Supplementary Experiments for "Spectral Mixture Kernels for Bayesian Optimization"

1. Approximate Conventional Kernels

1.1. Single Kernels

We begin by considering a commonly used Matérn kernel with a smoother covariance structure with $\nu = \frac{5}{2}$, as given in **??** with $\ell = 20$. We sample 200 points from a 1-dimensional GP with this kernel. Next, we attempt to reconstruct the kernel underlying the sample points by training a CSM and GSM kernel using $Q = 7$ mixtures. For comparison, we also perform the approximation using squared exponential (SE) in **??** and rational quadratic (RQ) in **??**.

Table 1. Marginal likelihood (MLL) of training on the sampled data. Ideally, the source kernel should yield the highest MLL. For other kernels, a higher MLL indicates a better approximation of the target kernel.

	CSM	GSM	RQ	SE	SOURCE
MA52	-0.68	-7.78	-4.70	-8.61	-0.37
MA32	-0.96	-9.82	-18.20	-12.46	-0.2
PE+MA52	-3.63	-4.67	-9.74	-22.71	-0.62
PE*MA52	-2.77	-6.35	-15.59	-25.32	-0.96

The marginal likelihood of the data using the CSM and GSM kernels are -0.68, -7.78, respectively, compared to -0.37 for the MA52[1] that generated the data. For the RQ and SE kernels, the log marginal likelihoods are -4.70 and -8.61, respectively. Detailed results are presented in Table 1.

Figure 1 shows the learned correlation functions for CSM, GSM, SE, and RQ kernels[2], compared to the correlation function of generating MA52. Nearly all the trained kernels demonstrate a similar correlation trend to the target kernel, but CSM performs the best. This is likely because CSM is more flexible than the SE and RQ kernels, enabling a better approximation of the target kernel. On the other hand, while the target kernel is only 2 times differentiable, GSM is infinitely differentiable, which is why CSM outperforms GSM.

[1]We use MA52 to denote Matérn kernels with $\nu = \frac{5}{2}$. Similarly, MA32 refers to those with $\nu = \frac{3}{2}$.

[2]The horizontal axis denotes the Euclidean distance between two points, while the vertical axis represents the corresponding covariance distance.

1.2. Composite Kernels

Apart from single kernels, we also attempt to approximate finitely differentiable composite kernels in different analytical forms. We consider both the addition and multiplication of MA52 in **??** with $\ell_{MA} = 2.$ and the PE kernel in **??** with $\ell_{PE} = 2, \omega = 1$. In these cases, CSM still demonstrates the best approximation results and captures the periodicity.

2. Numerical Experiments

Apart from the four benchmarks in the original submission, we further validate our approach against 3 alternatives across different domains. The first baseline is Adaptive Kernel Selection (ADA) (Roman et al., 2019). Similar to automatic Bayesian optimization (ABO) (Malkomes & Garnett, 2018), this algorithm maintains a set of kernels and dynamically chooses the most appropriate kernel in each iteration. Apart from AutoML algorithm, we also consider other spectral kernels for Gaussian Process, including Spectral Delta kernel (SDK) (Vargas-Hernández & Gardner, 2021), and SINC kernel (SINC) (Tobar, 2019) .Under this setup, we consider a wider range of synthetic, simulated,and hyperparameter optimization problems with increasing dimensionality and complexity, as summarized in **??**. Results are depicted in Figure 3.

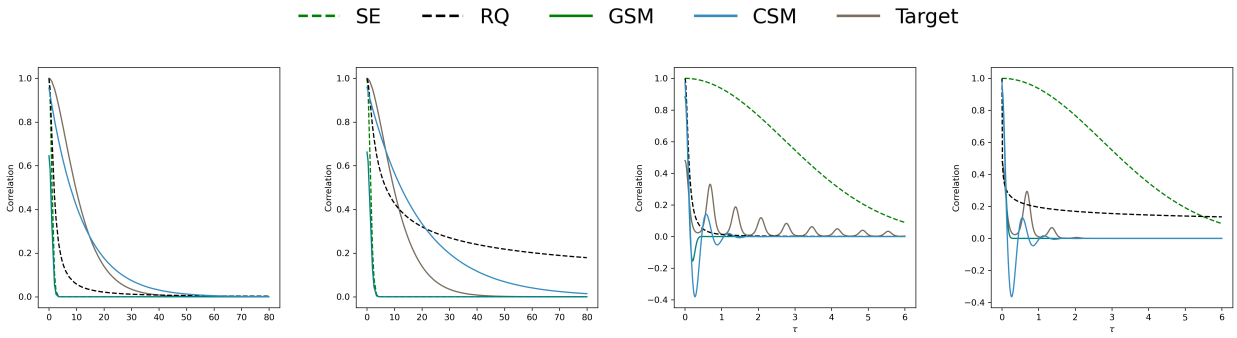
2.1. Synthetic Test Problems

2.2. Simulated Problems

Robot Pushing Detailed descriptions can be found in the submitted paper.

Portfolio Optimization In this test problem, our goal is to tune the hyperparameters of a trading strategy so as to maximize return under risk-aversion to random environmental conditions. The hyper-parameters to be optimized are the risk and trade aversion parameters, the holding cost multiplier, bid-ask spread and the borrow cost.

We use CVXPortfolio (Boyd et al., 2017) to simulate and optimize the evolution of a portfolio over a period of four years using open-source market data. The details of this simulator can be found in Sections 7.1-7.3 of Boyed et al. (2017). Since this simulator is indeed expensive-to-evaluate,



(a) MA52 (b) MA32 (c) PE+MA52 (d) PE*MA52

Figure 1. Learned correlation function in kernel approximation, with $\tau = x - x'$. The darker solid line denotes the kernel that generates sampling points. The closer the other curves are to the darker solid line, the better the corresponding training kernel captures the characteristics of the target kernel.

Table 2. Results for the performance metrics across 10 repetitions for different test functions and methods. '–' means the experiment hasn't finished yet.

OBJECTIVE	DIM	RBF	RQ	MA52	ABO	ADA	SDK	SINC	CSM7	GSM7	CSM+GSM
BRANIN	2	**-2.34**	-2.33	**-2.34**	-0.80	**-2.34**	1.22	2.64	-1.98	-2.29	**-2.34**
HARTMANN	3	-2.37	-1.18	-3.63	0.46	-3.05	-1.93	-0.28	-2.53	-4.55	-7.26
EXPONENTIAL	5	1.36	3.08	2.30	-0.71	–	–	–	1.44	-0.19	-2.25
HARTMANN	6	-1.42	0.74	-2.37	-2.43	-2.67	-0.90	-0.24	-2.59	-3.59	-4.00
ROSENBROCK	20	–	–	7.81	7.97	7.91	4.46	5.04	4.61	–	–
LEVY	30	–	–	–	3.59	3.55	–	–	–	–	–
ROBOT PUSHING	3	2.38	2.43	2.49	1.94	–	–	–	1.26	1.89	1.07
ROBOT PUSHING	4	2.20	2.07	1.99	2.07	–	–	–	1.11	1.22	0.76
PORTFOLIO	5	3.27	3.20	3.23	–	–	–	–	3.20	3.27	–
SVM	3	–	–	–	–	–	–	–	–	–	–
XGBOOST	9	–	–	–	–	–	–	–	–	–	–
LIGHTGBM	16	–	–	–	–	–	–	–	–	–	–

with each evaluation taking around 3 minutes, evaluating the performance of the various algorithms becomes prohibitively expensive. Therefore, following Cakmak et al. (2020) in our experiments we do not use the simulator directly. Instead, we build a surrogate function obtained as the mean function of a GP trained using evaluations of the actual simulator across 3, 000 points chosen according to a Sobol sampling design (Owen, 1998).

2.3. Hyperparameter Optimization

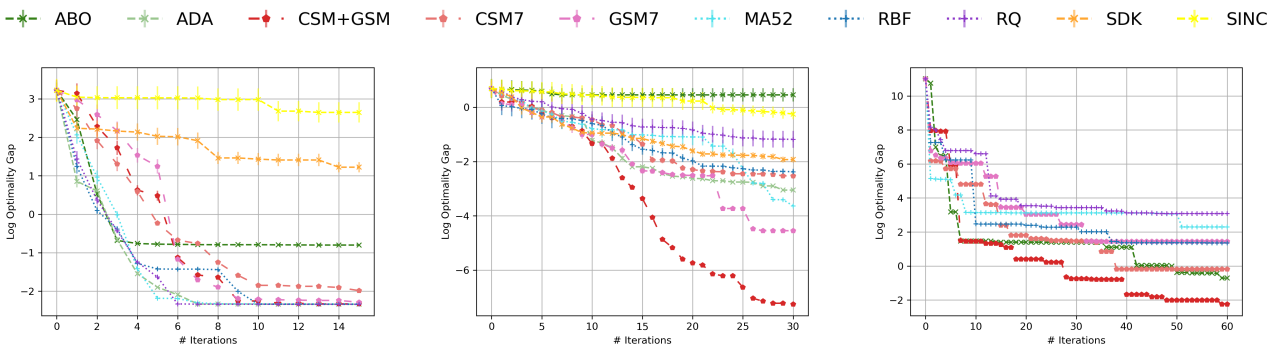
We further evaluate our proposed algorithm on hyperparameter tuning tasks for three representative models: SVM (3 hyperparameters), XGBoost (9 hyperparameters), and LightGBM (16 hyperparameters) using the MNIST dataset (LeCun & Cortes, 2010). All experiments follow a standardized protocol with 5-fold cross-validation.

2.4. Alternative Performance Metric: Mean Average Regret

Apart from optimal value and optimality gap, we also plotted the mean average regret against iterations. The results demonstrate consistent advantages of the proposed CSM7 and GSM7 kernels over conventional approaches (MA52, RBF, RQ) across multiple test functions. Traditional kernels demonstrate slower convergence and significant fluctuations, particularly in complex problem settings. Moreover, the performance gap widens as problem complexity increases, suggesting better scalability.

2.5. Alternative Acquisition Functions

To address potential bias from acquisition function selection, we repeated experiments using Probability of Improvement (PI) and Upper Confidence Bound (UCB) alongside Expected Improvement (EI). Results show the our approach outperformed baselines regardless of the choice acquisition function. The only exception is Hartmann_6d with PI as demonstrated in Figure 4-(b2).



(a) Branin_2d (b) Hartmann_3d (c) Exponential_5d

(d) Hartmann_6d (e) Rosenbrock_20d (f) Levy_30d

(g) RobotPushing_3d (h) RobotPushing_4d (e) Portfolio_5d

(f) SVM_3d (e) XGBoost_9d (f) LightGMB_16d

Figure 2. Results for the average gap performance for different test functions and methods. a-f: log optimality gap for synthetic test functions; g-h: log optimality gap for real-world problems. The plots are plotted against the number of iterations.

References

Boyd, S., Busseti, E., Diamond, S., Kahn, R. N., Koh, K., Nystrup, P., and Speth, J. 2017. doi: 10.1561/2400000023.

Cakmak, S., Astudillo, R., Frazier, P., and Zhou, E. Bayesian optimization of risk measures. In *Proceedings*

Table 3. Test functions used in our experiments. The analytic form of these functions as well as the global minima are available online (Surjanovic & Bingham, 2025).

OBJECTIVE	DIM	ITERS	INPUTS
BRANIN	2	15	$x \in [-3, 3]^2$
HARTMANN	3	30	$x \in [0, 1]^3$
EXPONENTIAL	5	60	$x \in [-5.12, 5.12]^5$
HARTMANN	6	80	$x \in [0, 1]^6$
ROSENBROCK	20	200	$x \in [-2.048, 2.048]^{20}$
LEVY	30	300	$x \in [-5, 5]^{30}$
ROBOT PUSHING	3	100	$x_0, x_1 \in [-5, 5]^2$, $x_2 \in [1, 30]$
ROBOT PUSHING	4	150	$x_0, x_1 \in [-5, 5]^2$, $x_2 \in [1, 30]$, $x_3 \in [0, 2\pi]$
PORTFOLIO	5	200	$x_0 \in [0.1, 1000]$ (RISK PARAMETER), $x_1 \in [5.5, 8.0]$ (TRADE AVERSION PARAMETER), $x_2 \in [0.1, 100]$ (HOLDING COST MULTIPLIER), $x_3 \in [10^{-4}, 10^{-2}]$ (BID-ASK SPREAD), $x_4 \in [10^{-4}, 10^{-3}]$ (BORROW COST)
SVM	3	100	$x_0 \in [0, 1]$ (C), $x_1 \in [0, 1]$ (GAMMA), $x_2 \in \{$'LINEAR','RBF','POLY'$\}$ (KERNEL),
XGBOOST	9	150	$x_0 \in [-3, 0]_{\log_{10}}$ (LEARNING_RATE), $x_1 \in [10, 210]_{\mathbb{Z}}$ (N_ESTIMATORS), $x_2 \in [3, 8]_{\mathbb{Z}}$ (MAX_DEPTH), $x_3 \in [2, 20]_{\mathbb{Z}}$ (MIN_CHILD_WEIGHT), $x_4 \in [0.5, 1.0]$ (SUBSAMPLE), $x_5 \in [0.5, 1.0]$ (COLSAMPLE_BYTREE), $x_6 \in [0, 5]$ (GAMMA), $x_7 \in [-3, 0]_{\log_{10}}$ (REG_ALPHA), $x_8 \in [-3, 0]_{\log_{10}}$ (REG_LAMBDA)
LIGHTGBM	16	200	$x_0 \in [-4, -1]_{\log_{10}}$ (LEARNING_RATE), $x_1 \in [10, 210]_{\mathbb{Z}}$ (NUM_LEAVES), $x_2 \in [3, 8]_{\mathbb{Z}}$ (MAX_DEPTH), $x_3 \in [2, 20]_{\mathbb{Z}}$ (MIN_CHILD_SAMPLES), $x_4 \in [0.5, 1.0]$ (SUBSAMPLE), $x_5 \in [0.5, 1.0]$ (COLSAMPLE_BYTREE), $x_6 \in [-3, -1]_{\log_{10}}$ (REG_ALPHA), $x_7 \in [-3, -1]_{\log_{10}}$ (REG_LAMBDA), $x_8 \in [0, 0.2]$ (MIN_SPLIT_GAIN), $x_9 \in [0.5, 0.95]$ (FEATURE_FRACTION), $x_{10} \in [0, 5]_{\mathbb{Z}}$ (BAGGING_FREQ), $x_{11} \in [0.5, 0.95]$ (BAGGING_FRACTION), $x_{12} \in [-3, -1]_{\log_{10}}$ (LAMBDA_L1), $x_{13} \in [-3, -1]_{\log_{10}}$ (LAMBDA_L2), $x_{14} \in [10, 50]_{\mathbb{Z}}$ (MIN_DATA_IN_LEAF), $x_{15} \in [0, 0.5]$ (PATH_SMOOTH)

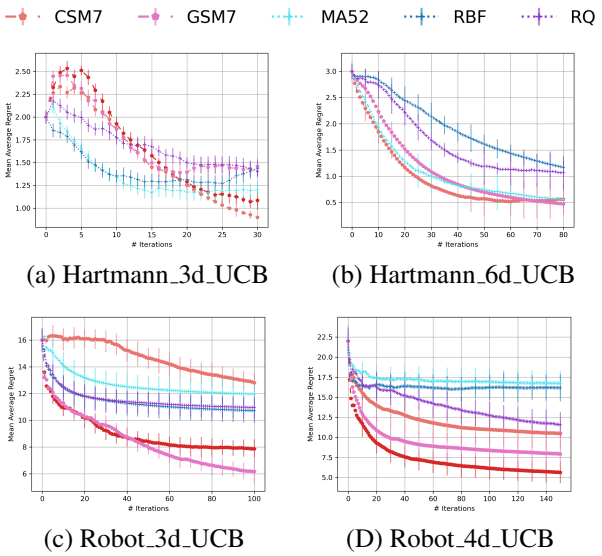
of the 34th International Conference on Neural Information Processing Systems, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

LeCun, Y. and Cortes, C. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/, 2010. Accessed: 2023-08-20.

Malkomes, G. and Garnett, R. Automating bayesian optimization with bayesian optimization. In *Proceedings of the 32th International Conference on Neural Information Processing Systems*, NIPS'18, pp. 5988–5997, Red Hook, NY, USA, 2018. Curran Associates Inc.

Owen, A. B. Scrambling sobol' and niederreiter–xing points. *Journal of Complexity*, 14(4):466–489, 1998. ISSN 0885-064X. doi: https://doi.org/10.1006/jcom.1998.0487. URL https://www.sciencedirect.com/science/article/pii/S0885064X98904873.

Roman, I., Santana, R., Mendiburu, A., and Lozano, J. A. An experimental study in adaptive kernel selection for



(a) Hartmann_3d_UCB (b) Hartmann_6d_UCB

(c) Robot_3d_UCB (D) Robot_4d_UCB

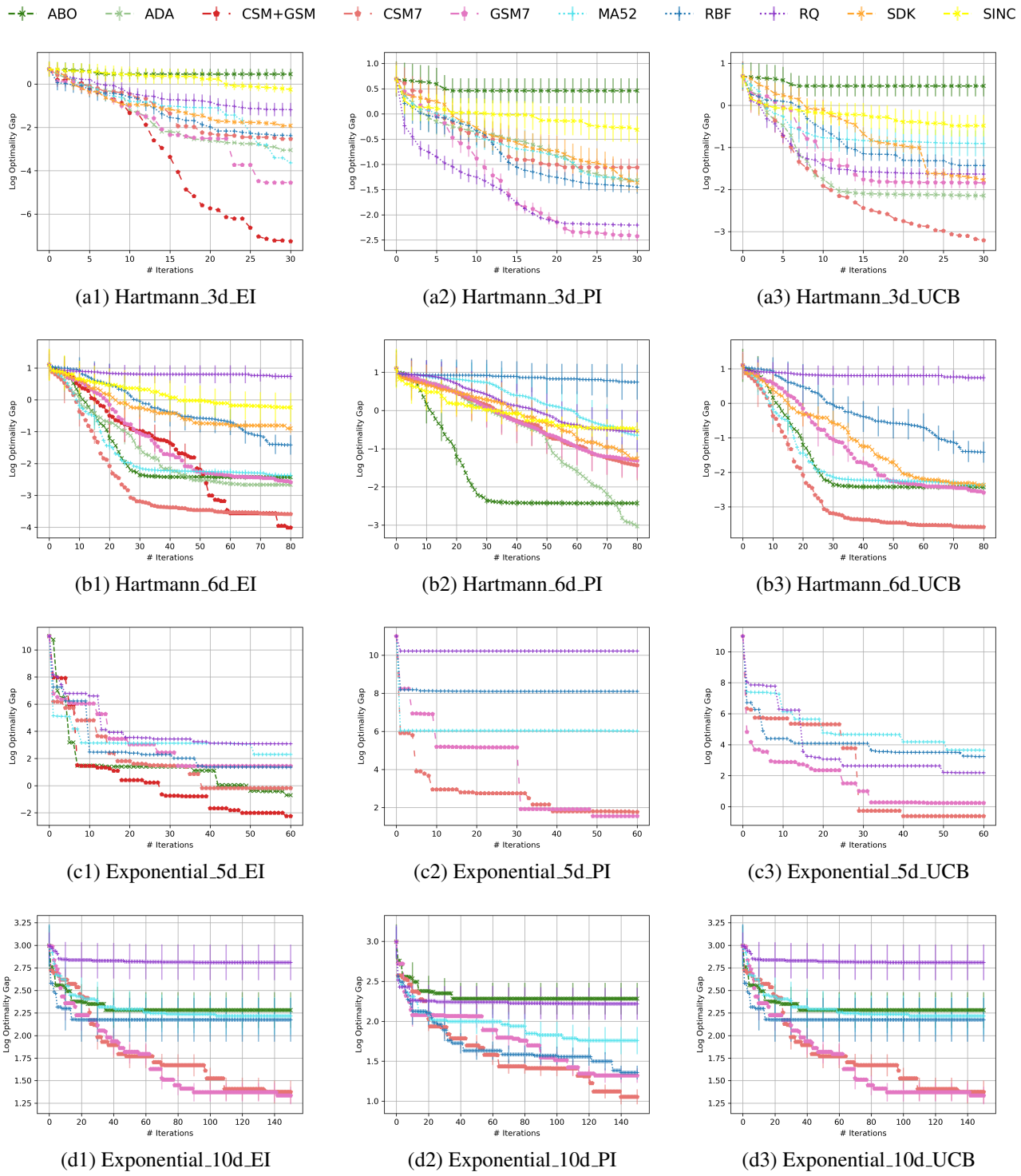
Figure 3. Results for the average mean regret over iterations.

bayesian optimization. *IEEE Access*, 7:184294–184302, 2019. doi: 10.1109/ACCESS.2019.2960498.

Surjanovic, S. and Bingham, D. Virtual library of simulation experiments: Test functions and datasets. Retrieved January 19, 2025, from http://www.sfu.ca/~ssurjano, 2025.

Tobar, F. *Band-limited gaussian processes: the sinc kernel*. Curran Associates Inc., Red Hook, NY, USA, 2019.

Vargas-Hernández, R. and Gardner, J. Gaussian processes with spectral delta kernel for higher accurate potential energy surfaces for large molecules, 09 2021.



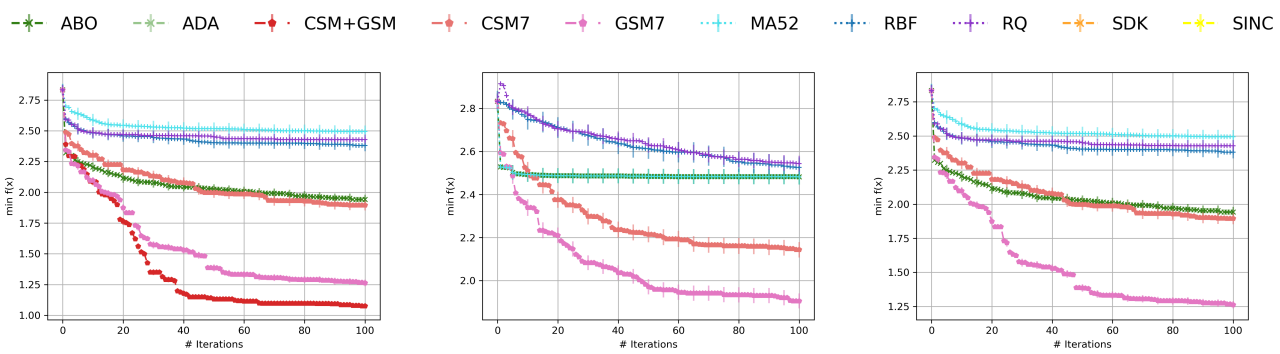
(a1) Hartmann_3d_EI (a2) Hartmann_3d_PI (a3) Hartmann_3d_UCB

(b1) Hartmann_6d_EI (b2) Hartmann_6d_PI (b3) Hartmann_6d_UCB

(c1) Exponential_5d_EI (c2) Exponential_5d_PI (c3) Exponential_5d_UCB

(d1) Exponential_10d_EI (d2) Exponential_10d_PI (d3) Exponential_10d_UCB

Figure 4. Results for the average performance using different acquisition functions. The plots are plotted against the number of iterations.



(e1) Robot_3d_EI (e2) Robot_3d_PI (e3) Robot_3d_UCB

(e1) Robot_4d_EI (e2) Robot_4d_PI (e3) Robot_4d_UCB

Figure 5. Results for the average performance using different acquisition functions. The plots are plotted against the number of iterations.