

## Supplementary materials for *Discrete and mixed-variable experimental design with surrogate-based approach*

Mengjia Zhu, Austin Mroz, Lingfeng Gui, Kim Jelfs, Alberto Bemporad, Ehecatl Antonio del Río Chanona\*, and Ye Seol Lee\*

\* *lauren.lee@ucl.ac.uk* , *a.del-rio-chanona@imperial.ac.uk*

September 29, 2024

### 1 Comparison study with Gryffin

*Gryffin* [1]: This algorithm is a Bayesian optimization (BO) framework designed specifically for categorical variables, leveraging expert knowledge in the form of physicochemical descriptors. *Gryffin* augments traditional Bayesian optimization by using kernel density estimation (KDE) directly on categorical spaces and incorporates smooth approximations to categorical distributions through the use of concrete distributions. In cases where domain knowledge is available, *Gryffin* utilizes descriptors to redefine the metric on the categorical space, enabling more efficient search by highlighting similarities between categorical options.

Like other Bayesian optimization approaches, *Gryffin* operates in two stages: the initial sampling stage, where the algorithm randomly samples from the search space, and the active-learning stage, where the acquisition function (influenced by the domain knowledge) guides the selection of the next points to evaluate. Furthermore, *Gryffin* can dynamically refine the descriptors during the optimization process to improve the quality of the search, allowing the algorithm not only to optimize but also to enhance the domain knowledge used. This dual capacity makes *Gryffin* particularly suitable for optimization problems in chemistry and materials science, where categorical variables are common and can be described by meaningful descriptors. Similar to other BO methods, it cannot handle discrete/mixed-variable constraints explicitly.

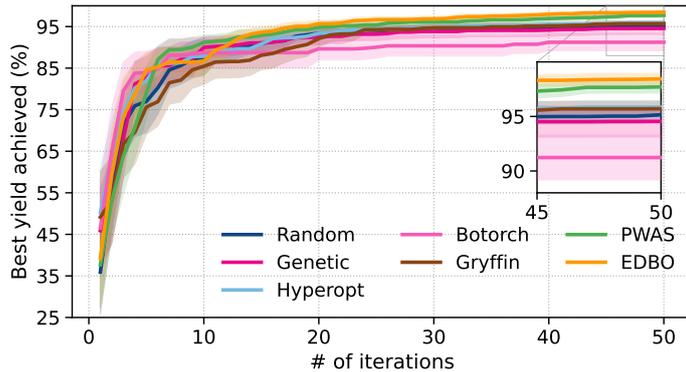
We use *Gryffin* to solve the first two case studies. The relevant code and results are included in the following folder of the code repository: [https://github.com/mjzhu-p/olympus/tree/pwas\\_comp/case\\_studies/case\\_study\\_pwas/z\\_comparisonStudy](https://github.com/mjzhu-p/olympus/tree/pwas_comp/case_studies/case_study_pwas/z_comparisonStudy)

#### 1.1 Suzuki-Miyaura cross-coupling

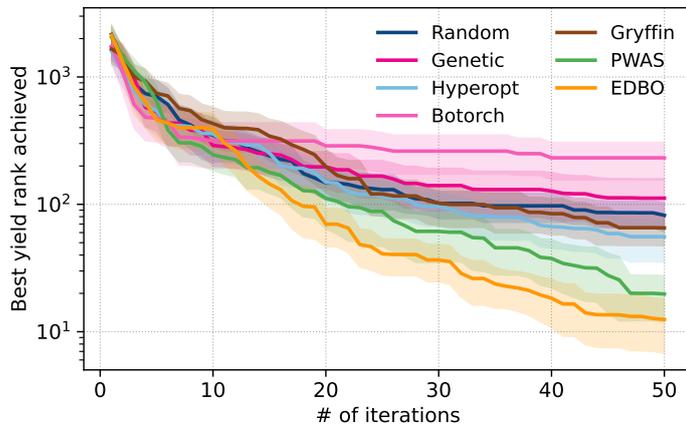
As shown in Figs. 1 and 2, the performance of *Gryffin* is similar to the performance of *Hyperopt* [2] but worse than *EDBO* [3] and *PWAS*.

#### 1.2 Crossed barrel

As for the crossed-barrel problem, the performance of *Gryffin* is similar to *random search* and worse than the rest of methods compared (see Fig. 3). Additionally, the computational time required by *Gryffin* is relatively high compared to *Genetic*, *Hyperopt*, and *PWAS* (see Table 1).



(a)



(b)

Figure 1: A comparison of the performance of *PWAS* and the benchmark methods on Suzuki-Miyaura cross-coupling reaction optimization. For each method, the solid line represents the mean value, and the filled area comprises the 95% confidence interval, *i.e.*, mean  $\pm$  1.96 std. (a) Best yield achieved (%) so far at different iterations, and (b) Best yield rank achieved so far at different iterations.

Table 1: CPU time (seconds) required by different methods for one run of the optimization for the crossed barrel design. Statistics were obtained from 30 random runs.

	Random	Genetic	Hyperopt	BoTorch	Gryffin	PWAS	EDBO_1	EDBO_2	EDBO_3
Average	1.85	1.77	2.80	398.68	148.38	35.36	272.54	227.54	212.92
std	0.44	0.35	0.71	260.71	105.89	2.00	67.61	2.52	20.38

## 2 Illustrative predictability comparisons

We focus on problems with mixed-variables and discontinuous functions, for which we argue that, when the number of training samples are limited, piecewise affine (PWA) function can be an efficient surrogate model. Here, we selected 1-D and 2-D examples for better visualization. We compare PWA surrogate model with Gaussian Process models, for which three types of kernels are considered, namely, radial basis function (RBF), Matern, and linear kernels. For non-biased comparisons, training samples are evenly selected from their domain. We consider the following two synthetic functions:

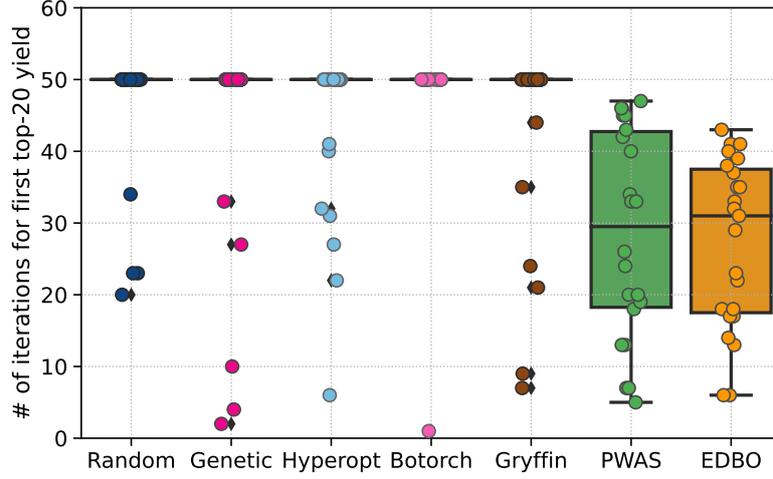


Figure 2: Number of iterations each method takes in each run to obtain the first top-20 ranked yield. The results for 30 repetitions are summarized in the boxplot. Each dot represents one run of the repetitions. The diamond-shaped points are the ones classified as outliers by the boxplot.

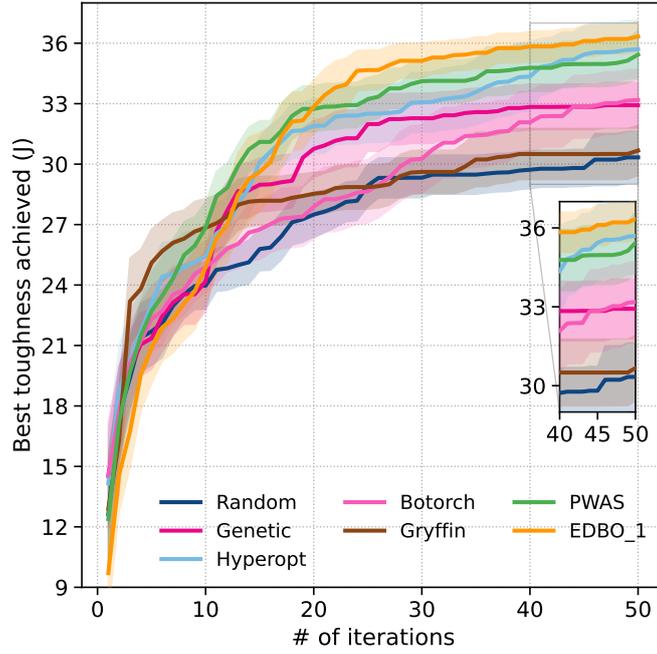


Figure 3: Best toughness achieved so far at different iterations for the designed structure at different iterations for crossed barrel design. Results are summarized over 30 repetitions. For each method, the solid line represents the mean value, and the filled area comprises the 95% confidence interval, *i.e.*,  $\text{mean} \pm 1.96 \text{ std}$ .

Function 1: 1-D, discontinuous function, continuous variables only

$$f(x_1, x_2) = \begin{cases} x^2 - 3 & -5 \leq x \leq 1 \\ 0.5x + 31 & 1 \leq x \leq 4 \\ 0.2x^3 & 4 \leq x \leq 10 \end{cases} \quad (1)$$

Function 2: 2-D, mixed-variable (1 continuous, 1 categorical), for different classes of the categorical variable ( $x_2$ ), the function behaves differently at the same continuous variable ( $x_1$ )

$$f(x_1, x_2) = \begin{cases} -(x_1 + .5)^2 + 2x_1 + 1 & x_2 = 0 \\ -x_1 - 100 & x_2 = 1 \\ (x_1 - 1)^3 & x_2 = 2 \\ 12x + 50 & x_2 = 3 \end{cases} \quad (2)$$

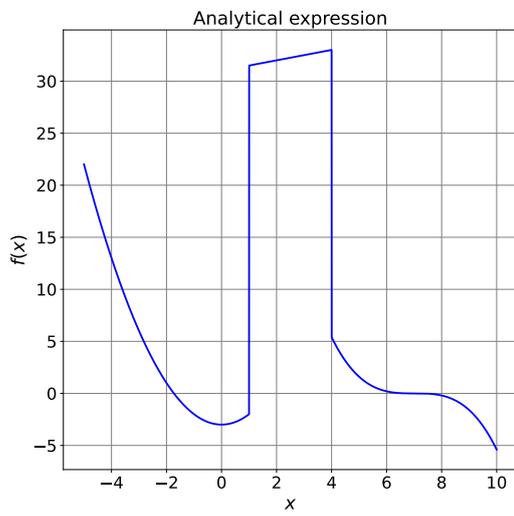
$-5 \leq x_1 \leq 5, x_2 \in \{0, 1, 2, 3\},$

The trend of both functions are depicted in Fig. 4 based on the analytic formula.

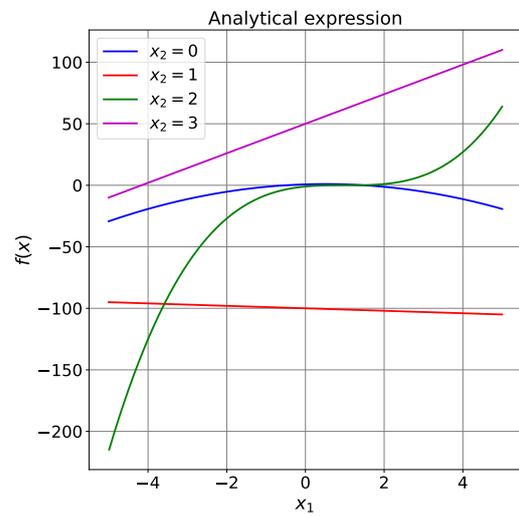
- Jupyter notebook: [https://github.com/MolChemML/ExpDesign/blob/main/surrogate\\_model\\_comparison/pred\\_comp\\_illustrative\\_example.ipynb](https://github.com/MolChemML/ExpDesign/blob/main/surrogate_model_comparison/pred_comp_illustrative_example.ipynb)
  - Figs. 5- 16 shows the surrogate fitting for Function 1 and 2 after different numbers of training samples.
  - As can be seen in Figs. 5- 16, the results highlight the benefits of using a piecewise affine (PWA) function as a surrogate model, particularly for problems with mixed-variables and discontinuous function behaviors, and when the number of training samples is limited.
- Notes:
  - We want to note that PWAS is proposed mainly as an optimization tool rather than a regression tool. It is used to adaptively suggest the next promising sample to test, and it is particularly useful for problems involves mixed-variable linear constraints. PWA surrogate is selected since it has a MILP reformulation, which makes it possible to incorporate constraints directly when solve the optimization problem of the acquisition function (formed by the surrogate and the exploration function) to propose feasible samples.
  - Also, due to the natural of PWA functions, it can handle problems with sharp transitions well, which can often occur in problems within chemistry domain.

## References

- [1] F. Häse, M. Aldeghi, R. J. Hickman, L. M. Roch, and A. Aspuru-Guzik, “Gryffin: An algorithm for bayesian optimization of categorical variables informed by expert knowledge,” *Applied Physics Reviews*, vol. 8, no. 3, p. 031 406, 2021.
- [2] J. Bergstra, D. Yamins, and D. Cox, “Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures,” in *International conference on machine learning*, PMLR, 2013, pp. 115–123.
- [3] B. J. Shields, J. Stevens, J. Li, *et al.*, “Bayesian reaction optimization as a tool for chemical synthesis,” *Nature*, vol. 590, no. 7844, pp. 89–96, 2021.



(a)



(b)

Figure 4: (a) Function 1: 1-D, discontinuous function with continuous variable, (b) Function 2: 2-D, mixed-variable (1 continuous, 1 categorical), for different classes of the categorical variable ( $x_2$ ), the function behaves differently at the same continuous variable ( $x_1$ )

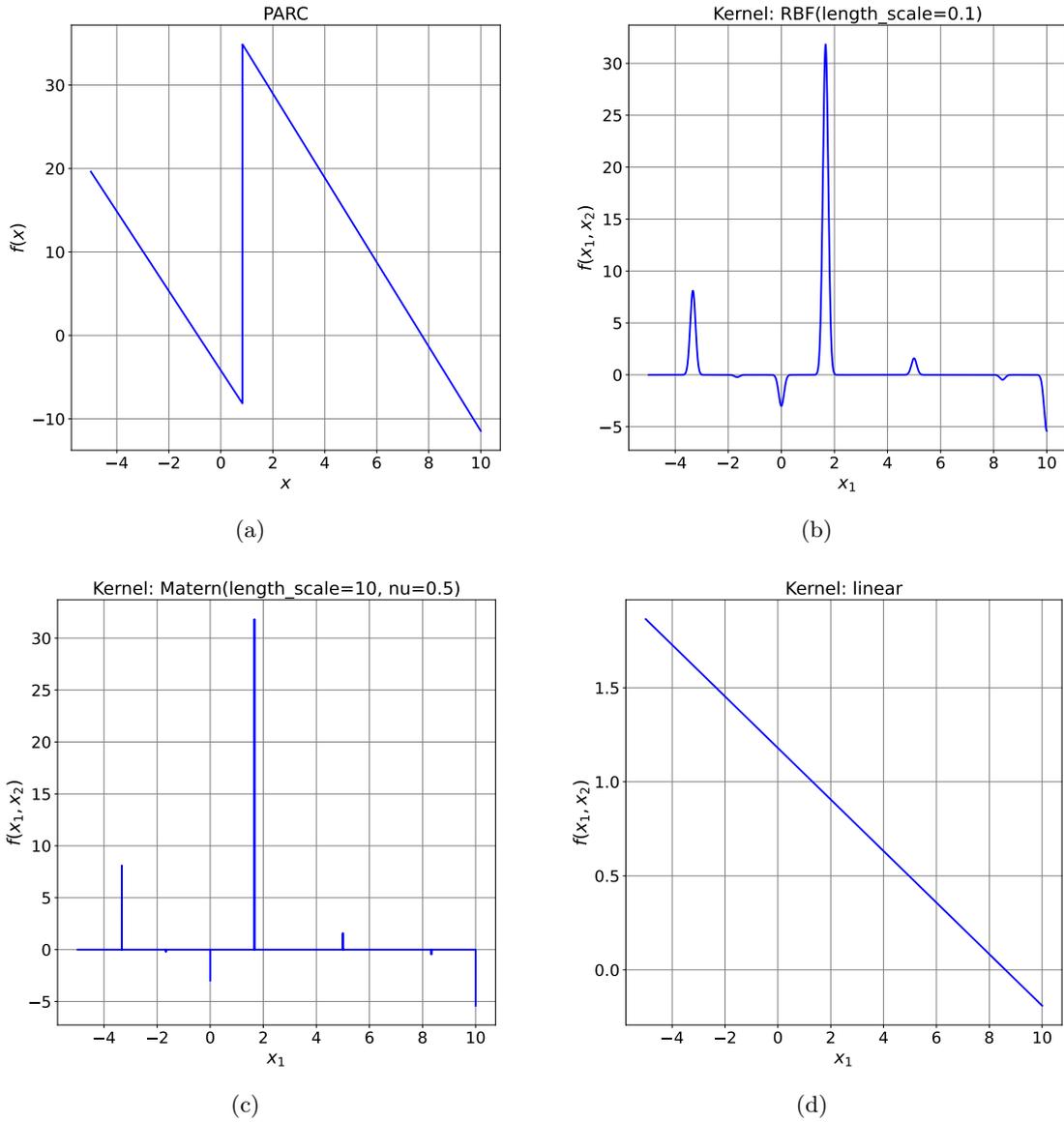
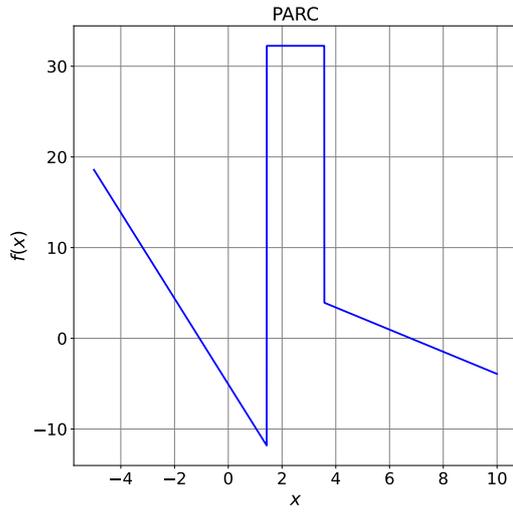
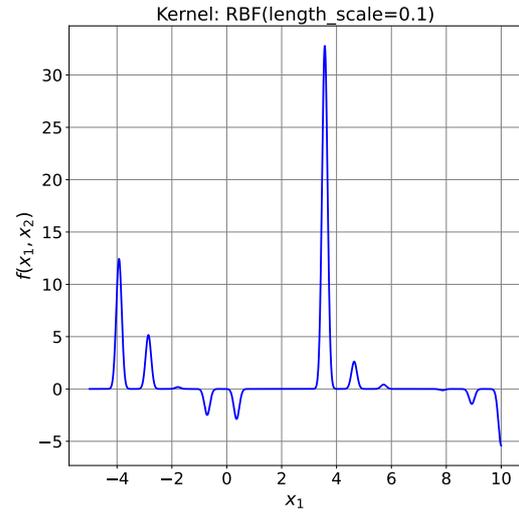


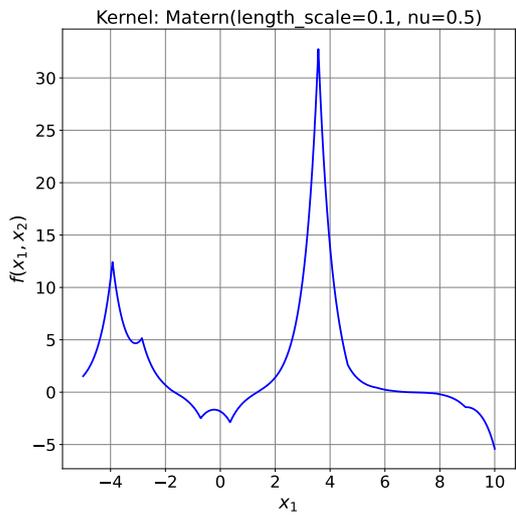
Figure 5: Surrogate model of Function 1 fitted with 10 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear



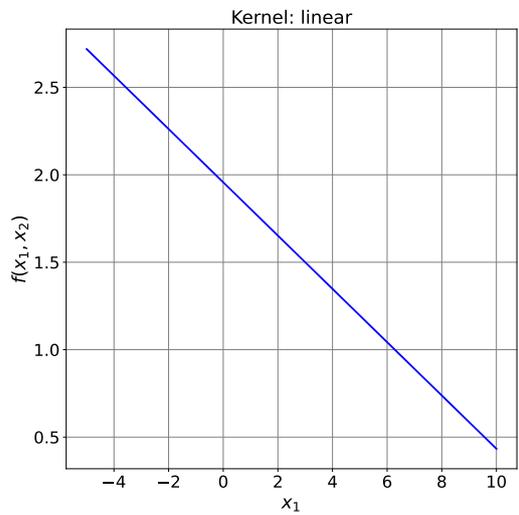
(a)



(b)

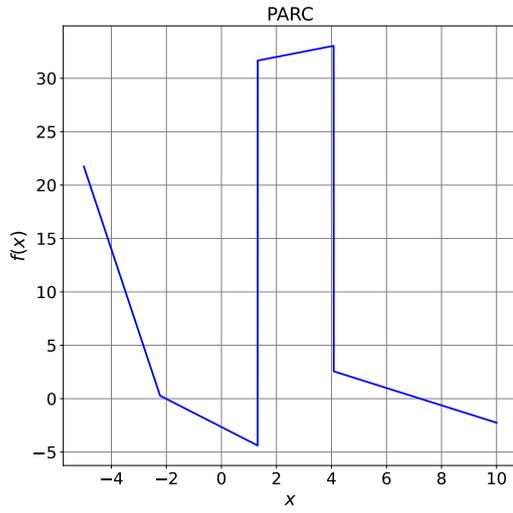


(c)

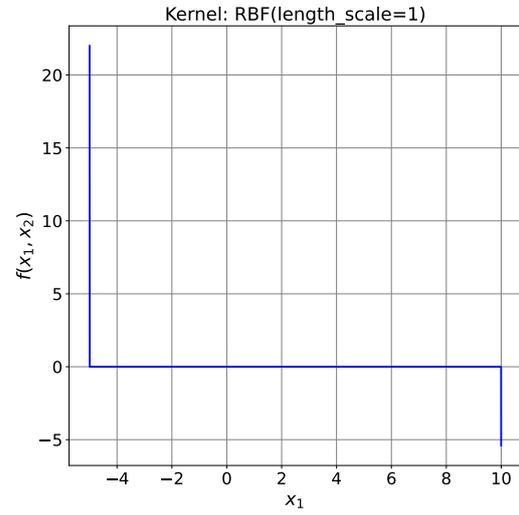


(d)

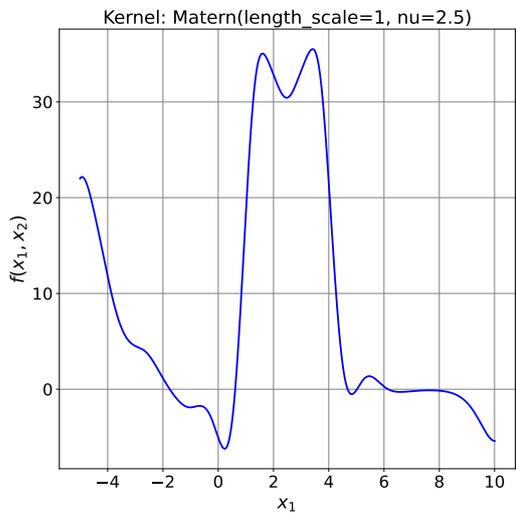
Figure 6: Surrogate model of Function 1 fitted with 15 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear



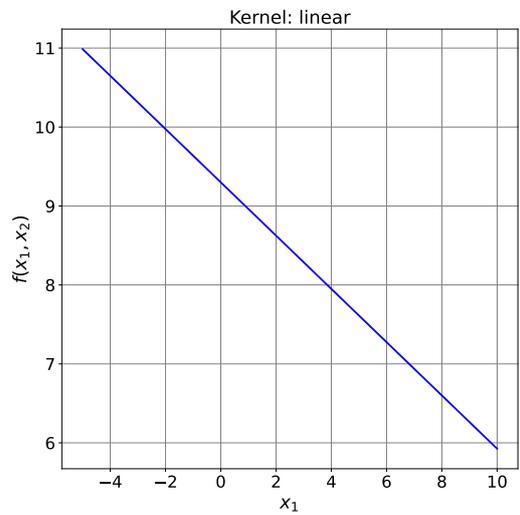
(a)



(b)



(c)



(d)

Figure 7: Surrogate model of Function 1 fitted with 20 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear

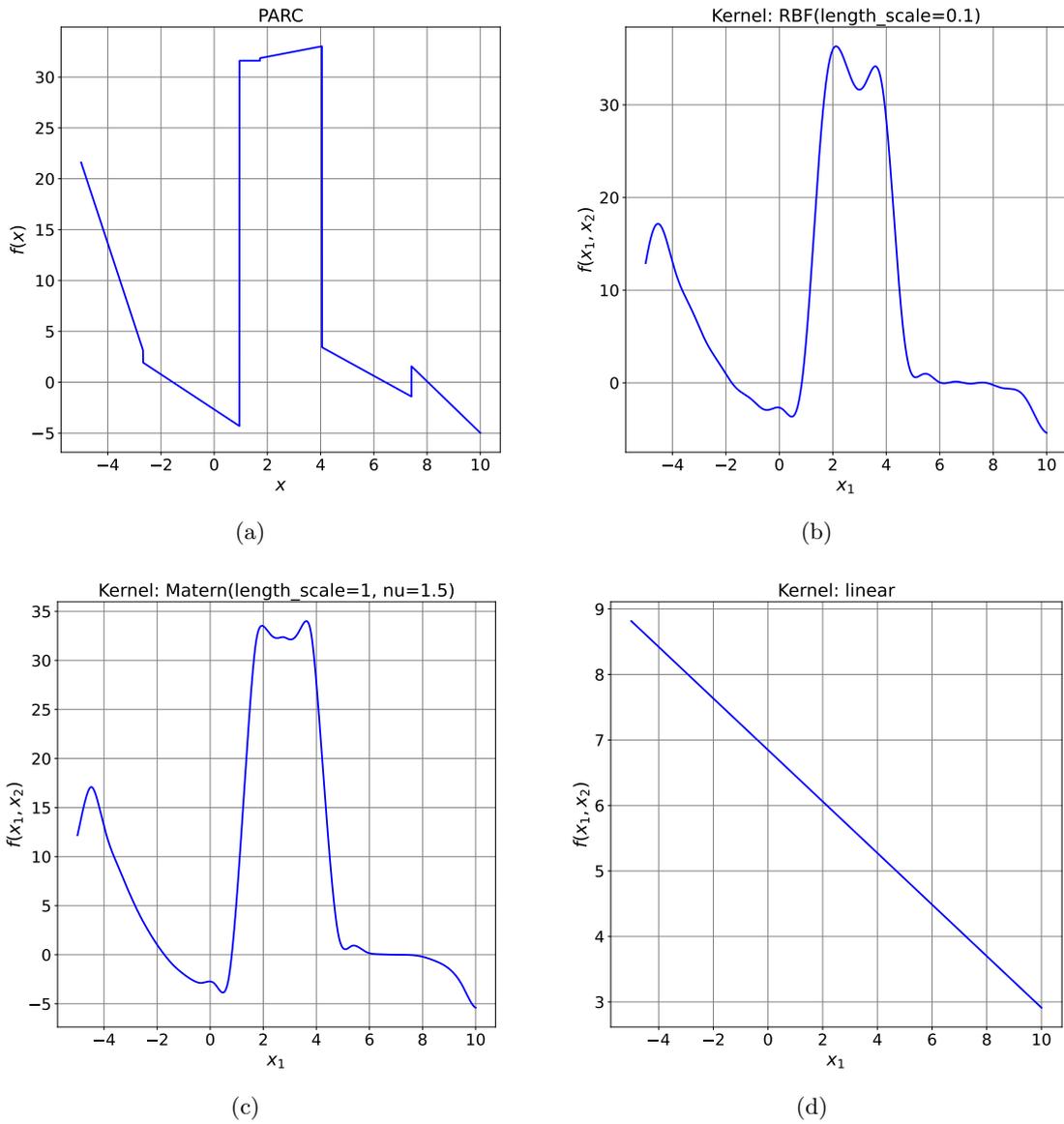
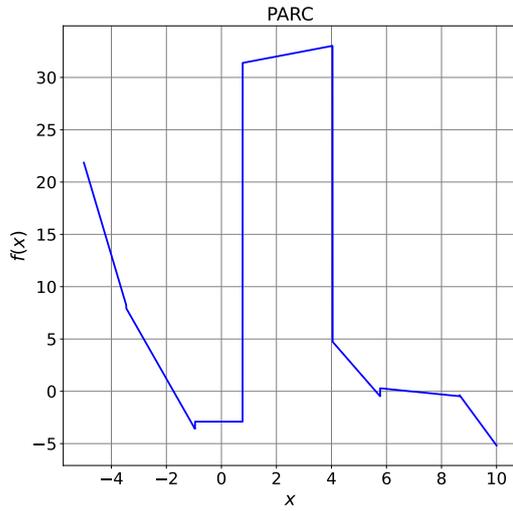
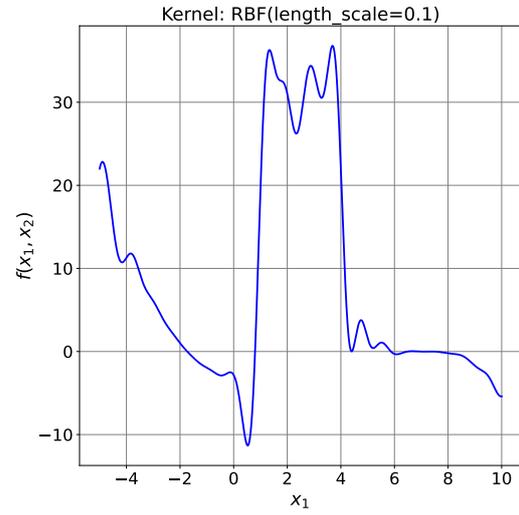


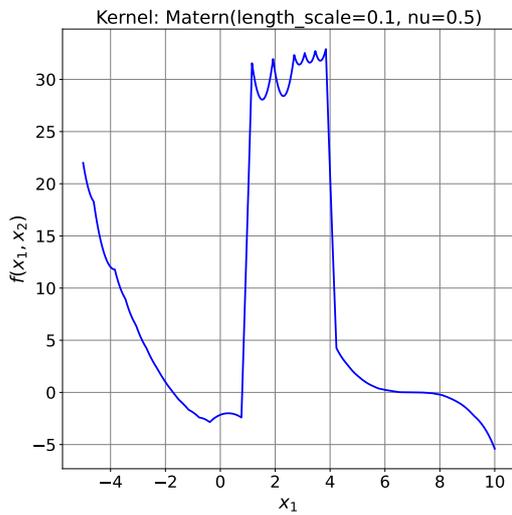
Figure 8: Surrogate model of Function 1 of Function 1 fitted with 30 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear



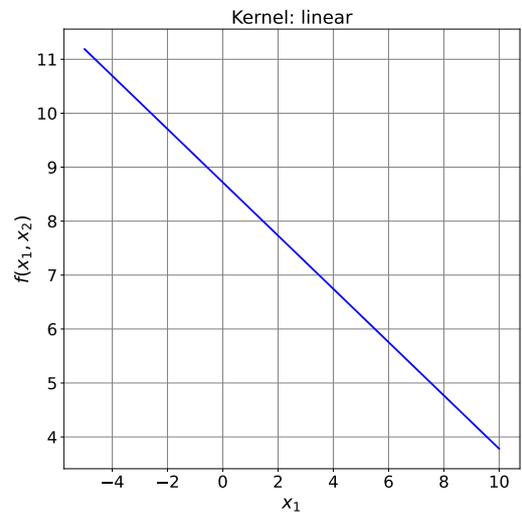
(a)



(b)

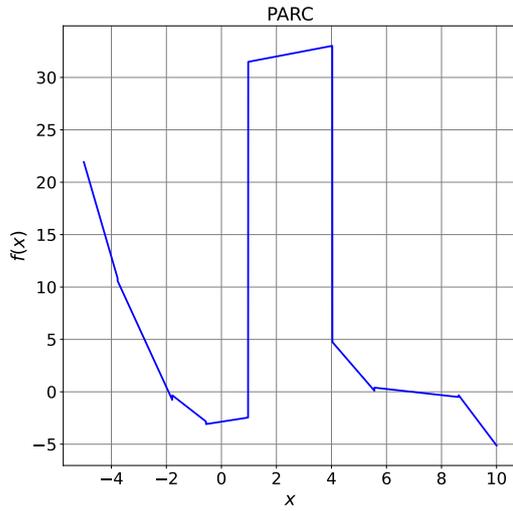


(c)

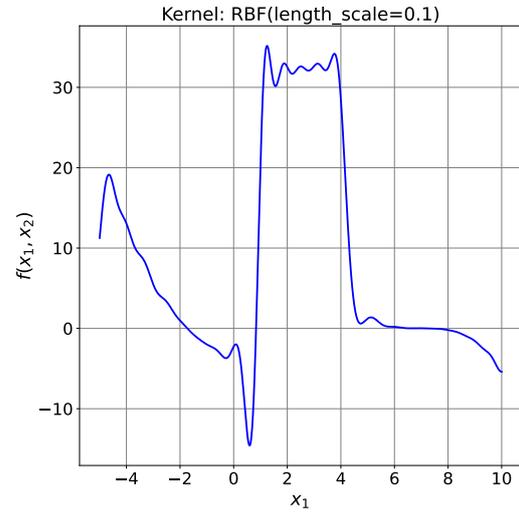


(d)

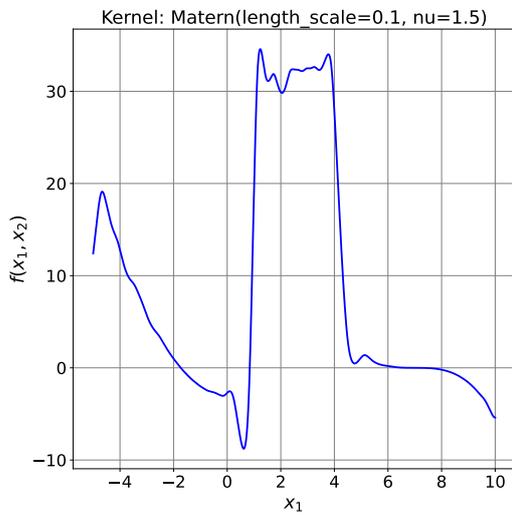
Figure 9: Surrogate model of Function 1 fitted with 40 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear



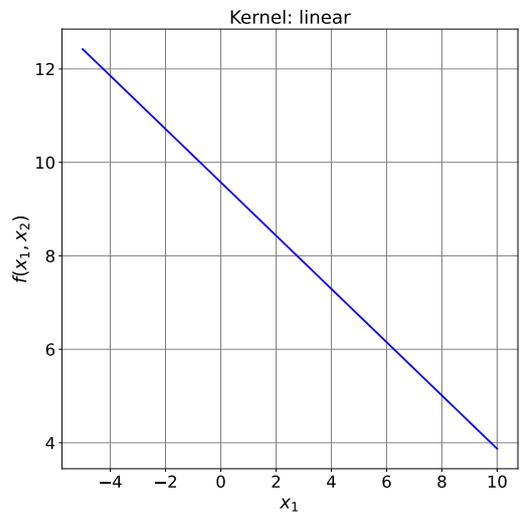
(a)



(b)



(c)



(d)

Figure 10: Surrogate model of Function 1 fitted with 50 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear

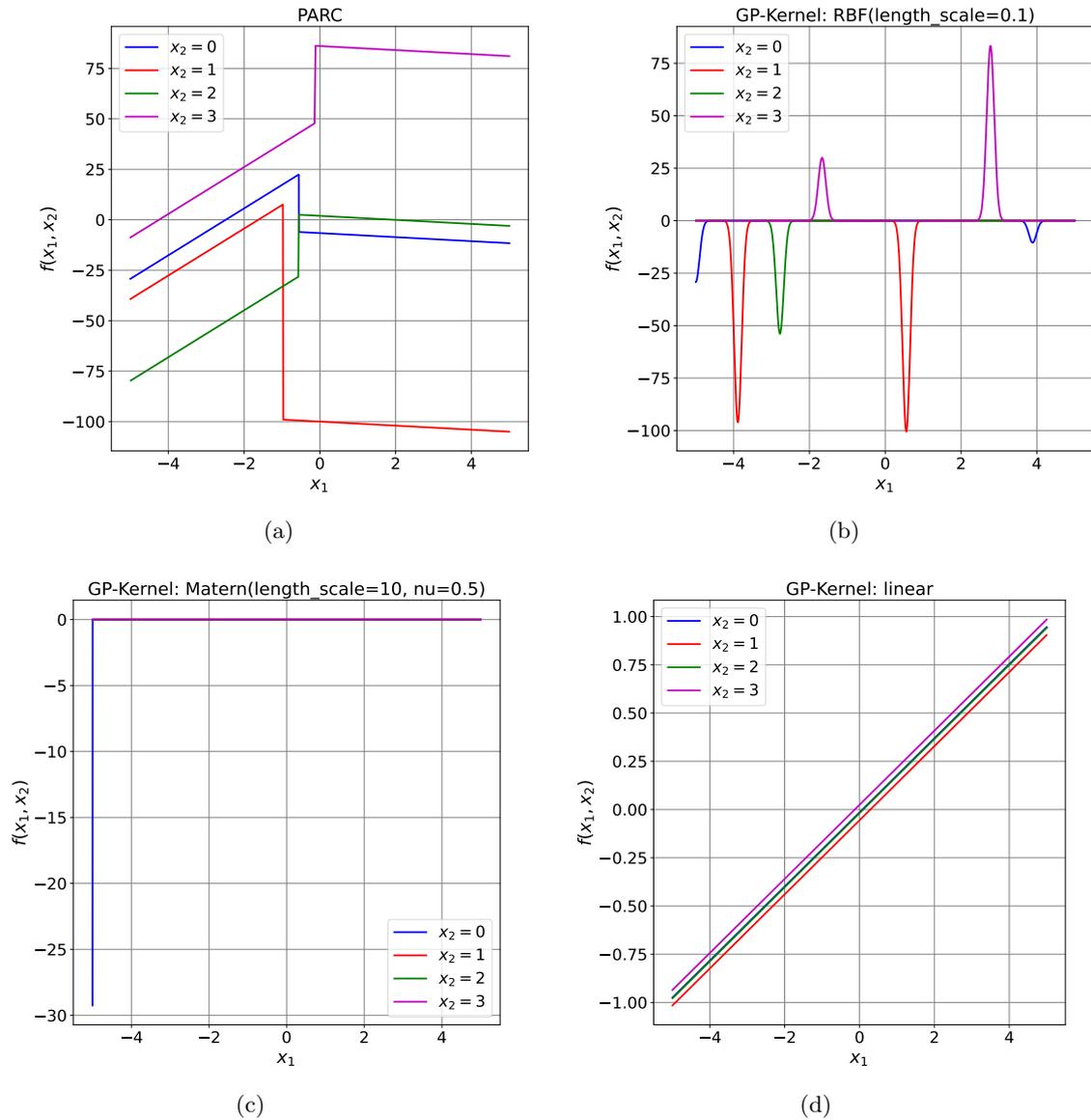


Figure 11: Surrogate model of Function 2 fitted with 10 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear

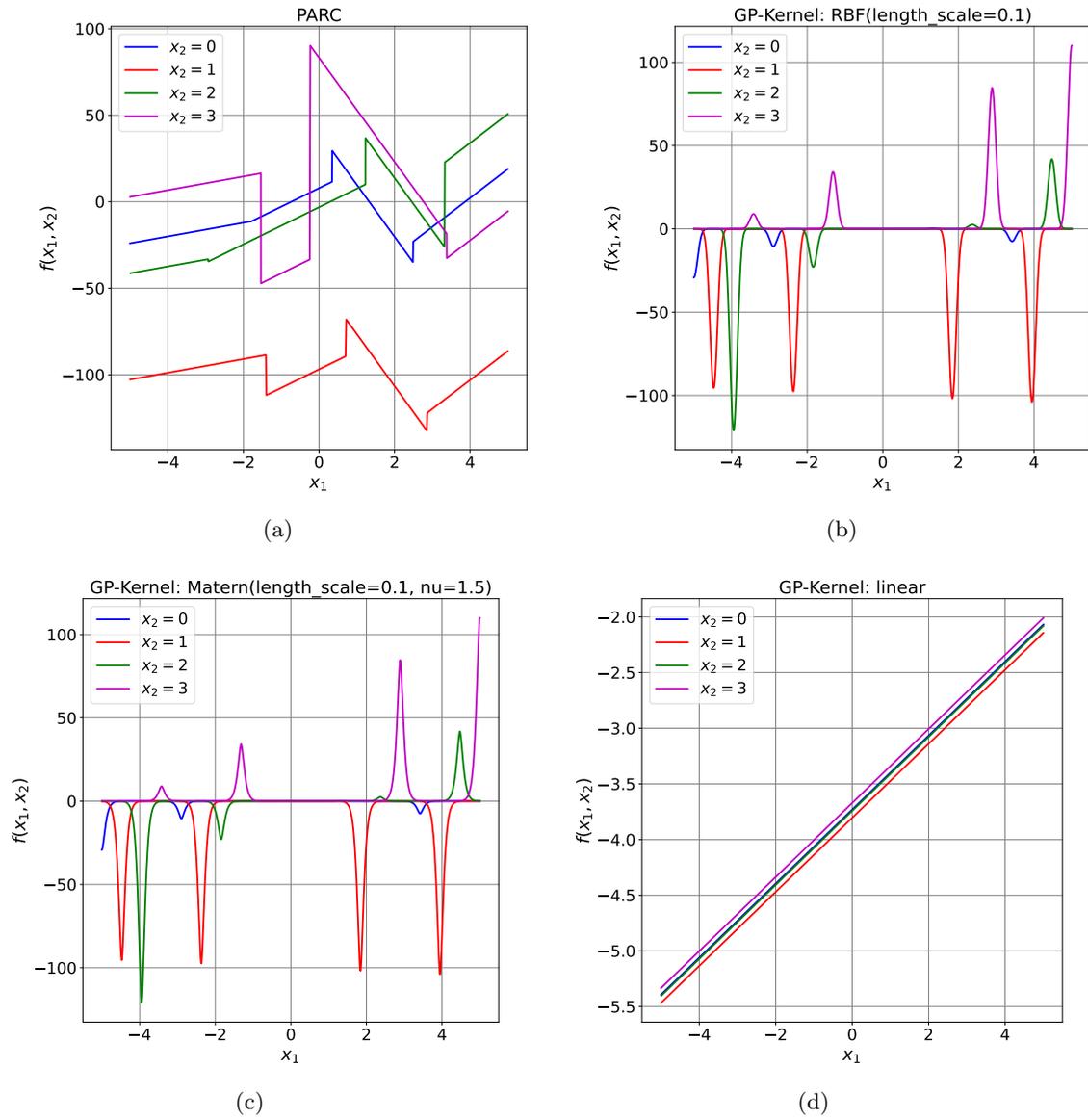


Figure 12: Surrogate model of Function 2 fitted with 20 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear

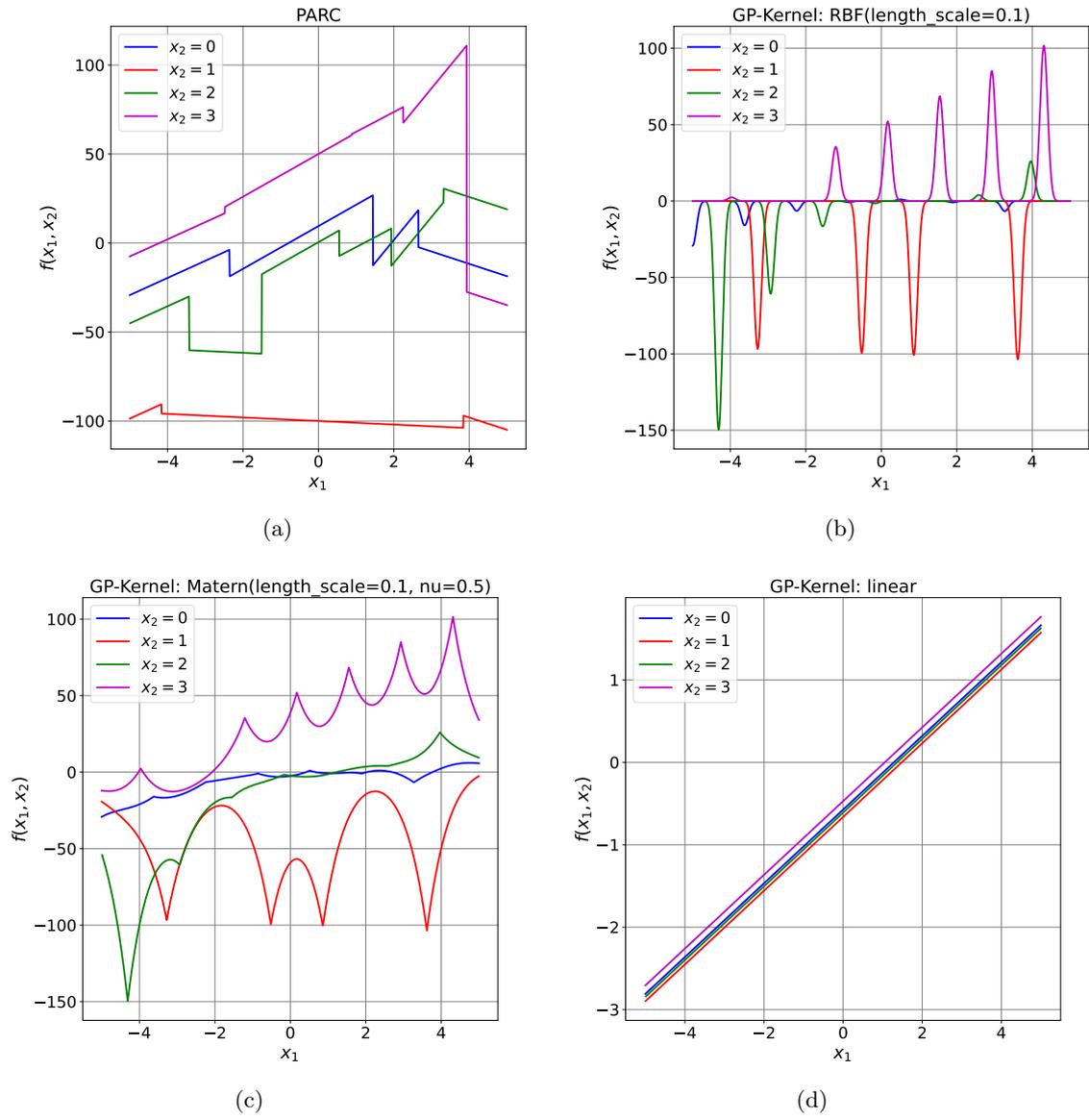


Figure 13: Surrogate model of Function 2 fitted with 30 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear

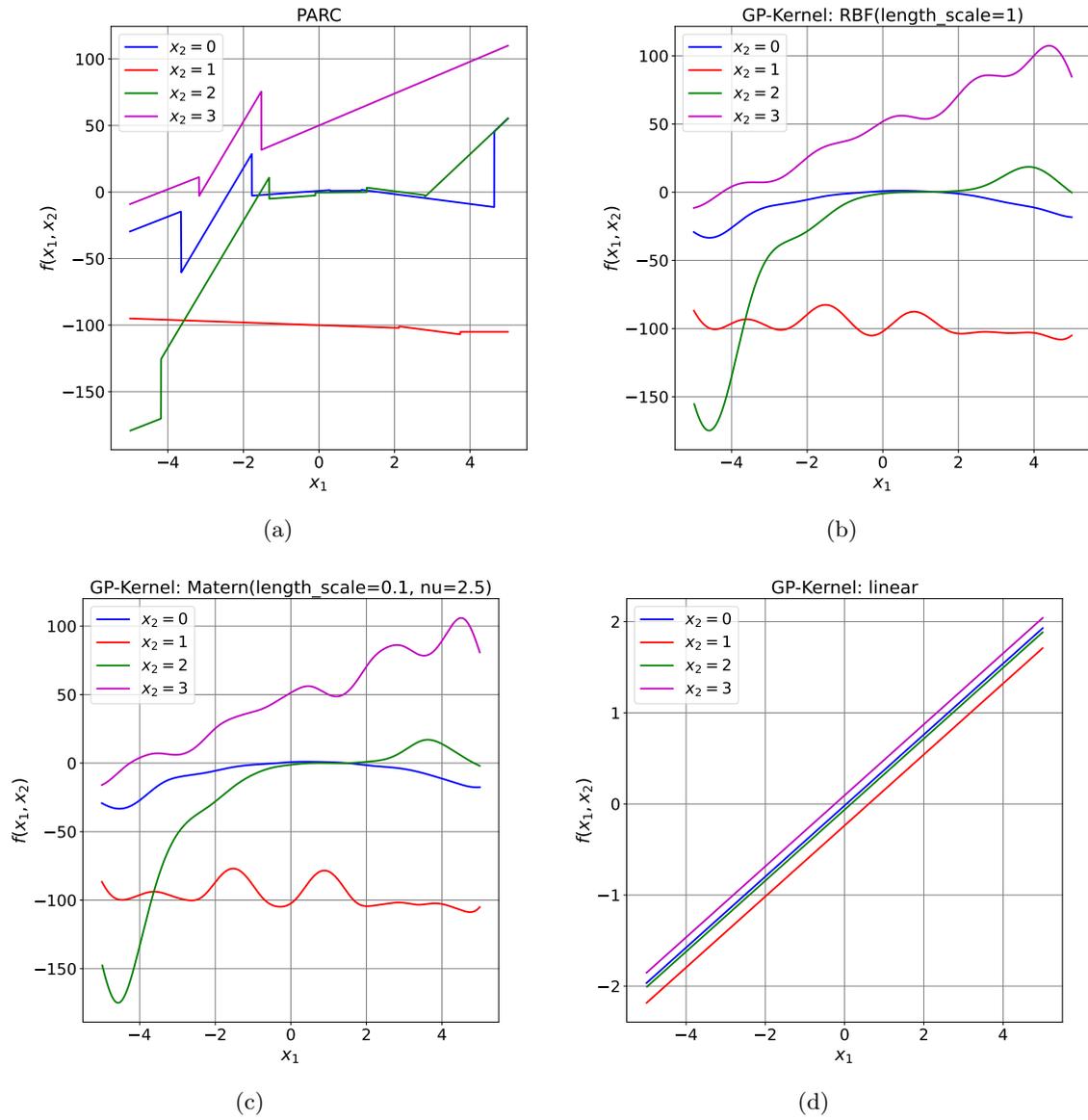


Figure 14: Surrogate model of Function 2 fitted with 50 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear

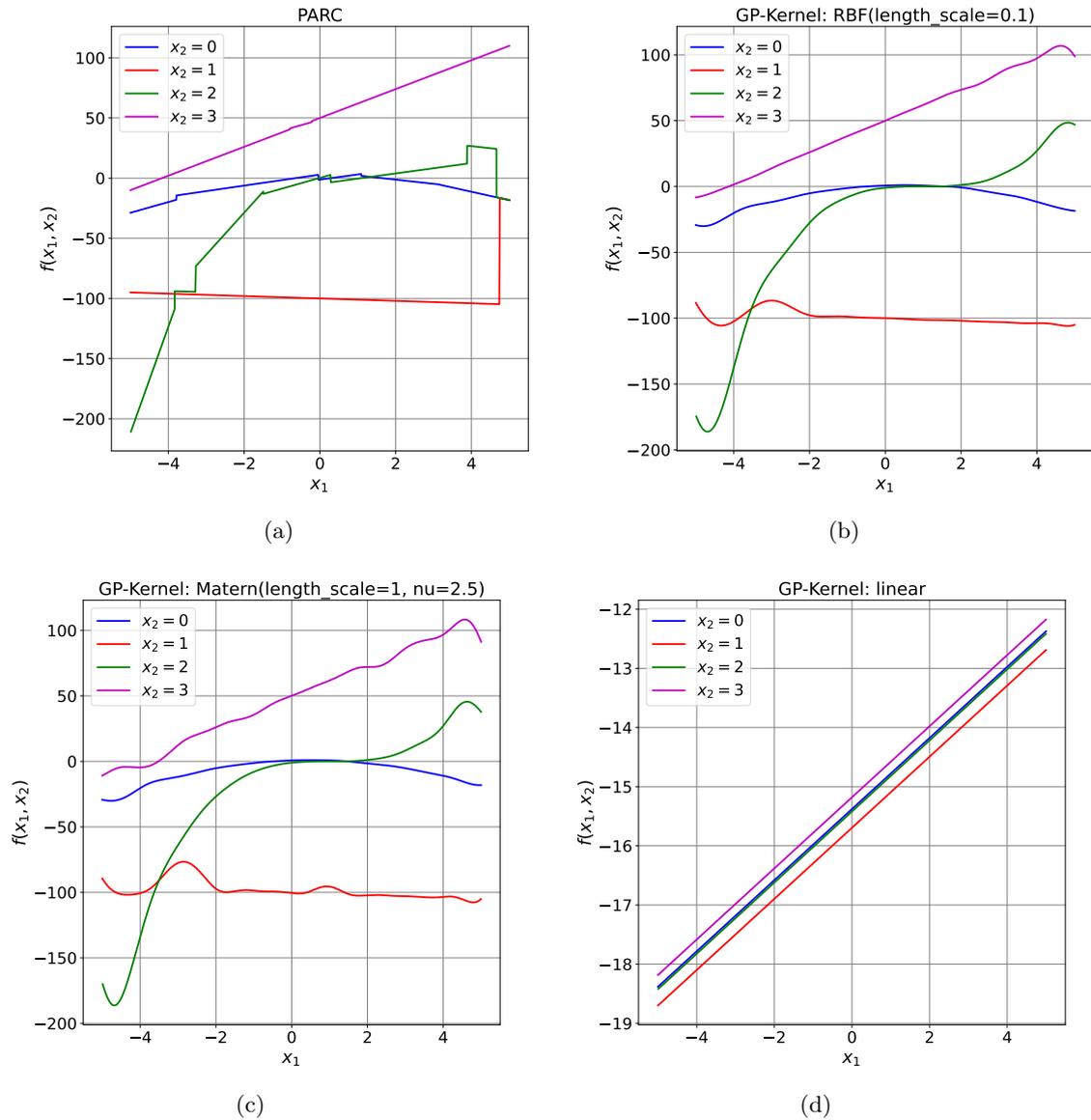


Figure 15: Surrogate model of Function 2 fitted with 70 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear

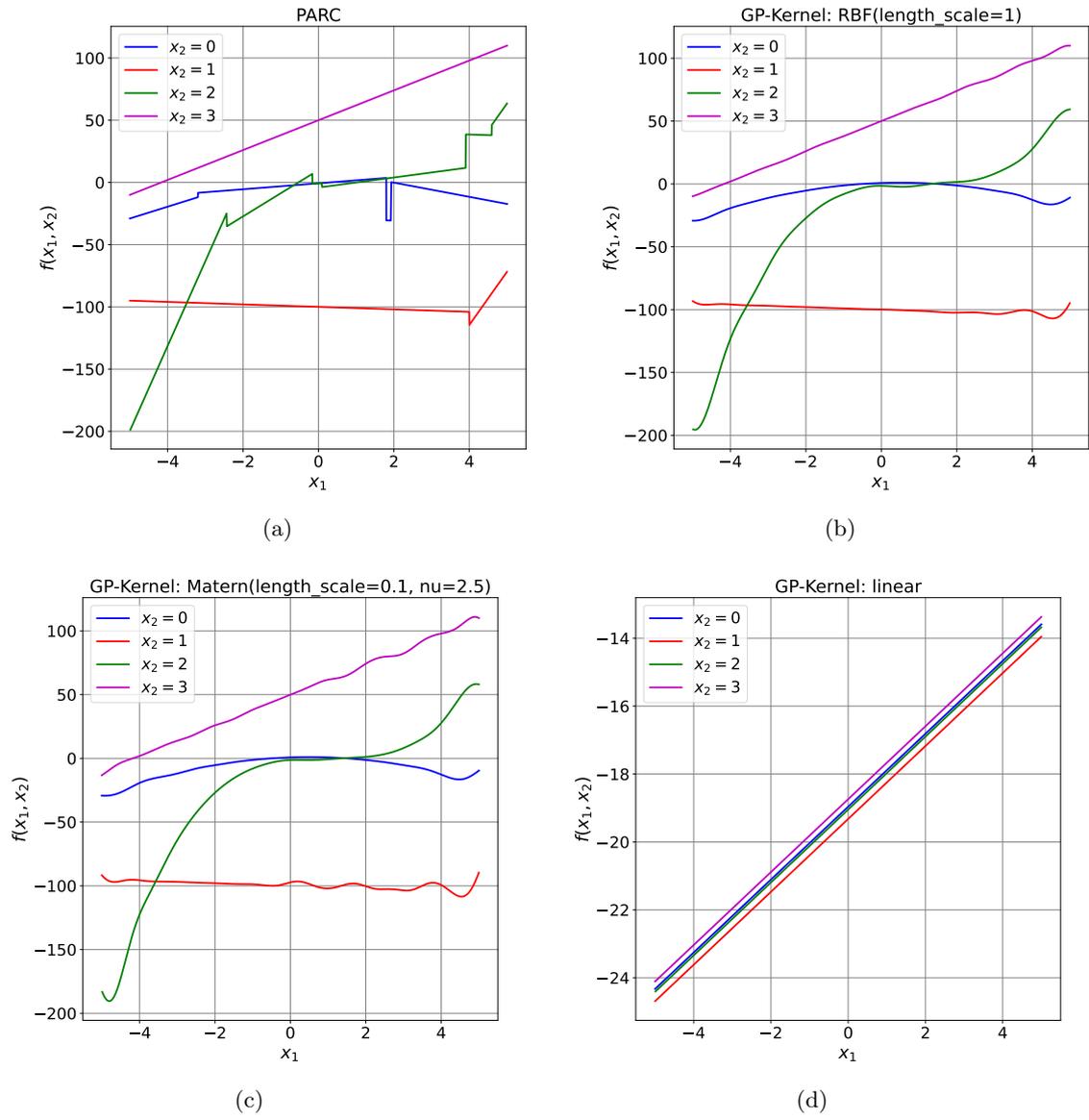


Figure 16: Surrogate model of Function 2 fitted with 80 training samples (a) PWAS, (b) GP-RBF, (c) GP-Matern, and (d) GP-Linear