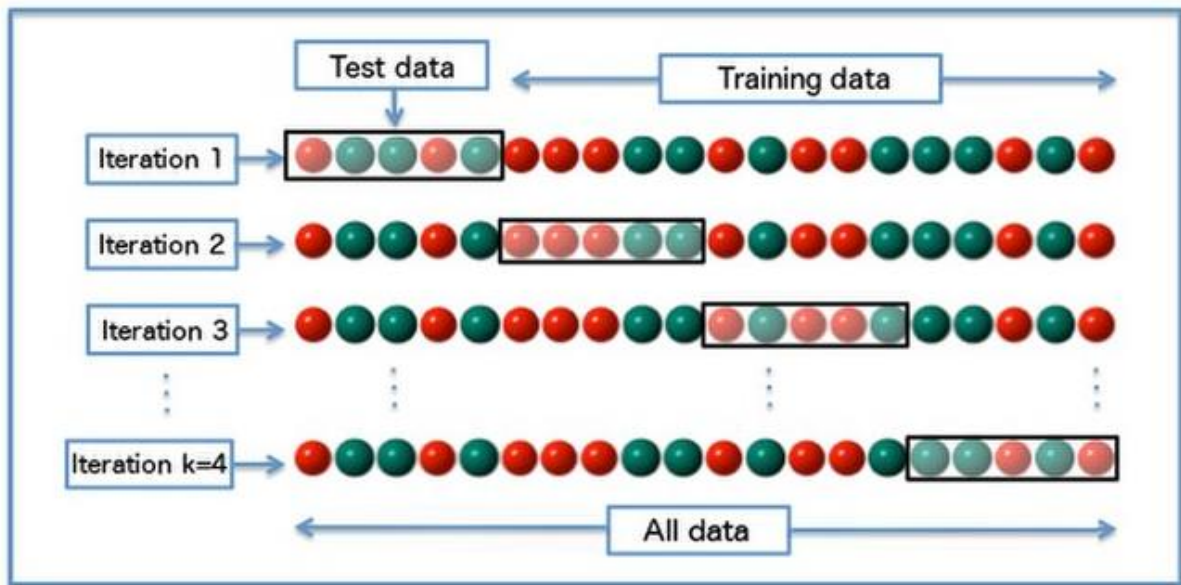
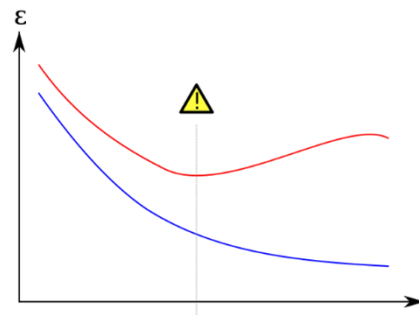
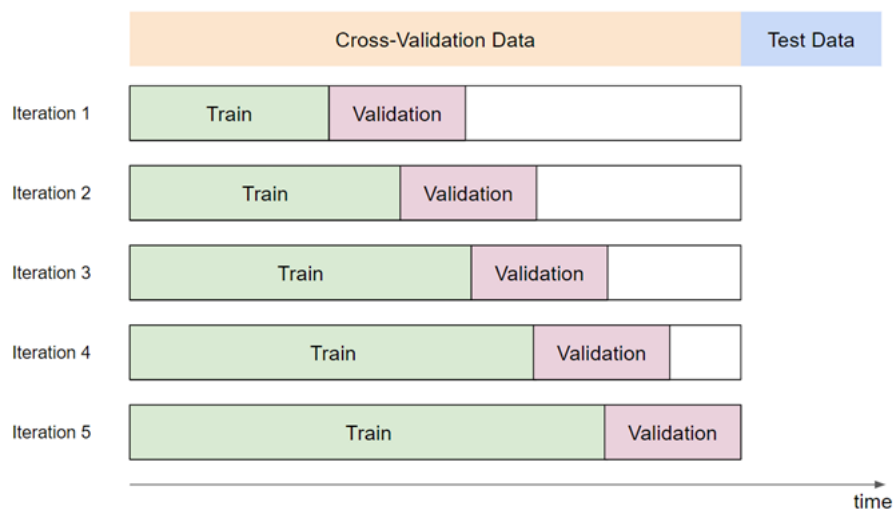
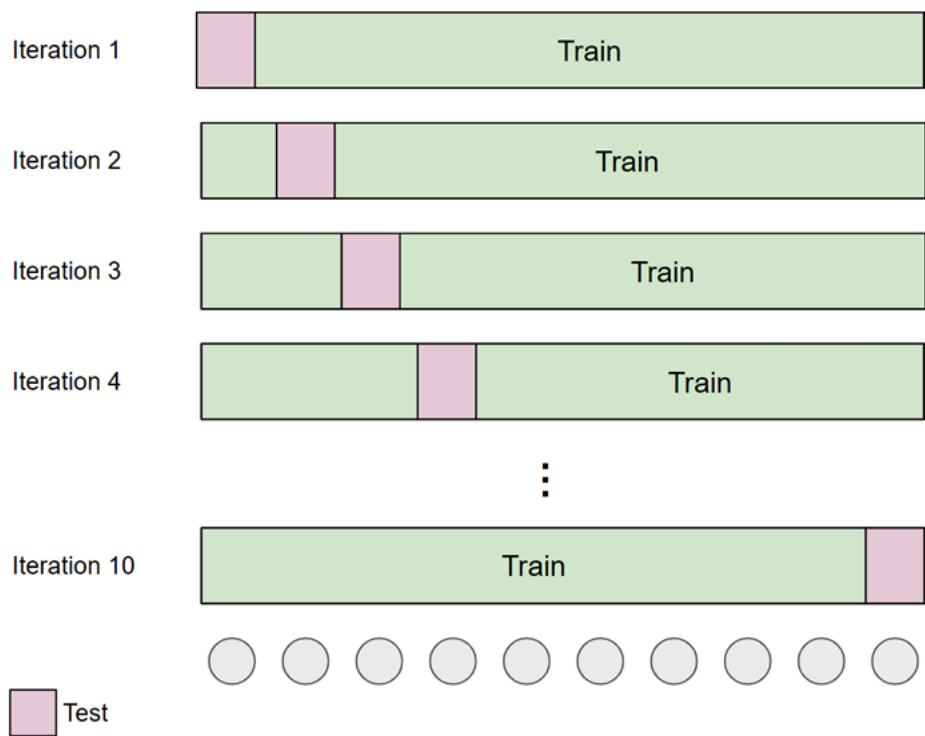


## Chapter 1: Evaluating Machine Learning Models

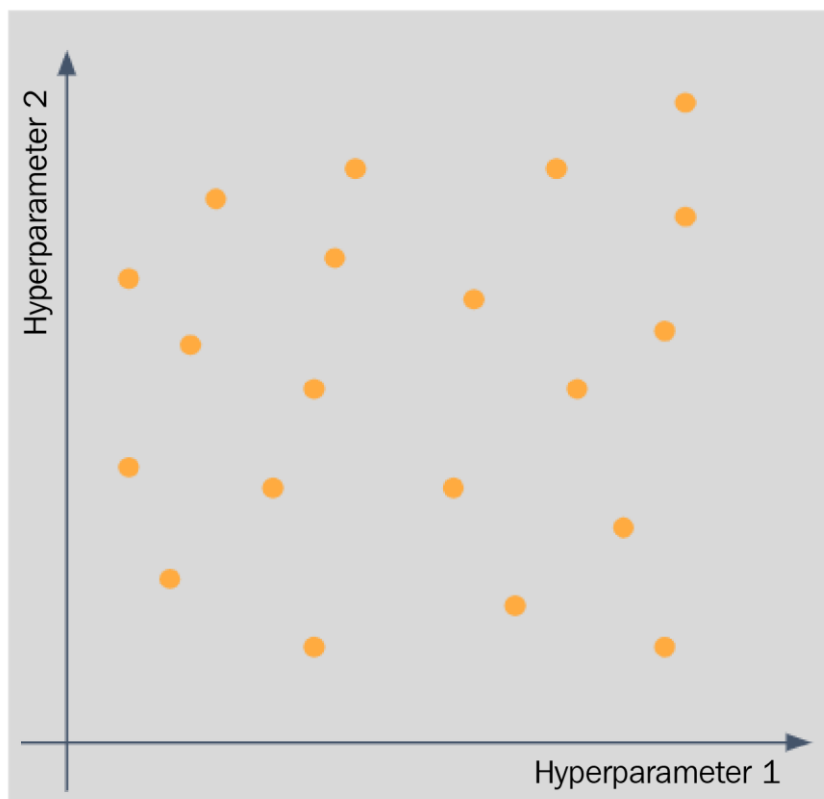
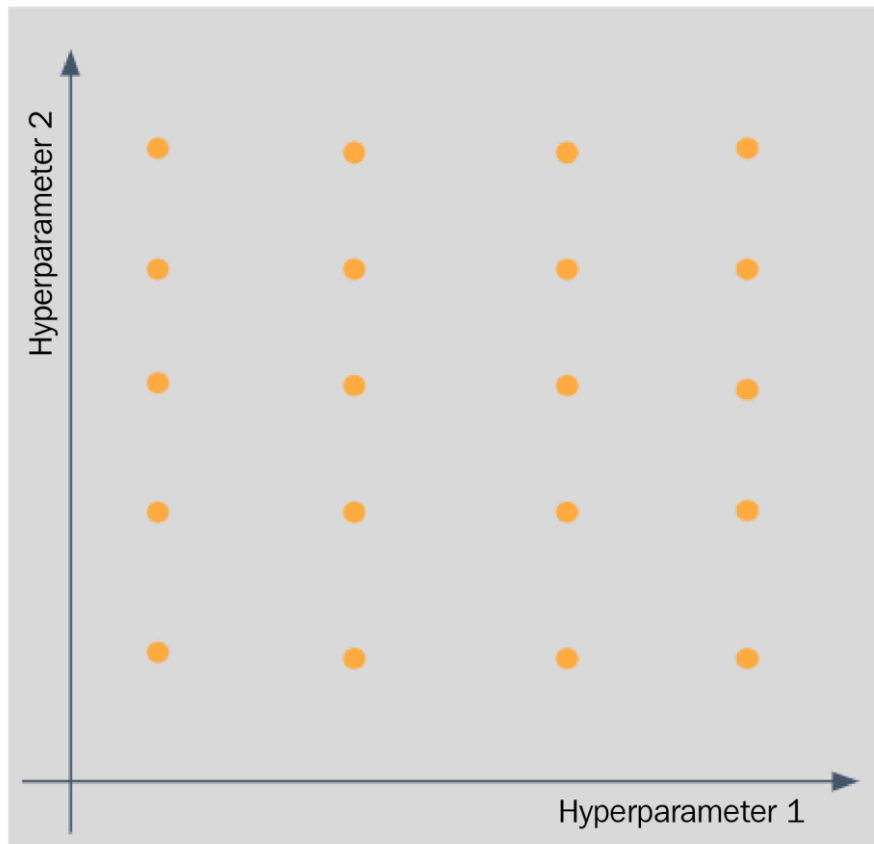




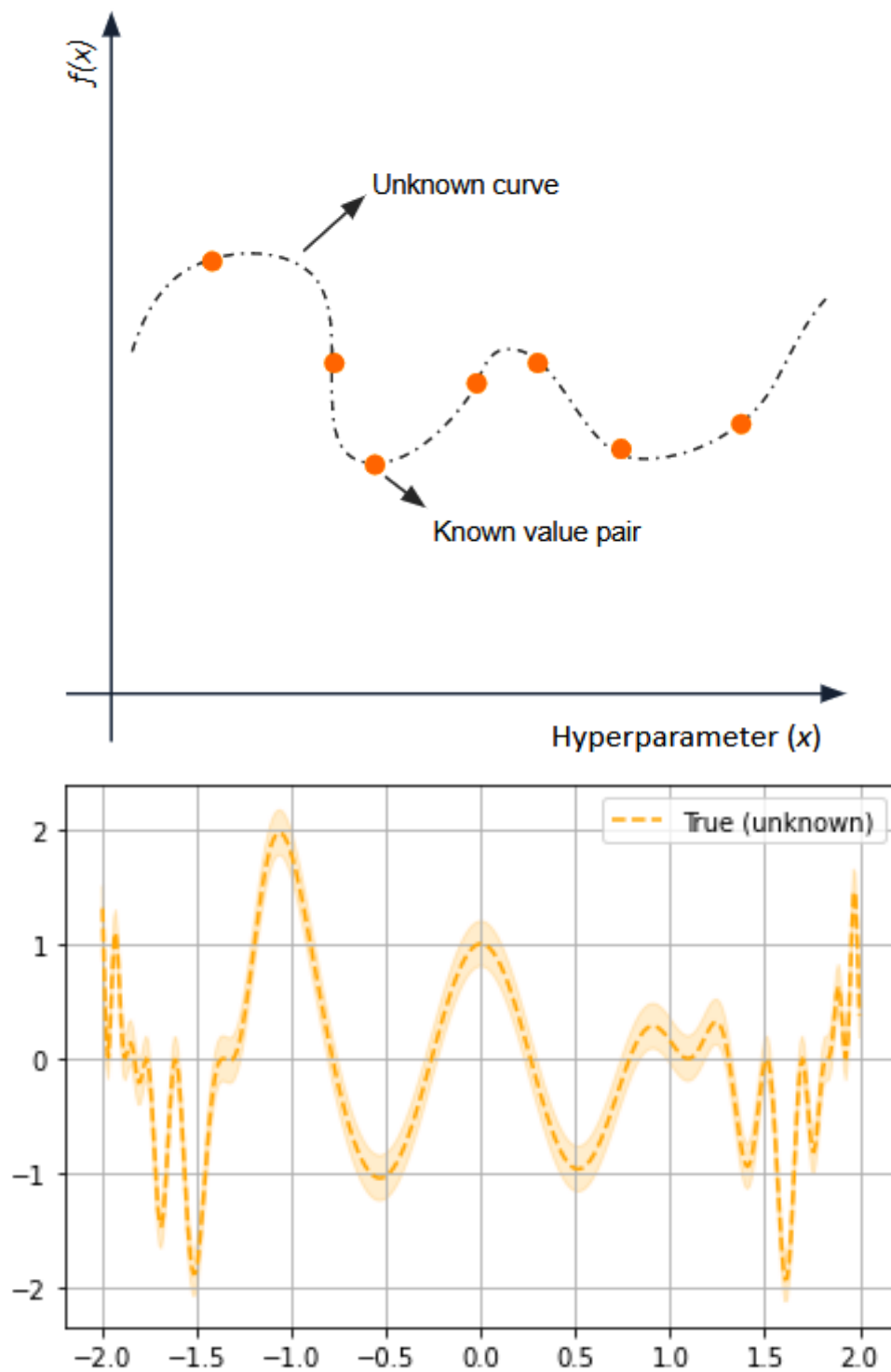
## Chapter 2: Introducing Hyperparameter Tuning

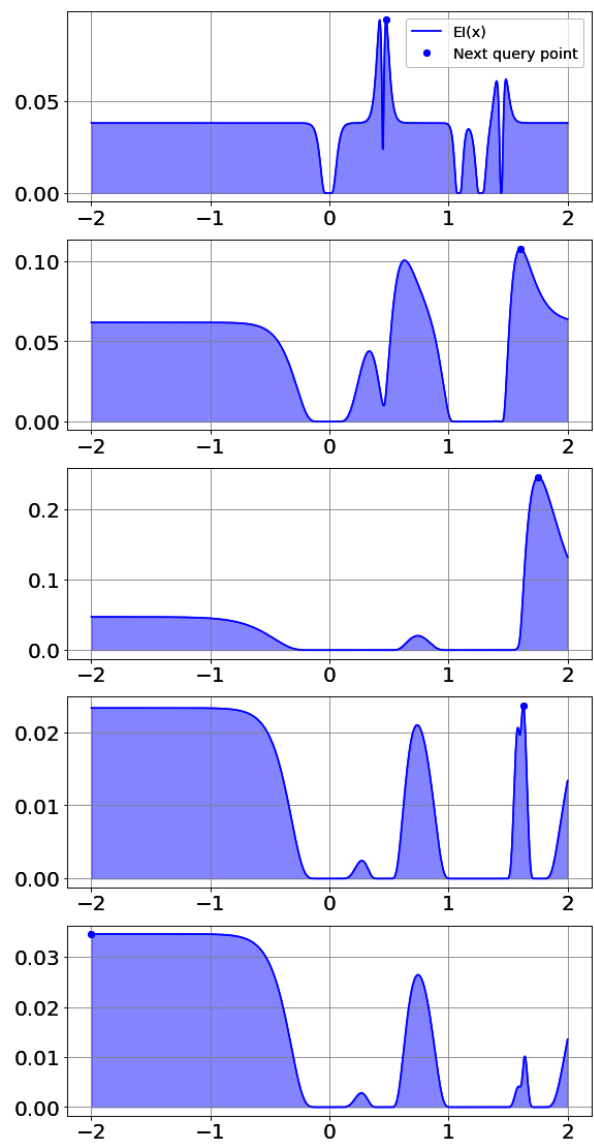
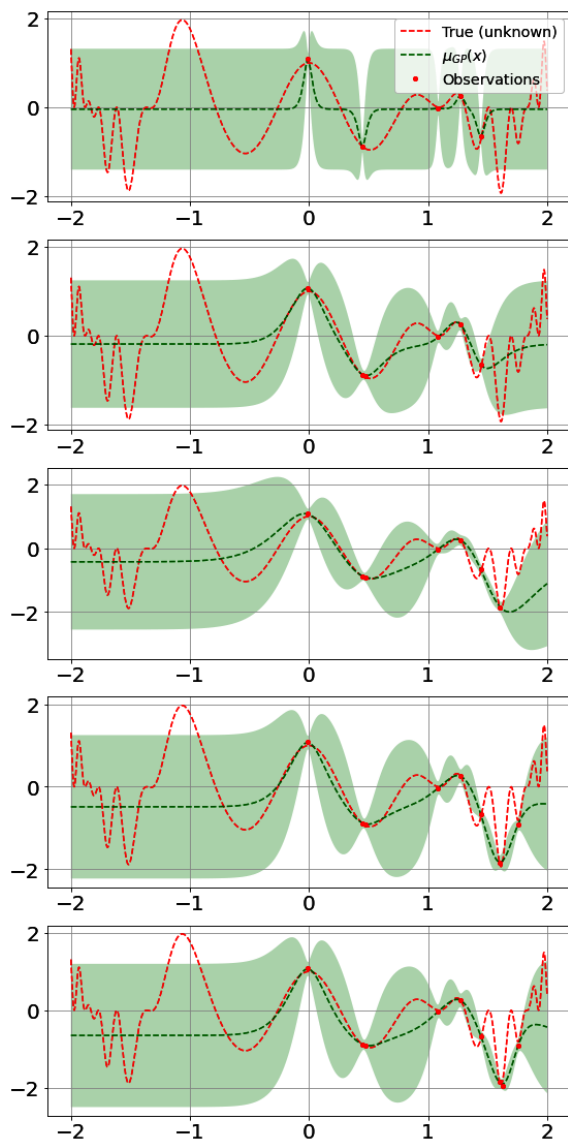
No images...

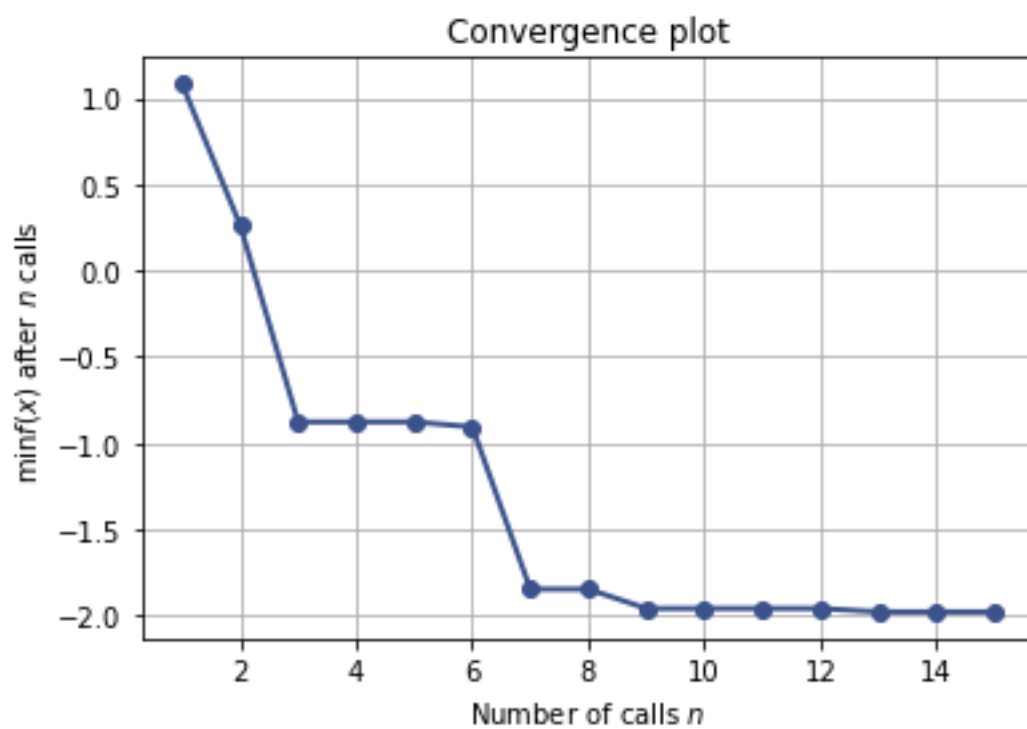
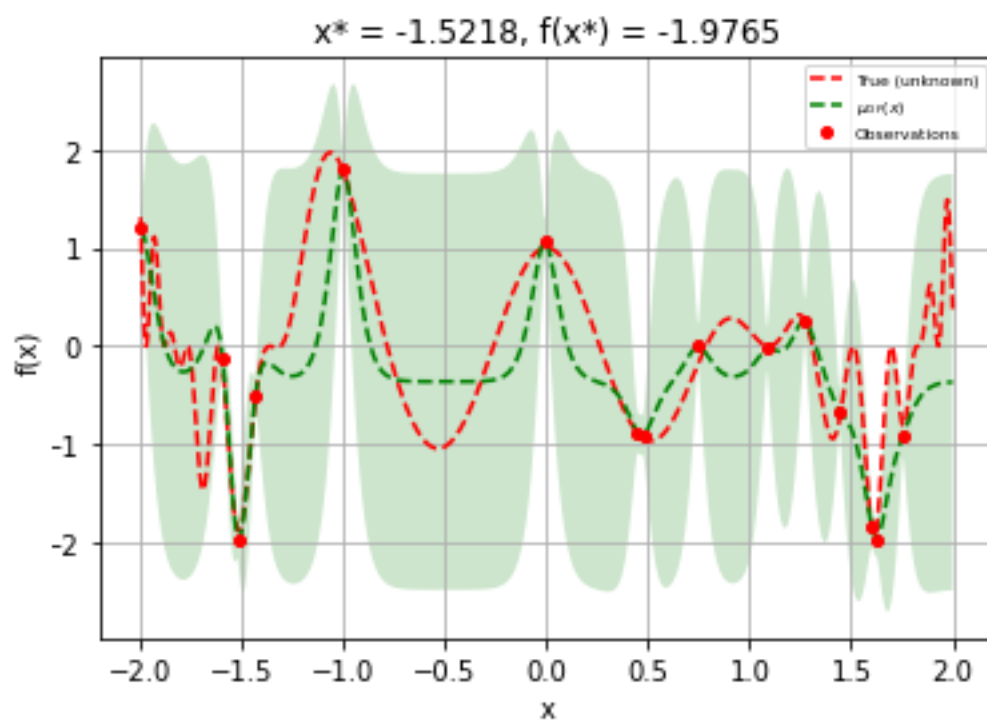
## Chapter 3: Exploring Exhaustive Search

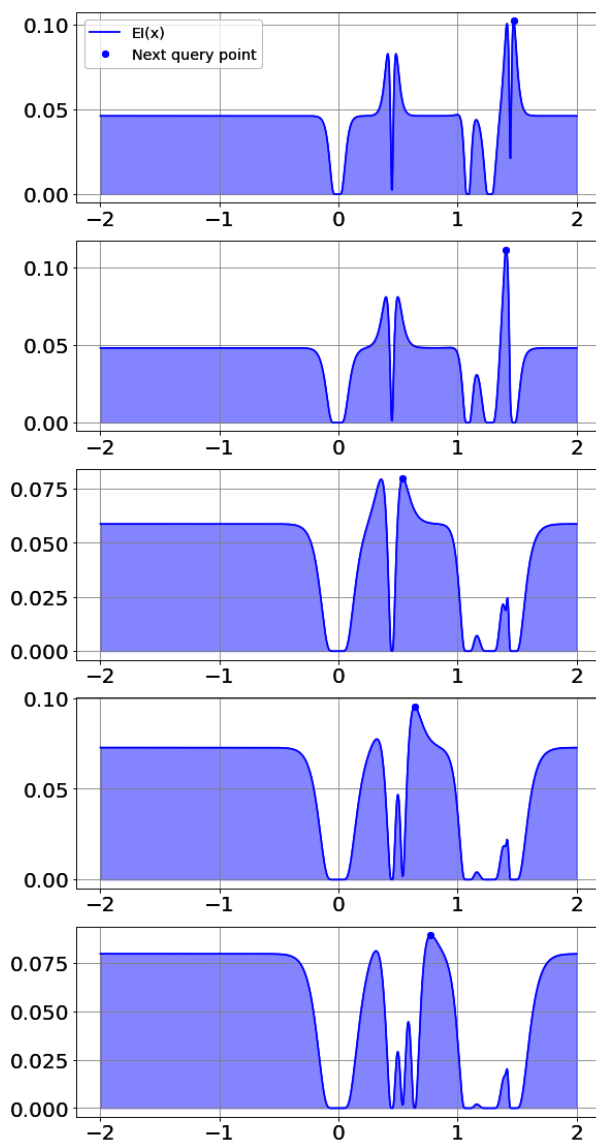
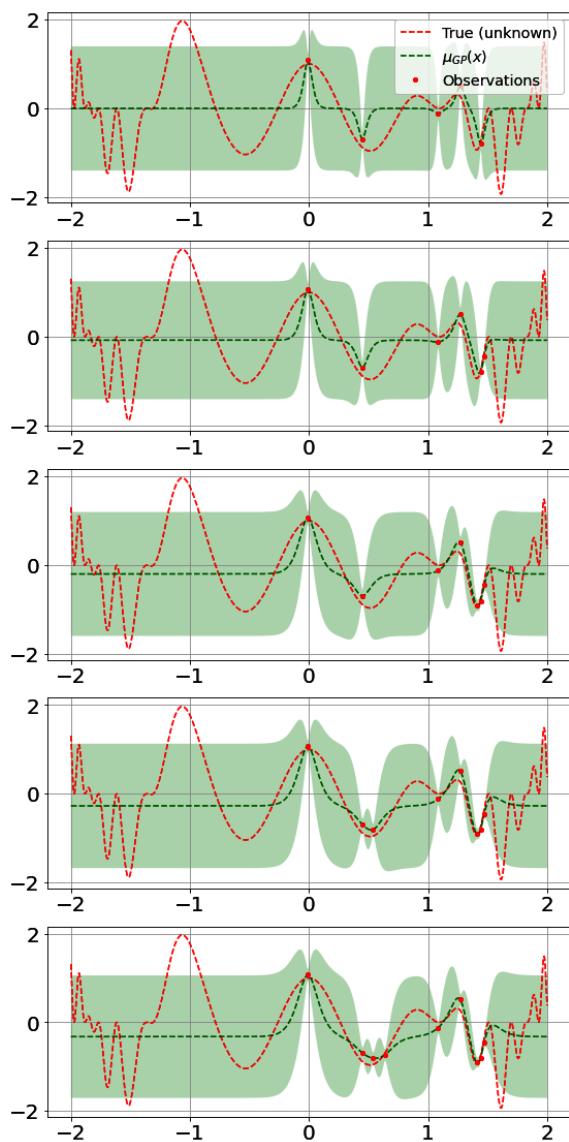


## Chapter 4: Exploring Bayesian Optimization

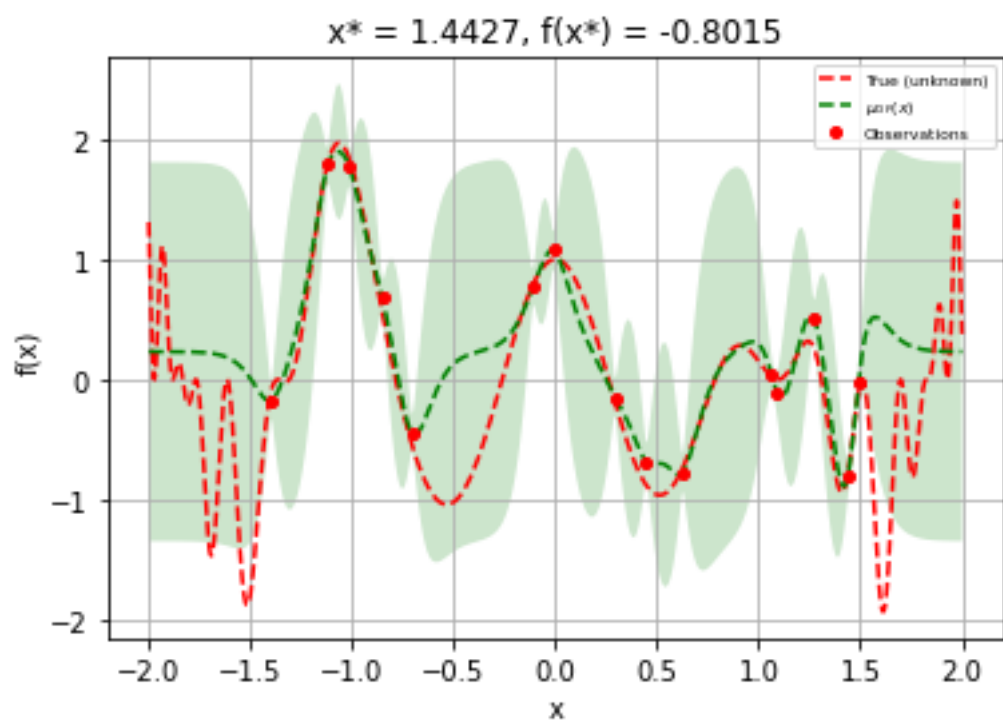
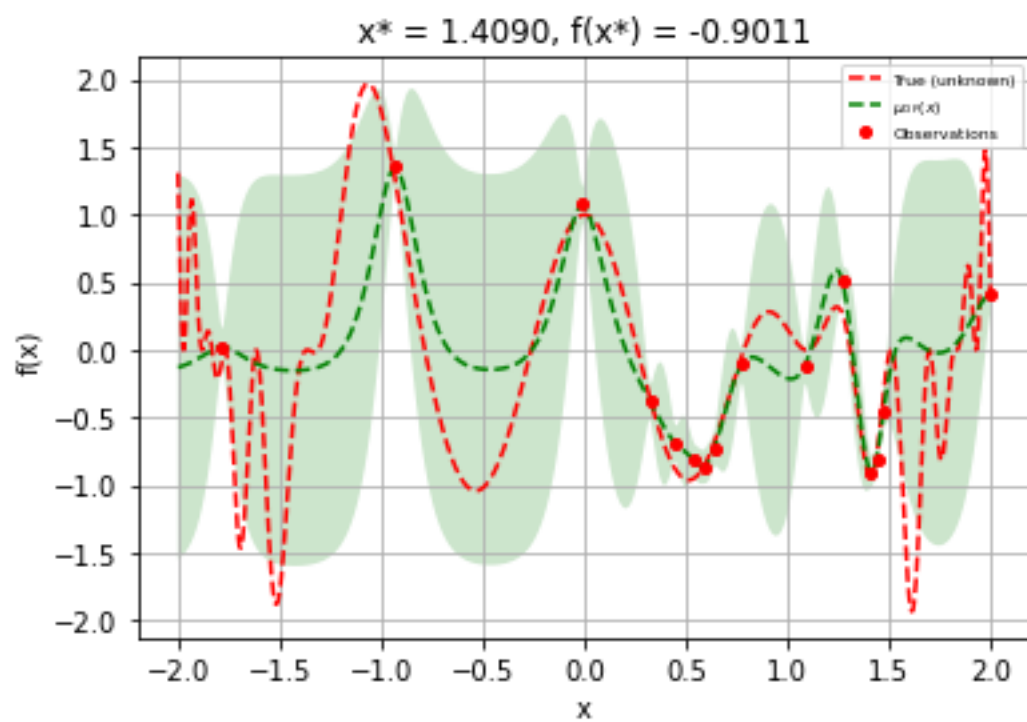


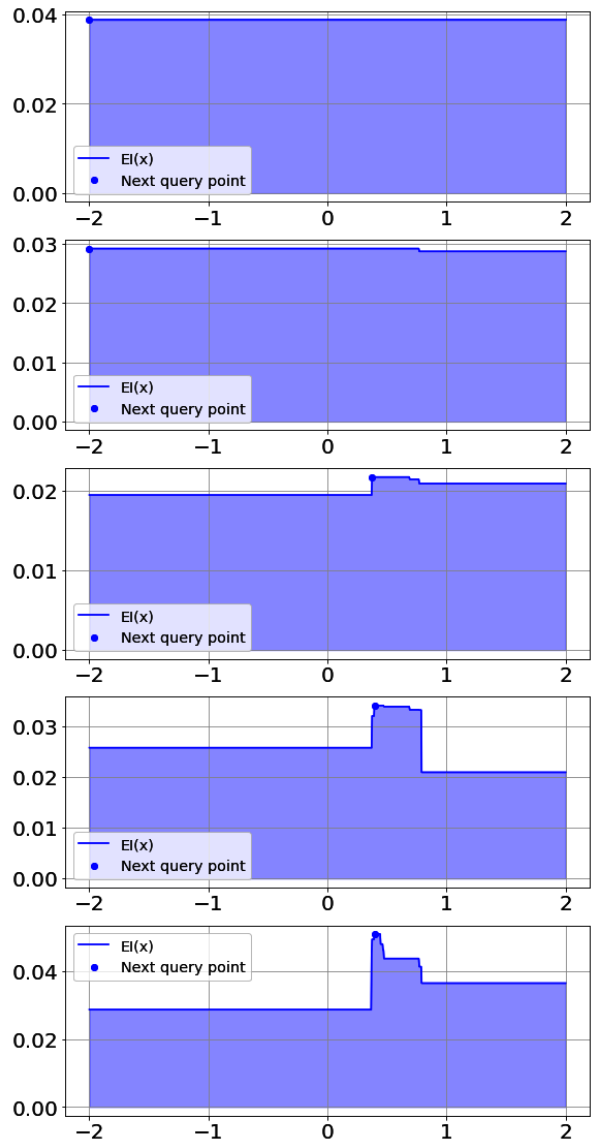
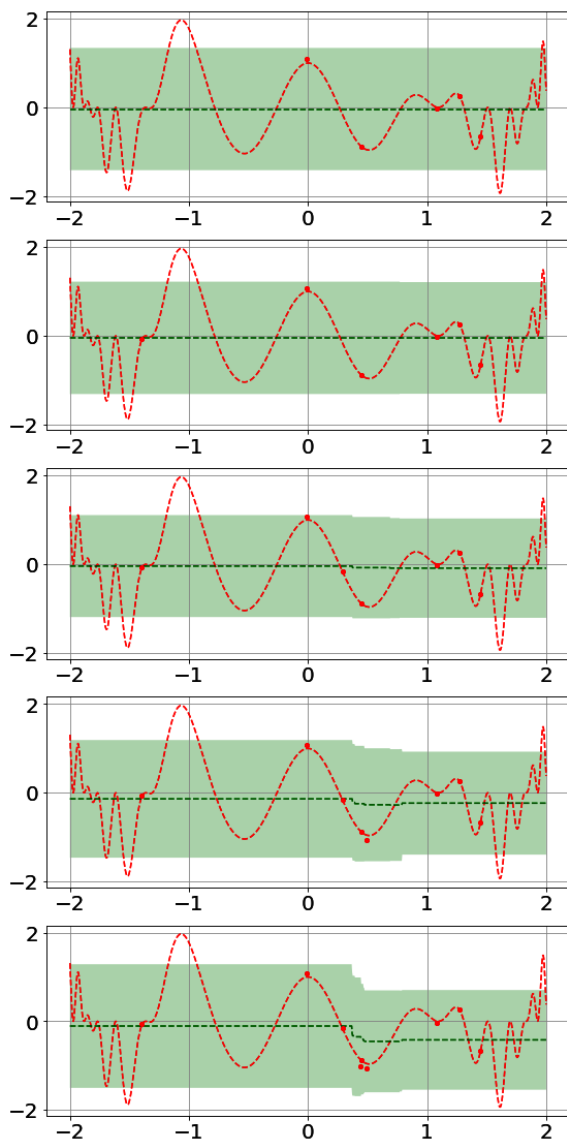


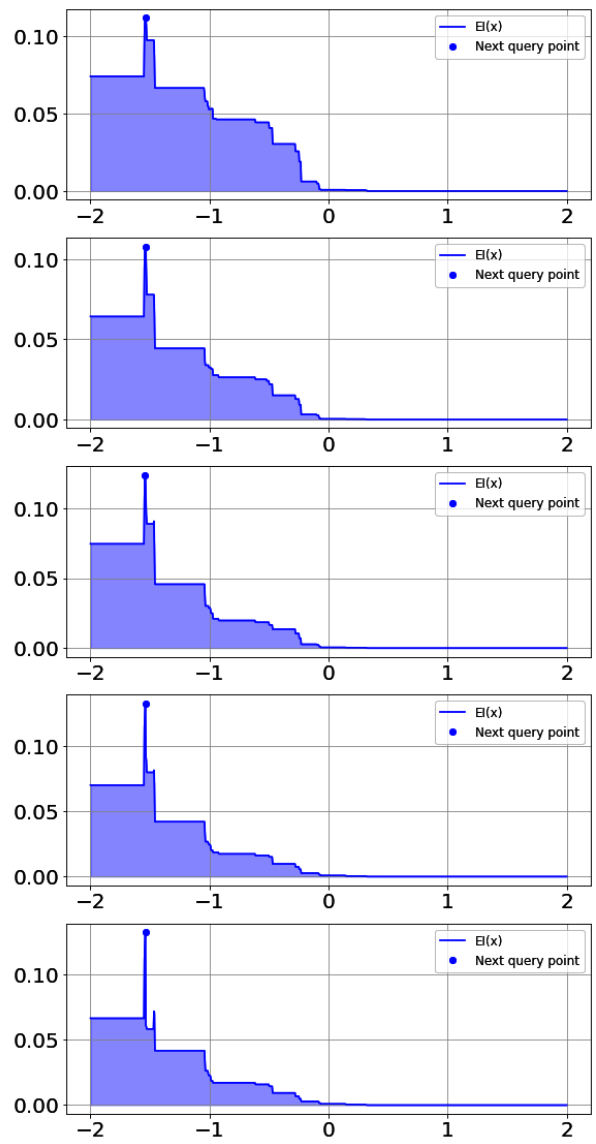
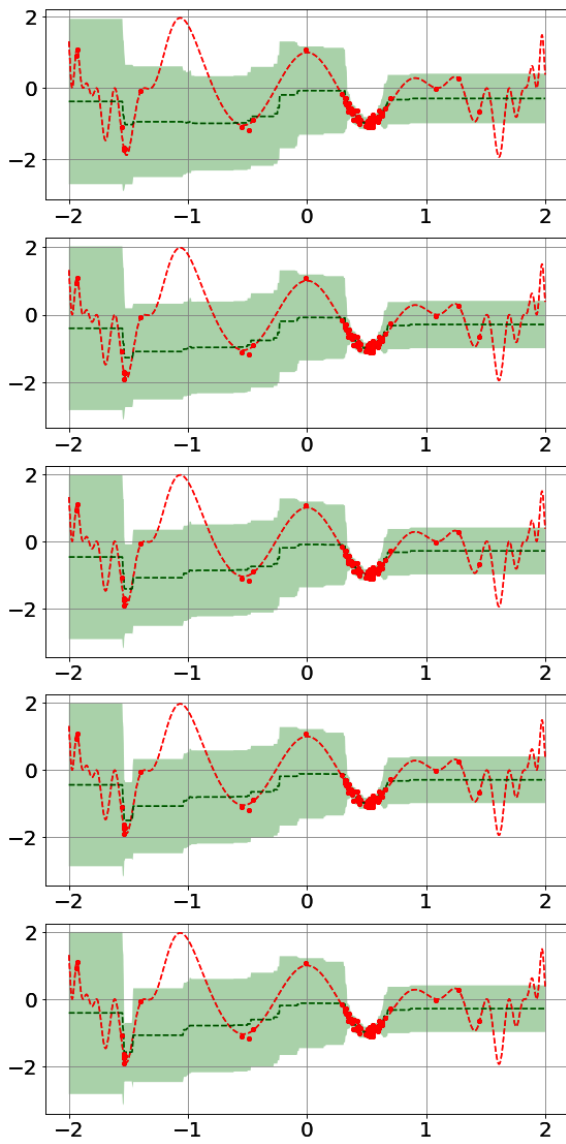


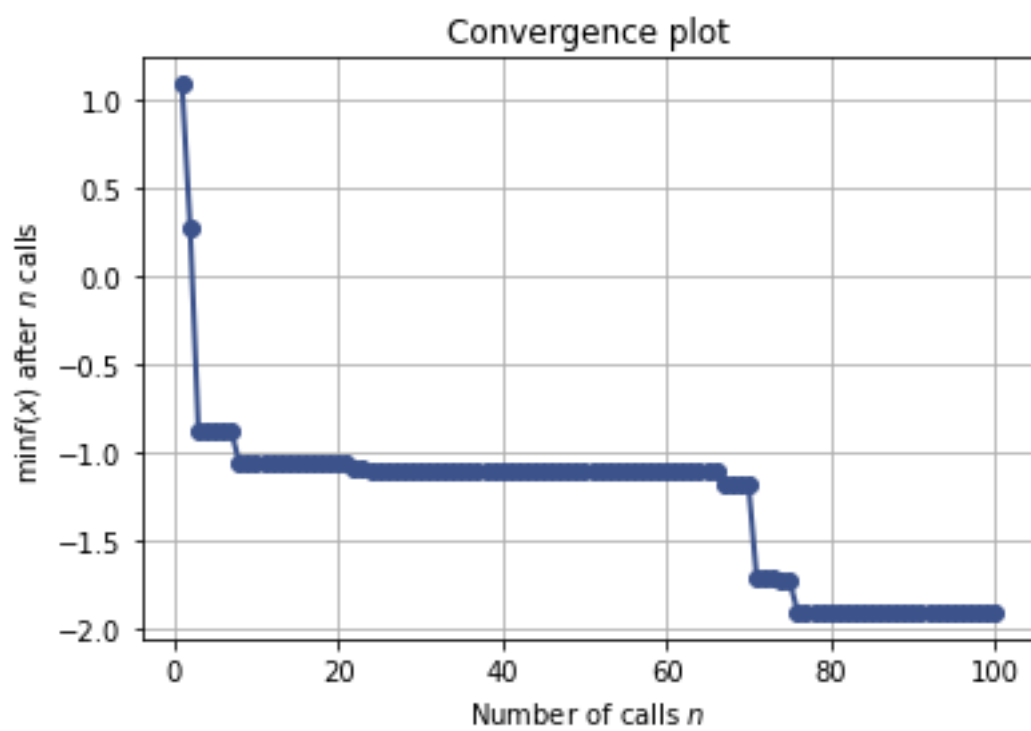
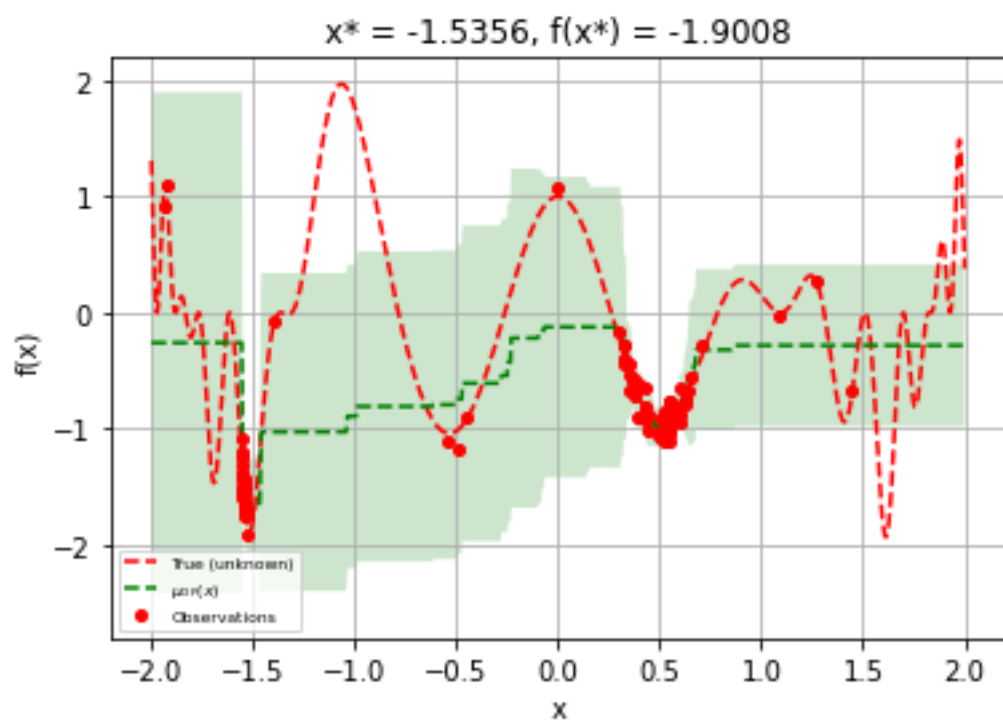


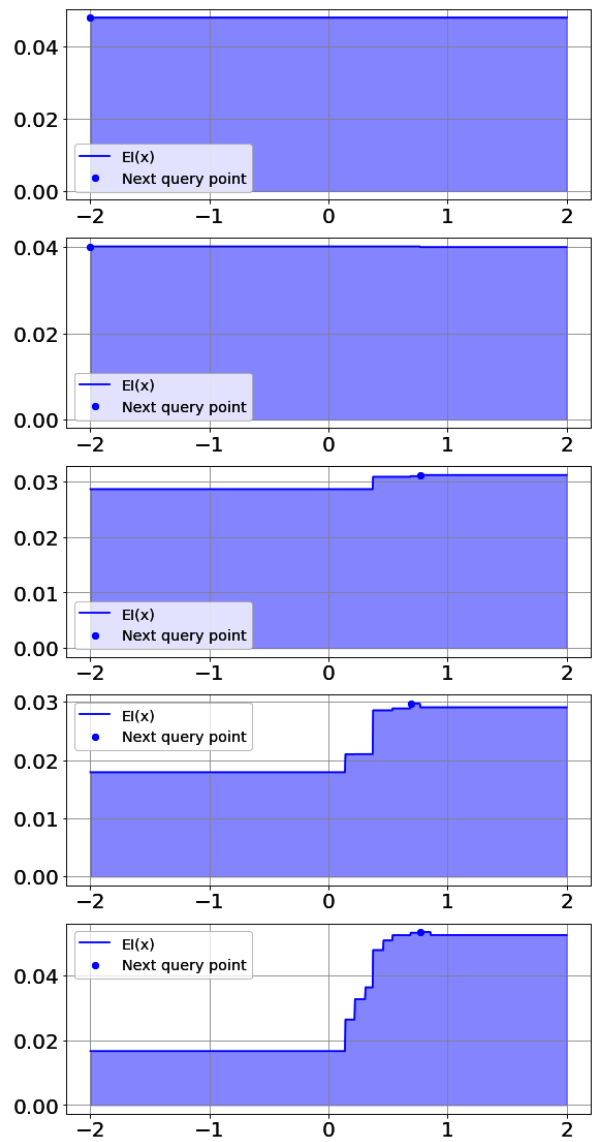
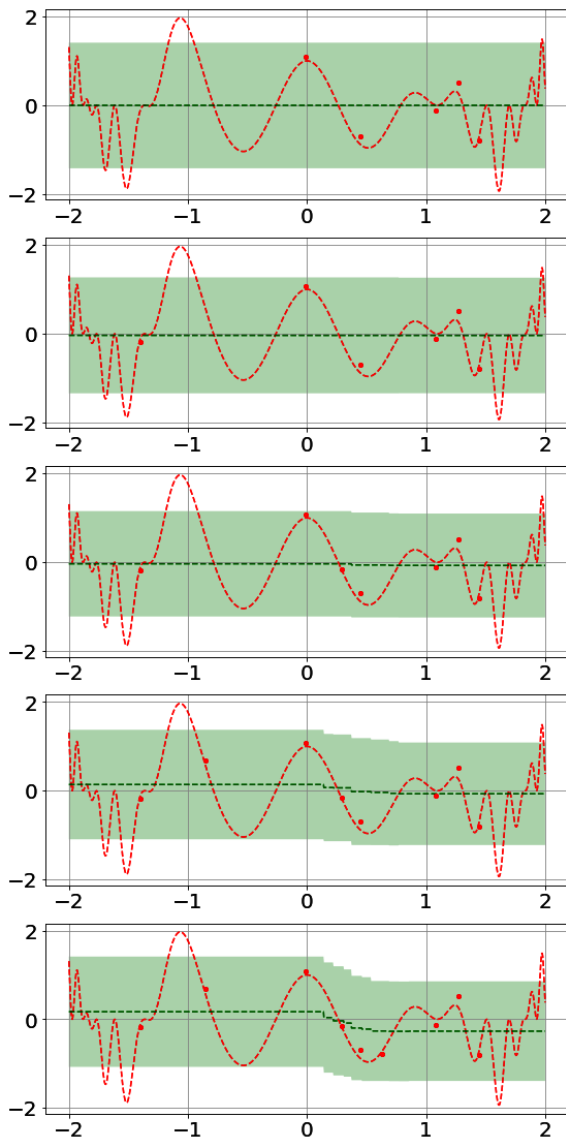


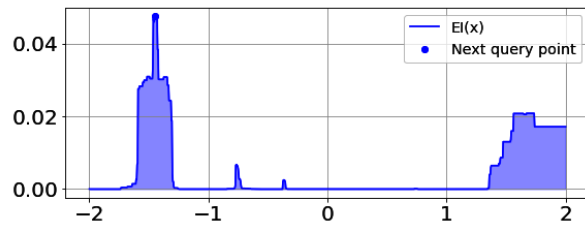
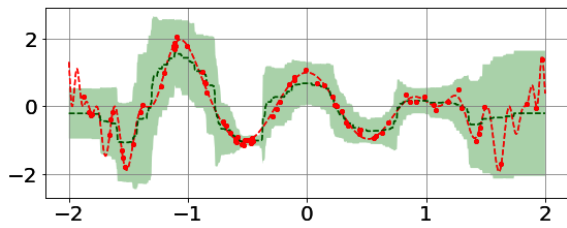
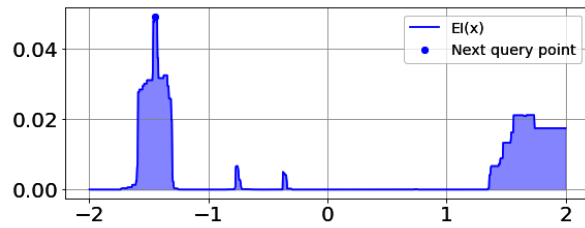
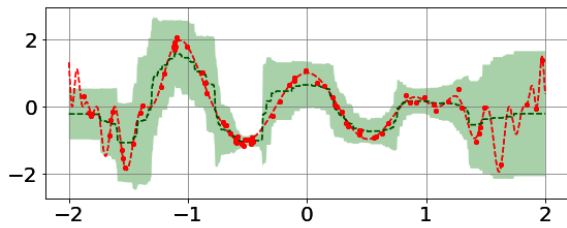
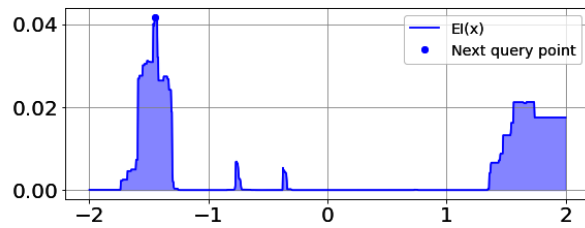
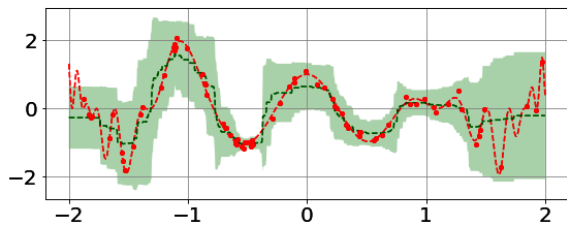
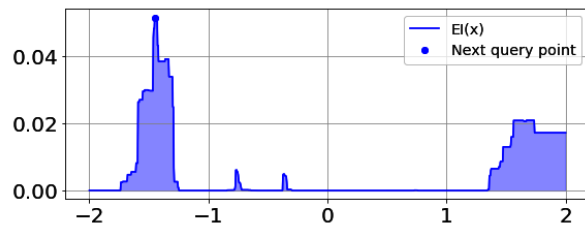
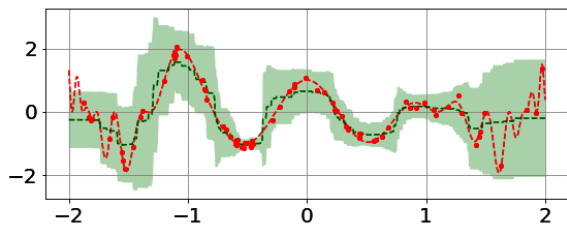
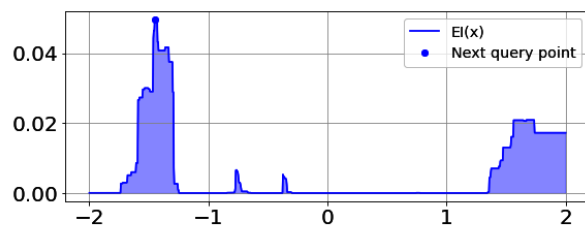
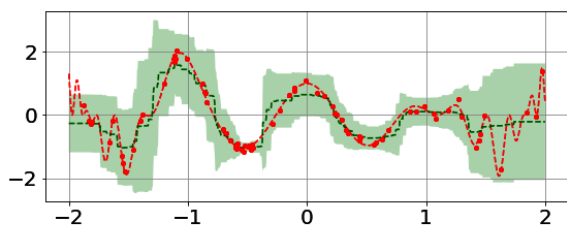


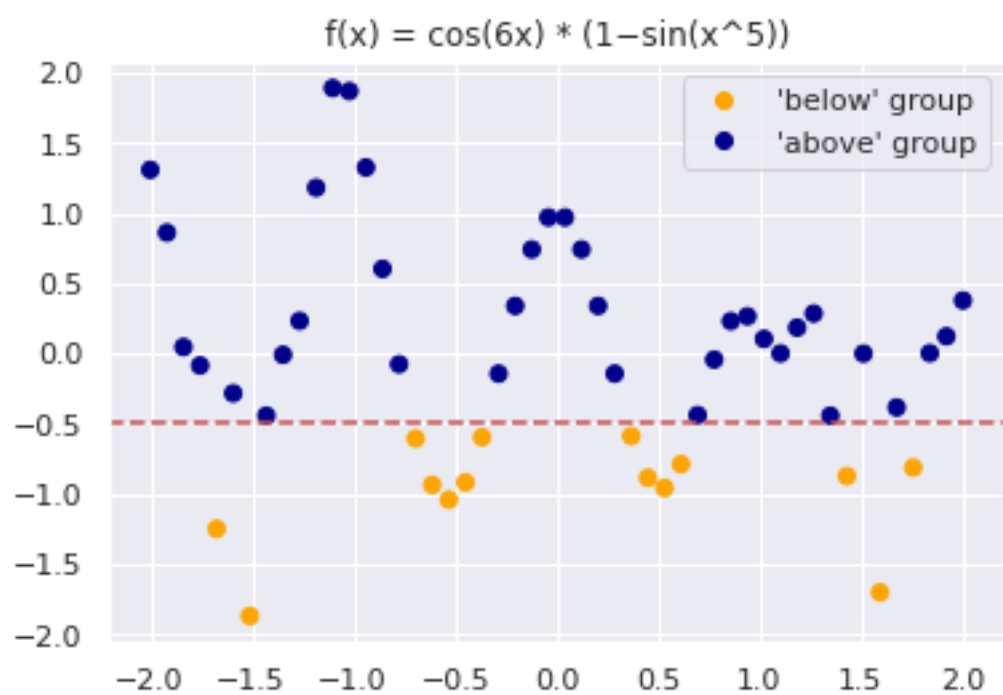
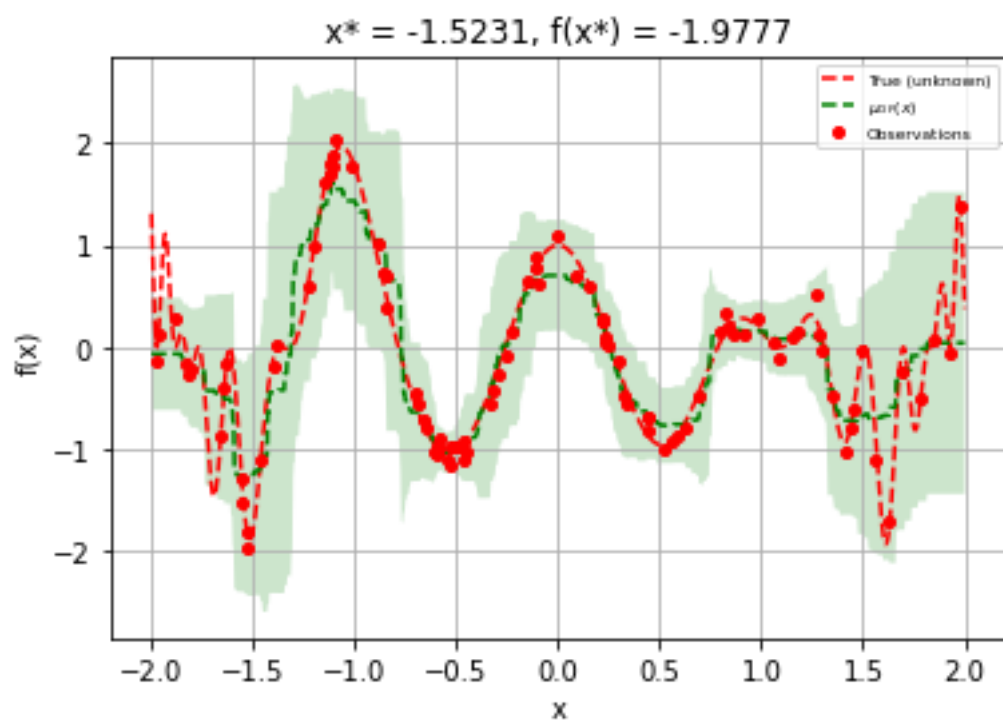




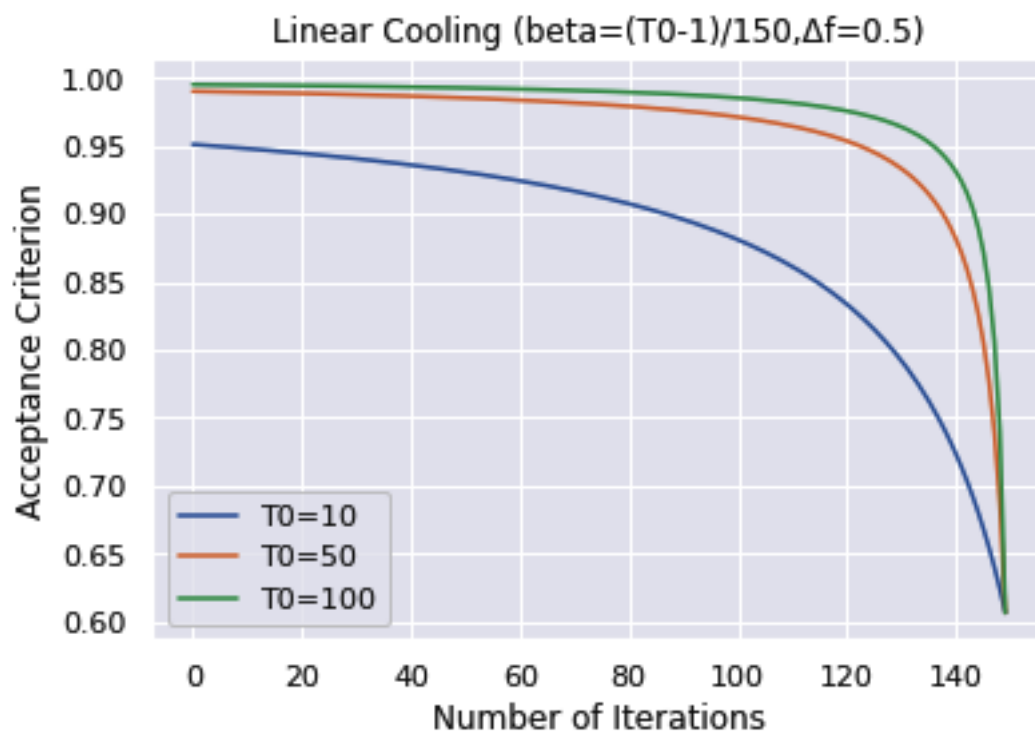
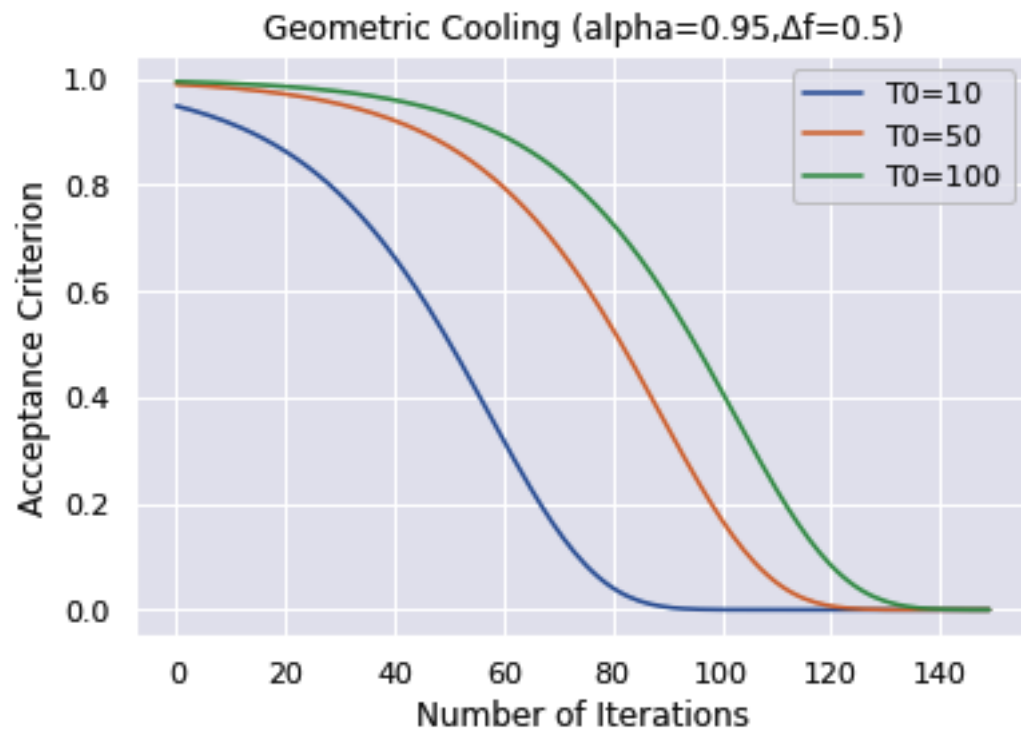




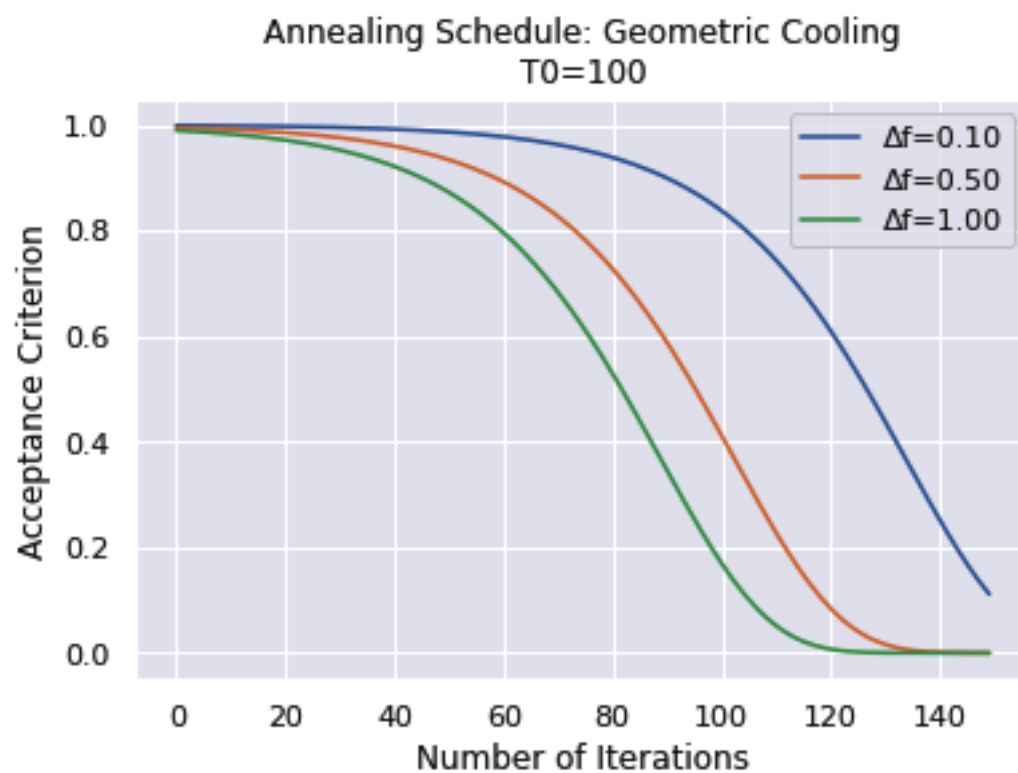
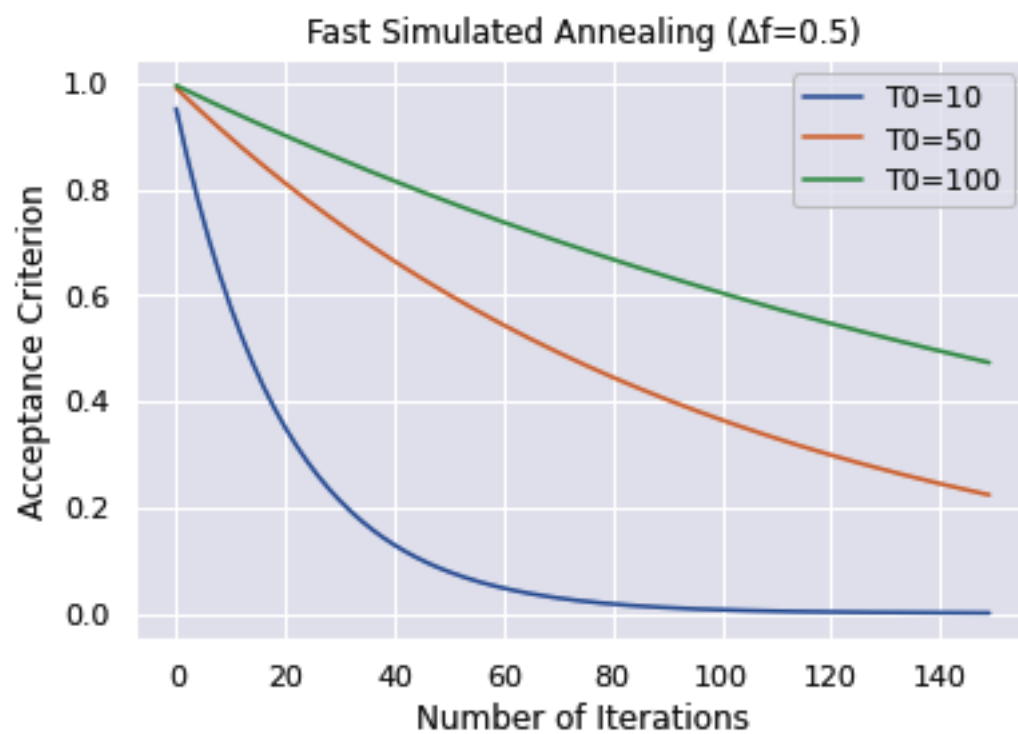


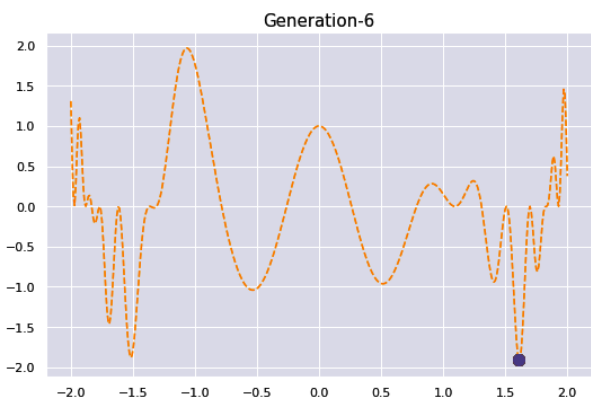
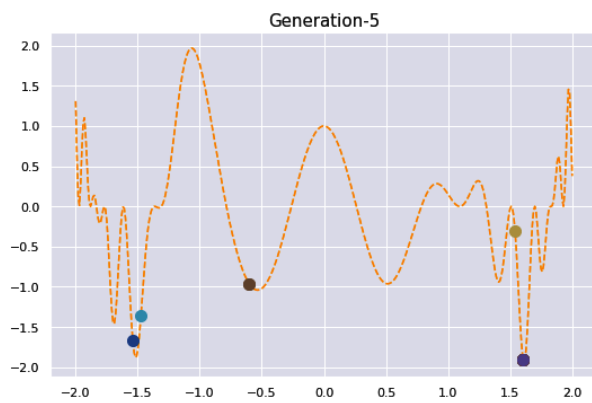
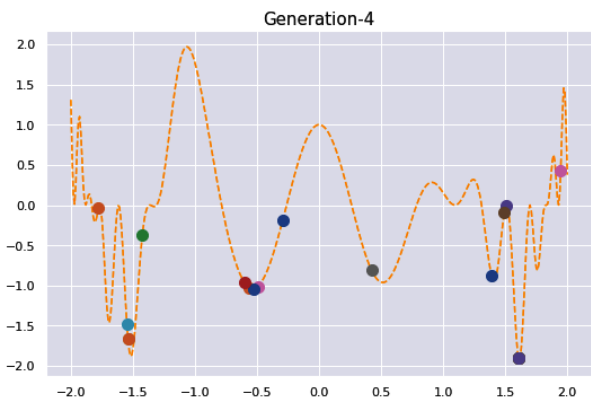
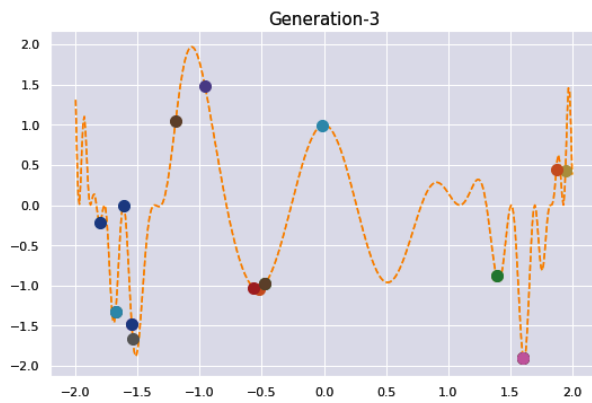
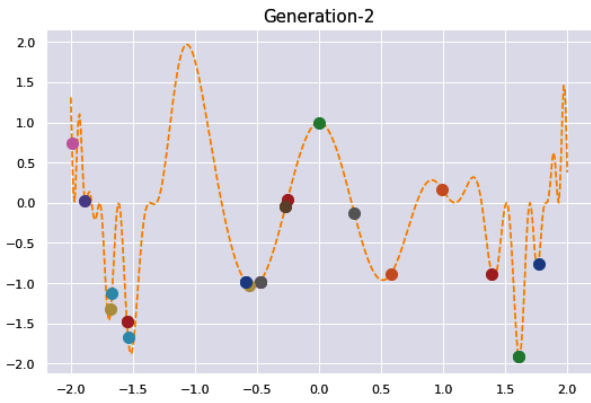
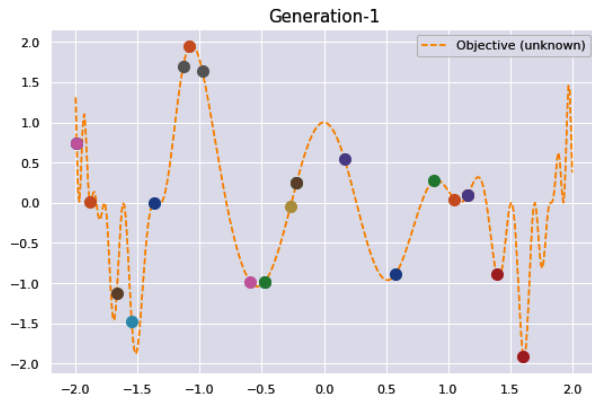
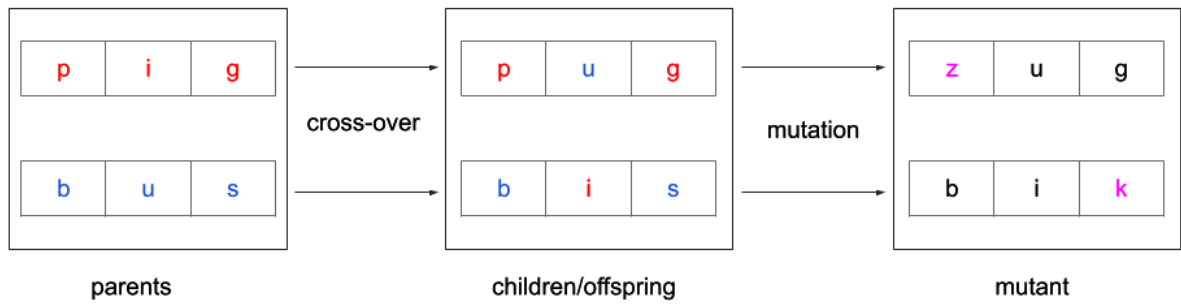


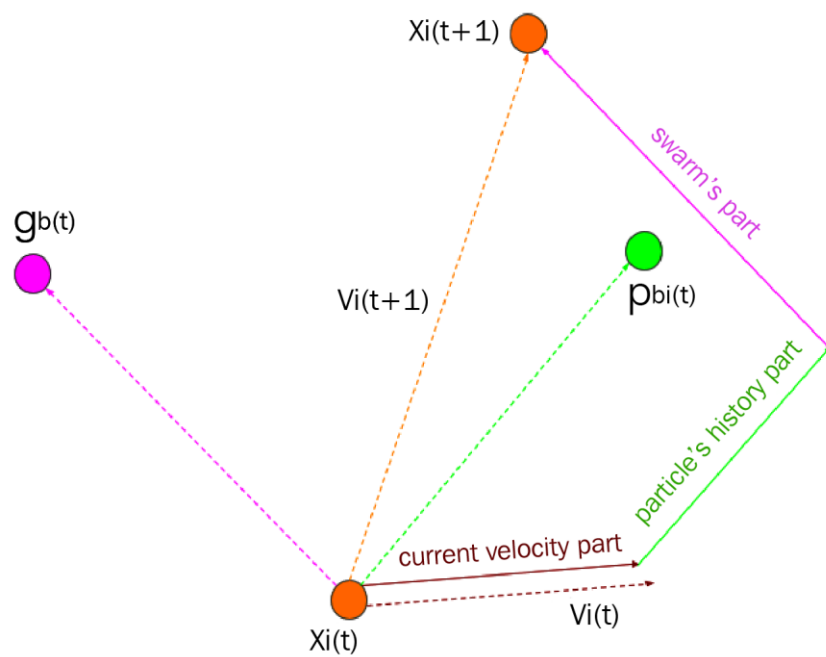
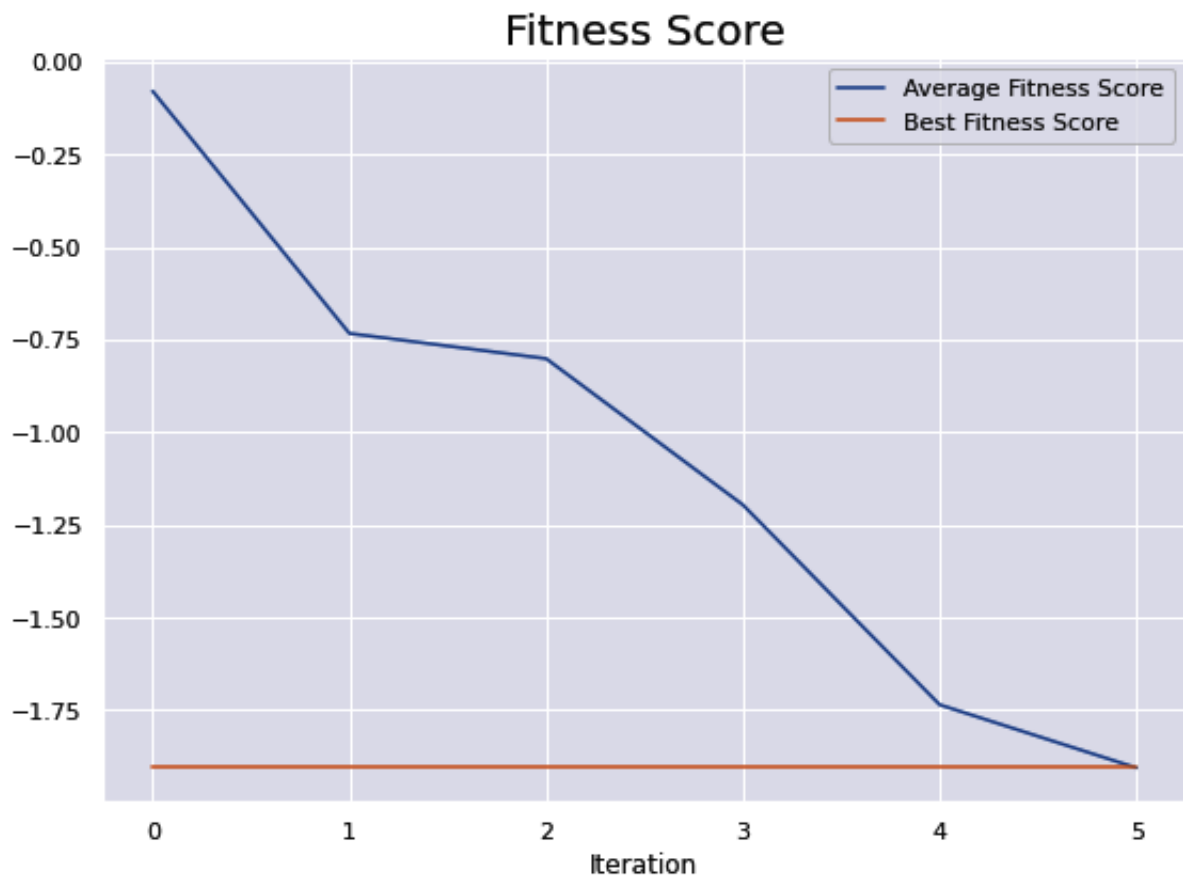
## Chapter 5: Exploring Heuristic Search

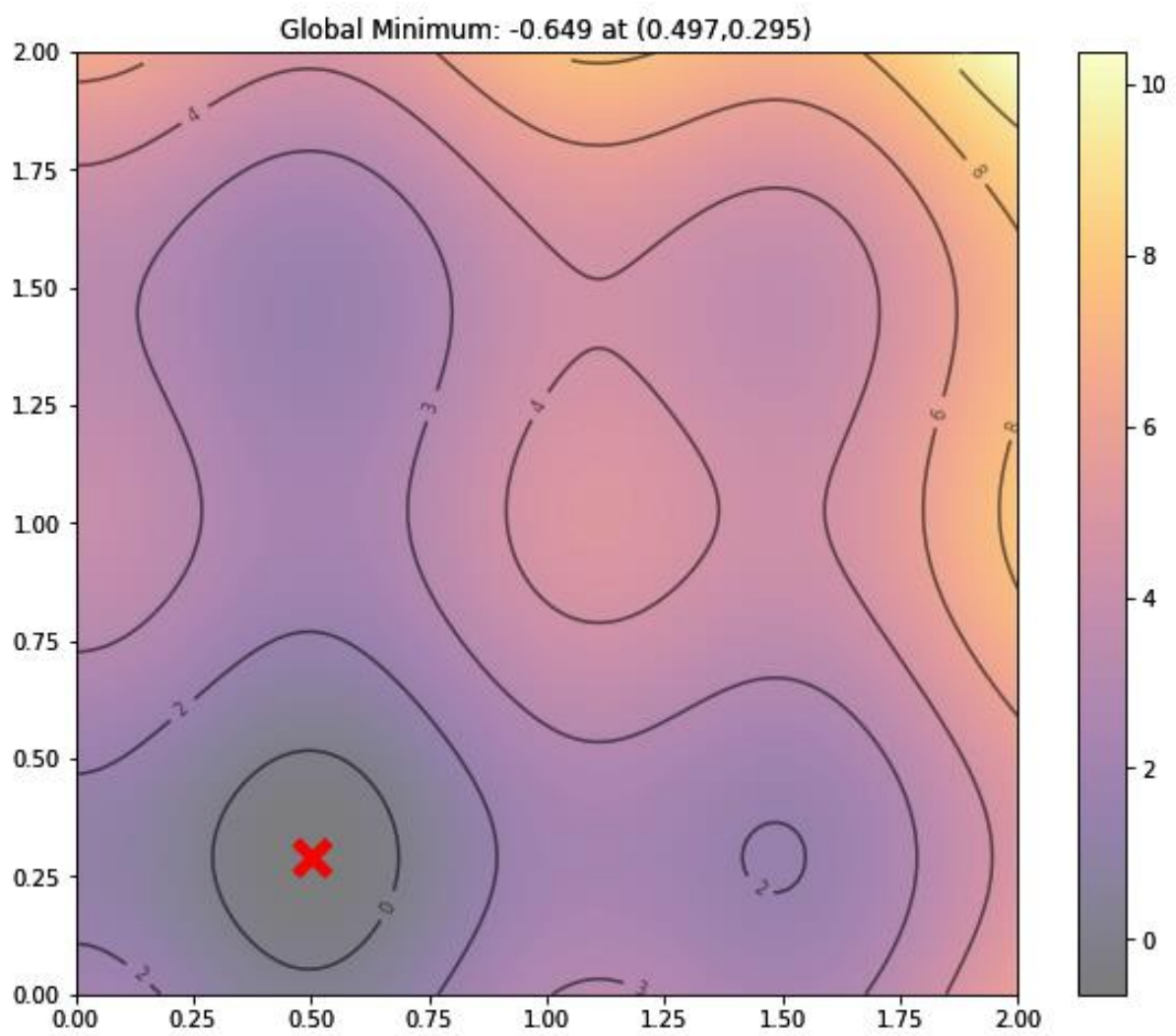




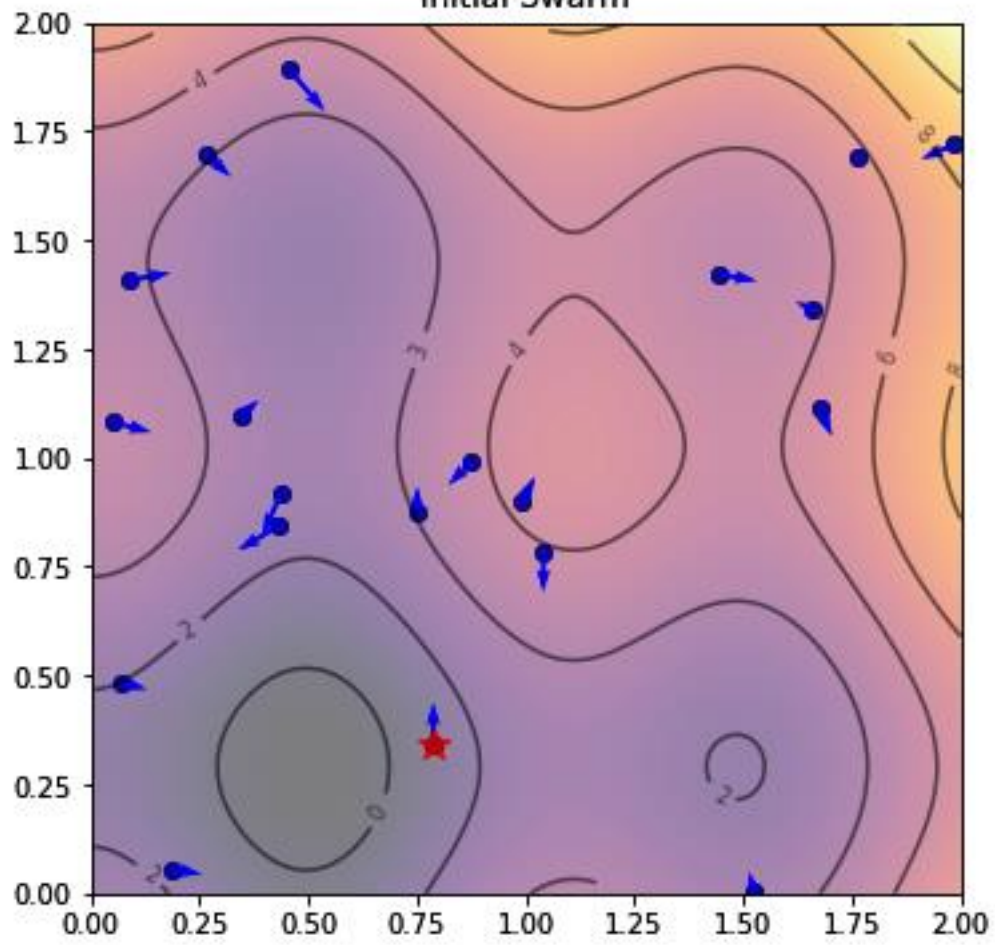


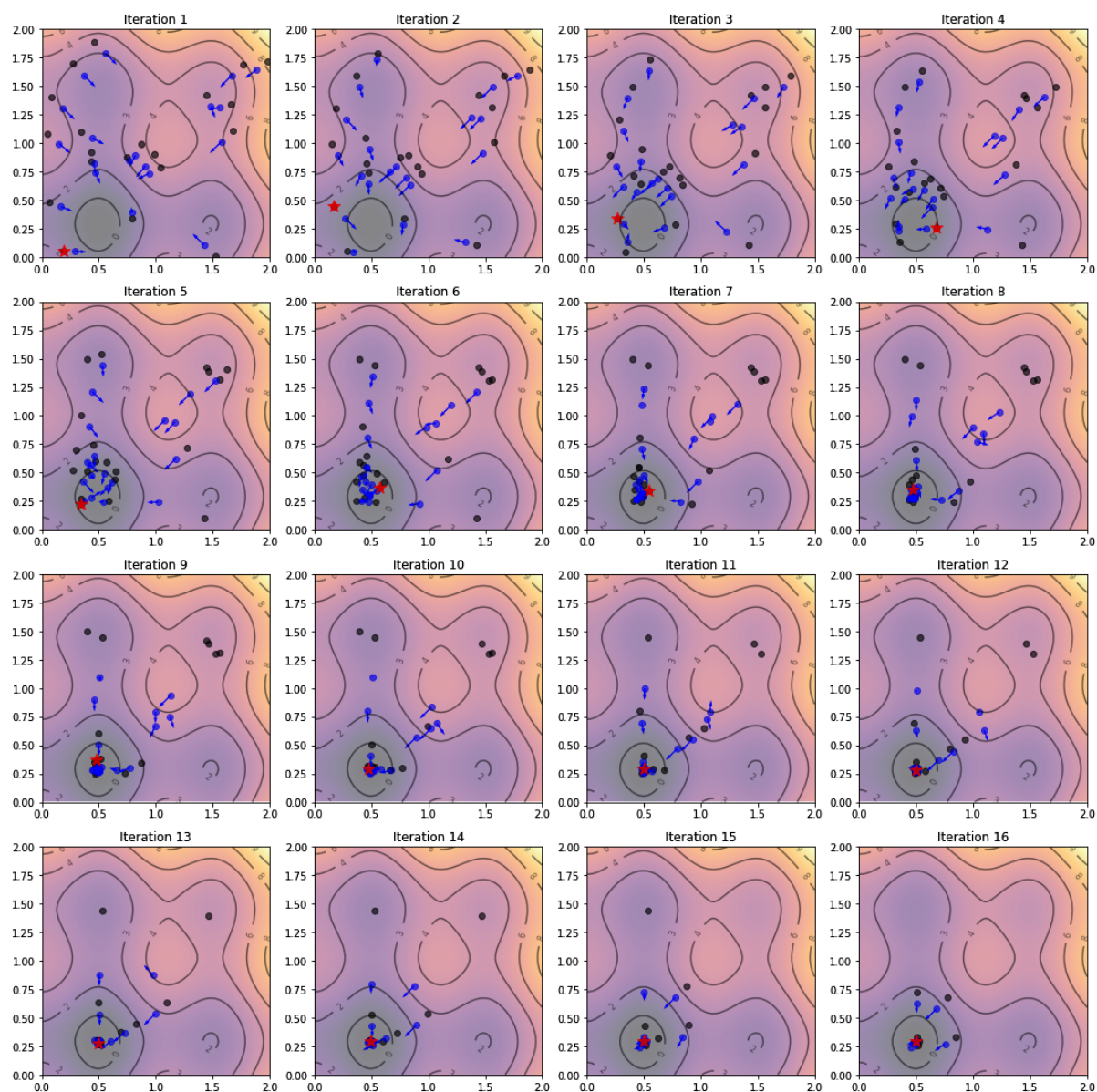




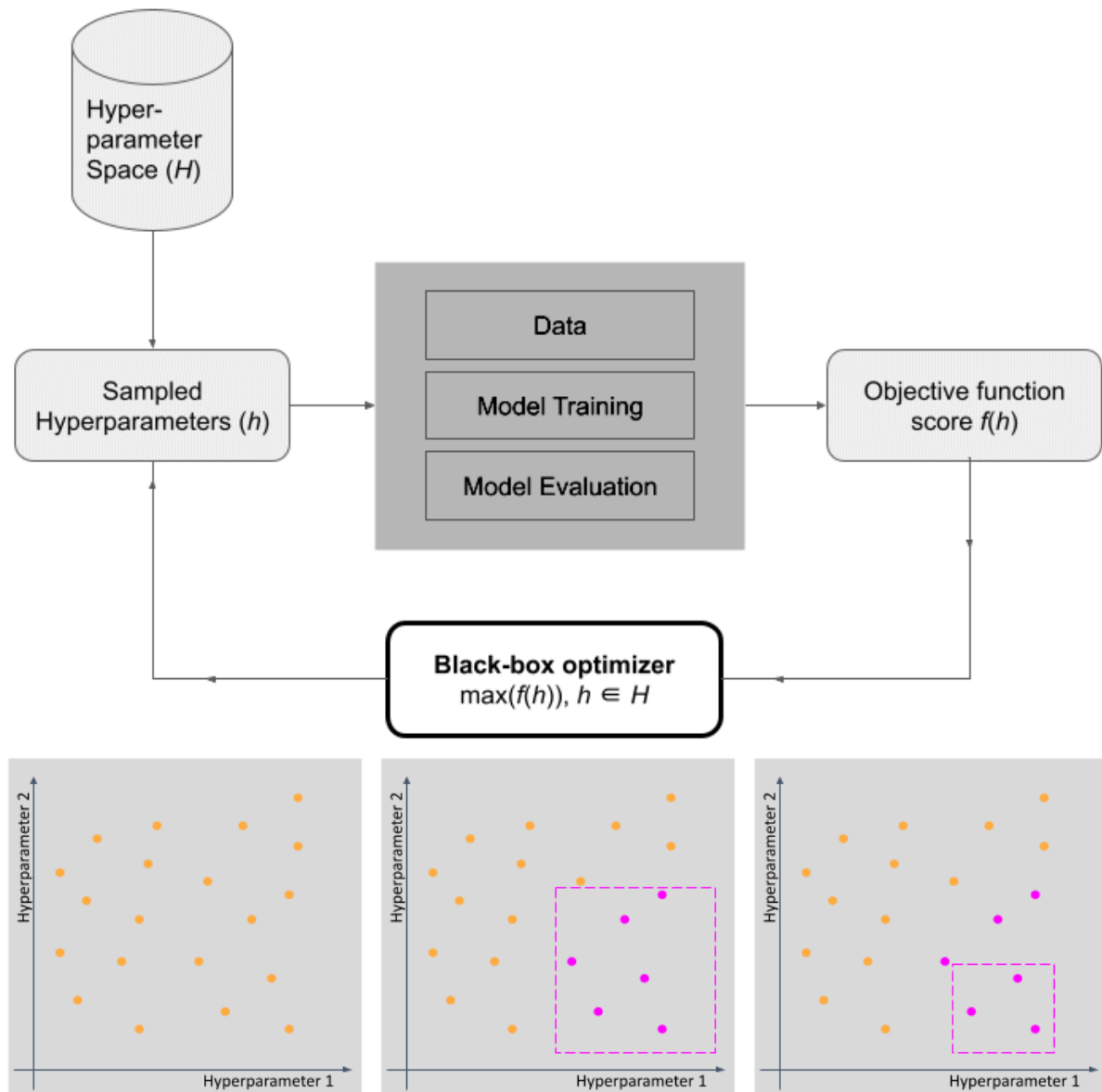


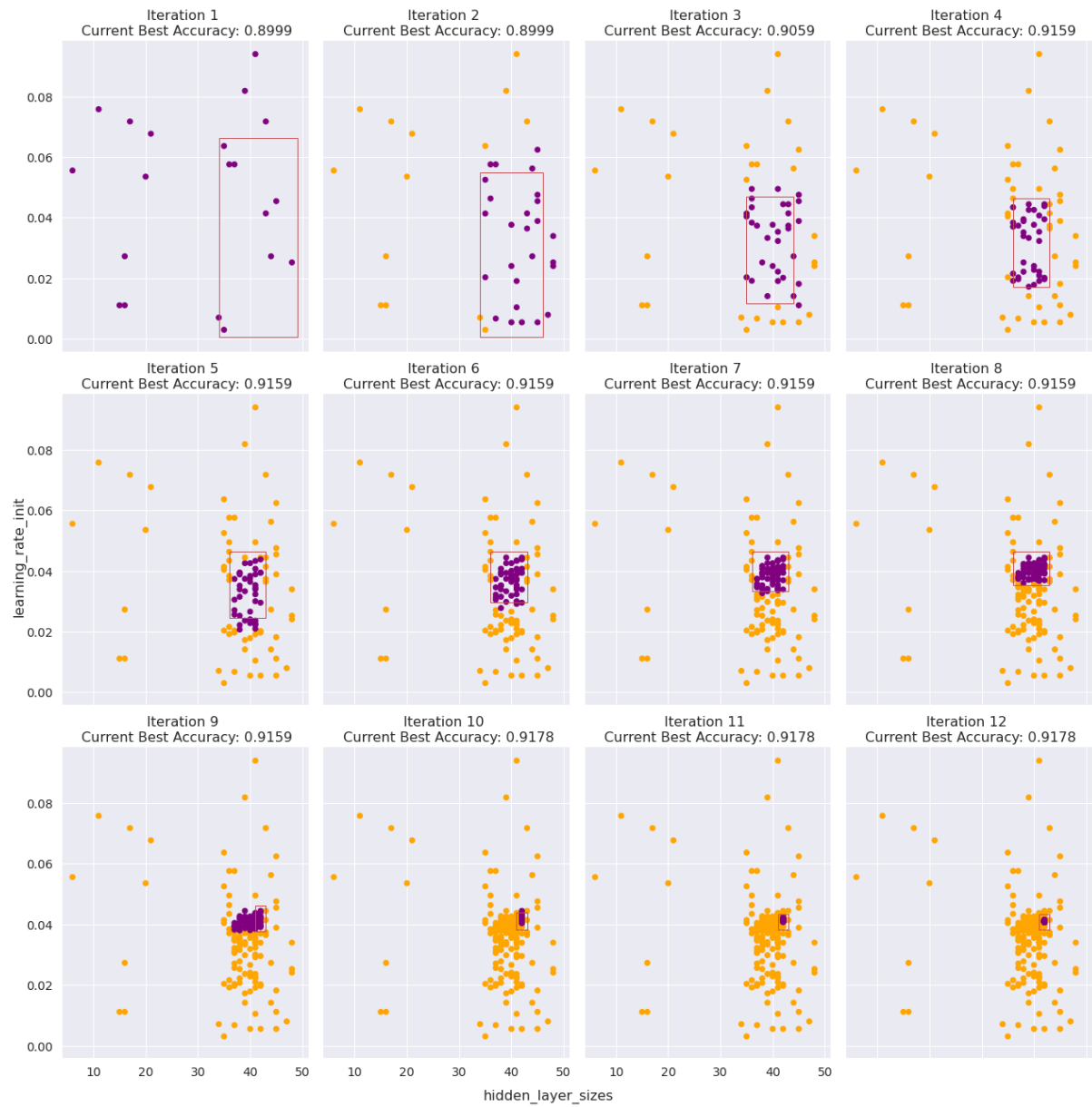
Initial Swarm



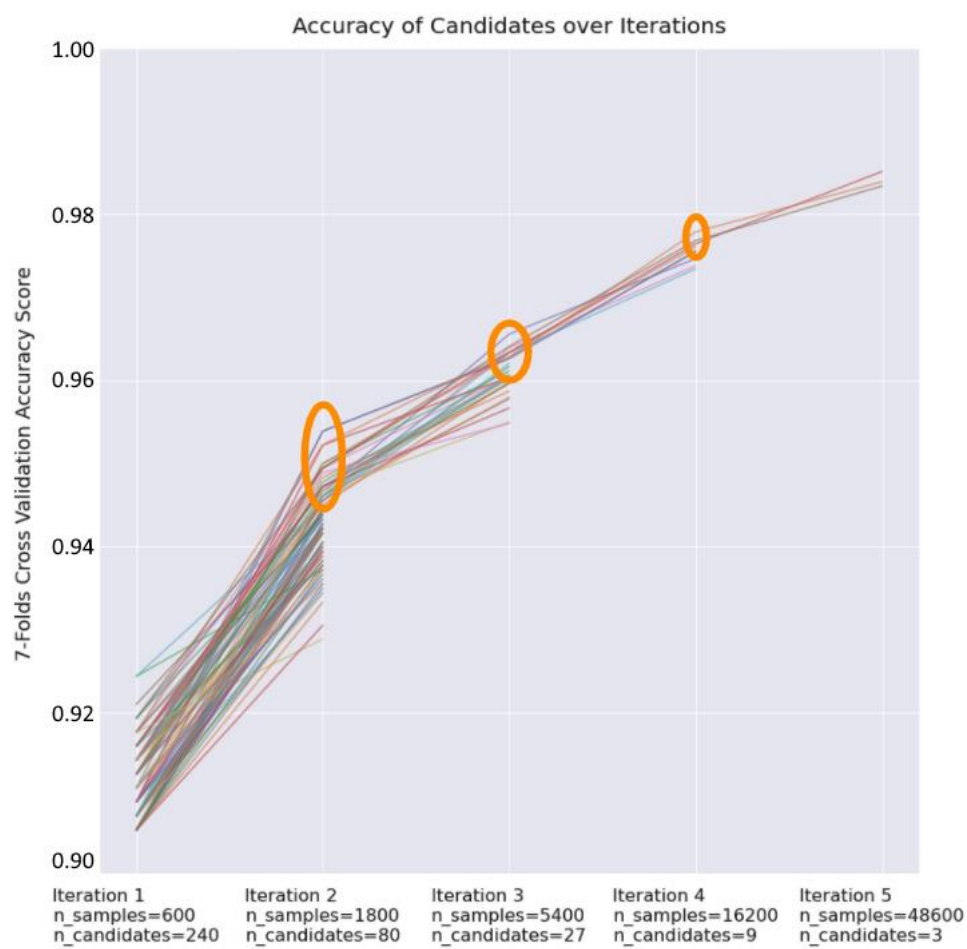
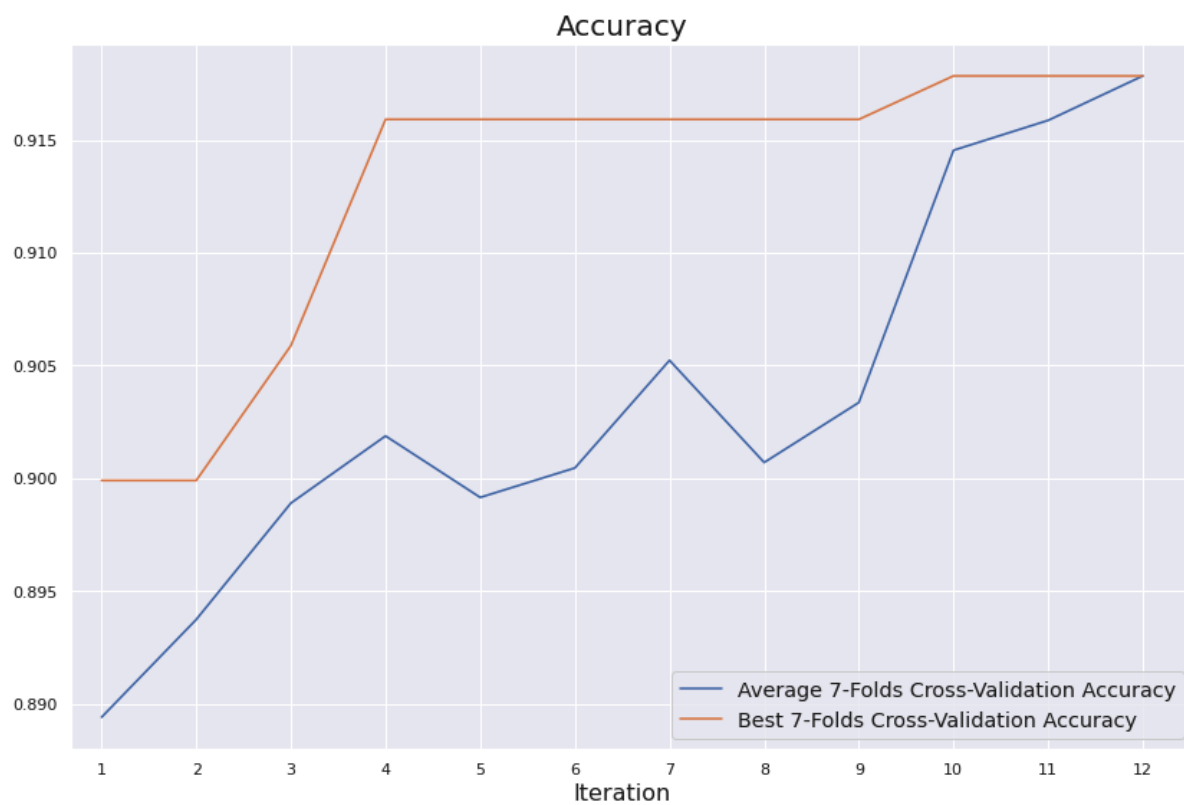


## Chapter 6: Exploring Multi-Fidelity Optimization

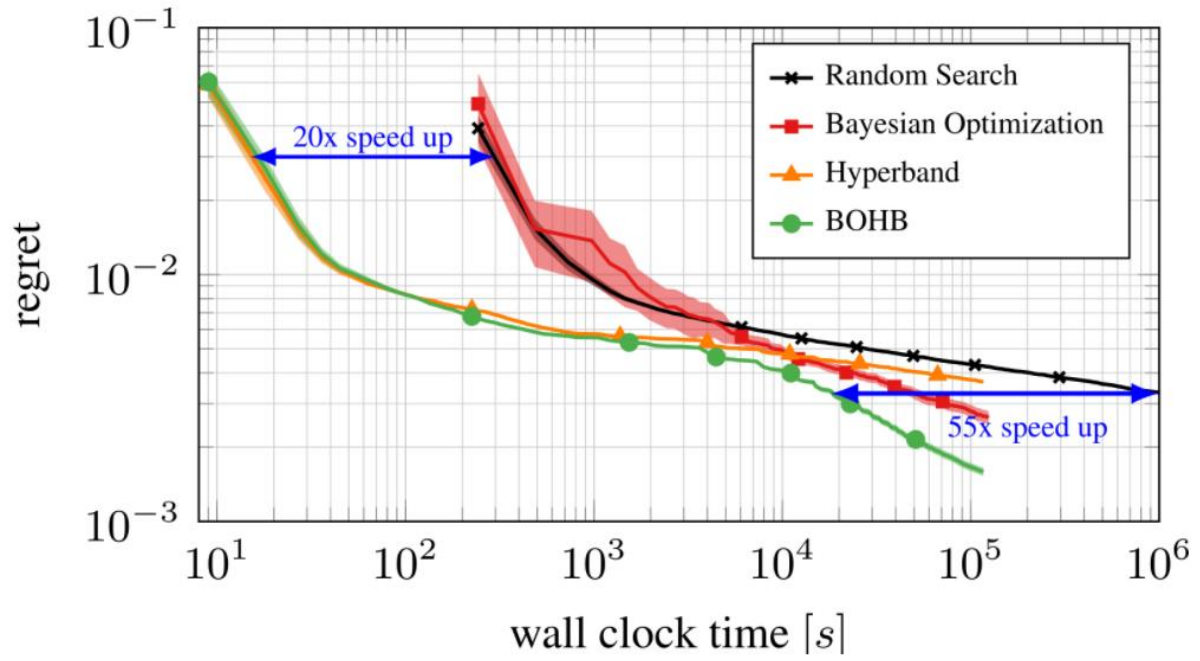




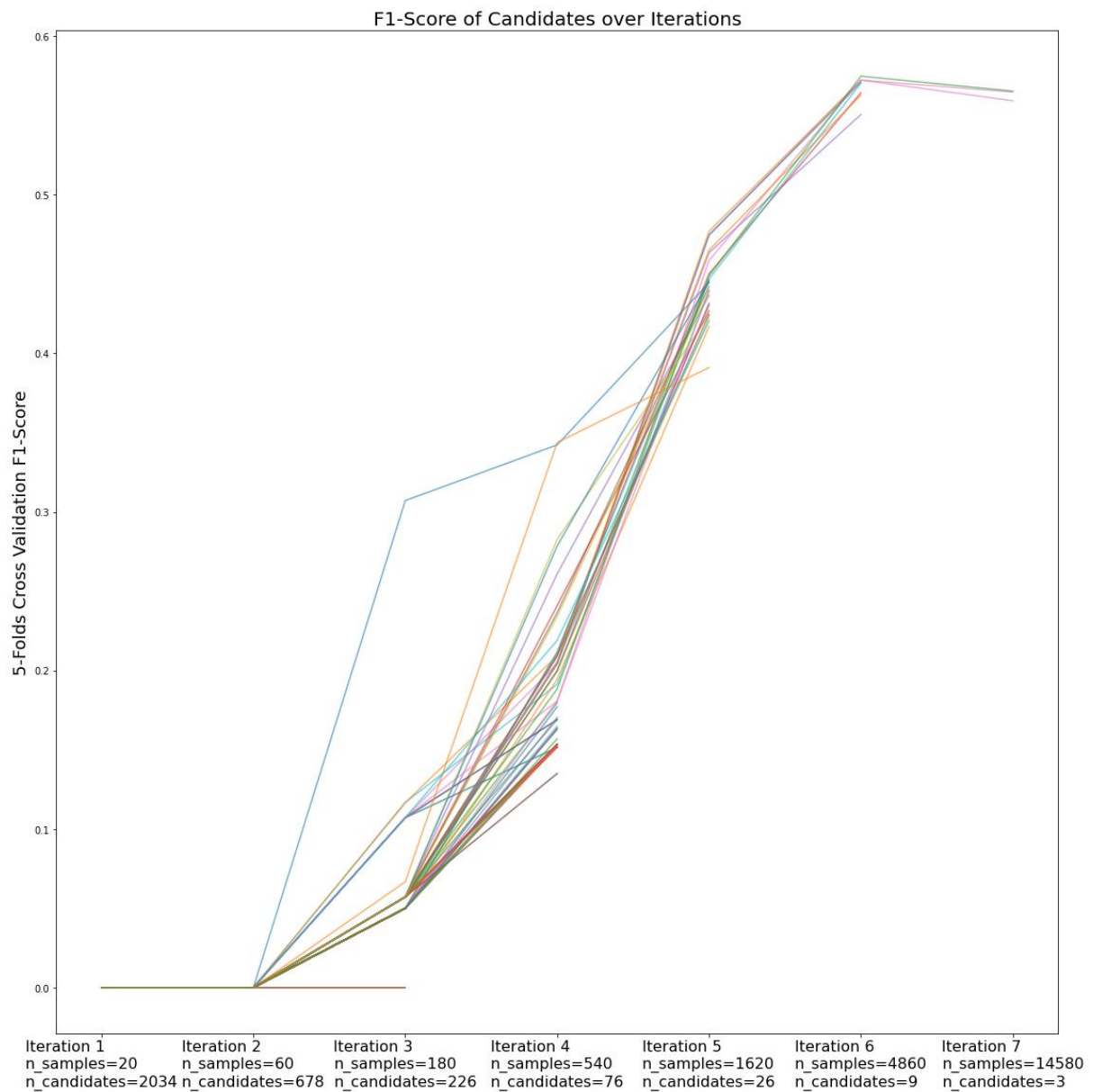




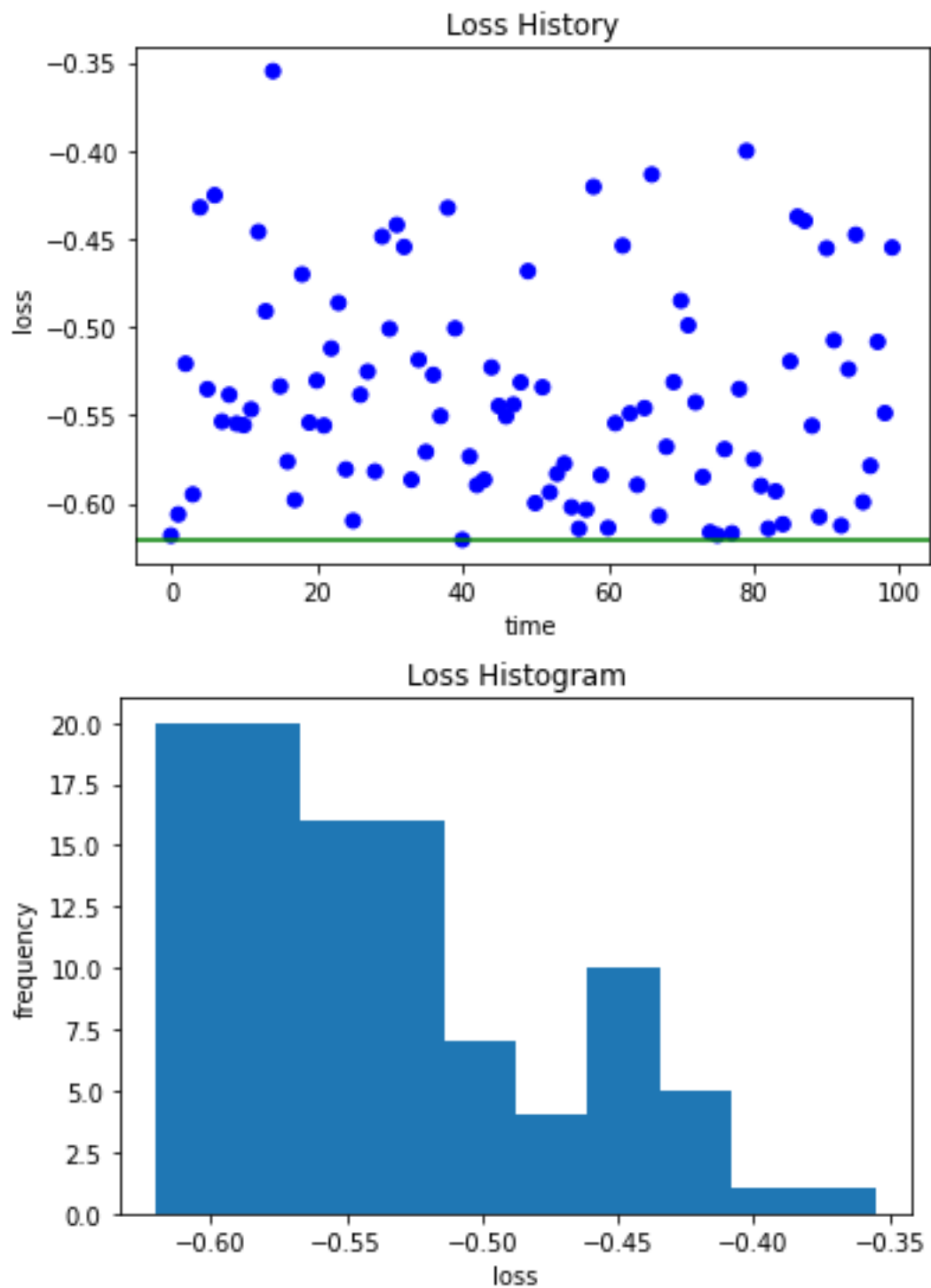
SH Iteration	Bracket-1		Bracket-2		Bracket-3		Bracket-4	
	$n_1$	$r_1$	$n_2$	$r_2$	$n_3$	$r_3$	$n_4$	$r_4$
1	27	1	12	3	6	9	4	27
2	9	3	4	9	2	27		
3	3	9	1	27				
4	1	27						

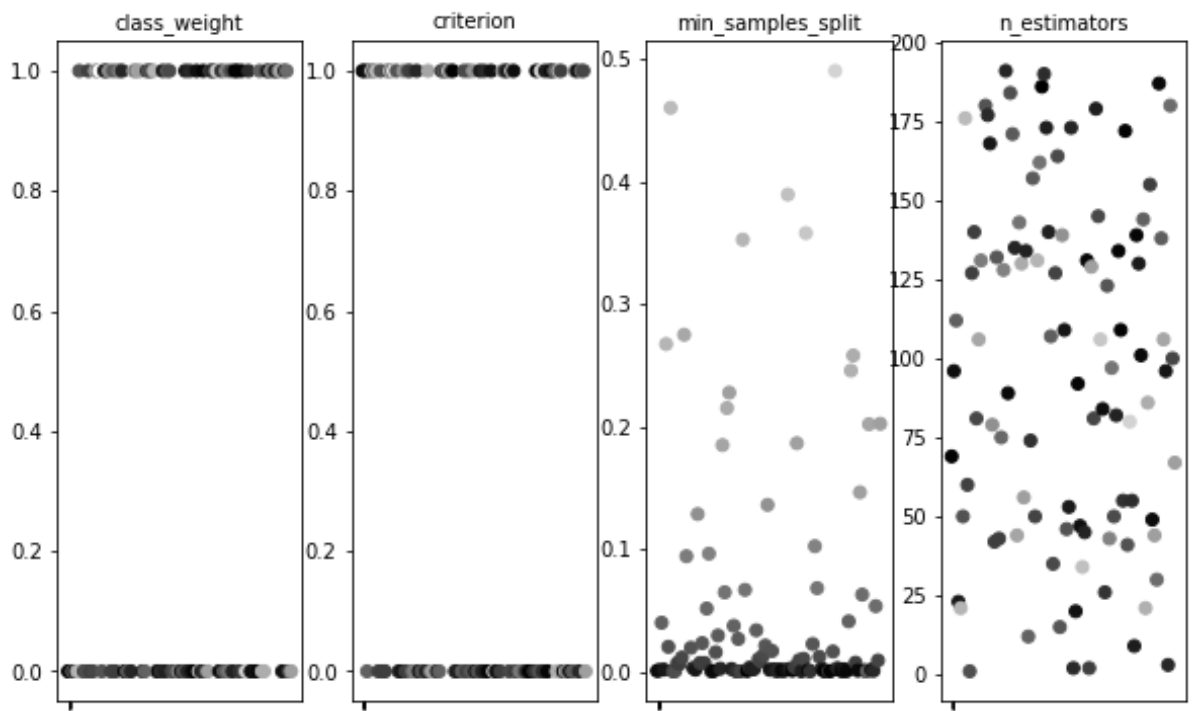


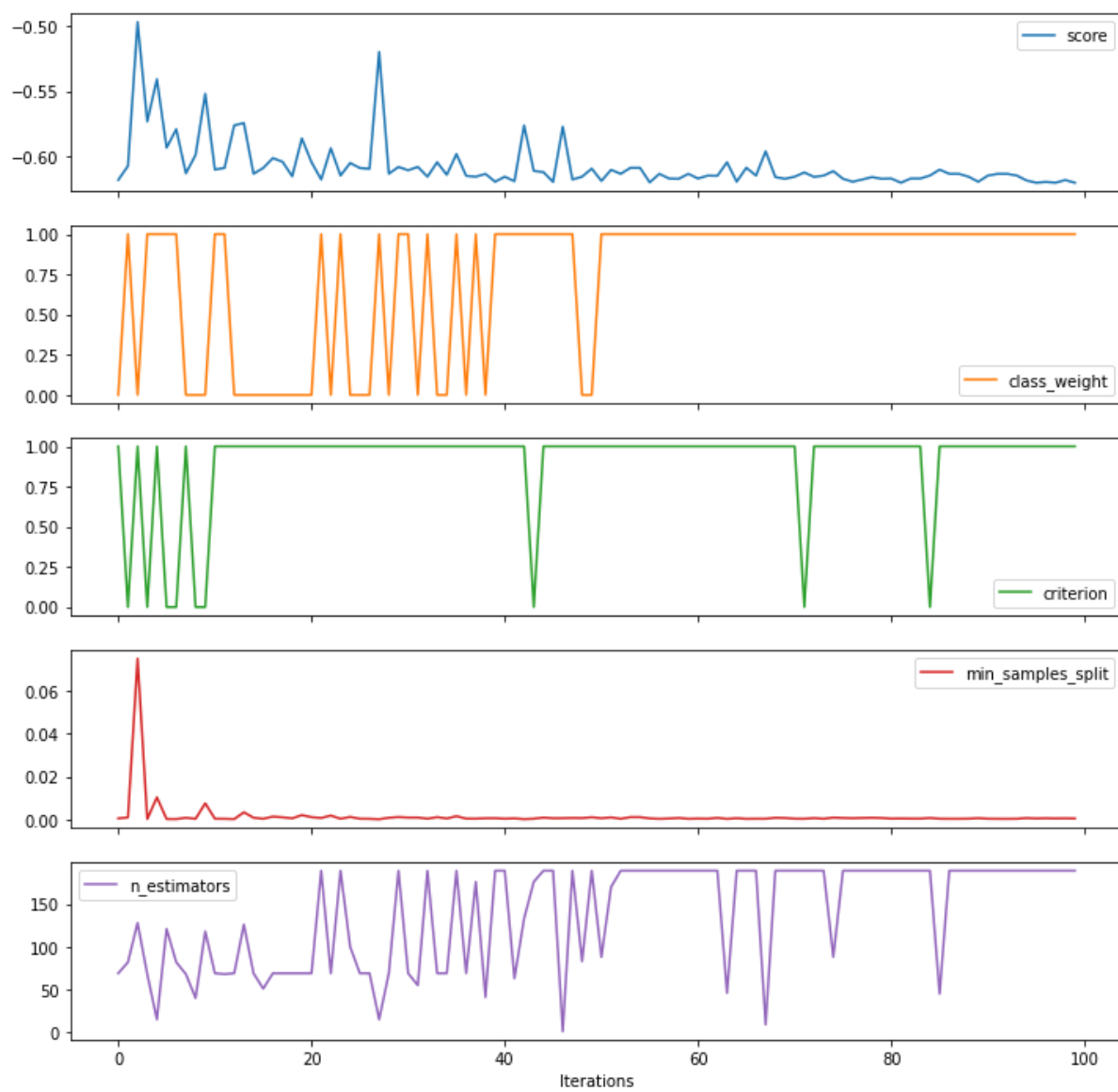
## Chapter 7: Hyperparameter Tuning via Scikit



## Chapter 8: Hyperparameter Tuning via Hyperopt



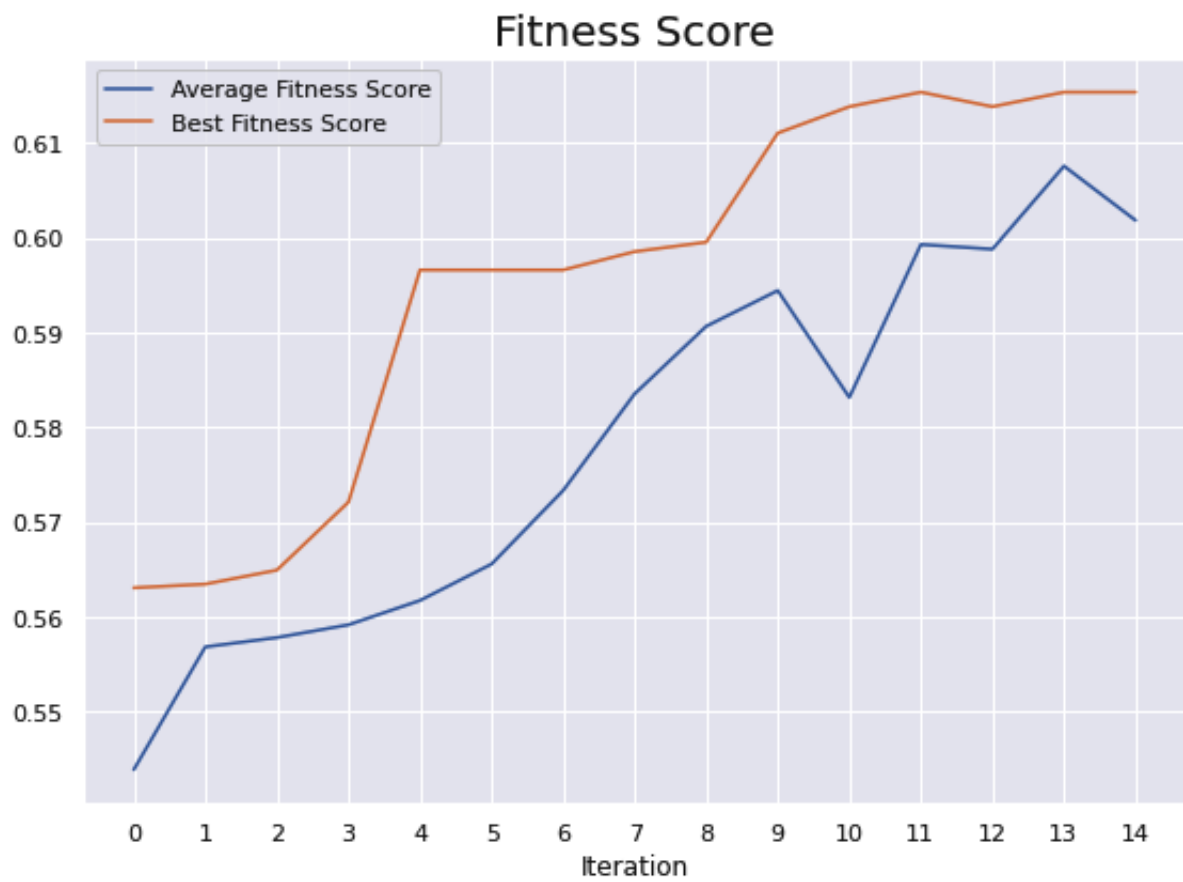




## Chapter 9: Hyperparameter Tuning via Optuna

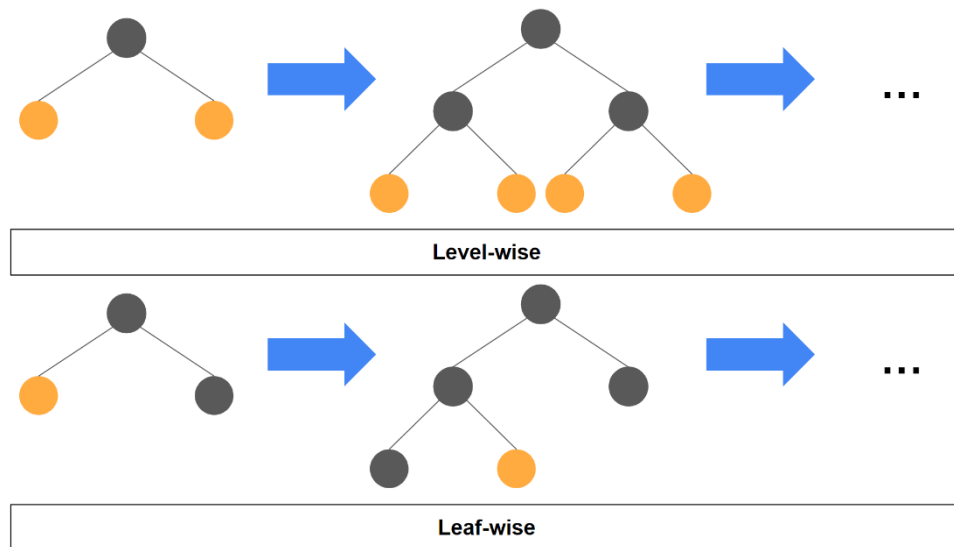
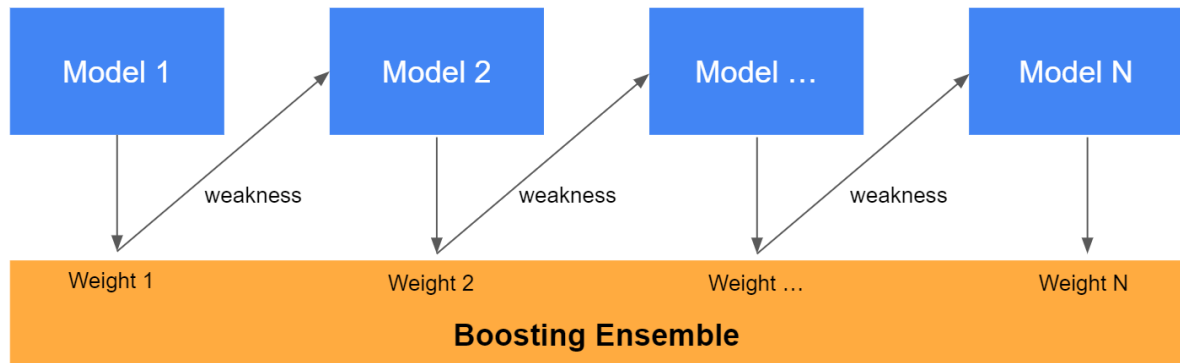
No images...

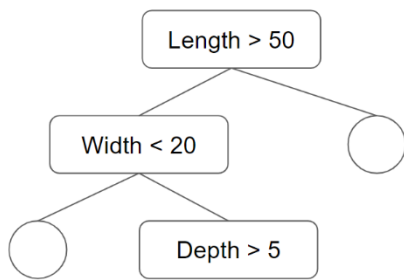
## Chapter 10: Advanced Hyperparameter Tuning with DEAP and Microsoft NNI



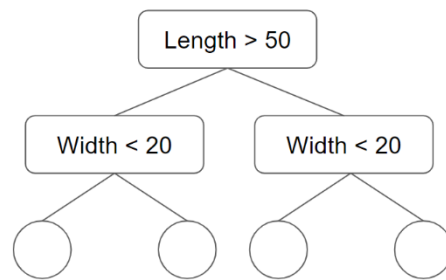


## Chapter 11: Understanding Hyperparameters of Popular Algorithms

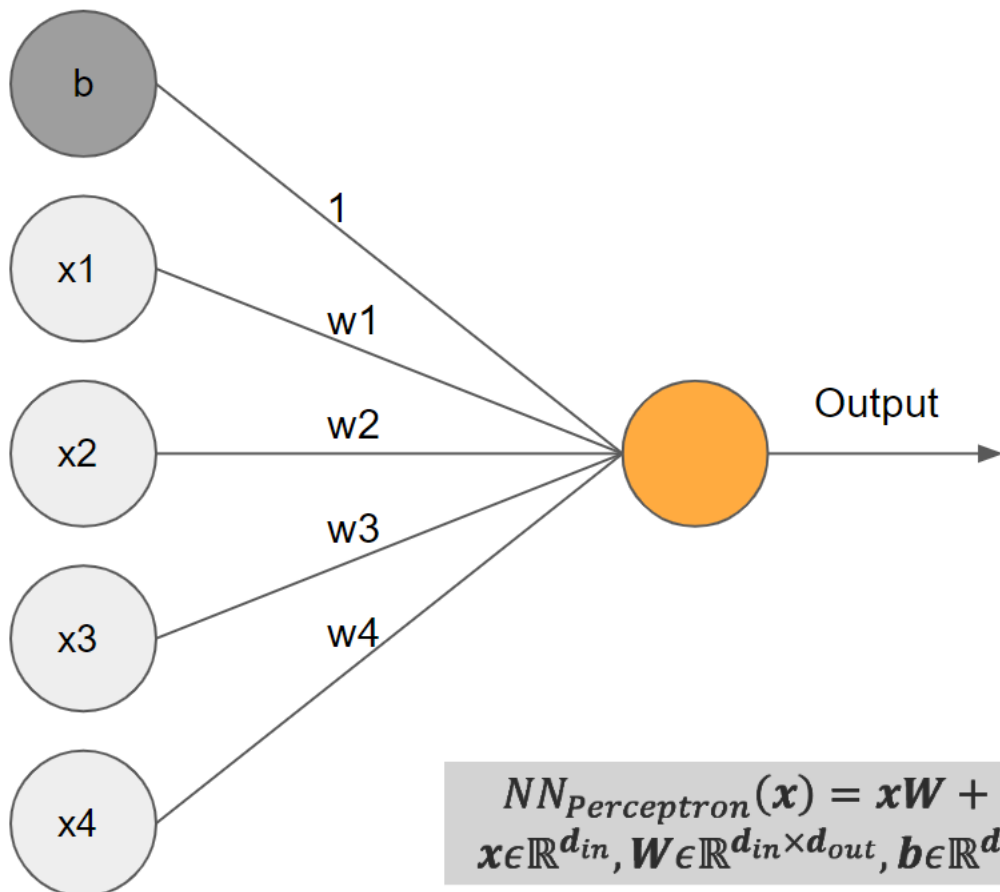




**Asymmetric Tree**



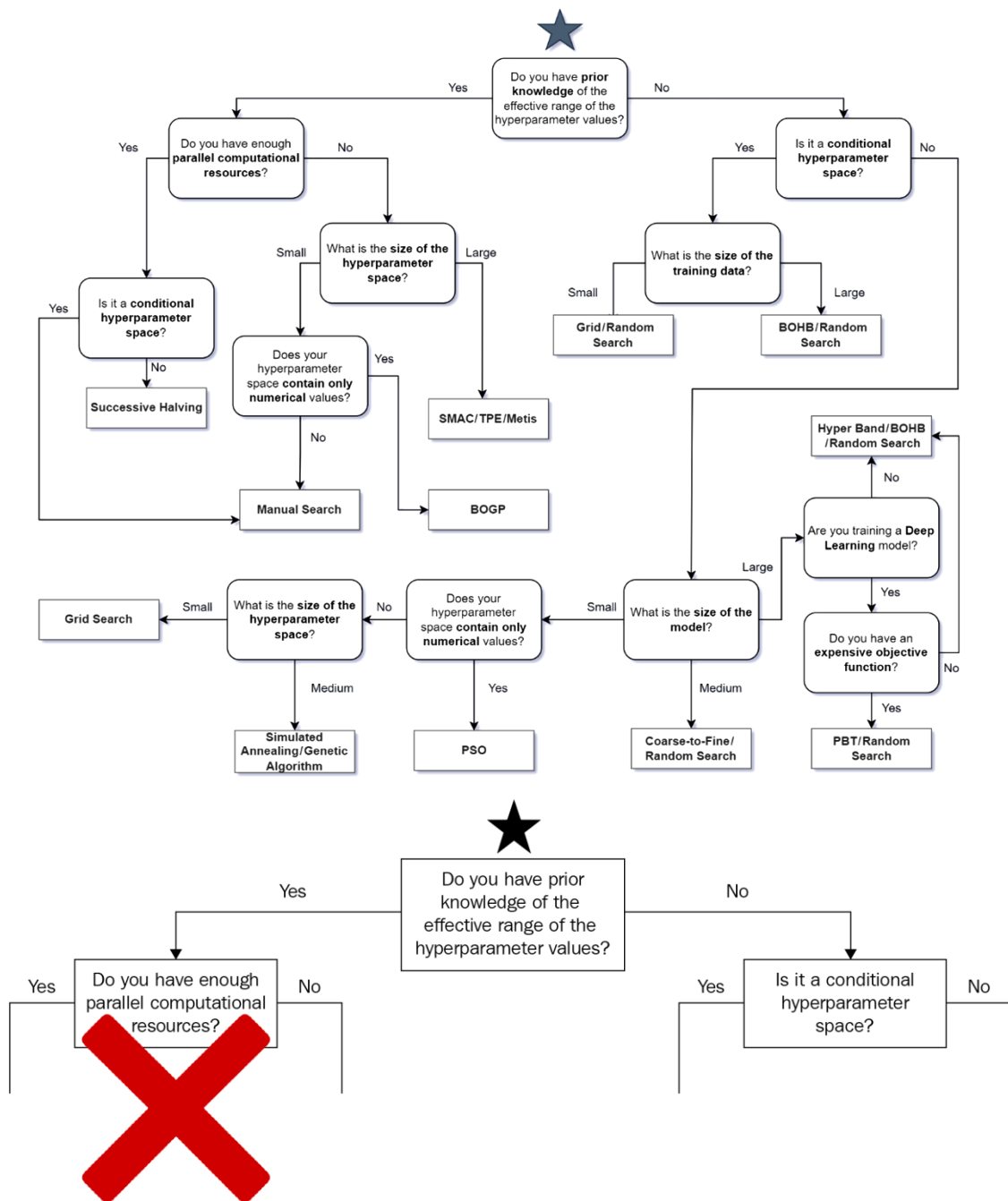
**Symmetric Tree**

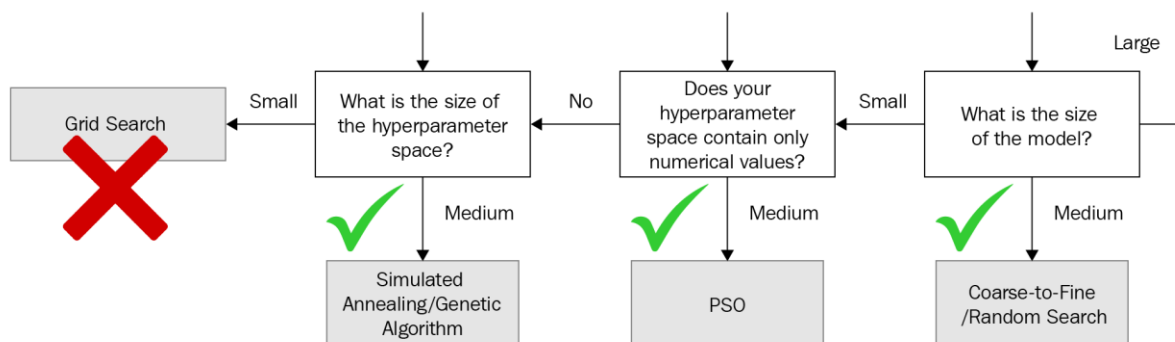
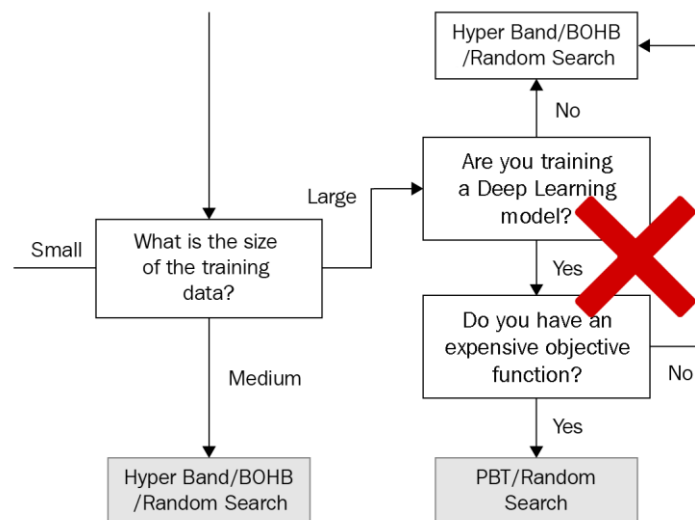
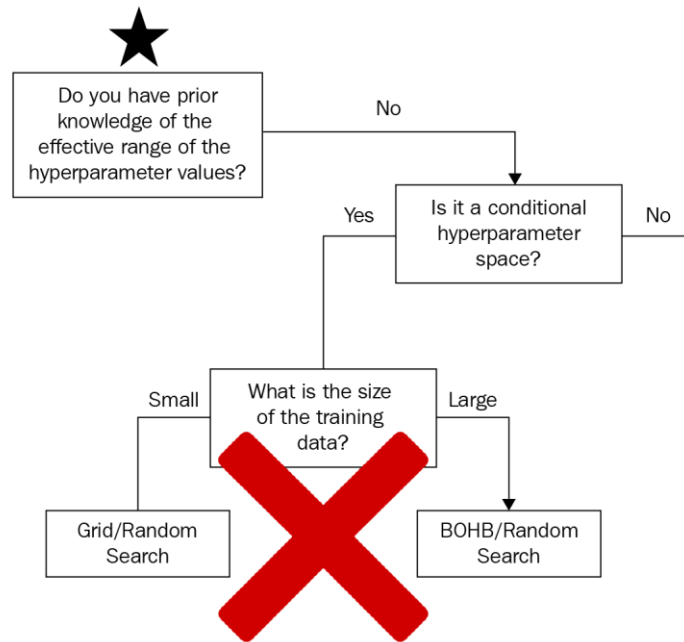


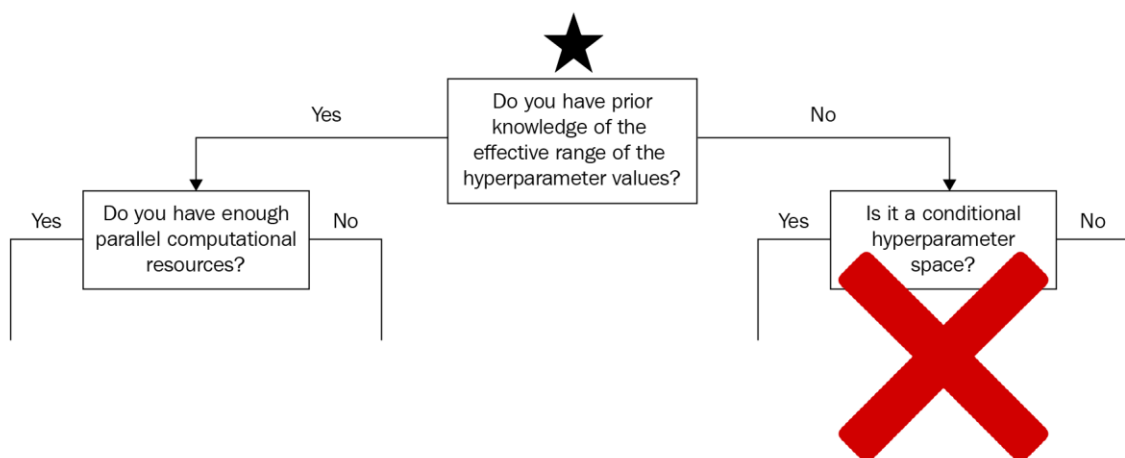
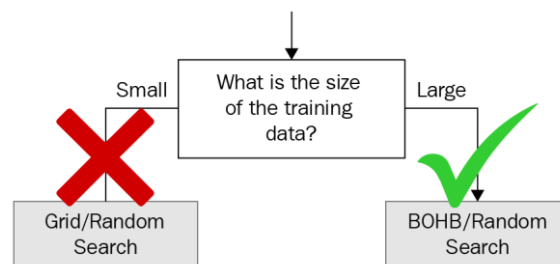
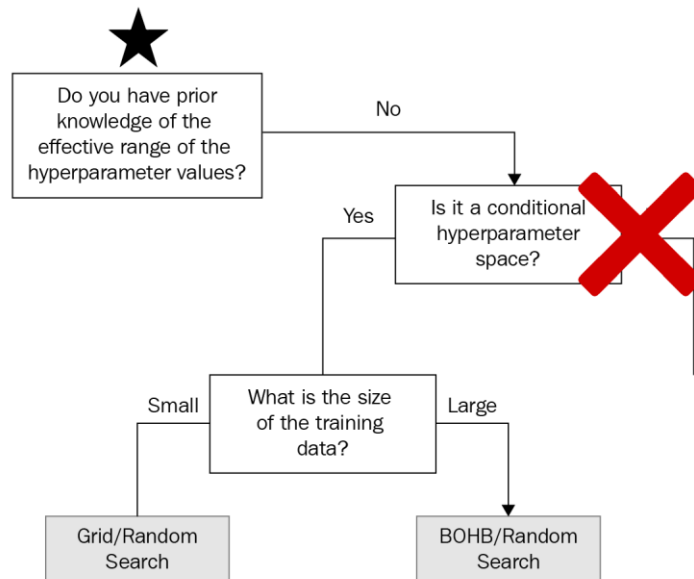
$$NN_{\text{Perceptron}}(x) = xW + b$$

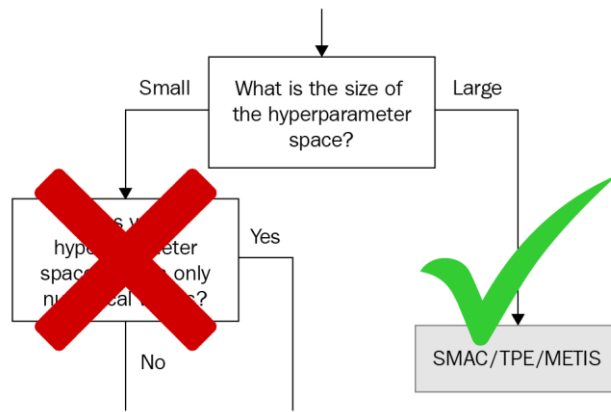
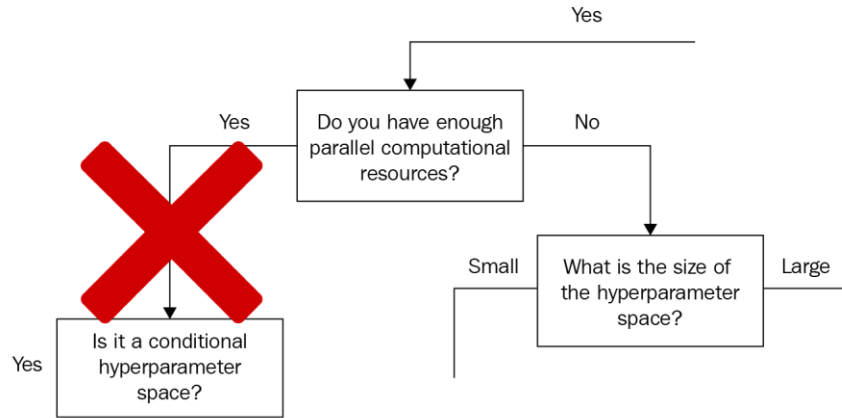
$$x \in \mathbb{R}^{d_{in}}, W \in \mathbb{R}^{d_{in} \times d_{out}}, b \in \mathbb{R}^{d_{out}}$$

# Chapter 12: Introducing Hyperparameter Tuning Decision Map











## Chapter 13: Tracking Hyperparameter Tuning Experiments

← → ↻

app.neptune.ai/ /-/projects



INDIVIDUAL



Projects 1

People

Subscription

Settings

Search

Sort by

Recently visited ▼

+


New project

example-project-tensorflow-keras


Private

Example project using TensorFlow / Keras.


10




0



1





Create new project

×

Project name

hpo-test-1

?

To log runs to this project, you need to pass **project name** into your Python code.  
run = neptune.init(project=' /hpo-test-1')  
[See docs to learn more.](#)

Project key

HPOT

Project color

▼

Project privacy

☒ Private - Only you can see this project.

☐ Public - Anyone on the Internet will be able to find and see this project.

Description

Hyperparameter tuning experiment example 1.

Cancel

Create

### Step 1: Install the client library

```
pip install neptune-client neptune-optuna
```

### Step 2: Create a run, then log whatever model building metadata you care about.

`train_optuna.py` [↓](#)

```
import optuna

import neptune.new as neptune
import neptune.new.integrations.optuna as optuna_utils

run = neptune.init(
    project="hpo-test",
    api_token="eyJhcGlFYWRkcmVzcyI6Imh0dHBzOi8vYXBwLm5lcHR1bmUuYWkiLCJhcGlfdXJ"
) # your credentials

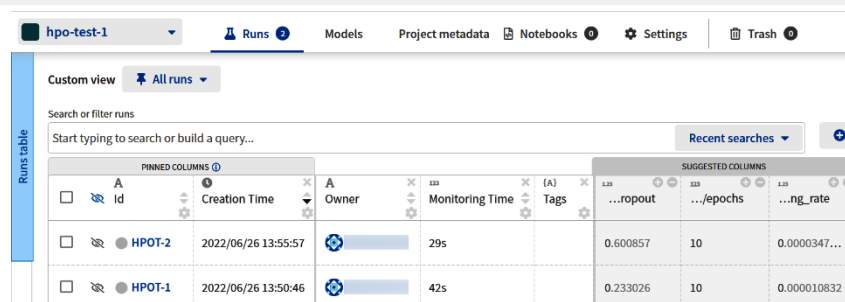
params = {"direction": "minimize", "n_trials": 20}
run["parameters"] = params

def objective(trial):
    param = {
```

### Step 3: Run it

```
python train_optuna.py
```

### Step 4: See your metadata displayed here in Neptune!



The screenshot shows the Neptune web interface for project 'hpo-test-1'. The 'Runs' tab is active, displaying a table of runs. The table has columns for 'Id', 'Creation Time', 'Owner', 'Monitoring Time', 'Tags', '...ropout', '.../epochs', and '...ng\_rate'. Two runs are listed: 'HPOT-2' and 'HPOT-1'.


Id	Creation Time	Owner	Monitoring Time	Tags	...ropout	.../epochs	...ng_rate
HPOT-2	2022/06/26 13:55:57		29s		0.600857	10	0.0000347...
HPOT-1	2022/06/26 13:50:46		42s		0.233026	10	0.000010832



 hpo-test-1

[< Back to runs](#)



 HPOT-1



Untitled

 [Details](#)


Last modified: 2022/06/26 13:51:28

Running time: 19s

 All metadata


 Charts

 Images

 Monitoring

 Source code

 Artifacts

 [Create your own dashboard!](#)

 [Add new dashboard](#)

hpo-test-1

Runs 2ModelsProject metadataNotebooks 0Set

Runs table

Compare runs<<

2 out of 2 runs

Charts

Images

Parallel coordinates

Side-by-side

Artifacts

Create your own dashboard!

Add new dashboard

Horizontal split

Compare runs 2

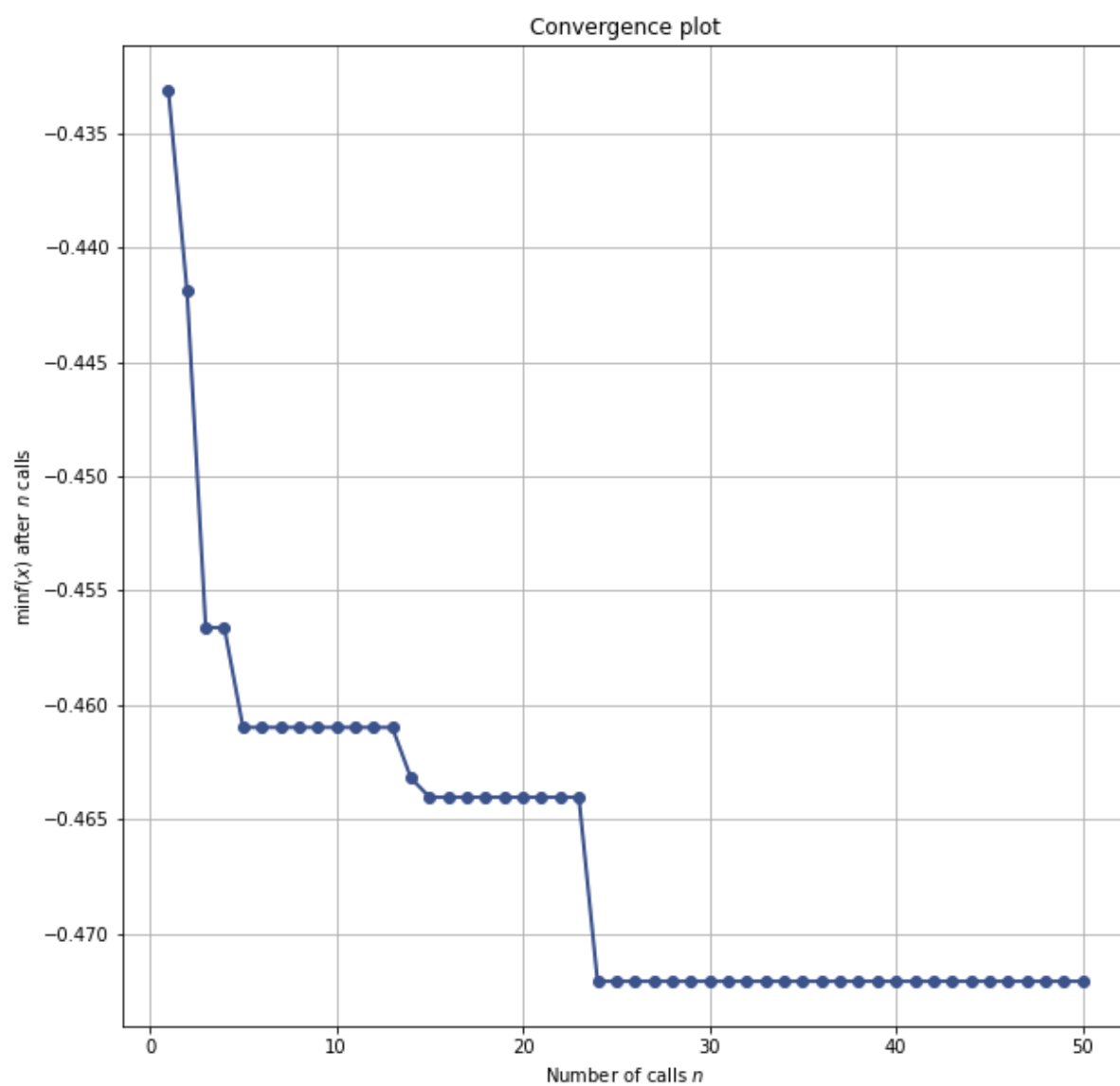
☐ Rows with diff only

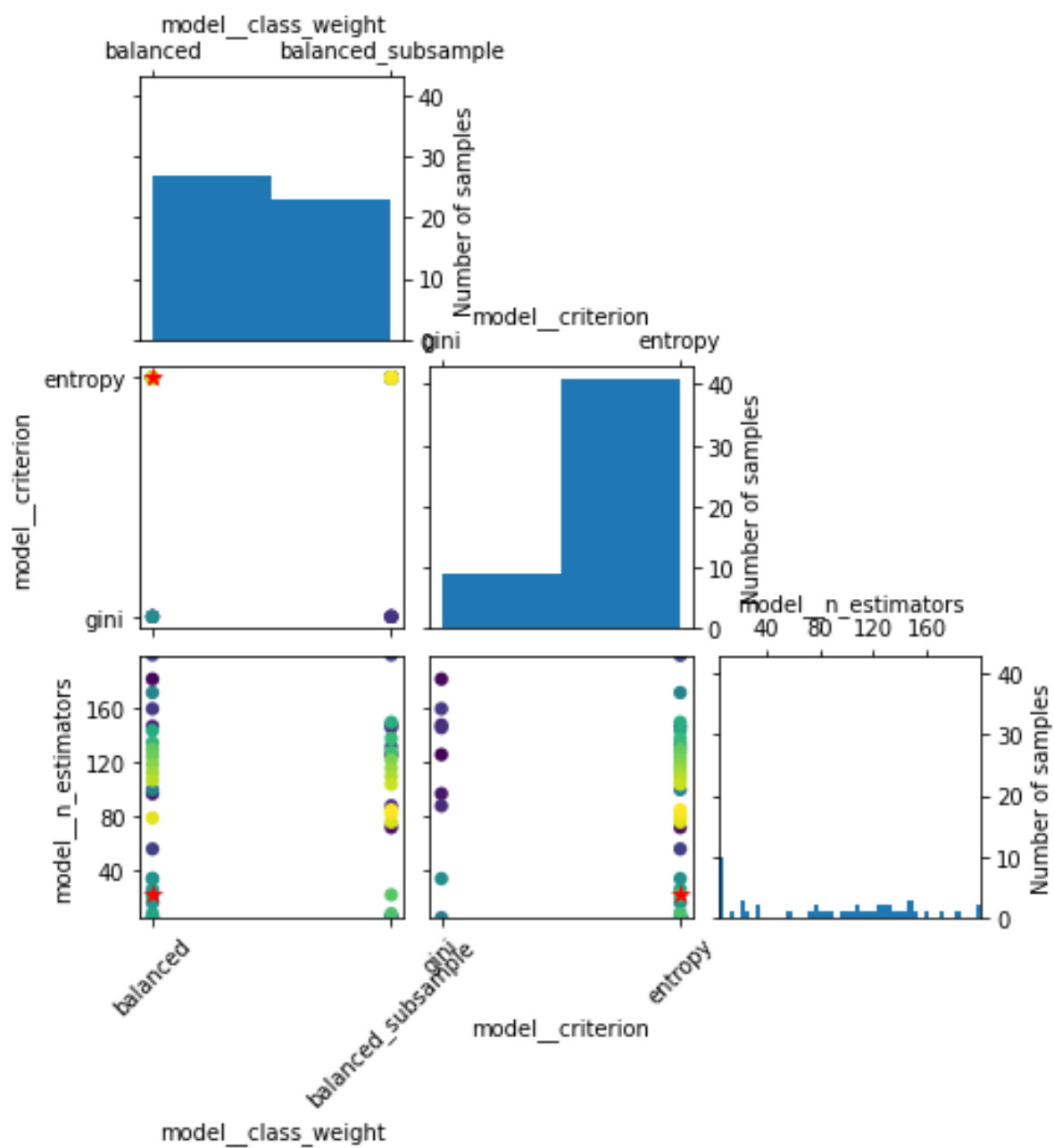
☐ Show cell changes

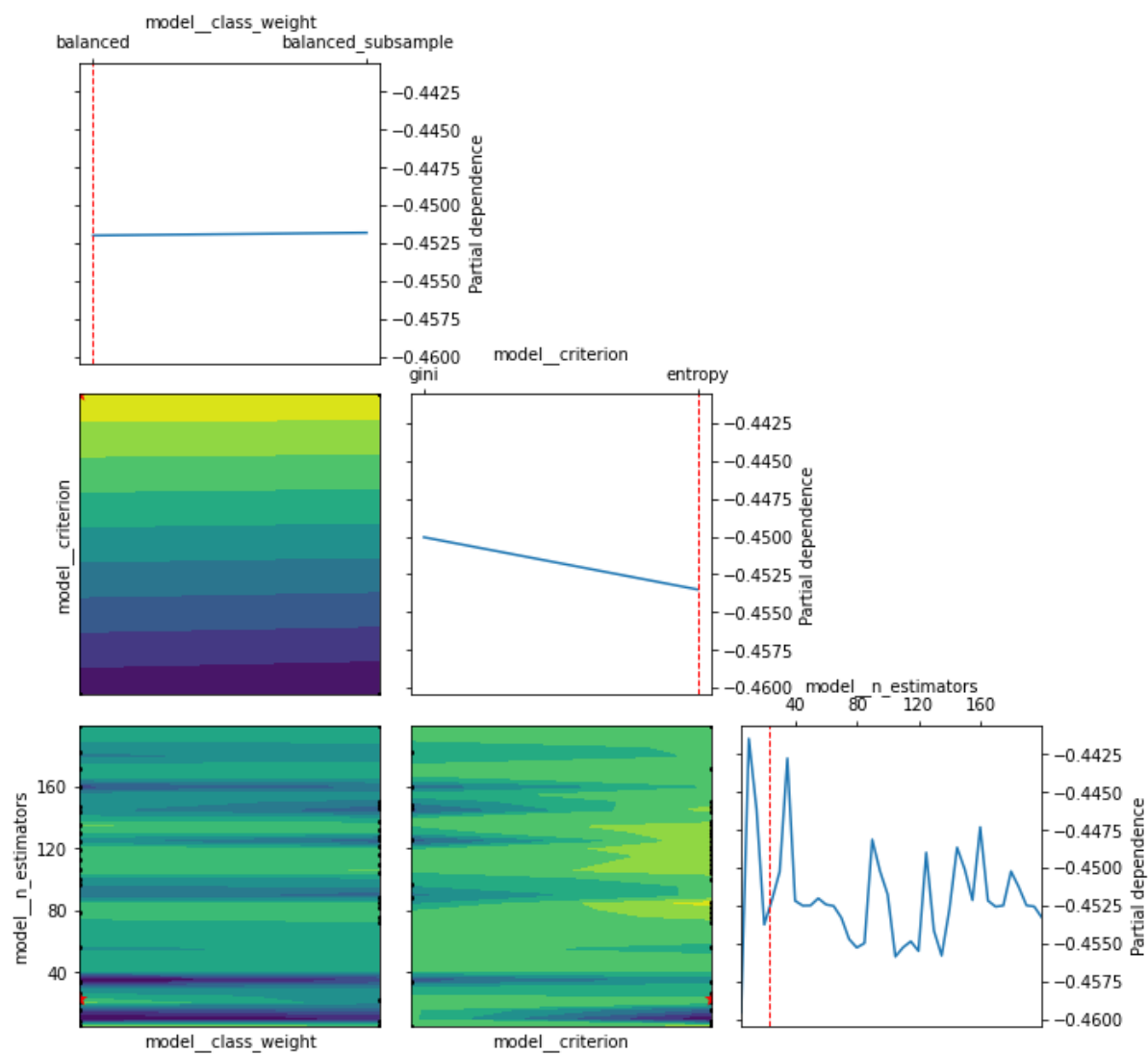
HPOT-2

HPOT-1

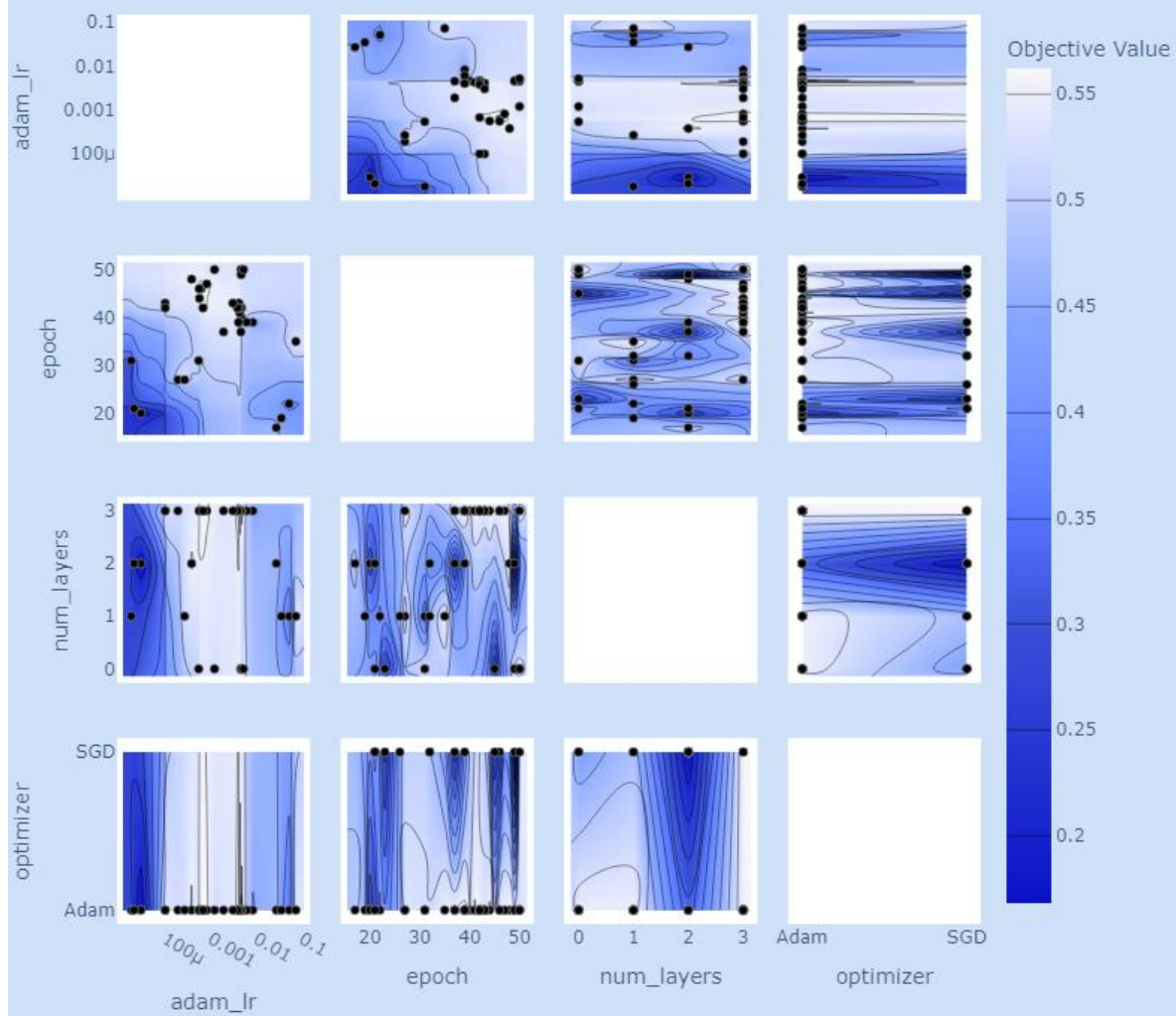
Creation Time	2022/06/26 13:55:57	2022/06/26 13:50:46
Owner		
Monitoring Time	29	42
best/params/dropout	0.600857	0.233026
best/params/epochs	10	10
...params/learning_rate	0.0000347923	0.000010832
...0/datetime_complete	2022/06/26 13:55:58	2022/06/26 13:50:47
...rials/0/datetime_start	2022/06/26 13:55:58	2022/06/26 13:50:47
...distributions/dropout	UniformDistribution(...	UniformDistribution(h...
.../distributions/epochs	IntUniformDistributio...	IntUniformDistributio...
...butions/learning_rate	LogUniformDistributi...	LogUniformDistributi...
best/trials/0/duration	0:00:00.000566	0:00:00.000769
...ls/0/params/dropout	0.226547	0.472201



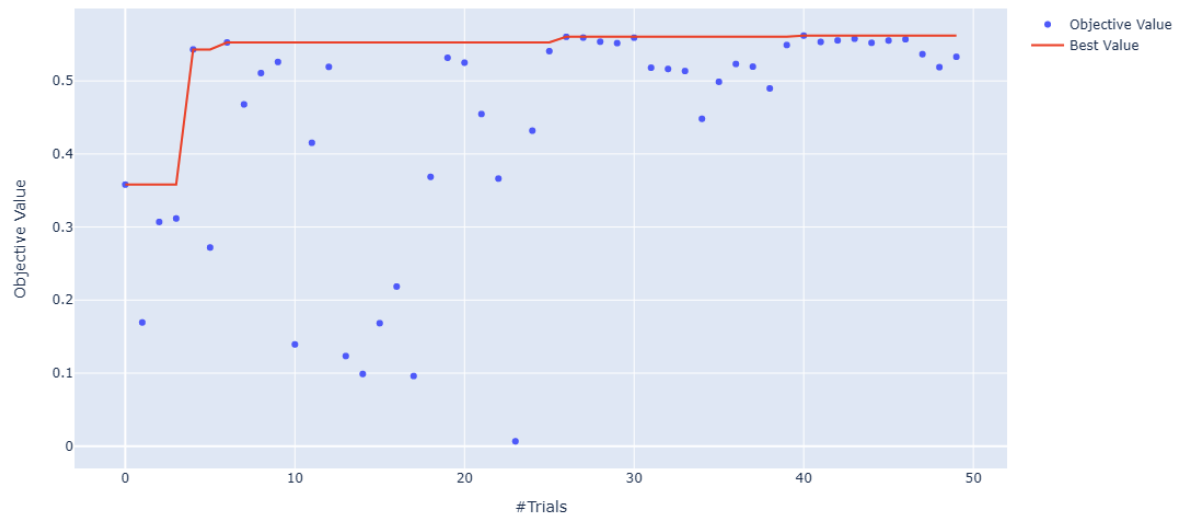




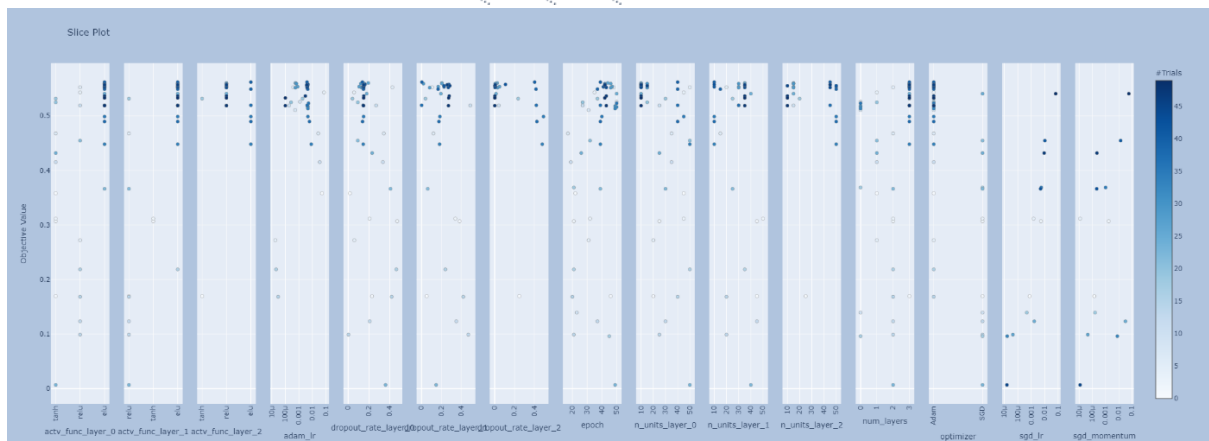
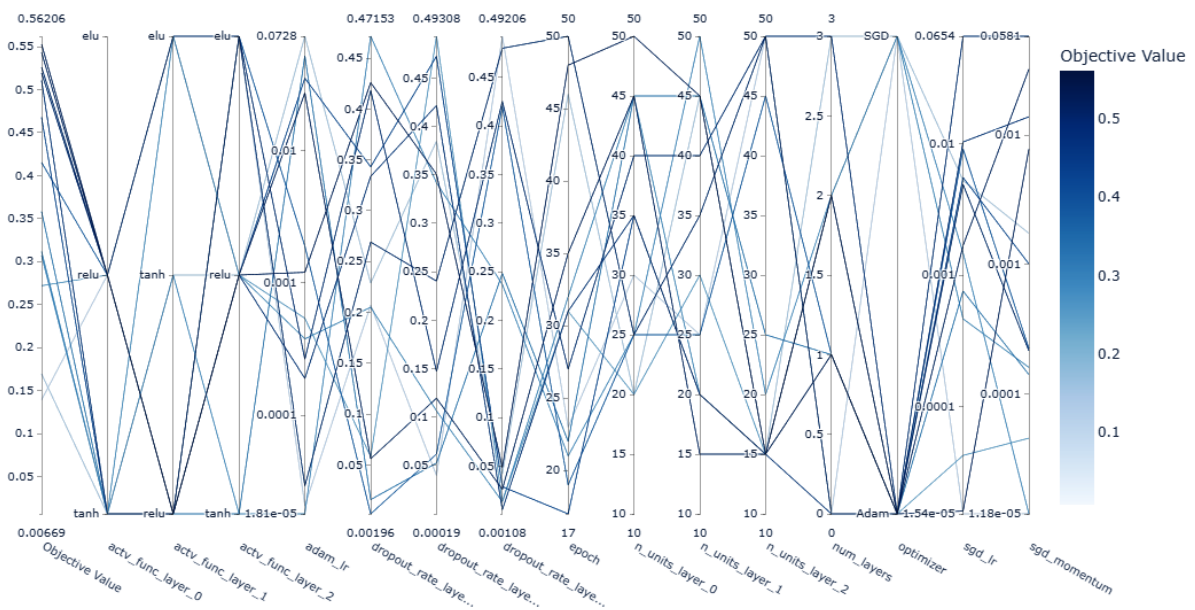
Contour Plot

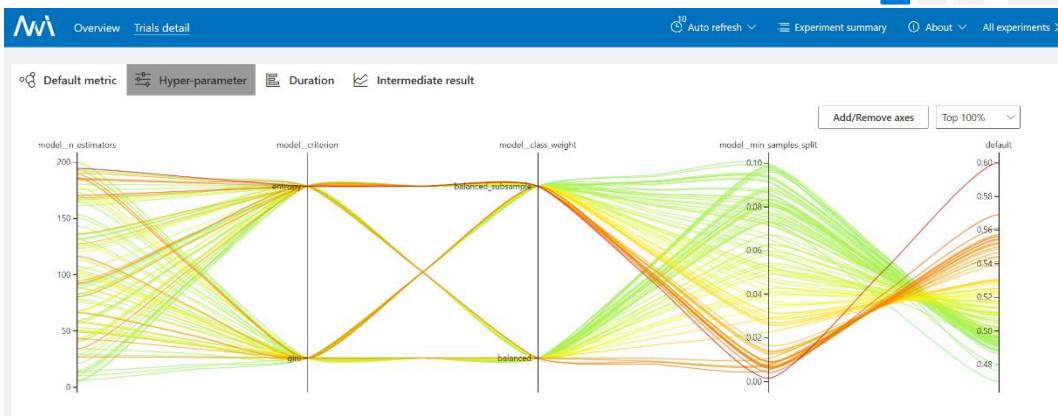
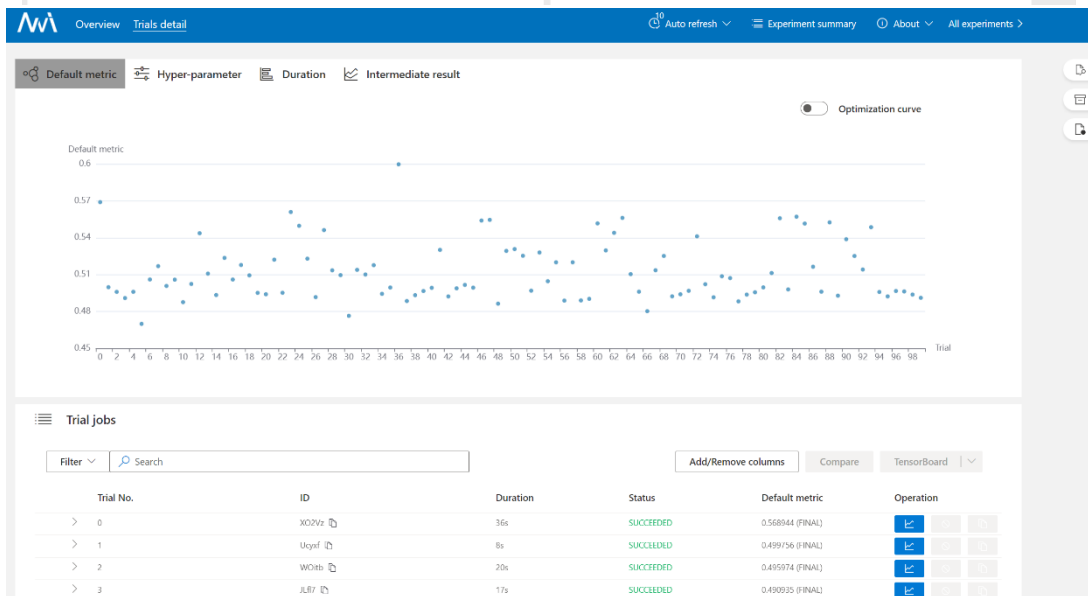
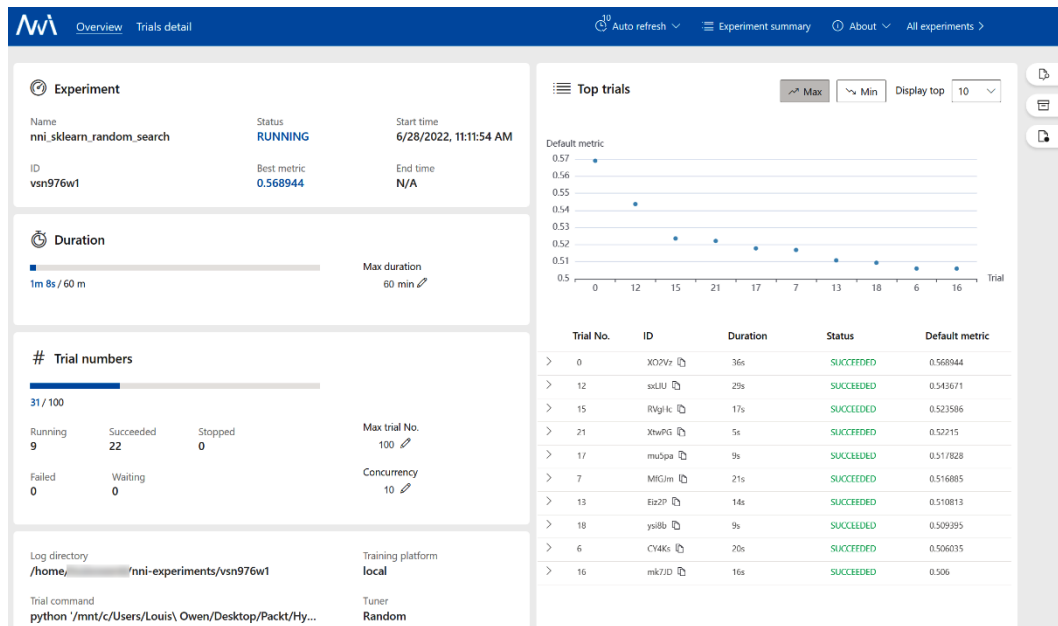


Optimization History Plot

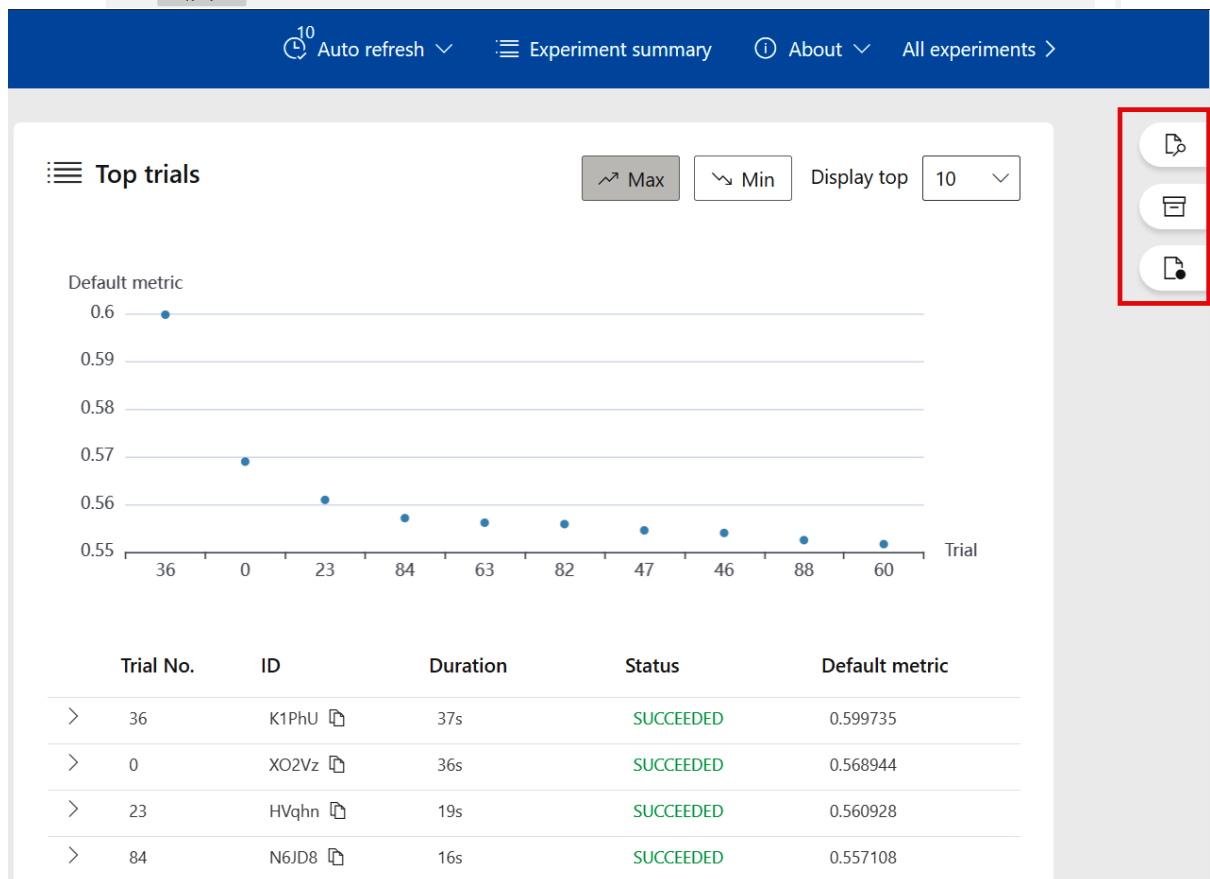
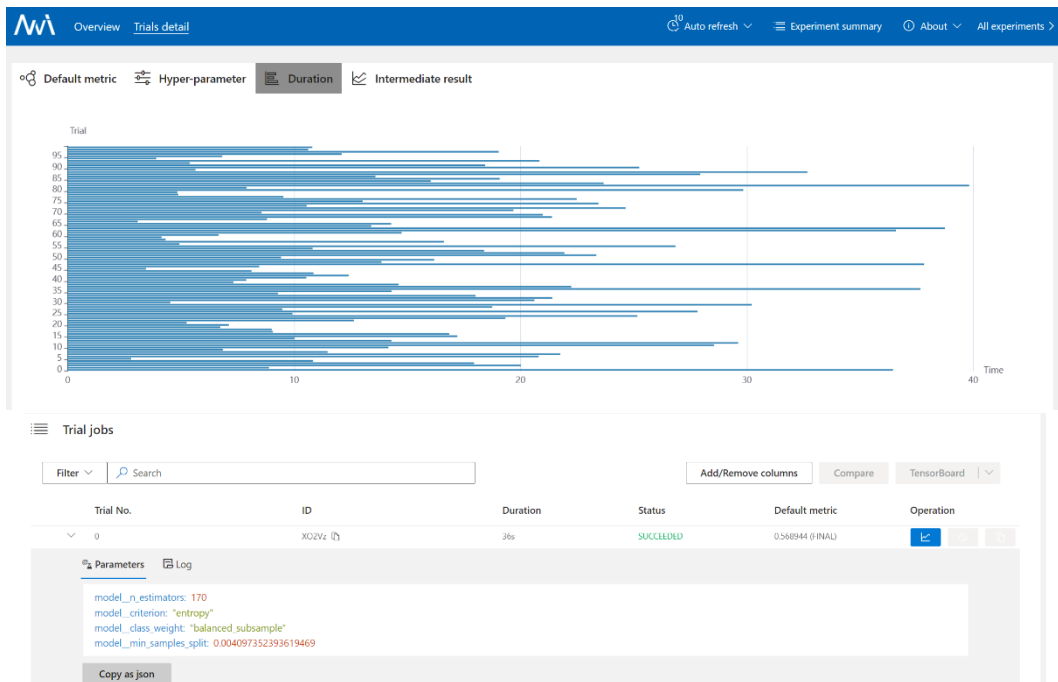


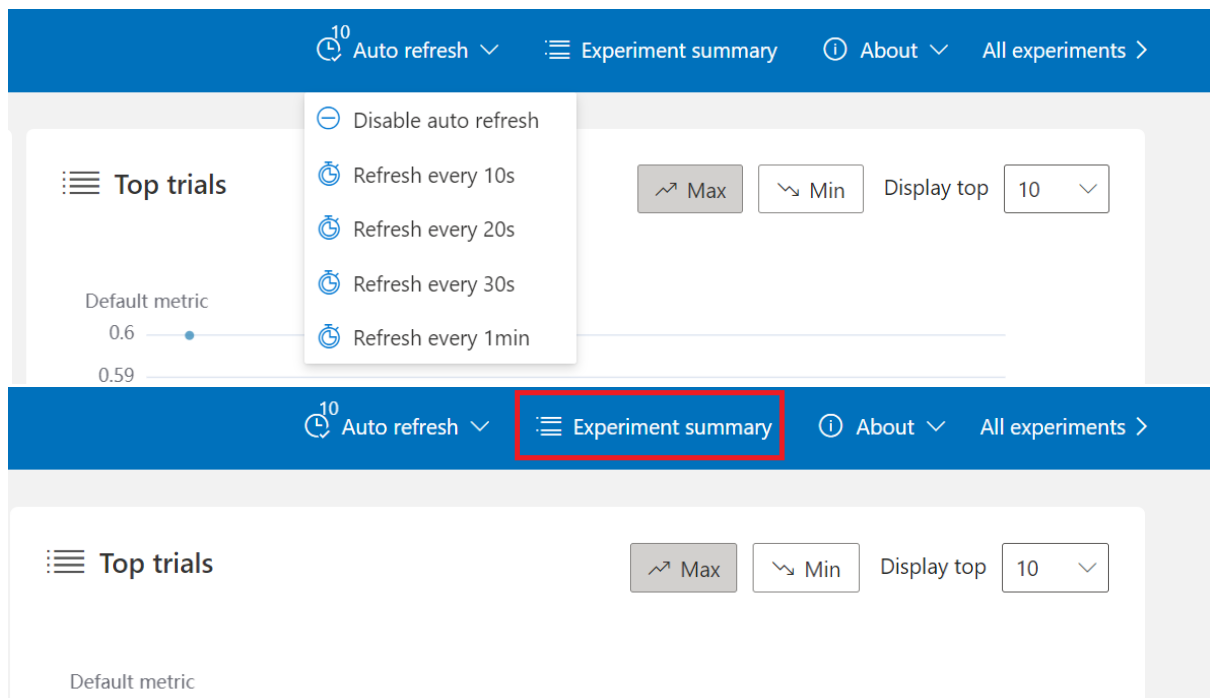
Parallel Coordinate Plot











## Chapter 14: Conclusions and Next Steps

