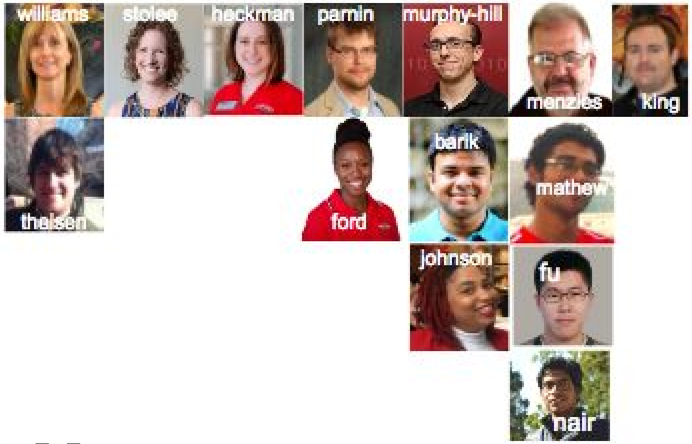
NC STATE UNIVERSITY

... if software engineering then NC State ...



Advanced Analytics

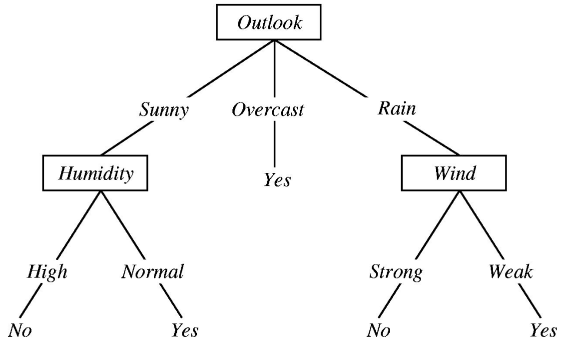
Plant a (decision) TREE and save the world*!

Vivek Nair
North Carolina State University
vivekaxl@gmail.com
vivekaxl.com

* Configure software using less resources

Most Valuable Point

"Information is a source of learning. But unless it is organized, processed, and available to the right people in a format for decision making, it is a burden, not a benefit" -- Dr. William Pollard



Decision Trees - Use Cases

TAR*(ZAN)***2** *[1]*
2002

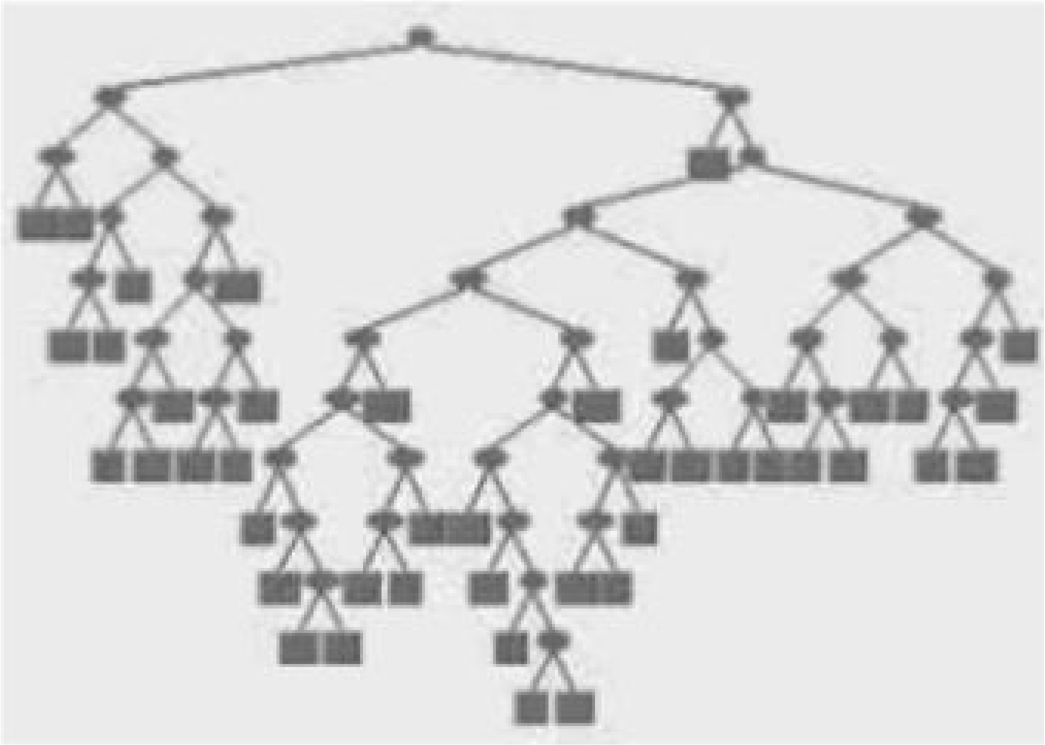
[1] Menzies, Tim, and Ying Hu. "Just enough learning (of association rules): the TAR2 “Treatment” learner." *Artificial Intelligence Review* 25.3 (2006): 211-229.

TAR(ZAN)2 Learns Small Theories

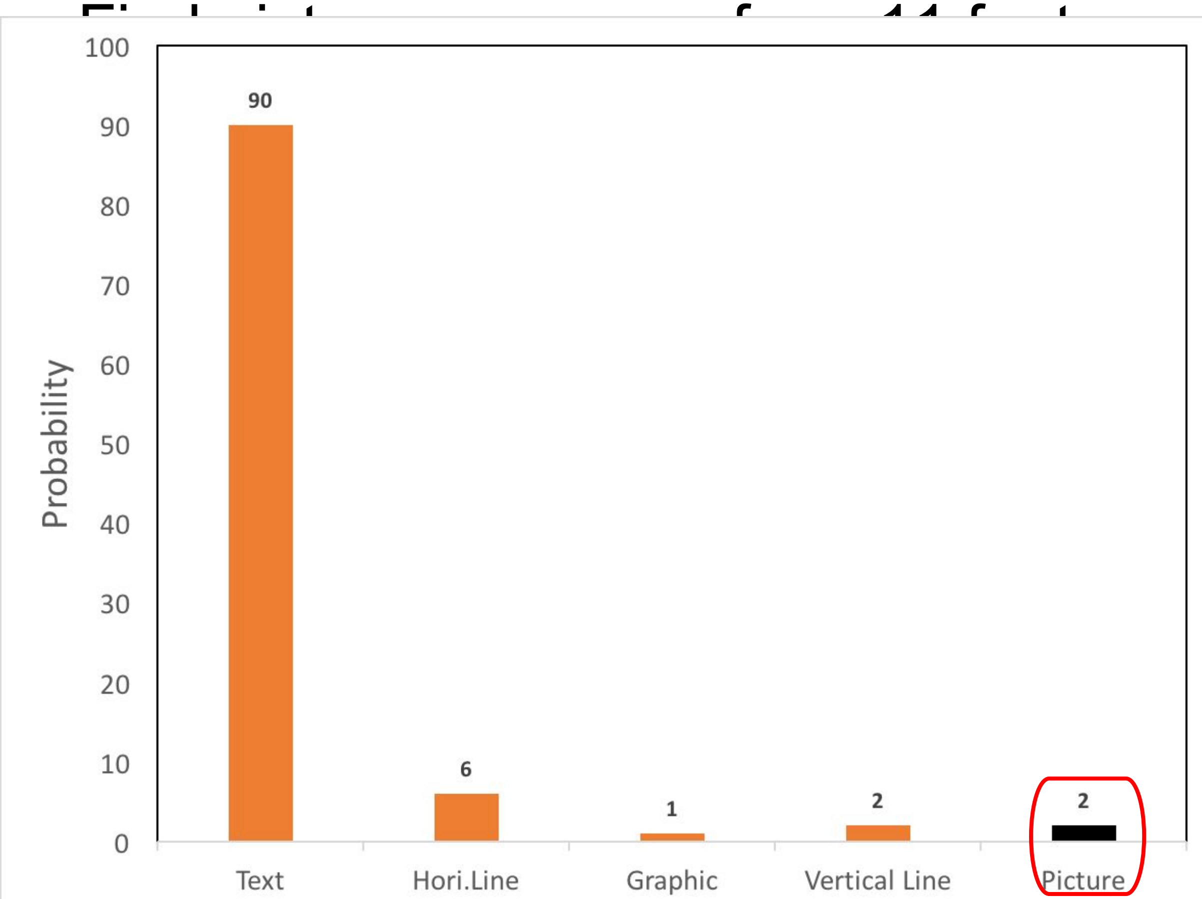
Problem: Find picture on a page from 11 features

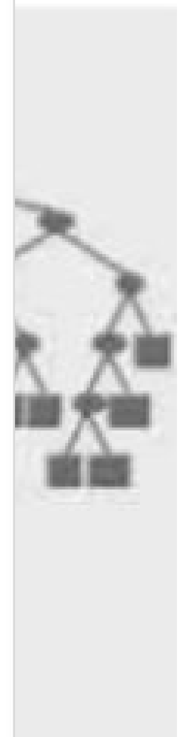
TAR(ZAN)2
Learns Small Theories

Problem: Find picture on a page from 11 features

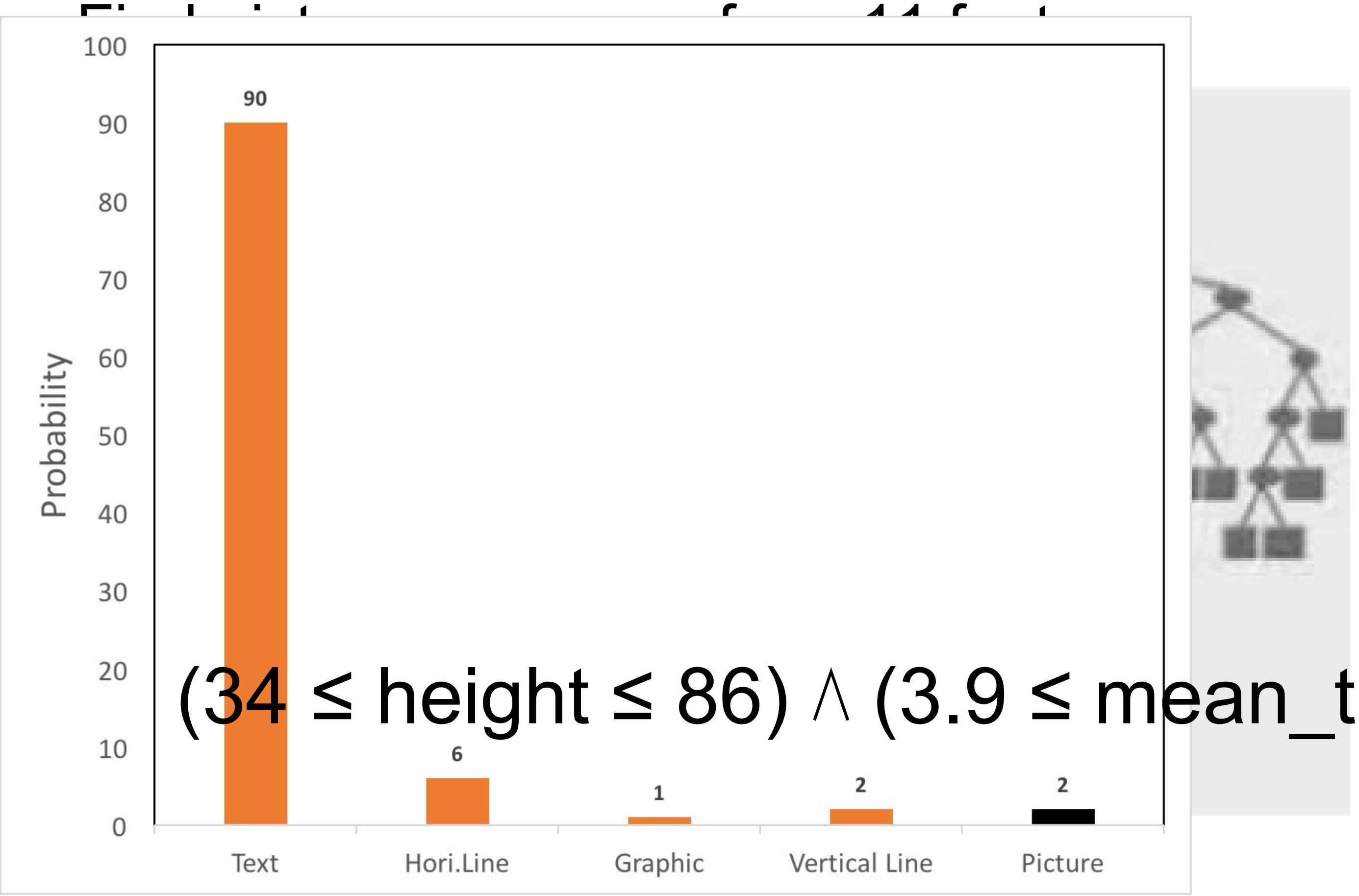


TAR(ZAN)2
Learns Small Theories



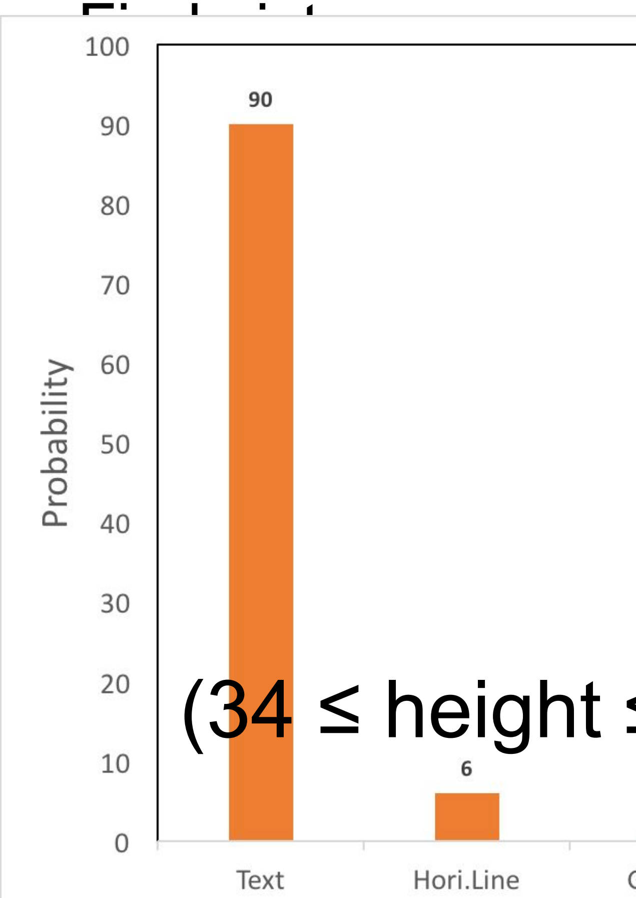


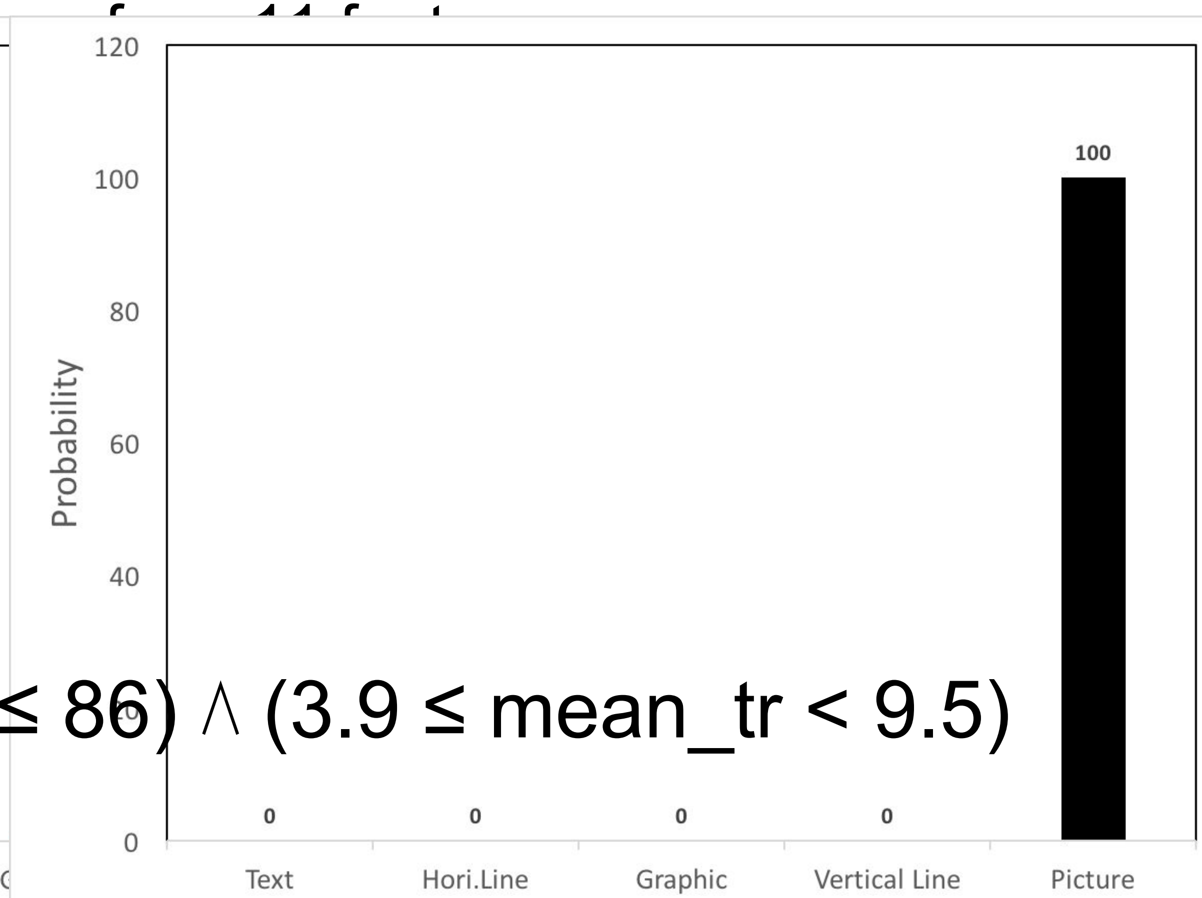
TAR(ZAN)2 Learns Small Theories



$(34 \le \text{height} \le 86) \wedge (3.9 \le \text{mean\_tr} < 9.5)$

TAR(ZAN)2
Learns Small Theories





$$(34 \leq \text{height} \leq 86) \wedge (3.9 \leq \text{mean\_tr} < 9.5)$$

Decision Trees - Use Cases

TAR**(ZAN)**2 [1]
2002

[1] Menzies, Tim, and Ying Hu. "Just enough learning (of association rules): the TAR2 “Treatment” learner." *Artificial Intelligence Review* 25.3 (2006): 211-229.

Decision Trees - Use Cases

TAR**(ZAN)**2 *[1]*
2002

SWAY*[2]*
2016

Performance Optimization*[3]*
2017

XTREE*[4]*
2017

[1] Menzies, Tim, and Ying Hu. "Just enough learning (of association rules): the TAR2 “Treatment” learner." *Artificial Intelligence Review* 25.3 (2006): 211-229.
[2] Nair et al. "An (accidental) exploration of alternatives to evolutionary algorithms for sbse." *SSBSE*- 2016.
[3] Guo et al. “Variability-aware performance prediction: A statistical learning approach." *ASE*-2013.
[4] Krishna et al.. "Less is more: Minimizing code reorganization using XTREE." *IST-2017*

Decision Trees - Use Cases

TAR**(ZAN)**2 *[1]*
2002

Optimization
SWAY*[2]*
2016

Software Variability
Performance Optimization*[3]*
2017

Planning
XTREE*[4]*
2017

[1] Menzies, Tim, and Ying Hu. "Just enough learning (of association rules): the TAR2 "Treatment" learner." *Artificial Intelligence Review* 25.3 (2006): 211-229.
[2] Nair et al. "An (accidental) exploration of alternatives to evolutionary algorithms for sbse." *SSBSE*- 2016.
[3] Guo et al. "Variability-aware performance prediction: A statistical learning approach." *ASE*-2013.
[4] Krishna et al.. "Less is more: Minimizing code reorganization using XTREE." *IST-2017*

Decision Trees - Use Cases

TAR(ZAN)2 [1]
2002

Optimization
SWAY[2]
2016

Software Variability
Performance Optimization[3]
2017

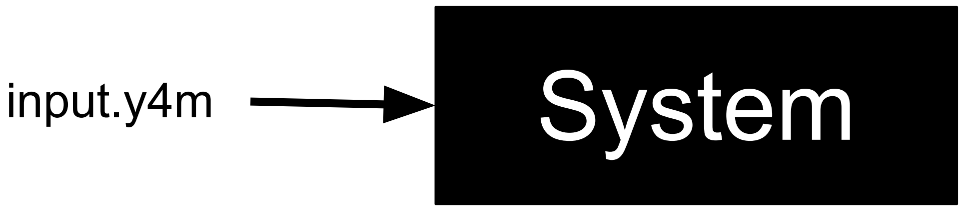
Planning
XTREE[4]
2017

[1] Menzies, Tim, and Ying Hu. "Just enough learning (of association rules): the TAR2 “Treatment” learner." *Artificial Intelligence Review* 25.3 (2006): 211-229.
[2] Nair et al. "An (accidental) exploration of alternatives to evolutionary algorithms for sbse." *SSBSE*- 2016.
[3] Guo et al. “Variability-aware performance prediction: A statistical learning approach." *ASE*-2013.
[4] Krishna et al.. "Less is more: Minimizing code reorganization using XTREE." *IST-2017*

Configurable Systems and Variability

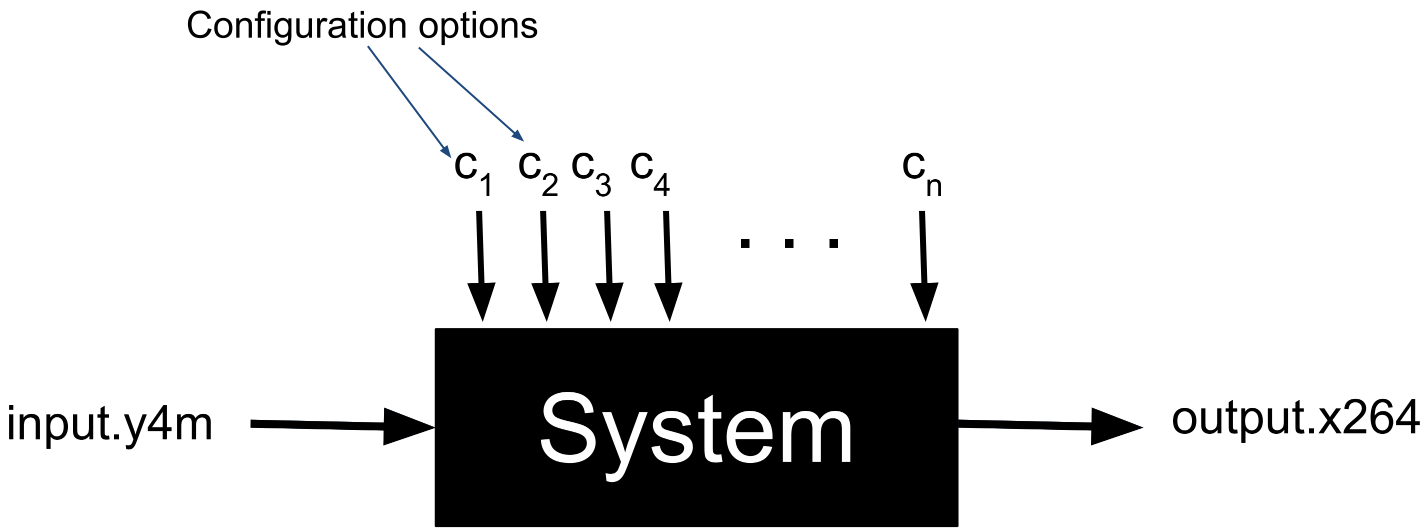
System

Configurable Systems and Variability

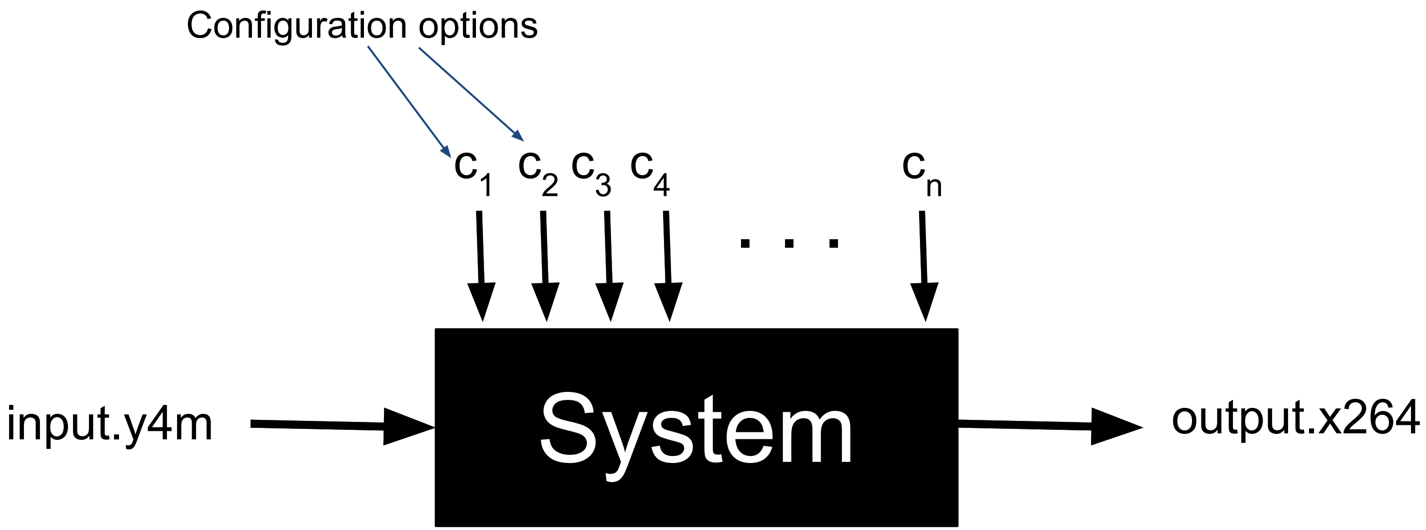


Configurable Systems and Variability

Configurable Systems and Variability

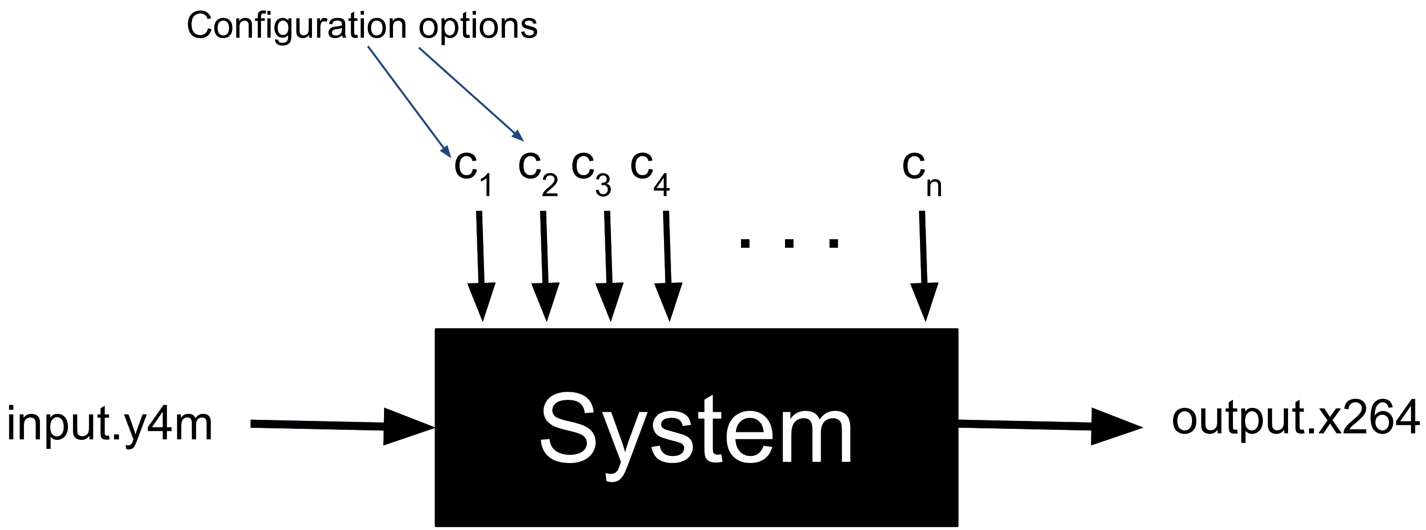


Configurable Systems and Variability



Non-functional behavior: response time, throughput, etc.

Configurable Systems and Variability



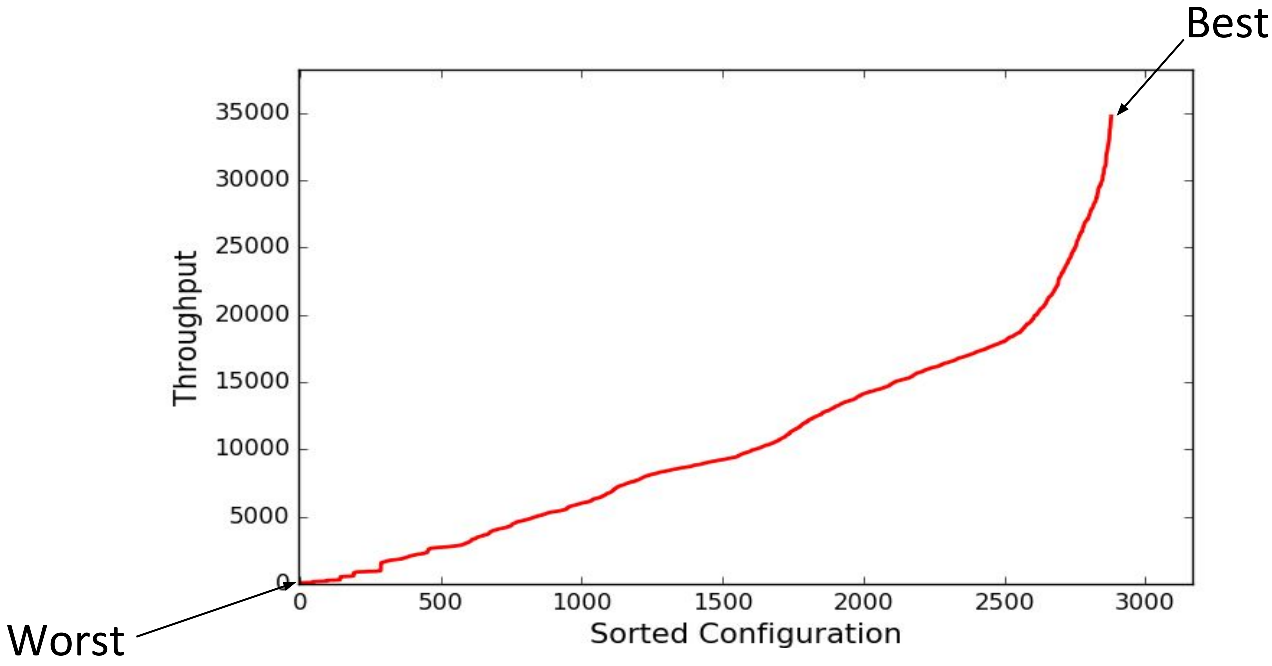
Non-functional behavior: response time, throughput, etc.

Objective: Find (near) **optimal configuration** of a system **with minimal effort**

Performance Optimization is Necessary!

System: Apache Storm
Workload: Word Count

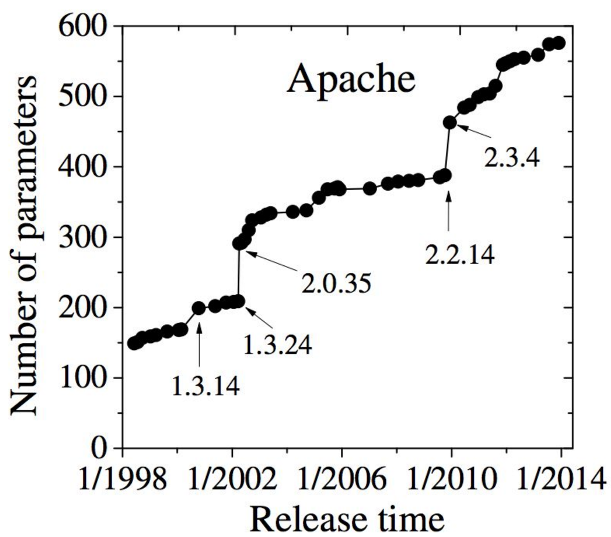
Performance: Throughput
 # Configurations: 6

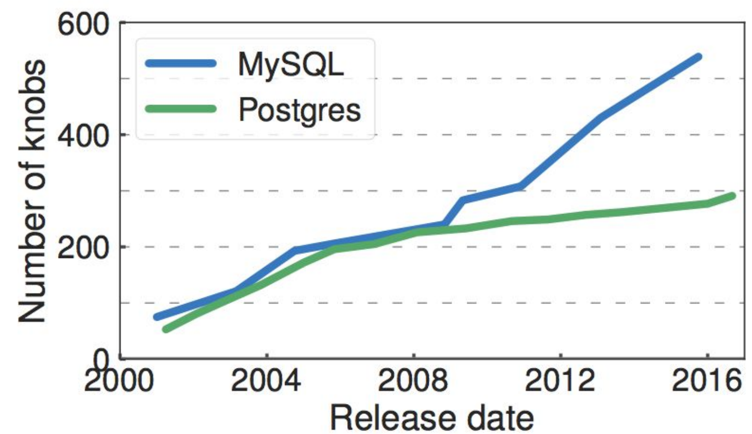


Best configuration is 480 times better than **Worst** configuration

Performance Optimization is getting more Complex!

Necessary





200 new configuration options added to Apache HTTP server between 2010 and 2014

250 new configuration options added to MySQL between 2012 and 2016

[1] Xu et. al. 2015. Hey, you have given me too many knobs!: understanding and dealing with over-designed configuration in system software.FSE 2015
[2] Van Aken, Dana, et al. "Automatic Database Management System Tuning Through Large-scale Machine Learning." *International Conference on Management of Data*. ACM, 2017.

Performance Optimization is required since Default Configuration is Bad!

- ✓ Necessary
- ✓ Complex

Default **MySQL** configuration in 2016 assumes that machine has only 160 MB of RAM[1]

Rule-of-thumb settings for WordCount (in **Hadoop**) gave one of its worst execution times[2]

[1] Van Aken, Dana, et al. "Automatic Database Management System Tuning Through Large-scale Machine Learning." *International Conference on Management of Data*. ACM, 2017.
[2] Herodotou, Herodotos, et al. "Starfish: A Self-tuning System for Big Data Analytics." *CIDR*

Performance Optimization can be Expensive!

- Necessary
- Complex
- Default is bad

- Evaluation of single instance of software/hardware co-design problem can take weeks[1]
- Rolling Sort use-case required 21 days, within a total experimental time of about 2.5 months[2]
- Test suite generation using Evolutionary Algorithm can take weeks[3]
- Image recognition workload and speech recognition workload, jobs ran for many hours or days[4]

[1] Zuluaga, Marcela, et al. "Active learning for multi-objective optimization." *International Conference on Machine Learning*. 2013.
[2] Jamshidi, Pooyan, and Giuliano Casale. "An uncertainty-aware approach to optimal configuration of stream processing systems."*MASCOTS-2016*
[3] Wang, Tiantian, et al. "Searching for better configurations: a rigorous approach to clone evaluation." *FSE-2013*
[4] Venkataraman, Shivaram, et al. "Ernest: Efficient Performance Prediction for Large-Scale Advanced Analytics." *NSDI*. 2016.

Is it pervasive?

- ✓ Necessary
- ✓ Complex
- ✓ Default is bad
- ✓ Expensive

Cloud Computing
- Ernest
- Cherrypick
- PARIS

Database
- otter-tune
- ituned

Machine Learning
- Hyperparameter Tuning
- Random search
- SMBO
- Fabolas

Software Engineering
- Tuning or Default Values?
- Tuning for Software Analytics
- Tuning for Defect Prediction
- Topic Modelling

Performance Optimization!

- Necessary
- Complex
- Default is bad
- Expensive
- Pervasive

Performance Optimization!

- Necessary
- Complex
- Default is bad
- Expensive
- Pervasive

SIGOPT

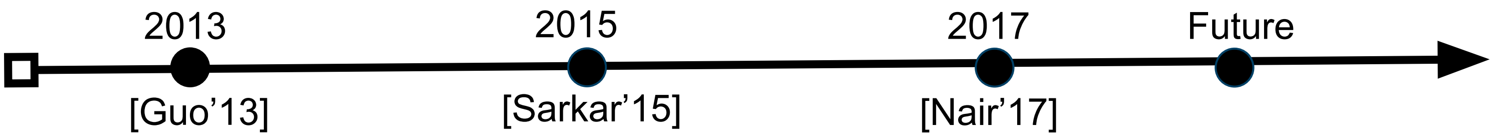
Quid

WHETLAB

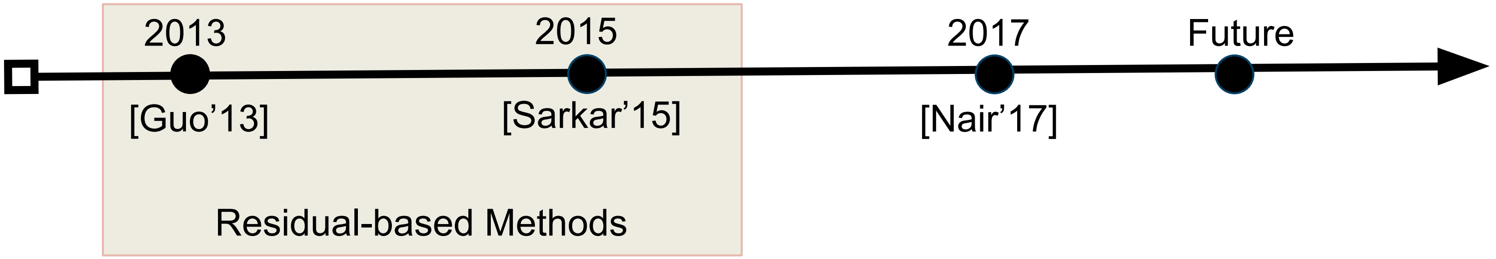
DataRobot

PETUUM

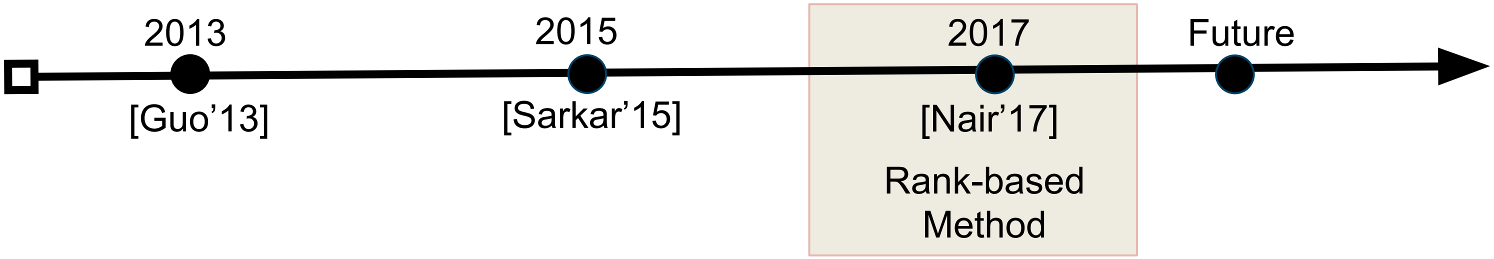
Road Map



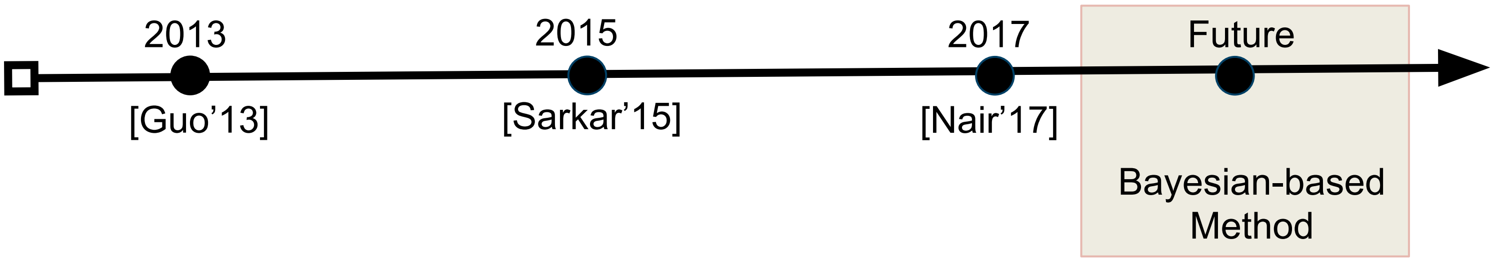
Road Map


2013
2015
2017
Future
[Guo'13]
[Sarkar'15]
[Nair'17]
Residual-based Methods

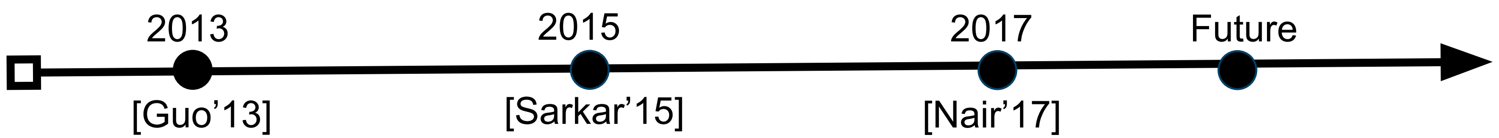
Road Map


2013
[Guo'13]
2015
[Sarkar'15]
2017
[Nair'17]
Rank-based
Method
Future

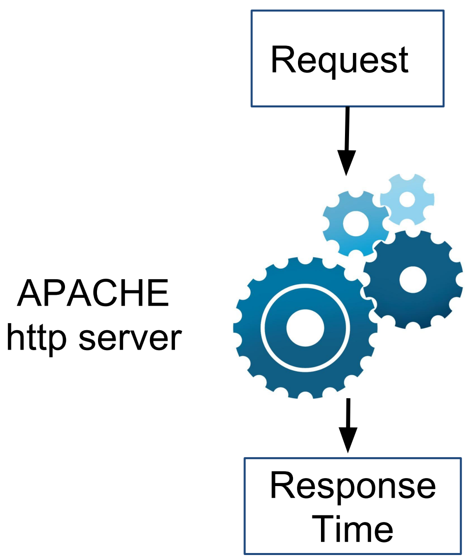
Road Map

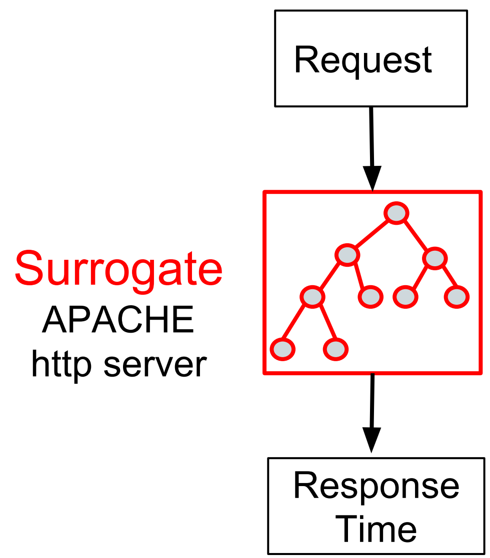

2013
[Guo'13]
2015
[Sarkar'15]
2017
[Nair'17]
Future
Bayesian-based Method

Road Map

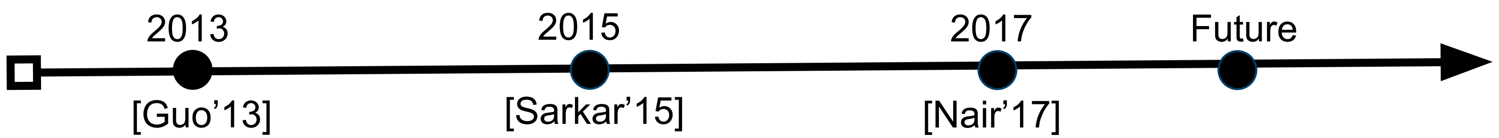


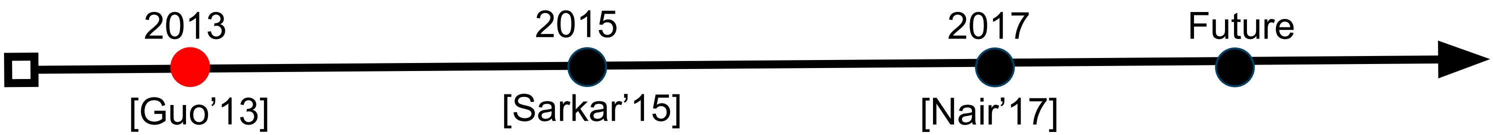
Surrogate is a cheap(er) version of the actual system





Road Map



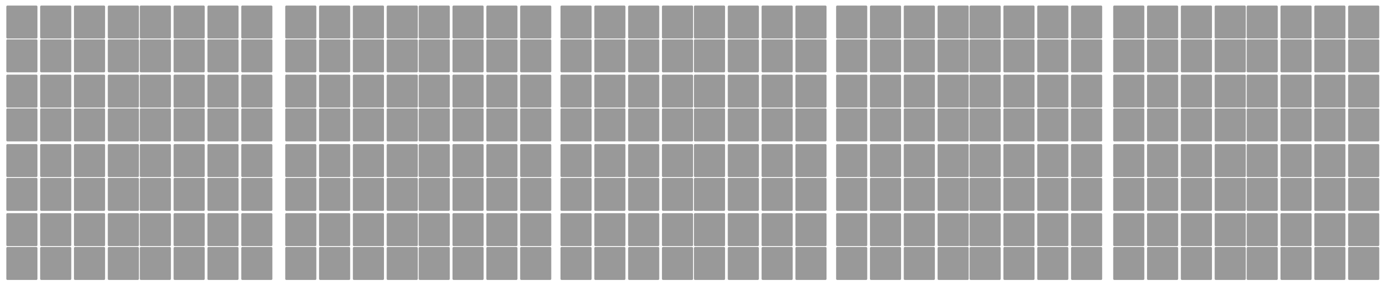


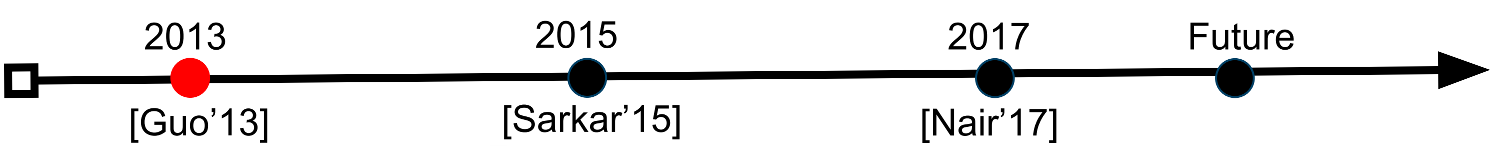
Progressive Sampling

Guo, Jianmei, et al. "Variability-aware performance prediction: A statistical learning approach." *ASE*-2013.

Residual-based Methods
Progressive Sampling

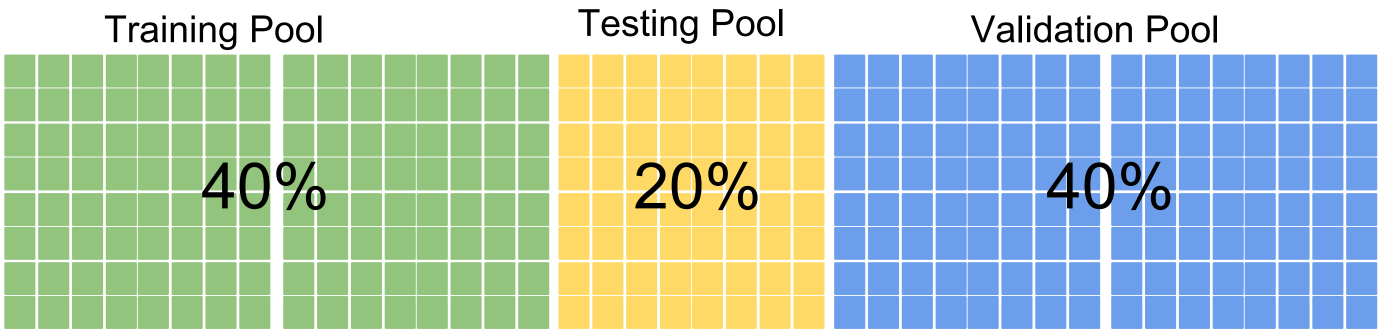
Configuration Space

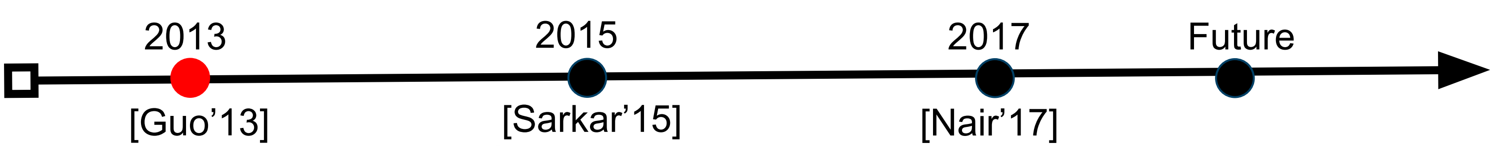




Residual-based Methods
Progressive Sampling

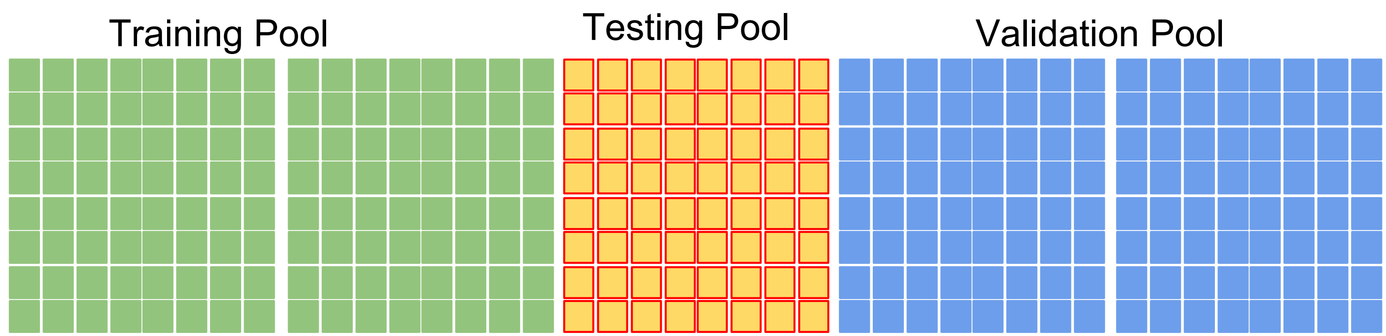
Configuration Space

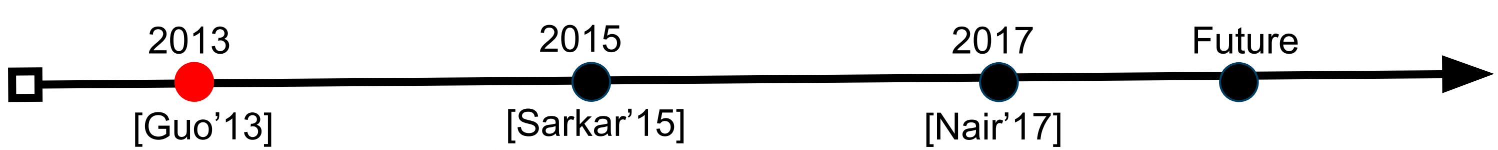




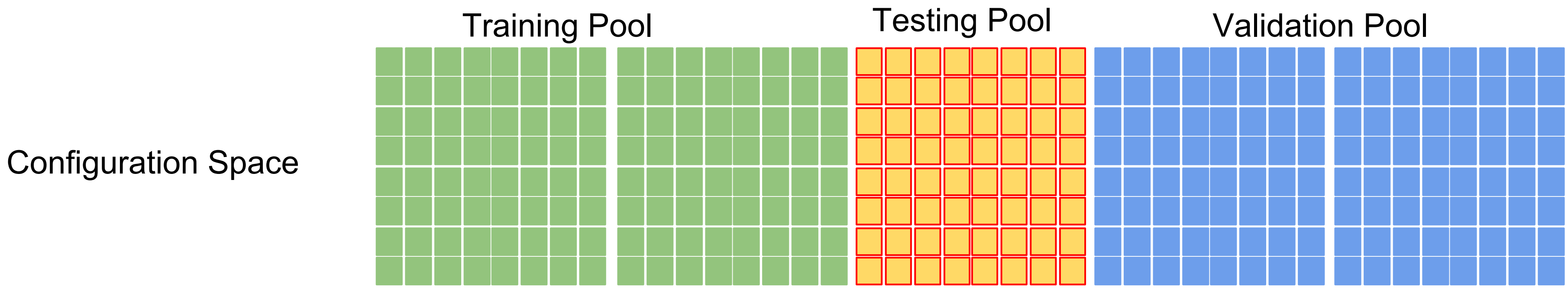
Residual-based Methods
Progressive Sampling

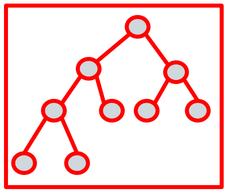
Configuration Space

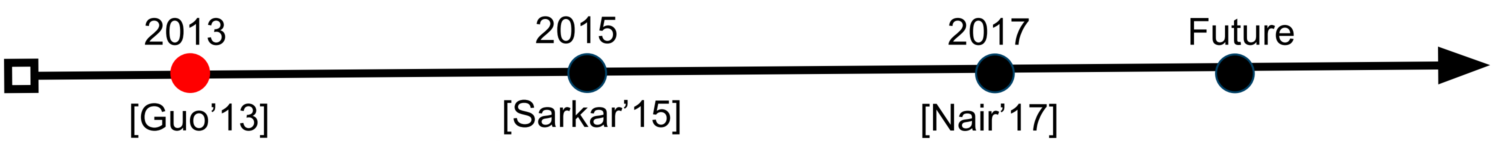




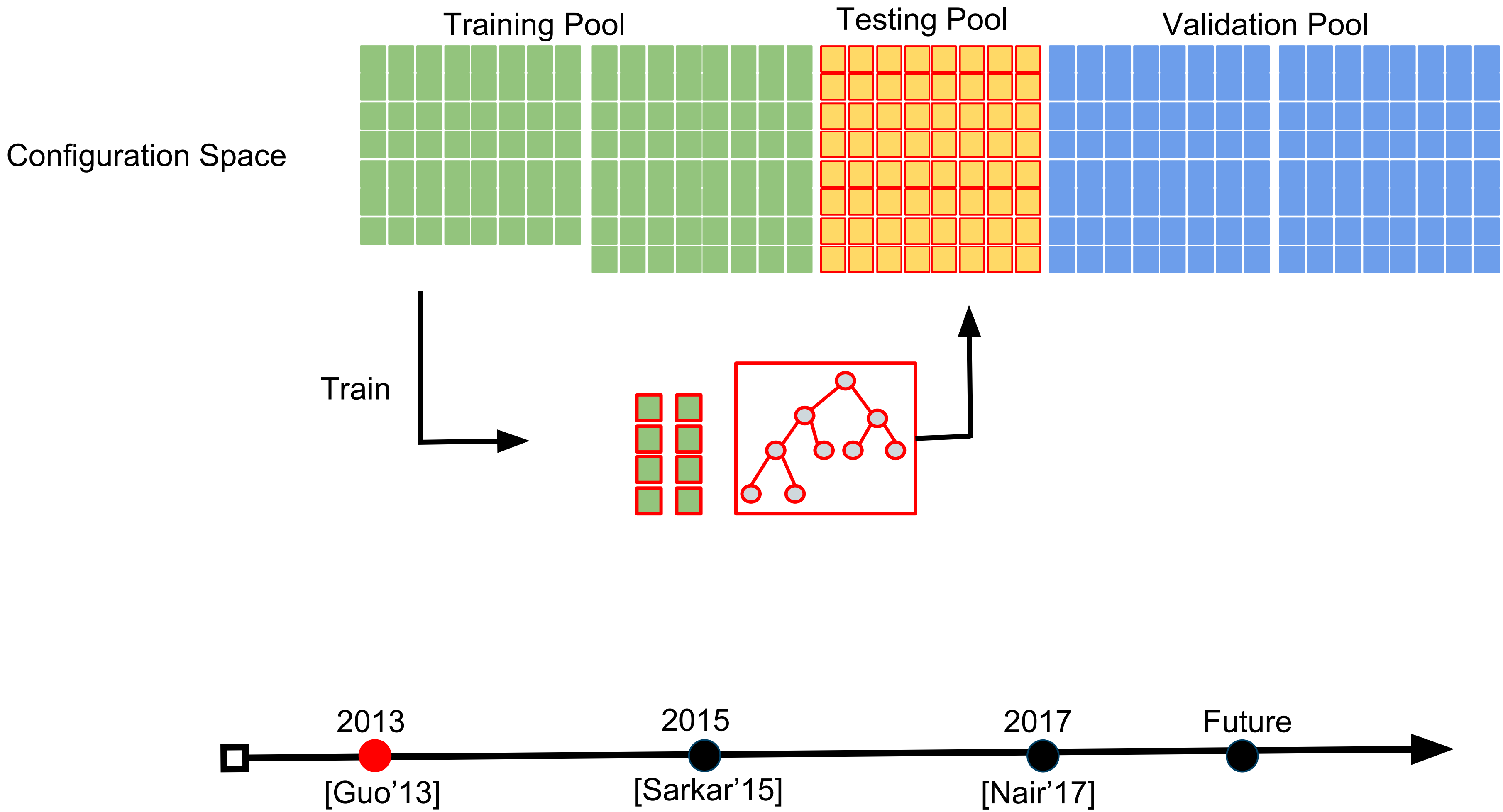
Residual-based Methods
Progressive Sampling



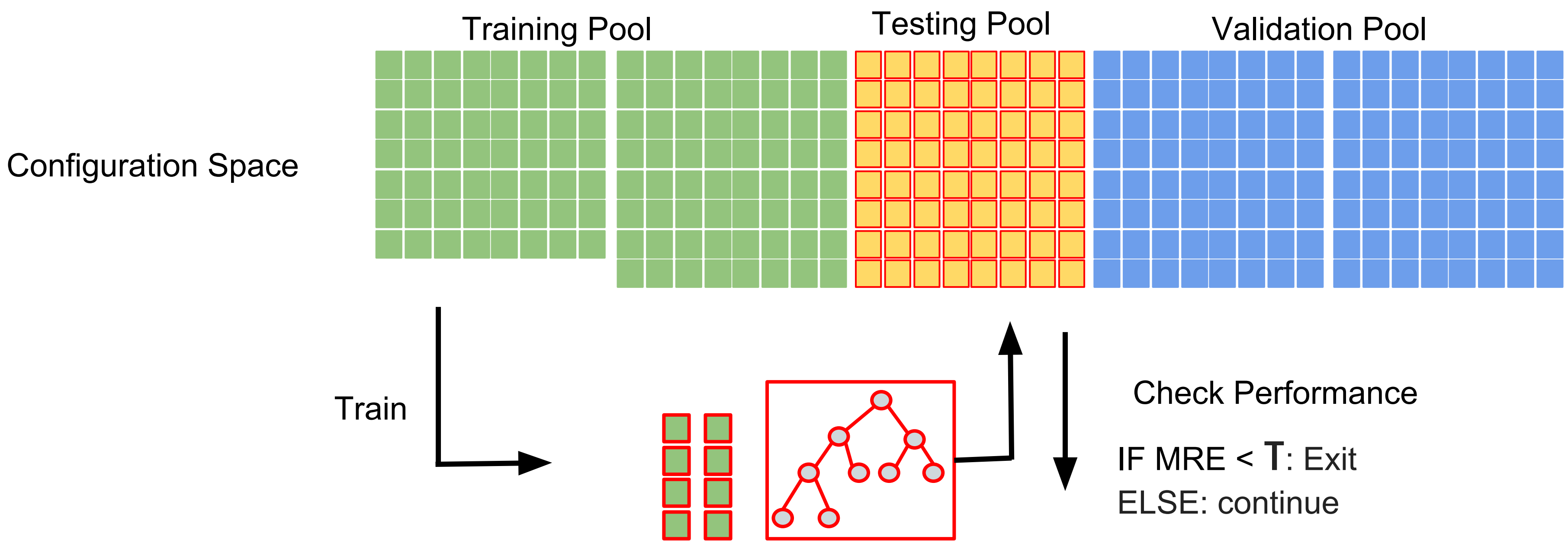


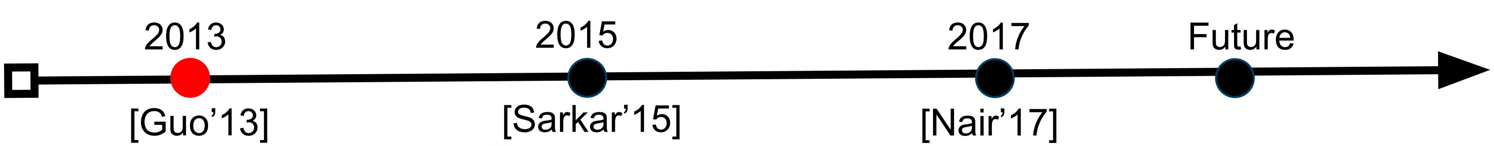


Residual-based Methods
Progressive Sampling

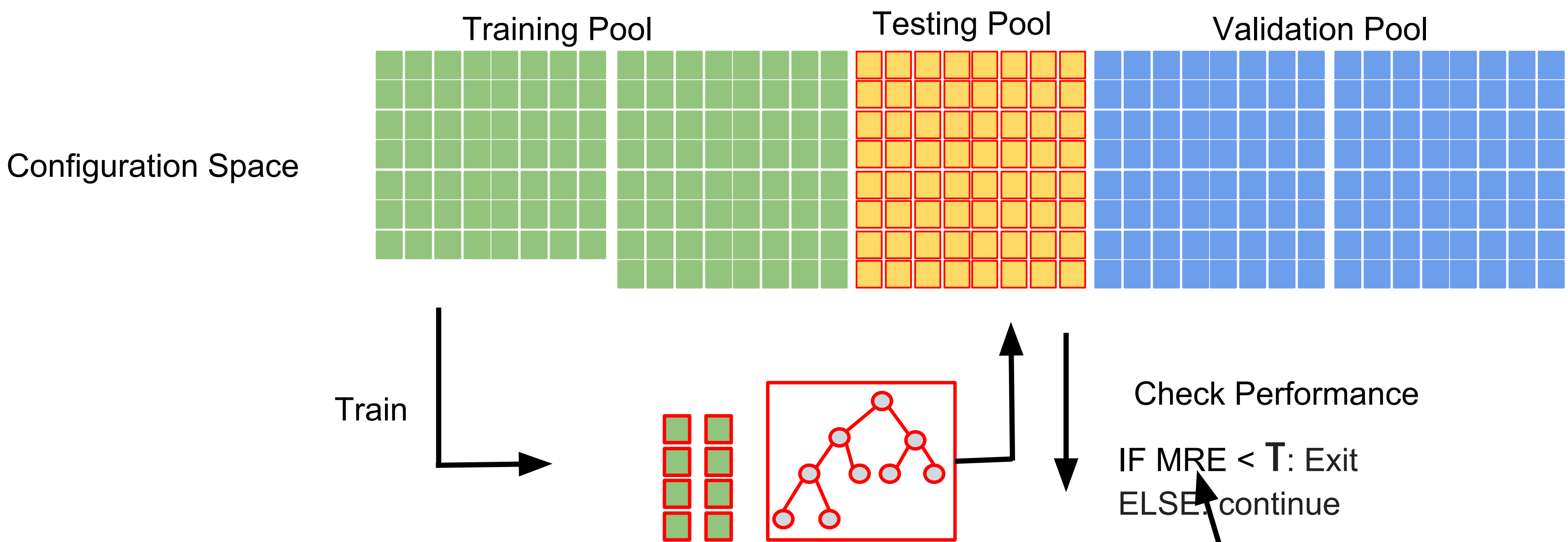


Residual-based Methods
Progressive Sampling





Residual-based Methods
Progressive Sampling

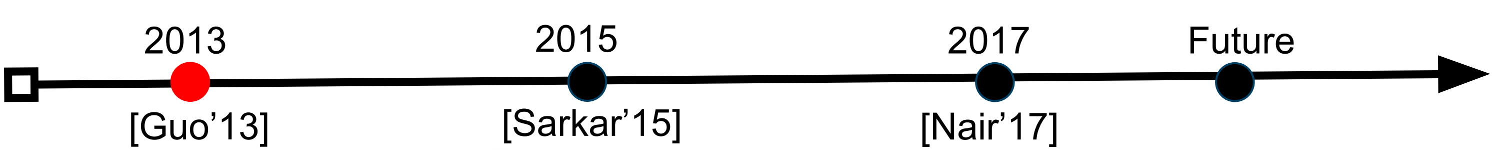


Check Performance

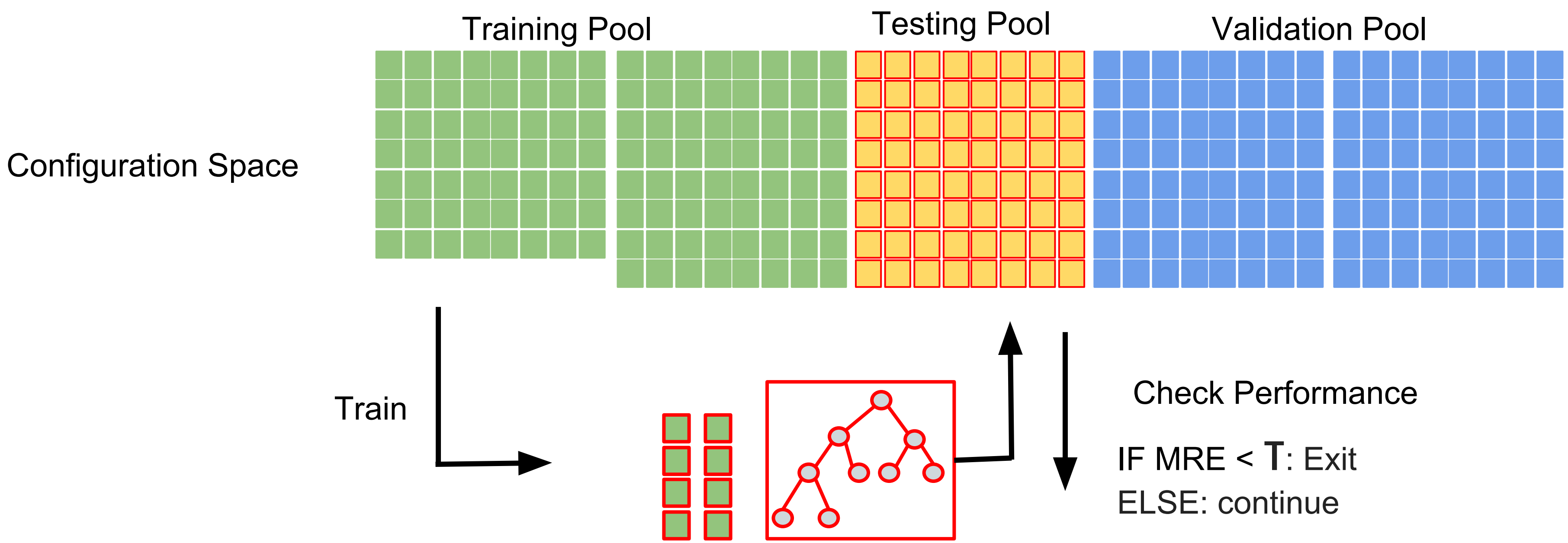
IF MRE < **T**: Exit
ELSE: continue

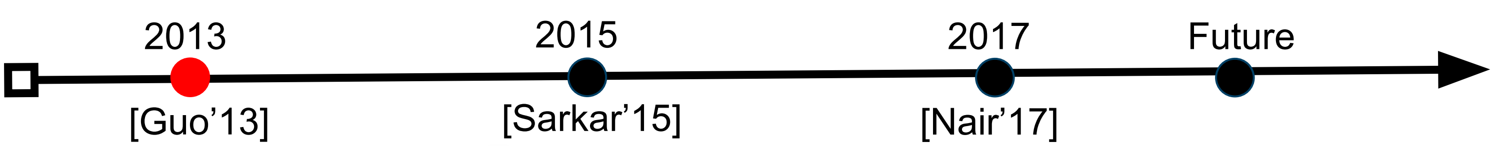
Mean Relative Error

$$MRE = \frac{|actual - predicted|}{actual}$$

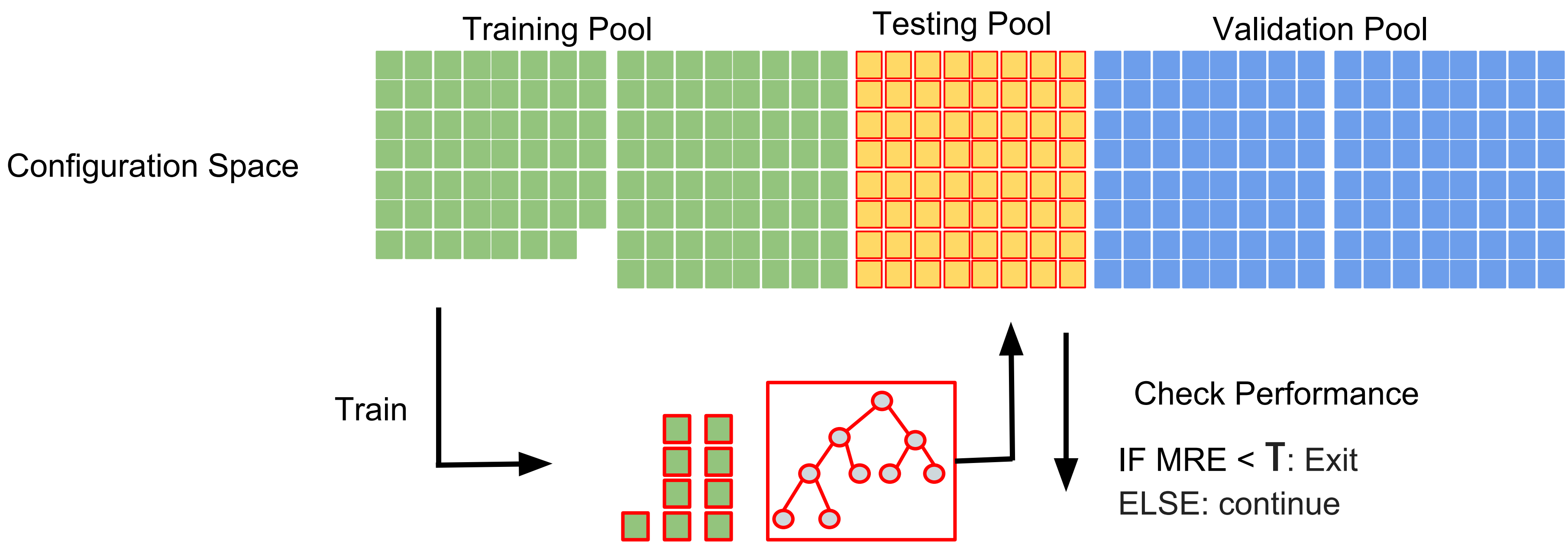


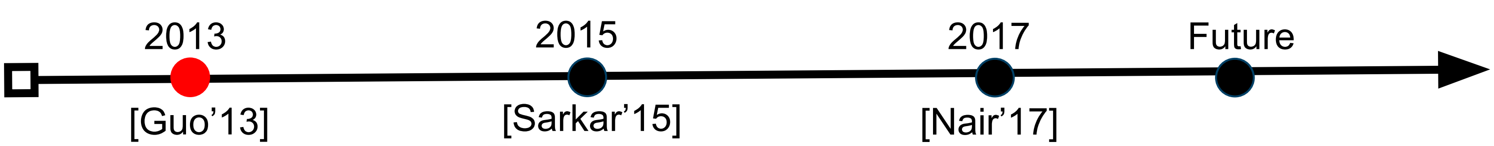
Residual-based Methods
Progressive Sampling



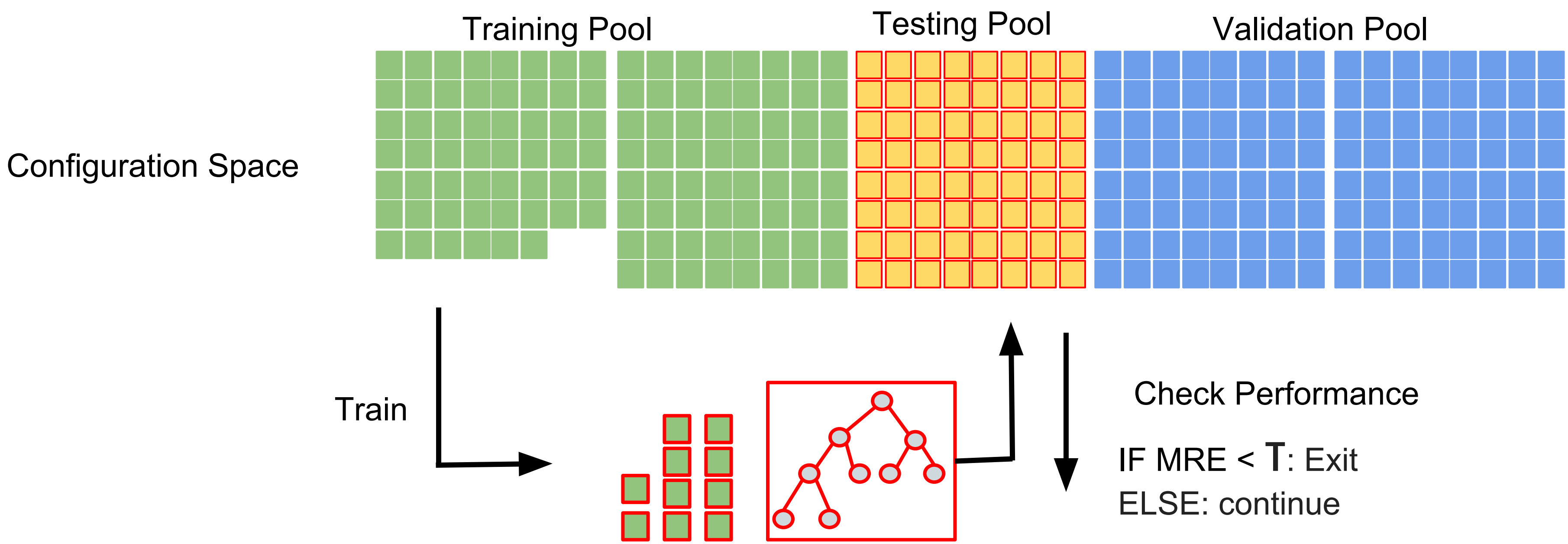


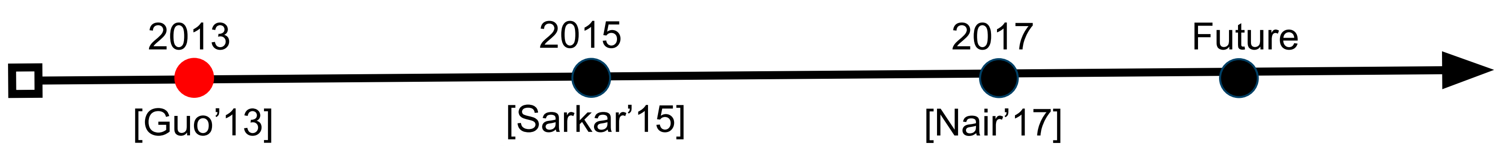
Residual-based Methods
Progressive Sampling



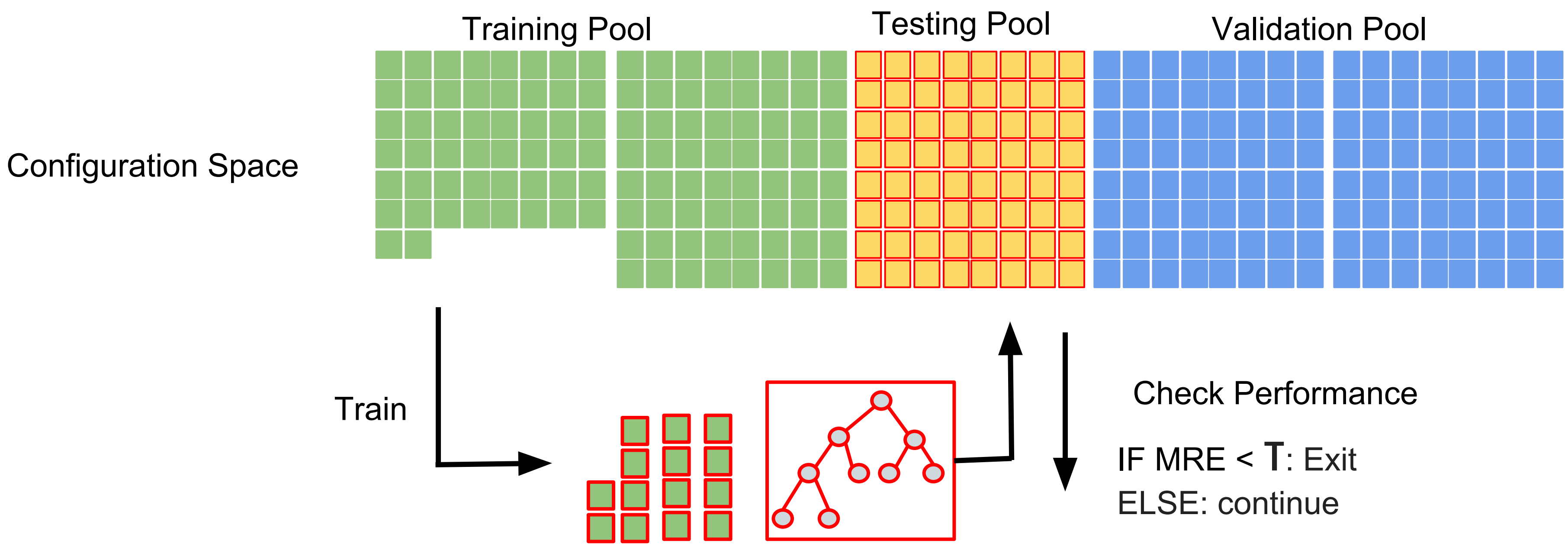


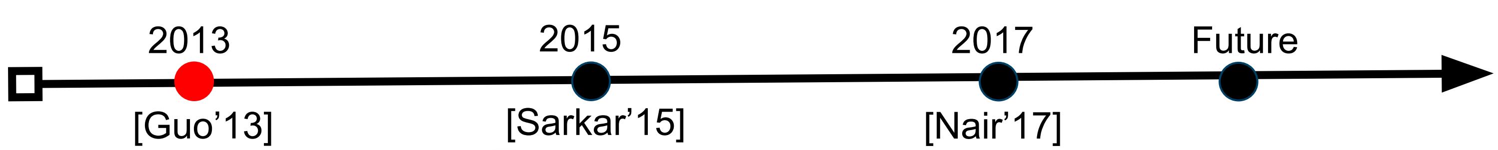
Residual-based Methods
Progressive Sampling



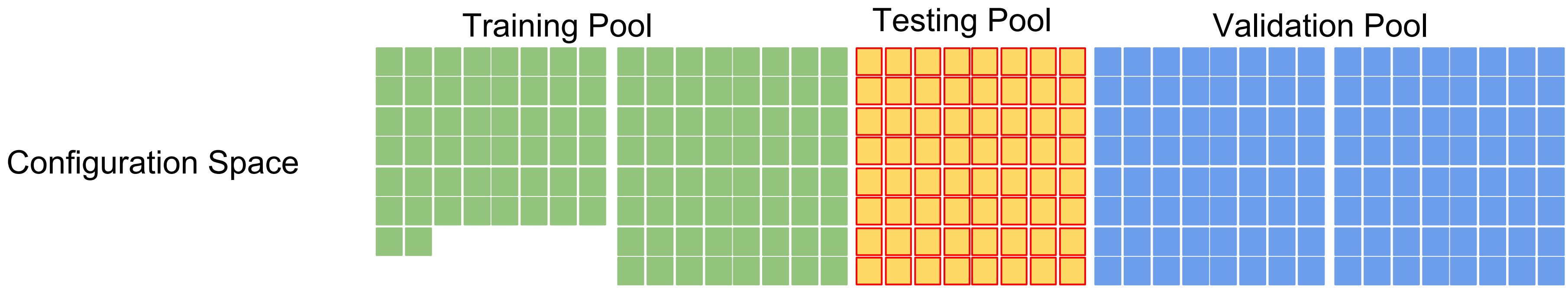


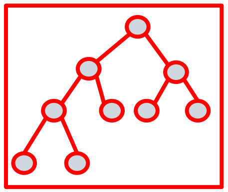
Residual-based Methods
Progressive Sampling

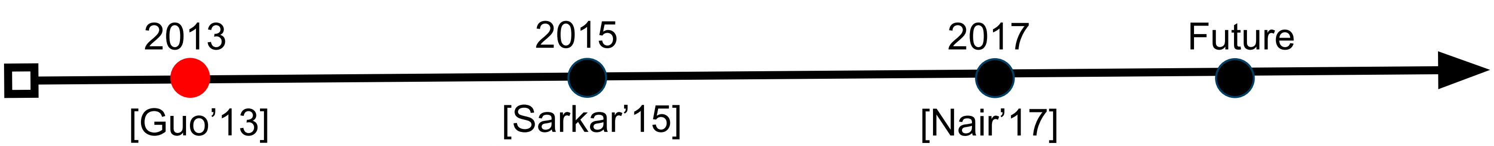




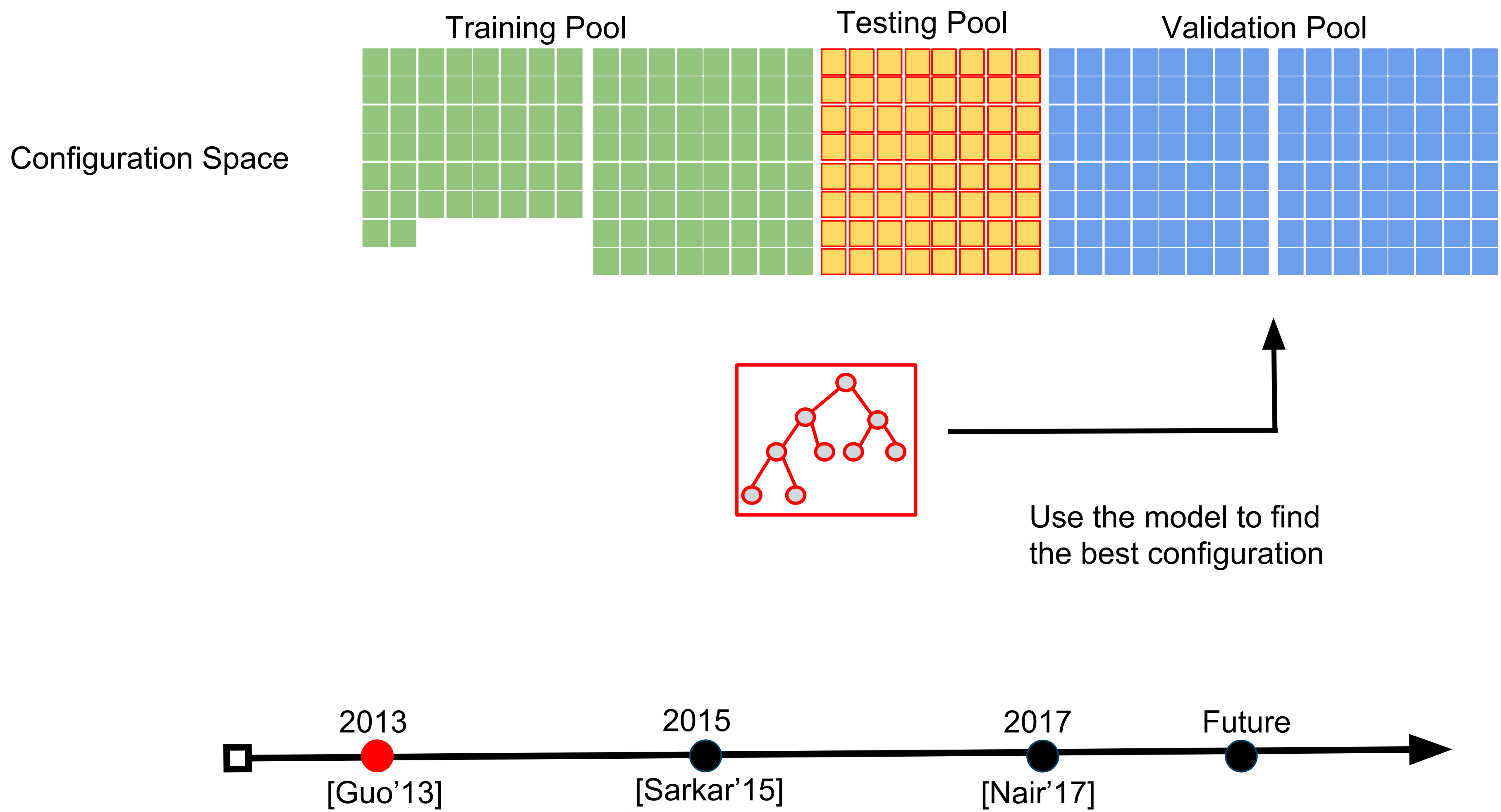
Residual-based Methods
Progressive Sampling





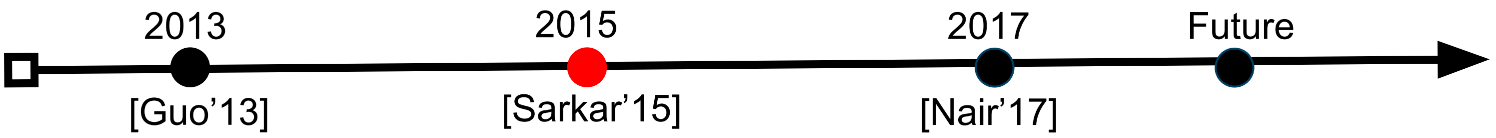


Residual-based Methods
Progressive Sampling



Residual-based Methods
Progressive Sampling - Limitation

- The stopping condition is arbitrary
- Cannot estimate cost required to build a surrogate

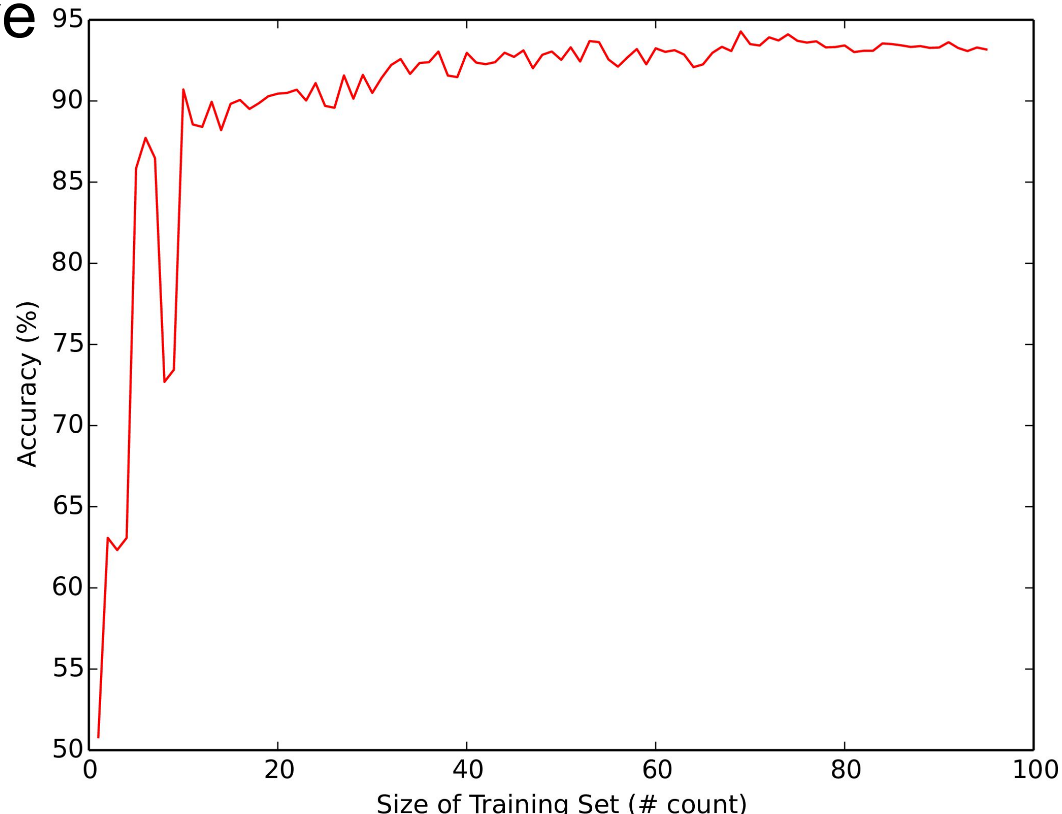


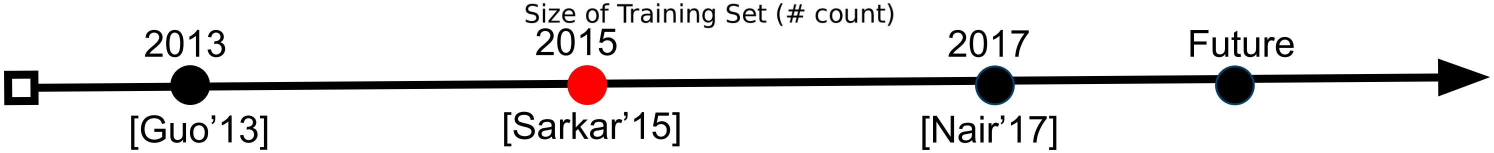
Projective Sampling

Sarkar, Atri, et al. "Cost-efficient sampling for performance prediction of configurable systems." *ASE* 2015.

Residual-based Methods
Projective Sampling

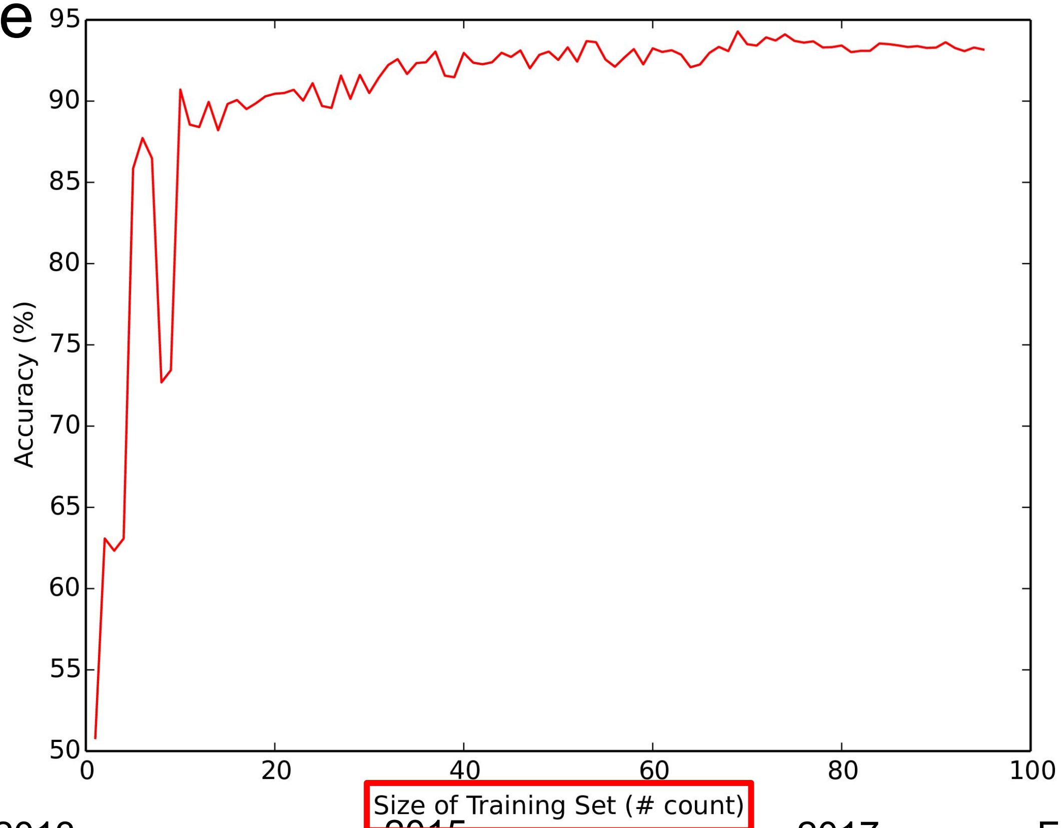
Learning Curve

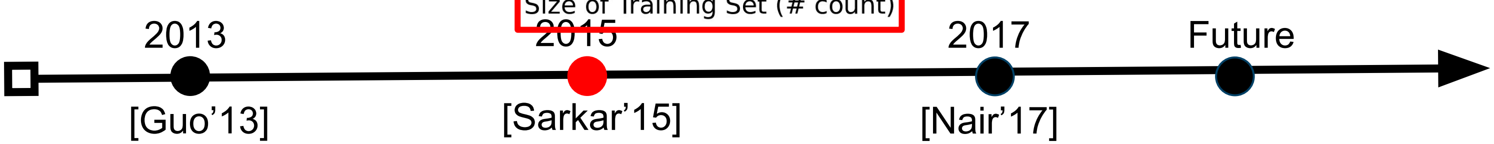




Residual-based Methods
Projective Sampling

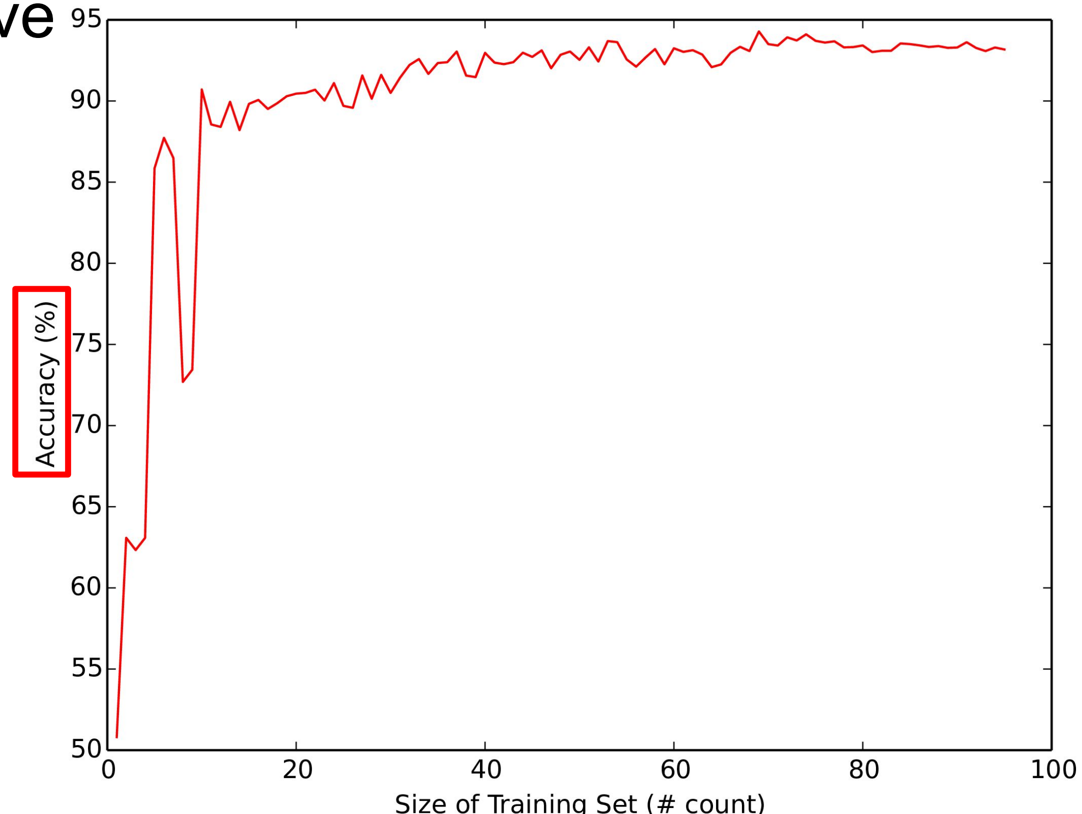
Learning Curve

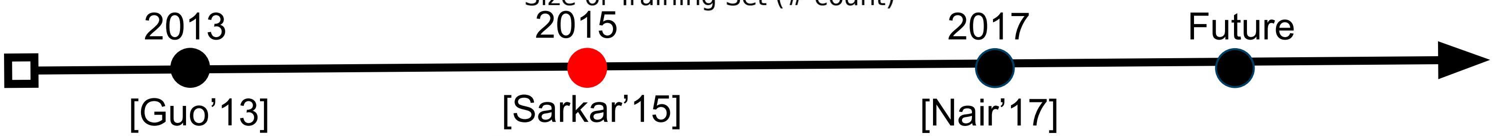




Residual-based Methods
Projective Sampling

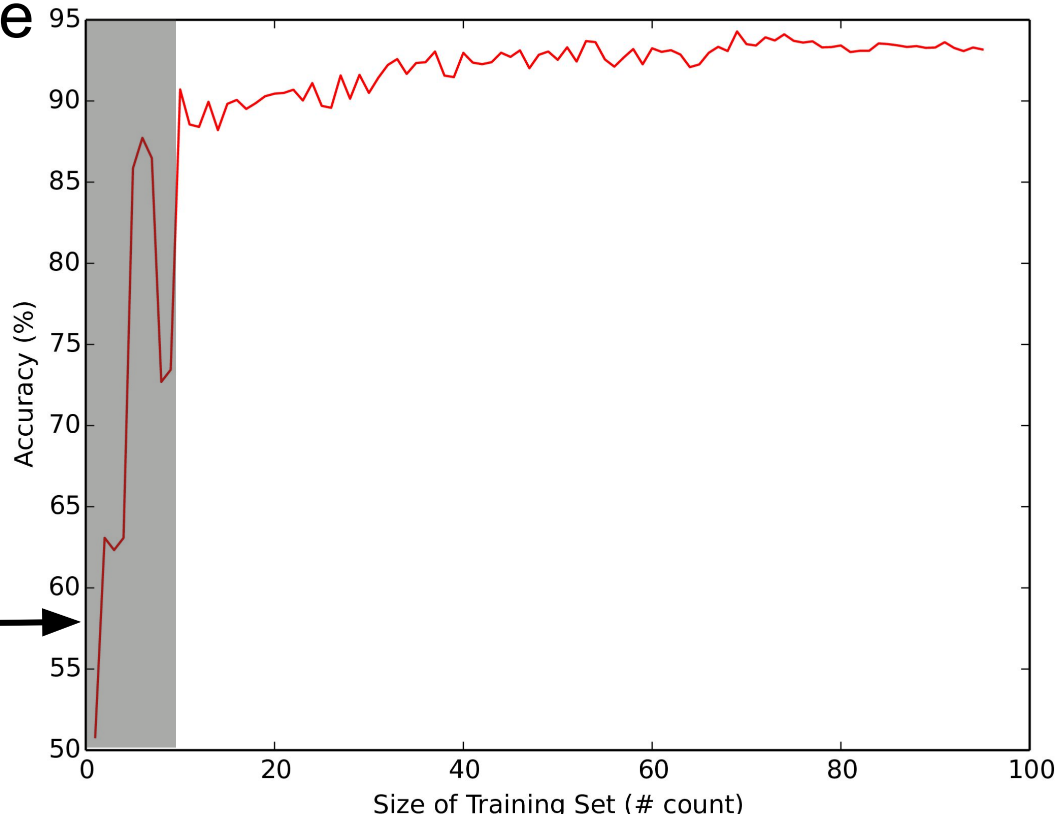
Learning Curve



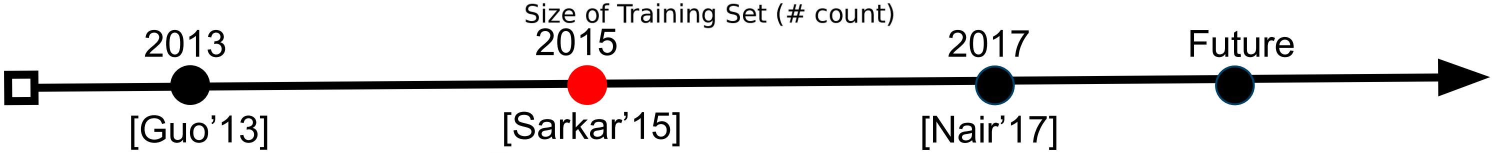


Residual-based Methods
Projective Sampling

Learning Curve

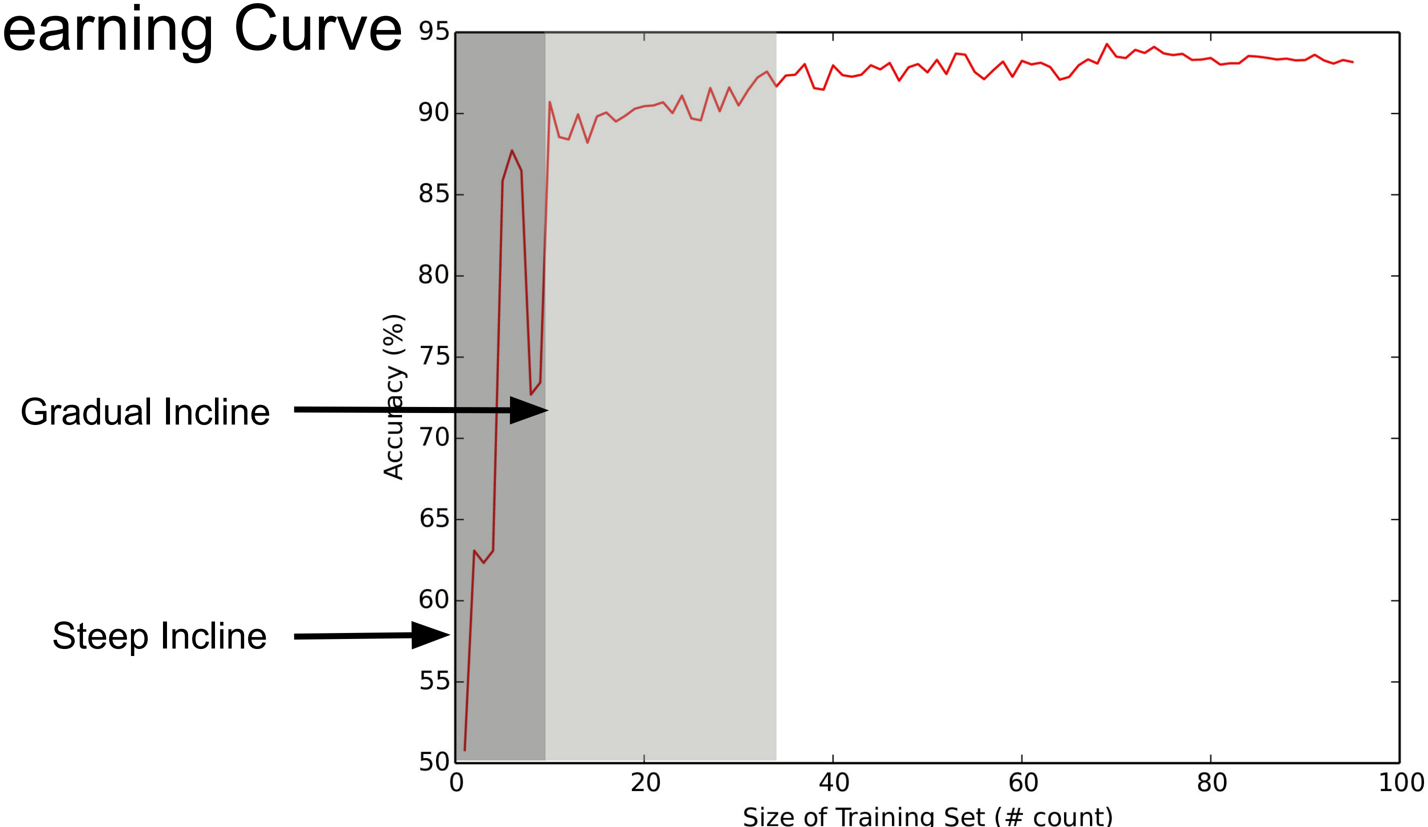


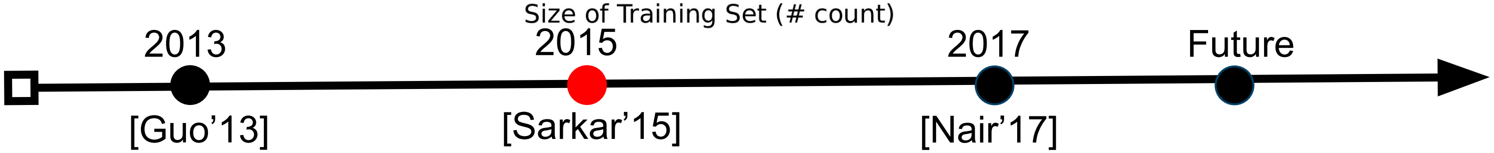
Steep Incline



Residual-based Methods
Projective Sampling

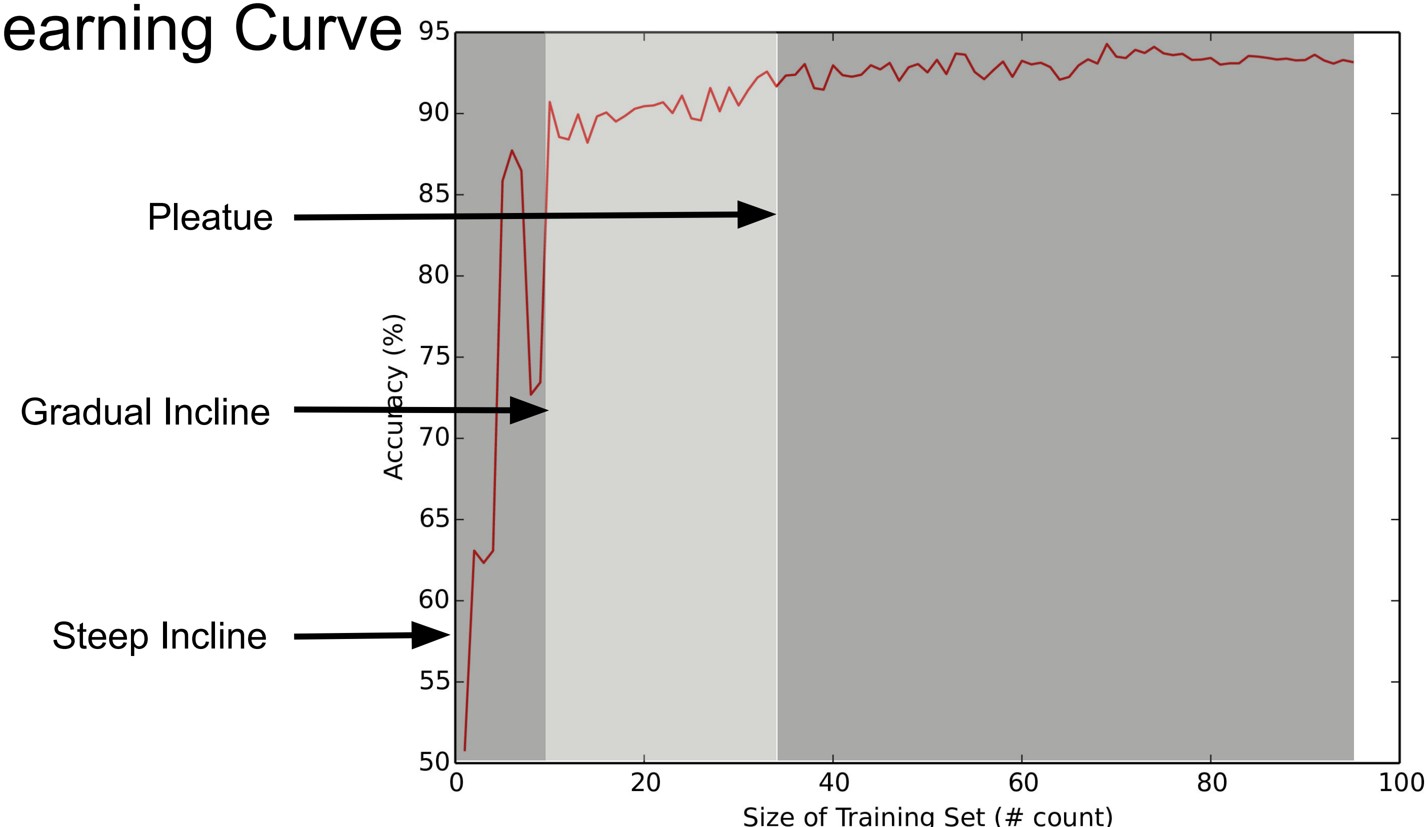
Learning Curve

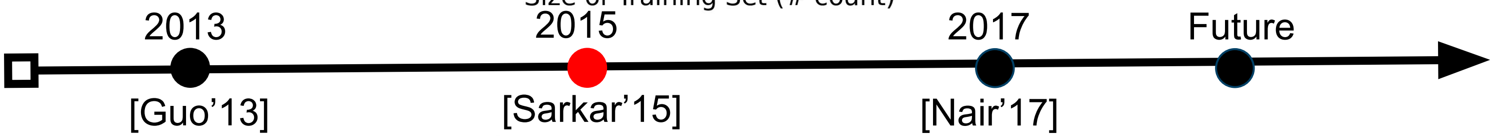




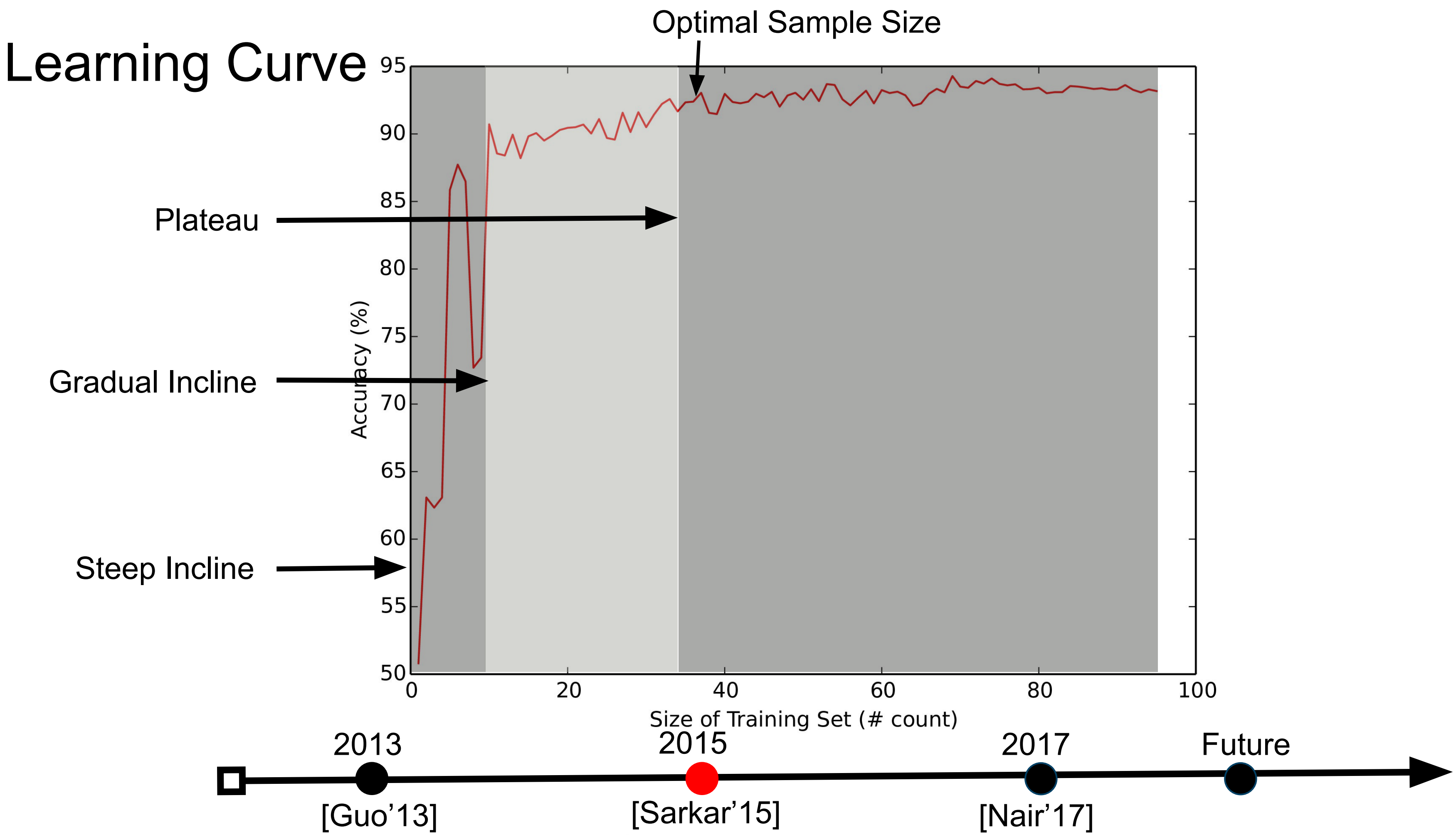
Residual-based Methods
Projective Sampling

Learning Curve





Residual-based Methods
Projective Sampling



Residual-based Methods
Projective Sampling

Estimates the Learning Curve

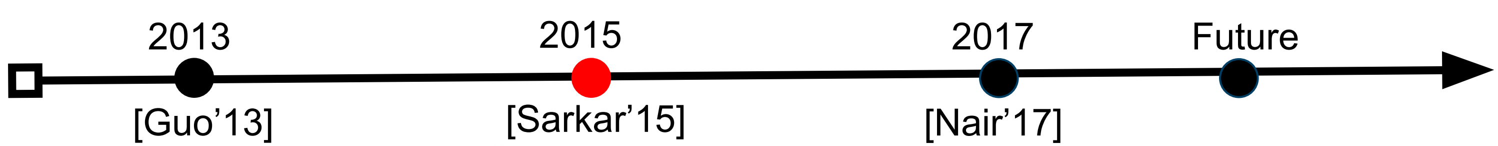
Generate initial data points for the learning curve → Find best fit projective function → Calculate optimal training set → Build surrogate

2013 [Guo'13] — 2015 [Sarkar'15] — 2017 [Nair'17] — Future

Residual-based Methods
Projective Sampling

Estimates the Learning Curve

Generate initial data points for the learning curve → Find best fit projective function → Calculate optimal training set → Build surrogate



Residual-based Methods
Projective Sampling

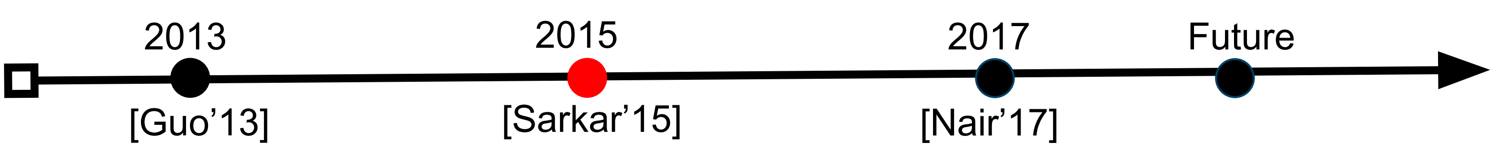
Estimates the Learning Curve

Generate initial data points for the learning curve → Find best fit projective function → Calculate optimal training set → Build surrogate

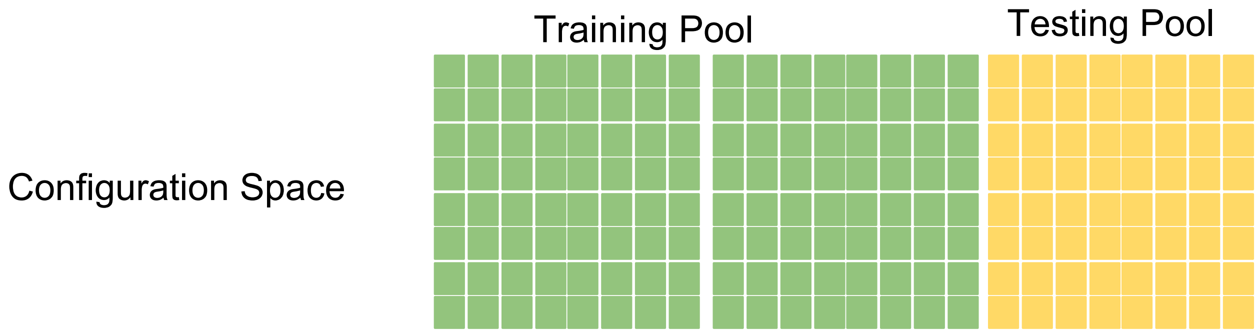
Requirement:
Initial samples should reflect relationship between all configuration options

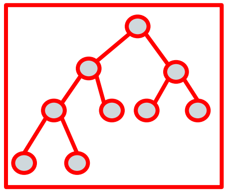
Intuition:
Performance depends if configuration option is selected or deselected

Heuristic:
Feature Frequency - initial samples have each option selected or deselected, at least, δ times

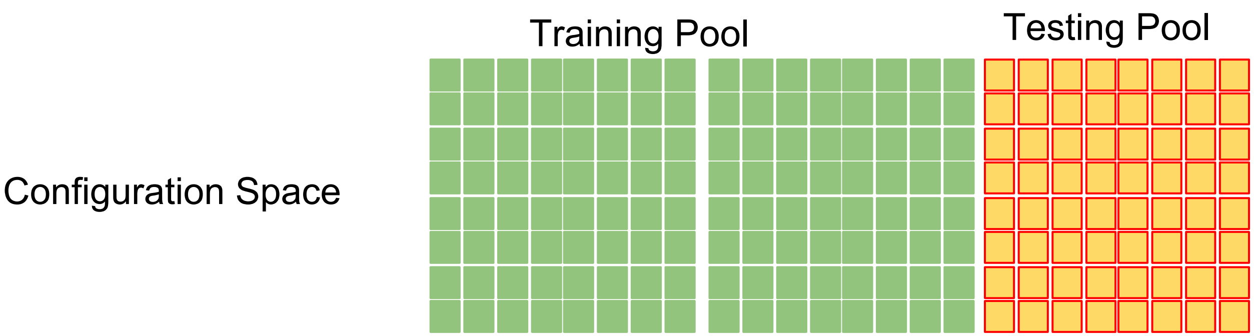


Residual-based Methods
Projective Sampling

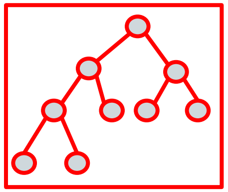




Residual-based Methods
Projective Sampling

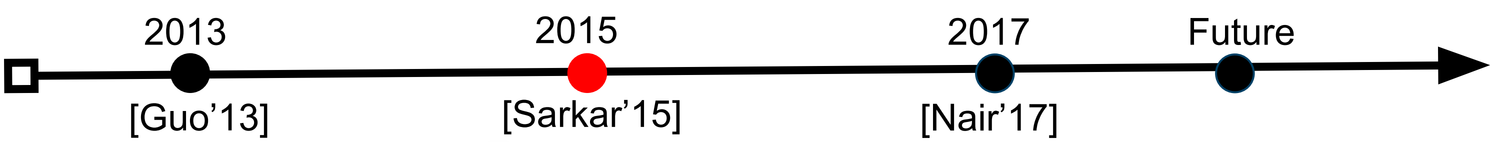


#Samples	Accuracy



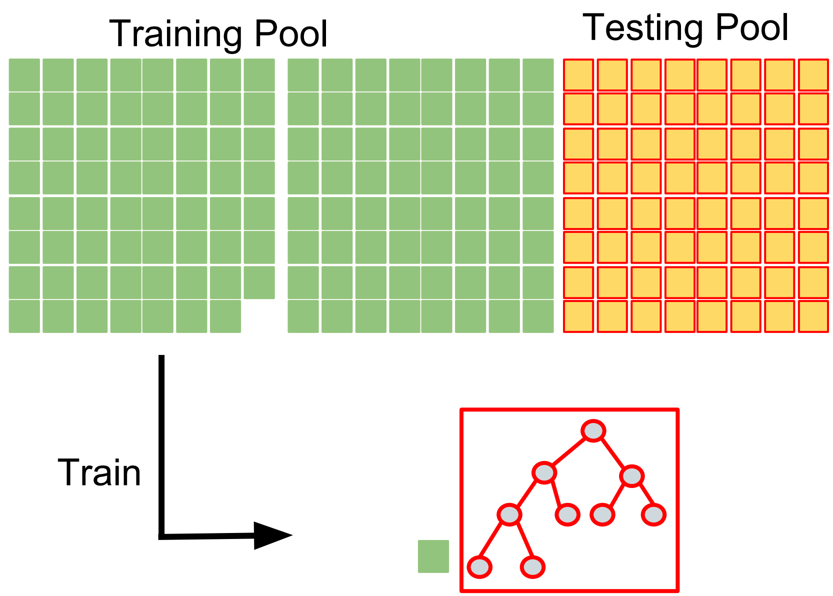
Feature frequency table (δ=2)

	c_1	c_2	c_3	c_4
Selected	0	0	0	0
Deselected	0	0	0	0



Residual-based Methods
Projective Sampling

Configuration Space

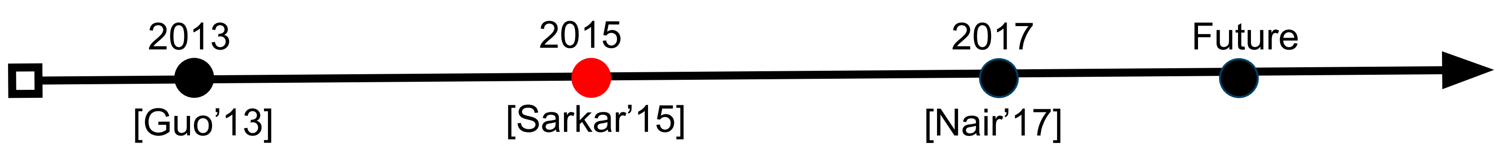


#Samples	Accuracy

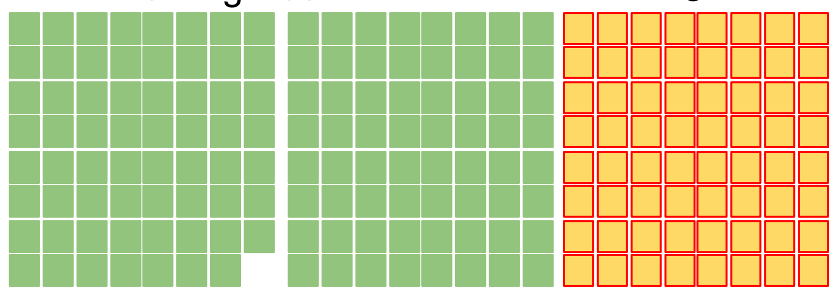
c_1	c_2	c_3	c_4
1	0	1	1

Feature frequency table (δ=2)

	c_1	c_2	c_3	c_4
Selected	0	0	0	0
Deselected	0	0	0	0



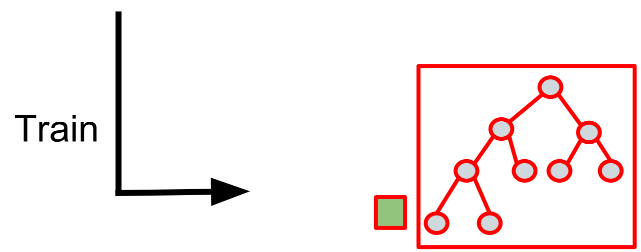
Residual-based Methods
Projective Sampling



#Samples	Accuracy
1	5%

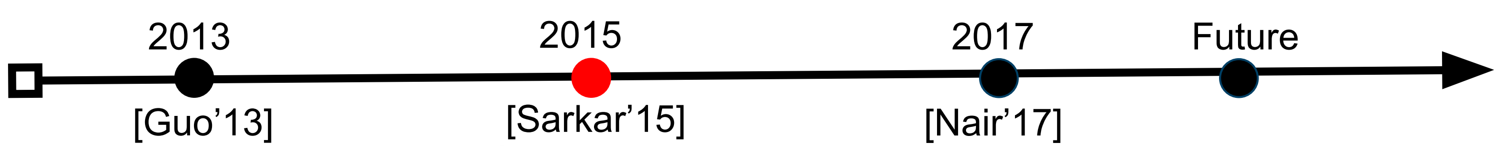
Configuration Space

c_1	c_2	c_3	c_4
1	0	1	1



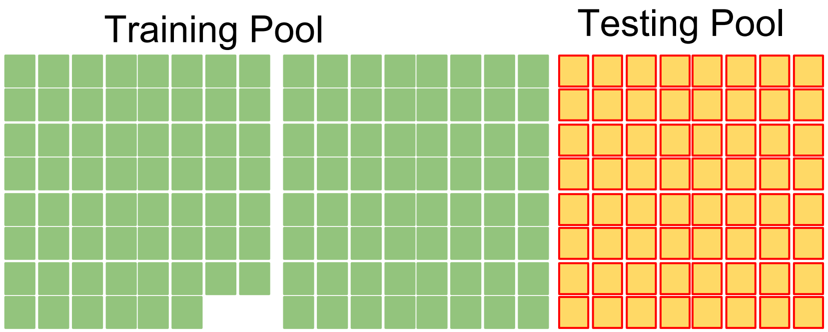
Feature frequency table (δ=2)

	c_1	c_2	c_3	c_4
Selected	1	0	1	1
Deselected	0	1	0	0

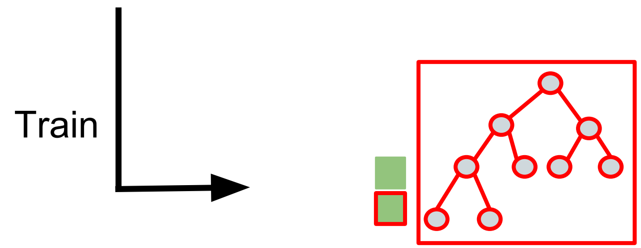


Residual-based Methods
Projective Sampling

Configuration Space



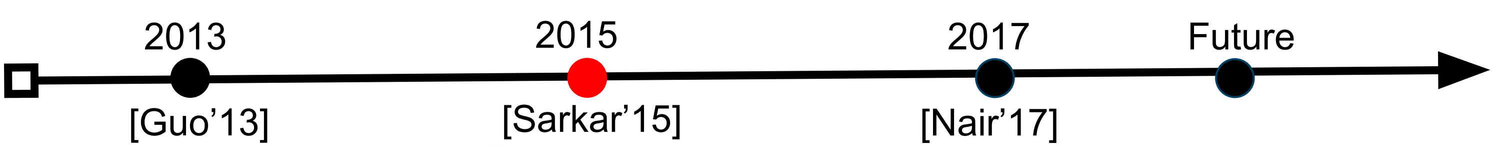
c_1	c_2	c_3	c_4
1	0	1	1
0	1	0	0



#Samples	Accuracy
1	5%

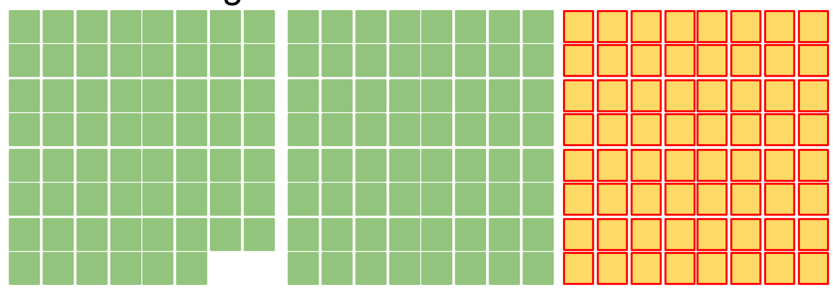
Feature frequency table (δ=2)

	c_1	c_2	c_3	c_4
Selected	1	0	1	1
Deselected	0	1	0	0



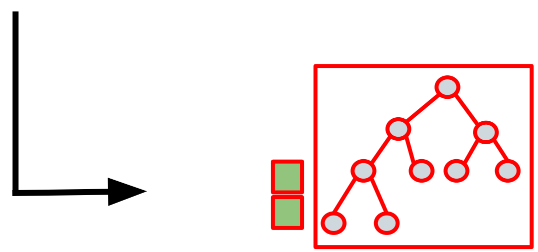
Residual-based Methods
Projective Sampling

Configuration Space



#Samples	Accuracy
1	5%
2	17%

c_1	c_2	c_3	c_4
1	0	1	1
0	1	1	0

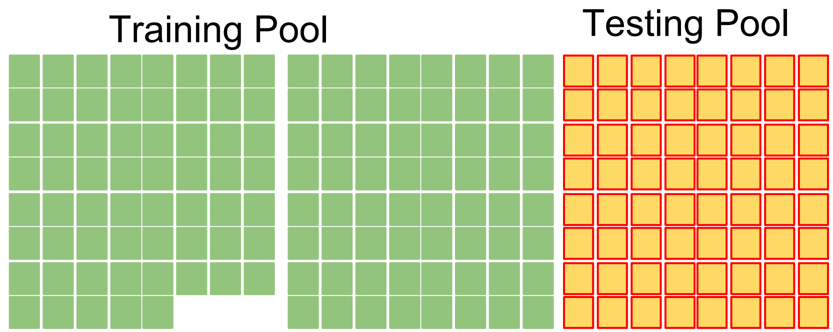


Feature frequency table (δ=2)

	c_1	c_2	c_3	c_4
Selected	1	1	2	1
Deselected	1	1	0	1

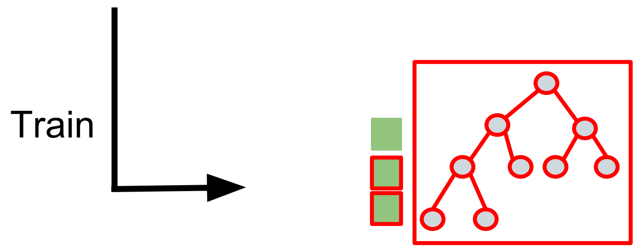
Residual-based Methods
Projective Sampling

Configuration Space



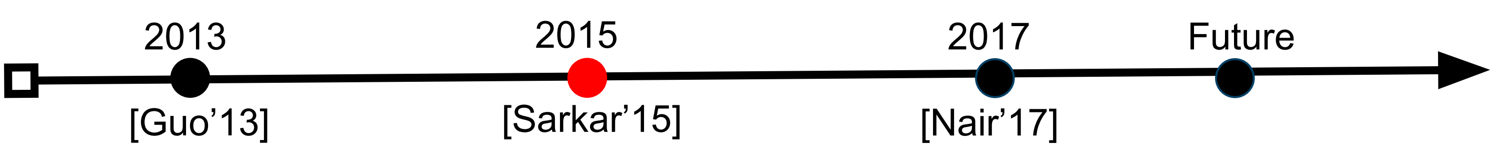
#Samples	Accuracy
1	5%
2	17%

c_1	c_2	c_3	c_4
1	0	1	1
0	1	1	0
1	1	0	0



Feature frequency table (δ=2)

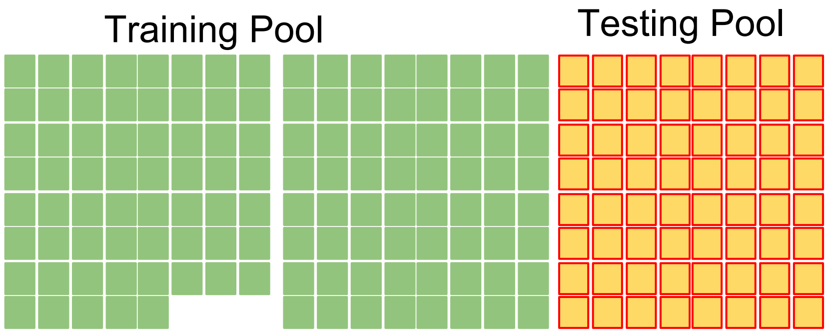
	c_1	c_2	c_3	c_4
Selected	1	1	2	1
Deselected	1	1	0	1



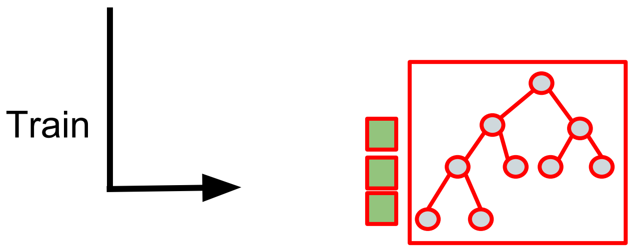
Residual-based Methods
Projective Sampling

Configuration Space

c_1	c_2	c_3	c_4
1	0	1	1
0	1	1	0
1	1	0	0

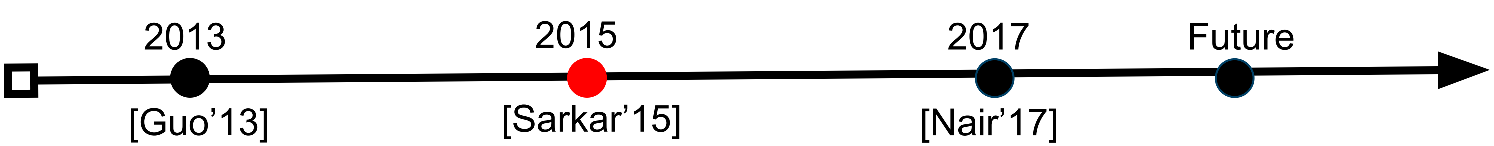


#Samples	Accuracy
1	5%
2	17%
3	29%



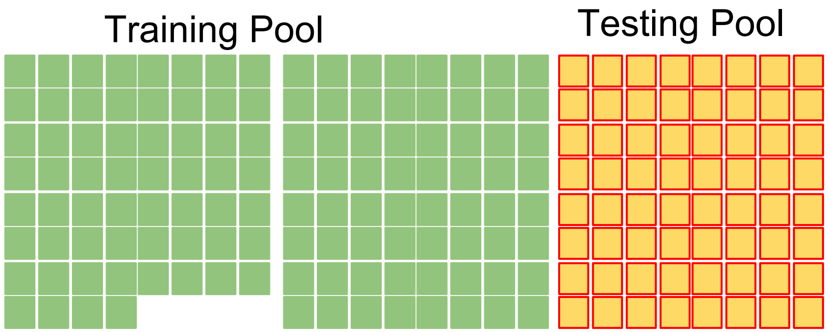
Feature frequency table (δ=2)

	c_1	c_2	c_3	c_4
Selected	2	2	2	1
Deselected	1	1	1	2



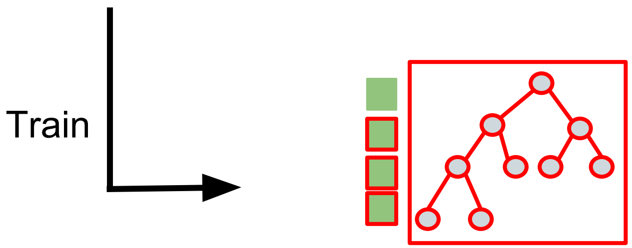
Residual-based Methods
Projective Sampling

Configuration Space



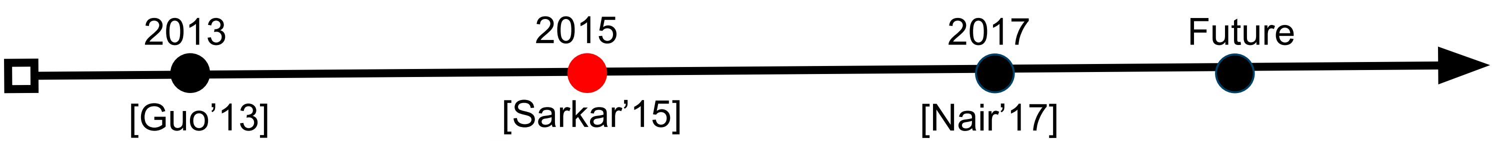
#Samples	Accuracy
1	5%
2	17%
3	29%

c_1	c_2	c_3	c_4
1	0	1	1
0	1	1	0
1	1	0	0
0	0	0	1



Feature frequency table (δ=2)

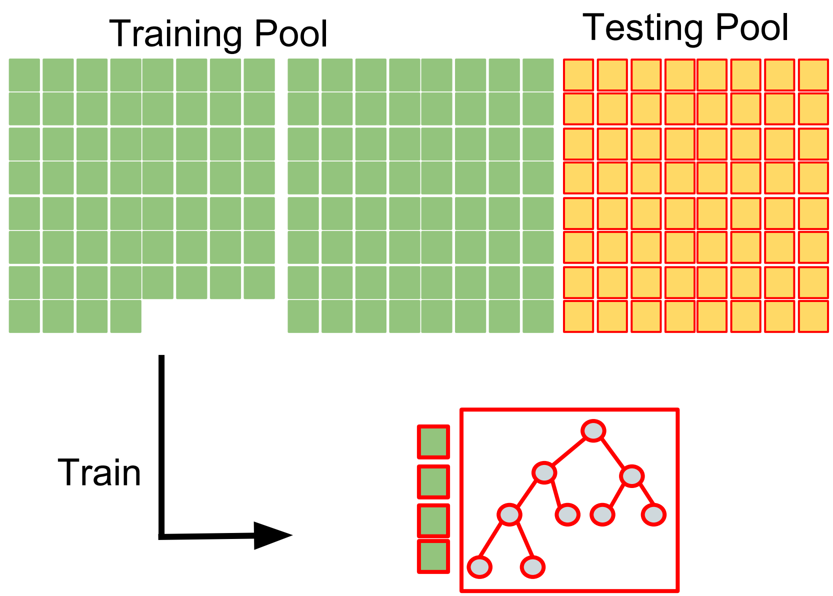
	c_1	c_2	c_3	c_4
Selected	2	2	2	1
Deselected	1	1	1	2



Residual-based Methods
Projective Sampling

Configuration Space

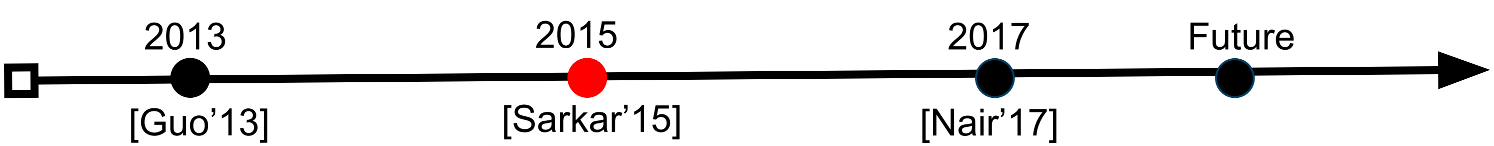
c_1	c_2	c_3	c_4
1	0	1	1
0	1	1	0
1	1	0	0
0	0	0	1



#Samples	Accuracy
1	5%
2	17%
3	29%
4	35%

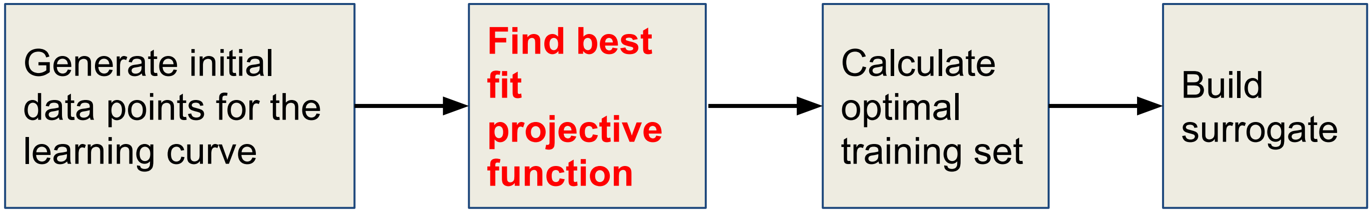
Feature frequency table (δ=2)

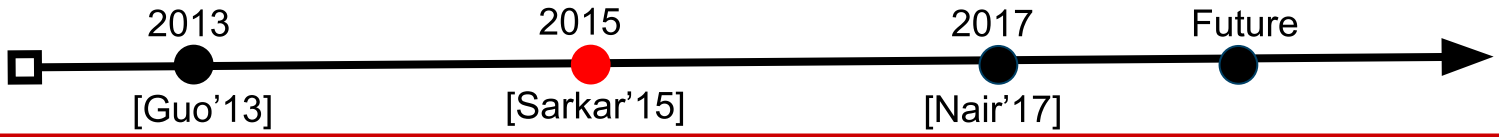
	c_1	c_2	c_3	c_4
Selected	2	2	2	2
Deselected	2	2	2	2



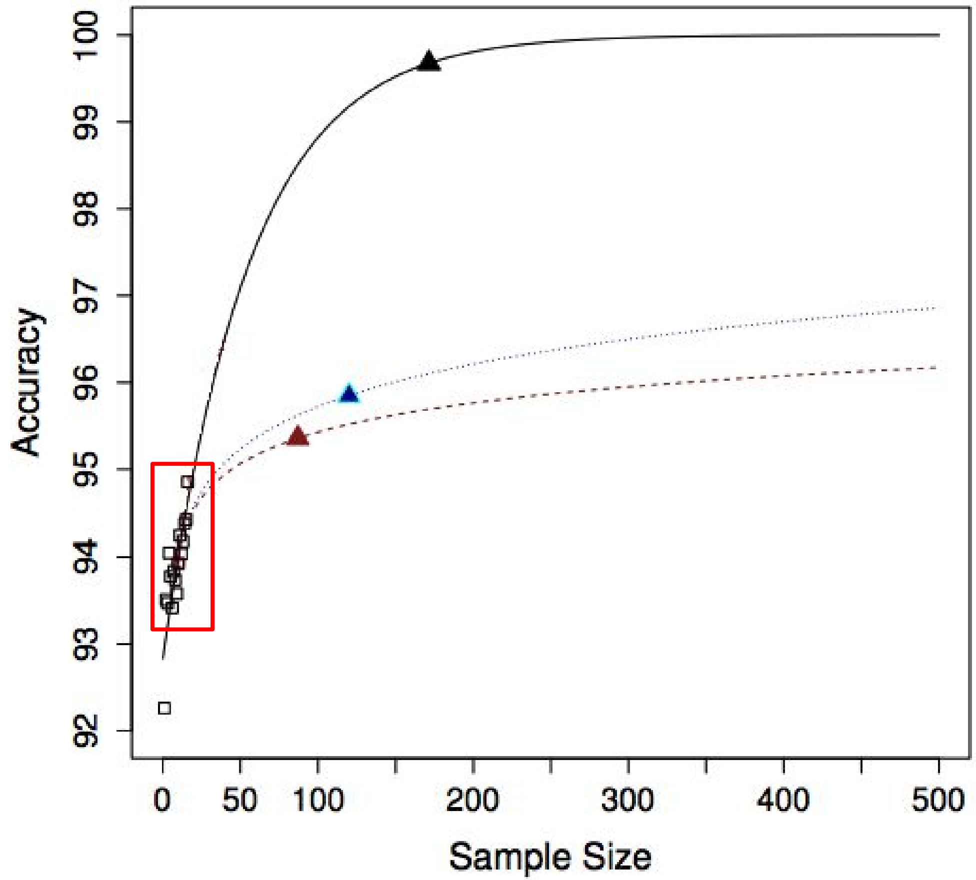
Residual-based Methods
Projective Sampling

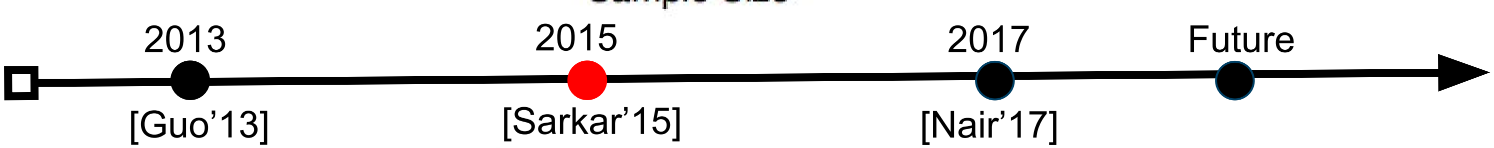
Estimates the Learning Curve



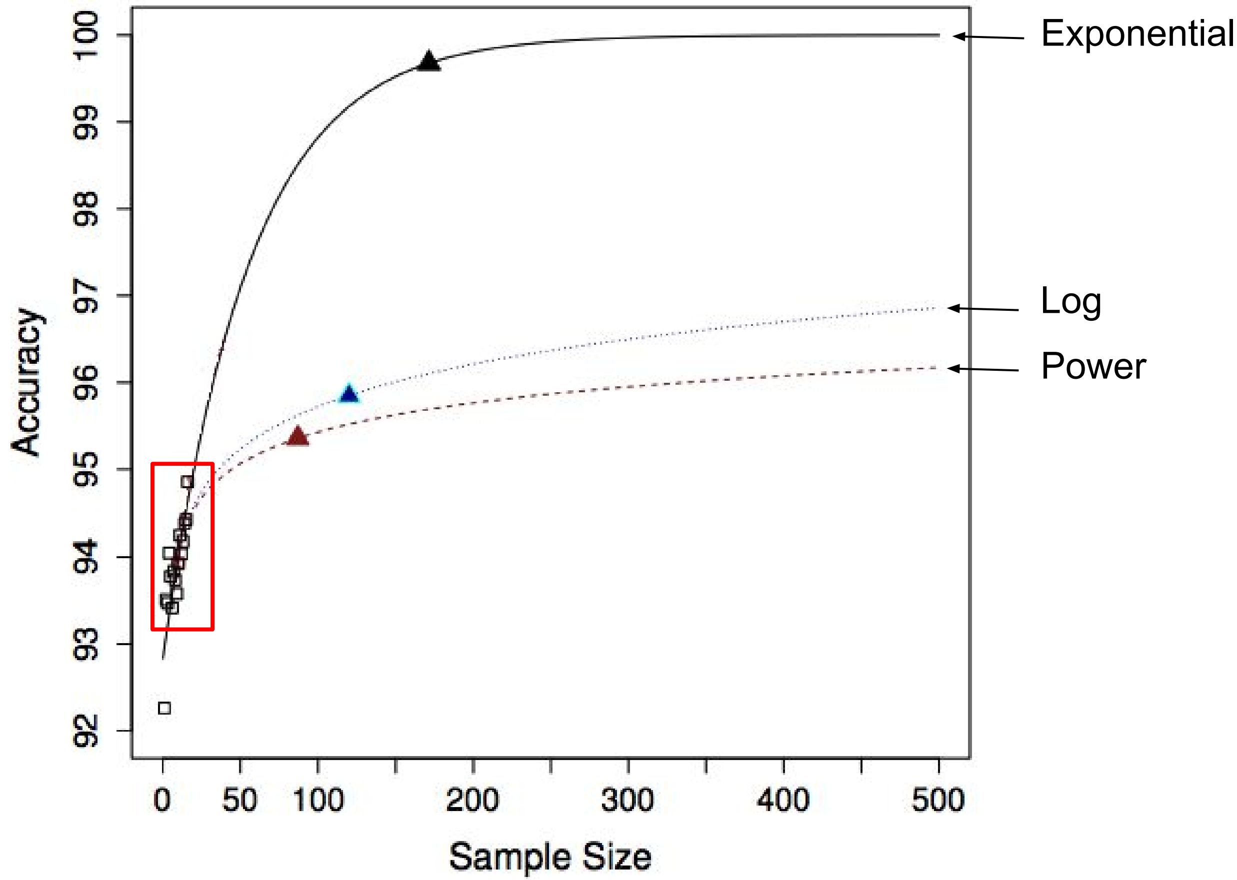


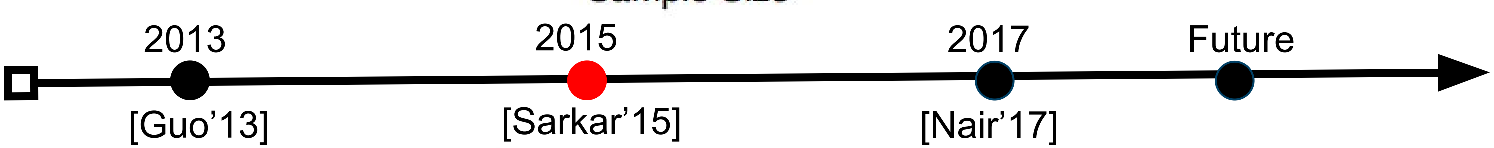
Residual-based Methods
Projective Sampling





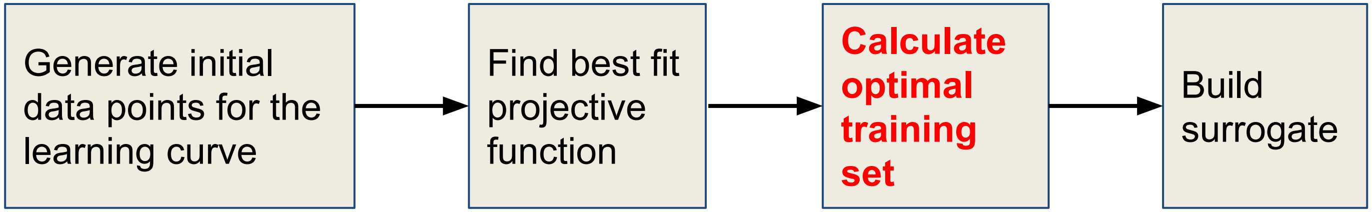
Residual-based Methods
Projective Sampling





Residual-based Methods
Projective Sampling

Estimates the Learning Curve



Residual-based Methods
Projective Sampling

Penalty factor

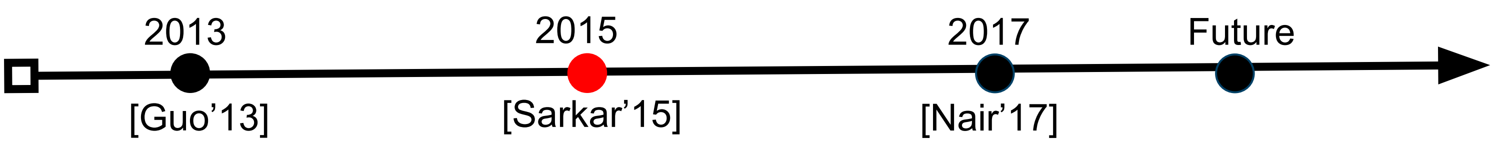
Table II: Projective functions of learning curves

Name	Equation	Optimal Sample Size
Logarithmic	$err(n) = a + b.log(n)$	$n^* = -(R \cdot \lvert S\rvert \cdot b)/2$
Weiss and Tian	$err(n) = a + bn/(n+1)$	$n^* = \sqrt{(-R \cdot \lvert S\rvert \cdot b)/2}$
Power Law	$err(n) = an^b$	$n^* = (\frac{-2}{R \cdot \lvert S\rvert \cdot a \cdot b})^{\frac{1}{b-1}}$
Exponential	$err(n) = ab^n$	$n^* = \log_b \left(\frac{-2}{R \cdot \lvert S\rvert \cdot a \cdot \ln b}\right)$

Taken from Sarkar et al.

Coefficients of projective function

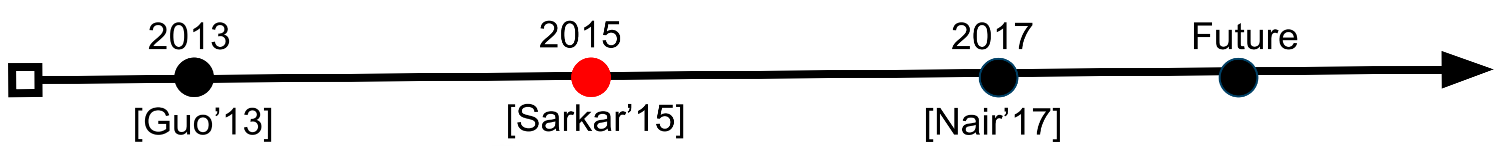
Number of configurations whose performance value will be predicted



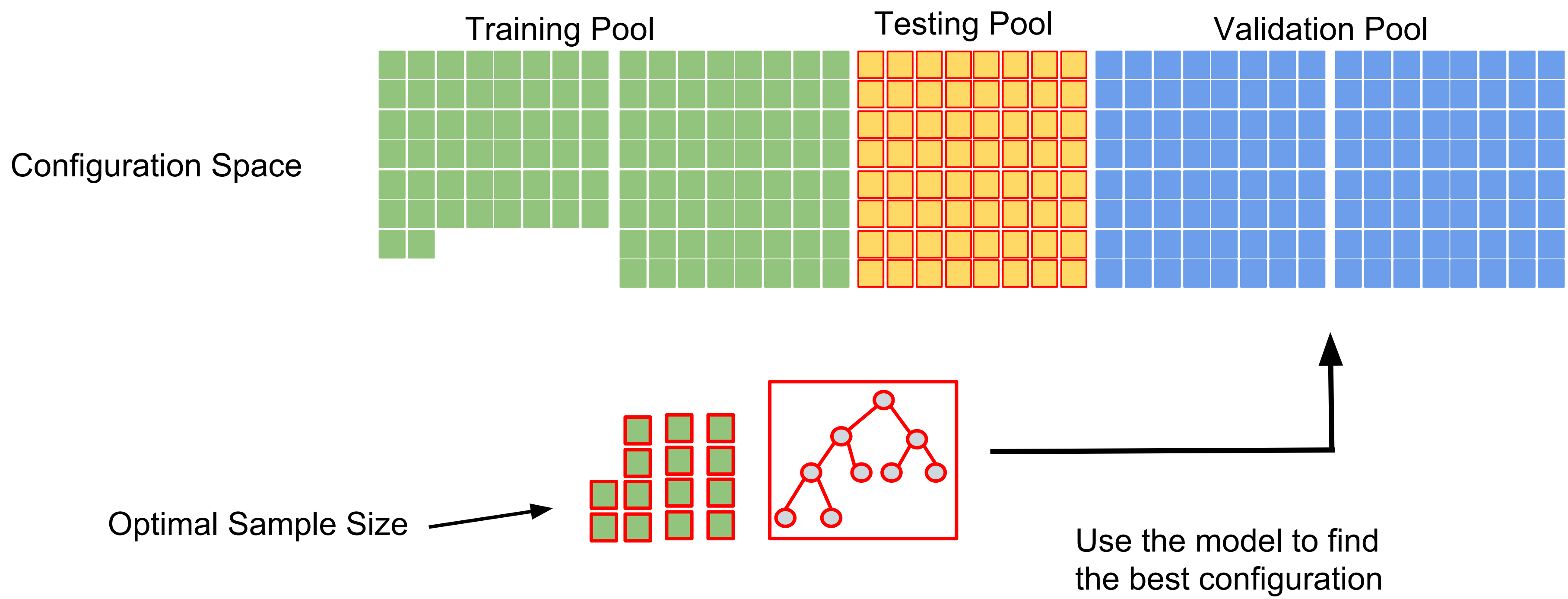
Residual-based Methods
Projective Sampling

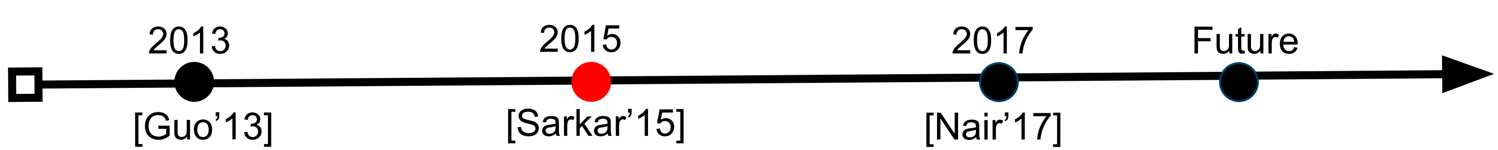
Estimates the Learning Curve

Generate initial data points for the learning curve → Find best fit projective function → Calculate optimal training set → **Build surrogate**



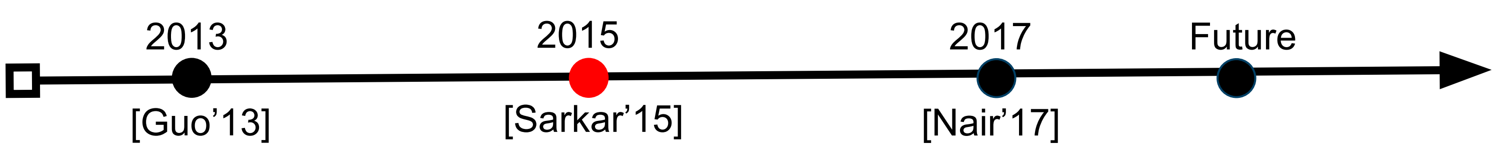
Residual-based Methods
Projective Sampling





Residual-based Methods
Projective Sampling - Limitation

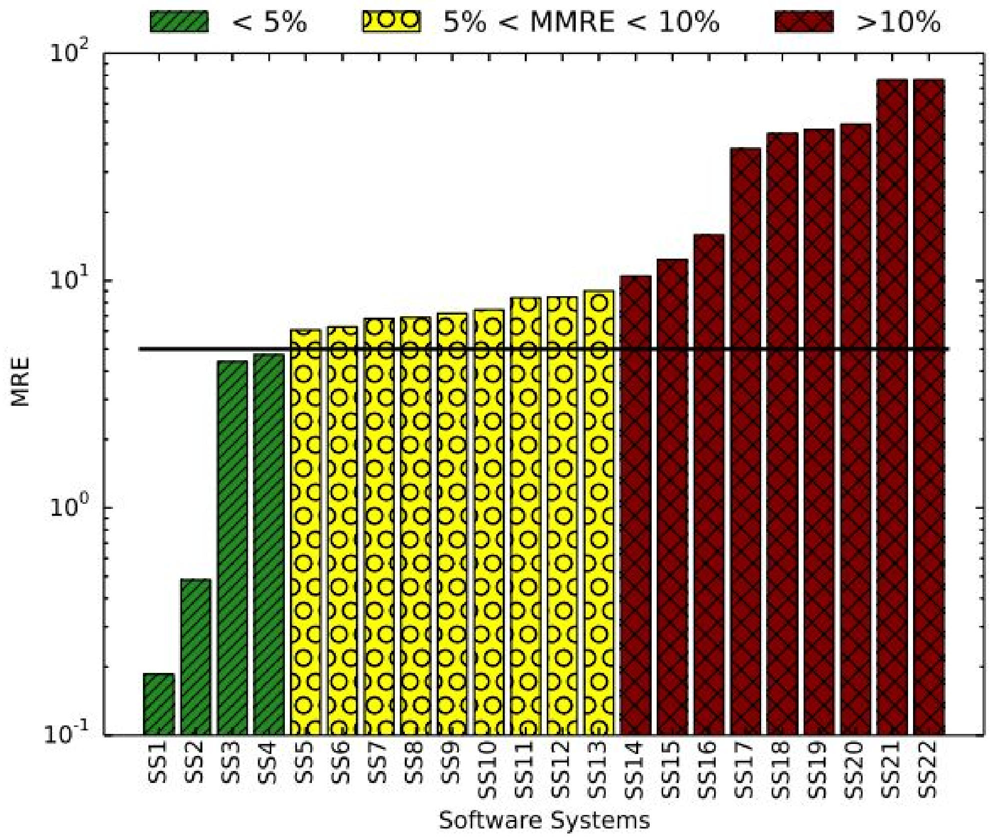
Assumes an accurate model can be built

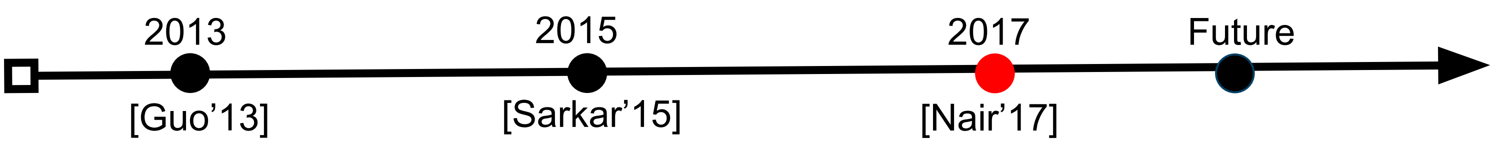


Rank-based Method

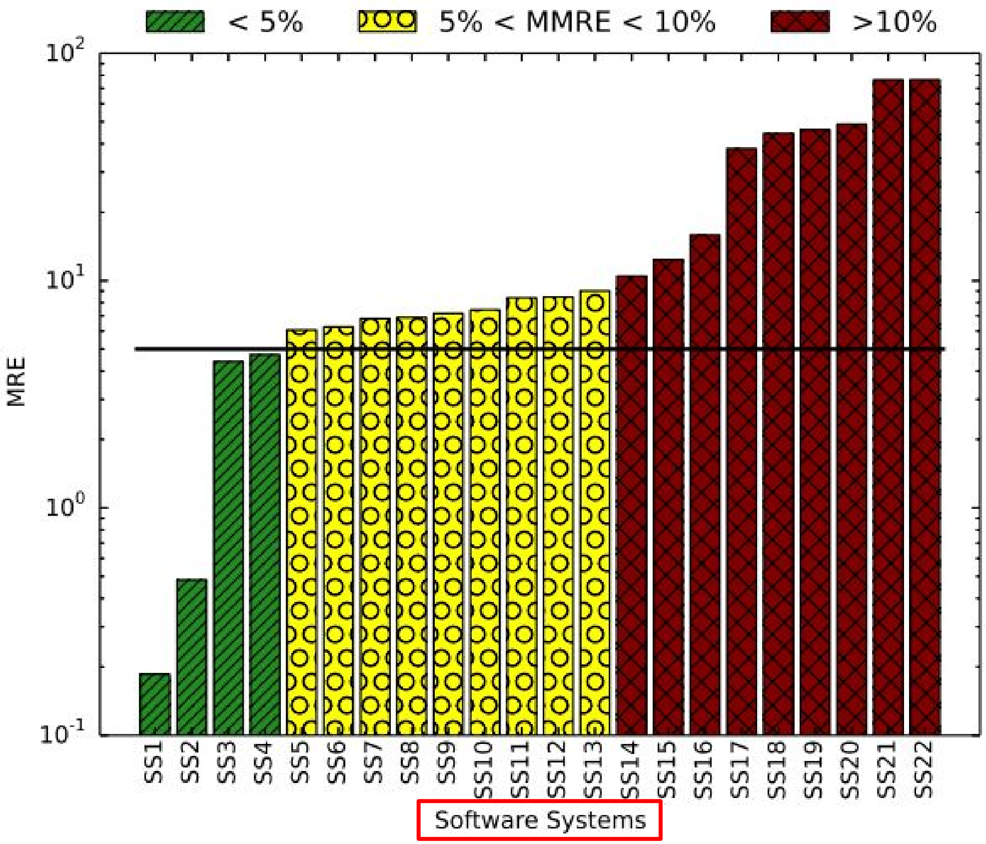
Nair, Vivek, et al. "Using Bad Learners to find Good Configurations." FSE 2017

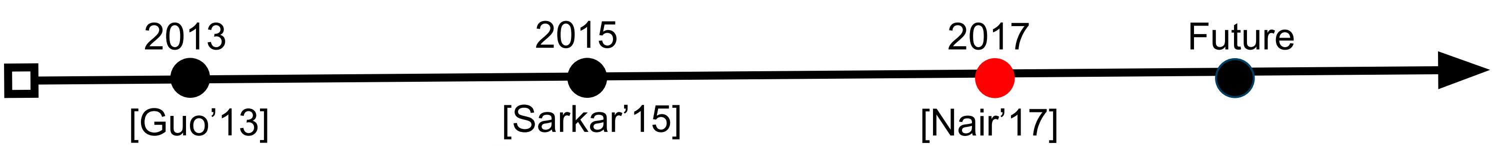
Rank-based Method



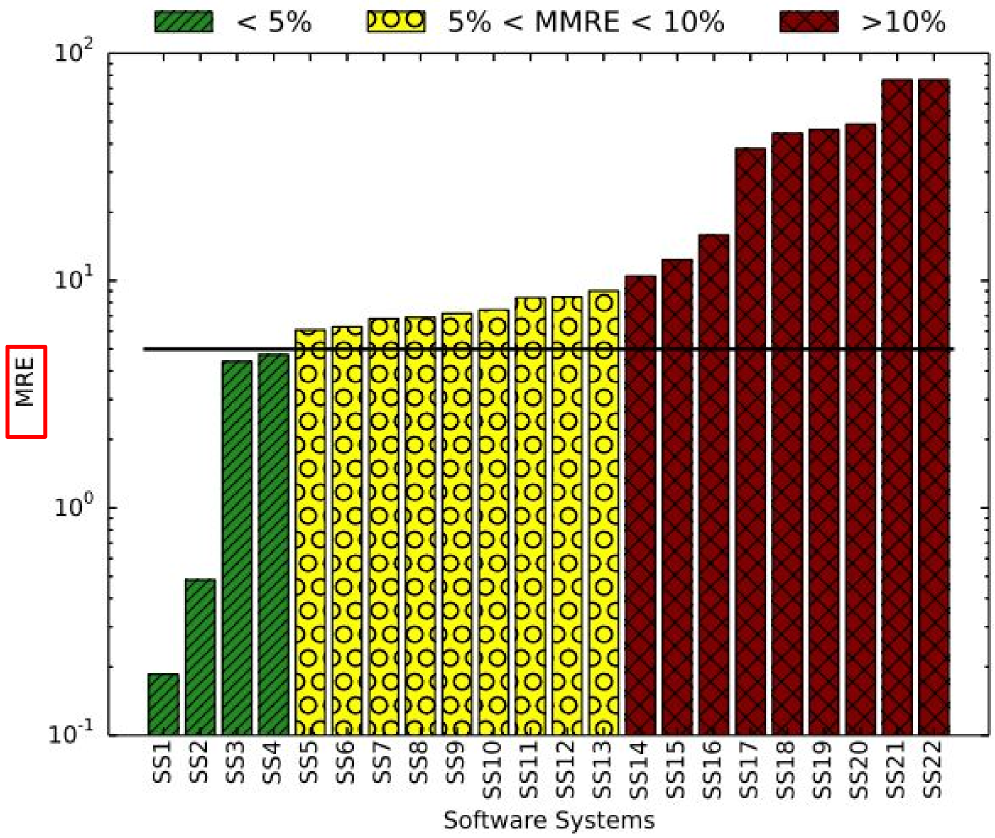


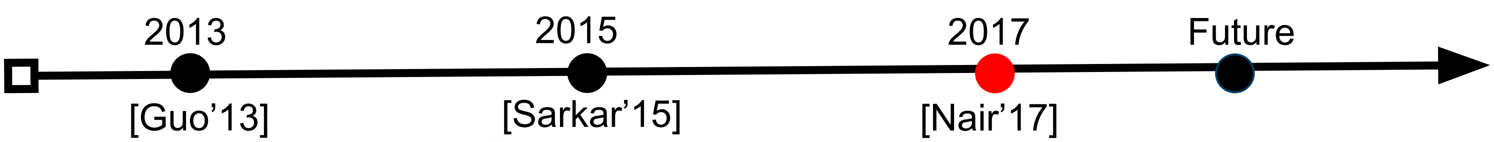
Rank-based Method



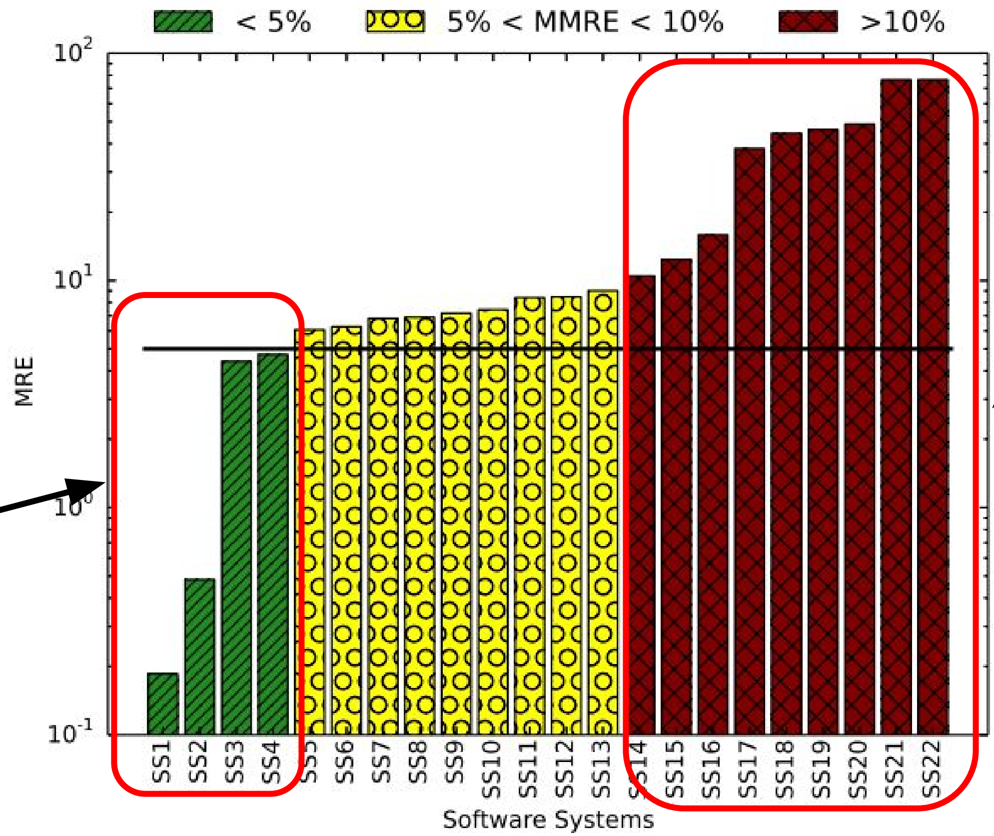


Rank-based Method



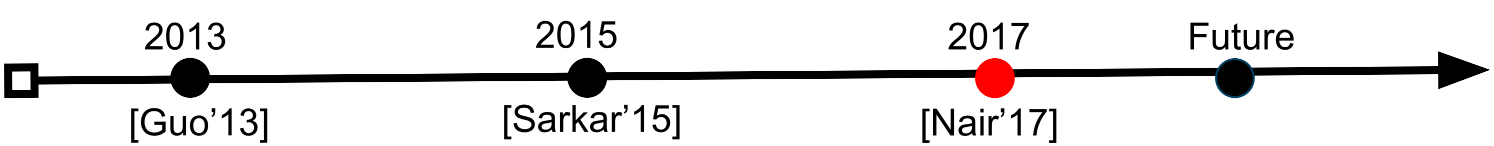


Rank-based Method



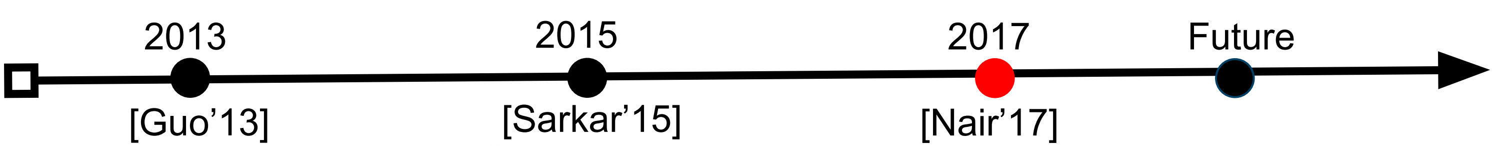
Software Systems cannot be accurately modelled

Software Systems used in existing works can be accurately modelled

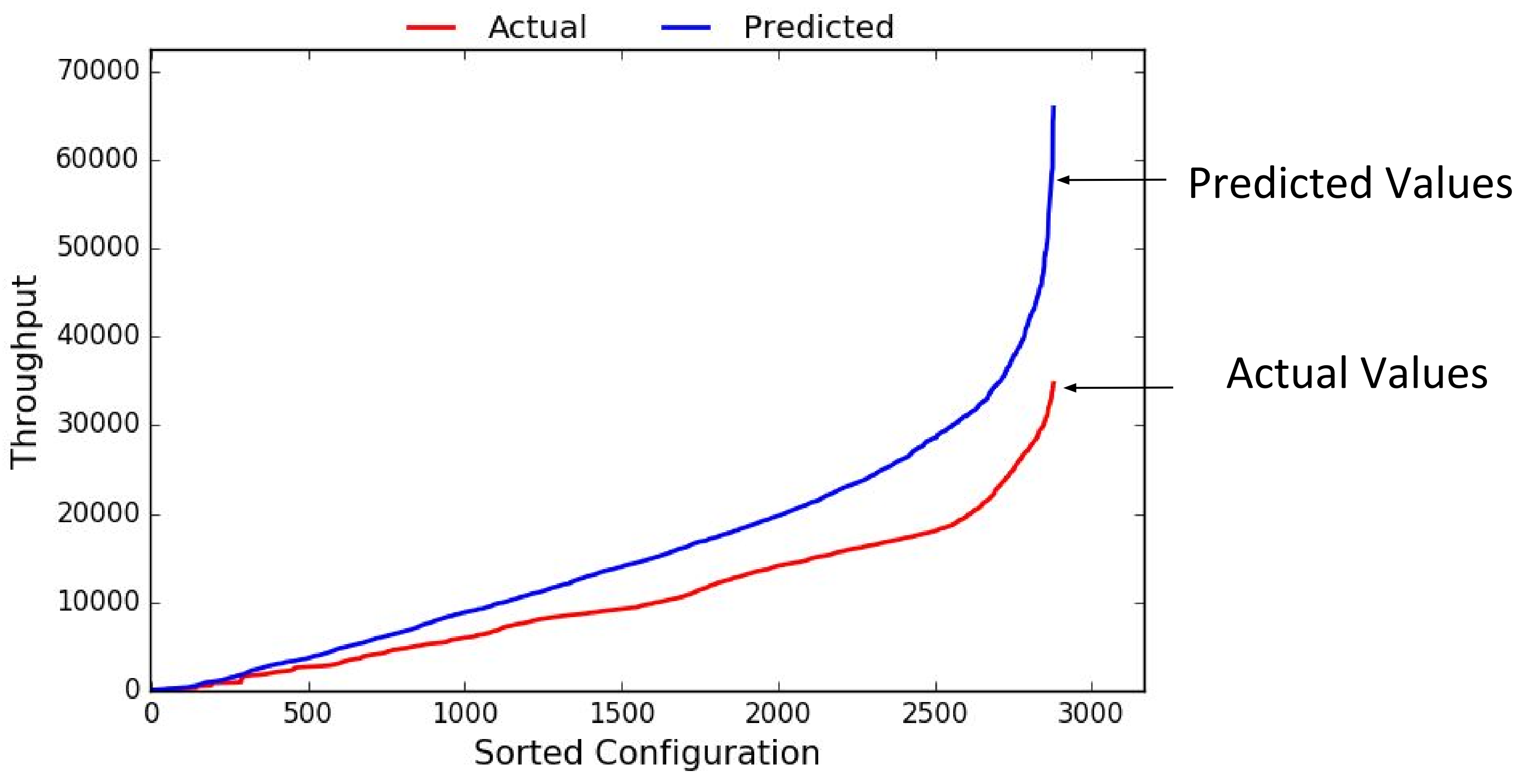


Rank-based Method
Core Insights

Rank-preserving model rather than highly accurate model

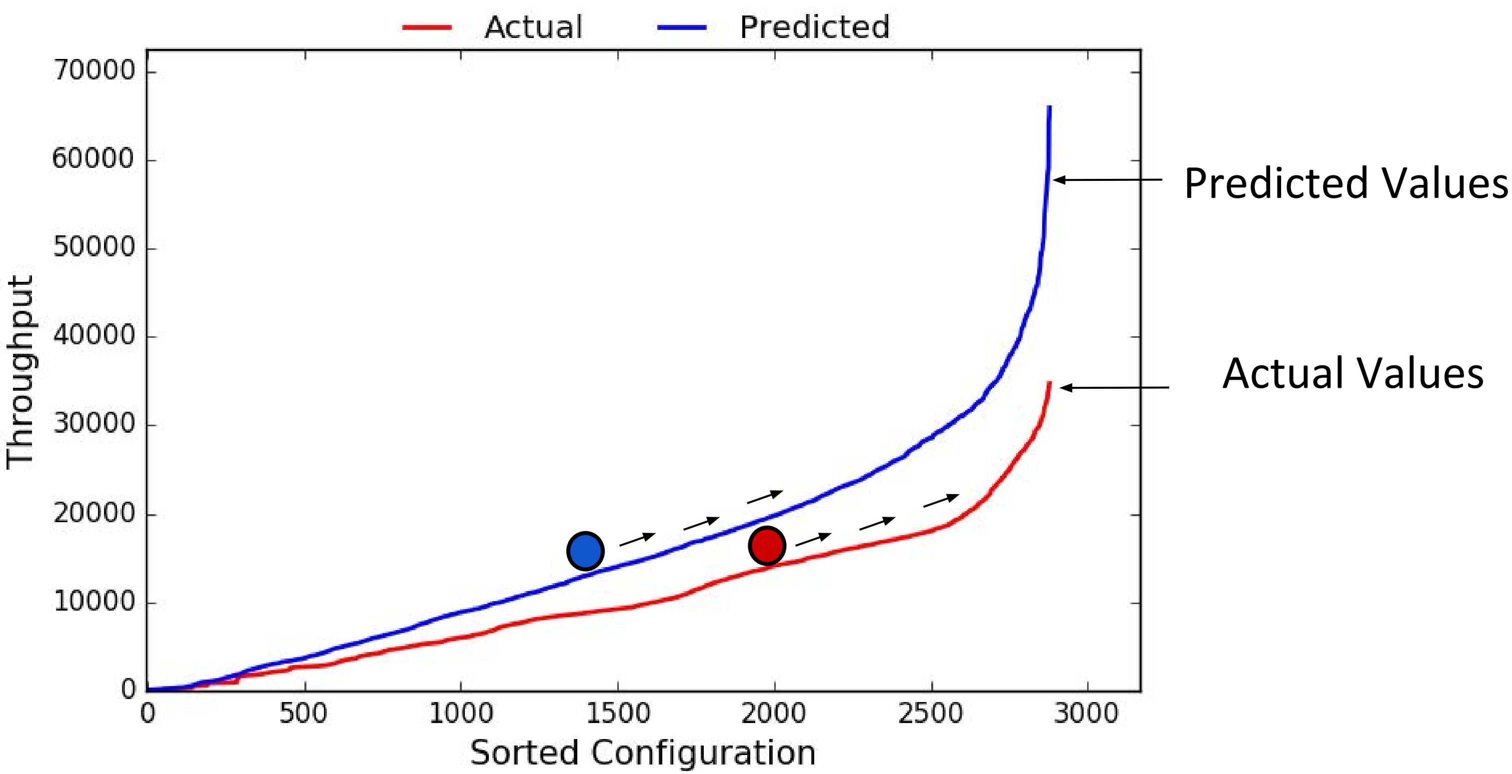


Rank Preserving Model



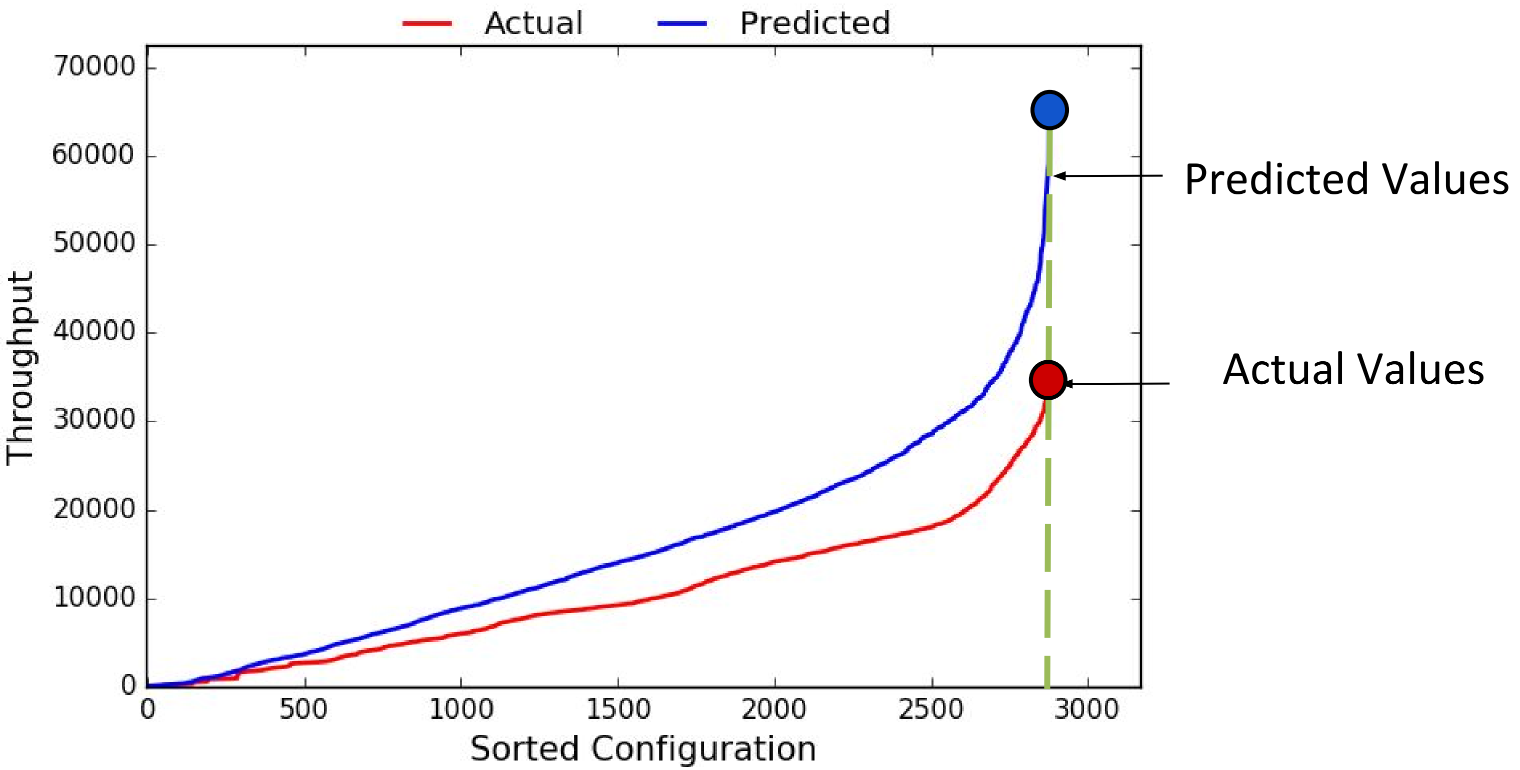
Best Configuration obtained using actual and the predicted values is the same

Rank Preserving Model



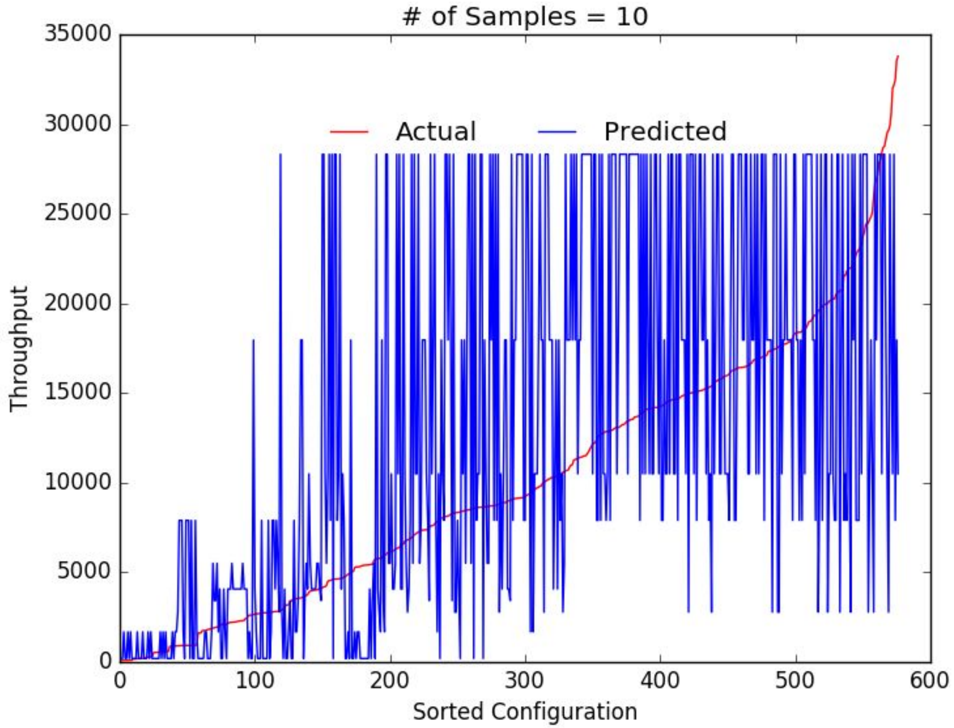
Best Configuration obtained using actual and the predicted values is the same

Rank Preserving Model

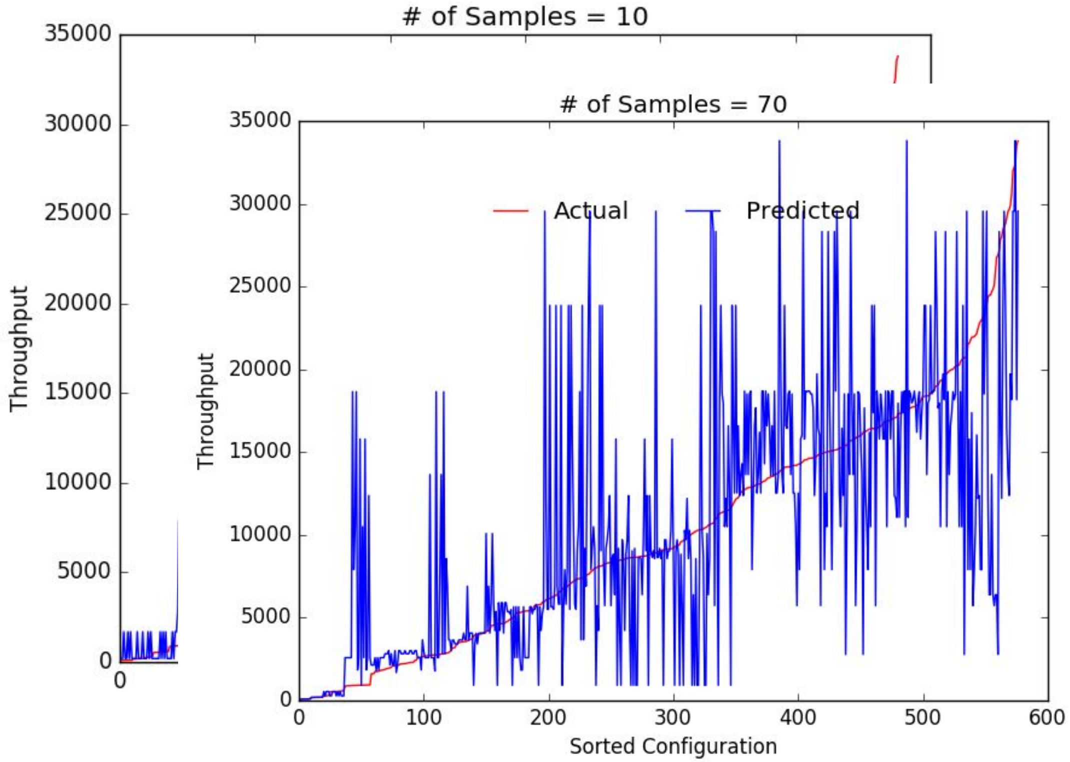


Best Configuration obtained using actual and the predicted values is the same

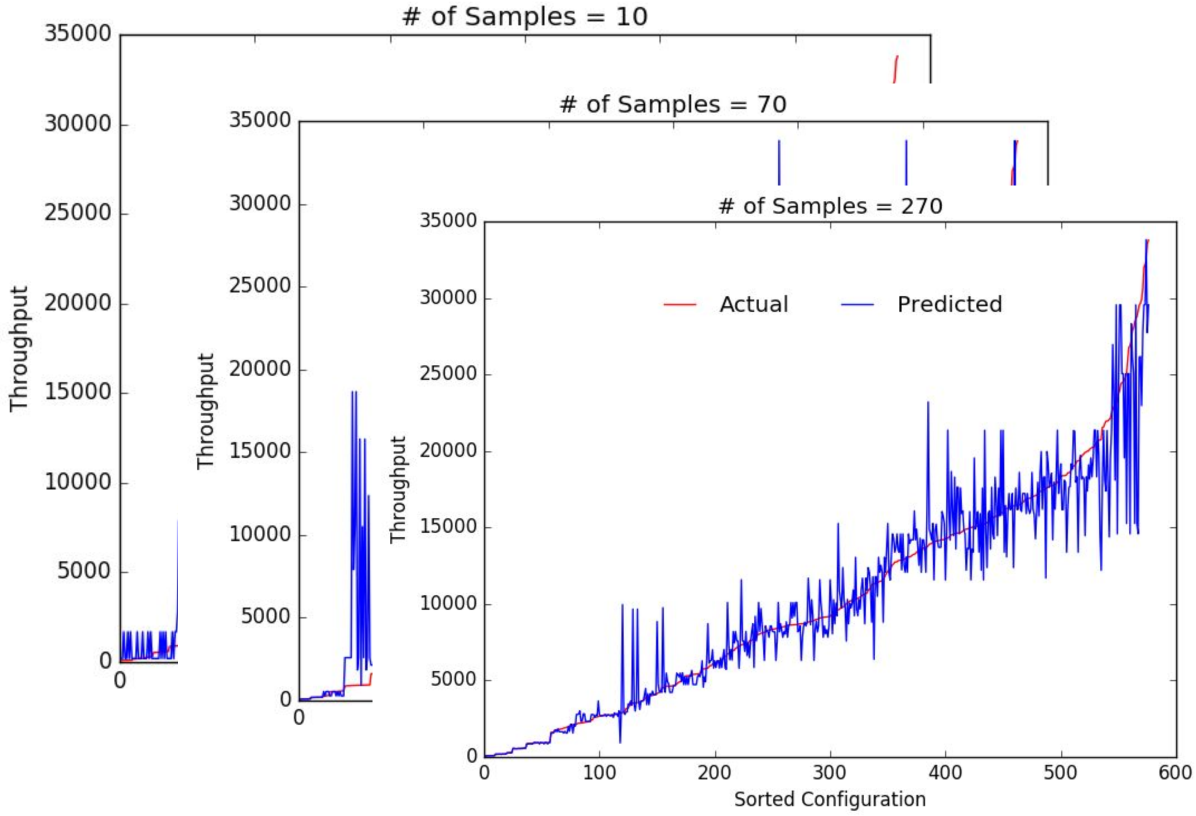
Rank Preserving Model



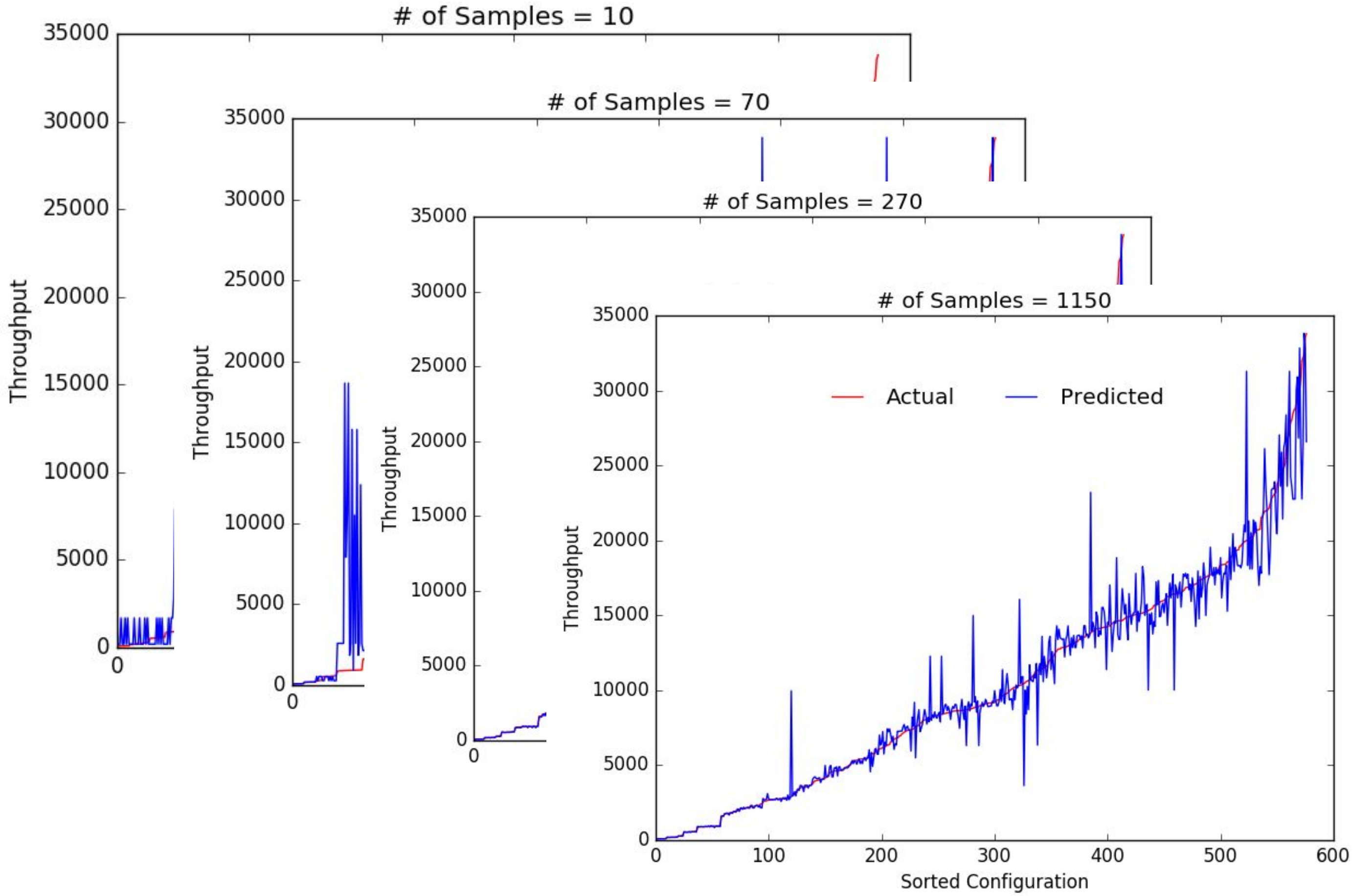
Rank Preserving Model



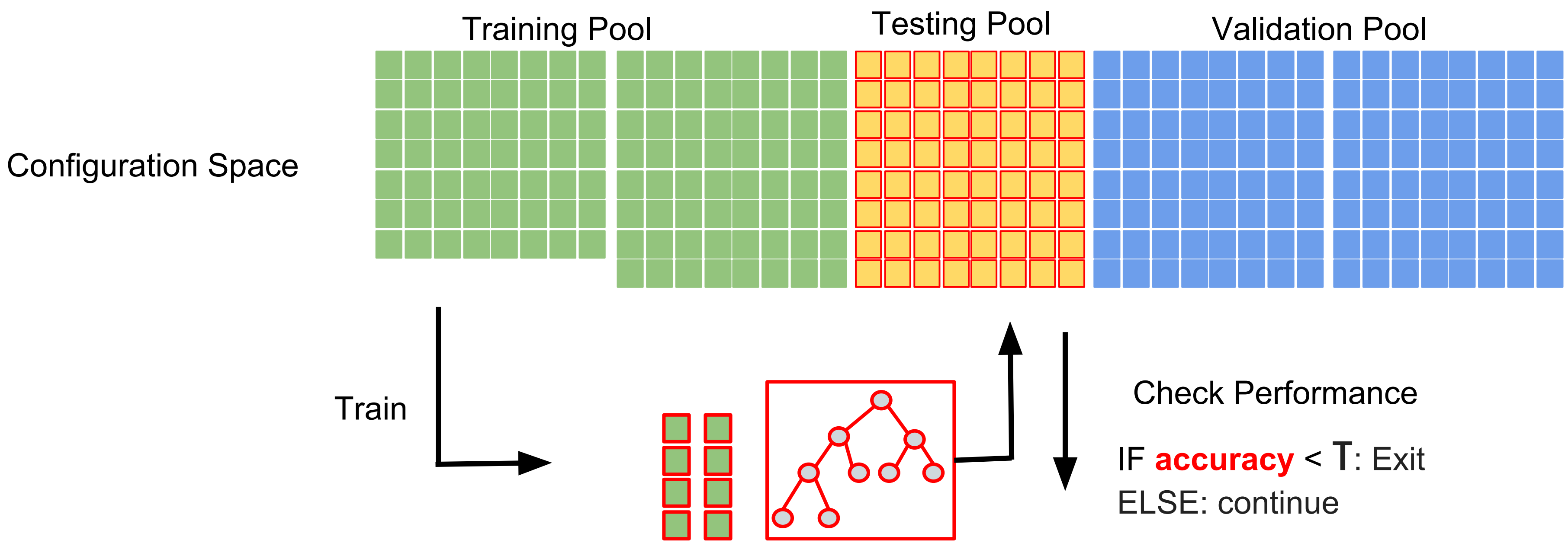
Rank Preserving Model

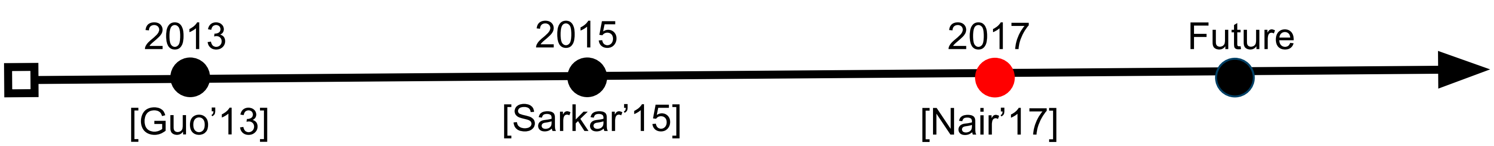


Rank Preserving Model

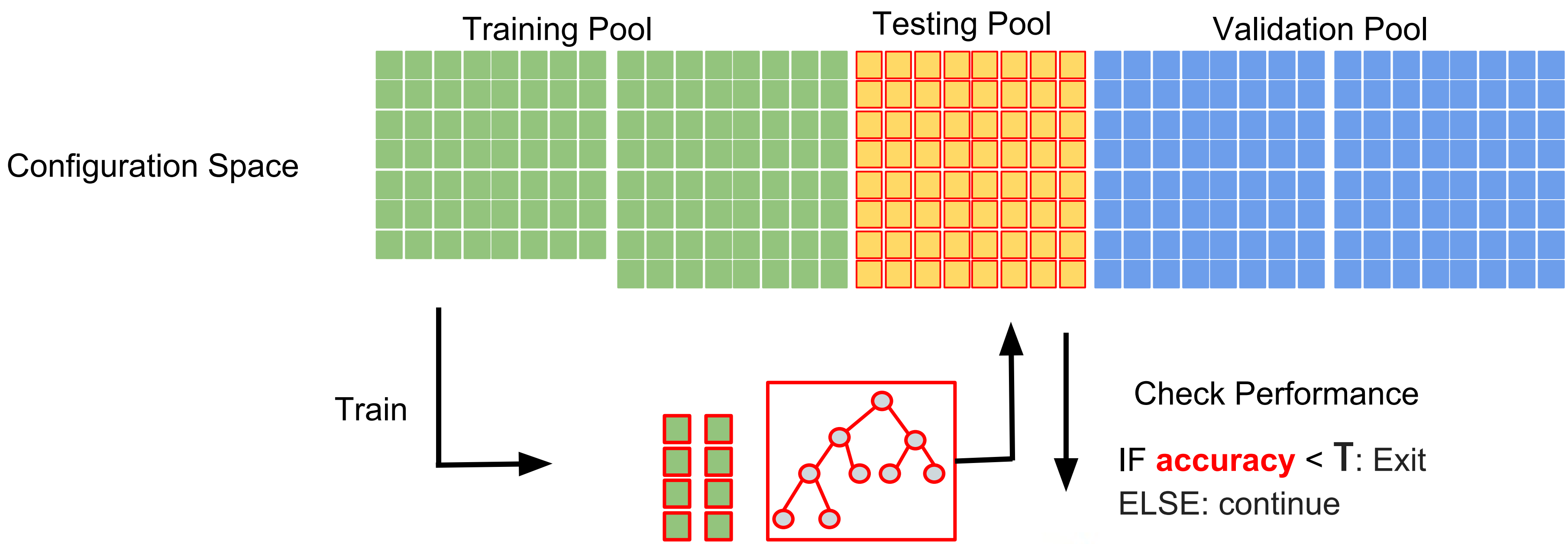


Rank Preserving Model

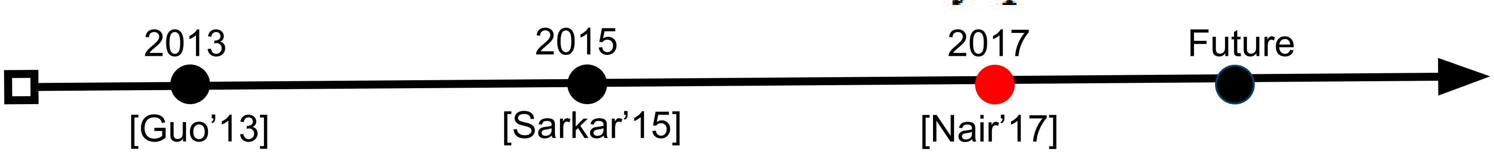




Rank Preserving Model

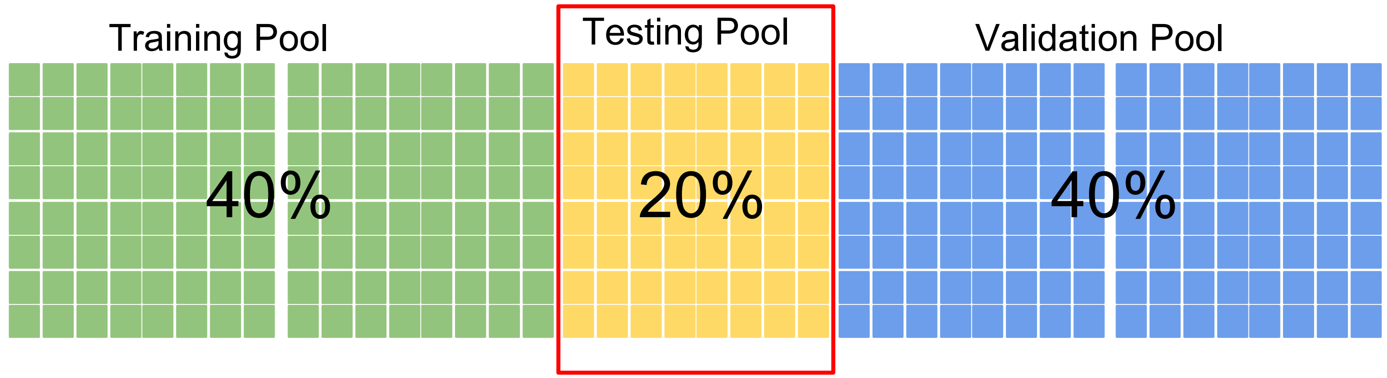


$$accuracy = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| rank(y_i) - rank(f(x_i)) \right|$$

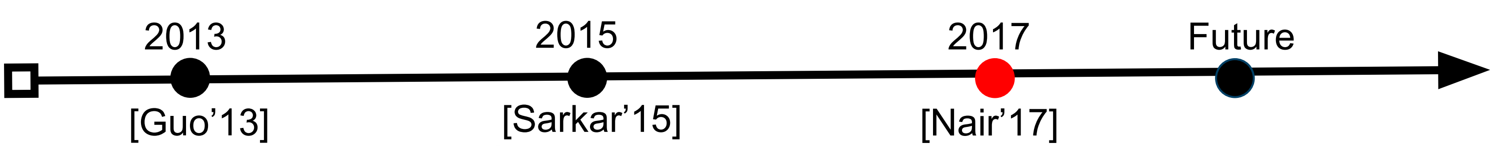


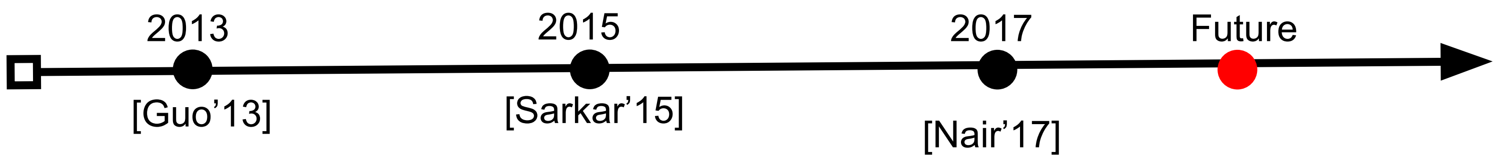
Rank Preserving Model - Limitation

Configuration Space



Requires Testing Pool - 20% of configuration space

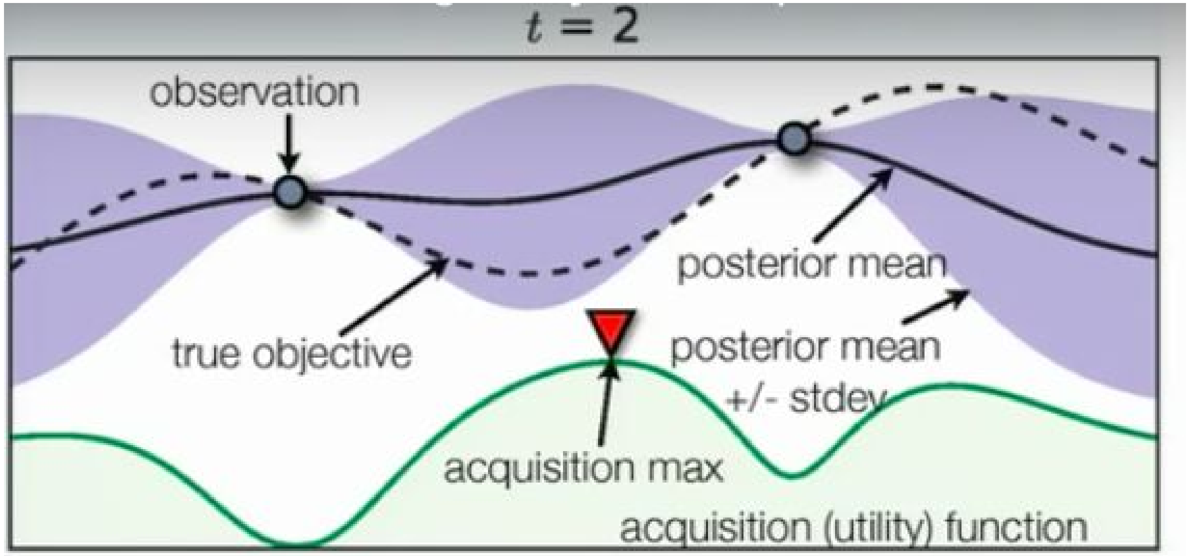




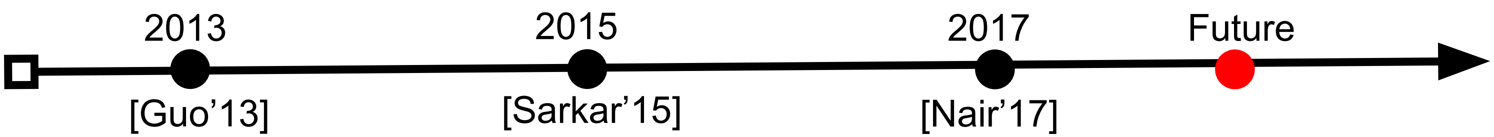
Future - Bayesian based Method

Nair, Vivek et al. "FLASH: A Faster Optimizer for SBSE Tasks." *preprint*

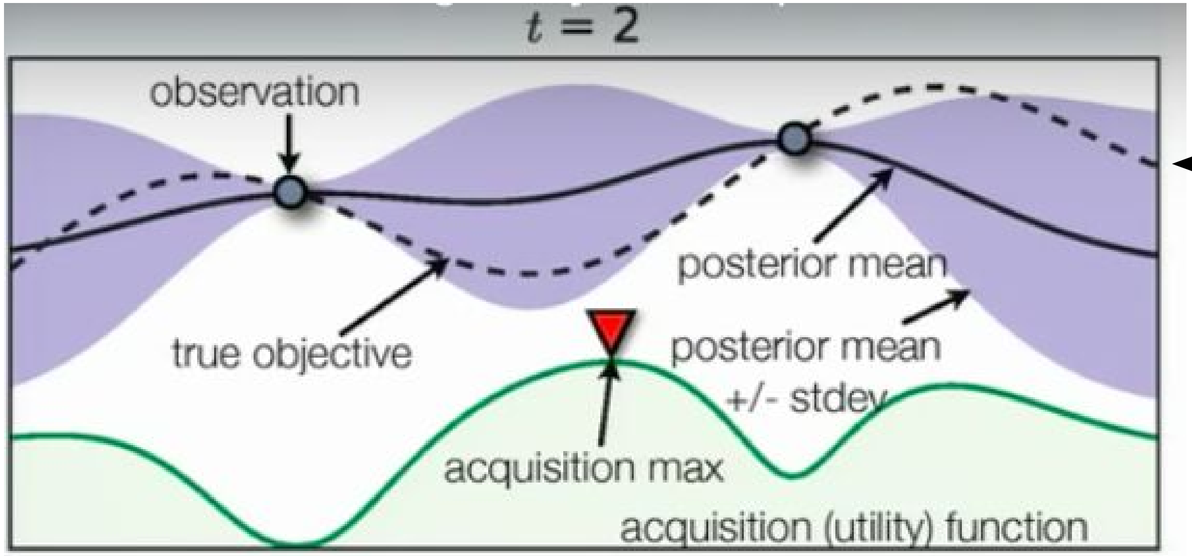
Bayesian Optimization



Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

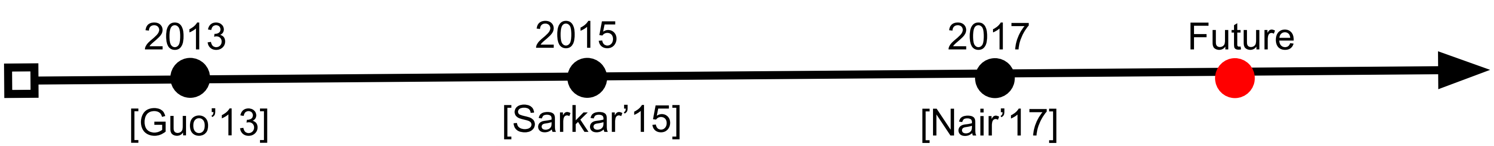


Bayesian Optimization

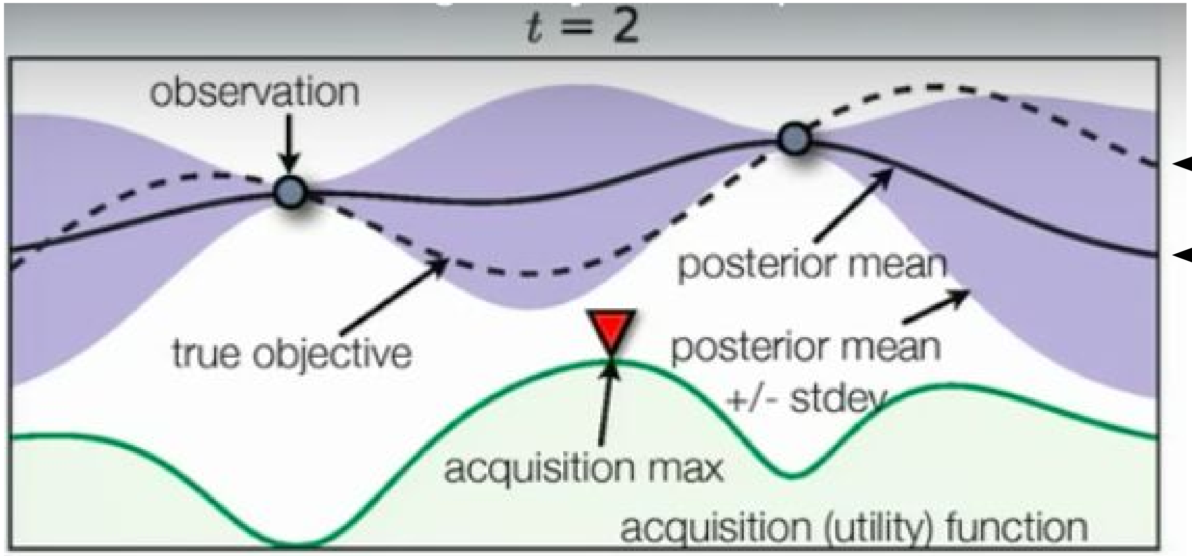


Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

True Performance Distribution



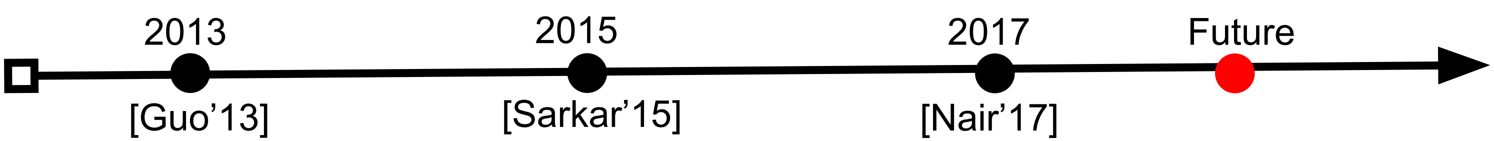
Bayesian Optimization



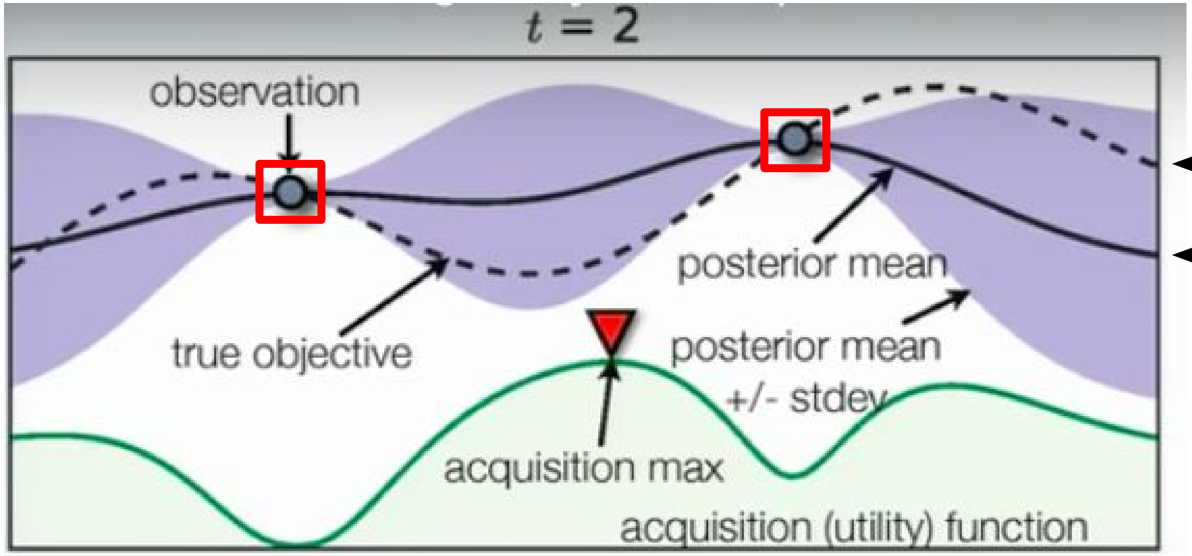
Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

True Performance Distribution

Predicted Performance Distribution



Bayesian Optimization

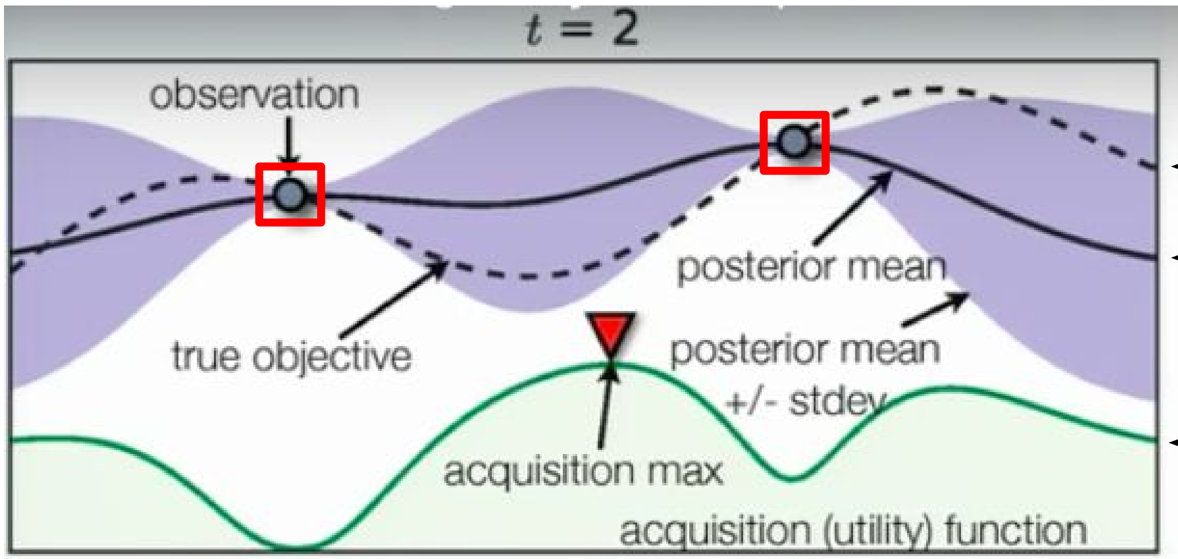


True Performance Distribution

Predicted Performance Distribution

Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

Bayesian Optimization



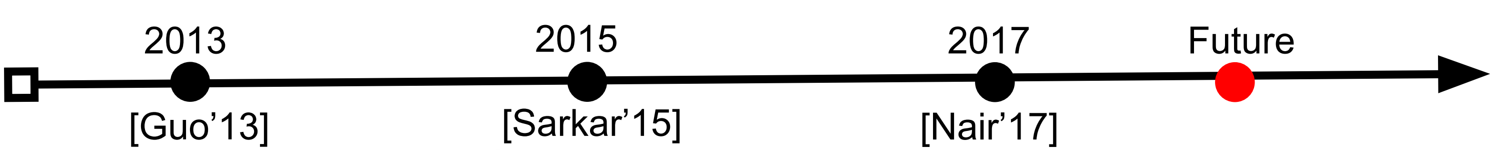
Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

True Performance Distribution

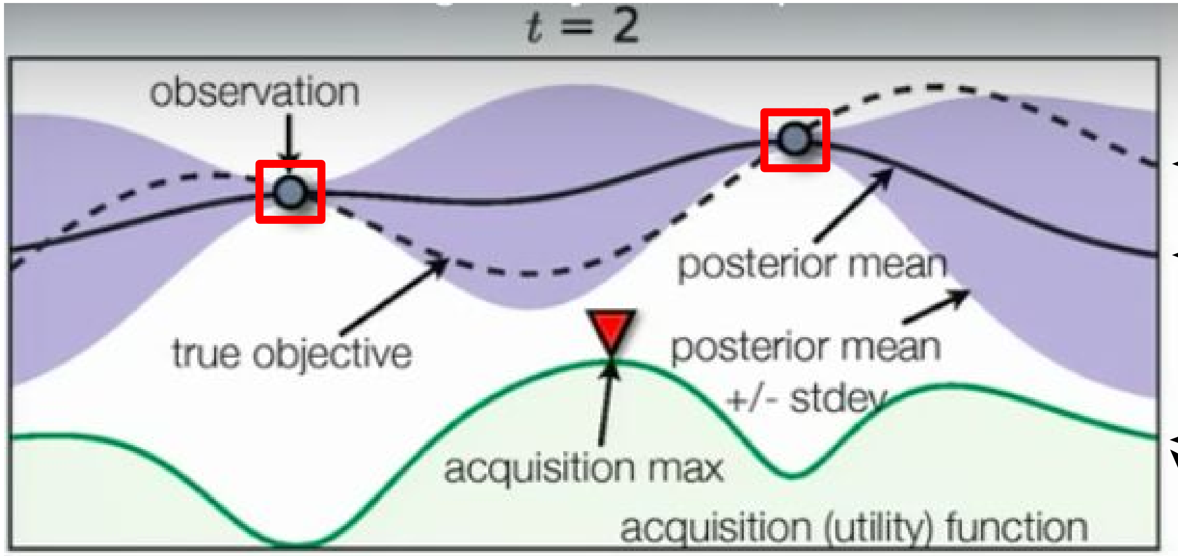
Predicted Performance Distribution

Acquisition Function

Which configuration should I evaluate next?



Bayesian Optimization



Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

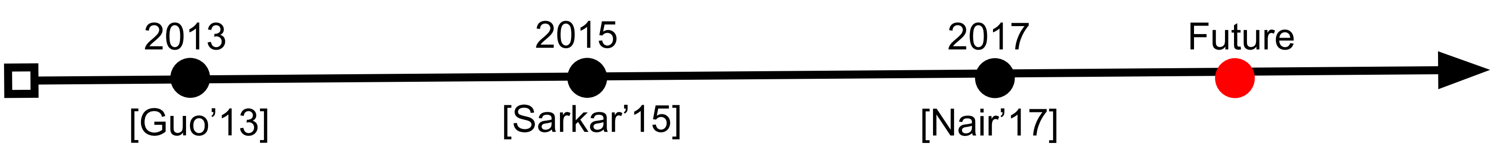
True Performance Distribution

Predicted Performance Distribution

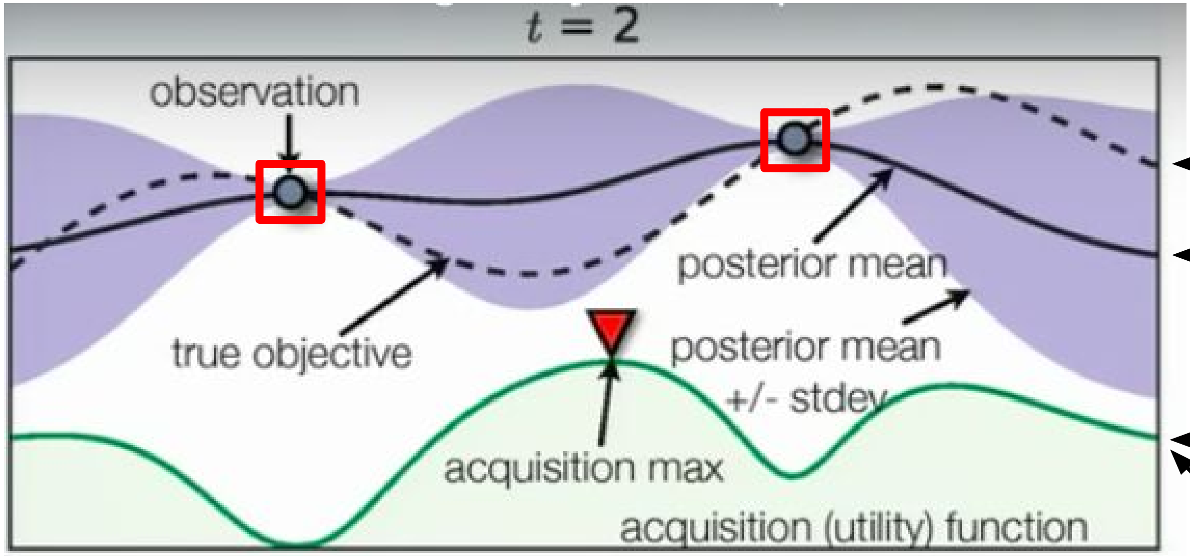
Acquisition Function

$$\mu(x) + \kappa\sigma(x)$$

Which configuration should I evaluate next?



Bayesian Optimization



Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

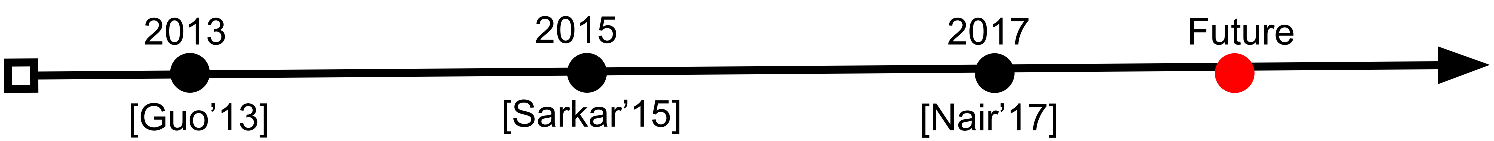
True Performance Distribution

Predicted Performance Distribution

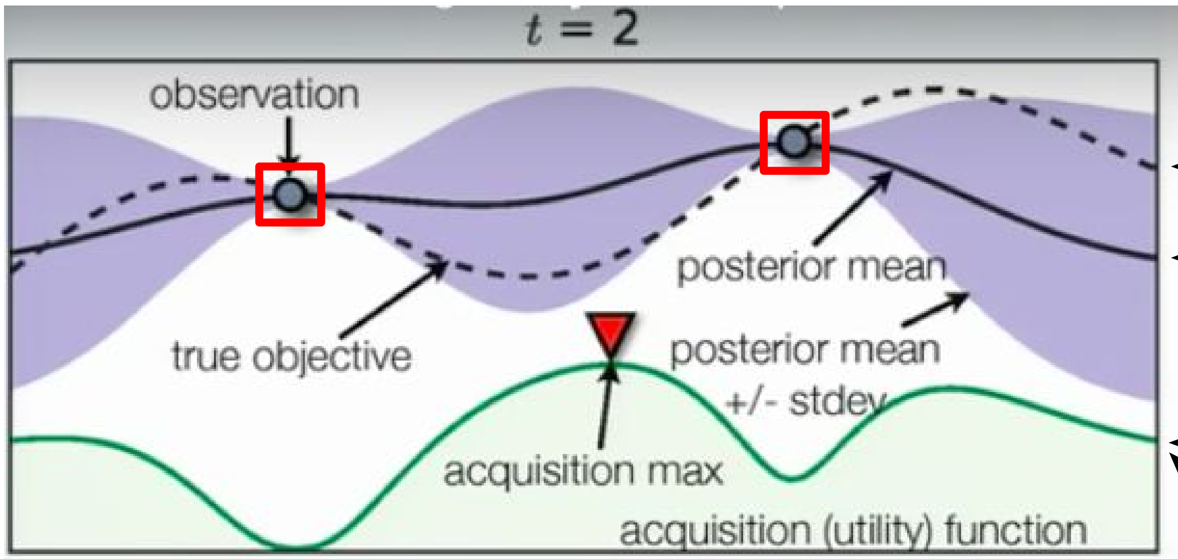
Acquisition Function

$$\mu(x) + \boxed{\kappa}\sigma(x)$$

Tradeoff between <u>Exploration vs Exploitation</u>



Bayesian Optimization



Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

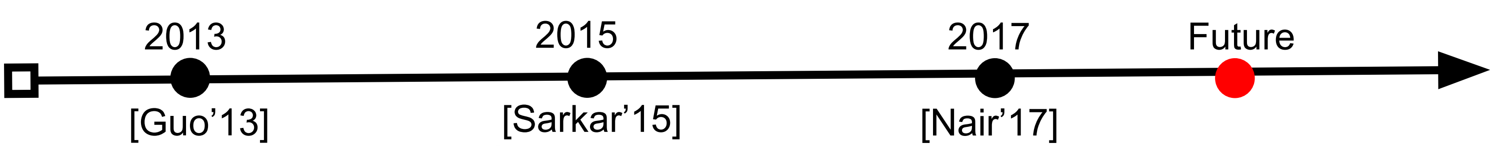
True Performance Distribution

Predicted Performance Distribution

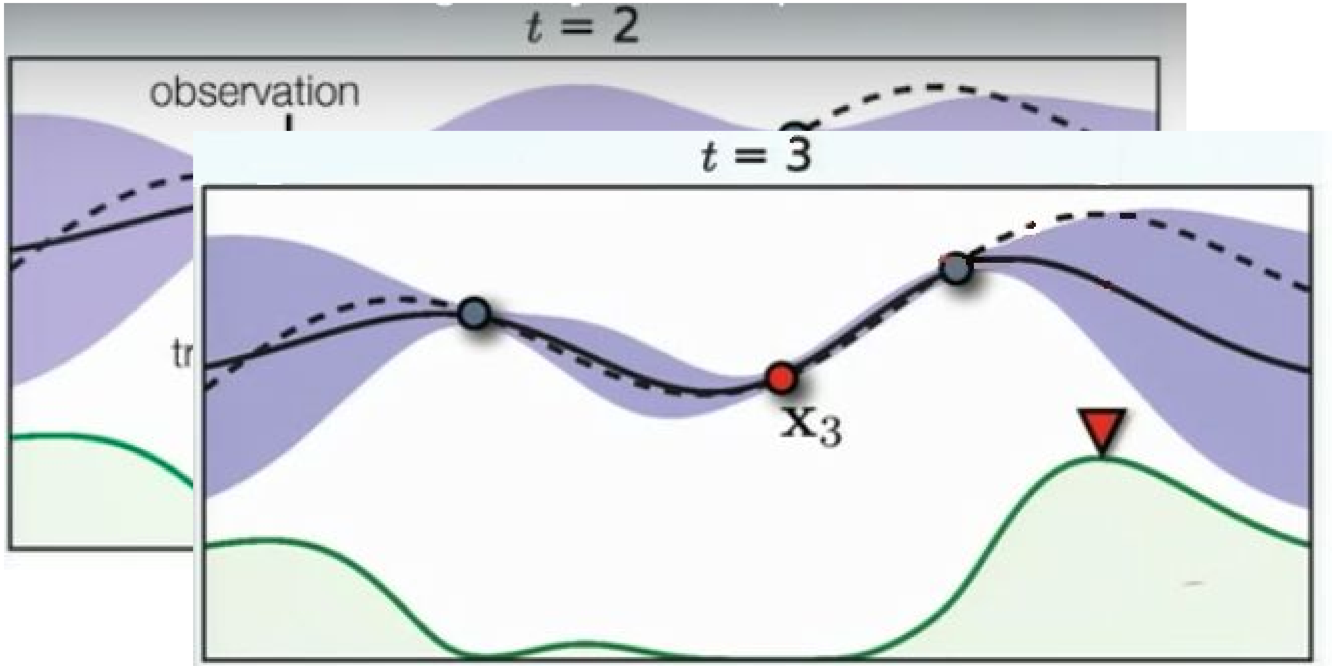
Acquisition Function

$$\mu(x) + \boxed{\kappa}\sigma(x)$$

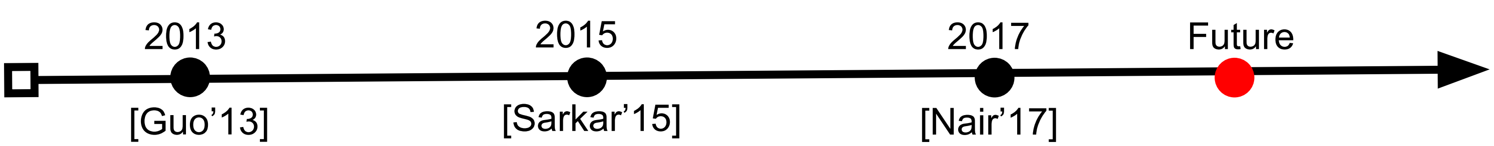
Surrogate of choice: Gaussian Processes (GP)



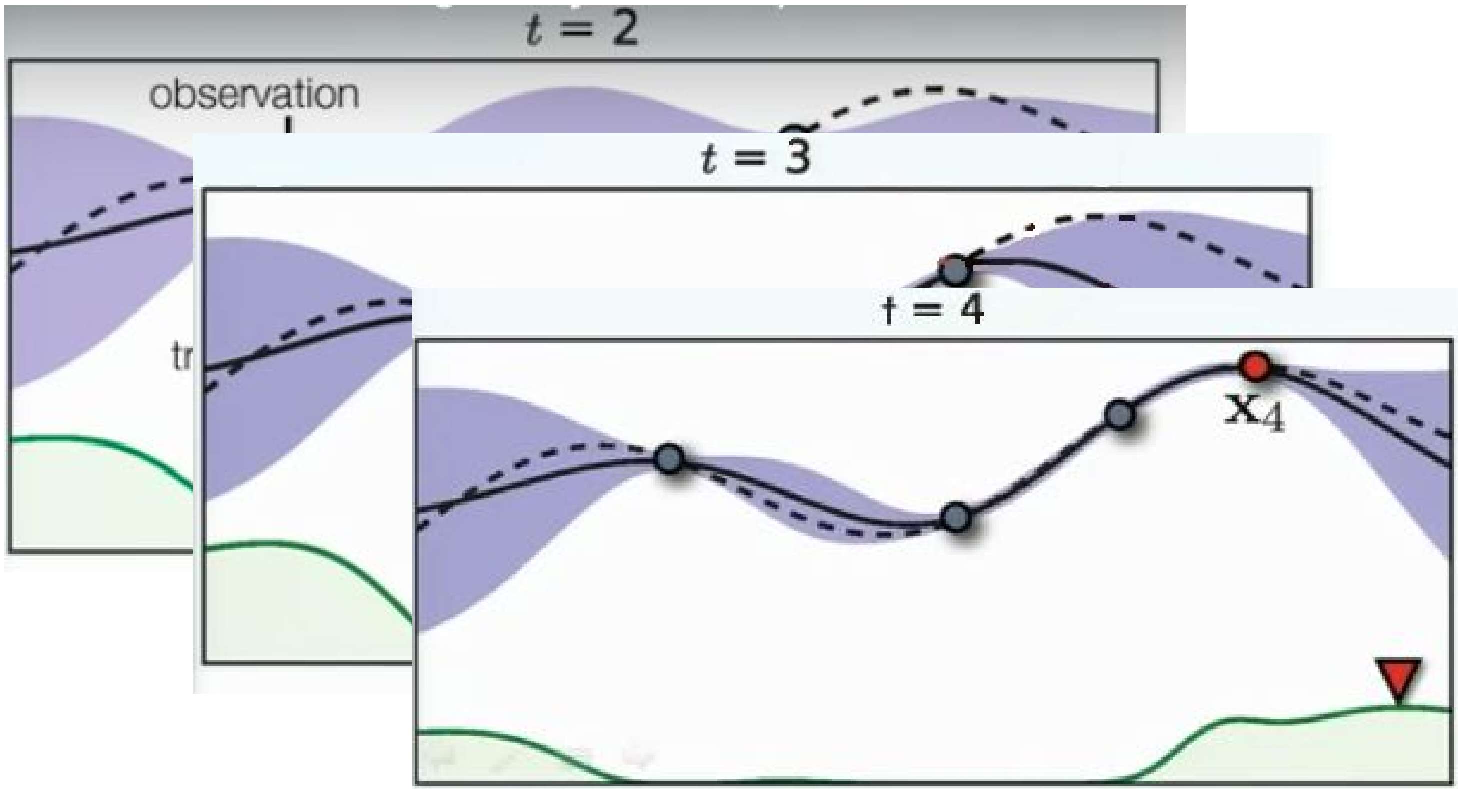
Bayesian Optimization



Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)



Bayesian Optimization



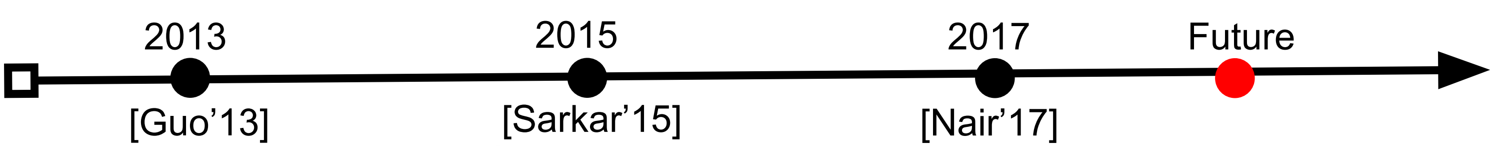
Taken from Dr. Nando de Freitas (tiny.cc/4tgeny)

2013 [Guo'13] — 2015 [Sarkar'15] — 2017 [Nair'17] — Future

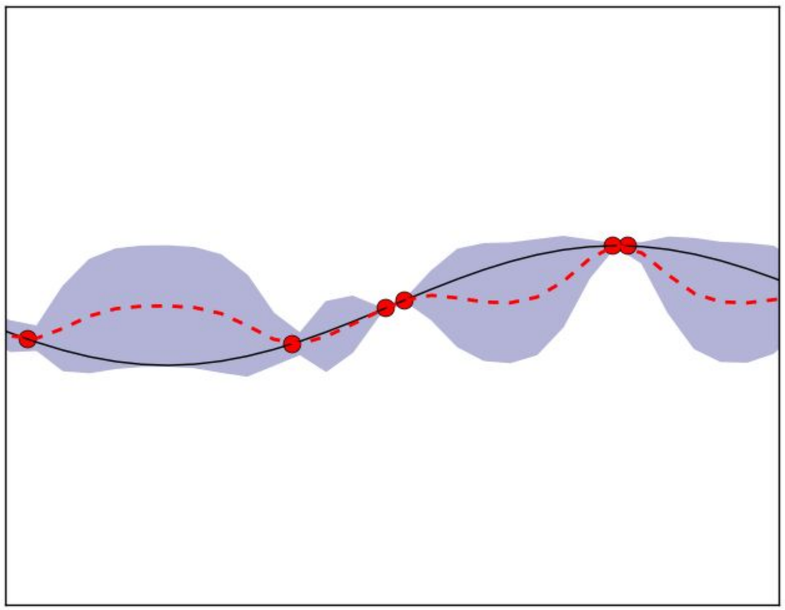
Bayesian Optimization

GP lose efficiency in high dimensional spaces

i.e. number of features exceeds a dozen



Bayesian-based Method - FLASH

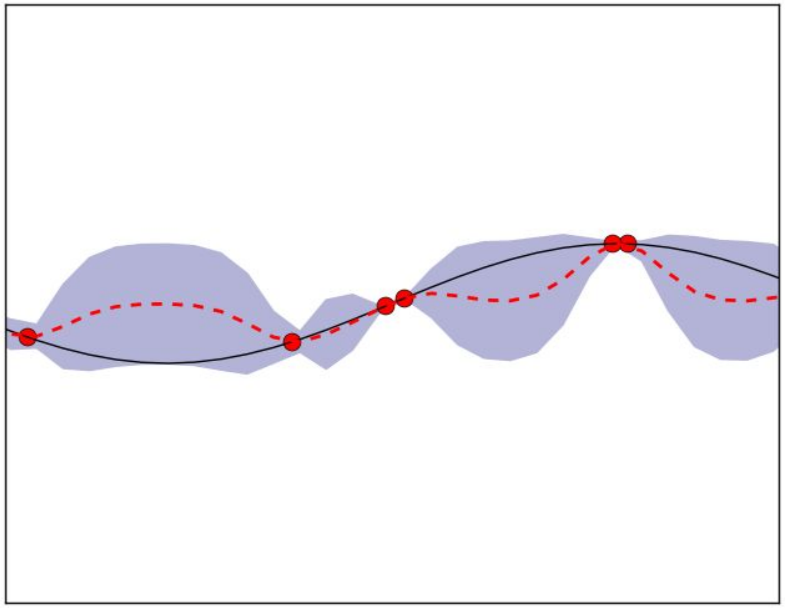


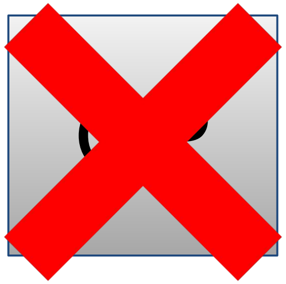
GP

$$\mu(x) + \kappa\sigma(x)$$

2013 [Guo'13] — 2015 [Sarkar'15] — 2017 [Nair'17] — Future

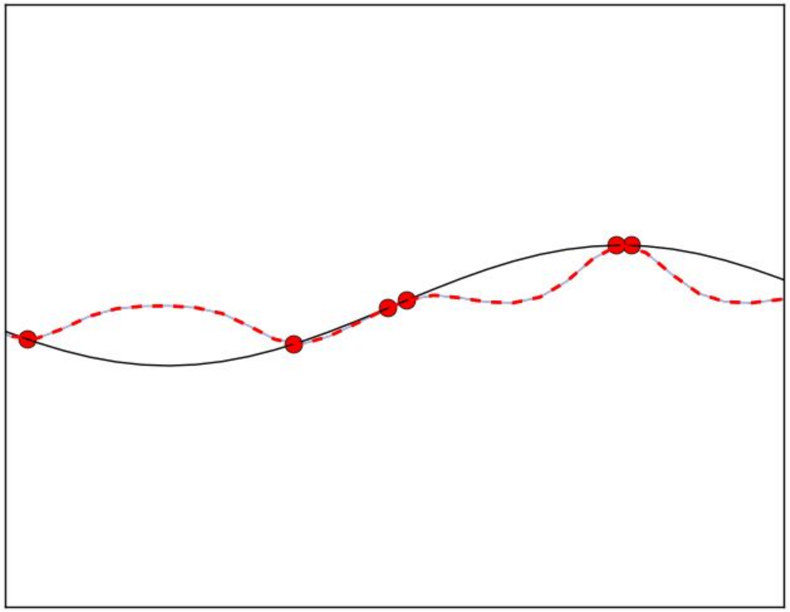
Bayesian-based Method - FLASH

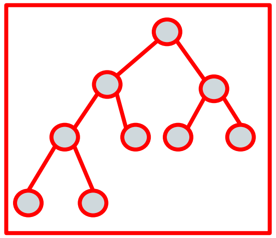




$$\mu(x) + \kappa\sigma(x)$$

Bayesian-based Method - FLASH



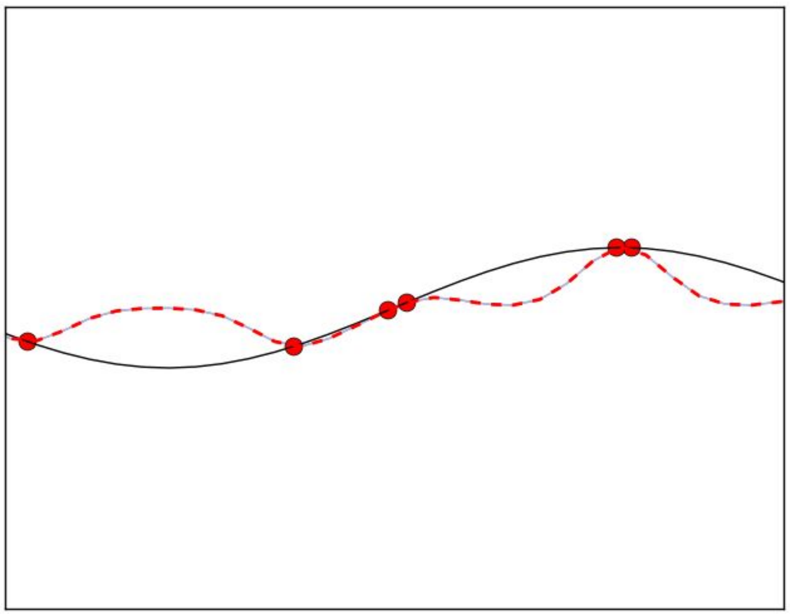


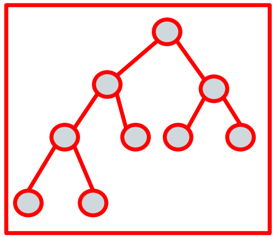
$$\mu(x) + \kappa\sigma(x)$$

2013 2015 2017 Future

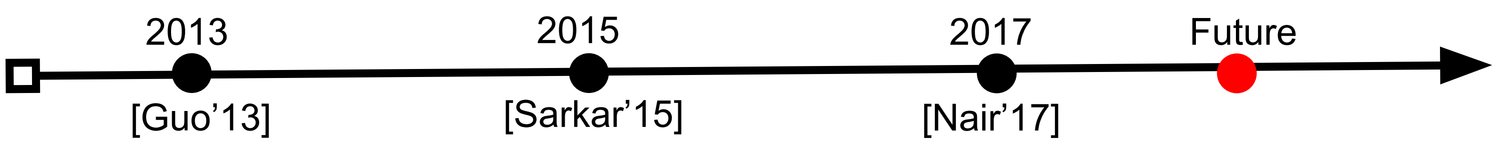
[Guo'13] [Sarkar'15] [Nair'17]

Bayesian-based Method - FLASH

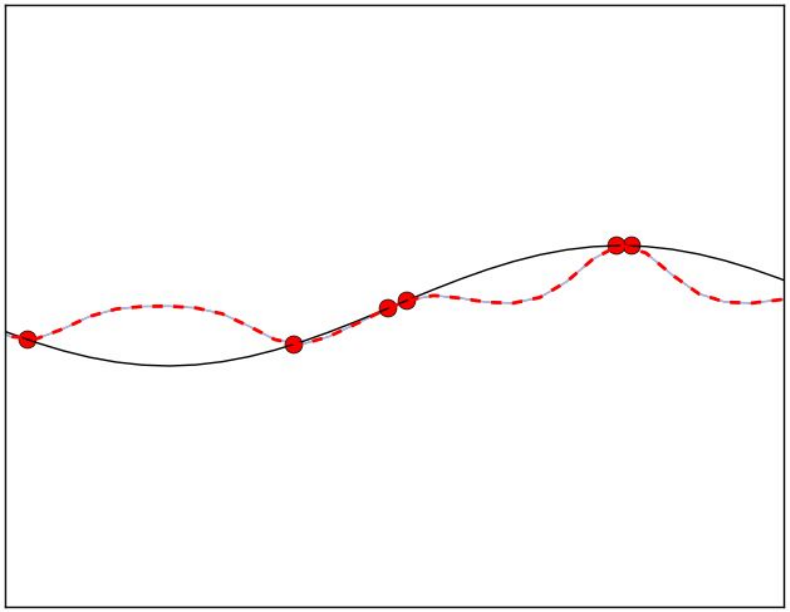


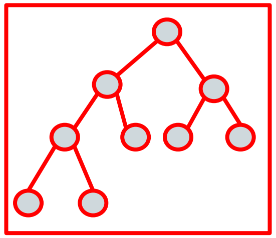


$$\cancel{\mu(x) + \kappa\sigma(x)}$$



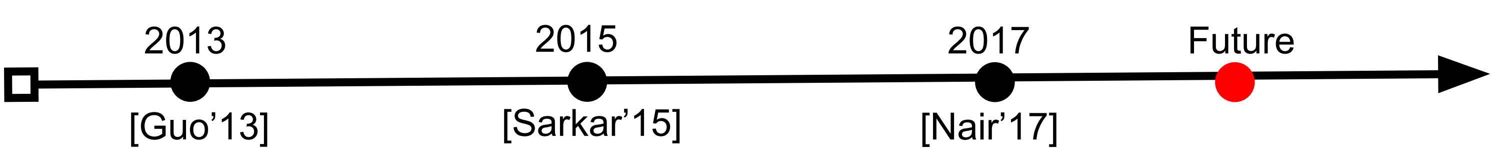
Bayesian-based Method - FLASH



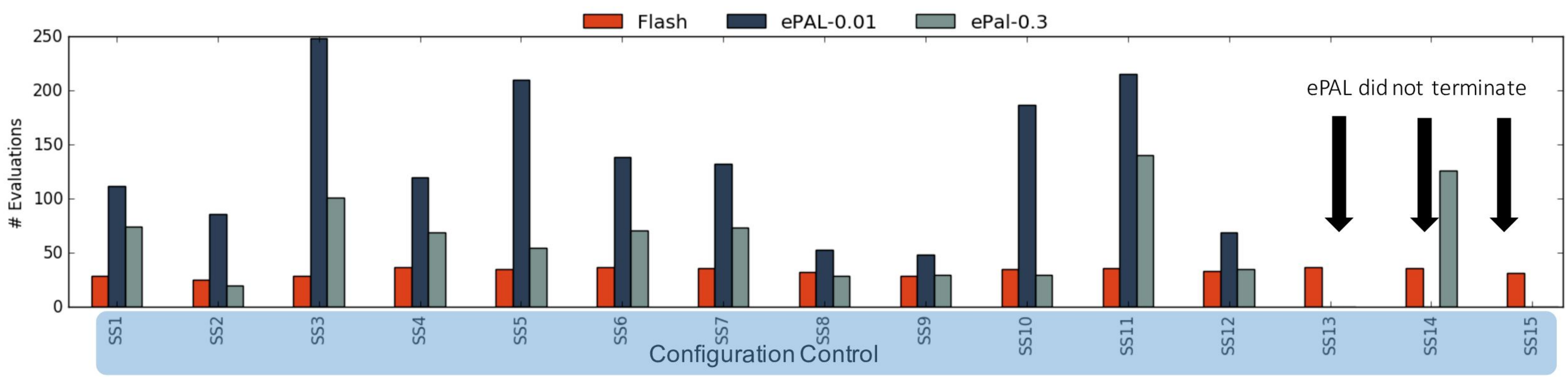


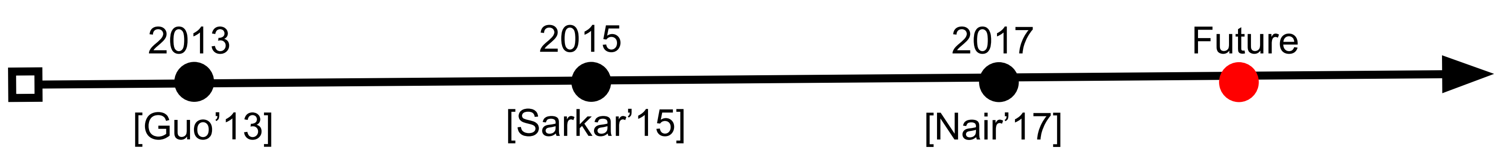
$$\mu(x)$$

$$\cancel{\mu(x) + \kappa\sigma(x)}$$



Bayesian-based Method - FLASH





Bayesian-based Method - FLASH

```
ipsccp-0
|       iv_users-0
|       |       sccp-0
|       |       |       print_used_types-0
|       |       |       |       jump_threading-0
|       |       |       |       |       time_passes-0
|       |       |       |       |       |       instcombine-0 (16)
|       |       |       |       |       |       instcombine-1 (12)
|       |       |       |       |       time_passes-1 (10)
|       |       |       |       jump_threading-1
|       |       |       |       |       simplifycfg-0 (32)
|       |       |       |       |       simplifycfg-1 (33)
|       |       |       print_used_types-1
|       |       |       |       |inline-0 (47)
|       |       |       |       |inline-1 (43)
|       |       sccp-1
|       |       |       print_used_types-0 (1)
|       |       |       |print_used_types-1 (7)
|       iv_users-1
|       |       sccp-0
|       |       |       print_used_types-0
|       |       |       |       jump_threading-0 (30)
|       |       |       |       jump_threading-1 (42)
|       |       |       print_used_types-1
|       |       |       |       inline-0 (56)
|       |       |       |       inline-1 (62)
|       |       sccp-1
|       |       |       instcombine-0 (26)
|       |       |       instcombine-1 (33)
ipsccp-1
|       sccp-0
|       |       print_used_types-0
|       |       |       iv_users-0
|       |       |       |       jump_threading-0 (50)
|       |       |       |       jump_threading-1
|       |       |       |       |       instcombine-0 (53)
|       |       |       |       |       instcombine-1 (54)
|       |       |       iv_users-1
|       |       |       |       simplifycfg-0 (59.5)
|       |       |       |       simplifycfg-1
|       |       |       |       |       time_passes-0 (63)
|       |       |       |       |       time_passes-1 (66)
|       |       print_used_types-1
|       |       |       iv_users-0
|       |       |       |       time_passes-0 (69)
|       |       |       |       time_passes-1
|       |       |       |       |       instcombine-0 (73)
|       |       |       |       |       instcombine-1 (71)
|       |       |       iv_users-1
|       |       |       |       gvn-0 (76)
|       |       |       |       gvn-1 (79)
|       sccp-1
|       |       print_used_types-0
|       |       |       jump_threading-0 (3)
|       |       |       jump_threading-1 (16)
|       |       print_used_types-1
|       |       |       iv_users-0 (35)
|       |       |       iv_users-1 (50)
```

Bayesian-based Method - FLASH

```
ipsccp=0
|   iv_users=0
|   |   sccp=0
|   |   |   print_used_types=0
|   |   |   |   jump_threading=0
|   |   |   |   |   time_passes=0
|   |   |   |   |   |   instcombine=0 (16)
|   |   |   |   |   |   instcombine=1 (12)
|   |   |   |   |   time_passes=1 (10)
|   |   |   |   jump_threading=1
|   |   |   |   |   simplifycfg=0 (32)
|   |   |   |   |   simplifycfg=1 (33)
|   |   |   print_used_types=1
|   |   |   |   |inline=0 (47)
|   |   |   |   |inline=1 (43)
|   |   sccp=1
|   |   |   print_used_types=0 (1)
|   |   |   |print_used_types=1 (7)
|   iv_users=1
|   |   sccp=0
|   |   |   print_used_types=0
|   |   |   |   jump_threading=0 (30)
|   |   |   |   jump_threading=1 (42)
|   |   |   print_used_types=1
|   |   |   |   inline=0 (56)
|   |   |   |   inline=1 (62)
|   |   sccp=1
|   |   |   instcombine=0 (26)
|   |   |   instcombine=1 (33)
ipsccp=1
|   sccp=0
|   |   print_used_types=0
|   |   |   iv_users=0
|   |   |   |   jump_threading=0 (50)
|   |   |   |   jump_threading=1
|   |   |   |   |   instcombine=0 (53)
|   |   |   |   |   instcombine=1 (54)
|   |   |   iv_users=1
|   |   |   |   simplifycfg=0 (59.5)
|   |   |   |   simplifycfg=1
|   |   |   |   |   time_passes=0 (63)
|   |   |   |   |   time_passes=1 (66)
|   |   print_used_types=1
|   |   |   iv_users=0
|   |   |   |   time_passes=0 (69)
|   |   |   |   time_passes=1
|   |   |   |   |   instcombine=0 (73)
|   |   |   |   |   instcombine=1 (71)
|   |   |   iv_users=1
|   |   |   |   gvn=0 (76)
|   |   |   |   gvn=1 (79)
|   sccp=1
|   |   print_used_types=0
|   |   |   jump_threading=0 (3)
|   |   |   jump_threading=1 (16)
|   |   print_used_types=1
|   |   |   iv_users=0 (35)
|   |   |   iv_users=1 (50)
```

```
sccp=0
|   print_used_types=0
|   |   ipsccp=0 (6.5)
|   |   ipsccp=1
|   |   |   x[10]=0 (12)
|   |   |   x[10]=1 (19)
|   print_used_types=1
|   |   ipsccp=0 (14)
|   |   ipsccp=1
|   |   |   time_passes=0  (24)
|   |   |   time_passes=1
|   |   |   |   jump_threading=0 (28)
|   |   |   |   jump_threading=1 (31)
sccp=1
|   ipsccp=0 (1.5)
|   ipsccp=1 (7.5)
```

Bayesian-based Method - FLASH

```
ipsccp=0
|       iv_users=0
|       |       sccp=0
|       |       |       print_used_types=0
|       |       |       |       jump_threading=0
|       |       |       |       |       time_passes=0
|       |       |       |       |       |       instcombine=0 (16)
|       |       |       |       |       |       instcombine=1 (12)
|       |       |       |       |       time_passes=1 (10)
|       |       |       |       jump_threading=1
|       |       |       |       |       simplifycfg=0 (32)
|       |       |       |       |       simplifycfg=1 (33)
|       |       |       print_used_types=1
|       |       |       |       |inline=0 (47)
|       |       |       |       |inline=1 (43)
|       |       sccp=1
|       |       |       print_used_types=0 (1)
|       |       |       |print_used_types=1 (7)
|       iv_users=1
|       |       sccp=0
|       |       |       print_used_types=0
|       |       |       |       jump_threading=0 (30)
|       |       |       |       jump_threading=1 (42)
|       |       |       print_used_types=1
|       |       |       |       inline=0 (56)
|       |       |       |       inline=1 (62)
|       |       sccp=1
|       |       |       instcombine=0 (26)
|       |       |       instcombine=1 (33)
ipsccp=1
|       sccp=0
|       |       print_used_types=0
|       |       |       iv_users=0
|       |       |       |       jump_threading=0 (50)
|       |       |       |       jump_threading=1
|       |       |       |       |       instcombine=0 (53)
|       |       |       |       |       instcombine=1 (54)
|       |       |       iv_users=1
|       |       |       |       simplifycfg=0 (59.5)
|       |       |       |       simplifycfg=1
|       |       |       |       |       time_passes=0 (63)
|       |       |       |       |       time_passes=1 (66)
|       |       print_used_types=1
|       |       |       iv_users=0
|       |       |       |       time_passes=0 (69)
|       |       |       |       time_passes=1
|       |       |       |       |       instcombine=0 (73)
|       |       |       |       |       instcombine=1 (71)
|       |       |       iv_users=1
|       |       |       |       gvn=0 (76)
|       |       |       |       gvn=1 (79)
|       sccp=1
|       |       print_used_types=0
|       |       |       jump_threading=0 (3)
|       |       |       jump_threading=1 (16)
|       |       print_used_types=1
|       |       |       iv_users=0 (35)
|       |       |       iv_users=1 (50)
```

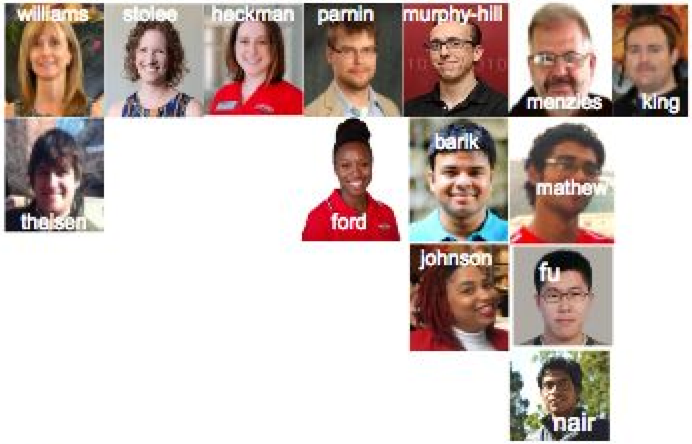
```
sccp=0
|       print_used_types=0
|       |       ipsccp=0 (6.5)
|       |       ipsccp=1
|       |       |       x[10]=0 (12)
|       |       |       x[10]=1 (19)
|       print_used_types=1
|       |       ipsccp=0 (14)
|       |       ipsccp=1
|       |       |       time_passes=0   (24)
|       |       |       time_passes=1
|       |       |       |       jump_threading=0 (28)
|       |       |       |       jump_threading=1 (31)
sccp=1
|       ipsccp=0 (1.5)
|       ipsccp=1 (7.5)
```

- ✓ Fast
- ✓ Effective
- ✓ Comprehensible

Conclusion

- ML Algorithms are not a black box
  - How to use Decision Tree in Planning?
  - Can I explain the results to the Decision Makers?
- Lazy is good
  - Only do what is required
  - Optimization does not require an accurate model
- Easy over Hard
  - Try simplest first
  - Tuning SVM outperforms DL

NC STATE UNIVERSITY



vivekaxl@gmail.com

@vivekaxl

vivekaxl.com

Expected Graduation: **May 2018**

Data Science, Performance Optimization, Evolutionary Algorithms

Resources

Rank-based Method: http://tiny.cc/wnheny
Flash: http://tiny.cc/hoheny
ePAL: http://www.spiral.net/software/pal.html
Bayesian Optimization: https://youtu.be/vz3D36VXefI

"Look for me...beneath the tree...North!"

—Three-eyed raven