Combining Gaussian Processes and Neural Networks as Surrogate Models in Bayesian Optimization

Submitted to Special Session 8: Bayesian optimization: recent achievements and challenges ahead

Omer Ekmekcioglu[1 [0000−0002−8343−8934]], Juergen Branke[1 [0000−0002−4343−5878]], and Nursen Aydin[1 [0000−0002−4569−7070]]

Warwick Business School, University of Warwick, UK

Abstract. Bayesian Optimization (BO) is a powerful method for optimizing expensive black-box optimization problems. Selecting an appropriate surrogate model is challenging due to limited prior information about the characteristics of the underlying black-box function. While Gaussian Processes (GPs) perform well on most synthetic benchmarks, their performance on real-world problems is not guaranteed. We propose using a combination of Bayesian and non-Bayesian surrogate models, i.e., GPs and Neural Networks (NNs), to mitigate model-mismatch risk while largely preserving the sample efficiency of the best model in the mixture.

Keywords: Bayesian Optimization · Surrogate Models

1 Introduction

In black-box optimization problems, where function evaluations are expensive, BO is a common, effective, and sample-efficient solution framework. Expensive black-box functions appear in a wide range of problems, including robotics (21), automated machine learning (14), experimental design (26), drug design (3), and efficient process design (24). At its core, BO relies on constructing a Bayesian surrogate model that fits a set of observations, mapping input parameters to their corresponding output values (10). The Bayesian nature of the surrogate model enables the quantification of uncertainty in unobserved regions of the domain and guides the selection of query points. This selection is performed by computing and optimizing an "acquisition" function. After evaluating the black-box function at these query points, the new information is added to the dataset, and the process is repeated until the evaluation budget is exhausted. Therefore, the quality of the surrogate model fit and the choice of acquisition function are the two key components affecting the performance of a BO algorithm (28).

Some of the most commonly used acquisition functions are Expected Improvement (EI) (30), Knowledge Gradient (KG) (8), and Information Theoretic functions (15; 11). Furthermore, TS has attracted interest across various applications due to its simplicity and effectiveness (20). These acquisition functions

are model-agnostic and can be optimized as long as the surrogate models provide posterior mean and variance.

GPs are by far the most common choice for surrogate models due to their flexibility, accurate uncertainty quantification, and ease of implementation (23; 29; 7). As long as the surrogate model is accurate, the acquisition functions can effectively optimize the black-box function. However, the black-box nature of the problem assumes there is no prior knowledge of the underlying function's characteristics, making the choice of surrogate model just as critical as the choice of acquisition function. Although GPs perform well on synthetic benchmarks, they struggle in high-dimensional problems when the number of available data points is large, and when the input or output spaces are not continuous (12). In such cases, alternative surrogate models can be used to address GPs' shortcomings, such as decision trees or Bayesian Neural Networks (BNNs) (18; 19). However, none of these surrogates are perfect, and it is difficult to pinpoint which model will perform best for a given problem. Furthermore, the computation time to fit a complex surrogate model, such as BNN, may be even more expensive than the evaluation cost of the real-life problem (1), making the use of such surrogates unrealistic, as BNN approaches can be significantly more computationally expensive (18).

In this paper, we propose using a combination of surrogate models, namely a GP and an NN, building on existing work on using a mixture of GPs for surrogate modeling (20; 9). We observe that the GP-NN mixture is effective, as selecting query points based on one surrogate model improves the fit of both surrogate models and thus helps identify high-value points. Due to the synergy from using different models, we observe that the probability of a catastrophic mismatch on a given problem is significantly lower, and the mixture is almost as efficient as the best model on the benchmark.

2 Background

2.1 Bayesian Optimization

BO is a widely used surrogate-based optimization approach for expensive black-box functions. BO seeks to find the minimizer of a black-box problem,

$$\arg\min_{x \in \mathcal{X}} f(x),$$

where $\mathcal{X} \subseteq \mathbb{R}^d$. Then a surrogate, $\hat{f}$, is created to approximate f, using the prior information contained in a dataset, $\mathcal{D} = (\mathbf{X}, \mathbf{y})$, where $\mathbf{X} \subseteq \mathbb{R}^{n \times d}$ is the previously evaluated data points and $\mathbf{y} = f(\mathbf{X})$ is the corresponding function values. Then the acquisition function, $\alpha(x)$, is optimized to select the next query point to evaluate on the black-box function: $x' = \arg\max \alpha(x)$. Once the evaluation budget is exhausted, the best observed point is returned (30).

2.2 GP Mixtures

The surrogate models used in BO should be robust, as fine-tuning model hyperparameters is a difficult online task that may not be feasible. Therefore, it is desirable to have a surrogate model that performs well when no prior information about the black-box system is available. (20; 25) use a mixture of GPs with different kernels to decrease the risk of model-mismatch. Similarly, (9) propose to use a convex combination of GPs and compare this weighted ensemble against a strategy that greedily selects only the single best-fitting model based on cross-validation. These approaches are effective because different kernels are suited to different functional characteristics. Our work follows a similar motivation, but extends beyond GP-only mixtures by incorporating multiple surrogate models.

2.3 Non-Bayesian NN

While BNNs are highly effective surrogate models, various studies have examined the effectiveness of using a single NN with a greedy selection mechanism to select query points (2). This approach queries the best solution as predicted by the NN. Although we refer to this NN model as non-Bayesian, the single neural network baseline can be interpreted as an instance of Thompson Sampling applied to a BNN, as it effectively represents a single realization drawn from the ensemble posterior, or a sample from a GP with Neural Tangent Kernel (NTK) (4). (2) briefly examines this approach, noting that one can modify the last layer to convert a Bayesian model into a deterministic NN. In prior work, it has been shown that greedy approaches perform as well as EI approaches in certain applications, such as bilevel optimization (27).

In our work, we implement the ϵ-greedy policy proposed in (5) when using deterministic NN models. This policy selects the maximizer of the posterior mean with probability $1 - \epsilon$, and samples uniformly at random with probability ϵ, balancing exploration and exploitation. We refer to this approach of using a single network for TS as NNTS.

3 Methodology

In this paper, we analyze the impact of using a surrogate model ensemble to improve the robustness of BO performance. Our ensemble consists of a GP and an NN, allowing us to leverage the complementary modeling capabilities of both approaches. GPs are often effective for smooth synthetic problems but are constrained by kernel assumptions, such as continuity. NNs complement them by excelling in regimes where GPs struggle, such as modeling sharp discontinuities or handling complex combinatorial inputs. In our proposed framework, a model is selected at each BO iteration based on a selection mechanism. We evaluate two selection mechanisms, random and Multi-Armed Bandit (MAB) assisted. In the random approach, with a pre-determined probability, p, we select which model to use in a given BO iteration. In the MAB framework, each surrogate model is

treated as an arm, and the next surrogate is selected using a bandit policy, such as the Upper Confidence Bound (UCB) policy. After selecting the surrogate, we fit it and perform the acquisition function optimization steps as in the standard BO algorithm. We also examine the effect of p on algorithm performance in the numerical results section. We present the modified BO algorithm using the surrogate mixtures in Algorithm 1 and Algorithm 2. In this study, we mainly use TS as our acquisition function.

Algorithm 1: Mixed-Surrogate Bayesian Optimization

Initialize $\mathcal{D}^0$ using Sobol sampling
Fit a surrogate model $\mathcal{S}$ to the initial dataset $\mathcal{D}^0$
for $n \leftarrow 0$ **to** $N-1$ **do**
  Sample $p \sim \mathcal{U}(0,1)$
  if $p > 0.5$ **then**
    Set surrogate $\mathcal{S} \leftarrow$ NN
  else
    Set surrogate $\mathcal{S} \leftarrow$ GP
  end
  $x^{n+1} = \arg\max_x \alpha(x)$; *// Optimize the acquisition function*
  Evaluate: $y^{n+1} = f(x^{n+1})$
  Update: $\mathcal{D}^{n+1} \leftarrow \mathcal{D}^n \cup \{x^{n+1}, y^{n+1}\}$
  Update: $\mathcal{S}$
end
return $\{x_i \mid i = \arg\max_j \{y_j \mid (x_j, y_j) \in D^N\}\}$

In the mixture, we focus on GPs and NNs as the two main surrogate models. Different surrogate models could be added to the mixture we use; however, this would likely slow down the algorithm. As shown in (29; 23; 19), GPs are highly efficient surrogate models, and when they underperform, NN-based models typically perform well (7). This complementarity makes their use as a sparse combination particularly effective.

Our preference for non-Bayesian models stems from the high computational overhead and implementation complexity associated with BNNs. Creating an accurate BNN requires domain-specific design choices, such as selecting between deep ensembles, Hamiltonian Monte Carlo (HMC), or Deep Kernel Learning. The fully Bayesian NN approaches based on HMC are not computationally tractable (16); therefore, NN ensemble approximations (18), Last Layer Bayesian alternatives (2), and similar approximations are preferred. However, (2) reports that the NNTS approach is a strong competitor to those BNN-based methods. Accordingly, in our mixture approach, we rely on the GP to provide principled uncertainty quantification and structured exploration, and we use NNTS for exploitation in promising regions of the search space. This allows us to balance exploration and exploitation while maintaining a lower computational cost than BNN-based alternatives.

The surrogate model selection mechanism is an important component in our algorithm. For this reason, we begin with a simple strategy that randomly alternates between surrogate models, and then evaluate a more sophisticated MAB-

Algorithm 2: MAB-Assisted Mixed-Surrogate Bayesian Optimization

Initialize $\mathcal{D}^0$ using Sobol sampling
Fit a surrogate model $\mathcal{S}$ to the initial dataset $\mathcal{D}^0$
Initialize MAB parameters: counts $C_k = 0$, values $V_k = 0$ for $k \in \{\text{GP}, \text{NN}\}$
for $n \leftarrow 0$ **to** $N - 1$ **do**
  Select surrogate $k^* = \arg\max_k \text{MAB\_Policy}(V_k, C_k, n)$; `// e.g., UCB`
  if $k^* = NN$ **then**
    Set surrogate $\mathcal{S} \leftarrow \text{NN}$
  else
    Set surrogate $\mathcal{S} \leftarrow \text{GP}$
  end
  $x^{n+1} = \arg\max_x \alpha(x; \mathcal{S})$; `// Optimize acquisition using` $\mathcal{S}$
  Evaluate: $y^{n+1} = f(x^{n+1})$
  Update: $\mathcal{D}^{n+1} \leftarrow \mathcal{D}^n \cup \{x^{n+1}, y^{n+1}\}$
  Compute reward r_n (e.g., improvement $y^{n+1} - \max(\mathcal{D}^n)$)
  Update MAB parameters V_{k^*}, C_{k^*} using r_n
  Update: $\mathcal{S}$ on $\mathcal{D}^{n+1}$
end
return $\{x_i \mid i = \arg\max_j \{y_j \mid (x_j, y_j) \in D^N\}\}$

based selection mechanism. MAB has been used in operator selection (6) and, in BO, as a feature selection tool in (22) and in a multi-context BO setting (13). In our setting, MAB allows us to focus on the surrogate that yields a better improvement throughout the BO process. Algorithm 2 shows the MAB-assisted variant of our algorithm. Using UCB for selection improves our algorithm's performance. We define the bandit reward for our approach as $R^{(t)} = y_{\max}^{(t)} - y_{\max}^{(t-1)}$, the incremental improvement in a given iteration. While the use of MAB in this context departs from its classical assumptions due to the decreasing nature of the rewards, designing a bandit framework that accommodates non-stationary rewards is left for future work.

4 Numerical Results

To understand the behavior of distinct surrogate models across various landscapes, we performed a comprehensive empirical analysis involving three stages. First, we evaluated each surrogate's standalone performance in a standard BO setting. Second, we evaluated model fit by decoupling the data acquisition phase from the model inference and testing phase. Our results demonstrate a synergistic effect: query points generated by one surrogate often enhance the predictive accuracy of the other, particularly near the optima and within "promising" regions. Finally, we validated that our proposed mixture method achieves robust performance even in tasks where individual models struggle. The full experimental configuration is detailed in Table 1. We used a very similar experimental setup (2) and (18). We keep the initial points the same for the synthetic problems and increase the number of BO iterations to improve convergence.

Table 1: Summary of experimental configurations and model hyperparameters for BO.

Parameter	**Hartmann (6D)**	**Ackley (10D)**	**Branin (2D)**	**Pest Control (25D)**
General Experiment Settings				
Trials (N_{trials})	10	10	10	5
BO Iterations	190	190	90	100
Initial Points	10	10	10	20
Batch Size	1	1	1	1
IBNN Model Hyperparameters				
Bias Variance (σ_b^2)	1.3	1.3	1.3	1.3
Weight Variance (σ_w^2)	10	10	10	10
Depth	3	3	3	3
NNTS Model Hyperparameters				
Architecture	$[128, 128, 128]$	$[128, 128, 128]$	$[128, 128, 128]$	$[128, 128, 128]$
Activation Function	ELU	ELU	ELU	ELU
Epochs	1000	1000	1000	1000
Learning Rate	10^{-3}	10^{-3}	10^{-3}	10^{-3}
Weight Decay	10^{-4}	10^{-4}	10^{-4}	10^{-4}
Noise Variance	0.01	0.01	0.01	0.01
Acquisition (ϵ-TS)	$\epsilon = 0.1$	$\epsilon = 0.1$	$\epsilon = 0.1$	$\epsilon = 0.1$

4.1 BO Performance

To evaluate surrogate model performance, we first assess BO performance with a single surrogate model. We compare the baseline GP and Infinite-Width Bayesian Neural Network (IBNN) models (17) against the NNTS model. We selected GP and IBNN models as they were identified as the top-performing algorithms in the comprehensive surrogate evaluation by (18), while maintaining high computational efficiency during training. For the evaluation, we use Branin (2D), Hartmann (6D), and Ackley (10D) synthetic problems, as well as the real-life Pest Control (25D) problem. We use TS for GPs and IBNNs as the acquisition function, selecting a single query point per iteration (batch size of 1). We ran the algorithms with sufficient budget to observe clear convergence in every BO run. All experiments are repeated ten times, and we report the mean and standard error across runs.

Figure 1 presents the synthetic test results. We observe that NNTS performs quite well on two of the standard synthetic functions, with GPs performing best. (19) find that GPs are superior to their competitors in synthetic functions but can underperform in real-life tasks. Branin and Hartmann functions are well-suited for GPs, allowing them to be the superior surrogate model, supporting this claim. Furthermore, we observe that while GPs are the best surrogate models on the synthetic benchmarks, they perform poorly on the real-life example, Pest Control. This highlights the inherent difficulty of selecting a surrogate model

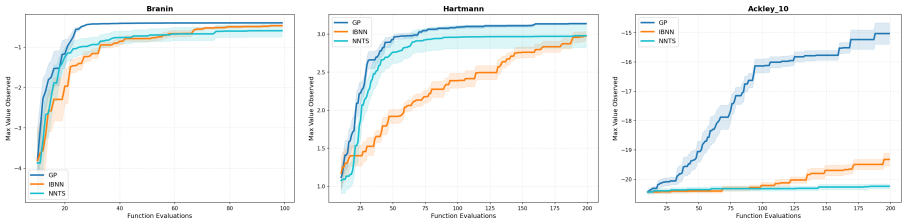


Fig. 1: Synthetic benchmarks: Branin (2D), Hartmann (6D) and Ackley (10D), comparison of mean performance and standard error over 10 runs across different surrogate models.

before the BO run, particularly in the absence of prior knowledge about the objective function. To address the drastic variation across benchmarks, we propose a simple yet effective approach: combining a Bayesian and a non-Bayesian surrogate model, i.e., a GP and an NN.

4.2 One Model Can Improve the Fit of Another

To better understand how model fit evolves, we evaluate each surrogate model on a fixed set of points obtained from a single BO run. By changing the surrogate model used in the BO run, we repeat the same experiment. This allows us to construct datasets collected using GPs, IBNNs, and NNTS in BO. We incrementally retrain the models on these datasets, increasing the training set size by 10 points at each step to evaluate each model's predictive performance. This performance was assessed on randomly sampled test points across the entire domain using standard metrics, including MSE, RMSE, MAE, and log-likelihood. We present the results in Figures 2. We also present the results for the remaining benchmarks and the detailed experiment settings in the appendices. We report the mean and standard error over five repeated experiments.

We observe that using Bayesian models (e.g., GP or IBNN) to select query points consistently minimizes loss across the entire domain because they tend to explore undersampled regions. In contrast, the greedy NNTS approach selects query points in close proximity, which often yields smaller immediate improvements in the fitness loss. Moreover, this tendency to collect closely clustered points can negatively affect GP and IBNN models by degrading their overall fit, despite strong BO performance. This stems from the fact that successful BO does not require a perfect global approximation of the landscape; instead, it necessitates accurate modeling of regions of interest (i.e., near the optima). In Figures 3 and 4, we evaluate the model fits on the relevant high-value points by restricting our attention to the top 10%; this reveals significant changes in performance.

To highlight the difference in the data collection on the model fit, we compare the MSE across the whole domain and the MSE on the top 10% of the points in Figures 5 and 6. The key observation from these figures is that, depending on

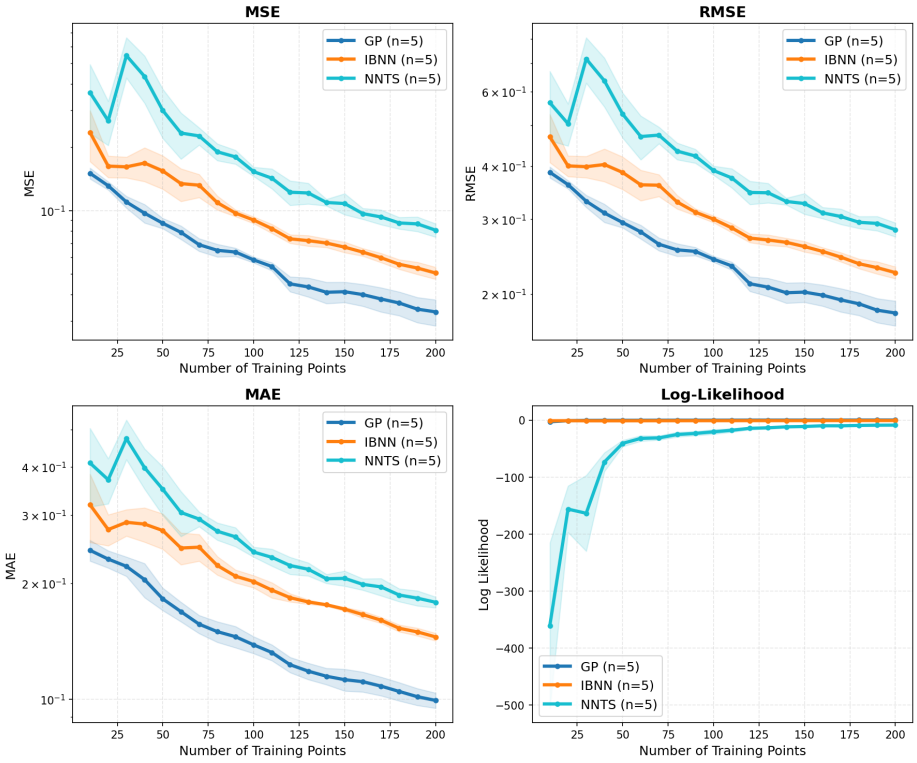


Fig. 2: Surrogate fitness using the data collected by a BO run using the GP model on the Hartmann function.

the problem, queries generated by one surrogate model can be very informative for all surrogates. Figure 5 shows that when data are collected using IBNNs on the Branin function, the remaining models learn to identify important regions more accurately and with greater sample efficiency than when data are collected with other models. Similarly, when using IBNNs to collect data on the Ackley function, Figure 6 shows that GPs learn the relevant regions faster than when data are collected using GPs directly. This transfer of information enables models to better fit the problems where their BO performance was initially weak. Overall, this cross-model learning effect helps explain why surrogate mixtures can outperform single-model approaches in practice.

4.3 Mixture of Surrogates Improves Performance

To assess whether a mixture approach enhances the robustness of individual models, we conducted a comparative analysis. Figures 7 and 8 present the results of three experiments: one in which GPs perform well, and two in which NNTS outperforms other surrogates. These experiments follow the same exper-

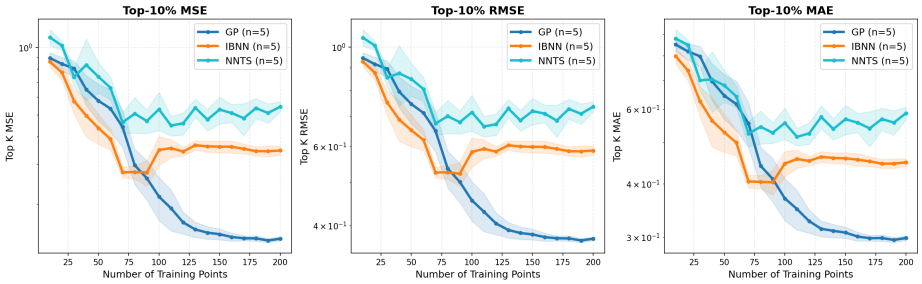


Fig. 3: Weighted fitness of surrogates using the data collected by a BO run using the GP model on the Ackley function.

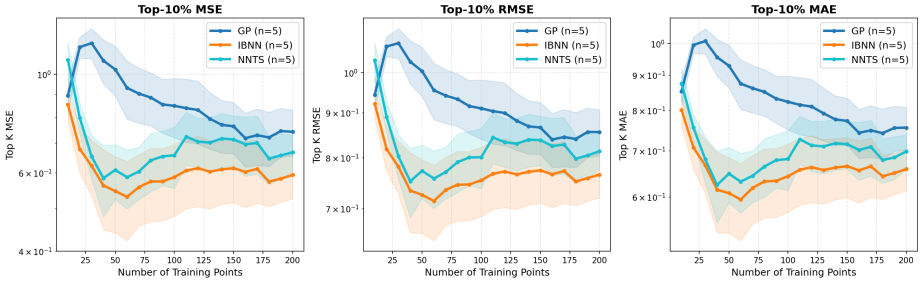


Fig. 4: Weighted fitness of surrogates using the data collected by a BO run using the NNTS model on the Ackley function.

imental setup as used in Section 4.1. In Figure 7, we use the mixture approach with randomized selection of the surrogate model to be used in a given iteration. We evaluate different mixture ratios to study how sampling from a single model influences overall performance. Specifically, we consider GP/NNTS splits of 70/30, 50/50, and 30/70. From the performance curves, we see that the mixture model outperforms the weaker individual surrogates and hedges against the risk of a single surrogate failing. We observe that a 50/50 selection ratio is an effective way to mix models while managing risk. We further evaluate MAB-based selection in Figure 8. We observe that using a UCB policy for surrogate selection improves the BO performance beyond the random selection baseline. We observe that on Hartmann and Pest problems, the mixture converges faster than the individual models. Across experiments, MAB-UCB maintains a balanced selection across all functions. The selection ratios are (47% GP / 53% NNTS) on Branin, (36%, 64%) on Hartmann, (50%,50%) on Ackley, and (48% GP / 52% NNTS) on pest control. The MAB's tendency to approximate a uniform distribution supports the competitive performance observed in the fixed 50/50 random selection baseline.

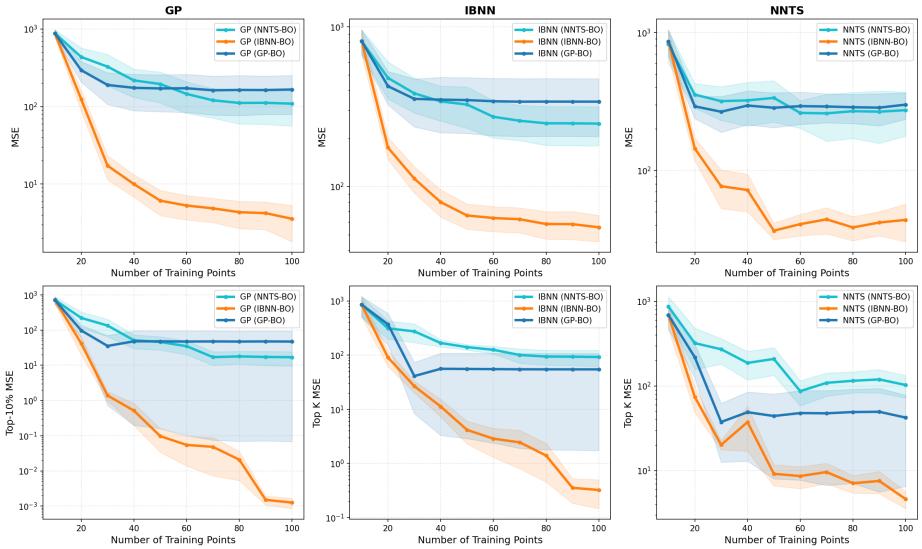


Fig. 5: Surrogate evaluation on Branin function when different surrogate models are used in the data collection using BO.

5 Conclusion

In this paper, we proposed a mixed-surrogate BO algorithm that alternates between a GP and an NN surrogate. We explore both a simple random selection and an MAB-assisted selection mechanism to choose which model to use at each iteration. This design allows us to switch between two different surrogates across iterations, hedging against model-mismatch risk. We also evaluate how query points collected by one surrogate can affect the fit of other models. We observe that random sampling from the mixture is effective and can be further improved by an MAB-assisted selection mechanism. For future work, there are several avenues to explore. First, one could extend the surrogate mixture to include additional models, such as decision trees, to better understand each model's contribution to the mixture. Second, motivated by the performance gain from MAB-assisted surrogate selection, one could further refine the selection mechanism to improve our algorithm's performance. Finally, while we used the same acquisition functions across all models, the mixture framework could also enable model-specific acquisition functions, potentially further amplifying BO performance.

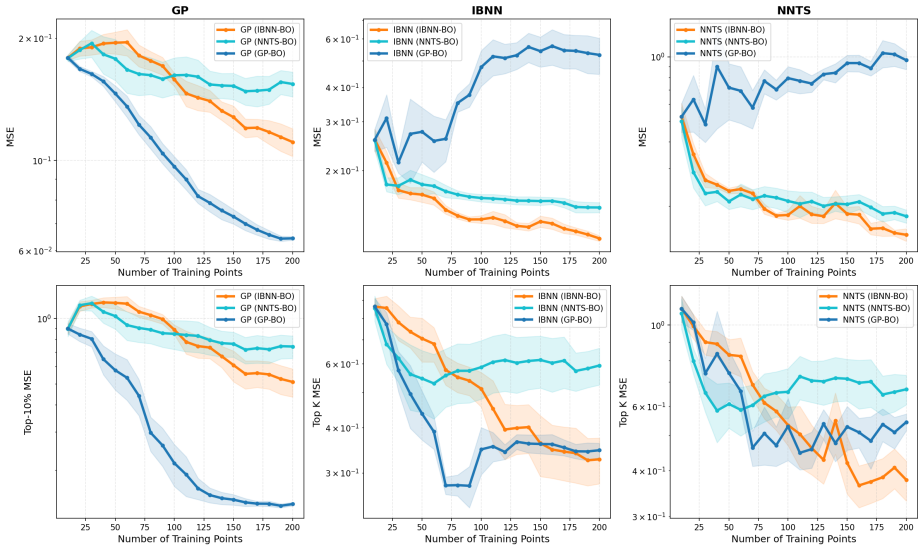
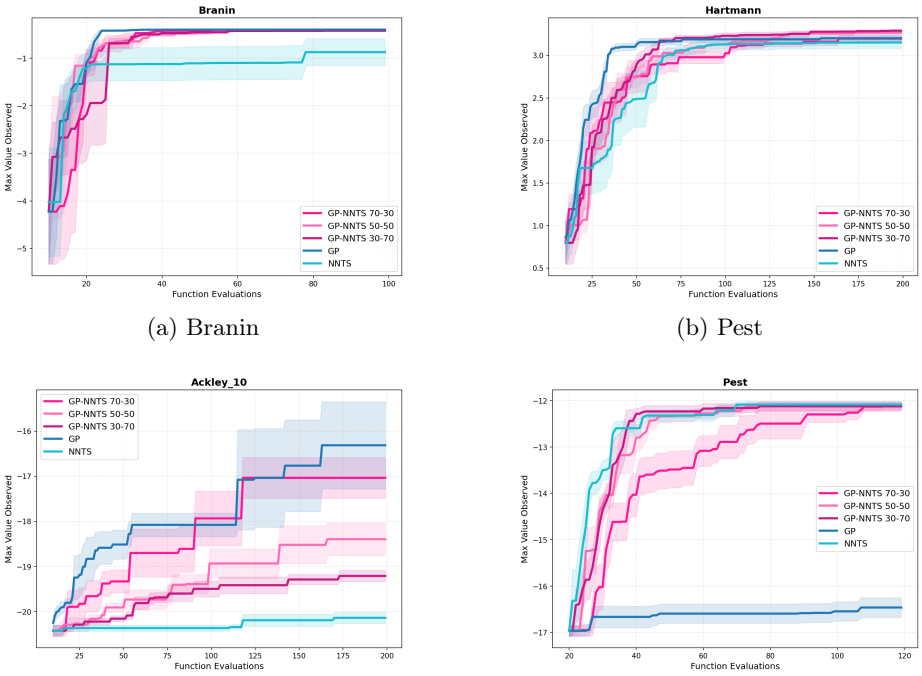


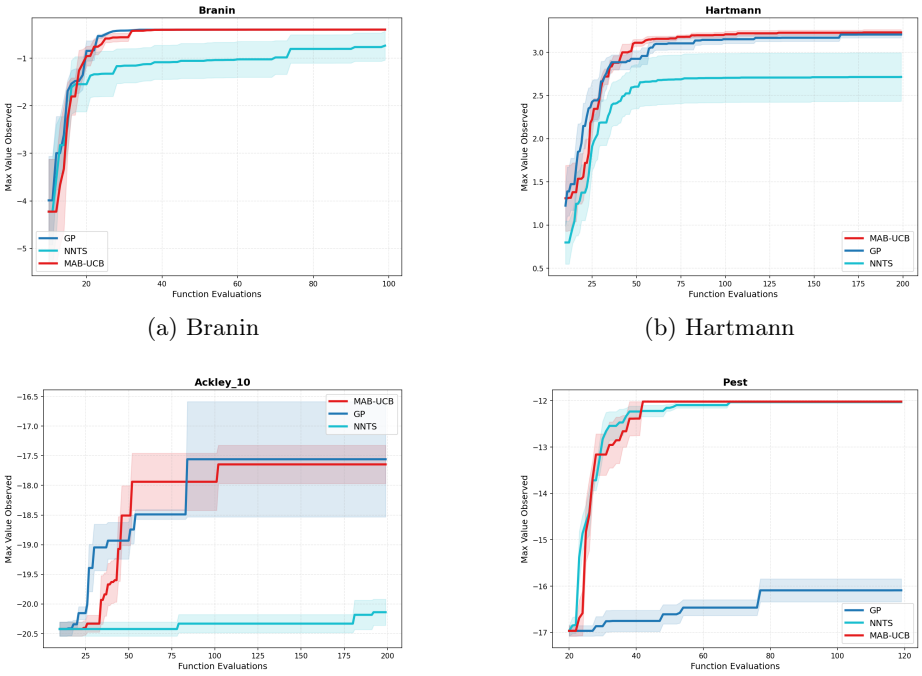
Fig. 6: Surrogate evaluation on Ackley function when different surrogate models are used in the data collection using BO.



(a) Branin (b) Pest

(c) Ackley 10 (d) Pest

Fig. 7: Branin, Pest, and Ackley benchmark for mixture of GP and NN models and standard error over 5 runs.



(a) Branin (b) Hartmann

(c) Ackley 10 (d) Pest

Fig. 8: Branin, Pest, and Ackley benchmark for mixture of GP and NN models and standard error over 5 runs.

Bibliography

[1] Bliek, L., Guijt, A., Karlsson, R., Verwer, S., de Weerdt, M.: Benchmarking surrogate-based optimisation algorithms on expensive black-box functions. Applied Soft Computing **147**, 110744 (2023)

[2] Brunzema, P., Jordahn, M., Willes, J., Trimpe, S., Snoek, J., Harrison, J.: Bayesian optimization via continual variational last layer training. In: International Conference on Learning Representations (2025)

[3] Colliandre, L., Muller, C.: Bayesian optimization in drug discovery. Methods in Molecular Biology p. 101–136 (2023)

[4] Dai, Z., Shu, Y., Low, B.K.H., Jaillet, P.: Sample-then-optimize batch neural thompson sampling (2022), https://arxiv.org/abs/2210.06850

[5] De Ath, G., Everson, R.M., Rahat, A.A.M., Fieldsend, J.E.: Greed is good: Exploration and exploitation trade-offs in Bayesian optimisation. ACM Transactions on Evolutionary Learning and Optimization **1**(1) (2021)

[6] Fialho, Á., Da Costa, L., Schoenauer, M., Sebag, M.: Analyzing bandit-based adaptive operator selection mechanisms. Annals of Mathematics and Artificial Intelligence **60**(1), 25–64 (2010)

[7] Foldager, J., Jordahn, M., Hansen, L.K., Andersen, M.R.: On the role of model uncertainties in Bayesian optimisation. In: Conference on Uncertainty in Artificial Intelligence. vol. 216, pp. 592–601. PMLR (2023)

[8] Frazier, P., Powell, W., Dayanik, S.: The knowledge-gradient policy for correlated normal beliefs. INFORMS Journal on Computing **21**(4), 599–613 (2009)

[9] Friese, M., Bartz-Beielstein, T., Bäck, T., Naujoks, B., Emmerich, M.: Weighted ensembles in model-based global optimization. In: AIP conference proceedings. vol. 2070, p. 020003. AIP Publishing LLC (2019)

[10] Garnett, R.: Bayesian Optimization. Cambridge University Press (2023)

[11] Garrido-Merchán, E.C.: Information-theoretic Bayesian optimization: Survey and tutorial. arXiv preprint arXiv:2502.06789 (2025)

[12] González-Duque, M., Michael, R., Bartels, S., Zainchkovskyy, Y., Hauberg, S., Boomsma, W.: A survey and benchmark of high-dimensional Bayesian optimization of discrete sequences. Advances in Neural Information Processing Systems **37**, 140478–140508 (2024)

[13] Gopakumar, S., Gupta, S., Rana, S., Nguyen, V., Venkatesh, S.: Algorithmic assurance: An active approach to algorithmic testing using Bayesian optimisation. In: Advances in Neural Information Processing Systems. vol. 31, p. 5465–5473 (2018)

[14] He, X., Zhao, K., Chu, X.: Automl: A survey of the state-of-the-art. Knowledge-Based Systems **212**, 106622 (2021)

[15] Henrández-Lobato, J.M., Hoffman, M.W., Ghahramani, Z.: Predictive entropy search for efficient global optimization of black-box functions. In: Neural Information Processing Systems. p. 918–926. MIT Press (2014)

[16] Izmailov, P., Vikram, S., Hoffman, M.D., Wilson, A.G.G.: What are Bayesian neural network posteriors really like? In: International Conference on Machine Learning. pp. 4629–4640. PMLR (2021)
[17] Lee, J., Bahri, Y., Novak, R., Schoenholz, S.S., Pennington, J., Sohl-Dickstein, J.: Deep neural networks as Gaussian processes. International Conference on Learning Representations (2018)
[18] Li, Y.L., Rudner, T.G.J., Wilson, A.G.: A study of Bayesian neural network surrogates for Bayesian optimization (2024), https://arxiv.org/abs/2305.20028
[19] Lim, Y.F., Ng, C.K., Vaitesswar, U., Hippalgaonkar, K.: Extrapolative bayesian optimization with Gaussian process and neural network ensemble surrogate models. Advanced Intelligent Systems **3**(11), 2100101 (2021)
[20] Lu, Q., Polyzos, K.D., Li, B., Giannakis, G.B.: Surrogate modeling for Bayesian optimization beyond a single Gaussian process. arXiv preprint arXiv:2205.14090 (2022)
[21] Martinez-Cantin, R.: Bayesian optimization with adaptive kernels for robot control. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). p. 3350–3356. IEEE Press (2017)
[22] Nguyen, D., Gupta, S., Rana, S., Shilton, A., Venkatesh, S.: Bayesian optimization for categorical and category-specific continuous inputs. AAAI Conference on Artificial Intelligence **34**(04), 5256–5263 (2020)
[23] Papenmeier, L., Poloczek, M., Nardi, L.: Understanding high-dimensional Bayesian optimization. In: International Conference on Machine Learning (2025), https://openreview.net/forum?id=1d2fpvyKvJ
[24] Paulson, J.A., Tsay, C.: Bayesian optimization as a flexible and efficient design framework for sustainable process systems. Current Opinion in Green and Sustainable Chemistry **51**, 100983 (2025)
[25] Polyzos, K.D., Lu, Q., Giannakis, G.B.: Bayesian optimization with ensemble learning models and adaptive expected improvement. In: IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 1–5. IEEE (2023)
[26] Shahriari, B., Swersky, K., Wang, Z., Adams, R.P., De Freitas, N.: Taking the human out of the loop: A review of Bayesian optimization. Proceedings of the IEEE **104**(1), 148–175 (2015)
[27] Wang, B., Singh, H.K., Ray, T.: Comparing expected improvement and kriging believer for expensive bilevel optimization. 2021 IEEE Congress on Evolutionary Computation (CEC) (2021)
[28] Wang, X., Jin, Y., Schmitt, S., Olhofer, M.: Recent advances in Bayesian optimization. ACM Computing Surveys **55**(13s) (2023)
[29] Xu, Z., Wang, H., Phillips, J.M., Zhe, S.: Standard Gaussian process is all you need for high-dimensional Bayesian optimization (2025), https://arxiv.org/abs/2402.02746
[30] Zhan, D., Xing, H.: Expected improvement for expensive optimization: a review. Journal of Global Optimization **78**(3), 507–544 (2020)

6 Appendix

6.1 Model Fitness

In this section, we present figures illustrating the surrogate models' fitness across different functions. Figures 9 and 10 show the surrogate fitness across the whole domain when a specific model is used to collect data points. In Figure 11, we compare the performance of every individual model when different models are used to collect data points. In Figures 12 and 13, we present the fitness metrics for the high-value points for the Branin function.

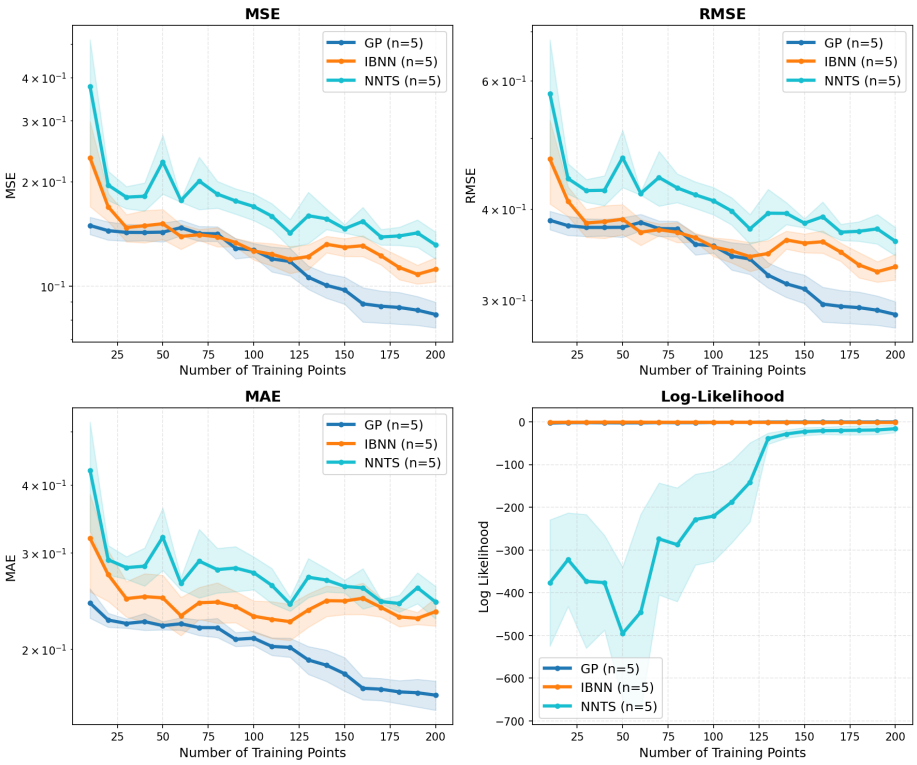


Fig. 9: Surrogate fitness using the data collected by a BO run using the NNTS model on the Hartmann function.

Figure 14 illustrates the progression of arm selections made by the algorithm. We observe that, across all problems, the arm selections are quite uniform. For the Hartmann function, we observe that selection prefers NNTS toward the end, leveraging the greedy approach it offers.

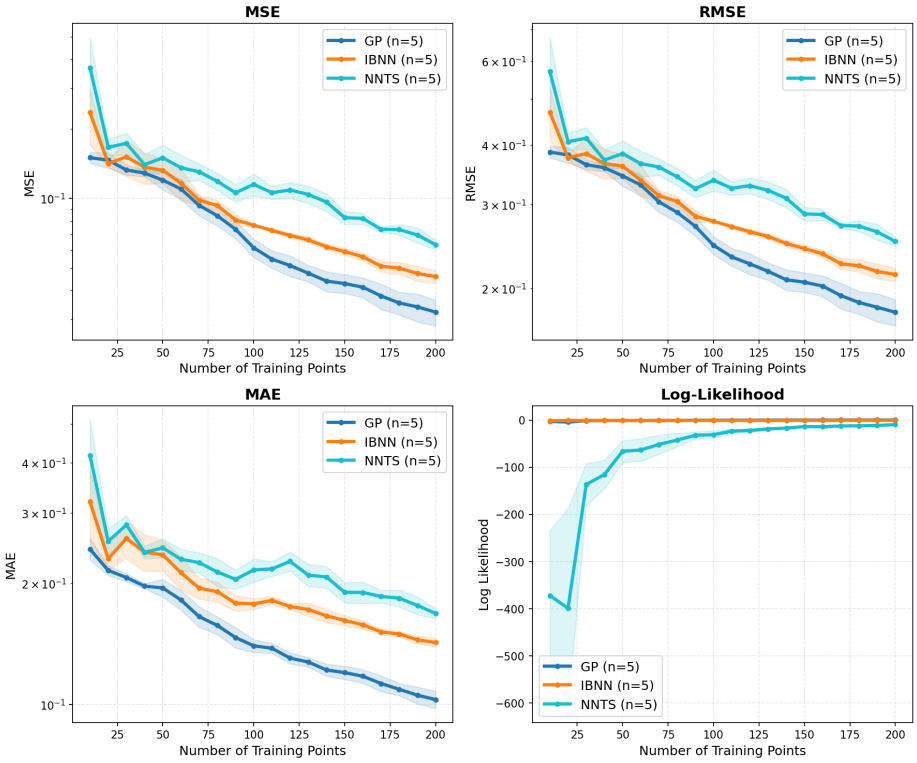


Fig. 10: Surrogate fitness using the data collected by a BO run using the IBNN model on the Hartmann function.

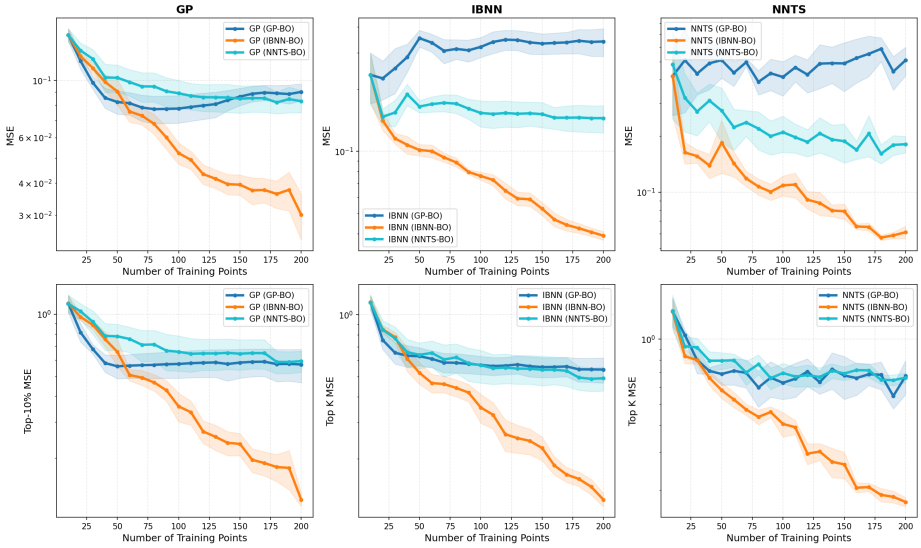


Fig. 11: Surrogate evaluation on Hartmann function when different surrogate models are used in the data collection using BO.

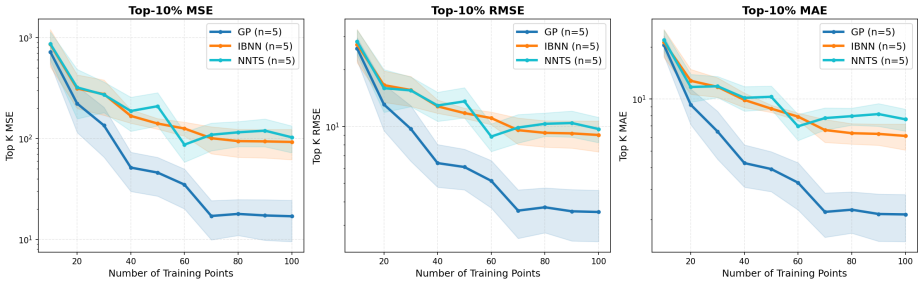


Fig. 12: Surrogate fitness using the data collected by a BO run using the NNTS model on the Branin function.

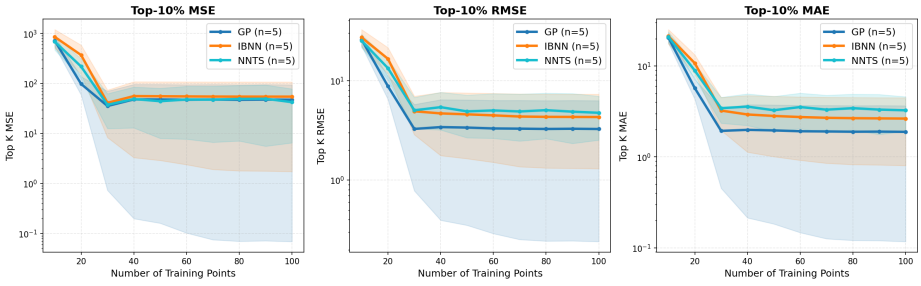


Fig. 13: Surrogate fitness using the data collected by a BO run using the GP model on the Branin function.

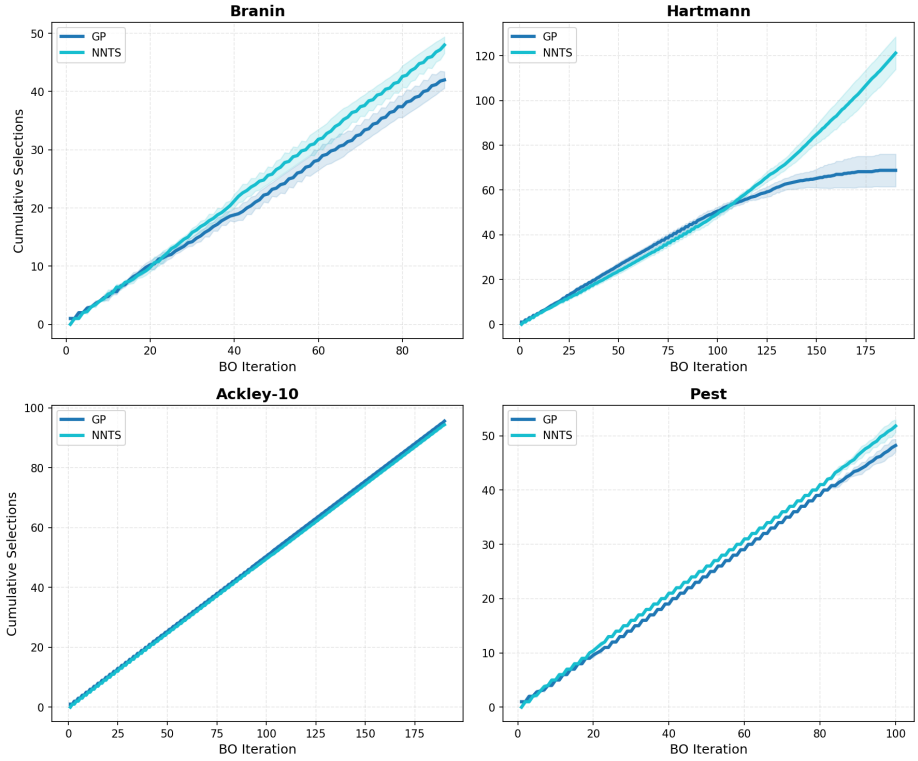


Fig. 14: Cumulative selection of surrogate models using MAB assistance.