Surrogate Benchmarks for Hyperparameter Optimization

Katharina Eggensperger
Frank Hutter

University of Freiburg
{eggenspk,fh}@cs.uni-freiburg.de

Holger Hoos
Kevin Leyton-Brown

University of British Columbia
{hoos,kevinlb}@cs.ubc.ca

UNI FREIBURG

Problem:

Evaluation of Methods for Hyperparameter Optimization is expensive !

UNI FREIBURG

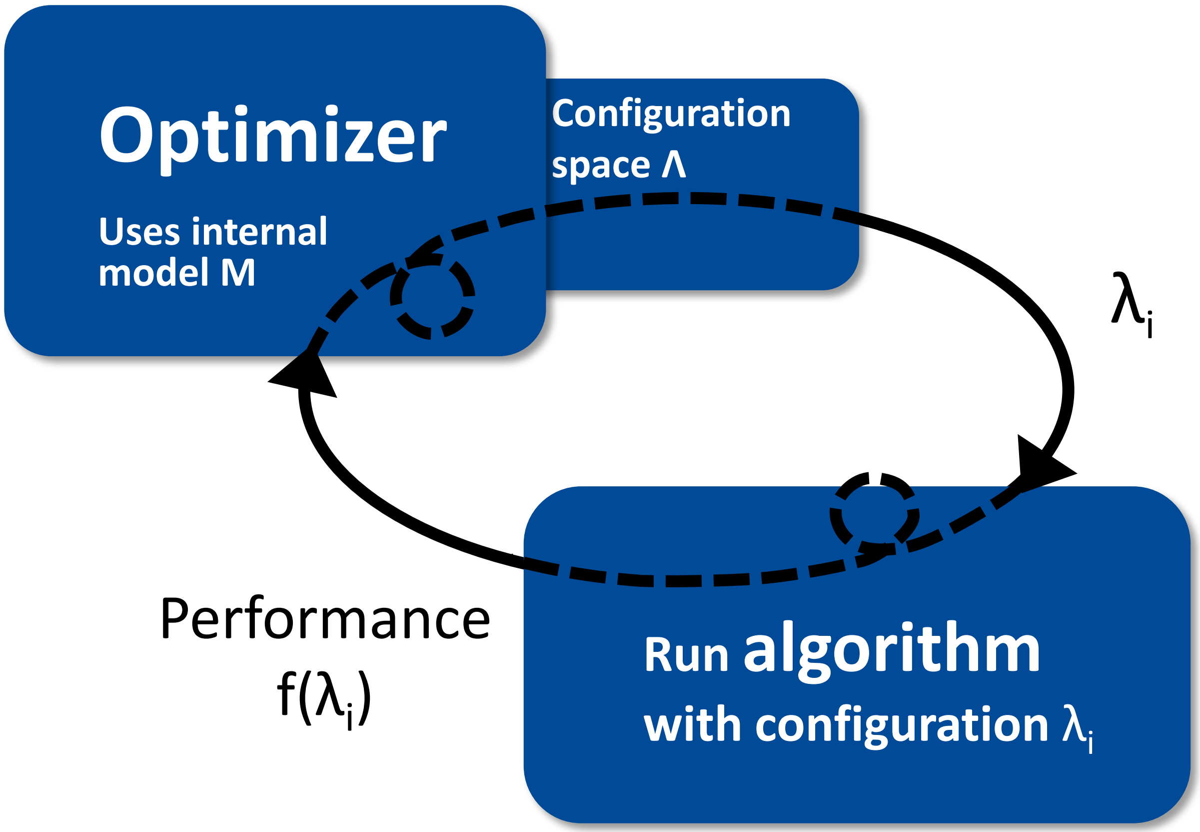
Outline

- Benchmarking Hyperparameter Optimization Methods
- Constructing Surrogates
- Using Surrogate Benchmarks

Outline

- Benchmarking Hyperparameter Optimization Methods
- Constructing Surrogates
- Using Surrogate Benchmarks

Bayesian Optimization Methods



What do we need for an empirical comparison

- Standard benchmark problems
- Easy-to-use software

Then:

- Run each optimizer on each benchmark X multiple times

What do we need for an empirical comparison

- Standard benchmark problems
- Easy-to-use software

Then:

- Run each optimizer on each benchmark X multiple times

Evaluation of X is expensive

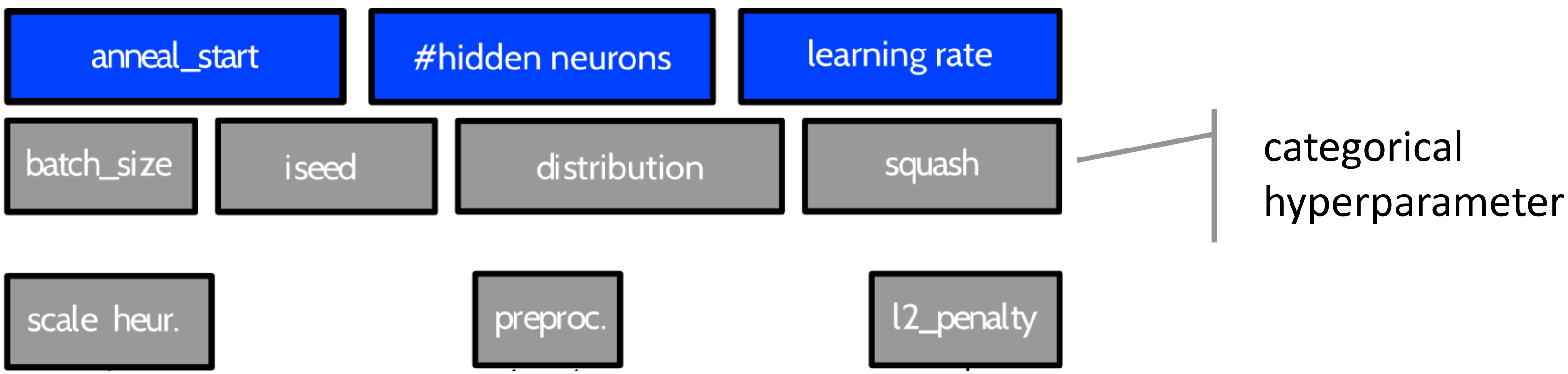
Benchmarking hyperparameter optimization methods

Neural Network, configuration space Λ:

- anneal_start
- #hidden neurons
- learning rate
- batch_size
- iseed
- distribution
- squash
- scale heur.
- preproc.
- l2_penalty

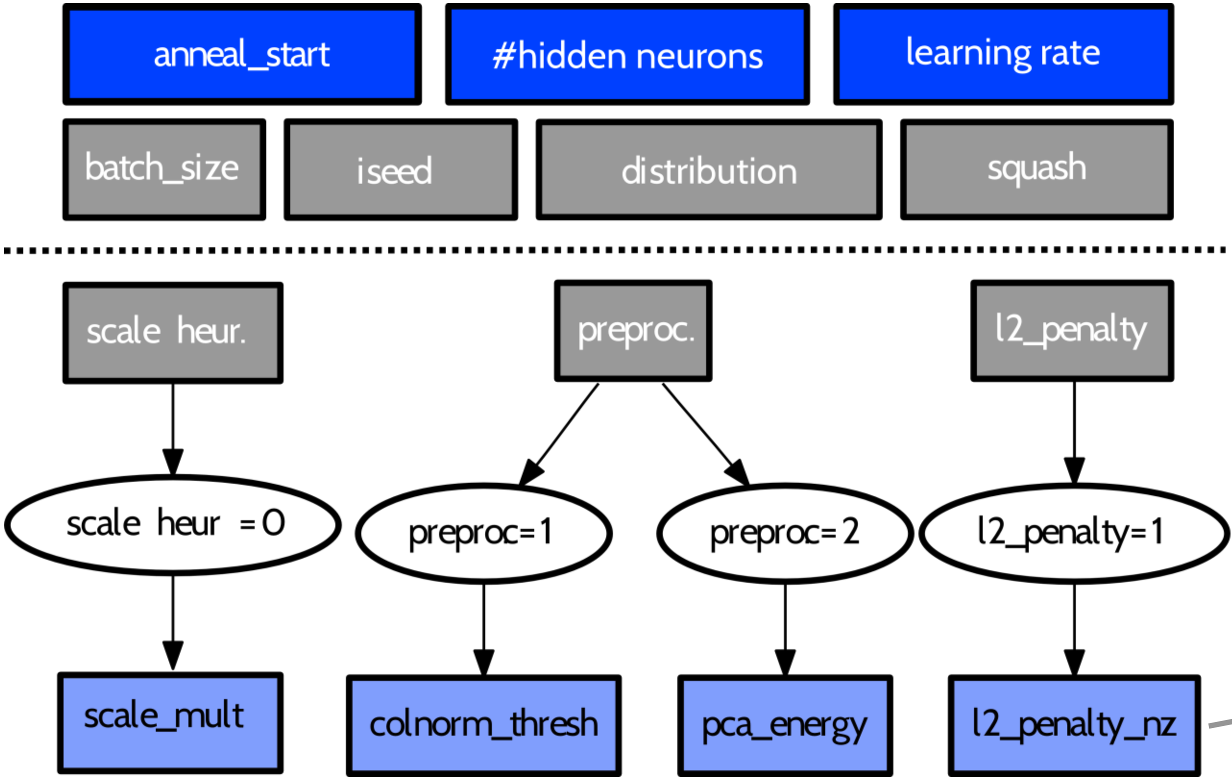
Benchmarking hyperparameter optimization methods

Neural Network, configuration space Λ:



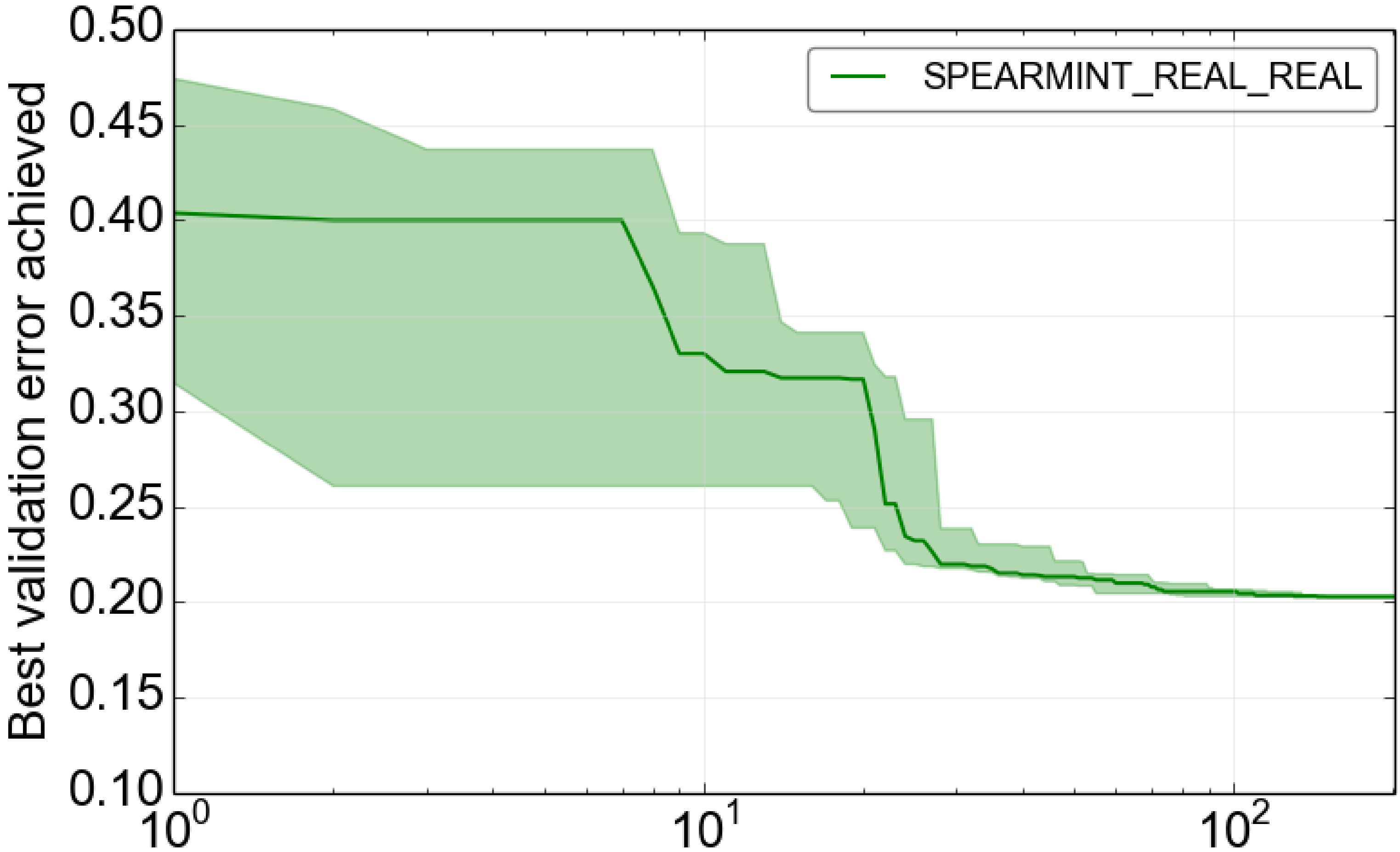
Benchmarking hyperparameter optimization methods

Neural Network, configuration space Λ:



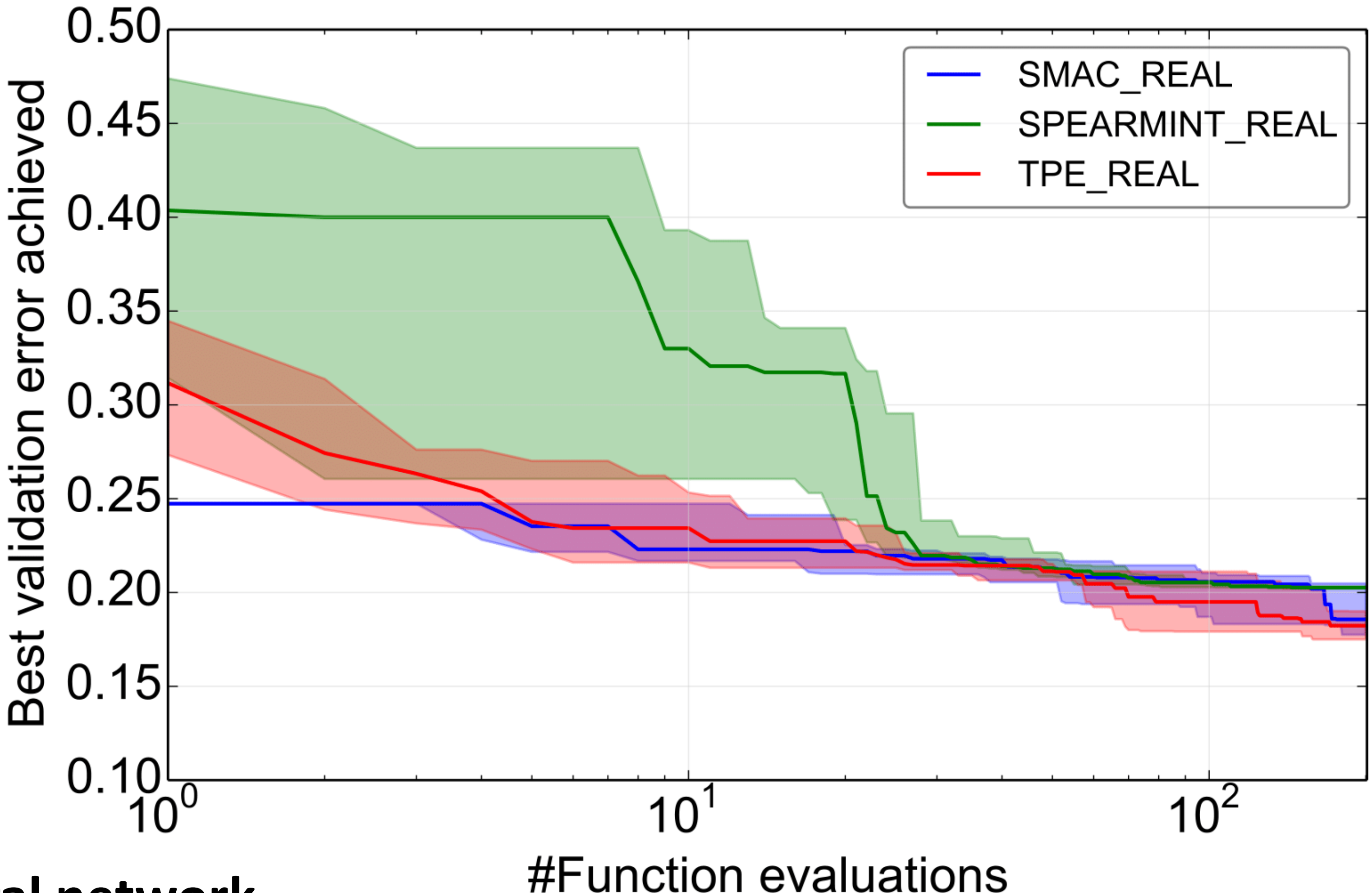
conditional hyperparameter

Benchmarking hyperparameter optimization methods



Neural network

Benchmarking hyperparameter optimization methods



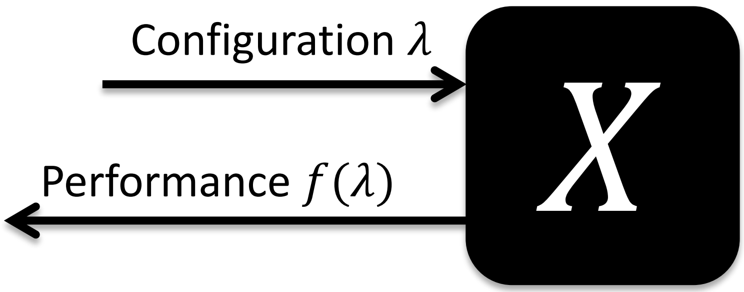
Neural network

Outline

- Benchmarking Hyperparameter Optimization Methods
- Constructing Surrogates
- Using Surrogate Benchmarks

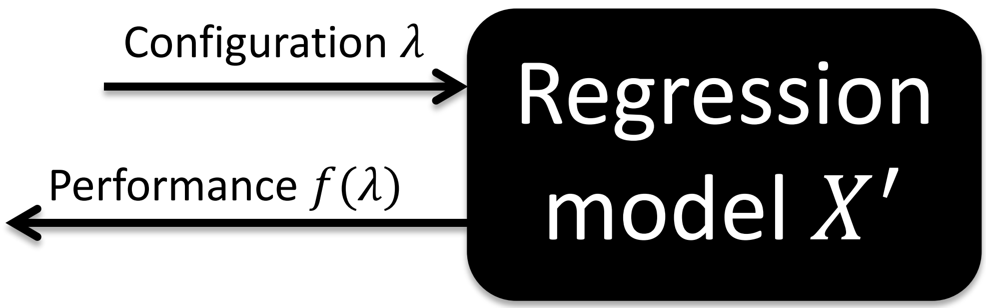
Surrogate Benchmark X'

- cheap-to-evaluate
- Can be used like the real benchmark X
- Behaves like X



Surrogate Benchmark X'

- cheap-to-evaluate
- Can be used like the real benchmark X
- Behaves like X



Constructing a Surrogate for Benchmark X

UNI FREIBURG

1. Collect data
2. Choose a regression model
3. Train and store model

1. Collect data for benchmark X

Training data: $\left((\lambda_1, f(\lambda_1)), \ldots, (\lambda_n, f(\lambda_n))\right)$

- Dense sampling in high performance regions

- Good overall coverage

1. Collect data for benchmark X

Training data: $\left((\lambda_1, f(\lambda_1)), \ldots, (\lambda_n, f(\lambda_n))\right)$

- Dense sampling in high performance regions

Run optimizers on benchmark X

- Good overall coverage

1. Collect data for benchmark X

Training data: $\left((\lambda_1, f(\lambda_1)), \dots, (\lambda_n, f(\lambda_n))\right)$

- Dense sampling in high performance regions

Run optimizers on benchmark X

- Good overall coverage

Run random search on benchmark X

2. Choice of Regression Models

Ridge Regression

K-nearest neighbour

Gradient Boosting

Linear Regression

Random Forests

Gaussian Processes

Bayesian Neural Network

SVM

2. Choice of Regression Models

Ridge Regression

K-nearest neighbour

Gradient Boosting

Linear Regression

Random Forests

Gaussian Processes

Bayesian Neural Network

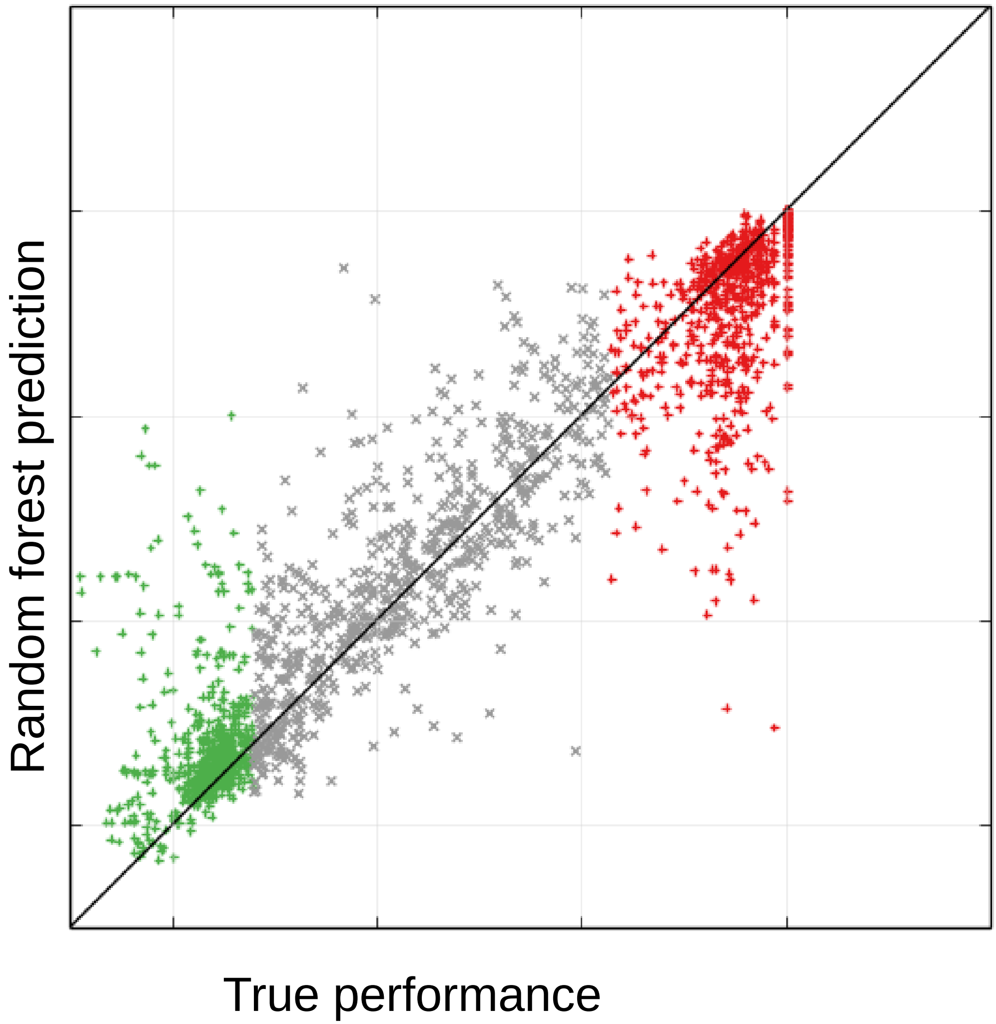
SVM

2. Choice of Regression Models

Can we quantify the performance of a new optimizer?

- **Leave-one-optimizer-out** setting
  - Train model on data gathered by all but one optimizer
  - Test on remaining data

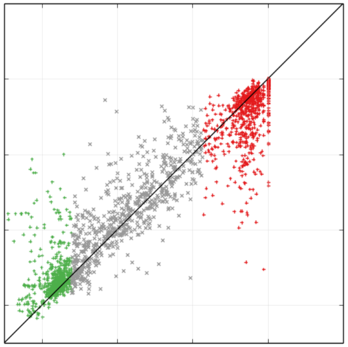
2. Choice of Regression Models
Leave-one-optimizer-out setting



Neural Network

2. Choice of Regression Models
Leave-one-optimizer-out setting

Random Forest



Neural Network

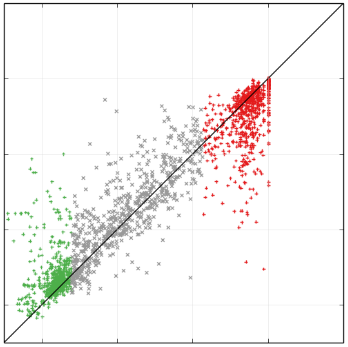
2. Choice of Regression Models

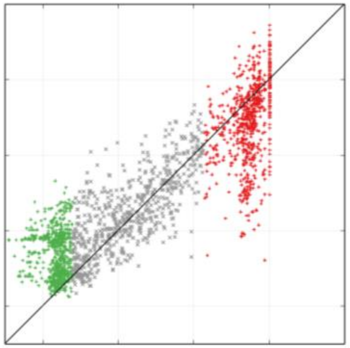
Leave-one-optimizer-out setting

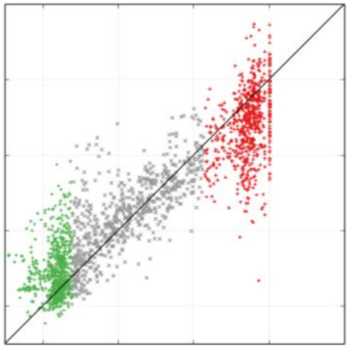
Random Forest

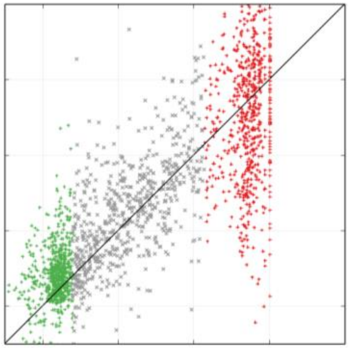
Gaussian Process

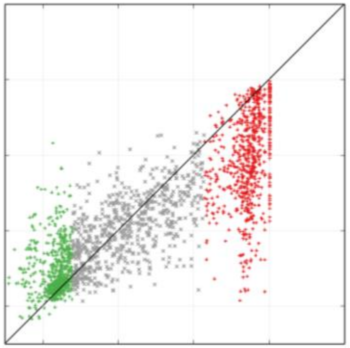
k-nearest-neighbour











Gradient Boosting

nuSVR

Neural Network

2. Choice of Regression Models

Ridge Regression

K-nearest neighbour

Gradient Boosting

Linear Regression

Random Forests

Gaussian Processes

Bayesian Neural Network

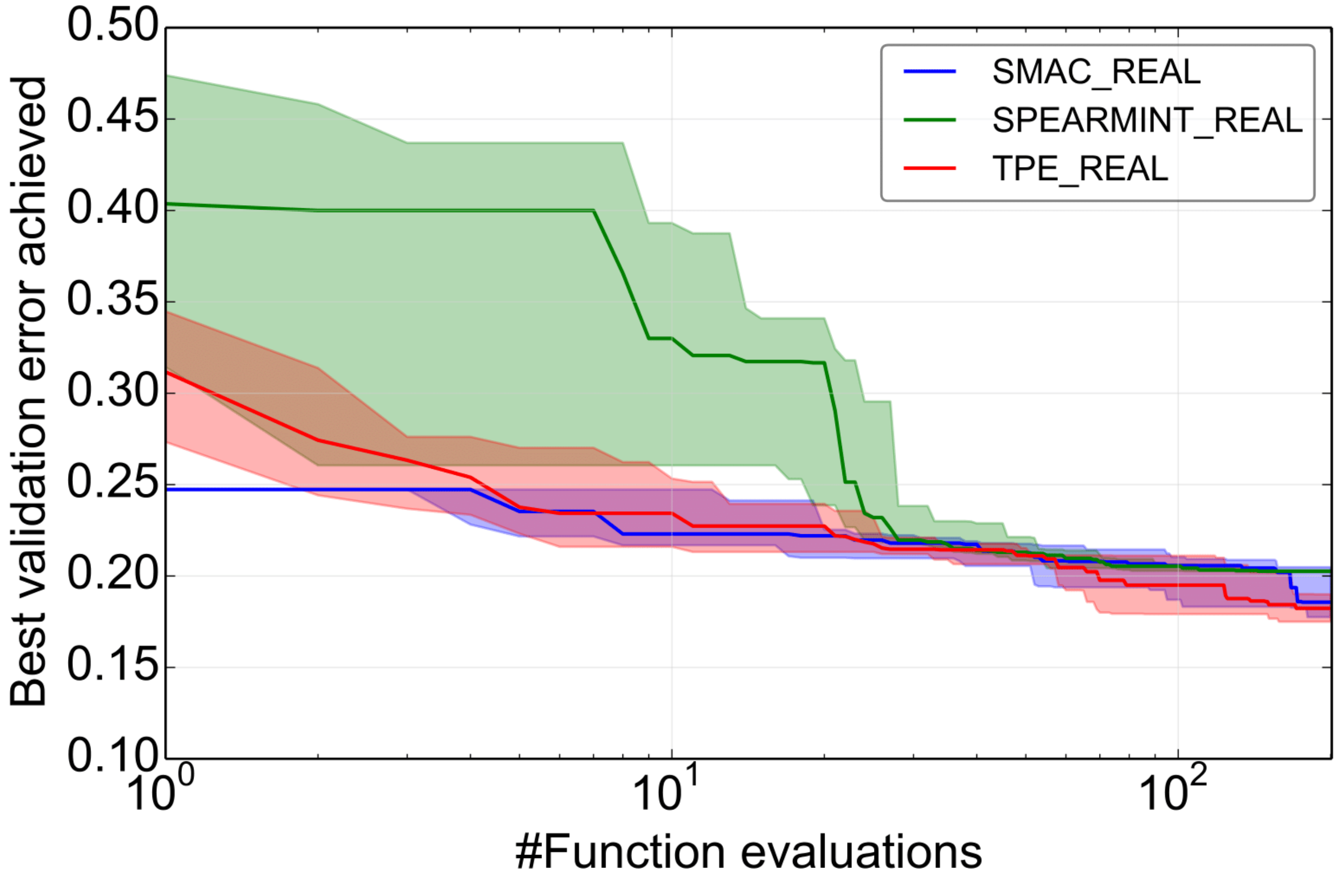
SVM

Outline

- Benchmarking Hyperparameter Optimization Methods
- Constructing Surrogates
- Using Surrogate Benchmarks

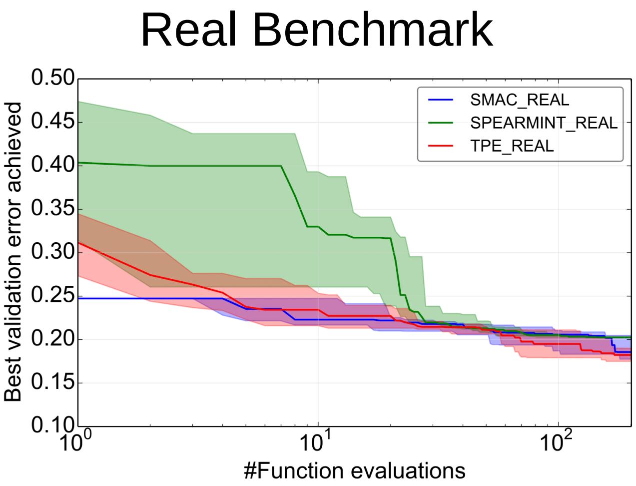
Using Surrogate Benchmarks

Neural Network



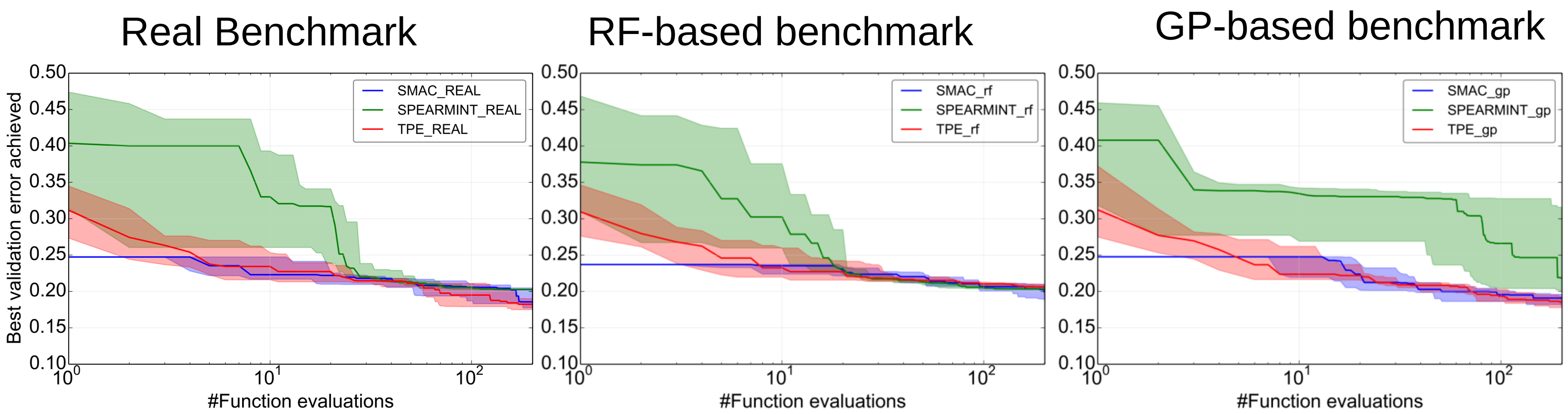
Using Surrogate Benchmarks

Neural Network



Using Surrogate Benchmarks

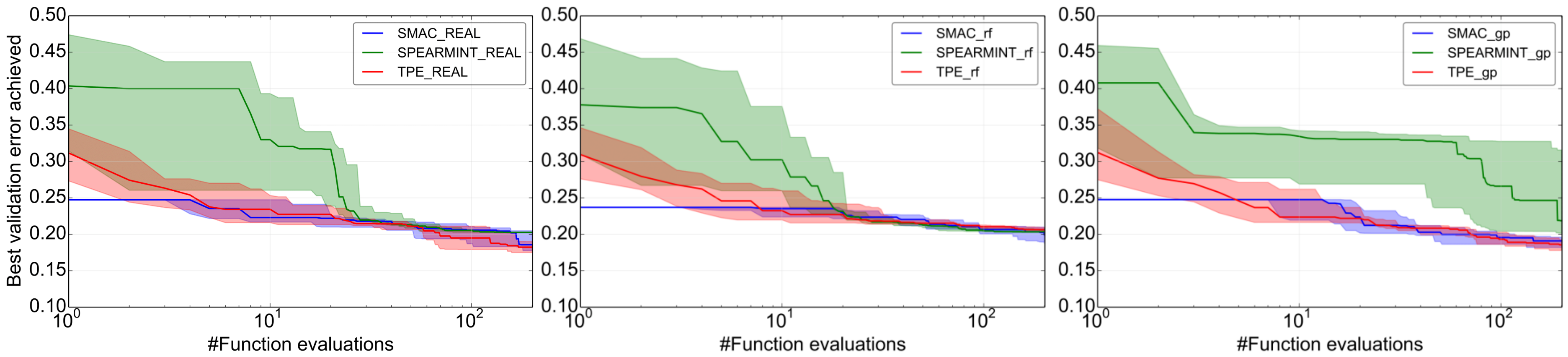
Neural Network



Using Surrogate Benchmarks

Neural Network

Real Benchmark | RF-based benchmark | GP-based benchmark



	Real Benchmark	RF-based benchmark	GP-based benchmark
One optimization run:	40h	<200s	<200s
Whole comparison:	50d	<1.5h	<1.5h

Applications

- Extensive testing at **early development stages**
- **Fast comparison** of different hyperparameter optimization methods
- **Metaoptimization** of existing hyperparameter optimization methods

Conclusion

Can we construct **cheap-to evaluate** and **realistic** hyperparameter optimization benchmarks?

Yes, based on random forests and
Gaussian process regression models

Conclusion

UNI FREIBURG

Can we construct **cheap-to evaluate** and **realistic** hyperparameter optimization benchmarks?

Yes, based on random forests and Gaussian process regression models

But, some work needs to be done for high dimensional benchmarks.

This presentation was supported by an *ECCAI travel award* and the *ECCAI sponsors*

Thank you for your attention

More information on hyperparameter optimization benchmarks can be found on **automl.org/hpolib**

Regression models

Model	Hyperparameter optimization
Random Forest (RF)	None
Gradient Boosting (GB)	None
Extra Trees	None
Gaussian process (GP), Matérn 5/2 kernel	MCMC sampling over hyperparameters
Support Vector Regression (SVR)	Random search for C and gamma
NuSVR	Random search for C, gamma and nu
Bayesian neural network (BNN)	None
k-nearest neighbour (KNN)	Random search for n_neighbors
Linear Regression	None
Least Angle Regression	None
Ridge Regression	None

Benchmarks

	#λ	hyperparameter cond.	cat. / cont.	Input dim.	#evals. per run	#data
Branin	2	-	- / 2	3	200	7402
Log. Reg. 5CV	4	-	- / 4	9	500	18521
HP-NNET convex	14	4	7 / 7	25	200	7750
HP-DBNET mrbi	36	27	19 / 17	82	200	7466