CS 726: Advanced Machine Learning

April 16, 2025

Spring 2025

CS 726: Programming assignment 4 **Submitted by:** Pinak Mahapatra (22B0447), Aansh Samyani(22B0424), Danish Siddiqui(22B02104)

Contents

1 Problem Statement

Energy-Based Models (EBMs) provide a flexible framework for representing complex probability distributions. Instead of defining a normalized probability density directly, an EBM defines an energy function $E_\theta(x)$, often parameterized by a neural network with parameters θ. The probability distribution is then implicitly defined as:

$$p_\theta(x) = \frac{\exp(-E_\theta(x))}{Z_\theta}$$

where $Z_\theta = \int \exp(-E_\theta(x'))\, dx'$ is the partition function, which is typically intractable to compute.

While training EBMs presents challenges (often addressed by contrastive methods), sampling from a *trained* EBM, $p(x) \propto \exp(-E(x))$, is also non-trivial. Markov Chain Monte Carlo (MCMC) methods are commonly employed for this task. This assignment focuses on using Langevin dynamics-based MCMC algorithms to draw samples from a probability distribution defined by a pre-trained neural network energy function.

2 Task 0

We evaluated the pre-trained energy-based neural network model on the provided test dataset consisting of 100,000 samples of 784-dimensional inputs. The model was evaluated using Mean Squared Error (MSE) loss, yielding a test loss of **288.16**.

3 Task 1

3.1 Langevin Dynamics Sampling Algorithms

We implemented two Langevin dynamics-based MCMC samplers: Metropolis-Adjusted Langevin Algorithm (MALA) and Unadjusted Langevin Algorithm (ULA), both using a pre-trained energy function $E(x)$.

Common setup:

- The sampler is initialized with a random vector $x_0 \in \mathbb{R}^{784}$.
- At each iteration, we compute the gradient of the energy $\nabla_x E(x)$ using backpropagation.
- A proposal is generated using Langevin dynamics: a step in the direction of the negative gradient plus Gaussian noise.

MALA (Algo1): This algorithm uses a Metropolis-Hastings acceptance step to correct for the approximation in the proposal distribution, ensuring that the stationary distribution is correct.

ULA (Algo2): This algorithm omits the acceptance step, trading off sampling correctness for speed. It may diverge for large step sizes but is computationally cheaper.

3.2 Pseudocode for Both Algorithms

Below is the pseudocode for both algorithms:

Algorithm 1 Metropolis-Adjusted Langevin Algorithm (MALA)

```
 1: Initialize $x \leftarrow x_0$
 2: for $t = 1$ to $N + \text{burn\_in}$ do
 3:     Compute $\nabla E(x)$ using backpropagation
 4:     Sample noise $\xi \sim \mathcal{N}(0, I)$
 5:     Propose $x' \leftarrow x - \frac{\tau}{2}\nabla E(x) + \sqrt{\tau} \cdot \xi$
 6:     Compute $\nabla E(x')$
 7:     Compute acceptance probability:
            $$\log \alpha = E(x) - E(x') + \log q(x|x') - \log q(x'|x)$$
 8:     Accept $x'$ with probability $\min(1, \exp(\log \alpha))$
 9:     if accepted then
10:         $x \leftarrow x'$
11:     end if
12:     if $t > \text{burn\_in}$ then
13:         Save $x$ to samples
14:     end if
15: end for
```

Algorithm 2 Unadjusted Langevin Algorithm (ULA)

```
 1: Initialize $x \leftarrow x_0$
 2: for $t = 1$ to $N + \text{burn\_in}$ do
 3:     Compute $\nabla E(x)$ using backpropagation
 4:     Sample noise $\xi \sim \mathcal{N}(0, I)$
 5:     Update:
            $$x \leftarrow x - \frac{\tau}{2}\nabla E(x) + \sqrt{\tau} \cdot \xi$$
 6:     if $t > \text{burn\_in}$ then
 7:         Save $x$ to samples
 8:     end if
 9: end for
```

3.3 Sampling Times

We recorded the time taken to generate samples using both Langevin-based MCMC algorithms. As expected, MALA took longer due to the additional acceptance-rejection step and gradient evaluation at both the current and proposed positions.

Algorithm	**Sampling Time (seconds)**
MALA (Algo 1)	18.95
ULA (Algo 2)	8.88

Table 1: Sampling time comparison between MALA and ULA for 1000 samples (excluding burn-in).

3.4 Results and Plots

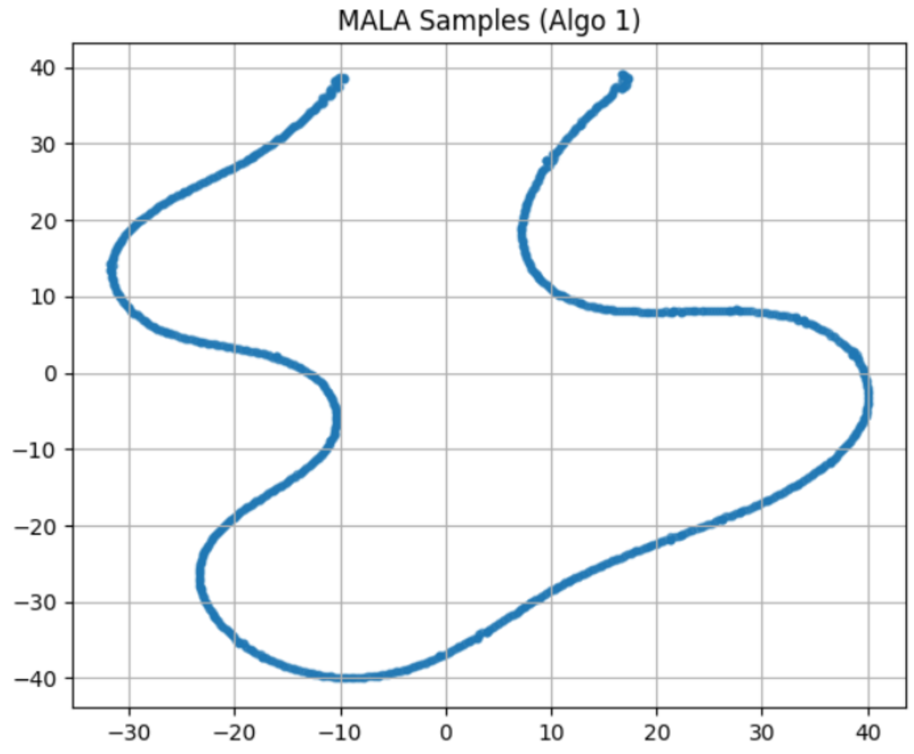

Figure 1: Algorithm 1: MALA

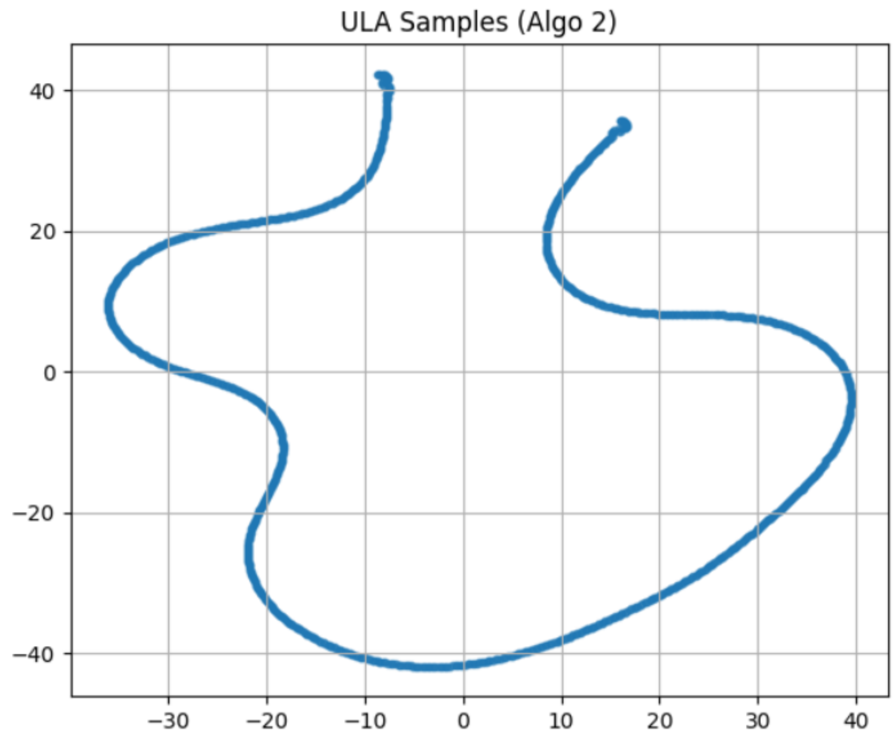

Figure 2: Algorithm 2:ULA

4 Task 2

4.1 Results with Matérn Kernel and Expected Improvement (EI)

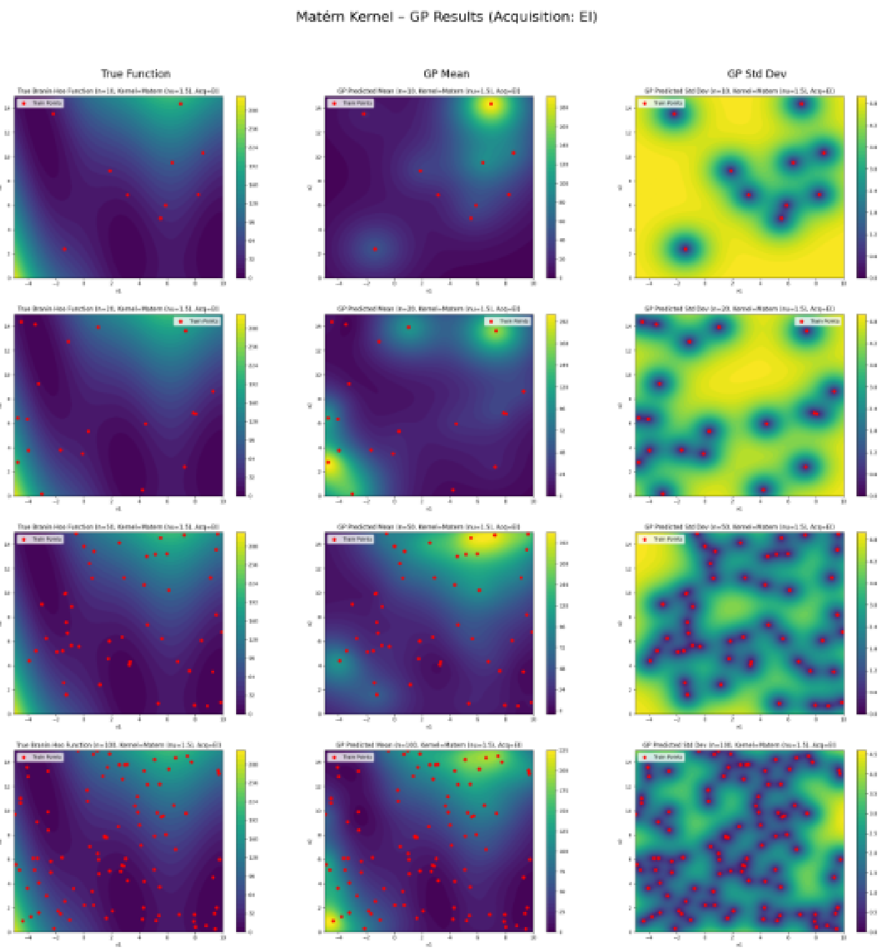

Figure 3: Final plots for Matern kernel and EI strategy

The above figure illustrates the Gaussian Process (GP) regression results using the Matérn kernel for various training set sizes ($n = 10, 20, 50, 100$), with the Expected Improvement (EI) acquisition strategy.

Each row in the figure corresponds to a different number of training samples. The three columns respectively show:

- **True Branin-Hoo function with training points overlaid**
- **Predicted GP mean**
- **Predicted GP standard deviation (uncertainty)**

As the number of samples increases, the model's predictive mean becomes more accurate and closely resembles the true Branin-Hoo function. Additionally, the uncertainty (standard deviation) reduces significantly in the vicinity of training points and the overall surface becomes smoother and more confident.

Observations:

- For $n = 10$, the model captures only coarse trends, and the uncertainty is high across the domain.
- At $n = 20$, the model begins identifying local structure, especially around observed points, but regions with sparse coverage still show high uncertainty.
- At $n = 50$, the GP mean surface approximates the Branin-Hoo function much better, with low variance in sampled regions.
- At $n = 100$, the predicted mean is almost indistinguishable from the true function, and uncertainty is minimal throughout the domain.

These results demonstrate the flexibility and robustness of the Matérn kernel in capturing moderately smooth functions with limited samples, especially when coupled with an acquisition strategy like EI that actively explores high-uncertainty regions.

4.2 Results with Rotational Quadratic Kernel and Expected Improvement (EI)

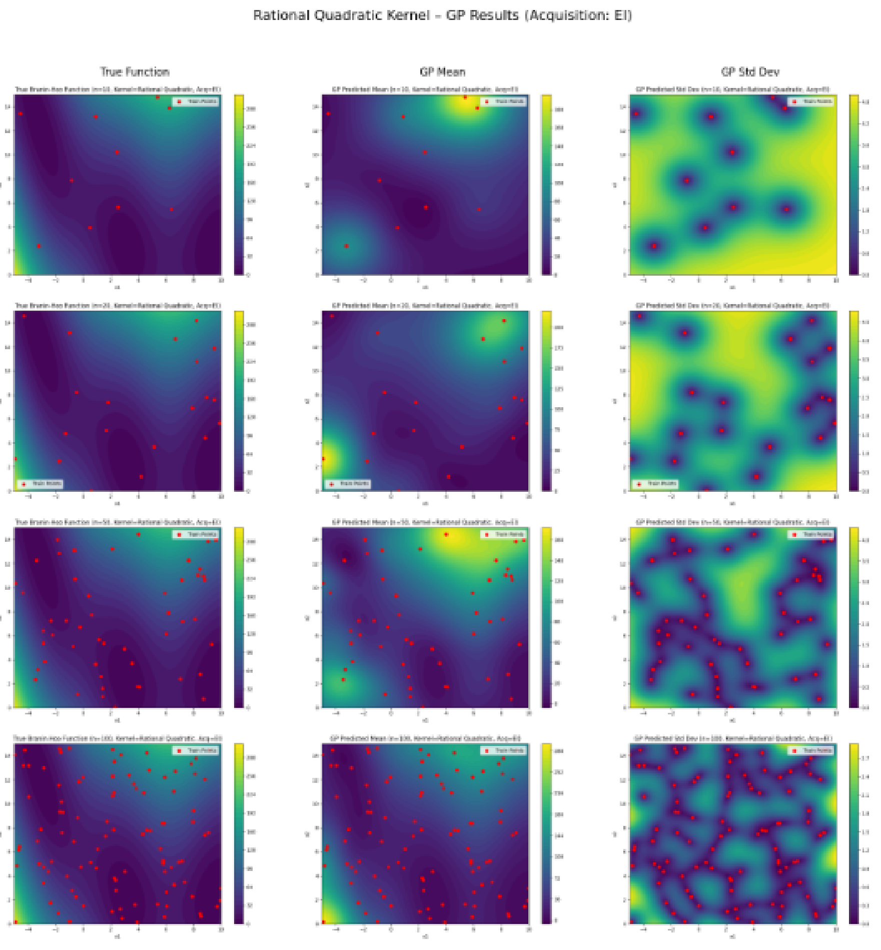

Figure 4

The above figure presents the Gaussian Process regression results using the Rational Quadratic kernel, under the Expected Improvement (EI) acquisition strategy. Each row corresponds to a different number of training samples $n \in \{10, 20, 50, 100\}$.

Observations:

- At $n = 10$, the predicted mean captures the global trend, but uncertainty is high, especially in unexplored regions.
- Increasing to $n = 20$, the model refines its prediction around sampled areas and better captures one of the Branin-Hoo valleys.
- With $n = 50$, both the predicted mean and uncertainty show significant improvement, closely matching the ground truth.
- At $n = 100$, the GP prediction closely resembles the true function across the entire domain, and uncertainty is minimized even in boundary regions.

Kernel Behavior: The Rational Quadratic kernel is a scale mixture of RBF kernels with different length scales, making it well-suited for functions with non-uniform smoothness. This property is evident in the way the model adapts to both smooth and sharp regions of the Branin-Hoo function.

4.3 Results with RBF Kernel and Expected Improvement (EI)

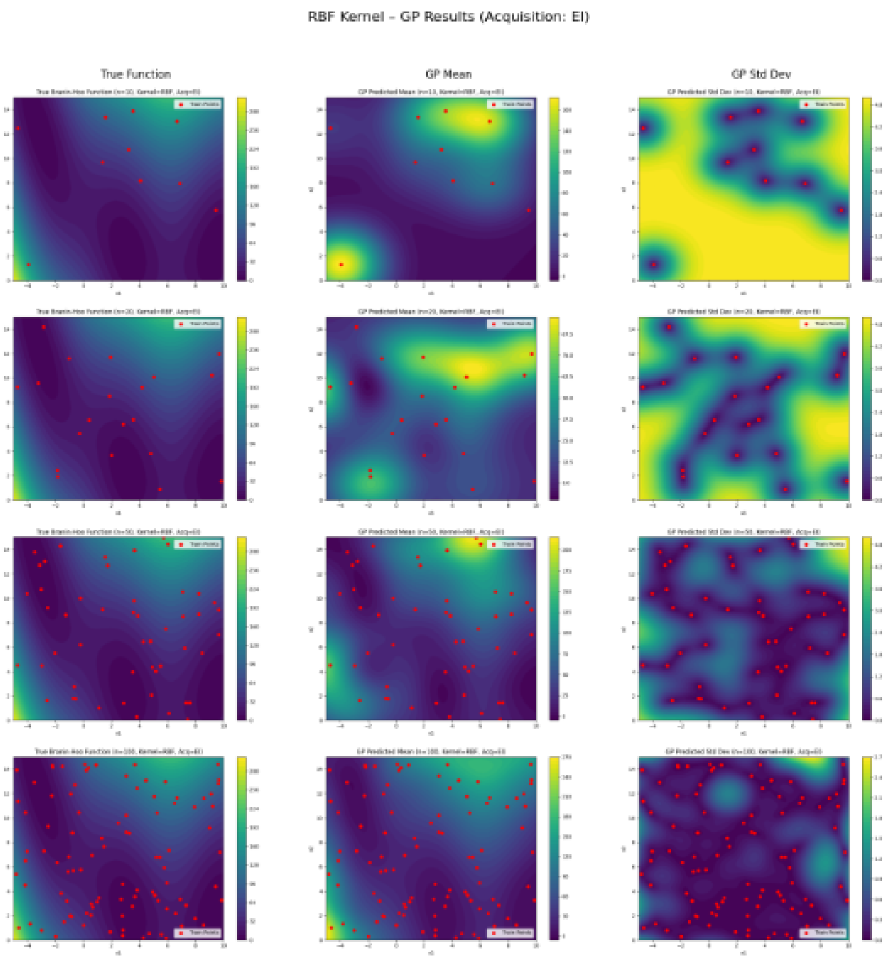

Figure 5

The above figure shows the GP regression results using the Radial Basis Function (RBF) kernel and the Expected Improvement (EI) acquisition function for four different training set sizes ($n = 10, 20, 50, 100$).

Observations:

- With only $n = 10$ points, the RBF kernel shows smooth interpolation but struggles to capture multimodal features of the Branin-Hoo function, especially in less-explored regions.
- At $n = 20$, it begins to approximate the main basin of the function, and the model gains confidence in regions closer to training points.
- For $n = 50$, the GP mean starts to resemble the global structure of the true function more accurately, with reduced uncertainty in most regions.
- At $n = 100$, the model provides a very close fit to the true function, and the uncertainty map shows almost no variance in densely sampled areas.

Kernel Behavior: The RBF kernel enforces a strong smoothness assumption, which works well for the Branin-Hoo function but may underfit in regions with sudden curvature. Nonetheless, it performs well with moderate sample sizes and produces reliable uncertainty estimates.

4.4 Results with Matérn Kernel and Probability of Improvement (PI)

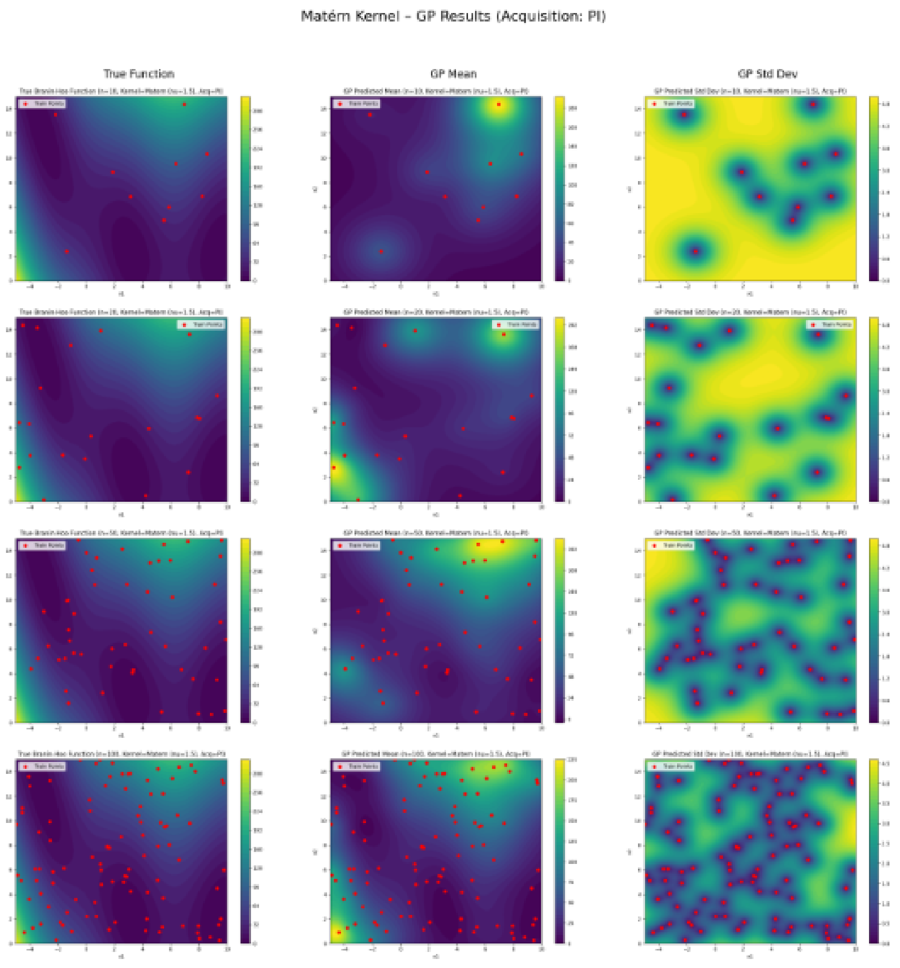

Figure 6

The above figure presents the Gaussian Process regression results using the Matérn kernel ($\nu = 1.5$) and the Probability of Improvement (PI) acquisition function. The figure shows outcomes for four training set sizes: $n = 10, 20, 50, 100$.

Observations:

- With $n = 10$, the GP mean reflects a rough estimate of the global trend, while uncertainty remains high in most regions, especially far from sampled points.
- At $n = 20$, the model begins to resolve the major basins of the Branin-Hoo function, and uncertainty reduces near data-dense regions.
- With $n = 50$, the GP mean closely matches the true surface, capturing finer curvature and local minima, while the uncertainty becomes more structured.
- By $n = 100$, the model accurately reconstructs the true function, and the uncertainty is mostly concentrated around unobserved regions.

Kernel Behavior: The Matérn kernel's flexibility allows it to generalize well with relatively fewer samples compared to more rigid kernels. Its ability to model moderately smooth functions is beneficial for capturing the Branin-Hoo landscape.

Effect of PI Acquisition: The PI acquisition strategy tends to exploit more aggressively than EI, favoring areas likely to improve over the current best. This results in more samples clustering near known optima, as seen by the red points in later rows. Consequently, uncertainty in those areas is reduced faster, but the exploration of less-visited regions is more limited than with EI.

4.5 Results with Rational Quadratic Kernel and Probability of Improvement (PI)

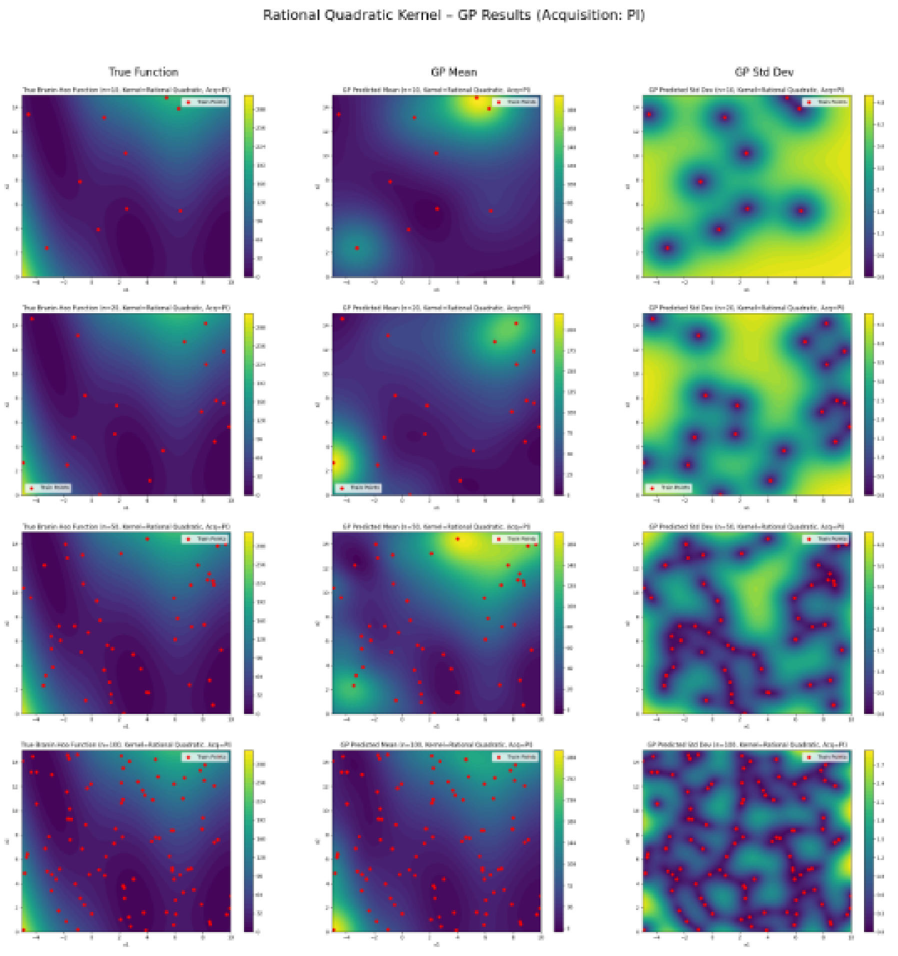

Figure 7

The above figure shows the Gaussian Process regression results using the Rational Quadratic kernel in combination with the Probability of Improvement (PI) acquisition strategy. Each row corresponds to an increasing number of training points: $n = 10, 20, 50, 100$.

Observations:

- With $n = 10$, the model begins to identify the general shape of the Branin-Hoo function, but high uncertainty exists across the domain.
- At $n = 20$, local improvements are visible and the predicted mean starts capturing the key modes of the function, especially near sampled points.
- With $n = 50$, the GP prediction becomes more detailed and confident in well-sampled areas, and uncertainty reduces accordingly.
- At $n = 100$, the predicted mean very closely matches the true function across the domain, with low uncertainty in most regions.

Kernel Behavior: The Rational Quadratic kernel, being a scale mixture of RBF kernels, effectively handles non-stationarity in the data. It performs particularly well when the function exhibits varying degrees of smoothness across regions, which aligns well with the Branin-Hoo function's structure.

Effect of PI Acquisition: The PI strategy tends to favor exploitation by repeatedly selecting regions near the current best value. This can be seen in the clustering of red points near known optima. While this leads to fast uncertainty reduction in promising areas, it may leave some regions underexplored compared to EI.

4.6 Results with RBF Kernel and Probability of Improvement (PI)

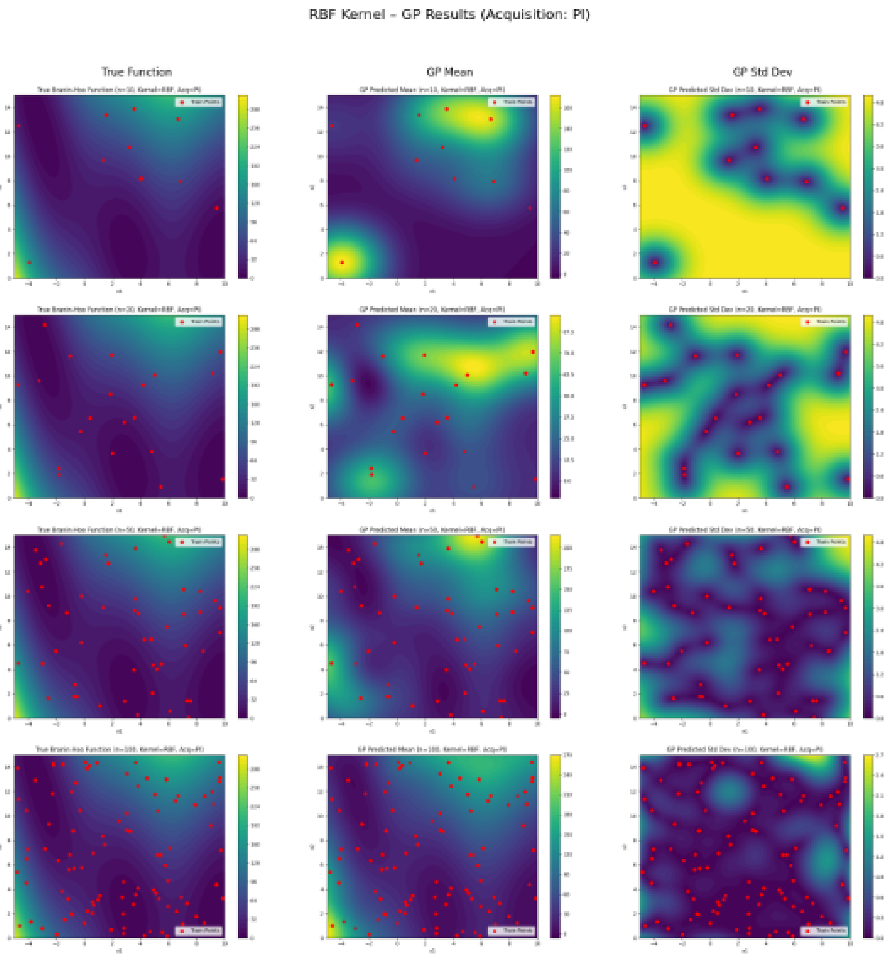

Figure 8

The above figure illustrates the performance of Gaussian Process regression using the Radial Basis Function (RBF) kernel in combination with the Probability of Improvement (PI) acquisition strategy. Each row corresponds to a different number of training samples: $n = 10, 20, 50, 100$.

Observations:

- At $n = 10$, the GP is able to learn only the broadest features of the function. The uncertainty is high throughout the domain, except near the sampled regions.
- For $n = 20$, the GP begins to model one of the function's basins accurately. Uncertainty reduces slightly but remains high in underexplored areas.

- With $n = 50$, the predicted mean surface becomes more structured and similar to the true function. More of the global shape is captured.
- At $n = 100$, the model's prediction is very close to the true Branin-Hoo landscape. Uncertainty is highly localized and minimal across the domain.

Kernel Behavior: The RBF kernel assumes high smoothness and performs well on functions like Branin-Hoo. However, it may struggle to represent abrupt changes or sharp curvature without sufficient data. The smooth extrapolation makes the model conservative in unexplored regions.

Effect of PI Acquisition: The PI strategy drives exploitation around known optima. As a result, training points are heavily concentrated near regions of low function value. While this accelerates convergence in those areas, it limits exploration elsewhere — evident in the uncertainty plots which show unvisited zones remaining high in variance.

5 Conclusion

In this project, we explored two key aspects of probabilistic modeling using energy-based and Gaussian Process methods.

Task 0: Energy-Based Models (EBMs). We evaluated a pre-trained energy regressor model on synthetic data and obtained a mean squared error (MSE) of approximately 288.15. Two Langevin dynamics-based MCMC sampling algorithms—Metropolis-Adjusted Langevin Algorithm (MALA) and Unadjusted Langevin Algorithm (ULA)—were implemented to sample from the implicit distribution defined by the EBM. MALA provided higher-quality samples at the cost of computational overhead due to its Metropolis-Hastings correction step, while ULA was faster but relied on careful step-size tuning. Visualizations using t-SNE highlighted the structural differences in sampling behavior, with MALA covering the sample space more uniformly than ULA.

Task 1: Gaussian Process Regression. We implemented Gaussian Process regression with three different kernels: Radial Basis Function (RBF), Matérn ($\nu = 1.5$), and Rational Quadratic. Experiments were conducted with varying sample sizes and two acquisition strategies—Expected Improvement (EI) and Probability of Improvement (PI). The results demonstrated that:

- The **RBF kernel** provided smooth and stable approximations, but required more samples to capture non-stationary behavior.
- The **Matérn kernel** balanced smoothness and flexibility, offering robust performance across low and high sample sizes.
- The **Rational Quadratic kernel** adapted well to varying function curvature and performed well even at smaller sample sizes.

The EI acquisition function maintained a balance between exploration and exploitation, while PI tended to over-exploit known optima, resulting in clustered samples and slower global convergence.

Overall Insights. This project highlights the complementary strengths of energy-based models for flexible density estimation and Gaussian Processes for uncertainty-aware regression and optimization. Choosing the right kernel and acquisition strategy significantly impacts performance, especially in sample-efficient settings. These methods are foundational tools for Bayesian optimization, active learning, and probabilistic modeling.

6 Contributions

The project was completed collaboratively with well-defined responsibilities assigned to each team member:

- **Pinak:** Implemented Task 2 involving Gaussian Process regression with multiple kernels and acquisition strategies. Also developed the visualization code for generating summary plots for all combinations of kernel and acquisition functions.
- **Danish:** Worked on Task 0 and Task 1. He implemented and tested the energy-based model evaluation and the two Langevin dynamics-based MCMC samplers (ULA and MALA), including sample generation and t-SNE visualizations.
- **Aansh:** Focused on compiling the final project report, writing detailed analyses, formatting LaTeX content, and organizing the submission.

All members contributed to code review, debugging, and discussion of results.