Memory Without Storage — Reproducibility Package

Paper: *Memory Without Storage, Learning Without a Learner: Observer Inference Across Four Computational Substrates*
Author: D. Neale, Goleudy.ai
Date: March 2026

Overview

This package contains all scripts needed to reproduce every result, table, and figure in the paper. All scripts are pure Python with NumPy as the only required dependency (SciPy used only for Spearman correlation in one script).

Requirements: Python 3.10+, NumPy

No external datasets. All data is generated from first principles by the scripts.

Scripts

Core Analysis (produces all main results)

Script	Produces	Paper sections
`final_analysis.py`	M/L/D for all 287 systems across 4 substrates. Multi-observer hierarchy test. Outputs: `final_ca.csv`, `final_gol.csv`, `final_gs.csv`, `final_sort.csv`	Tables in 4.1, 4.2. Cross-substrate top-20 in 4.2. Multi-observer results in 4.4.
`mld_dual_memory.py`	M_pred (model obsolescence) and M_cohen (Cohen's d) for 23 representative systems. Spearman correlation. M∩L∩D overlap using M_pred.	Tables in 4.2 (M_pred column), 4.5. Appendix C Tables C.6, C.7.
`gs_frequency_sweep.py`	Habituation frequency sweep: 7 tap intervals, 12 taps each, on stable Gray-Scott spots.	Section 4.3 frequency table. Appendix C Table C.8.

Script	Produces	Paper sections
`gs_habituation.py`	Extended habituation protocol: 3 conditions (standard, fast, strong taps) plus dishabituation test.	Section 4.3 supplementary detail.
`mld_hierarchy_investigation.py`	Feature-by-feature L decomposition for 5 systems. Identifies which features carry learning signal vs. add noise.	Section 4.4 feature table. Appendix C Table C.9.
`threshold_sensitivity.py`	Activity gate robustness test: L computed at 8 thresholds (0.001%-10%) for 10 systems + all 256 CA rules.	Section 2.3 robustness claim. Appendix B.4 robustness note.

Earlier Iterations

The scripts in this package are the final versions resulting from iterative development. All intermediate versions have been retired. The development history is documented in the project repository at goleudy.ai.

Reproduction Instructions

Full reproduction of all paper results

```bash

```

```
# 1. Core analysis — all 287 systems, 4 substrates
python3 final_analysis.py
# Outputs: final_ca.csv, final_gol.csv, final_gs.csv, final_sort.csv
# Runtime: ~3 minutes (single CPU)

# 2. Dual memory comparison — M_pred and M_cohen
python3 mld_dual_memory.py
# Outputs: console report with cross-substrate M comparison
# Runtime: ~3 minutes

# 3. Habituation frequency sweep
python3 gs_frequency_sweep.py
# Outputs: console report with 7-frequency tap response profiles
# Runtime: ~2 minutes

# 4. Feature investigation
python3 mld_hierarchy_investigation.py
# Outputs: console report with per-feature L decomposition
# Runtime: ~3 minutes

# 5. Threshold sensitivity
python3 threshold_sensitivity.py
# Outputs: console report with robustness test across 256 CA rules
# Runtime: ~2 minutes
```

Total wall-clock time for complete reproduction: approximately 15 minutes on a modern laptop.

Verifying specific results

Table 4.1 (CA by Wolfram class): Run `final_analysis.py`, check "CA RESULTS BY WOLFRAM CLASS" section.

Table 4.2 (Cross-substrate top 8): Run `mld_dual_memory.py`, check "M∩L∩D OVERLAP" section.

Table 4.3 (Frequency sweep): Run `gs_frequency_sweep.py`, check "SUMMARY TABLE" section.

Table 4.4 (Feature investigation): Run `mld_hierarchy_investigation.py`, check "FEATURE CONTRIBUTION ANALYSIS" for each system.

Section 4.5 (ρ = 0.75): Run `mld_dual_memory.py`, check Spearman correlation line.

Section 2.3 (threshold robustness): Run `threshold_sensitivity.py`, check "Sensitivity" line at end.

Random Seeds

All stochastic components use fixed seeds for reproducibility:

Component	Seed	Used in
CA initial conditions	42	All CA runs
GoL grid	N/A (deterministic from pattern)	All GoL runs
Gray-Scott initial perturbation	42	All GS runs
Sorting initial permutation	123	All sorting runs
Self-sorting visit order	42	Self-sort variants
Habituation tap location	deterministic (centre of highest-V cluster)	All habituation runs

Output Format

CSV files use standard format with header row. Fields:

final_ca.csv: name, category, M, L, D_rate, D_mag, D_centre, character, surprise_early, surprise_late, memory_early, memory_late, alive, wolfram_class

final_gol.csv: Same fields minus wolfram_class.

final_gs.csv: Same fields minus wolfram_class.

final_sort.csv: Same fields minus wolfram_class.

License

Contact

D. Neale — david.n@goleudy.ai — goleudy.ai