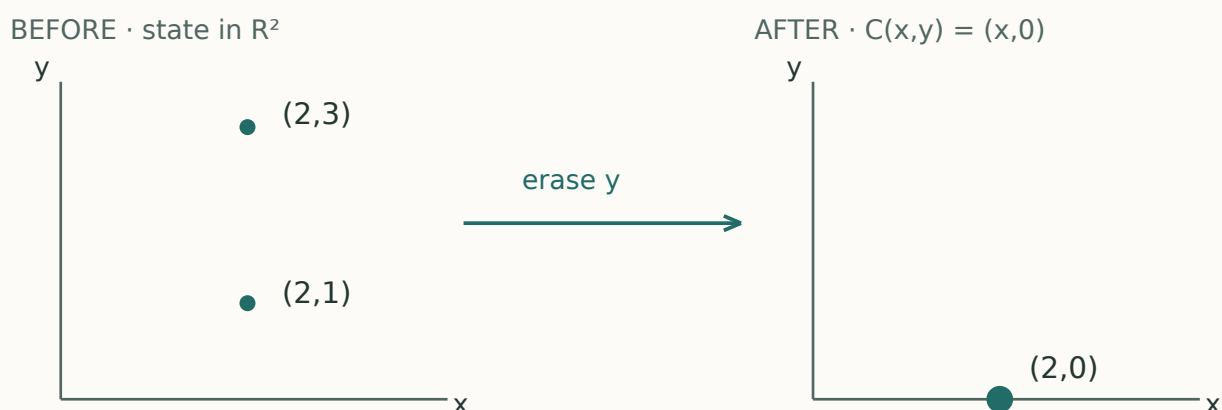


# What a rule actually says

From chosen operations to claims we can defend

Before asking whether a machine has discovered a law, ask what has actually been defined. A line of code can specify an operation. A proof can establish what follows from it. A test can reveal how a fitted model behaves. These are different kinds of knowledge.



The picture makes a collision visible. The proof establishes noninjectivity.

Figure 1 · A deliberately constructed operation, not a discovered physical regularity. Different states become indistinguishable after projection.

## Five words that should not blur together

A definition fixes meaning. An assumption states what we are taking as given. A theorem states a consequence proved under those meanings and assumptions. An empirical finding reports what a specified test observed. A physical or empirical law describes a well-supported regularity, with a domain and predictions that can be challenged by evidence—not metaphysical certainty.

This edition begins with a small, explicit mathematical object, then follows one completed experiment. It does not finish the specification of the intended new machine, establish a new physical law, or publish the research. Original reports remain unchanged.

# A projection, fully stated

## The object and the task

For this explanatory example, the state domain is  $\mathbb{R}^2$ : a state  $s=(x,y)$  has two real coordinates. The action set contains one supplied operation  $C$ . Its transition is  $C(x,y)=(x,0)$ . An observation after the action is the pair  $(x,0)$ ; it does not contain the old  $y$ . The readout task  $q_x(s)=x$  asks only for the first coordinate.

State  $s=(x,y)$  Action  $C$  Transition  $C(s)=(x,0)$

Representation  $\phi(s)=C(s)$  Readout  $h(u,v)=u$

For the  $x$ -only task,  $h(\phi(s))=q_x(s)$  for every admissible state. That equality is an exact consequence of these definitions. It is not evidence that  $x$  alone will answer every future question.

## The noninjectivity proof

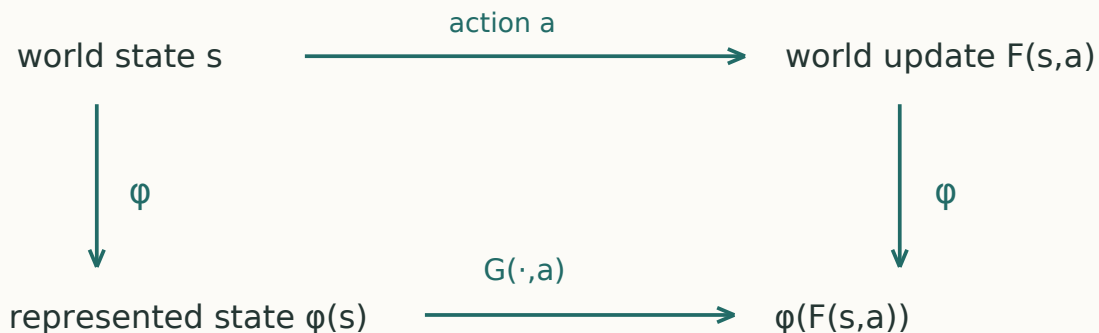
Choose  $s_1=(2,1)$  and  $s_2=(2,3)$ . They are distinct, yet  $C(s_1)=C(s_2)=(2,0)$ . Therefore  $C$  is not injective. Suppose a decoder  $D$  could recover every original state from  $C(s)$  alone. At the same input  $(2,0)$ , it would have to return both  $s_1$  and  $s_2$ . A single-valued function cannot do that; such a universal decoder does not exist.

The drawing shows the collision. The argument proves the claim. A restriction such as  $y=1$ , a remembered earlier observation, or an extra stored  $y$  would change the information available and therefore change the problem. No such side information is assumed here.

## Where the checked proof differs

The existing Lean core uses unbounded integer states  $\mathbb{Z}^2$ , not this real-coordinate example. Its checked projection witness uses  $(0,0)$  and  $(0,1)$ . The same elementary collision argument applies here, but this new explanatory statement has not been rerun through Lean in this editorial task. [S2]

# Enough for a question is not enough for a world



Required equality:  $G(\varphi(s), a) = \varphi(F(s, a))$

Figure 2 · The square must commute on the stated domain and action set. Drawing it does not establish that equality for a learned representation.

Let  $F(s,a)$  be the world transition,  $\varphi(s)$  its representation, and  $G$  its represented transition. Closure requires  $G(\varphi(s), a) = \varphi(F(s, a))$ . A separate readout requirement  $h(\varphi(s)) = q(s)$  connects the representation to the task. Neither condition may be silently assumed for an unspecified machine. [S3]

The projection is closed under  $C$  itself: applying  $C$  again changes nothing. But add a rotation  $F(x,y) = (-y,x)$ . States  $(0,0)$  and  $(0,1)$  begin with the same represented state; after rotation their represented states are  $(0,0)$  and  $(-1,0)$ . No deterministic  $G$  given only the original representation and that action can produce both answers. The action set matters.

## What has already been formalized

The prior bounded Lean core checked 16 declarations: exact affine identities, composition and associativity, finite-list execution, noninjectivity, and conditional coordinate-change results with supplied algebraic assumptions. No sorry, admit or new user-declared axiom was used; standard propext and Quot.sound dependencies remain where reported. [S2]

Bend ordinary checking and four small runtime fixtures passed. Independent --verdict validation failed with a kernel mismatch. F32 execution also has a retained counterexample: starting at 1, adding 16777216 then subtracting it gives 0 sequentially but 1 when offsets are composed first. Exact arithmetic proofs do not establish universal floating-point or compiler correspondence.

# One experiment, two routes to learning

The new question is narrow: does the observed endpoint-training advantage persist when target noise is not Gaussian? This completed simulator study changes the noise distribution, not the architecture, fitting rules, training designs or evaluation protocol. [S1]

## PRIMITIVE LABELS

state → one operation → target



OLS on 176 pairs per generator

T, S, C · 528 pairs · 18 coefficients

## SEQUENCE ENDPOINT LABELS

state → several operations → target



joint fit on 33 words × 16 pairs

528 pairs · 18 coefficients · 3 starts



Both predict held-out words by composition

Figure 3 · Equal counts do not make these observation designs equally informative. The intermediate operations and composition rule are supplied to both routes.

The supplied operations act on two coordinates: T rotates by  $35^\circ$ ;  $S(x,y)=(x+0.45y+0.2,y-0.15)$ ;  $C(x,y)=(x,0)$ . Both routes learn 18 affine coefficients. Primitive fitting uses ordinary least squares on single-operation pairs. Endpoint fitting learns the same three shared generators jointly from sequence endpoints, with three train-only starts.

All inputs used for training and evaluation are clean. Only training target coordinates are perturbed, with  $\sigma=0.03$  and population marginal variance 0.0009. The primary endpoint is the six-step TSTSTS; TS and the eight-step lossy TSTCSTST are prespecified secondary endpoints. Words execute left to right. [S1]

There are 32 paired seed replicates, 8101–8132, and 96 clean held-out states per endpoint. The shifted test rectangle is  $[-4,4] \times [-3,3]$ . Each distribution shares latent states and optimizer starts; route pairing reuses one target-noise draw in pair order. These are dependent comparisons across distributions, not three independent studies.

# Three noise distributions, stated precisely

The earlier report called these a “noise law.” Here that phrase means a probability distribution. It does not mean a discovered fundamental law. Let  $Z$  be a standard normal random variable; in the mixture,  $B$  is an independent Bernoulli variable with probability 0.05 of being 1.

## Gaussian reference

$$\varepsilon = 0.03 Z, Z \sim N(0,1)$$

The reference uses ordinary centered Gaussian target noise. It is one modeling choice, not a default truth about measurement.

## Standardized Student-t

$$\varepsilon = 0.03 \sqrt{3/5} U, U \sim t_5$$

Five degrees of freedom give heavier tails than the Gaussian. The multiplier standardizes population variance to one before applying  $\sigma$ . The standardized fourth moment is 9.

## Sparse broad-noise mixture

$$\varepsilon = 0.03 Z \times (10 \text{ if } B=1, \text{ otherwise } 1) / \sqrt{5.95}$$

Most draws use the narrow component; 5% use a tenfold scale. Since  $0.95 \times 1^2 + 0.05 \times 10^2 = 5.95$ , dividing by  $\sqrt{5.95}$  matches population variance. It does not match each realized sample’s variance or information content.

No clipping, sample renormalization, redraw or outlier deletion was used. Both alternatives have unbounded support and finite fourth moments. They are two specified alternatives—not a robustness guarantee over all heavy-tailed distributions. Actual draws and mixture masks remain in the original raw archive. [S1]

# The seed-level story

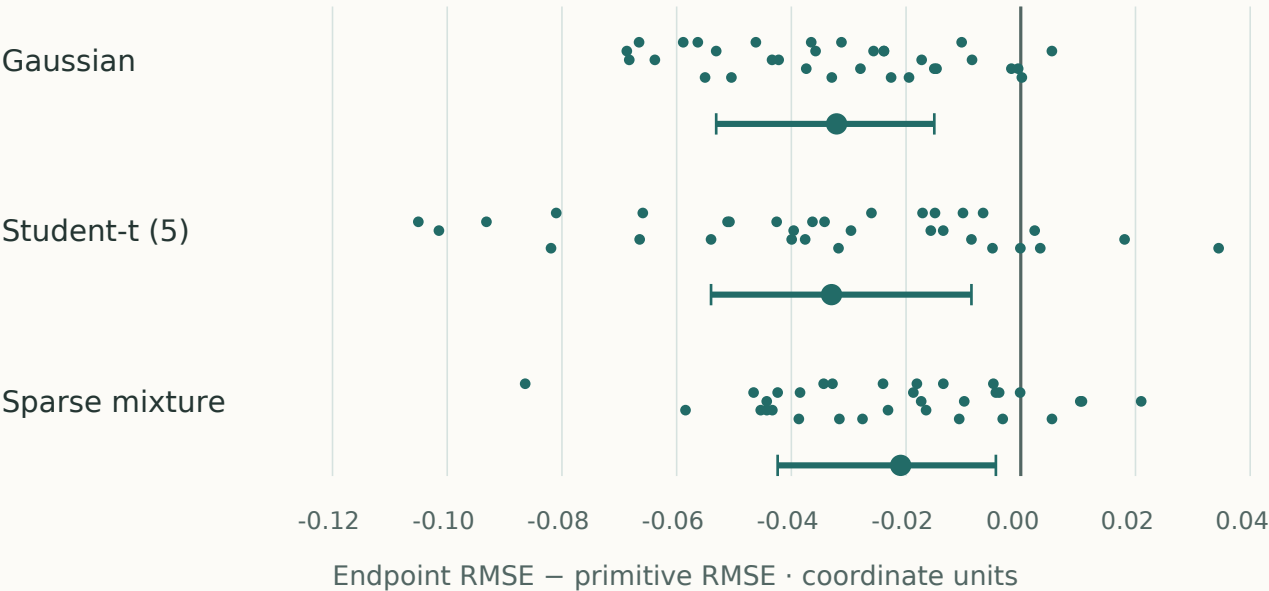


Figure 4 · Every small dot is an actual saved seed gap. Vertical offsets only separate marks. Large dots and bars show the median and simultaneous median interval. Negative means lower endpoint error.

| Distribution   | Mean primitive | Mean endpoint | Wins  |
|----------------|----------------|---------------|-------|
| Gaussian       | 0.058618       | 0.025726      | 30/32 |
| Student-t (5)  | 0.069593       | 0.035169      | 28/32 |
| Sparse mixture | 0.050321       | 0.027342      | 28/32 |

For six-step TSTSTS on shifted clean states, each of the three simultaneous median-gap intervals lies below zero. The endpoint route has lower error in 30 of 32 Gaussian seed pairs and 28 of 32 under each alternative. The measured gaps are in synthetic coordinate units. [S1]

The table’s arithmetic means describe this batch only. The inferential conclusion is about the population median paired seed gap under this simulator protocol. It is not a population-mean claim, a rare-tail failure bound, or a comparison proving one noise distribution easier than another.

# The result survives these tests—not every test

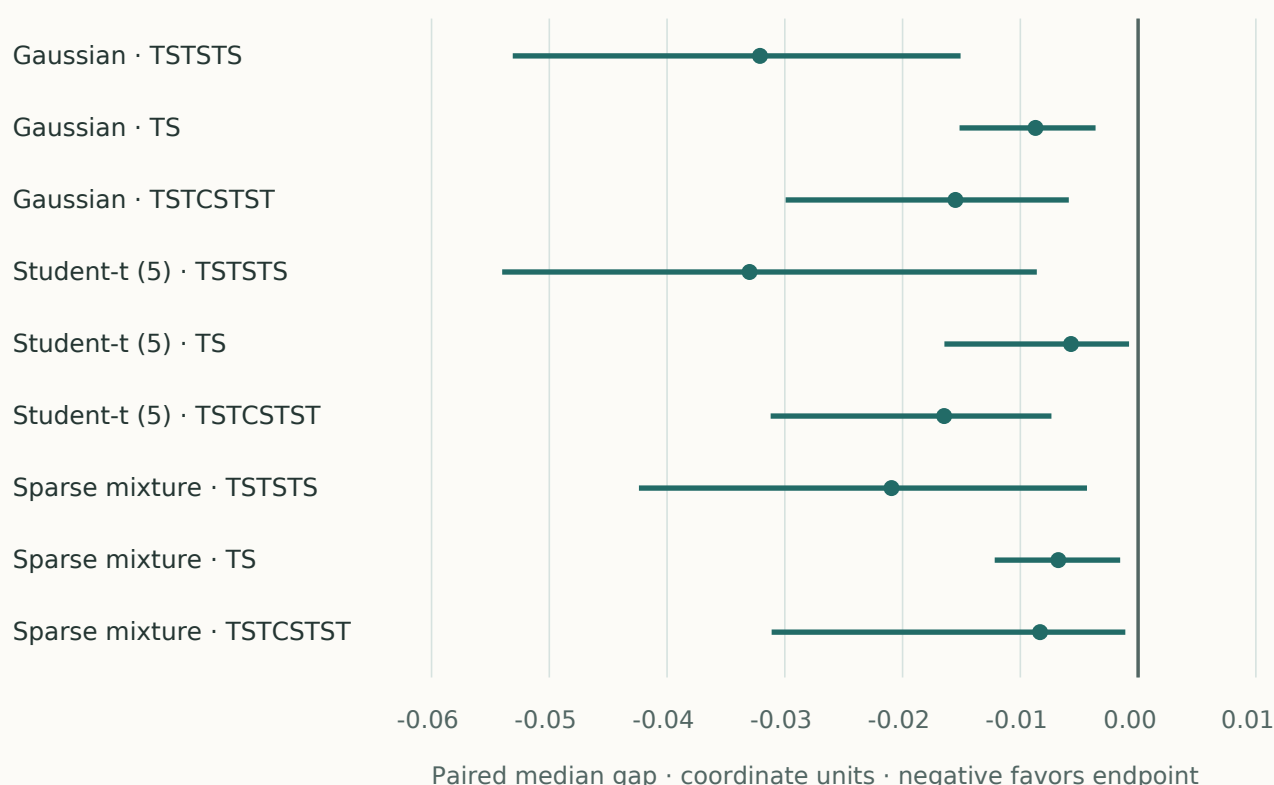


Figure 5 · All nine intervals lie below zero. These are intervals for paired median RMSE gaps, not intervals for each method's mean error or the range of individual seeds.

The same direction holds for the short held-out TS endpoint and the longer lossy TSTCSTST. The defensible claim is therefore limited but useful: the median endpoint-training advantage is not confined to Gaussian target noise in these two alternatives and three prespecified endpoints.

Projection in the longer chain erases information. Lower forward prediction error does not recover the erased original coordinate. Both learners still use the same supplied affine family, alphabet, coordinates and compositional recurrence. No architecture discovery, learned rule family, or full OaK agent follows.

No optimizer failure, excluded seed or nonfinite prediction was recorded in this batch's frozen diagnostics. That observed absence is not a guarantee that failures are impossible or that rare catastrophic losses have been measured precisely.  
[S1]

# What the uncertainty and the costs mean

## Why the median has an interval

For each of the nine cells, sort the 32 saved paired gaps. The frozen distribution-free rule takes the 8th and 25th values as interval endpoints. Bonferroni adjustment accounts for all nine prespecified cells. Under independent seed replicates and the sampling/median conditions, simultaneous coverage is at least 95%; discreteness gives a conservative lower bound of 98.1078%. [S1]

No Gaussian variance estimate or mean-based t interval is imported into this inference. Thirty-two seed replicates do not precisely characterize rare tails. The dots in Figure 4 show the actual batch; the interval concerns its generating median, not an interval containing 95% of dots.

## Equal records, unequal information and work

Each route receives 528 labeled pairs and uses 18 coefficients. Primitive labels describe individual actions; endpoint labels describe different states and sequences. Equal scalar slots do not establish equal acquisition, effective independent information, computational work or statistical efficiency.

| Recorded fitting work    | Count      | CPU wall time |
|--------------------------|------------|---------------|
| Primitive OLS            | 288        | 0.0211 s      |
| Length-one joint control | 288 starts | 1.009 s       |
| Endpoint joint fit       | 288 starts | 7.032 s       |

These are machine-specific local timings, not FLOPs, peak memory or physical sensing costs. The unchanged length-one joint control uses exactly the primitive observations and agrees numerically with OLS: coefficient difference  $\leq 5.995 \times 10^{-15}$  and prediction difference  $\leq 9.326 \times 10^{-14}$ . This checks one numerical route; it does not prove global solver optimality.

# Evidence stays visible behind the story

Each row contains 32 paired seed replicates.  $\Delta$  means endpoint RMSE minus primitive RMSE. Means are descriptive; intervals refer to the median paired  $\Delta$ . All numbers below come from saved groups.json, rounded only for display. [S1]

| Distribution / word       | Median $\Delta$ | Simultaneous interval  | Wins  |
|---------------------------|-----------------|------------------------|-------|
| Gaussian / TSTSTS         | -0.032108       | [-0.053104, -0.015076] | 30/32 |
| Gaussian / TS             | -0.008713       | [-0.015163, -0.003612] | 29/32 |
| Gaussian / TSTCSTST       | -0.015512       | [-0.029930, -0.005884] | 29/32 |
| Student-t (5) / TSTSTS    | -0.032994       | [-0.054002, -0.008601] | 28/32 |
| Student-t (5) / TS        | -0.005697       | [-0.016448, -0.000768] | 27/32 |
| Student-t (5) / TSTCSTST  | -0.016470       | [-0.031204, -0.007359] | 27/32 |
| Sparse mixture / TSTSTS   | -0.020940       | [-0.042389, -0.004340] | 28/32 |
| Sparse mixture / TS       | -0.006775       | [-0.012177, -0.001522] | 25/32 |
| Sparse mixture / TSTCSTST | -0.008325       | [-0.031120, -0.001085] | 25/32 |

## What the receipts establish

The retained saved-data audit reports 7,743 comparisons and maximum residual 5.55e-15. It checks recurrence, noise construction, costs and train-only selection, RMSE, median intervals and sign tails without optimizer refitting. One deterministic replay matches 7,968 arrays exactly. These are software/evidence checks, not independent replication. This edition recomputes all 288 saved deltas and nine displayed summaries and intervals without fitting.

The earlier covariance audit remains FAILED: empirical correlation  $-0.070475848$  exceeded the frozen absolute 0.06 threshold in a separate noise-reuse study. Its receipt is preserved; this target-only result neither repairs nor silently discards it. No new experiments, proof-tool runs or model calls were performed for this edition.

# Sources, revisions and a reusable reading language

## Source ledger

[S1] Non-Gaussian target-noise sensitivity, 3 October 2026. Primary sources are the frozen protocol, 288 seed-endpoint rows, nine group summaries, cost records, and original execution/audit receipts. The source report and full raw/replay packages remain unchanged. Selected source records are included in this edition's editable package; it is not the full raw-array replay archive.

[S2] "An executable core, and what its proofs guarantee," 1 October 2026, plus its Bend-Lean source/log package. Reviewed the complete three-page note and proof-package inventory. Historical proof and runtime receipts are cited, not rerun here. The projection example's real-coordinate proof is explanatory mathematics, not a newly checked Lean declaration.

[S3] "Composition is not reconstruction," evidence-synthesis edition 3, 3 October 2026. Selected extracted passages of the 22-page manuscript were reviewed for definitions, proof scope, access limits and failures; not a full review. Its restricted four-sensor FD001 test had 15.25% higher engine-mean MSE and failed noninferiority. This edition replaces neither that manuscript nor the 67-page learner book.

## What changed in the presentation

The original audit-first report is now a definition-to-evidence narrative. Original vector diagrams explain collisions and closure; plotted marks map to actual saved rows. Axes have numerical ticks, coordinate units, a sign convention and explicit uncertainty. Searchable embedded text replaces image-only pages. A restrained serif reading face, shorter measure and generous leading make both print and mobile reading gentler.

Two official public Substack guides were retrieved without login. Their initial viewports and selected paragraph styles were visually inspected: one has a spacious serif column; the other, a warm surface and monospaced body. We borrow readability and pacing, not branded assets, navigation or a fixed "Substack font." Review did not cover every illustration or publication.

Authorship remains unassigned; this is a prepared reading edition, not a submitted, peer-reviewed or publicly published paper. HTML source links open the original references. Design/claim revision notes and the complete figure-to-source map are in the editable package.

[Original experiment report](#)

[Technical synthesis](#)

[Checked proof note](#)

[Substack customization guide](#)

[Substack visual identity guide](#)