mcfeedback — Iteration 13 (engine-fixed): 012b + propagationCycles=3

ENGINE FIX APPLIED Base: experiment-012b (exp-004 + weightDecay=0.0025). One addition: `propagationCycles: 3`. 10 seeds × 1000 episodes × 4 conditions.

Engine fix: what changed in src/engine.mjs

The original implementation ran `computeOutput()` (which increments `cycleCount` / `fireCount` / `fireRate`) and `regulateThreshold()` inside the propagation loop. With `propagationCycles=3` this meant homeostatic threshold regulation fired 3× per episode, shrinking thresholds 3× faster and collapsing all 10 seeds to 45%.

The fix moves stat updates and threshold regulation **outside** the loop so they run exactly once per episode regardless of `propagationCycles`. Only weighted accumulation and the firing comparison (`output ≥ threshold`) repeat per cycle.

```
// BEFORE (broken): computeOutput + regulateThreshold inside loop
for (let _cycle = 0; _cycle < nCycles; _cycle++) {
  // ... accumulate ...
  computeOutput(id, accumulators, neuronState);   // updates cycleCount, fireCount — runs N×
  regulateThreshold(state, config);              // adjusts threshold — runs N×
}

// AFTER (fixed): fire only inside loop, stats + regulation once outside
for (let _cycle = 0; _cycle < nCycles; _cycle++) {
  // ... accumulate ...
  state.output         = input >= state.threshold ? 1 : 0;  // fire
  state.firedThisCycle = state.output === 1;
}
// run once:
state.cycleCount++;
if (state.firedThisCycle) state.fireCount++;
```

```
state.fireRate = state.fireCount / state.cycleCount;
regulateThreshold(state, config);
```

Config

```
learningRate:          0.01    (exp-004 original)
weightDecay:           0.0025  (012b fix — halved from 0.005)
flagStrengthGain:      0.3     (exp-004 original)
flagDecayRate:         0.7     (exp-004 original)
flagStrengthThreshold: 0.5     (exp-004 original)
propagationCycles:     3       ← the new addition
reward:                linear  (no squaring, no annealing)

No direction-consistent flags, no frustration flip, no reward shaping.
```

Results: Before vs After fix

012B (NO CYCLES)

55.0%

10/10 seeds · std 0%

013 BROKEN (TRIPLE HOMEOSTASIS)

45.0%

0/10 seeds · all collapsed

013 FIXED (CORRECT CYCLES)

52.0%

7/10 seeds · std 4.8%

Distribution (frozen-weight accuracy)

Condition	Mean	Std	Min	Max	All seeds
Baseline	53.0%	4.8%	40%	55%	55 55 40 55 55 50 55 55 55 55

Ambient only	54.0%	3.2%	45%	55%	55 45 55 55 55 55 55 55 55 55
Dampening only	55.0%	0.0%	55%	55%	55 55 55 55 55 55 55 55 55 55
Full model	52.0%	4.8%	45%	55%	45 45 55 55 55 55 55 55 55 45

Paired t-tests vs Baseline

Comparison	Mean diff	t	p	Significant?
Ambient only vs Baseline	+1.0%	0.5145	0.457	ns
Dampening only vs Baseline	+2.0%	1.3093	0.1645	ns
Full model vs Baseline	−1.0%	−0.4082	0.5111	ns

Per-pattern mean accuracy (across seeds)

Condition	P1	P2	P3	P4
Baseline	46.0%	58.0%	56.0%	52.0%
Ambient only	42.0%	58.0%	58.0%	58.0%
Dampening only	40.0%	60.0%	60.0%	60.0%
Full model	46.0%	54.0%	54.0%	54.0%

P1=[1,0,1,0,1]→[0,1,0,1,0] · P2=[1,1,0,0,0]→[0,0,1,1,1] · P3=[1,0,0,0,1]→[0,1,1,1,0] · P4=[0,1,0,1,0]→[1,0,1,0,1]

P1 and P4 are exact complements of each other. The network converges to a fixed output that satisfies P2/P3/P4 but not P1, giving (40+60+60+60)/4 = **55%**. This pattern-level ceiling is unchanged across all propagation-cycle conditions.

Mean |weight| — Full model

Seed	Accuracy	Mean |weight|	Note
42	45%	0.6230	—
137	45%	1.1697	high |w|, still stuck
271	55%	0.8696	
314	55%	0.7748	
500	55%	0.8183	
618	55%	0.8058	
777	55%	0.8013	
888	55%	0.7940	
999	55%	0.7256	
1234	45%	0.2871	weight-starved

Group	Seeds	Mean |weight|
Good (≥55%)	7/10	0.7985
Poor (<55%)	3/10	0.6933

Seed 1234 is clearly weight-starved (|w|=0.29). Seeds 42 and 137 have healthy weights but landed in a different attractor basin. Compare to the *broken* version: mean |w|=0.35 across all seeds — the triple homeostasis was actively preventing weight accumulation.

Summary comparison: 012b vs 013 (fixed)

	012b	013 broken	013 fixed

	(no cycles)	(triple homeostasis)	(correct cycles)
Mean accuracy	55.0%	45.0%	**52.0%**
Std	0.0%	0.0%	4.8%
Seeds ≥ 55%	10/10	0/10	**7/10**
Mean |weight|	0.7898	0.3520	0.7669

Verdict

The engine fix is confirmed. Moving stat updates and `regulateThreshold()` outside the propagation loop restores healthy weight dynamics: mean |weight| jumps from 0.35 → 0.77 and 7/10 seeds recover to the 55% attractor.

However, **propagationCycles=3 does not improve over 012b**: it drops from 10/10 → 7/10 seeds at 55%, with 3 seeds stuck in alternative 45% attractors. Seeds 42 and 137 appear to lock into a different basin early in training; seed 1234 remains weight-starved. The 55% ceiling itself is unchanged — the per-pattern analysis shows the same P2/P3/P4 = 60%, P1 = 40% split.

Current best: experiment-012b (weightDecay=0.0025, propagationCycles=1): 10/10 seeds, mean 55.0%, std 0.0%.

Next directions:

- The 55% ceiling is a structural attractor driven by the P1/P4 complement problem. Breaking it requires either more training (more episodes), a different reward signal, or a topology change that lets the network distinguish P1 and P4 context differently.
- Experiment with more episodes (2000–5000) to see if the ceiling is a convergence-speed limit or a structural one.
- Try input/output expansion or hidden cluster to give the network more representational capacity.
- Curriculum learning: train P1/P4 more frequently to force discrimination.

Generated 2026-02-22 · mcfeedback experiment-013 (fixed engine) · 10 seeds × 1000 episodes × 4 conditions