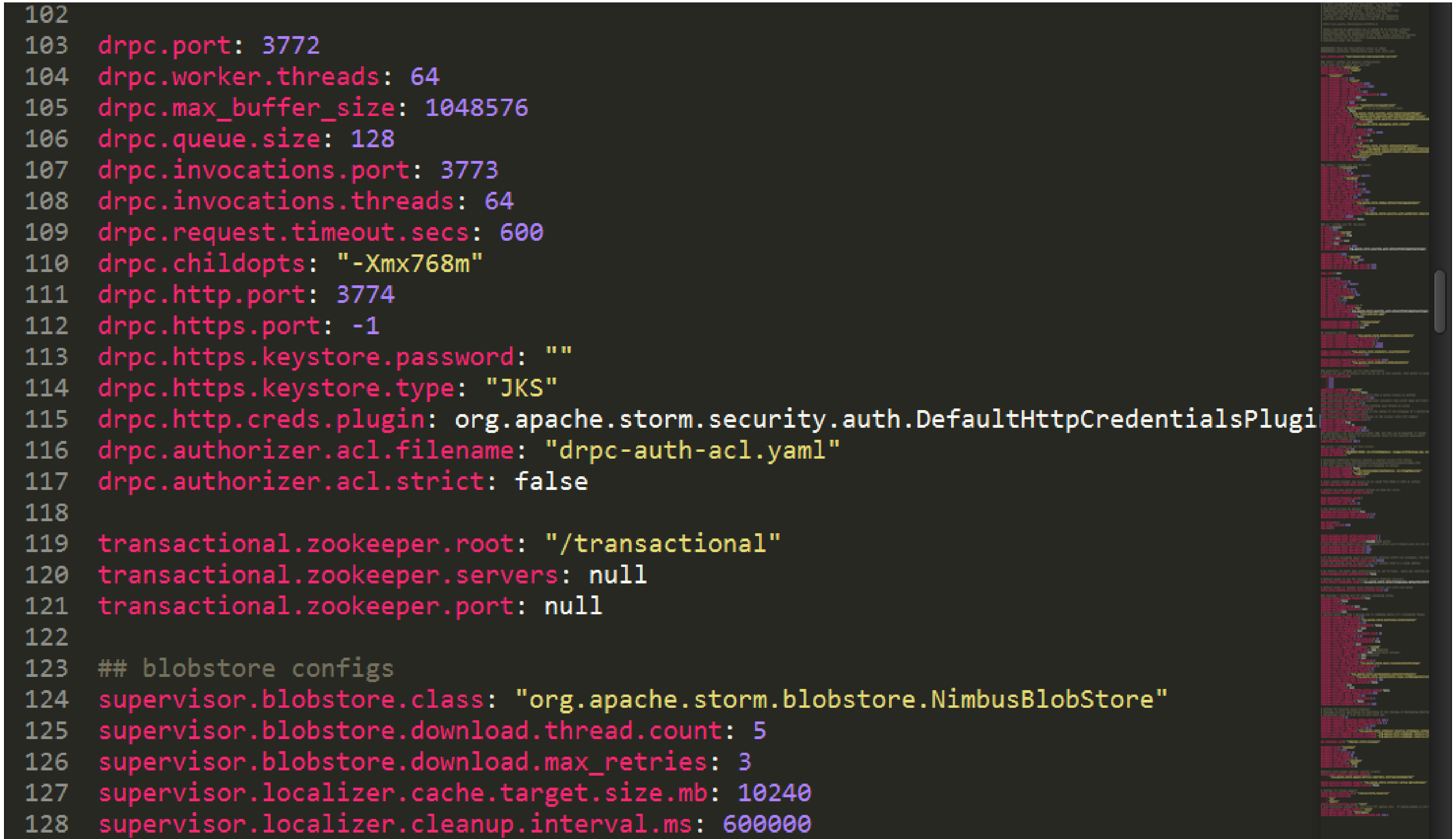
Transfer Learning for Improving Model Predictions in Highly Configurable Software

Pooyan Jamshidi, Miguel Velez, Christian Kästner
Norbert Siegmund, Prasad Kawthekar

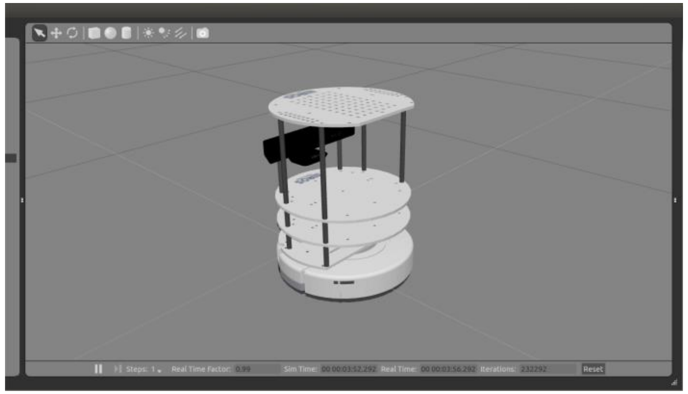
Carnegie Mellon University

Performance Influence of Configuration Parameters



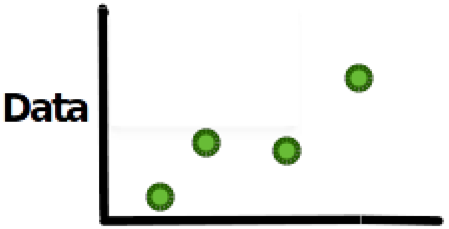
```

drpc.port: 3772
drpc.worker.threads: 64
drpc.max_buffer_size: 1048576
drpc.queue.size: 128
drpc.invocations.port: 3773
drpc.invocations.threads: 64
drpc.request.timeout.secs: 600
drpc.childopts: "-Xmx768m"
drpc.http.port: 3774
drpc.https.port: -1
drpc.https.keystore.password: ""
drpc.https.keystore.type: "JKS"
drpc.http.creds.plugin: org.apache.storm.security.auth.DefaultHttpCredentialsPlugi
drpc.authorizer.acl.filename: "drpc-auth-acl.yaml"
drpc.authorizer.acl.strict: false

transactional.zookeeper.root: "/transactional"
transactional.zookeeper.servers: null
transactional.zookeeper.port: null

## blobstore configs
supervisor.blobstore.class: "org.apache.storm.blobstore.NimbusBlobStore"
supervisor.blobstore.download.thread.count: 5
supervisor.blobstore.download.max_retries: 3
supervisor.localizer.cache.target.size.mb: 10240
supervisor.localizer.cleanup.interval.ms: 600000
```

Classic Sensitivity Analysis

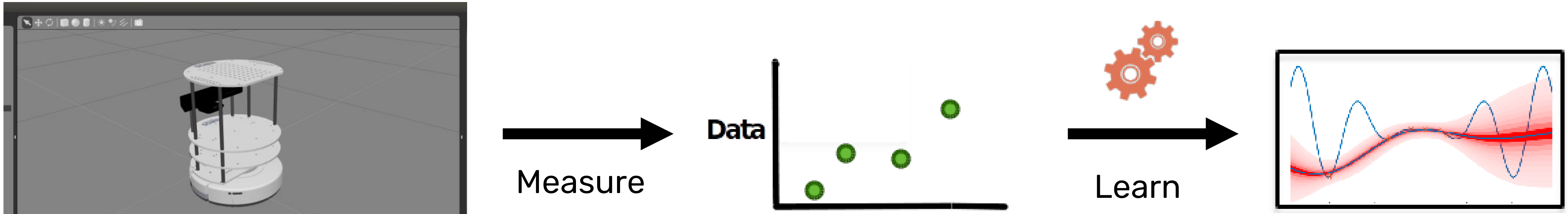


Steps: 1 Real Time Factor: 0.99 Sim Time: 00 00:03:52.292 Real Time: 00 00:03:56.292 Iterations: 232292

Measure



Classic Sensitivity Analysis



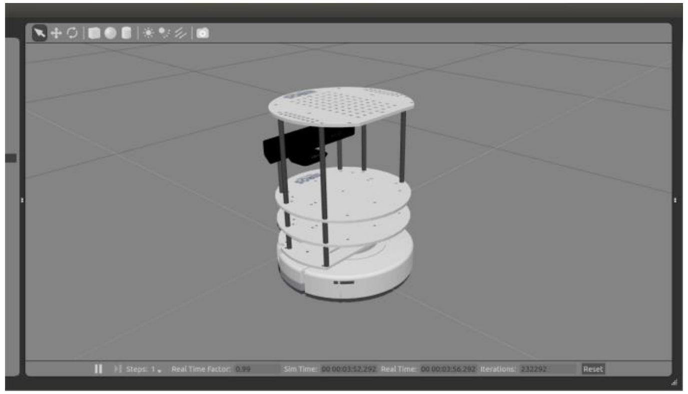
Measure

Data

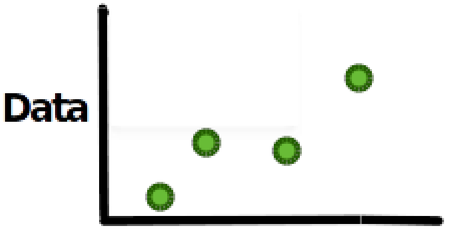
Learn

50 - 3*C1 + 20*C2
- 3*C1*C3

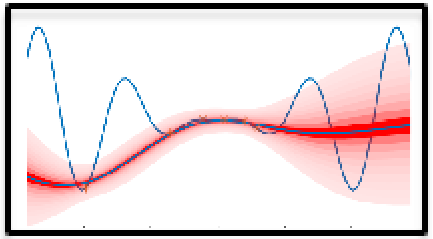
Classic Sensitivity Analysis



Measure



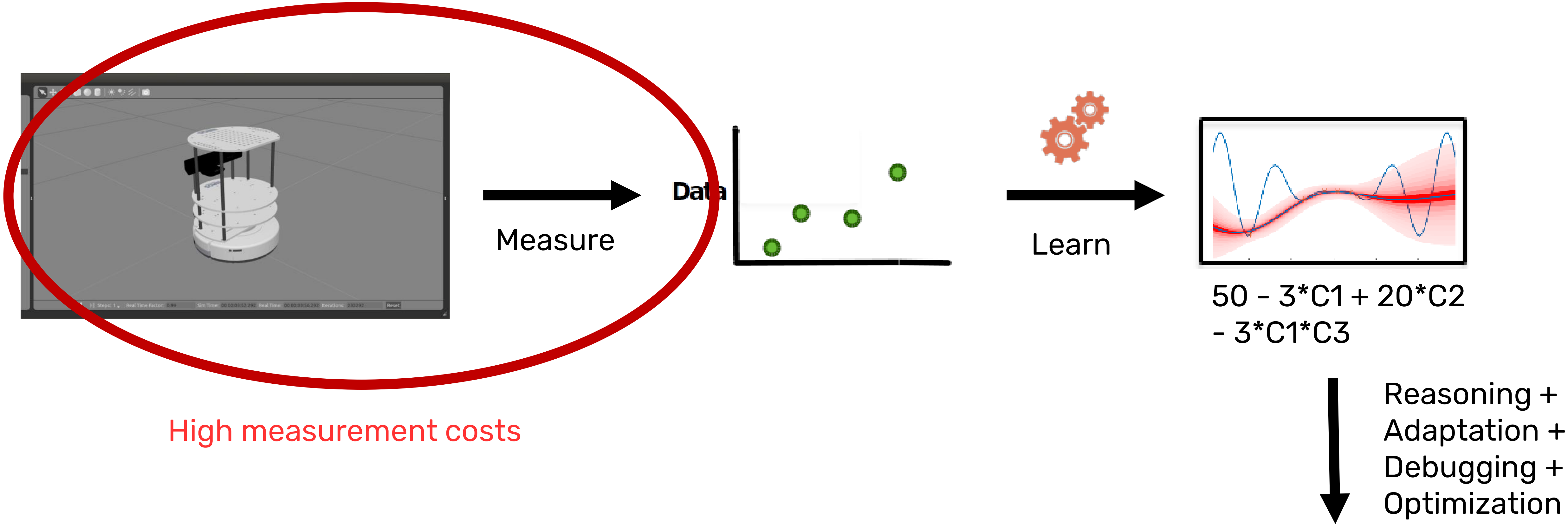
Learn



50 - 3*C1 + 20*C2 - 3*C1*C3

Reasoning +
Adaptation +
Debugging +
Optimization

Classic Sensitivity Analysis



Measure

Data

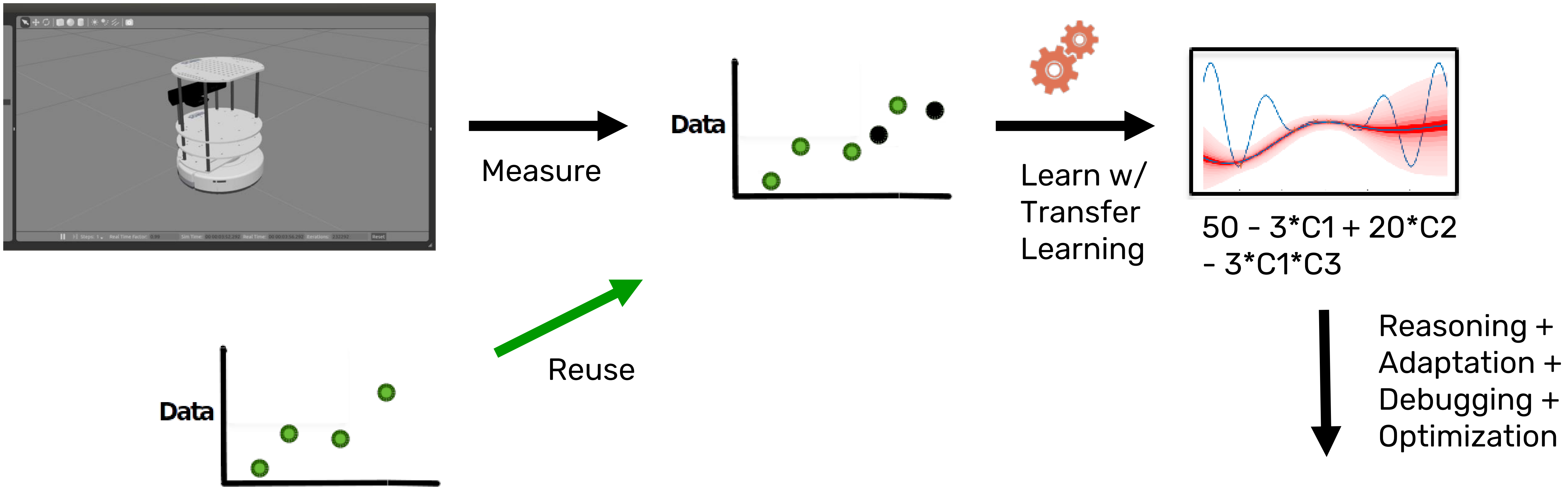
Learn

50 - 3*C1 + 20*C2
- 3*C1*C3

Reasoning +
Adaptation +
Debugging +
Optimization

High measurement costs

Idea: Transfer Learning



Measure

Data

Learn w/ Transfer Learning

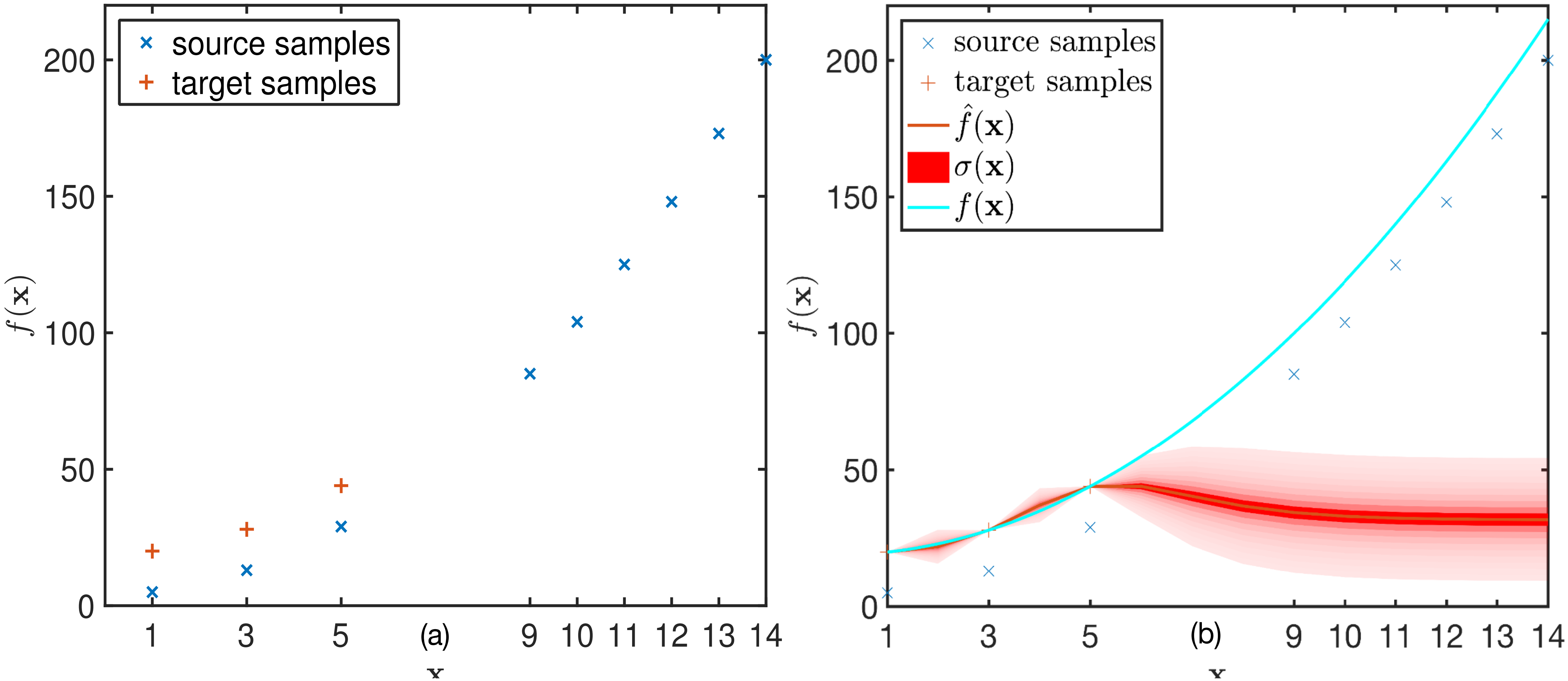
50 - 3*C1 + 20*C2 - 3*C1*C3

Reasoning + Adaptation + Debugging + Optimization

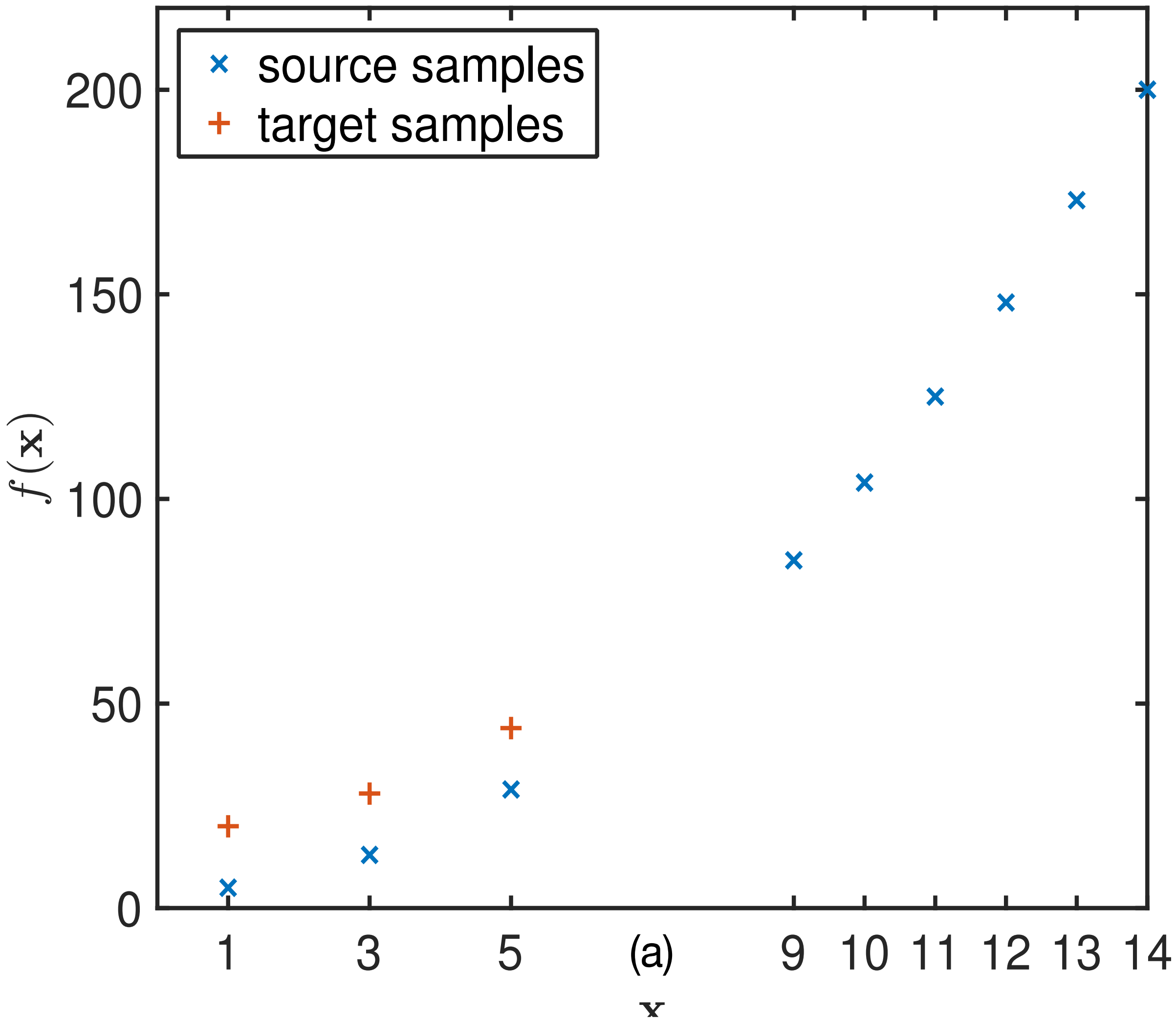
Data

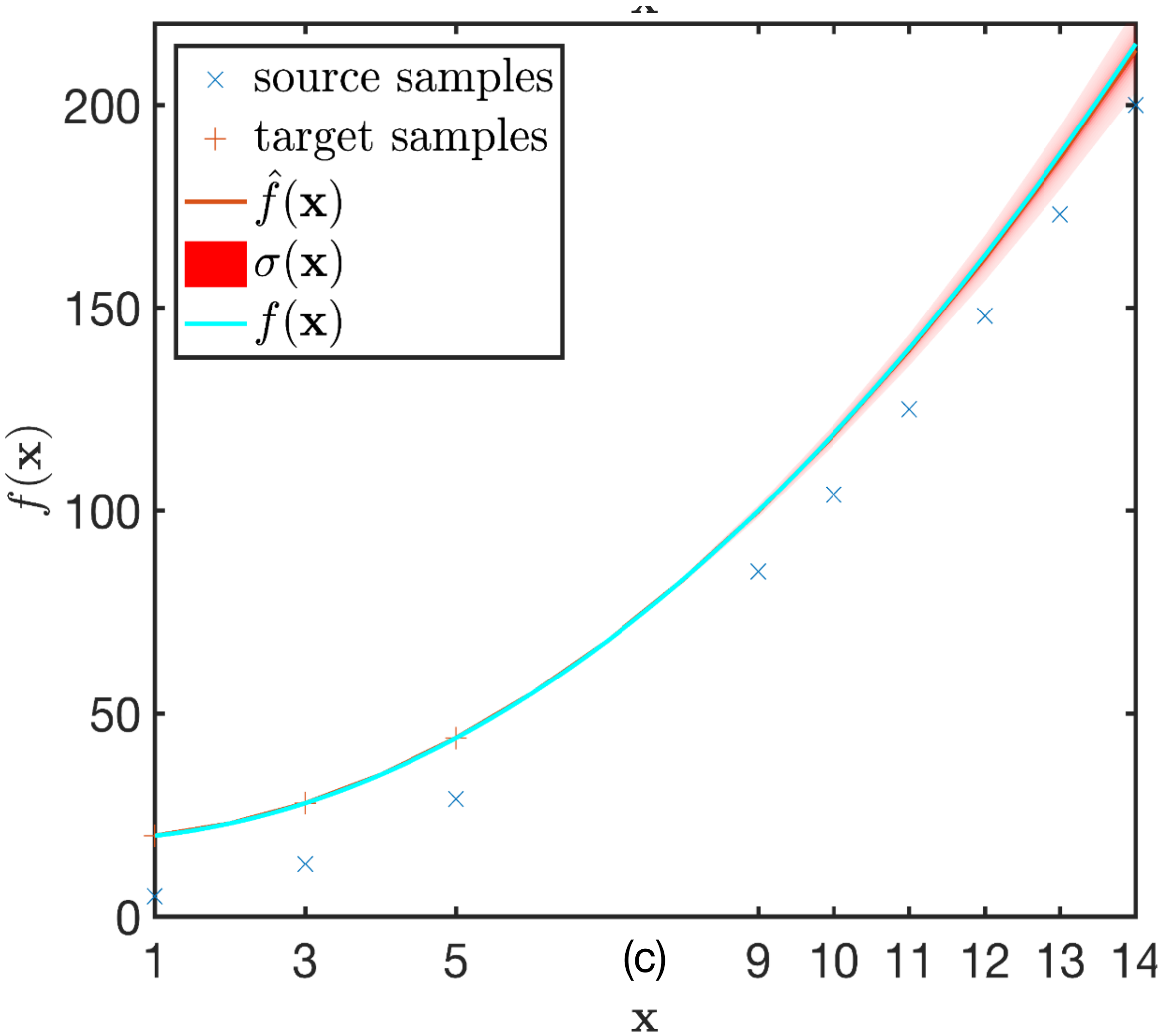
Reuse

Exploiting Similarity



Exploiting Similarity





GP for modeling black box response function

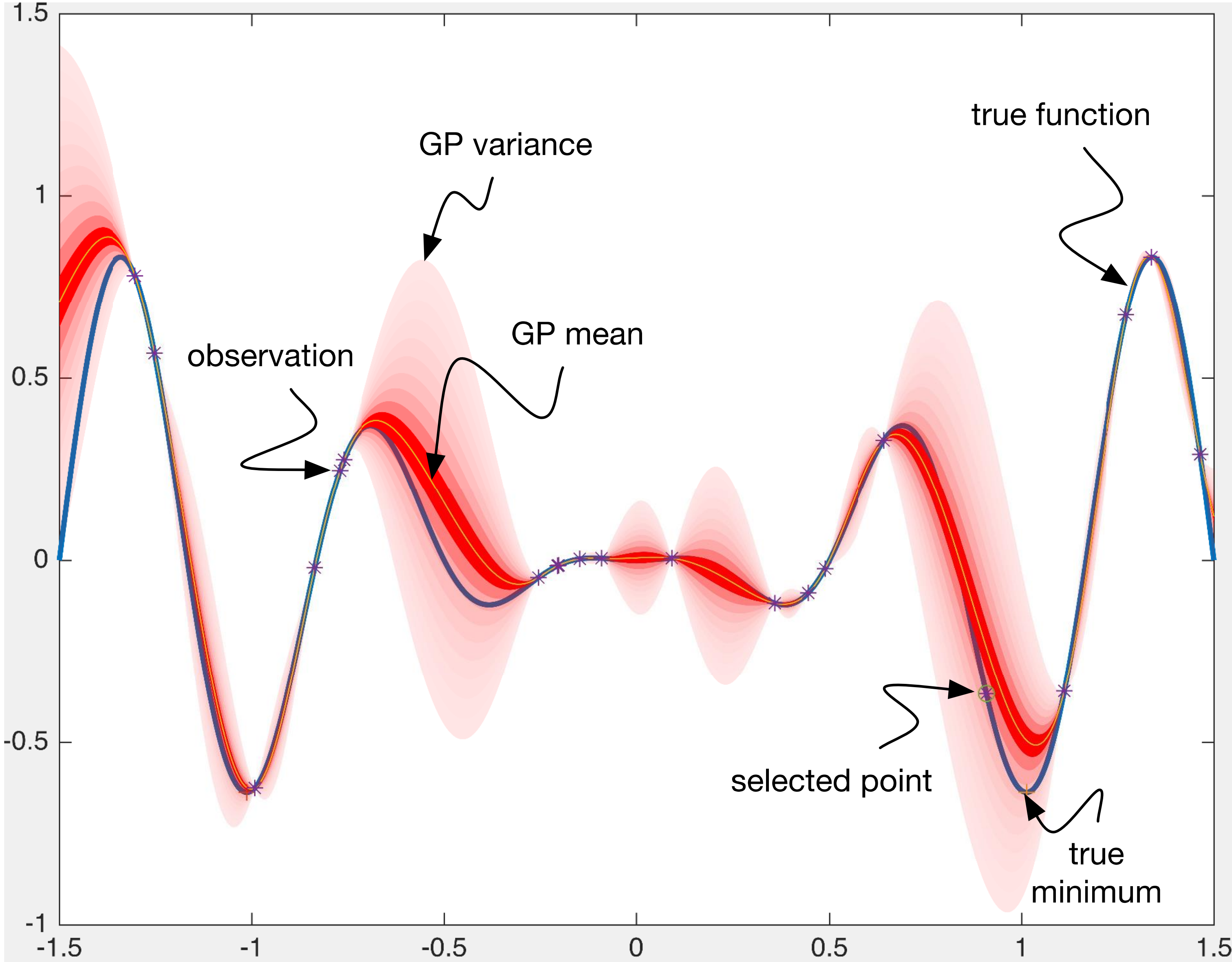
$$y = f(\boldsymbol{x}) \sim \mathcal{GP}(\mu(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')),$$

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

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

$$\boldsymbol{K} := \begin{bmatrix} k(\boldsymbol{x}_1, \boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_1, \boldsymbol{x}_t) \\ \vdots & \ddots & \vdots \\ k(\boldsymbol{x}_t, \boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_t, \boldsymbol{x}_t) \end{bmatrix}$$

$$k(f, g, \boldsymbol{x}, \boldsymbol{x}') = k_t(f, g) \times k_{xx}(\boldsymbol{x}, \boldsymbol{x}'),$$



Scenarios and Assumptions

Environment change (configuration option vs. environment change)

- Different benchmark/workload
- Different program version
- Different hardware

Shape of old and new model similar

Part

Some Results ("it works")

First Feasibility Demonstration

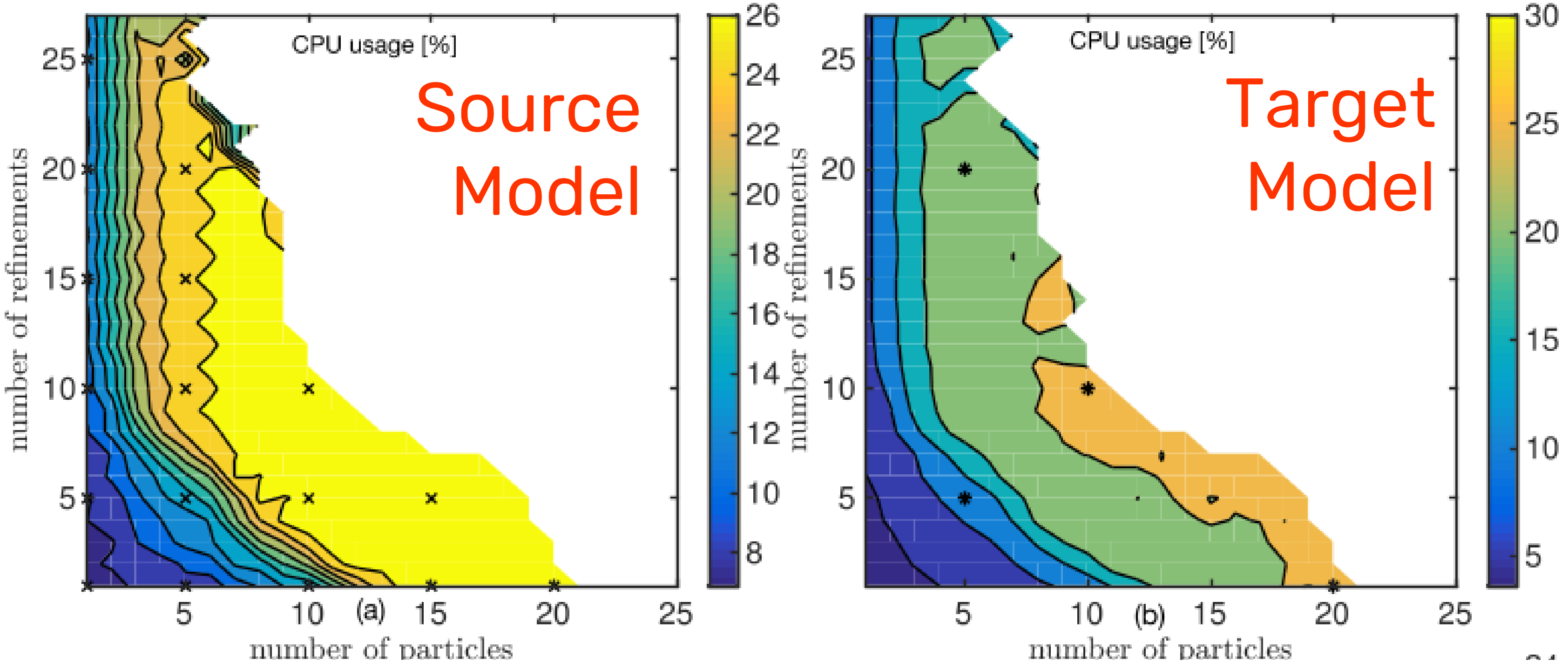
[SEAMS 2017]

Case study & controlled experiments

- Can we improve prediction accuracy?
- Tradeoffs among #source and #target samples?
- Fast enough?

Subject sys.: Cobot, Apache Storm, Cassandra

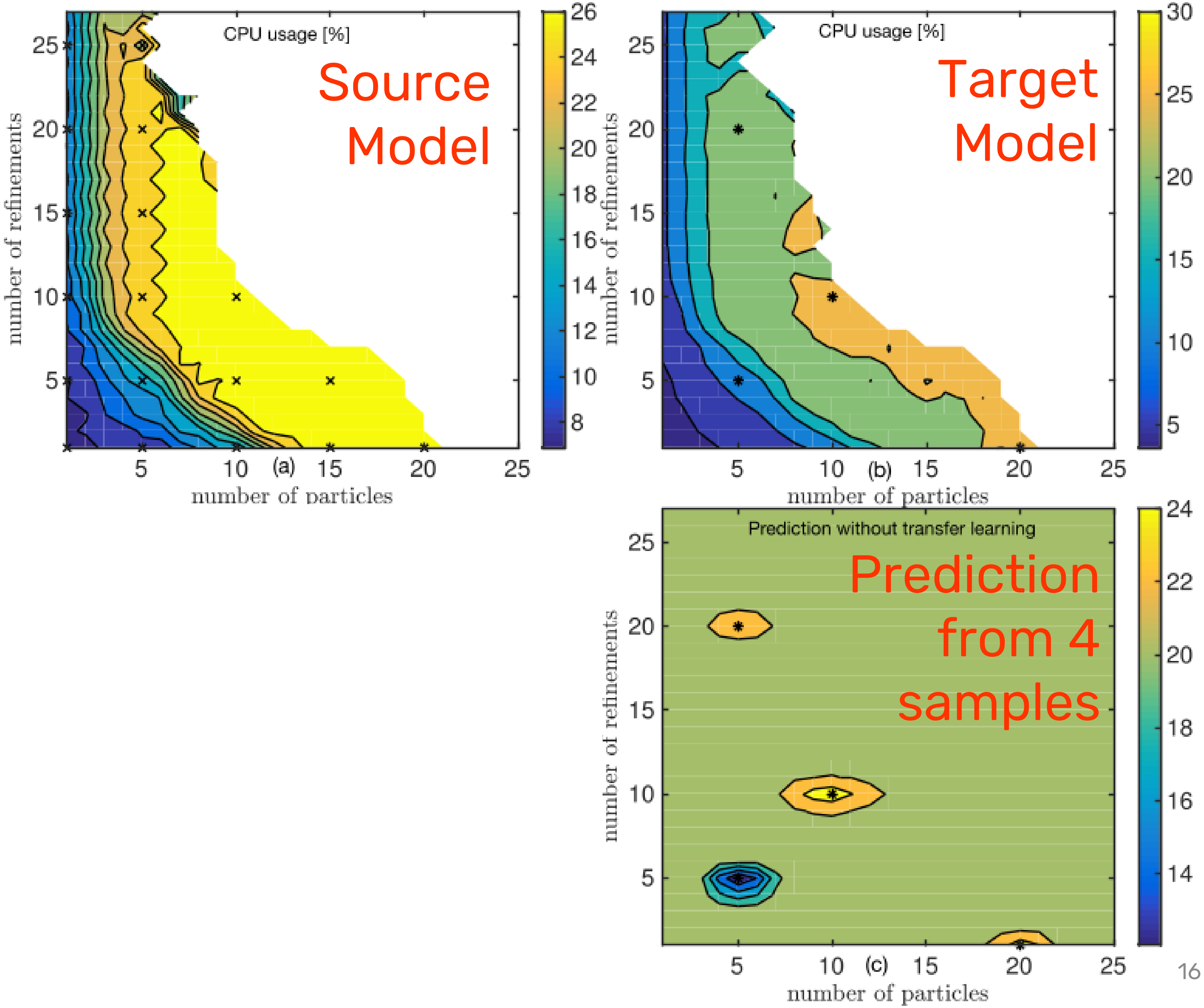
Example: Performance prediction for CoBot



Source Model

Target Model

Example: Performance prediction for CoBot

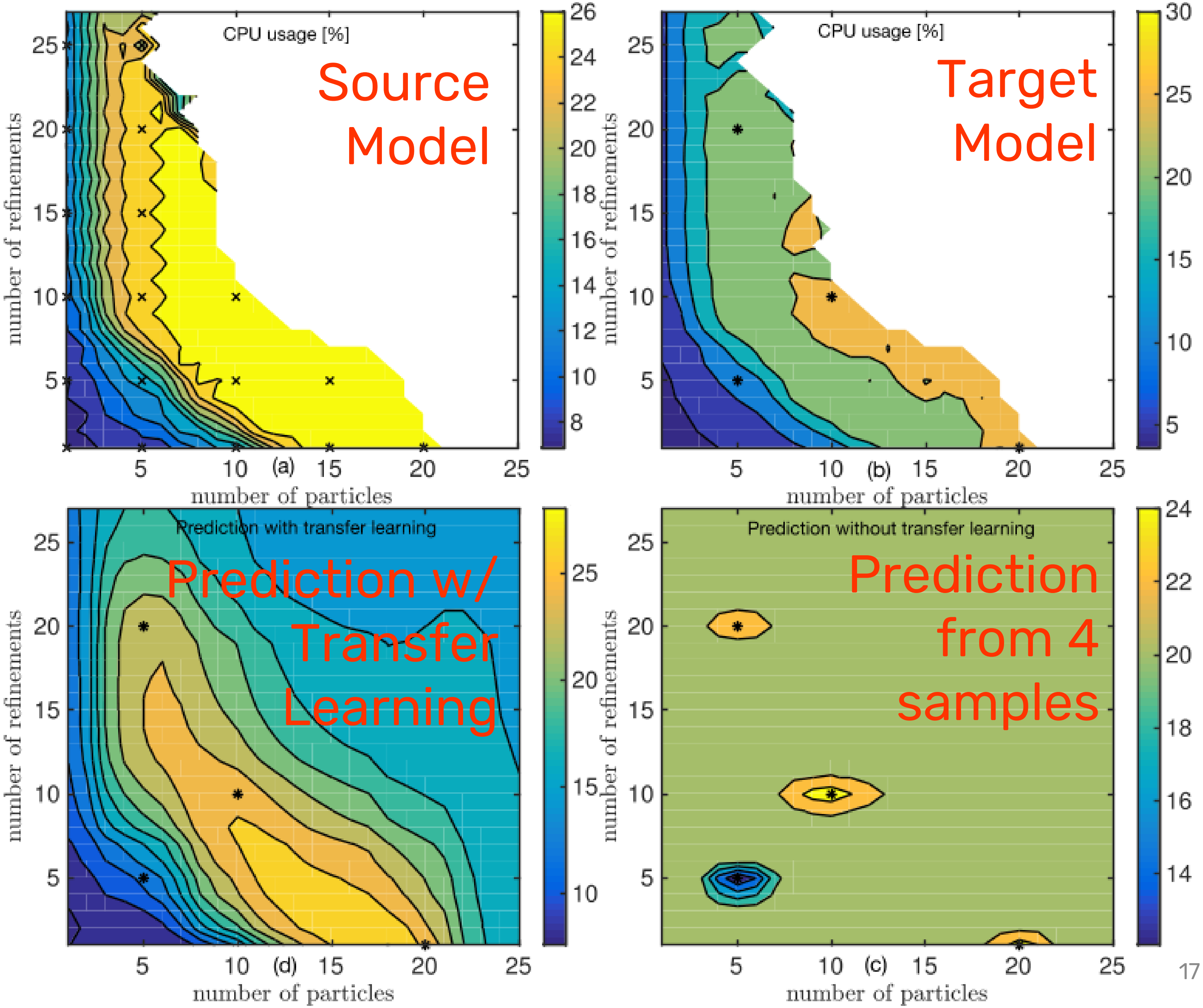


Source Model

Target Model

Prediction from 4 samples

Example: Performance prediction for CoBot



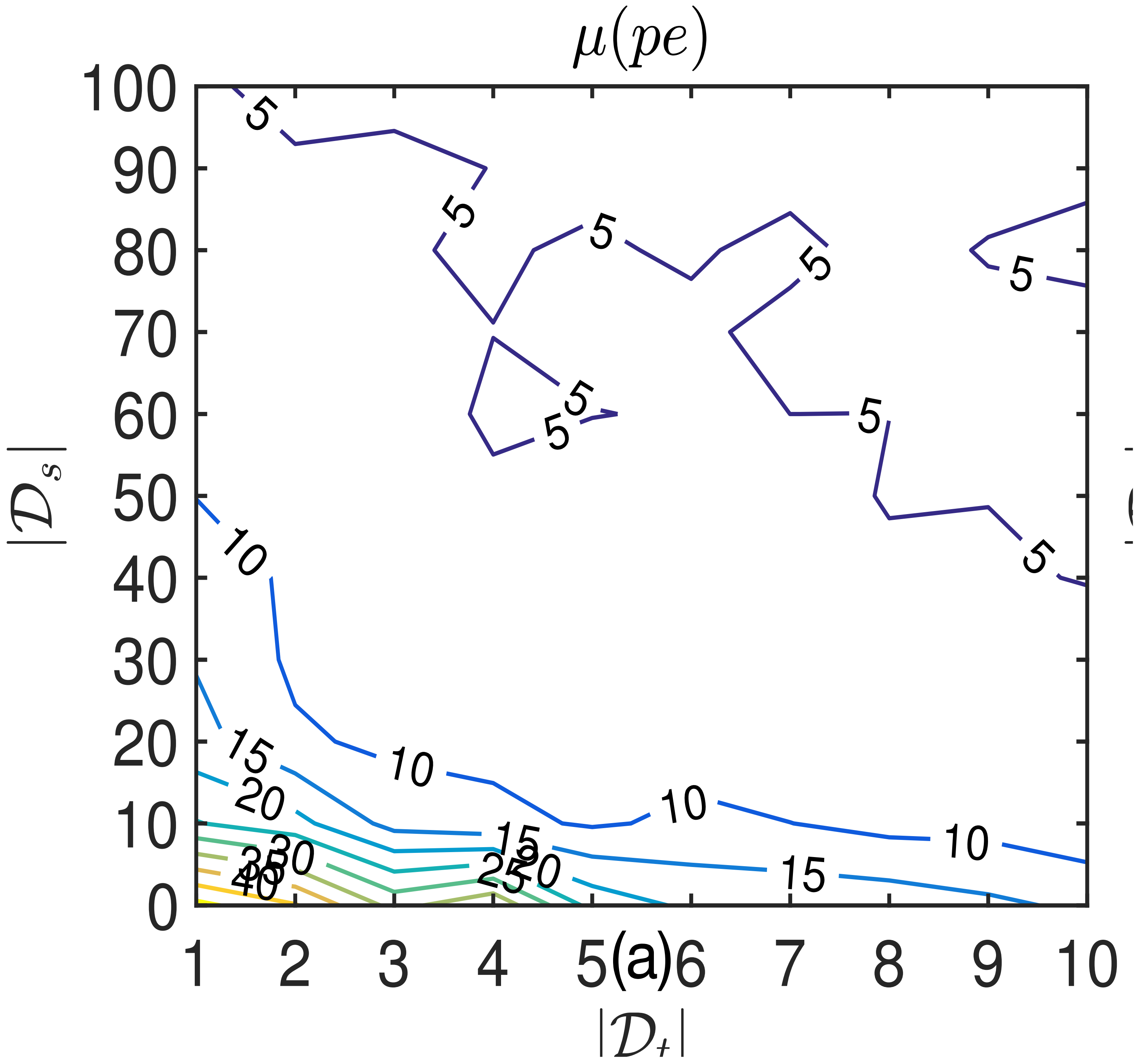
Source Model

Target Model

Prediction w/ Transfer Learning

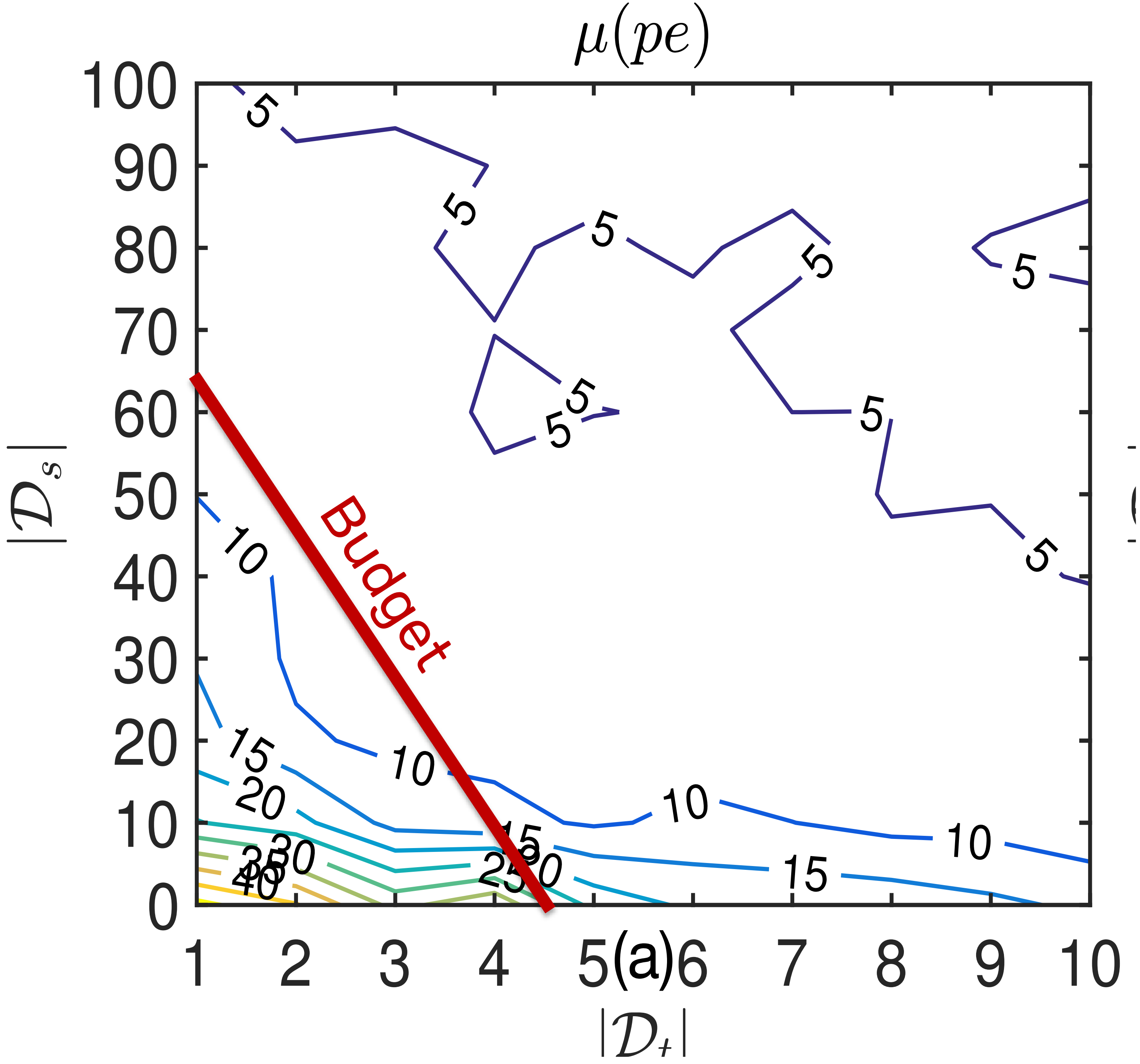
Prediction from 4 samples

Accuracy and Costs



(a)

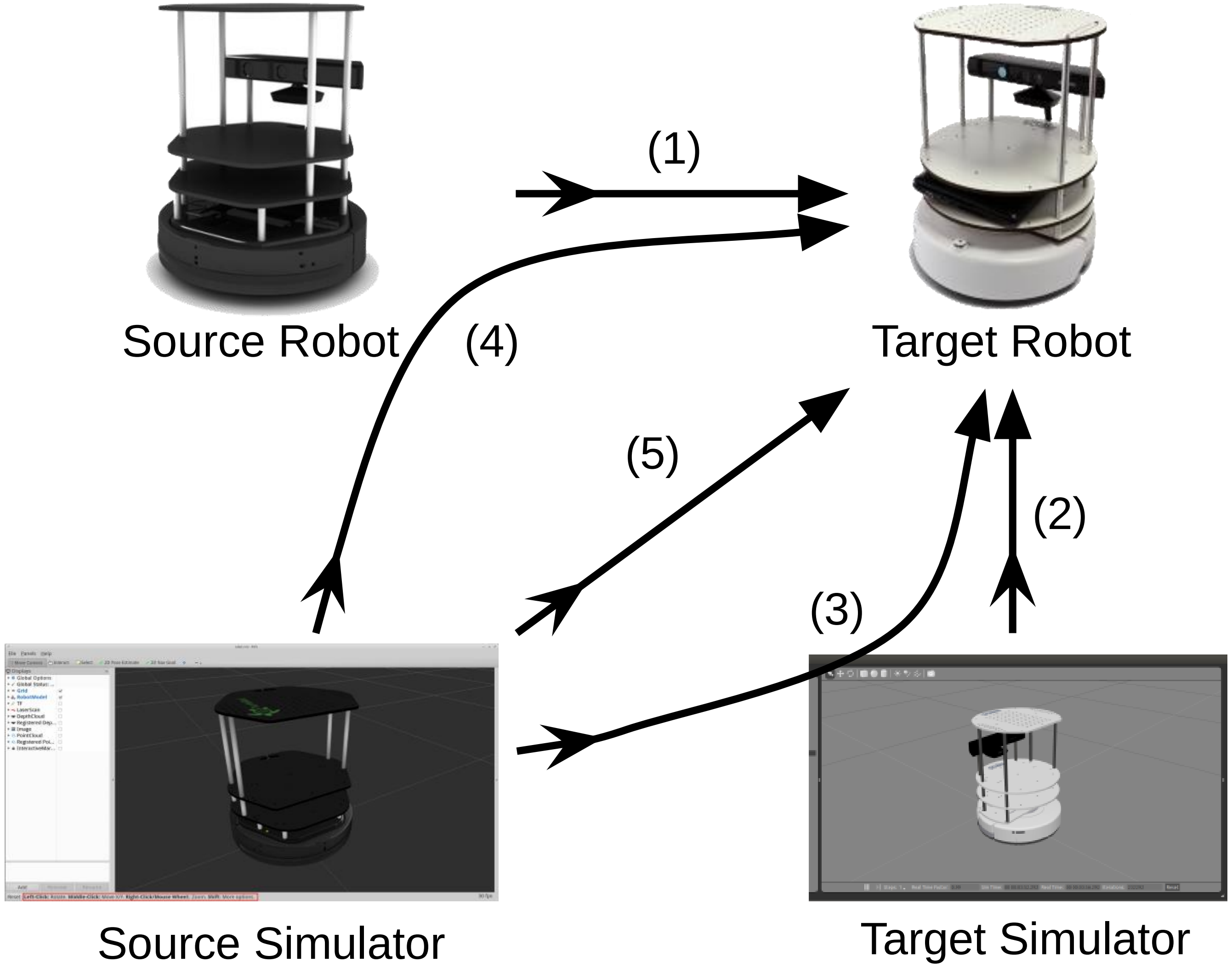
Accuracy and Costs



Part

Future work, insights and ideas

Selecting from Multiple Sources (Cost Model)



Source Robot

Target Robot

Source Simulator

Target Simulator

(1)

(2)

(3)

(4)

(5)

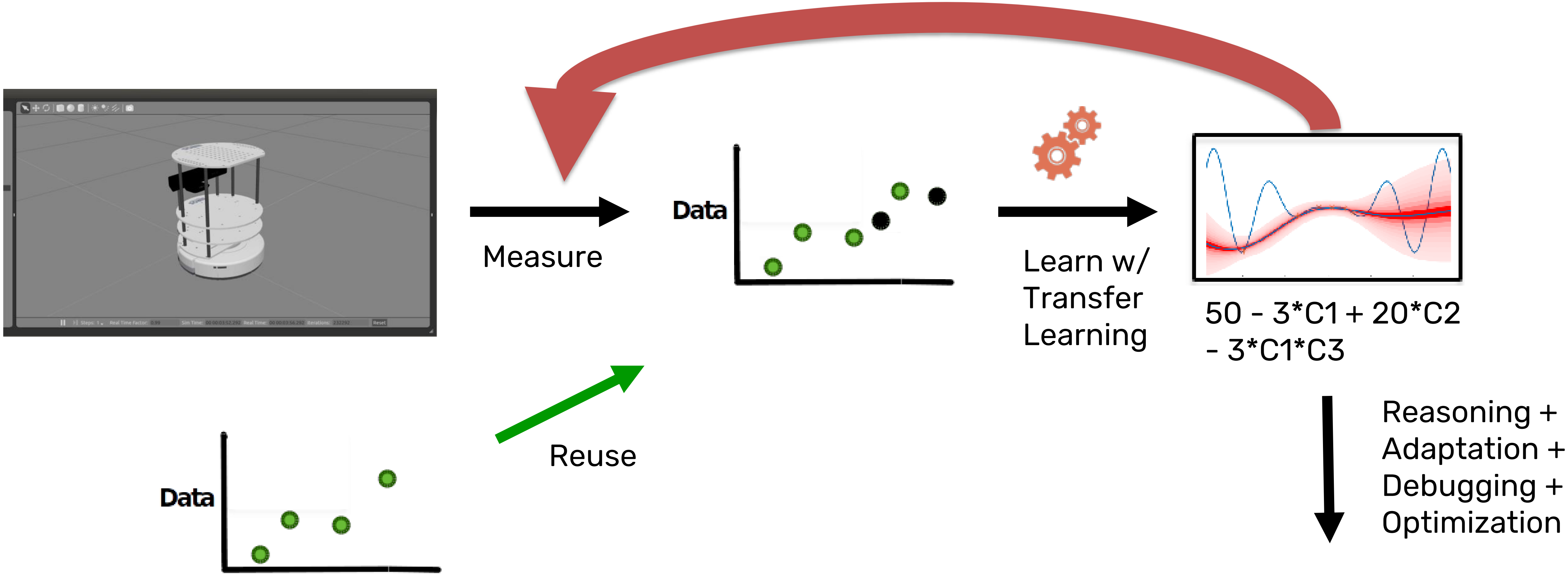
Checking Assumptions

How similar are source and target models for real environment changes (workload/infrastructure/code changes)?

Expected similarities:

- Constant
- Constant / proportional shift
- More noise but similar trends
- Many features and interactions with similar impact
- Many important features and interactions remain important

Active learning + transfer learning



Measure

Data

Learn w/ Transfer Learning

50 - 3*C1 + 20*C2 - 3*C1*C3

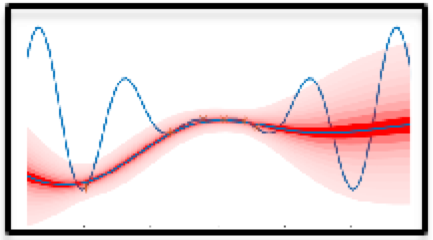
Reasoning + Adaptation + Debugging + Optimization

Data

Reuse

Goal: find best sample points iteratively by gaining knowledge from source and target domain

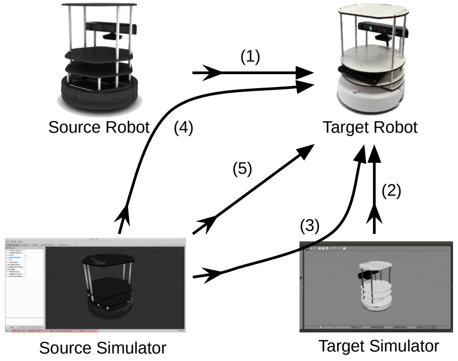
Transfer Learning
for Improving Model Predictions
in Highly Configurable Software



Improves the model accuracy up to several orders of magnitude

Is able to trade-off between different number of samples from source and target enabling a cost-aware model

Exploring similarity across environment changes and active learning


Source Robot
Target Robot
(1)
(4)
(5)
(2)
(3)
Source Simulator
Target Simulator