# Large Scale Bayesian Optimization with Subspace Decomposition

**Logan Mathesen**
*Joint Work with*: **Pedrielli G.**

ARIZONA STATE UNIVERSITY
SCHOOL OF **COMPUTING INFORMATICS & DECISION SYSTEMS ENGINEERING**

July 29th, 2019

# Global Optimization in High Dimension

- Increasingly complex problems are being researched in many domains
  - Large scale production of individualized cancer therapy;
  - Control of distributed robots;
  - Self driven cars.
- Increase in complexity (and parameterization) of novel algorithms developed to address these problems;
- Translation of hyper-parameter optimization problem to non-convex black-box optimization.

# Non-linear Non-convex Black-box Optimization

Assume there exists $f(\boldsymbol{x}) : \mathbb{X} \to \mathbb{R}$, where $\mathbb{X} \subset \mathbb{R}^d$.
We want to find and optimal solution $\boldsymbol{x}^*$:

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

We are specifically interested in high values of $d$.

# Bayesian Optimization

- State-of-the-art family within black-box optimization when function evaluations are costly and number of allowed function evaluations is low
- Surrogate model search, often Gaussian process
- Derivative free method of global optimization

**Algorithm 1** (Basic Pseudocode for Bayesian Optimization)

Place a Gaussian process prior on $f$
Observe $f$ at $n_0$ points according to an initial space-filling experimental design. Set $n = n_0$.
**while** $n \leq N$ **do**
  Update the posterior probability distribution on $f$ using all available data
  Let $x_n$ be a maximizer of the acquisition function over $x$, where the acquisition function is computed using the current posterior distribution.
  Observe $y_n = f(x_n)$.
  Increment $n$
**end while**
Return a solution: either the point evaluated with the largest $f(x)$ or the point with the largest posterior mean.

**Figure:** Peter I. Frazier. Bayesian Optimization. In INFORMS TutORials in Operations Research. Published online: 19 Oct 2018; 255-278.

# Challenges with Bayesian Optimiation

- Performance is not satisfactory after 10 dimensions;
- Three key and intertwined reasons for this:
  1. Gaussian process learning effort grows cubically with number of observations, $O(n^3)$
  2. To ensure reasonable closeness to $\boldsymbol{x}^*$ substantial *coverage* of $\mathbb{X}$ is required, number of observations needed for coverage exponentially increases with dimension
  3. Maximizing the acquisition function generally scales exponentially with the number of dimensions

# Embedding

- Assumption of "low effective dimensionality" or "approximate low effective dimensionality"
  - Dimensions do not impact objective function
  - Definition: $\epsilon$-effective subspace ($\mathcal{V}_\epsilon$)
    A function $f : \mathbb{R}^d \to \mathbb{R}$, has a valid $\mathcal{V}_\epsilon$ if $\exists$ linear subspace $\mathcal{V}_\epsilon \subseteq \mathbb{R}^d$ s.t., for all $\boldsymbol{x} \in \mathbb{R}^d$, $|f(\boldsymbol{x}) - f(\boldsymbol{x}_\epsilon)| \leq \epsilon$. Where $\boldsymbol{x}_\epsilon \in \mathcal{V}_\epsilon$ is the orthogonal projection of $\boldsymbol{x}$ onto $\mathcal{V}_\epsilon$
- REMBO (Random EMbedding Bayesian Optimization)
- SRE (Sequential Random Embeddings)
- SI-BO (Subspace Identification – Bayesian Optimization) Random Sampling

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., & de Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. Journal of Artificial Intelligence Research, 55, 361-387.
Qian, H., Hu, Y. Q., & Yu, Y. (2016, July). Derivative-Free Optimization of High-Dimensional Non-Convex Functions by Sequential Random Embeddings. In IJCAI (pp. 1946-1952).
Djolonga, J., Krause, A., & Cevher, V. (2013). High-dimensional gaussian process bandits. In Advances in Neural Information Processing Systems (pp. 1025-1033).

# Additive Modeling

- **structural assumption**: $f(\cdot)$ decomposes as:

$$f(\boldsymbol{x}) = f^1\left(\boldsymbol{x}^1\right) + f^2\left(\boldsymbol{x}^2\right) + \ldots + f^m\left(\boldsymbol{x}^m\right)$$

  where each $\boldsymbol{x}^j \in \mathbb{X}^j = \prod_i^{d_j} [0,1]_j$ represents a "group" constituting the decomposition.
- Each $f^j\left(\boldsymbol{x}^j\right)$ is then independently modeled as a Gaussian process, and when added together recover the full dimensional model
- Additive kernel based acquisition functions assessed over $\mathbb{X}$ can be optimized by maximizing over each component $\mathbb{X}^j$ to reproduce sampling in $\mathbb{X}$.

Kandasamy, K., Schneider, J., & Pŏczos, B. (2015, June). High dimensional Bayesian optimisation and bandits via additive models. In International Conference on Machine Learning (pp. 295-304).

- Work in selection of appropriate $\boldsymbol{x}^j$ "groups" include
  - MCMC methods;
  - Factor graph methods;
  - Gibbs sampling based structural kernel learning;
  - Gibbs sampling based graph learning;
  - Fourier Feature approximation.

Gardner, J., C. Guo, K. Weinberger, R. Garnett, and R. Grosse. 2017. "Discovering and exploiting additivestructure for Bayesian optimization". InArtificial Intelligence and Statistics, 1311–1319.
Trong Nghia Hoang and Quang Minh Hoang and Ruofei Ouyang and Kian Hsiang Low 2018. "Decentral-ized High-Dimensional Bayesian Optimization With Factor Graphs".
Wang, Z., C. Li, S. Jegelka, and P. Kohli. 2017. "Batched High-dimensional Bayesian Optimization viaStructural Kernel Learning". InInternational Conference on Machine Learning (ICML).
Rolland, P., J. Scarlett, I. Bogunovic, and V. Cevher. 2018. "High-Dimensional Bayesian Optimization viaAdditive Models with Overlapping Groups". InInternational Conference on Artificial Intelligence andStatistics, 298–307.
Mutny, M., and A. Krause. 2018. "Efficient High Dimensional Bayesian Optimization with Additivity andQuadrature Fourier Features". InAdvances in Neural Information Processing Systems, 9005–9016.

# Our Problem

We aim at finding the *global* minimum $\boldsymbol{x}^*$ such that:

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

- $f$ is non-linear and non-convex
- $\mathbb{X}$ is continuous
- No low dimensional structures can be easily learned about:
  - the changes of $f(\cdot)$ in $\mathbb{X}$ (embedding);
  - $f(\cdot)$ modeled as a realization of an additive structure.
- $f$ is smooth (implied assumption from algorithm implementation)

We propose the Subspace COmmunication based OPtimization (SCOOP) algorithm for efficient optimization.

# SCOOP Algorithm Overview

- Consider the original space to be separated into $k$ subspaces $\mathbb{X}_1, \mathbb{X}_2, \ldots, \mathbb{X}_k$ each with dimensionality $d_j = d - n_j, j = 1, \ldots, k$;
- The subspace generation has to consider that all the dimensions of the original space $\mathbb{X}$ need to be considered in at least one of the decompositions
- In each subspace we use Bayesian optimization to yield subspace optimizers, however, any optimization technique could be used.
- The key is the **information sharing** mechanism. In fact, every "batch" of iterations, the subspaces need to somewhat communicate the current state of the local optimization in order to change the value of the "currently fixed" variables.

# SCOOP Algorithm Overview

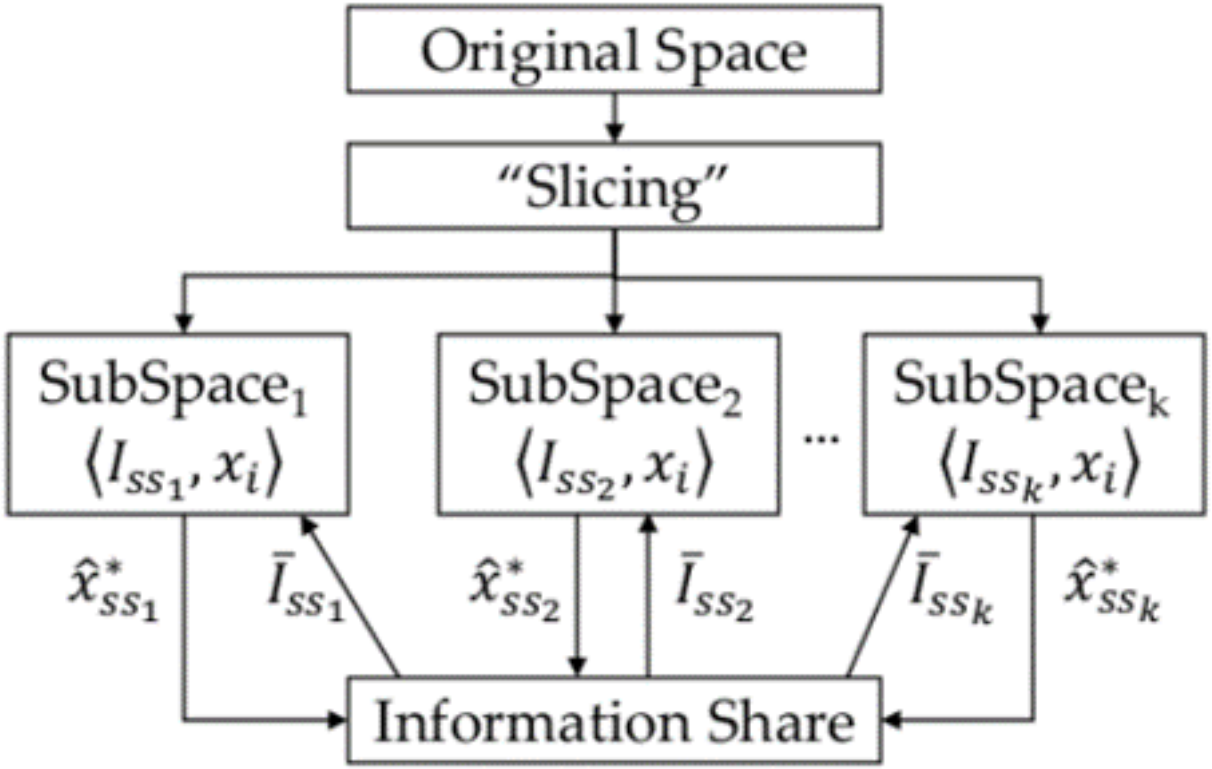
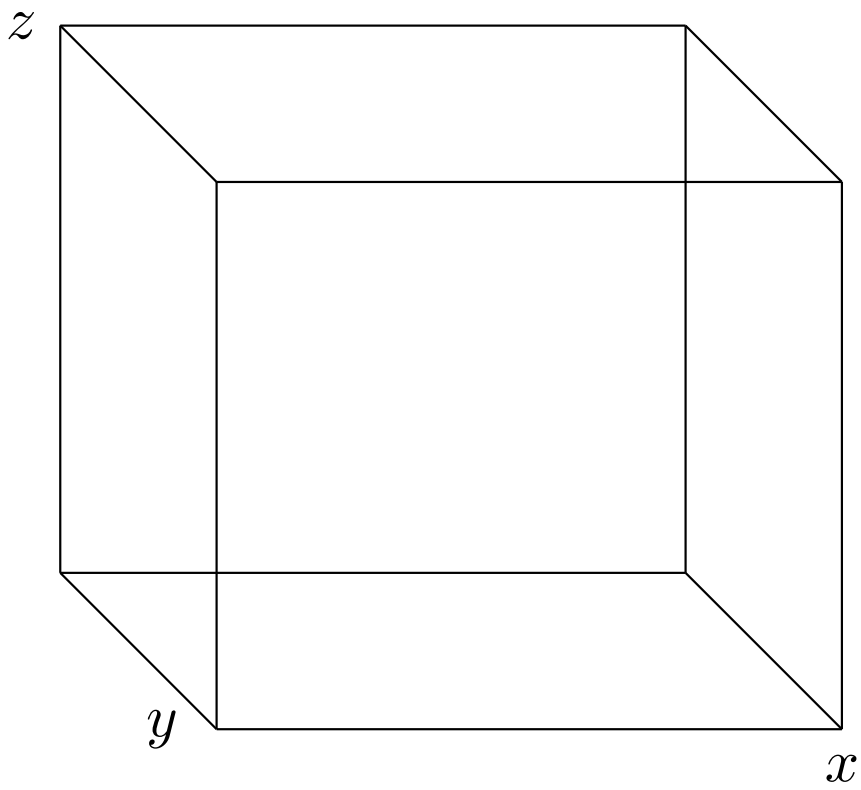
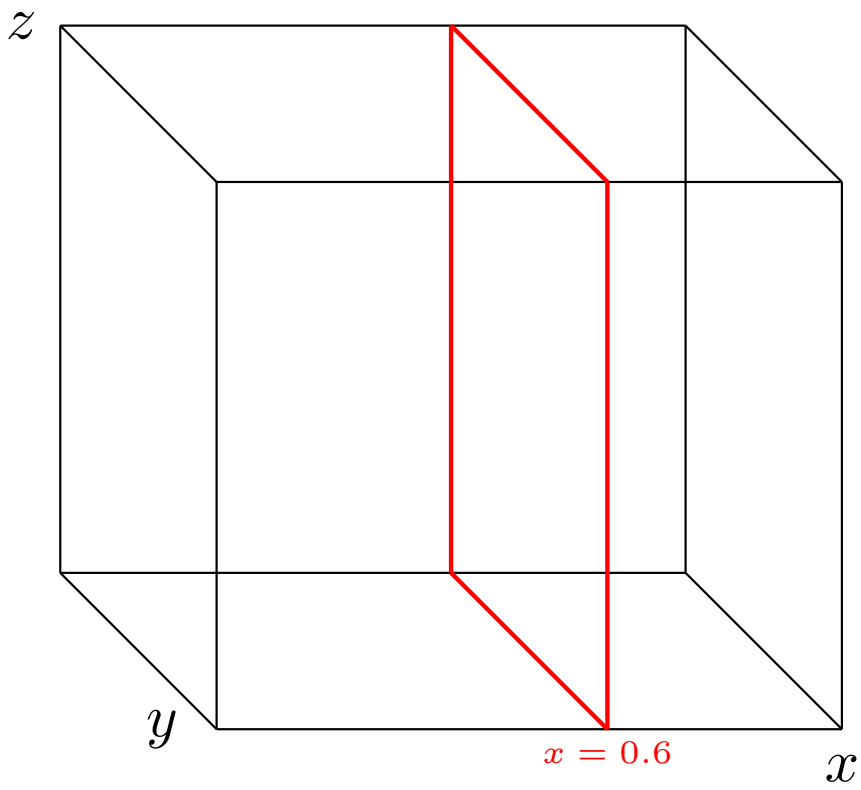



**Figure:** Basic Idea of SCOOP.

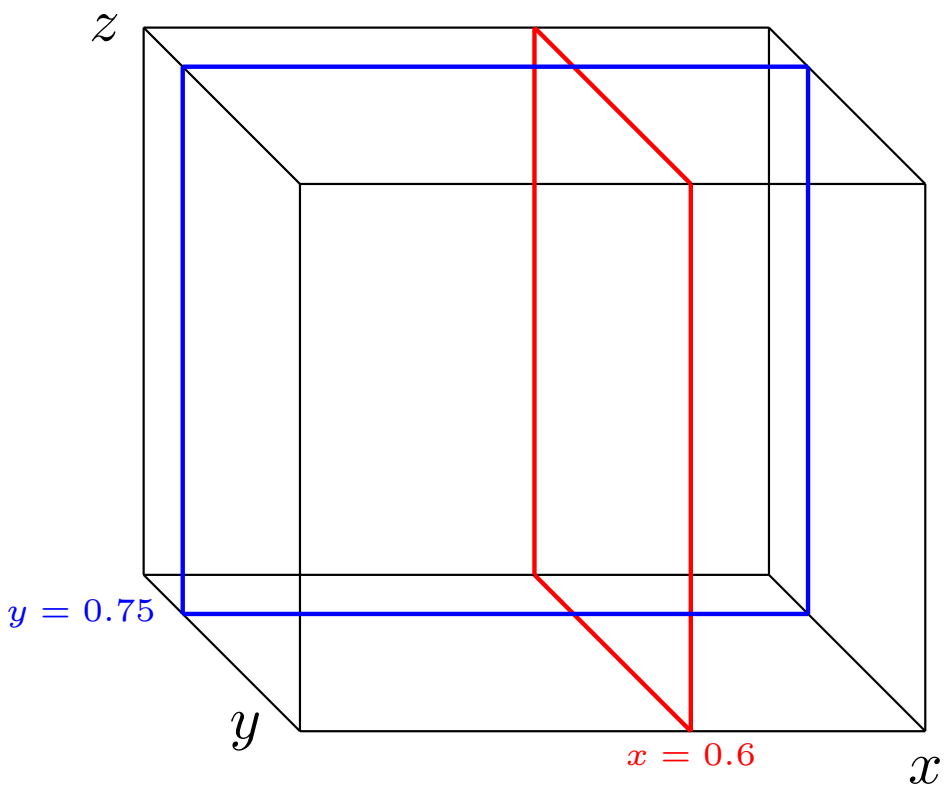
# SCOOP Visualization in $\mathbb{R}^3$



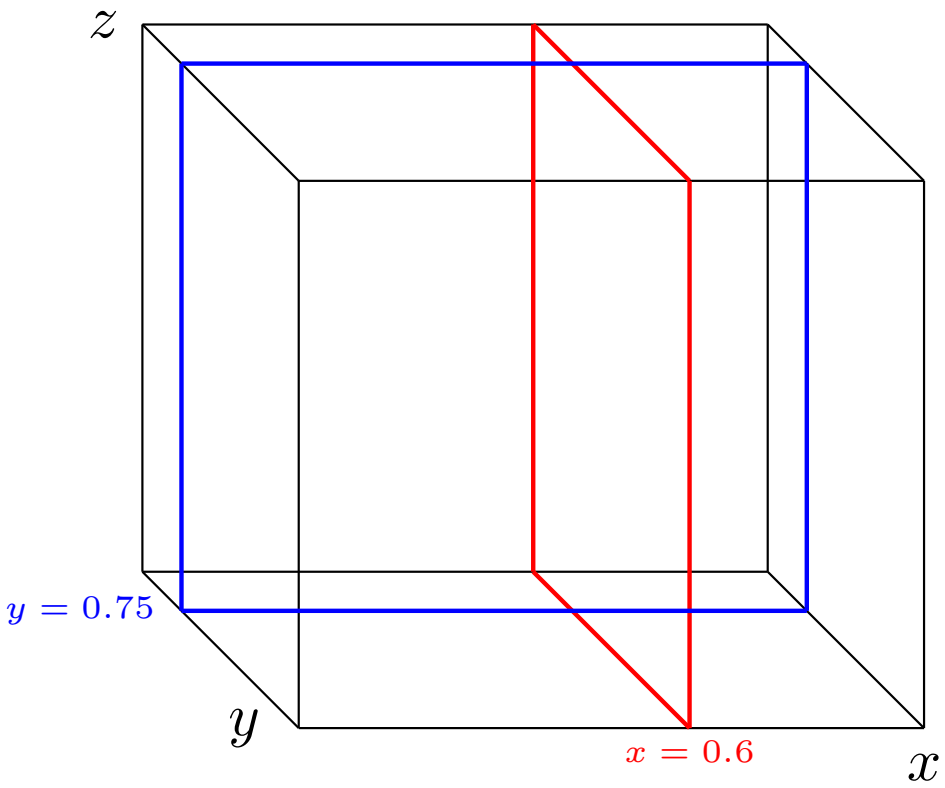
# SCOOP Visualization in $\mathbb{R}^3$

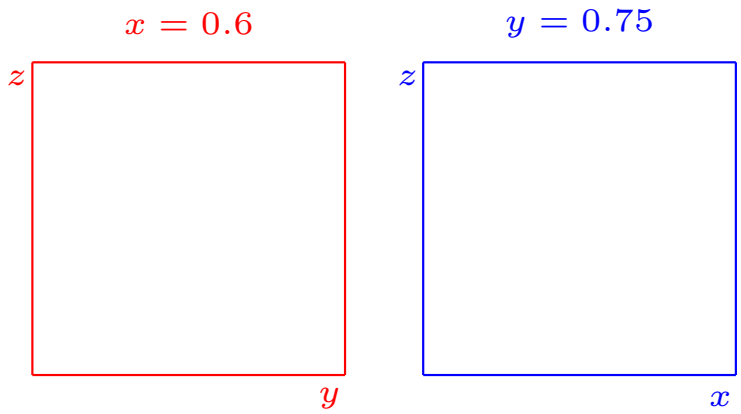


# SCOOP Visualization in $\mathbb{R}^3$

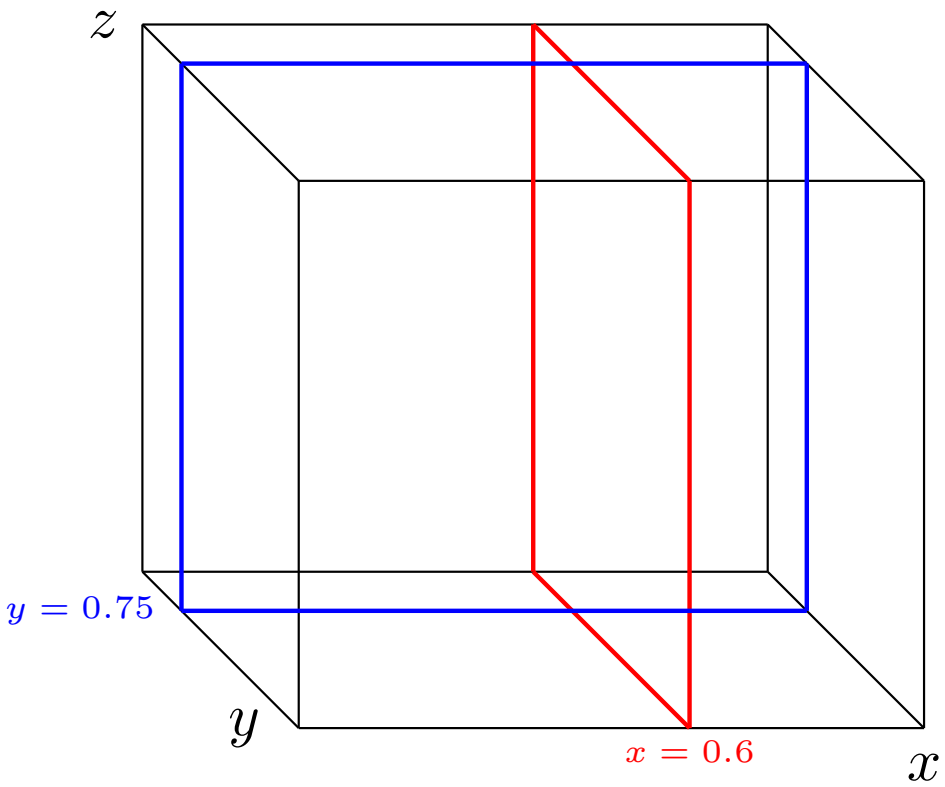


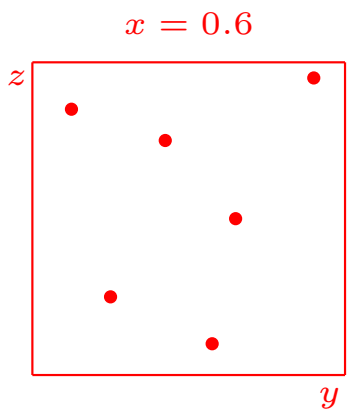
# SCOOP Visualization in $\mathbb{R}^3$

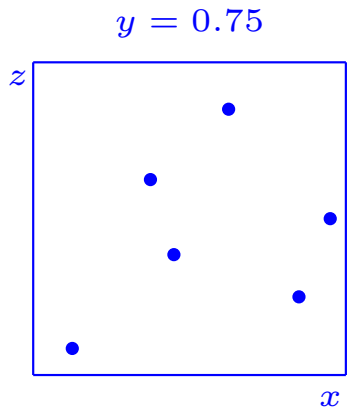


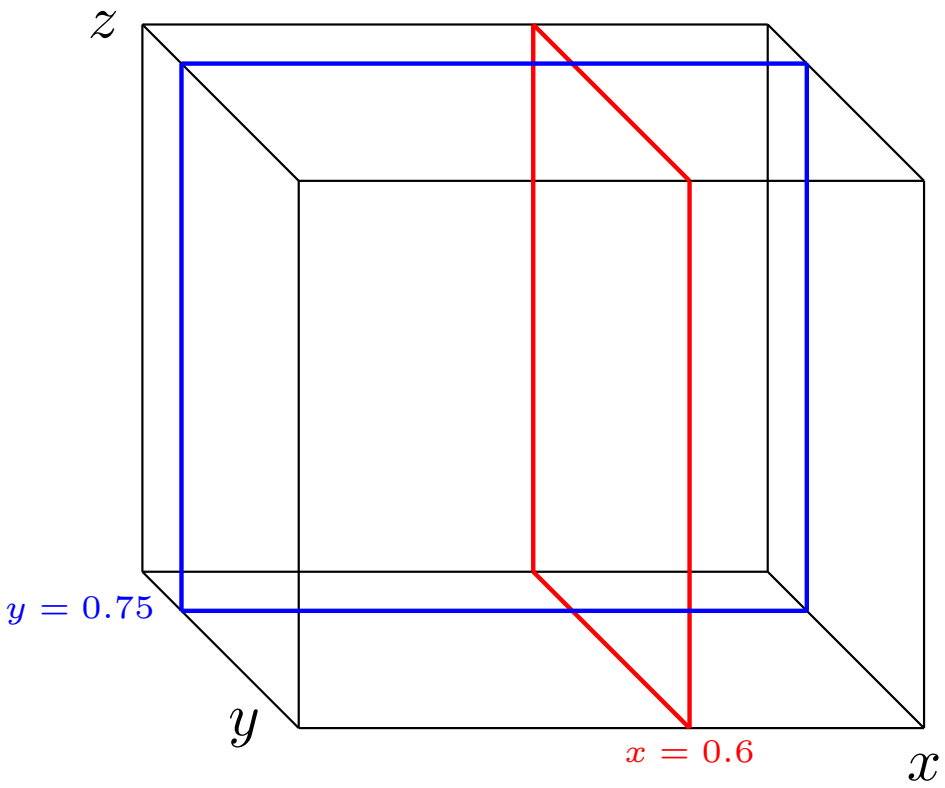
# SCOOP Visualization in $\mathbb{R}^3$

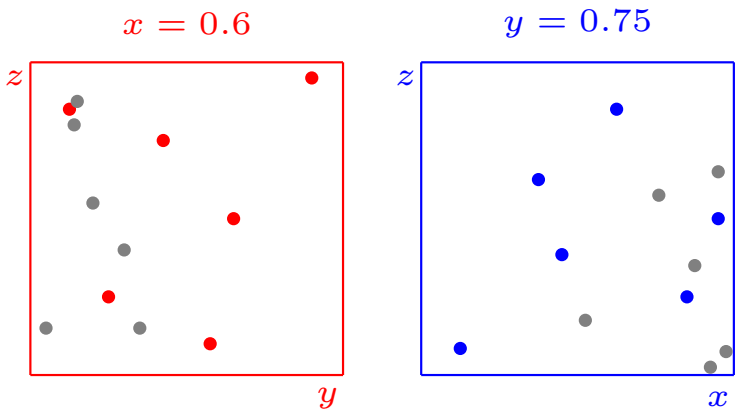


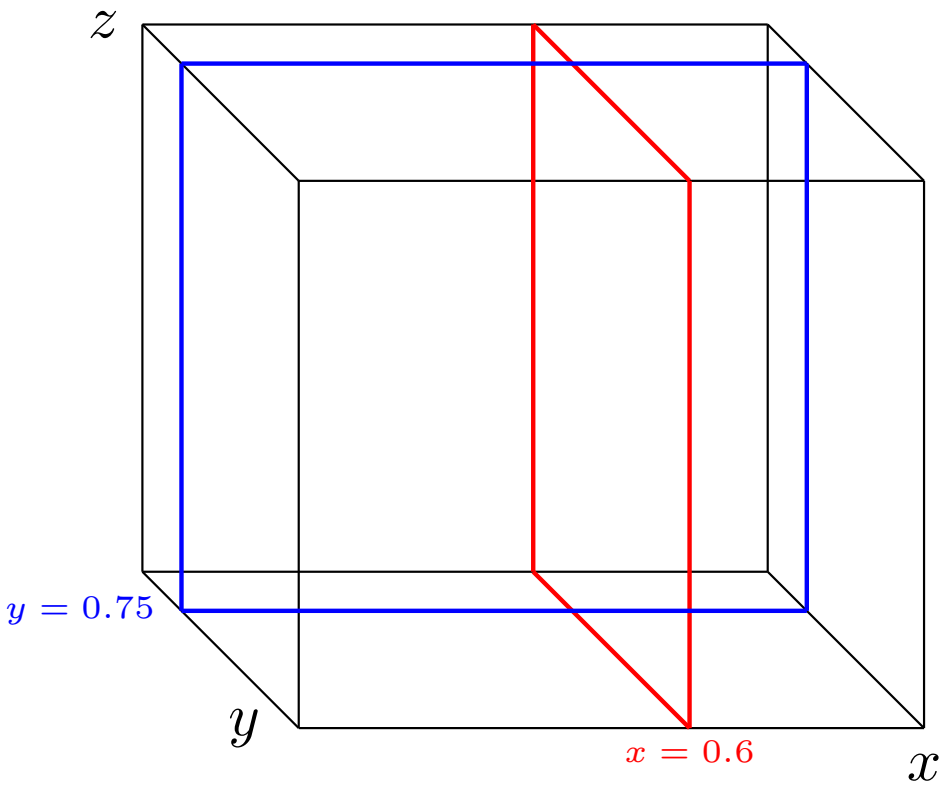



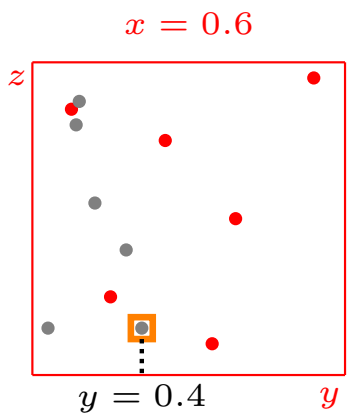
# SCOOP Visualization in $\mathbb{R}^3$

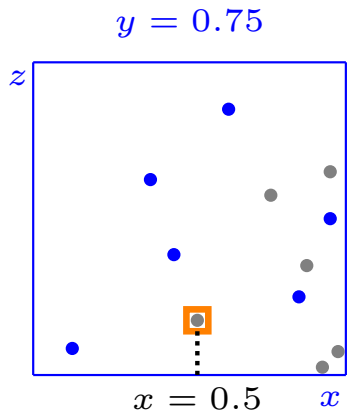


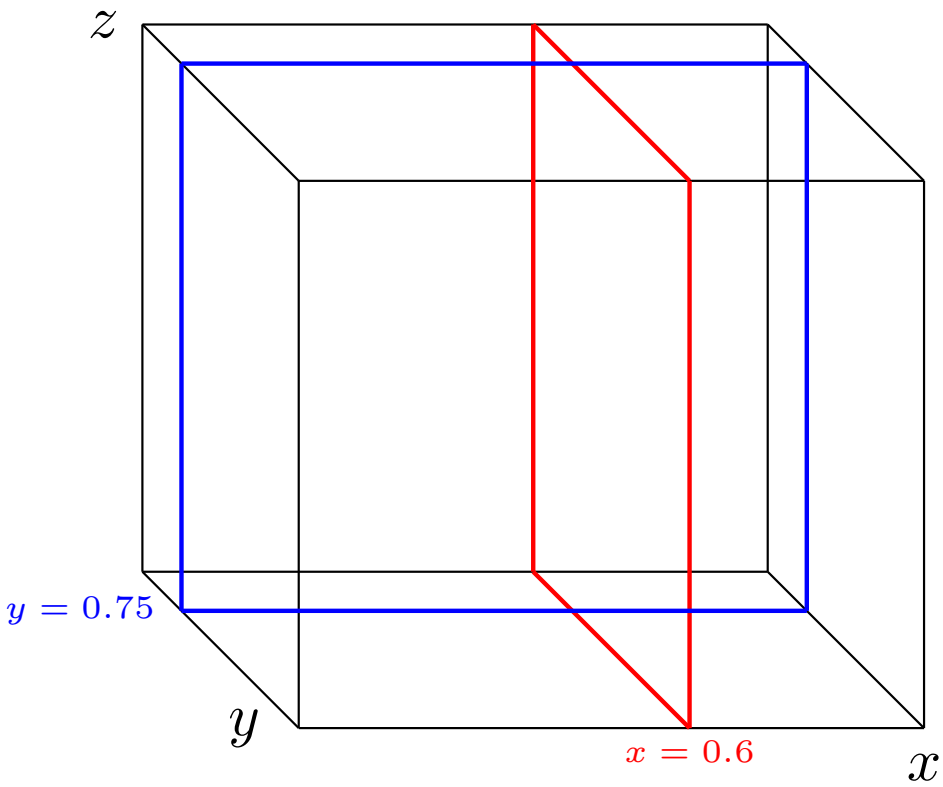
# SCOOP Visualization in $\mathbb{R}^3$

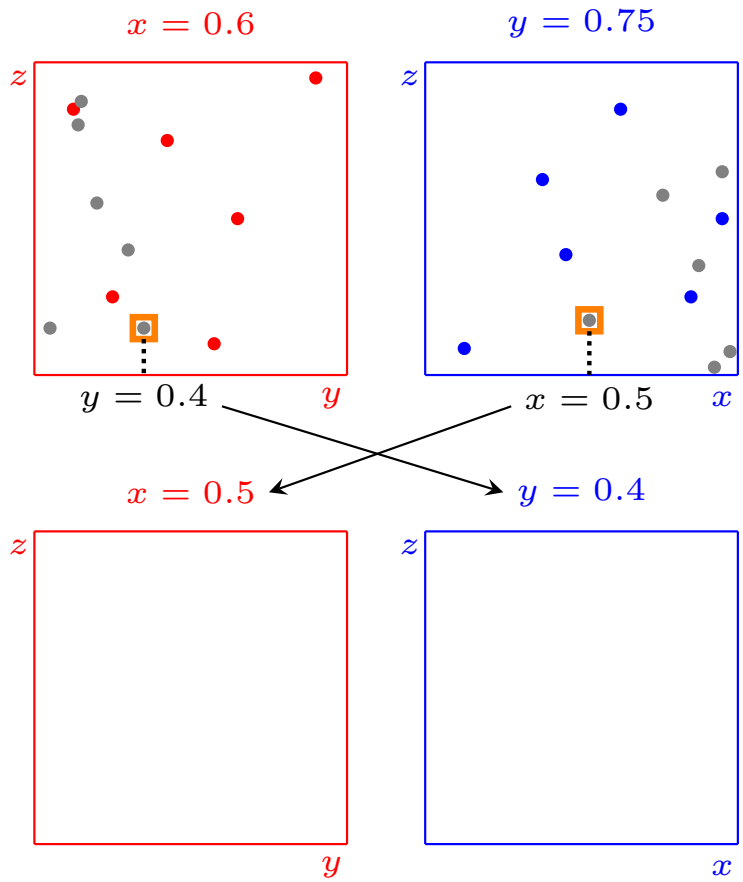


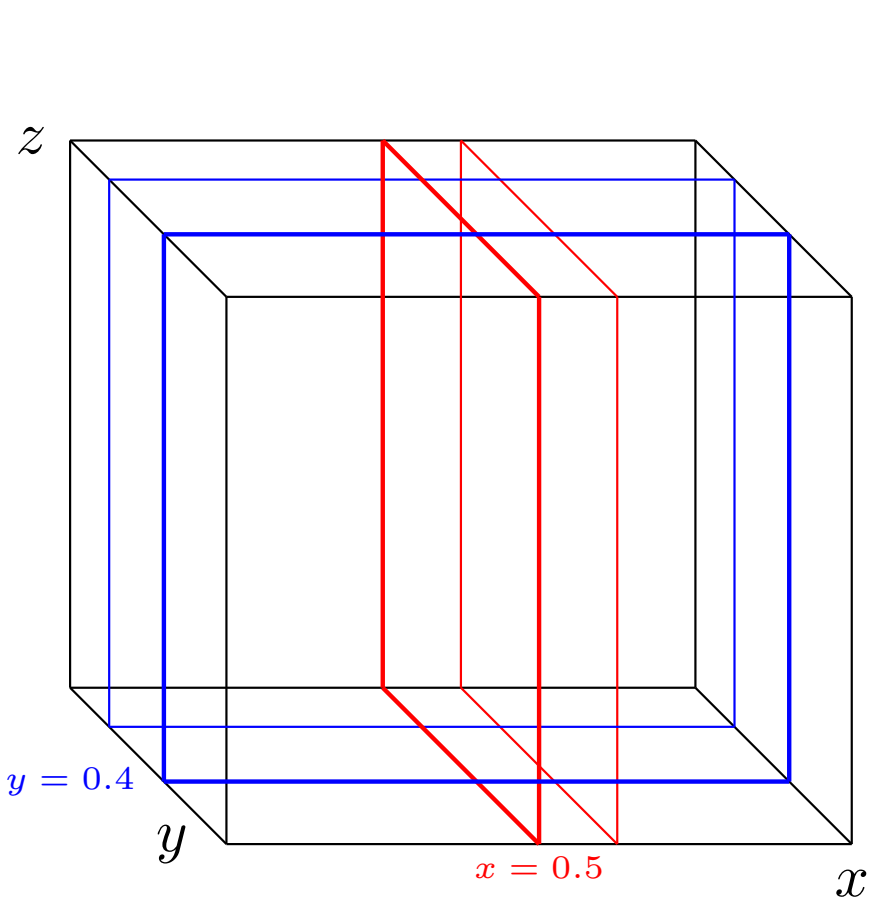



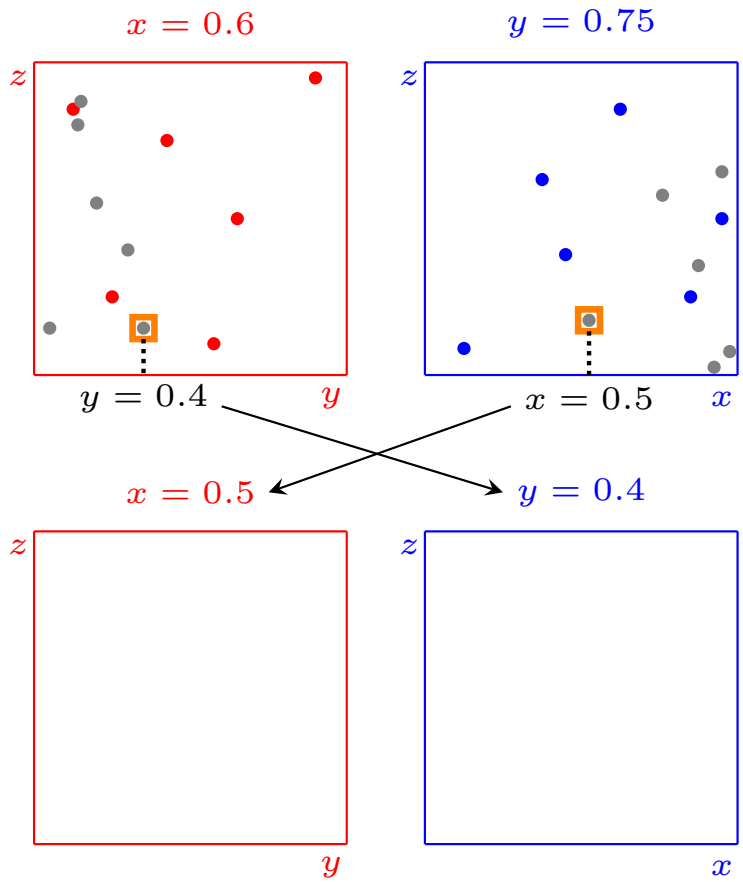
# SCOOP Visualization in $\mathbb{R}^3$



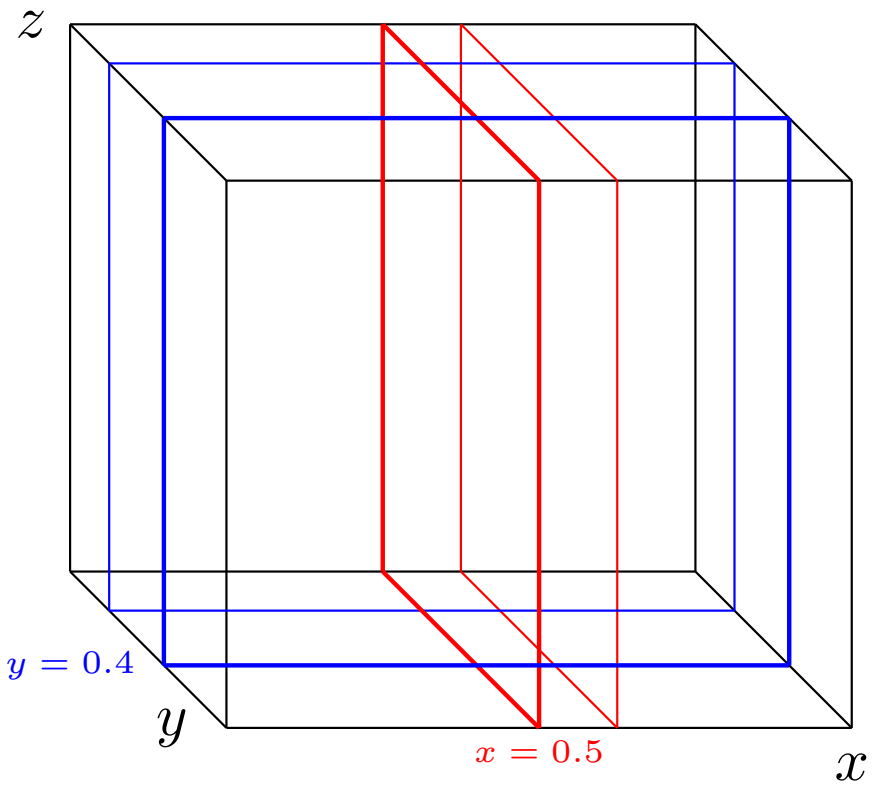



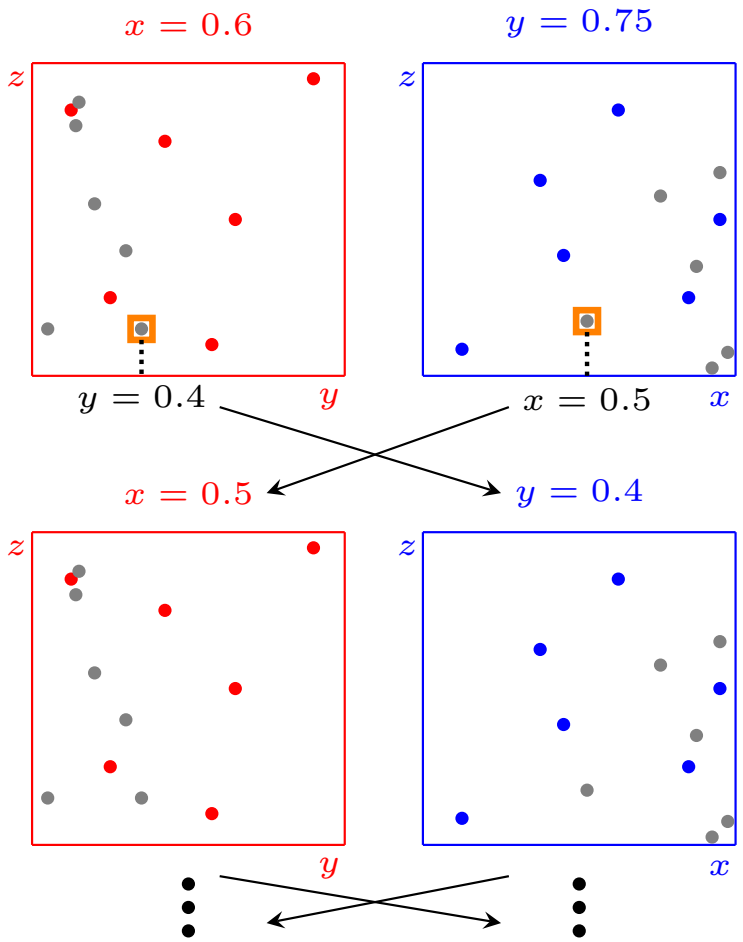
# SCOOP Visualization in $\mathbb{R}^3$



# SCOOP Visualization in $\mathbb{R}^3$



# SCOOP Overview

**Algorithm 1** Subspace COmmunication based OPtimization

**Input:** $f(\mathbf{x})$, $a(\mathbf{x})$, $B$, $n_{sub}$, $n_{hid}$, $b_{init}$, $b_{opt}$, information sharing strategy
**Output:** $\hat{\mathbf{x}}^*$, $f(\hat{\mathbf{x}}^*)$: the best sampled full dimensional location

**Initialize:** Randomly assign all hidden dimension values for each of subspaces, there will be a total of $(n_{hid} \times n_{sub})$ hidden dimensions assigned.
**while** $B > 0$ **do**
*Execute Subspace Optimizations*
  **for** $i = 1, \ldots, n_{sub}$ **do**
    Create initial space filling design with $b_{init}$ points over active/free dimensions of subspace $i$: $\mathbf{X}_{0i} \in \mathbb{R}^{(b_{init} \times d - n_{hid})}$
    Create $\mathbf{X}_{0i}^{full}$ by augmenting each design point in $\mathbf{X}_{0i}$ with current subspace hidden dimension values: $\mathbf{X}_{0i}^{full} \in \mathbb{R}^{(b_{init} \times d)}$
    Sample objective function at each of these points: $f(\mathbf{X}_{0i}^{full}) \in \mathbb{R}^{(b_{init} \times 1)}$. Update $B \leftarrow B - b_{init}$, break if $B \leq 0$
    Fit a subspace GP to $\mathbf{X}_{0i}$, $f(\mathbf{X}_{0i}^{full})$
    **for** $j = 1, \ldots, b_{opt}$ **do**
      Discover $\mathbf{x}_{next} = \arg\max_{\mathbf{x} \in \mathcal{X}} a(\mathbf{x})$: $\mathbf{x}_{next} \in \mathbb{R}^{(1 \times d - n_{hid})}$
      Augment $\mathbf{x}_{next}$ with hidden dimension values: $\mathbf{x}_{next}^{full} \in \mathbb{R}^{(1 \times d)}$
      Update $\mathbf{X}_{ji} \leftarrow \mathbf{X}_{(j-1)i} \cup \mathbf{x}_{next}$ and $\mathbf{X}_{ji}^{full} \leftarrow \mathbf{X}_{(j-1)i}^{full} \cup \mathbf{x}_{next}^{full}$
      Sample $f(\mathbf{x}_{next}^{full})$. Update $B \leftarrow B - 1$, break if $B = 0$
      Fit updated subspace GP to $\mathbf{X}_{ji}$, $f(\mathbf{X}_{ji}^{full})$
    **end for**
  **end for**
*Execute Subspace Information Sharing: Update Hidden Dimension Values*
  **for** $i = 1, \ldots, n_{sub}$ **do**
    **for** $j = 1, \ldots, n_{hid}$ **do**
      For hidden dimension $j$ of subspace $i$, determine the complementary subspace with a shared active parameter and free dimension corresponding to subspace $i$'s $j^{th}$ hidden dimension
      Update value of subspace $i$'s hidden dimension $j$ by sharing information with the identified complementary subspace
    **end for**
  **end for**
**end while**

# SCOOP Overview

**Algorithm 1** Subspace COmmunication based OPtimization

**Input:** $f(\mathbf{x})$, $a(\mathbf{x})$, $B$, $n_{sub}$, $n_{hid}$, $b_{init}$, $b_{opt}$, information sharing strategy
**Output:** $\hat{\mathbf{x}}^*$, $f(\hat{\mathbf{x}}^*)$: the best sampled full dimensional location

**Initialize: Select Constant/Hidden Dimensions of each Subspace**

**while** $B > 0$ **do**
*Execute Subspace Optimizations*
  **for** $i = 1, \ldots, n_{sub}$ **do**
    Create initial space filling design with $b_{init}$ points over active/free dimensions of subspace $i$: $\mathbf{X}_{0i} \in \mathbb{R}^{(b_{init} \times d - n_{hid})}$
    Create $\mathbf{X}_{0i}^{full}$ by augmenting each design point in $\mathbf{X}_{0i}$ with current subspace hidden dimension values: $\mathbf{X}_{0i}^{full} \in \mathbb{R}^{(b_{init} \times d)}$
    Sample objective function at each of these points: $f(\mathbf{X}_{0i}^{full}) \in \mathbb{R}^{(b_{init} \times 1)}$. Update $B \leftarrow B - b_{init}$, break if $B \leq 0$
    Fit a subspace GP to $\mathbf{X}_{0i}$, $f(\mathbf{X}_{0i}^{full})$
    **for** $j = 1, \ldots, b_{opt}$ **do**
      Discover $\mathbf{x}_{next} = \arg\max_{\mathbf{x} \in \mathcal{X}} a(\mathbf{x})$: $\mathbf{x}_{next} \in \mathbb{R}^{(1 \times d - n_{hid})}$
      Augment $\mathbf{x}_{next}$ with hidden dimension values: $\mathbf{x}_{next}^{full} \in \mathbb{R}^{(1 \times d)}$
      Update $\mathbf{X}_{ji} \leftarrow \mathbf{X}_{(j-1)i} \cup \mathbf{x}_{next}$ and $\mathbf{X}_{ji}^{full} \leftarrow \mathbf{X}_{(j-1)i}^{full} \cup \mathbf{x}_{next}^{full}$
      Sample $f(\mathbf{x}_{next}^{full})$. Update $B \leftarrow B - 1$, break if $B = 0$
      Fit updated subspace GP to $\mathbf{X}_{ji}$, $f(\mathbf{X}_{ji}^{full})$
    **end for**
  **end for**
*Execute Subspace Information Sharing: Update Hidden Dimension Values*
  **for** $i = 1, \ldots, n_{sub}$ **do**
    **for** $j = 1, \ldots, n_{hid}$ **do**
      For hidden dimension $j$ of subspace $i$, determine the complementary subspace with a shared active parameter and free dimension corresponding to subspace $i$'s $j^{th}$ hidden dimension
      Update value of subspace $i$'s hidden dimension $j$ by sharing information with the identified complementary subspace
    **end for**
  **end for**
**end while**

# SCOOP Overview

**Algorithm 1** Subspace COmmunication based OPtimization

**Input:** $f(\mathbf{x})$, $a(\mathbf{x})$, $B$, $n_{sub}$, $n_{hid}$, $b_{init}$, $b_{opt}$, information sharing strategy
**Output:** $\hat{\mathbf{x}}^*$, $f(\hat{\mathbf{x}}^*)$: the best sampled full dimensional location

Initialize: Select Constant/Hidden Dimensions of each Subspace

3: **while** $B > 0$ **do**
*Execute Subspace Optimizations*
4:
5:
6:
7:
8:
9:
10: Bayesian Optimization: Optimize over each Current Subspace
11:
12:
13:
14:
15:
16:
*Execute Subspace Information Sharing: Update Hidden Dimension Values*
17: **for** $i = 1, \ldots, n_{sub}$ **do**
18: **for** $j = 1, \ldots, n_{hid}$ **do**
19: For hidden dimension $j$ of subspace $i$, determine the complementary subspace with a shared active parameter and free dimension corresponding to subspace $i$'s $j^{th}$ hidden dimension
20: Update value of subspace $i$'s hidden dimension $j$ by sharing information with the identified complementary subspace
21: **end for**
22: **end for**
23: **end while**

# SCOOP Overview

**Algorithm 1** Subspace COmmunication based OPtimization

**Input:** $f(\mathbf{x})$, $a(\mathbf{x})$, $B$, $n_{sub}$, $n_{hid}$, $b_{init}$, $b_{opt}$, information sharing strategy
**Output:** $\hat{\mathbf{x}}^*$, $f(\hat{\mathbf{x}}^*)$: the best sampled full dimensional location

Initialize: Select Constant/Hidden Dimensions of each Subspace

**while** $B > 0$ **do**
*Execute Subspace Optimizations*

Bayesian Optimization: Optimize over each Current Subspace

*Execute Subspace Information Sharing: Update Hidden Dimension Values*

Information Sharing: Update Constant/Hidden Dimension *values of each Subspace* via complementary Subspaces

**end while**

# SCOOP Overview

**Algorithm 1** Subspace COmmunication based OPtimization

**Input:** $f(\mathbf{x})$, $a(\mathbf{x})$, $B$, $n_{sub}$, $n_{hid}$, $b_{init}$, $b_{opt}$, information sharing strategy
**Output:** $\hat{\mathbf{x}}^*$, $f(\hat{\mathbf{x}}^*)$: the best sampled full dimensional location

Initialize: Select Constant/Hidden Dimensions of each Subspace

3: **while** $B > 0$ **do**
*Execute Subspace Optimizations*
4:
5:
6:
7:
8:
9:
10:
11:
12:
13:
15:
16:

Bayesian Optimization: Optimize over each Current Subspace

*Execute Subspace Information Sharing: Update Hidden Dimension Values*
17:
18:
20:
21:
22:

Information Sharing: Update Constant/Hidden Dimension *values of each Subspace* via complementary Subspaces

23: **end while**

# Information Sharing

- Best Observed Sample Sharing (SCOOP-B)

$$\boldsymbol{x}_j^{\text{share}} \in \arg \min_{\boldsymbol{x} \in \mathbb{X}^{SS_j}} f(\boldsymbol{x})$$

- Marginalized Expected Improvement (SCOOP-E)

$$\boldsymbol{x}_j^{\text{share}} \in \arg \min_{\boldsymbol{x} \in \mathbb{X}^{SS_j}} EI_M$$

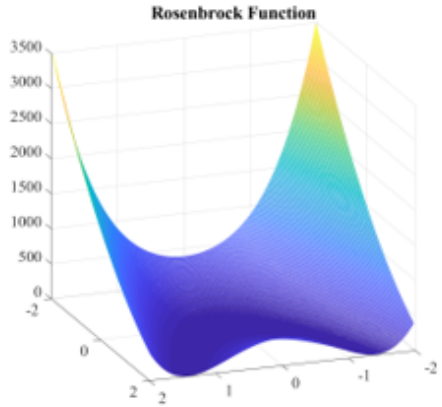
where

$$EI_M(\boldsymbol{x}|SS_j) = \int_{x_i} E\left[\Delta_f(\boldsymbol{x})\,\Phi\left(\frac{\Delta_f(\boldsymbol{x})}{s(\boldsymbol{x})}\right) + s(\boldsymbol{x})\,\phi\left(\frac{\Delta_f(\boldsymbol{x})}{s(\boldsymbol{x})}\right)\right] dx_i$$

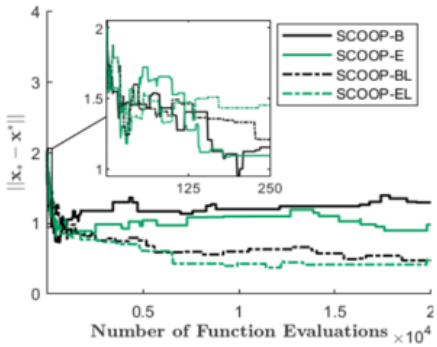
- Link Failure

$$x_{ij}^{\text{share}} = (1-\xi)\,\hat{x}_{ij}^{\text{share}} + \xi\eta \;\forall i, \xi \sim \text{Ber}(\alpha), \eta \sim \text{Beta}(b_1, b_2)$$

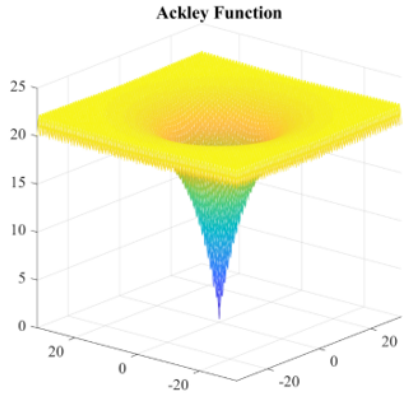
# Sharing Strategy Testing in $\mathbb{R}^3$



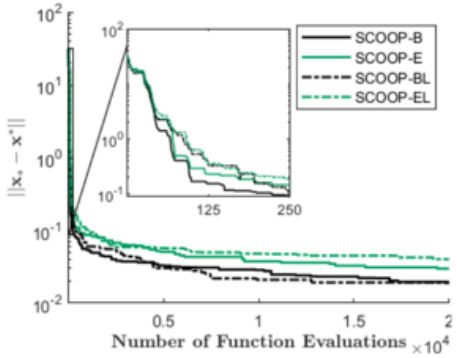

(a) Rosenbrock.




(b) Rosenbrock Results.




(c) Ackley.




(d) Ackley Results.

**Figure:** Performance of SCOOP in 2-d under several sharing strategies.

It appears that SCOOP-BL is the best performer under the different set-ups.

# Algorithm Parametrization Testing in $\mathbb{R}^{50}$

Experimentation executed to test 3 algorithm parameters:

- $SS_{dim}$ – number of dimensions active in each subspaces
  - $SS_{dim} = 2$, yielding 25 subspaces
  - $SS_{dim} = 5$, yielding 10 subspaces
- $b_{init} : b_{opt}$ – ratio of initializing to optimizing samples
  - $b_{init} : b_{opt} = 1 : 1$
  - $b_{init} : b_{opt} = 1 : 5$
- % Link Fail – % hidden dimension value updates that fail
  - 0% Link Fail
  - 10% Link Fail

# Algorithm Parametrization Testing in $\mathbb{R}^{50}$

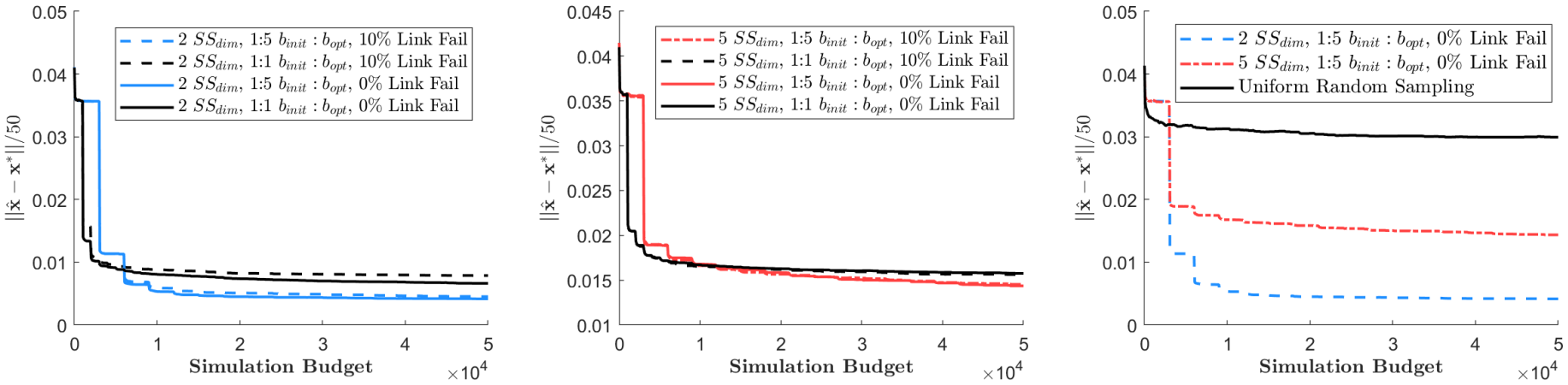
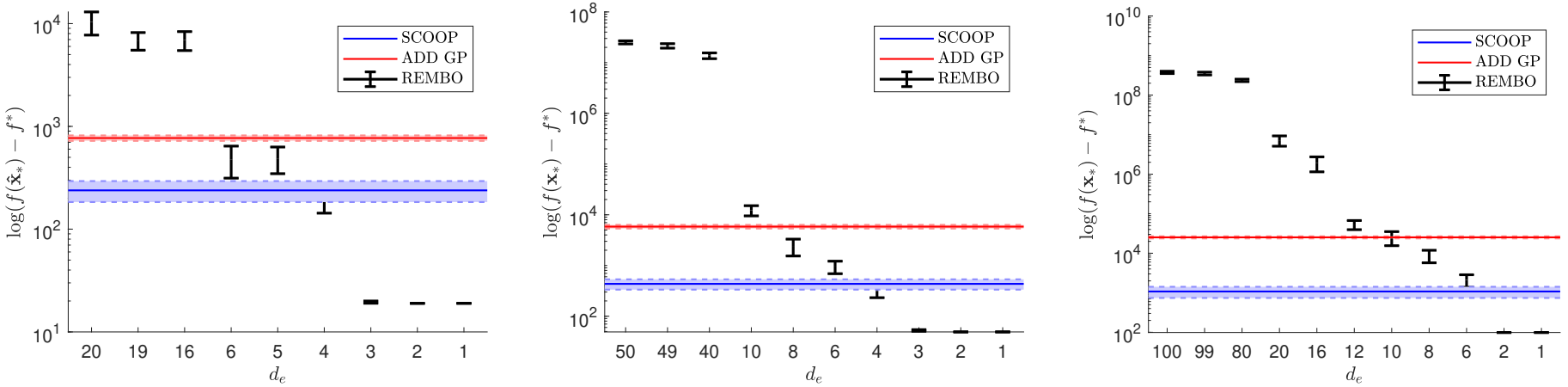



**Figure:** Alternative SCOOP formulations tested on 50 dimensional Ackley function, results presented are over 50 algorithm execution replications.

Better optimization at each iteration provides higher quality shared information, yielding better overall algorithm performance.

# State-of-the-Art Comparison

SCOOP against REMBO and an Additive Gaussian process approach over a 20, 50, and 100 dimensional Rosenbrock function.



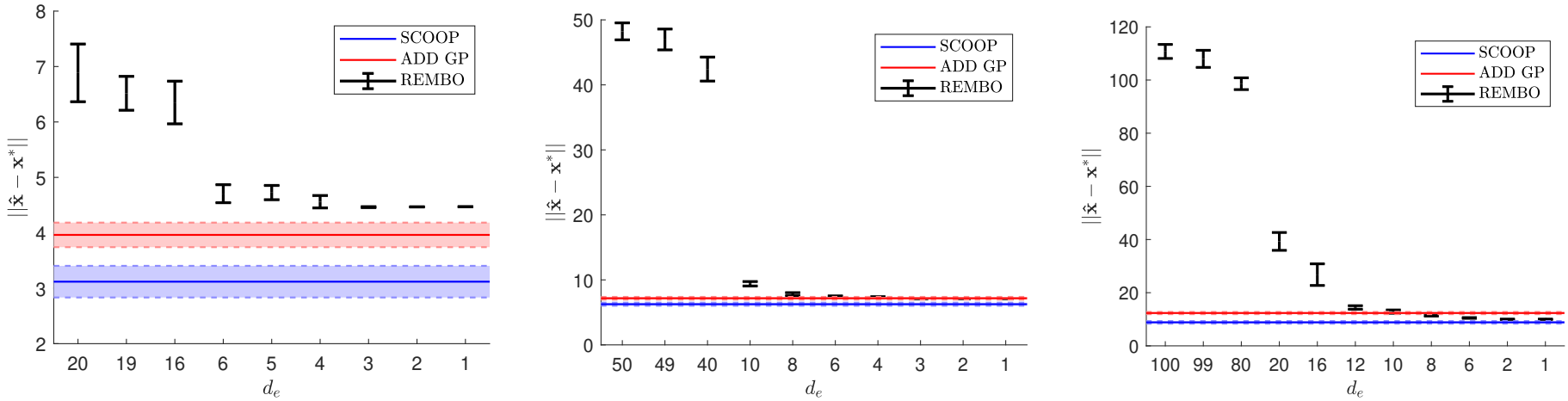

**(a)** Rosenbrock in $\mathbb{R}^{20}$ **(b)** Rosenbrock in $\mathbb{R}^{50}$ **(c)** Rosenbrock in $\mathbb{R}^{100}$

**Figure:** Average best function value achieved by SCOOP, ADD GP, and REMBO (with varying embedding dimension, $d_e$).

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., & de Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. Journal of Artificial Intelligence Research, 55, 361-387.
Wang, Z., C. Li, S. Jegelka, and P. Kohli. 2017. "Batched High-dimensional Bayesian Optimization viaStructural Kernel Learning". InInternational Conference on Machine Learning (ICML).

# State-of-the-Art Comparison

SCOOP against REMBO and an Additive Gaussian process approach over a 20, 50, and 100 dimensional Rosenbrock function.




**(a)** Rosenbrock in $\mathbb{R}^{20}$ **(b)** Rosenbrock in $\mathbb{R}^{50}$ **(c)** Rosenbrock in $\mathbb{R}^{100}$

**Figure:** Average Euclidean error between identified minimum and true optimum for SCOOP, ADD GP, and REMBO (with varying $d_e$).

Wang, Z., Hutter, F., Zoghi, M., Matheson, D., & de Feitas, N. (2016). Bayesian optimization in a billion dimensions via random embeddings. Journal of Artificial Intelligence Research, 55, 361-387.
Wang, Z., C. Li, S. Jegelka, and P. Kohli. 2017. "Batched High-dimensional Bayesian Optimization viaStructural Kernel Learning". InInternational Conference on Machine Learning (ICML).

# Conclusions and Future Work

- SCOOP directly optimizes over multiple low dimensional subspaces, leveraging information communication among these easy optimizations to navigate the hard to search full dimensional space.
- Further efforts are undergoing on tests in 1000 active dimensions environments and alternative information sharing strategies
- Optimal selection of active subspace dimensions and communication patterns/network amongst subspaces

# Thanks for Listening!

lmathese@asu.edu

Ackley function

$$f(\boldsymbol{x}) = -a \exp\left(-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(cx_i)\right) + a + \exp(1)$$

Rosenbrock function

$$f(\boldsymbol{x}) = \sum_{i=1}^{d-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

Configuring neural net topology with 6, 9, and 15 layers. Wall-clock time constrained optimization, accounting for both neural net selection and training.

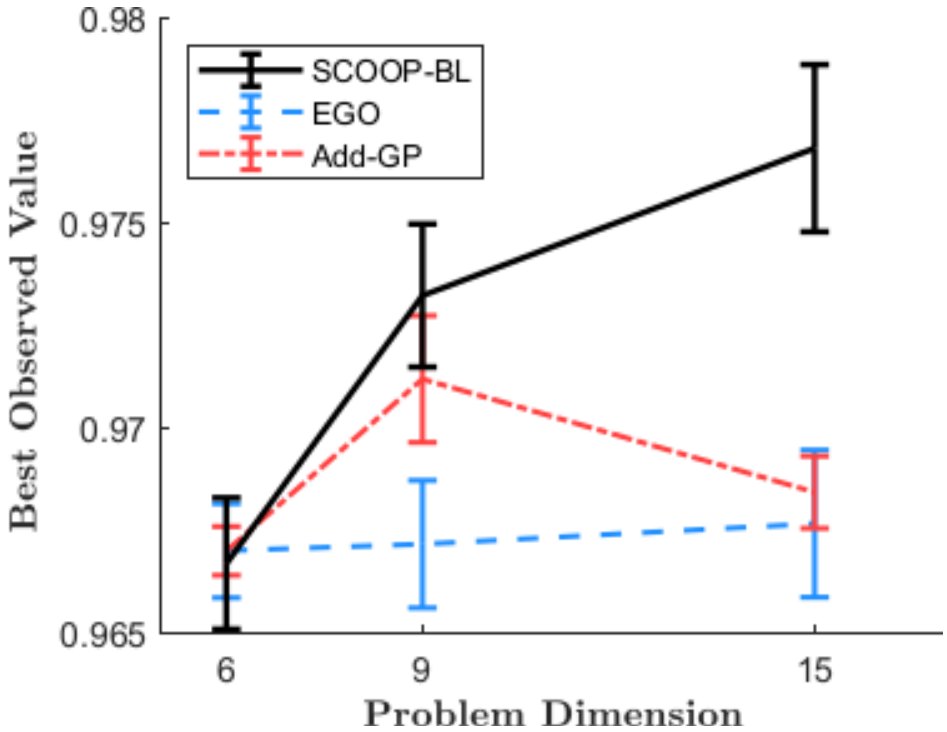



**Figure:** Neural net prediction accuracy results across nets with differing number of layers (dimension).

Efficient Global Optimization (EGO), Batched Additive Bayesian Optimization via Structured Kernel Learning (Add-GP)

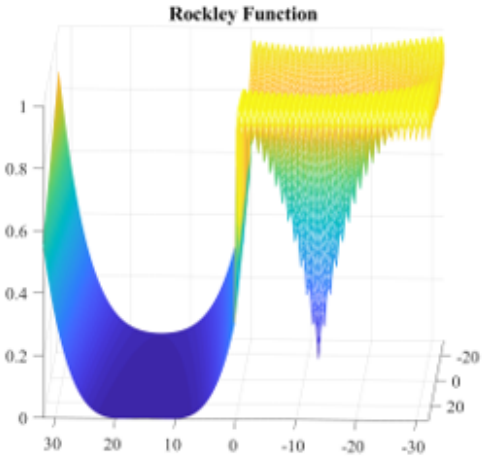
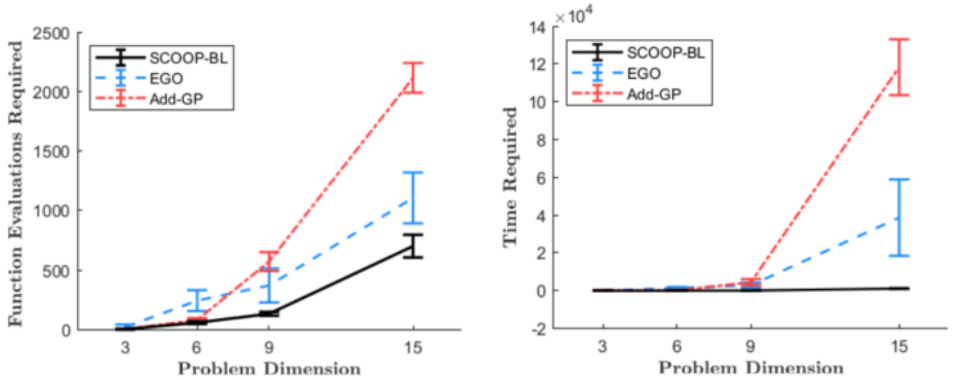


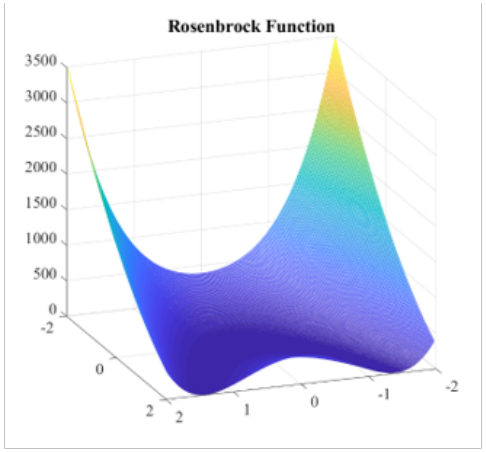

**Figure:** Combined Function

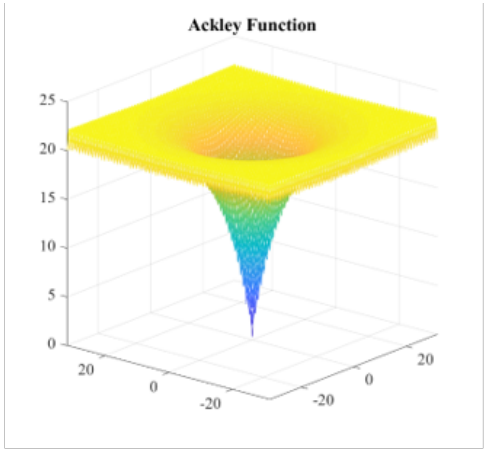


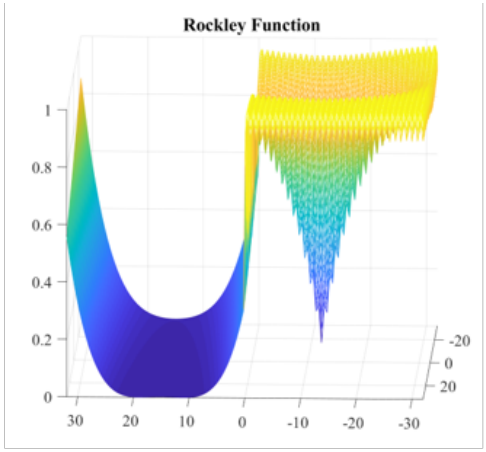

**(a)** Evaluations required to optimize. **(b)** Time required to optimize.

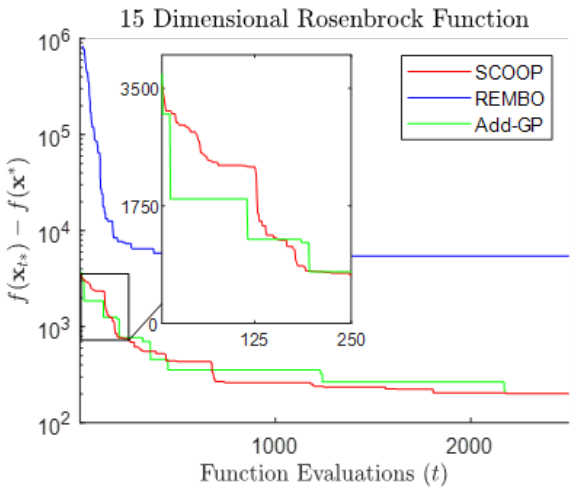
**Figure:** Performance of SCOOP.

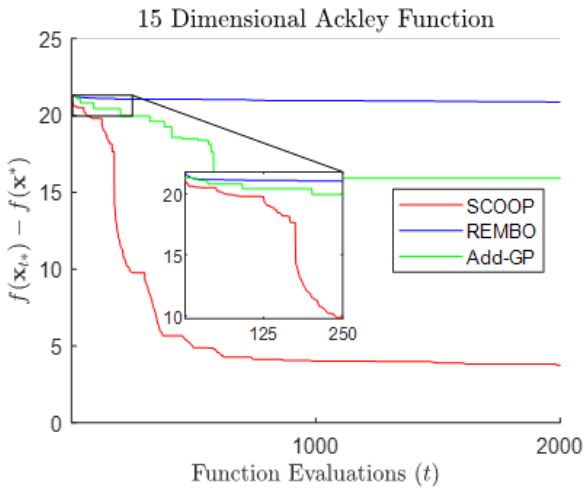
Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. Journal of Global optimization, 13(4), 455-492.
Wang, Z., Li, C., Jegelka, S., & Kohli, P. (2017). Batched high-dimensional bayesian optimization via structural kernel learning. arXiv preprint arXiv:1703.01973.



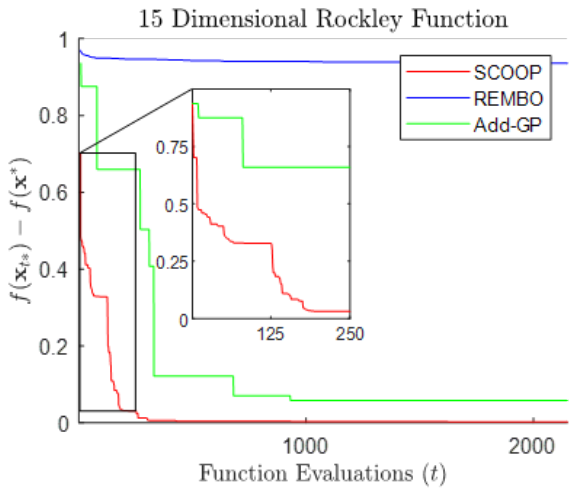




**Figure:** Higher Dimensional Results