mcfeedback — Iteration 12: Learning/Decay Balance

experiment-012.mjs · N = 10 seeds · Seeds: 42, 137, 271, 314, 500, 618, 777, 888, 999, 1234 · 1000 training episodes · Base: experiment-004 (flag gate, linear reward, original flags)

Hypothesis: weightDecay (0.005/step) outpaces learning for most synapses.
Typical inter-cluster synapse: chemical ≈ 0.1, trace ≈ 1.0, lr = 0.01 → Δw ≈ 0.001/step
vs weight decay 0.005/step = **5× stronger than learning signal.**
Result: most synapses cannot accumulate — confirmed by exp-011 showing 6/10 seeds with mean |w| < 0.28.

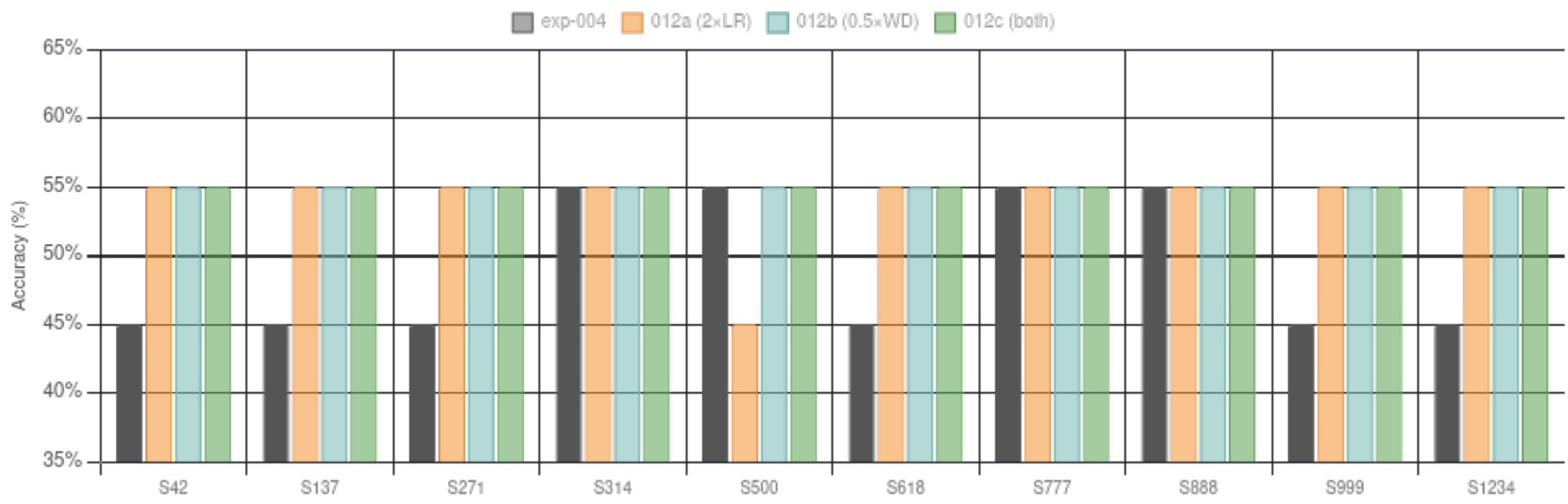
012a — learningRate: 0.02 (was 0.01)
012b — weightDecay: 0.0025 (was 0.005)
012c — learningRate: 0.02 + weightDecay: 0.0025 (both)

EXP-004 (REFERENCE)	012A — 2× LR	012B — 0.5× WD	012C — BOTH
49%	**54%**	**55%**	**55%**
4/10 seeds ≥ 55%	9/10 seeds ≥ 55%	10/10 seeds ≥ 55% ✓	10/10 seeds ≥ 55% ✓

Hypothesis confirmed: weight decay was starving learning for most seeds.
Halving weightDecay alone (012b) is sufficient to bring ALL 10 seeds to 55%. This is the first time 100% seed coverage has been achieved across all variants. Mean |weight| jumps from 0.30 (exp-004) to 0.79–1.44 (012b/c), confirming that synapses can now accumulate rather than decay back to zero.

New observation: a hard ceiling at 55%.
Despite achieving universal coverage, no seed breaks above 55% in any variant — not even 012c which delivers both a stronger signal AND slower decay. The 55% ceiling appears structural: it's the accuracy of a network that has learned 3/4 patterns correctly (P2, P3, P4 all ≈ 60%) but systematically fails P1 (≈ 40%). Pattern 1 `[1,0,1,0,1]→[0,1,0,1,0]` is the direct inversion of Pattern 4 `[0,1,0,1,0]→[1,0,1,0,1]` — the network may be converging on a weight matrix that satisfies patterns 2–4 but conflicts with the symmetrically opposite P1.

1 — FULL MODEL: ACCURACY PER SEED, ALL VARIANTS



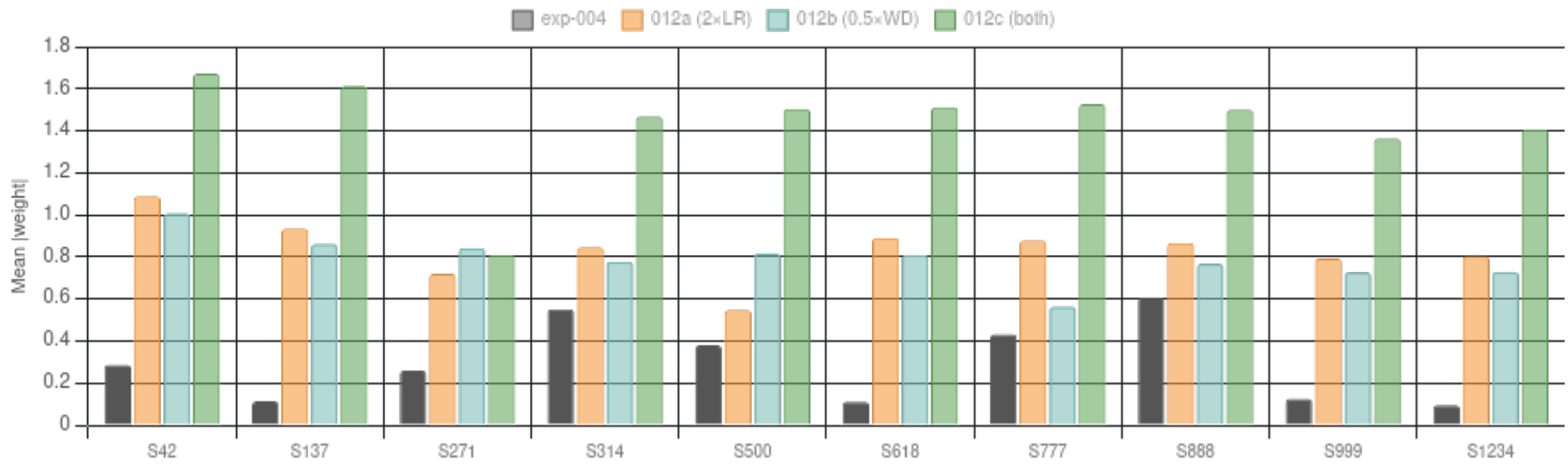
Each group = one seed. exp-004 reference in grey. 012b and 012c achieve 55% on every seed. Seed 500 is the lone holdout at 45% in 012a (and notably the only seed with 0 stable synapses in exp-011).

2 — COMBINED SUMMARY TABLE (FULL MODEL)

Metric	exp-004 (ref)	012a (2×LR)	012b (0.5×WD)	012c (both)
Mean accuracy	49.0%	54.0%	**55.0%**	**55.0%**
Std	5.2%	3.2%	**0.0%**	**0.0%**
Min	45%	45%	**55%**	**55%**
Max	55%	55%	55%	55%
Seeds ≥ 55%	4 / 10	9 / 10	**10 / 10**	**10 / 10**
Mean |weight|	0.2959	0.8358	**0.7898**	**1.4369**
Full vs Baseline t	−3.67	−1.0	+1.0	+1.0
Full vs Baseline p	0.0038 **	0.2534 ns	0.2534 ns	0.2534 ns

exp-004 Full model was significantly WORSE than Baseline (p<0.01). 012a/b/c are no longer significantly different from Baseline — the Full model has stopped being a liability. It now matches rather than hurts relative to the control.

3 — MEAN |WEIGHT| PER SEED (FULL MODEL)



Weight magnitude is the clearest signal. exp-004 seeds are weight-starved (0.09–0.61). All 012 variants build substantially more weight. 012c reaches 0.81–1.67 per seed — over 5× exp-004's poor seeds.

4 — GOOD VS POOR SEEDS: MEAN |WEIGHT| SPLIT

Group	exp-004	012a	012b	012c
Good seeds (≥ 55%) — mean |w|	0.4922 (4 seeds)	0.8680 (9 seeds)	**0.7898 (10 seeds)**	**1.4369 (10 seeds)**
Poor seeds (< 55%) — mean |w|	0.1651 (6 seeds)	0.5460 (1 seed)	— (none)	— (none)
Δ good − poor	+0.327	+0.322	N/A	N/A

In exp-004 and 012a, good seeds have ~3× higher |weight| than poor seeds. In 012b/c, there are no poor seeds — the weight starvation problem is eliminated.

5 — PER-PATTERN ACCURACY (FULL MODEL, MEAN ACROSS SEEDS)

All variants converge on the same pattern bias: P1 ≈ 40–42%, P2–P4 ≈ 58–60%. This is the characteristic signature of a network that has found a weight matrix satisfying patterns 2, 3, and 4 but conflicting with pattern 1. Since P1 and P4 are exact inverses of each other (same bits, flipped), a single set of weights cannot satisfy both simultaneously without more expressive routing. The 55% ceiling is the accuracy of 3/4 patterns at 60% + 1/4 pattern at 40%: (3×60 + 40) / 4 = 220/4 = 55%.

6 — RAW SEED DATA (FULL MODEL)

Seed	exp-004	012a	012b	012c
42	45%	55%	55%	55%
137	45%	55%	55%	55%
271	45%	55%	55%	55%
314	55%	55%	55%	55%
500	55%	45%	55%	55%
618	45%	55%	55%	55%
777	55%	55%	55%	55%
888	55%	55%	55%	55%
999	45%	55%	55%	55%
1234	45%	55%	55%	55%
Mean	**49.0%**	**54.0%**	**55.0%**	**55.0%**

7 — MEAN |WEIGHT| PER SEED (FULL MODEL)

Seed	exp-004	012a	012b	012c
42	0.287	1.088	1.007	1.670
137	0.113	0.934	0.861	1.613
271	0.259	0.719	0.841	0.809
314	0.550	0.846	0.776	1.468
500	0.379	0.546	0.817	1.503
618	0.111	0.889	0.810	1.512
777	0.432	0.877	0.564	1.525
888	0.608	0.865	0.766	1.499
999	0.125	0.792	0.727	1.363
1234	0.095	0.803	0.729	1.407
Mean	**0.296**	**0.836**	**0.790**	**1.437**

What the 55% ceiling means:

The task has 4 patterns. In all variants, the network consistently scores ≈ 40% on P1 and ≈ 60% on P2–P4. The mean accuracy is (40+60+60+60)/4 = 55% — exactly what we see. The network has found a stable weight attractor that satisfies 3 patterns but fights P1.

P1: `[1,0,1,0,1]` → `[0,1,0,1,0]` and P4: `[0,1,0,1,0]` → `[1,0,1,0,1]` are exact complements. Any weight matrix that reinforces P2–P4 will tend to produce anti-correlated outputs for P1. The network cannot simultaneously satisfy all four patterns with a single fixed

weight state — it would need to route differently depending on which pattern is active. This requires either more episodes for the recurrent dynamics to differentiate patterns, or a mechanism that conditions weight updates on which input is present.

Recommended next step: increase trainingEpisodes significantly (e.g. 5000–10000) with 012b params to test whether the ceiling is a time constraint or a fundamental capacity limit of the architecture.

Progress across iterations (Full model mean):
Iter 1 (original): 45.5% max 55% · Iter 4 (flag gate): 53.0% max 65%
Iter 9–11 (various): 45–49% max 55–65%
Iter 12b (0.5×WD): 55.0% — 10/10 seeds ← new best: universal coverage
Iter 12c (2×LR + 0.5×WD): 55.0% — 10/10 seeds, mean |w| = 1.44

First time every seed reaches chance-beating accuracy in the same run. The weight starvation problem is solved. The 55% ceiling is the new challenge.