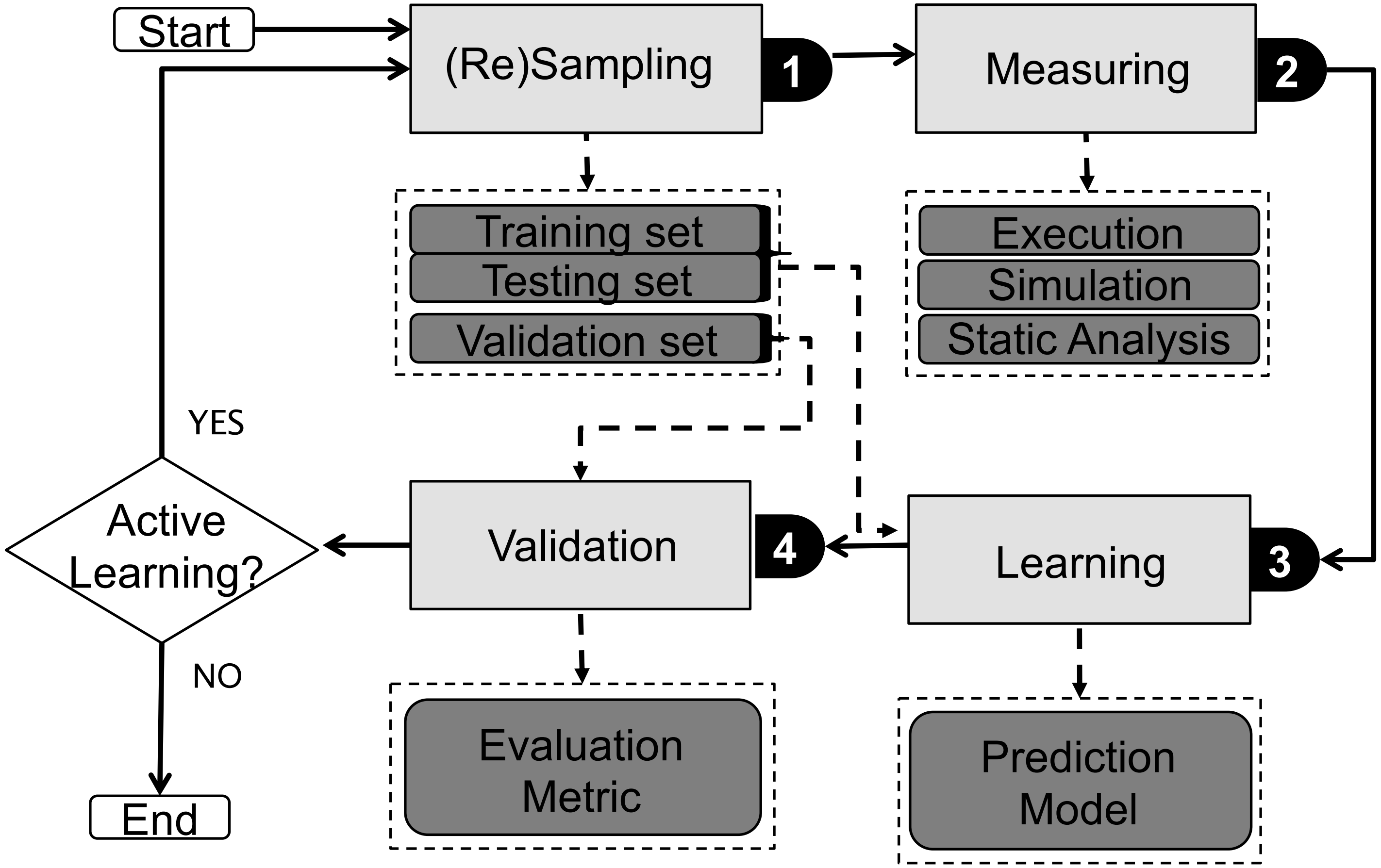
How they can be measured and evaluated?

Linux™

15,000+ features

Tri-state values and numerical options

Learning Stages



- **Research Questions:**

RQ1. What are the **applications** addressed for learning techniques to explore configuration spaces?

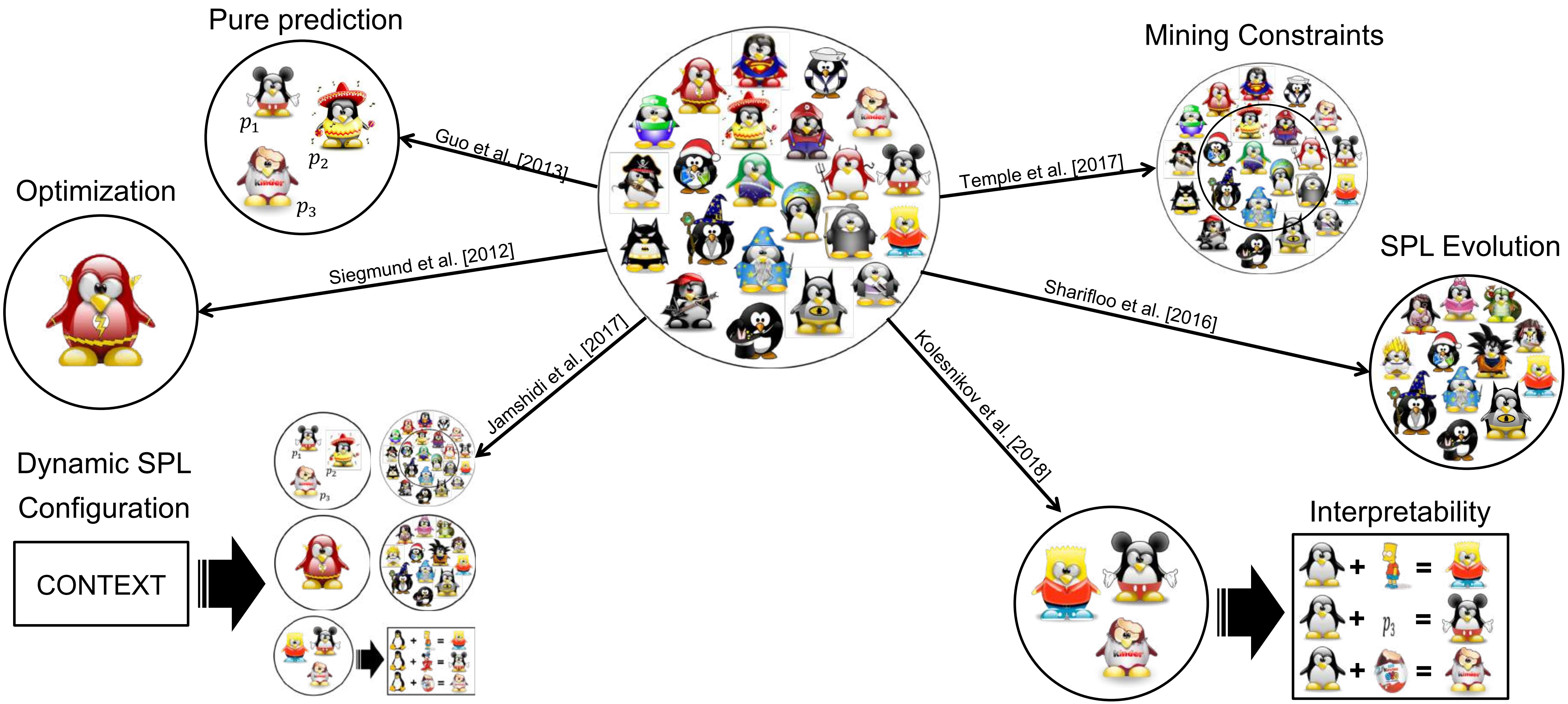
RQ2. Which **sampling** methods are used by these techniques?

RQ3. How do the proposed techniques deal with **measurements** of non-functional properties?

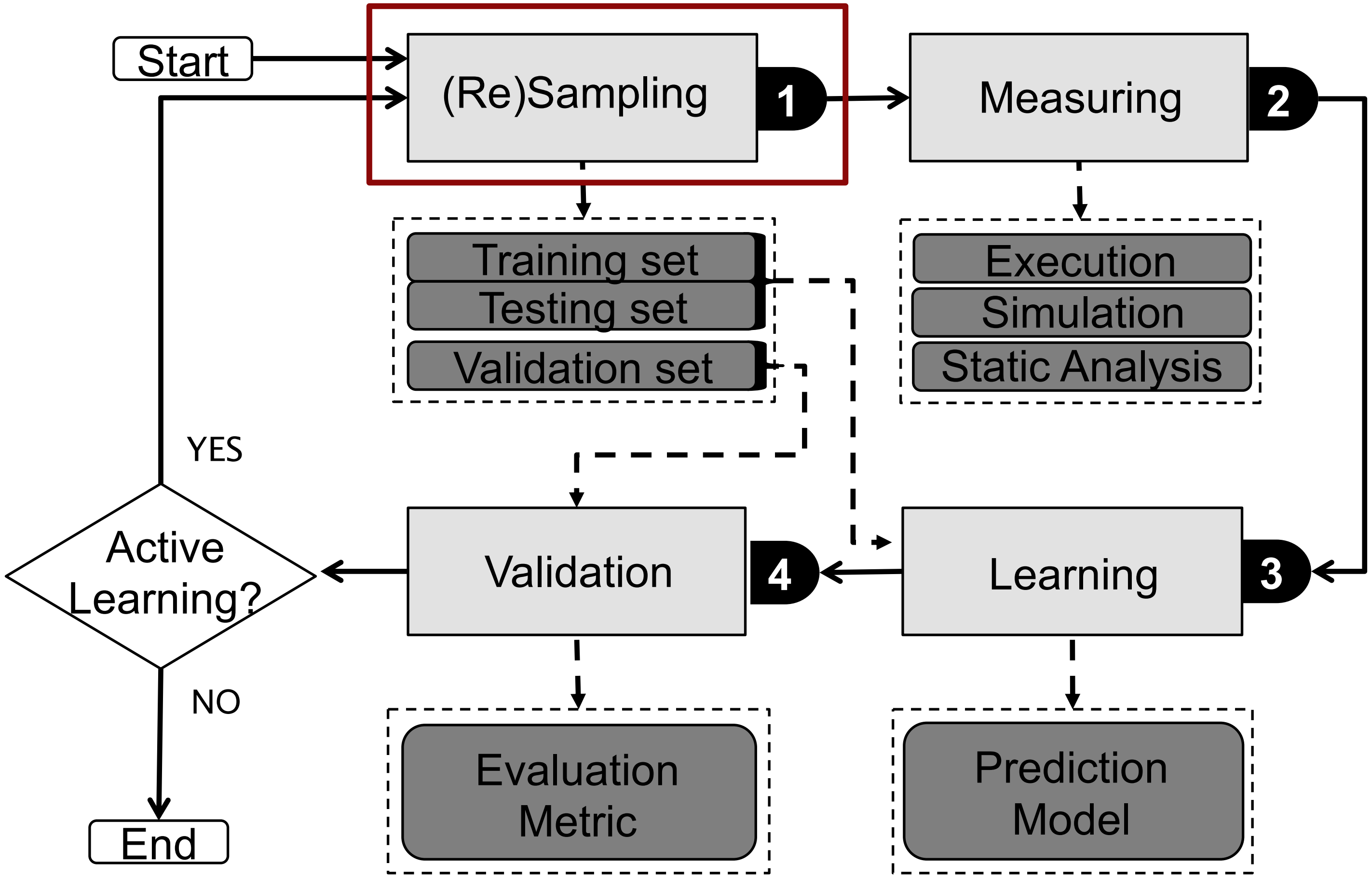
RQ4. Which **learning** techniques have been proposed in the literature?

RQ5. How are these techniques **validated**?

RQ1: Applicability



Learning Stages



- **Research Questions:**

RQ1. What are the **applications** addressed for learning techniques to explore configuration spaces?

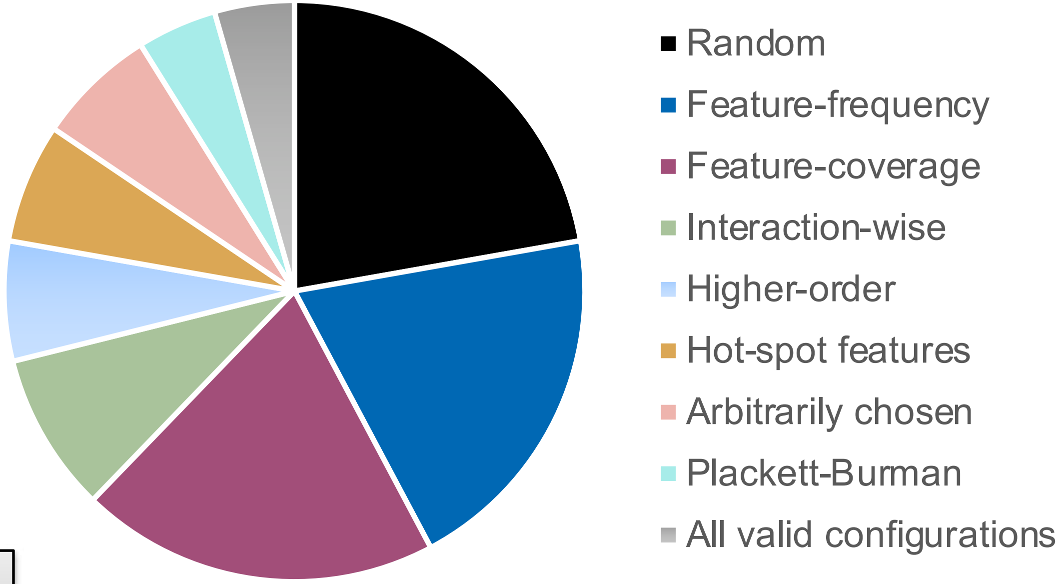
RQ2. Which **sampling** methods are used by these techniques?

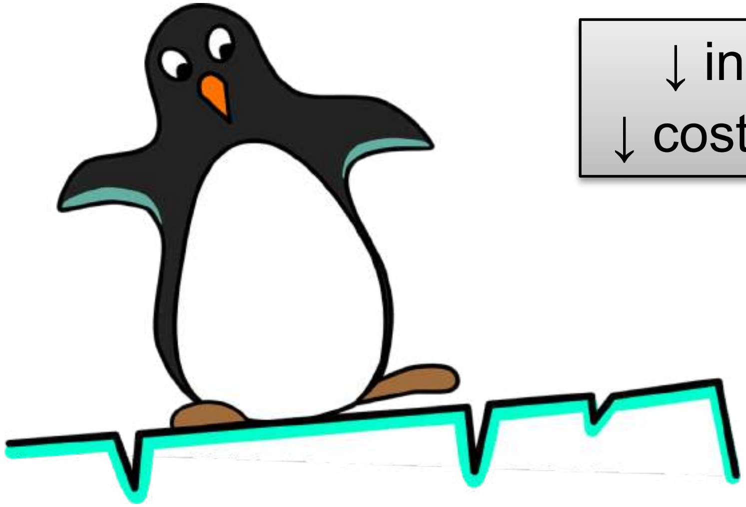
What sampling technique to use?

What is an ideal sample size?

RQ2: Sampling

- Multi-objective problem
  - SAT constrained sampling
  - The “training sampling” should be representative
  - Not infinite resources (limited budget)





↓ invalid configurations,
↓ cost and ↓ prediction error

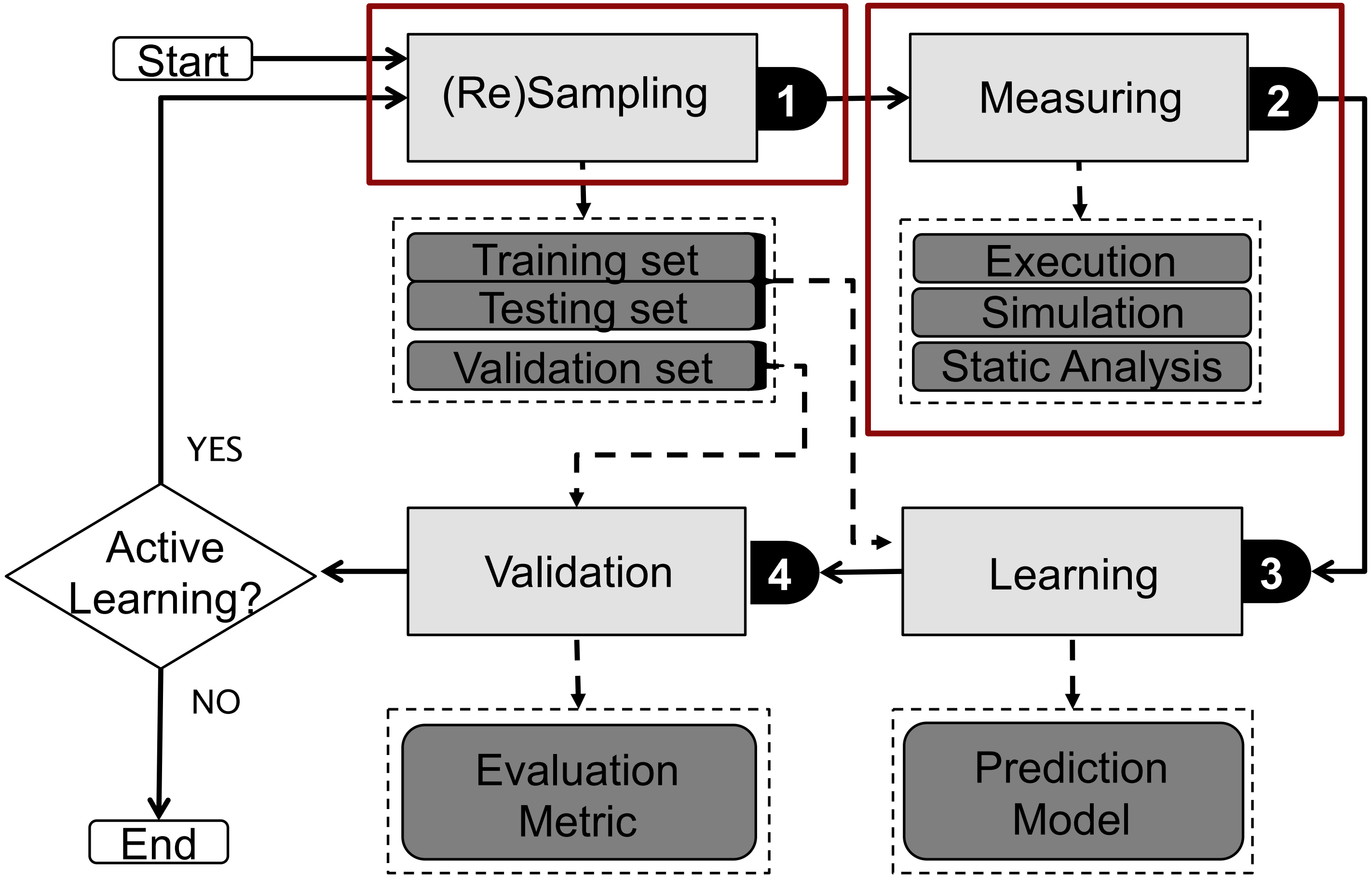
Random sampling

Feature coverage

Feature frequency

References	Sample Method
[10, 12, 18, 19, 33, 34, 36, 45, 51, 52]	Random sampling
[33, 41, 45, 46, 47, 48, 49, 57, 59]	Feature-frequency heuristic
[33, 41, 45, 46, 47, 48, 59, 61, 63]	Feature-coverage heuristic
[21, 47, 48, 63]	Interaction-wise heuristic
[46, 53, 56]	Higher-order interactions
[46, 56, 58]	Hot-spot features
[6, 32, 50]	Arbitrarily chosen
[45, 57]	Plackett-Burman design
[22, 21]	All valid configurations

Learning Stages



- **Research Questions:**

RQ1. What are the **applications** addressed for learning techniques to explore configuration spaces?

RQ2. Which **sampling** methods are used by these techniques?

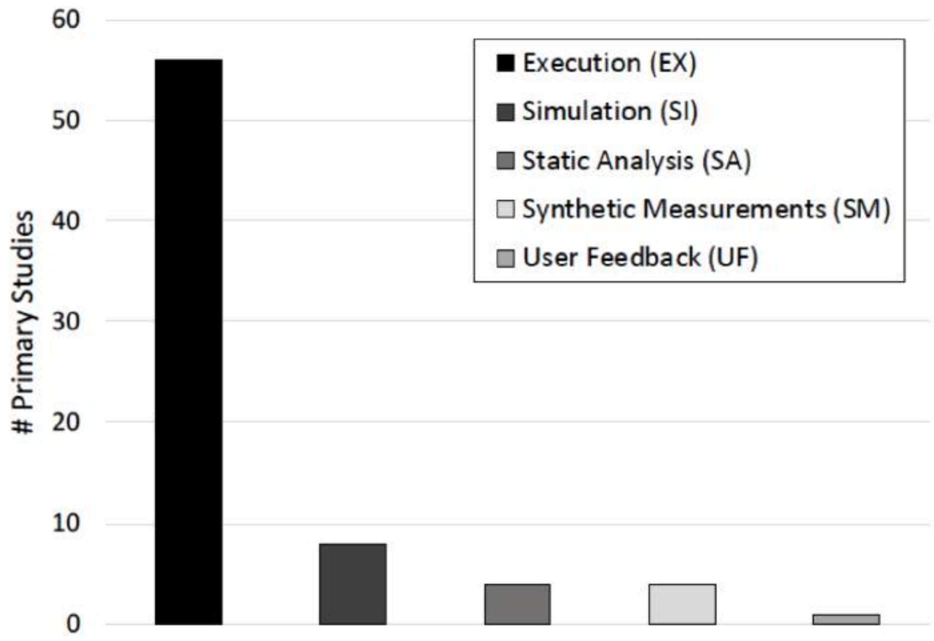
RQ3. How do the proposed techniques deal with **measurements** of non-functional properties?

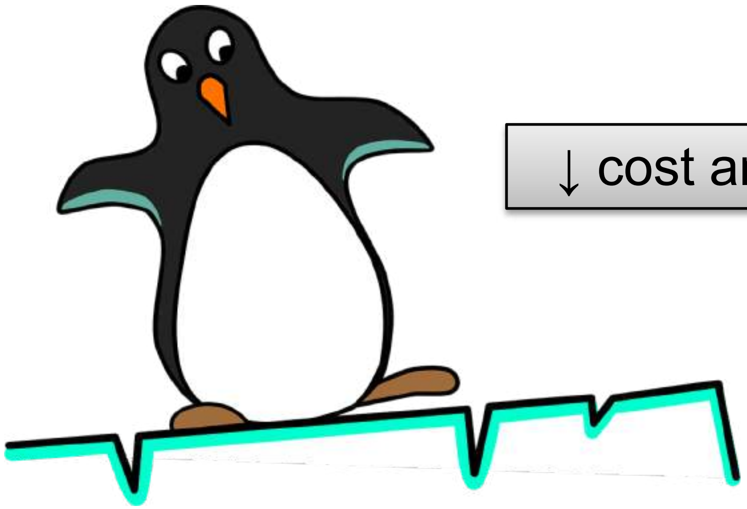
How to measure a sample?

Which properties are measured?

RQ3: Measuring

- Measuring properties of configurations
  - Quantitative and qualitative measurements
  - Execution vs. simulation vs. static analysis vs. user feedback vs. synthetic measurements
  - Consider hardware and external influences
    - Some properties are less costly





↓ cost and ↓ measurement error

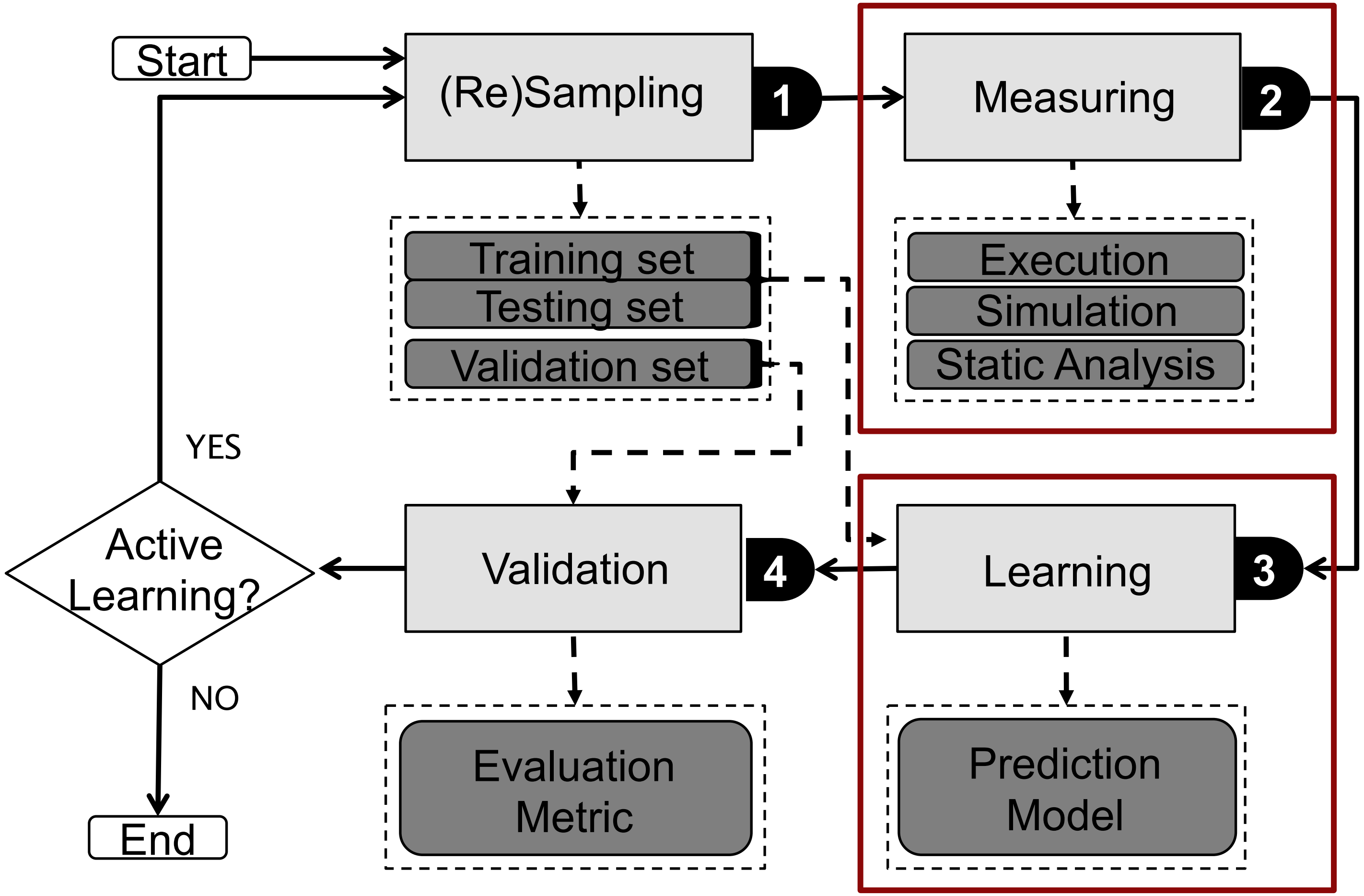
response time

code complexity

CPU power consumption

Non-Functional Property	Type	Non-Functional Property	Type
area	QT	Peak Signal to Noise Ratio	QT
compressed size	QT	response rate	QT
CPU power consumption	QT	response time	QT
CPU usage	QT	runtime	QT
encoding time	QT	throughput	QT
iterations	QT	video quality	QL
latency	QT	workload	QT
footprint	QT	code complexity	QT
performance[12]	QT	memory consumption	QT

Learning Stages



- **Research Questions:**

RQ1. What are the **applications** addressed for learning techniques to explore configuration spaces?

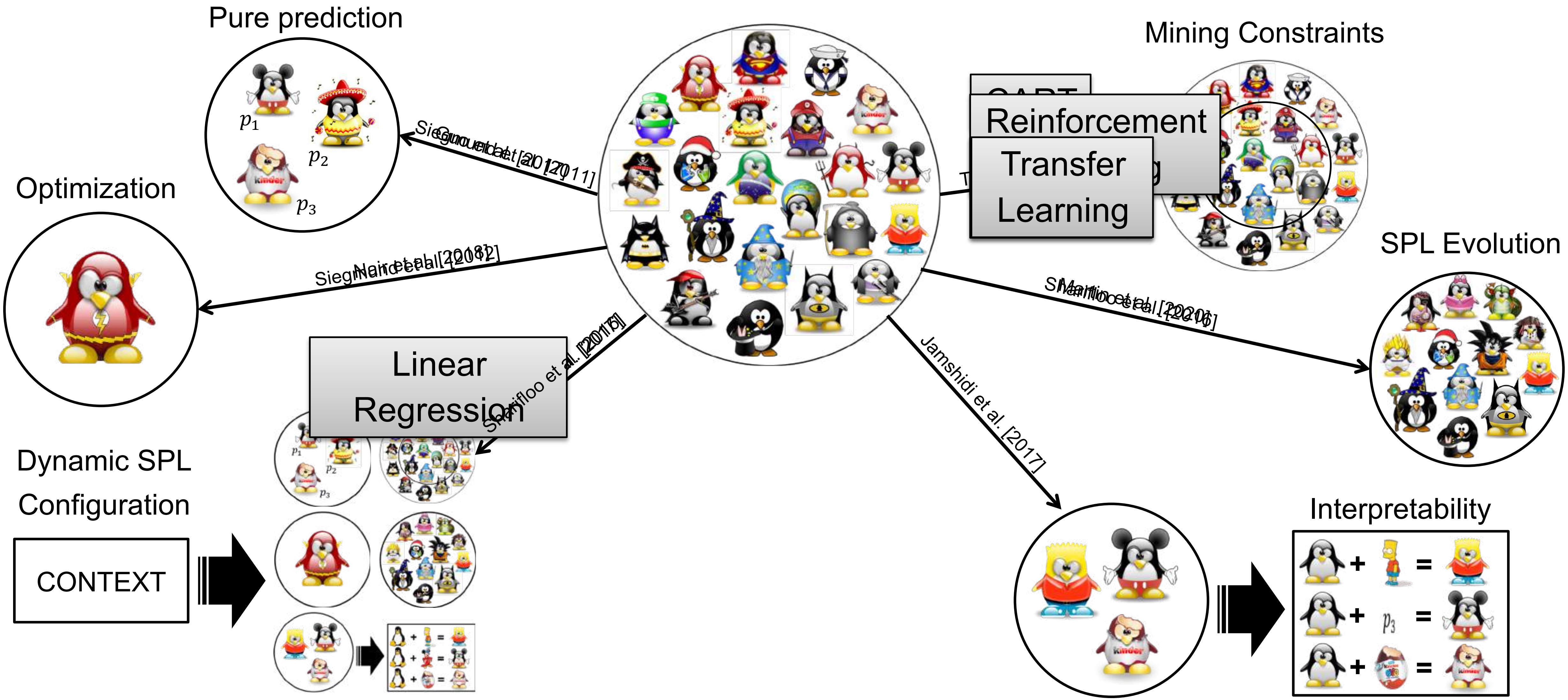
RQ2. Which **sampling** methods are used by these techniques?

RQ3. How do the proposed techniques deal with **measurements** of non-functional properties?

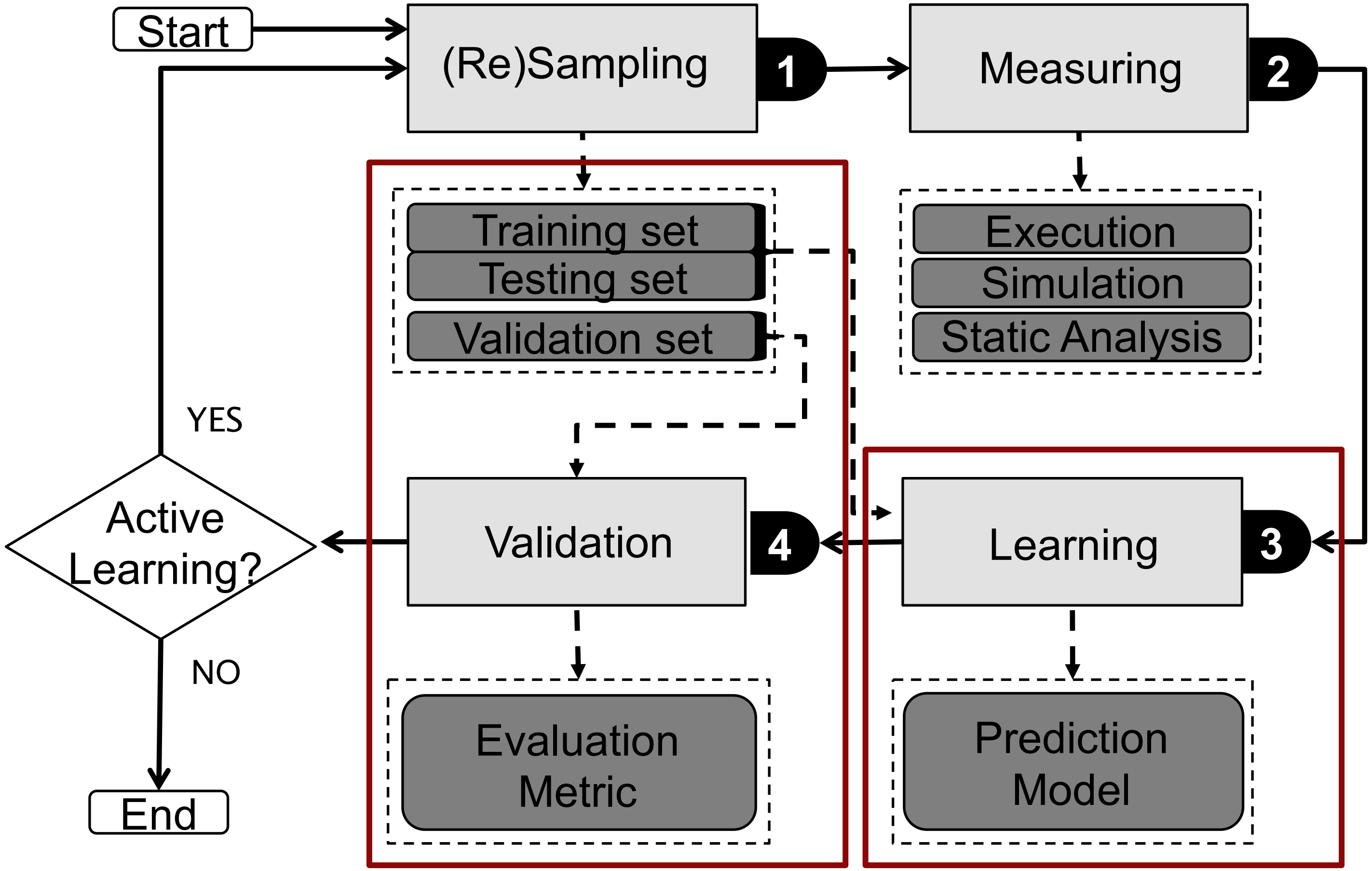
RQ4. Which **learning** techniques have been proposed in the literature?

What learning technique to use?

RQ4: Learning



Learning Stages



- **Research Questions:**

RQ1. What are the **applications** addressed for learning techniques to explore configuration spaces?

RQ2. Which **sampling** methods are used by these techniques?

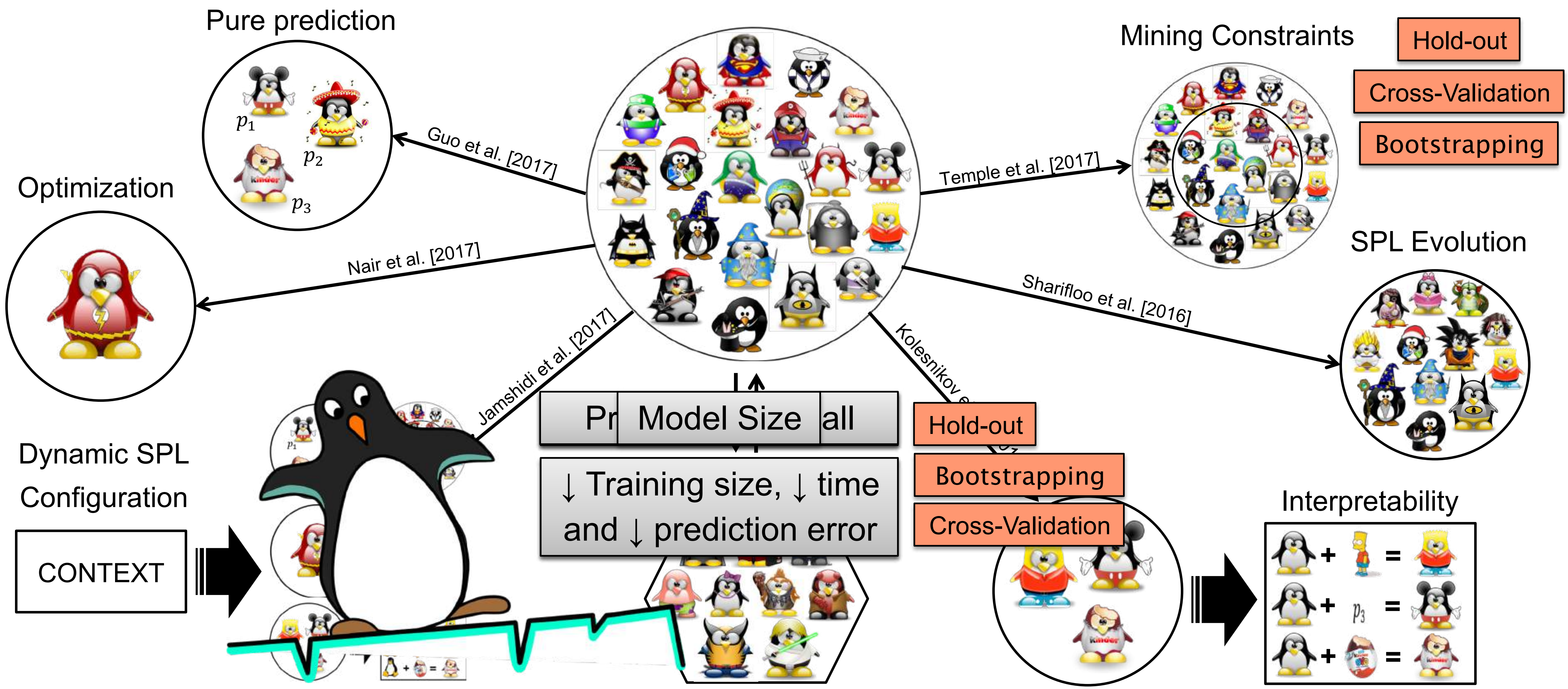
RQ3. How do the proposed techniques deal with **measurements** of non-functional properties?

RQ4. Which **learning** techniques have been proposed in the literature?

RQ5. How are these techniques **validated**?

Does it work in practice? How to evaluate? What are the challenging configurable systems?

RQ5: Validation



RQ5: Validation

Reference	Evaluation Metric	Applicability
[67]	Closeness Range, Winner Probability	A1
[76, 77]	Coverage	A1
[70]	Jaccard Similarity	A1
[76]	Performance-Relevant Feature Interactions Detection Accuracy	A1
[69]	t-masked metric	A1
[75]	True Positive (TP) Rate, False Positive (FP) Rate, Receiver Operating Characteristic (ROC)	A1
[65, 46, 29, 47, 36, 37, 38, 39, 67, 45, 32, 68, 84, 71, 72, 59, 44, 35, 94, 79]	Mean Relative Error (MRE)	A1, A2, A3, A4
[86, 66, 56, 31, 84, 35, 79, 77]	Sampling Cost	A1, A2, A3, A4
[32, 68]	Highest Error (HE)	A1, A3
[42, 30, 81, 83, 44, 85, 74, 88, 73, 92, 90]	Mean Absolute Error (MAE)	A1, A3, A4
[72, 101, 96]	Mann-Whitney U-test	A1, A3, A5
[57, 72]	F-test	A1, A4
[60, 9, 100, 70, 64]	Precision, Recall	A1, A4, A5
[62, 69, 64, 101, 75]	Precision, Recall, F-measure	A1, A5
[80]	GAP	A2
[43]	Kullback-Leibler (KL), Pearson Linear Correlation, Spearman Correlation Coefficient	A2
[45, 47]	Structure of Prediction Models	A2
[43, 59]	Rank Correlation	A2, A4
[41, 35]	Domain Experts Feedback	A3
[97]	Error Probability	A3
[83]	Global Confidence, Neighbors Density Confidence, Neighbors Similarity Confidence	A3
[32]	LT15, LT30	A3
[54, 31]	Mean Rank Difference	A3
[81]	Median Magnitude of the Relative Error (MdMRE)	A3
[33]	Pareto Prediction Error	A3
[90, 91, 89]	Rank Accuracy (RA)	A3
[41, 93, 95, 97]	Tuning Time	A3
[81, 92, 57]	Mean Square Error (MSE)	A3, A4
[57]	p-value, R2, Residual Standard Error (RSE)	A4
[99]	Reward	A4
[58]	Statistical Significance	A4
[61, 58]	Qualitative Analysis	A4, A5
[55]	Ranking Constraints	A4, A5
[101]	Delaney's Statistics	A5
[101]	Distance Function, Hyper-volume (HV)	A5
[63]	Equivalence among Combinatorial Models	A5
[63]	Failure Index Delta (FID), Totally Repaired Models (TRM)	A5

Pure prediction: *Siegmund et al. Predicting Performancevia Automated Feature-Interaction* Detection. ICSE, 2012.

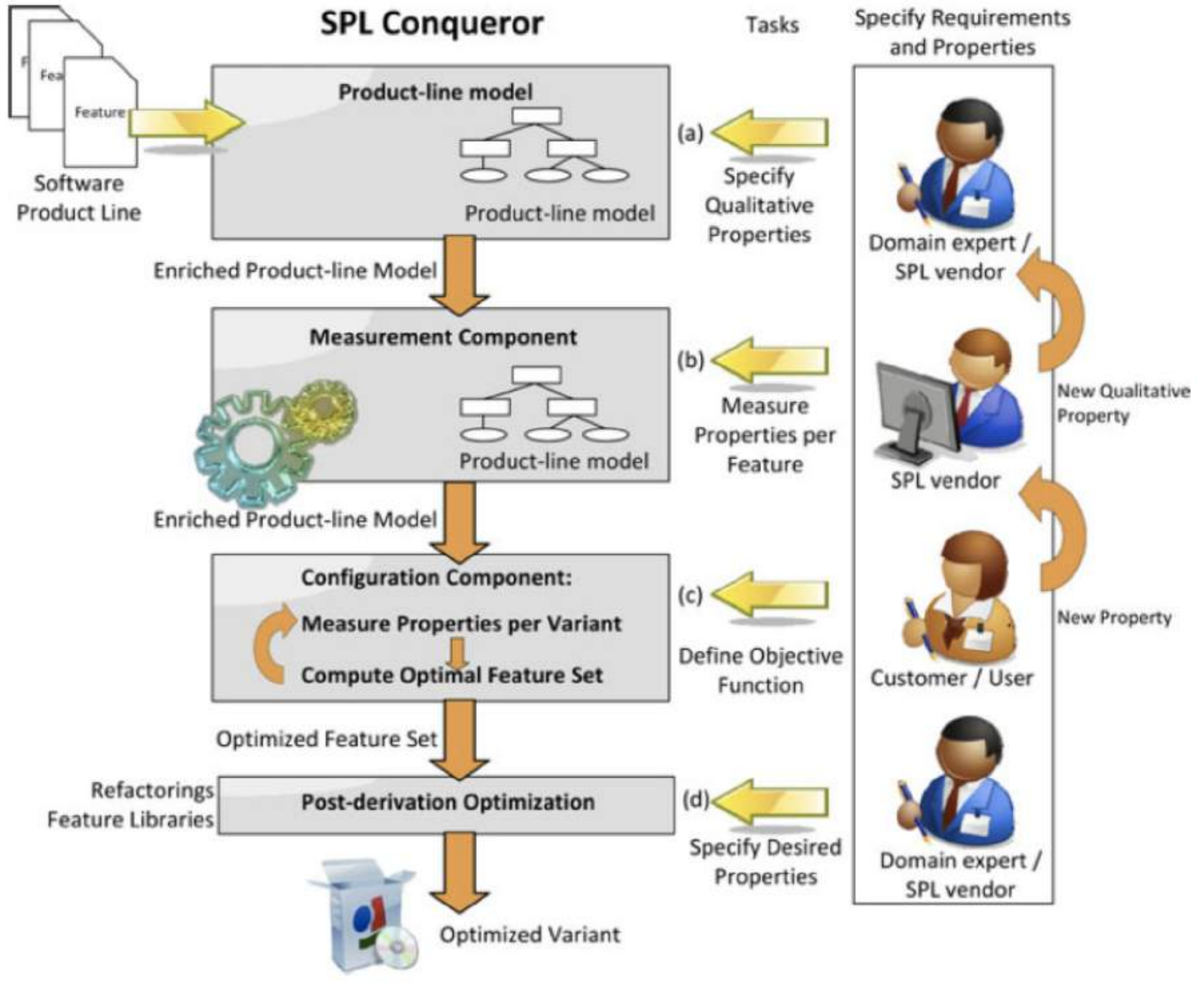


Fig. 5 Process of SPL Conqueror including the different tasks of measurement, configuration and optimization

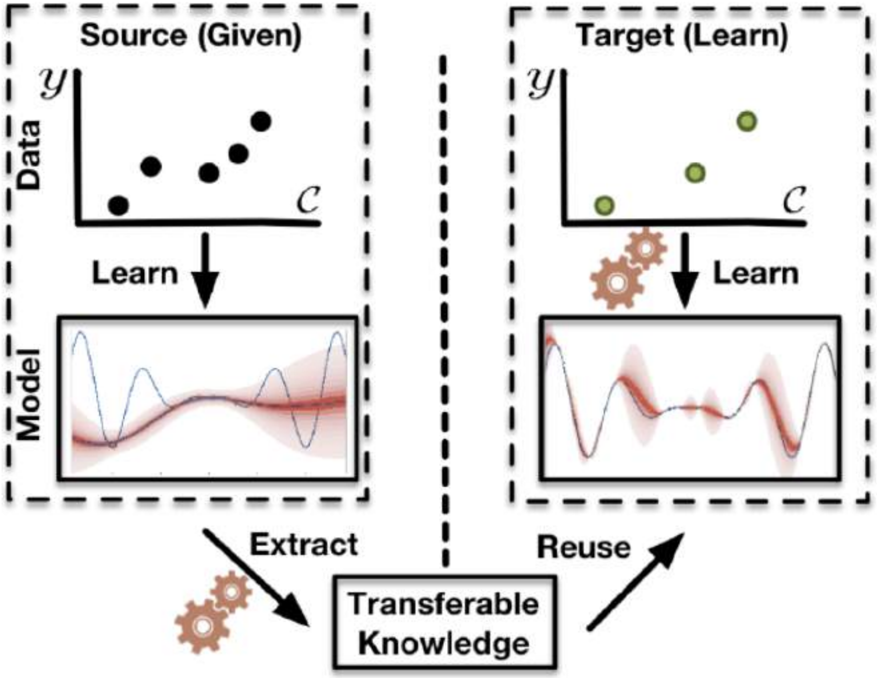
- Sampling: FW, PW, HO, HS, and BF
- Measuring: request rate and time
- Learning: Feature's Influence Delta
- Validation: Mean Fault Rate

EVALUATION RESULTS FOR SIX CASE STUDIES; APPROACHES (APPR.): FEATURE-WISE (FW), PAIR-WISE (PW), HIGHER-ORDER (HO), HOT-SPOT (HS), BRUTE FORCE (BF). MEAN: MEAN FAULT RATE OF PREDICTIONS, STD: STANDARD DEVIATION OF PREDICTIONS.

		Effort			Fault Rate (in %)	
Program	Appr.	Measurements	Time (in h)	Interactions	Distribution	Mean±Std
Berkeley CE	FW	15 (0.6 %)	3	0		44.1±42.3
	PW	139 (5.4 %)	23	14		3.9±5.3
	HO	160 (6.3 %)	27	22		2.8±3.7
	HS	164 (6.4 %)	27	22		2.8±3.7
	BF	2560 (100 %)	426	-	-40 -20 0 20 40	—
Berkeley JE	FW	10 (3 %)	8.4	0		17.7±19.6
	PW	48 (12 %)	40	24		8.5±9.6
	HO	116 (29 %)	97	51		3.8±5.7
	HS	162 (40.5 %)	137	69		1.7±3.5
	BF	400 (100 %)	335	-	-40 -20 0 20 40	—
Apache	FW	9 (4.7 %)	10	0		14.9±24.8
	PW	29 (15.1 %)	32	18		7.7±11.2
	HO	80 (41.7 %)	89	44		11.6±22.7
	HS	143 (74.5 %)	159	73		5.3±10.8
	BF	192 (100 %)	213	-	-40 -20 0 20 40	—

Interpretability: *Jamshidi et al. Transfer Learning for PerformanceModeling of Configurable Systems: An Exploratory Analysis*. ASE, 2017.

- Sampling: Random
- Measuring: time
- Learning: Linear Regression , CART, and Multinomial Logistic Regression
- Validation: Kullback-Leibler, Pearson Linear Correlation, Spearman Correlation Coefficient, Rank correlation,



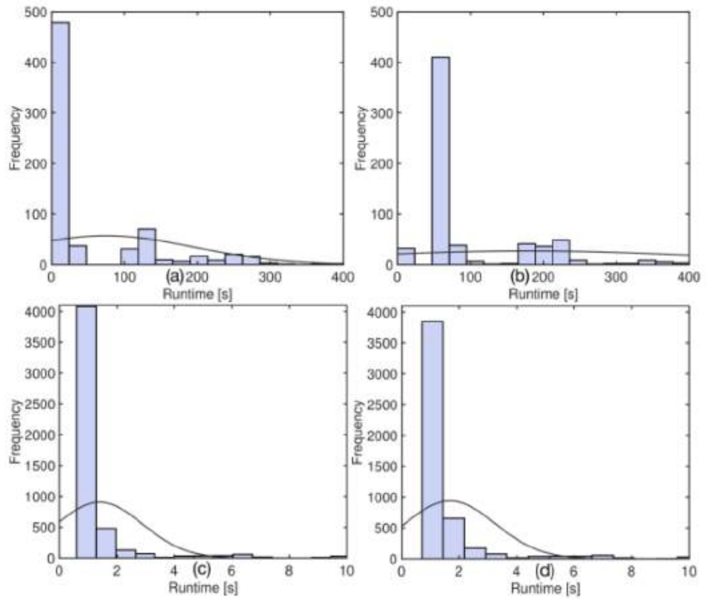


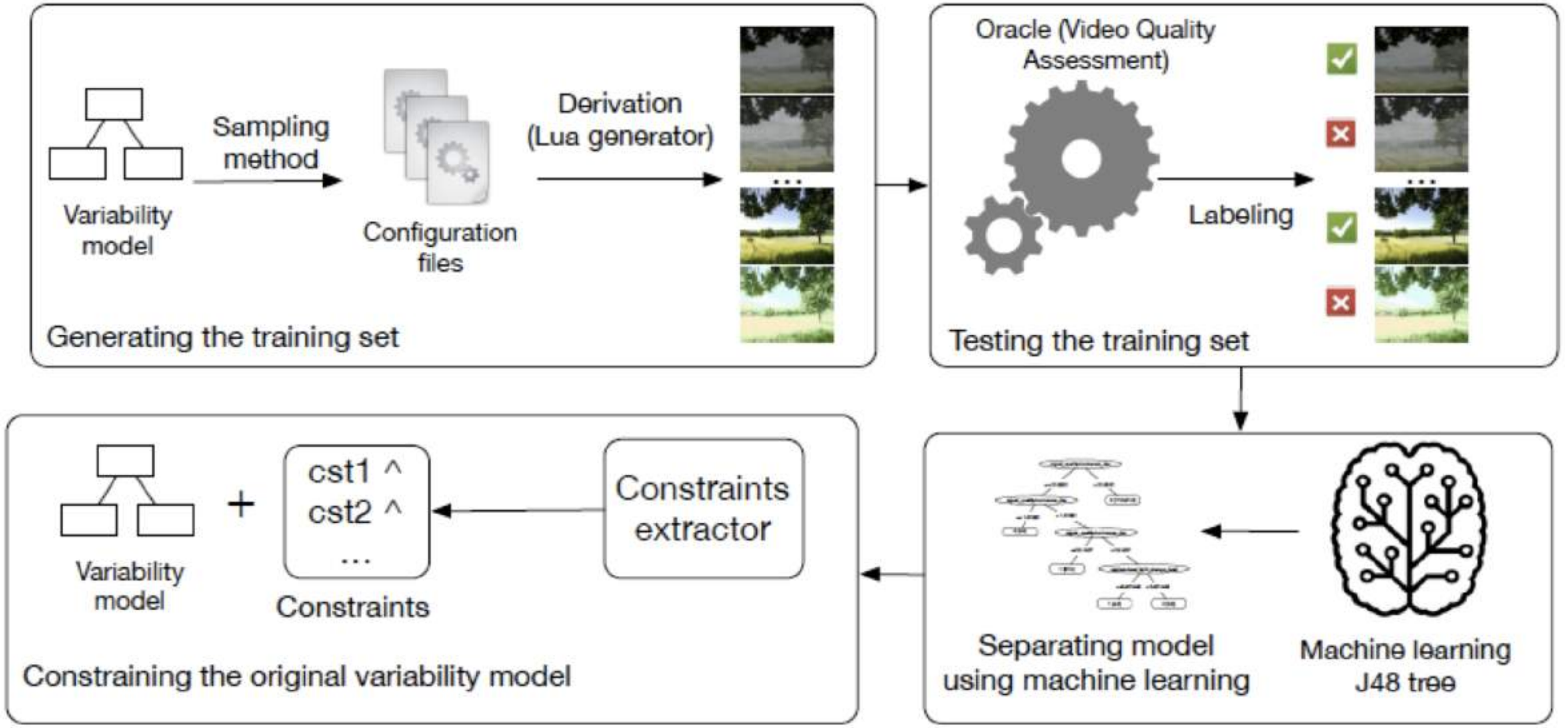
Fig. 2: Performance distributions of environments, depending on the severity of change, may be dissimilar, $D_{KL}^{ec_1} = 25.02$ (a,b), or very similar, $D_{KL}^{ec_{13}} = 0.32$ (c,d).

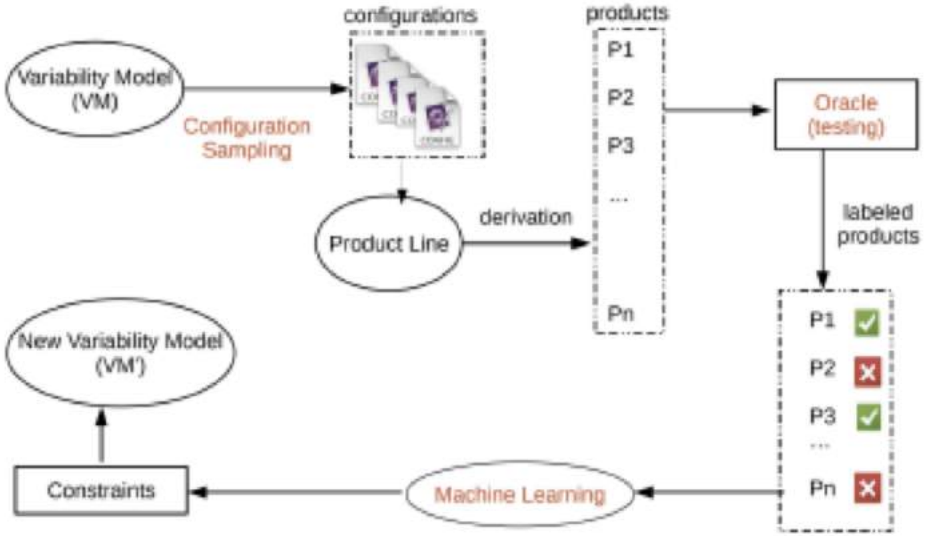
System	Domain	d	$\lvert\mathcal{C}\rvert$	$\lvert H\rvert$	$\lvert W\rvert$	$\lvert V\rvert$
SPEAR	SAT solver	14	16 384	3	4	2
x264	Video encoder	16	4 000	2	3	3
SQLite	Database	14	1 000	2	14	2
SaC	Compiler	50	71 267	1	10	1

d: configuration options; $\mathcal{C}$: configurations; H: hardware environments; W: analyzed workload; V: analyzed versions.

Mining Constraints: *Temple et al. Using Machine Learningto Infer Constraints for Product Lines*. SPLC, 2016.

- Sampling: Random
- Measuring: video quality
- Learning: CART
- Validation: Precision and Recall





		Oracle	
		Faulty	Non-faulty
variability model (VM')	Faulty (invalid)	234 (±57.899)	69.5 (±26.973)
	Non-faulty (valid)	141.1 (±60.440)	3566.2 (±25.804)