System Workflow WITH Full Ollama Cache

PHASE 1: ONE-TIME CACHE GENERATION

⏱ Time: 90-120 minutes (once)

generate_real_data.py

Loop: 5000 jobs × 4 prompts = 20,000 pairs

1. Load HDFS jobs
(5000 log entries)

2. Load 4 prompt templates
[simple, standard, fewshot_1, fewshot_3]

3. **For each (job, prompt) pair:**
• Check cache (skip if exists)
• Call Ollama API
• Save to cache/{hash}.json

🌐 Ollama API
(Local LLM)

Model: gemma3
~3 sec/call

📁 results/ollama_cache/

Creates ~20,000 JSON files:
{hash}.json → {label, response}

Cache ready → Fast

✅ OUTPUT:
20,000 cached LLM responses
Ready for instant reuse!

PHASE 2: FAST TRAINING AND OPTIMIZATION (Repeatable)

⚡ Time: 10-15 seconds per run (NO API CALLS)

1. create_training_data()
GROUNDING_RATIO = 1.0 ✓

Loop: 20,000 job-prompt pairs

For each pair:
✓ query_gemini(job, prompt)
✓ Checks cache FIRST
✓ Returns instantly (no API wait)
✓ label = 1 if correct, 0 if wrong

Result: X (20000×7), y (20000×1)

📁 **Cache Hit (100%)**

All 20,000 pairs cached
0 API calls needed

2. train_predictor(X, y)
XGBoost on 100% REAL data
Expected: 85%+ accuracy

3. predict_probabilities()
P(success | job, prompt)
Matrix: 5000 × 4

Baseline
(4 strategies)

Each prompt
for ALL jobs

Random
Search

1000 iter

NSGA-II

Pop: 100
Gen: 50

BEST!

SPEA2

Archive
based

4. Pareto Front Extraction
+ Metrics (IGD, Δ, M_n)
+ Visualization

📊 **Final Outputs:**
• tables/baseline_results.csv
• tables/table3_optimizer_comparison.csv
• figures/pareto_with_baselines.png

✅ KEY BENEFITS

1. 100% Real LLM Data
No synthetic heuristics
Learns actual LLM behavior

2. Higher Accuracy
Expected: 79% → 85%+
Better predictions

3. Instant Reruns
0 API calls after cache
10 sec vs 60 sec

4. Reproducible Experiments
Same cache → same results
Scientific rigor

5. Fast Iteration
Tune hyperparameters instantly
Test 100 configs in 20 min

📊 BEFORE vs AFTER

Metric	Before (5%)	After (100%)
Real Training Data	1,000	20,000
Synthetic Data	19,000	0
Model Accuracy	79%	85%+
API Calls/Run	1,000	0
Training Time	60 sec	10 sec
Cache Files	1,000	20,000
Setup Cost	5 min	120 min (once)