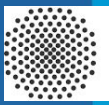


Universität Stuttgart

Latent Space Bayesian Optimization with Transfer Learning

Jagan Shanmugam

Outline

- Motivation
- Problem statement
- Background
  - Bayesian Optimization
- Existing methods
  - Random Embeddings for Bayesian Optimization
  - Multi-Task Adaptive Bayesian Linear Regression (ABLR)
- Latent space models and Joint learning
  - Projection Multi-Task ABLR
  - Auto Encoder Multi-Task ABLR
- Latent space Bayesian Optimization with Transfer Learning
- Experiments
  - Synthetic functions
  - Results

Motivation

- Hyperparameters of industrial processes **(black-box functions)** are manually tuned in different context/settings
- Oftentimes, only a subset or linear/nonlinear combination of parameters are relevant
- Should adapt to unseen tasks by learning from previous evaluations (metadata)

Goal: Find the optimal parameters of a black-box function by optimizing in low dimensional space and leveraging collected information from previous runs

Optimization run	***Settings/Context in Welding process***	***Dataset***
Run/Task 1	*Material type 1, Electrode type 1, Size, ..*	*Dataset 1*
Run/Task 2	*Material type 2, Electrode type 2, Size, ..*	*Dataset 2*
…..	…..	…..
Run/Task T	***Material type T, Electrode type 1, Size, ..***	***Warm start from previous runs/tasks, to increase sample efficiency in Task T***

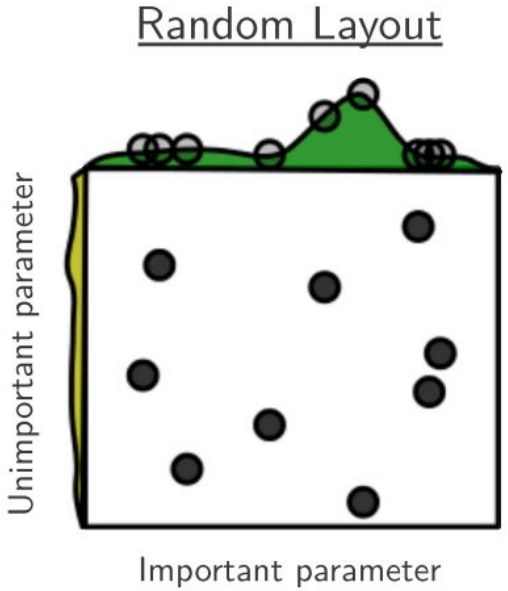
Problem statement

- Following black-box optimization problem:

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

  - Evaluation is expensive and noisy
  - No analytical form or gradient, possibly non-convex
  - Parameter space $\mathcal{X}$ - high but *Intrinsically low dimensional*

- Leverage data from evaluations of related black-box functions: $\{(\mathrm{f}_t)\}_{t=1}^{T}$

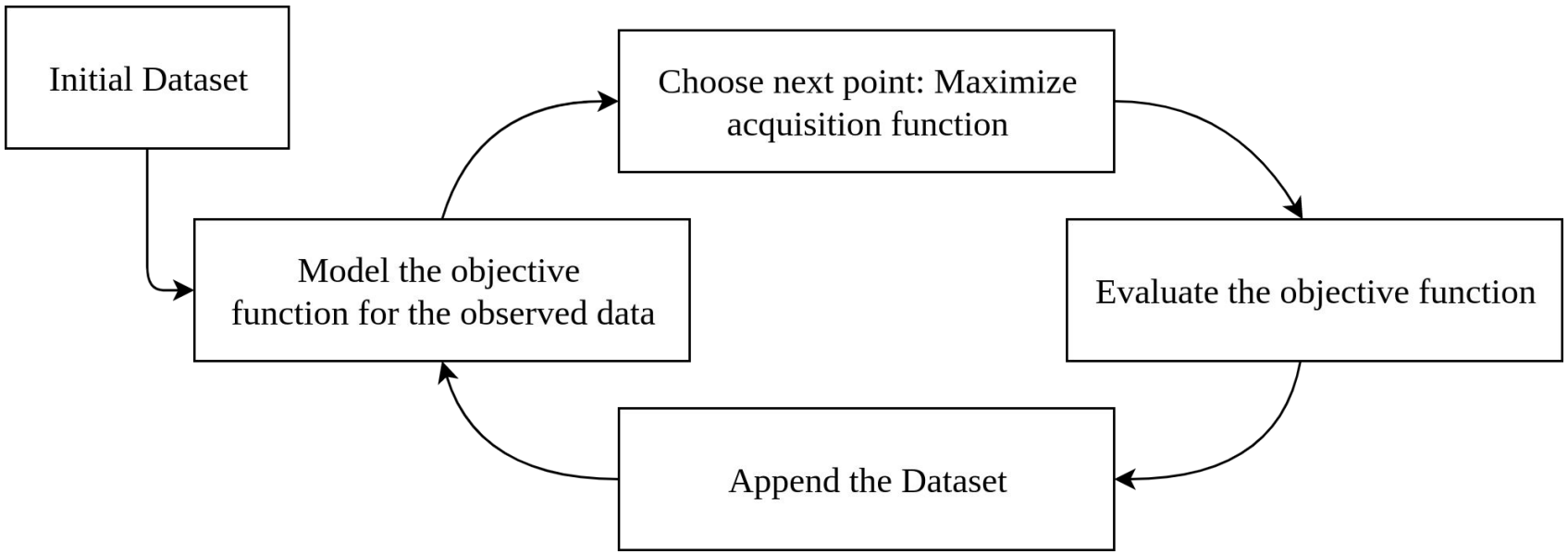


Random layout of parameters in a function with one important and one unimportant parameter*

*Image source: Bergstra, J. & Bengio, Y. (2012). Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.*, 13, 281-305.

BOSCH

Bayesian Optimization (BO)

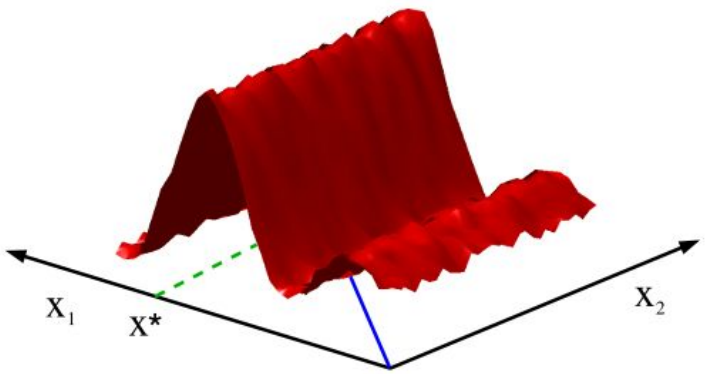


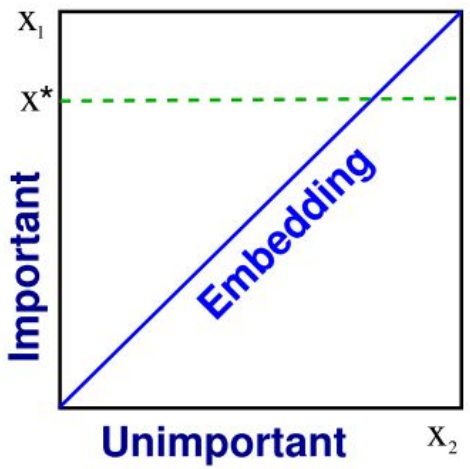
Limitations:

- Cold start problem: When started from scratch, initial iterations explore the input space randomly
- High Dimensional BO: Maximizing nonconvex acquisition function in high dimensional input spaces does not lead to reliable estimate of next input point to evaluate

Existing methods -

Random Embeddings for BO (REMBO)





- Random projection matrix, $A \in \mathbb{R}^{D \times d}, d < D$
- Sample a point $z \in \mathbb{R}^d$ in low dimensional embedding within $[-\sqrt{d}, \sqrt{d}]^d$ to optimize the acquisition function in low dimensional space
- Evaluate on the function: $f(Az)$

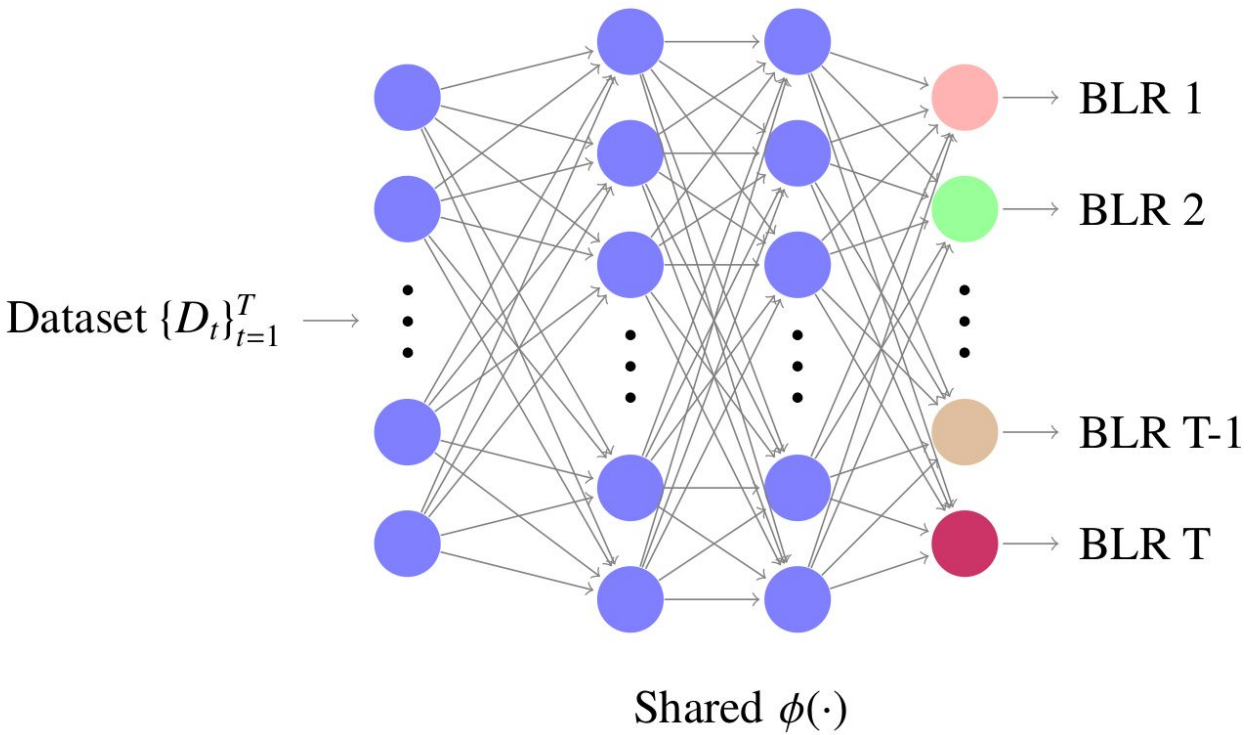
*2D function with 1 effective dimension, d=1, D=2

*Image source: Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando De Freitas. 2016. Bayesian optimization in a billion dimensions via random embeddings.

BOSCH

Existing methods -

Multi-Task Adaptive Bayesian Linear Regression (ABLR)



- Extension of Adaptive Bayesian Linear Regression to multi-task cases
- Shared feature learning network for all tasks
- One Bayesian Linear Regressor as a last layer for each task
- **Training:** By minimizing the sum of Negative Log Likelihood of all tasks with points from one task forming a batch

Existing methods -

Multi-Task Adaptive Bayesian Linear Regression (ABLR)

T Black-box functions: $\{(\mathrm{f}_t)\}_{t=1}^{T}$

Data evaluated on each function: $\mathrm{D}_t = \{(x_t^n, y_t^n)\}_{n=1}^{N_t}$

Dataset: $\{\mathrm{D}_t\}_{t=1}^{T}$

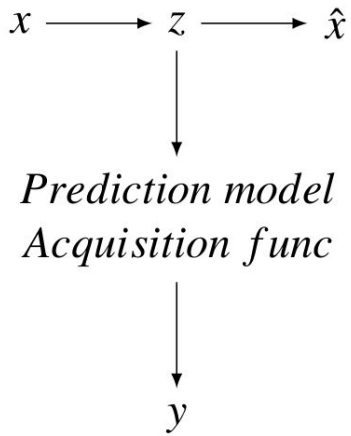
$$P(w_t \mid \alpha_t) = \mathcal{N}(0,\, \alpha_t^{-1}\mathbb{I}_P)$$
$$P(y_t \mid w_t, z, \beta_t) = \mathcal{N}(\Phi_t w_t,\, \beta_t^{-1}\mathbb{I}_{N_t})$$

$$\Phi_t = [\phi_z(x_t^n)]_n \in \mathbb{R}^{N_t \times D}$$

$$\beta_t > 0,\ \alpha_t > 0$$

Latent space models

Idea: Joint learning of low dimensional latent space and prediction model on the latent space by learning shared features from the latent space



➔ Enables BO in the low dimensional space

➔ Speeds up BO by learning from other tasks in multi-task cases

Joint training: Minimizing the sum of Negative Log Likelihood and MSE for all tasks

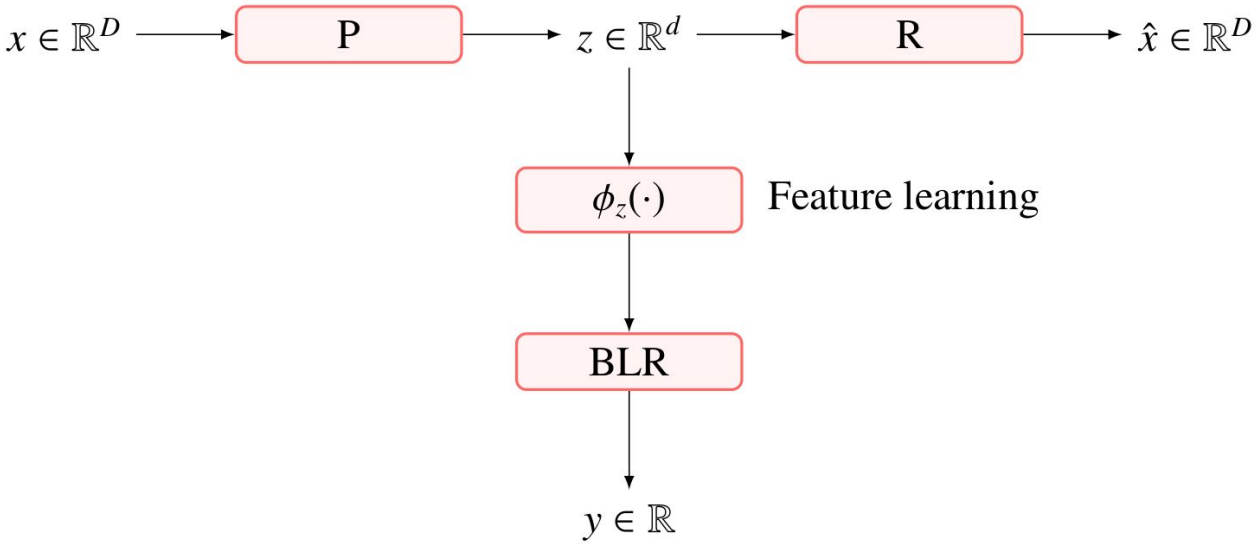
$$Total\ loss = \sum_{t=1}^{T}[-log\ P(y_t/z, \alpha_t, \beta_t) + \|x_t - \hat{x}_t\|_2^2]$$

Two variants:

- Projection ABLR model
- Auto Encoder ABLR model

Latent space models -

Projection ABLR model



- Single Task case - Typical BO
- Linear transformation of input space to latent space and vice-versa
- Projection (*P*) and Reconstruction (*R*) matrices are learned during BO

Latent space models -

Projection Multi-Task ABLR model

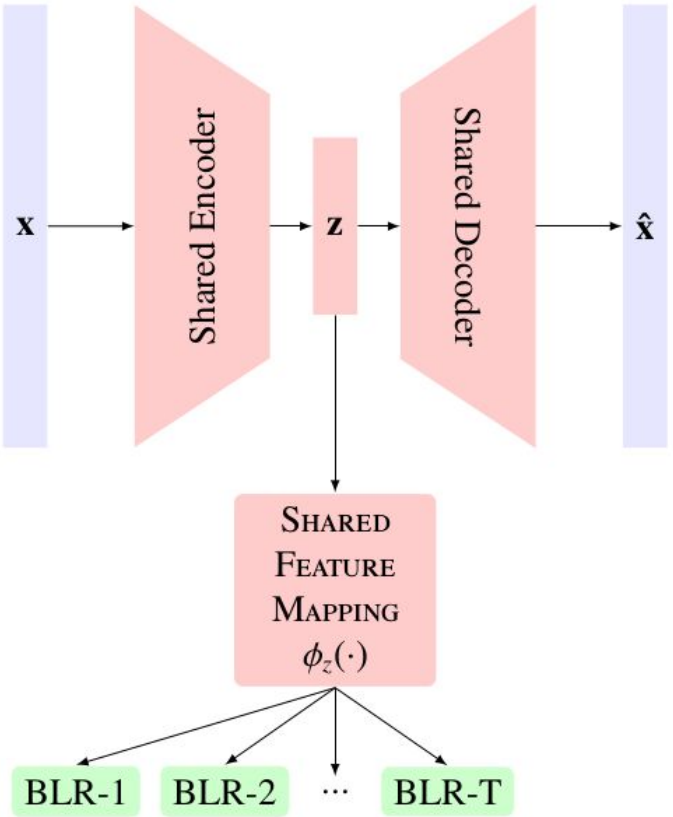
- Learn shared Projection and Reconstruction matrices for all tasks along with ABLR parameters from metadata

$$z = Px \qquad \mathrm{P} \in \mathbb{R}^{d \times D}$$
$$\hat{x} = Rz \qquad \mathrm{x} \in \mathbb{R}^{D}$$
$$\mathrm{z} \in \mathbb{R}^{d}$$
$$\mathrm{R} \in \mathbb{R}^{D \times d}$$

- With one BLR layer for each task and learning a linear shared Projection/Reconstruction mapping from metadata, a Multi-Task model is constructed

Latent space models -

Auto Encoder Multi-Task ABLR model



- Shared Encoder, Decoder, Feature mapping for all tasks
- Nonlinear transformation of input space to latent space
- Constrain the latent space to be gaussian using Variational Auto Encoder approach, a variant implemented and tested, which is conceptually similar to VAE-BO

Latent space Bayesian Optimization

while *current step <= max iterations* **do**
- *Train the Latent model: Target-train*
  - Dimensionality reduction: $z = Mapping(x)$
  - Learn the low dimensional surface: $p(f_z/z, y)$
- *Select next input for evaluation:*
  - Maximize Acquisition function over: z_{next}
  - Map z_{next} to input space: x_{next}
  - Clip x_{next} if projected outside bounds
- Evaluate x_{next} on Target Task T
- Append the dataset $D_T = \{x_{next}, f_T(x_{next})\}$
- Increment current step by 1

- Model resides in low dimensional space -> Acquisition function is optimized in low dimensional latent space
- During BO, point to evaluate on the target task's objective function in input space is reconstructed from the point in latent space
- **Latent Space BO with Transfer Learning**: Multi-Task model is trained on metadata offline
  - Mapping (Projection or AE model) is learned during meta-training and adapted during BO
  - Shared features are learned from latent space by using Multi-Task ABLR model

Experiments -

Synthetic functions

- Parameterized Quadratic function:
  - Multiple tasks are generated by sampling the parameters (a, b, c)

$$f(z) = \frac{1}{2}a||z||^2 + b\mathbf{1}^T z + 3c$$

- Parameterized Rosenbrock function (log-scaled):
  - Number of settings: n_s (Values of settings differentiate tasks)

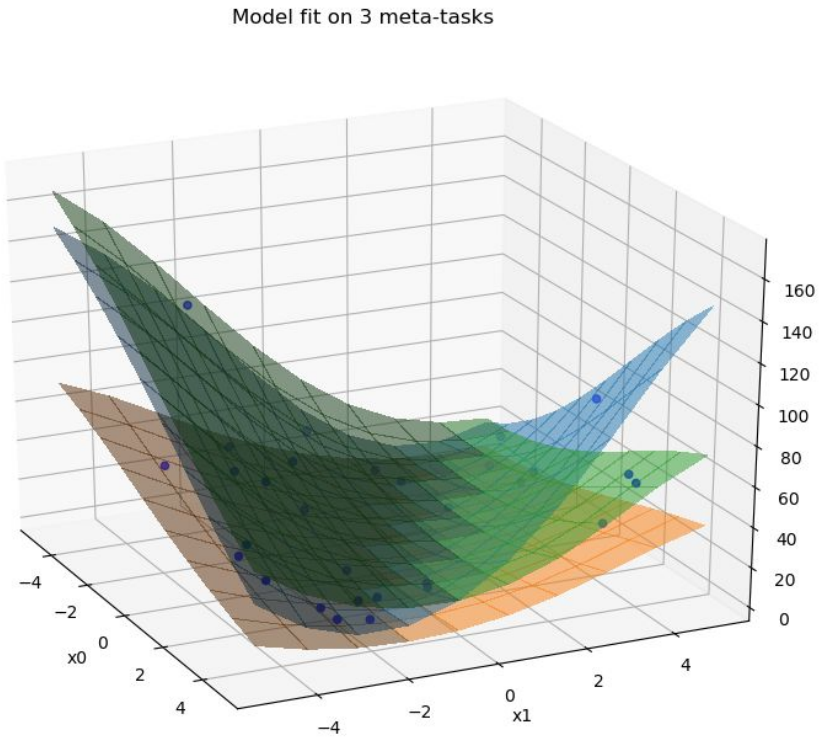
$$f(z) = \sum_{i=1}^{(n_s+d)-1} [100(z_{i+1} - z_i^2)^2 + (z_i - 1)^2]$$

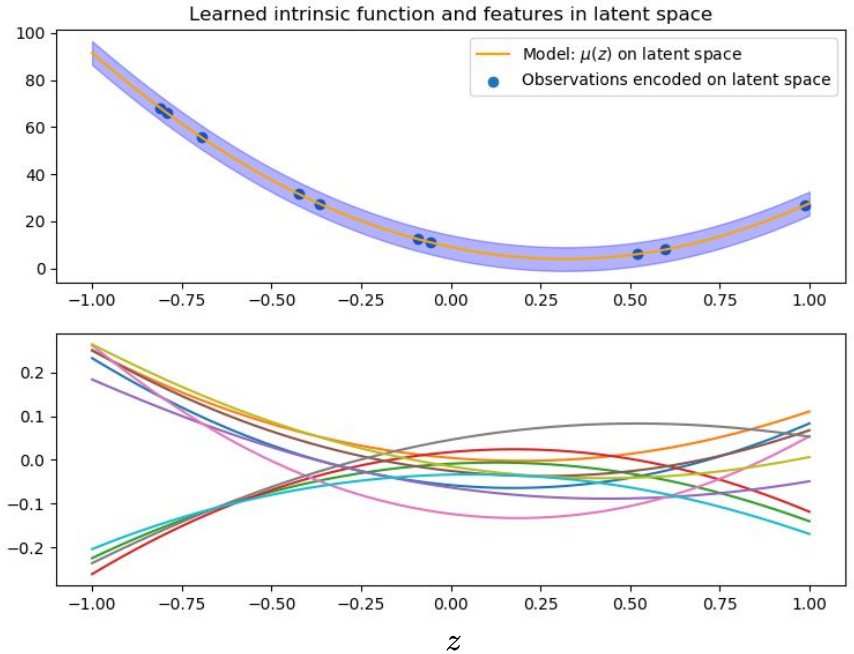
- **High dimensional function with low intrinsic dimensionality**, with matrix $A \in \mathbb{R}^{d \times D}$ with orthogonal rows

$$f_X(x) = f(z) = f(Ax)$$

Experiments -

Meta-learning the intrinsic function



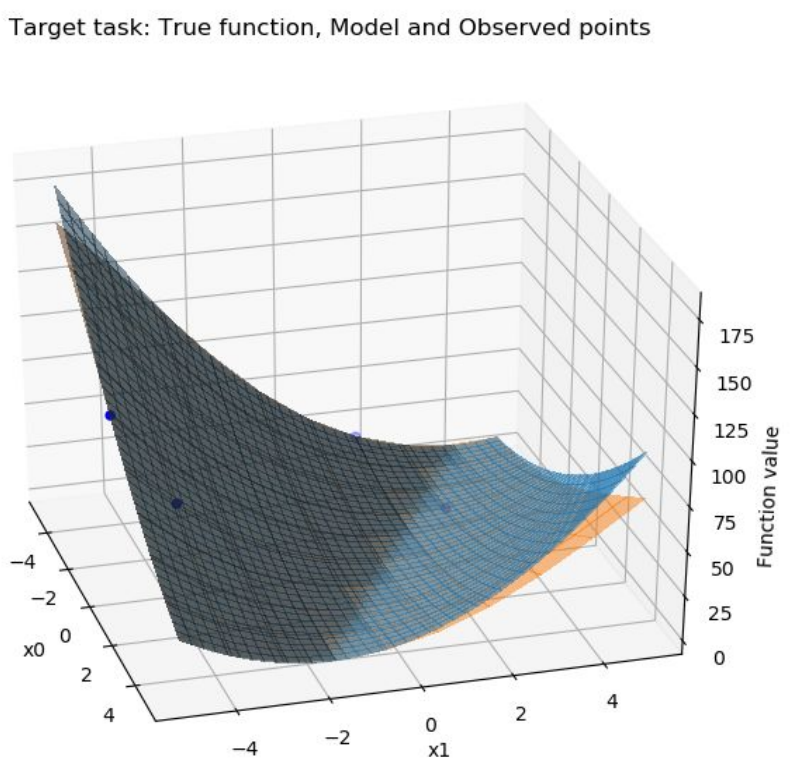


Projection-MT-ABLR model trained on metadata (30 tasks and 10 points each) of 2D Quadratic function with 1D intrinsic space

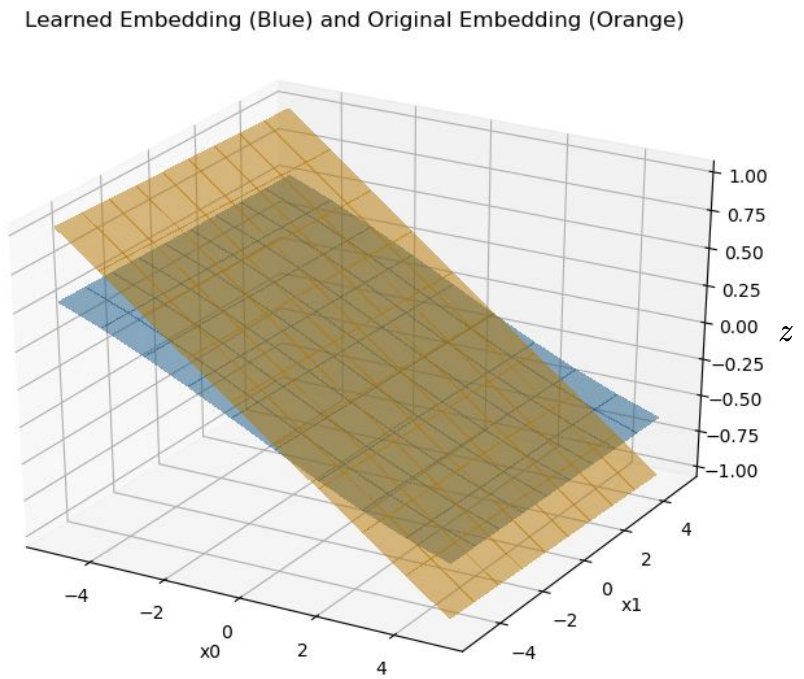
BOSCH

Experiments -

Meta-learning the intrinsic function



Projection-MT-ABLR model on Target task with only 4 randomly sampled points:Original function (Orange), Model fit (Blue)

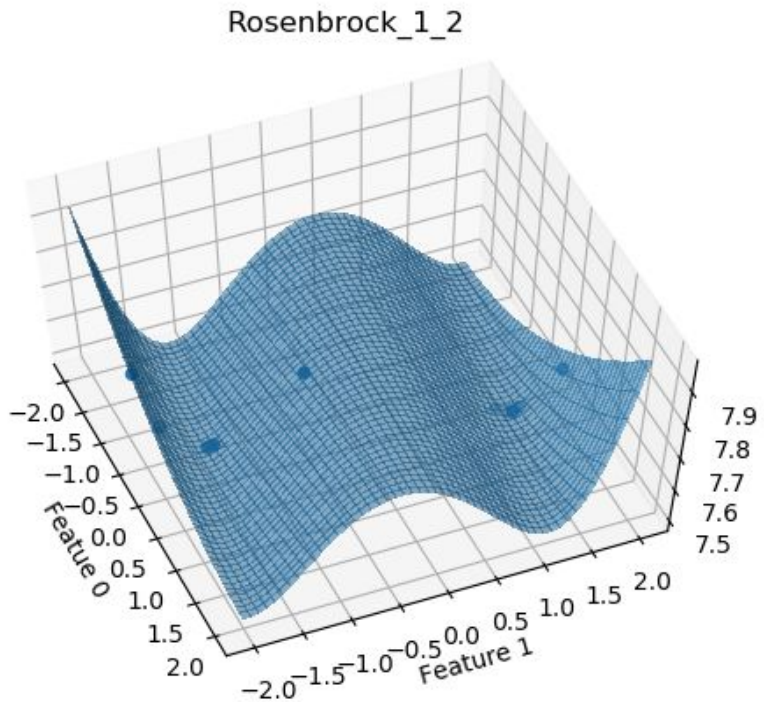


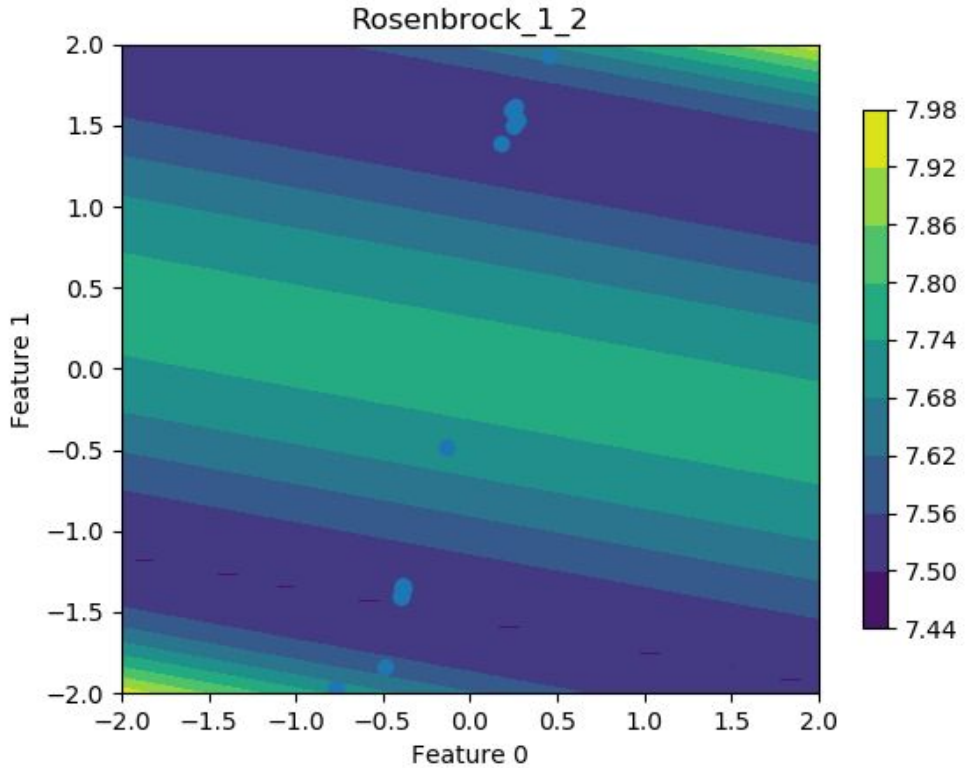
Learned and Original linear embedding in the input space: Every point in 2D input space is mapped via learned projection matrix to latent space

BOSCH

Experiments -

Low dimensional BO with Transfer Learning





2D Rosenbrock function with intrinsic dimensionality of one and points sampled during BO in the corresponding contour plot

BOSCH

Results

High Dimensional BO for parameterized Quadratic and Rosenbrock functions for two cases:

1. without Transfer Learning
2. with Transfer Learning

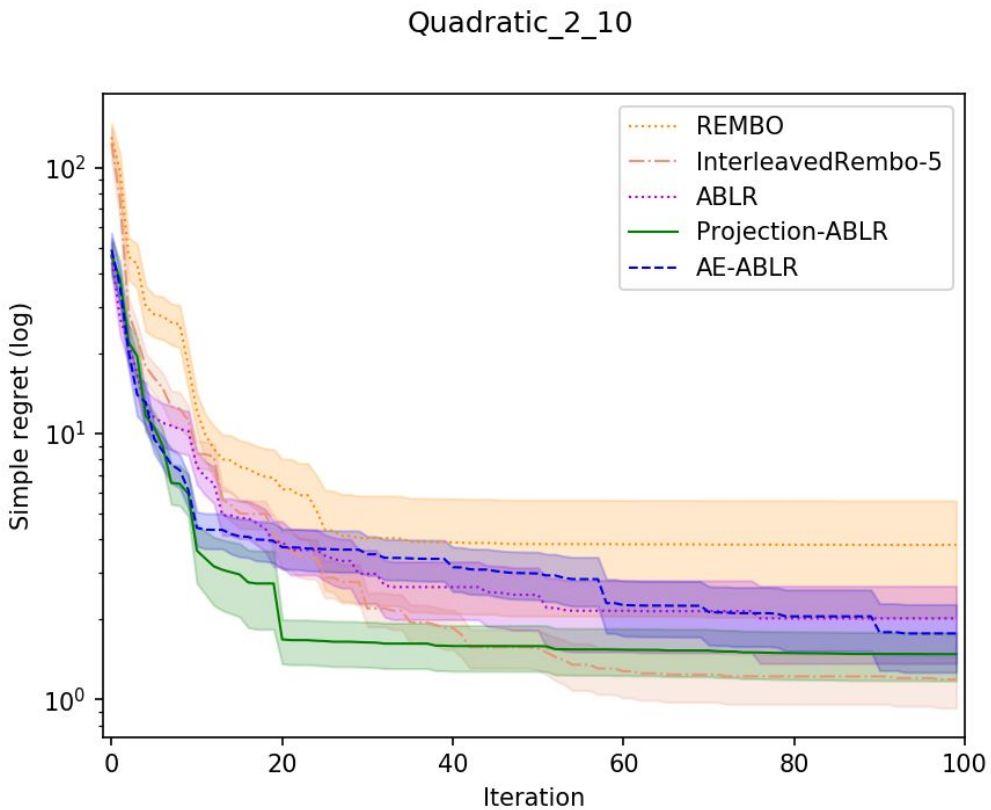
Metric for comparison - Simple regret or Optimality gap

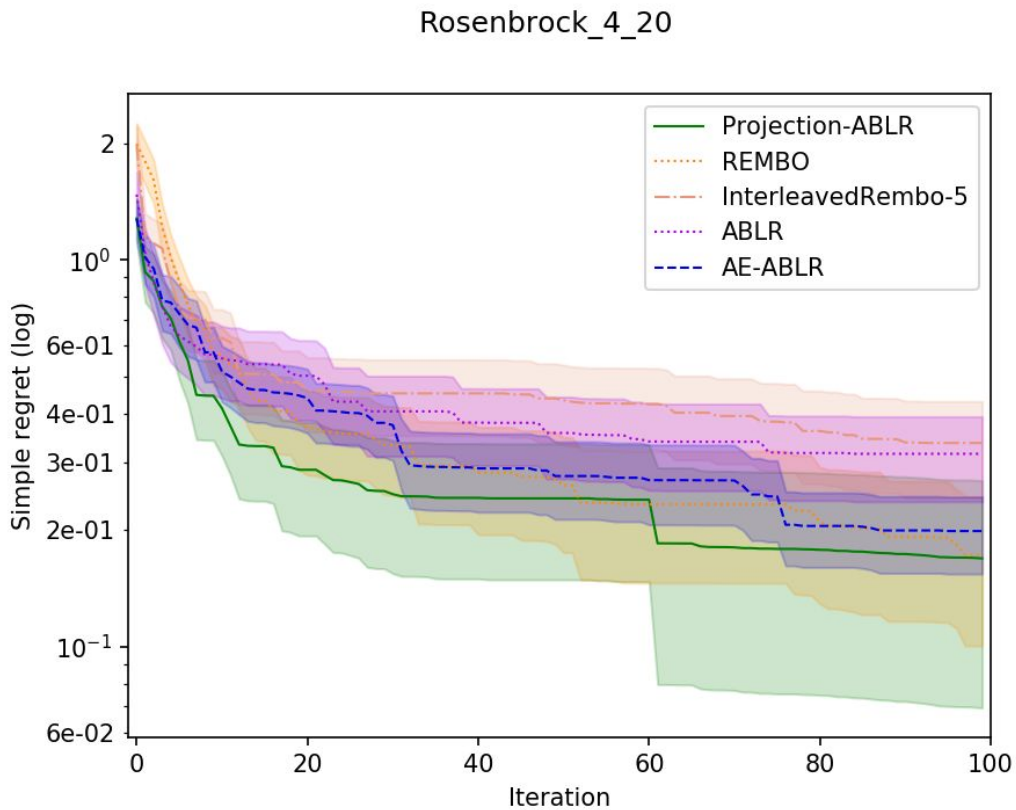
$$\text{Simple regret} = y_{current\_best} - y_{optimum}$$

Display format of benchmarks: *<function>_<d>_<D>*, where *d* is the intrinsic dimensionality and *D* is the actual number of input dimensions.

Results -

High Dimensional BO - without Transfer Learning



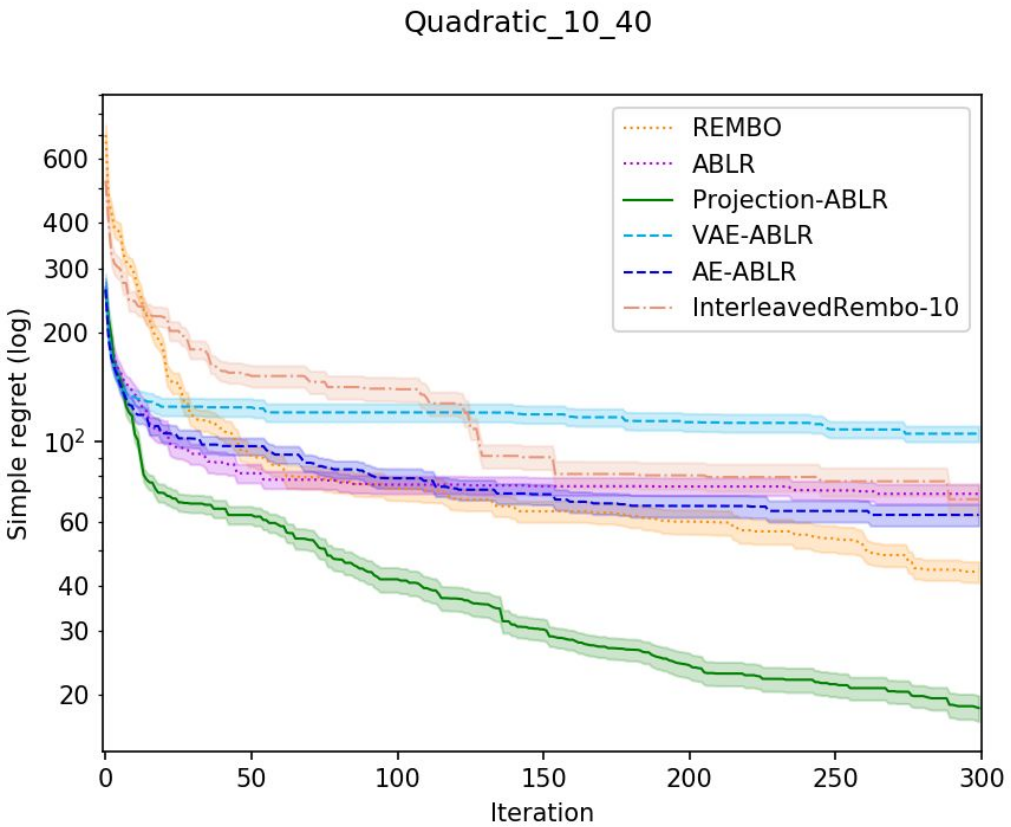


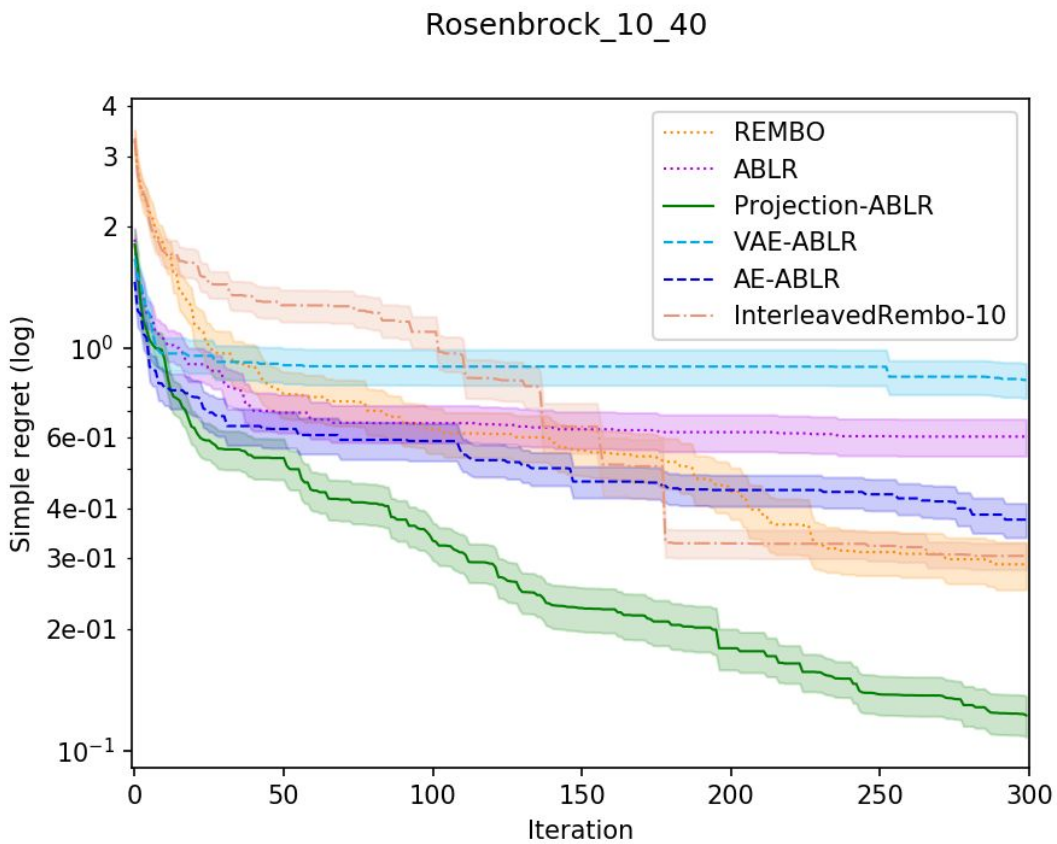
Typical HD-BO without meta-training the model

BOSCH

Results -

High Dimensional BO - with Transfer Learning





HD-BO with Transfer Learning models trained using metadata (30 tasks 10 points per task)

BOSCH

Summary

- Method which jointly learns low dimensional latent space mappings and multi-task ABLR model
- In the Transfer Learning setting, *Projection-MT-ABLR* achieves lower regret than other baselines
- Increasing metadata - Scales linearly with number of tasks and points per task
  - our model learn tasks better when points per task are moderate

Open challenges:

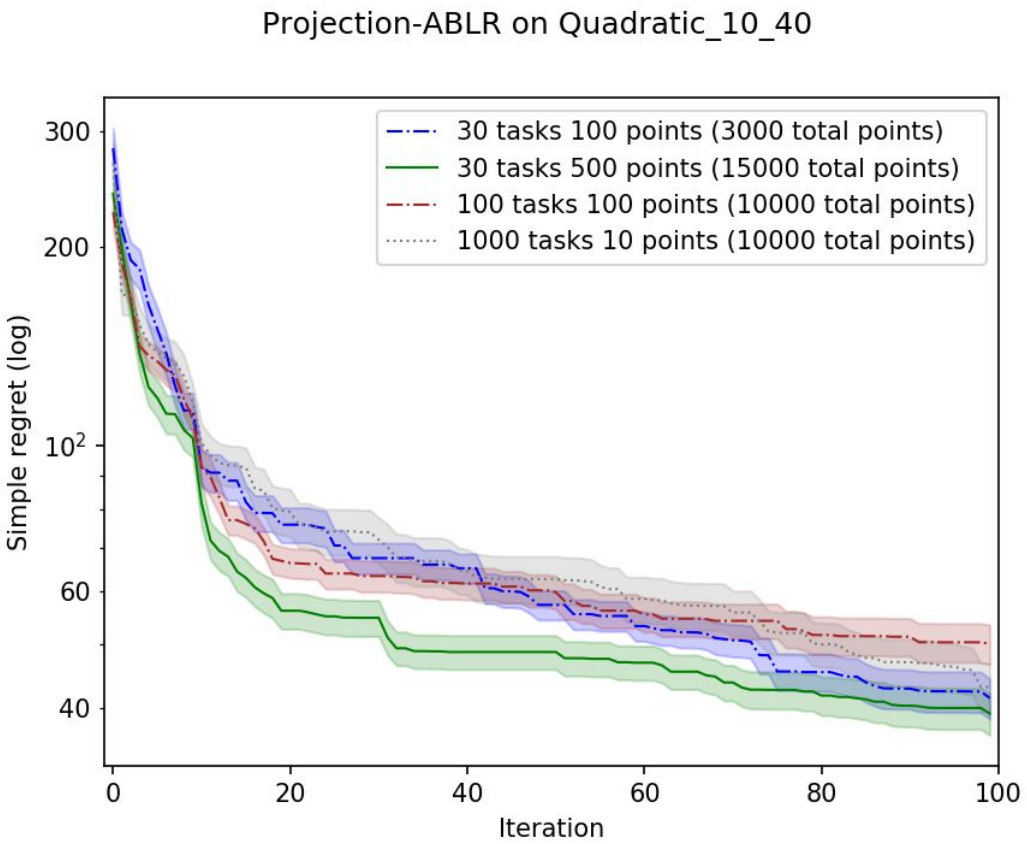
- Estimating intrinsic dimensionality from metadata
  - Tried out cross validation on multiple latent models with varying latent space dimension

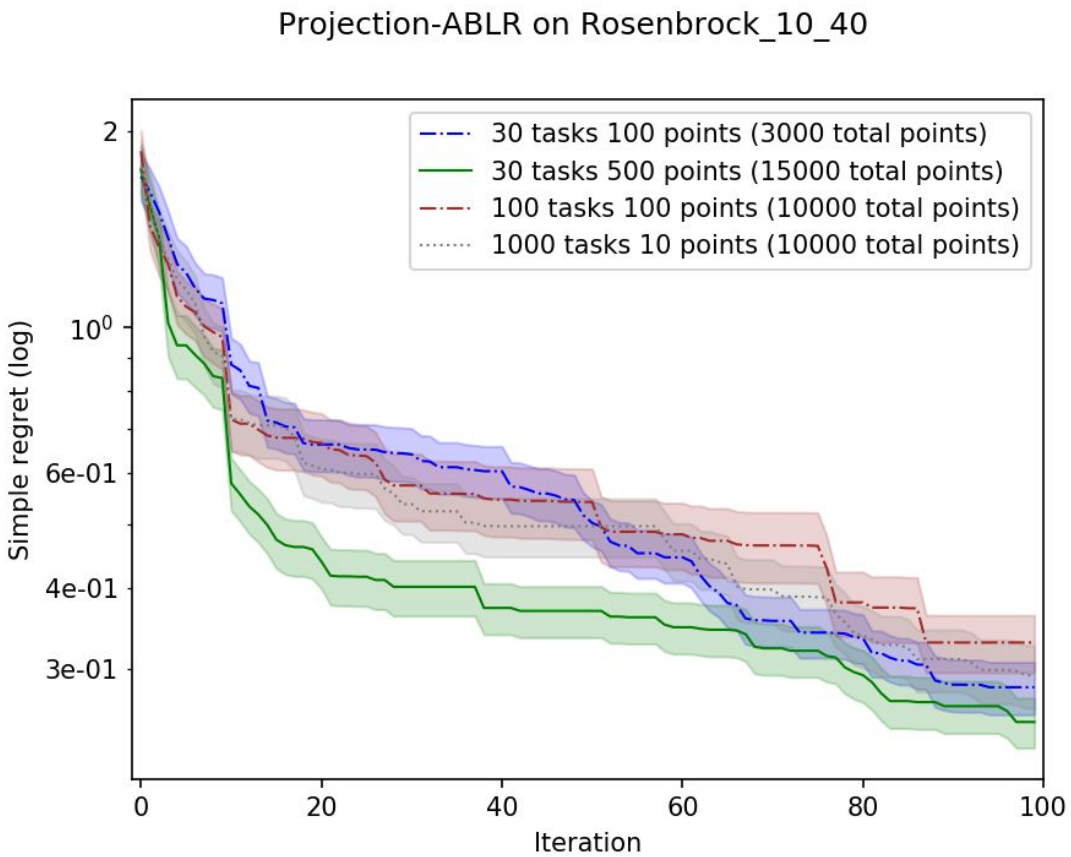
Thank you!
Questions?

Additional slides

Results -

Increasing metadata



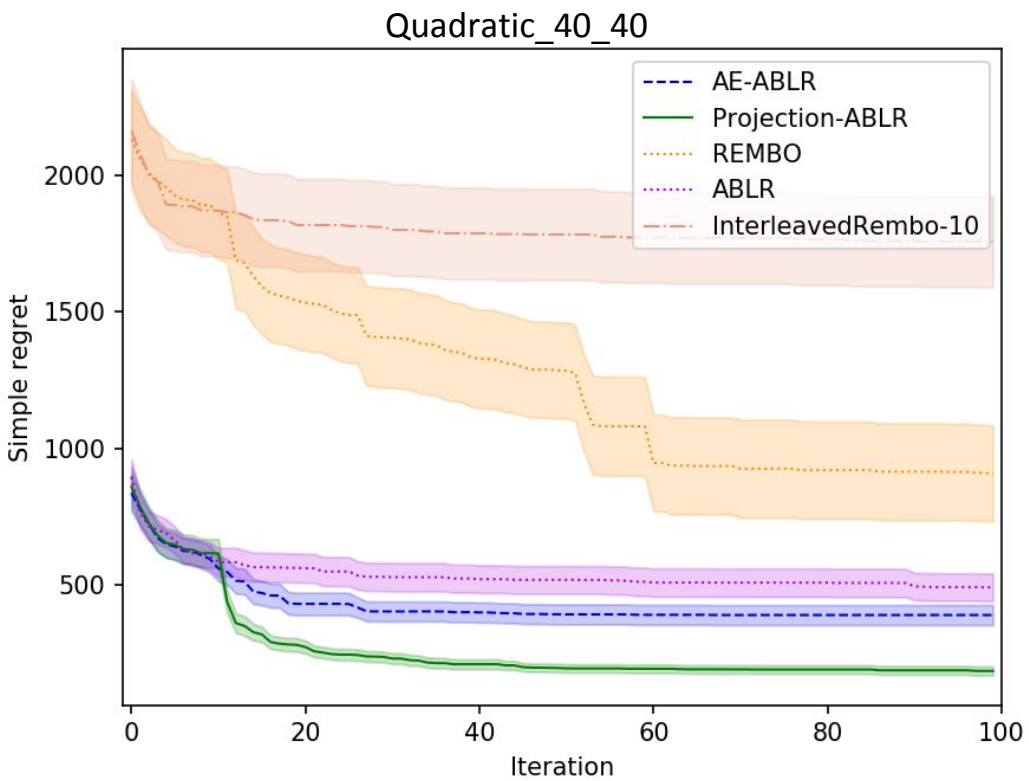


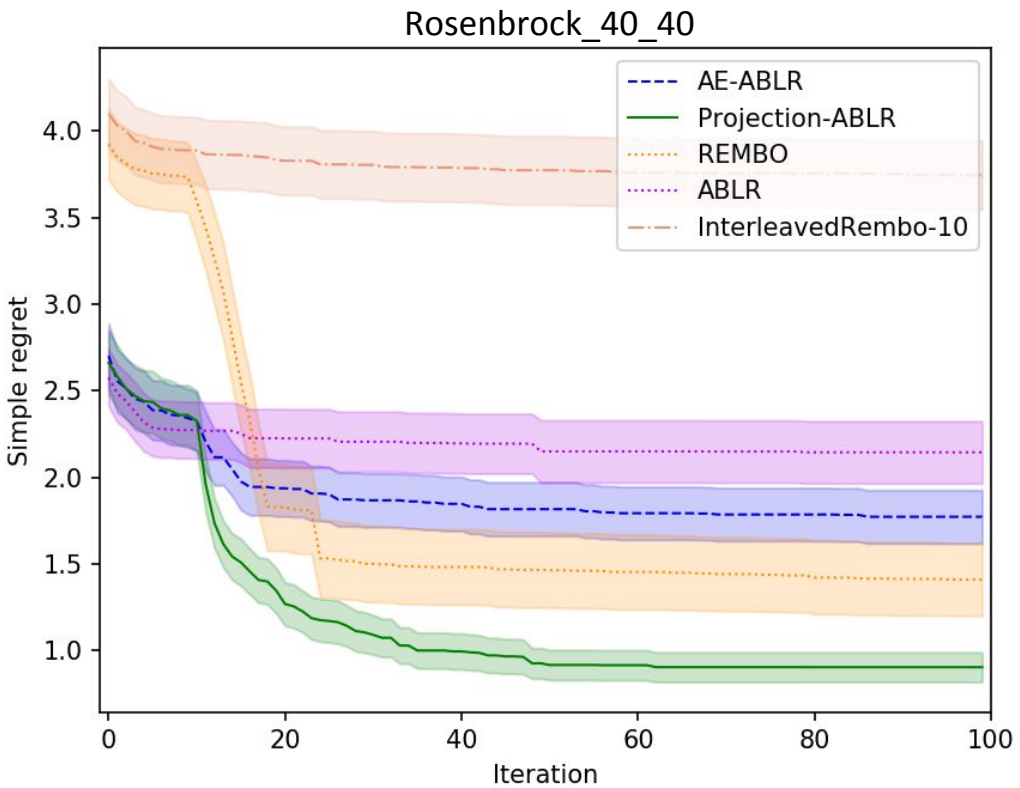
BOSCH

Results -

High Dimensional BO - without intrinsic structure

with Transfer Learning:





BOSCH

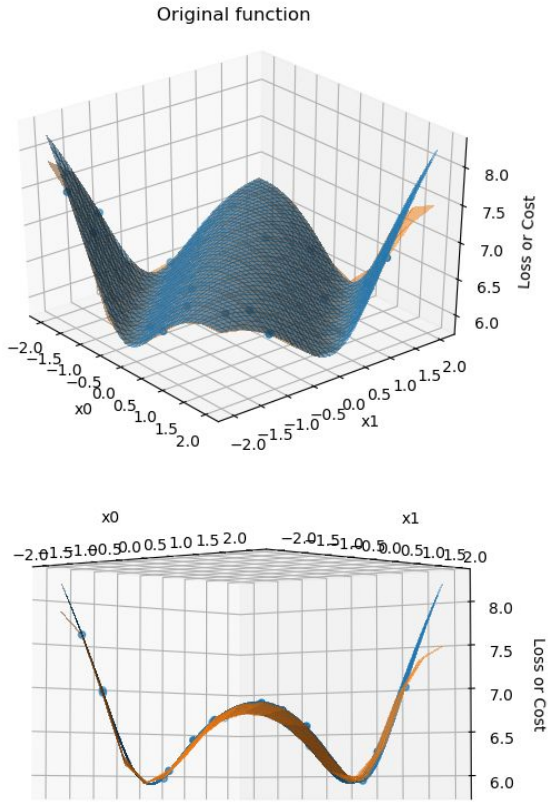
Experiments -

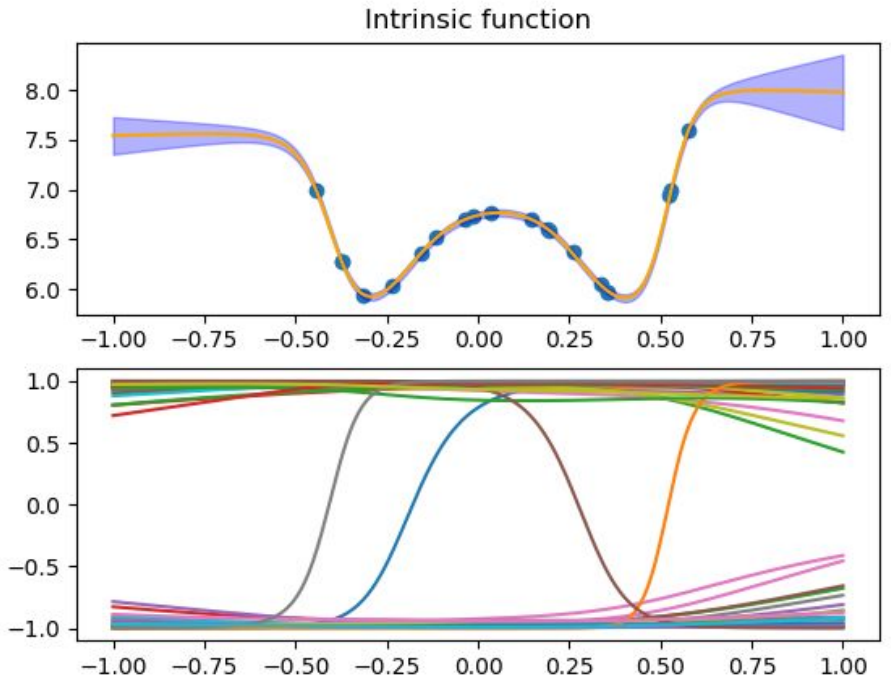
Models used in experiments

Model	Modeling space	Based on	Transfer Learning
REMBO, InterleavedRembo	Latent space $\mathbb{R}^d$	GP	No
MultiTaskABLR	Original space $\mathbb{R}^{\mathbb{D}}$	BLR	Yes
Projection-ABLR, AE-ABLR, VAE-ABLR	Latent space $\mathbb{R}^d$	BLR	Yes

Results -

Learning the intrinsic function





BOSCH