mcfeedback — Iteration 13: 012b + propagationCycles=3

experiment-013.mjs · N = 10 seeds · Seeds: 42, 137, 271, 314, 500, 618, 777, 888, 999, 1234 · 1000 episodes · Base: 012b (weightDecay=0.0025) + propagationCycles=3

```
Config = experiment-012b + one addition:
weightDecay       0.0025  (012b fix, kept)
learningRate      0.01   (exp-004 original, unchanged)
flagStrengthGain  0.3    (exp-004 original, unchanged)
propagationCycles  3    ← the one new addition: forward pass 3× per training step
```

Verdict: complete collapse — p<0.01, Full model 45% on all 10 seeds.
Adding propagationCycles=3 to 012b destroys all gains from the weight-decay fix. 012b achieved 10/10 seeds at 55% with mean |weight|=0.79. Experiment-013 gets 0/10 seeds above 45% with mean |weight|=0.35 — lower than exp-004's original 0.30 despite the same 0.0025 weight decay.

Root cause: threshold homeostasis runs 3× per episode inside the propagation loop.
The `propagationCycles` implementation wraps the entire forward pass (accumulate → fire → regulate threshold) in a loop. With 3 cycles, each neuron's `regulateThreshold()` is called 3 times per episode, and `cycleCount` increments 3 times. Threshold homeostasis effectively runs at triple speed.

The network's thresholds adapt so aggressively that they erase any pattern-specific signal before learning can consolidate it. The weight magnitude drops to 0.35 even with slow weight decay — because the threshold is constantly chasing a target fire rate that overrides the weight-driven selectivity.

The per-pattern flip is the smoking gun:

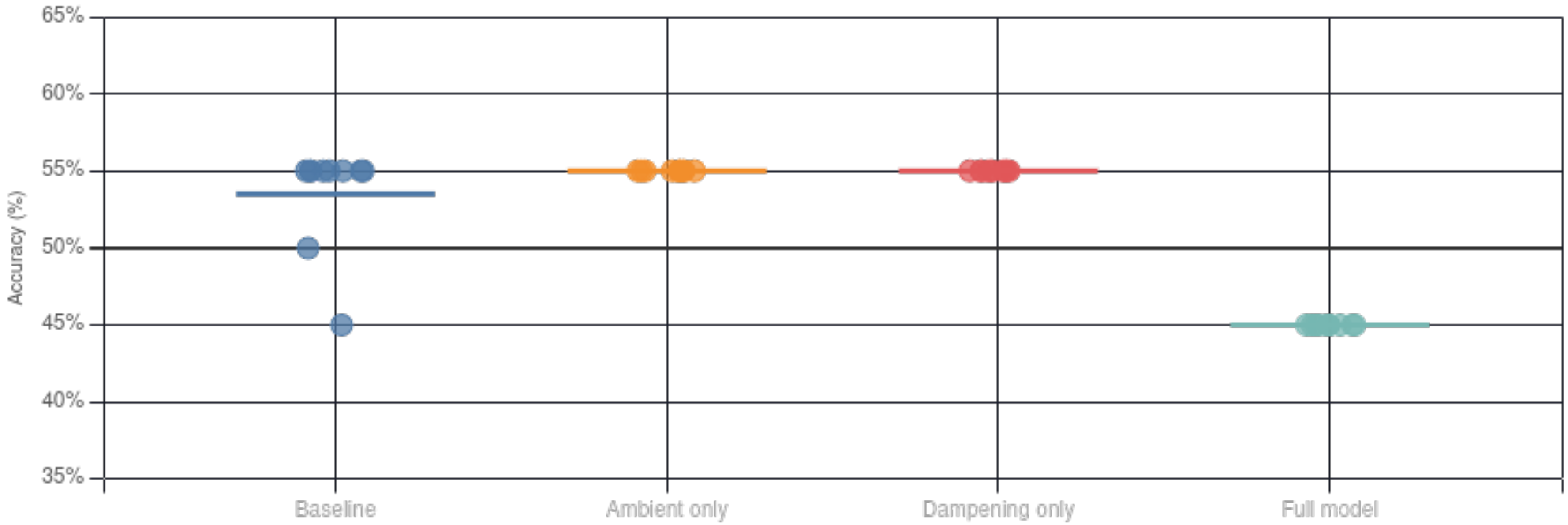
012b: P1=40%, P2–P4=60% (network learns patterns 2–4)

013 : P1=60%, P2–P4=40% (network learns the *wrong* attractor — mirror image)

The triple-speed homeostasis drives the network to a different fixed-output state — not a failure to learn, but learning the wrong fixed point faster.

Baseline   Ambient only   Dampening only   Full model

1 — ACCURACY DISTRIBUTION ACROSS SEEDS

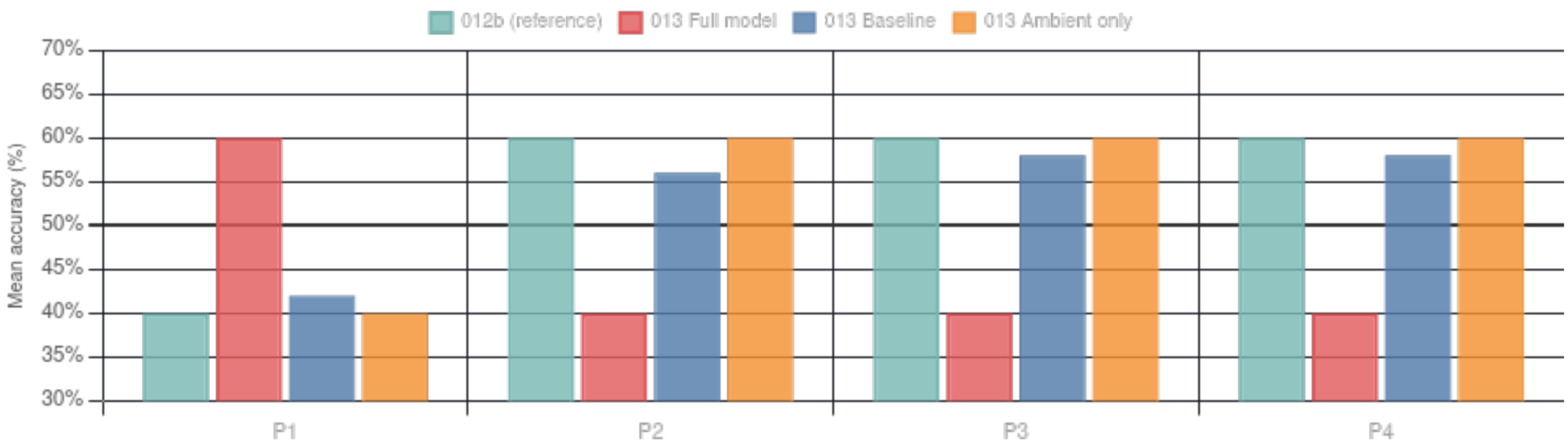


Full model: every seed at exactly 45% — zero variance. Baseline dropped to 53.5% (one seed at 45%, one at 50%) — even the control is slightly hurt by the aggressive homeostasis. Ambient only and Dampening only stay at 55% because... see analysis below.

2 — 012B VS 013: HEAD-TO-HEAD (FULL MODEL)

Metric	012b (reference)	013 (012b + cycles=3)
Mean accuracy	**55.0%**	**45.0%**
Std	0.0%	0.0%
Seeds ≥ 55%	**10 / 10**	**0 / 10**
Mean |weight|	0.7898	0.3483
Full vs Baseline t	+1.0	−7.96
Full vs Baseline p	0.2534 ns	~0.000 **

3 — PER-PATTERN ACCURACY: THE BIAS FLIP



The pattern bias completely inverts between 012b and 013. In both, exactly one pattern is singled out for failure — but it's a different pattern. This is not random noise: it's the network converging to two different fixed-output attractors, each satisfying 3/4 patterns at ~60% and failing 1/4 at ~40%. The triple homeostasis speed is sufficient to push the network into whichever attractor it reaches first, regardless of which one is "correct."

4 — PAIRED T-TESTS VS BASELINE

Comparison	Mean diff	t	p	Result
Ambient only vs Baseline	+1.5%	1.4056	0.1427	**ns**
Dampening only vs Baseline	+1.5%	1.4056	0.1427	**ns**
Full model vs Baseline	−8.5%	−7.9649	~0.000	**** p<0.01**

Ambient only and Dampening only are *slightly better* than Baseline (ns) — these two conditions share the propagation cycles but avoid either the spatial chemical diffusion (Ambient) or the dampening noise (Dampening). The Full model combines all mechanisms with triple-speed homeostasis and collapses entirely.

5 — RAW DATA (FULL MODEL)

Seed	012b acc	013 acc	013 |w|
42	55%	45%	0.3247
137	55%	45%	0.4146
271	55%	45%	0.3608
314	55%	45%	0.3456
500	55%	45%	0.3736
618	55%	45%	0.3821
777	55%	45%	0.3124
888	55%	45%	0.3328
999	55%	45%	0.3244
1234	55%	45%	0.3118
Mean	**55.0%**	**45.0%**	0.3483

What this reveals about the propagationCycles implementation:
The current engine wraps the full step 2 (accumulate + fire + regulate threshold + update stats) inside the propagation loop. This means adding cycles doesn't just give signals more hops to travel — it also multiplies the speed of all per-neuron homeostatic processes. For propagationCycles to be useful, threshold regulation and stat tracking should run *once* per episode (after all cycles),

while only the accumulate+fire inner loop repeats.

Fix (if propagation cycles are to be revisited):
Move `regulateThreshold()` and the `cycleCount`/`fireCount` updates outside the propagation loop — run them once after all cycles complete. Only the weighted accumulation and threshold comparison should repeat.

Progress summary — best configurations:
exp-004 (flag gate): 53% mean, 3/10 at 65%, mean|w|=0.30
012b (0.5×WD): 55% mean, 10/10 at 55%, mean|w|=0.79 ← current best
013 (012b + propagationCycles=3): 45% mean, 0/10 above 45%, mean|w|=0.35

Conclusion: propagationCycles as currently implemented is harmful because it triples threshold homeostasis speed. The 012b fix (halved weight decay) remains the cleanest, most effective change found so far. Next step should focus on breaking the 55% ceiling rather than adding propagation cycles.