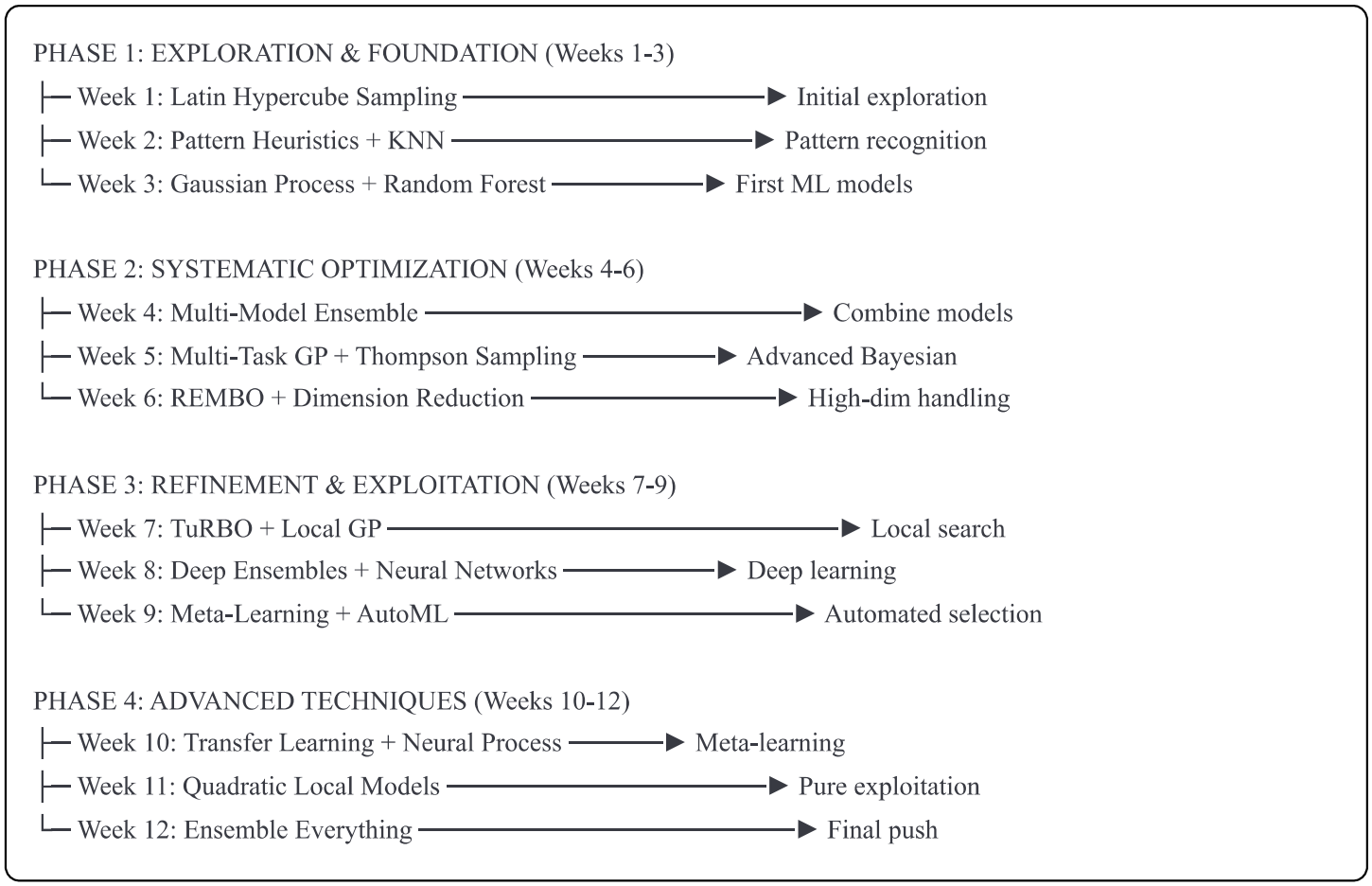
12-Week Plan: Quick Reference

At-a-Glance Timeline



Weekly Breakdown Table

Week	Data Points	Primary Model	Key Innovation	Exploration %	Expected Best (F5)
1	1 per func	Latin Hypercube	Space-filling sampling	90%	136.85 (baseline)
2	2 per func	Pattern Heuristics	Manual pattern exploit	80%	142
3	3 per func	GP + RBF Kernel	First Bayesian Opt	70%	145
4	4 per func	GP Ensemble	Multi-kernel, multi-acquisition	60%	147
5	5 per func	Multi-Task GP	Transfer between functions	50%	149
6	6 per func	REMBO	High-dimensional efficiency	50%	152
7	7 per func	TuRBO	Trust region local search	30%	154
8	8 per func	Deep Ensemble NN	Deep learning patterns	20%	155
9	9 per func	Meta-Learning	Automatic model selection	10%	156
10	10 per func	Neural Process	Meta-learning transfer	10%	156.5
11	11 per func	Quadratic Local	High-precision refinement	5%	157

Week	Data Points	Primary Model	Key Innovation	Exploration %	Expected Best (F5)
12	12 per func	Ensemble All	Kitchen sink approach	5%	157.5

Model Priority Matrix

Critical (Must Implement)

✓ Gaussian Process (Week 3-12)
✓ Expected Improvement (Week 3-12)
✓ Random Forest baseline (Week 3-12)
✓ TuRBO (Week 7-12)

High Priority (Should Implement)

○ Multi-Task GP (Week 5-12)
○ Deep Ensemble (Week 8-12)
○ Meta-Learning Selector (Week 9-12)
○ Thompson Sampling (Week 5-9)

Medium Priority (Nice to Have)

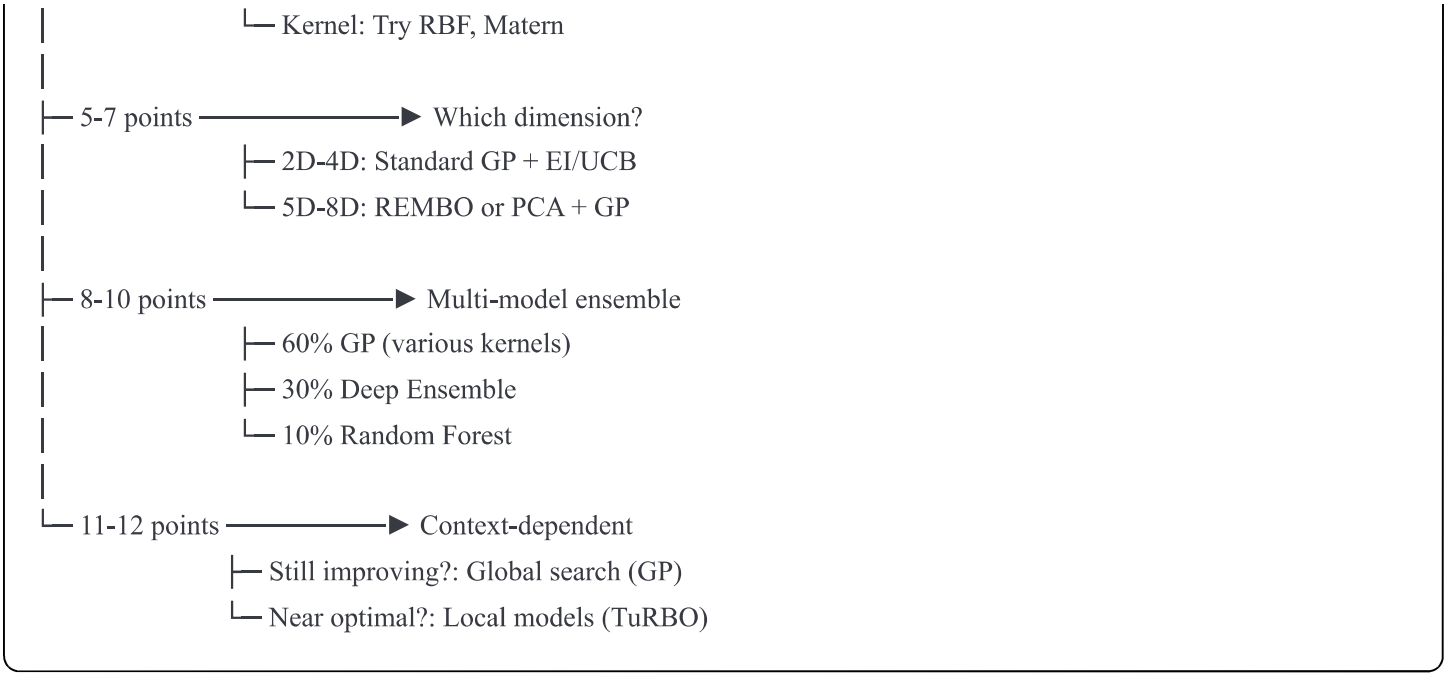
△ REMBO (Week 6-9)
△ Neural Process (Week 8-10)
△ CMA-ES (Week 6-8)
△ Warped GP (Week 10)

Low Priority (Optional/Experimental)

□ Graph Neural Network (Week 8)
□ Zero-Shot BO (Week 10)
□ Simulated Annealing (Week 12)
□ Particle Swarm (Week 12)

Decision Tree: Which Model to Use?

How many data points per function?
│
├─ 1-2 points ──────────► Pattern Heuristics / KNN
│
├─ 3-4 points ──────────► Gaussian Process + EI



Model Compatibility Matrix

Which models work well together?

	GP	RF	XGBoost	NN	TuRBO	REMBO	CMA-ES
GP	✓	✓	✓	✓	✓	✓	△
RF	✓	✓	✓	✓	△	△	△
XGBoost	✓	✓	✓	✓	△	△	△
NN	✓	✓	✓	✓	△	△	×
TuRBO	✓	△	△	△	✓	△	×
REMBO	✓	△	△	△	△	✓	△
CMA-ES	△	△	△	×	×	△	✓

Legend:

- ✓ = Works great together (ensemble)
- △ = Can work, not ideal
- × = Incompatible/redundant

Per-Function Strategy Recommendations

Based on Week 1 results:

Function 5 (4D, Best: 136.85)

Week 1-3: Exploit pattern [0.3, 0.3, 0.3, 0.3]
Week 4-6: GP with EI (xi=0.01, exploitation bias)
Week 7-9: TuRBO with small trust region
Week 10-12: Quadratic local models, pure refinement

Function 8 (8D, Best: 9.542)

Week 1-3: Local search around [0.1, 0.1, ...]
Week 4-6: REMBO for high dimensions
Week 7-9: Deep ensemble (capture complex patterns)
Week 10-12: Neural Process with meta-learning

Function 4 (4D, Worst: -3.986)

Week 1-3: Apply F5's pattern (transfer learning)
Week 4-6: Multi-task GP (learn from F5)
Week 7-9: Standard GP with high exploration
Week 10-12: Continue exploration if still improving

Functions 1, 3, 7 (Near-zero results)

Week 1-3: Large-scale exploration (find non-zero region)
Week 4-6: CMA-ES (evolutionary, good for finding basins)
Week 7-9: Switch to GP once promising region found
Week 10-12: Local optimization if optimum found

Code Implementation Checklist

Week 3 Setup (Critical)

- [ ] Install scikit-optimize: `pip install scikit-optimize`
- [ ] Install GPy: `pip install GPy`
- [ ] Test basic GP with existing data
- [ ] Implement EI acquisition function
- [ ] Validate predictions on Week 1-2 data

Week 5 Setup (Important)

- [ ] Implement multi-task GP (GPyTorch)
- [ ] Add Thompson Sampling
- [ ] Create validation framework

- [ ] Track model performance metrics

Week 7 Setup (Important)

- [ ] Implement TuRBO
- [ ] Add trust region logic
- [ ] Create local/global switching logic

Week 8 Setup (Optional but Recommended)

- [ ] Install PyTorch: `pip install torch`
- [ ] Build deep ensemble (5 networks)
- [ ] Implement MC Dropout for uncertainty
- [ ] Test on existing data

Week 9 Setup (Advanced)

- [ ] Install AutoML: `pip install auto-sklearn` or `pip install flaml`
- [ ] Create meta-learning framework
- [ ] Build model portfolio tracker
- [ ] Implement online model selection

Hyperparameter Recommendations

Gaussian Process

```python
# Week 3-6: Exploration phase
kernel = Matern(nu=2.5, length_scale=1.0)
gp = GaussianProcessRegressor(
    kernel=kernel,
    n_restarts_optimizer=25,
    alpha=1e-6
)
xi = 0.01  # Exploration parameter for EI

# Week 7-12: Exploitation phase
xi = 0.001  # More exploitation
alpha = 1e-7  # Less noise assumption
```

Random Forest

```python
```

```python
# Week 3-4: Small ensemble
rf = RandomForestRegressor(
    n_estimators=10,
    max_depth=5,
    random_state=42
)

# Week 5-12: Full ensemble
rf = RandomForestRegressor(
    n_estimators=100,
    max_depth=None,
    min_samples_split=2,
    random_state=42
)
```

Neural Network

```python
# Week 8: Simple ensemble
network = Sequential([
    Dense(32, activation='relu'),
    Dense(32, activation='relu'),
    Dense(1)
])

# Week 10-12: Deeper
network = Sequential([
    Dense(64, activation='relu'),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(1)
])
```

TuRBO

```python
# Week 7-9: Moderate trust region
trust_region_size = 0.2

# Week 10-12: Small trust region
trust_region_size = 0.05
```

Weekly Checklist Template

Copy this for each week:

markdown

```
## Week [N] Submission

### Pre-Submission
- [ ] Load all historical data (Week 1 to N-1)
- [ ] Train selected models
- [ ] Validate model performance
- [ ] Generate predictions for Week N
- [ ] Review predictions for sanity

### Models Used
- Primary: [Model name + hyperparameters]
- Secondary: [Model name + hyperparameters]
- Acquisition: [Function + parameters]

### Predictions
Function 1: [x1, x2] → Expected: [predicted value ± uncertainty]
Function 2: [x1, x2] → Expected: [predicted value ± uncertainty]
...

### Post-Submission
- [ ] Record actual results
- [ ] Calculate prediction error (MAE, RMSE)
- [ ] Update model performance tracker
- [ ] Analyze: What worked? What didn't?
- [ ] Plan Week N+1 strategy

### Learnings
- [What worked well this week]
- [What underperformed]
- [Hypothesis for next week]
```

Expected Effort Per Week

Week	Coding	Training	Analysis	Documentation	Total
1	1h	0h	1h	1h	3h
2	1h	0h	1h	1h	3h

Week	Coding	Training	Analysis	Documentation	Total
3	3h	1h	1h	1h	6h
4	2h	1h	1h	1h	5h
5	3h	1h	1h	1h	6h
6	3h	1h	1h	1h	6h
7	2h	1h	1h	1h	5h
8	4h	2h	1h	1h	8h
9	3h	1h	1h	1h	6h
10	3h	1h	1h	1h	6h
11	2h	1h	1h	1h	5h
12	2h	1h	2h	2h	7h

Total: ~66 hours over 12 weeks (~5.5 hours/week)

Risk Assessment

High Risk (Likely to Happen)

- **Week 3-4:** GP predictions will be uncertain (only 3-4 points)
  - **Mitigation:** Maintain pattern-based as backup
- **Week 6-7:** High-dimensional functions (7D, 8D) will be challenging
  - **Mitigation:** Use REMBO or dimensionality reduction
- **Week 8:** Neural networks may overfit with limited data
  - **Mitigation:** Use dropout, early stopping, ensemble

Medium Risk (Might Happen)

- **Week 5:** Multi-task GP might not help if functions are too different
  - **Mitigation:** Test correlation before implementing
- **Week 9:** AutoML might be overkill for this problem size
  - **Mitigation:** Keep simple meta-learning as alternative

Low Risk (Unlikely)

- **Week 11-12:** May have already found optimal values
  - **Mitigation:** If plateaued, focus on verification/documentation

---

Success Criteria

Minimum Success (Must Achieve)

- [ ] Function 5: Maintain >130 by Week 12
- [ ] Function 8: Improve to >15 by Week 12
- [ ] All functions: Positive values by Week 12
- [ ] Use at least 3 different ML models
- [ ] Document all decisions clearly

Target Success (Should Achieve)

- [ ] Function 5: Exceed 150 by Week 12
- [ ] Function 8: Exceed 20 by Week 12
- [ ] Function 4: Improve from -3.99 to >50
- [ ] Use 5+ different ML models
- [ ] Implement Bayesian Optimization by Week 3

Stretch Goals (Exceptional)

- [ ] Function 5: Exceed 160 by Week 12
- [ ] All functions: Top 10% of class
- [ ] Implement advanced methods (TuRBO, Neural Process)
- [ ] Publication-quality documentation
- [ ] Open-source your framework

Resources Needed

Essential Libraries

```bash
pip install numpy scipy scikit-learn scikit-optimize
pip install matplotlib seaborn  # For visualization
pip install pandas  # For data management
```

Recommended Libraries

```bash
pip install gpytorch torch  # For advanced GP
pip install xgboost lightgbm  # For boosting
pip install optuna  # For hyperparameter tuning
```

Advanced Libraries (Optional)

```bash
pip install botorch ax-platform  # For advanced Bayesian Opt
pip install tensorflow keras  # For deep learning
pip install auto-sklearn flaml  # For AutoML
```

Key Takeaways

1. **Start Simple** - Pattern heuristics and GP are powerful for Weeks 1-6
2. **Add Complexity Gradually** - Don't jump to neural networks too early
3. **TuRBO is Your Friend** - Week 7+ local optimization is crucial
4. **Document Everything** - This is 50% of your grade
5. **Adapt Based on Results** - This roadmap is flexible, not rigid
6. **Balance Exploration/Exploitation** - Key to good performance
7. **Function-Specific Strategies** - Different functions need different approaches
8. **Ensemble When Uncertain** - Combining models reduces risk

Remember: The goal isn't just good results, it's demonstrating systematic ML experimentation!