CS726 : Advanced Machine Learning

Indian Institute Of Technology, Bombay

Programming Assignment 4

Author:
Arijeet Mondal : 24m0812
Daksh Goyal : 24m0756

Instructor:
Prof. Sunita Sarawagi

April 17, 2025

Contents

1 Task 0: Environment Setup and Result Reproduction

We begin by setting up a Python environment with only the permitted libraries: PyTorch, NumPy, and Matplotlib. The script `get_results.py` defines a deep neural network model, `EnergyRegressor`, that estimates an energy value $E_\theta(x)$ for a given input vector $x \in \mathbb{R}^{784}$. This model acts as the energy function in an Energy-Based Model (EBM) framework.

The `EnergyRegressor` is initialized and its weights are loaded from the provided file `trained_model_weights.pth`. The model architecture is shown below:

```
EnergyRegressor(
  (net): Sequential(
    (0): Linear(in_features=784, out_features=4096, bias=True)
    (1): ReLU(inplace=True)
    (2): Linear(in_features=4096, out_features=2048, bias=True)
    (3): ReLU(inplace=True)
    (4): Linear(in_features=2048, out_features=1024, bias=True)
    (5): ReLU(inplace=True)
    (6): Linear(in_features=1024, out_features=512, bias=True)
    (7): ReLU(inplace=True)
    (8): Linear(in_features=512, out_features=256, bias=True)
    (9): ReLU(inplace=True)
    (10): Linear(in_features=256, out_features=128, bias=True)
    (11): ReLU(inplace=True)
    (12): Linear(in_features=128, out_features=64, bias=True)
    (13): ReLU(inplace=True)
    (14): Linear(in_features=64, out_features=32, bias=True)
    (15): ReLU(inplace=True)
    (16): Linear(in_features=32, out_features=16, bias=True)
    (17): ReLU(inplace=True)
    (18): Linear(in_features=16, out_features=8, bias=True)
    (19): ReLU(inplace=True)
    (20): Linear(in_features=8, out_features=4, bias=True)
    (21): ReLU(inplace=True)
    (22): Linear(in_features=4, out_features=2, bias=True)
    (23): ReLU(inplace=True)
    (24): Linear(in_features=2, out_features=1, bias=True)
  )
)
```

A custom PyTorch `Dataset` class is used to load the synthetic test dataset (`A4_test_data.pt`) containing input vectors and corresponding energy values. It has 100000 samples. The model's performance is evaluated using Mean Squared Error (MSE) on this dataset via the `Tester` class.

The final test loss **288.1554** is used as a baseline for subsequent sampling algorithm evaluations.

2 Task 1: MCMC Sampling Implementation

In this task, we implemented two MCMC algorithms to sample from a probability distribution defined by a pre-trained neural network energy function $E(X)$. The target distribution is given by $p(X) \propto \exp(-E(X))$.

Algorithm 1 uses Metropolis-adjusted Langevin dynamics. At each step, it computes the gradient of the energy function, proposes a new sample using Langevin dynamics with added Gaussian noise, and applies a Metropolis-Hastings acceptance criterion to decide whether to accept the proposed sample.

Algorithm 2 uses Unadjusted Langevin dynamics, where each new sample is computed directly using the gradient and Gaussian noise without a Metropolis correction.

The initial sample X_0 is initialized as a random Gaussian Noise sample. We generated samples after a burn-in period and visualized the high-dimensional samples using t-SNE. The sampling time and burn-in time for both algorithms were recorded for performance comparison.

2.1 Report

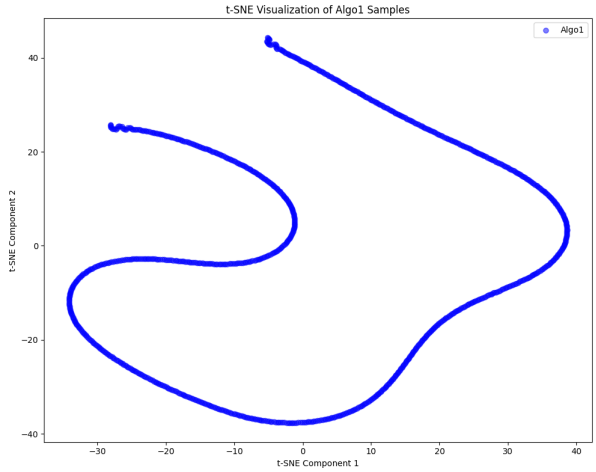
The following are reports on the experiments done on the two algorithms with different numbers of samples, burn-in times, and tau values.

2.1.1 Comparison

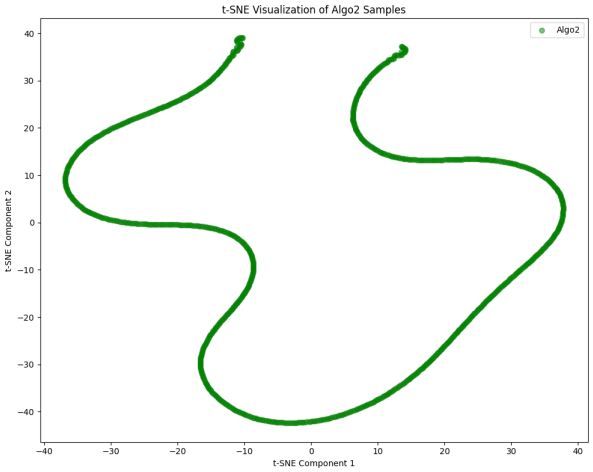
Table 1: Comparison of different Algorithms

Algorithm	**Samples**	**Burn-in Count**	**Tau**	**Burn-in Time (s)**	**Total Time (s)**
Algorithm 1	1000	100	0.01	0.6185	5.8457
Algorithm 2	1000	100	0.01	0.2266	2.5528
Algorithm 1	10000	2500	0.001	13.5855	74.8149
Algorithm 2	10000	2500	0.001	6.0685	33.0869
Algorithm 1	50000	5000	0.001	28.6487	312.0505
Algorithm 2	50000	5000	0.001	11.3556	146.7482

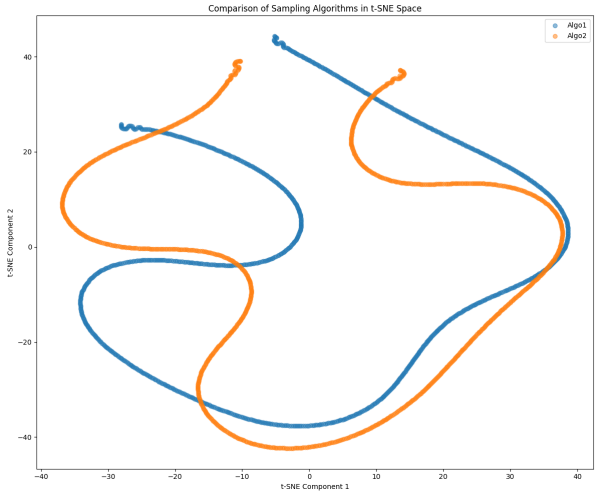
2.1.2 1000 samples



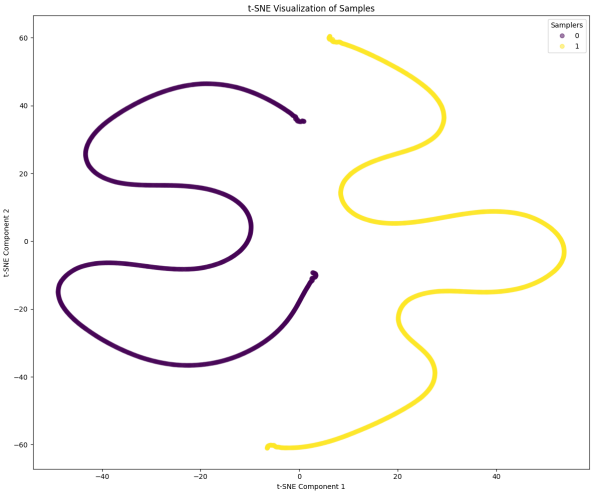
(a) t-SNE: Algo1 Samples



(b) t-SNE: Algo2 Samples



(c) Overlay: Algo1 and Algo2

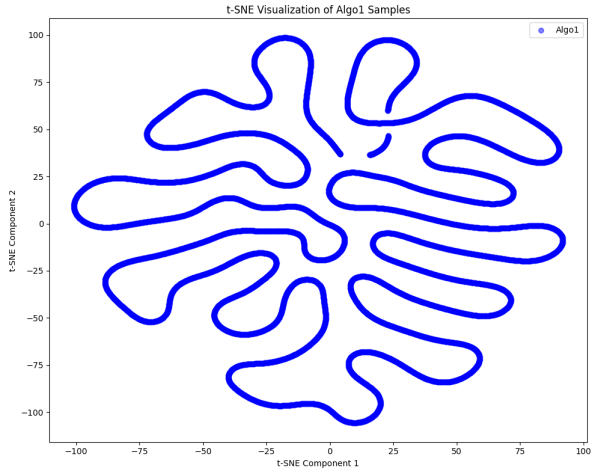


(d) Combined t-SNE with Labels

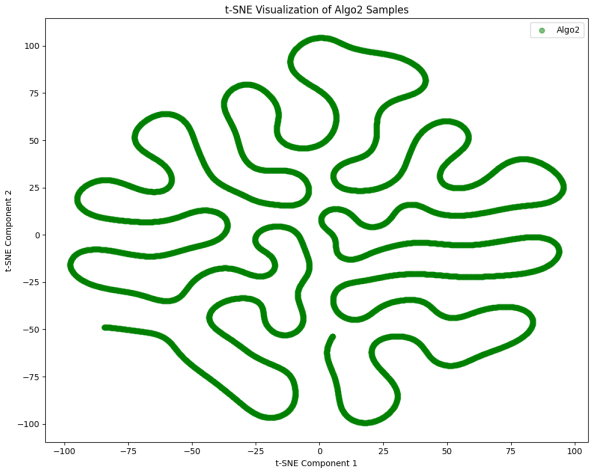
Figure 1: t-SNE visualization comparing sampling algorithms.

Fig 1 shows the t-SNE samples for both the algorithms, along with the overlays and combined t-SNE. Burn-in count is 100 and τ is 0.01.

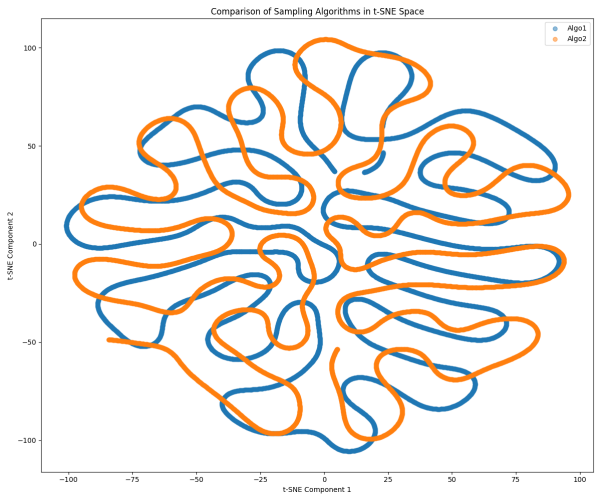
2.1.3 10000 samples



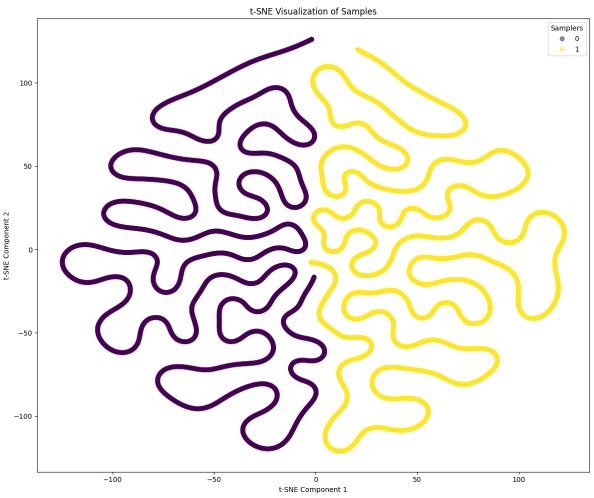
(a) t-SNE: Algo1 Samples



(b) t-SNE: Algo2 Samples



(c) Overlay: Algo1 and Algo2

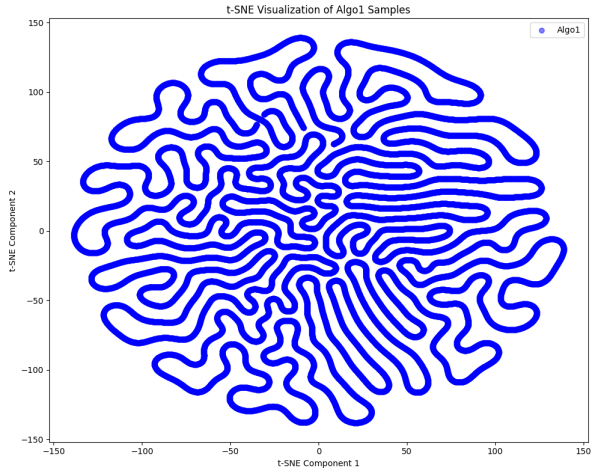


(d) Combined t-SNE with Labels

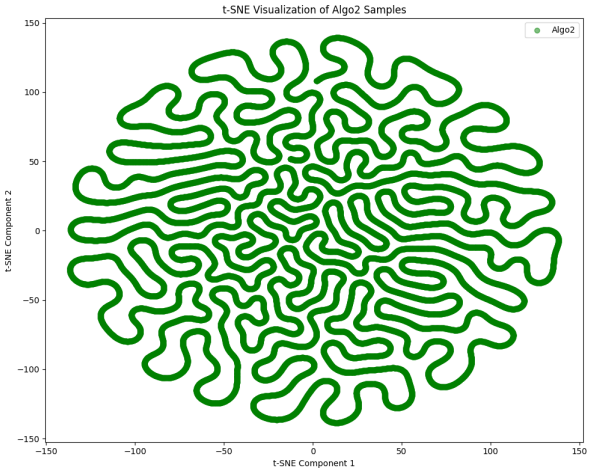
Figure 2: t-SNE visualization comparing sampling algorithms.

Fig 2 shows the t-SNE samples for both the algorithms, along with the overlays and combined t-SNE. Burn-in count is 2500 and τ is 0.001.

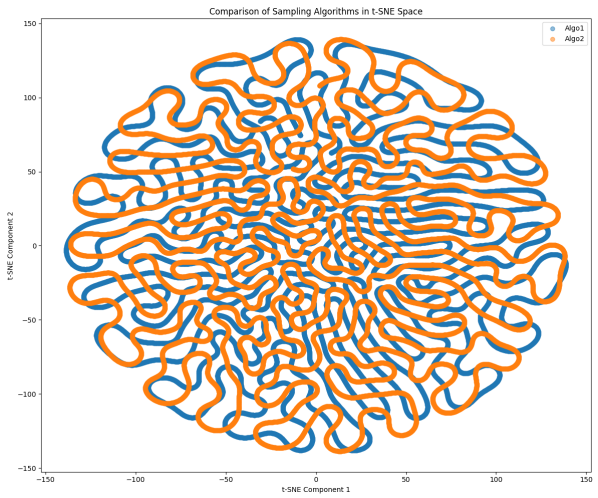
2.1.4 50000 samples



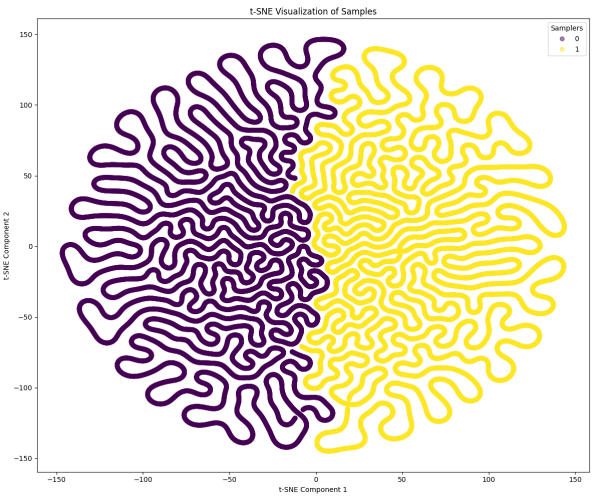
(a) t-SNE: Algo1 Samples



(b) t-SNE: Algo2 Samples



(c) Overlay: Algo1 and Algo2



(d) Combined t-SNE with Labels

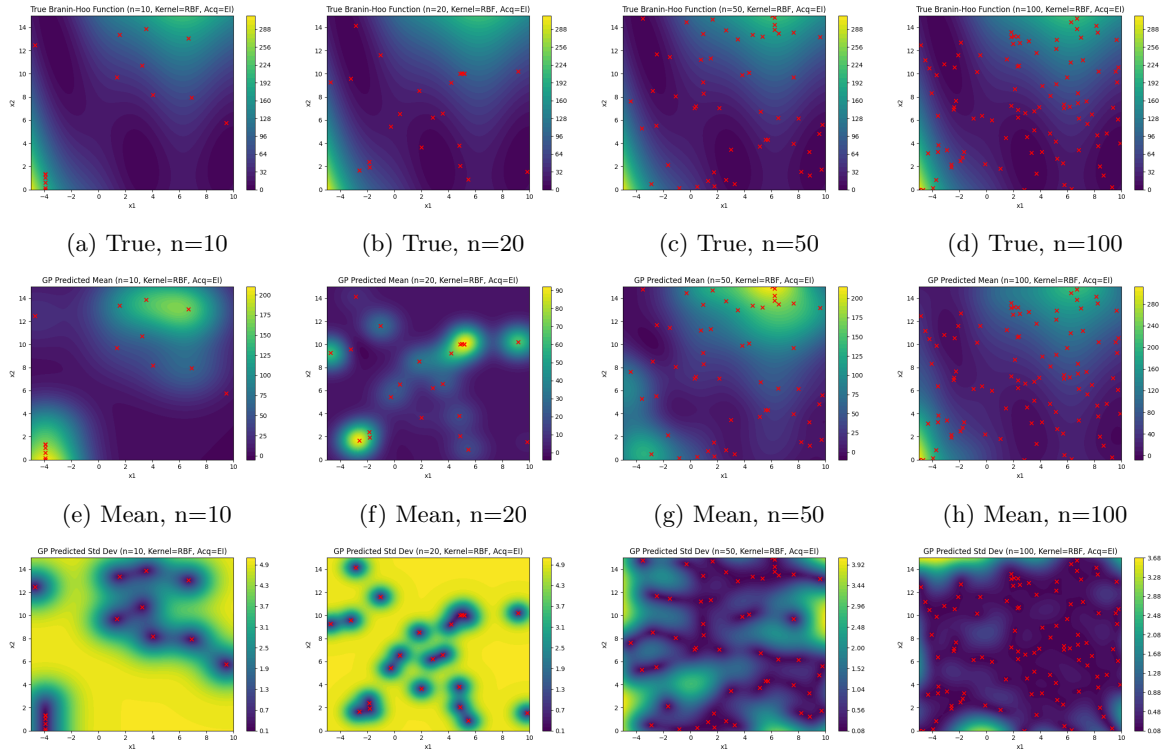
Figure 3: t-SNE visualization comparing sampling algorithms.

Fig 3 shows the t-SNE samples for both the algorithms, along with the overlays and combined t-SNE. Burn-in count is 5000 and τ is 0.001.

3 Task 2: Approximating a Black-Box Function Using Gaussian Processes

3.1 RBF Kernel

3.1.1 Expected Improvement (EI)



(a) True, n=10 (b) True, n=20 (c) True, n=50 (d) True, n=100

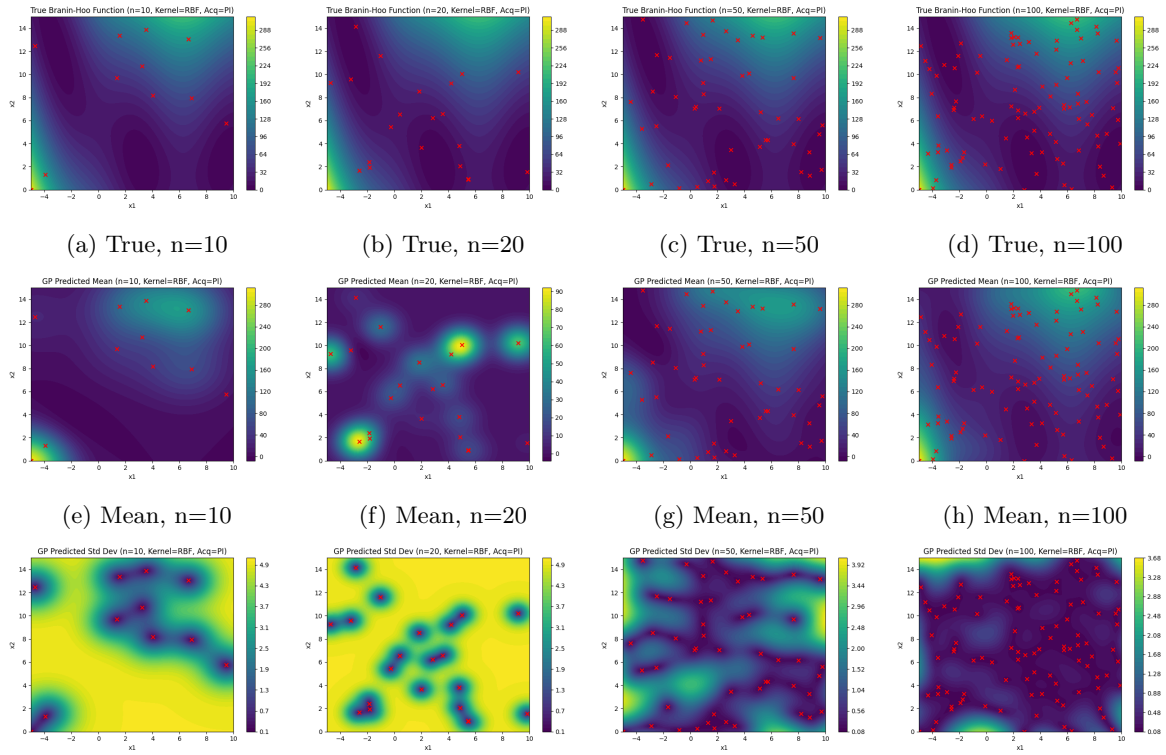
(e) Mean, n=10 (f) Mean, n=20 (g) Mean, n=50 (h) Mean, n=100

(i) Std, n=10 (j) Std, n=20 (k) Std, n=50 (l) Std, n=100

Figure 4: RBF Kernel — Acquisition: Expected Improvement (EI)

- **n = 10**
  The GP captures just one basin with an overly flat surface elsewhere, indicating underfitting. Uncertainty is high across most of the domain, only reducing narrowly around the few sampled points.
- **n = 20**
  A second minimum starts to emerge, but transitions remain overly smooth and shallow. Uncertainty decreases slightly but remains large over sharp ridges and unexplored regions.
- **n = 50**
  All three minima are visible, though the surface is still smoothened, particularly around steep valleys. Uncertainty drops substantially near sampled regions, with moderate levels remaining in high-curvature areas.
- **n = 100**
  The predictive mean is highly accurate, with correctly shaped and located minima. Uncertainty is low and uniform, showing strong confidence in the approximation.

3.1.2 Probability of Improvement (PI)



(a) True, n=10 (b) True, n=20 (c) True, n=50 (d) True, n=100

(e) Mean, n=10 (f) Mean, n=20 (g) Mean, n=50 (h) Mean, n=100

(i) Std, n=10 (j) Std, n=20 (k) Std, n=50 (l) Std, n=100

Figure 5: RBF Kernel — Acquisition: Probability of Improvement (PI)

- **n = 10**
  The model heavily focuses on one basin, ignoring the rest of the domain, which limits exploration. Uncertainty stays very high outside the sampled region.

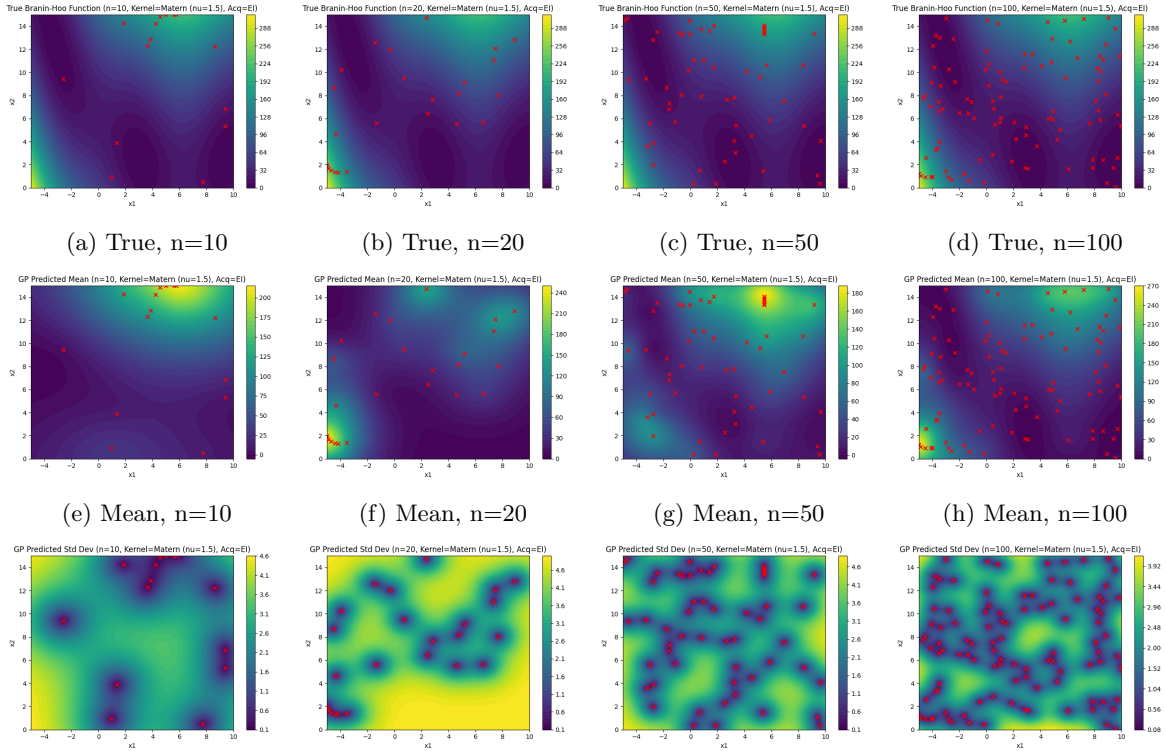
- **n = 20**
  Slight progress is made, but the model still clusters around the initial optimum. Variance remains broad, with minimal reduction outside the primary sampling zone.

- **n = 50**
  All minima begin to surface, but only after excessive focus on the first one. Variance shrinks around densely sampled areas, yet overconfidence can persist in underexplored regions.

- **n = 100**
  The mean aligns well with the true function, though exploration bias is visible. Uncertainty becomes uniformly small, albeit with delayed convergence compared to EI.

3.2 Matérn Kernel

3.2.1 Expected Improvement (EI)



(a) True, n=10 (b) True, n=20 (c) True, n=50 (d) True, n=100

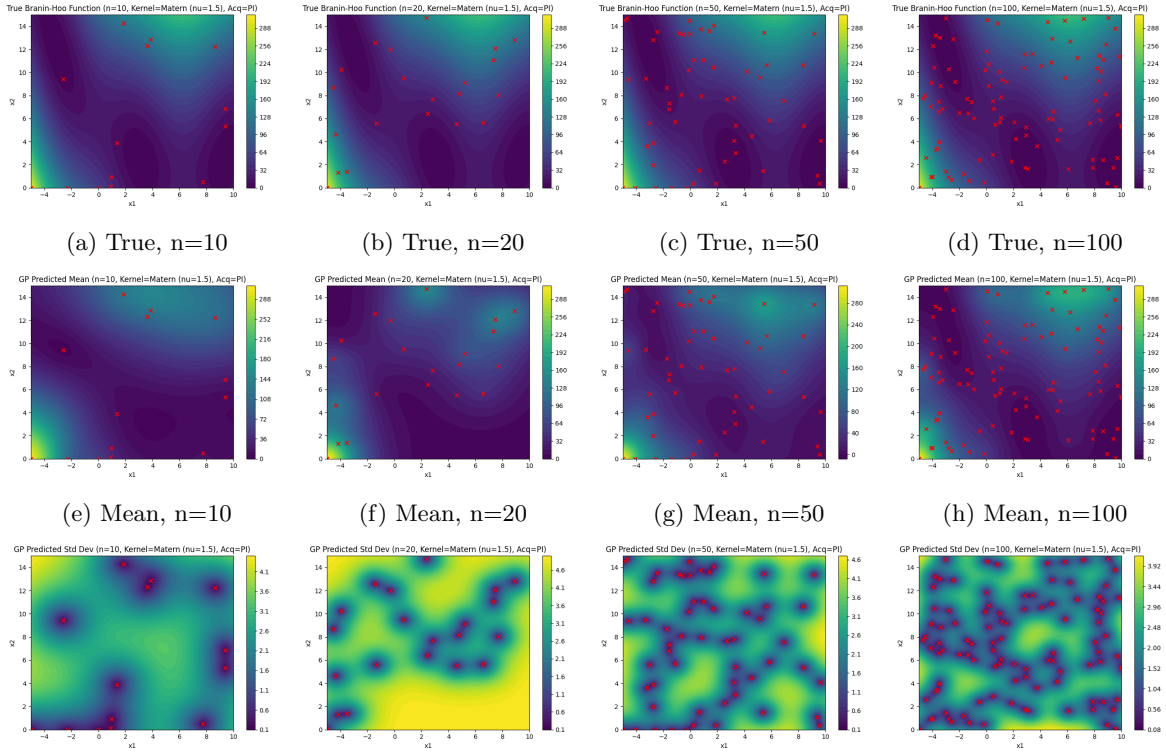
(e) Mean, n=10 (f) Mean, n=20 (g) Mean, n=50 (h) Mean, n=100

(i) Std, n=10 (j) Std, n=20 (k) Std, n=50 (l) Std, n=100

Figure 6: Matérn Kernel — Acquisition: Expected Improvement (EI)

- **n = 10**
  Two minima are partially captured, and the GP displays sharper, more realistic curvature than RBF. Uncertainty is high overall but falls off more sharply around sampled steep ridges.
- **n = 20**
  The model now reveals a third minimum, with improved fidelity in ridge and valley transitions. Uncertainty is well localized, reducing significantly along sampled contours.
- **n = 50**
  All minima and ridges are modeled accurately, with sharp features preserved. Uncertainty is minimal across most of the domain, except at distant boundaries.
- **n = 100**
  The mean mirrors the Branin–Hoo function with near-perfect fidelity. Uncertainty is uniformly low, indicating full confidence in predictions.

3.2.2 Probability of Improvement (PI)



(a) True, n=10 (b) True, n=20 (c) True, n=50 (d) True, n=100

(e) Mean, n=10 (f) Mean, n=20 (g) Mean, n=50 (h) Mean, n=100

(i) Std, n=10 (j) Std, n=20 (k) Std, n=50 (l) Std, n=100

Figure 7: Matérn Kernel — Acquisition: Probability of Improvement (PI)

- **n = 10**
  The GP tends to explore slightly more than RBF+PI, capturing hints of a second basin. Variance is still high in unexplored regions but transitions more smoothly.

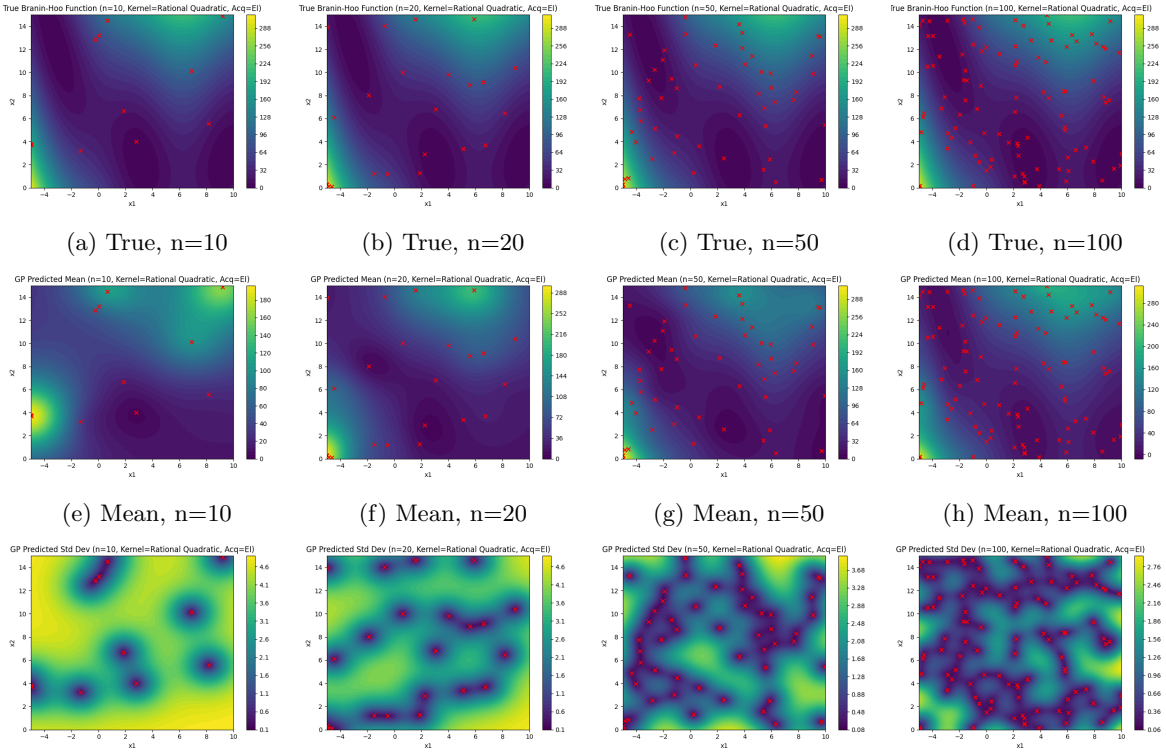
- **n = 20**
  The second minimum becomes clearer, with more even sample distribution. Uncertainty reduces around valleys while staying moderate elsewhere.

- **n = 50**
  All minima are resolved, and the structure resembles the EI case. Variance patterns closely follow EI, with slightly higher levels in remote regions.

- **n = 100**
  The GP closely approximates the true function. Uncertainty becomes uniformly low across the space.

3.3 Rational Quadratic Kernel

3.3.1 Expected Improvement (EI)



(a) True, n=10 (b) True, n=20 (c) True, n=50 (d) True, n=100

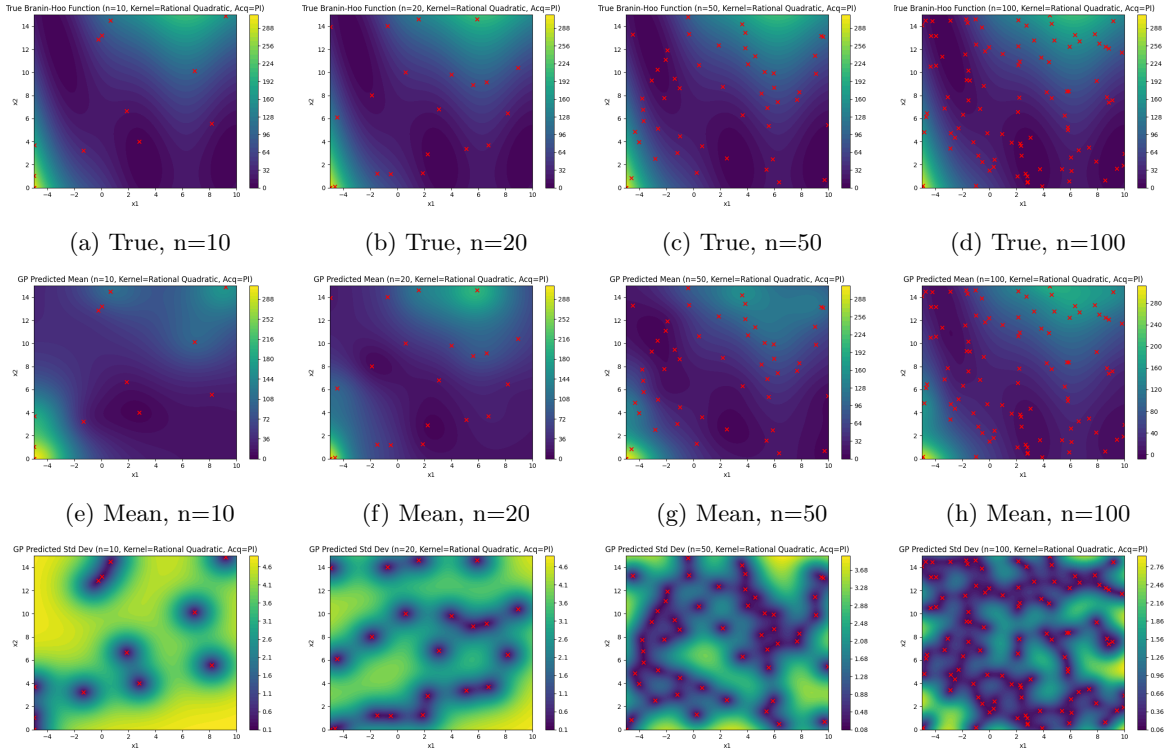
(e) Mean, n=10 (f) Mean, n=20 (g) Mean, n=50 (h) Mean, n=100

(i) Std, n=10 (j) Std, n=20 (k) Std, n=50 (l) Std, n=100

Figure 8: Rational Quadratic Kernel — Acquisition: Expected Improvement (EI)

- **n = 10**
  The primary basin is captured well, and a second minimum begins to show due to the kernel's variable length-scale. Uncertainty is high but adapts better than RBF, showing smoother decay.
- **n = 20**
  Two minima are now distinct, with ridges rendered more sharply than RBF. Uncertainty reduces more evenly across the domain, especially in curved areas.
- **n = 50**
  All three minima and their connecting ridges are accurately approximated. Variance is low near samples but slightly elevated in sharp transitions.
- **n = 100**
  The GP reaches near-perfect mean approximation with smooth yet detailed structure. Uncertainty is minimal with slight increases only near extrapolated edges.

3.3.2 Probability of Improvement (PI)



(a) True, n=10 (b) True, n=20 (c) True, n=50 (d) True, n=100

(e) Mean, n=10 (f) Mean, n=20 (g) Mean, n=50 (h) Mean, n=100

(i) Std, n=10 (j) Std, n=20 (k) Std, n=50 (l) Std, n=100

Figure 9: Rational Quadratic Kernel — Acquisition: Probability of Improvement (PI)

- **n = 10**
  One basin dominates, but due to the kernel's flexibility, subtle hints of other regions appear. Uncertainty is still high but decays gradually around the sampled zone.

- **n = 20**
  A second minimum becomes visible with improved contour representation. Variance falls off more gracefully than in RBF+PI, with clearer spatial structure.

- **n = 50**
  All critical points are modeled correctly, and the transitions are smooth. Uncertainty shrinks effectively, with slightly higher levels than EI in a few areas.

- **n = 100**
  The GP converges to the true function with strong fidelity. Variance is low across the board, showing robust model certainty.

3.4 Final Summary

RBF

Pros: Extremely smooth surfaces; very low variance once heavily sampled.

Cons: Oversmooths sharp ridges; needs large n to reveal multiple minima; PI is overly exploitative.

Matérn ($\nu = 1.5$)

Pros: Excellent at modeling sharp features; tight, localized uncertainty; EI finds all minima quickly.

Cons: Slightly rougher priors can lead to small oscillations in very flat regions.

Rational Quadratic

Pros: Balances RBF's smoothness with Matérn's adaptability; handles multi-scale features elegantly; EI+RQ recovers all minima by $n \approx 50$ with crisp yet smooth contours.

Cons: Uncertainty reduction is a bit slower around the steepest transitions compared to Matérn; PI still somewhat exploitative at very low n.

Acquisition Functions

EI remains the safest choice for global modeling across all kernels.

PI can be useful for rapid local refinement—best paired with Matérn or Rational Quadratic to mitigate its myopia.

Sample Efficiency & Recommendation

$n \leq 20$: Only Matérn+EI or RQ+EI give any hope of finding multiple minima.

$n \approx 50$: RQ+EI and Matérn+EI both achieve near-true reconstructions; RQ+PI is a close third.

$n \approx 100$: All methods converge, but Matérn+EI still edges out slightly on uncertainty localization, while RQ+EI offers the smoothest fidelity.

Overall Best-in-Class:

For aggressive discovery of an unknown, multimodal surface: Matérn ($\nu = 1.5$) + EI.

For a balance of smooth interpolation and adaptability at moderate sample budgets: Rational Quadratic + EI.