**Model.py**

✅ This model.py now:

* Supports **dynamic acquisition functions** (ucb, ei, pi)
* Allows easy switching between **Matern 5/2 kernel** and **RBF kernel**
* Handles **cold start** (no training points yet)
* Includes **robust fallback if optimizer fails**

**Utils.py**

✅ Now:

* **When plotting**, you will **automatically get a comment**: good fit or poor fit.
* **After every new point**, your system will **log**:
  + Acquisition strategy
  + Kernel used
  + Best value observed
  + Iteration number

✅ Helps in tracking and reporting!

**Main.py**

✅ main.py now:

* Loops through all 8 functions automatically
* Decides acquisition + kernel smartly based on number of observations
* Suggests next input
* Plots GP prediction and checks model fit quality
* Logs iteration details into a CSV file for each function

**Plot\_convergence.py**

✅ **What This Does:**

* Reads the logs/ folder
* For each function:
  + Plots **iteration number** (week) vs **best output value so far**
  + Saves the plot to plots/convergence\_func\_{func\_id}.png
* Helps you **visualize your model’s learning progress over time** 📈

✅ You can run this **any time** after a few weeks to track improvement!

**Function 1: Searching for Contamination Sources**

**Traits**: Sparse signal, mostly zero, two distinct peaks (multi-modal), narrow peaks.

**Strategy Considerations**:

* Use **UCB** early on with **high exploration** (kappa=2.5+) to cover space.
* Later consider **EI**, but avoid early exploitation — you may miss the second peak.
* Be cautious of zero outputs — they can mislead GP into under-exploring.
* You could also try **two GPs** later if you see two peaks emerging.

✅ Set kappa=2.5 and consider plotting regularly to track unexplored regions.

**Function 2: Noisy ML Model Optimization**

**Traits**: Highly noisy, multiple local optima, model initialization sensitivity.

**Strategy Considerations**:

* Stick with **UCB** longer — you want to explore more to average out noise.
* Consider adding **noise\_level parameter** to GP kernel (currently WhiteKernel() does this).
* You may want to **retrain with a rolling window** of the best N points to prevent overfitting to noise.

✅ Consider increasing WhiteKernel(noise\_level=0.1) to let GP trust less in early points.

**Function 3: Drug Discovery (minimize adverse reaction)**

**Traits**: 3D, one variable might be irrelevant, goal is close to zero.

**Strategy Considerations**:

* Watch for GP fitting quality — one flat dimension may confuse it.
* Track how well GP models uncertainty — it should shrink only on relevant dimensions.
* Use **EI or PI** earlier than usual if you spot a minimum cluster.

✅ Try removing dimensions (manually or via variance analysis) if you suspect one is flat.

**Function 4: Fast but Inaccurate Modelling**

**Traits**: Expensive surrogate approximation, lots of local optima, 4D.

**Strategy Considerations**:

* Use **EI** from the beginning (exploitation is okay since this is an approximation).
* Consider **multiple restarts** (increase n\_restarts) in suggest\_next() to avoid local traps.

✅ Try n\_restarts=20 to make acquisition optimization more robust.

**Function 5: Chemical Reaction Yield**

**Traits**: 4D, unimodal, smoother behavior.

**Strategy Considerations**:

* Very GP-friendly.
* Start with **UCB**, switch to **EI** quickly.
* Focus on convergence and exploitation.

✅ Your default strategy will likely perform very well here.

**Function 6: Cake Recipe (multi-objective)**

**Traits**: 5D, multiple competing objectives combined into one score.

**Strategy Considerations**:

* Outputs may have subtle tradeoffs.
* GP will behave reasonably, but don't over-tune.
* Consider logging individual components (if available later).

✅ No changes needed now; stick to UCB→EI transition.

**Function 7: ML Hyperparameter Tuning**

**Traits**: 6D, known models, possibly researchable.

**Strategy Considerations**:

* You could look up common hyperparameter defaults (as a meta prior).
* Start with **default hyperparameter values** and **then explore**.
* If the surface is known to be bumpy, use more acquisition restarts.

✅ Consider using known good values as starting points in addition to GP.

**Function 8: High-Dimensional Black-Box**

**Traits**: 8D, curse of dimensionality, hard to optimize, local search preferred.

**Strategy Considerations**:

* Avoid pure exploration — too expensive in 8D.
* Use **EI or PI** early and aggressively.
* Use **lower number of restarts**, and maybe switch to **local search** mode (greedy) near the end.

✅ Your model should exploit fast and stick to promising regions once found.