DOES THE FUTURE FOLLOW THE ACTION? MEASURING ACTION FIDELITY IN WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Action-conditioned video world models are used to plan, to evaluate policies, and to compare models; each use depends on whether the predicted video follows the supplied action, a property we call *action fidelity*. Existing evaluations measure visual quality, VLM-judged plausibility, or downstream policy performance; none isolates action fidelity, and reference-based metrics can point the wrong way: on a world model's own predictions, PSNR ranks the pixel average of the executed futures above every checkpoint. Learning the judgment directly invites a shortcut. A contrastive scorer trained to match futures with their histories reaches 99% accuracy on its training task, yet exactly chance (12.5%) when asked which of eight futures from the same state an action produced: what a contrastive score measures is decided by what it is contrasted against. Contrasting each future with futures of alternative actions executed from the same state removes the shortcut and yields LAFS (Learned Action-Fidelity Score), which rates a prediction from its history and action without a reference rollout. We introduce BranchPairs, which pairs an executed future that follows the queried action, clean or under an appearance change, with one that deviates from it, on LIBERO, RoboTwin, and Action-MNIST. Trained only on clean executions, LAFS prefers the future that follows the action in 89% of LIBERO pairs (chance 50%), versus 57% for PSNR and 75% for an inverse dynamics model trained on the same data, and keeps its lead on pairs matched in PSNR (93%, versus 80% and at most 60% for visual metrics). On a Cosmos world model fine-tuned on LIBERO-10, LAFS orders three checkpoints and three controls as an executed-future reference does, unlike PSNR and a VLM critic.

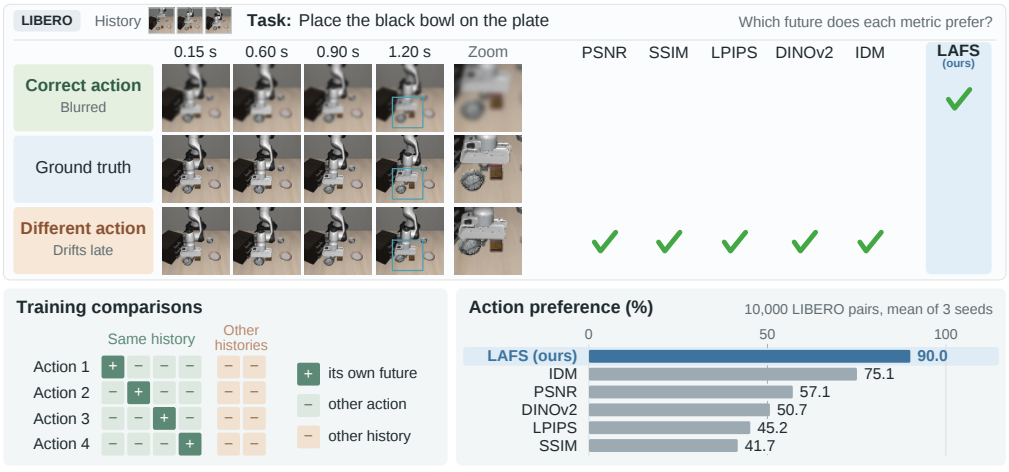


Figure 1: **Visual metrics barely tell which future follows the action; LAFS does. Top:** a future of the queried action, only blurred as by a hedging prediction, against a sharp future that follows the action and then drifts to a different one; checks mark the future each metric prefers (selected example). **Bottom:** LAFS training pairs (left) and preference over 10,000 LIBERO pairs (right); IDM: inverse dynamics model.

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.

1 INTRODUCTION

An agent that can foresee the consequences of its actions can choose among them before it acts. Action-conditioned video world models provide this foresight: given an observation history and a sequence of actions, they predict the observations that the actions would bring about (Ha & Schmidhuber, 2018; Kim et al., 2020; Yang et al., 2024b; Bruce et al., 2024; NVIDIA, 2025). They are used to plan by simulating and ranking candidate actions (Zhou et al., 2025; Assran et al., 2025), to evaluate policies without a robot (Tseng et al., 2026), and are ranked by the videos they produce. These uses rest on a property that realism alone does not provide: the predicted future must be the one that this action, rather than another action available in the same state, would produce. We call this property *action fidelity*, as distinct from *visual fidelity*; related notions appear as action controllability in driving and camera control (Arai et al., 2024; Gao et al., 2024; Wang et al., 2026).

World models are currently evaluated along three lines. Visual metrics compare a prediction with the recorded future, as PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and DINOv2 feature distance (Oquab et al., 2024) do, or compare distributions of videos, as FID and FVD do (Heusel et al., 2017; Unterthiner et al., 2018), and benchmark suites extend them (Huang et al., 2024; Liu et al., 2024; Duan et al., 2025). VLM judges assess whether a rollout completes its instruction and is physically plausible (Li et al., 2025; Bansal et al., 2025); they receive the instruction and the video but not the action. Downstream evaluations measure policy success or policy ranking (Shang et al., 2026; Tseng et al., 2026), or how well a VLM critic chooses actions from imagined rollouts (Alzayer et al., 2026), and mix the effects of the policy, the task, and prediction errors. None isolates action fidelity. Marginal distributional metrics are blind to it, since permuting futures across actions leaves their distribution unchanged, and reference-based metrics rest on an assumption that fails: that a prediction closer to the recorded future better follows its action. A sharp video that drifts to a different action can score above a blurred one that performs exactly the commanded motion (Figure 1); on a world model's own predictions, PSNR ranks the pixel average of the executed futures above every checkpoint (Section 4.2); and on the controlled pairs of Section 3, SSIM and LPIPS prefer the future that deviates from the action more often than not.

Scores that do take the action as input recover it from the generated video with an inverse dynamics model and compare it with the command (Kim et al., 2021; Arai et al., 2024; Rigter et al., 2025; Guo et al., 2025), which asks whether the action can be read off a single video. We ask a relative question: among the futures that a state admits, which one does this action produce? This holds a prediction responsible only for consequences that a video can show (Appendix B). Judging whether a future is consistent with a history and an action is simpler than generating it, an asymmetry that work on action verification also exploits (Liu et al., 2026), and contrastive learning is designed to learn such judgments (van den Oord et al., 2018; Zheng et al., 2023; 2024). Concurrently, WorldEcho (Chen et al., 2026a) replays candidate actions from a restored state as a reference for a trajectory error; we use such alternatives as the negatives of a reference-free score (Appendix A reviews related work).

Learning the judgment, however, raises a problem of its own: *what a contrastive score measures is decided by what it is contrasted against*. Against futures of other trajectories, a future can be matched to its history by recognizing the scene, without the action. A scorer trained this way on LIBERO demonstrations selects the right future in 99.29% of its training comparisons; asked which of eight futures executed from one state an action produced, it is right in exactly 12.50%, the accuracy of any score that ignores the action. Section 2.1 shows why: a score that ignores the action can drive the cross-history loss toward zero, but not a same-state loss below $\log K$. With data fixed, the negatives alone decide: on RoboTwin, replacing eight of fifteen cross-history negatives with futures of other actions from the same state raises retrieval from 55.2% to 97.3%.

We therefore contrast each future with futures produced by executing alternative actions from the same state. They share the scene and the history, so only the consequences of the actions tell them apart. The resulting score, LAFS (Learned Action-Fidelity Score), is a small network (26M parameters) trained only on clean executions. It rates a predicted future from its history and action alone, without a reference rollout, so it also applies to actions that were never executed, as in planning.

To test metrics against action outcomes established independently of them, we introduce BranchPairs, in which every case pairs an executed future that follows the queried action with one from the same state that deviates from it, each standing for an error that world models make: hedging blur, generation noise, exposure drift, a plausible rollout of a different action, late divergence, or wrong

timing. It covers manipulation in LIBERO (Liu et al., 2023) and RoboTwin (Chen et al., 2026b) and a controlled domain built from MNIST digits (LeCun et al., 1998), and none of its appearance degradations appears in training. On LIBERO, LAFS prefers the future that follows the action in 89% of pairs, against 57% for PSNR and 75% for an inverse dynamics model trained on the same data, and in 93% of pairs matched in PSNR (versus 80% and at most 60% for visual metrics). On a Cosmos world model fine-tuned on LIBERO-10, it orders three checkpoints and three controls as an executed-future reference does, unlike PSNR and a VLM critic. Our contributions are:

- We show that what a contrastive score measures is decided by what it is contrasted against. A scorer that picks a future among futures of other histories can succeed by recognizing the scene while ignoring the action; we observe this on LIBERO and characterize the estimand of each choice of negatives (Proposition 1): cross-history negatives yield a scene term plus an action term, same-state negatives the action term alone, so among same-state futures no score that ignores the action beats chance (Section 2.1).
- We propose LAFS, a score trained on such same-state comparisons of clean executions, which rates a predicted future from its history and action without a reference video (Section 2).
- We introduce BranchPairs, which pairs, from one state, an executed future that follows the queried action with one that deviates from it, and use it with a fine-tuned Cosmos world model to compare LAFS with pixel, perceptual, inverse-dynamics, and VLM metrics (Sections 3 and 4).

2 Learning an Action-Fidelity Score

Setup. At time t, a history $H = (o_{t-L+1}, \dots, o_t)$ of L frames and T controls $A = (a_t, \dots, a_{t+T-1})$ form a query (H, A), whose observed future $Y = (o_{t+1}, \dots, o_{t+T})$ is drawn from $p(Y \mid H, A)$; an action-conditioned world model predicts $\widehat{Y}$ from the same query. We seek a score $s_\theta(H, A, Y)$ (Figure 2) that measures whether $\widehat{Y}$ depicts the consequences of A rather than those of another action available in the same state. It sees only observations and controls, so any world model can be evaluated through its input–output interface. A branch group $\mathcal{B} = \{(H, A_j, Y_j)\}_{j=1}^{K}$ collects the observed futures of K action sequences executed from the state of one history; $\pi(A \mid H)$ is the distribution of branch actions, $p(Y \mid H) = \int p(Y \mid H, a)\, \pi(da \mid H)$ the resulting distribution of futures from that state, and $p(Y)$ the distribution of futures across histories.

2.1 Training comparisons that depend on the action

A contrastive objective learns a score by identifying a matched observation among alternatives (van den Oord et al., 2018). For a query with observed future Y^+ and a candidate set $\mathcal{C} = \{Y^+\} \cup \mathcal{N}$ that adds negative futures $\mathcal{N}$, the InfoNCE loss is

$$\mathcal{L}(\theta) = -\mathbb{E}\left[\log \frac{\exp s_\theta(H, A, Y^+)}{\sum_{Y \in \mathcal{C}} \exp s_\theta(H, A, Y)}\right]. \tag{1}$$

What the score learns depends on what the negatives share with the positive. *Cross-history negatives* follow other histories, typically the other examples in a batch (Chen et al., 2020), as in action-conditioned objectives for control (Kipf et al., 2020; Eysenbach et al., 2022; Zheng et al., 2023); we call them *cross-task* when they come from other tasks. *Within-trajectory negatives* come from other times of the same demonstration (Sermanet et al., 2018; Zheng et al., 2024). Neither holds the state fixed while varying the action. We add *same-state branches*: futures of other action sequences executed from the positive's state. They share its history, so only the consequences of the actions tell them apart. For the query (H_i, A_{ij}) of branch j in group i, the positive is Y_{ij}, the same-state negatives are futures Y_{ik} with $k \neq j$, and the cross-history negatives are futures of other histories; the robotic configuration uses eight same-state and seven cross-history negatives per positive. Every branch is a positive for its own action, and the candidate set is needed only in training (Appendix B).

What the negatives estimate. The minimizer of Eq. (1) is, up to a term that depends only on the query, the log density ratio between the positive's distribution and that of the negatives (van den Oord et al., 2018; Ma & Collins, 2018; Poole et al., 2019), the principle by which negatives that share a variable remove its information (Tsai et al., 2022).

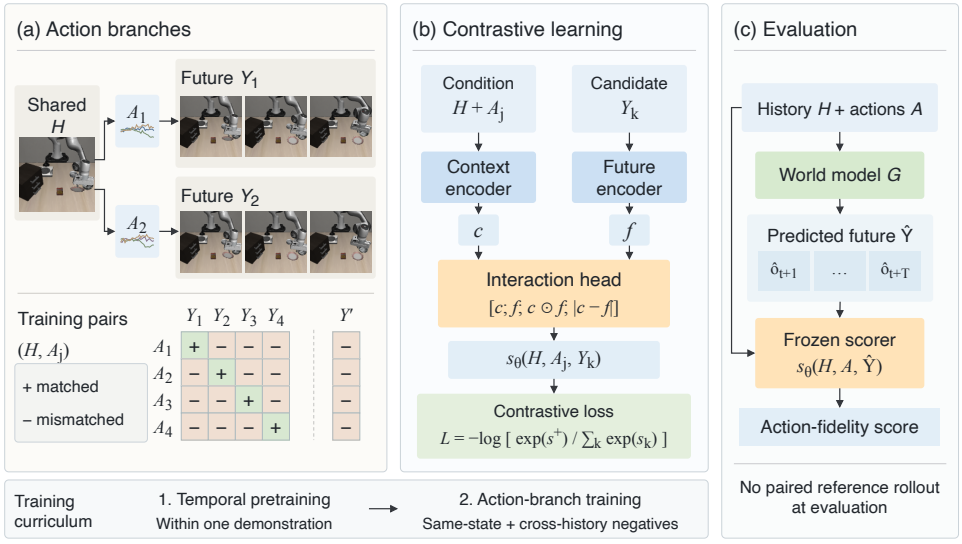


Figure 2: **Learning and using an action-fidelity score. (a)** Alternative action sequences executed from a shared LIBERO state; each future is positive for its own action, and other branches and cross-history futures (Y') supply negatives (schematic labels, not measured scores). **(b)** Context and future representations interact to produce a scalar score, trained first on within-demonstration windows and then on mixed branch and cross-history comparisons. **(c)** The frozen scorer rates a world-model prediction from its history and action, without a reference rollout.

Proposition 1 (cf. van den Oord et al., 2018; Ma & Collins, 2018)**.** *Let negatives be drawn i.i.d. and independently of the positive from a distribution $q(Y \mid H)$ that does not depend on the queried action and is mutually absolutely continuous with $p(Y \mid H, A)$. Over unrestricted scores, any minimizer of Eq. (1) satisfies $s^*(H, A, Y) = \log \frac{p(Y|H,A)}{q(Y|H)} + c(H, A)$ for some offset $c(H, A)$ that does not depend on Y. In particular,*

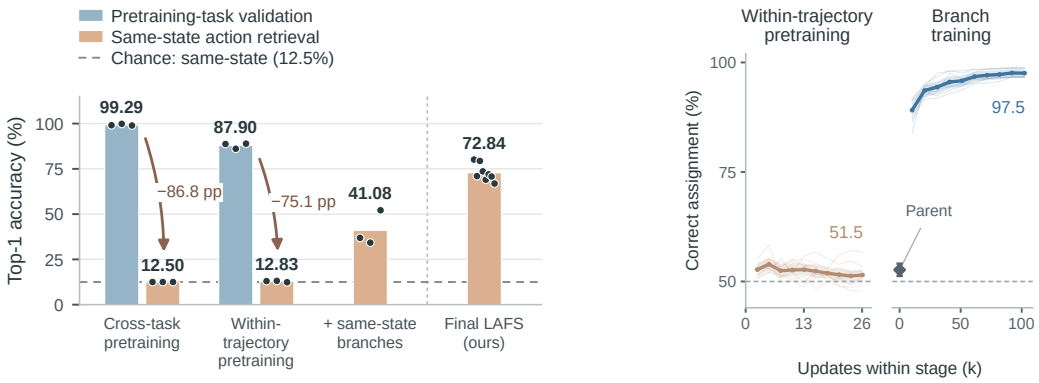
$$\textit{cross-history negatives, } q = p(Y): \quad s^* = \underbrace{\log \frac{p(Y \mid H)}{p(Y)}}_{\textit{history}} + \underbrace{\log \frac{p(Y \mid H, A)}{p(Y \mid H)}}_{\textit{action}} + c(H, A), \tag{2}$$

$$\textit{same-state negatives, } q = p(Y \mid H): \quad s^* = \log \frac{p(Y \mid H, A)}{p(Y \mid H)} + c(H, A). \tag{3}$$

With independently drawn branch actions, the action term equals $\log \pi(A \mid H, Y)/\pi(A \mid H)$, where $\pi(A \mid H, Y)$ is the posterior of the branch action given its future: the evidence that the future gives for the queried action among the branches. Training does not identify the offset $c(H, A)$; it cancels within a query and across models on the same queries, and comparability across conditions (Section 4.1) is a property of the trained network, not of the objective. The contrast between the two kinds of negatives holds for any collected group of branches:

Lemma 1. *Consider K queries $(H, A_1), \ldots, (H, A_K)$ that share a history and whose candidate sets contain their futures $Y_1, \ldots, Y_K$, with Y_j the positive of query j. If a score does not depend on the queried action, then (a) its retrieval accuracy among $Y_1, \ldots, Y_K$ is exactly $1/K$ under any query-independent tie-breaking rule, and (b) its loss in Eq. (1), averaged over the K queries, is at least $\log K$, with equality only if it scores them equally and the candidate sets contain nothing else.*

When futures of different histories are separated, a score that ignores the action can drive the cross-history loss arbitrarily close to zero (Proposition 2), so low loss against cross-history negatives carries no information about the action, whereas a same-state loss below $\log K$ requires using it. On demonstration data the point is sharper: each history appears with one action, so no choice of negatives lets the objective tell a score that uses the action from one that ignores it. Branches make the action vary within a state, and same-state negatives make that variation the only thing to tell apart.



(a) High accuracy, weak action discrimination

(b) Action–future assignment

Figure 3: **Training-task accuracy does not ensure action discrimination. (a)** Same-state retrieval of the future produced by the supplied action among eight executed branches (100 pre-grasp groups, 800 queries; chance 12.5%), beside each pretraining task's own validation accuracy. The first three groups share an earlier architecture and three seeds and differ only in their negatives; right of the separator is final LAFS (eight seeds). Bars: seed means; dots: seeds. **(b)** Fraction of action pairs whose correct action–future assignment scores above the swapped one (chance 50%) during within-trajectory pretraining and branch training, on 100 development conditions; thick lines average eight seeds, shading: 95% condition-bootstrap intervals; the diamond marks the start of branch training.

Unlike an inverse model regressing the action, the same-state estimand ranks futures identically for actions with indistinguishable outcomes (Appendix B).

We therefore report same-state retrieval beside each training task's own validation accuracy (Figure 3a). On LIBERO, cross-task or within-trajectory negatives alone reach high validation accuracy but leave retrieval at chance, cross-task pretraining at exactly 12.50% in all three seeds, as Lemma 1(a) predicts; same-state branches lift it, and final LAFS reaches 72.8%.

2.2 Training schedule and scoring network

We first train with within-trajectory negatives on recorded demonstrations and then fine-tune on branch comparisons with the same loss and network; the first stage never contrasts two actions from one state, but it makes the branch task learnable (Section 4.4). The query and the candidate future are encoded separately: convolutional stems and temporal transformers encode the videos (e_H, e_Y), and a transformer encodes the controls (e_A). With projection heads P_ϕ and P_ψ and ℓ_2 normalization,

$$\phi = \mathrm{norm}(P_\phi([e_H(H); e_A(A)])), \qquad \psi = \mathrm{norm}(P_\psi(e_Y(Y))), \tag{4}$$

so that ϕ is a unit-norm embedding of the query (history and action) and ψ one of the candidate future. A small head h_θ scores their interaction rather than their cosine similarity alone:

$$u = h_\theta([\phi; \psi; \phi \odot \psi; |\phi - \psi|]), \qquad s_\theta(H, A, Y) = \frac{\mathrm{norm}(w)^\top \mathrm{norm}(u)}{\tau}. \tag{5}$$

Here w is the learned weight vector of the final linear layer and τ a temperature. Fixing $\tau = 0.1$ bounds the score, $|s_\theta| \le 10$, keeping logits and their gradients bounded while permitting a logit gap of $2/\tau$; without it, the logit scale differed by orders of magnitude across seeds (Appendix B). After training, the scorer is frozen and rates a prediction with $s_\theta(H, A, \widehat{Y})$, without a paired reference rollout or alternative futures. Action-MNIST uses a joint history–action–future encoder (Appendix G).

3 Evaluation Protocol

We evaluate a metric by whether its judgments agree with action consequences established independently of it, and collect such tests in BranchPairs, built on executed action branches. A condition fixes a history and the actions executed from its state; a pair contrasts a future that follows the

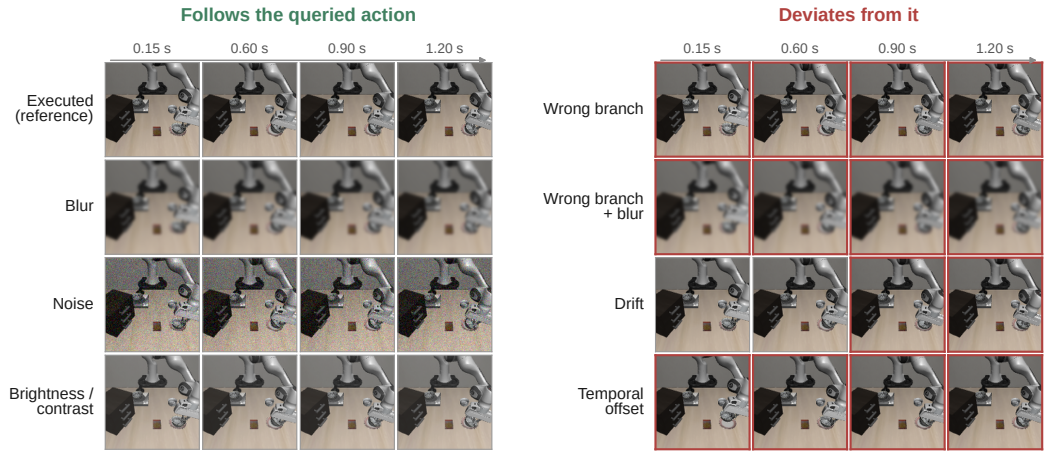


Figure 4: **BranchPairs pairs futures that follow the queried action with futures that deviate from it.** One LIBERO condition, four of the eight frames of each 1.2 s future. **Left:** the executed future of the queried action and appearance changes that preserve it: Gaussian blur ($\sigma = 2$), and noise and a brightness/contrast shift of matched PSNR. **Right:** the executed future of another action from the same state, clean or blurred; drift, which follows the queried action for the first half; and a two-frame delay. Red frames deviate from the queried action. A metric passes a pair if it prefers the left future; every case and severity is listed in Appendix Table 6.

queried action, possibly under an appearance change, with one that deviates from it, and a metric passes if it prefers the former (Figure 4). Each case stands for an error that world models make, fixed before any metric is scored (Appendix Table 6). LAFS and the inverse-dynamics baseline train only on clean executed futures, which only temporal offsets resemble. On LIBERO, BranchPairs uses 100 pre-grasp conditions with 100 pairs each, so every condition and case has a fixed weight, as on RoboTwin and Action-MNIST (Appendix H).

For a metric m, let $m_i(Y)$ be its value for a candidate Y of condition i, oriented so that larger values are preferred. Over a set $\mathcal{T}$ of triples $(i, \tilde{Y}^+, \tilde{Y}^-)$, each pairing a candidate $\tilde{Y}^+$ that preserves the query's action consequences with a candidate $\tilde{Y}^-$ that has an action error, its pairwise accuracy is

$$\mathrm{PairAcc}(m; \mathcal{T}) = \frac{1}{|\mathcal{T}|} \sum_{(i,\tilde{Y}^+,\tilde{Y}^-)\in\mathcal{T}} \left[\mathbf{1}\{m_i(\tilde{Y}^+) > m_i(\tilde{Y}^-)\} + \tfrac{1}{2}\mathbf{1}\{m_i(\tilde{Y}^+) = m_i(\tilde{Y}^-)\} \right]. \quad (6)$$

where $\mathbf{1}\{\cdot\}$ is the indicator function, so ties count one half. Action-MNIST also reports AUC. As a control for pixel error, we keep the 499 LIBERO pairs whose PSNR against the recorded future differs by less than 0.25 dB, also weighting the nine case combinations equally (case-balanced); the control is needed and feasible only on LIBERO, whose alternative actions differ little in pixels (on RoboTwin, 17 of 40,950 pairs qualify). Intervals resample condition groups with all methods paired.

4 Experiments

LIBERO (Liu et al., 2023) is our primary testbed, with 100 pre-grasp condition groups of the spatial suite and eight action queries each (rendering in Appendix B); RoboTwin (Chen et al., 2026b), with 525 two-arm groups, replicates the main comparisons; and Action-MNIST, built from MNIST digits (LeCun et al., 1998), whose action outcomes are all known, provides controlled diagnostics. Outside ablations, each suite uses one final LAFS model.

Baselines. We compare with PSNR, SSIM (Wang et al., 2004), LPIPS (Zhang et al., 2018), and DINOv2 feature distance (Oquab et al., 2024) against the recorded future. For action recovery (Arai et al., 2024; Rigter et al., 2025), an inverse dynamics model trained on the same data, with the same video encoders and initialization, regresses the controls, and a candidate is scored by the negative squared error between recovered and commanded actions; like our score, it needs no reference. Set-level distances such as FID (Heusel et al., 2017) do not score individual videos.

Table 1: **Action-correctness accuracy.** Percentage of pairs in which a metric prefers the action-consistent future (chance 50) on the main cases of BranchPairs: 10,000 pairs from 100 conditions on LIBERO and 21,000 pairs from 525 conditions on RoboTwin; LAFS is averaged over eight seeds. Paired GT: whether a reference future is required. †Trained on the same data as our score; mean of three seeds (our score reaches 90.04 on LIBERO with the same seeds). ‡AUC over the labeled construction set, with the joint encoder of Appendix G.

Metric	Paired GT	Action-MNIST‡	LIBERO	RoboTwin
PSNR	Yes	82.85	57.11	99.70
SSIM	Yes	72.36	41.68	77.58
LPIPS	Yes	45.78	45.20	71.75
DINOv2 distance	Yes	65.32	50.69	75.37
Inverse dynamics†	No	–	75.08	53.60
LAFS (ours)	No	**91.38**	**88.63**	99.06

4.1 Do metrics tell action errors from visual changes?

On the 10,000 main LIBERO pairs (Table 1), our score prefers the action-consistent future in 88.6% of pairs, against 75.1% for inverse dynamics, 57.1% for PSNR, and 50.7% for DINOv2; SSIM (41.7%) and LPIPS (45.2%) fall below chance, because appearance changes that preserve the action can lower visual similarity more than an alternative action does. Reference-based metrics pass the 1,000 clean pairs by construction; without them, PSNR falls to 52.3% and DINOv2 to 45.2%, while ours stays at 87.8%. With the nine case combinations weighted equally, the ordering is unchanged, and the hardest case, a noisy correct future against a drift, is 47.9% for our score and at most 26.0% for the others (Appendix H). On Action-MNIST, PSNR is strong, yet our score leads.

Controlling for pixel error. On the 499 LIBERO pairs matched in PSNR, our score reaches 93.5% (92.8% case-balanced), against 80.1% for inverse dynamics and at most 59.9% for the visual metrics (Table 8): it is not a proxy for pixel error. With the same three training seeds, it exceeds inverse dynamics by 15.0 and 14.0 points on the main and matched pairs (95% intervals [13.0, 16.8] and [10.5, 17.7]) and for every paired seed. Regression must recover every control value, including those the video does not determine, whereas our score only decides whether a future matches the action.

RoboTwin. Alternative bimanual executions differ strongly in pixels, so PSNR separates nearly all 21,000 main pairs (99.7%; ours 99.1%), whereas inverse dynamics reaches only 53.6%: recovering fourteen bimanual control values per step from video is harder than deciding which executed future an action produced. Pixel error suffices when an action error changes many pixels and the executed future is available.

One threshold across conditions. Flagging poor predictions needs one threshold across conditions, where the offset $c(H, A)$ no longer cancels (Figure 5). On LIBERO, the ROC area of our score falls only from 0.89 within conditions to 0.85, against 0.41–0.56 for the visual metrics, which are precise only on positives they cannot tell from the recorded future; its average precision is 0.82. On Action-MNIST it leads in ROC area (0.91 versus 0.83 for PSNR).

4.2 Does LAFS track a world model through training?

A world model that learns to follow its actions should, as training proceeds, predict futures that increasingly depict the outcome of the commanded action. We fine-tune a Cosmos3-Edge world model (NVIDIA, 2026) on LIBERO-10 as a forward-dynamics model (Appendix D). Checkpoints at 1,000, 5,000, and 9,000 iterations predict the future under each of eight executed actions in 40 held-out conditions, and LAFS is trained on branches from the training demonstrations (Appendix F).

Figure 6 ranks six predictors: the three checkpoints and three controls built from the executed futures, namely the futures themselves, their pixel average (branch mean), the squared-error optimum among predictions that ignore the action, and the futures paired with another action (permuted). Ac-

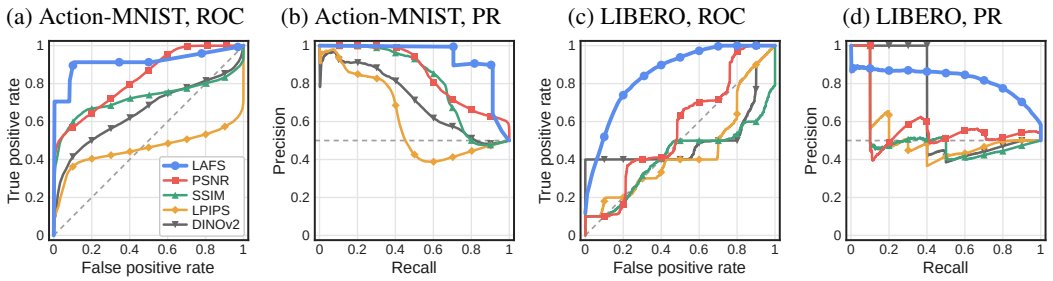


Figure 5: **LAFS stays accurate with one threshold shared across all conditions.** ROC and precision–recall curves on Action-MNIST (3,200 anchors) and the LIBERO main cases (100 conditions; 1,000 candidates per class, JPEG excluded); precision is at equal class proportions, and dashed lines mark chance. LIBERO LAFS: mean over eight seeds. Average precision of LAFS is 0.93 on Action-MNIST (PSNR 0.86) and 0.82 on LIBERO (DINOv2 0.67, PSNR 0.58).

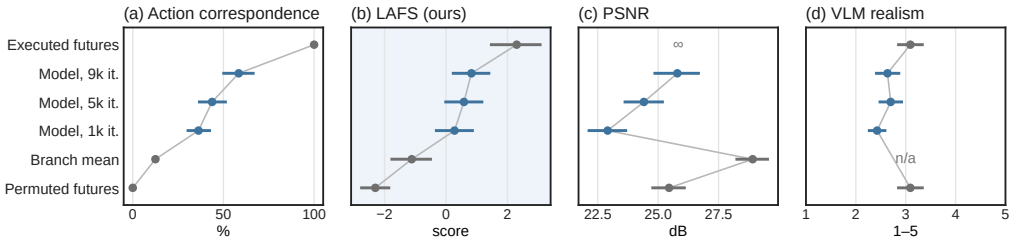


Figure 6: **Only LAFS ranks predictions like the reference.** Six predictors on 40 LIBERO-10 conditions, 8 actions each; rows follow the reference order, bars are 95% condition-bootstrap intervals, and a line joins the rows. **(a)** Action correspondence (chance 12.5%). **(b)** Mean LAFS score, reference-free. **(c)** PSNR to the queried action's executed future. **(d)** Mean VLM physical-realism rating from the instruction and the video.

tion correspondence, the reference, asks whether a prediction is nearest, among the eight executed futures, to that of its own action (Appendix D); it rises from 36.3% to 43.8% and 58.4% over training (9,000 versus 1,000 iterations: 22.2 points, 95% CI [14.1, 30.6]). LAFS, which scores each prediction from its history and action alone, ranks all six predictors in the reference's order and resolves the gain from 1,000 to 9,000 iterations (0.55, 95% CI [0.13, 0.97]). PSNR ranks the branch mean first, and a VLM critic that rates physical realism from the instruction and the video, as world-model benchmarks do (Appendix E), rates the permuted futures as highly as the executed ones.

4.3 Does the score use the supplied action?

Holding histories and candidate futures fixed, we intervene on the action input. On RoboTwin (eight seeds, 40,950 pairs), zeroing the action embedding lowers the preference for the action-consistent future from 99.4% to 33.7%, and reassigning actions within the condition to 43.8% (paired drop 55.6 points, 95% CI [53.1, 57.9]); on LIBERO, a variant trained with fifteen same-state negatives falls from 78.6% to 52.0% and 40.6% on an earlier set of 659 PSNR-matched pairs. Figure 3b tracks this through training with $D_{jk} = (s_{jj} + s_{kk} - s_{jk} - s_{kj})/2$, $s_{jk} = s(H, A_j, Y_k)$, which compares the correct assignment of two actions to their futures with the swapped one and cancels additive action and future offsets: the fraction of pairs with $D_{jk} > 0$ stays near chance in pretraining and rises from 52.6% at the start of branch training to 97.5% (95% CI of the gain [43.1, 46.6]; every seed improves). Figure 7 shows where this lives in the representation: we split the variance of the normalized readout features of all 6,400 action–future pairs exactly into the part shared within a condition (scene), the part that depends on which action is paired with which future (pairing, the feature counterpart of D_{jk}), and the rest. Before branch training, the scene holds 91.7% and the pairing 0.1%; after it, the pairing is the largest part (38.9%), in every condition, while task identity stays linearly decodable (89% and 83% on held-out conditions; chance 10%). The two checkpoints also differ in training time; the negatives alone are compared next.

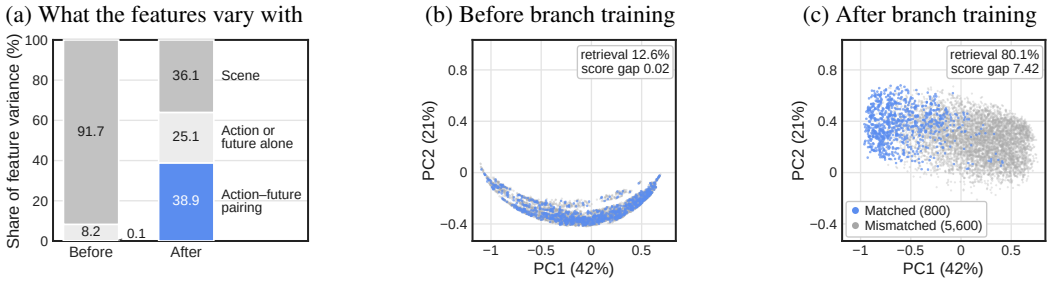


Figure 7: **Branch training reorganizes the features around the action–future pairing.** Readout features (Eq. 5) of all 6,400 action–future pairs in 100 development conditions (Figure 3b), before and after branch training (one preselected seed). **(a)** Exact variance split into scene, pairing, and the rest (Section 4.3). **(b, c)** One PCA fitted to both checkpoints; blue: a future with the action that produced it. Retrieval: same-state 8-way (chance 12.5%); gap: matched minus mismatched score.

Table 2: **The negatives decide what the score learns.** Same-state 8-way retrieval on LIBERO (%; chance 12.5), mean ± SD. A (earlier architecture): + rows add pretraining to branch training from scratch. B: + rows add one change each; indented rows: variants. Shaded: final setting.

Configuration	Pre-grasp ↑	At-grasp ↑	Free-space ↑
A Curriculum			
Cross-task pretraining only	12.50 ± 0.00	12.46 ± 0.07	12.50 ± 0.00
Within-trajectory pretraining only	12.83 ± 0.40	12.62 ± 0.38	13.09 ± 0.29
Branch training from scratch	23.17 ± 19.13	–	–
+ cross-task pretraining	20.88 ± 13.53	20.04 ± 14.50	26.30 ± 23.70
+ within-trajectory pretraining	41.08 ± 9.65	45.92 ± 15.38	58.12 ± 23.19
+ both (cross-task, then within-trajectory)	47.54 ± 2.57	59.62 ± 9.43	74.54 ± 6.11
B Readout and negatives			
Branch training, 15 same-state negatives	56.02 ± 7.41	69.00 ± 10.08	83.60 ± 7.62
+ bounded readout	69.34 ± 4.71	83.11 ± 5.24	92.91 ± 3.91
w/ shorter warmup (5%)	62.80 ± 3.63	74.72 ± 6.55	90.25 ± 3.57
+ mixed negatives (8 same-state, 7 cross-history)	72.84 ± 4.71	84.67 ± 4.48	93.43 ± 2.75
w/o the 7 cross-history negatives	68.91 ± 5.35	81.39 ± 6.71	92.50 ± 3.04

4.4 Which training choices matter?

In an earlier study (Table 2A), cross-task or within-trajectory negatives alone leave same-state retrieval at chance and the same-state loss above $\log 8$ in every seed, whereas branch training after within-trajectory pretraining lowers it to 0.87–1.52, below what Lemma 1(b) allows a score that ignores the action: branch training from scratch is unreliable, and within-trajectory pretraining makes it learnable without solving it. In the current model (B), the bounded readout and mixing eight same-state with seven cross-history negatives each raise retrieval; the mix also raises metric quality by 2.76 points (95% CI [2.12, 3.44]; matched: 2.14, [0.87, 3.38]); on RoboTwin, they raise retrieval from 55.2% to 97.3% (Table 4).

5 Conclusion

What a contrastive score measures is decided by what it is contrasted against: futures executed from the same state teach it the action, whereas futures of other histories teach it the scene, so a learned metric should be reported with its accuracy on comparisons that vary only the property it claims to measure. Trained this way on clean executions alone, LAFS leads pixel, perceptual, and inverse-dynamics baselines on BranchPairs, including pairs matched in PSNR, and ranks a world model's checkpoints and controls as an executed-future reference does, whereas PSNR ranks the hedged average first. Its cost is paid once, at training time, by executing alternative actions in the evaluated domain, and the construction applies wherever such alternatives can be executed (Appendix C discusses limitations).

ETHICS STATEMENT

This work evaluates action-conditioned video world models in simulation. It involves no human subjects, user studies, or personal data: all videos are rendered in the LIBERO and RoboTwin simulators or built from MNIST digits, and the world models we evaluate are fine-tuned on public simulated demonstrations. LAFS measures only whether a predicted future follows the supplied action; it does not measure task success, safety, or visual quality, and a high score is not evidence that a model is safe to use on a physical robot. Better tests of action fidelity can help reveal world models that would mislead planning or policy evaluation before they are used on real systems.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix B specify the scoring network, the objective, and the training schedule, and Appendix J contains the proofs. Section 3 and Appendix H define the cases of BranchPairs, and Appendices B and D describe rendering and world-model training. Training seeds, checkpoint selection, and confidence intervals are stated with each result. The supplementary material contains an anonymized excerpt of the scorer, its training and evaluation code, and the re-rendering script. We will release the full code, the BranchPairs constructions, and the trained scores.

AI USE STATEMENT

We acknowledge the use of large language models (LLMs) as auxiliary tools in the preparation and implementation of this work. Specifically, LLMs were used for language polishing, literature search and retrieval, assistance with selected implementation and figure-generation code, and assistance with mathematical derivations and proofs in the appendix. All AI-assisted outputs, including code, references, mathematical arguments, and manuscript edits, were critically reviewed and verified by the authors before inclusion.

The research problem, methodological design, experimental design, interpretation of results, and scientific conclusions were determined by the authors. We critically reviewed and edited all text generated or modified by the LLM to ensure it accurately reflects our original thoughts and findings.

REFERENCES

Hadi Alzayer, Wenlong Huang, Haonan Chen, Christopher Luey, Lvmin Zhang, Maneesh Agrawala, Gordon Wetzstein, Li Fei-Fei, Yilun Du, Jiajun Wu, and Jia-Bin Huang. Masked visual actions for unified world modeling. *arXiv preprint arXiv:2607.19343*, 2026. URL `https://arxiv.org/abs/2607.19343v1`. Version 1.

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.

Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, et al. V-JEPA 2: Self-supervised video models enable understanding, prediction and planning. *arXiv preprint arXiv:2506.09985*, 2025.

Bowen Baker, Ilge Akkaya, Peter Zhokhov, Joost Huizinga, Jie Tang, Adrien Ecoffet, Brandon Houghton, Raul Sampedro, and Jeff Clune. Video PreTraining (VPT): Learning to act by watching unlabeled online videos. In *Advances in Neural Information Processing Systems*, 2022. arXiv:2206.11795.

Hritik Bansal, Zongyu Lin, Tianyi Xie, Zeshun Zong, Michal Yarom, Yonatan Bitton, Chenfanfu Jiang, Yizhou Sun, Kai-Wei Chang, and Aditya Grover. VideoPhy: Evaluating physical commonsense for video generation. In *International Conference on Learning Representations*, 2025.

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*, 2024. arXiv:2402.15391.

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*, 2026a.

Tianxing Chen, Zanxin Chen, Baijun Chen, Zijian Cai, Yibin Liu, Zixuan Li, Qiwei Liang, Xianliang Lin, Yiheng Ge, Zhenyu Gu, et al. RoboTwin 2.0: A scalable data generator and benchmark with strong domain randomization for robust bimanual robotic manipulation. In *International Conference on Machine Learning*, 2026b.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International Conference on Machine Learning*, 2020.

Peterson Co, Sicheng Hu, Chunxuan Jiao, et al. WorldSimProbe: Diagnosing simulator faithfulness in action-conditioned world models for embodied manipulation. *arXiv preprint arXiv:2608.09298*, 2026.

Terrance DeVries, Adriana Romero, Luis Pineda, Graham W. Taylor, and Michal Drozdzal. On the evaluation of conditional GANs. *arXiv preprint arXiv:1907.08175*, 2019.

Haoyi Duan, Hong-Xing Yu, Sirui Chen, Li Fei-Fei, and Jiajun Wu. WorldScore: A unified evaluation benchmark for world generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 27713–27724, 2025.

Benjamin Eysenbach, Ruslan Salakhutdinov, and Sergey Levine. C-learning: Learning to achieve goals via recursive classification. In *International Conference on Learning Representations*, 2021. arXiv:2011.08909.

Benjamin Eysenbach, Tianjun Zhang, Sergey Levine, and Ruslan Salakhutdinov. Contrastive learning as goal-conditioned reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

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. arXiv:2405.17398.

Junliang Guo, Yang Ye, Tianyu He, Haoyu Wu, Yushu Jiang, Tim Pearce, and Jiang Bian. MineWorld: A real-time and open-source interactive world model on Minecraft. *arXiv preprint arXiv:2504.08388*, 2025.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, 2018.

Xuan He, Dongfu Jiang, Ge Zhang, et al. VideoScore: Building automatic metrics to simulate fine-grained human feedback for video generation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024.

Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. arXiv:1706.08500.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. arXiv:2311.17982.

Seung Wook Kim, Yuhao Zhou, Jonah Philion, Antonio Torralba, and Sanja Fidler. Learning to simulate dynamic environments with GameGAN. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1231–1240, 2020. arXiv:2005.12126.

Seung Wook Kim, Jonah Philion, Antonio Torralba, and Sanja Fidler. DriveGAN: Towards a controllable high-quality neural simulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. arXiv:2104.15060.

Thomas Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020. arXiv:1911.12247.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Dacheng Li, Yunhao Fang, Yukang Chen, Shuo Yang, Shiyi Cao, Justin Wong, Michael Luo, Xiaolong Wang, Hongxu Yin, Joseph E. Gonzalez, Ion Stoica, Song Han, and Yao Lu. WorldModelBench: Judging video generation models as world models. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

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. arXiv:2306.03310.

Yaofang Liu, Xiaodong Cun, Xuebo Liu, Xintao Wang, Yong Zhang, Haoxin Chen, Yang Liu, Tieyong Zeng, Raymond Chan, and Ying Shan. EvalCrafter: Benchmarking and evaluating large video generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. arXiv:2310.11440.

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. URL `https://arxiv.org/abs/2604.01985v2`. Version 2.

Zhuang Ma and Michael Collins. Noise contrastive estimation and negative sampling for conditional models: Consistency and statistical efficiency. In *Conference on Empirical Methods in Natural Language Processing*, 2018.

David McAllester and Karl Stratos. Formal limitations on the measurement of mutual information. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2020.

NVIDIA. Cosmos world foundation model platform for physical AI. *arXiv preprint arXiv:2501.03575*, 2025.

NVIDIA. Cosmos 3: Omnimodal world models for physical AI. *arXiv preprint arXiv:2606.02800*, 2026.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. arXiv:2304.07193.

Ben Poole, Sherjil Ozair, Aäron van den Oord, Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *International Conference on Machine Learning*, 2019.

Marc Rigter, Tarun Gupta, Agrin Hilmkil, and Chao Ma. AVID: Adapting video diffusion models to world models. *Reinforcement Learning Journal*, 2025. arXiv:2410.12822.

Pierre Sermanet, Corey Lynch, Yevgen Chebotar, Jasmine Hsu, Eric Jang, Stefan Schaal, and Sergey Levine. Time-contrastive networks: Self-supervised learning from video. In *IEEE International Conference on Robotics and Automation*, 2018.

Yu Shang, Zhuohang Li, Yiding Ma, Weikang Su, Xin Jin, Ziyou Wang, Lei Jin, Xin Zhang, Yinzhou Tang, Haisheng Su, Chen Gao, Wei Wu, Xihui Liu, Dhruv Shah, Zhaoxiang Zhang, Zhibo Chen, Jun Zhu, Yonghong Tian, Tat-Seng Chua, Wenwu Zhu, and Yong Li. WorldArena: A unified benchmark for evaluating perception and functional utility of embodied world models. *arXiv preprint arXiv:2602.08971*, 2026.

Michael Soloveitchik, Tzvi Diskin, Efrat Morin, and Ami Wiesel. Conditional Frechet inception distance. *arXiv preprint arXiv:2103.11521*, 2021.

Nitish Srivastava, Elman Mansimov, and Ruslan Salakhutdinov. Unsupervised learning of video representations using LSTMs. In *International Conference on Machine Learning (ICML)*, 2015.

George Stein, Jesse C. Cresswell, Rasa Hosseinzadeh, Yi Sui, Brendan Leigh Ross, Valentin Villecroze, Zhaoyan Liu, Anthony L. Caterini, J. Eric T. Taylor, and Gabriel Loaiza-Ganem. Exposing flaws of generative model evaluation metrics and their unfair treatment of diffusion models. In *Advances in Neural Information Processing Systems*, 2023.

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.

Yao-Hung Hubert Tsai, Tianqin Li, Martin Q. Ma, Han Zhao, Kun Zhang, Louis-Philippe Morency, and Ruslan Salakhutdinov. Conditional contrastive learning with kernel. In *International Conference on Learning Representations*, 2022.

Wei-Cheng Tseng, Gashon Hussein, Yuzhu Dong, Allen Z. Ren, Lucy X. Shi, XuDong Wang, Sergey Levine, Zhaoshuo Li, Jinwei Gu, Florian Shkurti, Ming-Yu Liu, and Quan Vuong. SC3-Eval: Evaluating robot foundation models via self-consistent video generation. *arXiv preprint arXiv:2606.18610*, 2026.

Thomas Unterthiner, Sjoerd van Steenkiste, Karol Kurach, Raphael Marinier, Marcin Michalski, and Sylvain Gelly. Towards accurate generative models of video: A new metric & challenges. *arXiv preprint arXiv:1812.01717*, 2018.

Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. URL https://arxiv.org/abs/1807.03748.

Zehan Wang, Tengfei Wang, Haiyu Zhang, Xuhui Zuo, Junta Wu, Haoyuan Wang, Wenqiang Sun, Zhenwei Wang, Chenjie Cao, Hengshuang Zhao, Chunchao Guo, and Zhou Zhao. WorldCompass: Reinforcement learning for long-horizon world models. *arXiv preprint arXiv:2602.09022*, 2026.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

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. URL https://arxiv.org/abs/2607.09185v1. Version 1.

Jiazhi Yang, Shenyuan Gao, Yihang Qiu, Li Chen, Tianyu Li, Bo Dai, Kashyap Chitta, Penghao Wu, Jia Zeng, Ping Luo, Jun Zhang, Andreas Geiger, Yu Qiao, and Hongyang Li. Generalized predictive model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024a. arXiv:2403.09630.

Sherry Yang, Yilun Du, Seyed Kamyar Seyed Ghasemipour, Jonathan Tompson, Leslie Pack Kaelbling, Dale Schuurmans, and Pieter Abbeel. Learning interactive real-world simulators. In *International Conference on Learning Representations*, 2024b.

Yang Yu, Shiyuan Zhang, Yifei Sheng, Haoxiang Ren, and Haoxin Lin. How should world models be evaluated for embodied decision-making? A decision-making-centric position. *arXiv preprint arXiv:2606.15032*, 2026. URL https://arxiv.org/abs/2606.15032v2. Version 2.

Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? In *International Conference on Learning Representations (ICLR)*, 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. arXiv:1801.03924.

Xiangteng Zhang, Yang Guan, Bo Zhang, Hongyang Li, Ya-Qin Zhang, and Shengbo Eben Li. On the identifiability of controlled world models. *arXiv preprint arXiv:2607.22430*, 2026. URL `https://arxiv.org/abs/2607.22430v2`. Version 2.

Yifan Zhang, Chunli Peng, Boyang Wang, Puyi Wang, Qingcheng Zhu, Fei Kang, Biao Jiang, Zedong Gao, Eric Li, Yang Liu, and Yahui Zhou. Matrix-Game: Interactive world foundation model. *arXiv preprint arXiv:2506.18701*, 2025.

Ruijie Zheng, Xiyao Wang, Yanchao Sun, Shuang Ma, Jieyu Zhao, Huazhe Xu, Hal Daumé III, and Furong Huang. TACO: Temporal latent action-driven contrastive loss for visual reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023. arXiv:2306.13229.

Ruijie Zheng, Yongyuan Liang, Xiyao Wang, Shuang Ma, Hal Daumé III, Huazhe Xu, John Langford, Praveen Palanisamy, Kalyan Shankar Basu, and Furong Huang. Premier-TACO is a few-shot policy learner: Pretraining multitask representation via temporal action-driven contrastive loss. In *International Conference on Machine Learning*, 2024. arXiv:2402.06187.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *International Conference on Machine Learning*, 2025. arXiv:2411.04983.

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 Related work

Evaluating world models. World models are most often evaluated by the fidelity of the videos they produce. Distributional metrics such as FID and FVD compare sets of generated and real videos (Heusel et al., 2017; Unterthiner et al., 2018); Inception features misjudge modern generators, and DINOv2 features (Oquab et al., 2024) have been recommended in their place (Stein et al., 2023). Conditional variants such as FJD and CFID compare joint distributions of conditions and outputs and can detect a mismatch between them, but they need many paired samples (DeVries et al., 2019; Soloveitchik et al., 2021). Paired metrics compare each prediction with the recorded future, from PSNR and SSIM (Wang et al., 2004) to deep features that better match human similarity judgments (Zhang et al., 2018). VLM judges rate instruction following and physical commonsense from the instruction and the video (Li et al., 2025; Bansal et al., 2025), and a VLM critic can score imagined rollouts during planning (Alzayer et al., 2026); we include a single-video VLM rating in Section 4.2. Learned reference-free metrics score an output from its conditioning input, as CLIPScore does for image captions (Hessel et al., 2021), and VideoScore regresses human ratings of generated videos (He et al., 2024); LAFS is trained on executed alternatives rather than human ratings. Closed-loop evaluations run a policy inside the model and report its success or how well the model ranks policies (Shang et al., 2026; Tseng et al., 2026), and a recent position paper argues that claims about world models outrun the evidence these evaluations provide (Yu et al., 2026). For action fidelity, each of these misses something different: a marginal set-level comparison is unchanged when futures are permuted across actions, a paired error mixes action errors with appearance changes (Section 4.1), a judge that is not given the action cannot condition on it, and a closed-loop score also depends on the policy and the task. LAFS takes the conditioning action as an explicit input and complements these evaluations.

Action-conditioned evaluation. The standard test of whether a generated video follows its command is action recovery: an inverse dynamics model infers the action from the generated video, and the inferred action is compared with the commanded one. DriveGAN (Kim et al., 2021), GenAD (Yang et al., 2024a), and Vista (Gao et al., 2024) evaluate driving world models this way, and ACT-Bench turns the test into a benchmark (Arai et al., 2024). MineWorld and Matrix-Game apply the VPT inverse dynamics model to Minecraft keyboard and camera actions (Guo et al., 2025; Baker et al., 2022; Zhang et al., 2025), WorldCompass recovers camera motion (Wang et al., 2026), and AVID reports the action error relative to the recovery model's own error on real video (Rigter et al., 2025). World Action Verifier exploits the asymmetry between forward prediction and inverse verification (Liu et al., 2026), CD-LAM addresses action-irrelevant bias during training (Wei et al., 2026), and identifiability theory characterizes when latent states and action-conditioned dynamics can be recovered from observation–action data (Zhang et al., 2026). Action recovery judges a single video and requires the action to be identifiable from observations. We instead compare the futures that one state admits under different actions and learn this comparison from executed alternatives; an inverse dynamics model trained on the same data is our main learned baseline (Section 4.1).

Contrastive learning of dynamics. Contrastive objectives over transitions are widely used to learn representations for control. C-SWM contrasts structured latent transitions (Kipf et al., 2020); contrastive RL casts goal reaching as classification over futures (Eysenbach et al., 2021; 2022); TACO maximizes the mutual information between a representation of the state and action sequence and a representation of the future state, and Premier-TACO extends it to multitask pretraining (Zheng et al., 2023; 2024); and action-conditioned predictors are increasingly trained on frozen self-supervised features (Zhou et al., 2025; Assran et al., 2025). In these works, negatives serve representation learning and are chosen mainly for their difficulty. Conditional contrastive learning draws negatives that share a variable in order to remove its information (Tsai et al., 2022). In vision–language models, high retrieval accuracy coexists with insensitivity to word order, and negatives built by perturbing captions repair it (Yuksekgonul et al., 2023); our same-state negatives come from executed interventions rather than constructed perturbations. We apply this idea to evaluation: negatives from the same state make the learned score measure the effect of the action (Section 2.1).

Concurrent work. WorldEcho (Chen et al., 2026a) and WorldSimProbe (Co et al., 2026) also obtain ground truth by executing actions in a simulator and score a generated rollout against the simulator reference, WorldEcho through the alignment of extracted end-effector trajectories. We

use executed alternatives only for training; at evaluation, LAFS needs no simulator, matched rollout, or pose extractor.

B METHOD DETAILS

Branch groups. We collect groups of action branches $\mathcal{B}_i = \{(H_i, A_{ij}, Y_{ij})\}_{j=1}^{K}$ by executing alternative action sequences from the same initial environment state. All branches share H_i, and Y_{ij} is the observed outcome of A_{ij}; thus (H_i, A_{ij}, Y_{ij}) is a matched example, while (H_i, A_{ij}, Y_{ik}), $k \neq j$, changes the action–future correspondence without changing the starting scene. The branch actions are drawn from a branching distribution π, and the estimand of Proposition 1, hence the score, is defined relative to it: a different branching distribution defines a different score. In BranchPairs on LIBERO, a branch point lies eight control steps before the demonstration's first gripper sign change (pre-grasp), two steps after it (at-grasp), or at least 20 steps from any gripper event (free space). Each branch adds to the next 24 demonstration actions a smooth zero-mean perturbation with random Fourier phases and a shared amplitude spectrum, scaled after clipping so that its root-mean-square value in each action dimension is 15% of that dimension's largest absolute action (within 10%); eight branches are kept only if their perturbations are pairwise dissimilar (cosine below 0.5) and matched in smoothness. Training groups on LIBERO have sixteen branches at each of thirty branch points per demonstration, so a query has at most fifteen same-state negatives; BranchPairs conditions have eight. These comparisons concern consequences visible within the observation horizon. Cross-history negatives retain comparisons across scenes alongside the finer distinctions within a state. Comparisons with only same-state or only cross-history negatives, at equal candidate count, test the composition (Section 4.4).

Observational equivalence. Finally, the same-state estimand holds a prediction responsible only for consequences that can be observed. If two actions produce the same distribution of futures, $p(\cdot \mid H, A) = p(\cdot \mid H, A')$, then $s^*(H, A, \cdot)$ and $s^*(H, A', \cdot)$ differ by a constant and rank candidate futures identically. An inverse model that regresses a point estimate $\hat{A}(H, Y)$ and scores $-\|\hat{A}(H, Y) - A\|^2$ has no such guarantee: its ranking depends on whether the recovered action lies closer to A or to A', a distinction that no video can show.

Training schedule. We first train with within-trajectory negatives on recorded demonstrations, then fine-tune on the branch comparisons of Section 2.1. In the first stage, each query is a history and the actions that follow it at a time t of a demonstration, over windows at every time step; its positive is the recorded future window, and its negatives are future windows of the same demonstration starting at least four steps from t. These examples share a task and scene but differ in temporal context and execution. They provide training experience with action–future matching before the model must distinguish alternative continuations of one state. The second stage holds the initial history fixed and varies the executed actions. It therefore removes the temporal-context differences that can help resolve the first stage's comparisons. We retain cross-history negatives in this stage rather than relying exclusively on same-state alternatives. Both stages use the same future-selection loss and scoring network, with weights transferred between stages. The progression is set by how comparisons are constructed, rather than by mining examples by the model's current loss.

Bounded readout. The head consists of layer normalization, a linear layer, and a SiLU nonlinearity before the final projection. We omit the final bias and fix $\tau = 0.1$, giving $|s_\theta| \leq 10$. The same quantity is used as the logit in Eq. (1); no second temperature is applied. This caps norm growth in the final features and projection weights. Rankings are invariant to rescaling the score, whereas Eq. (1) keeps decreasing along αs_θ as $\alpha \to \infty$ once every positive is ranked strictly first, so an unbounded readout can lower the loss by growing its scale alone, to a different degree in each seed. With the bound, the logit gap is at most $2/\tau$, and the loss is bounded below by $\log\big(1 + (|\mathcal{C}| - 1)e^{-2/\tau}\big)$, which is below 10^{-7} for $\tau = 0.1$ and $|\mathcal{C}| = 16$. The bound therefore leaves a negligible loss floor; its purpose is optimization stability, keeping logits and their gradients bounded. In unbounded runs of an earlier configuration, the median logit differed by a factor of about 2,000 between two seeds. Table 3 lists the training settings; the learning rate warms up over the first 10% of each stage and then follows a cosine decay. Readout normalization and warmup are

evaluated through separate ablations. Every LAFS run, including the ablations, trains on a single GPU in float32, on AMD Instinct MI325X or NVIDIA RTX PRO 6000 Blackwell GPUs.

Table 3: **LAFS training settings on LIBERO.** Both stages use the same network and loss; the optimizer is reset between them. The RoboTwin model uses the same optimizer, schedule, batch, and candidate composition.

Setting	Value
Observations	agentview RGB, 128×128
History H	$L = 8$ frames spanning the last 24 control steps
Actions A	$T = 24$ control steps (1.2 s at 20 Hz)
Future Y	8 frames, every third of the 24 steps
Video encoders	history and future each: convolutional stem, spatial pooling, 14-layer temporal transformer (width 256)
Action encoder	4-layer transformer
Parameters	26,464,897
Readout	bounded, $\tau = 0.1$
Branch-stage candidates	1 positive, 8 same-state and 7 cross-history negatives per query
Batch	64 queries per step
Optimizer	AdamW, $\beta = (0.9,\ 0.95)$, weight decay 0.05
Learning rate	peak 3×10^{-4}; warm-up over 10% of each stage, then cosine decay
Gradient clipping	global norm 1.0
Training length	20 epochs per stage (102,280 steps in the branch stage)
Checkpoint	epoch with the lowest validation loss in each stage
Seeds	eight, 20260902–20260909
Hardware	one AMD Instinct MI325X GPU per run; float32

Deployment. The output is a compatibility score, not a calibrated probability of correct execution. Bounding its scale does not by itself establish metric quality, and retrieval accuracy on training-style candidates answers a different question from ranking world-model errors. We therefore evaluate the frozen score on controlled changes to action correctness and appearance, as well as on generated videos, separately from the retrieval objective.

Negatives on RoboTwin. Table 4 gives the controlled RoboTwin comparison of Section 4.4.

Table 4: **Negatives on RoboTwin.** Same-state eight-way retrieval (%; chance 12.5%) on 525 validation groups (4,200 queries, 35 tasks), mean of three paired seeds with 95% intervals from paired resampling. Both arms start from the same pretrained checkpoints and train on the same data for five epochs with fifteen negatives per query; only the source of the negatives differs.

Negatives per query (same-state / cross-history)	Retrieval	95% CI
After within-trajectory pretraining	16.1	[14.6, 17.8]
0 / 15	55.2	[51.3, 58.9]
8 / 7	97.3	[94.9, 98.7]
Paired difference	+42.1	[38.8, 45.3]

Number versus composition of negatives. Figure 8 varies the negatives per query for the current LIBERO model (eight seeds, same data and optimizer steps, pre-grasp same-state retrieval). Raising the same-state negatives from 8 to 15 changes retrieval by 0.4 points (5 of 8 seeds improve), whereas replacing 7 of the 15 with cross-history futures raises it by 3.5 points (7 of 8 seeds, the final setting of Table 2); adding 15 cross-history negatives to 15 same-state ones changes it by a further 0.2 points (6 of 8 seeds). The composition of the negatives, not their number, accounts for the gain.

Rendering. We render LIBERO with the offscreen renderer of robosuite (Zhu et al., 2020) under MuJoCo 2.3.7 (Todorov et al., 2012), in which the robot's collision geometry remains visible over its visual meshes. The frames of Sections 4.1, 4.3, and 4.4 are rendered this way, and every metric and training configuration receives the same frames. Figures 1, 2, and 4 show the same trajectories re-rendered with the collision geometry hidden.

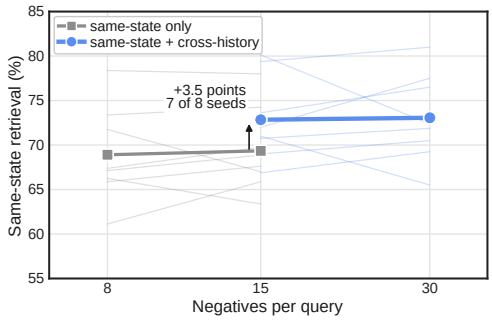


Figure 8: **Composition, not count, of the negatives.** Same-state eight-way retrieval on LIBERO (%; chance 12.5) against negatives per query: same-state only (8 or 15) and mixed (8 same-state and 7 cross-history, or 15 and 15). Thin lines: individual seeds; markers: means over eight seeds.

C LIMITATIONS

Our score must be trained on the domain it evaluates, from alternative actions executed from a common state, which requires a simulator or a resettable system; a single score across suites remains open. It judges only consequences visible within its horizon and does not measure task success or visual quality. BranchPairs constructs its errors from executed futures and does not cover every error of world models, such as objects that appear or vanish; the design comparisons are confined to LIBERO, and Action-MNIST uses a joint encoder. The training schedule was not tuned: in the earlier study, a cross-task stage before within-trajectory pretraining raised pre-grasp retrieval from 41.1% to 47.5% and reduced its spread across seeds (Table 2A), but the current model keeps the two-stage schedule, so its results may understate what a tuned schedule achieves.

D WORLD-MODEL TRAINING DETAILS

We fine-tune a Cosmos3-Edge world model (NVIDIA, 2026), the 4B variant of the Cosmos 3 family released after the report,[1] on the LIBERO-10 demonstrations, holding out 1% of the episodes for validation. The model receives the agentview and wrist camera images side by side at 256×256 per view and 20 frames per second, a text description of the task, and chunks of sixteen actions. Actions are ten-dimensional frame-to-frame changes of the end-effector pose in the robot's native frame (three for position, six for a continuous rotation representation, and one for the gripper), normalized by quantiles. All sixteen actions of each chunk are given as clean conditions in every training sample (forward dynamics); the layers that embed and decode actions are initialized from scratch. Table 5 lists the training settings. Figure 6 evaluates the checkpoints at 1,000, 5,000, and 9,000 iterations of this run; each panel uses the conditions in which every prediction is scored (40, or 39 for the VLM), and the VLM does not rate the branch mean (n/a). The world model is trained on the standard LIBERO renders, so its evaluation uses frames re-rendered from the stored simulator states with the collision geometry hidden, in the orientation of its training data. At evaluation, the model receives all sixteen actions and predicts only the video they produce. We predict 32 steps as two consecutive chunks, the second conditioned on the last predicted frame, and score the first 24 steps. Action correspondence compares each prediction with the eight executed futures of its condition and credits it when the nearest of them by PSNR is the future of its own action (ties share the credit). Used this way, as a choice among futures of one state rather than a distance to one future, PSNR no longer rewards hedging: a blurred prediction moves away from all eight futures alike, and the branch mean scores at chance (12.5%).

[1] `https://huggingface.co/nvidia/Cosmos3-Edge`

Table 5: **World-model training settings.** A single fine-tuning run; every checkpoint in Figure 6 is taken from it.

Setting	Value
Initialization	Cosmos3-Edge (4B); action layers from scratch
Conditioning	16 clean actions per chunk (forward dynamics)
Views and frame rate	agentview and wrist, 256×256 each; 20 fps
Caption dropout	0.1 (classifier-free guidance)
Optimizer	AdamW, $\beta = (0.9, 0.99)$, $\epsilon = 10^{-8}$, weight decay 0.05
Learning rate	5×10^{-5}; $5\times$ for the action layers
Schedule	linear warm-up over 500 iterations, then linear decay reaching zero at 16,000 iterations
Training length	9,000 iterations; checkpoint every 1,000
Batch	global batch 512; sequence packing, at most 64 clips and 45,056 tokens per GPU; 2 gradient-accumulation steps
Hardware	4 NVIDIA H100 GPUs, fully sharded data parallel
Precision	bfloat16, float32 master weights
Gradient clipping	global norm 1.0
Weight averaging (EMA)	not used
Random seed	42

E VLM SCORING PROTOCOL

Each prediction is rated independently by `gemini-3.1-pro-preview` with temperature zero. The input contains the original task instruction, a textual success criterion, and nine dual-view PNG images: the initial frame and future frames 4, 8, 12, 16, 20, 24, 28, and 32. Each image places agentview on the left and the wrist view on the right, each at 160×160 pixels. As in instruction-following and physical-commonsense judges (Li et al., 2025; Bansal et al., 2025), the critic rates each video on its own from the instruction and the frames, without a reference video, other candidates, or checkpoint identity.

Figure 6d uses the returned `physics_realism.score`, an integer from 1 to 5. For each checkpoint, we first average its eight query ratings within each condition and then average the 40 condition means equally. The 95% percentile intervals use 10,000 condition-bootstrap resamples (seed 20260926), with the same sampled condition indices across checkpoints and all eight queries retained together. Missing responses are not scored: one truncated reply removes its condition from the 1,000-iteration mean. The separate `task_success.success` field is not part of this plotted score and is not an environment-measured success rate.

F LAFS ON LIBERO-10

For each LIBERO-10 task, we split the demonstrations into 40 for training, of which 4 are kept for validation, and 10 for evaluation, with a fixed seed and before collecting any branches. From the training demonstrations we collect 1,500 groups of executed action branches for training and 167 for validation, with the same candidate construction and acceptance rules as for LIBERO-Spatial and LIBERO-Goal. We initialize from the eight seeds of the LIBERO-Spatial and LIBERO-Goal model and train on these branches with the same objective and hyperparameters, changing only the number of epochs, and select checkpoints on the validation demonstrations. With 10 epochs every seed selected one of its last two epochs, so we also trained for 30 epochs and, by a rule fixed before any evaluation, report the schedule with the lower mean validation loss (30 epochs: 1.32 versus 1.42).

On the 50 evaluation conditions, the adapted score retrieves the executed future of a queried action among the eight futures of its condition in 58.3% of queries (95% CI [52.9, 63.5]; chance 12.5%) and the action of an executed future among the eight actions in 50.2% ([45.5, 55.0]; means of eight seeds; condition bootstrap). The 10-epoch schedule gives 51.2% and 43.5%, and the exponential moving average of the 30-epoch weights 59.5% and 50.9%. Figure 6 excludes ten stove-scene conditions, whose world-model inputs were rendered without the burner state.

Table 6: **Cases in BranchPairs.** Each pair contrasts a future that follows the queried action with one that does not, both from the same condition. Every case in the main comparison stands for an error that video world models make. Severities are those used on LIBERO; futures have eight frames, and noise and brightness/contrast strengths are set to match the PSNR of the three blur levels.

Case	Construction	World-model error it represents	Severities
Follows the queried action			
Clean	executed future of the queried action	none (faithful prediction)	–
Blur	Gaussian blur	hedging between possible outcomes	$\sigma = 1, 2, 4$
Noise	additive Gaussian noise	high-frequency generation artifacts	3, PSNR-matched
Brightness/contrast	photometric shift	color and exposure drift	3, PSNR-matched
Deviates from the queried action			
Wrong branch	executed future of another action from the same state, clean or blurred	plausible rollout of the wrong action	blur $\sigma = 0, 1, 2, 4$
Drift	queried future for the first steps, then another branch	correct start, later divergence	first $k = 2, 4, 6$ frames correct
Temporal offset	queried future shifted in time	right content, wrong timing	delay of 1, 2, 4 frames

G THE ACTION-MNIST SCORER

Action-MNIST follows Moving MNIST (Srivastava et al., 2015) in animating MNIST digits, with the motion set by an action whose outcome is known. On Action-MNIST, a single encoder reads the history, the action, and a candidate future together and outputs $s_{jk} = s(H, A_j, Y_k)$ for the $K = 8$ branches of a group. Two digit branches can produce nearly the same motion, so the objective uses a set of positives and a set of valid comparisons for each branch j, based on the root-mean-square distance between digit trajectories (in pixels, with rotation angles weighted by four). The positive set $\mathcal{P}(j)$ contains j and every branch within distance one of it; no branch has more than two positives. The valid set $\mathcal{V}(j)$ adds every branch at distance at least three whose displacement from branch j reaches two pixels in at least four of the twelve frames. Pairs between these thresholds are excluded from the comparison. The loss averages the two directions of retrieval,

$$\mathcal{L}_{\text{MNIST}} = -\frac{1}{2K} \sum_{j=1}^{K} \left[\log \frac{\sum_{k \in \mathcal{P}(j)} e^{s_{jk}}}{\sum_{k \in \mathcal{V}(j)} e^{s_{jk}}} + \log \frac{\sum_{k \in \mathcal{P}(j)} e^{s_{kj}}}{\sum_{k \in \mathcal{V}(j)} e^{s_{kj}}} \right], \tag{7}$$

where the first term selects the future of an action and the second the action of a future. Treating branches with indistinguishable outcomes as positives follows the observational-equivalence argument of Section 2.1: a future is not penalized for failing to show a difference that the outcome does not contain. In the evaluation records, 3.25% of branches have two positives. The robotic scorer uses a single positive per query and the objective of Section 2.1.

H BRANCHPAIRS: ALL CASES

Table 7 reports every metric on every case of BranchPairs, including the cases outside the main comparison. Rows pair a case that follows the queried action with a case that deviates from it. The clean rows are the unperturbed judgment, which reference-based metrics pass trivially.

Examples. Figures 9–11 show examples of BranchPairs pairs on each benchmark, with the queried action and the alternate action executed from the same state.

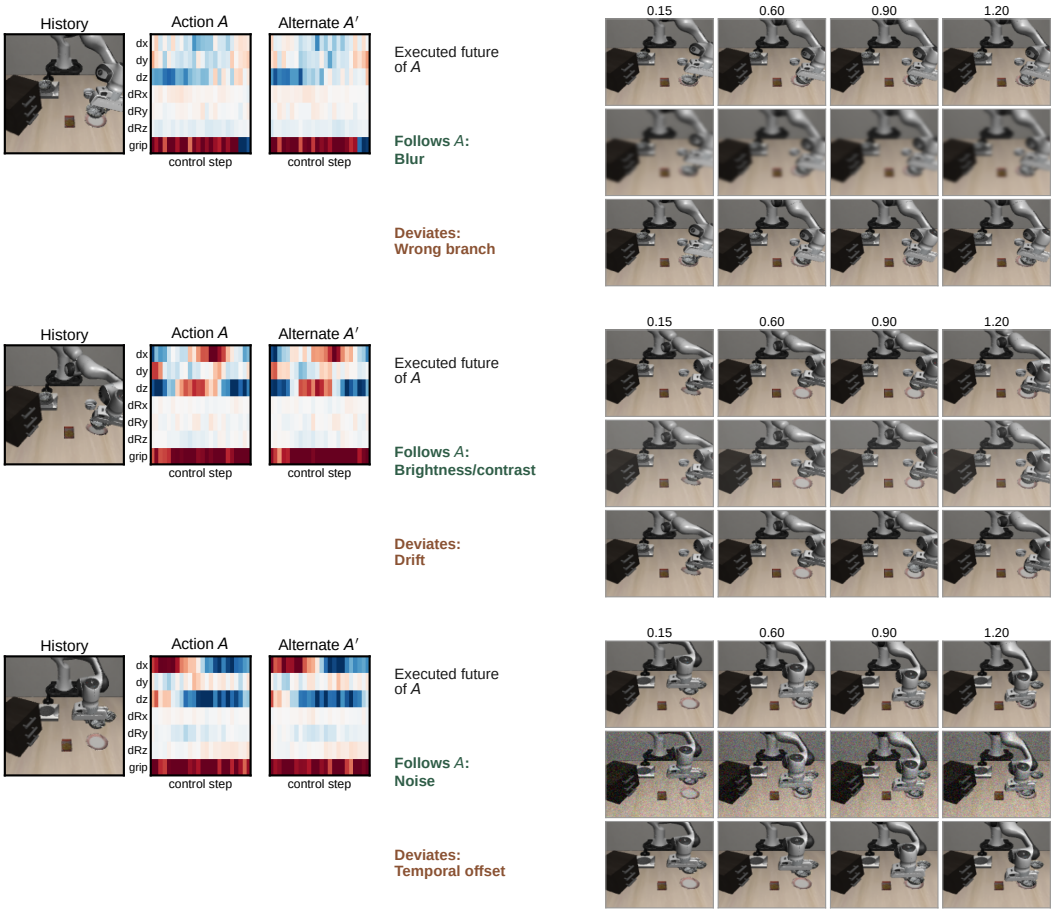


Figure 9: **BranchPairs examples on LIBERO.** Each block shows the last history frame, the queried action A and the alternate action A' executed from the same state (rows: end-effector translation, rotation, and gripper command; columns: the 24 control steps; one symmetric color scale per block), and four frames of three futures: the executed future of A, a future that follows A under an appearance change, and a future that deviates from it. From top: blur against the executed future of A' (wrong branch); brightness/contrast against a drift that switches to A''s future after four frames; noise against the executed future of A delayed by two frames. Constructions as in Figure 4; frames re-rendered with the collision geometry hidden. Scores are not shown.

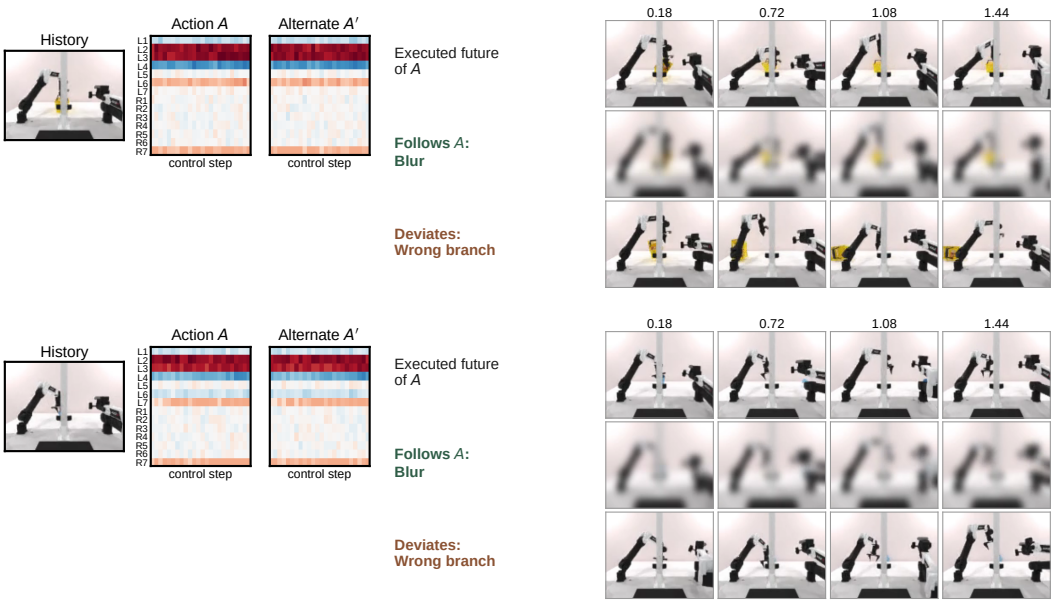


Figure 10: **BranchPairs examples on RoboTwin** (front camera). Layout as in Figure 9; actions have fourteen dimensions, seven per arm (L1–L7, R1–R7). Both blocks pair the blurred executed future of A with the executed future of A'. Scores are not shown.

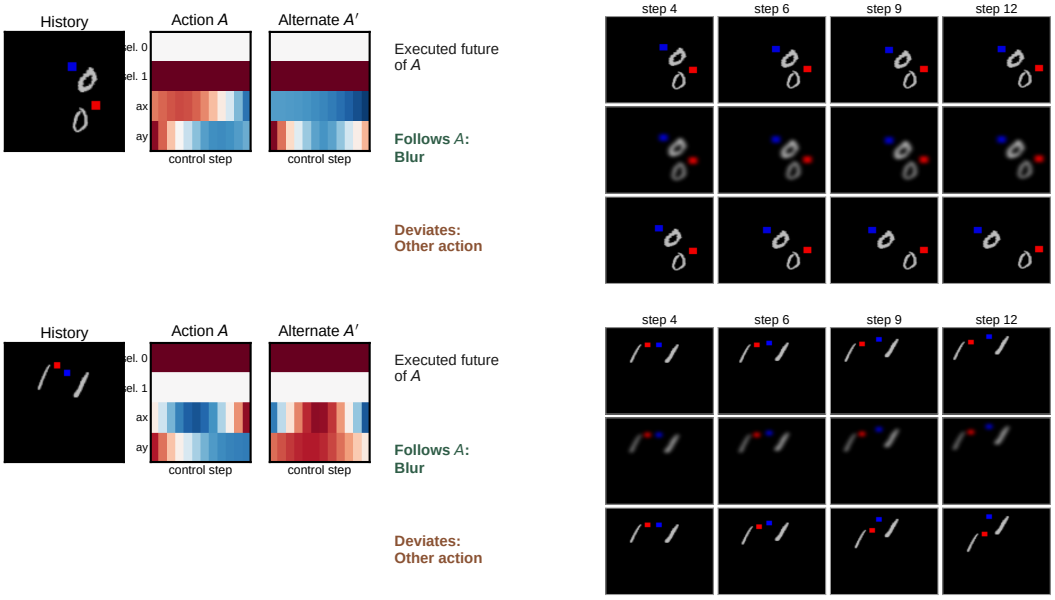


Figure 11: **BranchPairs examples on Action-MNIST.** Layout as in Figure 9; at each of the 12 steps the action selects which digit moves (sel. 0, sel. 1) and sets its acceleration (a_x, a_y); the red and blue squares are the environment's markers of the two digits. Both blocks pair the blurred executed future of A with the executed future of A'. Scores are not shown.

Table 7: **Per-case results on LIBERO.** Pairwise accuracy (%) on the main cases of BranchPairs (MAIN-Full, 100 conditions): each row pairs a case that follows the queried action with a case that deviates from it. Wrong branches include their blurred versions. LAFS is averaged over eight seeds and inverse dynamics (IDM) over three. The last row, JPEG compression, is not among the main cases; reference-based metrics were not computed for it.

Case pair	Pairs	LAFS	IDM	PSNR	SSIM	LPIPS	DINOv2
Clean vs. wrong branch	400	99.4	94.6	100.0	100.0	100.0	100.0
Clean vs. drift	300	86.8	98.1	100.0	100.0	100.0	100.0
Clean vs. temporal offset	300	99.9	98.2	100.0	100.0	100.0	100.0
Blur vs. wrong branch	1,200	98.4	79.6	60.9	51.7	48.8	47.3
Blur vs. drift	900	77.1	64.4	15.6	0.8	0.0	0.0
Blur vs. temporal offset	900	99.9	97.6	83.7	28.7	0.0	14.1
Noise vs. wrong branch	1,200	84.7	46.6	55.9	3.4	57.7	31.2
Noise vs. drift	900	47.9	26.0	15.7	0.0	0.0	0.0
Noise vs. temporal offset	900	99.5	95.4	81.2	0.0	0.1	0.0
Bright./contr. vs. wrong branch	1,200	98.3	81.0	55.9	93.8	91.2	100.0
Bright./contr. vs. drift	900	78.8	69.6	15.7	25.7	34.4	100.0
Bright./contr. vs. temporal offset	900	99.9	97.5	81.3	98.4	92.9	100.0
JPEG vs. all action errors	3,000	52.7	31.2	–	–	–	–

Table 8: **Composition and reweighting of the matched comparison on LIBERO.** Left: number of PSNR-matched pairs for each combination of an appearance case (rows) and an action case (columns), 499 in total over 100 conditions. Right: pairwise accuracy (%) pooled over pairs and averaged over the nine combinations (case-balanced). LAFS averages eight seeds; IDM, three.

	Drift	Offset	Wrong br.
Blur	43	33	116
Noise	31	32	86
Bright./contr.	31	38	89

Metric	Pooled	Case-balanced
PSNR	59.9	53.8
SSIM	47.9	37.9
LPIPS	59.5	48.6
DINOv2	52.5	44.4
IDM	80.1	81.8
LAFS	93.5	92.8

I SEEDS IN FIGURE 1

The bars of Figure 1 average each learned metric over the three seeds with which both LAFS and IDM were trained (20260902–04); the example in its top row is scored with seed 20260904 for LAFS and seed 20260902 for IDM. Every other LIBERO number in the paper averages over all trained seeds (Table 1). PSNR, SSIM, LPIPS, and DINOv2 are not trained and have a single value. Table 9 lists every seed. LAFS varies more across seeds than IDM (standard deviation 5.9 versus 0.6 points), but its worst seed (81.2%) still exceeds the best IDM seed (75.7%).

J PROOFS

Setting. Throughout, H denotes the history, A the action sequence, and Y the future window. For a query $x = (H, A)$ with positive Y^+ and negatives $Y_1^-, \dots, Y_{M-1}^-$, the per-query term of Eq. (1) is

$$\ell(s; x, \mathcal{C}) = -\log \frac{\exp s(x, Y^+)}{\sum_{Y \in \mathcal{C}} \exp s(x, Y)} = \log\Big(1 + \sum_{i=1}^{M-1} \exp\big(s(x, Y_i^-) - s(x, Y^+)\big)\Big), \qquad (8)$$

with $\mathcal{C} = \{Y^+, Y_1^-, \dots, Y_{M-1}^-\}$ and $M \geq 2$. Lemma 1 and Proposition 2 below concern fixed candidate lists and hold for any collected data. Proposition 1 and Corollary 1 are population statements about the following idealized collection model. From an environment state S, branch actions $A_1, \dots, A_K$ are drawn independently from a branching distribution $\pi(\cdot \mid S)$, each is executed, and the futures $Y_j \sim p(\cdot \mid S, A_j)$ are independent given S. The history determines the state, so that

Table 9: **Every training seed on the 10,000 LIBERO MAIN pairs.** Preference for the action-consistent future (%) with 95% bootstrap intervals over the 100 conditions. Bold: the three-seed means shown in Figure 1; the three seeds are those with which both learned metrics were trained. Means over all seeds are used everywhere else in the paper.

Seed	LAFS	IDM
20260902	93.4 [92.1, 94.7]	75.7 [73.8, 77.6]
20260903	81.2 [80.1, 82.3]	74.4 [72.4, 76.5]
20260904	95.5 [94.2, 96.7]	75.1 [72.9, 77.3]
20260905	94.5 [93.2, 95.7]	–
20260906	82.0 [80.9, 83.2]	–
20260907	84.0 [82.7, 85.1]	–
20260908	86.5 [85.3, 87.7]	–
20260909	91.9 [90.3, 93.4]	–
Mean over seeds	88.6 [87.9, 89.4]	75.1 [73.3, 76.9]
Mean over 20260902–04	**90.0** [89.1, 91.0]	**75.1** [73.3, 76.9]
Standard deviation across seeds	5.9	0.6
Range across seeds	81.2–95.5	74.4–75.7

$p(\cdot \mid H, A) = p(\cdot \mid S, A)$, and $p(Y \mid H) = \int p(Y \mid H, a)\, \pi(da \mid H)$ is the marginal distribution of a branch future. For query (H, A_j), the pairs (A_l, Y_l) with $l \neq j$ are then independent of (A_j, Y_j) given H and identically distributed, so the same-state negatives are i.i.d. draws from $p(\cdot \mid H)$ that do not depend on the queried action. Cross-history negatives are futures of other training examples, drawn independently of the query, and are i.i.d. draws from the marginal $p(Y)$. The training configurations of Section 4.4 use fixed numbers of each kind of negative. The theory describes the two pure cases; a fixed composition such as eight same-state and seven cross-history negatives is not an i.i.d. sample from a mixture, its population optimum is not a simple density ratio (see the example below), and it is an empirical choice (Table 2B). In a deterministic simulator $p(\cdot \mid H, A)$ is a point mass and the density ratio degenerates; Proposition 1 then describes the limit of an environment or observation process with noise, of which the trained score is a smooth surrogate. Lemma 1 and Proposition 2 concern fixed candidate lists and need no densities.

Proof of Proposition 1. Let $p(\cdot \mid x)$ and $q(\cdot \mid H)$ be mutually absolutely continuous, let $r = dp(\cdot \mid x)/dq(\cdot \mid H)$ be finite and positive almost everywhere, and minimize the population loss over all measurable scores.

Step 1: the loss is a log-loss for the position of the positive. Present the M candidates in a uniformly random order and let $k^\star$ denote the position of the positive. Eq. (8) depends only on which candidate is the positive, not on the order of the negatives, so

$$\mathbb{E}\big[\ell(s; x, \mathcal{C})\big] = \mathbb{E}\big[-\log Q_s(k^\star \mid x, Y_{1:M})\big], \qquad Q_s(k \mid x, Y_{1:M}) = \frac{\exp s(x, Y_k)}{\sum_{m=1}^{M} \exp s(x, Y_m)}.$$

Step 2: the posterior of the position. Given x, the joint law of $(k^\star, Y_{1:M})$ has density $\frac{1}{M}\, r(Y_{k^\star})$ with respect to $\mathrm{Unif}\{1, \dots, M\} \otimes q^{\otimes M}$, so

$$P(k \mid x, Y_{1:M}) = \frac{r(Y_k)}{\sum_{m=1}^{M} r(Y_m)}.$$

Step 3: the minimizers. For every s,

$$\mathbb{E}\big[-\log Q_s(k^\star \mid x, Y_{1:M})\big] = \mathbb{E}\big[\mathcal{H}\big(P(\cdot \mid x, Y_{1:M})\big)\big] + \mathbb{E}\big[\mathrm{KL}\big(P(\cdot \mid x, Y_{1:M}) \,\|\, Q_s(\cdot \mid x, Y_{1:M})\big)\big],$$

where $\mathcal{H}$ is the entropy of the posterior. The first term does not depend on s, and the second is nonnegative and vanishes exactly when $Q_s = P$ almost everywhere. The score $s = \log r$ gives $Q_s = P$, so the minimizers are exactly the scores with $Q_s = P$ almost everywhere.

Step 4: the minimizers are $\log r$ *up to an offset.* If $Q_s = P$ almost everywhere, then for almost every candidate set and every pair of positions, $s(x, Y_k) - s(x, Y_l) = \log r(Y_k) - \log r(Y_l)$. Let

$d(Y) = s(x, Y) - \log r(Y)$. Then $d(Y) = d(Y')$ for almost every pair (Y, Y') under the joint law of two candidates, which is $p \otimes q$ for the positive and a negative and $q \otimes q$ for two negatives. By mutual absolute continuity both laws have the same null sets as $q \otimes q$, so $d(Y) = d(Y')$ for $q \otimes q$-almost every pair. By Fubini's theorem there is a Y' with $d(Y) = d(Y')$ for q-almost every Y; call this value $c(x)$. Hence $s(x, Y) = \log r(Y) + c(x)$ almost everywhere. Conversely, the offset cancels in Q_s, so every such score is a minimizer.

Step 5: the two negative distributions. With cross-history negatives, $q = p(Y)$ and $\log r = \log \frac{p(Y|H)}{p(Y)} + \log \frac{p(Y|H,A)}{p(Y|H)}$, which is Eq. (2). With same-state negatives, $q = p(Y \mid H)$, which is Eq. (3). □

Reading the action term. Under the collection model, Bayes' rule gives $p(Y \mid H, A)/p(Y \mid H) = \pi(A \mid H, Y)/\pi(A \mid H)$, the factor by which observing Y changes the probability of the queried action among the branches. Its expectation over $A \sim \pi(\cdot \mid H)$ and $Y \sim p(\cdot \mid H, A)$ is the conditional mutual information $I_\pi(A; Y \mid H)$, and the InfoNCE bound (van den Oord et al., 2018; Poole et al., 2019), applied conditionally on H, gives $\log M - \mathbb{E}[\ell] \le I_\pi(A; Y \mid H)$ for every score trained with $M - 1$ same-state negatives; like any sample-based lower bound on mutual information, it cannot exceed $\log M$ (McAllester & Stratos, 2020). The fixed-composition training of Section 4.4 has a different population optimum. For example, with two outcomes, a positive distribution $(0.8, 0.2)$, and one negative from each of $(0.9, 0.1)$ and $(0.1, 0.9)$, the optimal score gap is 1.162, whereas i.i.d. negatives from their average $(0.5, 0.5)$ give $\log 4 = 1.386$.

Proof of Lemma 1. Let v_k be the score of Y_k, which is the same for every query because the score does not depend on the action. (a) All queries share the candidates $Y_1, \dots, Y_K$ and assign them the same scores. Under a tie-breaking rule that does not depend on the query, all K queries select the same candidate $Y_{j'}$, which is the positive of query j' only, so exactly one of the K queries is answered correctly. Under independent uniform tie-breaking among the top-scoring set $\mathcal{M}$, query j is correct with probability $\mathbf{1}\{j \in \mathcal{M}\}/|\mathcal{M}|$, so the expected accuracy is $\frac{1}{K} \sum_{j \in \mathcal{M}} 1/|\mathcal{M}| = 1/K$. (b) Additional candidates only increase each term of Eq. (8), so the average loss is at least

$$\frac{1}{K} \sum_{j=1}^{K} \Big(\log \sum_{k=1}^{K} e^{v_k} - v_j \Big) = \log \sum_{k=1}^{K} e^{v_k} - \frac{1}{K} \sum_{k=1}^{K} v_k = \log K + \log \Big(\frac{1}{K} \sum_{k=1}^{K} e^{v_k} \Big) - \frac{1}{K} \sum_{k=1}^{K} v_k.$$

By Jensen's inequality, $\frac{1}{K} \sum_k e^{v_k} \ge \exp\big(\frac{1}{K} \sum_k v_k\big)$, so the right-hand side is at least $\log K$, with equality only when all v_k are equal and no other candidates are present. □

Loss certificates and the cross-history shortcut. The following statements concern fixed candidate lists and need no densities.

Proposition 2. *(a) For any score s and candidate set $\mathcal{C}$, if $\ell(s; x, \mathcal{C}) < \log 2$, then $s(x, Y^+) > s(x, Y)$ for every $Y \in \mathcal{C} \setminus \{Y^+\}$. Consequently, over any distribution of queries and candidate sets, the probability that the positive fails to outrank every other candidate is at most $\mathbb{E}[\ell]/\log 2$.*

(b) Suppose that the history can be read from its future: there is a measurable map $\hat{H}$ with $\hat{H}(Y) = H$ almost surely under $p(\cdot \mid H)$ for every H, and that all negatives come from other histories. Then for every $\varepsilon > 0$ there is a score $s(H, Y)$ that does not depend on the action and has expected loss below ε.

Proof. (a) If $\ell < \log 2$, Eq. (8) gives $\sum_i \exp\big(s(x, Y_i^-) - s(x, Y^+)\big) < 1$, so every summand is below one and $s(x, Y_i^-) < s(x, Y^+)$ for all i. Conversely, if some candidate scores at least as high as the positive, the corresponding summand is at least one and $\ell \ge \log 2$; Markov's inequality gives the bound on the probability. (b) Let $s_\beta(H, Y) = \beta\, \mathbf{1}\{\hat{H}(Y) = H\}$. Almost surely the positive satisfies $\hat{H}(Y^+) = H$ and every negative, a future of another history H', satisfies $\hat{H}(Y^-) = H' \ne H$, so $\ell(s_\beta; x, \mathcal{C}) = \log\big(1 + (M - 1)e^{-\beta}\big)$, which falls below ε for large β. □

Part (a) applies to mixed candidate sets: a query whose loss is below $\log 2$ outranks every candidate it was trained against, including its same-state alternatives. By part (b) and Lemma 1, near-zero loss against cross-history negatives is compatible with exactly chance-level same-state retrieval, whereas an average same-state loss below $\log K$ is not attainable without using the action.

Propositions 1 and 2(b) describe two ends of the cross-history case. When futures of different histories overlap, the estimand is finite and contains the history term of Eq. (2); when a future reveals its history, that term diverges and the history alone drives the loss to zero. Real data lie between; the same-state case has no such tension because H is fixed. Conversely to Lemma 1(b), in the setting of Proposition 1 with same-state negatives the optimal loss is below $\log K$ exactly when $I_\pi(A; Y \mid H) > 0$, that is, when the branch actions have distinguishable consequences: by Step 3 of its proof, the optimal loss is the expected entropy of the position posterior, which equals $\log K$ only if the density ratio is constant. A same-state loss below $\log K$ is therefore attainable, and by Lemma 1 only by using the action.

Appearance nuisance. The two estimands of Proposition 1 treat appearance factors differently.

Corollary 1. *Under the assumptions of Proposition 1, suppose $Y = g(Z_a, Z_n)$ for a diffeomorphism g, where Z_a carries the consequences of the action and the nuisance Z_n is independent of (A, Z_a) given H. Then Eq. (3) depends on Y only through Z_a, whereas the history term of Eq. (2) in general depends on Z_n when the distribution of Z_n differs across histories.*

Proof. Write $(z_a, z_n) = g^{-1}(y)$. By the change of variables formula, every conditional density of Y has the form $p_Y(y \mid \cdot) = p_Z(z_a, z_n \mid \cdot)\, J(y)$ with $J(y) = |\det Dg^{-1}(y)|$, a factor that depends on neither H nor A. Conditional independence of Z_n and (A, Z_a) given H gives

$$p(z_a, z_n \mid H, A) = p(z_a \mid H, A)\, p(z_n \mid H),$$
$$p(z_a, z_n \mid H) = \int p(z_a \mid H, a)\, p(z_n \mid H)\, \pi(da \mid H) = p(z_a \mid H)\, p(z_n \mid H).$$

Hence

$$\log \frac{p(y \mid H, A)}{p(y \mid H)} = \log \frac{p(z_a \mid H, A)\, p(z_n \mid H)\, J(y)}{p(z_a \mid H)\, p(z_n \mid H)\, J(y)} = \log \frac{p(z_a \mid H, A)}{p(z_a \mid H)},$$

which depends on y only through z_a. The history term is

$$\log \frac{p(y \mid H)}{p(y)} = \log \frac{p(z_a \mid H)\, p(z_n \mid H)}{\mathbb{E}_{H'}\big[p(z_a \mid H')\, p(z_n \mid H')\big]}.$$

For example, let H take two values h_1, h_2 with probability $\frac{1}{2}$ each, let $p(z_a \mid h_1) = p(z_a \mid h_2)$, and let $p(z_n \mid h_i) = f_i(z_n)$ with f_1, f_2 positive on a common support. The history term for h_1 is then $\log\big(2f_1(z_n)/(f_1(z_n) + f_2(z_n))\big)$, which varies with z_n unless $f_1 = f_2$. Among futures of h_1, it prefers those whose nuisance is typical of h_1, whatever the action. □

The corollary concerns variation within the data distribution that preserves Z_a. Degradations such as blur can also remove action information and are evaluated empirically in Section 4.

Observationally equivalent actions. Let $p(\cdot \mid H, A) = p(\cdot \mid H, A')$. By Proposition 1, for almost every Y,

$$s^*(H, A, Y) - s^*(H, A', Y) = \log \frac{p(Y \mid H, A)}{q(Y \mid H)} - \log \frac{p(Y \mid H, A')}{q(Y \mid H)} + c(H, A) - c(H, A')$$
$$= c(H, A) - c(H, A'),$$

so the two queries rank candidate sets identically. Consider a point-estimate recovery score $m(H, A, Y) = -\|n(\hat{A}(H, Y)) - n(A)\|^2$, where n is the normalization used by the inverse-dynamics baseline (n is the identity for the unnormalized form). Then

$$m(H, A, Y) - m(H, A', Y) = 2\big\langle n(\hat{A}(H, Y)),\, n(A) - n(A')\big\rangle + \|n(A')\|^2 - \|n(A)\|^2,$$

which varies with Y whenever the recovered action varies along $n(A) - n(A')$. For example, if $\hat{A}(H, Y_1) = B$ and $\hat{A}(H, Y_2) = C$ with $\|n(B) - n(A)\| < \|n(C) - n(A)\|$ and $\|n(C) - n(A')\| < \|n(B) - n(A')\|$, as when B lies near A and C near A', then m ranks Y_1 above Y_2 under A and Y_2 above Y_1 under A'. By the Bayes identity above, Eq. (3) equals $\log \pi(A \mid H, Y)/\pi(A \mid H)$, the log-ratio a probabilistic inverse model would need a density over whole action sequences to evaluate; the same-state comparison obtains it from futures alone, without modeling that density. □

Lower bound with a bounded readout. The readout is an inner product of unit vectors divided by τ, so $|s| \leq 1/\tau$ and every difference $s(x, Y_i^-) - s(x, Y^+)$ is at least $-2/\tau$. By Eq. (8), each term of Eq. (1) is at least $\log\big(1 + (|\mathcal{C}| - 1)e^{-2/\tau}\big)$. For $\tau = 0.1$ and $|\mathcal{C}| = 16$, this is $\log(1 + 15e^{-20}) \leq 15e^{-20} < 10^{-7}$. □