# An Uncertainty-Aware Approach to Optimal Configuration of Stream Processing Systems

Pooyan Jamshidi
(joint work work Giuliano Casale)
Imperial College London
p.jamshidi@imperial.ac.uk

University of Bern
1st Nov 2016

Imperial College London

# Motivation

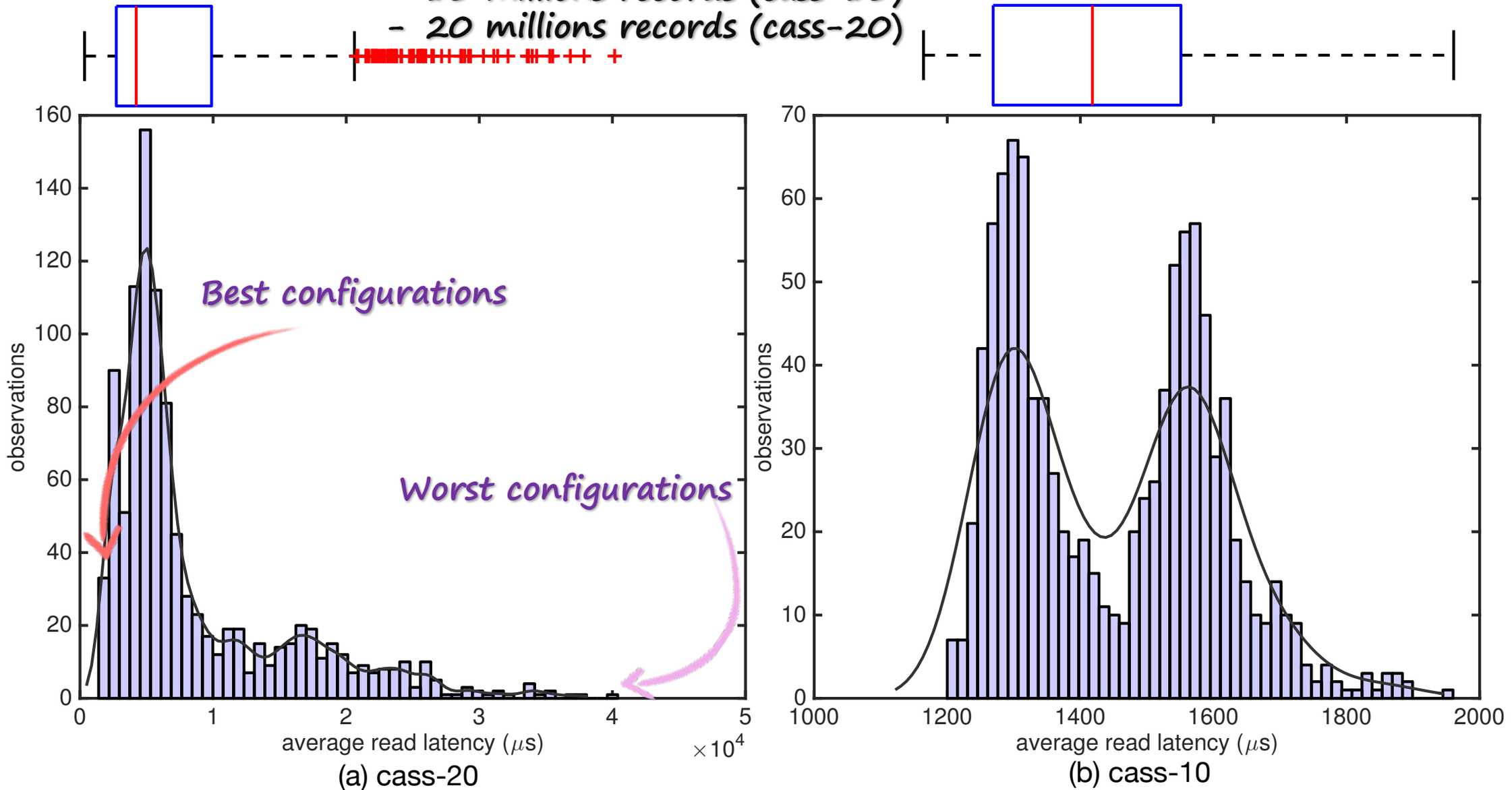
1- Many different Parameters =>
- large state space
- interactions

2- Defaults are typically used =>
- poor performance

```
102
103 drpc.port: 3772
104 drpc.worker.threads: 64
105 drpc.max_buffer_size: 1048576
106 drpc.queue.size: 128
107 drpc.invocations.port: 3773
108 drpc.invocations.threads: 64
109 drpc.request.timeout.secs: 600
110 drpc.childopts: "-Xmx768m"
111 drpc.http.port: 3774
112 drpc.https.port: -1
113 drpc.https.keystore.password: ""
114 drpc.https.keystore.type: "JKS"
115 drpc.http.creds.plugin: org.apache.storm.security.auth.DefaultHttpCredentialsPlugi
116 drpc.authorizer.acl.filename: "drpc-auth-acl.yaml"
117 drpc.authorizer.acl.strict: false
118
119 transactional.zookeeper.root: "/transactional"
120 transactional.zookeeper.servers: null
121 transactional.zookeeper.port: null
122
123 ## blobstore configs
124 supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
125 supervisor.blobstore.download.thread.count: 5
126 supervisor.blobstore.download.max_retries: 3
127 supervisor.localizer.cache.target.size.mb: 10240
128 supervisor.localizer.cleanup.interval.ms: 600000
129
```

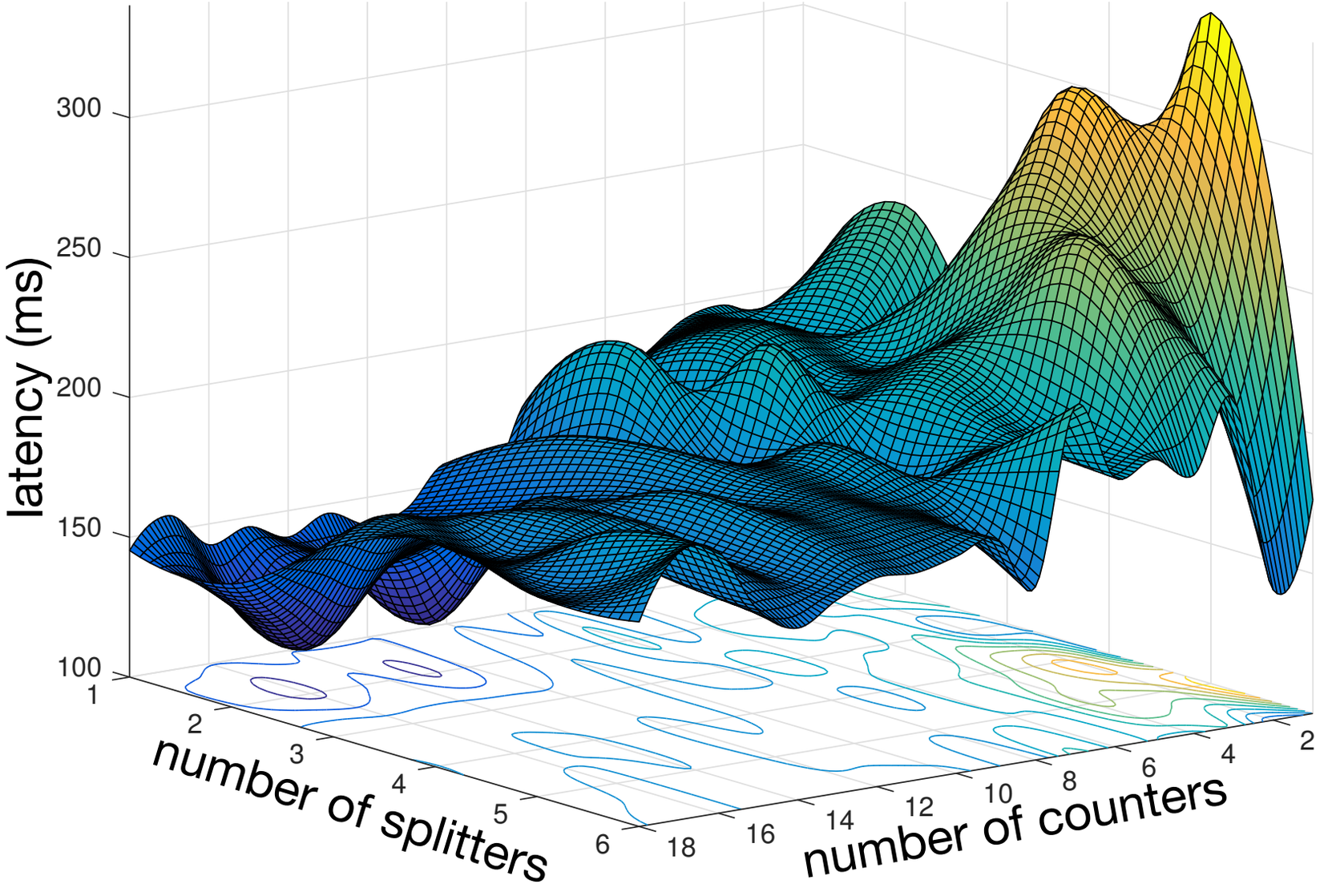
# Motivation

Experiments on
Apache Cassandra:
- 6 parameters, 1024 configurations
- Average read latency
- 10 millions records (cass-10)
- 20 millions records (cass-20)



# Motivation (Apache Storm)

*In our experiments we observed improvement up to 100%*



# Goal

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}\in\mathbb{X}} f(\boldsymbol{x})$$

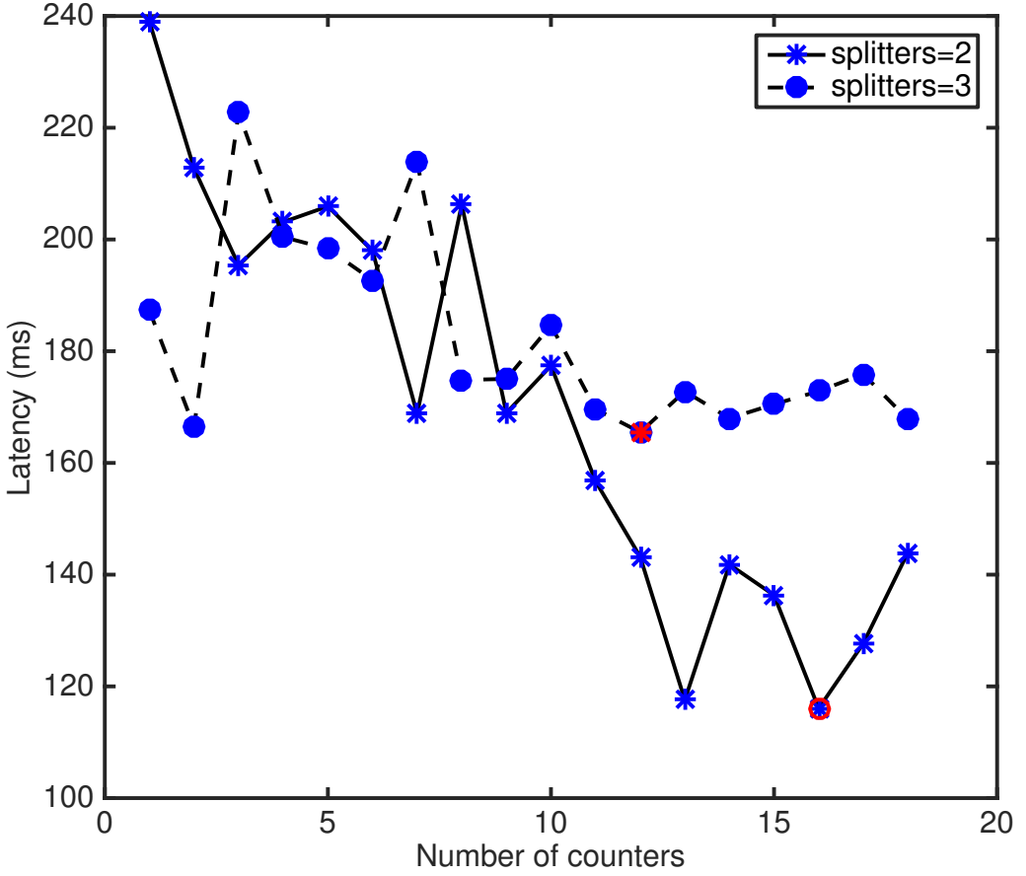
$$\mathbb{X} = Dom(X_1) \times \cdots \times Dom(X_d)$$ Configuration space

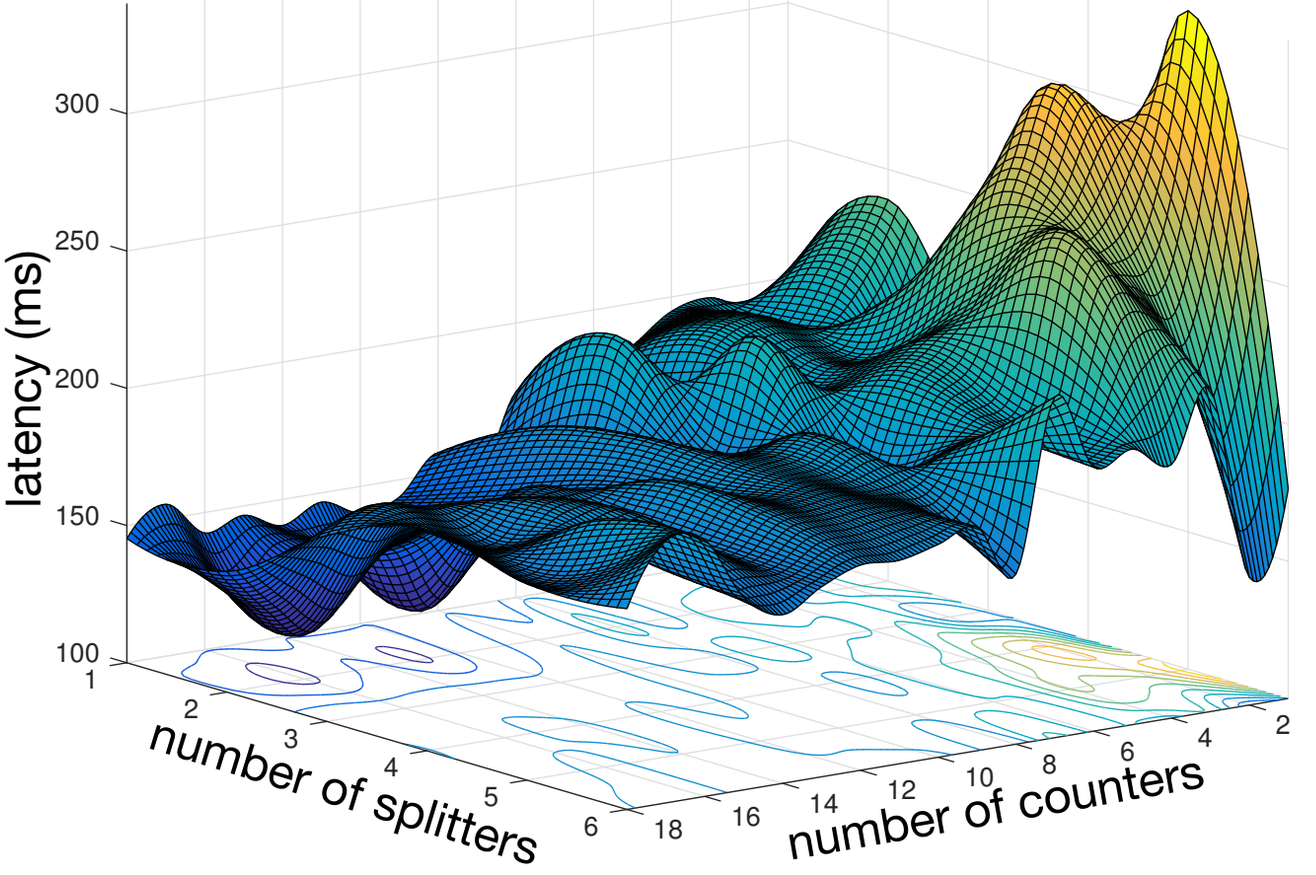
$$y_i = f(\boldsymbol{x}_i), \boldsymbol{x}_i \subset \mathbb{X}$$ Partially known

$$y_i = f(\boldsymbol{x}_i) + \epsilon$$ Measurements subject to noise

# Non-linear interactions

*Response surface is:*
- *Non-linear*
- *Non convex*
- *Multi-modal*

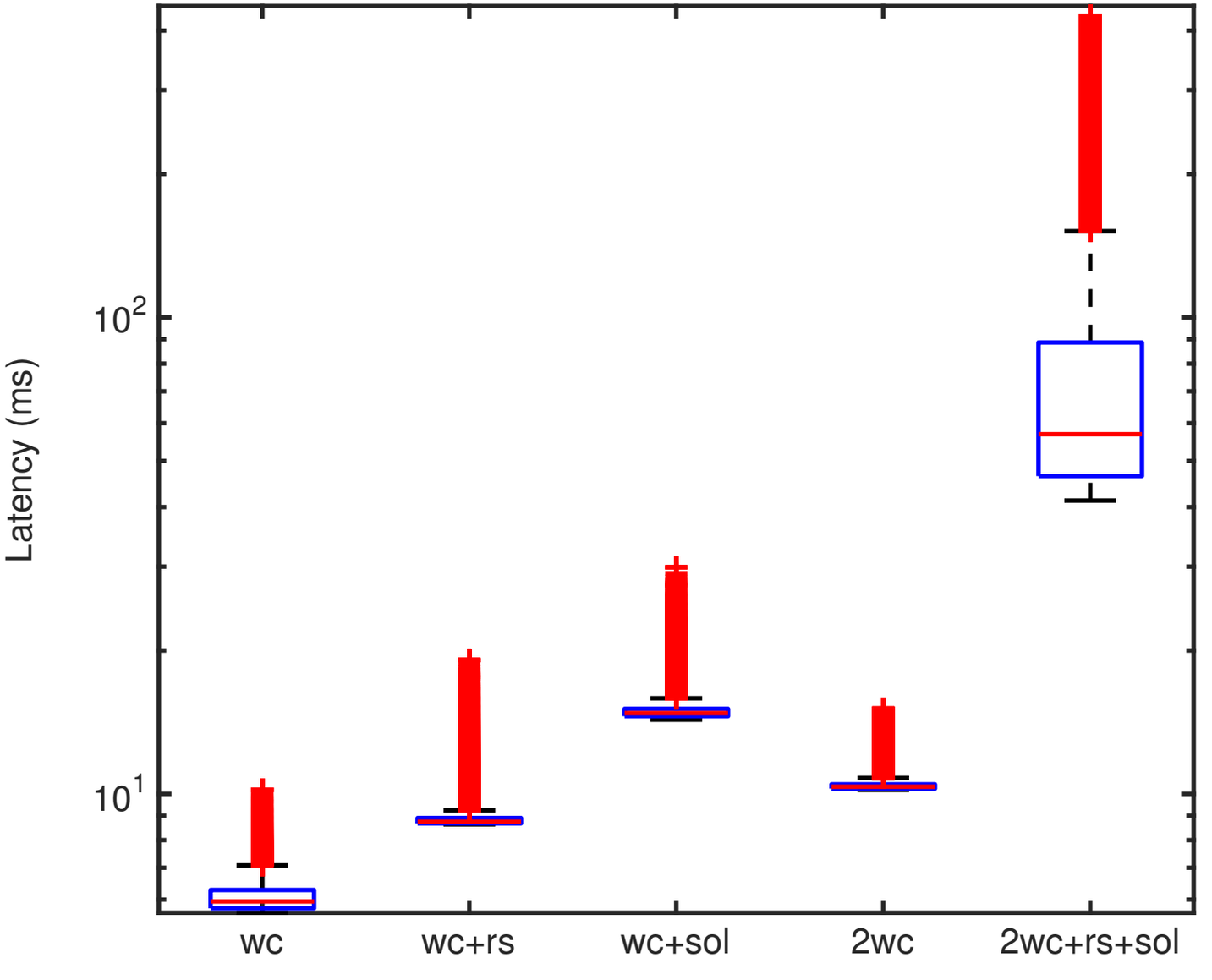


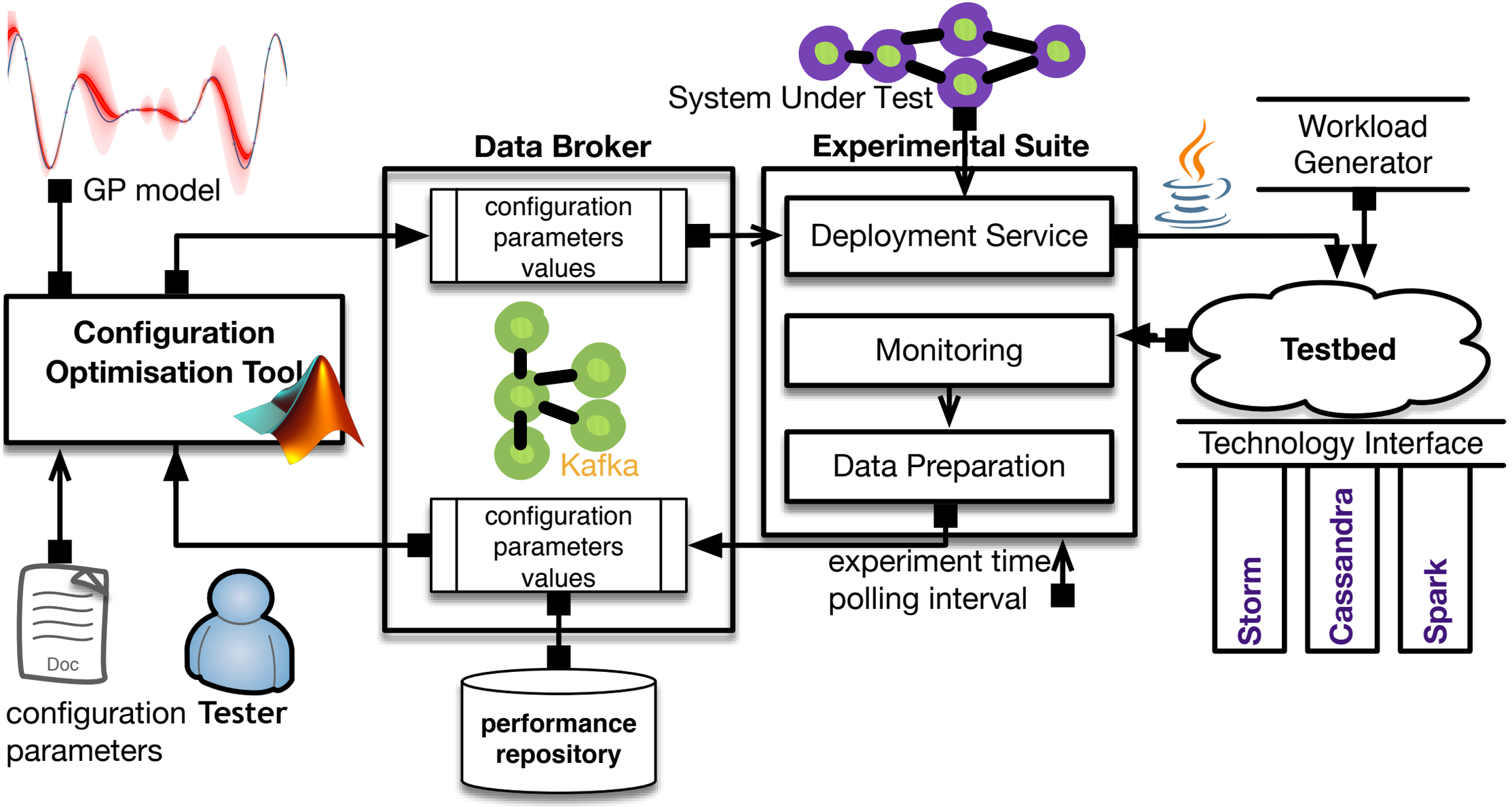
# The measurements are subject to variability

*The scale of measurement variability is different in different deployments (heteroscedastic noise)*

$$y_i = f(\boldsymbol{x}_i) + \epsilon_i$$



# BO4CO architecture

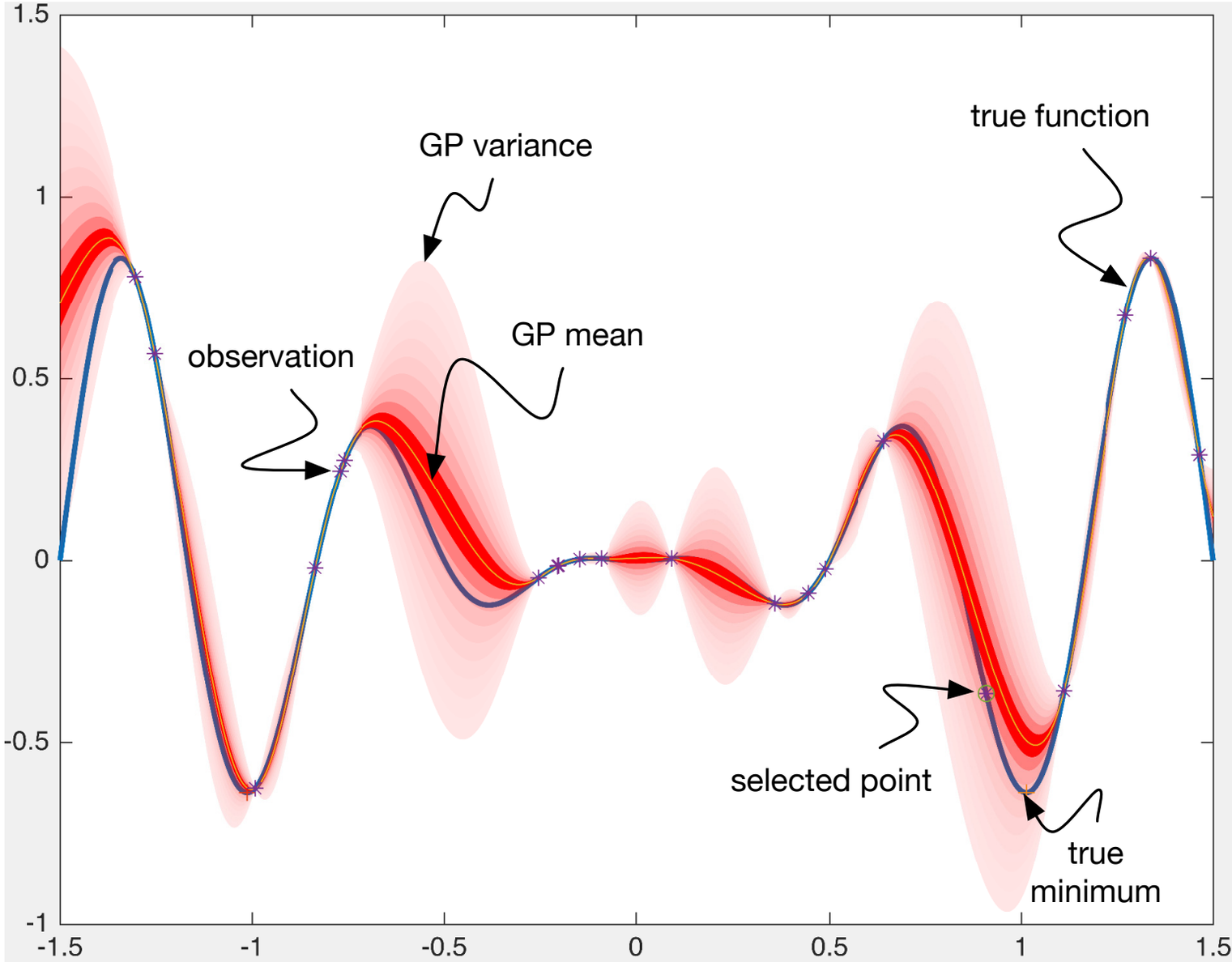


# GP for modeling blackbox response function

$$y = f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')),$$

$$\mu_t(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})^\intercal (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{\mu})$$
$$\sigma_t^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2 \boldsymbol{I} - \boldsymbol{k}(\boldsymbol{x})^\intercal (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k}(\boldsymbol{x})$$

**Motivations:**
1- mean estimates + variance
2- all computations are linear algebra
3- good estimations when few data

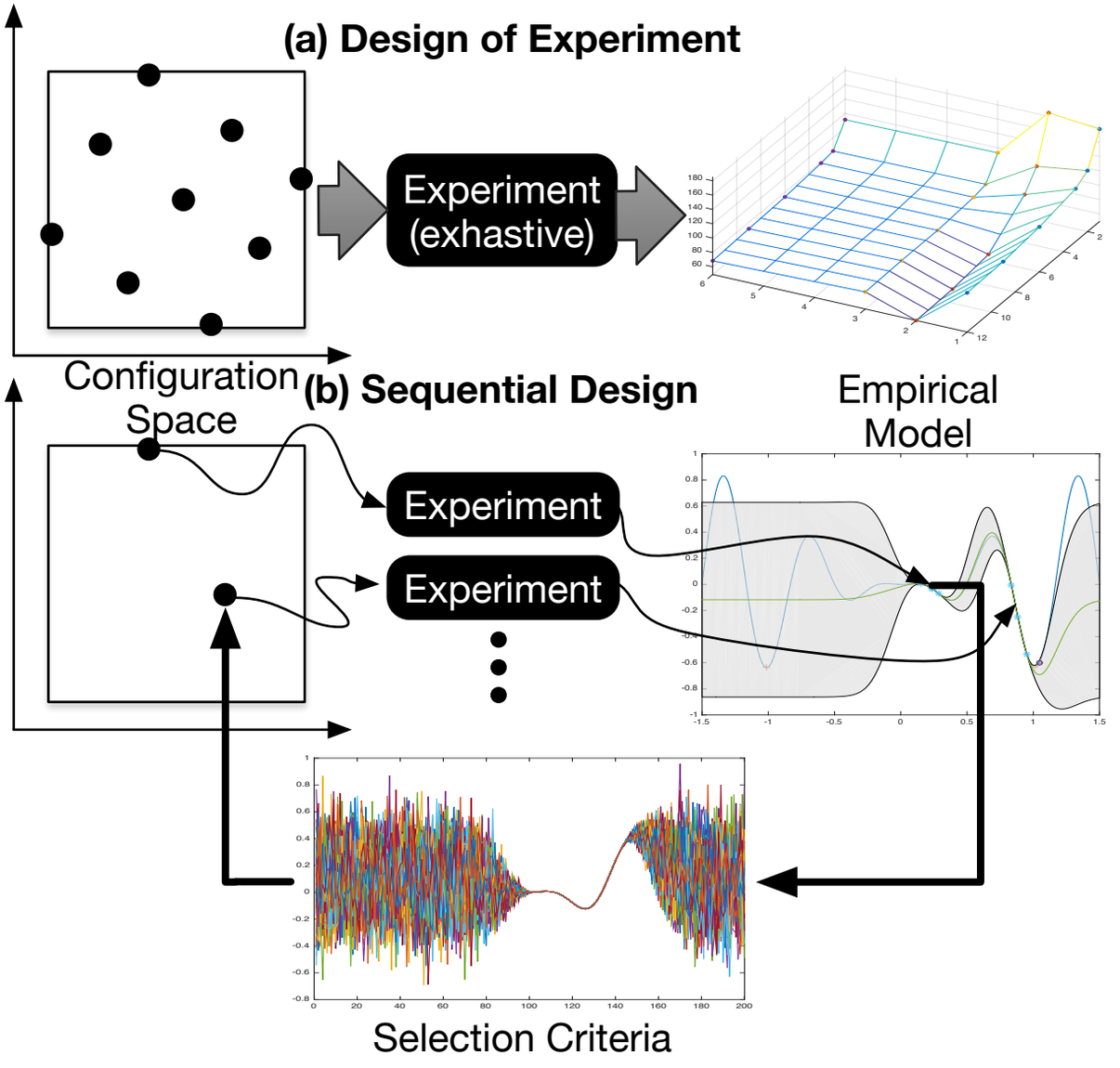


# Sparsity of Effects

- Correlation-based feature selector
- Merit is used to select subsets that are highly correlated with the response variable
- At most 2-3 parameters were strongly interacting with each other

| | Topol. | Parameters | Main factors | Merit | Size | Testbed |
|---|---|---|---|---|---|---|
| 1 | wc(6D) | 1-spouts, 2-max_spout, 3-spout_wait, 4-splitters, 5-counters, 6-netty_min_wait | {1, 2, 5} | 0.787 | 2880 | C1 |
| 2 | sol(6D) | 1-spouts, 2-max_spout, 3-top_level, 4-netty_min_wait, 5-message_size, 6-bolts | {1, 2, 3} | 0.447 | 2866 | C2 |
| 3 | rs(6D) | 1-spouts, 2-max_spout, 3-sorters, 4-emit_freq, 5-chunk_size, 6-message_size | {3} | 0.385 | 3840 | C3 |
| 4 | wc(3D) | 1-max_spout, 2-splitters, 3-counters | {1, 2} | 0.480 | 756 | C4 |
| 5 | wc(5D) | 1-spouts, 2-splitters, 3-counters, 4-buffer-size, 5-heap | {1} | 0.851 | 1080 | C5 |

Experiments on:
1. C1: OpenNebula (X)
2. C2: Amazon EC2 (Y)
3. C3: OpenNebula (3X)
4. C4: Amazon EC2 (2Y)
5. C5: Microsoft Azure (X)




**Algorithm 1 : BO4CO**

**Input:** Configuration space $\mathbb{X}$, Maximum budget $N_{max}$, Response function $f$, Kernel function $\boldsymbol{K}_\theta$, Hyper-parameters $\boldsymbol{\theta}$, Design sample size $n$, learning cycle $N_l$

**Output:** Optimal configurations $\boldsymbol{x}^*$ and learned model $\mathcal{M}$

choose an initial sparse design (*lhd*) to find an initial design samples $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$

obtain *performance measurements* of the initial design, $y_i \leftarrow f(\boldsymbol{x}_i) + \epsilon_i, \forall \boldsymbol{x}_i \in \mathcal{D}$

$\mathbb{S}_{1:n} \leftarrow \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n; t \leftarrow n+1$

$\mathcal{M}(\boldsymbol{x}|\mathbb{S}_{1:n}, \boldsymbol{\theta}) \leftarrow$ fit a $\mathcal{GP}$ model to the design ▷ Eq.(3)

**while** $t \leq N_{max}$ **do**

  if $(t \mod N_l = 0)$ $\boldsymbol{\theta} \leftarrow$ *learn* the kernel hyper-parameters by maximizing the likelihood

  find *next configuration* $\boldsymbol{x}_t$ by optimizing the selection criteria over the estimated response surface given the data, $\boldsymbol{x}_t \leftarrow \arg max_{\boldsymbol{x}} u(\boldsymbol{x}|\mathcal{M}, \mathbb{S}_{1:t-1})$ ▷ Eq.(9)

  obtain performance for the *new configuration* $\boldsymbol{x}_t$, $y_t \leftarrow f(\boldsymbol{x}_t) + \epsilon_t$

  Augment the configuration $\mathbb{S}_{1:t} = \{\mathbb{S}_{1:t-1}, (\boldsymbol{x}_t, y_t)\}$

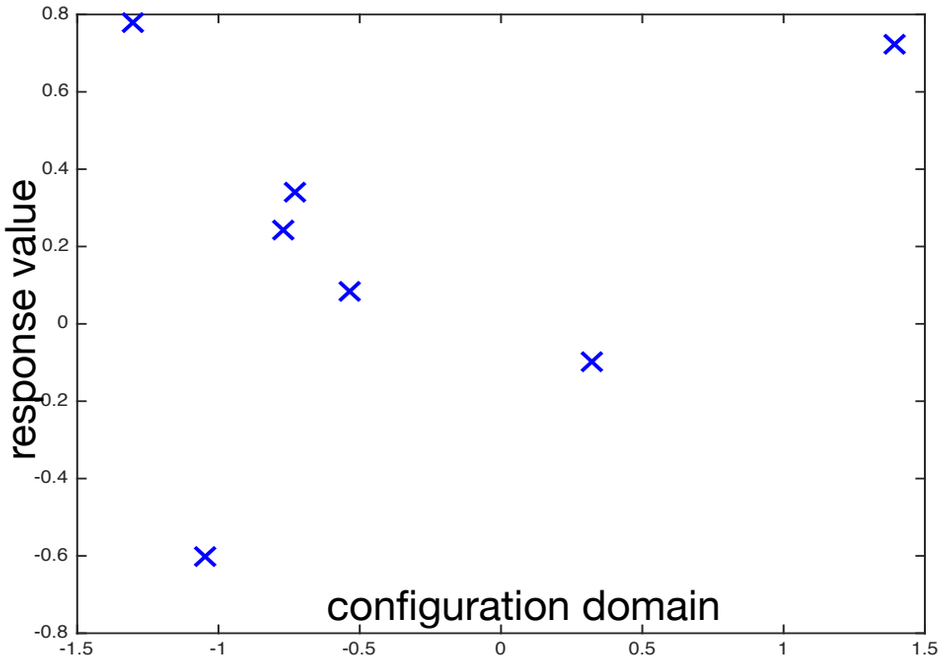
  $\mathcal{M}(\boldsymbol{x}|\mathbb{S}_{1:t}, \boldsymbol{\theta}) \leftarrow$ *re-fit* a new GP model ▷ Eq.(7)

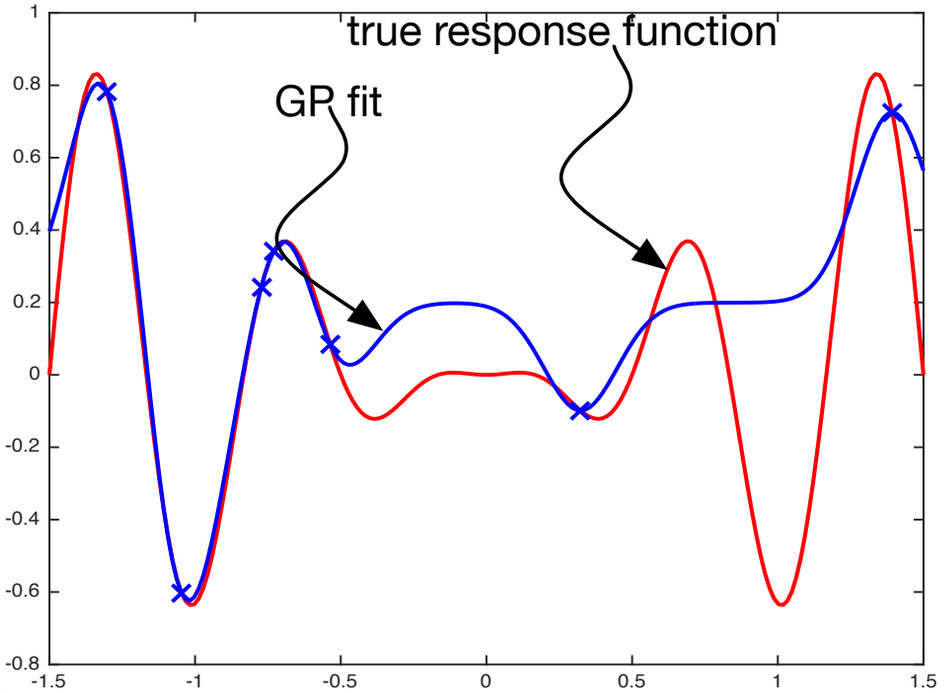
  $t \leftarrow t+1$

**end while**

$(\boldsymbol{x}^*, y^*) = \min \mathbb{S}_{1:N_{max}}$

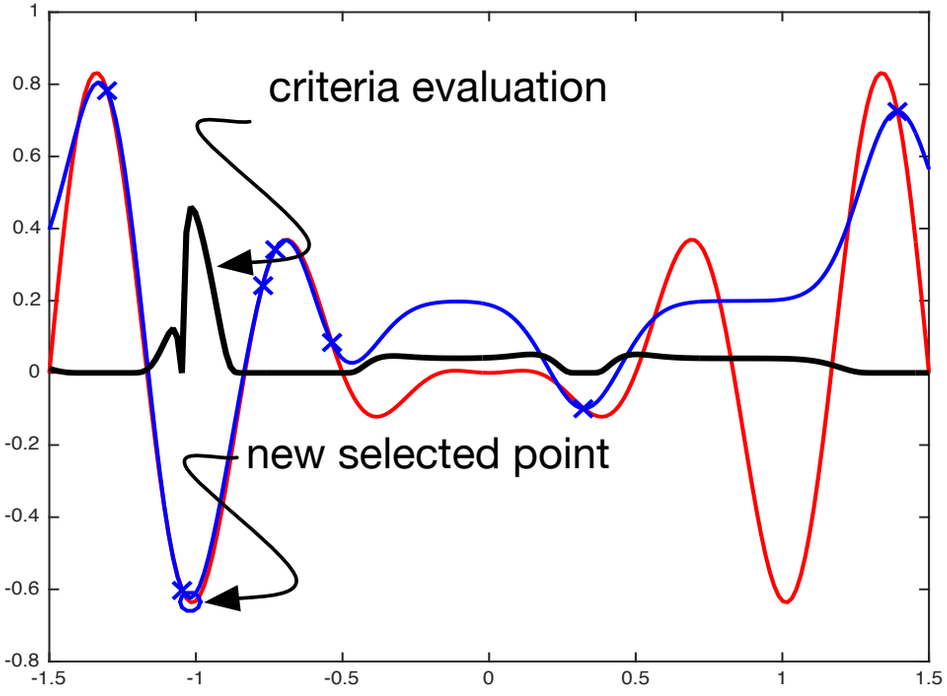
$\mathcal{M}(\boldsymbol{x})$

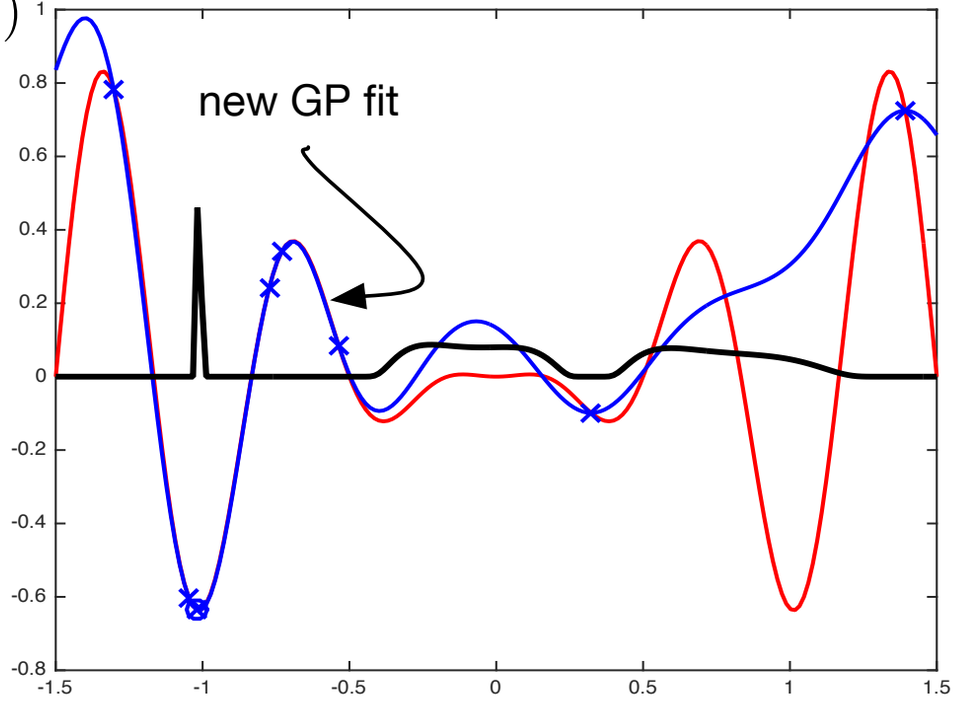


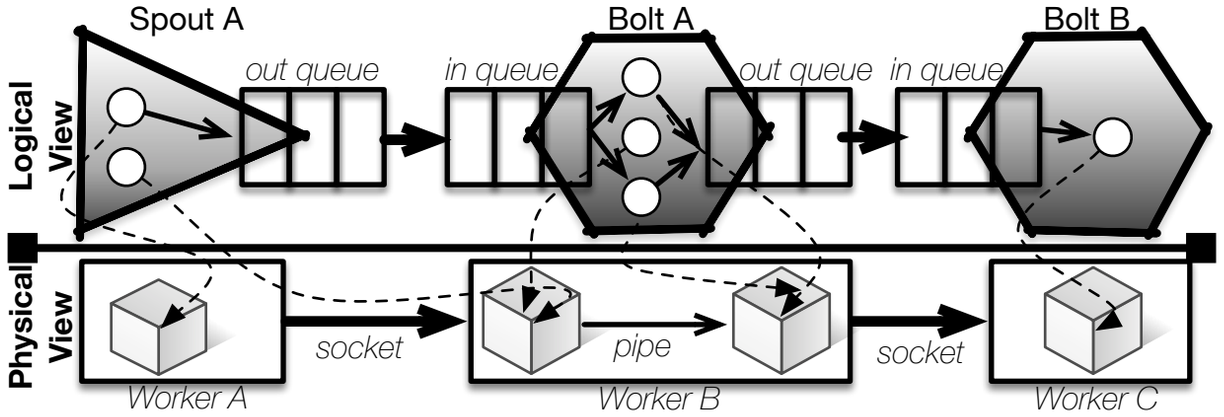

Acquisition function:

$$\boldsymbol{x}_{t+1} = \underset{\boldsymbol{x} \in \mathbb{X}}{\operatorname{argmax}}\, u(\boldsymbol{x} | \mathcal{M}, \mathbb{S}_{1:t})$$

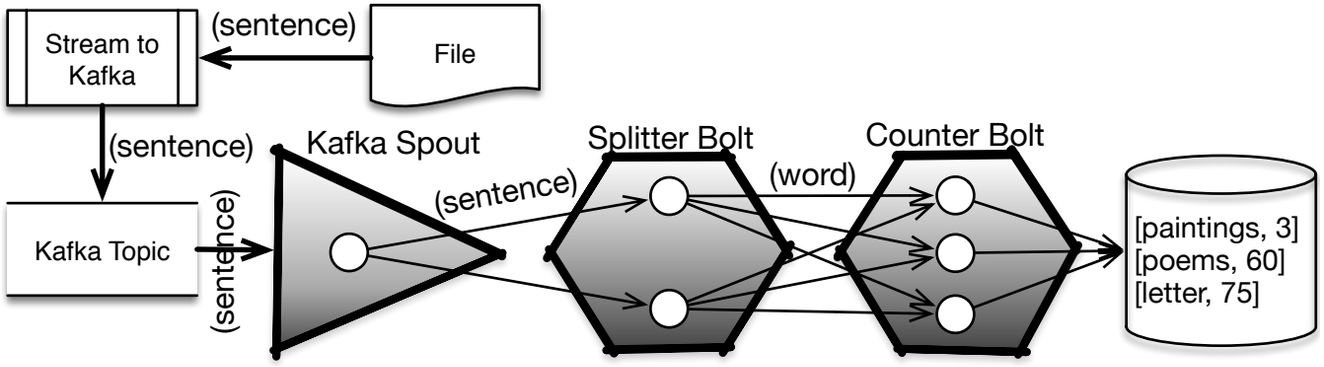


# Stream Processing Systems




**Storm Architecture**

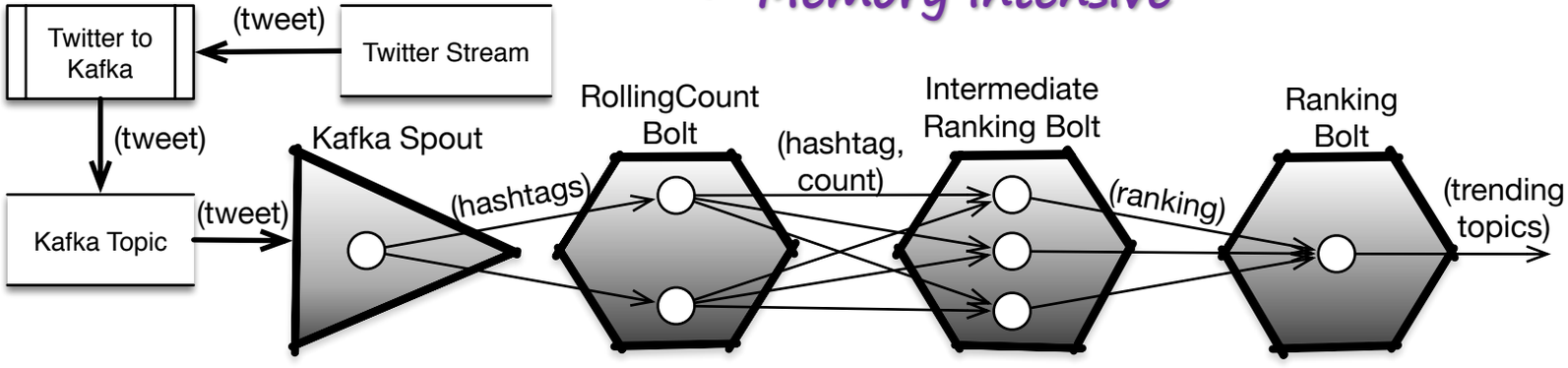



**Word Count Architecture**

- CPU intensive

**Rolling Sort Architecture**

- Memory intensive

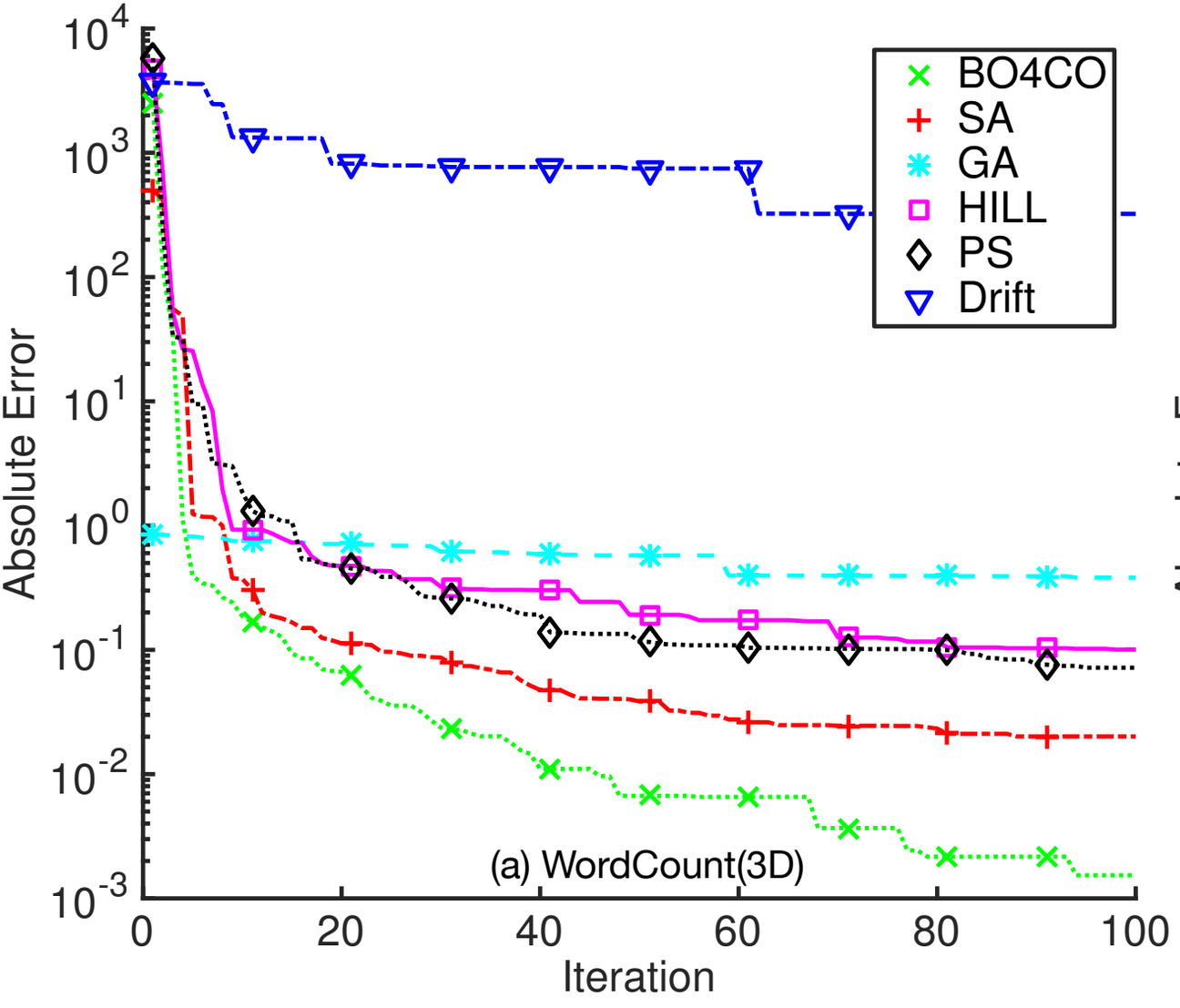


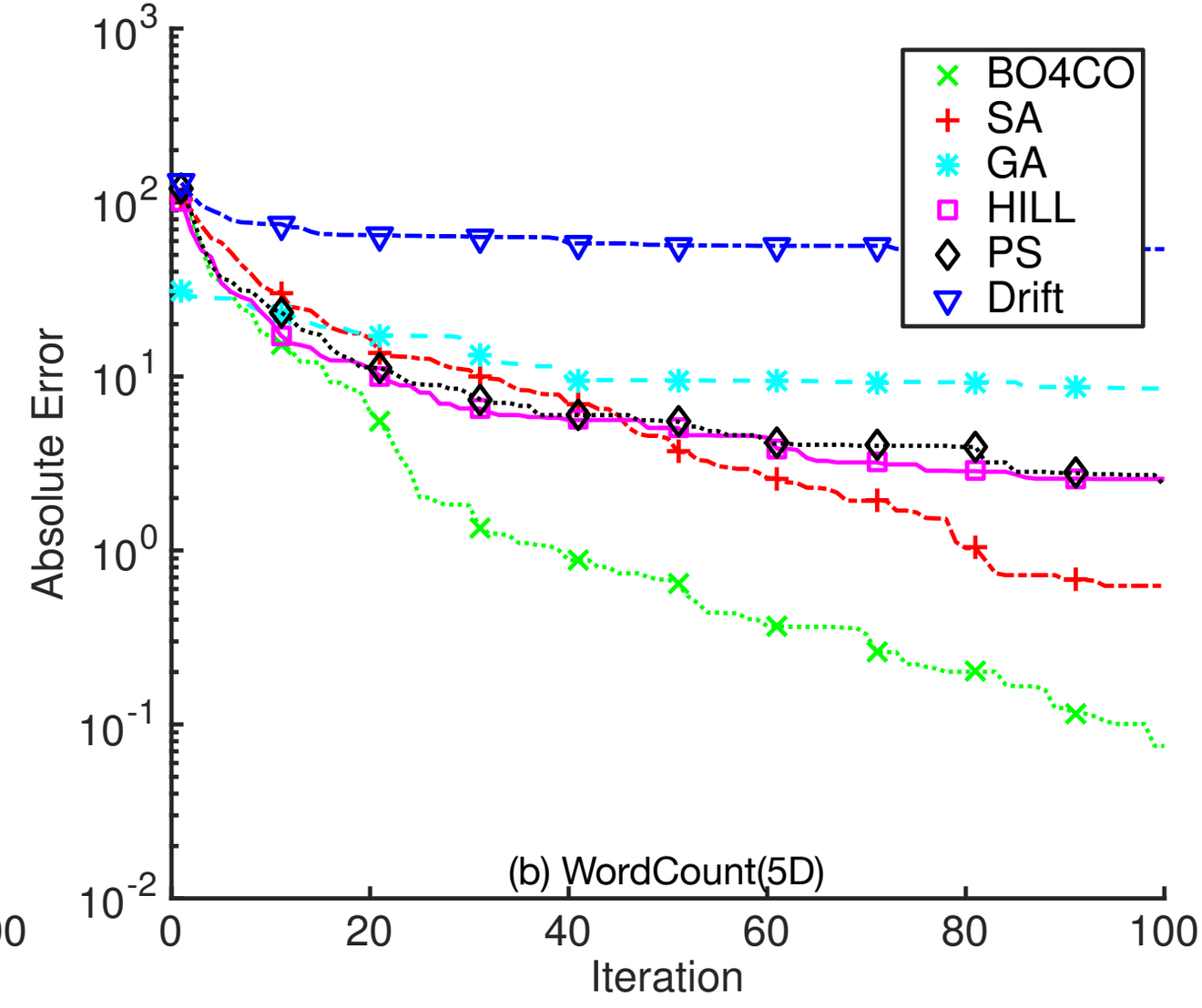

**Applications:**

- Fraud detection
- Trending topics

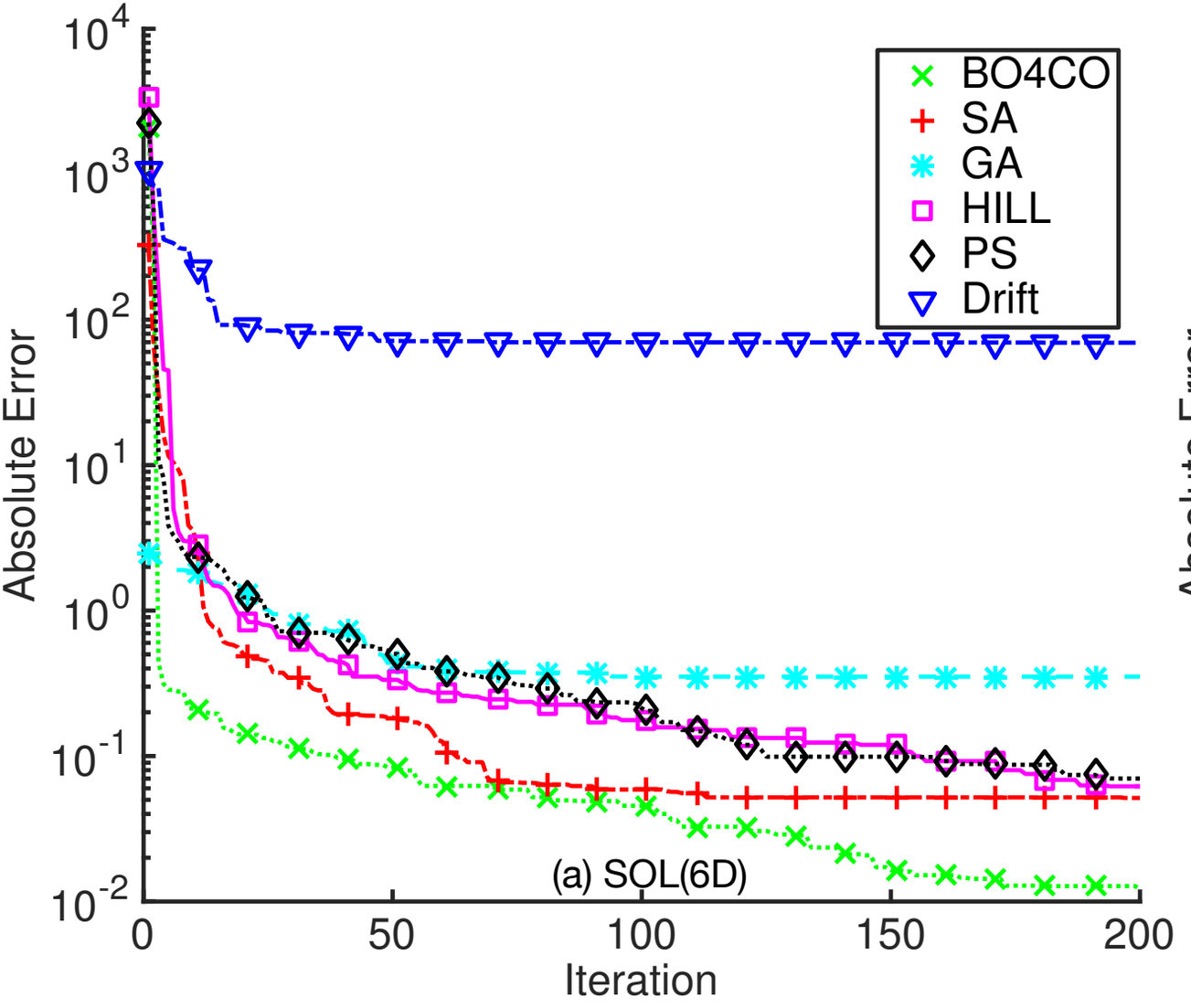
# Experimental results

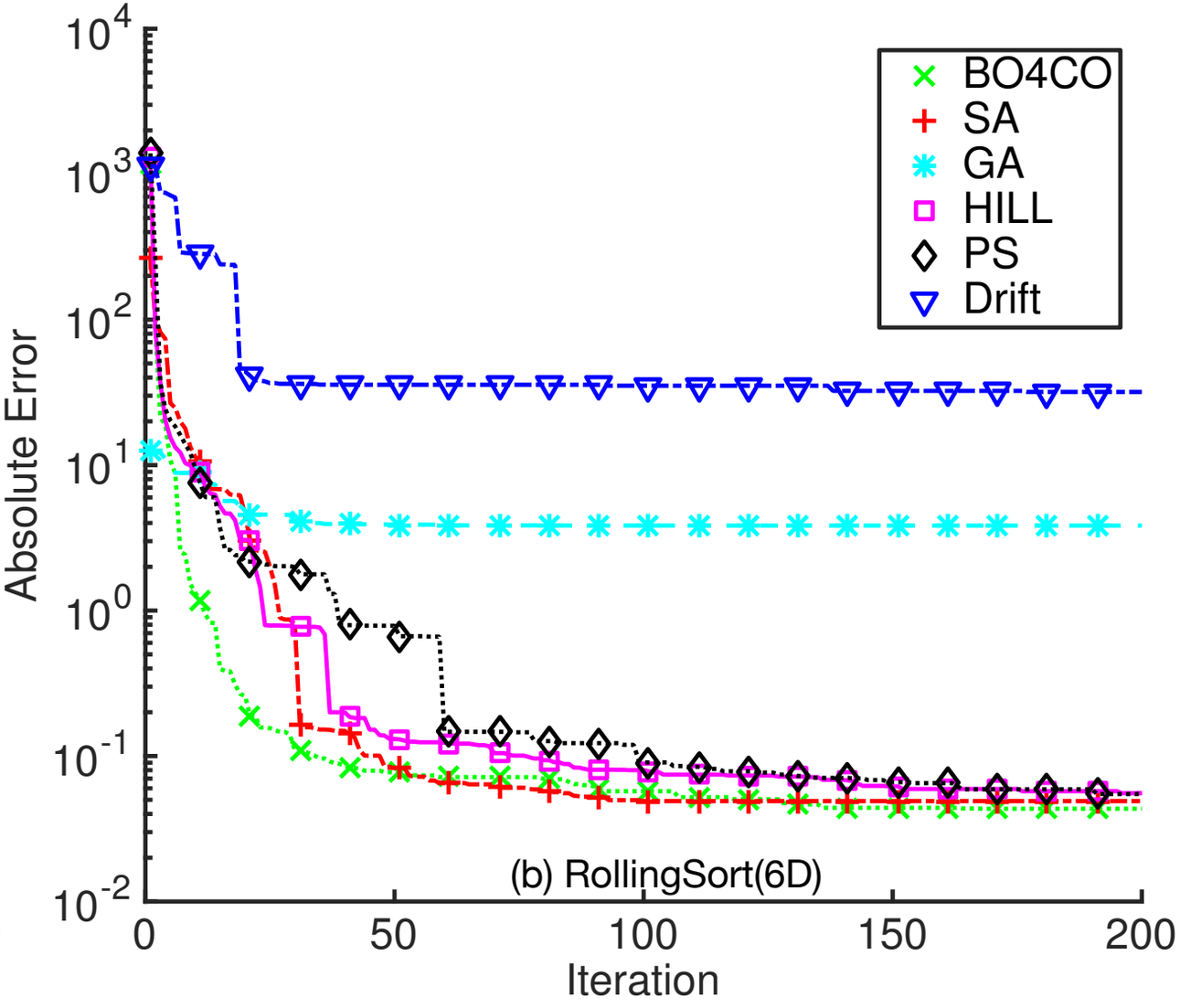
- 30 runs, report average performance
- Yes, we did full factorial measurements and we know where global min is…



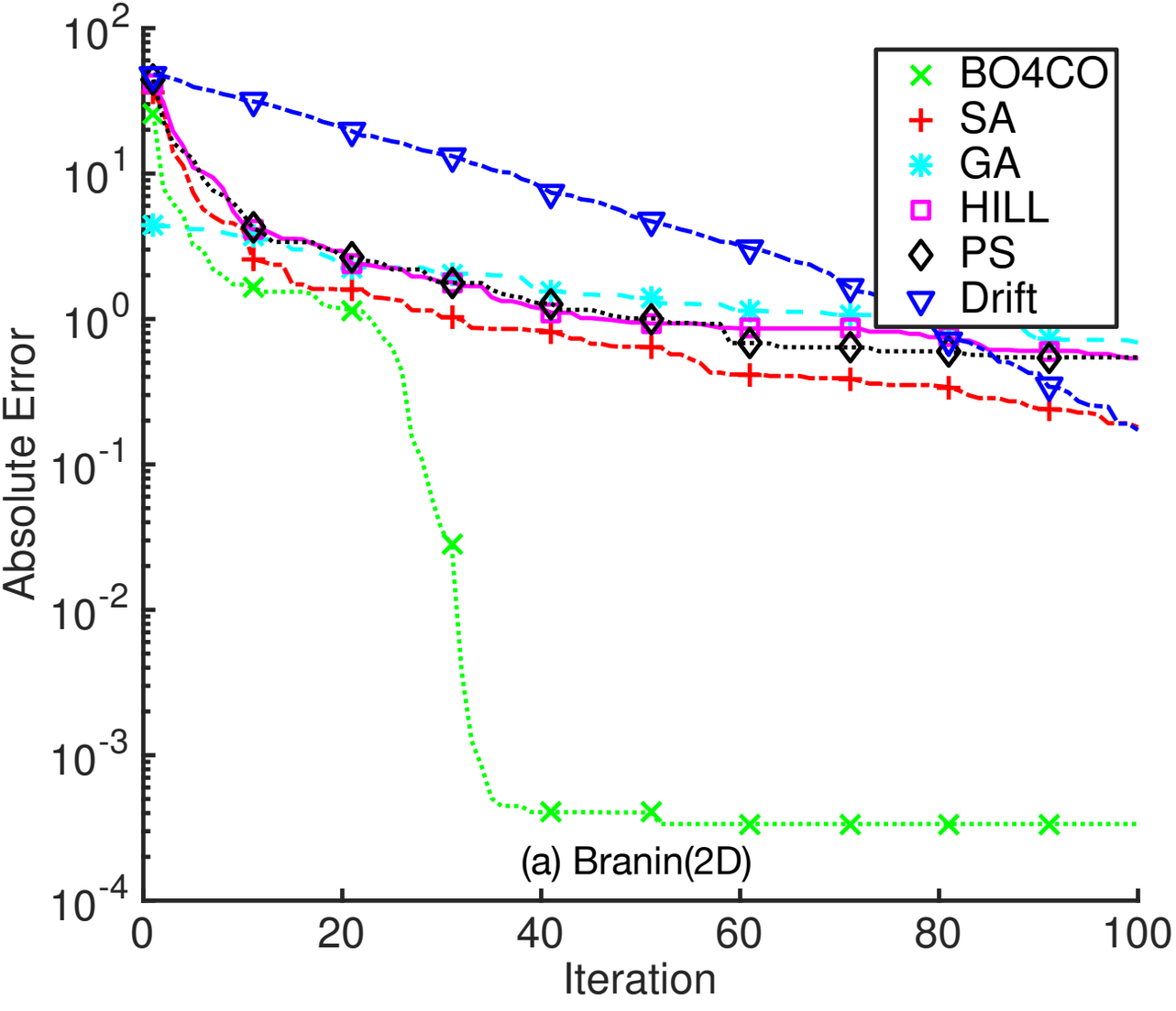



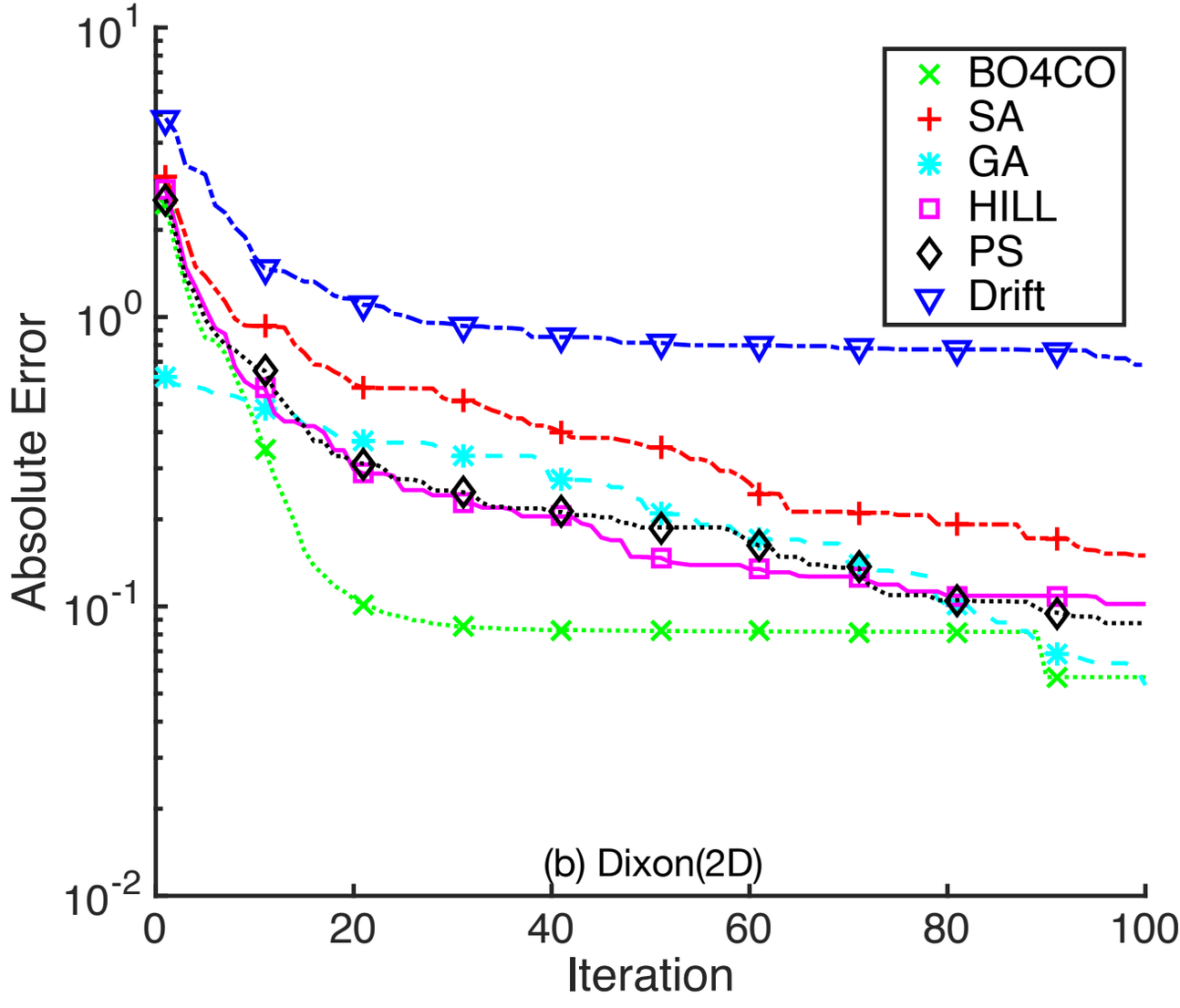
# Experimental results



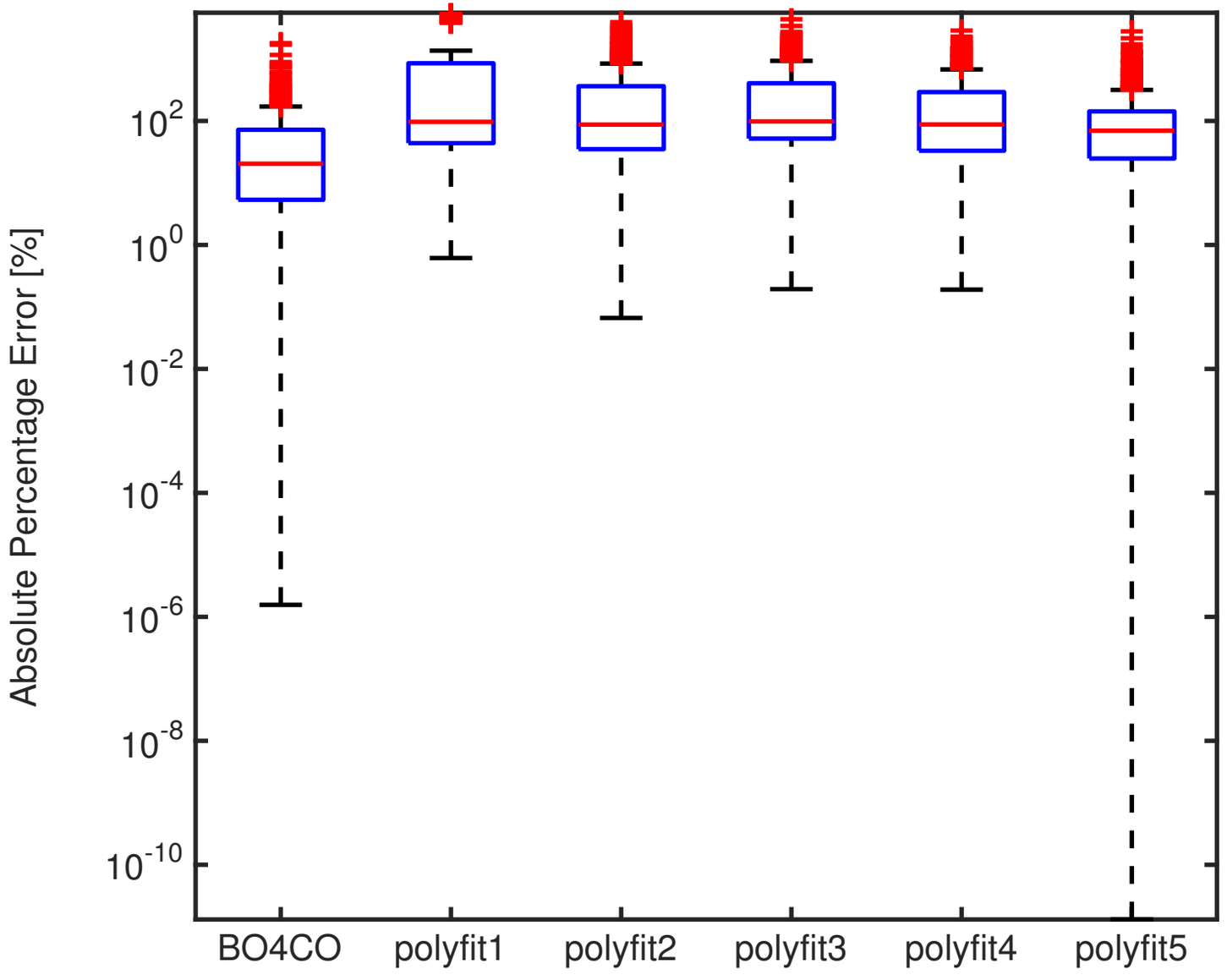



# Experimental results

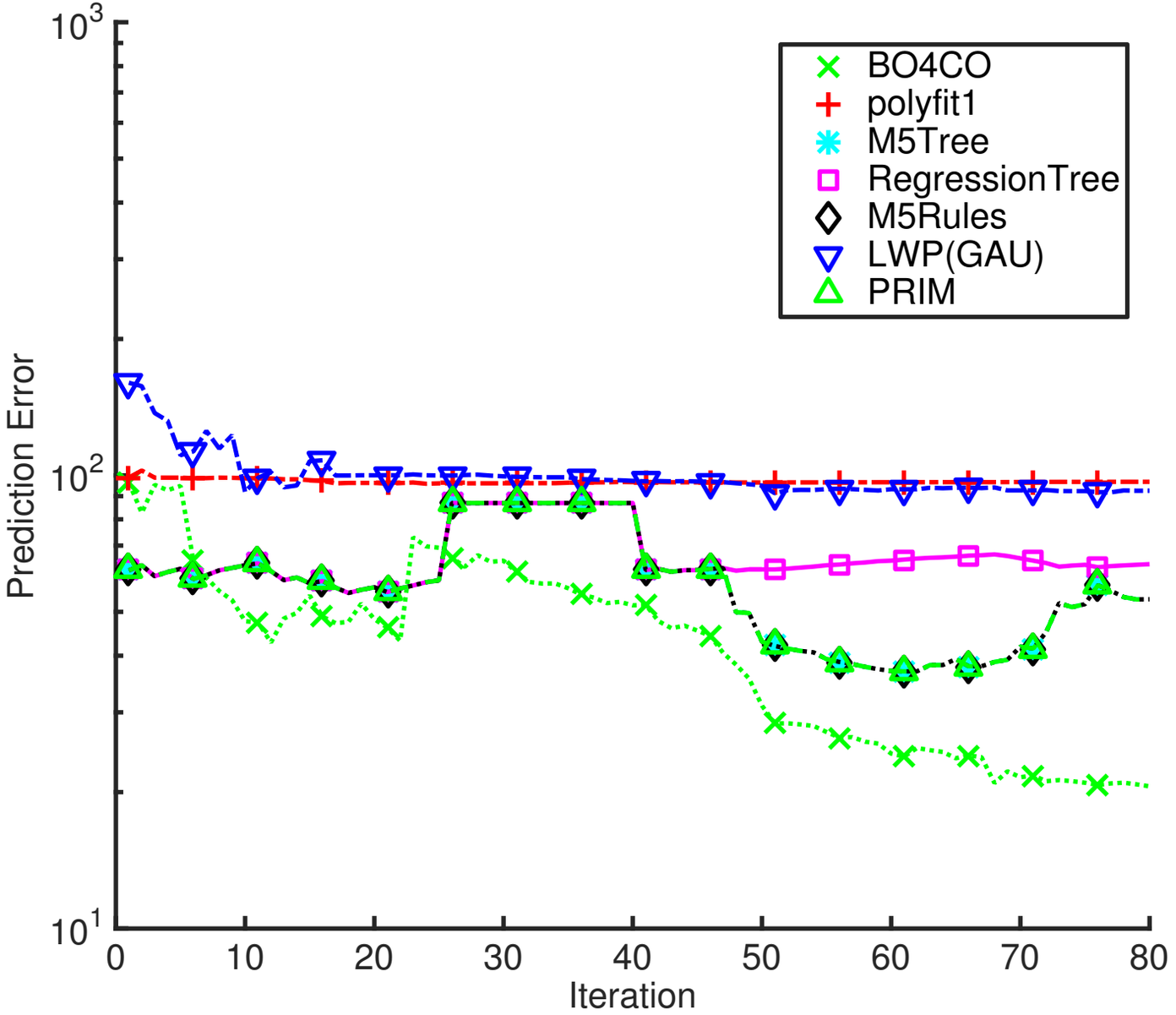


# Model accuracy (comparison with polynomial regression models)



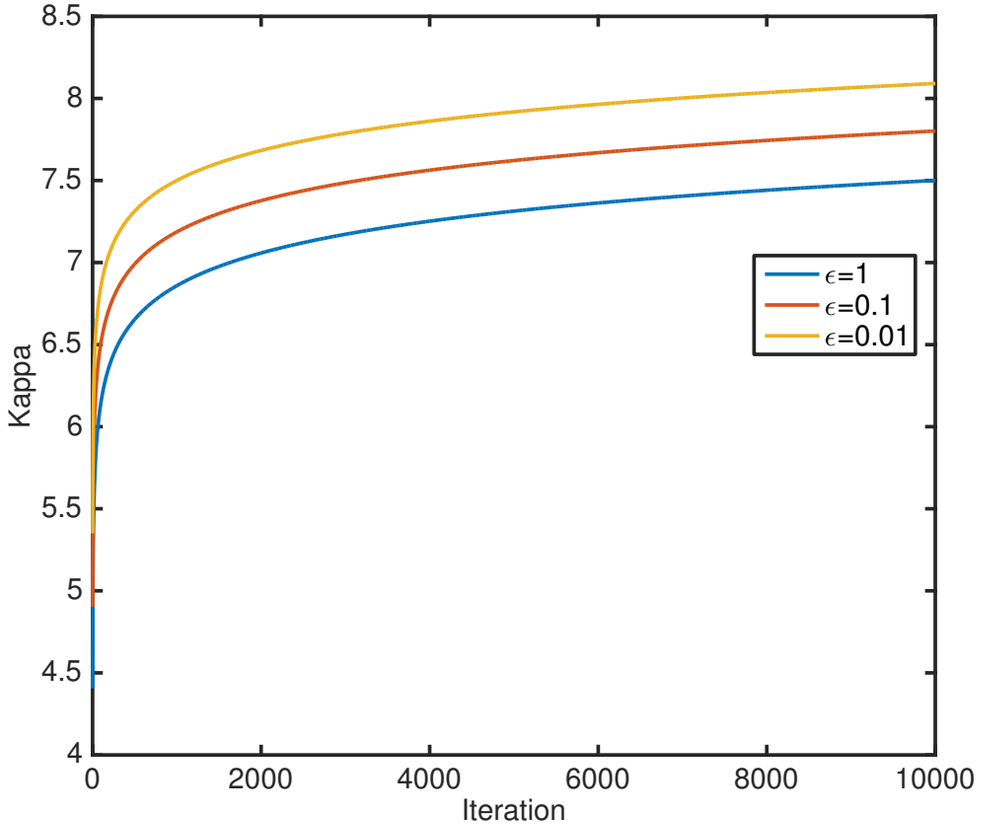
# Prediction accuracy over time

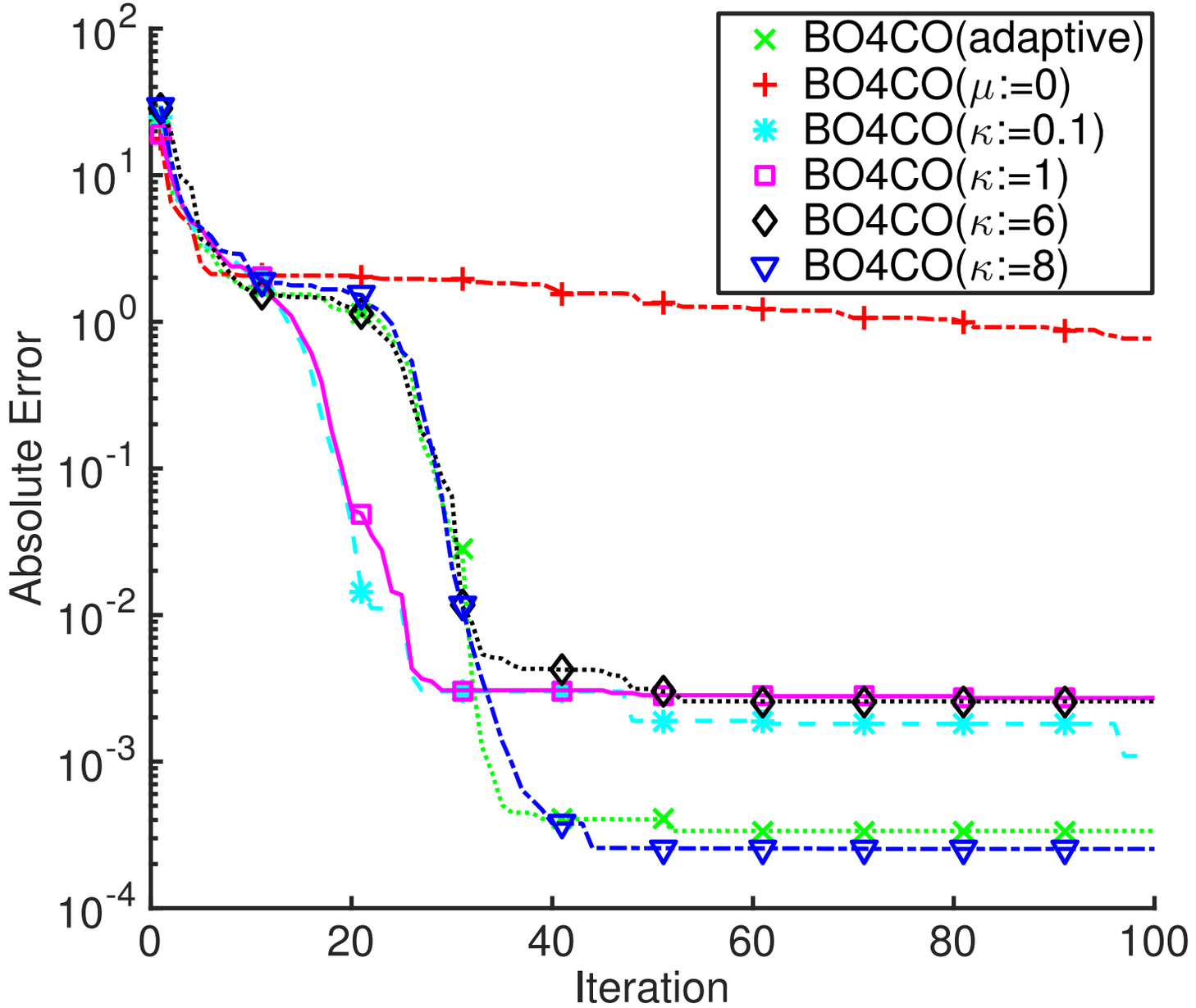


# Exploitation vs exploration

$$\boldsymbol{x}_{t+1} = \underset{\boldsymbol{x}\in\mathbb{X}}{\operatorname{argmax}}\, u(\boldsymbol{x}|\mathcal{M}, \mathbb{S}^1_{1:t})$$

$$u_{LCB}(\boldsymbol{x}|\mathcal{M}, \mathbb{S}^1_{1:t}) = \underset{\boldsymbol{x}\in\mathbb{X}}{\operatorname{argmin}}\, \mu_t(\boldsymbol{x}) - \kappa\sigma_t(\boldsymbol{x}),$$



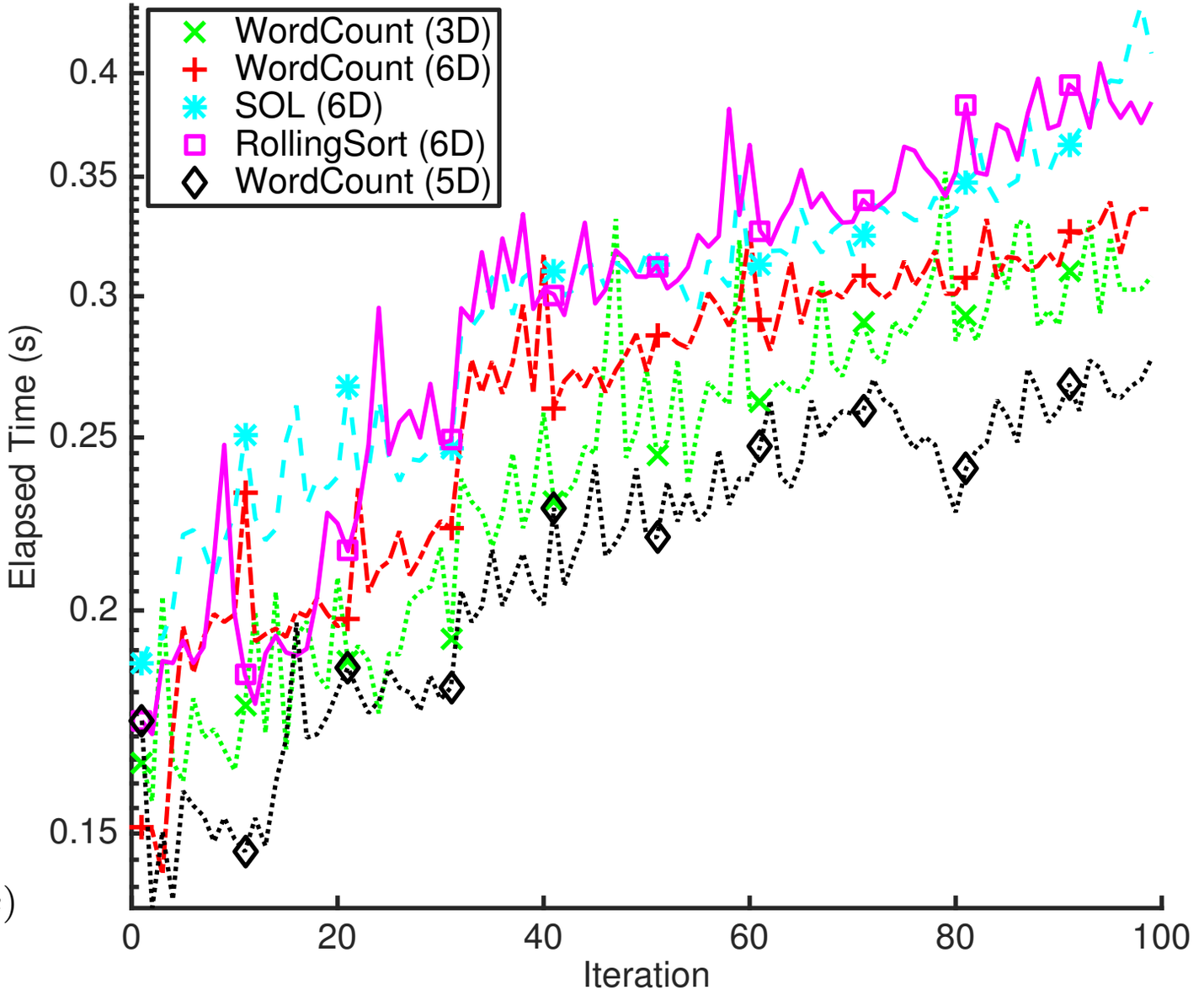
# Runtime overhead

- The computation time in larger datasets is higher than those with less data and lower.

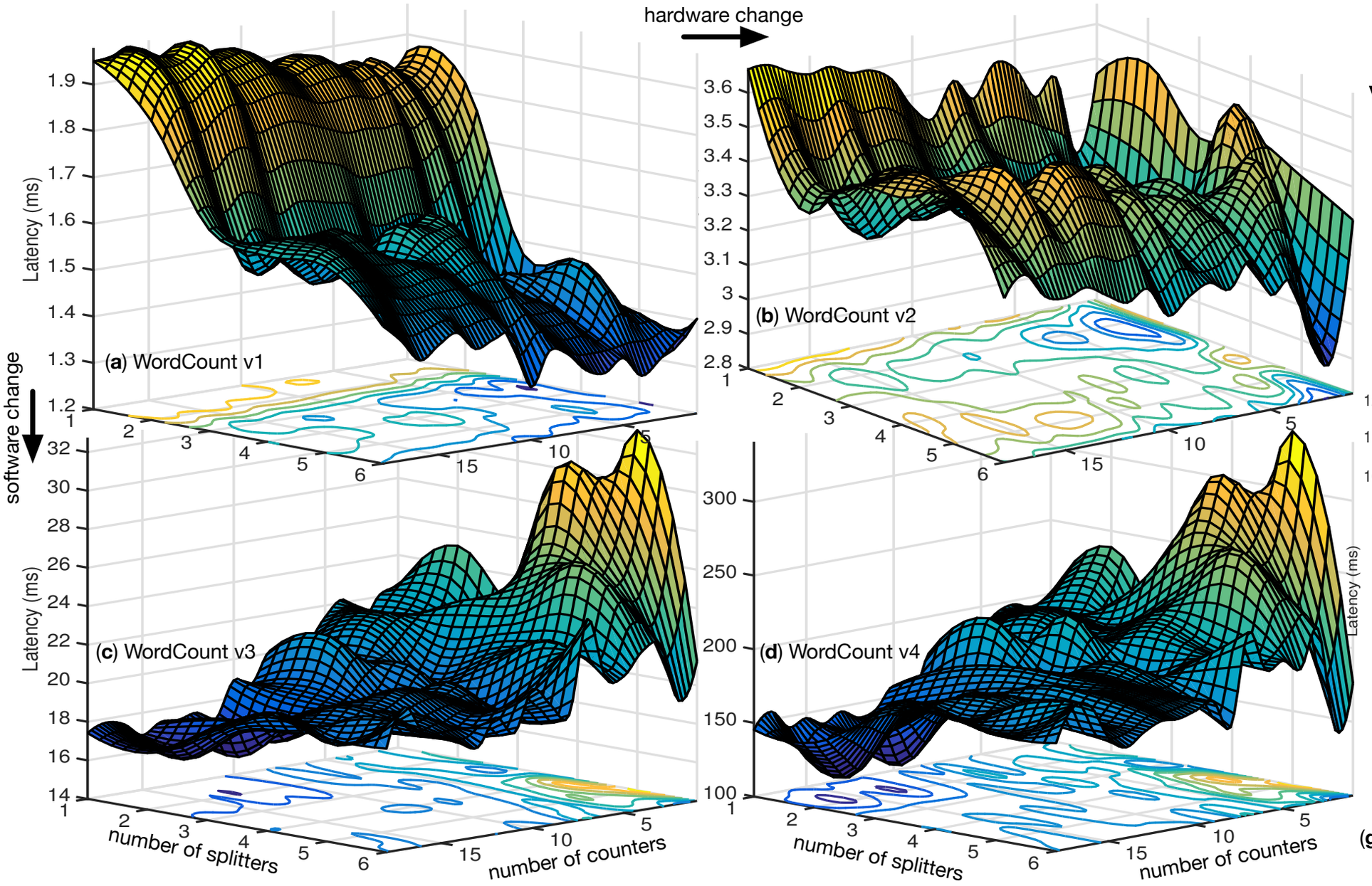
- The computation time increases over time since the matrix size for Cholesky inversion gets larger.

$$y = f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')),$$

$$\mu_t(\boldsymbol{x}) = \mu(\boldsymbol{x}) + \boldsymbol{k}(\boldsymbol{x})^\intercal (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} (\boldsymbol{y} - \boldsymbol{\mu})$$
$$\sigma_t^2(\boldsymbol{x}) = k(\boldsymbol{x}, \boldsymbol{x}) + \sigma^2 \boldsymbol{I} - \boldsymbol{k}(\boldsymbol{x})^\intercal (\boldsymbol{K} + \sigma^2 \boldsymbol{I})^{-1} \boldsymbol{k}(\boldsymbol{x})$$



# Correlations: SPS experiments




correlation coefficient

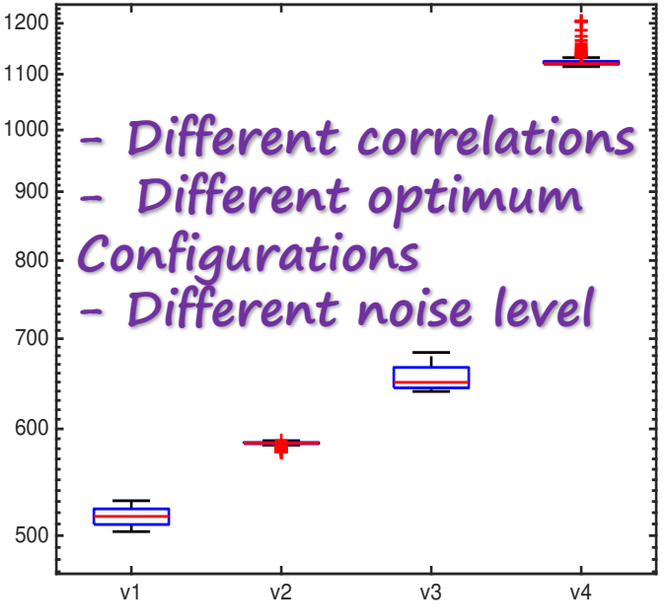
p-value

| | v1 | v2 | v3 | v4 |
|---|---|---|---|---|
| v1 | 1 | 0.41 | -0.46 | -0.50 |
| v2 | 7.36E-06 | 1 | -0.20 | -0.18 |
| v3 | 6.92E-07 | 0.04 | 1 | 0.94 |
| v4 | 2.54E-08 | 0.07 | 1.16E-52 | 1 |

(e) Pearson correlation coefficients

| | v1 | v2 | v3 | v4 |
|---|---|---|---|---|
| v1 | 1 | 0.49 | -0.51 | -0.51 |
| v2 | 5.50E-08 | 1 | -0.2793 | -0.24 |
| v3 | 1.30E-08 | 0.003 | 1 | 0.88 |
| v4 | 1.40E-08 | 0.01 | 8.30E-36 | 1 |

(f) Spearman correlation coefficients

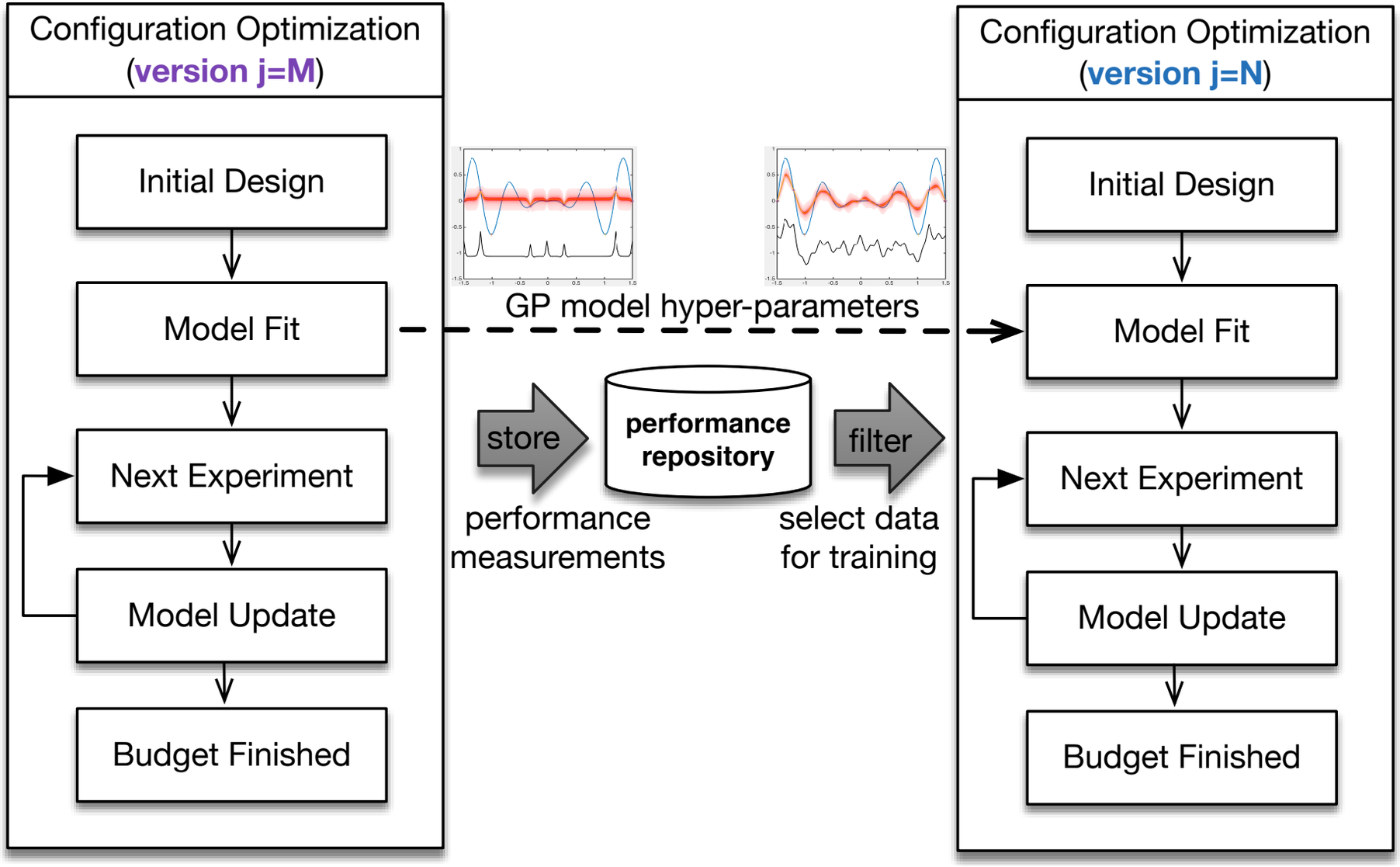



(g) Measurement noise across WordCount versions

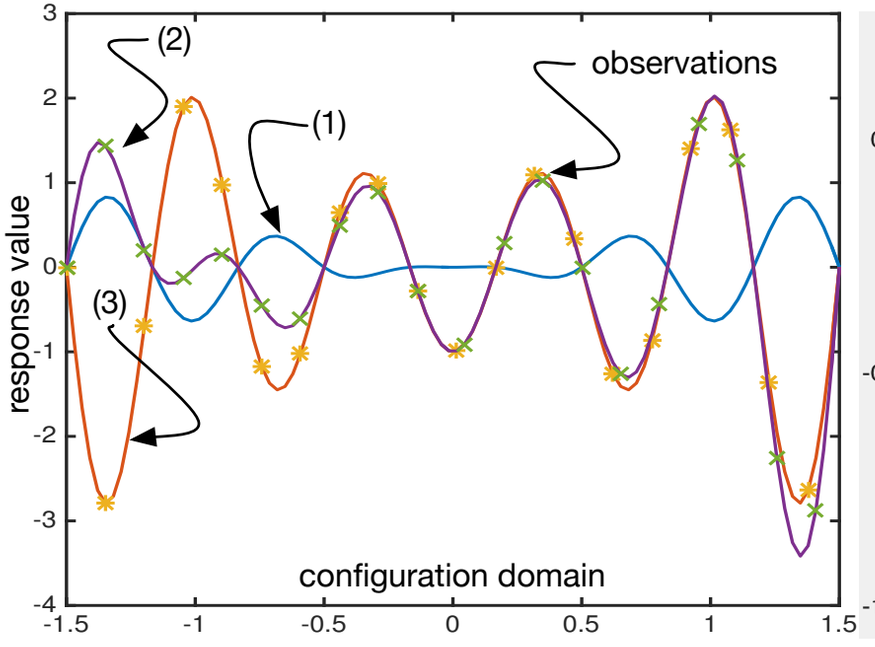
# DevOps

- Different versions are continuously delivered (daily basis).
- Big Data systems are developed using similar frameworks (Apache Storm, Spark, Hadoop, Kafka, etc).
- Different versions share similar business logics.

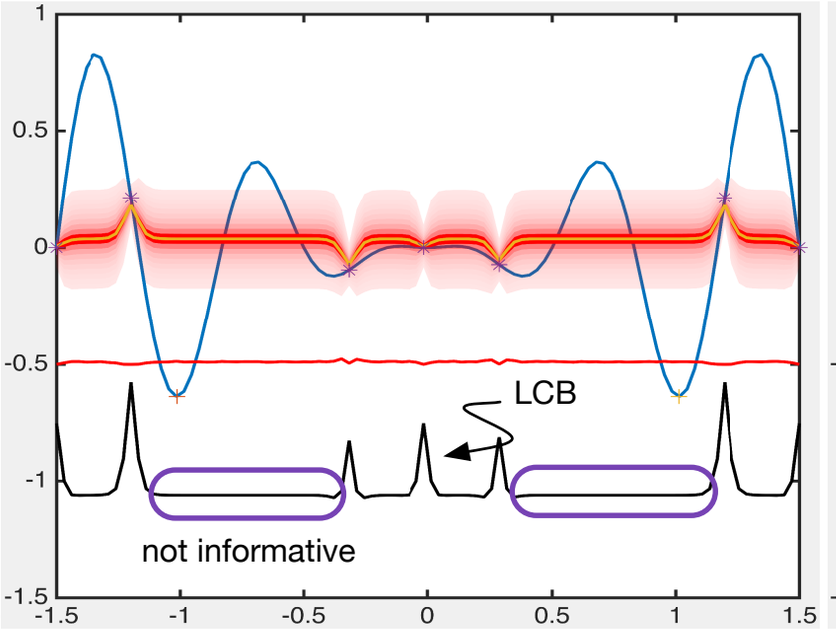
# Solution: Transfer Learning for Configuration Optimization



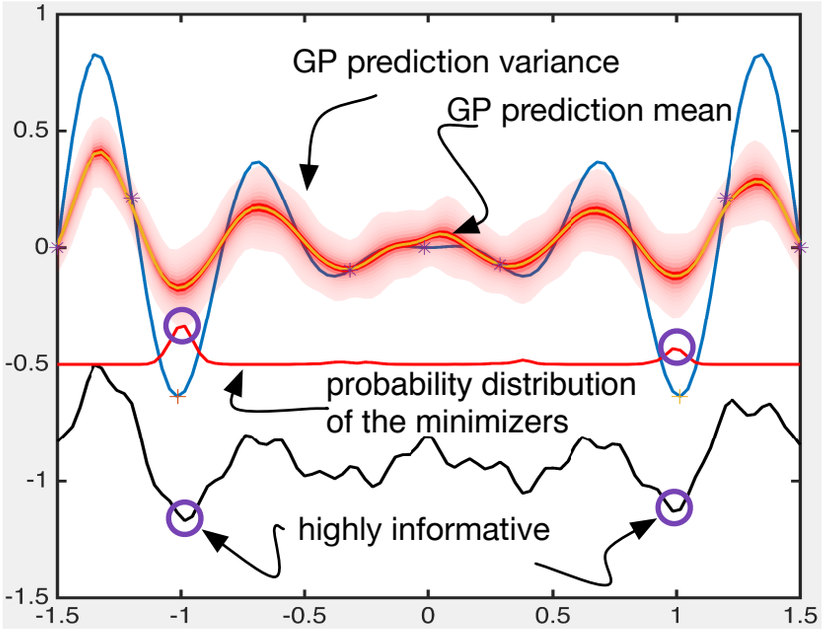
# The case where we learn from correlated responses




(a) 3 sample response functions

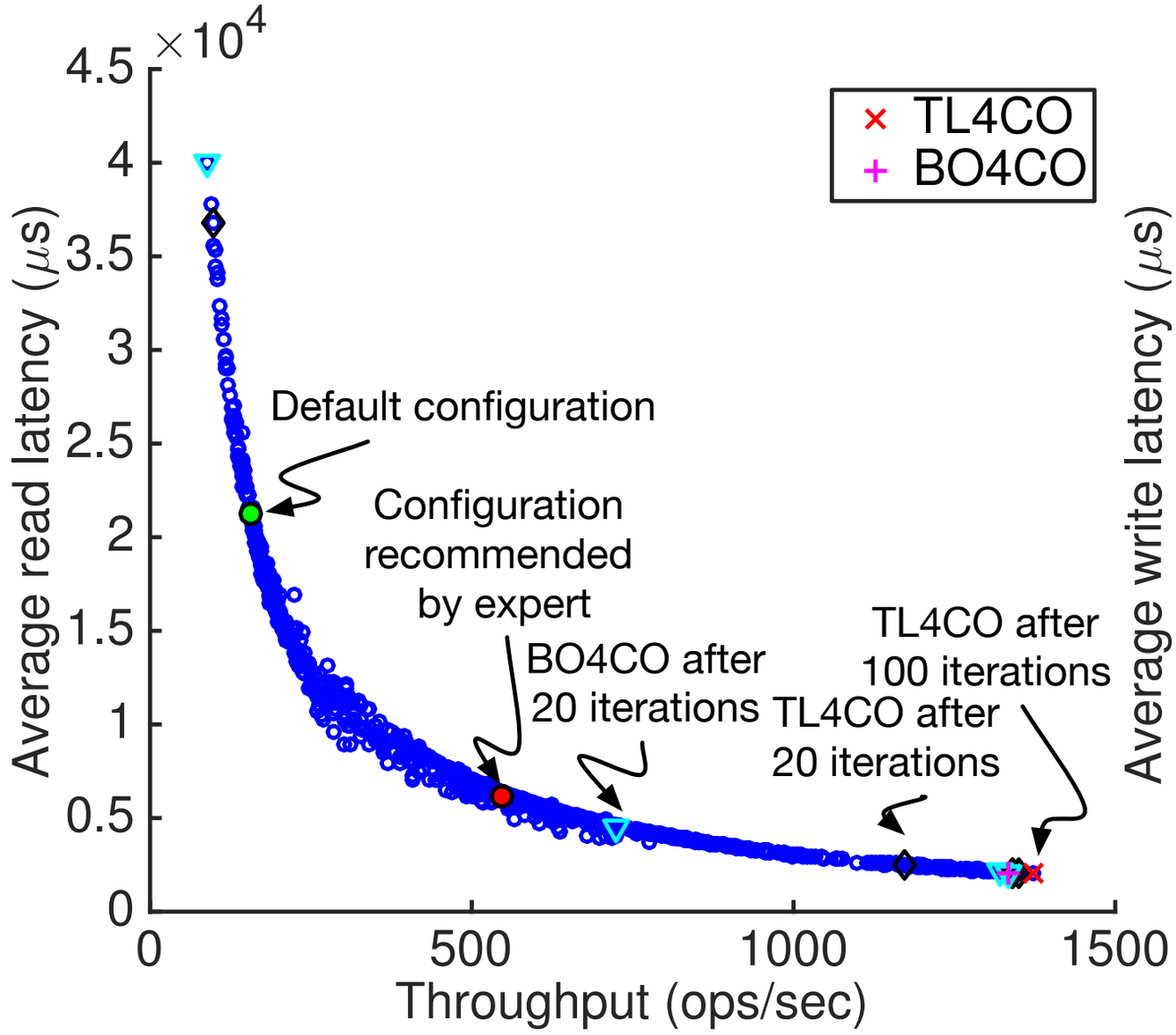


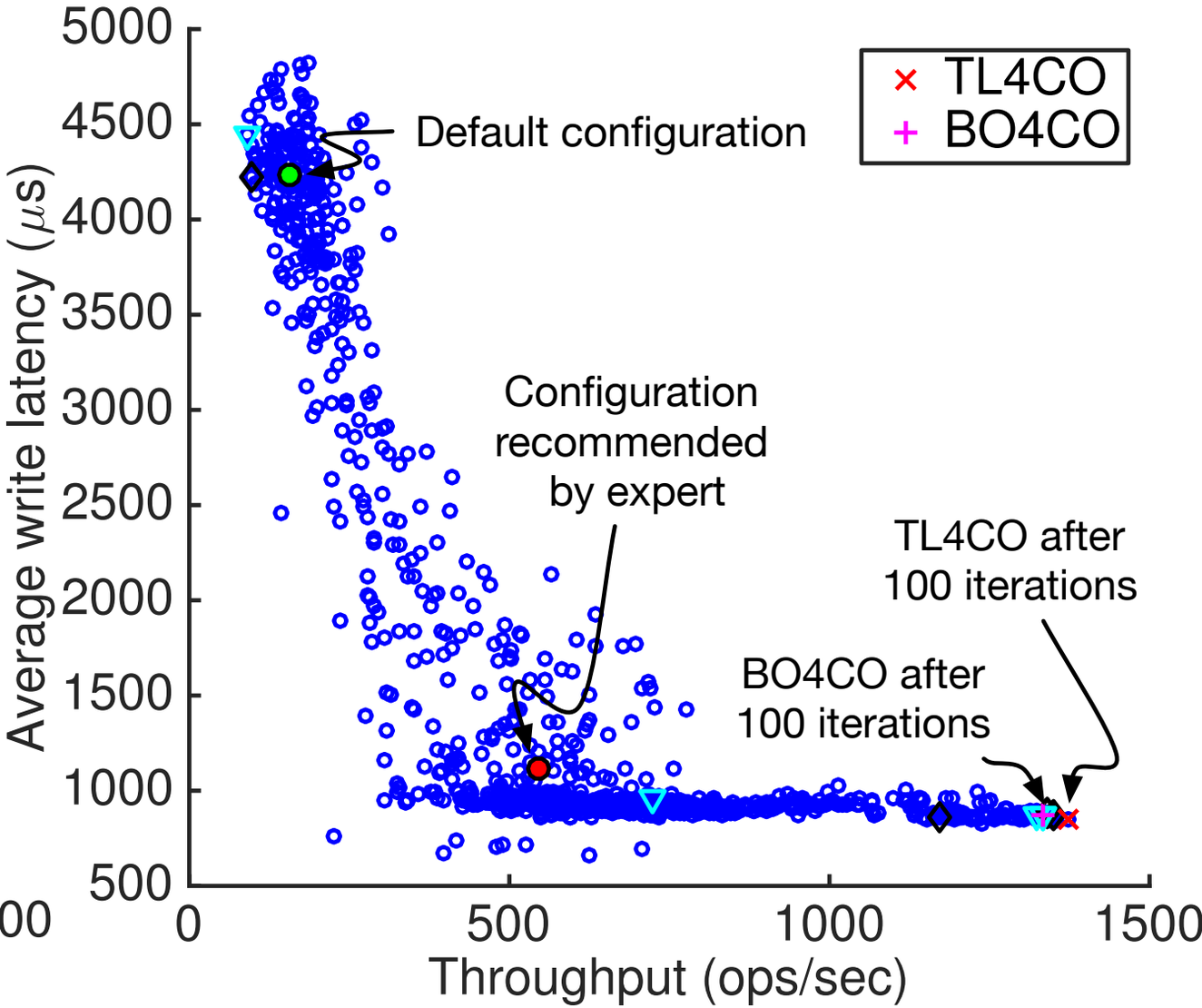

(b) GP fit for (1) ignoring observations for (2),(3)



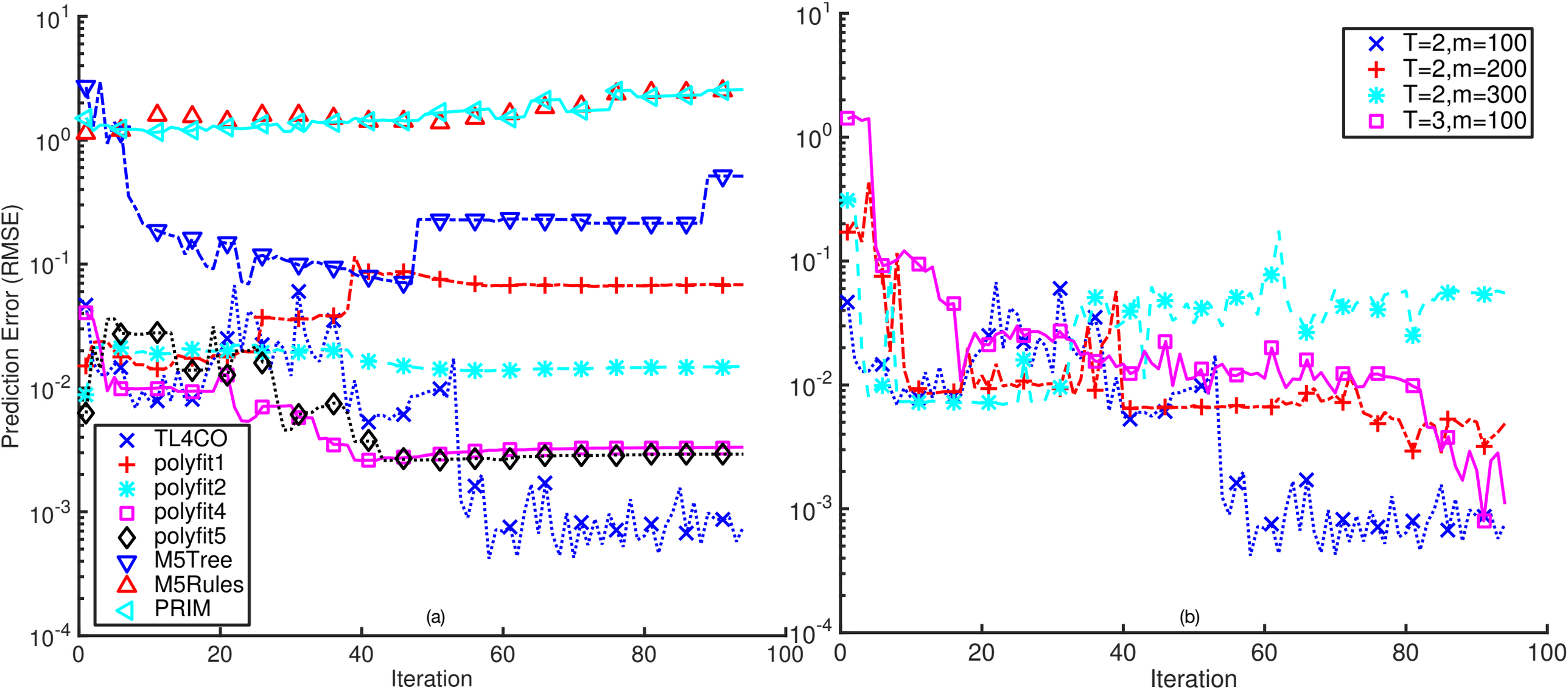

(c) multi-task GP fit for (1) by transfer learning from (2),(3)

# Comparison with default and expert prescription



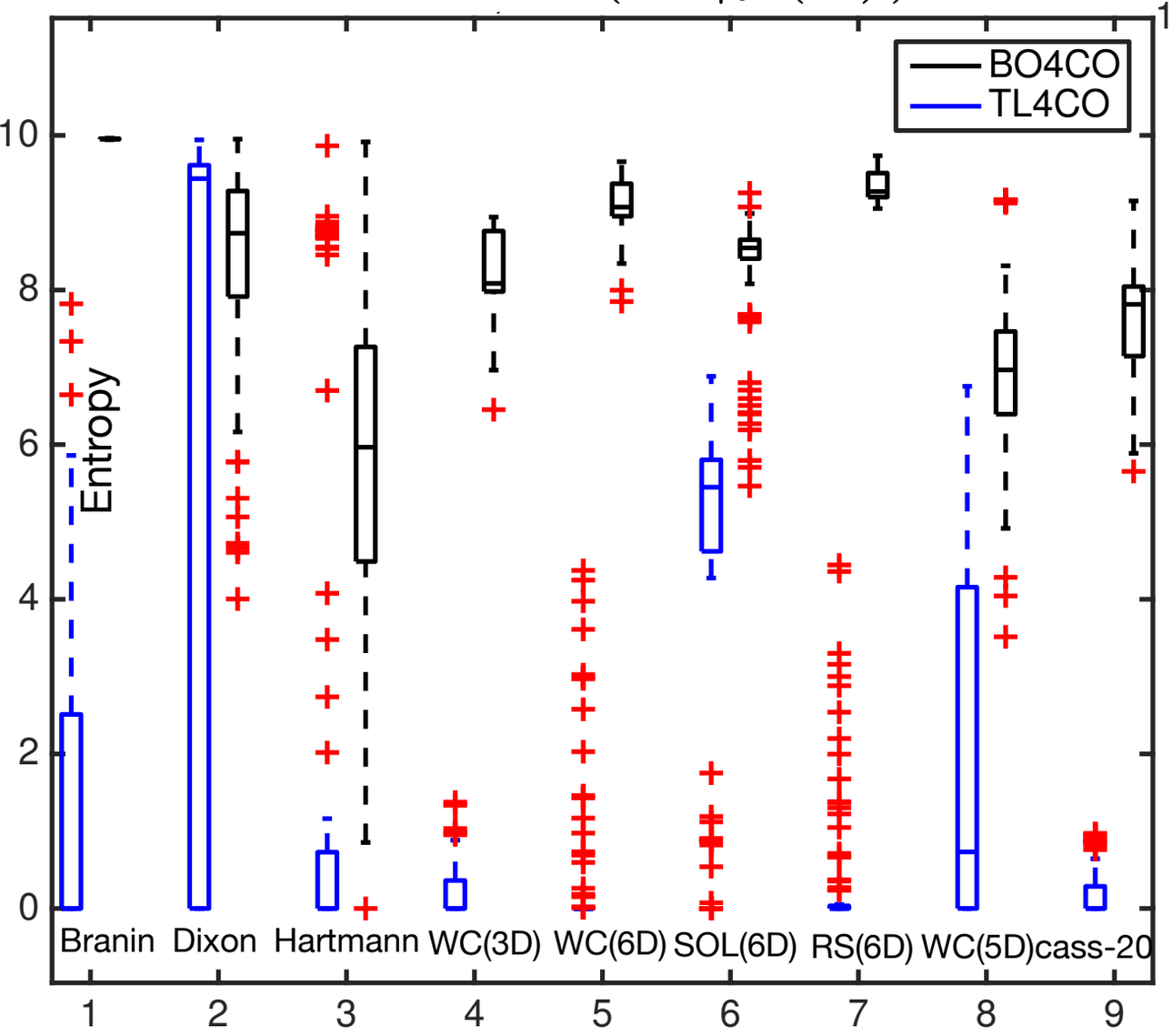



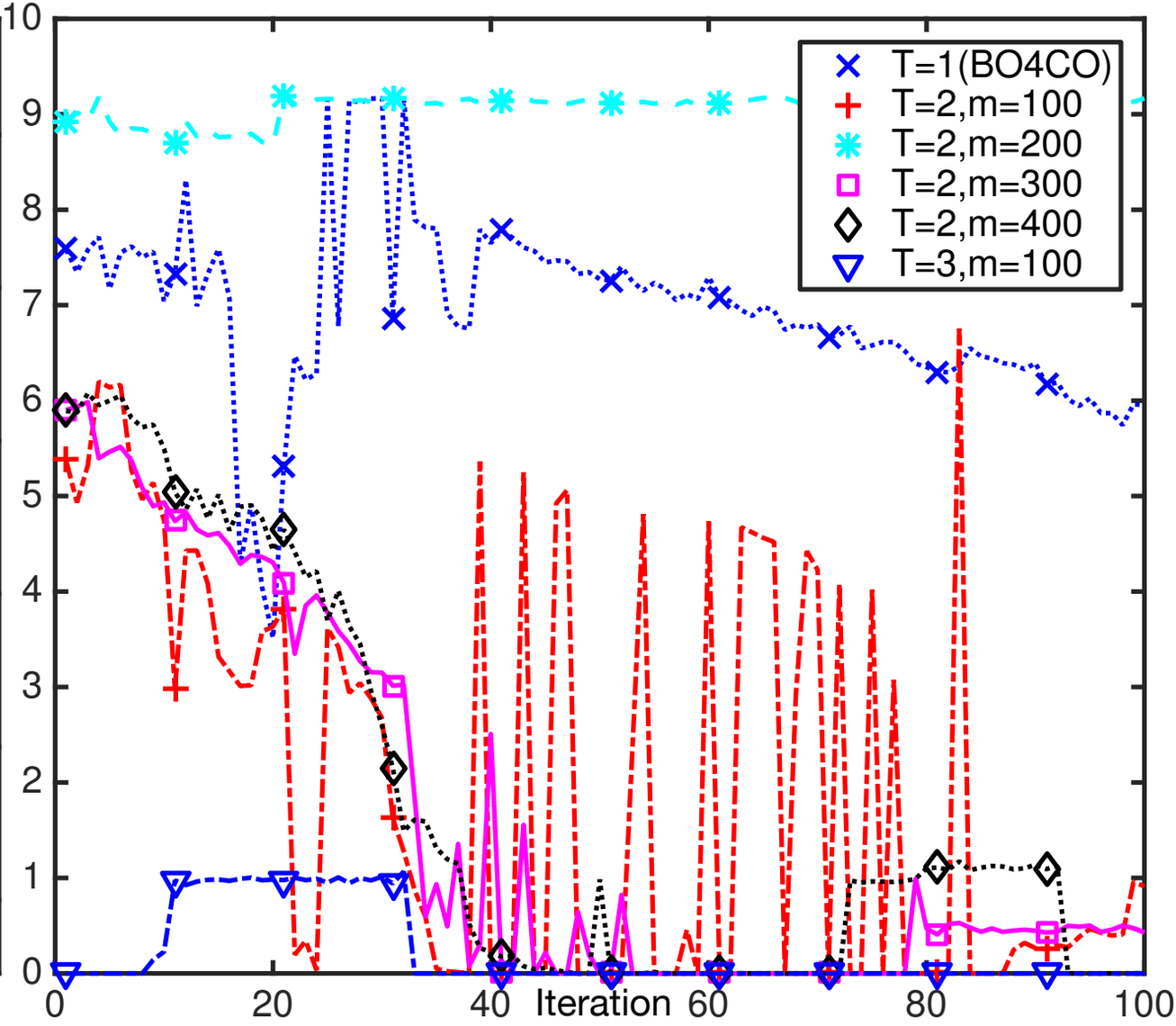
# Prediction accuracy over time



# Entropy of the density function of the minimizers

$$\boldsymbol{X}^* = Pr(\boldsymbol{x}^* | f(\boldsymbol{x}))$$ *Knowledge about the location of the minimizer*

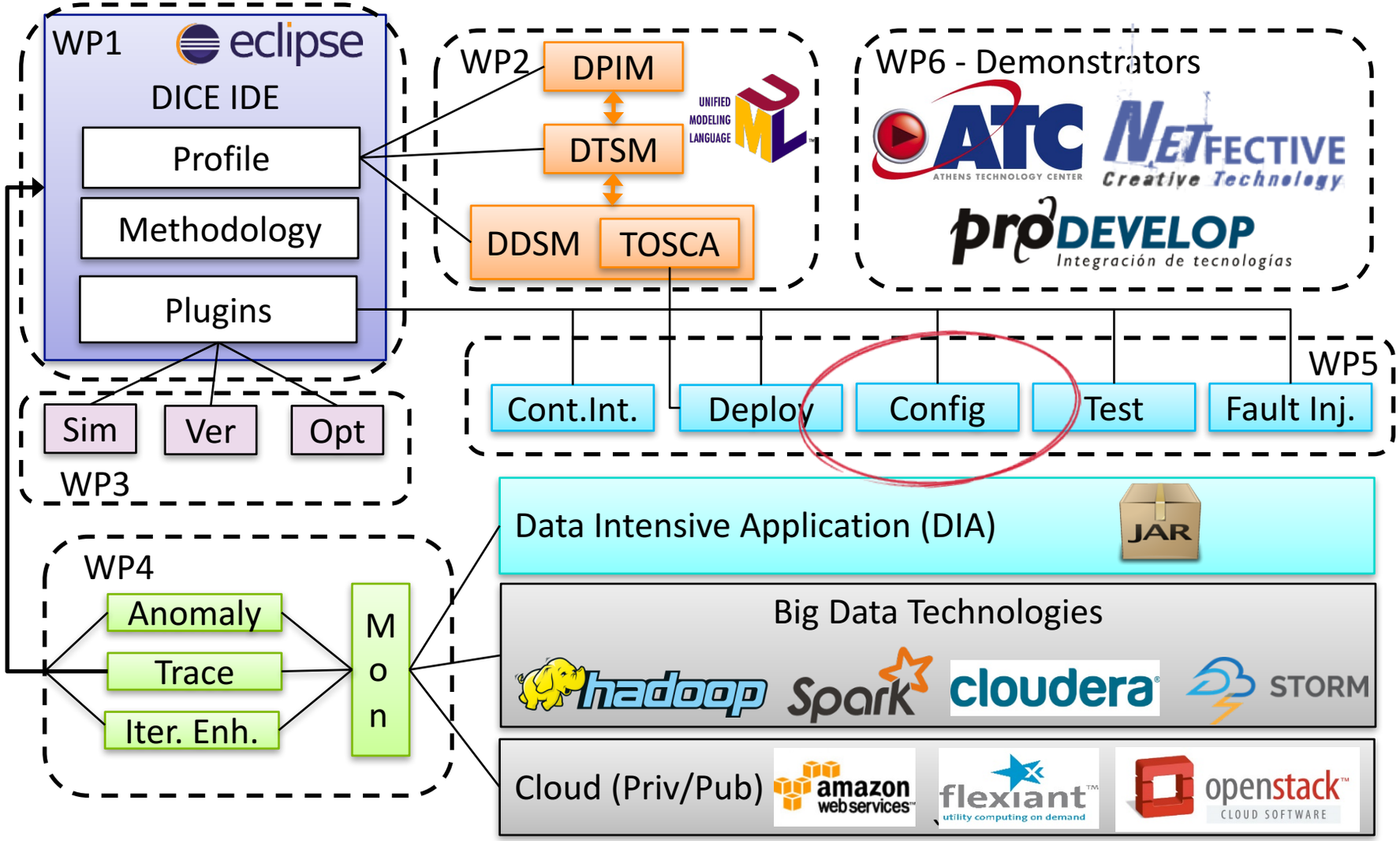


# Takeaways

- Be aware of Uncertainty
  - By **quantifying** the uncertainty
  - Make decisions taking into account the right level of uncertainty (homoscedastic vs heteroscedastic)
  - Uncertainty sometimes helps (models that provide an estimation of the uncertainty are typically more **informative**)
  - By exploiting this knowledge you can only explore **interesting zones** rather than learning the whole performance function
- You can learn from operational data
  - Not only from the current version, but from **previous measurements** as well
  - Use the learning from past measurements as **prior knowledge**
  - **Too much data** can be also harmful, it would slow down or blur the proper learning (**negative transfer**)

Acknowledgement:
-BO4CO as a part of DevOps pipeline in H2020 DICE
-BO4CO is being acquired by TATA (TCS)




Code and data: https://github.com/dice-project/DICE-Configuration-BO4CO

# Submit to **SEAMS 2017**

- **Any work on Self-***
- **Abstract Submission**: 6 Jan, 2017 (**firm**)
- **Paper Submission**: 13 Jan, 2017 (**firm**)
- **Page limit**:
  - Long: 10+2,
  - New ideas and tools: 6+1
- **More info**: https://wp.doc.ic.ac.uk/seams2017/
- **Symposium**: 22-23 May, 2017
- **We accept artifacts submissions (tool, data, model)**

**12th International Symposium on Software Engineering for Adaptive and Self-Managing Systems**
**Buenos Aires, Argentina, May 22-23, 2017, http://wp.doc.ic.ac.uk/seams2017**


Co-located with


IEEE computer society tcse

acm Association for Computing Machinery SIGSOFT

## Call for Papers

Self-adaptation and self-management are key objectives in many modern and emerging software systems, including the industrial internet of things, cyber-physical systems, cloud computing, and mobile computing. These systems must be able to adapt themselves at run time to preserve and optimize their operation in the presence of uncertain changes in their operating environment, resource variability, new user needs, attacks, intrusions, and faults.

Approaches to complement software-based systems with self-managing and self-adaptive capabilities are an important area of research and development, offering solutions that leverage advances in fields such as software architecture, fault-tolerant computing, programming languages, robotics, and run-time program analysis and verification. Additionally, research in this field is informed by related areas like biologically-inspired computing, artificial intelligence, machine learning, control systems, and agent-based systems. The SEAMS symposium focuses on applying software engineering to these approaches, including methods, techniques, and tools that can be used to support self-* properties like self-adaptation, self-management, self-healing, self-optimization, and self-configuration.

The objective of SEAMS is to bring together researchers and practitioners from diverse areas to investigate, discuss, and examine the fundamental principles, state of the art, and critical challenges of engineering self-adaptive and self-managing systems.

**Topics of Interest:** All topics related to engineering self-adaptive and self-managing systems, including:

**Foundational Concepts**
- self-* properties
- control theory
- algorithms
- decision-making and planning
- managing uncertainty
- mixed-initiative and human-in-the-loop systems

**Languages**
- formal notations for modeling and analyzing self-* properties
- programming language support for self-adaptation

**Constructive methods**
- requirements elicitation techniques
- reuse support (e.g., patterns, designs, code)
- architectural techniques
- legacy systems

**Analytical Methods for Self-Adaptation and -Management**
- evaluation and assurance
- verification and validation
- analysis and testing frameworks

**Application Areas**
- Industrial internet of things
- Cyber-physical systems
- Cloud computing
- Mobile computing
- Robotics
- Smart user interfaces
- Security and privacy
- Wearables and ubiquitous/pervasive systems

**Artifacts* and Evaluations**
- model problems and exemplars
- resources, metrics, or software that can be used to compare self-adaptive approaches
- experiences in applying tools to real problems

**Paper Submission Details**

SEAMS solicits three types of papers: **long papers** (10 pages for the main text, inclusive of figures, tables, appendices, etc.; references may be included in up to two additional pages), short papers for **new ideas and early results** (6 pages + 1 for references) and **artifact papers** (6 pages + 1 reference). Long papers should clearly describe innovative and original research or explain how existing techniques have been applied to real-world examples. Short papers should describe novel and promising ideas and/or techniques that are in an early stage of development. Artifact papers must describe why and how the accompanying artifact may be useful for the broader community. Papers must not have been previously published or concurrently submitted elsewhere. Papers must conform to IEEE formatting guidelines (see ICSE 2017 style guidelines), and submitted via EasyChair. Accepted papers will appear in the symposium proceedings that will be published in the ACM and IEEE digital libraries. Accepted artifact papers will also be archived on the Dagstuhl Artifacts Series (DARTS).

**Further Information**
Symposia-related email should be addressed to:
seams17-org@lists.andrew.cmu.edu

**Important Dates:**
Abstract Submission: 6 Jan, 2017 (AoE,firm)
Paper Submission: 13 Jan, 2017 (AoE,firm)
Notification: 21 February, 2017
Camera ready: 6 Mar, 2017

***There will be a specific session to be dedicated to artifacts that may be useful for the community as a whole. Please see http://wp.doc.ic.ac.uk/seams2017/call-for-artifacts/ for more details.**

Selected papers will be invited to submit to the ACM Transactions on Autonomous and Adaptive Systems (TAAS).

**General Chair**
David Garlan, USA
**Program Chair**
Bashar Nuseibeh, UK & Ireland
**Artifacts Chair**
Javier Cámara, USA
**Publicity Chair**
Pooyan Jamshidi, UK
**Local Chair**
Nicolás D'Ippolito, Argentina
**Program Committee**
Dalal Alrajeh, UK
Jesper Andersson, Sweden
Rami Bahsoon, UK
Arosha Bandara, UK
Luciano Baresi, Italy
Jacob Beal, USA
Nelly Bencomo, UK
Amel, Bennaceur, UK
Victor Braberman, Argentina
Tomas Bures, Czech Republic
Radu Calinescu, UK
Javier Camara, USA
Betty Cheng, USA
Siobhán Clarke, Ireland
Rogério de Lemos, UK
Elisabetta di Nitto, Italy
Nicolás D'Ippolito, Argentina
Ada Diaconescu, France
Gregor Engels, Germany
Antonio Filieri, UK
Erik Fredericks, USA
Holger Giese, Germany
Hassan Gomaa, USA
Joel Greenyer, Germany
Mark Harman, UK
Valerie Issarny, France
Pooyan Jamshidi, UK
Jean-Marc Jézéquel, France
Samuel Kounev, Germany
Philippe Lalanda, France
Seok–Won Lee, South Korea
Marin Litoiu, Canada
Xiaoxing Ma, China
Martina Maggio, Sweden
Sam Malek, USA
Nenad Medvidovic, USA
Hausi Müller, Canada
Henry Muccini, Italy
John Mylopoulos, Canada
Ingrid Nunes, Brazil
Liliana Pasquale, Ireland
Patrizio Pelliccione, Sweden
Xin Peng, China
David Rosenblum, Singapore
Bradley Schmerl, USA
Hella Seebach, Germany
Amir Molzam Sharifloo, Germany
Vitor Silva Sousa, Brazil
Jan-Philipp Steghöfer, Sweden
Ladan Tahvildari, Canada
Kenji Tei, Japan
Axel van Lamsweerde, Belgium
Giuseppe Valetto, Italy
Mirko Viroli, Italy
Danny Weyns, Belgium
Yijun Yu, UK
**Artifact Evaluation Committee**
Konstantinos Angelopoulos, UK
Nuno Antunes, Portugal
Amel Bennaceur, UK
Javier Cámara, USA
Ilias Gerostathopoulos, Germany
Mahmoud Hammad, USA
Muhammad Usman Iftikhar, Sweden
Ashutosh Pandey, USA
Roykrong Sukkerd, USA
Christos Tsigkanos, Italy