

The world a model must keep

A guided Phi9 book of information, learning and action

Shubham Attri · Independent researcher and founder, Phi9
Learner edition 2 · 3 October 2026

A model keeps a view of the world, not the world itself. Follow what survives an observation, a transformation and a decision—and learn to separate a worked example, a mathematical guarantee and an experimental result.

This complete fourteen-chapter learning edition revisits the chapter names and progression, redraws the explanatory diagrams, and adds traced examples and worked exercises throughout. Precise technical passages and older experiment results are retained where needed. It is an assisted pedagogical rewrite, not fourteen new studies. Original/live editions remain unchanged; no experiment was rerun and no new generalization is claimed.

Contents

How to read and test this book	3
1 What a Number Leaves Out	4
2 Keep the Distinctions an Action Needs	7
3 When a Representation Can Predict	10
4 Learn a Rule Without Inventing Information	13
5 Follow Motion, Not Just Position	16
6 Change Coordinates; Check What Survives	19
7 Remember, Condition, and Update Beliefs	22
8 Separate a Proof from a Working Program	25
9 Discover Rules—and Count the Search	29
10 Know Where a Model Is Unreliable	33
11 Observe More, Pay More, Update Earlier	37
12 Refit the Rule or Change the Observation?	41
13 Choose Without Fooling Yourself	45
14 Turn Predictions into Accountable Actions	49
Appendix A The technical contract	53
Appendix B Evidence ledger, not a combined benchmark	55
Appendix C Terms that should stay distinct	58
Appendix D Sources, coverage and editorial method	59
Appendix E What the later transformation studies establish	64

How to read and test this book

Start with each chapter's scene. Trace the states and transformations by hand; then inspect the equations. Attempt its exercise before the worked answer. The progression runs from information loss to actions, a predictive contract, learning, motion and coordinate changes; then memory, proof, empirical reliability, sensing, diagnosis, calibration and accountable action.

All fourteen names were revisited. The chapter order was reviewed and retained because later questions depend on earlier distinctions. This complete learner edition rewrites the teaching entry points, adds traces and exercises, and redraws every explanatory diagram. It retains the precise technical core and older source trail rather than inventing replacement experiments.

Worked arithmetic and schematic drawings are not measurements. Empirical comparisons remain tied to their own protocols; they are not a combined league table. New affine results and a proposed seeded-familiarity protocol are separately identified. The mathematical appendix, evidence ledger, glossary and source guide offer a second reading route.

The original/live edition is unchanged. No study was rerun, no new Codex review was performed, and physical or medical applications remain proposals rather than validated robot or clinical systems. The full Welch Labs book was unavailable: only the previously documented public sample and exercises were inspected. Its text and artwork are not reproduced.

1 | What a Number Leaves Out

Start with a total is not an arrangement

Put four counters on the left and six on the right. Then put seven on the left and three on the right. Both displays read ten. Before reading further, double only the counters on the left: the displays become fourteen and seventeen. A representation can answer the present question and still fail the next one.

Trace: $(4,6) \rightarrow \text{total } 10 \rightarrow \text{double left} \rightarrow \text{total } 14$. Trace: $(7,3) \rightarrow \text{total } 10 \rightarrow \text{double left} \rightarrow \text{total } 17$. The collision is at the middle measurement, not in the arithmetic.

The question this chapter answers

When does a measurement retain enough for the next action?

Two boxes, one measurement

There are ten counters on a table, divided between a left box and a right box. A display shows their total. Ask how many counters there are, and it gives an exact answer: ten.

Now double only the contents of the left box. Start with four on the left and six on the right, and the new total is fourteen. Start with seven and three, and it is seventeen. Both arrangements gave the same original reading. After the action, they no longer do.

The display was not wrong. We asked it a question it could answer, then used that answer for a question it could not.

ONE MEASUREMENT · TWO FUTURES

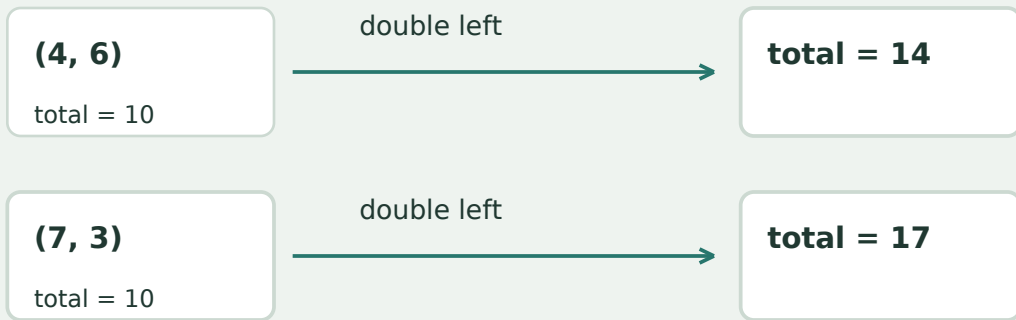


Figure 1. Constructed example. The same total conceals two arrangements; the same left-only intervention produces different successors.

Let u be the left count and v the right count. The full arrangement is $x = (u, v)$. The display retains only $z = u + v$. Doubling the left changes the arrangement to $(2u, v)$, so the next displayed total is:

$$z' = 2u + v = z + u$$

The missing term is u . A deterministic rule given only z and “double left” must return the same answer for both arrangements. It cannot return fourteen for one and seventeen for the other.

More computation cannot recover a distinction that the input never recorded. The obstacle here is information, not the speed of the calculation.

Enough for which action?

Change the instruction to “double both”. The new total is $2u + 2v = 2z$. Now the display has everything needed to compute the next reading. The same representation succeeds for one action and fails for another.

Enough information means enough for the outputs we require and the actions we allow—not simply a large representation or an accurate answer to an earlier task.

We can repair the left-only example by retaining u alongside z , allowing only actions whose effects depend on z , or restricting the arrangements—for

example, to $u = v$. The last choice also requires actions that preserve the restriction. These repairs change the information, the task or the assumptions.

Doubling only the left destroys equal box contents. Equality may let us predict the first step, but it cannot support the same rule indefinitely. The world need not maintain a convenient assumption.

An uncertain answer can still help

Suppose the left count is equally likely to be four or seven, with total ten in either case. After doubling the left, the total is fourteen or seventeen. The mean is 15.5. That average can be useful without being the exact outcome in either arrangement. Here, uncertainty belongs in the answer.

Try another question. Does a total of ten tell us whether the left box is empty? Compare (0,10) with (5,5). The display agrees; the answer does not. Does that total determine the result of doubling both? Yes.

We will call this a collision: two different states have the same representation. A collision is harmless when it merges distinctions the task never uses. It becomes a problem when a required answer or action separates those states again.

A model need not remember everything. It must keep the distinctions its promises depend on. This constructed arithmetic example is not a measured experiment. We can now give the distinctions coordinates and make the promise precise. [P1, P2]

Try it before reading the answer

Compare (2,8) with (9,1). Find the total after doubling both boxes and after doubling only the right box. Is the total alone sufficient for each action?

Worked answer and limit

Doubling both gives 20 in both cases: the compact update is $z' = 2z$. Doubling right gives 18 and 11; no function of the shared total 10 can return both answers. Keep the right count as well, or restrict the action set. This proves failure on these states, not on every possible domain.

Carry this forward

Next, give those distinctions coordinates and trace actions in order.

2 | Keep the Distinctions an Action Needs

Start with order is part of the operation

Start at (3,7). Swap the coordinates, then double the first: $(3,7) \rightarrow (7,3) \rightarrow (14,3)$. Reverse the order: $(3,7) \rightarrow (6,7) \rightarrow (7,6)$. The same ingredients do not imply the same transformation. Coordinates let us preserve the distinctions and trace the order explicitly.

Trace: the rightmost matrix acts first. B S x means swap first, then double. S B x means double first, then swap. Their totals are 17 and 13.

The question this chapter answers

How do coordinates and operation order preserve or destroy distinctions?

A vector keeps distinctions separate

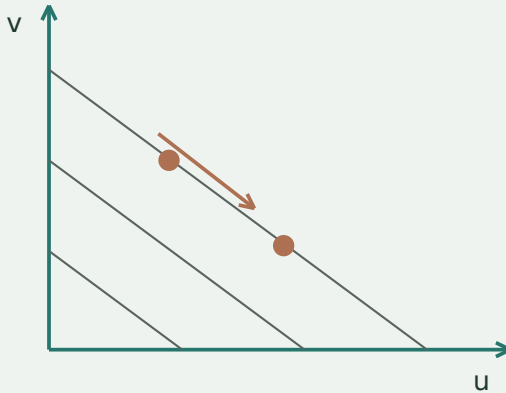
A scalar is one number. A vector keeps several coordinates in a specified order. For our boxes, $x = (3,7)$ means three on the left and seven on the right. Swap the coordinates to $(7,3)$, and we change the arrangement without changing the total. The meaning and order of the coordinates matter.

Adding $(1,0)$ gives $(4,7)$: one counter has been added to the left. Adding $(1,-1)$ transfers one from right to left, provided there is one to transfer. The physical states are nonnegative integers. Extending the formulas to real vectors helps us reason algebraically; it does not turn fractional or negative counters into physical objects.

$$P = \begin{bmatrix} 1 & 1 \end{bmatrix} \quad z = Px = u + v$$

The row matrix P describes what we retain. Its kernel contains all differences of the form $(t, -t)$. These differences leave the display unchanged. When a difference describes a legal transfer between our boxes, it moves counters without changing the total. A kernel contains invisible directions; it is not itself a set of legal physical states.

WHAT THE TOTAL CANNOT SEE



$$P(u,v) = u + v$$

One diagonal: one total.
Different points: different
arrangements.

Hidden direction: $(1, -1)$. Domain
restrictions still matter.

Figure 2. Schematic state space. Each diagonal is a fibre of the total map. Motion along a fibre changes the hidden arrangement, not the total.

A matrix is a linear transformation. Doubling both boxes uses $A = \text{diag}(2,2)$; doubling only the left uses $B = \text{diag}(2,1)$. Swapping uses $S(u,v) = (v,u)$. Apply B after S to $(3,7)$ and obtain $(14,3)$. Apply S after B and obtain $(7,6)$. Composition has an order. Before multiplying the matrices, trace the two instructions on the boxes and ask which arrangement each order produces.

What multiplication means

The expression ABx applies B first and A second. Associativity lets us regroup a fixed exact product, $(AB)C = A(BC)$. It does not allow us to reorder its factors. A map adding a counter is affine rather than linear: $T(x) = Ax + b$, with the offset b recording the addition.

The dimension of a representation is the number of its coordinates, not the number of vectors appearing in an equation. A two-dimensional state can have one current state vector, one velocity vector and many candidate actions. The number of arrows we draw is not the number of coordinates needed to describe a state.

For motion in a constrained world, ambient dimension and usable freedom may differ. A point moving around a circle occupies a two-coordinate plane, but locally has one degree of freedom along the circle. A robot's joints, object pose and contacts introduce different coordinates and constraints. We must name those meanings before counting them.

A worked compressed update

If A doubles both boxes, $PA = [2 \ 2] = 2P$. Thus we can update the total by multiplying it by two. For B, $PB = [2 \ 1]$. There is no scalar g such that $gP = [2 \ 1]$, because gP has equal entries. A compact rule exists in the first case and fails in the second.

Try an affine version: double both boxes and add one to the left. The total obeys $z' = 2z + 1$. The offset did not disappear during compression; it was mapped into the retained coordinate. The total can update itself because the action respects the distinctions it retains. We next turn that observation into a contract for a general representation. [P3, P4]

Try it before reading the answer

For $P(u,v)=u+v$ and $A(u,v)=(u+1,v)$, derive the compact update. Then test whether the same representation closes under $B(u,v)=(2u,v)$.

Worked answer and limit

$PA(u,v)=u+v+1$, so $G(z)=z+1$ works. $PB(u,v)=2u+v$, which differs for (4,6) and (7,3) despite equal totals. An affine offset is compatible with closure when it is represented; doubling left is not.

Carry this forward

Next, turn these checks into a precise contract for represented futures.

3 | When a Representation Can Predict

Start with test collisions before trusting the future

Suppose a model assigns the same label to two rooms. It may safely forget the room identity if every allowed action leads to the same next label and both rooms answer the required question identically. One counterexample is enough to break this promise. The contract is about all admitted states and actions, not a few successful predictions.

Trace: full state $x \rightarrow$ action $a \rightarrow F(x,a) \rightarrow$ representation $\varphi(F(x,a))$; compare with $x \rightarrow \varphi(x) \rightarrow$ represented update $G(\varphi(x),a)$. The destinations must agree.

The question this chapter answers

What exact condition lets a represented update exist?

Two routes should meet

Imagine two routes to the next reading. Follow the action in the full world, then measure the result. Or measure the starting state first, then update that measurement directly. If the representation is enough, the routes should meet.

Let X be the admitted states, A the admitted actions, $F(x,a)$ the deterministic successor, $q(x)$ the answer required and $\varphi(x)$ the representation retained. We seek a readout h and a represented update G , defined on the image $\varphi(X)$:

$$h(\varphi(x)) = q(x)$$

$$G(\varphi(x),a) = \varphi(F(x,a))$$

THE TWO ROUTES MUST AGREE



Figure 3. The closure square. Updating then representing must agree with representing then updating, on every admitted state and action.

The first equation preserves the answer. The second says the represented dynamics are closed: we can compute the next summary from the current summary and the action, without recovering the full state. Together they preserve the requested output after every finite admitted action word, provided the assumptions hold at each visited state. For actions chosen during execution, the policy and action-legality conditions must also preserve the contract.

The collision test

Such maps exist precisely when equal representations imply equal answers and equal represented successors under each action. Necessity is immediate: a function applied to identical inputs has identical outputs. For sufficiency, choose any state representing z and define $h(z) = q(x)$, $G(z,a) = \varphi(F(x,a))$. The collision conditions make those definitions independent of the state chosen. An induction extends the one-step relation along any finite action sequence.

This classical quotient argument is closely related to automata and state abstraction. It establishes when a represented update exists. It does not tell us how to compute that update efficiently, learn the representation or handle values outside $\varphi(X)$; nor does it require the represented state set to be finite. [R1, R4]

Closure is stronger than task-only prediction

Suppose the required answer is always zero. A representation also carries an irrelevant label. Two states share label 0, but their successors have labels 1 and 2. The answer remains zero forever; it is perfectly predictable. The entire chosen representation nevertheless fails to close, because $G(0)$ cannot be both 1 and 2.

Failure of whole-state closure does not mean every task predictor must fail. A coarser representation may discard the inconsistent label; a richer history may preserve a needed distinction. The task determines which matters.

The action chooser can hide information too

Even if each fixed action closes, a policy using hidden information may break the closed loop. Let $\varphi(x,h) = x$. “Advance” increments x ; “stay” leaves it unchanged. Both close individually. A scheduler advances when hidden h is zero and stays otherwise. States $(0,0)$ and $(0,1)$ then have equal representations and different represented successors.

A sufficient safeguard is for the policy to depend only on $\varphi(x)$. It is not always necessary: different actions can have identical represented effects. The essential check is the resulting closed-loop transition. If legal actions, timing, scheduling or context affect the result, the contract must account for them too. We have supplied the rule. Can examples identify one that meets this contract? [P4, R4]

Try it before reading the answer

Let $X = \{a,b,c\}$, $\varphi(a)=\varphi(b)=0$ and $\varphi(c)=1$. An action maps $a \rightarrow c$, $b \rightarrow b$, $c \rightarrow c$. Can G exist? What if the action instead maps $b \rightarrow c$?

Worked answer and limit

Initially $G(0)$ would have to be both 1 and 0, so it cannot exist. After changing $b \rightarrow c$, the successors of a and b agree: $G(0)=1$ and $G(1)=1$ work. A readout contract must still be checked separately; closure alone does not establish the required answer.

Carry this forward

Next, ask what observations can teach about an update that the contract permits.

4 | Learn a Rule Without Inventing Information

Start with one update, followed all the way

A learner sees $x=2$ and target $y=6$. With weight $w=1$ it predicts 2. Half the squared error is 8. The derivative with respect to w is -8 ; a step of 0.1 gives $w=1.8$, prediction 3.6 and loss 2.88. We improved a fitted rule. We did not create an observation that was never present.

Trace: input 2 $\rightarrow$ multiply by 1 $\rightarrow$ prediction 2 $\rightarrow$ residual -4 $\rightarrow$ gradient -8 $\rightarrow$ weight 1.8 $\rightarrow$ prediction 3.6. These are worked calculations, not a new experiment.

The question this chapter answers

What changes during fitting, and what must evaluation keep separate?

What a fit changes

In the box example, we supplied the rule “double both”. Now suppose we give a learner pairs of observed totals z and next totals z' instead. It estimates a multiplier g by fitting $z' \approx gz$. Minimize the sum of squared residuals, differentiate with respect to g , and obtain:

$$g = \Sigma(zz') / \Sigma(z^2), \text{ when } \Sigma(z^2) > 0$$

In the executed foundation toy, 66 exact training pairs gave $g = 2$. On 55 withheld starting states, the maximum exact error was zero. These are new inputs under the same constructed rule, not independent environments or evidence of general intelligence. Finite testing and an all-state algebraic argument have different roles. [P3]

Follow one gradient all the way

Consider a deliberately small new teaching example, not a new training run. A predictor uses $\hat{y} = wx$. On $x = 2$, desired $y = 6$, current $w = 1$, it predicts two. Which way would you move the weight to bring the prediction closer to the desired output?

Set $L = (\hat{y} - y)^2/2$. The error is -4 and the derivative is $dL/dw = (wx - y)x = -8$. A step of size 0.1 gives $w' = 1 - 0.1(-8) = 1.8$. The new prediction is 3.6 and the new loss is 2.88 , down from eight.

FOLLOW ONE LEARNING STEP

input $x = 2$



weight $w = 1$



prediction 2

target $6 \rightarrow$ residual $-4 \rightarrow$ derivative -8

step $0.1 \rightarrow w' = 1.8 \rightarrow$ prediction 3.6

half-squared loss: $8 \rightarrow 2.88$; worked arithmetic

Figure 4. Worked arithmetic, not a new training run. A derivative tells how loss responds locally to the parameter; the chosen step changes the prediction.

Trace the figure from weight to prediction, error and update. The arithmetic shows why the gradient points toward a local reduction in this example. It does not prove that arbitrary large steps improve every loss, that a neural network finds a global optimum, or that one example determines the right rule.

For a chain of operations, backpropagation applies the chain rule. If $h = w_1x$ and $\hat{y} = w_2h$, then $dL/dw_1 = (\hat{y} - y)w_2x$ and $dL/dw_2 = (\hat{y} - y)h$. The forward pass computes predictions; the backward pass distributes derivatives; an optimizer changes parameters. Each stage does a different job.

Expressiveness is not a guarantee

A network stacks parameterized maps and nonlinearities. Without nonlinearities, several linear layers compose into one linear map. Nonlinearities permit richer functions, but the information available at the input still limits the output. If two hidden arrangements produce the identical retained total, a deterministic network seeing only that total cannot separate them.

Training changes a fitted function. It may also change a learned encoder. Neither operation makes discarded distinctions recoverable without assumptions or additional observations. Approximation results about network families do not say that finite data, an optimizer and a deployment distribution will deliver every desired function.

Keep evaluation out of the fit

We learn on training data, choose settings using a declared validation procedure, and measure generalization on held-out data. After repeated choices, validation performance itself can become optimistic. Later chapters show this in a retained experiment.

A learned model has a domain, a target, a loss and a resource budget. An explanation of learning should name each of them. Ask what changed: weights, features, memory, available sensing or the evaluator's rules? To follow a learned system through time, we also need to separate its state from its motion and its readout. [P5, R17]

Try it before reading the answer

Take the same example with step size 0.25. Compute the updated weight and loss. Does success on this one point establish generalization?

Worked answer and limit

$w' = 1 - 0.25(-8) = 3$, prediction=6 and loss=0. The single point is fit exactly. It says nothing about other inputs, changing targets or missing state; held-out evaluation is needed. A favorable step here is not a safe universal learning rate.

Carry this forward

Next, separate the state that evolves from the readout we ask for.

5 | Follow Motion, Not Just Position

Start with two identical positions can move apart

A pendulum can pass through its center while moving left or right. A photograph of position alone cannot tell those futures apart. For the ideal oscillator, state (r,s) keeps position and velocity; the readout may still ask only for r . Learning a state update and answering a readout are different jobs.

Trace: position $r=0$ with velocity $+\omega$ moves toward positive r ; the same position with velocity $-\omega$ moves toward negative r . Same current readout, different derivative.

The question this chapter answers

What state predicts motion while a readout answers a narrower question?

A point is not a trajectory

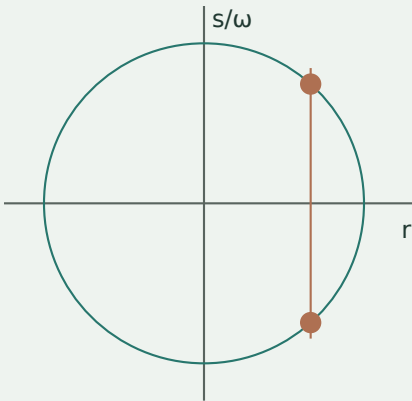
A state $x(t)$ describes the system at one instant. A trajectory is the path $t \mapsto x(t)$ through those states. Velocity tells us the direction and rate of change at an instant. The equation $dx/dt = b(x,a)$ assigns a velocity to each state, possibly depending on action a . That assignment is a vector field, not a decoder or, by itself, a learned representation.

For an oscillator, let r be position and s velocity. With frequency $\omega > 0$:

$$dr/dt = s \quad ds/dt = -\omega^2 r$$

Starting at $r(0) = 1$, $s(0) = 0$ gives $r(t) = \cos(\omega t)$, $s(t) = -\omega \sin(\omega t)$. The phase portrait is an ellipse in (r,s) coordinates and a circle in $(r,s/\omega)$ coordinates. Reading position only merges moments with different velocities. The same position can be moving left or right. Picture the oscillator passing the same point in opposite directions: what must the state retain to tell those moments apart?

POSITION IS A READOUT, NOT THE WHOLE STATE



Same position r .

Opposite velocity s . Different next positions.

$$r(t) = \cos(\omega t)$$

$$s(t) = -\omega \sin(\omega t)$$

Known nonzero ω ; ideal oscillator.

Figure 5. Analytic schematic, not a rerun. Position alone merges opposite directions of motion. The scaled state $(r, s/\omega)$ follows a circle.

The decoder answers a specified question

An encoder maps an observation or state to z . A decoder maps z to a requested output. In a generative model it may reconstruct pixels or parameters of a distribution; in our closure notation h reads the task output. “Decoder” in a Transformer also names an architectural stack. These meanings are related by data flow, not identical definitions.

A readout producing an image is not the transition law moving the state. A low reconstruction error does not alone establish that the latent state predicts actions and futures. Conversely, a good task predictor may deliberately lose visual details that the task does not need. VAE, Transformer, Dreamer and Neural ODE papers provide different examples of these roles. [R19–R23]

Skipped states can mean different things

Sampling records points at chosen times; the continuous trajectory can pass through intermediate states not recorded. A numerical step approximates motion between samples. A projection can merge different points. A physically unreachable point cannot be visited under the admitted dynamics. These are four different reasons for a picture to appear to “skip” something.

The retained oscillator demonstration ran Euler and RK4 for ten periods at three step sizes, comparing six trajectories with an analytic solution. Three deterministic assertions passed. At the middle resolution, final scaled-state error was approximately 3.67 for Euler and 3.04×10^{-6} for RK4. This is a numerical-method demonstration, not a learned world model. [P6]

Explicit Euler increases oscillator energy by a factor $1 + (\omega\Delta t)^2$ per step in exact arithmetic. Smaller steps reduce that error; they do not turn Euler into the exact flow. The earlier demo checked this identity to roundoff. No trajectories were recomputed for this book.

The important division is now visible: state describes what is, motion describes how it changes, and readout describes what we ask to see. Keeping these jobs separate lets us ask whether a new coordinate system simplifies the motion without changing what we can read from it.

Try it before reading the answer

For $\dot{r}=s$ and $\dot{s}=-\omega^2 r$, compare states $(0,+1)$ and $(0,-1)$. Is r alone closed? Is adding s sufficient for this specified ideal model?

Worked answer and limit

Their r derivatives are $+1$ and -1 , so position alone is not closed. The pair (r,s) supplies both derivatives under the stated equations. Sufficiency depends on the model: forcing, changing ω or damping would require their parameters or context too.

Carry this forward

Next, examine coordinate changes and spectral descriptions of that motion.

6 | Change Coordinates; Check What Survives

Start with a new coordinate is not automatically less information

Imagine writing the same circular motion as two real coordinates or one complex number. Nothing need be lost: $c = r - i s/\omega$ encodes both. Keeping only one Fourier coefficient is different—it discards part of a general signal. First ask whether a change is reversible, then ask whether the retained coordinates are closed under the dynamics.

Trace: $(r, s) \rightarrow c = r - i s/\omega \rightarrow \dot{c} = i\omega c \rightarrow$ recover $r = \text{Re}(c)$, $s = -\omega \text{Im}(c)$, for known nonzero ω . This is an analytic coordinate change, not learned compression.

The question this chapter answers

When is a transform invertible, truncated or closed?

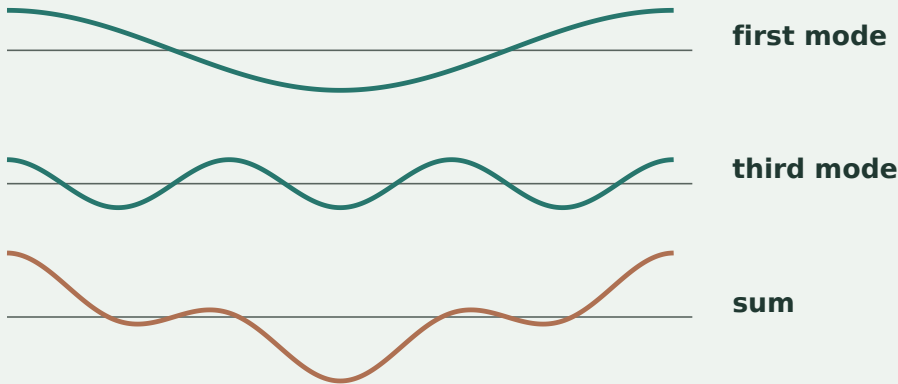
A wave can become a rotating number

Picture the oscillator as a point moving around an ellipse. Rescale its velocity coordinate, and the ellipse becomes a circle. A cosine and a sine of the same frequency can then be written as one rotating complex number. Fourier analysis uses oscillatory components to represent suitable functions. For the oscillator $\dot{r} = s$, $\dot{s} = -\omega^2 r$ with constant $\omega > 0$, define $c = r - is/\omega$. Then $dc/dt = i\omega c$. Instead of following two coupled real coordinates, we follow one complex rotation.

The dynamics were specified before we changed coordinates. The complex representation makes them easier to express; it does not discover their physical law.

$$\hat{f}(\omega) = \int f(t)e^{-i\omega t} dt$$

This convention uses angular frequency; inverse reconstruction has a $1/(2\pi)$ factor. Transform and inverse statements require an appropriate function space and convergence interpretation. Under suitable differentiability and decay, transformation sends f' to $i\omega f$. Periodic versions use Fourier series and periodic boundary conditions.

RETAINING ALL MODES $\neq$ DROPPING ONE

Analytic schematic; different amplitudes, not measured error.

Figure 6. Schematic superposition. Components add to a signal. Retaining all coefficients can change coordinates; dropping a component changes the retained information.

What a finite transform preserves

For N samples, the full discrete Fourier transform is an invertible linear change of coordinates when all N coefficients and the inverse convention are retained. Keeping only a few frequencies is compression and may lose information. A finite Fourier reconstruction is continuous as a function of time; that does not prove the sampled source was continuous or uniquely recover its behavior between samples.

Band-limited reconstruction needs sampling and signal assumptions. A sharp event, unobserved contact or missing sensor distinction does not become known because the retained numbers have been written as waves. Fourier cannot invert the information-losing sum of our two boxes unless other assumptions supply the missing separation.

Nonlinearity couples modes

If a motion includes a cubic term, cubing a cosine produces both its original frequency and a third harmonic:

$$\cos^3\theta = (3\cos\theta + \cos 3\theta)/4$$

A basis that diagonalizes one linear operator need not diagonalize a nonlinear dynamics. Modes can interact. Multiplication in one domain becomes convolution in the other under the relevant transform conventions.

Oscillatory coordinates are a candidate tool, not a certificate that every representation becomes independent continuous motion.

Koopman moves observables

For a discrete deterministic transition F , define $Kg = g \circ F$ on a suitable invariant space of observable functions. This operator is linear in g even when F is nonlinear. That trades nonlinear state dynamics for potentially infinite-dimensional linear observable dynamics. A finite chosen dictionary closes exactly only if its span is invariant. A fitted matrix can approximate its evolution without proving invariance. [R24]

For continuous-time dynamics, the corresponding generator acts as $Lg = \nabla g \cdot b$. The following example closes under that generator. A worked special case is $\dot{x} = \mu x$, $\dot{y} = \lambda(y - x^2)$. Retain $\psi = (x, y, x^2)$. Differentiation gives $\dot{x} = \mu x$, $\dot{y} = \lambda y - \lambda x^2$ and $(x^2) = 2\mu x^2$. This particular three-coordinate lifted system is linear and exact. It adds a coordinate; it is not automatically compression. For $\dot{x} = x^2$ on an unrestricted real domain, the dictionary $(x, \dots, x^n)$, $n \geq 1$, generates nx^{n+1} when its highest-degree member is differentiated and fails this closure test. That does not rule out every other transform.

The research question is which coordinates simplify the admitted dynamics while preserving the desired output at acceptable cost. Changing coordinates cannot supply missing observations. History sometimes can—but only if it contains the distinctions a snapshot leaves out.

Try it before reading the answer

A signal contains $\cos(t) + 0.3\cos(3t)$. If only its first harmonic is retained, can the original be reconstructed over this two-component family? Contrast retaining both coefficients.

Worked answer and limit

No: changing the third-harmonic coefficient leaves the first-harmonic representation unchanged. Retaining both coefficients preserves this specified family. That does not prove sufficiency for arbitrary signals or nonlinear dynamics that generate additional modes.

Carry this forward

Next, ask whether histories and context can supply missing distinctions.

7 | Remember, Condition, and Update Beliefs

Start with use another observation to separate possibilities

Return to the pendulum at its center. A second nearby position can reveal direction, provided the observations and timing are informative. Memory is useful because it can separate possible worlds—not because a longer transcript is automatically a sufficient state. A belief represents remaining uncertainty instead of silently choosing a world.

Trace: prior possibilities → action-conditioned prediction → observed evidence → normalized posterior. Context such as units and time spacing changes the meaning of the same number.

The question this chapter answers

When do memory and context provide information rather than just more text?

History can reveal what a snapshot hides

The current total does not tell us the left count. Earlier actions and measurements might. If they do, an observer can retain that information in memory. History helps because of the distinctions it reveals, not merely because it is long. Repeating the same uninformative measurement need not reveal the arrangement.

A fair comparison must say whether a predictor sees one observation, a fixed window or the full history. If one method receives a history and another receives only a snapshot, their comparison changes information access as well as architecture.

Under partial observation, a belief state records possible worlds and their probabilities. In a controlled hidden-state model, one schematic Bayesian update is:

$$b'(x') \propto p(o' | x') \sum_x p(x' | x, a) b(x)$$

It predicts possible successor states and reweights them by the new observation. Continuous states replace the sum with an integral.

Normalization, the transition law and the observation likelihood must be specified. A practical approximation may be much smaller than this full distribution, without carrying an exact guarantee.

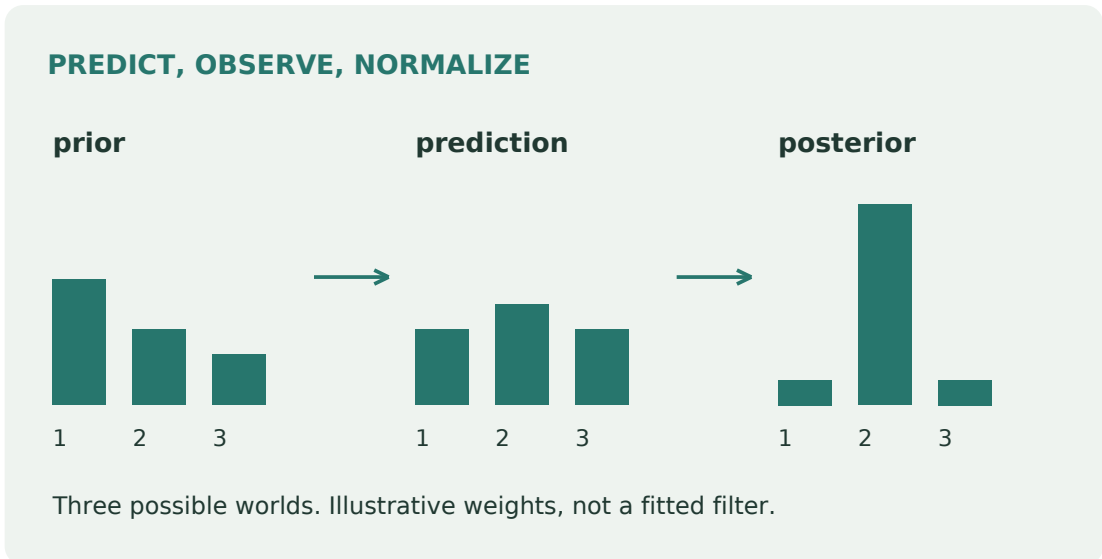


Figure 7. A belief update first predicts possible worlds and then reweights them using the observation. This schematic is not a fitted filter.

Context belongs in the model

An observation can be numerically identical while its meaning changes. Millimeters are not meters. A sensor may have moved. A coefficient may refer to an old version. A task may now ask about duration rather than endpoint. Context determines how the values should be interpreted.

The foundation implementation guarded shape, nonnegative counts, an expected context string, source ID and an age bound. These checks rejected declared malformed or stale inputs. The age bound was a teaching policy, not a calibrated scientific threshold. A supplied provenance label is not authenticated provenance. A guard cannot detect a world change nobody records. [P3]

In-context learning changes the computation

A model with fixed weights can produce different predictions after receiving examples in its context. The forward computation depends on those examples. That does not mean every weight has been updated, nor that the model has inferred the true law of an arbitrary world.

For the box problem, context might provide examples revealing the left count or the intervention rule. If every demonstration remains compatible with two hidden explanations, uncertainty survives. Results on Transformers learning function classes in context show a capability under specified distributions, not universal scientific identification. [R22]

A distribution is a different promise

Statistical sufficiency for a model parameter is not automatically predictive sufficiency for a future target. A strong predictive statement is $P(Y | H) = P(Y | Z)$ for $Z = \varphi(H)$, under a specified law. Equality of conditional means is weaker: it preserves a squared-loss average, not every tail event or multimodal outcome.

In controlled systems, prediction under one behavior policy need not preserve counterfactual effects of other actions. Predictive-state representations organize information through action-conditioned future tests; their existence results do not guarantee that finite observations learn an optimal representation of every changing world. [R2, R3, R6]

The safe reading of “the model understands the context” is operational: the declared context changes a specified calculation. It does not mean every hidden fact is available. Once we put the calculation into a program, we need a further check: does its execution preserve the mathematics?

Try it before reading the answer

Two possible states have prior probabilities 0.5 each. An observation has likelihood 0.8 in the first and 0.2 in the second. Compute the posterior. Would an observation with likelihood 0.5 in both help?

Worked answer and limit

Posterior weights are $0.4/(0.4+0.1)=0.8$ and 0.2. Equal likelihoods keep the posterior at 0.5 each: that observation does not distinguish the states. These specified probabilities are a teaching example, not a calibrated deployed filter.

Carry this forward

Next, check how a mathematical promise survives implementation.

8 | Separate a Proof from a Working Program

Start with test the meaning of “the same calculation”

With exact arithmetic, $(a+b)+c$ and $a+(b+c)$ agree. On a floating-point machine, take $a=1$, $b=2^{53}$, $c=-2^{53}$: rounding can make the first path 0 and the second 1. Algebra, executable agreement and evidence about the world are three layers. A test at one layer does not certify the others.

Trace: exact identity $\rightarrow$ chosen numeric representation $\rightarrow$ finite test suite $\rightarrow$ task validation. State which layer each claim belongs to.

The question this chapter answers

What do a theorem, a runtime test and external validity each establish?

Agreement has more than one layer

A mathematical rule, an executable program and a physical system are different objects. A proof about the mathematical rule does not, by itself, certify its implementation or its physical measurements. The exact foundation compared a step-by-step interpreter with composed matrices over 85 action sequences of length zero through three on 121 starting states: 10,285 agreements. It also passed 31 unit tests. [P3]

The tests establish agreement on the cases checked. The specified linear identities support the all-state argument. Increasing the test count does not turn the former into the latter. Exact integer or rational arithmetic can remove one source of numerical uncertainty without making the world's measurements exact.

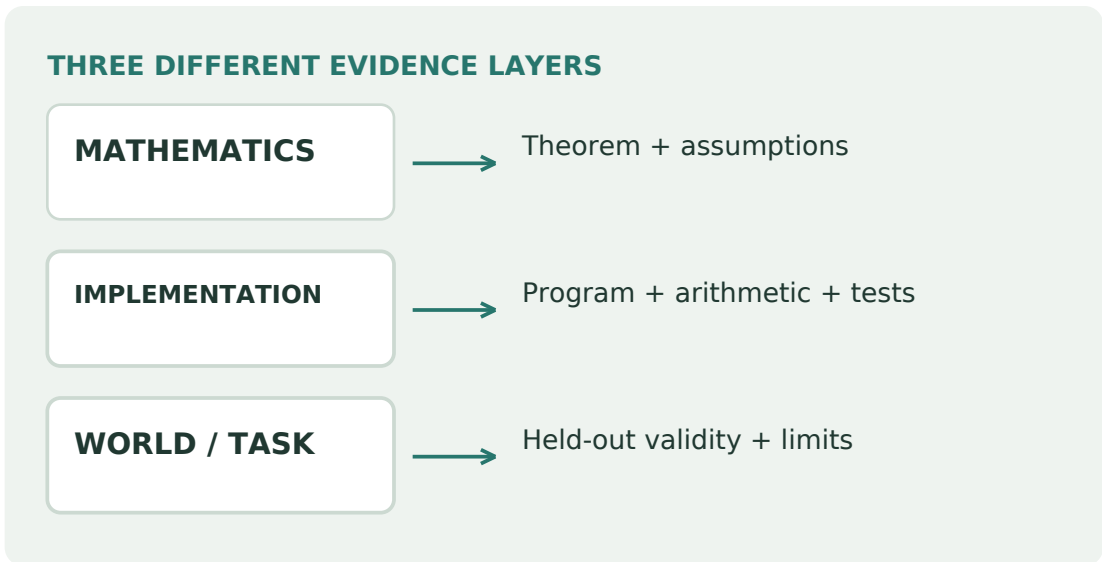


Figure 8. Three evidence layers. A mathematical theorem, runtime agreement and external validity require different evidence; none automatically certifies the next.

What the formal artifacts established

The retained Lean reports include a necessary collision obstruction and fixed-sequence commutation given one-step premises. A separate affine supplement covers exact integer identities, composition and compilation of lists of supplied actions. Their proofs do not discover a representation, construct every possible reduced rule, establish stochastic closure or certify a learned world model. They were reviewed in retained sources, not rerun for this edition. [P4, P7]

The Bend source check, JavaScript compilation and fixture execution were reported as working engineering steps. Independent optional BendTT/kernel validation remained unresolved. The working compiler path and the unresolved kernel check are different claims; the former does not establish the latter.

Associativity can fail in the executable numbers

In binary64 floating arithmetic, rounding changes $(1 + 2^{53}) - 2^{53}$ into zero, while $1 + (2^{53} - 2^{53})$ is one. The exact integer expressions are equal; the evaluated floating expressions are not. The retained F32 fixture gives the analogous issue at 2^{24} .

Thus algebraic simplification may change numerical behavior. A compressed execution can agree to a measured tolerance rather than

exactly. Its error needs a bound or a protocol, not a slogan about identical mathematics.

A derivative condition needs a uniqueness condition

For smooth φ and full dynamics $\dot{x} = b(x)$, a proposed latent field V must satisfy $D\varphi(x)b(x) = V(\varphi(x))$. This expresses consistent projected velocity. It does not alone guarantee that the latent initial-value problem has a unique solution.

Take $\dot{x} = 1$ and $\varphi(x) = x^3$. Then $\dot{z} = 3|z|^{2/3}$. From $z(0) = 0$, the projected full trajectory is $z(t) = t^3$. The latent equation also permits waiting at zero until any $c \geq 0$ and then following $(t - c)^3$. Its field is not locally Lipschitz at zero. The full trajectory and its actual projected flow remain uniquely specified here; the latent differential equation alone admits additional solutions. The projected velocity relation alone did not select a unique latent flow. Suitable regularity, a specified solution class or the actual projected flow supplies the missing structure. This separates agreement of formulas from agreement of behavior—a distinction we will also need when evaluating learned rules. [P6]

A CLOSED DERIVATIVE NEED NOT PICK ONE FLOW

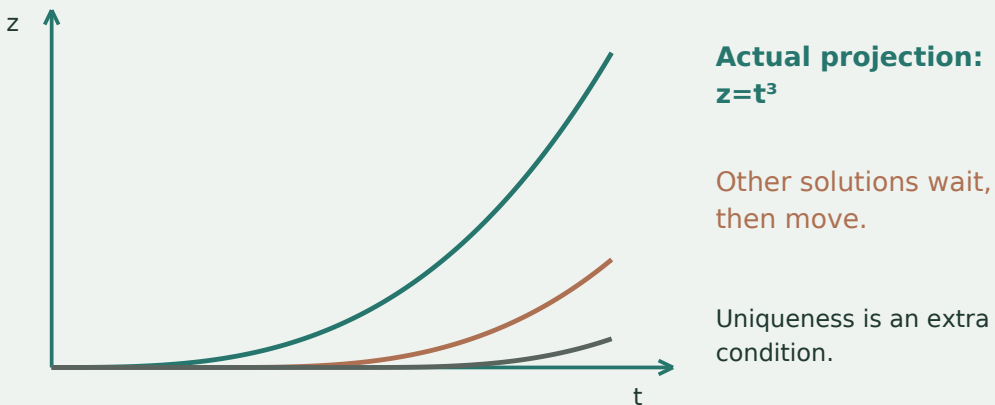


Figure 9. Exact constructed solutions of the latent equation. The actual projected full flow is t^3 ; the differential equation also permits arbitrary waiting at zero. These curves are not numerical runs.

What no general procedure can promise

For effectively specified computable total program-state transitions, an output flag can remain zero until a simulated program halts. Deciding whether it agrees forever with an always-zero output would decide nonhalting. A universal all-program equivalence decider is therefore unavailable. Finite horizons, explicit finite systems and restricted algebraic families are practical escape routes. This is a prose reduction, not a new machine-checked theorem. [P5, R14]

Try it before reading the answer

A program agrees with a theorem on 121 enumerated states. What follows for those tested states, and what does not follow for unseen states or a real robot?

Worked answer and limit

The observed implementation agrees on those cases under the tested arithmetic and comparison rule. Universal correctness needs a proof covering the implementation and domain; robot validity additionally needs suitable sensing, dynamics and safety evidence. A finite pass count alone provides neither.

Carry this forward

Next, examine what the rule-learning experiments actually paid for and found.

9 | Discover Rules—and Count the Search

Start with a small formula can be expensive to discover

Suppose one candidate formula uses just a few interaction terms and another uses a dense library. A short final rule is not evidence of a cheap discovery process: the search may fit many candidates, and a physics baseline may start with valuable supplied knowledge. Compare what each method sees, tries and pays before comparing its final error.

Trace: observations → candidate library → fitting and selection → frozen rule → held-out rollout. Count selection work as well as the final executable rule.

The question this chapter answers

What was fitted, selected and paid for in the retained rule-learning studies?

A compact formula can carry a lot of work

Consider coupled particles whose acceleration depends on position, velocity, fields and interactions. We can give a learner a library of candidate terms, let it fit their coefficients and retain a small active subset. The fitted rule then chooses from that library. This is not the same as supplying the correct physical terms in advance.

The retained interaction study used a 105-term library and thresholded ridge fitting. In a clean interpolation condition, sparse-rule rollout RMSE was approximately 0.00012015, compared with 0.038743 for the tested generic dense fit. It nearly matched the hand-selected physics model in that setting. [P8]

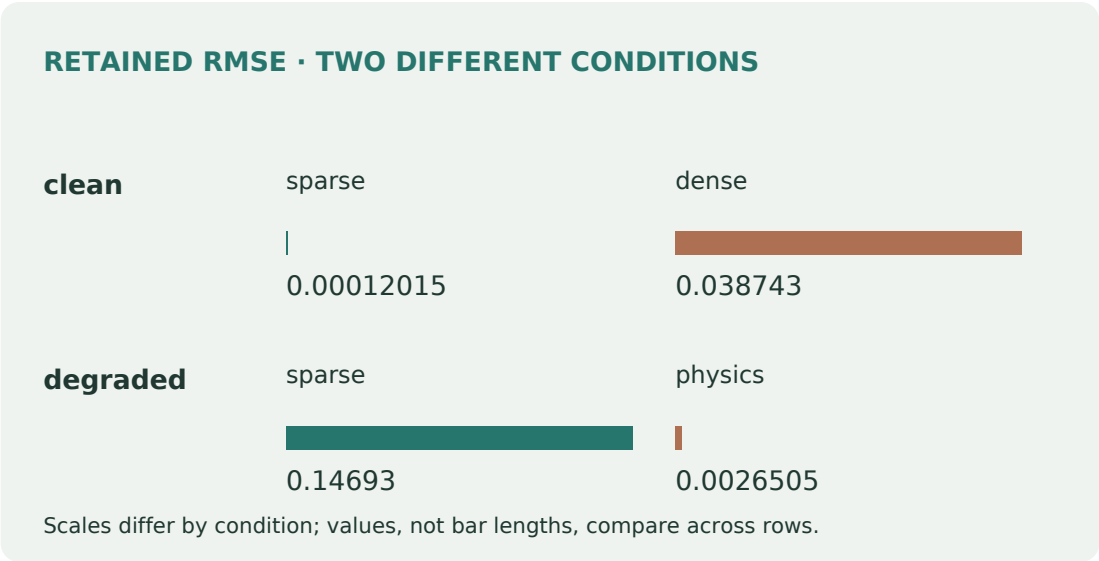


Figure 10. Retained result values. Clean interpolation and jointly degraded extrapolation are different conditions, not one pooled benchmark. Lower RMSE is better.

The interesting part is the reversal. Under the jointly noisy, sparsely observed extrapolation condition, sparse RMSE was approximately 0.14693 while the physics-selected model achieved 0.0026505. The strong prior helped. It was not learned from scratch, and its supplied family makes it a different comparator.

A fair bill includes the search

The study retained 160 fits, twenty independent training seeds and 10,240 model trajectory evaluations across its conditions. Sparse fitting searched twelve settings while the cited baseline search used three. The resulting accuracy comparison is not an equal-learning-budget contest.

A separate execution comparison held the learned rule fixed and changed how it was evaluated. Compact execution was faster in the reported local CPU timings, with median speed ratios about 3.20 in clean and 2.35 in degraded conditions; maximum discrepancy was around 1.23×10^{-13} . This comparison supports faster execution of the fixed learned rule in the reported setting. Learning that rule has a separate search cost, so the timing result does not establish that sparse learning is universally faster.

The dense generic model had 163 divergent rollouts assigned penalty fifty. A method’s failure policy materially affects averages. Nonfinite trajectories cannot simply disappear from the report, but the chosen penalty is not a physical error unit.

Differentiation amplifies sensing noise

Estimating velocity or acceleration by differencing nearby noisy measurements can be unstable. Roughly, central derivative noise grows as measurement noise divided by time spacing. An integral-window fit avoids explicit point differentiation but averages over a wider temporal support and changes the regression design.

A retained scalar comparison ran 360 fits and 7,200 held-out rollouts across nine noise/spacing cells. In its finest-spacing, highest-noise cell, failures fell from 80/400 to 13/400, with penalized-RMSE gain 8.4032 and ordinary paired interval [1.9095,14.8968]. Five of six noisy-cell intervals crossed zero. A descriptive decomposition assigned 8.375 of the highlighted gain to failure penalties. That is 99.66% of the mean contrast, not a causal attribution proving why the algorithms differ. [P4]

The conclusion is narrower than “integral methods solve noise”. Averaging support, overlapping rows, wider held-out support and failure accounting all changed the comparison. A stronger experiment would match the conditions it can and report the differences it cannot remove. Even a useful rule will make mistakes. The next question is whether a system can identify where its own predictions are less reliable.

What the model actually learned

It learned coefficients and selected support inside supplied families. It did not learn every measurement convention, simulator law, detector threshold or search strategy. Compactness is a property of the resulting program. Sufficiency is a promise about its future use. One does not establish the other.

A newer affine comparison: one budget is not every budget

The later matched-observation study compares one direct affine fit with composed primitive fits under 120 transition records and 720 scalar supervised slots each. Across six cells, direct prediction has 3.7%–9.1% lower average latent RMSE, with six-family simultaneous paired intervals supporting the contrast. Both use a common PCA calibration, which is additional shared access rather than part of that supervised-slot budget. Composition has more fitted parameters and fits; equal counts do not establish equal information or compute.

An exploratory same-trajectory control slightly favored composition in one low-noise lossy cell. A subsequent 32-fresh-seed study reused intermediate observation noise: all three low-noise reused contrasts were inconclusive, while high-noise contrasts favored direct prediction. Five of six shifts favored composition; the sixth shift was inconclusive. One prespecified covariance audit failed (-0.07048 against an absolute threshold of 0.06); no seeds or tolerances were changed. Appendix E retains the complete six-cell summaries and the failed check. These results concern sensitivity to observation design, not a universal winner.

Try it before reading the answer

A direct affine fit uses six coefficients; a two-stage chain uses two six-coefficient fits. Both receive 120 transition records. Does matching record counts match model capacity, information or compute?

Worked answer and limit

No. The chain has 12 fitted coefficients and two fits. Record counts do not specify distinct sensor readings, correlations, intermediate access or tuning. Report each budget separately; equal counts are one controlled axis, not a complete fairness claim.

Carry this forward

Next, ask when the learned predictor should be trusted or withheld.

10 | Know Where a Model Is Unreliable

Start with familiarity must predict a failure boundary

A model recognizes a familiar-looking input but gives a bad answer. Visual or geometric resemblance is not enough. A useful familiarity map should predict where frozen transformations work, where uncertainty is high, and where a new observation is needed. That is a measurable operational claim, not evidence of consciousness.

Trace: seeded primitive examples → fixed skill library → unseen composition → prediction and uncertainty → held-out error. Separate new inputs, new parameters, new order and new family.

The question this chapter answers

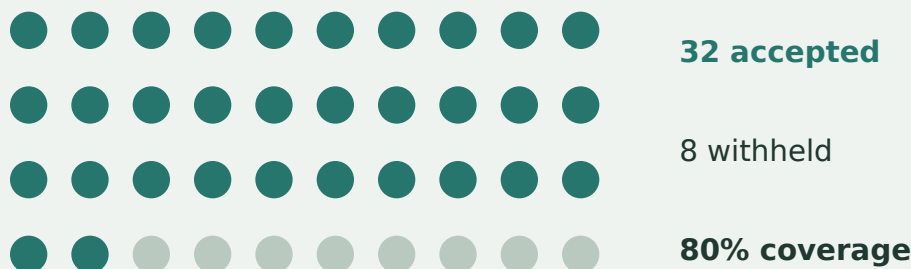
What would make familiarity an honest prediction of reliability?

Predicting error is a task too

A world model predicts an event. An error monitor makes a second prediction: where is that model likely to be wrong? In the retained study, descriptors included ensemble spread, distance from observed support, predicted motion and initial velocity. A fitted regression used those descriptors to rank candidate trajectories. [P9]

The monitor accepted 32 of forty trajectories: 80% coverage. It did not improve every underlying prediction. All selection policies shared the same numerical predictor. Withholding harder cases can lower the error on accepted cases without improving the predictor. Report coverage beside that error so the reader can see the trade.

SELECTIVE ERROR NEEDS ITS DENOMINATOR



Forty predictions exist. Selection changes which are acted on.

Figure 11. Schematic forty-case cohort at 80% coverage: thirty-two accepted, eight withheld. Lower selective error is not full-coverage improvement.

After a common model update, mean accepted trajectory RMSE was 0.004732855 for the monitor versus 0.005014883 for random selection within range. Under shifted field/coupling it was 0.01088819 versus 0.01140392. Paired twenty-seed intervals favored the monitor over random selection in those comparisons.

The stronger baseline changed the interpretation: monitor-versus-ensemble-spread intervals crossed zero in every reported contrast. Incremental benefit beyond that cheaper uncertainty heuristic was not established. An inconclusive interval is not proof of equality, but it cannot support the claimed superiority either.

An index makes a calculation reproducible

The implementation retained 680 stable keys linking model versions, fitted values, support, inputs, predictions, decisions and evaluator errors. Replaying eighty cohort predictions and monitor scores reproduced the stored values exactly. Recorded selected IDs agreed. Deliberately corrupted or stale references were rejected.

Those are useful engineering properties. An index can say which model produced a decision and reconstruct the calculation. It cannot recover observations that were never retained, authenticate every source declaration or establish that the prediction was true. Replay fidelity and predictive validity are independent claims.

An operational self-model is not consciousness

The modest meaning of self-representation here is a stored, executable description of the learner's predictive reliability that affects acceptance decisions. That is already a useful computational object. Nothing in the study measures a subjective point of view or recursive general self-awareness.

The later event-monitor implementation had guarded windows, recorded action states and fresh-window processing. Its warnings remained heuristics. It did not introduce a continuous autonomous daemon, guaranteed drift detection or automatic safe retraining. Engineering completion did not resolve the earlier monitor-versus-spread uncertainty. [P4, P9]

What would earn the stronger claim

A protected new regime, a fixed total budget and a predeclared useful improvement over spread-only acceptance would make a meaningful test. Refreshed, stale and shuffled monitors should share the same predictor and coverage. If the uncertainty interval does not clear the required improvement, the claim of incremental value should be rejected or left unresolved.

A system can make its uncertainty visible without claiming that uncertainty is calibrated. An ensemble's disagreement is a signal to test, not automatically a probability of failure. Once we have tested such a signal, we can ask whether it helps choose when to observe more. That choice has a cost, and the cost arrives over time. [R18]

Seeding a learner: a proposed protocol, not a result

A random-number seed controls a simulated run; a seeded skill library is a different idea. The proposal is to teach a small set of primitive transformations, freeze that library, and test compositions never shown during fitting. Familiarity is a vector of measured axes—state-domain support, parameter support, operation-family support, order/depth support, calibration and task-relevant information retention—not a single resemblance score.

Hold out complete sequences as well as fresh states and random seeds. Compare composed primitives with a direct learner given the same declared access, an unfitted baseline and an oracle only where its privileged

knowledge is explicitly labeled. Report prediction error by holdout type and depth, coverage versus selective error, interval calibration, acquisition cost and failure on noninjective recovery requests. Fit familiarity thresholds on validation data only; assess whether those scores predict error on the final test. Low error on a new state inside an old sequence is not evidence of unseen-sequence transfer. We have not executed this protocol.

Try it before reading the answer

A selector accepts 32 of 40 predictions and reports lower error on those 32. What two quantities must be reported? What test would support a seeded skill transfer claim?

Worked answer and limit

Report selective error and coverage ($32/40=80\%$), plus the full-coverage comparison. For transfer, freeze primitives and thresholds, withhold complete operation sequences, then evaluate new seeds and states without fitting to the withheld outputs. No such generalization result is established here.

Carry this forward

Next, include the cost and timing of the information a model requests.

11 | Observe More, Pay More, Update Earlier

Start with the timing of information changes its value

A useful sensor reading arrives too late to help the current decision. In the retained protocol, 32 examples detect a change and a 64-example scan follows: 96 examples pass before the full scan is available. Updating after the first eight scan examples changes response timing, not the scan's final size.

Trace: regime change → 32-example detection → first 8 scan examples → early refit → continue to all 64 scan examples. Keep latency, total measurements and update count distinct.

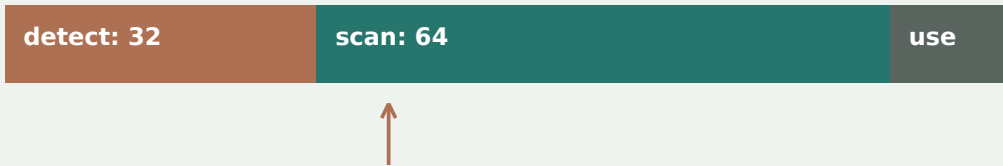
The question this chapter answers

How do observation cost, latency and refit timing affect comparison?

Information arrives on a schedule

A compact online learner senses a few coordinates. After residual error rises, it scans more coordinates and refits. The aim is to use little information until the world changes. But the learner needs time to detect that change, collect observations and fit a new rule. Its mistakes during that delay belong in the result.

In the first retained sensing study, a change detector needed thirty-two examples and the diagnostic scan needed sixty-four more. A ninety-six-example delay occupies half of a regime lasting 192 examples. Good predictions after recovery do not undo the mistakes made while waiting. [P10]

TIME TO INFORMATION $\neq$ TOTAL SCAN COST

First 8 scan examples: an early refit can start here.

Full detection + scan: 96 examples. The scan still ends at 64.

Figure 12. Protocol timing schematic. A 32-example detection delay plus a 64-example scan consumes 96 examples; early updates can start before the scan ends.

At sixteen dimensions with sparse changes and output noise 0.1, the compact adaptive policy used about 26.2% of the full sensor scalars, but postchange MSE was about 0.430 versus 0.277 for the tested full-observation policy. Less sensing bought more error. The study did not establish readiness to replace full observation.

Use information before the scan ends

The follow-up fitted useful prefixes after eight or sixteen diagnostic examples instead of waiting for sixty-four. This attacked latency rather than inventing faster access to the entire scan. All scans still acquired sixty-four examples. Updating earlier is not the same as sensing fewer.

The incremental study retained 400 streams, 2,400 learner runs and 1,520,640 scored-before-feedback predictions. Early updating improved some sparse cases. Dense changes still defeated the narrow selected representation. The preserved reports separate native detectors and shared-controller analyses. [P11]

Match recovery before crediting compression

In the matched-recovery study, full-state learners received the same trigger events and early refit opportunities. In the sixteen-dimensional, low-noise sparse cell, selective update8 achieved postchange MSE 0.2225 and full reset8 achieved 0.2298. Their difference was -0.0073 , with paired interval $[-0.0164, +0.0018]$. This did not establish postchange superiority, equivalence or noninferiority.

For the specified first-sixty-four-after-change comparison, the reported interval lay below zero: -0.0360 $[-0.0627, -0.0093]$. Full reset8 improved on full reset64 by 0.1087 $[0.0946, 0.1227]$. Earlier useful recovery explained substantial gain. These are ordinary paired Student-t 95% intervals over twenty independent seeds per cell, unadjusted for multiple comparisons. Prediction rows are workload, not independent replicates. [P12]

At sixty-four dimensions, a selective advantage remained against the tested small-prefix full ridge fit. A full-window control improved that comparator, and high-dimensional fits on tiny prefixes were weak. The finding does not overturn the optimum obtainable with full information. All four dense-change cells favored richer methods over selective recovery.

More information cannot worsen the free optimum

Let H be available information and $Z = s(H)$ a measurable summary, so $\sigma(Z)$ is contained in $\sigma(H)$. Define $m(H) = E[Y|H]$ and $mZ = E[Y|Z]$. For square-integrable Y and unrestricted squared-loss prediction:

$$R^*(Z) - R^*(H) = E[(m(H) - mZ)^2] \geq 0$$

The unrestricted full-information predictor can imitate any summary-based predictor. A finite trained full model can nevertheless lose through variance, regularization, model mismatch, poor updates or compute limits. Those practical losses are important; they are not proof that removal creates information. [P5]

Equal ceilings are not equal actual spending. Shared trigger access is a mechanism-analysis privilege, not automatically a deployable independent algorithm. To assess the trade, we need the accuracy over time and the complete bill. We also need to know what caused the failure: a stale rule, or information the representation never kept.

Try it before reading the answer

A sensor costs one unit per reading. Two learners both finish a 64-reading scan, but one updates after 8 readings. Which has lower sensor cost? Which may reduce delay?

Worked answer and limit

Their scan costs are equal at 64 units. The early learner may reduce time before using information, but may also fit a noisier estimate or pay more refit work. Only measured errors and a complete cost account decide the net advantage.

Carry this forward

Next, decide whether to repair the fitted rule or change what is observed.

12 | Refit the Rule or Change the Observation?

Start with repair the right failure

Your scale has changed from kilograms to pounds, or your total conceals which box will be doubled. Both can cause a residual, but the repairs differ. Refit a stale rule when the observation is sufficient; acquire a missing distinction when the observation cannot support the required answer. A confident diagnostic label cannot replace access to evidence.

Trace: error appears → test competing causes → estimate value of another observation → choose refit, probe or uncertainty. The cause is not determined by the residual alone.

The question this chapter answers

Which evidence separates stale rules from missing information?

Two causes can look like one residual

A predictor used to work and now fails. Return to the boxes. If both used to double and now both triple, the total is still enough: we need to replace $2z$ with $3z$. The rule is stale. If the action instead doubles only the left, the total is no longer enough. No refit of a deterministic function of total alone can predict the exact result for every arrangement. The problem is a missing distinction.

The two cases isolate useful differences. They are not a complete account of real failures, which can also involve noise, optimization and finite samples.

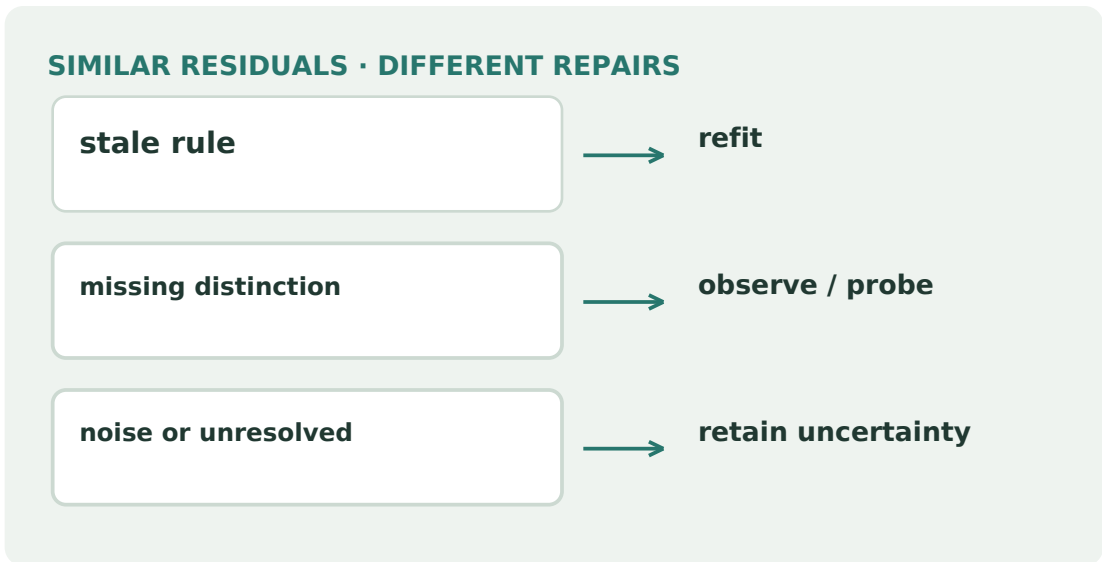


Figure 13. Different failure explanations require different responses. A residual alone need not distinguish stale mapping, missing information and irreducible noise.

Diagnosis needs access, not a confident label

If two hidden causes produce the same distribution of visible histories under every permitted adaptive probing policy, no diagnostic function of those histories can distinguish them. With equal priors, expected binary accuracy is at most one half. At a fixed policy and horizon, optimal equal-prior accuracy is $(1 + TV)/2$, where TV is total variation distance between the history laws.

The statement concerns distributional equality under declared access. A new sensor or intervention can break that equality. A prior can change the decision. Neither should be smuggled into an apparently information-matched comparison. [P2, P5]

For independent standard Gaussian x_2 and u , the targets $x_1 + x_2 + \text{small noise}$ and $x_1 + u + \text{small noise}$ have the same compact-view conditional law when only x_1 is observed and the hidden variables are identically distributed. Observing x_2 can distinguish their predictive relation. A residual alarm alone cannot tell us which explanation generated the unseen term.

What the retained diagnosis experiment found

The bounded diagnosis study compared eight policies over two hundred paired streams: fifty seeds in each of four regimes. It retained 1,600 policy runs and 819,200 scored-before-feedback predictions. The selector refitted on all fifty stale-fit streams and acquired the additional sensor on all fifty planted missing-information streams. Those observations describe the constructed conditions, not a universal diagnosis guarantee. [P13]

In the independent-noise null, it acquired unnecessarily on nine of fifty streams. Its online MSE exceeded the fixed compact predictor by 0.01113, with paired interval [0.00381, 0.01845]. Rolling compact performed better in the stale-fit condition; rolling full performed better in the missing-information condition. Waiting for a diagnosis had a price.

The later calibration study continued this generator, not the separate sixteen/sixty-four-coordinate latency study. Similar percentages do not make them the same experiment.

What remains a proposal

The working note proposes a benchmark contract: observations, actions, change distributions, train/test split, horizon, costs and baselines fixed before outcomes. The executed pilots test narrower procedures; they do not complete the benchmark or establish exclusive causal diagnosis.

A useful selector must beat strong refit-only, periodic, random and uncertainty-triggered acquisition methods under matched probe access and charged overhead. If it does not, that is a negative finding to retain. Before choosing a new model, we must also account for a simpler hazard: a candidate can win because the selection data were lucky. [R7-R10]

Try it before reading the answer

In world A, $y=2x$; in world B, y is an independent hidden variable with the same marginal distribution. A learner sees only y values without x . Can it tell the worlds apart from this observation stream alone?

Worked answer and limit

Not when the observed-history laws are identical. Observing paired x and y may distinguish correlation, but that changes access. More computation on an indistinguishable record cannot establish which world generated it.

Carry this forward

Next, prevent selection itself from manufacturing apparent evidence.

13 | Choose Without Fooling Yourself

Start with a winner needs a fresh test

Select the best of many noisy candidates and its validation score may overstate its value. Freeze the selection rule and test on fresh cases. Calibration asks how often the rule acts under specified null conditions; it does not turn a pilot count into a guarantee for every future distribution.

Trace: exploratory candidates → frozen threshold on separate calibration data → fresh evaluation → action count and uncertainty → deployment cost. Do not tune using the last step.

The question this chapter answers

How does selection bias enter, and what does calibration actually measure?

A winner can be a lucky winner

Compare candidate models on a small validation set and choose the one with lower loss. Some candidates win partly through luck. Once we select the winner, its apparent gain can overstate what we will see on fresh data. Repeated selection can increase this optimism. This is an established statistical effect; the pilot below illustrates it. [R17]

The retained calibration study allowed one residual alarm and one forty-eight-example paid probe. It fitted compact and full ordinary least-squares candidates on thirty-two probe examples and selected on the next sixteen. A chosen candidate was refitted on all forty-eight and affected predictions only afterward. [P14]

Freeze the margin before evaluation

Calibration used two hundred seeds in each of two null conditions. It compared a forced fixed window with alarm-conditioned probe windows. For each mode, the margin was the maximum of two empirical 95th percentiles, bounded below by the inherited 0.02 margin. The frozen margins were 0.1144 and 0.1193. These empirical quantiles set thresholds; they do not certify a population error bound. Evaluation used one hundred fresh paired seeds in each of three conditions and five policies: 1,500 runs, 768,000 scored-before-feedback predictions.

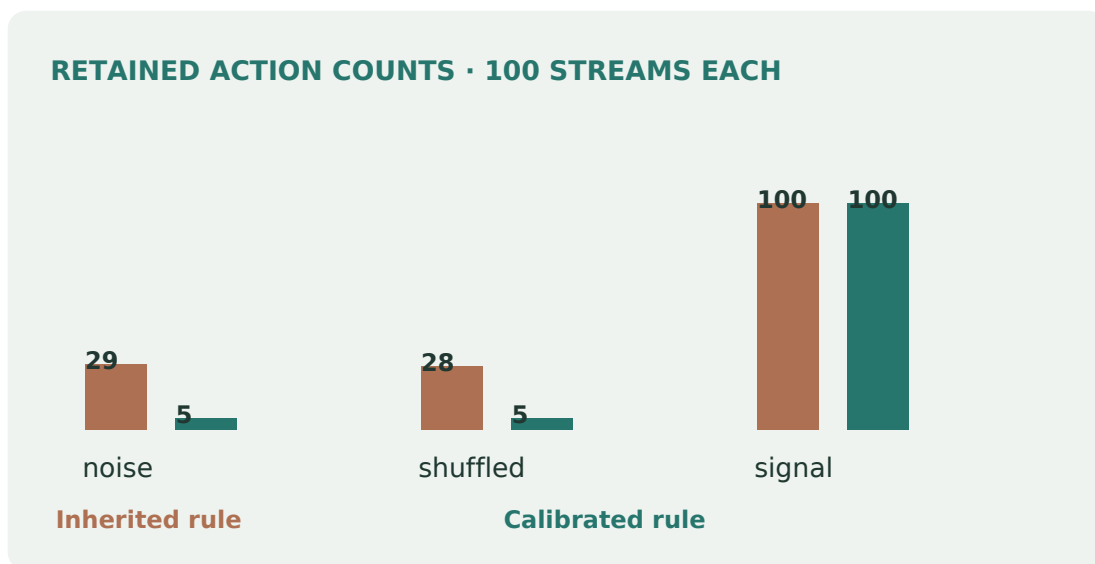


Figure 14. Retained deployment counts, 100 fresh seeds per condition. Both calibrated selectors had the same counts. Bars are descriptive; intervals are in the text.

Independent-noise full-model deployments fell from 29/100 to 5/100; shuffled-null deployments fell from 28/100 to 5/100. Every selector deployed the full model on all one hundred strong planted-signal streams. Fixed-window and alarm-conditioned methods were nearly indistinguishable here. There was no demonstrated extra benefit of conditioning on the alarm.

Five percent is a count, not a guarantee

A five-in-one-hundred estimate has a 95% Wilson interval of approximately [2.15%,11.18%]. It is not proof of a universal 5% false-acquisition bound. One hundred successes have a lower Wilson bound near 96.3%, not proof of perfect detection outside these draws. The signal amplitude was strong and no weak-signal sweep was performed.

Among the five alarm-calibrated noise selections, average compact-minus-full validation gain was $+0.1732$. On the subsequent fresh evaluator-only diagnostic it was -0.2143 . These selected subsets are small; their averages describe the retained cases but are unstable. The shadow diagnostic scored the original thirty-two-row candidates, not the forty-eight-row deployed refits; it never influenced policy decisions. Actual online MSE separately scored the deployed models.

The shuffled null preserved targets and the sensor multiset while permuting the sensor column across the finite stream. It created dependence and was an approximate additional null, not an exact conditional-randomization guarantee.

Keep the losing result and the invoice

Signal-condition online MSE was 0.14988 for the selectors versus 0.08996 for rolling full. The difference, $+0.05992$ [0.05578,0.06406], favored rolling full. Discovery latency remained even after calibration improved false deployments.

Every run had a 24,000-unit abstract-work ceiling. Both calibrated policies paid 7,014.11 units per run for conservative amortization of the combined calibration work over three hundred planned streams. Maximum realized work was 23,261. Rolling full had eleven denied refits per stream. The amortized calibration charge and maximum realized work use the same declared accounting basis, including the charged calibration where applicable. These are proxies and reuse assumptions, not dollars, instructions, energy or an optimized economic allocation.

Nineteen recorded audit checks passed, including frozen hashes, disjoint seeds, noninterference, accounting and deterministic replay. They are internal engineering evidence. Ordinary paired intervals are not familywise or time-uniform guarantees. Calibration reduced a specified failure rate; it did not remove selection optimism or beat the information-rich baseline. A system that uses these predictions to act must carry those limits into its decisions. [R17, R25, R26]

Try it before reading the answer

A frozen selector acts in 5 of 100 null streams. Is its true false-action probability known to be at most 5%? What should be reported?

Worked answer and limit

No. The observed rate is 5%; the retained Wilson 95% interval is approximately 2.15%–11.18%. Report the count, interval, null construction and test conditions. A distribution shift or repeated selection can change the operating rate.

Carry this forward

Next, connect prediction to skills, returns, duration and safety.

14 | Turn Predictions into Accountable Actions

Start with an endpoint is not the whole skill

Move every counter from the left box to the right. Both (4,6) and (7,3) end at (0,10), but take four and seven one-counter moves. The total predicts the endpoint; it does not predict duration or reward –u. An agent needs the distinctions that matter for action, cost and safety.

Trace: start state → skill policy → repeated actions → termination → endpoint, elapsed steps and reward. For an option, the endpoint is only one part of its outcome.

The question this chapter answers

Which distinctions must an agent retain for rewards, duration and safe actions?

Prediction becomes a decision

A model predicts what may happen. A policy chooses an action. A reward states the preference used for a particular task, and a value estimates the reward accumulated under a policy. These definitions let us analyze a decision system. They do not make its reward a complete account of human value.

$$V\pi(x) = E\pi[\sum_t \gamma^t r_{t+1} \mid x_0 = x], 0 \leq \gamma < 1$$

The expectation must be well-defined. Bounded rewards are one sufficient condition for finite value when $\gamma < 1$; finite horizons still require appropriate reward integrability. Additional assumptions can support other discount choices. A temporally extended action, or option, has an initiation set, a policy and a termination condition. Its planning model needs consequences, accumulated reward and relevant timing or discount information, not merely the final picture. [P3, R13]

A skill can preserve an endpoint and lose its cost

Our hand-written option transfers one counter at a time from left to right until the left is empty. It ends at $(0, u + v)$, takes u steps and returns $-u$ with reward -1 per step and no discount over its finite execution. All retained starting states passed these exact checks.

States $(0, 10)$ and $(5, 5)$ have the same total and final total, but durations zero and five. Total is sufficient for the endpoint total, not for planning the time or cost. Storing $(\text{total}, \text{left})$ resolves the ambiguity, but no longer reduces the number of coordinates of this toy.

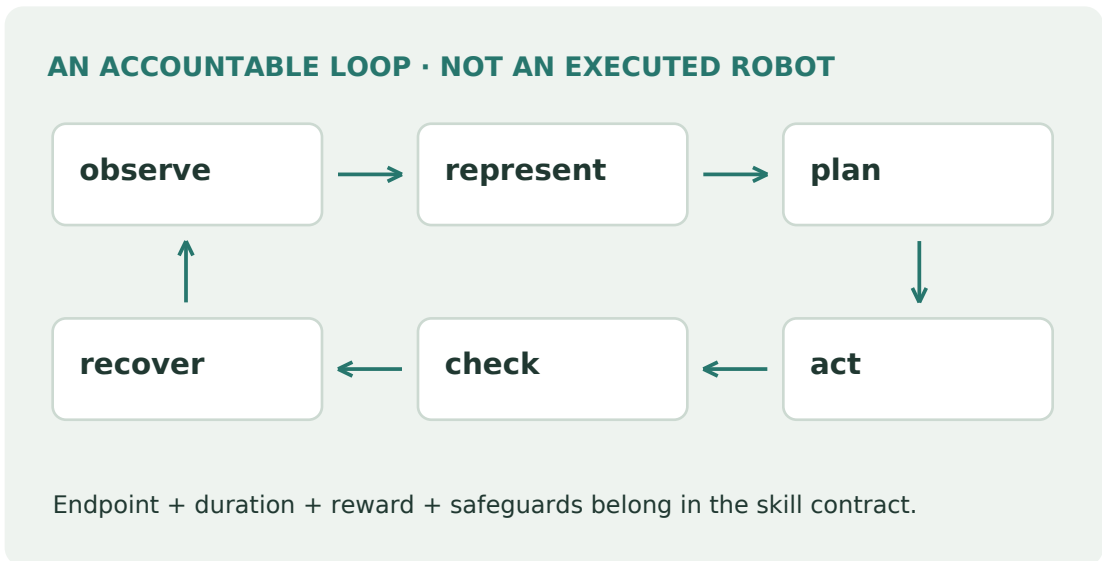


Figure 15. A proposed agent loop. The option must model timing and cost as well as endpoint; feedback and safeguards are part of the contract. Not an executed robot system.

The OaK direction links feature construction, subtasks, options, option models and planning. Our foundation supplied the feature, subtask and option. It did not execute a continually learning complete architecture, learned option discovery or a meta-learned step-size system. The missing components and tests are work still to do, not capabilities already demonstrated. [P3, R13]

Physical agents sharpen the contract

Shubham reports that Phi9 has robotic arms and has trained some compression-based policies. This edition has not inspected checkpoints, datasets, controller code, force limits or task-performance logs. This is an attributed, unverified project report. The supplied evidence does not verify the hardware configuration, trained policies or task capability.

A proposed first demonstration should use supervised, nonclinical tabletop tasks with safe props: sort objects into marked trays, move a blunt training tool, and operate a zipper pouch fixed to a jig. These are proposals, not tasks executed for this book. A human supervisor, emergency stop, limited workspace, conservative speed/force limits and a defined fallback belong in any actual protocol.

The medical-workflow motivation is assistance with routine manipulation around a clinician's work. This is not permission to operate near a patient, handle contaminated sharps or claim surgical autonomy, sterility or clinical validity. A safe-prop bench test cannot establish those properties. Patient data, infection control and clinical deployment would require separate qualified processes.

A demonstration should reveal failures

A future protocol could score object-placement success, zipper completion, contact or drop events, completion time, human intervention and recovery. Hold out objects and layouts; distinguish taught tasks from generalization; compare compression variants with a strong full-observation policy under the same permissible information and safety conditions. Record failures as prominently as successes.

That protocol has not been frozen or run here. It is a place to begin the next experiment: choose a useful task, identify the distinctions it needs, and decide in advance how to test the system. Some tests will support the promise; others will show where it fails. We need both. The aim is not to describe a representation as universal. It is to make a promise the system can keep.

Try it before reading the answer

With reward -1 per transfer and no discount, compare the two starts above. What extra state makes return and duration predictable? Would this establish a physical robot skill?

Worked answer and limit

Returns are -4 and -7 ; durations are 4 and 7. Retaining the left count u separates them. This constructed option is not a verified robot skill: sensing, motion control, environment, failure handling and safety would require separate evidence.

Carry this forward

The appendices now let you inspect the contracts, evidence and source trail without conflating them.

Appendix A | The technical contract

A1. Exact deterministic closure

Let $F_a: X \rightarrow X$ be total for each admitted action a , Q a family of readouts and $\varphi: X \rightarrow Z$. Readout and represented transition maps on $\varphi(X)$ exist if and only if, for every collision $\varphi(x) = \varphi(x')$, both $q(x) = q(x')$ for each q in Q and $\varphi(F_a(x)) = \varphi(F_a(x'))$ for each a . Necessity applies a function to equal inputs. Sufficiency chooses any representative, defines the maps using it and invokes the collision conditions. Induction preserves any finite action word. This proof is elementary and classical; no new Lean run certified the converse construction.

Define $x \approx x'$ when all specified future outputs agree after every finite action word, including the empty word. Prefixing a word by an action establishes successor stability. Any representation that closes under every admitted action and preserves every specified readout can merge only $\approx$ -equivalent states. The quotient by $\approx$ is coarsest in this behavioral sense, not necessarily cheapest to compute, fewest bits under every encoding or easiest to learn. Finite-horizon equivalence at H gives successor equivalence at $H-1$, not necessarily at H ; a remaining-horizon index may be needed.

A2. Linear and affine special case

On finite-dimensional vector spaces, $\varphi(x) = Px$, $F_a(x) = A_ax + b_a$ and $q(x) = Qx$ close precisely when $A_a \ker(P)$ is contained in $\ker(P)$ for every a and $\ker(P)$ is contained in $\ker(Q)$. Differences of colliding states lie in $\ker(P)$. Their transformed differences must remain invisible, and their task differences must vanish. Define $G_a(Px) = PA_ax + Pb_a$ and $h(Px) = Qx$; these conditions make them well-defined on the image. An arbitrary extension outside the image is not determined by the contract. Restricted physical domains require their own collision analysis rather than assuming the unrestricted vector-space necessity.

A3. Continuous dynamics

With a specified differentiable projection and full vector field, the chain rule gives $\dot{z} = D\varphi(x)b(x)$. Fiber consistency makes a single-valued projected field possible. A well-posed autonomous reduced initial-value problem additionally needs appropriate regularity and solution conditions. The cubic projection in chapter eight separates consistent velocity from unique flow. Numerical integration error is a third issue, independent of both.

A4. Stochastic and predictive promises

For a finite controlled Markov system, equality of the joint conditional law of the next represented state and required emitted output or reward, for colliding states under each action, is sufficient to preserve their joint trace law. General spaces need measurability and appropriate conditional kernels. Expected additive-return tasks may require expected rewards rather than their full distributions, under the specified Markov model. State-space bisimulation, output-trace equality and prediction under one behavior policy are not interchangeable promises.

Let $Z = s(H)$ be a measurable summary, so $\sigma(Z)$ is contained in $\sigma(H)$. For square-integrable Y , conditional expectation minimizes squared loss. Write $Y - mZ = (Y - m) + (m - mZ)$; the cross term has expectation zero because $m - mZ$ is H -measurable and $E[Y - m|H] = 0$. The difference of optimal risks is $E[(m - mZ)^2]$. Zero defect implies conditional-mean sufficiency under that law, not full-distributional sufficiency or exact dynamic closure.

A5. Distinguishability

Equal history laws under every permitted policy imply equal output laws for any history-based diagnostic. Under equal binary priors its expected accuracy is at most one half. The fixed-experiment optimal accuracy $(1 + TV)/2$ is a standard testing identity. Different priors, longer horizons or revealing probes change the experiment. Finite-sample difficulty is not the same as exact indistinguishability.

Appendix B | Evidence ledger, not a combined benchmark

Each row below belongs to its own protocol. Different outputs, losses, horizons, sensors, randomization units and proxies prevent pooling these studies into one average “Phi9 improvement”. Counts are retained execution reports, not new runs. Mechanical audits are internal checks, not external replication.

Study	Retained scale	Main boundary
Exact foundation	121 states; 31 tests; 10,285 comparisons	Supplied deterministic integer/rational toy; no universal learning
Interaction fitting	160 fits; 20 seeds; 10,240 trajectory evaluations	Unequal tuning budgets; strong physics prior; degradation reversal
Scalar identification	360 fits; 7,200 rollouts; 9 cells	Failure penalties dominate one highlighted contrast
Self-monitor	200 ensemble fits; 3,200 predictions; 12,800 decisions	80% coverage; spread-only superiority unresolved
Initial sensing	320 streams; 960 runs; 552,960 predictions	Detection and scan latency; sparse/dense difference
Incremental sensing	400 streams; 2,400 runs; 1,520,640 predictions	Early use of same 64-example scan
Matched recovery	240 streams; 1,680 learner runs; 967,680 predictions	Shared controller; update opportunities not equal cost
Diagnosis	200 streams; 1,600 runs; 819,200 predictions	9/50 noise acquisitions; rolling baselines can win
Calibration	1,500 runs; 768,000 predictions; 19 audit checks	5/100 is a pilot estimate; no global error guarantee
Oscillator demo	6 trajectories; 3 assertions	No model trained; numerical rather than robotics evidence

The calibration comparison in full

The following table reproduces the retained online MSE means. Lower is better. One hundred paired seeds per condition; predictors are scored before their own feedback. The names refer to the frozen policies, not hypothetical optimally tuned families.

Condition	Policy	Online MSE	Postchange MSE
Noise	Inherited	0.58790	1.04408
Noise	Fixed-window	0.57834	1.02670
Noise	Alarm-conditioned	0.57809	1.02623
Noise	Rolling compact	0.58044	1.03032
Noise	Rolling full	0.58450	1.03732
Signal	Inherited	0.14988	0.26118
Signal	Fixed-window	0.14988	0.26118
Signal	Alarm-conditioned	0.14988	0.26118
Signal	Rolling compact	0.57176	1.01483
Signal	Rolling full	0.08996	0.15375
Shuffled	Inherited	0.57796	1.02568
Shuffled	Fixed-window	0.56964	1.01083
Shuffled	Alarm-conditioned	0.56964	1.01083
Shuffled	Rolling compact	0.57176	1.01483
Shuffled	Rolling full	0.57765	1.02556

What this edition does not rerun

No neural model, robot policy, C-MAPSS model or new empirical campaign was trained or evaluated. The oscillator curves are analytic explanatory drawings here; six earlier numerical trajectories remain in their original archive. The new gradient example and Koopman lift are worked algebra, not execution receipts. Existing study wall times and checks are cited from retained reports. No formal binaries or compiler chain were independently recertified.

The consolidated v2 source bundle has known historical raw-part gaps documented in its archive. This book does not manufacture missing rows or imply that textual reports replace every original raw array. Diagnosis and calibration retain separate raw-data packaging. Reproduction of the book requires only its own source and rendering dependencies; reproduction of each experiment requires that experiment's preserved code, settings and data companions.

Appendix C | Terms that should stay distinct

State: a description of the system at one instant, with a stated domain.

Observation: what the sensing process provides, which may not determine the state. Representation: retained coordinates or features of state/history.

Readout: a map from the representation to a requested output.

Transition: a next-state rule under an action. Vector field: an instantaneous direction and rate. Flow: the time-evolution map, with existence/uniqueness conditions. Trajectory: the resulting path. Basis change: an invertible re-expression when all coordinates are retained. Compression: a reduction that may merge distinctions.

Closure: self-consistent evolution of the specified representation. Task sufficiency: enough information for the specified output promise; it may be weaker than whole-representation closure. Reconstruction: recovery of an input/state, a different objective. Conditional mean: the squared-loss optimal average under a specified law, not the whole predictive distribution.

Learning: changing a fitted rule, representation or other specified component from experience. In-context adaptation: a computation depending on supplied examples, possibly with fixed weights. Calibration: a procedure relating scores or decisions to outcomes under a protocol, not automatic immunity to future shift. Selection optimism: apparent performance inflated by choosing on finite noisy data.

Policy: an action-selection rule. Option: a temporally extended action with start, continuation and stop conditions. Coverage: the fraction of cases on which a selective system acts. Risk: the declared loss on the declared population or selected cases. Cost: a stated resource measure, not an interchangeable synonym for milliseconds, sensing, energy and money.

A useful habit is to ask four questions before accepting any claim: which variable is represented, which operation changes it, which output is evaluated, and which receipt supports the result? If the words slide between meanings, the diagram should slow us down rather than conceal the slide.

Appendix D | Sources, coverage and editorial method

Preserved Phi9 source families

[P1] The world behind the number, candidate chapter 01, edition 1, 2 October 2026, nine pages in the actual attached PDF. This supplies the core box narrative, elementary closure proof and proposed experiment. It is the one explicitly numbered pre-consolidation book chapter.

[P2] Representation closure and diagnosis after change, working-note.pdf, four-page paper scaffold. Its analysis plan and proposed benchmark are not results. [P3] From addition to OaK, ten-unit foundation notebook, preserved in consolidated v2. [P4] Task-relative sufficiency technical study, revision 2, and What a model is allowed to forget, essay revision 2, cover scheduler, formal/runtime and monitor boundaries.

[P5] Sufficiency is a contract, not a representation size, fifteen-page literature/thesis revision, 1 October 2026. It replaces overbroad claims and retains a primary-text source ledger. [P6] The state, the motion and the readout, eleven-section companion and source/demo archive, 2 October 2026. Its earlier Codex critique and six numerical trajectories remain separate from this book critique.

[P7] An executable core, and what its proofs guarantee, verified-core supplement. [P8] Technical-paper-measured and Research-article-measured: interaction learning and execution comparisons. [P9] Self-monitor research note and technical supplement: fixed-coverage error selection and indexed replay.

[P10] Sensing latency and representation cost. [P11] Incremental diagnostics results. [P12] Matched recovery results. These are separate one-step sensing studies, not a single dynamical world-model trial. [P13] Budgeted diagnosis code/results and retained protocol, 2 October 2026. [P14] Calibrating selective sensor acquisition after a residual alarm, seven-page actual PDF and original study source, 2 October 2026.

Coverage is thematic consolidation of sixteen inspected/inventoried PDFs plus diagnosis protocol/results and relevant retained source documents. Redundant versions, full coefficient dumps, every exploratory contrast, raw prediction rows and all original notebooks are not reproduced in the reading text. The input inventory and attachment mapping in the editable archive locate them. Originals were not edited. Companion sections and notebook units were reorganized, not miscounted as completed chapters.

Established references and reading trail

The earlier thesis review inspected selected primary-text passages; this edition uses that retained ledger rather than claiming a fresh full-paper review of every citation. Links locate the primary sources. A reference supplies context, not proof that our different implementation inherits its guarantee.

[R1] Hopcroft (1971), An $n \log n$ Algorithm for Minimizing States in a Finite Automaton. <https://www.cs.cmu.edu/~cdm/resources/Hopcroft1971.pdf>

[R2] Fisher (1922), On the Mathematical Foundations of Theoretical Statistics. <https://errorstatistics.com/wp-content/uploads/2021/04/fisher-1922.pdf>

[R3] Shalizi and Crutchfield (2001), Computational Mechanics: Pattern and Prediction, Structure and Simplicity. <https://bactra.org/research/cmppss.pdf>

[R4] Li, Walsh and Littman (2006), Towards a Unified Theory of State Abstraction for MDPs. <https://thomasjwalsh.net/pub/aima06Towards.pdf>

[R5] Ferns, Panangaden and Precup (2004), Metrics for Finite Markov Decision Processes. <https://arxiv.org/pdf/1207.4114>

[R6] Littman, Sutton and Singh (2001/2002), Predictive Representations of State. https://proceedings.neurips.cc/paper_files/paper/2001/file/1e4d36177d71bbb3558e43af9577d70e-Paper.pdf

[R7] Shim, Hwang and Yang (2018), Joint Active Feature Acquisition and Classification with Variable-Size Set Encoding. https://proceedings.neurips.cc/paper_files/paper/2018/file/e5841df2166dd424a57127423d276bbe-Paper.pdf

[R8] Bifet and Gavaldà (2007), Learning from Time-Changing Data with Adaptive Windowing. Author manuscript: <https://www.cs.upc.edu/~Gavalda/papers/adwin06.pdf>

- [R9] Rahbar et al. (2023), Efficient Online Decision Tree Learning with Active Feature Acquisition.
<https://www.ijcai.org/proceedings/2023/0463.pdf>
- [R10] Wang and Stadler (2021 preprint), Online Feature Selection for Efficient Learning in Networked Systems. <https://arxiv.org/pdf/2112.08253>
- [R11] Tishby, Pereira and Bialek (1999), The Information Bottleneck Method. https://www.princeton.edu/~wbialek/our_papers/tishby+al_99.pdf
- [R12] Zhang et al. (2021), Learning Invariant Representations for Reinforcement Learning without Reconstruction.
<https://arxiv.org/pdf/2006.10742v2>
- [R13] Sutton, Bowling and Pilarski (2022), The Alberta Plan for AI Research. <https://arxiv.org/abs/2208.11173> ; Sutton, Oak architecture invited talk: <https://neurips.cc/virtual/2025/loc/san-diego/invited-talk/109601>
- [R14] Rice (1953), Classes of Recursively Enumerable Sets and Their Decision Problems. <https://www.ams.org/journals/tran/1953-074-02/S0002-9947-1953-0053041-6/S0002-9947-1953-0053041-6.pdf>
- [R15] Shao, Mao and Zhang (2022), Learning Task-Oriented Communication for Edge Inference: An Information Bottleneck Approach.
<https://arxiv.org/pdf/2102.04170v3>
- [R16] Kemertas and Aumentado-Armstrong (2021), Towards Robust Bisimulation Metric Learning. <https://proceedings.neurips.cc/paper/2021/file/e/256bf8e6923a52fda8ddf7dc050a1148-Paper.pdf>
- [R17] Cawley and Talbot (2010), On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation.
<https://jmlr.org/papers/v11/cawley10a.html>
- [R18] Geifman and El-Yaniv (2019), SelectiveNet; Lakshminarayanan, Pritzel and Blundell (2017), Deep Ensembles.
<https://arxiv.org/abs/1901.09192> ; <https://arxiv.org/abs/1612.01474>
- [R19] Kingma and Welling (2013), Auto-Encoding Variational Bayes.
<https://arxiv.org/abs/1312.6114>
- [R20] Vaswani et al. (2017), Attention Is All You Need.
<https://arxiv.org/abs/1706.03762>

[R21] Hafner et al. (2019), Dream to Control: Learning Behaviors by Latent Imagination. <https://arxiv.org/abs/1912.01603>

[R22] Garg et al. (2022), What Can Transformers Learn In-Context? <https://arxiv.org/abs/2208.01066>

[R23] Chen et al. (2018), Neural Ordinary Differential Equations. <https://arxiv.org/abs/1806.07366>

[R24] Brunton, Brunton, Proctor and Kutz (2016), Koopman Invariant Subspaces and Finite Linear Representations of Nonlinear Dynamical Systems for Control. Retained primary PDF check from editorial copy 1: introductory invariant-subspace discussion and operator/generator definitions, printed pp. 2–5; the polynomial lifting construction on printed pp. 8–9 was also checked against the example in chapter six. Not a claim to have reviewed all nineteen pages.

<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0150171>

[R25] Dwork et al. (2015), Generalization in Adaptive Data Analysis and Holdout Reuse. <https://arxiv.org/abs/1506.02629>

[R26] Howard et al. (2021), Time-uniform, nonparametric, nonasymptotic confidence sequences. <https://doi.org/10.1214/20-AOS1991>

What was learned from Welch Labs

The official title located is The Welch Labs Illustrated Guide to AI. Its official page describes 376 pages, nine chapters, full-color visuals and exercises, spanning perceptrons through AI video generation. One current book was established, not a verified collection of additional volumes.

<https://www.welchlabs.com/ai-book>

Access was the official public product page, the freely linked fifty-seven-page exercises PDF and the author’s linked public code directory. The complete paid book was not acquired or read. Selected rendered sample pages inspected were PDF pages 3, 24, 25, 50 and 51, corresponding to printed pages 54, 139, 140, 292 and 293. These show concrete perceptron cases, forward/backward calculation tables and attention matrices annotated with dimensions. The exercises document was text-extracted to identify its coverage; not every exercise was solved.

The useful principles were to keep one concrete object in view, draw what each operation changes, expose shapes and dimensions, let a worked

calculation bridge intuition and equations, and pair explanations with an activity that can reveal misunderstanding. This learner edition applies those general principles to original Phi9 box, flow, error and acquisition diagrams. Welch prose, exercises and artwork are not reproduced. Public code was identified and its directory inspected, not executed. No purchase or unauthorized copy was made.

Official sample:

https://www.welchlabs.com/s/exercises_rev_12_nov_14_1.pdf Author code:
https://github.com/stephencwelch/ai_book

Editorial review and reproducibility

A bounded text-only Codex review informed the earlier consolidated edition. It covered complete chapters 3, 6, 8 and 11–14, Appendix A, the chapter map and part of a teaching excerpt from chapter 4; it did not inspect the final PDF or replicate the experiments. Its suggestions were independently assessed. The earlier edition already made nested information explicit in the Bayes-risk identity, required closure and readout preservation for behavioral minimality, retained joint stochastic laws where joint traces were promised, distinguished Koopman operators from generators, and stated value-integrability and statistical sampling assumptions. This copy preserves those conditions. No new Codex review was run for this editorial pass. Private review prompts, execution receipts and provider metadata are not included in its source package. The review is critique, not proof or replication.

This editorial source package contains the editable Markdown, renderer with original figure instructions and result tables, change summary and build notes. It can rebuild the reading copy without credentials, robot access or rerunning experiments. It is not a complete raw-data reproduction archive. Layout checks concern the delivered pages; scientific claims remain bounded by the retained evidence described above.

Appendix E | What the later transformation studies establish

Prediction is not inversion

In the original affine robustness campaign, 20 seeds crossed three training sizes, three positive observation-noise levels and three transformation sequences: 540 records in 27 cells. Direct prediction had lower mean latent RMSE in all 27 cells and lower error in 475 of 540 paired runs. Twenty-six ordinary paired intervals and 22 conservative Bonferroni intervals favored direct prediction; Holm correction rejected 26 of 27 nulls under the paired t assumptions. The original contrast is composed minus direct; the newer tables below use direct minus composed. Signs therefore differ without a scientific contradiction. The direct learner received separate endpoint labels; this campaign was not a matched-information comparison.

The fitted determinant guard wrongly accepted inverse requests for 18/180 lossy rows. A supplied information-loss contract rejected all 180 historical and 36 fresh lossy cases; all 27 unit tests passed. It did not alter the predictor, infer the contract or restore lost information. Algebraically invertible fitted maps do not certify world-state recoverability.

Match declared observations, not “everything”

Sequence / noise	Direct / composed RMSE	D – C; simultaneous interval
turn-shear / 0.03	0.043811 / 0.045495	-0.001684 [-0.002597, -0.000771]
shear-turn / 0.03	0.044072 / 0.045888	-0.001816 [-0.002902, -0.000731]
turn-compress-shear / 0.03	0.037143 / 0.039221	-0.002078 [-0.003431, -0.000725]
turn-shear / 0.2	0.281236 / 0.292247	-0.011011 [-0.016912, -0.005111]
shear-turn / 0.2	0.283712 / 0.298145	-0.014433 [-0.020659, -0.008208]
turn-compress-shear / 0.2	0.243436 / 0.267873	-0.024437 [-0.035878, -0.012995]

Negative D – C favors direct prediction. These are six-family simultaneous paired intervals, 20 seeds per cell, not error bars around individual predictions. Matching 120 records and 720 scalar supervised slots does not match intermediate access, unique sensor readings, correlation, number of fits or parameters. Shared PCA uses additional calibration access. The low-noise lossy same-trajectory control had D – C=+0.000337 with an unadjusted interval approximately [+0.000059,+0.000615]; it was exploratory.

Intermediate-noise reuse changes the comparison

Sequence / noise	Independent D – C	Reused D – C	Reuse – independent
turn-shear / 0.03	-0.001388 [-0.002270, -0.000506]	-0.000035 [-0.000152, +0.000082]	+0.001353 [+0.000455, +0.002252]
shear-turn / 0.03	-0.001439 [-0.002144, -0.000733]	-0.000016 [-0.000151, +0.000119]	+0.001423 [+0.000732, +0.002114]
turn-compress-shear / 0.03	-0.001702 [-0.003265, -0.000138]	-0.000006 [-0.000490, +0.000477]	+0.001695 [+0.000309, +0.003082]
turn-shear / 0.2	-0.011864 [-0.017212, -0.006517]	-0.006339 [-0.011103, -0.001575]	+0.005526 [+0.001465, +0.009587]
shear-turn / 0.2	-0.013507 [-0.018034, -0.008981]	-0.007279 [-0.010528, -0.004030]	+0.006229 [+0.002237, +0.010220]
turn-compress-shear / 0.2	-0.013188 [-0.023021, -0.003354]	-0.011132 [-0.016760, -0.005505]	+0.002055 [-0.005198, +0.009308]

The intervals in this table use the frozen 18-contrast Bonferroni family (12 prediction contrasts and six shifts) over 32 fresh seeds. The reused low-noise intervals contain zero: inconclusive is not equivalent. High-noise reused contrasts favor direct prediction. Five shift intervals exclude zero toward composition; the high-noise lossy shift does not. Reuse changes distinct sensor readings: 540 or 480 scalar readings for two- or three-stage complete trajectories versus 720 independent scalar slots. Equal records are not equal information.

The audit failure stays in the result

The independent-noise shear-turn cell at $\sigma=0.03$, channel 0, produced correlation -0.07048 across 1,920 samples. The frozen absolute limit was 0.06. This was one failed covariance audit among 36; variance ratios stayed within their frozen bands. No seed, tolerance or observation array was changed. Exact same-runtime replay and nonfinite/rank checks passed, but those checks do not erase the failed criterion or establish independent scientific replication. The sensitivity estimates remain reported with this limitation.

What changed since the earlier paper

A later raw archive contains all 540 original rows and all 15 exported columns; the old eight-column projection gap is resolved. Historical source-to-campaign byte linkage and original per-point test predictions are not established. The updated paper supplied with this edition retains the original 27-cell table and analyzes these later studies separately. None was rerun while writing this book.

An experiment we have not yet done

A seeded familiarity map would need whole-sequence holdouts, frozen primitive fits and thresholds, fresh states and random seeds, separate axes for parameter shift, order, depth, family and information loss, and a declared observation/computation budget. Compare calibrated selective risk at its actual coverage, not only accepted-case error. This is a proposed protocol, not demonstrated generalization.