

AUG 07, 2020

Hyperparameter Optimization and DeepHyper

Argonne
NATIONAL LABORATORY



Prasanna Balaprakash and Misha Salim
Argonne National Laboratory

Team



Prasanna Balaprakash,
MCS & ALCF



Romain Egele,
MCS



Misha Salim, ALCF



Stefan Wild, MCS



Venkat Vishwanath,
ALCF



Taylor Childers,
ALCF



Tom Uram,
ALCF



Elise Jennings,
ALCF-Alumni



Romit Maulick, ALCF



Bethany Lusch, ALCF



Epoch
001,644

Learning rate
0.03

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



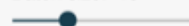
Ratio of training to test data: 50%



Noise: 0



Batch size: 10



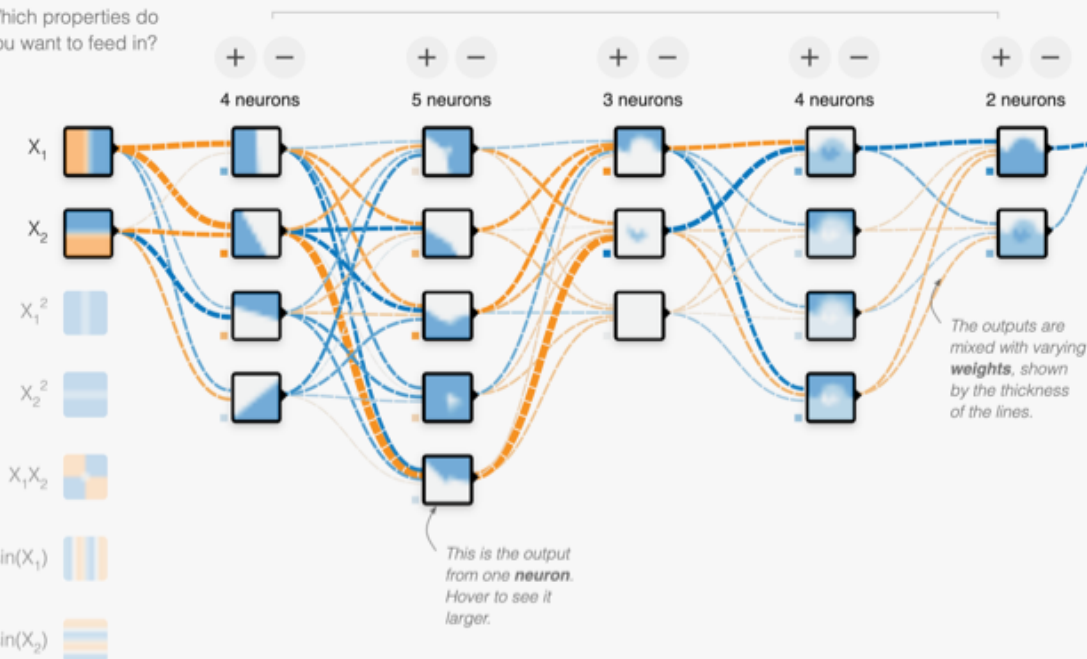
REGENERATE

FEATURES

Which properties do you want to feed in?

- ☒ X_1
- ☒ X_2
- ☐ X_1^2
- ☐ X_2^2
- ☐ $X_1 X_2$
- ☐ $\sin(X_1)$
- ☐ $\sin(X_2)$

+ - 5 HIDDEN LAYERS

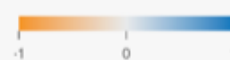


OUTPUT

Test loss 0.316
Training loss 0.321



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output



Epoch
001,142

Learning rate
0.03

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

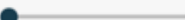
Which dataset do you want to use?



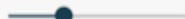
Ratio of training to test data: 50%



Noise: 0



Batch size: 10



REGENERATE

FEATURES

Which properties do you want to feed in?

X_1



X_2



X_1^2



X_2^2



$X_1 X_2$



$\sin(X_1)$



$\sin(X_2)$



+ - 5 HIDDEN LAYERS

+ -

4 neurons

+ -

4 neurons

+ -

4 neurons

+ -

4 neurons

+ -

2 neurons



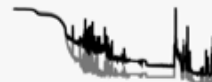
This is the output from one **neuron**. Hover to see it larger.

The outputs are mixed with varying **weights**, shown by the thickness of the lines.

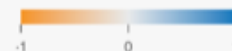
OUTPUT

Test loss 0.063

Training loss 0.015



Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output



Epoch
001,442

Learning rate
0.03

Activation
ReLU

Regularization
None

Regularization rate
0

Problem type
Classification

DATA

Which dataset do you want to use?



Ratio of training to test data: 50%



Noise: 0



Batch size: 10



REGENERATE

FEATURES

Which properties do you want to feed in?

X_1



X_2



X_1^2



X_2^2



$X_1 X_2$



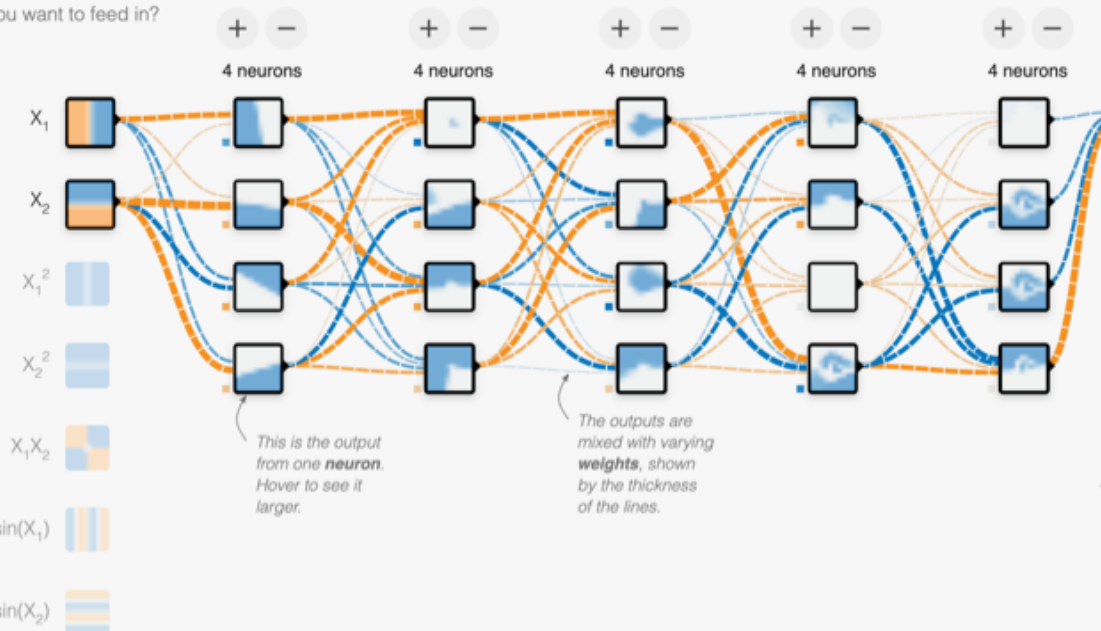
$\sin(X_1)$



$\sin(X_2)$

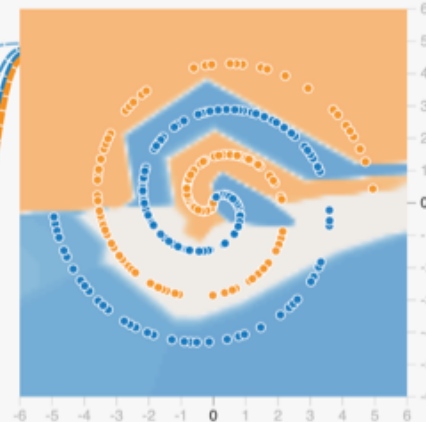
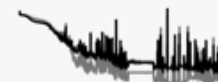


+ - 5 HIDDEN LAYERS

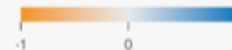


OUTPUT

Test loss 0.239
Training loss 0.146



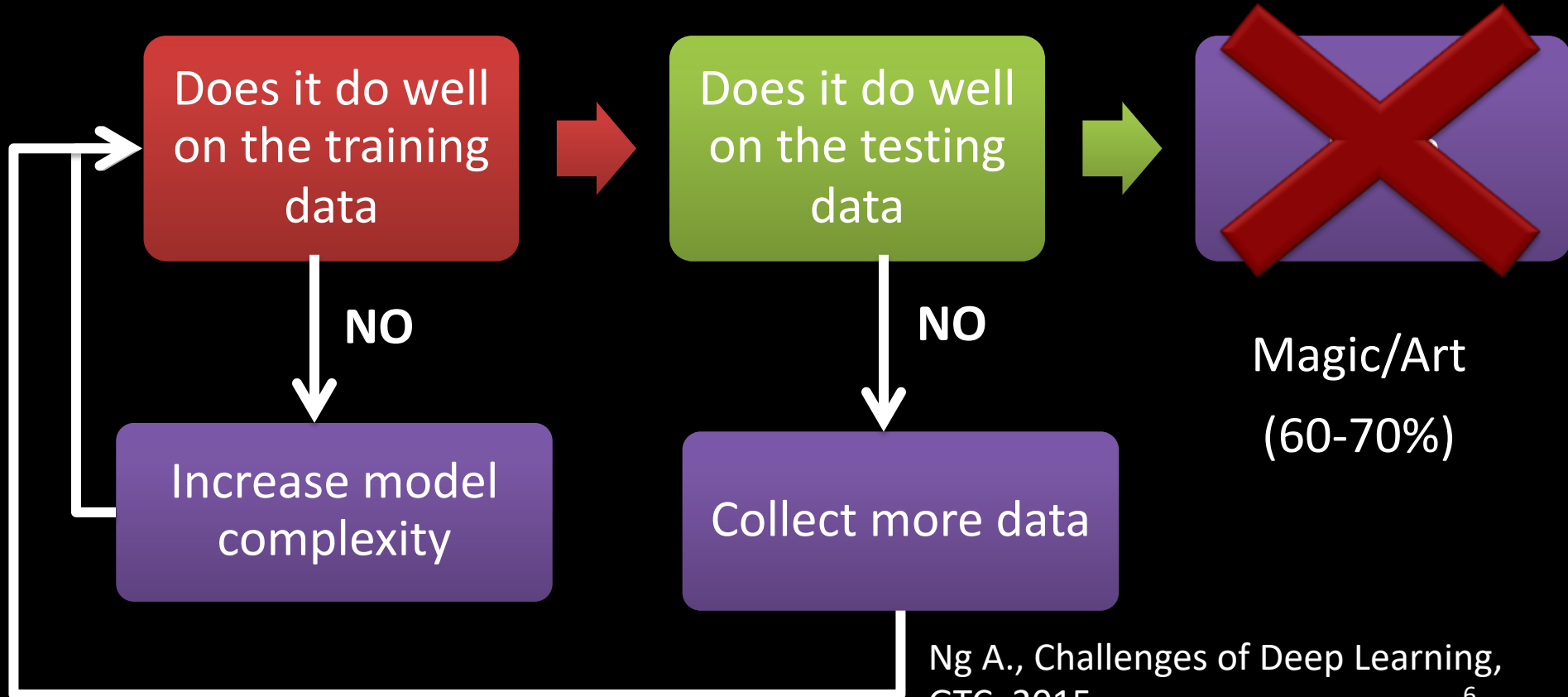
Colors shows data, neuron and weight values.



☐ Show test data

☐ Discretize output

Supervised (deep) learning



Automated machine learning for deep learning

Lower-level problem:

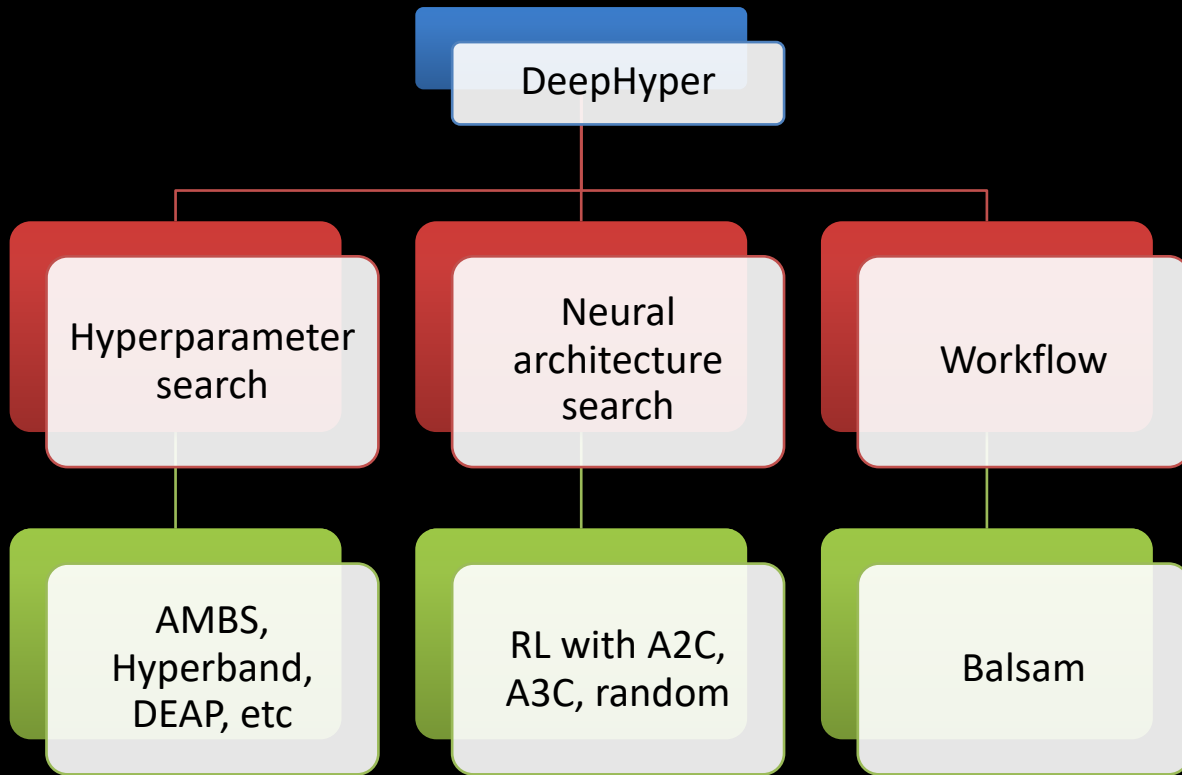
$$\text{solve} \quad \underset{w}{\text{minimize}} \quad \text{err}_T([\mathcal{X}_A, \mathcal{X}_P]; \mathcal{T}; w)$$

Upper-level problem:

$$\text{solve} \quad \underset{\mathcal{X}_A, \mathcal{X}_P}{\text{minimize}} \quad \text{err}_V([\mathcal{X}_A, \mathcal{X}_P]; \mathcal{V}; w^*[\mathcal{X}_A, \mathcal{X}_P])$$

Architecture space Hyperparameter space

DeepHyper: Scalable AutoML package

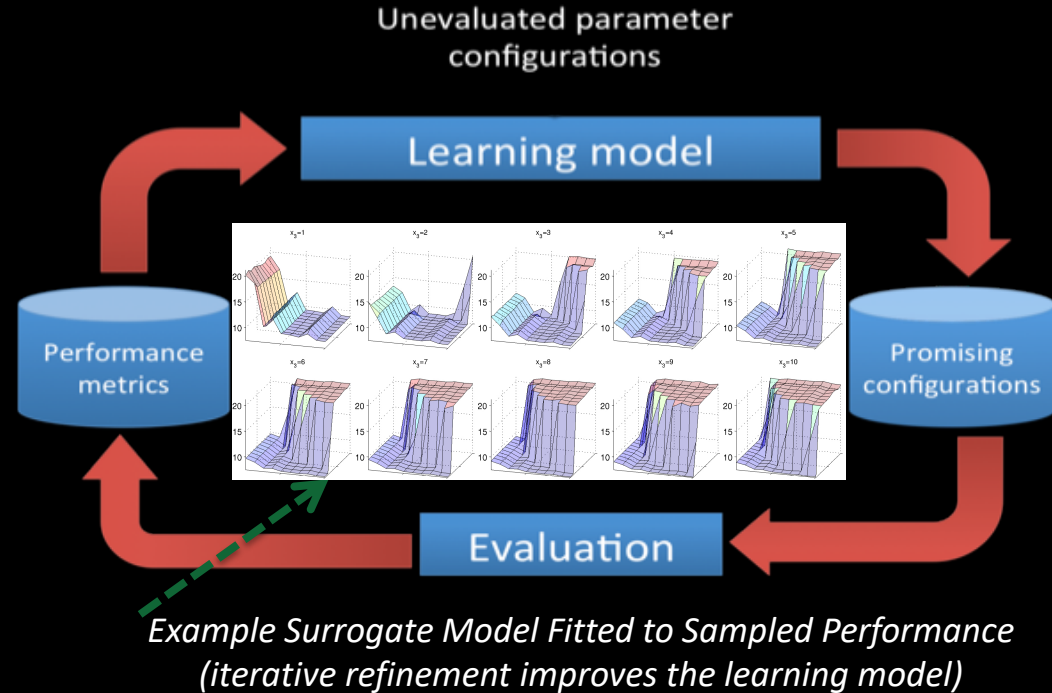


<https://github.com/deephyper/deephyper>

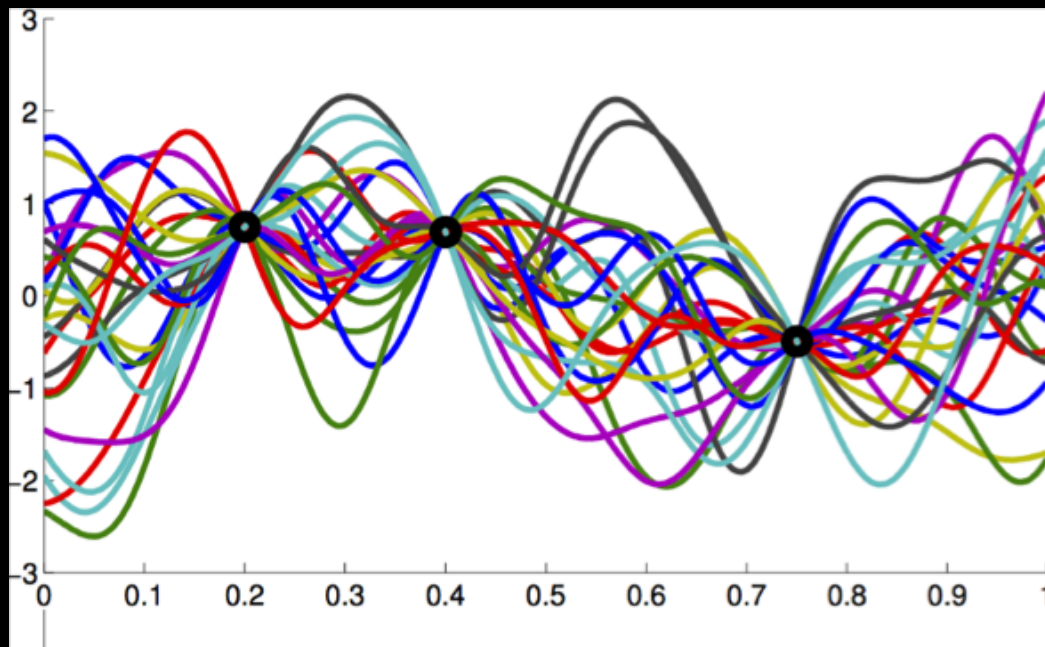
AMBS: Asynchronous model-based search

– Framework:

- Initialization phase
 - Random or Latin hypercube sampling
- Iterative phase
 - Fit model
 - Sample using the model



Bayesian optimization



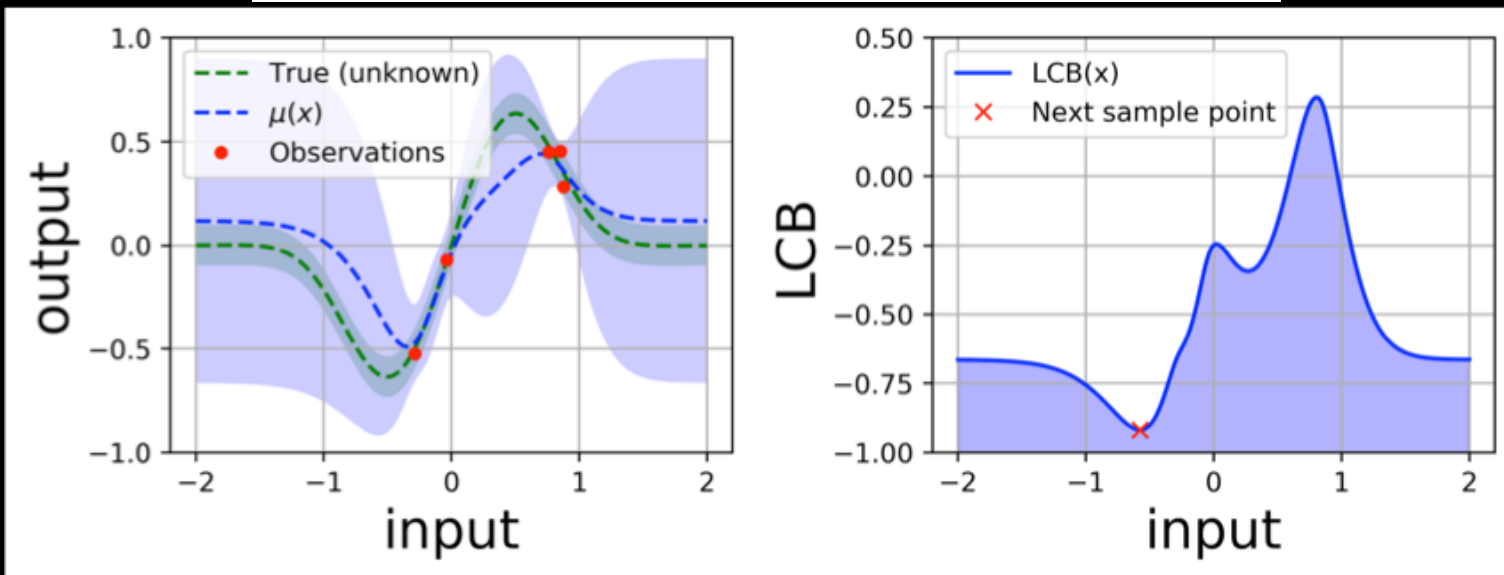
functions

- Usual Gaussian process regression cannot handle nonordinal space natively
- Appropriate methods: random forest, extra tree regressor, Bayesian NN
- We use **Random Forest**

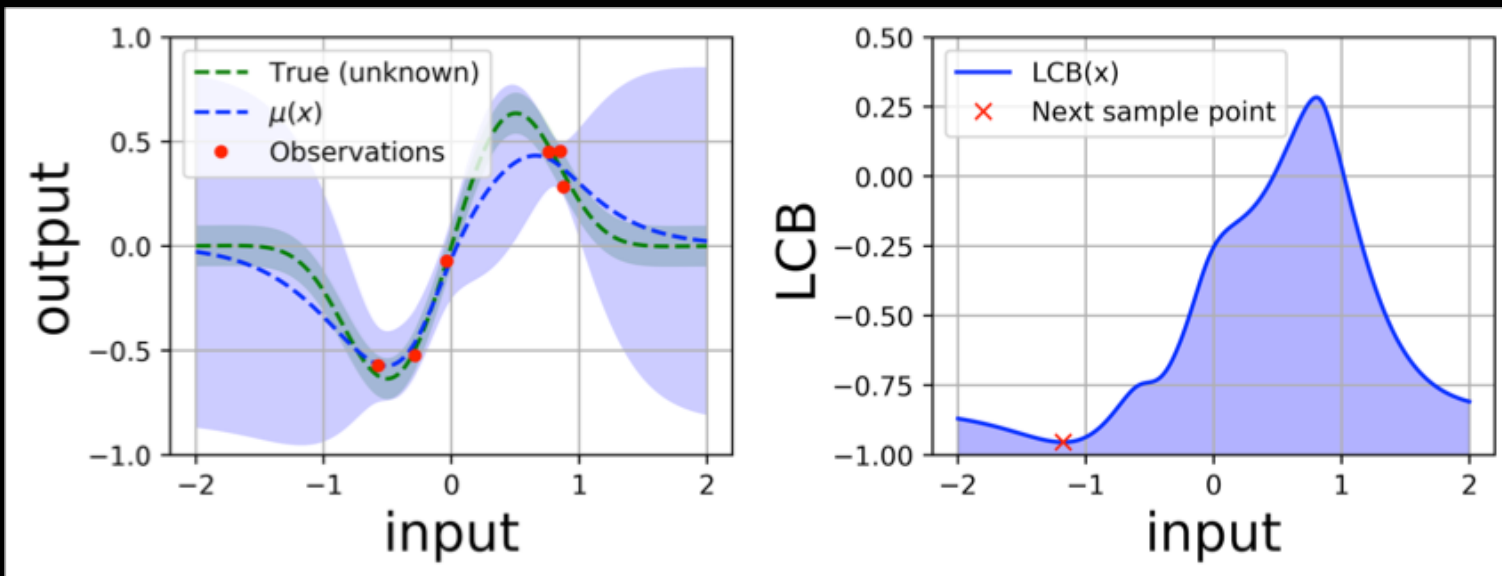
Bayesian optimization

$$LCB(x, \beta) = \mu(x) - \beta \times \sigma(x)$$

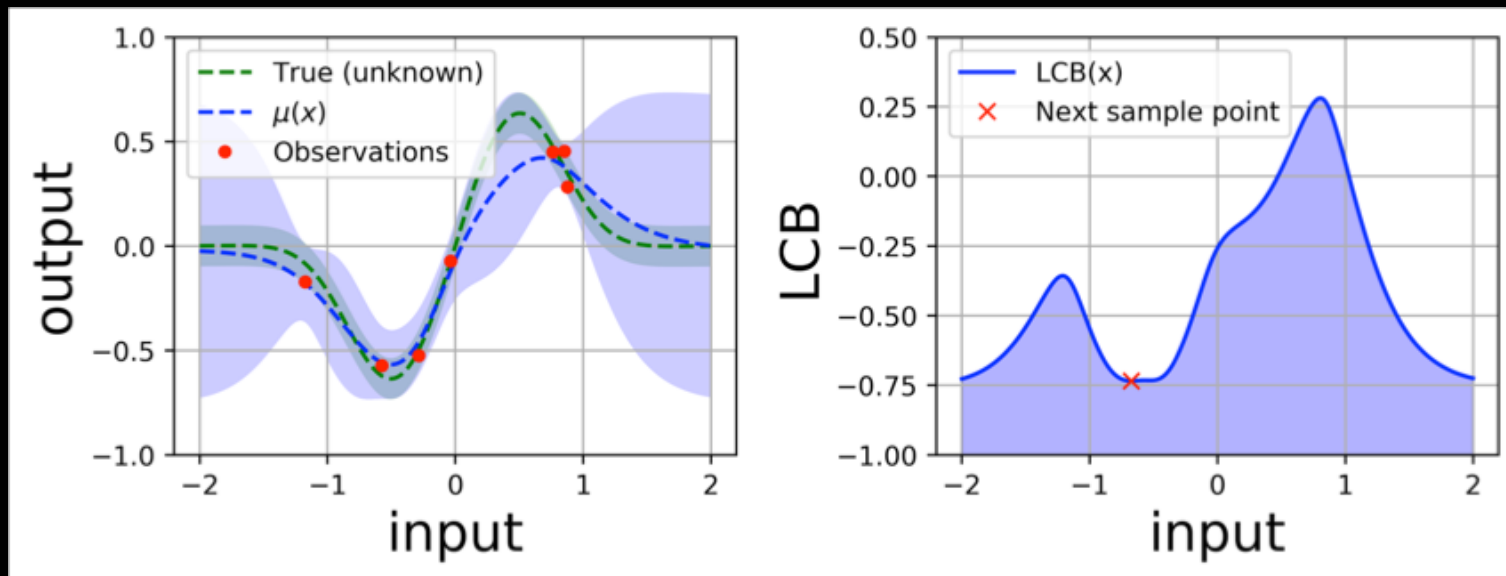
1.96



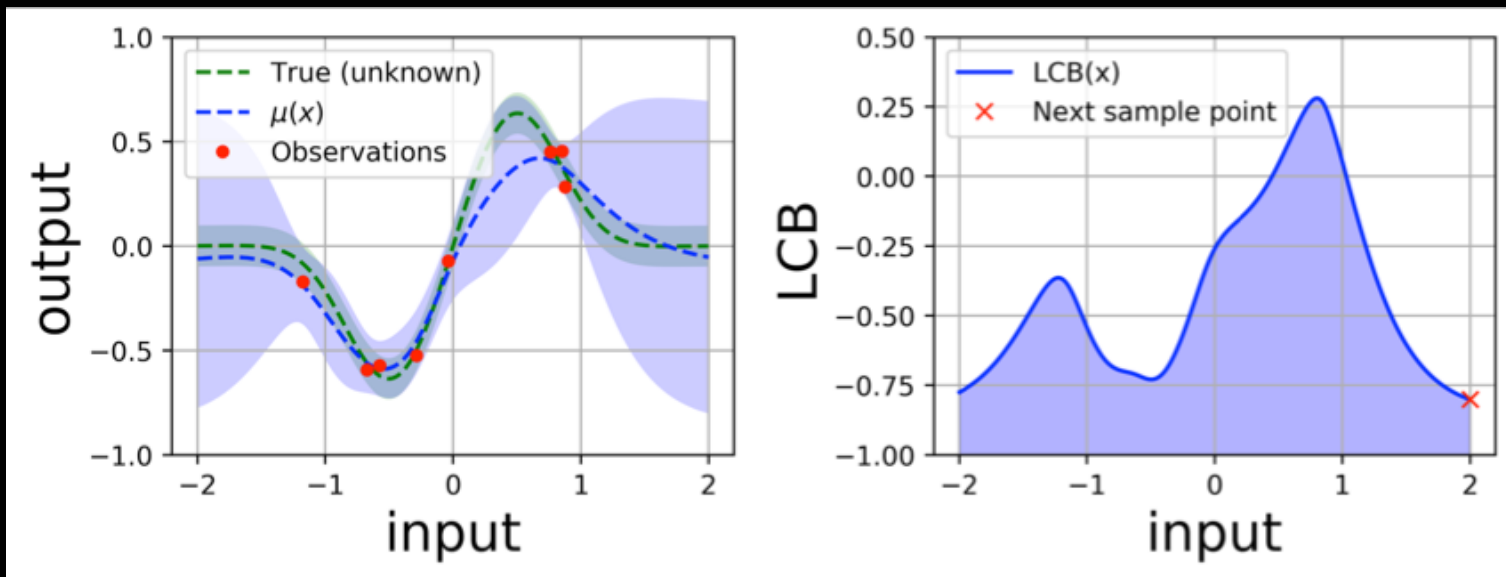
Bayesian optimization



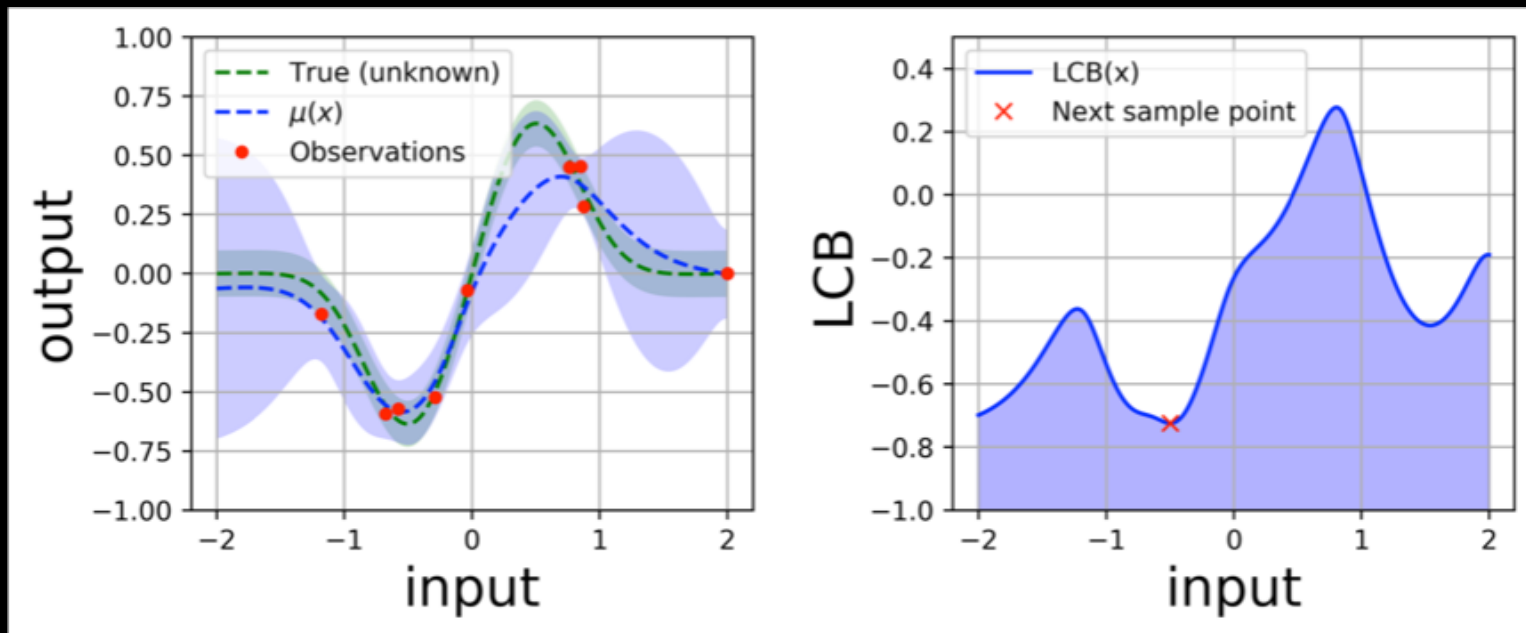
Bayesian optimization



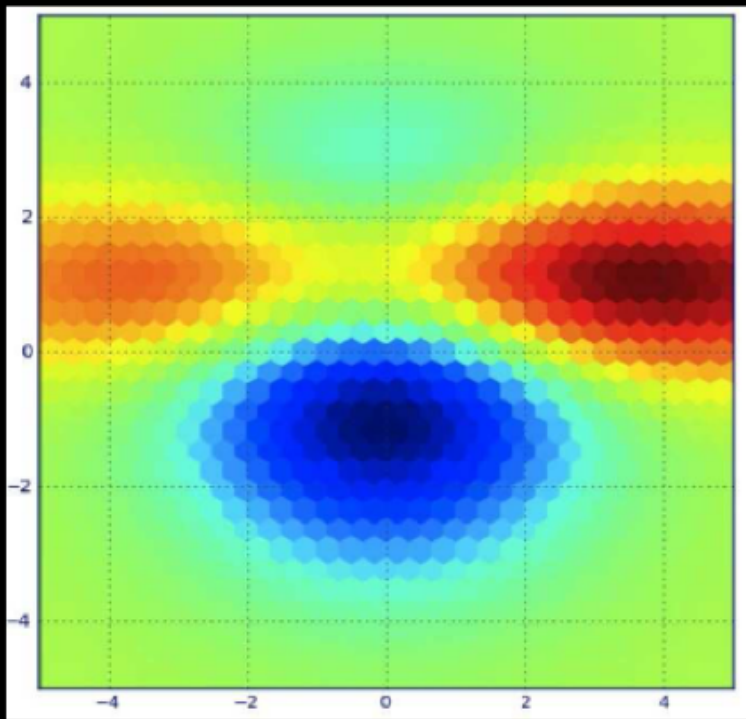
Bayesian optimization



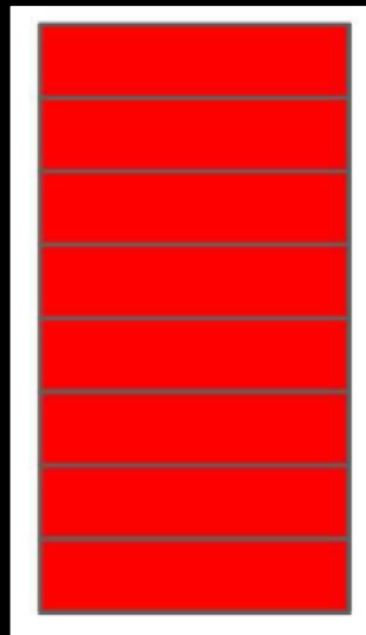
Bayesian optimization



Multipoint asynchronous sampling



Naive



Conditioned



Constant liar scheme for asynchronous update

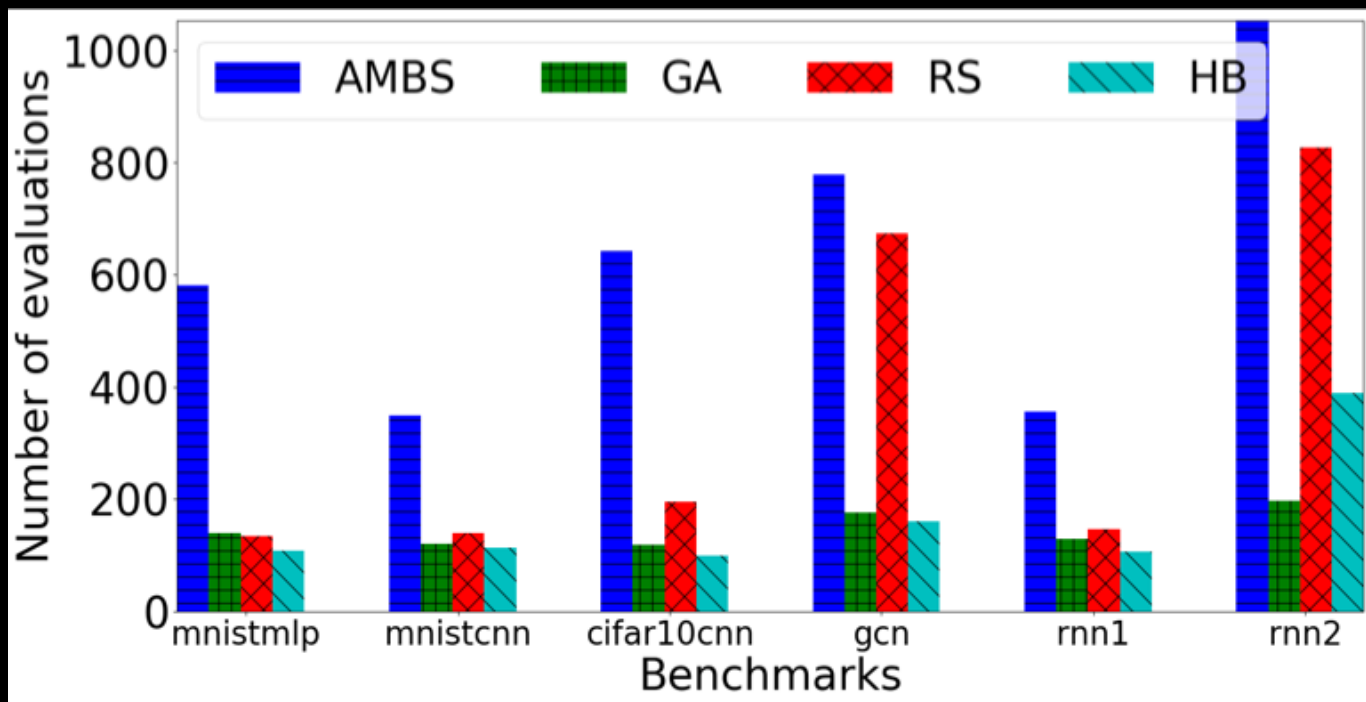
$$LCB_k(x, \beta) = \mu(x) - \beta \times \sigma(x)$$

$$f(\hat{x}_{LCB_k}) = \mu(\hat{x}_{LCB_k})$$

$$LCB_{k+1}(x, \beta) = \mu'(x) - \beta \times \sigma'(x)$$

$$f(\hat{x}_{LCB_{k+1}}) = \mu'(\hat{x}_{LCB_{k+1}})$$

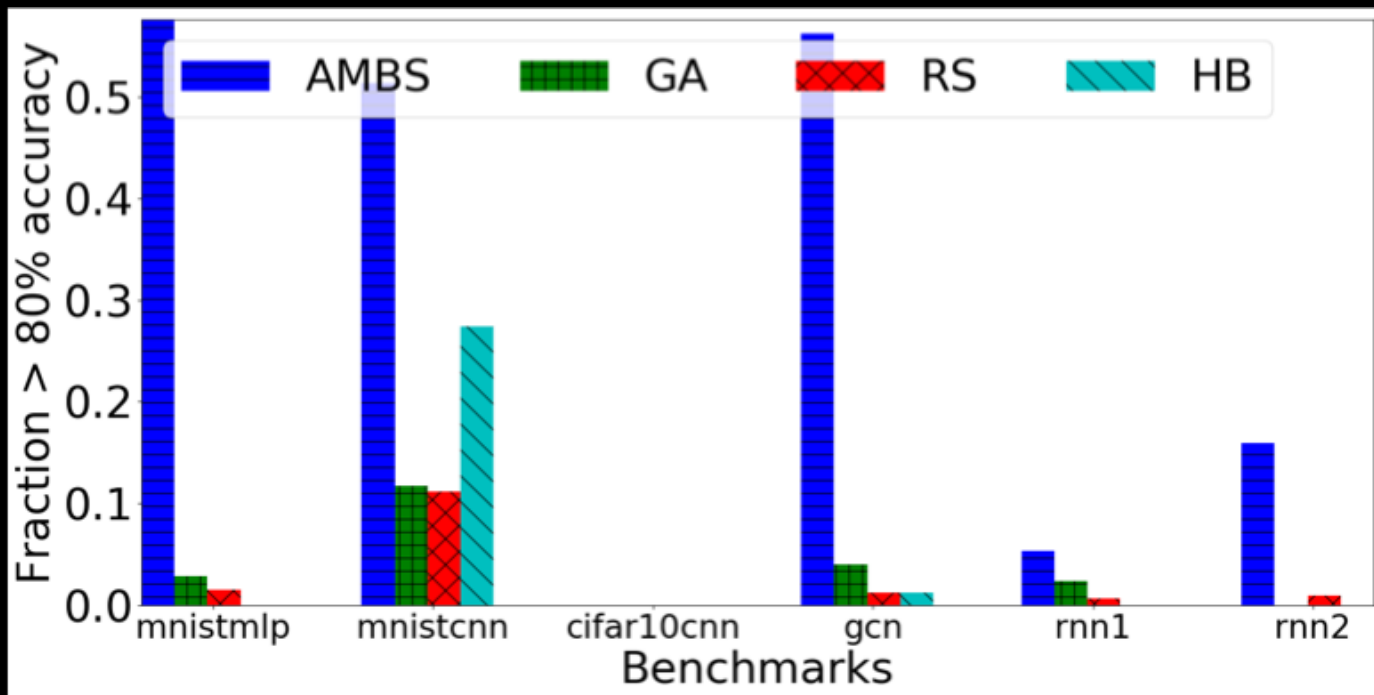
Comparison of search methods



Target platform: Theta@ALCF (128 KNL nodes)

Stopping criterion: 2 hours

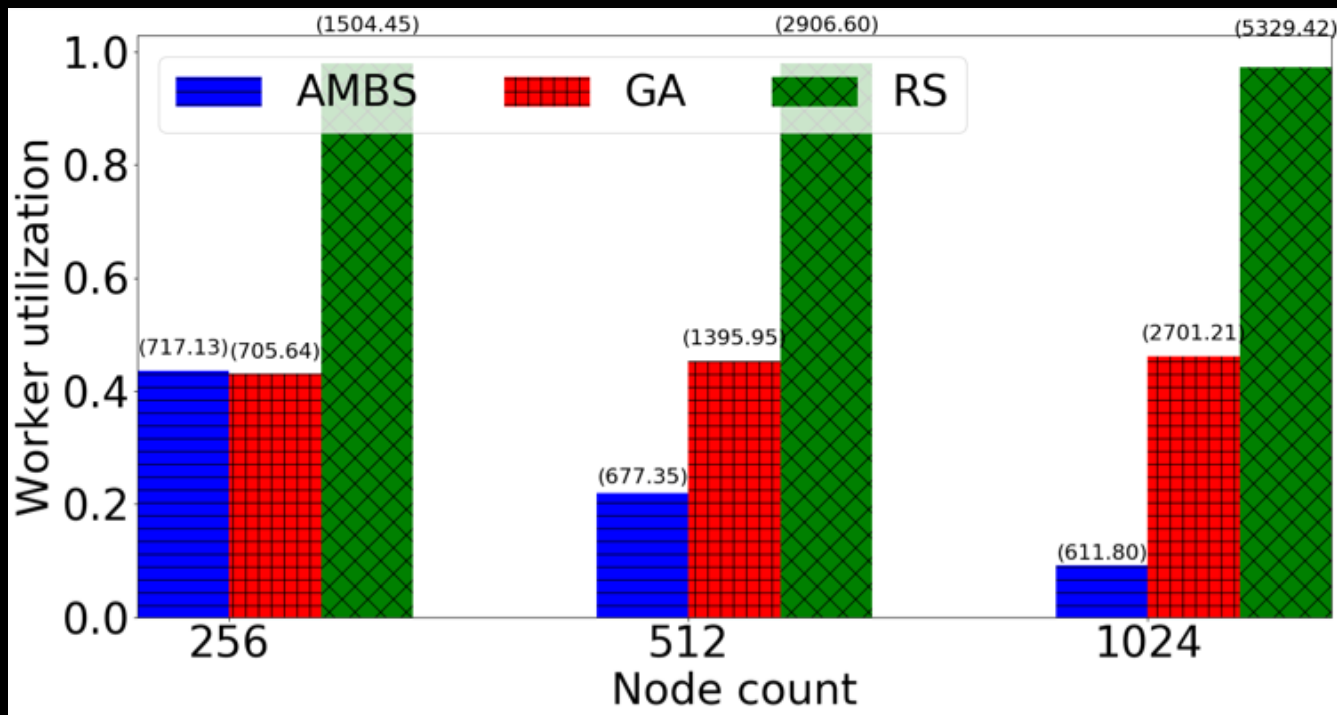
Comparison of search methods



Target platform: Theta@ALCF (128 KNL nodes)

Stopping criterion: 2 hours

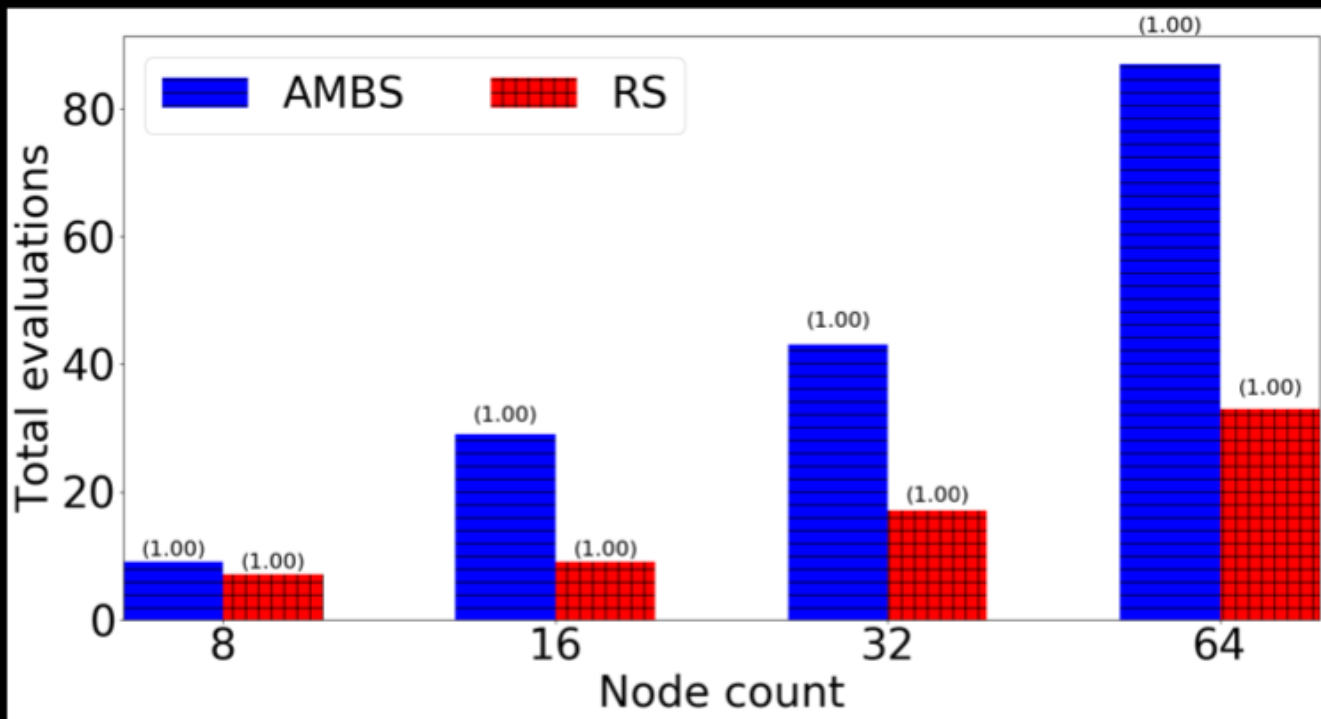
Scaling search methods



Target platform: Theta@ALCF (128 KNL nodes)

Benchmark: rnn2; Stopping criterion: 2 hours

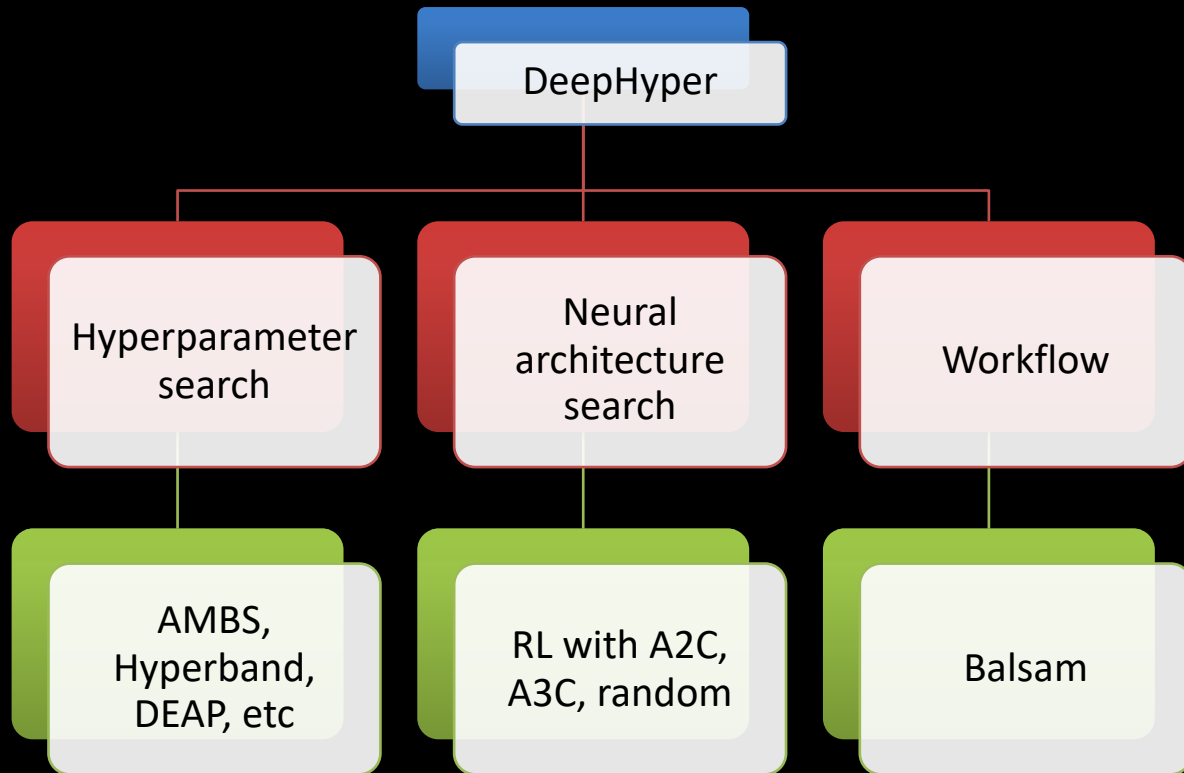
AMBS vs RS



Target platform: Cooley (64 nodes Haswell + NVIDIA Tesla K80)

Benchmark: cifar10cnn; Stopping criterion: 1 hour

DeepHyper: Scalable AutoML package



<https://github.com/deephyper/deephyper>

Acknowledgements



DOE Early Career Research Program, ASCR

Argonne Leadership Computing Facility



Laboratory Directed Research and Development (LDRD)

DeepHyper

