

Why a Self-Improving System Cannot Trust Its Own Score

Michael P. Laing

mllaing@post.harvard.edu · ORCID 0009-0004-5593-5759

Written with Anthropic Claude Fable 5.1

September 2026; revised September 28, 2026

Expository companion to “The Fixed Point of Self-Improvement” (September 2026), which rests on “What a Memory Must Preserve” and “Measure-Relational Algebra: The Core” (September 2026; the DOI is in the references). Written on the occasion of Cade Metz’s report on recursive self-improvement in The New York Times, September 16, 2026. Canonical reference: realql.org.

The premise both sides share

In 1965 I. J. Good wrote that the first ultraintelligent machine would be the last invention humanity need ever make, “provided that the machine is docile enough to tell us how to keep it under control” [9]. Sixty years on, the proviso is still doing the work. The companies now pursuing recursive self-improvement and the researchers who fear it [1, 10] disagree about nearly everything except one premise, which neither side states because neither side doubts it: that a system improving itself can tell whether it is improving. The optimists expect the loop to compound gains and the pessimists expect it to compound power, and both expect the loop’s own score to report which is happening.

This note is about that premise. It argues, from an elementary theorem and one recent geophysical result, that a self-improving system built on today’s memory conventions cannot tell whether it is improving, and that the reason is not a limit on its intelligence but a property of how its scores are written down. The argument has five steps. Each is short, and each rests on a result stated and proved in the technical papers listed at the end.

1. A score is a measurement

The accounts now current [1] describe three mechanisms: reinforcement learning against a scored environment, evolutionary search over scored candidate designs, and an agent trained on the records its predecessor wrote. The report explains the first two at length and describes the third in its opening; all three are in use. They differ in what they optimize and in how they generate candidates, but they share a step. At some point each generation is judged by a number, and the next generation is built toward a higher one. The number may be a reward, a

benchmark score, a judge model’s verdict, a loss against the previous generation’s outputs, or a fitness value; the name does not matter.

What matters is that the number is produced by an instrument, and every instrument has error. A benchmark is a finite sample of tasks, so its score is a sample estimate of a capability the sample does not exhaust. A judge model’s verdict is one noisy reading of a quality it does not observe directly. A reward from a simulator is a reading of an outcome under a model of the world that is not the world. In every case there is a quantity the loop cares about, call it X , and a reading of it, call it W , and W is not X . The difference is not a nuisance to be averaged away. It has a structure, the instrument’s error model, and the instrument’s builders usually know it. That knowledge is where the trouble begins, because of what happens to it next.

2. Writing the reading as a number loses something specific

The reading goes into the memory store — the record from which the next generation will be trained, selected, or judged — as a value. The error model does not go with it. This is the universal convention: a cell holds a number, and whatever was known about the number’s uncertainty is at best a note in a separate column that nothing downstream reads. The core paper [2] asks a precise question about the convention: when is it harmless? When does computing from the number give the same answer as computing from the full reading and reducing to a number at the end?

The answer is a theorem, and it is sharp. Reducing to a number before a computation agrees with reducing after it, for every possible input law, exactly when the computation is affine — a weighted sum, and nothing else. For any curved use of the number the two disagree, and the disagreement is not noise. It is a bias: of definite sign, set by the direction of the curvature; of a size set by the curvature and by the dispersion that was discarded; and immune to more data, because more readings from the same instrument do not restore what each reading lost at the moment it was written. The proof is Jensen’s inequality, which is more than a century old. What is new is the observation that the convention puts the bias everywhere, since almost nothing a self-improving loop does with a score is affine. It thresholds, it ranks, it selects the maximum, it passes the score through a nonlinear reward, it compares this generation’s number to the last one’s.

3. A learner trained on the record cannot recover the truth

Suppose the loop is not merely selecting on the score but learning from it: a reward model, a value function, a predictor of which candidate designs will score well. The memory note [3] shows what such a learner converges to. A consistent supervised learner trained on records that pair readings W with outcomes Y estimates the regression of the outcome on the reading — the best guess at Y given W — and not the regression of the outcome on the quantity, which is the response to X the loop was trying to learn. The two differ whenever W carries error, and under the observation model of the core paper the difference has a closed form: a linear response is learned with its slope attenuated by the instrument’s reliability — regression dilution, more than a century old — which in the paper’s coordinate shows as a concave power law. The papers call this the artifact, because it is structure present in the record and absent from the

world. It is a different failure from step 2’s: there, a curved use of a number that was already collapsed; here, the reading standing in for the quantity, with no curvature needed at all. In the closed-form case the response is linear in the quantity and the whole effect is the second kind.

The learner cannot do otherwise. It is not making a mistake; it is estimating exactly what the record contains, and, in the model of the papers, a record of noisy readings and noisy outcomes does not contain what it would need in order to do better: no regression run on it, in either direction, recovers the response without the error model, which is the one thing the record does not hold. This is why the usual remedies fail in a specific and predictable way. More data, more capacity, and a richer model class all produce faster and more faithful convergence to the artifact, and better optimization leaves the target where it was. Capability is spent on distance to the target; the target is set by the memory store.

4. Iteration compounds it, at a rate the instrument sets

Now let the loop run. The fixed-point paper [4] defines the population operator of a self-improvement loop and proves what happens to the artifact under three regimes, which between them cover the real pipelines.

In the first, each generation keeps the original outcomes as its labels and refreshes only its readings. This is the honest loop, and the artifact is its fixed point: every generation converges to the same flattened response as the first, and no generation improves on it, however capable. In the second, each generation is labeled by the previous generation’s predictions and trained against a fresh reading of the same quantity — which is what distillation does, what pseudo-labeling does, what a reward model trained on the last policy’s outputs does, and what an agent trained on its predecessor’s records does. Here the learned response does not stay put. It contracts, generation by generation, toward a constant, the unconditional mean of the outcome, and the contraction is geometric at a rate fixed by one number: the squared maximal correlation between a reading and the quantity it measures. That number is a property of the instrument. A reading that pins the quantity down nearly exactly forgets slowly; a noisy reading forgets in a few generations. Nothing downstream changes the rate — not the architecture, not the optimizer, not the sample size — because the theorem is about the target, not the fit. The third regime is the mixture, which retains a fraction of ground truth each generation and settles at its own fixed point between the other two.

This is regression to the mean, compounded once per generation. It should be distinguished from model collapse [6, 7], which it is not. Model collapse is a finite-sample phenomenon: tails are undersampled, the fitted distribution narrows, and the effect is weakened by more data and by retaining real data. The contraction here has infinite data per generation, samples nothing, loses no tails, and is sharpened rather than cured by more data; and retaining real data does not help when the real data is itself point-valued, because the first generation is already at the artifact.

5. The loop’s own evaluator cannot see it

This is the step that answers Good’s proviso. Under the second regime, the loop’s own training loss — the error of each generation against the labels the previous generation gave it — goes

to zero as the response is forgotten. The loop reports success at exactly the moment it has lost what it set out to learn. Under the first regime the situation is worse in a quieter way: no evaluation against the record can detect the artifact at all, because the artifact is the record's own regression function, and any evaluator that reads the same memory store will confirm it. An evaluator that kept the original outcomes, outside the loop, would see the error rising toward the variance of the true response. But the record the loop trains from cannot represent that, and a self-improving system, by construction, trains from its own record.

So whether a loop is improving is not a question its score can answer. It is a question about the memory store: does it keep, for each reading, the dispersion the theorem needs, or only the number? That is a design property. It cannot be tested from inside the loop, and it can be checked from outside; the memory note [3] gives three conformance checks that test whether an implementation carries a reading as a law rather than as a point. This is the kind of question Codd asked of relational systems in 1981 [8]: not how well a system performs, but whether it has the capability at all.

The repair is at the memory store, and it is cheap

The fixed-point paper's last theorem says what happens when the memory store keeps the reading as what it is: a law over the quantity, with the instrument's error model attached, conditioned anew on each fresh reading. Then the same readings, under the same learner, converge on the truth. In the simplest case the cost is two numbers per cell instead of one, and the second number is already on the instrument's calibration certificate. What changes is the write convention — not the learner, not the compute, and not the pace of the research.

What is and is not claimed

What is claimed is narrow: the loss sits in the memory store, the loop's own evaluator cannot see it, and a write convention removes it. What is not claimed is wider. The theorems are about measured quantities: scores, rewards, fitness values, losses, and the telemetry, sensor readings, and logged metrics a system keeps about its own operation. They say nothing about text or code as such. What a self-improving system trains on is mostly text and code; what it selects on is a number, and the number is covered. The results are population statements about targets, so they bound what any learner can achieve from a given record and say nothing about how fast a particular system approaches the bound. No claim is made about any deployed system.

The evidence that this is not hypothetical

For as long as the record of strong solar driving had been kept, Earth's magnetosphere was believed to saturate under it. The saturation was in the data, was modeled as physics, and was built into estimates of space-weather risk. In 2026 Sivadas and colleagues showed in *Nature* [5] that it is what a linear response looks like when the driver is stored as a value and its measurement error is stored nowhere — regression to the mean, read as a ceiling — and that removing the artifact roughly doubles the estimated impact of a storm on the scale of the 1859

Carrington event. No self-improving system was involved. There was only a memory store built on the universal convention, and generations of careful people computing carefully from it.

The question to ask

The current debate asks whether a self-improving system will become too capable to control. The question this note would put first is smaller, and it can be answered today for any loop that scores itself: what does its memory store keep of a reading, besides the number? If the answer is nothing, then the loop’s fixed point has already been chosen by the write convention, its score will report success on the way there, and Good’s proviso — that the machine be able to tell us how it is doing — was failed before the machine was built.

The property at stake is not, in the end, a safety property, though safety is where it bites first. It is a correctness property, and correctness is prior: if machines are going to reason about the physical world — in science, engineering, and medicine, wherever their inputs are measured — they must be able to reason from what was actually observed to conclusions that are actually warranted, and a memory that asserts a precision it never measured denies them that at the first step. Safety is the special case where being wrong hurts.

References

1. C. Metz, “The Liftoff Scenario That Terrifies A.I. Doomsayers,” *The New York Times*, September 16, 2026.
2. M. P. Laing, “Measure-Relational Algebra: The Core,” realql.org, September 2026, <https://doi.org/10.5281/zenodo.22999911>. Theorem 3.2, Corollary 3.4, Remark 3.5, Proposition 4.1.
3. M. P. Laing, “What a Memory Must Preserve: Machine Reasoning over Uncertain Observations,” realql.org, September 2026. Proposition 1, Corollary 2, the identification bounds of §4, and the conformance checks of §5.
4. M. P. Laing, “The Fixed Point of Self-Improvement: Three Limits of a Learner That Trains on Its Own Memory,” realql.org, September 2026. The theorems for Loops A, B, and C, the corollary on the loop’s own loss, and §§5–7.
5. N. Sivadas, D. Sibeck, V. Subramanyan, M.-T. Walach, et al., “Regression to the mean can explain saturation of geomagnetic storms,” *Nature* 655, 1143–1147 (2026).
6. I. Shumailov, Z. Shumaylov, Y. Zhao, N. Papernot, R. Anderson, Y. Gal, “AI models collapse when trained on recursively generated data,” *Nature* 631, 755–759 (2024).
7. M. Gerstgrasser et al., “Is model collapse inevitable? Breaking the curse of recursion by accumulating real and synthetic data,” arXiv:2404.01413 (2024).
8. E. F. Codd, “Relational database: a practical foundation for productivity,” *Communications of the ACM* 25(2), 109–117 (1982); the 1981 ACM Turing Award lecture.
9. I. J. Good, “Speculations concerning the first ultraintelligent machine,” *Advances in Computers* 6, 31–88 (1965).
10. M. Favaro and J. Clark, “When AI Builds Itself,” Anthropic Institute, June 2026.