09 What Is Still Broken in World Models?

BY NILUSHANAN KULASINGHAM

11 MIN READ · INTERACTIVE

> Scenes do not fail all at once, and different systems fail under different contracts. The closing argument: what to test, which benchmark claims compose, and which do not.

The figures in this chapter are interactive. Press and drag them online at worldmodels101.com/chapters/whats-broken.

Every system in this course ends on something that does not work yet. I think the failures belong together, because the confusion around the phrase was built out of them. Five kept turning up: persistence, action, horizon, target and verification. Each was found by a different community, and each was measured late or not at all.

Chapter 1 met the five machines the phrase can mean: the renderer, the simulator, the dynamics model, the representation and the implicit model.

PERSISTENCE

Does the same object remain itself after it leaves view and returns?

ACTION

Do different commands reliably cause the corresponding different futures?

HORIZON

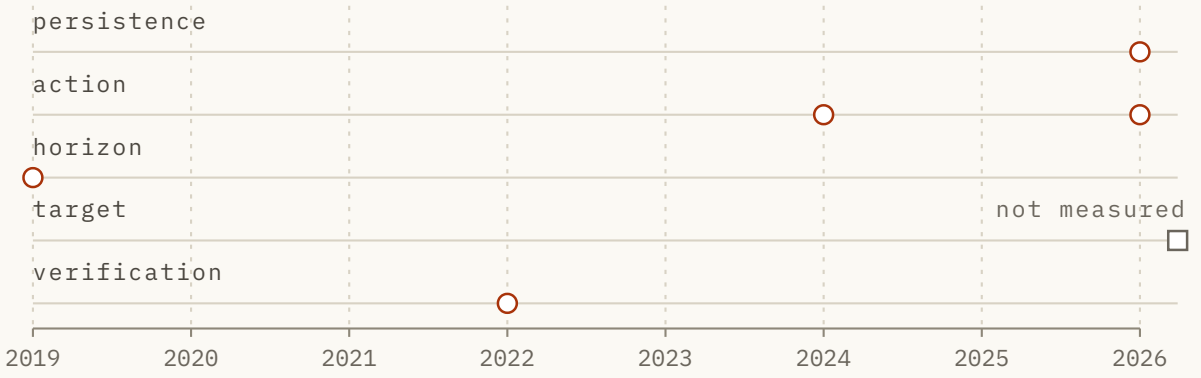
How far can it run on its own outputs before the answer stops supporting a decision?

TARGET

What information is the training target allowed to discard or collapse?

VERIFICATION

What observation could prove the claimed internal model is not there?



Five failures, five communities. Read the dates from left to right.

WHO RAN INTO IT FIRST

Click a card above.

FIRST MEASURED

Click a card above.

FIG. 9.1 The five failures, laid out. Click one to read the question it asks, which community ran into it first, and when somebody first measured it. Then read the dates along the bottom from left to right. The order is the odd part.

Click through the five above before you read on, and notice which one each system hits first.

Rollouts do not fail all at once

Let's start with the rollout, because that is where most of the failures show up. A rollout is the model's predictions run forward step after step, each one built on the last. The usual report is one average error over the whole run. That melts several failures into one number and hides the order they happened in.

Identity can go while texture stays sharp. Action fidelity, whether the action you gave it is the action it took, can go while both still look fine. Which goes first depends on the system and the task. So the useful report is one horizon per property: how many steps each one held up for.

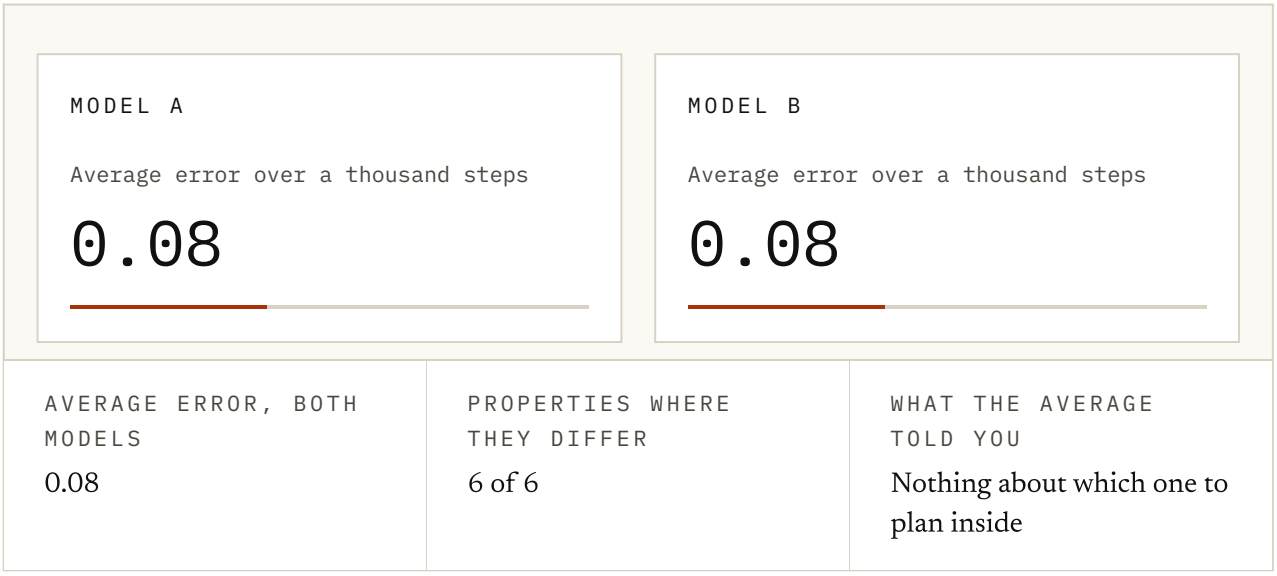


FIG. 9.2 Two models with the same average error over a thousand steps. Leave the report on one number and there is nothing to choose between them. Switch it to one horizon per property and read the two rulers: the same average was hiding two different machines. Click a property to see how far apart they are on it. The profiles are illustrative.

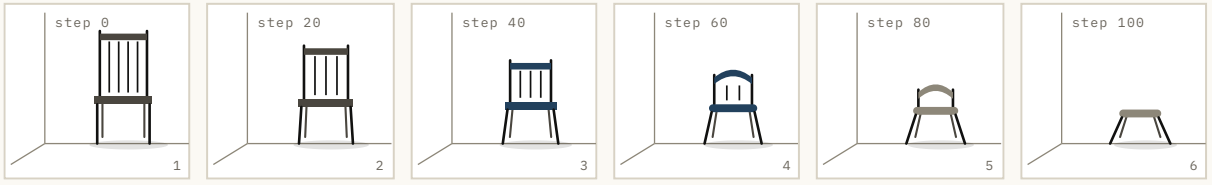
In both profiles, colour and texture outlast identity and physics, and the loss explains why. A per-frame loss pays the model for getting most of the pixels about right, and most pixels are surfaces and light. Nothing in it is watching whether the chair you walked past is the same chair.

The result is what I call the single-frame alibi. You get a rollout a hundred steps past the point of being useful, and any one frame of it still looks completely fine. The mistake only shows when you look along the sequence.

Three failures that outlived every fix

Things stop being themselves. Object permanence is the expectation that a thing is still there, and still itself, when you look back. It is the failure everyone notices, and the one chapter 1's turn-around test was built to catch and could not.

Take a renderer, a model whose output is pictures. It has no object store, so identity lives in the predictor's hidden state, the numbers it carries between steps. When the chair becomes another chair, it does so a little at a time and without an error message. A simulator stores the object explicitly, but then needs the structure right first.



SELECTED FRAME	THIS FRAME ON ITS OWN	SAME CHAIR AS FRAME 1?
Frame 6, step 100	looks fine	no
HORIZON	**FRAMES THAT LOOK FINE ON THEIR OWN**	**FRAMES SHOWING THE SAME CHAIR AS FRAME 1**
100 steps	6 of 6	1 of 6

FIG. 9.3 Drag the horizon and look at any one frame on its own: sharp edges, plausible light, nothing to complain about. Now look along the strip. Leave the switch on hidden state and the chair turns into a different chair a little at a time, with no frame you can point to as the mistake. Flip it to object store and every frame shows the same chair, because a stored object is being read back. The frames are illustrative.

What if is not really available. A model can tell you what is likely to happen next. What would have happened if you had done something else is a different question, and it is the one a decision needs. The model's answer to it comes from the model, not from the world.

Bingyi Kang and colleagues measured the gap in 2024, in *How Far is Video Generation from World Model*. They tested video generators on physical laws inside and outside their training data. Inside, the generators obeyed the laws. Outside, they broke them, and nothing in the output said which case you were in.

> I think this one gets under-rated because the model always answers. A system that said "I have never seen anything like that" would be more useful and less impressive, and nobody is rewarded for building it.

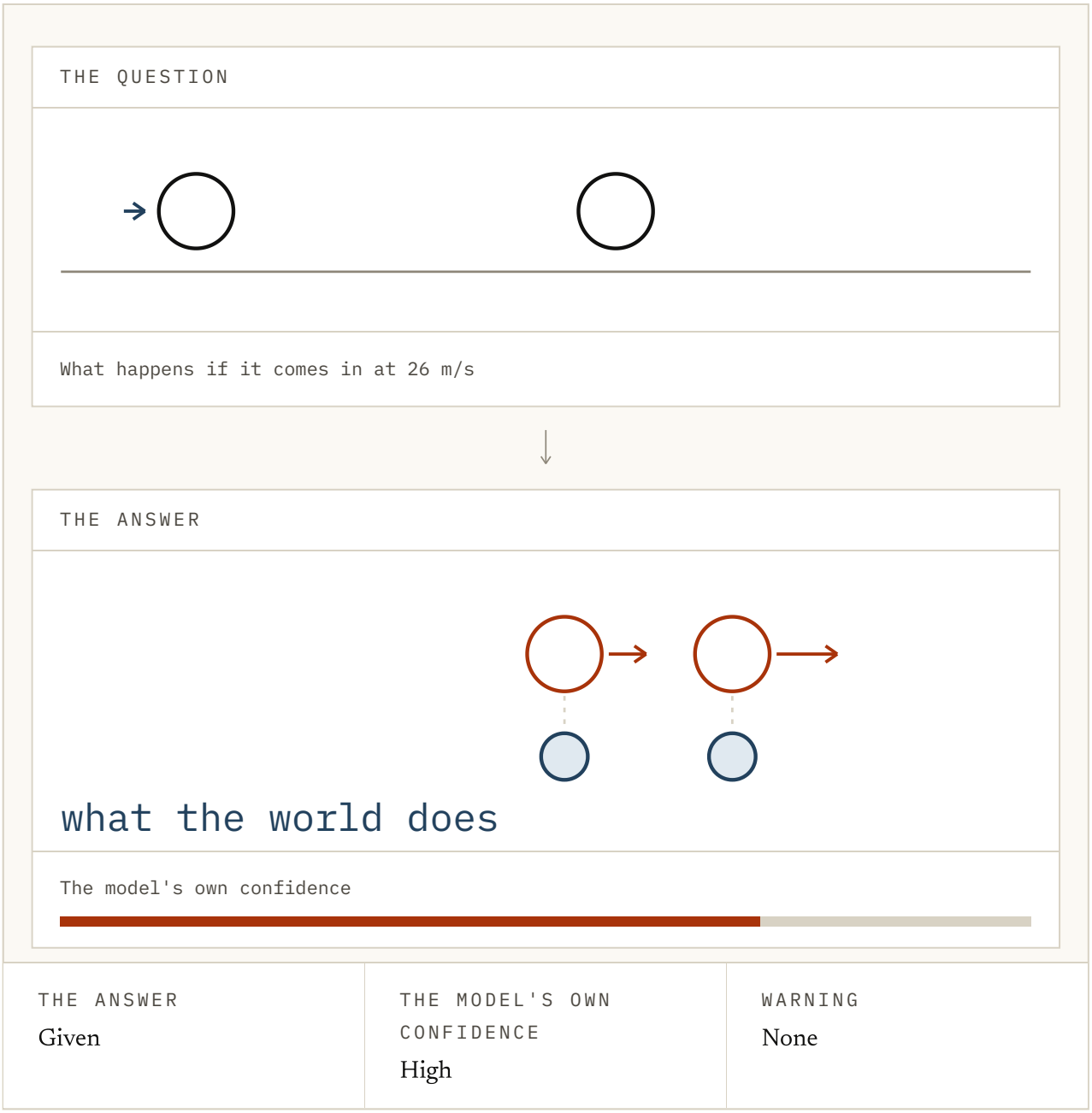
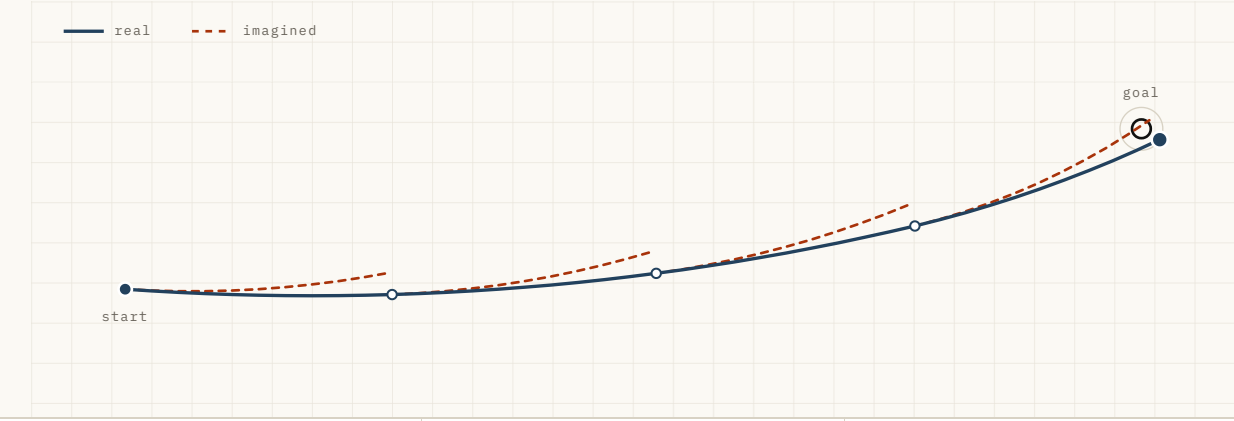


FIG. 9.4 Three questions put to the same generator: a speed the footage had, one just past it, and one nothing like it. Step through them and watch the answer card. It arrives every time, drawn the same way, with the same confidence, and only the slate outcome beside it says when it was wrong. Then switch on the flag a useful system would raise and see which questions it catches. The speeds and the outcomes are illustrative.

Long horizons are still the binding constraint. Most tricks in model-based control are ways of not needing the model to be right for very long. Horizon was the first of the five to be measured. It was measured by the people who had to act on the model's predictions, in reinforcement learning.

Tingwu Wang and colleagues did it in 2019, in *Benchmarking Model-Based Reinforcement Learning*. They ran the methods that learn a model of their environment and plan inside it side by side. The planning horizon, meaning how many steps ahead the model was trusted, decided where each won and lost.

Michael Janner and colleagues answered the same year, under a title that is still the question: *When to Trust Your Model*. Their answer was only briefly, and from a real state. Short rollouts and replanning every few steps work because they stop relying on the model before it drifts. The horizon over which it holds still decides what you can build.



STEPS TRUSTED	REPLANS	DISTANCE FROM GOAL AT THE END
10	3	reached

FIG. 9.5 Janner's answer, made playable. The slate path is what really happens and the vermilion path is what the model imagines. Drag the steps trusted out to thirty and press Run: the imagined path leaves the real one, so the plan is for a world that is not there. Pull it back to three and run again. Now the planner snaps back to the real state every few steps, and the errors never get the chance to pile up. The numbers are illustrative.

The league table nobody can publish

Measurement arrived in the wrong order, and I want to walk you through why. Reinforcement learning measured horizon first, because an agent that trusts a bad model loses, and the loss shows up in the score. The video people came next and

measured what they could, which was the frame.

VBench came in 2023, from Ziqi Huang and colleagues. It scored generated video on separate dimensions, such as frame quality and how smoothly things moved. It used enough of them that the ordering stopped being obvious.

Benchmarks that test worlds you act inside, rather than watch, came last. PlayWorld, released in August 2026, tests 171 scenarios. They cover geometry, interaction fidelity (does an action do what it should) and what happens to objects as they leave and re-enter view. That is persistence and action, the two failures a renderer hits first, tested seven years after horizon.

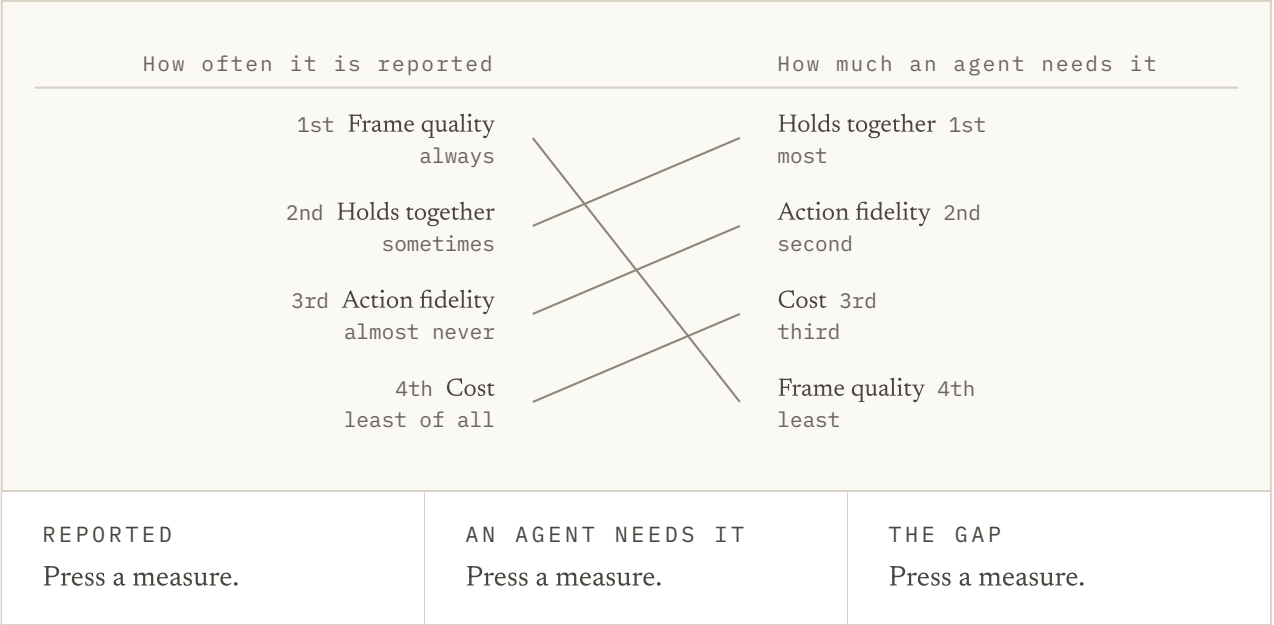


FIG. 9.6 The four measures, ordered twice. On the left, how often they get reported; on the right, how much an agent needs them. Press a measure to draw its line across. Nearly every line crosses, and that is the whole of what I mean by the wrong order. The orderings are my judgement rather than a survey.

Frame quality is measurable, and it is what demos are made of. Whether the world holds together over a thousand steps is what an agent needs, and no demo is long enough to show it. Action fidelity separates a world model from a video generator, and it is almost never reported. Cost is reported least of all.

That order is close to the reverse of how much each matters, because the easy measures were taken first. Downstream task success looks like the way round this. But it scores the model and the policy together, and a good policy hides a bad model.

Progress is real. Genie 3, from Google DeepMind in 2025, is said to stay coherent for several minutes. That is reported by the lab that built it, on a measure it chose. "Better" means nothing until the measure is named, and the system contract with it, meaning which of the five machines is claimed.

Match the failure test to the system

The phrase covers five machines, and for most of this history they sat the same exam. That is where the confusion came from. A renderer depends on persistence and action fidelity, and the single-frame alibi is its usual failure. A simulator is the easy one to check, because it exports structure that another program can test.

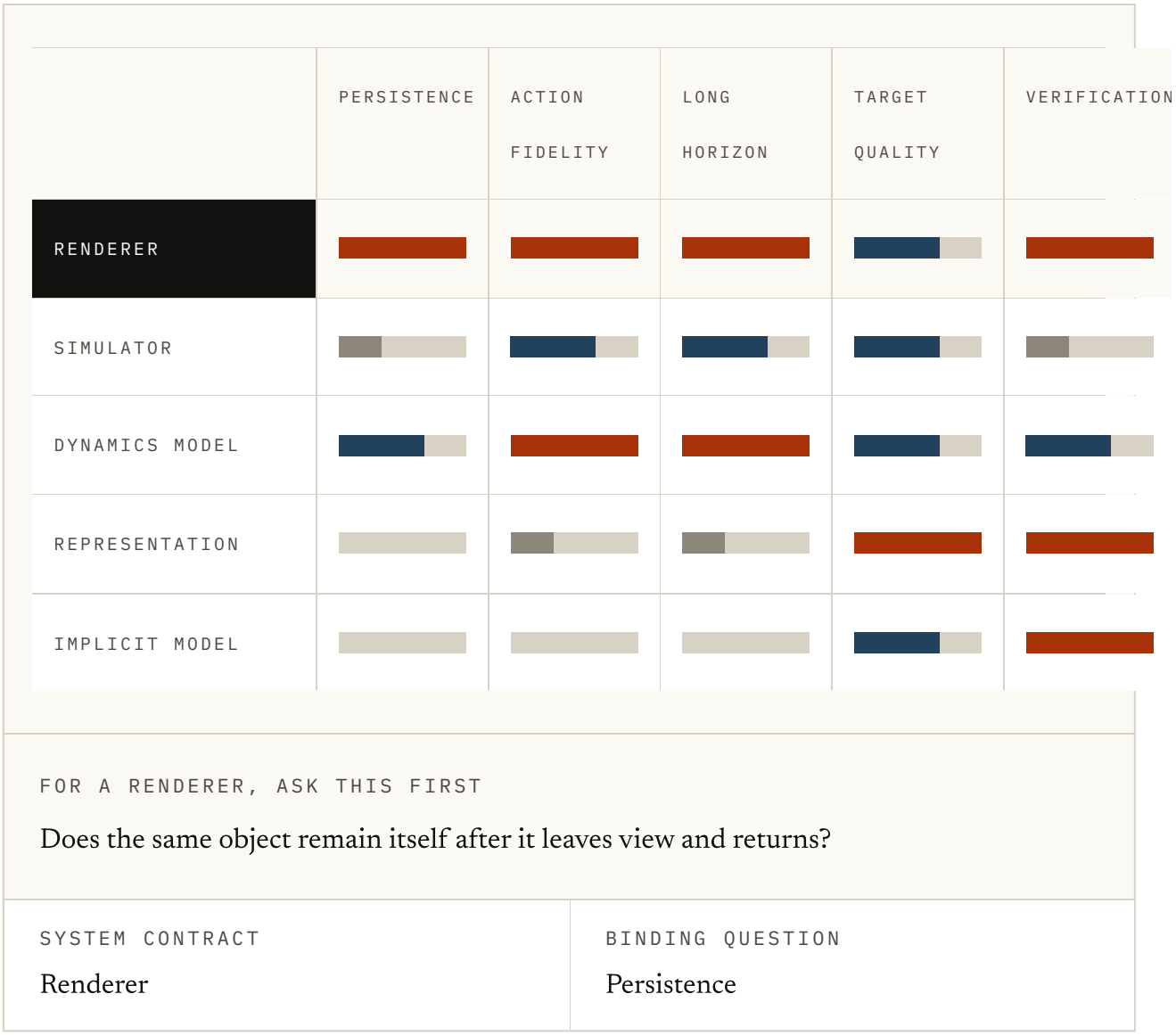


FIG. 9.7 Five system contracts crossed with the five failure questions. The levels are editorial judgements, mine, rather than benchmark scores. Select a row to see the question that most often binds that kind of system, and read the renderer and simulator rows against the paragraph above. The next three paragraphs walk the other three rows. A single "world model quality" axis would flatten all five.

A compact dynamics model predicts a small state rather than pixels. It depends on rollout horizon, and on whether planning search can exploit it. A representation, an embedding learned for something else to use, can look stable while throwing away the variable the task needed. I call that the target problem: predicting the wrong thing well.

An implicit model is a claim that a model formed inside a network trained for some other job. Checking it needs a probe, a small second network trained to read one fact out of the first's internals. Kenneth Li and colleagues showed what that takes in 2022, in *Emergent World Representations*.

A network trained only to predict the next move in Othello (a board game played with two-sided discs) turned out, under probes, to hold a picture of the board. The probe could have found nothing.

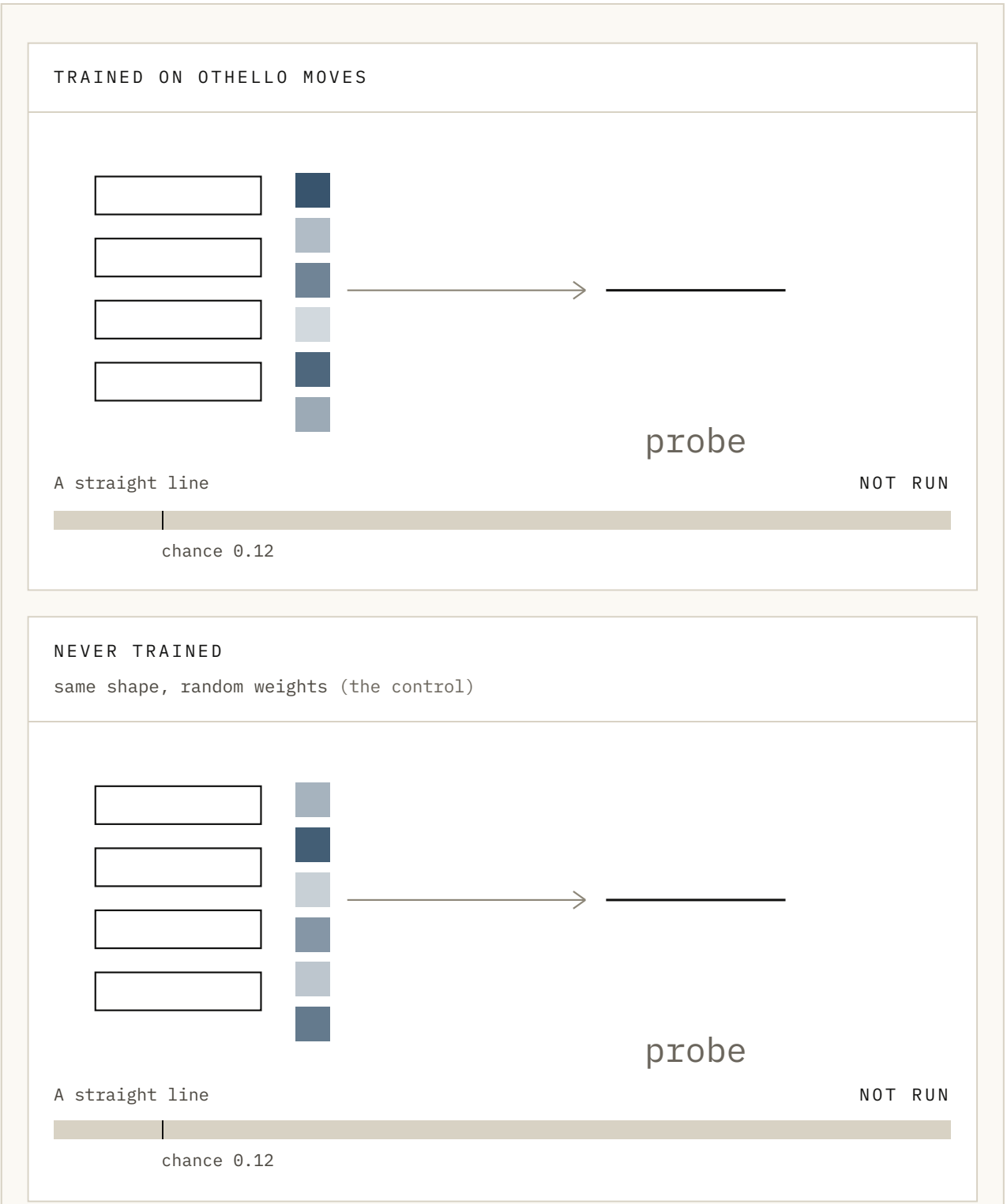


FIG. 9.8 The same probe run against two networks: the one trained on Othello moves, and one of the same shape that never trained. Press Run the probe and compare the two bars against the chance line. Then let the probe be a big network instead of a straight line, run it again, and it reads a board out of the untrained one too. That is why the probe has to be small. The accuracies are illustrative.

Four questions still handle most papers, so let me leave you with them. The first is **what does it output?** A picture, a structure, a compact state or an embedding, and the other three questions depend on the answer.

The second is **can you roll it forward under actions nobody took?** If not, it will not support a decision, however good it looks.

The third is **over what horizon does it hold up, and who measured that?** The number matters more than the architecture, and it is usually missing.

The fourth is **which measure is the claim using, and what does it ignore?** There is no overall score, so anyone implying one has picked a measure and not told you which.

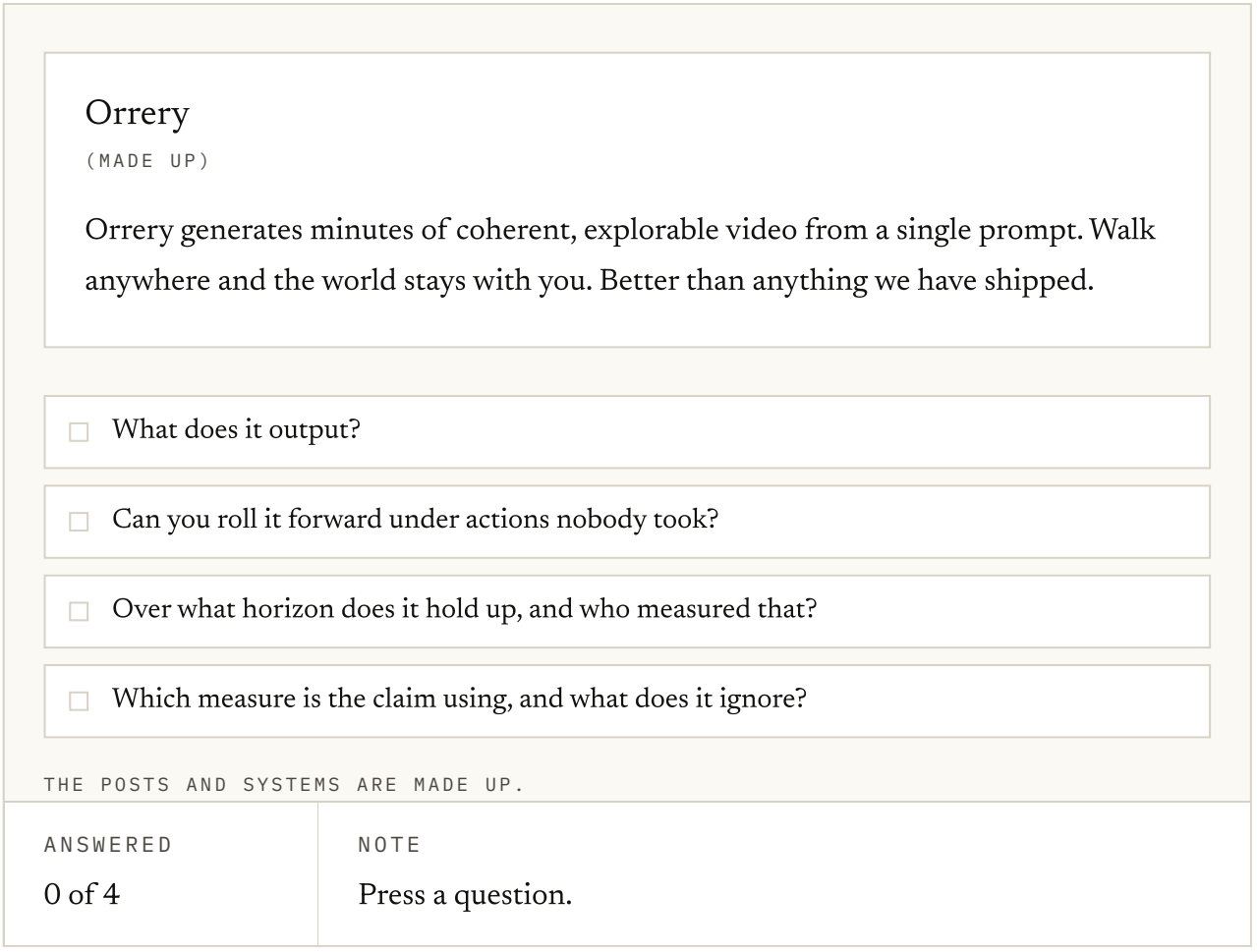


FIG. 9.9 The four questions, put to three made-up launch posts. Pick a post, then press each question in turn. Where the post answers it, the words that answer it light up. Where it does not, you get "not stated", and the counter at the bottom keeps score. Try all three posts, and you should find that the most exciting one answers the fewest. The posts are invented, and so are the systems in them.

Yann LeCun, then Meta's chief AI scientist and a Turing Award winner, wrote the fullest statement of the target in 2022. The paper is *A Path Towards Autonomous Machine Intelligence*, and it sets out what a world model would have to do to deserve the name. Read against the four questions, it is a list of what is still missing, and naming an architecture answers none of them.

That brings us to the end of the course. The phrase still means five different things, and the people using it still rarely say which. What has changed is on your side: you can now tell which of the five is being offered, what it would take for the claim to be true, and what to ask when somebody is not saying.

Thank you for reading this far. Hopefully these nine chapters helped, and there is a short quiz below if you want to check the ideas stuck. If I have got something wrong anywhere in the course, or you know a better source than the ones I used, the About page has a corrections section. I would be glad to hear from you.

Try it

1 Why is one average error insufficient as a long-rollout report?

a. Different properties can fail at different horizons, and an average hides which contract was lost
b. The error is measured in the wrong units
c. Rollouts are too short to average over
d. Averages are always misleading

> ANSWER a. Different properties can fail at different horizons, and an average hides which contract was lost. There is no universal failure order. The point is to report separate horizons for identity, physics, control and appearance instead of allowing one strong dimension to conceal another.

2 A rollout still looks completely fine in any single frame. What does that tell you?

a. The horizon has not been reached
b. It is still usable
c. Very little. Most of the pixels are surfaces and light, and nothing in a per-frame loss is watching whether objects stayed themselves
d. The model has learned physics

> ANSWER c. Very little. Most of the pixels are surfaces and light, and nothing in a per-frame loss is watching whether objects stayed themselves. This is how the subject fools you, and it is why demos are weaker evidence than they feel.

3 Why is object permanence fragile in a renderer with no exposed object store?

a. Occlusion is computationally expensive
b. Training data lacks occluded objects
c. The models are too small
d. Identity is carried implicitly by predictor state rather than enforced by a persistent record

> ANSWER d. Identity is carried implicitly by predictor state rather than enforced by a persistent record. A structured simulator may store objects explicitly. In the renderer case, identity can drift gradually because persistence is an inferred behaviour rather than an inspectable record.

4 You ask a model what would have happened if you had acted differently. What have you got?

a. A fact about the world
b. An answer that is true inside the model, with nothing in it indicating whether the model had data there
c. A proof
d. Nothing, models cannot answer that

> ANSWER b. An answer that is true inside the model, with nothing in it indicating whether the model had data there. Where the model has data the two are close. Where it does not, they are not, and the model answers with equal confidence either way.

5 Short rollouts, replanning and starting from real states all work. What do they have in common?

a. They are all ways of not needing the model to be right for very long
b. They only apply to robotics
c. They remove the need for a policy
d. They make the model more accurate

> ANSWER a. They are all ways of not needing the model to be right for very long. Read uncharitably, the whole toolkit is an admission. The horizon over which a model holds up is the number that decides what you can build.

6 Why does downstream task success not settle which model is better?

a. It only works in simulation
b. Tasks are too easy
c. It scores the model and the policy together, so a good policy hides a bad model
d. It cannot be measured reliably

> ANSWER c. It scores the model and the policy together, so a good policy hides a bad model. It sounds careful, and it conflates the two things you were trying to tell apart.

7 A benchmark winner is best on geometry but mediocre on action fidelity. What is the first thing to ask?

a. What hardware it runs on
b. How long it trained for
c. How large is it
d. Which measure, and what does that measure ignore

> ANSWER d. Which measure, and what does that measure ignore. Benchmarks such as PlayWorld make useful dimensions explicit. They do not remove the need to say which contract matters for the intended use or how competing dimensions were combined.

8 Across the whole course, which single question does the most work on an unfamiliar system?

a. How many parameters does it have
b. What does it output, and can you roll it forward under actions nobody took
c. Is it open source
d. Does it use a transformer

ANSWER b. What does it output, and can you roll it forward under actions nobody took. The first half decides what it can be used for. The second decides whether it can support a decision. Almost everything else follows, and neither depends on knowing the architecture.

FIG. 9.10 Eight questions on what is still broken. If you have read the whole course, the last one should be easy and slightly annoying.

Sources

PlayWorld: A Benchmark for Interactive World Models ZHANG ET AL., 2026 ↗

A current benchmark spanning 171 scenarios and separating geometry, interaction fidelity, and out-of-sight evolution rather than pretending world quality is one number.

https://arxiv.org/abs/2608.13552

How Far is Video Generation from World Model: A Physical Law Perspective

KANG ET AL., 2024 ↗

Physical laws matched inside the training distribution and not extrapolated outside it. The clearest measured statement of the gap this chapter is about.

https://arxiv.org/abs/2411.02385

VBench: Comprehensive Benchmark Suite for Video Generative Models

HUANG ET AL., 2023 ↗

A serious attempt at the measurement problem, and useful for seeing how many separate dimensions it takes before the ordering stops being obvious.

https://arxiv.org/abs/2311.17982

Benchmarking Model-Based Reinforcement Learning WANG ET AL., 2019 ↗

Where model-based methods win and lose, and the discovery that the planning horizon is the number that decides it.

https://arxiv.org/abs/1907.02057

When to Trust Your Model: Model-Based Policy Optimisation JANNER ET AL., 2019 ↗

Short rollouts as an answer to a model you cannot trust for long. The title is this chapter's question.

https://arxiv.org/abs/1906.08253

Emergent World Representations LI ET AL., 2022 ↗

The strongest available evidence that something structured forms inside a predictor, and a good example of what it takes to show it rather than assert it.

https://arxiv.org/abs/2210.13382

Genie 3 GOOGLE DEEPMIND, 2025 ↗

Reported coherence over minutes, from the lab that built it. Included as an example of exactly the kind of claim this chapter is asking you to read carefully.

https://deepmind.google/blog/genie-3-a-new-frontier-for-world-models/

A Path Towards Autonomous Machine Intelligence LECUN, 2022 ↗

The most complete statement of what a world model would have to do to be worth the name, which doubles as a list of what is still missing.

https://openreview.net/forum?id=BZ5a1r-kVsf

FIG. 9.11 I've ordered these for someone building on this rather than for historical completeness, benchmarks first and LeCun last.