LLM Enhanced Bayesian Optimization for Scientific Applications like Fusion

Sahand Ghorbanpour†, Ricardo Luna Gutierrez†, Vineet Gundecha†, Desik Rengarajan, Ashwin Ramesh Babu, Soumyendu Sarkar†∗

Hewlett Packard Enterprise

{sahand.ghorbanpour, rluna, vineet.gundecha, desik.rengarajan ashwin.ramesh-babu, soumyendu.sarkar}@hpe.com

Abstract

Although Bayesian optimization (BO) is commonly used to optimize many scientific and industrial experiments, it requires specialized acquisition functions (AFs) to navigate exploration spaces efficiently for various application domains. Moreover, traditional BO struggles to incorporate prior data from simulations or previous experiments, resulting in lower sample efficiency. This paper explores freely available LLMs, fine-tuned to generate Python code, to refine AFs while learning from simulations and experiments. Our results show that this method generates novel acquisition functions that outperform traditional BO AFs, such as Expected Improvement (EI) and Upper Confidence Bound (UCB). We apply this approach to inertial confinement fusion (ICF), a costly and complex problem in which a laser pulse shape is optimized across multiple shots to maximize yield. We demonstrate that this LLM-based technique outperforms classic AFs, such as EI and UCB, in ICF optimization, leading to more efficient and effective optimization and accelerating scientific innovation.

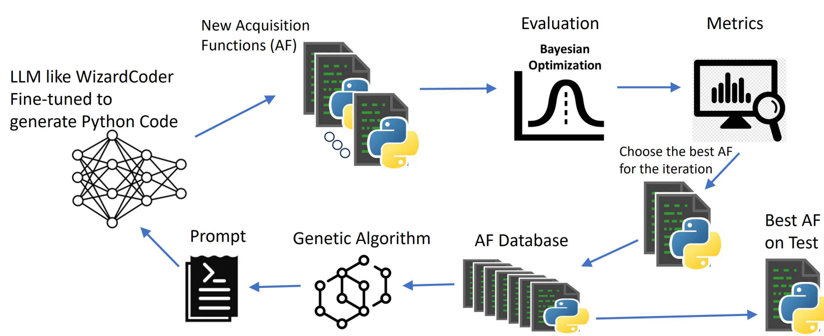

Figure 1: LLM can refine acquisition functions in optimizing scientific experiments like inertial confinement fusion. Diagram based on FunBO.

∗Corresponding author. †These authors contributed equally. Presentation: https://tinyurl.com/y94jfwmn

Machine Learning and the Physical Sciences Workshop, NeurIPS 2024.

1 Introduction

Experiments across various scientific domains are often prohibitively expensive due to their inherent complexity, the need for specialized expertise, and the requirement for advanced equipment. In light of these challenges, developing techniques that maximize experimental outcomes while ensuring sample efficiency is critical. One such technique, Bayesian Optimization (BO), focuses on sample-efficient optimization by combining a surrogate model with an acquisition function (AF) to guide the efficient selection of new experiments. However, traditional BO approaches rely on generic AFs, such as Expected Improvement (EI) or Upper Confidence Bound (UCB), or require handcrafted AFs, which are often challenging to design, especially for complex problems. Furthermore, standard BO methods typically fail to leverage data from previous experiments or simulations to tailor AFs to specific problems.

Recent studies have demonstrated the potential of powerful models, such as Large Language Models (LLMs), to discover novel AFs that enhance the effectiveness of BO across various domains (Aglietti et al., 2024; Liu et al., 2024; Ramos et al., 2023; Yang et al., 2024). In this work, we investigate the benefits of such approaches for optimizing complex functions. Specifically, we focus on FunBO (Aglietti et al., 2024), an AF discovery method that harnesses the capabilities of LLMs and a limited set of source functions—derived from prior experiments or simulations—to design innovative AFs.

To evaluate our approach, we use two widely recognized benchmark functions: the Goldstein-Price and Branin functions. The Goldstein-Price function (Dixon, 1978) is a two-dimensional function known for its challenging optimization landscape, featuring ridges, valleys, and multiple local minima. Similarly, the Branin function (Branin, 1972) is a global optimization benchmark with a complex, multimodal structure. Both functions are ideal for testing optimization algorithms as they challenge the ability to navigate non-linear, multimodal, and non-separable problems effectively.

In addition to these benchmarks, we apply FunBO (Aglietti et al., 2024) to a real-world problem: energy optimization in inertial confinement fusion (ICF). Fusion energy offers significant advantages over fission, including minimal waste production and no risk of catastrophic meltdowns. ICF achieves fusion conditions by using powerful lasers to compress fuel and initiate reactions (Bahrami et al., 2019). However, the high costs and complexities of laser systems, along with the scarcity of specialized facilities capable of conducting such experiments, severely limit practical experimentation. This makes efficient optimization algorithms essential for accelerating progress in ICF research.

Unlike the FunBO approach, which employs Codey, a proprietary LLM-based code generation solution built on PaLM 2 with an expensive API, our study emphasizes the use of smaller, open-source models. These models, freely accessible on platforms like Hugging Face (Wolf et al., 2020), make the methodology more accessible and adaptable to a broader range of researchers, ensuring cost-effective and versatile solutions for scientific optimization challenges.

2 Background

Bayesian Optimization (BO) seeks the global optimum of a black-box objective function by using a surrogate model, typically a Gaussian process, to iteratively select new points via an acquisition function until a predefined budget is exhausted.

2.1 LLM for refining BO Acquisition Function

Designing AFs can use LLMs and a set of source functions $\mathcal{F}$ as an algorithm discovery problem (Aglietti et al., 2024). For real-world problems such as ICF, this set $\mathcal{F}$ can be derived from simulations or historical data from previous experiments.

Starting with an initial AF α_{init} defined in code, this method (Figure 1) iteratively prompts an LLM to generate new AFs aimed at improving BO performance over a set of source (training) functions $\mathcal{F}_s$. At each step τ of the FunBO process, a prompt is created using two previously designed AFs stored in a database (DB). If the DB is empty, as in the initial steps, only α_{init} is used in the prompt. Based on this prompt, the LLM generates B new AFs α_τ. These newly designed AFs are then evaluated over $\mathcal{F}_s$ through N BO steps and are given a numeric score s_{α_τ} based on:

$$s_{\alpha_\tau}(\mathcal{F}_s) = \frac{1}{|\mathcal{F}_s|}\sum_{j=1}^{J}\left[\left(1 - \frac{f_j(x_j^*, \alpha_\tau) - y_j^*}{f_j(x_j^{n=0}) - y_j^*}\right) + \left(1 - \frac{N_{\alpha_\tau}}{N}\right)\right] \tag{1}$$

```python
def acquisition_function(x_points, predictive_mean, predictive_var, incumbent, beta=1.0):
    z = (incumbent - predictive_mean) / np.sqrt(predictive_var)
    predictive_std = np.sqrt(predictive_var)
    vals = (incumbent - predictive_mean) * norm.cdf(z) + predictive_std * norm.pdf(z)
    point = x_points[np.argmax(vals)]
    return point
```

Figure 2: Initial AF used for the LLM method's AF discovery process

.

```python
def acquisition_function_ICF(x_points, predictive_meanpredictive_var, incumbent,
                             beta =0.4):
  z = (incumbent - predictive_mean) / np.sqrt(predictive_var)
  predictive_std = np.sqrt(predictive_var)
  vals = beta * predictive_std * stats.norm.pdf(z) + (incumbent - predictive_mean) \
        * stats.norm.cdf(z)
  point = x_points[np.argmax(vals)]
  return point
```

Figure 3: AF discovered by the adapted LLM approach for the ICF task. The code shown was generated by the LLM and formated to fit the paper width. We can see that the LLM is able to output AFs in correct Python code.

where $f_j \in \mathcal{F}_s$, y_j^* is the true optimum of f_j, $x_j^{n=0}$ is the input value at $n = 0$, x_j^*, α_τ is the best input value found with α_τ, and N_{α_τ} is the number of trials out of N that α_τ took to achieve y_j^*. If the optimum value y_j^* is not found, N_{α_τ} is set equal to N. The first term in the scoring equation rewards the acquisition function for finding values close to the optimum, while the second term rewards it for achieving the optimum in the fewest possible steps N_{α_τ}. The score s_{α_τ} is used to store the generated AFs in the database (DB), and a population-based genetic algorithm uses these scores to determine the pair of AFs to use for prompt generation, favoring those that are shorter and have higher scores.

This process is repeated until the time budget $\mathcal{T}$ is exhausted. From the higher-scoring AFs, the method then selects the AF that performs the best on a set of validation functions $\mathcal{F}_v$. If a validation set is not available, $\mathcal{F}_s$ is used instead. An overview of the FunBO algorithm is shown in Algorithm 4. For our experiments, we use EI as α_{init} as shown in Figure 2

2.2 Inertial Confinement Fusion

Inertial Confinement Fusion (ICF) (Betti and Hurricane, 2016; Lees et al., 2021; Gopalaswamy et al., 2019, 2024) seeks to achieve nuclear fusion by using high-energy lasers to compress a small fuel pellet of deuterium and tritium to extreme temperatures and pressures. The goal is to fuse the nuclei, thereby releasing significant energy with the potential for providing nearly limitless clean power. The energy yield depends on the precise shaping and timing of the laser pulse, governed by five key parameters. However, ICF experiments are costly and limited, with only tens of shots per year for a research team, slowing progress. Sample-efficient optimization techniques are critical for accelerating advancements in fusion research. Other optimization studies this paper draws ideas from include (Gundecha et al., 2024; Gutierrez et al., 2024; Shmakov et al., 2023).

3 Experiments

We ran and assessed the performance of the AFs found by FunBO in the Branin and Goldstein-Price synthetic functions (Surjano, 2024), and the task of ICF energy yield optimization. For ICF's dataset creation, we utilized the LOTUS library (Ejaz et al., 2024) to generate various laser pulse profiles based on a custom parameterization. These parameters determined the laser power and timing, which were then used as inputs for the LILAC simulator (Delettrez et al., 1987), a simulator for laser-driven fusion physics. The chosen laser pulse shape significantly affects the experimental results (energy yield), influencing both the compression of the fusion fuel and the development of hydrodynamic instabilities (Williams et al., 2021). To develop a response surface reflecting different entropy shapes, we varied 5 parameters related to the laser pulse. Using Latin hypercube sampling, we generated 50,000 samples within the design constraints of the laser system. These pulse shapes were tested with the LILAC simulator on a fixed fusion fuel target, and the resulting neutron yields were

Algorithm 1 LLM to refine BO Acquisition Function

Inputs: $\mathcal{F}_s$, $\mathcal{F}_v$, DB, B, $\mathcal{T}$
Setup: Initialize α_{init}, e, and DB with N_{DB} islands.
Assign α_{init} to each island.
while $\tau < \mathcal{T}$ **do**
    Sample two AFs from DB and create prompt
    Get a batch of B AFs from the LLM.
    for each correct α_τ in the batch **do**
        Compute $s_{\alpha_\tau}(\mathcal{F}_s)$.
    end for
    Add correct α_τ to DB.
    Update step $\tau = \tau + 1$.
end while
Output: Return α in DB with score in the top 20^{th} percentile for $\mathcal{F}_s$ and highest score on $\mathcal{F}_v$.

Figure 4: Left: Algorithm of LLM to refine BO Acquisition Functions. Right: An example laser pulse shape used in ICF experiments, controlled by 5 parameters. The parameters are adjusted during an optimization campaign to maximize energy yield.

analyzed to construct a response surface based on these 5 parameters. Additionally, to create a diverse evaluation, we generated two source tasks and one test task by modifying the simulator's physics models, specifically, the equation of state, which affects shock behavior in fusion fuel, thus changing the response surfaces across different simulation versions. This discrepancy between training and test tasks enables us to approximate the divergence observed between simulation and real-world experiments in ICF.

Due to computational costs, we limited the FunBO's steps $\mathcal{T}$ to 1000. The AFs discovered are shown in Figure 3 for ICF, and in the Figures 6 and 7 in the Appendix for Branin and Goldstein-Price.

We used WizardCoder-Python-34B-V1.0 (Luo et al., 2023) as the base LLM for AF discovery, as it is open-source and deployable. Despite its smaller size compared to proprietary models like PaLM, WizardCoder effectively discovered performant AFs within the computational budget. By contrast, Codestral-22B-v0.1 (Mistral AI, 2024) and Phind-CodeLlama-34B (Phind AI, 2024) struggled to generate effective AFs within the defined steps $\mathcal{T}$. This may highlight differences in task-specific training or optimization capability among these models.

3.1 LLM driven Acquisition Function Performance

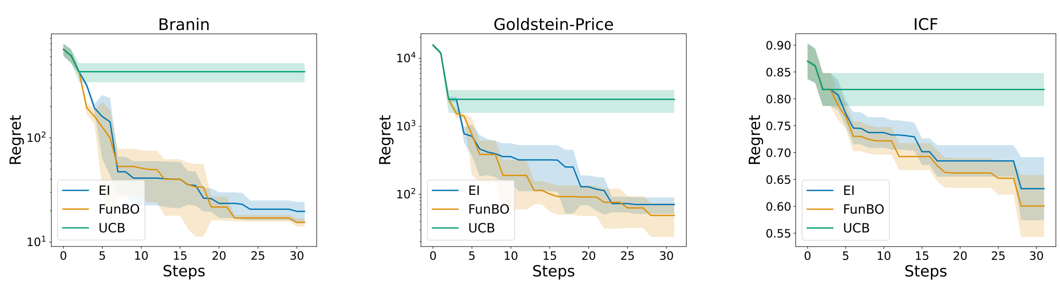

Figure 5: Comparison between the AF discovered by LLM and EI. The LLM approach is able to achieve lower regret in all 3 tasks. The shaded areas represent a ± 1 standard error.

We compare the LLM approach with α_{init} (EI) and UCB to assess this method's abilities to discover new AFs that improve BO performance (Garnett, 2023). We run all experiments using 10 different seeds and used 5 initial random samples to initialize the GP for all tasks. The results of this evaluation are presented in Figure 5. The LLM approach demonstrates superior performance compared to the classic BO baselines in all three tasks. These results show that even with smaller open-source models, complex problems such as ICF can benefit from leveraging LLMs for AF discovery, opening up new opportunities for research and development in applied AI for scientific domains.

4 Conclusion

Our study shows that LLM-based methods can greatly improve experimental optimization in complex scientific fields such as ICF. By enhancing performance and sample efficiency, these methods reduce experimental costs, potentially accelerating fusion energy development, a key step toward sustainable clean energy. The choice of the right LLM can significantly impact computational costs and the number of required samples.

In contrast to FunBO's original work, we use open-source LLM models for acquisition function discovery, which makes the method more accessible and adaptable to a broader range of research contexts for scientific optimization. Our experiment showed that a freely available 34B model designed for code generation outperformed larger LLMs, requiring fewer samples to improve the acquisition function. For some of the benchmark functions, it exceeded results from the FunBO paper, which used the Codey model (PaLM family).

Future work will explore applications in ICF, transfer learning, and dynamic prompting for AF discovery, using feedback from humans or LLMs. We aim to enhance the Funsearch evolutionary algorithm to reduce sample sizes and explore new Bayesian optimization methods like Neural and likelihood-free acquisition functions.

5 Acknowledgement

This work was financially supported by the U.S. Department of Energy under Award No. DE-SC0024408, Hewlett Packard Enterprise, and the University of Rochester.

References

V. Aglietti, I. Ktena, J. Schrouff, E. Sgouritsa, F. J. R. Ruiz, A. Malek, A. Bellot, S. Chiappa, Funbo: Discovering acquisition functions for bayesian optimization with funsearch, 2024. URL: https://arxiv.org/abs/2406.04824. arXiv:2406.04824.

T. Liu, N. Astorga, N. Seedat, M. van der Schaar, Large language models to enhance bayesian optimization, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=OOxotBmGol.

M. C. Ramos, S. S. Michtavy, M. D. Porosoff, A. D. White, Bayesian optimization of catalysts with in-context learning, 2023. URL: https://arxiv.org/abs/2304.05341. arXiv:2304.05341.

C. Yang, X. Wang, Y. Lu, H. Liu, Q. V. Le, D. Zhou, X. Chen, Large language models as optimizers, in: The Twelfth International Conference on Learning Representations, 2024. URL: https://openreview.net/forum?id=Bb4VGOWELI.

L. C. W. Dixon, The global optimization problem: an introduction, Towards Global Optimiation 2 (1978) 1–15.

F. H. Branin, Widely convergent method for finding multiple solutions of simultaneous nonlinear equations, IBM Journal of Research and Development 16 (1972) 504–522.

A. Bahrami, A. Teimourian, C. O. Okoye, H. Shiri, Technical and economic analysis of wind energy potential in uzbekistan, Journal of cleaner production 223 (2019) 801–814.

T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, A. M. Rush, Huggingface's transformers: State-of-the-art natural language processing, 2020. URL: https://arxiv.org/abs/1910.03771. arXiv:1910.03771.

R. Betti, O. A. Hurricane, Inertial-confinement fusion with lasers, Nature Physics 12 (2016) 435–448. URL: https://doi.org/10.1038/nphys3736. doi:10.1038/nphys3736.

A. Lees, R. Betti, J. P. Knauer, V. Gopalaswamy, D. Patel, K. M. Woo, K. S. Anderson, E. M. Campbell, D. Cao, J. Carroll-Nellenback, R. Epstein, C. Forrest, V. N. Goncharov, D. R. Harding, S. X. Hu, I. V. Igumenshchev, R. T. Janezic, O. M. Mannion, P. B. Radha, S. P. Regan, A. Shvydky,

R. C. Shah, W. T. Shmayda, C. Stoeckl, W. Theobald, C. Thomas, Experimentally inferred fusion yield dependencies of omega inertial confinement fusion implosions, Phys. Rev. Lett. 127 (2021) 105001. URL: https://link.aps.org/doi/10.1103/PhysRevLett.127.105001. doi:10.1103/PhysRevLett.127.105001.

V. Gopalaswamy, R. Betti, J. P. Knauer, N. Luciani, D. Patel, K. M. Woo, A. Bose, I. V. Igumenshchev, E. M. Campbell, K. S. Anderson, K. A. Bauer, M. J. Bonino, D. Cao, A. R. Christopherson, G. W. Collins, T. J. B. Collins, J. R. Davies, J. A. Delettrez, D. H. Edgell, R. Epstein, C. J. Forrest, D. H. Froula, V. Y. Glebov, V. N. Goncharov, D. R. Harding, S. X. Hu, D. W. Jacobs-Perkins, R. T. Janezic, J. H. Kelly, O. M. Mannion, A. Maximov, F. J. Marshall, D. T. Michel, S. Miller, S. F. B. Morse, J. Palastro, J. Peebles, P. B. Radha, S. P. Regan, S. Sampat, T. C. Sangster, A. B. Sefkow, W. Seka, R. C. Shah, W. T. Shmyada, A. Shvydky, C. Stoeckl, A. A. Solodov, W. Theobald, J. D. Zuegel, M. G. Johnson, R. D. Petrasso, C. K. Li, J. A. Frenje, Tripled yield in direct-drive laser fusion through statistical modelling, Nature 565 (2019) 581–586.

V. Gopalaswamy, C. A. Williams, R. Betti, D. Patel, J. P. Knauer, A. Lees, D. Cao, E. M. Campbell, P. Farmakis, R. Ejaz, K. S. Anderson, R. Epstein, J. Carroll-Nellenbeck, I. V. Igumenshchev, J. A. Marozas, P. B. Radha, A. A. Solodov, C. A. Thomas, K. M. Woo, T. J. B. Collins, S. X. Hu, W. Scullin, D. Turnbull, V. N. Goncharov, K. Churnetski, C. J. Forrest, V. Y. Glebov, P. V. Heuer, H. McClow, R. C. Shah, C. Stoeckl, W. Theobald, D. H. Edgell, S. Ivancic, M. J. Rosenberg, S. P. Regan, D. Bredesen, C. Fella, M. Koch, R. T. Janezic, M. J. Bonino, D. R. Harding, K. A. Bauer, S. Sampat, L. J. Waxer, M. Labuzeta, S. F. B. Morse, M. Gatu-Johnson, R. D. Petrasso, J. A. Frenje, J. Murray, B. Serrato, D. Guzman, C. Shuldberg, M. Farrell, C. Deeney, Demonstration of a hydrodynamically equivalent burning plasma in direct-drive inertial confinement fusion, Nature Physics 20 (2024) 751–757.

V. Gundecha, R. Gutierrez Luna, S. Ghorbanpour, R. Ejaz, V. Gopalaswamy, R. Betti, A. Naug, P. Faraboschi, S. Sarkar, Meta-learned bayesian optimization for energy yield in inertial confinement fusion, in: NeurIPS 2024 Workshop on Tackling Climate Change with Machine Learning, 2024.

R. L. Gutierrez, S. Ghorbanpour, V. Gundecha, R. Ejaz, V. Gopalaswamy, R. Betti, A. Naug, D. Rengarajan, A. R. Babu, P. Faraboschi, S. Sarkar, Explainable meta bayesian optimization with human feedback for scientific applications like fusion energy, in: NeurIPS 2024 Workshop on Tackling Climate Change with Machine Learning, 2024.

A. Shmakov, A. Naug, V. Gundecha, S. Ghorbanpour, R. L. Gutierrez, A. R. Babu, A. Guillen, S. Sarkar, Rtdk-bo: High dimensional bayesian optimization with reinforced transformer deep kernels, in: 2023 IEEE 19th International Conference on Automation Science and Engineering (CASE), IEEE, 2023, pp. 1–8.

S. Surjano, Goldstein-price function, 2024. URL: https://www.sfu.ca/~ssurjano/goldpr.html, accessed: 2024-09-06.

R. Ejaz, V. Gopalaswamy, A. Lees, C. Kanan, D. Cao, R. Betti, Deep learning-based predictive models for laser direct drive at the omega laser facility, Physics of Plasmas 31 (2024). doi:10.1063/5.0195675.

J. Delettrez, R. Epstein, M. C. Richardson, P. A. Jaanimagi, B. L. Henke, Effect of laser illumination nonuniformity on the analysis of time-resolved x-ray measurements in uv spherical transport experiments, Phys. Rev. A 36 (1987) 3926–3934. URL: https://link.aps.org/doi/10.1103/PhysRevA.36.3926. doi:10.1103/PhysRevA.36.3926.

C. A. Williams, R. Betti, V. Gopalaswamy, A. Lees, High yields in direct-drive inertial confinement fusion using thin-ice DT liner targets, Physics of Plasmas 28 (2021) 122708. URL: https://doi.org/10.1063/5.0069372. doi:10.1063/5.0069372.

Z. Luo, C. Xu, P. Zhao, Q. Sun, X. Geng, W. Hu, C. Tao, J. Ma, Q. Lin, D. Jiang, Wizardcoder: Empowering code large language models with evol-instruct, 2023.

Mistral AI, Mistral announces codestral: A new initiative, 2024. URL: https://mistral.ai/news/codestral/, accessed: 2024-09-09.

Phind AI, Code llama beats gpt-4: The latest in ai coding models, 2024. URL: https://www.phind.com/blog/code-llama-beats-gpt4, accessed: 2024-09-09.

R. Garnett, Bayesian Optimization, Cambridge University Press, 2023.

A Acquisition Functions

The UCB AF used for eveluation is shown in Figure 8. The AFs discovered on the Branin and Goldstein-Price are shown in Figure 6 and 7 respectively. The LLM is able to output the AFs in standard Python code.

```
def acquisition_function_Branin(x_points, predictive_mean, predictive_var, incumbent,
                                beta=1.0):
  z = (incumbent - predictive_mean) / np.sqrt( predictive_var )
  predictive_std = np.sqrt( predictive_var )
  vals = beta * (incumbent - predictive_mean) * stats.norm.cdf(z) + predictive_std \
        * stats.norm.pdf(z)
  vals += np.random.normal(0, 0.01, size=vals.shape)
  vals *= 1 / (predictive_std + 1e-5)
  return x_points[np.argmax(vals)]
```

Figure 6: AF discovered by FunBO for the Branin function.

```
def acquisition_function_GS(x_points, predictive_meanpredictive_var, incumbent,
                            beta=0.4):
  z = (incumbent - predictive_mean) / np.sqrt( predictive_var )
  predictive_std = np.sqrt( predictive_var )
  vals = beta * (incumbent - predictive_mean) * stats.norm.cdf(z) + predictive_std \
        * stats.norm.pdf(z)
  vals += np.sqrt(predictive_var) * 0.1
  vals += np.random.normal(0, 0.01, len(x_points)).reshape(-1,1)
  return x_points[np.argmax(vals)]
```

Figure 7: AF discovered by FunBO for the Goldstein-Price function.

```
def acquisition_function_UCB(x_points, predictive_meanpredictive_var, incumbent,
                             beta=0.4):
  vals = predictive_mean  + predictive_std * beta
  return x_points[np.argmax(vals)]
```

Figure 8: Code of the UCB AF used for evaluation.

A.1 Experimental Setup

The experiments were performed on a server with a Intel(R) Xeon(R) Platinum 8470 CPU and two H-100 GPUs for 130 hours at around 80% capacity.

B LLM Prompt

The template of the prompt used for the ICF experiments is shown in Figure 9

```python
"""Find best acquisition function consider trying different exploration and exploitation
terms, make sure high variations in finding the best acquistion functions.
Be sure you are not just adding a bunch of constant terms that doen't make any changes
in the results.
On every iteration, improve the last acquisition_function_v over the
acquisition_function_vX methods from previous iterations.
Make sure the new acquisition function is different than the previous ones.
"""
import itertools
import numpy as np
from scipy import stats
import funsearch
import GPy
import math
import pickle

@funsearch.evolve
def acquisition_function_v0(eval_points, predictive_mean , predictive_var , incumbent ,
                            beta = 1.0):
  """Returns the index of the point to collect in a vector of eval points.
  Given the posterior mean and posterior variance of a GP model for the objective
  function, this function computes an heuristic and find its optimum. The next
  function evaluation  will be placed at the point corresponding to the selected
  index in a vector of possible eval points.
  Args:
  predictive_mean: an array of shape [num_points , dim] containing the predicted mean
  values for the GP model on the objective function for 'num_points' points of
  dimensionality 'dim'.
  predictive_var: an array of shape [num_points , dim] containing the predicted variance
  values for the GP model on the objective function for 'num_points' points
  of dimensionality 'dim '.
  incumbent: current optimum value of objective function observed.
  beta: a possible hyperparameter to construct the heuristic.
  Returns:
  An integer representing the index of the point in the array of shape [num_points , dim]
  that needs to be selected for function evaluation.
  """
  z = (incumbent - predictive_mean) / np.sqrt(predictive_var)
  predictive_std = np.sqrt(predictive_var)
  vals = (incumbent - predictive_mean) * norm.cdf(z) + predictive_std * norm.pdf(z)
  point = x_points[np.argmax(vals)]
  return point

def acquisition_function_v1(eval_points, predictive_mean, predictive_var, incumbent,
                            beta=1.0):
  """Improved version of `acquisition_function_v0`."""
```

Figure 9: Template of the prompt used for FunBO AF discovery in the ICF task.