CAN WE CHECK WHETHER A WORLD MODEL FOLLOWS ACTIONS WITHOUT A SIMULATOR?

Anonymous authors
Paper under double-blind review

ABSTRACT

World models predict how an environment changes when an agent acts, and agents plan by comparing these predictions across actions. This only works if changing the action moves the prediction the way the real environment would move. Many existing evaluations check this with cheap probes that need no simulator. We test whether these probes notice when a model's prediction moves the wrong way. We use simulators that can return to a saved state and run again under a different action. This gives the true effect of each action change. We study 92 robot world models that we trained and 143 released models, including models that predict Atari frames. We also reflect each model's response to actions, which reverses its direction and keeps its size. We prove that probes that look only at the size of the response cannot detect this, and in our experiments they do not. A popular probe, *logged action retrieval*, even scores the reflected models higher. Training models on its criterion improves their retrieval score while their true direction gets worse. We also propose the matched probe, which compares the model's response with pairs of similar situations in the recorded data. It tracks the true direction in all three groups of models. We recommend testing any action probe on reflected models before trusting it.

1 INTRODUCTION

A world model predicts how an environment will change in response to an agent's actions. Agents use world models to plan, or to learn behavior inside them (Hansen et al., 2024; Alonso et al., 2024). A planner tries many action sequences inside the model and carries out the one with the best predicted outcome. This only works if the model predicts correctly what would happen under a different action. When the planner changes an action, the predicted future must change the way the real future would. We call this property *action faithfulness*. It has two parts, *(i) the direction in which the prediction moves* and *(ii) how far it moves*. Direction matters most. In our control experiment we reflect the responses of trained models, which keeps their size and reverses their direction. The gripper then ends up 32 cm away from a goal that is 4 cm away, on average (Section 5.4).

Measuring *action faithfulness* needs *counterfactuals*, meaning what the environment would have done under a different action from the same state. Only a simulator or a real intervention provides them. Evaluations therefore often use probes, which are scores computed without running a simulator for the model under test. A rollout is the sequence of states a model predicts. One kind of probe measures how far the rollout moves when a null action replaces the logged action, which is the action recorded in the data. Another checks whether the logged action explains the recorded outcome better than other candidate actions (Bruce et al., 2024; Chen et al., 2025). A third compares the sizes of the model's responses with the sizes in a simulator (Co et al., 2026). Several concurrent benchmarks and probes appeared in recent times (Yang et al., 2026; Shi et al., 2026; Co et al., 2026; Chen et al., 2026b). MiraBench scores action following with vision language model judges checked against human annotations (Yang et al., 2026). Concurrent work audits models against real environment branches (Vakalis, 2026) and against recorded futures (Chen, 2026). We ask a narrower question about the probes themselves. *If a model responds with the right size but in the wrong direction, does the probe notice?*

To answer this, we need the true response to every action change we ask a model about. We get it from simulators that can save a state and run again from it, in three environments. These are LIBERO

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

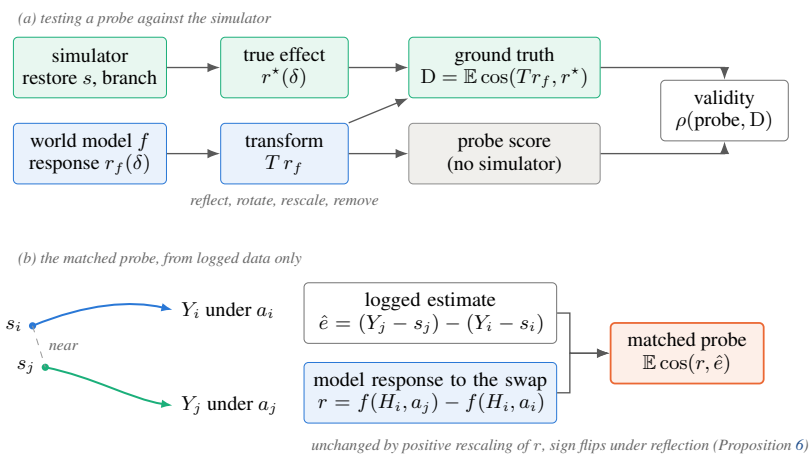

Figure 1: Our test and our probe. (a) From a saved simulator state we run the logged actions and a changed version of them, which gives the true effect of the change. We then reflect, rotate, rescale or remove the model's response to the change, while its predictions for the logged actions stay the same. A probe is only useful if its score follows the true direction through these edits. (b) Two recorded segments start from nearby states s_i and s_j but take different actions. The difference between their outcomes estimates what swapping the actions would do. The matched probe asks the model to predict the same swap and scores the cosine between the prediction and the estimate.

robot manipulation (Liu et al., 2023), the DeepMind Control Suite (Tassa et al., 2018) and Atari games in the Arcade Learning Environment (Bellemare et al., 2013). From each saved state we run the logged actions and a changed version of them and we record what the environment did. We test 235 world models. We trained 92 of them on LIBERO. The other 143 were released by others, namely 117 TD-MPC2 models for control tasks (Hansen et al., 2024) and 26 DIAMOND models that predict Atari frames (Alonso et al., 2024). We also build models whose faithfulness we know exactly. We take a trained model and reflect, rotate, rescale or remove its response to action changes (Figure 1a). These transforms keep the model's prediction for the logged actions. They change its direction and size in a way the simulator can measure. A probe that measures faithfulness must score a reflected model as worse than the original.

We find four things. First, a probe that reads only the size of a response cannot detect its direction. We prove that any such probe gives a model and its reflected copy the same score (Proposition 1, proved in Appendix B). Our experiments confirm this in all three groups of models.

Second, logged action retrieval rewards a large response whichever way it points (Proposition 3). It scores reflected and rotated models as better than the originals. When we train models on retrieval's own criterion, the retrieval score improves and the true direction gets worse.

Third, direction can be measured from logged data, for the actions the data contain. Our matched probe pairs two logged segments that start from similar states but took different actions. The difference between their outcomes estimates what swapping the actions does. We ask the model about the same swap and take the cosine between its answer and the estimate (Figure 1b). Rescaling a response cannot change this score, and reflecting it flips the sign (Proposition 6). The matched probe tracks the true direction in all three groups of models (Table 2). Among 30 LIBERO models that differ only in training objective and seed, it orders the six objectives exactly as the simulator does. Retrieval's order shows no relation to the simulator's. Among the seeds of released TD-MPC2 models, retrieval does slightly better and the difference is not significant. An inverse model check scored by direction also works where its inverse model is accurate and close to linear.

Fourth, the same logged pairs can serve as a training loss, which we call matched swap alignment. On our LIBERO models it raises direction under random action changes by 42%, with no loss in prediction accuracy. It does not improve control. It also lowers direction when we fine tune released TD-MPC2 models, whose predictions are already more accurate than the logged estimate.

Contributions.

- We test five existing action probes that need no simulator, and two direction probes, against simulator counterfactuals. We use three groups of world models, 143 of them released by others.
- We prove that probes which read only response size cannot detect direction. We give two direction probes. The matched probe ignores rescaling and flips sign under reflection, and a direction form of the inverse model check does the same when its inverse model is affine.
- We show that training on retrieval's criterion lowers true direction. Matched swap alignment raises it on the models we trained but not on released models.

Our niche is validating whether the metrics measure what they claim. Our source code is shared as supplementary material.

2 Responses, direction and probes

Let H be the history of states and actions so far, A the logged action sequence that follows, and Y the states that followed it. Logged means recorded in the data. A world model $f(H, a)$ predicts these states for any action sequence a. When we change the actions by δ, the prediction moves by $r_f(\delta) = f(H, A+\delta) - f(H, A)$. We call this the model's *response*. The true response $r^\star(\delta)$ is how the real states move. A simulator measures it by restoring the saved state and running both action sequences. We compare the two responses with two numbers,

$$\mathrm{D}(f) = \mathbb{E}\cos\big(r_f(\delta),\, r^\star(\delta)\big), \qquad \mathrm{M}(f) = \frac{\mathbb{E}\|r_f(\delta)\|}{\mathbb{E}\|r^\star(\delta)\|}. \tag{1}$$

We set $\cos(x, 0) = 0$, so a model with no response scores 0. The direction D is 1 when the prediction points exactly the way the environment moves. It is 0 when the two movements are unrelated and negative when they point in opposite directions. The size ratio M is 1 when the two movements have the same average size.

The value of D depends on which action changes we average over, and we use two kinds. For D we add random Gaussian noise to the logged actions. This often gives action sequences that never occur in the data. For $\mathrm{D}_{\log}$ we replace the logged actions with those of another logged segment. These actions occur in the data, so we say they lie on the *support* of the data. For rollouts of several steps, the responses are whole trajectories and we take the cosine over the flattened sequence.

A *probe* scores a model without running a simulator for it. Appendix A defines each probe exactly. Zero action divergence (C_1) measures how far the predicted rollout moves when a null action replaces the logged one. The calibration score of WorldSimProbe (Task 1, Co et al., 2026) takes the ratio of the model's responses to a small and a large action change. It compares this ratio with the same ratio in the simulator. It uses the simulator only for this reference ratio. Both probes depend on the model only through the sizes $\|r_f(\delta)\|$ of its responses, and we call such probes *size probes*. Logged action retrieval ranks the logged action among 15 alternatives by how close each one step prediction comes to the recorded next state. It is the only existing probe we test that compares a prediction with a recorded outcome. The inverse model check fits an inverse model $g(x_t, x_{t+1}) \mapsto a_t$ to logged transitions. It then reads the model's rollout through g, as in evaluations of driving world models (Gao et al., 2024; Arai et al., 2024) and the verifier of Liu et al. (2026). Its cycle error $\sum_t \|g(\hat{x}_t, \hat{x}_{t+1}) - a'_t\|^2$ is small when g recovers the queried actions a' from the states the model predicts under them.

3 Theoretical results

Each result below has a complete proof in Appendix B, and Lean 4 (de Moura & Ullrich, 2021; The mathlib Community, 2020) checks every one as stated, together with the examples in its plain explanation. For a matrix T we write $f_T(H, a) = f(H, A) + T\, r_f(a - A)$ for the model whose response is T times that of f, with the logged actions A as reference. So f_{-I} is the reflected model and f_{cI} is the model rescaled by c.

Proposition 1 (size probes cannot detect direction)**.** *Let Q be a linear isometry of the state space, and let $f_Q(H, a) = f(H, A) + Q\, r_f(a - A)$ be the model whose response is Q times that of f. Then $\|r_{f_Q}(\delta)\| = \|r_f(\delta)\|$ for every δ, so every size probe gives f_Q and f the same score. For the reflection $Q = -I$, $\mathrm{D}(f_{-I}) = -\mathrm{D}(f)$.*

In words. Rotating or reflecting a response changes where the prediction goes but not how far it moves. A size probe looks only at how far it moves. So it gives a model the same score as a copy that responds in the opposite direction. It also gives that score to a copy that responds in an unrelated direction. The true direction D tells them apart, because reflection flips its sign. Appendix D shows when the same holds for latent empowerment (Klyubin et al., 2005) and for the consistency conditions of Wang et al. (2026b).

Proposition 2 (the calibration score is the same for every nonzero linear response)**.** *Let the small action change be s times the large one, $s\,d$ and d. Our run uses $s = 0.25$, the default Task 1 setting of the WorldSimProbe code. If $r_f(\delta) = J\delta$ with $Jd \neq 0$, then $\|r_f(s\,d)\|^2 / \|r_f(d)\|^2 = s^2$.*

In words. The calibration score asks whether the model's response to a small action change is the right fraction of its response to a large one. For any model whose response is linear in the action and not zero, this fraction is always s^2, whatever the model has learned. The score then depends only on the simulator's own ratios. With the default setting of the WorldSimProbe code, a constant that ignores the state beats every model we trained (Appendix N).

Proposition 3 (what retrieval rewards)**.** *Let $e = f(H, A) - Y$ be the model's error at the logged action and $r = r_f(\varepsilon)$ its response to an alternative $A + \varepsilon$. Then*

$$\|f(H, A + \varepsilon) - Y\|^2 - \|f(H, A) - Y\|^2 = 2\, e^\top r + \|r\|^2. \tag{2}$$

In words. Retrieval asks whether the logged action gives a smaller prediction error than an alternative, which happens when this difference is positive. The second term is the squared size of the response. It is never negative, so a larger response makes the logged action win more often, whichever way the response points. Multiplying the response by c multiplies the first term by c and the second by c^2. If every alternative gets a nonzero response, a large enough response therefore puts the logged action first. Direction enters only through the first term, and only relative to the model's own error e. This is why retrieval gives its best score to the models whose response we multiply by 4 (Table 1).

Proposition 4 (logged data do not fix the response away from the logged actions)**.** *Suppose the logged actions come from a deterministic policy, $A = \pi(H)$. For any matrix valued function $B(H)$, the model $f_B(H, a) = f(H, a) + B(H)(a - \pi(H))$ makes the same prediction as f at every logged action. So any loss that compares its prediction at the logged action with the observed states has the same expected value on the logged data. Its response is $r_{f_B}(\delta) = r_f(\delta) + B(H)\delta$.*

This is the positivity condition of causal inference and we claim no novelty for it.

Corollary 5 (no certificate away from the logged actions)**.** *Call a probe* data grounded *if it depends on the model only through its predictions at the logged actions. Under Proposition 4, every data grounded probe gives f and f_B the same score. If the true response is linear, $r^\star(\delta) = J^\star\delta$, with $J^\star\delta \neq 0$ on every item that* D *averages over, then $\mathrm{D}(f_B) \to 1$ for $B = cJ^\star$ and $\mathrm{D}(f_B) \to -1$ for $B = -cJ^\star$ as $c \to \infty$.*

In words. If the data always pair a state with the same action, they say nothing about what happens under other actions. Adding a term that is zero at the logged action leaves every prediction the data can check unchanged. Yet the same term can turn the model's response in any direction. So no probe that uses only the model's predictions at the logged actions can certify direction, however much data it has. To measure direction from data, the data must contain different actions from similar states. The matched probe below uses such pairs. It also needs the state change under a fixed action to vary little between the two start states of a pair. Section 5.3 measures what the gap between them costs.

The matched probe. Take held out logged segments i with history H_i, start state s_i, actions a_i and following states Y_i. Pair each segment with the segment j from a different trajectory whose start state is nearest, and let

$$\hat{e}_i = (Y_j - s_j) - (Y_i - s_i), \qquad r_i = f(H_i, a_j) - f(H_i, a_i), \qquad \mathrm{MP}(f) = \tfrac{1}{n}\textstyle\sum_i \cos(r_i, \hat{e}_i). \tag{3}$$

Here $\hat{e}_i$ is the logged estimate of what swapping the actions does, and r_i is the model's answer to the same swap. The simulator's response to this swap from s_i is $r_i^\star$, so $\mathrm{D}_{\log}(f) = \frac{1}{n}\sum_i \cos(r_i, r_i^\star)$.

The estimate differs from $r_i^\star$ because s_j is not exactly s_i and because both segments contain noise. Retrieval never observes what an alternative action does. The matched probe takes that outcome from a segment in which the alternative action was actually taken.

Proposition 6 (what the matched probe measures)**.** *For every model f the following hold. (i)* $\mathrm{MP}(f_{cI}) = \mathrm{MP}(f)$ *for every $c > 0$, so rescaling a response cannot change the score. (ii)* $\mathrm{MP}(f_{-I}) = -\mathrm{MP}(f)$*. (iii) For every item with $\hat{e}_i, r_i^\star \neq 0$ and every r_i,* $|\cos(r_i, \hat{e}_i) - \cos(r_i, r_i^\star)| \leq \sqrt{2 - 2\cos(\hat{e}_i, r_i^\star)}$.

In words. The matched probe is an average cosine, so it reads direction and ignores size. Making a response larger or smaller cannot change it and reflecting a response flips its sign, just as reflection flips the true direction. Part (iii) bounds the error on one pair by how far the logged estimate points from the truth. For example, if the estimate's cosine with the true response is at least 0.9, the probe's value on that pair is within 0.45 of the true cosine. Proposition 7 in Appendix B shows that symmetric noise that is independent of the model shrinks the probe toward zero in expectation. It keeps the order of models by on support direction when the shrink factor is similar across pairs.

The inverse check scored by direction. Let $\hat{X}_a$ be the model's rollout under a. The direction form of the inverse model check is $\mathrm{IC}(f) = \mathbb{E}\cos\left(g(\hat{X}_{a'}) - g(\hat{X}_A),\ a' - A\right)$. If g is affine, IC has properties (i) and (ii) as well. If g is nonlinear, it need not have either (Proposition 8 in Appendix B). In words, a linear inverse model turns a reflected rollout into a reflected action change, so the check inherits the matched probe's behavior. A nonlinear inverse model can break this, unless it is close to linear over the size of the responses. IC measures direction only as far as g inverts the true dynamics.

4 Experimental setup

The true response. LIBERO (Liu et al., 2023) is a suite of simulated robot manipulation tasks whose demonstrations store full MuJoCo (Todorov et al., 2012) states. We pick 360 held out positions across the 30 tasks of LIBERO Goal, Object and Spatial. At each one we restore the state and run the logged 8 step action sequence. We also run three Gaussian perturbations of it ($\sigma \in \{0.25, 0.5, 1\}$, 1080 items, for D). Each is one Gaussian offset in raw action units, added to all 8 actions and clipped to $[-1, 1]$. We also run the action sequences of the 5 nearest matched segments (1800 swaps). The 360 nearest swaps give $\mathrm{D}_{\log}$, and Appendix H uses the rest. Restoring a MuJoCo state misses some controller state, which we reset before every run (Appendix E). After this fix, a repeat of the logged run differs from it by at most 0.008 in any coordinate over all 1080 items. It differs by 0.005 on average against a perturbation effect of 1.75, a signal to noise ratio of 356. We measure the 21 dimensional proprioceptive state, which holds the arm's joint, end effector and gripper readings. Appendix E explains why we could not use images here. Runs in the DeepMind Control Suite (Tassa et al., 2018) and the Atari emulator repeat exactly.

Response transforms. For each test item we replace a model's predicted rollout $P(a)$ with $P_T(a) = P(a_{\mathrm{ref}}) + T\left(P(a) - P(a_{\mathrm{ref}})\right)$, where a_{ref} is the logged action sequence. The matrix T is the identity, $-I$ (reflection), a random orthogonal matrix (rotation), $0.25I$, $4I$, or 0 (no response). The transformed model predicts exactly what the original predicts at each item's logged actions. The simulator tells us exactly what each transform does to direction and size. Where the data contain other actions from nearby states, a transformed model disagrees with them. The transforms are therefore a necessary test for a direction probe, but passing them is not enough (Section 5.2).

Models. On LIBERO we trained 92 models with one architecture, a 4 layer causal transformer (Vaswani et al., 2017). It adds a learned embedding of the action to its hidden state after every block. They learn from the proprioception of the 1051 training demonstrations. They differ in training objective, training length, seed and negative control (Appendices E and M). The objectives are next state MSE, alone or with an auxiliary loss, and delta prediction, which predicts the change in state (Nagabandi et al., 2018). One auxiliary loss, AGM (action gradient maximization), rewards response size. The other predicts the action from the current state and the model's predicted next state (inverse dynamics), at weights 0.001, 0.01 and 1. The negative controls break the link between action and outcome during training, for example by shuffling the actions. We also test two groups of released models. The first is every single task DeepMind Control model released by the TD-MPC2 authors, 117 models over 39 tasks with 3 seeds each. The second is the 26 released DIAMOND models, which predict Atari frames.

Table 1: Size probes cannot detect a response that points the wrong way, and retrieval prefers it. We transform the responses of 15 trained LIBERO models and report means. The simulator columns are the truth. D random and D logged are the true direction under random perturbations and logged swaps, where $+1$ means the prediction points exactly the way the environment moves, 0 means unrelated and -1 means opposite. Size is the predicted response size divided by the true one, so 1 is correct. Cf. error is the median relative error of the predicted effect, and miss is the final distance to a goal 4 cm away under model predictive control. The probe columns show what each probe concludes, and the arrows show which way it counts as more faithful. Retrieval is the mean rank of the logged action among 16 candidates, with chance at 7.5.

	simulator ground truth				control	probe			
model	D random ↑	D logged ↑	size	cf. error ↓	miss (cm) ↓	C_1 ↑	Task 1 ↑	ret. rank ↓	matched ↑
trained model	+0.312	+0.810	1.94	1.82	0.7	0.0353	49.8	0.92	+0.756
reflected response	-0.312	-0.810	1.94	2.39	32.0	0.0353	49.8	0.79	-0.756
rotated response	-0.013	+0.028	1.94	2.13	24.9	0.0353	49.8	0.41	+0.038
response $\times 0.25$	+0.312	+0.810	0.49	0.98	1.7	0.0022	49.8	4.14	+0.756
response $\times 4$	+0.312	+0.810	7.78	7.14	1.2	0.5643	49.8	0.02	+0.756
no response	+0.000	+0.000	0.00	1.00	4.7	0.0000	0.0	7.81	+0.000

Statistics. We compare a probe with the truth by the Spearman rank correlation across models (Spearman, 1904). A probe and its truth always use the same items. Intervals are 95% bootstrap intervals (Efron, 1979) that resample the independent unit, which is the model with its transformed copies, the task for TD-MPC2, or the objective for the 30 models that differ only in objective and seed. We compare training objectives by paired t tests over seeds with Holm correction (Holm, 1979).

Research questions. RQ1 asks whether probes notice a response that we turn the wrong way (Section 5.1). RQ2 asks whether probes rank ordinary models the way the simulator does (Section 5.2). RQ3 asks how accurate the two direction probes are (Section 5.3). RQ4 asks whether direction matters for training and for control (Section 5.4).

5 Results

Each subsection answers one research question. It describes the behavior we see and ends with the takeaway in italics. Appendix C gives every number behind these summaries.

5.1 RQ1. Do probes notice a response that points the wrong way?

We transform the responses of 15 trained LIBERO models (Liu et al., 2023) and ask each probe what it concludes (Table 1). Reflection reverses the true direction of every model, from $+0.31$ to -0.31 under random perturbations and from $+0.81$ to -0.81 on logged swaps. The two size probes give the reflected and rotated models exactly the same score as the originals, as Proposition 1 predicts. Retrieval ranks the reflected models above the originals, and the rotated models higher still. Its best score goes to the models whose response we multiply by 4, which leaves direction unchanged and makes the predicted effect much less accurate. The matched probe follows the true direction in every row. It flips sign under reflection, drops to near zero under rotation and does not move under rescaling. The released TD-MPC2 and Atari models behave the same way (Tables 4 and 6).

Takeaway. A probe that reads only response size cannot tell a model from a copy that responds in the opposite direction, and retrieval even prefers that copy. The probes that read direction pass this test.

5.2 RQ2. Do probes rank ordinary models the way the simulator does?

Passing RQ1 shows that a probe can detect direction. Real models differ in direction and size at the same time, so a useful probe must also rank them. Table 2 compares each probe's ranking of models with the simulator's.

On LIBERO, retrieval looks good across all 92 models. This is mostly because it separates models that learned to use the action from control models that did not. The harder test is the 30 models that differ only in training objective and seed (Table 9). There the matched probe orders the six objectives exactly as the simulator does, and retrieval's order shows no relation to it. Across the 30 models the

Table 2: The matched probe tracks the true direction in all three groups of models, while the size probes, retrieval and the cycle error fail once transformed copies are included. Each entry is the Spearman correlation across models between a probe and the true direction. A value of $+1$ means the probe orders the models exactly as the simulator does, 0 means no relation and -1 means the reverse order, so higher is better. We negate ranks and errors so that this holds for every row. As trained uses the models as they are, and transformed adds their five transformed copies. The last row scores each transformed copy $+1$, -1 or 0 as its transform keeps, reverses or removes direction, using no data and no model. A probe near this row has only shown that it detects the transforms. On Atari the logging policy is uniform random, so direction uses logged swaps. Empty cells are probes that do not apply. Discrete Atari actions have no small and large version of one action, and we drew real alternatives for retrieval on LIBERO only. Table 7 gives 95% bootstrap intervals.

	LIBERO, 92 models we trained				TD-MPC2, 117 released		Atari, 26 released	
	random perturbations		logged action swaps		random perturbations		logged action swaps	
probe	as trained	transformed	as trained	transformed	as trained	transformed	as trained	transformed
zero action divergence	+0.62	+0.17	+0.62	+0.26	-0.25	+0.15	+0.26	+0.20
WSP Task 1	+0.07	+0.10	-0.02	+0.12	+0.81	+0.31		
retrieval	+0.83	+0.15	+0.90	+0.25	+0.88	+0.10	+0.85	-0.14
retrieval, real negatives	+0.86	+0.06	+0.91	+0.17				
inverse model, cycle error	+0.12	+0.11	+0.55	+0.15	+0.86	+0.22	+0.16	+0.29
inverse model, direction	+0.94	+0.94	+0.98	+0.89	+0.83	+0.94	+0.74	+0.75
matched probe (ours)	+0.88	+0.78	+0.99	+0.98	+0.83	+0.97	+0.94	+0.94
reference, no data or model		+0.64		+0.67		+0.91		+0.92

matched probe reaches 0.92 and retrieval 0.14. Choosing a model by retrieval would pick one with below median direction, while the matched probe picks one within 0.004 of the best.

On the released TD-MPC2 models, retrieval and the matched probe track direction about equally well. Among the three seeds of each task, retrieval is somewhat better, but the difference is not significant. On the released Atari models the matched probe tracks direction across games at 0.94 and retrieval at 0.85, and this difference is not significant either. With transformed copies included, the size probes, retrieval and the cycle error all fall to 0.31 or below. The direction probes score higher, but the reference row shows that symmetry alone accounts for much of that.

Takeaway. The matched probe ranks ordinary models well in all three groups. Among the seeds of one TD-MPC2 task it is weaker, and retrieval is slightly ahead. Retrieval ranks the released models well but fails on the LIBERO models that differ only in training objective.

5.3 RQ3. How accurate are the direction probes?

For the matched probe, we ran the simulator on the same swaps the probe uses on LIBERO. The logged estimate points close to the true response, with a median cosine of 0.90, although the two start states of a pair are never identical. It gets less accurate as the two start states move apart. On these pairs the probe averages 0.76 where the truth averages 0.81. So it is close, but where the data and the truth disagree it can reward a model for agreeing with the data.

The inverse check scored by direction does about as well as the matched probe on LIBERO and TD-MPC2. There the inverse model is accurate and close to linear at the scale of the responses. On Atari frames the inverse model recovers the action at only 1.8 times chance, as a median over games. There the check falls clearly below the matched probe, and it no longer flips sign under reflection, which Proposition 8 allows for a nonlinear inverse model. Scored by its usual cycle error, the check reads size and behaves like the size probes (Tables 2 and 11).

Takeaway. The logged estimate is accurate enough to rank models by direction. The inverse check scored by direction works as well where its inverse model is accurate and close to linear, as with continuous actions. Elsewhere the matched probe is the safer choice.

5.4 RQ4. Does direction matter for training and for control?

Training on a probe's criterion. If a probe measures faithfulness, training a model to raise the probe should raise faithfulness. We add each probe's criterion to the MSE loss and judge the trained models only with the simulator (Table 3). Retrieval's own criterion, a contrastive loss, improves the retrieval

Table 3: Training on retrieval's criterion makes models less faithful, and matched swap alignment makes our LIBERO models more faithful. Each row changes the MSE objective, and each value is the mean change from the MSE baseline over 5 paired seeds. Arrows show which change is better, and size has no arrow because 1 is correct and the baseline is already above 1. A negative change in retrieval rank means retrieval scores the model as more faithful. An asterisk marks a Holm adjusted paired t test below 0.05 within the column (Holm, 1979). The contrastive loss is retrieval's criterion and AGM rewards response size. Indented rows are ablations of matched swap alignment. Appendix K gives the protocol and every weight.

objective	Δ D random ↑	Δ D logged ↑	Δ cf. error ↓	Δ size	Δ ret. rank	Δ val. MSE ($\times 10^{-4}$) ↓
delta prediction	+0.050*	+0.027*	+0.113	+0.134	+0.15	-1.64*
+ AGM (size)	+0.001*	0.000	+0.008*	+0.009*	-0.10*	+0.05*
+ contrastive ($w = 0.01$)	-0.097*	-0.239*	+0.465*	+0.228*	-0.43*	+5.94*
+ contrastive ($w = 0.1$)	-0.133*	-0.291*	+0.414*	+0.138*	-0.75*	+14.39*
+ matched swap alignment ($w = 0.01$)	+0.124*	+0.046*	-0.152*	-0.011	-0.40*	-0.22*
random partner ($w = 0.01$)	+0.082*	+0.020*	-0.064*	+0.016	-0.11	-0.06*
shuffled targets ($w = 0.01$)	-0.061*	-0.164*	+0.066*	-0.005	+0.78*	+0.94*
+ squared error imputation ($w = 1$)	+0.040*	+0.037*	-0.085*	-0.035*	-0.06	-0.78*

score a lot and makes the models less faithful under both kinds of action change. Their responses also grow, as Proposition 3 allows. AGM, which rewards response size, barely changes direction (by 0.001), yet retrieval rates it as better.

Matched swap alignment. We also turn the matched probe into a training loss, using pairs from the training demonstrations only (Appendix K). On our LIBERO models it raises direction under random perturbations by 42% on average, with a gain in every seed, and prediction error does not get worse. In the ablations, a random partner keeps two thirds of the gain and shuffled targets make direction worse. The gain does not transfer to released models. When we fine tune the released TD-MPC2 models with the same loss, direction improves in only 3 of 39 tasks at weight 0.01 (Table 14). The logged estimate there is less accurate than the models themselves, 0.61 against 0.87 in cosine with the simulator. So the loss pulls the models toward a worse target. A rise in held out prediction error warns of this without a simulator.

Control. We use model predictive control to move the gripper to a point 4 cm away (Appendix Q). Errors of size cost little, because feedback corrects them. A response four times too large or too small misses by 1.2 or 1.7 cm, against 0.66 cm for the trained models. Errors of direction are costly. A reflected response sends the gripper 32 cm away and a rotated one 25 cm (Figure 5). Across all models and transforms, the true direction, the matched probe and the inverse check predict the miss, and the size probes and retrieval do not. This mostly reflects the transforms themselves, since a reference that scores each transform $+1$, -1 or 0 without any data reaches 0.91. Among the 15 models as trained the misses are small, and no probe predicts them significantly. Matched swap alignment does not improve control.

Takeaway. Optimizing retrieval makes models less faithful, so retrieval should not be a training target. Control needs the right direction far more than the right size. In our two settings, training on the logged estimate helped where the estimate was about as accurate as the model and hurt where it was much less accurate.

6 Related work

Checking whether world models follow actions. World models can ignore or distort actions (Co et al., 2026; Shi et al., 2026; Wei et al., 2026), especially actions unlike the expert demonstrations they learned from (Chen et al., 2026b). The latents of video world models can also carry little information about the action (Yeom et al., 2026). Measures that see the direction of a response compare it with an executed reference. Examples are the flow alignment of Co et al. (2026), the SE(3) trajectory alignment of Chen et al. (2026b) and the matched interventions of Vakalis (2026). Measures that do not compare a response with an executed one mostly read size. Examples are drift under the zero action (Shi et al., 2026), the calibration ratio of Co et al. (2026), which uses a simulator only for a reference ratio, and empowerment (Klyubin et al., 2005). Genie's controllability score (Bruce et al., 2024) compares predictions with recorded frames, as retrieval does. Evaluations of driving world models score the trajectory that an inverse model reads off generated video (Gao et al., 2024; Arai

et al., 2024), and Liu et al. (2026) check cycle consistency through an inverse model. Other work evaluates action following with judges grounded in human annotations (Yang et al., 2026), with the consistency of composed actions (Wang et al., 2026b) or with admissibility tests (Oefinger et al., 2026), or studies visual perturbations (An et al., 2026) and design choices (Huang et al., 2026).

We are not the first to observe that world models can ignore actions, and two concurrent works are closest. Vakalis (2026) audits latent world models, including released TD-MPC2 checkpoints, against matched environment interventions. Chen (2026) audits robot video models against recorded futures and also finds that response size does not show movement toward them. Our probes need no environment, and our results explain which probes fail and why.

Counterfactual ground truth and training on action effects. Wang et al. (2026a) argue for judging world models by counterfactual controllability. Zhang et al. (2026b) build a driving benchmark with matched counterfactual outcomes in simulation, and Shi et al. (2026) a benchmark with several actions from the same state. During training, Chen et al. (2026a) learn action effects from synchronized counterfactual rollouts, and Chen et al. (2026b) align predicted changes under action interventions with the changes in ground truth futures. Other objectives predict state deltas (Nagabandi et al., 2018), regularize a latent Jacobian (Fang et al., 2024), or align models with control (Hutson et al., 2024; Parthasarathy et al., 2025). In causal inference, nearest neighbors impute missing counterfactual outcomes during training (Johansson et al., 2016; Schwab et al., 2018) and score effect models (Shalit et al., 2017).

Matched swap alignment applies the nearest neighbor idea to a world model's response to actions. It scores the response by direction and needs no simulator.

Checking a measurement. Control tasks (Hewitt & Liang, 2019), randomization checks (Adebayo et al., 2018) and negative controls (Lipsitch et al., 2010) test whether a measurement tracks its target. Model selection for heterogeneous treatment effects faces the same problem, because the true effect is never observed. It scores estimators against matched treated and control pairs (Rolling & Yang, 2014), an idea related to matching estimators (Rubin, 1973; Abadie & Imbens, 2006), or against plug in estimates (Schuler et al., 2018; Saito & Yasui, 2020; Curth & van der Schaar, 2023; Mahajan et al., 2024).

The matched probe applies this model selection idea to a world model's response to actions. It scores direction only, because an estimate built from two different states need not have the right size.

7 Limitations and conclusion

The matched probe needs logged data with different actions from similar states. No probe that reads the model only at the logged actions can certify direction away from them (Corollary 5), and planners also query there. We cannot bound the error of the logged estimate without assuming that the state change under a fixed action varies slowly with the start state. The inverse check needs an accurate inverse model, which we could not fit on Atari frames. On TD-MPC2 we measure the true response in each model's own latent space. Our pixel models predict Atari games, and we did not test robot video models. Matched swap alignment helped only the models we trained and did not improve control.

So can we check whether a world model follows actions without a simulator? Yes, but only for actions supported by the logged data. Across 235 world models, we show that using matched state pairs allows us to check response direction on the actions the data contain, but not with probes that read only response size, and not away from the logged actions. We recommend testing any action probe on reflected and rescaled copies of a model, and reporting direction separately from size.

Reproducibility statement

Every number in this paper is regenerated from saved checkpoints and simulator runs by the released scripts, and every table is generated from the resulting files (Appendix E). The supplementary code contains the simulator code, the probes, the response transforms, the loader for released models and its checks, and the Lean proofs. Appendix A defines every probe, Appendix F gives the protocol for the released models and Appendix Q the control experiment.

AI USE STATEMENT

Generative AI was used for language polishing, literature retrieval/discovery and assisting in the formalization and translation of proof steps into Lean 4. All literature was independently verified, all mathematical proofs were derived, audited, and mechanically verified by the Lean 4 compiler.

REFERENCES

Alberto Abadie and Guido W. Imbens. Large sample properties of matching estimators for average treatment effects. *Econometrica*, 74(1):235–267, 2006. doi: 10.1111/j.1468-0262.2006.00655.x.

Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, 2018.

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in Atari. In *Advances in Neural Information Processing Systems*, 2024.

Guo An, Zijing Wu, Honghua Dong, Yuhao Yan, Zixuan Gui, Haochong Chen, Shanzhao Ruan, Xiang Wang, Yurong Ling, and Qi Tian. Diagnosing JEPA world models with action-conditioned predictive consistency. arXiv preprint arXiv:2608.12939, 2026.

Hidehisa Arai, Keishi Ishihara, Tsubasa Takahashi, and Yu Yamaguchi. ACT-Bench: Towards action controllable world models for autonomous driving. arXiv preprint arXiv:2412.05337, 2024.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013. doi: 10.1613/jair.3912.

Jake Bruce, Michael Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, Yusuf Aytar, Sarah Bechtle, Feryal Behbahani, Stephanie Chan, Nicolas Heess, Lucy Gonzalez, Simon Osindero, Sherjil Ozair, Scott Reed, Jingwei Zhang, Konrad Zolna, Jeff Clune, Nando de Freitas, Satinder Singh, and Tim Rocktäschel. Genie: Generative interactive environments. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4603–4623, 2024.

Delong Chen, Willy Chung, Yejin Bang, Ziwei Ji, and Pascale Fung. WorldPrediction: A benchmark for high-level world modeling and long-horizon procedural planning. arXiv preprint arXiv:2506.04363, 2025.

Junlin Chen, Ruijie Wang, and Jianxin Li. Counterfactual quotient models: Learning what actions change, not what the world does. arXiv preprint arXiv:2608.22092, 2026a.

Sixiang Chen, Jiaming Liu, Jixian Wu, Yichen Guo, Tinghao Wang, Siyuan Qian, Hao Chen, Jiajun Cao, Jian Tang, and Shanghang Zhang. Do robotic world models really follow actions? Diagnosing and aligning action-conditioned generation for policy learning. arXiv preprint arXiv:2608.24885, 2026b.

Yinuo Chen. Action response is not factual direction: A prospective audit of robot video world models. engrXiv preprint, doi:10.31224/8244, 2026.

Peterson Co, Sicheng Hu, Chunxuan Jiao, Hongyang Cheng, Yulin Luo, Yijie Xu, Sixiang Chen, Zhongxia Zhao, Zihao Wang, DaFeng Chi, Peidong Liu, YuTong Chen, Henghua Liu, Zhihao Yuan, Huizhu Jia, Yuzheng Zhuang, Tianle Zhang, Liang Lin, Huajie Tan, and Shanghang Zhang. WorldSimProbe: Diagnosing simulator faithfulness in action-conditioned world models for embodied manipulation. arXiv preprint arXiv:2608.09298, 2026.

Alicia Curth and Mihaela van der Schaar. In search of insights, not magic bullets: Towards demystification of the model selection dilemma in heterogeneous treatment effect estimation. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 6623–6642, 2023.

Leonardo de Moura and Sebastian Ullrich. The Lean 4 theorem prover and programming language. In *International Conference on Automated Deduction (CADE)*, pp. 625–635, 2021. doi: 10.1007/978-3-030-79876-5_37.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552.

Qiaoyi Fang, Weiyu Du, Hang Wang, and Junshan Zhang. Towards unraveling and improving generalization in world models. arXiv preprint arXiv:2501.00195, 2024.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems*, 2024.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, 2019. doi: 10.18653/v1/D19-1275.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Siqiao Huang, Partha Kaushik, Michael Chen, Hengkai Pan, Kaiwen Geng, Omar Chehab, Fernando Moreno-Pino, and Max Simchowitz. Nano world models: A minimalist implementation of future video prediction. arXiv preprint arXiv:2605.23993, 2026.

Miles Hutson, Isaac Kauvar, and Nick Haber. Policy-shaped prediction: avoiding distractions in model-based reinforcement learning. In *Advances in Neural Information Processing Systems*, 2024.

Fredrik D. Johansson, Uri Shalit, and David Sontag. Learning representations for counterfactual inference. In *International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 3020–3029, 2016.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłoś, Błażej Osiński, Roy H. Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for Atari. In *International Conference on Learning Representations*, 2020.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Diederik P. Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Alexander S. Klyubin, Daniel Polani, and Chrystopher L. Nehaniv. Empowerment: A universal agent-centric measure of control. In *IEEE Congress on Evolutionary Computation*, volume 1, pp. 128–135, 2005. doi: 10.1109/CEC.2005.1554676.

Marc Lipsitch, Eric Tchetgen Tchetgen, and Ted Cohen. Negative controls: A tool for detecting confounding and bias in observational studies. *Epidemiology*, 21(3):383–388, 2010. doi: 10.1097/EDE.0b013e3181d61eeb.

Bo Liu, Yifeng Zhu, Chongkai Gao, Yihao Feng, Qiang Liu, Yuke Zhu, and Peter Stone. LIBERO: Benchmarking knowledge transfer for lifelong robot learning. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023.

Yuejiang Liu, Fan Feng, Lingjing Kong, Weifeng Lu, Jinzhou Tang, Kun Zhang, Kevin Murphy, Chelsea Finn, and Yilun Du. World action verifier: Self-improving world models via forward-inverse asymmetry. arXiv preprint arXiv:2604.01985, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Divyat Mahajan, Ioannis Mitliagkas, Brady Neal, and Vasilis Syrgkanis. Empirical analysis of model selection for heterogeneous causal effect estimation. In *International Conference on Learning Representations*, 2024.

David Manheim and Scott Garrabrant. Categorizing variants of Goodhart's law. arXiv preprint arXiv:1803.04585, 2018.

Diganta Misra. Mish: A self regularized non-monotonic activation function. In *British Machine Vision Conference*, 2020.

Anusha Nagabandi, Gregory Kahn, Ronald S. Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7559–7566, 2018. doi: 10.1109/ICRA.2018.8463189.

Christian Oefinger, Finn Rasmus Schäfer, Korbinian Moller, Mattia Piccinini, and Johannes Betz. Validate the dream before you trust its verdict: Admissibility for world-model simulators. arXiv preprint arXiv:2607.07196, 2026.

Markus Ojala and Gemma C. Garriga. Permutation tests for studying classifier performance. *Journal of Machine Learning Research*, 11(62):1833–1863, 2010.

Arjun Parthasarathy, Nimit Kalra, Rohun Agrawal, Yann LeCun, Oumayma Bounou, Pavel Izmailov, and Micah Goldblum. Closing the train-test gap in world models for gradient-based planning. arXiv preprint arXiv:2512.09929, 2025.

Craig A. Rolling and Yuhong Yang. Model selection for estimating treatment effects. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 76(4):749–769, 2014. doi: 10.1111/rssb.12043.

Donald B. Rubin. Matching to remove bias in observational studies. *Biometrics*, 29(1):159–184, 1973. doi: 10.2307/2529684.

Reuven Rubinstein. The cross-entropy method for combinatorial and continuous optimization. *Methodology and Computing in Applied Probability*, 1(2):127–190, 1999. doi: 10.1023/A:1010091220143.

Yuta Saito and Shota Yasui. Counterfactual cross-validation: Stable model selection procedure for causal inference models. In *International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8398–8407, 2020.

Alejandro Schuler, Michael Baiocchi, Robert Tibshirani, and Nigam Shah. A comparison of methods for model selection when estimating individual treatment effects. arXiv preprint arXiv:1804.05146, 2018.

Patrick Schwab, Lorenz Linhardt, and Walter Karlen. Perfect match: A simple method for learning representations for counterfactual inference with neural networks. arXiv preprint arXiv:1810.00656, 2018.

Uri Shalit, Fredrik D. Johansson, and David Sontag. Estimating individual treatment effect: Generalization bounds and algorithms. In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3076–3085, 2017.

Yuhong Shi, Zhenhao Chu, Jie Wei, Jun Hao, Jianyi Liu, and Jingwen Fu. Overcoming statistical bias in action-controllable world models. arXiv preprint arXiv:2608.04653, 2026.

C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101, 1904. doi: 10.2307/1412159.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. DeepMind control suite. arXiv preprint arXiv:1801.00690, 2018.

The mathlib Community. The Lean mathematical library. In *Proceedings of the 9th ACM SIGPLAN International Conference on Certified Programs and Proofs*, pp. 367–381, 2020. doi: 10.1145/3372885.3373824.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012. doi: 10.1109/IROS.2012.6386109.

Saran Tunyasuvunakool, Alistair Muldal, Yotam Doron, Siqi Liu, Steven Bohez, Josh Merel, Tom Erez, Timothy Lillicrap, Nicolas Heess, and Yuval Tassa. dm_control: Software and tasks for continuous control. *Software Impacts*, 6:100022, 2020. doi: 10.1016/j.simpa.2020.100022.

Donna Vakalis. The intervention gap in latent world models. arXiv preprint arXiv:2608.29998, 2026.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Xin Wang, Wenxuan Liu, Tongtong Feng, and Wenwu Zhu. Autonomous video generation with counterfactual controllability for self-evolving world models. arXiv preprint arXiv:2606.24152, 2026a.

Zijie Wang, Wei Zhang, Weiming Zhang, Fanqi Zhang, Xiao Tan, Yipeng Qin, and Guanbin Li. World models as group actions. arXiv preprint arXiv:2605.24578, 2026b.

Yufan Wei, Kun Zhou, Lingjun Mao, Zijun Zhang, Ziming Xu, Ziqiao Xi, Shuang Liang, Ruobing Han, Yuchen Yan, Xinyue Wang, Fan Feng, and Biwei Huang. Causally debiased latent action model for embodied action conditioned world models. arXiv preprint arXiv:2607.09185, 2026.

Tianzhuo Yang, Zihan Shen, Zirui Mi, Zhaoyi Zhang, Jiayi Zhou, Jiaming Ji, Juntao Dai, Jiawei Chen, Boyuan Chen, and Yaodong Yang. MiraBench: Evaluating action-conditioned reliability in robotic world models. arXiv preprint arXiv:2605.29360, 2026.

Jewon Yeom, Hanseul Kim, Jeongjae Park, Sungmok Jung, Jaejin Lee, and Taesup Kim. What makes video world model latents action-relevant: Prediction over reconstruction. arXiv preprint arXiv:2606.07687, 2026.

Jiahan Zhang, Muqing Jiang, Nanru Dai, Taiming Lu, Arda Uzunoglu, Shunchi Zhang, Yana Wei, Jiahao Wang, Vishal M. Patel, Paul Pu Liang, Daniel Khashabi, Cheng Peng, Rama Chellappa, Tianmin Shu, Alan Yuille, Yilun Du, and Jieneng Chen. World-in-world: World models in a closed-loop world. In *International Conference on Learning Representations*, 2026a.

Jiaru Zhang, Can Cui, Yi Xu, Xin Ye, Ruqi Zhang, and Ziran Wang. How can driving world models do counterfactual prediction? arXiv preprint arXiv:2608.11601, 2026b.

Yuke Zhu, Josiah Wong, Ajay Mandlekar, Roberto Martín-Martín, Abhishek Joshi, Kevin Lin, Abhiram Maddukuri, Soroush Nasiriany, and Yifeng Zhu. robosuite: A modular simulation framework and benchmark for robot learning. arXiv preprint arXiv:2009.12293, 2020.

A EXACT PROBE DEFINITIONS

We compute every quantity in the space the models are trained in. This is normalized proprioception on LIBERO and the model's own latent space on TD-MPC2. LIBERO rollouts are 8 steps long and open loop, and TD-MPC2 rollouts are 3 steps long.

Zero action divergence. $C_1 = \text{mean}\,\|P(A) - P(0)\|^2$ is the squared distance between the rollout under the logged actions and the rollout under a null action. The null action is zero in the normalized units of the LIBERO models, which is the mean action of the data. It is zero on TD-MPC2, and on Atari it is NOOP, action 0 of every game's action set. Drift under the zero action (Shi et al., 2026) is a related measure. It measures motion within the null action rollout itself, and our results do not cover it.

WorldSimProbe Task 1. Let D_s and D_l be the mean squared differences between the logged rollout and the rollouts under the small and the large perturbation. Let $D_s^\star$ and $D_l^\star$ be the same quantities in the simulator. The item score is the authors' triangular function of D_s/D_l against $D_s^\star/D_l^\star$. It is 1 when the two ratios are equal and falls linearly to 0 at 0 and at 1. As in the authors' code, we drop items where the simulator's large response does not exceed its small one. The reported value is 100 times the mean.

Logged action retrieval. There are 16 candidates, the logged action and 15 alternatives. The alternatives are either Gaussian perturbations with standard deviation 0.5 or actions drawn from the whole dataset. We rank the candidates by the root mean squared distance from their one step prediction to the recorded next state, and we break ties with jitter of relative size 10^{-9}. We report the mean rank of the logged action. Here 0 is best and chance is $(n-1)/2 = 7.5$ (`mean_rank_chance`).

Matched probe. For each held out position we take the held out segment from a different demonstration or episode of the same task whose standardized start state is nearest. We form $\hat{e}$ and r as in Section 3 over the full rollout. The probe is the mean cosine, and a model with no response scores 0.

Ground truth. D and $\text{D}_{\log}$ are mean cosines between the flattened predicted and true responses. M is the sum of the predicted response norms divided by the sum of the true ones. The counterfactual error is the median of $\|r_f - r^\star\|/\|r^\star\|$.

B PROOFS

This appendix proves every formal result in the paper and states Propositions 7 and 8. Lean 4 with Mathlib v4.15.0 checks every result in this appendix, and the examples in the explanations of Section 3. The file `proofs/PaperResults.lean` states each proposition and the corollary as here and proves it, using lemmas from `proofs/WamProofs.lean`. Both files build with no errors and no unfinished proofs, and `#print axioms` reports only `propext`, `Classical.choice` and `Quot.sound` for every theorem. It contains 32 theorems in `WamProofs.lean` and 25 theorems and 12 lemmas in `PaperResults.lean`. After each proof we name its Lean theorems. The script `analysis/verify_math.py` also checks Propositions 3, 7 and 8 numerically.

Proposition 1. The idea is that an isometry keeps lengths. Since $r_f(0) = 0$, the response of f_Q is $r_{f_Q}(\delta) = Q\,r_f(\delta) - Q\,r_f(0) = Q\,r_f(\delta)$. An isometry keeps norms, so $\|r_{f_Q}(\delta)\| = \|r_f(\delta)\|$ for every δ. A size probe depends on the model only through these norms, so its score does not change. For $Q = -I$ each cosine changes sign, $\cos(-r, r^\star) = -\cos(r, r^\star)$, and so does their mean. Lean: `response_transformed`, `prop1_size_probe_invariant` and `prop1_reflection_direction`.

Proposition 2. The idea is that a linear response scales exactly with the action. Linearity gives $J(s\,d) = s\,Jd$, so $\|J(s\,d)\|^2 = s^2\|Jd\|^2$, and the ratio is s^2 because $Jd \neq 0$. Lean: `task1_ratio_linear`. The claim after it, that a constant beats every model we trained, is measured (Appendix N). Lean also proves `tri_self`, that the score is highest when the two ratios agree, and `tri_const_lower`, a lower bound on the score of a constant.

Proposition 3. The idea is to expand a squared distance. We have $f(H, A+\varepsilon) - Y = e + r$, and $\|e+r\|^2 - \|e\|^2 = 2e^\top r + \|r\|^2$. For the model f_{cI} the prediction at the logged action is unchanged and the response is $c\,r$, so the difference becomes $2c\,e^\top r + c^2\|r\|^2$. It is positive whenever $r \neq 0$ and $|c| > 2|e^\top r|/\|r\|^2$, for either sign of c. With finitely many alternatives, each with a nonzero response,

a large enough $|c|$ therefore puts the logged action first. Lean: `prop3_identity`, `prop3_scaled` and `prop3_large_response`.

Proposition 4. The idea is that the added term is zero wherever the data can look. At a logged action $a = A = \pi(H)$ the added term $B(H)(A - \pi(H))$ is zero, so $f_B(H, A) = f(H, A)$. Any loss of the prediction and the observed states then takes the same value on every logged sample, and so has the same expectation. The response is $f_B(H, A + \delta) - f_B(H, A) = r_f(\delta) + B(H)(A + \delta - \pi(H)) - B(H)(A - \pi(H)) = r_f(\delta) + B(H)\delta$. Lean: `offSupport_value` and `offSupport_risk` (the prediction and the expected loss), `prop4_response` (the response), and `offSupport_deriv` with `offSupport_unbounded` (the action Jacobian of f_B is that of f plus B, which a scaled B makes exceed any bound).

Corollary 5. The idea is to make the added term dominate the response. A data grounded probe sees only $f_B(H, A) = f(H, A)$, so it gives f and f_B the same score. Take $B(H) = c\,J^\star(H)$. Then $\cos(r_{f_B}(\delta), r^\star(\delta)) = \cos(r_f(\delta)/c + J^\star\delta,\, J^\star\delta)$ for $c > 0$. The cosine is continuous at the nonzero vector $J^\star\delta$, so this tends to 1 as $c \to \infty$ on every item, and so does the average $\mathrm{D}(f_B)$. With $B = -c\,J^\star$ the same argument gives -1. Lean: `cor5_data_grounded` and `cor5_direction_limits`.

Proposition 6. The idea is that a cosine ignores length and changes sign with its argument. The transforms use the logged actions a_i as reference, so the model f_{cI} answers the swap with $c\,r_i$. (i) A cosine does not change when one argument is multiplied by $c > 0$ (`cosSim_smul_pos`). (ii) It changes sign when one argument is negated (`cosSim_neg`). Both hold item by item and so for the mean. (iii) Write $\bar{x} = x/\|x\|$. Then $\cos(r, x) = \langle \bar{r}, \bar{x} \rangle$, and the Cauchy Schwarz inequality with $\|\bar{r}\| \le 1$ gives $|\cos(r, \hat{e}) - \cos(r, r^\star)| = |\langle \bar{r}, \bar{e} - \bar{r}^\star \rangle| \le \|\bar{e} - \bar{r}^\star\|$ (`cosSim_sub_le`). Expanding the square of a difference of unit vectors gives $\|\bar{e} - \bar{r}^\star\|^2 = 2 - 2\cos(\hat{e}, r^\star)$ (`normalized_dist_sq`). Lean: `prop6_scaled_swap` (the transform multiplies the answer to a swap by c), `prop6_i`, `prop6_ii`, `prop6_iii`, and `prop6_example` for the example after Proposition 6.

Proposition 7 (symmetric noise shrinks the matched probe)**.** *Suppose $r_i^\star \neq 0$ and $\hat{e}_i = r_i^\star + \xi_i$, where ξ_i is independent of r_i and its distribution is invariant under every orthogonal map that sends the line through $r_i^\star$ to itself. Isotropic noise is one example. Then $\mathbb{E}\,\mathrm{MP}(f) = \frac{1}{n}\sum_i \kappa_i \cos(r_i, r_i^\star)$, with $\kappa_i \in [0, 1]$ depending only on $r_i^\star$ and the noise. Two models scored on the same pairs are therefore ordered in expectation by the same κ weighted on support direction. This ordering is the ordering by $\mathrm{D}_{\log}$ when κ_i is the same positive number for every item.*

In words. Suppose the error of the logged estimate is equally likely to point in any direction around the truth. Then it pulls the probe toward zero, by a factor κ_i between 0 and 1 on each pair, but it favors no model. On the same pairs, a model with a better true direction therefore scores higher on average, as long as the factor is similar across pairs. Very large noise can push the factor to 0, and the probe then carries no information about the model.

Proof. The idea is that symmetric noise has no preferred direction except along the truth. Fix an item and write $v = r_i^\star \neq 0$, $u = v/\|v\|$ and $w = v + \xi$. Let G be the group of orthogonal maps that fix v. By hypothesis the law of ξ is invariant under G and under the reflection $x \mapsto x - 2\langle x, u\rangle u$, which sends the line through v to itself but not v to itself. Since $gv = v$ for $g \in G$, the law of w is G invariant, and so is the law of $w/\|w\|$ (defined as 0 at $w = 0$). Its mean m therefore satisfies $gm = m$ for every $g \in G$. The group G contains the map that fixes u and negates every vector orthogonal to u, so m has no component orthogonal to u. So $m = \kappa u$ with $\kappa = \mathbb{E}\langle u, w/\|w\|\rangle \le 1$. Without the reflection κ can be negative, since a point mass at $\xi = -2v$ is invariant under G and gives $\kappa = -1$. Now pair ξ with its reflection, which has the same law. Write $t = \langle u, \xi\rangle$, $c = \|\xi - tu\|^2$ and $h(x) = x/\sqrt{x^2 + c}$. The two values of $\langle u, w/\|w\|\rangle$ are $h(\|v\| + t)$ and $h(\|v\| - t)$. The function h is odd and increasing, so $h(\|v\| + t) + h(\|v\| - t) = h(\|v\| + t) - h(t - \|v\|) \ge 0$, and hence $\kappa \ge 0$. For any response r independent of ξ, $\mathbb{E}\cos(r, w) = \langle r/\|r\|, m\rangle = \kappa \cos(r, v)$. Averaging over items gives the statement. `verify_math.py` draws isotropic Gaussian noise at several scales and confirms the linear relation. Lean: `prop7_item` for one item, under invariance under the two reflections used above, `prop7_mean` for the average over items with noises of any dependence, `prop7_ordering` for the ordering, and `prop7_zero_example` for noise equal to $\pm 3r^\star$, which gives $\kappa = 0$.

Proposition 8 (when the inverse check reads only direction)**.** *Suppose g reads each consecutive pair of states in a rollout and $g(x, y) = W_1 x + W_2 y + b$ for matrices W_1, W_2 and a vector b. Then*

$\mathrm{IC}(f_{cI}) = \mathrm{IC}(f)$ *for every* $c > 0$ *and* $\mathrm{IC}(f_{-I}) = -\mathrm{IC}(f)$*, for rollouts of any length. For a nonlinear* g *neither holds in general.*

In words. A linear inverse model turns a rescaled or reflected rollout into an equally rescaled or reflected action change. So the direction form keeps the two properties of the matched probe. A nonlinear inverse model can map a small or reflected change to a different action, and the proof ends with an example where both properties fail.

Proof. The idea is that an affine map commutes with scaling differences. Write the rollout under a as $\hat{x}_0, \hat{x}_1, \ldots, \hat{x}_T$, where $\hat{x}_0$ is the last observed state and is the same for every a. The transform with the logged actions as reference turns the rollout under a' into $\hat{x}_t^A + T\Delta_t$, with $\Delta_t = \hat{x}_t^{a'} - \hat{x}_t^A$ and $\Delta_0 = 0$. For an affine g, step t of $g(\hat{X}_{a'}) - g(\hat{X}_A)$ becomes $W_1 T\Delta_t + W_2 T\Delta_{t+1}$, which is linear in $T\Delta$. So $T = cI$ multiplies the whole vector by c. Its cosine with $a' - A$ is unchanged for $c > 0$ and changes sign for $c = -1$. The same holds for an inverse model that reads $(x_t, x_{t+1} - x_t)$, which is also affine in the pair. For the nonlinear case take one state dimension, one step, $\hat{x}_0 = \hat{x}_1^A = 0$ and $g(x, y) = y^2 - y$. In one dimension the cosine is the product of the signs. Suppose the model responds to the action change $a' - A = 1$ with $\Delta_1 = 2$. Then g changes by 2 and $\mathrm{IC} = 1$. After rescaling by 0.25 the response is 0.5, g changes by -0.25, and $\mathrm{IC} = -1$. After reflection the response is -2, g changes by 6, and IC stays 1. Lean: `prop8_affine_step`, `prop8` and `prop8_nonlinear`.

`WamProofs.lean` also contains `mean_rank_chance`, used in Appendix A, and auxiliary results that the paper does not rely on. These are `agm_unbounded_below`, `agm_zero_grad_at_blindness`, `agm_zero_grad_batch`, `fd_second_moment` and `fd_expectation` (facts about an objective that rewards response size), `risk_gap` and `blind_never_beats` (squared error prefers a predictor that uses the action), and `did_cancels` and `ratio_not_invariant`.

C Detailed results

This appendix gives the numbers behind each summary in Section 5, question by question.

C.1 RQ1, response transforms

No probe that reads size notices a reflected response, and retrieval prefers it (Table 1). Reflection takes the true direction of each model from $+0.312$ to -0.312 under random perturbations and from $+0.810$ to -0.810 on logged swaps. Zero action divergence and the calibration score do not change in any printed digit, as Proposition 1 requires. Retrieval gives the reflected models a better mean rank (0.79) than the originals (0.92). It gives the rotated models, whose response is unrelated to the true one, a better rank still (0.41). Multiplying the response by 4 leaves direction unchanged and raises the counterfactual error from 1.82 to 7.14. Retrieval gives these models its best score (0.02), as Proposition 3 predicts. The matched probe follows the true direction in every row. It gives $+0.756$ for the trained models, -0.756 when reflected and $+0.038$ when rotated, and it does not change under rescaling. The released TD-MPC2 and Atari models show the same pattern (Tables 4 and 6), and Figure 6 shows it over a continuous range of transforms.

C.2 RQ2, ranking models as trained

Passing RQ1 shows that a probe can see direction. Real models differ in direction and size at the same time, so a probe must also rank them correctly. Table 2 gives the evidence.

LIBERO. Across all 92 LIBERO models as trained, retrieval tracks direction well (0.83 under random perturbations and 0.90 on logged swaps). Much of this comes from separating models that learned to use the action from control models that did not. The cleanest test uses the 30 models that differ only in training objective and seed (Table 9). On logged swaps the six objectives differ in direction, while seeds of the same objective barely do, with a standard deviation of 0.005. The matched probe orders the six objectives exactly as the simulator does (Spearman 1.00 over objective means), and retrieval does not (-0.09). Over all 30 models, with intervals that resample objectives, the matched probe tracks direction at 0.92 [0.61, 0.97] and retrieval at 0.14. Without inverse dynamics at weight 1, whose loss swamps the state loss, these become 0.86 and -0.49. This matters for choosing a model.

The model with the best retrieval score has below median direction under random perturbations (0.275 against a median of 0.304). The model with the best matched probe score is within 0.004 of the best.

With the transformed copies included, every size probe falls to 0.26 or below and the matched probe reaches 0.98. Symmetry explains much of this. The reference row of Table 2 uses no data and no model and still reaches 0.64 to 0.92 on the transformed columns. Passing the transforms shows that a probe can see direction. The columns for models as trained show how well it measures it.

Released TD-MPC2 models. TD-MPC2 (Hansen et al., 2024) has no decoder, so we measure the true response in each model's latent space. We encode the simulator's next states with the model's own encoder, which gives the model's training target (Appendix F). Direction averages 0.77, from 0.23 (walker walk backwards) to 1.00, and size 0.93. Reflection leaves zero action divergence and the calibration score unchanged and improves retrieval's mean rank from 1.72 to 0.86. Rotation improves it to 0.07. As trained, the calibration score correlates with direction (0.81) only because both vary with the task. Within tasks it drops to 0.14. Retrieval (0.88) and the matched probe (0.83) track direction about equally well. Within tasks, where the three models differ only in their seed, retrieval does better, 0.76 against 0.62. Under a bootstrap that resamples tasks this difference is not significant (retrieval minus matched probe 0.15 $[-0.06, 0.39]$).

Released Atari models. Alonso et al. (2024) released one diffusion world model per Atari 100k game (Kaiser et al., 2020), which predicts the next 64×64 frame. Emulator branches are exact. A uniform random policy gives 100 positions per game, and every action it takes is on the support (Appendix G). Direction averages 0.49 across games, from 0.09 to 0.99. Zero action divergence does not change under reflection or rotation. Retrieval again rewards both. It scores the reflected models at 0.09 of chance and the rotated ones at 0.03, against 0.54 as trained. As trained, the matched probe tracks direction across games at 0.94 [0.85, 0.98], retrieval at 0.85 [0.63, 0.95] and zero action divergence at 0.26. The matched probe's lead over retrieval, 0.09 $[-0.03, 0.29]$, is not significant.

C.3 RQ3, ACCURACY OF THE DIRECTION PROBES

The matched probe. We ran the simulator for the same swaps that the matched probe uses. The logged estimate and the simulator's response have a median cosine of 0.90, and 93% of pairs are above 0.5. The two start states of a pair are a median standardized distance of 0.63 apart. Accuracy falls as this gap grows, from a mean cosine of 0.92 for the closest quarter of pairs to 0.75 for the farthest. The worst case bound of Proposition 6(iii) has a median of 0.44 per item. On the same pairs, the probe's mean ($+0.756$) is within 0.054 of the on support truth ($+0.810$). If the estimate's errors were symmetric and independent of the model, Proposition 7 would predict a probe of about 0.69, using each pair's own estimate accuracy as its shrink factor. The probe is closer to the truth than that. So the estimate's errors are not symmetric about the truth, or not independent of the model. Where the data and the truth disagree, the probe can then reward a model for agreeing with the data.

Models are much more faithful on logged swaps ($\mathrm{D}_{\log} = 0.81$) than under random perturbations ($\mathrm{D} = 0.31$), which is consistent with Proposition 4. The two are correlated across all 92 models (0.88), mostly because the control models are poor under both. Figure 7 shows typical responses on logged swaps. Appendix P shows the matched probe rejecting adversarial models that have no dynamics but copy the logging policy.

The inverse check. We fit an inverse model g to logged transitions in each group of models, with held out R^2 0.95 on LIBERO. We ask it about the same action changes as each ground truth (Appendix J). Its cycle error reads size. It tracks random perturbation direction at 0.12 across the 92 models, and removing the response lowers the error for 26 of the 30 ordinary models. Scored by direction, the check passes the transforms wherever g is accurate and close to linear. On LIBERO it matches the matched probe on logged swaps (0.89 within the 30 models, against 0.92). Under random perturbations it is ahead across the 92 models, at 0.94 against 0.88. The paired difference of 0.07 is not significant when we resample the seeds of a configuration together ($[-0.01, 0.22]$). Within the 30 models its lead, 0.84 against 0.61, is not significant either. On Atari frames a small convolutional classifier recovers the action at a median 1.8 times chance. The direction form then tracks direction at 0.74 [0.48, 0.88], significantly below the matched probe (lead 0.20 [0.06, 0.42]). Reflection leaves it near zero (-0.005 against 0.138), which Proposition 8 allows for a nonlinear g. Appendix J gives the TD-MPC2 results, where the two direction probes are close. We suggest the direction form of the

inverse check where its inverse model is accurate and close to linear, as with continuous actions, and the matched probe otherwise.

C.4 RQ4, TRAINING AND CONTROL

Training on a probe's criterion. If a probe measures faithfulness, training a model to raise the probe should raise faithfulness. We add each probe's criterion to the MSE loss at weights fixed in advance and judge the trained models only with the simulator (Table 3). A contrastive loss that makes the logged action's prediction beat perturbed ones is retrieval's own criterion. At weight 0.1 it drives retrieval's mean rank from 0.90 to 0.15. At the same time, direction falls from 0.295 to 0.162 under random perturbations and from 0.801 to 0.510 on logged swaps. Responses also grow, as Proposition 3 allows. This is Goodhart's law (Manheim & Garrabrant, 2018). AGM, which rewards response size, leaves direction unchanged, yet retrieval rates it better by 0.10 in mean rank ($p = 0.004$). Appendix I compares the other objectives.

Matched swap alignment. We turn the matched probe into a training loss using pairs from the training demonstrations only. For a pair (p, q), r_{pq} is the model's one step response to swapping the two actions and $\hat{e}_{pq}$ is the logged estimate. We add $w\,(1 - \cos(r_{pq}, \hat{e}_{pq}))$ to the MSE loss (Appendix K). At weight 0.01 it raises direction under random perturbations from 0.295 to 0.419 in every seed, a gain of 42%. This is about 2.5 times the gain of delta prediction, the best existing objective we tried. It also raises direction on logged swaps and lowers the counterfactual error, and validation MSE does not get worse. The ablations show where the gain comes from. Shuffled targets make direction worse. A random partner from another demonstration keeps two thirds of the gain. Squared error imputation of the swapped next state, which adapts the nearest neighbor outcome term of Johansson et al. (2016), keeps a third. The matched probe overstates the gain, at $+0.073$ against a true $+0.046$, because the models also learn the errors of the estimate. So a model trained on the probe's criterion should not be judged by that probe alone.

Released models. For each task we took the released seed 1 TD-MPC2 model and fine tuned its latent dynamics on four of its own logged episodes. We did this with and without alignment, from the same weights and with the same batches. We scored direction against the simulator on two held out episodes (Appendix L). Alignment lowered direction by 0.048 at weight 0.01 and improved it in only 3 of 39 tasks. On the same swaps the logged estimate agrees with the simulator at 0.61 on average and the released models at 0.87. The models beat their training target in 36 of 39 tasks. On LIBERO, the estimate (0.82) edges the MSE models (0.80). Held out prediction error detects the problem without a simulator. Its change predicts the change in direction (Spearman -0.63 at weight 0.01). We therefore keep alignment only when it does not raise held out error.

Control. In each of 12 tasks we start from a held out state. Model predictive control then moves the gripper toward a point 4 cm away in a random direction (Appendix Q). With the 15 trained models of Table 1, the gripper ends 0.66 cm from the goal on average. A response four times too large or too small misses by 1.2 and 1.7 cm, because feedback corrects most errors of size. A reflected response sends the gripper 32 cm away and a rotated one 25 cm (Figure 5). Zhang et al. (2026a) also find that controllability matters more than visual quality for task success. Across the 90 combinations of model and transform, direction predicts the miss (Spearman 0.84). So do the matched probe (0.82) and the inverse check (0.87). The size probes, retrieval and the cycle error do not (0.16 or less). Most of this comes from the transforms, since a reference that scores each transform $+1$, -1 or 0 without any data reaches 0.91. The 15 models as trained miss by 0.3 to 1.1 cm. Delta prediction has the best on support direction and also misses least (0.48 cm against 0.69 and 0.80 cm), but three objectives are too few to test this. Within one objective, direction does not predict the miss (0.29, $p = 0.29$), and no probe predicts it significantly even with all 15 models pooled. Models trained with matched swap alignment at weight 0.01 do not control better. On the same goals they miss by 0.75 cm against 0.69 cm for their MSE baselines, which is not significant. Delta prediction, with the lowest validation error, misses by 0.48 cm.

D EMPOWERMENT AND GROUP ACTION CONSISTENCY

The argument of Proposition 1 extends, under conditions stated below, to two measures it was not written for. The first is empowerment, a long standing measure of how far actions can

steer the future. The second is the set of group action consistency conditions proposed in 2026. Lean checks both, `appD_reflection_cov`, `appD_rotation_det` and `appD_consistency`, and `analysis/verify_math.py` checks both numerically.

Latent empowerment. Let $S = \mathrm{Cov}[r_f(\delta)]$ be the covariance of a model's final step responses to action changes, and let E be its residual covariance at the logged action. The usual local linear Gaussian estimate of empowerment is $m = \frac{1}{2}\log\det(I + E^{-1}S)$. A response transform T sends S to $TST^\top$ and leaves E unchanged, because predictions at the logged actions do not change. For $T = -I$ we have $TST^\top = S$, so a reflection leaves m unchanged whatever E is. For a general isometry Q and isotropic $E = \sigma^2 I$, we have $\det(I + \sigma^{-2}QSQ^\top) = \det(Q(I + \sigma^{-2}S)Q^\top) = \det(I + \sigma^{-2}S)$, so m is unchanged there too. Rescaling does change m, which makes it a measure of size.

Group action consistency. Wang et al. (2026b) score a model with three distances. The first is the drift under the zero action, $d(P(0), s)$. The second is the residual after an action and its inverse, $d(P(a, -a), s)$. The third is the distance between the endpoints of two decompositions of the same net motion, $d(P(a^A), P(a^B))$. Take the zero action as the reference of the transform, so $P_T(a) = P(0) + T(P(a) - P(0))$. The first distance is then unchanged for every T, because $P_T(0) = P(0)$. The third is unchanged for every isometry, because $P_T(a^A) - P_T(a^B) = T(P(a^A) - P(a^B))$ and T keeps norms. For the second, write $u = s - P(0)$ for the drift and $v = P(a, -a) - s$ for the residual before the transform. After the transform the residual is $(T - I)u + Tv$. With no drift this is Tv, whose size every isometry keeps. With drift and exact recovery, a reflection leaves a residual of size $2\|u\|$. So a model that meets their first condition gets the same three scores after its response is reflected or rotated. The second condition can see a reflection only as far as the model already fails the first.

E Reproducibility

Restoring a MuJoCo state leaves two pieces of controller state untouched. The first is the orientation goal of the operational space controller, which the controller keeps whenever the rotation part of an action is exactly zero. The second is the gripper's internal command, which moves by a fixed step in the direction of the action's sign at each step. Both therefore carry over from whatever branch ran before. Before every branch we reset the controller goal to the restored pose. We also set the gripper command to its value in the demonstration, which we reconstruct by replaying that demonstration's own gripper action signs. This fix took the largest disagreement between two runs of the same branch from 0.15 to 0.008 in a coordinate, and the mean noise floor from 0.021 to 0.005.

The LIBERO backbone is a 4 layer pre norm causal transformer with hidden width 128, 4 heads and 0.81 million parameters. It adds a learned embedding of the action after every block. We train with AdamW (Loshchilov & Hutter, 2019) at learning rate 3×10^{-4} and weight decay 10^{-2}. The schedule has a 200 step warm up and then cosine decay, with batch size 64, for 5000 steps unless stated. Observations are 21 dimensional proprioception, and actions are 7 dimensional. Normalization statistics come from the training split. The splits are a deterministic hash of the demonstration identifier at 70, 15 and 15 percent. AGM uses weight 0.5 and perturbation 0.1. The 92 models are six objectives with 5 seeds (30), ten negative control configurations with 5 seeds (50), and next state MSE and delta prediction trained for 500 or 1500 steps with seeds 0 to 2 (12). The controls are sequence permutation, derangement, a per timestep shift and an action blind input for next state MSE and for delta prediction, and sequence permutation for AGM and for inverse dynamics at weight 1. Everything runs on one 8 GB laptop GPU, and a 5000 step training run takes 1 to 6 minutes. Simulation uses MuJoCo 3.2.7, robosuite 1.4.1 (Zhu et al., 2020) and LIBERO with two compatibility patches applied at import time, and dm_control 1.0.27 (Tunyasuvunakool et al., 2020) for the released models. Matched segments come from the validation and test splits only.

We measure LIBERO ground truth in the proprioceptive state because images did not work. Our renderer does not reproduce the dataset's images exactly, and a VAE (Kingma & Welling, 2014) trained with a small KL weight amplifies such small pixel differences in its latents, so they do not cancel when we subtract two latents.

A script regenerates every number in this paper from saved checkpoints, and every table is generated from the resulting files. We recommend this practice for a reason we learned first hand. In our own pipeline, most cached probe values that had not been rebuilt disagreed with a regeneration.

F RELEASED TD-MPC2 MODELS

Table 5 lists the results per task, and Figure 2 plots them.

Table 4: The released TD-MPC2 models show the pattern of Table 1. The size probes do not change under reflection or rotation, retrieval rates both as better, and the matched probe follows the true direction. We transform 117 released TD-MPC2 world models, 39 DeepMind Control tasks with 3 seeds each, and report means. We measure ground truth in each model's own latent space, as the encoding of the simulator's actual next states, which is the target the model was trained to predict. Direction runs from -1 to $+1$, size is 1 when correct, and arrows show which way each probe counts as more faithful.

	simulator		probe			
model	direction ↑	size	C_1 ↑	Task 1 ↑	ret. rank ↓	matched ↑
trained model	+0.769	0.93	0.0067	89.0	1.72	+0.584
reflected response	-0.769	0.93	0.0067	89.0	0.86	-0.584
rotated response	+0.000	0.93	0.0067	89.0	0.07	-0.001
response $\times 0.25$	+0.769	0.23	0.0004	89.0	4.06	+0.584
response $\times 4$	+0.769	3.72	0.1065	89.0	0.19	+0.584
no response	+0.000	0.00	0.0000	0.0	7.51	+0.000

We load the encoder, latent dynamics and policy prior of each checkpoint with the architecture of the authors' code, with LayerNorm (Ba et al., 2016), Mish (Misra, 2020) and SimNorm with groups of 8. We check each load by reward prediction. Ten checkpoints use an older module layout than the released code. They store each MLP as a flat sequence of modules, with a parameter free first module in the encoder. These are cartpole balance seeds 1 and 2, cartpole swingup seed 1, cartpole swingup sparse seeds 1 and 2, humanoid run, stand and walk seed 3, pendulum swingup seed 2, and walker walk backwards seed 1. We put a symmetric log of the observation in that first position. With it, the reward head of each of these checkpoints correlates with the simulator's reward at 0.75 to 1.00, for the 9 whose reward varies during the check. Without it, the correlation is 0.04 to 0.51 for the humanoid checkpoints and 0.12 for walker walk backwards. On the cartpole and pendulum checkpoints, the symmetric log also reduces the one step latent consistency error by 3.6 to 63 times. We apply the symmetric log to these ten only.

Over all 117 checkpoints, reward prediction after 2 latent steps has a median correlation of 0.975. It is above 0.78 for 88% of the 110 checkpoints with a defined correlation. The lowest, 0.41, is reacher three hard seed 2, a task whose reward is nearly all or nothing. Seven checkpoints give no defined correlation after 2 latent steps, because a reward or its prediction is constant over the check. These are four sparse cartpole checkpoints and three finger turn easy checkpoints. We step the environment below dm_control's episode logic with action repeat 2. We take branches before an episode is reset, because some tasks keep their target in the simulator's model parameters, which a state restore does not reset. The logged data come from the seed 1 model's policy prior with Gaussian exploration noise of standard deviation 0.3. Random perturbations have standard deviation 0.2 and 0.5. We draw matched pairs across episodes by standardized observation distance.

G RELEASED ATARI MODELS

We use every released DIAMOND checkpoint for Atari 100k (Kaiser et al., 2020), 26 games. We load the denoiser with the authors' architecture and their released configuration, which uses 4 conditioning frames, 3 denoising steps and Euler sampling (Karras et al., 2022). The data come from a uniform random policy. The matched probe needs different actions from similar states, and a uniform random policy provides them by design. Per game we take 100 branch positions, at least 120 steps after a reset, across 7 to 16 episodes, each started from its own seed.

At each position we clone the emulator state. We run branches of 4 steps under the logged actions and under the actions of the matched position, and we also roll the model out under the null action. Restoring the cloned state makes a branch exact, and a repeat differs by 0 in every pixel in all 26 games. A branch that ends the episode is unusable, because the frame skip wrapper then returns a stale frame. We drop such positions (17 over all games). We roll the model out with the same

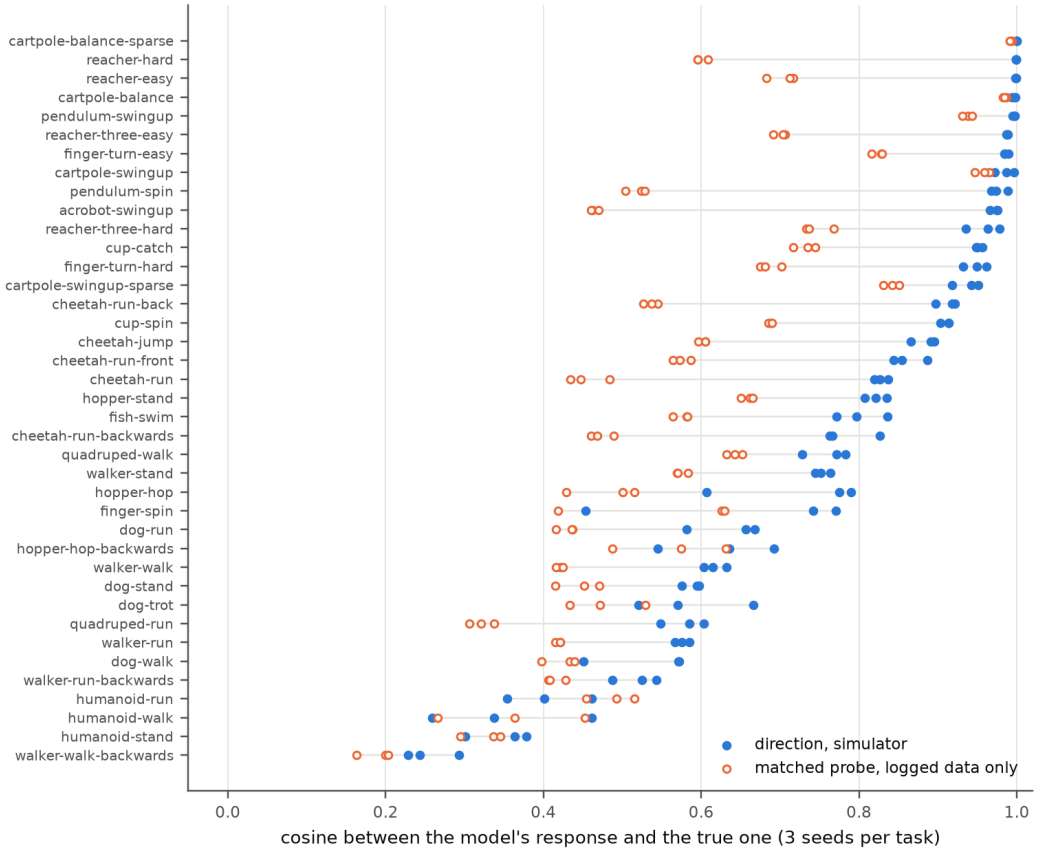

Figure 2: The matched probe orders the released TD-MPC2 models much as the simulator does, but it reads lower. Each row is one task and each marker one of its 3 seeds, sorted by the task's mean direction. The horizontal axis is the cosine between the model's response and the true one. A value of 1 means the prediction moves exactly as the environment does, and 0 means the two are unrelated. Filled markers show the true direction from the simulator and open markers the matched probe, which uses logged data only. The probe sits lower for 112 of the 117 models, as symmetric noise in the logged estimate would cause (Proposition 7), by an amount that varies between tasks. Across models it correlates with the truth at 0.83.

noise for every branch from a position, so a repeat is bit identical. Scoring the models again with independent noise per branch changes mean direction by 0.003, changes each game's direction by 0.004 on average, and changes any game by at most 0.047. So this choice makes the measurement repeatable and does not drive the result. A matched pair is the nearest position from a different episode, by distance between context frames, excluding pairs whose logged action sequences coincide. Retrieval ranks the logged action among the game's actions, each listed once, and counts a tie as half a rank. So a model with no response scores exactly chance. The rotation transform is a random permutation of the response coordinates. This is an isometry, and it avoids a dense 12288×12288 orthogonal matrix.

Repeating the collection on 6 games with a different seed moves direction by 0.051 on average and ranks the games almost identically (Spearman 0.94). It moves the matched probe by 0.027 (0.83).

H Confidence intervals and the number of matched pairs

Figure 3 shows the LIBERO and TD-MPC2 columns of Table 2 with their 95% bootstrap intervals, and Table 7 lists the intervals for every column. The matched probe barely changes when it averages

Table 5: Per task results on the released TD-MPC2 models, as trained, mean over 3 seeds. True direction varies widely between tasks, which is why correlations across all 117 models partly reflect the task. Direction is shown with its standard deviation over seeds.

task	direction	size	C_1	Task 1	ret. rank	matched
acrobot-swingup	+0.972±0.006	1.00	0.0005	99.7	2.01	+0.464
cartpole-balance	+0.996±0.002	0.99	0.0034	99.1	0.08	+0.984
cartpole-balance-sparse	+1.000±0.000	1.00	0.0016	100.0	0.04	+0.992
cartpole-swingup	+0.985±0.013	0.96	0.0008	99.6	0.39	+0.957
cartpole-swingup-sparse	+0.937±0.017	1.02	0.0028	98.9	1.66	+0.842
cheetah-jump	+0.884±0.016	0.99	0.0079	96.4	1.12	+0.602
cheetah-run	+0.828±0.009	0.89	0.0061	96.4	1.36	+0.455
cheetah-run-back	+0.912±0.013	0.96	0.0095	97.3	0.57	+0.537
cheetah-run-backwards	+0.785±0.036	0.96	0.0147	95.0	1.38	+0.473
cheetah-run-front	+0.862±0.022	0.95	0.0055	97.2	1.03	+0.575
cup-catch	+0.952±0.004	0.98	0.0129	76.5	0.24	+0.732
cup-spin	+0.910±0.006	0.92	0.0288	95.8	0.62	+0.688
dog-run	+0.635±0.047	0.76	0.0014	82.3	1.79	+0.429
dog-stand	+0.589±0.012	0.76	0.0020	82.5	2.80	+0.446
dog-trot	+0.585±0.074	0.87	0.0010	86.0	2.66	+0.478
dog-walk	+0.532±0.070	0.75	0.0023	83.5	3.06	+0.424
finger-spin	+0.655±0.175	0.84	0.0252	78.4	1.66	+0.558
finger-turn-easy	+0.987±0.003	1.00	0.0025	98.9	0.11	+0.825
finger-turn-hard	+0.948±0.015	0.97	0.0044	95.1	0.48	+0.686
fish-swim	+0.802±0.033	0.95	0.0126	98.2	1.41	+0.576
hopper-hop	+0.724±0.102	0.92	0.0125	83.5	2.60	+0.482
hopper-hop-backwards	+0.624±0.074	0.90	0.0046	88.5	4.08	+0.564
hopper-stand	+0.821±0.014	0.93	0.0122	84.4	0.94	+0.659
humanoid-run	+0.405±0.054	1.03	0.0017	79.4	6.54	+0.488
humanoid-stand	+0.348±0.041	0.58	0.0051	73.6	4.59	+0.326
humanoid-walk	+0.353±0.102	0.67	0.0021	69.5	4.53	+0.361
pendulum-spin	+0.977±0.011	0.99	0.0001	98.9	0.28	+0.519
pendulum-swingup	+0.996±0.001	1.01	0.0008	99.7	0.31	+0.938
quadruped-run	+0.579±0.028	0.73	0.0117	78.5	0.98	+0.321
quadruped-walk	+0.761±0.029	0.87	0.0058	74.6	0.29	+0.642
reacher-easy	+0.999±0.000	1.00	0.0045	99.7	0.05	+0.704
reacher-hard	+0.999±0.000	1.00	0.0038	99.8	0.02	+0.600
reacher-three-easy	+0.988±0.001	0.98	0.0042	94.0	0.06	+0.700
reacher-three-hard	+0.959±0.022	0.99	0.0037	94.8	0.58	+0.746
walker-run	+0.576±0.009	0.84	0.0127	74.8	2.66	+0.419
walker-run-backwards	+0.518±0.028	0.92	0.0104	76.7	3.29	+0.414
walker-stand	+0.753±0.010	0.92	0.0046	78.3	1.69	+0.574
walker-walk	+0.617±0.015	0.88	0.0075	80.6	3.15	+0.421
walker-walk-backwards	+0.255±0.034	1.61	0.0060	85.4	5.90	+0.189

Table 6: The released Atari models show the same pattern. Zero action divergence does not change under reflection or rotation, retrieval rates both as much better, and the matched probe follows the true direction. We transform 26 released DIAMOND diffusion world models, one per Atari 100k game (Alonso et al., 2024), which predict the next frame in pixels, and report means over games. Retrieval is the rank of the logged action among the game's actions divided by chance, so 1 is chance and lower counts as more faithful.

	simulator		probe (no simulator)		
model	direction ↑	size	C_1 ↑	ret. rank ↓	matched ↑
trained model	+0.488	1.16	0.0059	0.54	+0.178
reflected response	-0.488	1.16	0.0059	0.09	-0.178
rotated response	+0.000	1.16	0.0059	0.03	+0.000
response $\times 0.25$	+0.488	0.29	0.0004	0.95	+0.178
response $\times 4$	+0.488	4.66	0.0940	0.04	+0.178
no response	+0.000	0.00	0.0000	1.00	+0.000

over 1, 3 or 5 matched pairs per position (Table 8). Table 9 restricts LIBERO to the 30 models that differ only in training objective and seed.

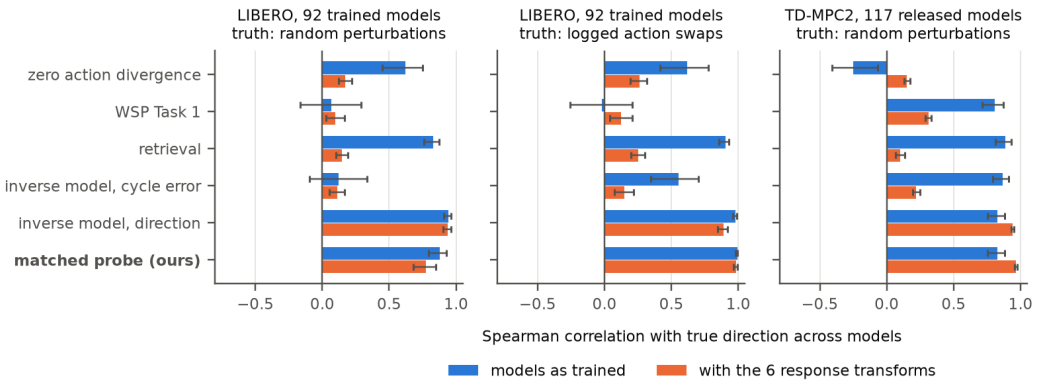

Figure 3: Only the two direction probes, the inverse model check scored by direction and the matched probe, keep a high correlation once the transformed copies are included. Each bar is the Spearman correlation across models between a probe and the true direction, with its 95% bootstrap interval, for the LIBERO and TD-MPC2 columns of Table 2. A value of 1 means the probe orders the models exactly as the simulator does and 0 means no relation. Blue bars use the models as trained. Orange bars add each model's transformed copies, which change direction and size independently.

Table 7: Bootstrap intervals for Table 2, which show that the matched probe's correlations are well above zero in every group. Each entry is a 95% interval for the Spearman correlation. We resample models and keep each model's transformed copies together, and for TD-MPC2 we resample tasks.

	LIBERO, 92 models we trained random perturbations		LIBERO, 92 models we trained logged action swaps		TD-MPC2, 117 released random perturbations		Atari, 26 released logged action swaps	
probe	as trained	transformed	as trained	transformed	as trained	transformed	as trained	transformed
zero action divergence	[+0.45, +0.75]	[+0.12, +0.22]	[+0.41, +0.78]	[+0.19, +0.31]	[-0.51, +0.06]	[+0.12, +0.18]	[-0.18, +0.63]	[+0.16, +0.24]
WSP Task 1	[-0.16, +0.29]	[+0.03, +0.17]	[-0.26, +0.21]	[+0.04, +0.21]	[+0.62, +0.91]	[+0.27, +0.34]		
retrieval	[+0.76, +0.87]	[+0.10, +0.19]	[+0.85, +0.93]	[+0.20, +0.30]	[+0.75, +0.95]	[+0.05, +0.14]	[+0.63, +0.95]	[-0.20, -0.09]
retrieval, real negatives	[+0.79, +0.91]	[+0.01, +0.11]	[+0.85, +0.95]	[+0.11, +0.23]				
inverse model, cycle error	[-0.10, +0.33]	[+0.05, +0.17]	[+0.35, +0.70]	[+0.07, +0.22]	[+0.74, +0.93]	[+0.18, +0.26]	[-0.27, +0.53]	[+0.14, +0.46]
inverse model, direction	[+0.91, +0.96]	[+0.90, +0.96]	[+0.96, +0.99]	[+0.84, +0.92]	[+0.69, +0.90]	[+0.92, +0.95]	[+0.48, +0.88]	[+0.69, +0.80]
matched probe (ours)	[+0.79, +0.93]	[+0.68, +0.85]	[+0.98, +1.00]	[+0.96, +0.99]	[+0.68, +0.91]	[+0.95, +0.98]	[+0.85, +0.98]	[+0.88, +0.97]

I OBJECTIVES COMPARED

Table 10 compares the objectives with the additive baseline. Delta prediction is the most faithful objective under both ground truths, and inverse dynamics at weight 0.01 also improves on the baseline. Retrieval ranks both below the baseline, significantly for inverse dynamics ($p < 10^{-3}$) but not for delta prediction ($p = 0.08$). So choosing objectives by retrieval would at best miss what the simulator prefers. The matched probe ranks them as the simulator does, $+0.020$ and $+0.017$, both with $p < 0.002$ and significant after Holm correction over five comparisons. It also detects the one large failure, inverse dynamics at weight 1, whose loss swamps the state loss. There it gives -0.280 against -0.303 in on support truth. These effects are small in absolute size, and their sign is what matters here. For AGM every direction measure and the matched probe move by less than 0.001, while retrieval's mean rank improves by 0.10 ($p = 0.004$). This is what Proposition 6(i) and Proposition 3 lead us to expect for an objective that rewards response size. A cosine ignores size, and retrieval rewards it. The tiny changes in direction reach significance only because each AGM model is nearly identical to its paired baseline.

J THE INVERSE MODEL CHECK

Protocol. The inverse model g reads a state and the change to the next state, $(x_t,\ x_{t+1} - x_t)$. We fit it to logged transitions only, by least squares for continuous actions and by cross entropy for discrete ones. It never sees a simulator counterfactual. On LIBERO it is a network with 3 hidden layers of width 512. We fit it to the 138455 transitions of the training demonstrations, the data the world models were trained on. On the 30670 transitions of the validation demonstrations its R^2 is 0.88 to

Table 8: The matched probe hardly depends on how many matched pairs it averages per position. Each entry is the Spearman correlation with the true direction over the 92 LIBERO models, as trained and with transforms, where higher is better.

	logged action swaps		random perturbations	
pairs per position	as trained	transformed	as trained	transformed
1	+0.992	+0.984	+0.877	+0.775
3	+0.989	+0.974	+0.866	+0.766
5	+0.993	+0.985	+0.872	+0.773

Table 9: Among the 30 LIBERO models that differ only in training objective and seed, the two direction probes track the true direction and retrieval does not. Each entry is the Spearman correlation with the true direction, where higher is better. Objective means averages the five seeds of each objective. Seed level removes each objective's mean, so it asks whether a probe ranks the seeds of one objective. The last column drops inverse dynamics at weight 1, whose loss swamps the state loss.

	D logged			D random	D logged
probe	all 30	objective means	seed level	seed level	without inv. dyn. $w = 1$
zero action divergence	-0.44	-0.60	+0.28	-0.24	-0.05
WSP Task 1	-0.57	-1.00	-0.02	+0.32	-0.48
retrieval	+0.14	-0.09	+0.40	+0.04	-0.49
retrieval, real negatives	+0.38	+0.26	+0.24	+0.19	-0.07
inverse model, cycle error	+0.55	+0.66	-0.13	+0.50	+0.23
inverse model, direction	+0.89	+1.00	+0.57	+0.77	+0.82
matched probe (ours)	+0.92	+1.00	+0.68	+0.64	+0.86

0.98 per action dimension. On TD-MPC2 we fit one inverse model per checkpoint, with 2 hidden layers of width 256, in that checkpoint's latent space. We fit it to 5 of the 6 logged episodes and measure it on the sixth. Its R^2 has a median of 0.87. It is above 0.5 for 84% of models, lowest -2.75. On Atari, 3 convolutional layers read the frame and the frame difference. We fit them to the random policy's transitions of every episode but the last (1428 to 3064 per game). On the last episode the classifier recovers the action at a median 1.8 times chance (0.6 to 4.8). We fixed each inverse model's architecture and training before computing any inverse check score, and we tuned none of them. Our cycle error follows the idea of Liu et al. (2026) with a dense inverse model, without its sparse state features or subgoal generator.

We ask the check about the same action changes as the ground truth it is compared with. These are the random perturbations behind D and the nearest logged swaps behind $\mathrm{D}_{\log}$. We read the first predicted step against the last observed state and later steps against the previous prediction. We apply a transform to the rollout before g reads it. For discrete actions, IC compares the change in g's action probabilities with the change in the one hot action, and the error is the cross entropy of the queried action.

Results. Table 11 gives the mean of each quantity per transform, and Table 12 gives every combination of query set and ground truth. In proprioception and in TD-MPC2's latent space, g is accurate. There IC reverses under reflection, stays near zero under rotation and moves by at most 0.02 under either rescaling, so the check behaves as Proposition 8 predicts for an affine g. This suggests that g is close to linear over the range of the responses. On Atari frames neither property holds. Reflection moves IC to -0.005 instead of reversing it, and rescaling by 0.25 more than halves it. The cycle error rises under rescaling by 4 in every group of models, as a measure of size must. On LIBERO under random perturbations it falls when the response is shrunk or removed. Asked about random perturbations, IC tracks random perturbation truth best. Asked about logged swaps, it tracks on support truth best. The matched probe can only be asked about swaps.

Released TD-MPC2 models. We fit one inverse model per checkpoint in its latent space (median held out R^2 0.87). The direction form then matches the matched probe as trained (0.83) and trails it with transforms (0.94 against 0.97) and within tasks (0.56 against 0.62). Its cycle error falls to 0.22 with transforms, but as trained it tracks direction at 0.86, and within tasks at 0.69.

Table 10: Retrieval ranks the two objectives that the simulator prefers below the baseline, and the matched probe ranks them correctly. Values are mean changes from the additive baseline over 5 paired seeds, and arrows show which change is better. A positive change in retrieval rank means retrieval scores the objective as less faithful. p is a paired t test on the matched probe.

	simulator: Δ direction		probe: Δ score		
objective vs. additive	random ↑	logged ↑	ret. rank	matched ↑	p (matched)
+ AGM	+0.001	-0.000	-0.10	-0.001	<0.001
+ inv. dyn. ($w = 1$)	-0.027	-0.303	+1.46	-0.280	0.002
+ inv. dyn. ($w = 0.01$)	+0.039	+0.010	+0.12	+0.017	<0.001
+ inv. dyn. ($w = 0.001$)	+0.004	+0.004	-0.02	+0.005	<0.001
delta prediction	+0.050	+0.027	+0.15	+0.020	0.001

Table 11: Scored by direction, the inverse check reverses under reflection and ignores rescaling wherever the inverse model is accurate, and it loses both properties on Atari frames. Its cycle error changes with rescaling, as a measure of size must. Values are means over models, next to the true direction. IC is the direction form and runs from -1 to $+1$ like the truth. For the cycle error, lower means the model is scored as more consistent.

	LIBERO, 15 trained models						TD-MPC2, 117			Atari, 26		
	random perturbations			logged swaps			random perturbations			swaps		
model	truth	IC	error	truth	IC	error	truth	IC	error	truth	IC	error
trained model	+0.312	+0.392	2885	+0.810	+0.857	9.6	+0.769	+0.829	2.89	+0.488	+0.138	10.80
reflected response	-0.312	-0.381	4436	-0.810	-0.835	100.6	-0.769	-0.816	9.32	-0.488	-0.005	16.81
rotated response	-0.013	-0.033	3241	+0.028	-0.059	57.4	+0.000	-0.018	5.94	+0.000	+0.011	14.81
response $\times 0.25$	+0.312	+0.385	2033	+0.810	+0.848	22.0	+0.769	+0.826	3.74	+0.488	+0.052	13.14
response $\times 4$	+0.312	+0.387	21232	+0.810	+0.839	215.1	+0.769	+0.820	21.10	+0.488	+0.117	15.25
no response	+0.000	+0.000	2122	+0.000	+0.000	32.0	+0.000	+0.000	4.44	+0.000	+0.000	14.49

K TRAINING ON THE LOGGED ESTIMATE

Pairs. Matched swap alignment uses only the training split. We take every second position p of a training demonstration that has a full context window of 16 states. For each, we find the transition q in another training demonstration of the same task whose normalized state is nearest. We only accept a transition whose normalized action differs from the one at p by at least 0.5 in Euclidean norm. This gives 61613 pairs with a median state gap of 0.525 in normalized units. For each pair the loss is $1 - \cos(r_{pq}, \hat{e}_{pq})$. Here $r_{pq} = f(H_p, a_q) - f(H_p, a_p)$ is the model's one step response, and $\hat{e}_{pq} = (x_{q+1} - x_q) - (x_{p+1} - x_p)$ is the logged estimate. We average the loss over 128 pairs per step and add it to the next state MSE with weight w.

Training and evaluation. We draw pairs from their own random stream, so each model sees the same MSE batches as its baseline with the same seed. The architecture, optimizer, schedule and 5000 steps match the baseline. The contrastive loss is an InfoNCE objective (van den Oord et al., 2018). The one step predictions of the logged action window must match the recorded next states better than those of 4 Gaussian perturbations of the whole window, with standard deviation 0.3. Its logits are negative squared errors, which differ by about 10^{-3}, so we divide them by a temperature of 10^{-3}. Without it the softmax stays uniform and the loss never leaves $\log 5$. We fixed the weights of every loss before training. We report every run with the final settings (Table 13). Our first contrastive runs had no temperature, never trained, and changed direction by less than 0.001. All scores use the held out positions of Section 4, so no evaluated pair or perturbation was seen in training. A training run takes a few minutes on one laptop GPU.

Ablations. The random partner ablation keeps the loss and the action filter. It replaces the nearest partner with a random transition of another demonstration of the same task. The shuffled target ablation permutes the logged estimates across pairs. Squared error imputation predicts the swapped next state as $x_p + (x_{q+1} - x_q)$ and uses the same pairs. This adapts the nearest neighbor outcome term of Johansson et al. (2016) to a world model, with a squared error in place of their absolute error.

Control. We chose the weight for the control experiment before running it, by the lowest mean validation MSE, which is a training criterion. This selected $w = 0.01$. The tasks, goals and planner are those of Appendix Q. So each episode pairs with its baseline episode on the same goal and with

Table 12: The inverse check tracks the true direction best when it is asked about the same kind of action change as the truth. Each entry is the Spearman correlation with the true direction, with the matched probe for comparison. We negate errors so that higher always means better agreement.

	LIBERO, 92 models				LIBERO, 30 ordinary		TD-MPC2, 117		
	D random		D logged		D random	D logged	direction		
query set	trained	transf.	trained	transf.	trained	trained	trained	transf.	within task
IC, logged swaps	+0.88	+0.80	+0.98	+0.89	+0.58	+0.89	+0.81	+0.94	+0.69
IC, random queries	+0.94	+0.94	+0.81	+0.75	+0.84	+0.53	+0.83	+0.94	+0.56
cycle error, logged swaps	+0.50	+0.18	+0.55	+0.15	+0.27	+0.55	+0.85	+0.28	+0.62
cycle error, random queries	+0.12	+0.11	+0.00	+0.01	+0.65	+0.65	+0.86	+0.22	+0.69
matched probe	+0.88	+0.78	+0.99	+0.98	+0.61	+0.92	+0.83	+0.97	+0.62

Table 13: Every objective and weight we trained on top of next state MSE, in the format of Table 3, except inverse dynamics at weights 0.001 and 1 (Table 10) and three contrastive runs without the temperature, whose loss stayed at $\log 5$. The contrastive loss lowers the true direction at both weights, and matched swap alignment raises direction under random perturbations at every weight. Values are mean changes from next state MSE over 5 paired seeds, and arrows show which change is better. An asterisk marks a Holm adjusted paired t test below 0.05 over all rows of the column.

objective	Δ D random ↑	Δ D logged ↑	Δ cf. error ↓	Δ size	Δ ret. rank	Δ val. MSE ($\times 10^{-4}$) ↓
delta prediction	+0.050*	+0.027*	+0.113	+0.134	+0.15	-1.64*
+ inv. dyn. ($w = 0.01$)	+0.039*	+0.010*	+0.024	+0.060*	+0.12*	+1.73*
+ AGM (size)	+0.001*	0.000	+0.008*	+0.009*	-0.10*	+0.05*
+ contrastive ($w = 0.01$)	-0.097*	-0.239*	+0.465*	+0.228*	-0.43*	+5.94*
+ contrastive ($w = 0.1$)	-0.133*	-0.291*	+0.414*	+0.138*	-0.75*	+14.39*
+ matched swap alignment ($w = 0.001$)	+0.065*	+0.041*	-0.062*	+0.007	-0.31*	-0.10*
+ matched swap alignment ($w = 0.01$)	+0.124*	+0.046*	-0.152*	-0.011	-0.40*	-0.22*
+ matched swap alignment ($w = 0.1$)	+0.172*	+0.011	-0.139*	+0.019	-0.08	+1.31*
random partner ($w = 0.01$)	+0.082*	+0.020*	-0.064*	+0.016	-0.11	-0.06*
shuffled targets ($w = 0.01$)	-0.061*	-0.164*	+0.066*	-0.005	+0.78*	+0.94*
+ squared error imputation ($w = 0.1$)	+0.013*	+0.016*	-0.023*	-0.006	-0.02	-0.30*
+ squared error imputation ($w = 1$)	+0.040*	+0.037*	-0.085*	-0.035*	-0.06	-0.78*

the same planner random stream. Over 60 paired episodes, the models trained with matched swap alignment miss by 0.056 cm more than their MSE baselines. This is not significant (paired t test over seeds, $p = 0.26$).

L Fine tuning released models

Table 14 gives the result. For each of the 39 tasks, the seed 1 checkpoint's policy prior logs 6 episodes of 500 steps, with exploration noise of standard deviation 0.3. We branch the simulator at 25 states per episode under random perturbations of the logged actions, as in Appendix F. We freeze the encoder, so the space in which we measure the simulator's truth does not move. We fine tune only the latent dynamics, for 1000 steps with Adam (Kingma & Ba, 2015) at learning rate 10^{-4}. Every arm starts from the released weights and sees the same batches. Matched swap alignment pairs each one step transition of episodes 0 to 3 with the nearest transition of another of those episodes, by standardized observation. It only accepts transitions whose action differs by at least 0.5. Direction is the mean cosine between the model's and the simulator's response over the branch states of episodes 4 and 5. Held out error is the consistency loss on the transitions of those episodes. We also measure the logged estimate against the simulator. For 300 pairs we branch from the saved state of the first transition under its partner's action. We then compare the logged estimate and the model's own one step response with the simulator's response, in the model's latent space.

M Negative controls

Table 15 gives every probe's value under each control.

When the causal link is removed in training, the matched probe collapses together with ground truth, from 0.749 to between 0 and 0.069 under every control. Retrieval keeps much of its advantage under the sequence level controls, with a mean rank of 2.96 against chance 7.5. A model can recover the current action from past ones when whole sequences are permuted. The calibration score does not

Table 14: On released TD-MPC2 models, matched swap alignment lowers direction at every weight, and a larger weight gives a larger drop and a higher held out error. We fine tune the 39 released seed 1 models and measure direction against the simulator on held out episodes, where higher is better. Δ direction is paired by task against fine tuning with the consistency loss only. Held out error is the median ratio to that arm, so a value above 1 is worse.

fine tuning	direction ↑	Δ direction ↑	tasks improved	held out error ↓
released model	0.788	-0.023		×1.13
consistency loss only	0.810			×1.00
+ matched swap alignment ($w = 0.001$)	0.796	-0.015	14 of 39	×1.01
+ matched swap alignment ($w = 0.01$)	0.762	-0.048	3 of 39	×1.23
+ matched swap alignment ($w = 0.1$)	0.714	-0.096	1 of 39	×1.81

Table 15: When training breaks the link between action and outcome, the true direction and the matched probe collapse, while retrieval keeps much of its score under the sequence level controls. Values are for the additive objective, mean over 5 seeds, and arrows show which way each quantity counts as more faithful.

training	D random ↑	D logged ↑	C_1 ↑	Task 1 ↑	ret. rank ↓	matched ↑
logged	0.295	0.801	0.035	49.9	0.90	0.749
sequence perm.	0.037	0.073	0.008	49.9	2.96	0.069
derangement	0.034	0.048	0.009	49.8	2.98	0.046
per timestep	0.018	0.009	0.005	50.1	4.07	0.001
action blind	0.000	0.000	0.000	0.0	7.29	0.000

move unless the model has no response at all, as Proposition 2 suggests. The derangement gives the same result as the leaky permutation, so the one fixed point per batch was not what kept retrieval high. Drawing retrieval's alternatives from real actions elsewhere in the dataset does not remove its remaining advantage either.

N THE CALIBRATION SCORE WITH ITS DEFAULT SETTING

Figure 4 shows the simulator's ratios and every model's score.

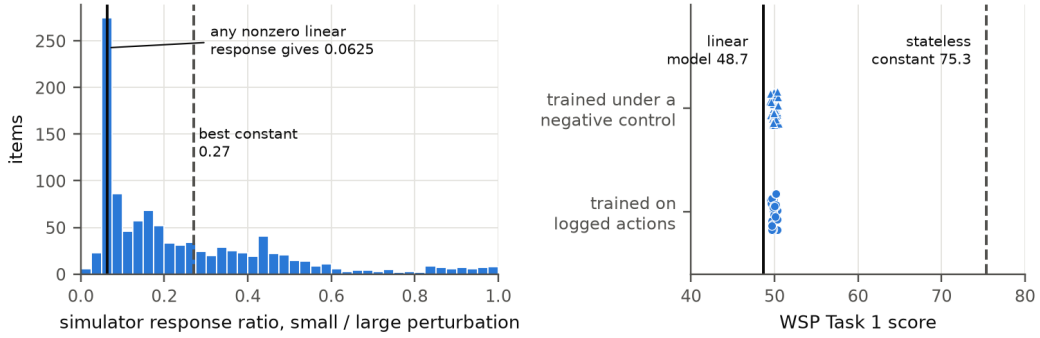

Figure 4: The calibration score gives nearly the same value to every model that responds to the action. The left panel shows, over 1066 items, the ratio between the simulator's responses to a small and a large perturbation. Every model whose response is linear and not zero gives 0.0625 (Proposition 2), and a constant ratio of 0.27 that ignores the state gets the best score. The right panel shows the score, from 0 to 100 where higher counts as more faithful, for the 70 models trained for 5000 steps that respond to the action, on logged actions or under a negative control. All of them land near 50, close to an exactly linear model (48.7) and far below the best constant (75.3). The 12 shorter runs also land near 50 and the 10 action blind controls score 0, and neither is shown.

We ran WorldSimProbe Task 1 with its authors' scoring function and the default Task 1 setting of their released code. That setting perturbs one action dimension by 0.14, and by 0.25 times 0.14 in the same direction. The benchmark in their paper instead draws two magnitudes per item from a small set, and Proposition 2 applies to either choice. We perturb each of the three translation dimensions in turn, over all 8 actions of the window. Each item uses four branches from one restored state, a discarded warm up, the logged run, and the small and the large perturbation. We measure responses as state MSE instead of video MSE. Over 1066 valid items in 30 tasks, every model we train that responds to the action scores between 49.4 and 50.7. This includes models trained under every negative control that leaves an action pathway in place. An exactly linear model scores 48.66. Between the 10th and 90th percentiles, the simulator's ratios range from 0.06 to 0.54, with medians of 0.070, 0.162 and 0.397 along the three translation axes. A constant ratio of 0.27 that ignores the state scores 75.3, above every model.

O CHANCE LEVEL OF RETRIEVAL

Retrieval scores are often compared with a chance level of $1/n$ for picking the logged action out of n candidates. That level holds only for a model that ignores the action entirely and breaks ties at random. For other models the candidates are not interchangeable, because the logged action sits at the center of the cloud of perturbed candidates. A better reference is a permutation null (Ojala & Garriga, 2010). We score the same candidates against the recorded next state of an unrelated position, where the logged action has no reason to win. This null exposes adversaries that score far above $1/n$ without having learned anything (Appendix P). We use the null to test one model at a time, and we do not subtract it before comparing models. The null depends on the shape of each model's responses. Its mean rank over the 360 positions is 9.07 for the additive objective, 7.35 for delta prediction and 7.54 for inverse dynamics at weight 1. Subtracting it would turn a raw rank difference of 0.15 between delta prediction and the baseline into an apparent deficit of 1.86, opposite to what the simulator shows.

P SYNTHETIC WORLDS

Table 16 gives the results for world G, where the action drives the state through a known matrix.

Table 16: A model with no dynamics can raise retrieval but not the matched probe. In world G the action drives the state through a known matrix. We build the adversaries on an action blind base, so they contain no learned dynamics. True direction runs from -1 to $+1$. Retrieval excess is retrieval's score above the permutation null, where higher counts as more faithful. Means over 5 seeds.

model (world G)	true direction	C_1	response	ret. excess	matched
trained, MSE	+0.996	1.72	1.91	+0.924	+0.805
trained, + AGM	+0.992	2.83	2.45	+0.895	+0.803
action blind base	-0.102	0.14	0.52	+0.013	-0.101
blind + unconditional inflation	-0.164	8.66	3.93	-0.005	-0.131
blind + policy aligned inflation	-0.164	8.13	3.93	+0.341	-0.131
blind + policy aligned, $c = 8$	-0.157	124.37	15.48	+0.090	-0.124

The adversaries add an inflation to an action blind base. One kind, $cB(a - \bar{a})$, ignores the behavior policy. The other, $cB(a - \hat{\pi}(H))$, follows a learned predictor of the policy. In both, B is a fixed random matrix. Unconditional inflation leaves retrieval at the null while raising zero action divergence fivefold. At $c = 8$ the permutation null rises with the policy aligned adversary, and its excess falls to $+0.090$. In world N, where the action has no effect, every model's matched probe is within 0.03 of 0. The whole sequence negative control collapses the genuine model's retrieval excess but not the policy aligned adversary's. Permuting whole sequences keeps the action autocorrelation that the adversary's policy predictor uses.

Q CLOSED LOOP CONTROL

We use 12 tasks, 4 from each of the three suites LIBERO Goal, Object and Spatial. For each we take one held out position and restore its state. We run one discarded warm up branch, restore again, and execute the logged action once, so every episode starts from a live simulator state. The goal is the gripper position plus 4 cm along a direction drawn at random per position. Every model and transform uses the same goals. Control runs for 20 steps, which is one second. At each step the cross entropy method (Rubinstein, 1999) samples 256 sequences of 8 translation actions, with rotation held at zero and the gripper at its logged command. It scores each sequence by the mean predicted distance of the gripper to the goal over the 8 steps. It refits the sampling distribution to the 32 best for 4 rounds and executes the first action of the final mean. We warm start the plan from the previous step's plan, shifted by one. A planner has no logged action to transform around. So a transformed model is $P(a_0) + T(P(a) - P(a_0))$, where a_0 is the sequence that does not move the arm. A model with no response predicts the same future for every sequence, and its planner then acts on random tie breaks.

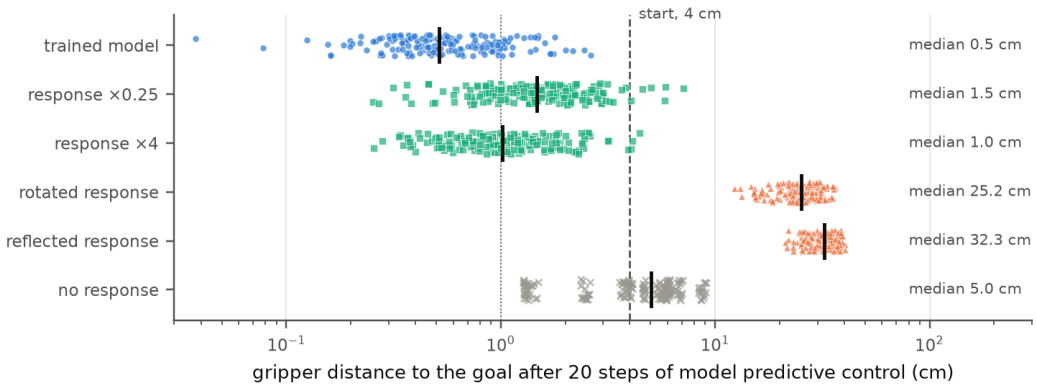

Figure 5: Errors of direction make control fail, and errors of size do not. Each point is one episode. It shows the final distance of the gripper to a goal that starts 4 cm away, after 20 steps of model predictive control. The axis is logarithmic, and lower is better. Rows are the 15 trained LIBERO models of Table 1 and their transforms over 12 tasks, 180 episodes per row, and black bars mark the medians. Points to the right of the dashed line ended farther from the goal than they started, and the dotted line marks 1 cm.

R ADDITIONAL FIGURES

Figure 8 summarizes what each metric concludes about each transformed model, and Figure 9 plots every probe against the true direction.

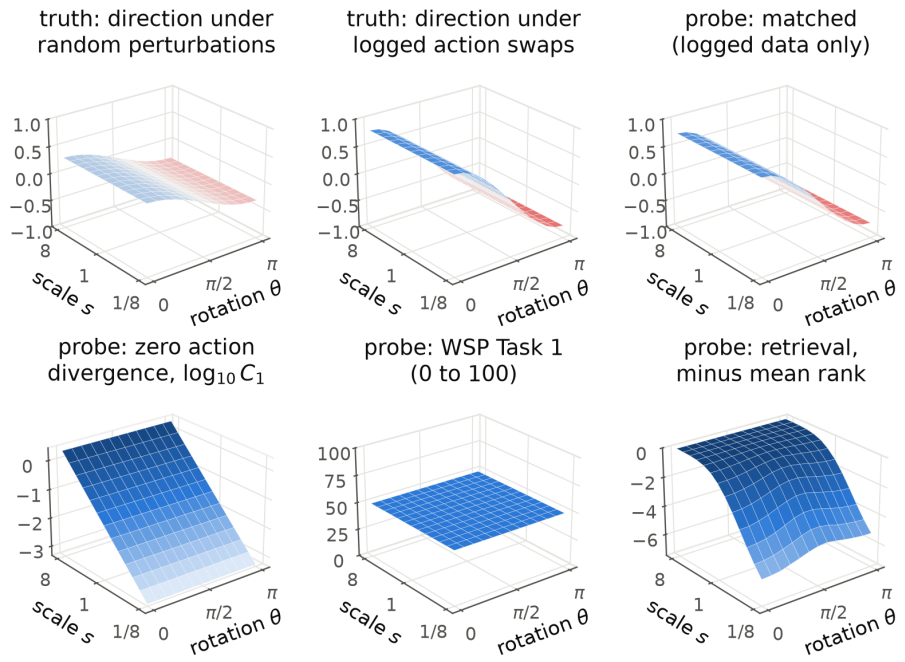

Figure 6: The two ground truths and the matched probe change with the direction of the response and ignore its size, and zero action divergence does the opposite. We rotate the response of each trained LIBERO model by an angle θ and rescale it by s, and plot the mean over 15 models. $\theta = \pi$ reflects 20 of the 21 state dimensions, and $s = 1$ is the original size. The top row shows both ground truths and the matched probe, from -1 to $+1$. The bottom row shows zero action divergence on a log scale, the calibration score from 0 to 100, and minus the retrieval mean rank. In every panel, higher counts as more faithful. Zero action divergence changes with s and not with θ, the calibration score changes with neither, and retrieval changes mostly with s.

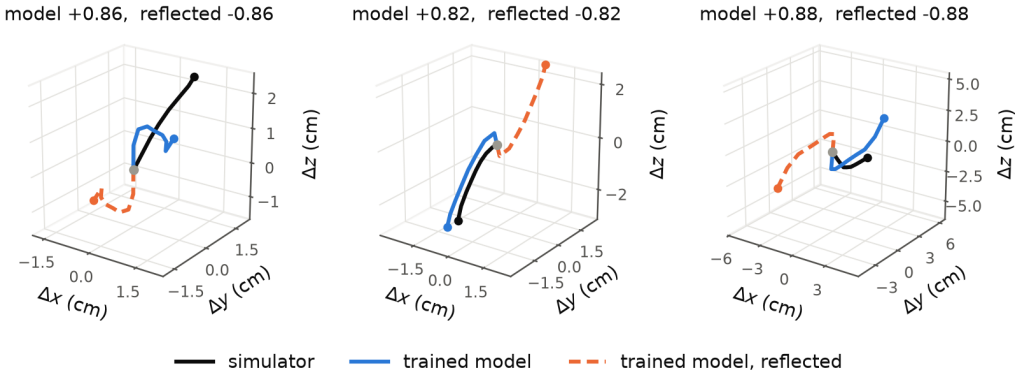

Figure 7: A reflected prediction has the same size as the original but points the other way, so size probes cannot tell them apart and the matched probe can. Each panel shows, for one held out LIBERO state, how swapping in another logged action sequence moves the gripper over 8 steps, in cm, starting from the gray dot. Black is what the simulator did, blue is the trained model's prediction, and dashed orange is the same prediction reflected. Titles give each prediction's cosine with the simulator, from -1 to $+1$. The model is the next state MSE model with seed 0. We chose the items closest to its median gripper cosine (0.85) among those whose true gripper effect exceeds 3 cm, from different tasks.

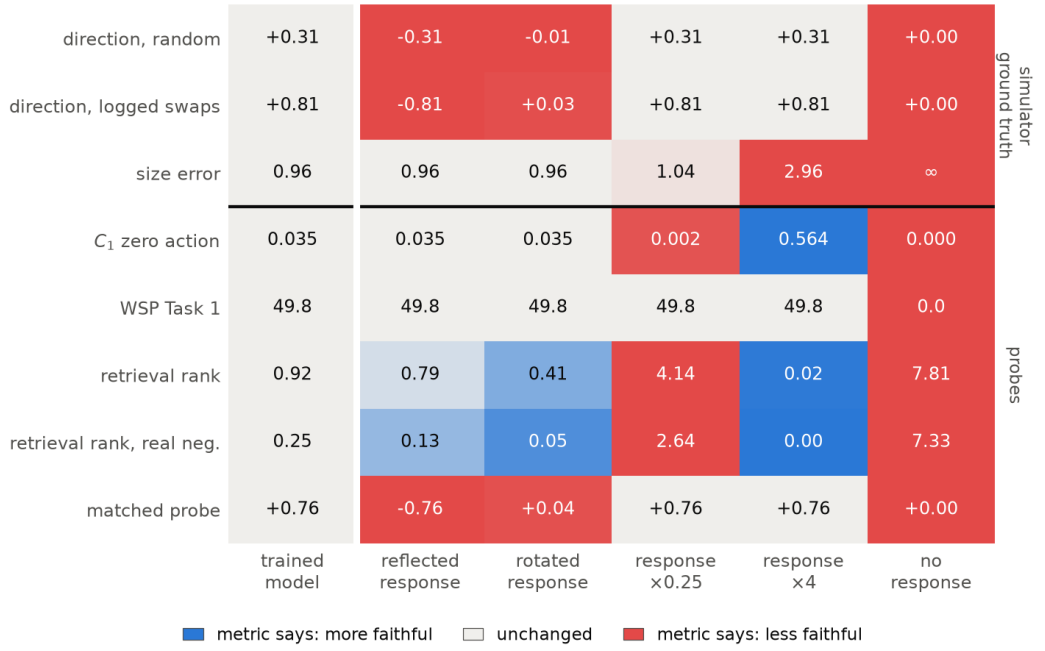

Figure 8: Only the direction rows and the matched probe mark the reflected and rotated models as less faithful. The size probes leave them unchanged, and retrieval marks them as more faithful. Each cell is the mean over 15 trained LIBERO models of one quantity (row) for one version of the model (column). The color shows what that quantity concludes compared with the trained model, blue for more faithful, red for less faithful and gray for unchanged. Direction runs from -1 to $+1$. Size error is $|\log_2 \mathrm{M}|$, the number of doublings by which the predicted response size is off, so 0 is correct and 0.96 means about twice too large. C_1 and the calibration score count higher as more faithful. Retrieval ranks count lower as more faithful, with chance at 7.5.

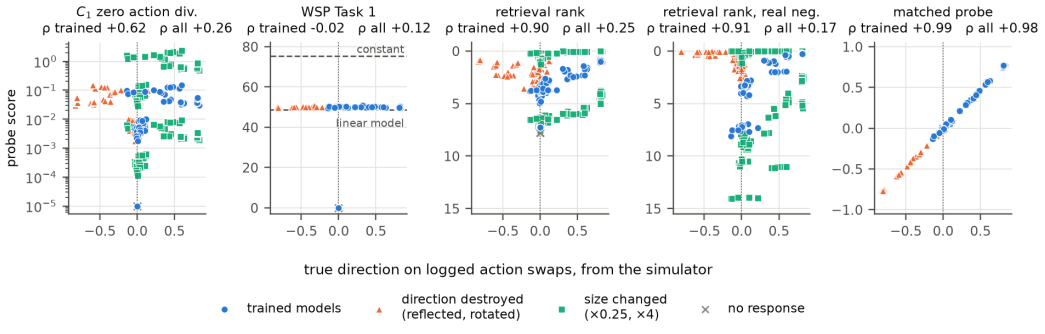

Figure 9: Only the matched probe lines up with the true direction once the reflected and rotated copies are included. Each panel plots one probe against the true direction on logged swaps, from -1 to $+1$, for the 92 LIBERO models and their transformed copies. The vertical axes are oriented so that up counts as more faithful. Titles give the Spearman correlation ρ for the trained models alone and for all points.