What Memory Cannot Supply: Budgeted Repair and Convention Inheritance in Small Systems

Flyxion
Independent Researcher

November 2026

Abstract

A set of small experiments in plain Python examines how bounded search retention, bounded lexicon size and bounded output affect two kinds of self-correcting systems. In the first, a beam-search repairer restores damaged arithmetic expressions, dependency graphs and boolean circuits. In the second, populations of weighted-lexicon agents settle on shared names through a repair game and must carry those names across population turnover. The repair experiments show that, in the conditions tested, search retention matters when damage is large relative to the beam and that enlarging the beam beyond a modest size buys search effort but not further correctness. In the circuit domain, finding a repair that passes the visible tests is a different outcome from recovering the full behaviour, and the tested increases in beam size did not reliably close the gap. The convention experiments yield a correction to their own first result. A design in which each inheritance rule shapes its own donor population shows a large advantage for inheriting a donor's strongest lexicon entries. A matched-state design, which applies every rule to identical donors, shows that most of that advantage came from the rules' cumulative effects on the donor population, that how widely a rule spreads meanings across newcomers matters as much as how agreeable its items are, and that recent corrections deliver fewer distinct meanings and more conflicting associations than the alternatives. What survives is a modest advantage of strong entries over random sampling when ties are broken randomly, and a clear disadvantage for corrections at larger budgets. Search capacity, observational discrimination and the selection of transmitted content behave as distinct factors in these systems, and the explanations offered for each remain hypotheses where stated.

1 Introduction

Systems that repair themselves are usually described by what they can do with unlimited resources, and the interesting questions often appear only when a resource is fixed. A repairer that may hold every candidate it has ever generated is a different object from one that must decide at each step what to keep. A community that may consult an unbounded record of its past exchanges is a different object from one whose members carry a few dozen entries each. The experiments reported here treat the constraint as part of the design. Each retains only a bounded set of search candidates or lexicon entries, each caps what a single run may write to disk, and each is small enough that the whole project, including every result discussed below, fits in a few tens of kilobytes and runs in seconds without a model download or a GPU.

The project began as a search for experiments with the satisfaction of a trained model and the verifiability of a unit test. Repair of a known object has the second property, since the original is available to the experimenter and success can be checked against it. Convention formation among simple agents has the first, since a shared vocabulary appears from nothing, fails under some conditions and survives under others. The two halves are connected by a common question about what a limited memory can and cannot carry, whether the memory is a beam of candidate repairs or a lexicon passed from one generation of agents to the next.

The essay first states what the budgets do and do not guarantee, then reports the repair experiments, then the convention experiments, including a revision of an early inheritance result, and then states what the combined results support and what they leave open. Throughout, success is reported with a stated denominator and a stated notion of correctness, because the central findings turn on distinctions that a single success rate conceals.

2 Budgets and What They Guarantee

Three budgets appear. In the repair scripts the beam width bounds the number of candidates retained between search rounds. This is best described as a search-retention budget and not as total memory. Each round generates every one-edit successor of every retained candidate before scoring and pruning them, so the transient population is larger than the beam by a factor equal to the size of the edit neighbourhood, which is 56 toggles for the eight-node graphs and a few dozen edits for the circuits. A visited-state set also grows until it reaches fifty times the beam width and is then cleared down to the current beam, which bounds it in proportion to the beam but does not remove it. Total working state therefore scales with the beam width without being equal to it, and no memory measurement was taken.

In the convention scripts each agent holds a weighted lexicon mapping meanings to symbols, and entries beyond a cap of twenty-four are forgotten, weakest first. A further optional budget bounds a log of an agent's most recent corrections. The symbol inventory is a community-level budget: when it is smaller than the number of meanings, compression forces ambiguity. These bounds are hard caps enforced by the data structures themselves.

The third budget is output. A capped writer refuses any line that would take a run's detail or transcript file past a byte ceiling and counts the lines it refuses. The ceiling limits a single run. Summary files are small and uncapped, and repeated runs into new folders accumulate without limit, so the guarantee is per run and is not a bound on the project. In every run reported below the ceiling was not reached and no lines were dropped, which means the cap was set but not exercised, and a smaller ceiling would be needed to observe the writer degrading.

3 Search Retention and Repair Outcomes

3.1 Design

Three domains share one structure. A known object is damaged by k random point edits, and a beam-search repairer, holding at most b candidates between rounds, applies single edits of cost 1 and stops when its objective reaches zero or after $k + 2$ edits. In the first domain the objects are arithmetic expressions over one variable, edited by changing an operator or a leaf,

and the objective is the summed absolute difference from the original on a fixed range of eight inputs. In the second the objects are dependency graphs on eight nodes, edited by toggling an edge, and the objective counts mismatches in reachability on a random sixty percent of ordered node pairs plus a penalty for any cycle. In the third the objects are boolean circuits with four inputs and six gates drawn from four gate types, edited by changing a gate type or rewiring one input, and the repairer sees ten of the sixteen truth-table rows.

Success in these experiments is behavioural. The scorer checks agreement with the original on whatever checks it is given, so a repaired object may differ from the original in structure and still count as repaired. For the circuits the final check covers all sixteen rows, which is equivalence of the computed function and not of the wiring. The experiments therefore establish recovery of behaviour on the stated checks and cannot establish recovery of the original object. Because k random edits can cancel one another or leave behaviour unchanged, the damaged object need not be at edit distance k from the original. The statistic reported as the share repaired within k edits compares the repair cost with the number of edits applied and is not a measure of optimal repair cost.

3.2 Beam Size and Repair Cost

Table 1 gives results for expressions from 200 trials per cell, with success defined as the repairer reaching zero error on the test inputs and the second column of each pair counting repairs that used no more edits than were applied. With a single corruption every beam solved every trial. At two corruptions a beam of 2 solved 94 percent and a beam of 32 solved all of them, and at three corruptions the figures were 90.5 and 100 percent. The share repaired within the applied number of edits is more sensitive, and for three corruptions it ran from 89 percent at a beam of 2 to 100 percent at a beam of 128. Small beams sometimes found a repair that cost more edits than were applied. The mean number of states examined rose steeply with the beam, from about 84 at a beam of 2 to about 694 at a beam of 128 for three corruptions, and most of that increase bought no further success once the beam reached about 32.

Table 1: Expression repair, 200 trials per cell. Success is zero error on the test inputs; the second figure in each pair is the share repaired within the applied number of edits k.

	$k = 2$		$k = 3$	
Beam	Success	Within k	Success	Within k
2	0.940	0.905	0.905	0.890
8	0.995	0.980	0.965	0.930
32	1.000	1.000	1.000	0.985
128	1.000	1.000	1.000	1.000

The graph domain showed the same shape with a steeper cost, because each state has 56 successors. From 100 trials per cell, six toggles were solved in 92 percent of trials with a beam of 2, 95 percent with a beam of 8, and every trial with beams of 32 and 128, while the mean states examined climbed from about 460 to about 12,600. A beam of 32 already reached complete success for roughly a third of the effort of a beam of 128. These patterns describe the conditions tested, which are a few damage levels and four beam sizes in toy domains, and they do not

establish a general threshold relation between beam and damage. Different beam conditions also use different damaged instances, since the instance seed depends on both k and the beam, so comparisons across beams are unpaired and no uncertainty is reported for them.

3.3 Passing the Visible Tests and Recovering the Behaviour

The circuit experiment separates two outcomes that the other domains conflate. Because the repairer sees only ten of sixteen rows, a circuit can satisfy every visible row and still differ from the original on a hidden one. Table 2 reports three quantities with explicit denominators. Visible success is the share of all trials in which the repairer found a circuit passing the ten visible rows. Correctness over all trials is the share of all trials in which the repaired circuit matched the original on all sixteen rows. Correctness given visible success divides the same count only by the trials that passed the visible rows. The last two coincide wherever visible success is 1.0 and differ otherwise, as at three corruptions with a beam of 8, where visible success is 0.98, correctness over all trials is 0.79 and correctness among visible passes is 0.81. They answer different questions, the first about the whole population of damaged circuits and the second about the quality of repairs the repairer considers finished.

Table 2: Circuit repair, 100 trials per cell, ten visible rows of sixteen. Correctness is agreement with the original on all sixteen rows.

k	Beam	Visible success	Correct, all trials	Correct, given visible
1	2	1.00	0.96	0.96
1	8	1.00	0.95	0.95
1	32	1.00	0.94	0.94
1	128	1.00	0.96	0.96
2	2	0.99	0.86	0.87
2	8	1.00	0.94	0.94
2	32	1.00	0.92	0.92
2	128	1.00	0.91	0.91
3	2	0.95	0.81	0.85
3	8	0.98	0.79	0.81
3	32	1.00	0.88	0.88
3	128	1.00	0.89	0.89

Visible success lay between 0.95 and 1.0 in every cell, while correctness lay between 0.79 and 0.96 under either denominator. Enlarging the beam raised visible success to 1.0 and then did not reliably raise correctness: at three corruptions, moving from a beam of 32 to a beam of 128 changed it from 0.88 to 0.89. The tested beam increases therefore did not eliminate the gap between visible success and full-table correctness. The inference to be drawn from this is limited. The repairer ranks candidates only by visible error and breaks ties by the order in which successors were generated, which uses no preference for fewer edits and no knowledge of how damage was produced. A different ranking rule, a preference for small edits, or an informative prior over corruptions could use information that this repairer ignores and could raise average hidden-row accuracy. Whether any procedure could guarantee recovery would require an identifiability argument showing that distinct targets remain compatible with

everything the repairer is entitled to use. The experiment shows a plateau for one ranking rule, and the remaining shortfall is consistent with ambiguity in the visible rows, but it does not demonstrate that the ambiguity is irreducible. The pattern parallels the literature on automated program repair, where patches that pass the available tests are frequently incorrect and overfit the tests they were validated against [4, 5, 7], and the circuit setting reproduces the effect in a case small enough to check against the complete truth table.

4 Convention Formation and Its Transmission

4.1 The Naming Game and the Compression Budget

The second half of the project uses a minimal version of the naming game familiar from work on the self-organisation of vocabularies [8, 2]. Ten agents hold weighted lexicons over twelve meanings. In each interaction a speaker names a randomly chosen meaning, inventing a symbol from a shared pool if it has none, and a hearer guesses the meaning from its own lexicon, guessing at random when the symbol is unknown. Success reinforces the used association and weakly inhibits competing symbols for the same meaning. Failure is repaired by revealing the intended meaning to the hearer, who adds the association, while the speaker weakens the symbol it used. An optional bounded log records the hearer's recent failures and is consulted as a bonus when interpreting a symbol. Every 400 interactions a fraction of the population is replaced by agents with empty lexicons.

The size of the symbol inventory dominates the outcome. With 6 symbols, success in the final stretch of each epoch was about 46 percent and most meanings shared a symbol with another meaning. With 12 symbols, which equals the number of meanings, success was about 75 percent and the ambiguity measure, the share of meanings whose consensus symbol is shared with another meaning, was about 0.4. With 24 symbols success was about 91 percent and ambiguity about 0.1, and the residual ambiguity at the largest inventory reflects early accidental collisions that become locked in. Compression therefore creates ambiguity in the expected way. A correction log gave small changes that were inconsistent in direction. At 12 symbols it moved success from 0.745 to 0.770 and ambiguity from 0.42 to 0.375, while at 6 symbols no improvement was observed (success 0.463 without a log and 0.448 with one), as Table 3 shows from 20 runs per cell. A log of 32 entries did no better than a log of 8. With these sample sizes, differences of a point or two should be treated as suggestive.

Table 3: Convention formation with 20 percent turnover, 20 runs per cell. Success is measured over the last 100 interactions of each epoch after warm-up; ambiguity is the share of meanings whose consensus symbol is shared.

	No log		Log of 32	
Symbols	Success	Ambiguity	Success	Ambiguity
6	0.463	0.858	0.448	0.848
12	0.745	0.421	0.770	0.375
24	0.908	0.101	0.917	0.087

4.2 Definitions and Turnover

Consensus is computed as follows. Each agent holding any entry for a meaning votes for its highest-weight symbol, ties inside an agent going to the alphabetically smallest symbol, and the most common vote wins, ties across agents going to the symbol voted first in agent order. A meaning with no entries in any agent has no consensus. Retention compares a later consensus with an earlier one and is the number of meanings whose earlier consensus exists and equals the later consensus, divided by all twelve meanings. A meaning with no inherited entry therefore has no consensus and counts as not retained, so a missing convention is never counted as a retained one, and immediate retention for newcomers who inherit nothing is exactly zero. If an unrelated convention were formed independently with 12 equally likely symbols, chance agreement would be about $1/12 \approx 0.08$, and the observed retention of 0.09 after 400 interactions with no inheritance at full replacement is consistent with that value, though the selection procedure does not strictly satisfy the independence assumption behind it.

Replacing twenty percent of the population per epoch left the consensus vocabulary nearly intact, with retention of about 99 percent, so that setting says little about survival. Raising the replacement fraction changed this sharply. With no inheritance at all, about 84 percent of the consensus vocabulary survived an epoch at 50 percent replacement, about 26 percent at 80 percent, and about 9 percent at full replacement, where the convention can survive only by being rebuilt from nothing. Success immediately after a replacement was between 12 and 27 percent and recovered during the epoch to between 30 and 56 percent. Two inheritance mechanisms were then tried. Copying a veteran's correction log to each newcomer did not help and was slightly worse than no history at 50 percent replacement, with retention of 0.78 against 0.84 and end-of-epoch success of 0.49 against 0.56. Seeding each newcomer with the K strongest lexicon entries of a random veteran at half weight raised retention at full replacement from 0.09 at $K = 0$ to 0.39 at $K = 3$, 0.64 at $K = 6$ and 0.99 at $K = 12$. The bounded channel plays the role of the transmission bottleneck in iterated learning, where what survives a generation is what can be learned from the limited sample the previous generation provides [6, 3]. These two scripts differ in contents, selection rule, information per item and insertion weight, so they compare two particular mechanisms and do not isolate any single factor.

4.3 A Cumulative Comparison of What Is Inherited

A comparison that equalises representation holds the insertion weight at 0.25 and gives every newcomer up to K meaning–symbol pairs from one randomly chosen donor, under full replacement, with the regimes differing in the selection of pairs: the donor's K strongest entries, its K most recent distinct logged corrections, or K entries drawn at random from its lexicon. Every agent keeps the same correction-log capacity in every regime. This design applies a replacement every cycle, and each regime then continues with the population that its own inheritance rule produced. The regimes share a warm-up community within a run, so the first replacement acts on identical donors, but from the second cycle onward their donor populations differ. The estimand is therefore the cumulative effect of a policy over ten cycles, and not the immediate effect of selecting content from a given donor. Table 4 gives means over 30 independent runs, each averaging its ten cycles, with standard errors across runs of at most 0.014.

Read at face value the table shows a large advantage for strong entries, and an explanation

Table 4: Cumulative design, full replacement, 30 independent runs. Each rule shapes its own donor population. Agreement is the share of inherited pairs matching the consensus before replacement. Retention is measured immediately, after 100 and after 400 interactions. The no-inheritance baseline has retention 0.09 after 400 interactions and last-100 success 0.33.

K	Content	Agreement	Retention, 0	Retention, 100	Retention, 400	Success
3	Strong	0.86	0.58	0.38	0.34	0.39
3	Corrections	0.49	0.41	0.23	0.20	0.35
3	Random	0.40	0.41	0.19	0.16	0.34
6	Strong	0.92	0.80	0.65	0.63	0.53
6	Corrections	0.51	0.53	0.38	0.35	0.39
6	Random	0.41	0.51	0.33	0.27	0.36
12	Strong	0.96	0.98	0.98	0.98	0.81
12	Corrections	0.55	0.67	0.62	0.61	0.46
12	Random	0.47	0.65	0.59	0.56	0.48

in terms of agreement is tempting, since about 86 to 96 percent of strong entries matched the earlier consensus against roughly half of logged corrections and 40 to 47 percent of random entries. The table does not support that reading on its own. The agreement figures describe donors that differ between regimes, because a community that has been fed random or recent-correction slices for several cycles is itself less coherent. The design also did not measure how many distinct meanings each channel delivered, nor how many conflicting symbols a newcomer received. Both gaps are addressed by the matched design that follows.

4.4 A Matched-State Comparison

In the matched design a single reference community is evolved without any replacement, and every 400 interactions its complete state is taken as a snapshot. Every rule and every budget is then applied to that same snapshot with the same donor assignment, so that newcomer i copies from the same donor under every rule. Each branch then runs 400 interactions with the same random seed, and the donor community is unaffected by any branch. The reference community is stable (retention of its own consensus over 400 interactions without replacement is 0.995) and every meaning has a consensus at every snapshot. In addition to the three rules, a fourth was added after the first matched run exposed a property of the strong rule. Strong entries are chosen by sorting on weight and then, among equal weights, on meaning index, and many weights tie at their cap, so every newcomer received the same high-index meanings. The added rule, strong entries with ties broken at random, differs from the first only in that respect. The design reports the pairs actually delivered, the distinct meanings covered per newcomer, the share of the twelve meanings carried by at least one newcomer, the number of meanings for which a newcomer received conflicting symbols, and the agreement of delivered pairs with the earlier consensus. Table 5 gives means over 30 independent runs, each averaging ten snapshots. The independent unit is the run, and the cycles within a run are not treated as replicates.

The matched comparison changes the earlier picture in four ways. First, the gaps are much smaller than in the cumulative design. At $K = 6$ the success advantage of strong entries over

Table 5: Matched-state design, full replacement, 30 independent runs. Delivered is pairs inserted per newcomer. Coverage is the share of the twelve meanings carried by at least one newcomer. Retention is measured immediately and after 400 interactions. The no-inheritance baseline has retention 0.09 after 400 interactions and last-100 success 0.34.

K	Rule	Delivered	Coverage	Agreement	Retention, 0	Retention, 400	Success
3	Strong, index ties	3.00	0.40	0.995	0.40	0.35	0.44
3	Strong, random ties	3.00	0.88	0.994	0.87	0.41	0.40
3	Corrections	2.99	0.46	0.938	0.43	0.34	0.38
3	Random	3.00	0.93	0.881	0.86	0.35	0.37
6	Strong, index ties	6.00	0.76	0.992	0.75	0.61	0.55
6	Strong, random ties	6.00	0.97	0.993	0.96	0.72	0.57
6	Corrections	5.55	0.59	0.863	0.50	0.47	0.40
6	Random	6.00	1.00	0.885	0.94	0.68	0.49
12	Strong, index ties	12.00	1.00	0.977	0.99	0.99	0.79
12	Strong, random ties	12.00	1.00	0.977	0.99	0.99	0.78
12	Corrections	8.01	0.70	0.783	0.56	0.53	0.43
12	Random	12.00	1.00	0.882	0.95	0.94	0.74

random entries, 0.17 in Table 4, becomes 0.06 for the index-tie rule and 0.08 for random ties in Table 5, and agreement among random entries is 0.88 when donors are identical against 0.41 when each regime's donors were produced by that regime. A large part of the earlier advantage was therefore an effect of inheritance policy on the donor population over successive cycles. Second, spreading meanings across newcomers matters. The index-tie rule left only 40 percent of meanings carried by any newcomer at $K = 3$ and 76 percent at $K = 6$, so its immediate retention was far below that of random sampling (the paired difference at $K = 3$ was -0.465 with standard error 0.022, and at $K = 6$ it was -0.189 with standard error 0.019) even though nearly every one of its entries agreed with the earlier consensus. Agreement is a property of items, whereas coverage is a property of the whole newcomer population, and the second can dominate the first. Third, once ties are broken at random, strong entries retain a modest advantage over random sampling after 400 interactions: the paired differences in retention were $+0.057$ (0.012), $+0.045$ (0.009) and $+0.048$ (0.006) at $K = 3$, 6 and 12, and in end-of-epoch success they were $+0.022$ (0.006), $+0.076$ (0.006) and $+0.042$ (0.003), with standard errors in parentheses. Fourth, logged corrections were worse than random sampling at $K = 6$ and $K = 12$, with paired retention differences after 400 interactions of -0.207 (0.024) and -0.414 (0.018) and success differences of -0.089 (0.010) and -0.317 (0.016). At $K = 12$ the channel delivered only 8.0 pairs on average, covering 6.6 distinct meanings per newcomer, and about one meaning per newcomer received conflicting symbols, whereas random sampling delivered 12 pairs covering 10.9 meanings. A log holds repeated failures about the same few symbols, so its capacity to carry distinct information is lower than its nominal length. At $K = 3$ corrections and random sampling were indistinguishable after 400 interactions (a paired retention difference of -0.011 with standard error 0.011), although random sampling had far higher immediate retention because its coverage was twice as large.

The immediate and later figures also distinguish inherited persistence from relearning. For small budgets the inherited convention eroded after replacement, as with random-tie strong entries at $K = 3$, whose retention fell from 0.87 immediately to 0.46 after 100 interactions and 0.41 after 400. For $K = 12$ the strong rules were essentially lossless, with retention of 0.99

throughout. A retention figure measured only after a full epoch can reflect a community that lost its convention and rebuilt part of it, whereas the immediate figure reflects what the channel carried.

5 What the Results Support

The two halves agree on a general point, which is that the amount a system retains and the information that retention carries are separate quantities, and that each can be varied and measured separately in systems this small. The central claim that the manuscript supports is that search capacity, observational discrimination and the selection of transmitted content are distinct experimental factors, and the experiments allow their effects to be examined separately. In the repair experiments a larger beam raised visible success to its ceiling without reliably raising correctness, so beam size and the discriminating power of the tests acted as separate factors. In the convention experiments the rule that selected transmitted content mattered, and so did how that rule distributed content across newcomers and how it shaped the donors of later generations.

Several explanations were offered along the way and should be held as hypotheses unless stated otherwise. The proposal that agreement among items explains the success of strong-entry inheritance was not supported as an account of the whole effect, because coverage proved at least as important and because the large cumulative gaps were mostly cumulative effects on donors. The proposal that a correction log concentrates on contested symbols remains plausible and is consistent with the low delivered counts, repeated entries and conflicts that were measured, but concentration itself was not measured. What the matched design does establish, for these agents, this update rule and these donors, is that recent corrections carry less distinct and more conflicting information than a random slice of the donor's lexicon at larger budgets, and that strong entries with random ties modestly outperform random sampling. Transmission from noisy donors, or from a donor holding a minority convention, remains untested, and strong-entry seeding could do badly there because it would transmit a minority convention with high confidence. The statement that a convention can be retained across complete turnover only through strong items is not supported: corrections and random entries at $K = 12$ retained 0.53 and 0.94 of the vocabulary after 400 interactions in the matched design against 0.09 with no inheritance, so every channel carried a substantial amount of convention, and the channels differ in how much.

The repair results carry parallel limits. The three domains are toy-sized, the scorer defines correctness behaviourally, and the circuit plateau depends on the fraction of rows shown and on one ranking rule. The expression and graph experiments use a greedy ordering by error, which is a strong signal in these domains and would be weaker where intermediate states give misleading scores. Trial counts range from 20 to 200 per cell, beam conditions are unpaired, and uncertainty is reported only for the two inheritance comparisons, so differences of a point or two elsewhere are suggestive and not established.

6 Reproducibility

The scripts, saved results and this manuscript are in the public repository at `https://github.`

com/standardgalactic/budgeted-repair. The results reported here were produced by the scripts as added in commit f2c8581, and the scripts have not changed since. The commands, all with seed 1, were repair.py with 200 trials, repair_graph.py and repair_circuit.py with 100, conventions.py, turnover.py and seeded_turnover.py with 20, and inheritance_comparison.py and matched_inheritance.py with 30. The seed schedules are as follows. In the three repair scripts the generator for a cell is seeded with 1000 times the seed plus 17 times k plus the beam width, so cells are unpaired across beams. In conventions.py a run is seeded with 100000 times the seed plus 1000 times the symbol count plus 10 times the log size plus the trial index. In turnover.py and seeded_turnover.py it is 100000 times the seed plus 100 times the turnover percentage plus the trial index, which is shared across regimes at one turnover level. In inheritance_comparison.py and matched_inheritance.py it is 100000 times the seed plus 7 times the trial index, and in the matched design branch seeds are derived from the run seed and the snapshot index so that every rule at a snapshot uses the same interaction stream seed and donor assignment. The repairers sort candidates by error using a stable sort, so ties keep the order in which successors were generated. Tables report means over trials or over independent runs, in the inheritance comparisons after averaging the ten cycles within each run, and standard errors are computed across runs. The paired differences in the matched design are computed within run and snapshot before averaging. The scoring, tie-breaking and retention definitions are those stated in the text and in the headers of the scripts.

7 Further Work

The most direct extension of the circuit result is to let the repairer choose which rows to inspect, subject to a query budget, so that spending on tests competes with spending on search retention, and to add a ranking rule that prefers small edits so that the contribution of the ranking rule can be separated from that of the tests. The question of how many informative queries are needed to pin down a concept has a long history in learning theory [1], and the present setting would measure it empirically in a case where success is exactly checkable. For the convention experiments the natural next steps are donors of mixed quality, newcomers who sample from several donors, a rule that selects strong entries while enforcing coverage across newcomers, and a log that records resolved disagreements together with the symbol that ended them. A smaller output ceiling would also exercise the writer, which has so far been set but not reached.

8 Conclusion

Two ordinary-looking constraints produced distinctions that a single success rate would have hidden, and one early result did not survive a better-controlled version of itself. A repairer can succeed on the evidence it was given and still be wrong about the object, and in the tested conditions larger beams did not reliably repair that. A community can carry a convention across complete turnover through any of several channels, but how much survives depends on which items are carried, how they are spread across newcomers, and what the channel has already done to its donors. In both halves the limiting resource was not how many items could be held

but what the held items said and how they were distributed, and in both the experiments were small enough to inspect completely, which is what made the correction of the early inheritance result possible.

References

[1] Angluin, D. (1988). Queries and Concept Learning. *Machine Learning*, 2(4), 319–342.

[2] Baronchelli, A., Felici, M., Loreto, V., Caglioti, E., and Steels, L. (2006). Sharp Transition Towards Shared Vocabularies in Multi-Agent Systems. *Journal of Statistical Mechanics: Theory and Experiment*, P06014.

[3] Kirby, S., Cornish, H., and Smith, K. (2008). Cumulative Cultural Evolution in the Laboratory: An Experimental Approach to the Origins of Structure in Human Language. *Proceedings of the National Academy of Sciences*, 105(31), 10681–10686.

[4] Le Goues, C., Nguyen, T., Forrest, S., and Weimer, W. (2012). GenProg: A Generic Method for Automatic Software Repair. *IEEE Transactions on Software Engineering*, 38(1), 54–72.

[5] Qi, Z., Long, F., Achour, S., and Rinard, M. (2015). An Analysis of Patch Plausibility and Correctness for Generate-and-Validate Patch Generation Systems. *Proceedings of the International Symposium on Software Testing and Analysis.*

[6] Smith, K., Kirby, S., and Brighton, H. (2003). Iterated Learning: A Framework for the Emergence of Language. *Artificial Life*, 9(4), 371–386.

[7] Smith, E. K., Barr, E. T., Le Goues, C., and Brun, Y. (2015). Is the Cure Worse Than the Disease? Overfitting in Automated Program Repair. *Proceedings of the Joint Meeting on Foundations of Software Engineering.*

[8] Steels, L. (1995). A Self-Organizing Spatial Vocabulary. *Artificial Life*, 2(3), 319–332.