OFF THE RECORD: WHAT A WORLD MODEL GUARANTEES AT UNRECORDED ACTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

World models let an agent plan by asking what an action would do instead of taking it. They are trained and validated on transitions recorded under a behavior policy, yet planning asks about actions that policy never took. We ask what a world model can guarantee at such actions. It is elementary that the data alone guarantees nothing there, since a model can fit it perfectly and still be arbitrarily wrong. Our two results say what a guarantee rests on. First, a Lipschitz constant declared for the model's predictor turns distance from the data into a bound. The bound takes two assumptions: the true dynamics respect that constant, and the model's encoder retains the relevant state. Under them, the next-state error of every model that fits the data and respects the constant is bounded from above by a band that widens with that distance. The band is attained in the worst case, and a constant declared too low can make it silently false. Half the disagreement between two models bounds the larger of their errors from below, with no constant. Second, the encoder sets that distance, and discarding nuisance shrinks it. On MuJoCo datasets, models that score alike on held-out data diverge at unrecorded actions, and their error and disagreement grow with that distance. An adversarial search finds an error within a small factor of the band. Replacing part of a dataset with widely varied actions lowers error far from the data. We turn these results into guiding principles. Distance from the data, not a held-out score, should govern how far a world model's prediction is trusted.

1 INTRODUCTION

A world model is meant to capture its environment well enough that an agent can plan by asking it what-if questions instead of acting in the world. World models are among the field's most active directions, yet there is no shared definition of what one is. Each paper means something slightly different by the term (Oh, 2026), so claims are hard to compare, and what a world model guarantees about a question its data does not answer is not settled.

Dreamer 4 (Hafner et al., 2025b), for example, optimizes a policy purely inside a world model learned from a fixed dataset of human Minecraft play, and the policy collects diamonds in the real game. Competent play leaves out the unusual actions such a policy seeks out, and Dreamer 4 penalizes the policy for straying from the recorded behavior. Its test of generalization scores predictions under the players' own actions, so that test never asks about an action no player took. Across the field, world models are evaluated in ways that do not say how large a prediction's error is at an action the data lacks. More data from the same behavior policy adds only the actions it takes, and error on recorded transitions says nothing on its own about an action not taken.

A guarantee at such an action rests on two assumptions: that the dynamics change no faster than a limit the modeler declares, and that the model's encoder keeps what matters. For simplicity, we take that limit to be a Lipschitz constant, one of several possible forms. Under both, every model that fits the data and respects that constant can drift from the world only gradually away from the data, and so predicts the next state to within a band that widens with the distance from the question, or *query*, to the data. The farther a query lies from the data, the less its answer can be trusted.

The inequalities behind these results are elementary. For states measured directly, the band is a form of the bound in set-membership identification (Milanese & Novara, 2004). What we add is an account of what the modeler must supply when the state is learned by a world model: the declared

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.

constant, the encoder and how distance is measured. We make three contributions, and §5 gives empirical evidence on MuJoCo (Todorov et al., 2012) datasets.

- **What a bound from distance rests on** (§3.2). Under the two assumptions above, a Lipschitz constant declared for the predictor turns distance from the data into the band, an upper bound on the query error of every model that fits the data and respects the constant. The band is tight in the worst case, and a constant declared too low can silently make it false.
- **The encoder sets the distance** (§3.3). Discarding nuisance brings a query closer to the data. Discarding relevant state also shrinks the distance, but the encoder error it adds keeps the band wide.
- **Guiding principles** (§4) derived from these results: judge a prediction by its distance from the data, declare the constant as an assumption behind the band's upper bound on the error and report two models' disagreement as a lower bound that needs no constant, buy coverage with varied actions rather than with more data from the same policy, discard only nuisance, and plan with distance as a penalty.

2 Related Work

Evaluation of learned world models. Learned world models differ in construction but are evaluated in the same few ways: a held-out score, task success, video quality, and a probe reading a physical quantity off the representation. This paper asks what such evaluation establishes about questions the data does not answer. Among these models, World Models (Ha & Schmidhuber, 2018) is trained once on a random policy's rollouts and DreamerV3 (Hafner et al., 2025a) is updated as its policy collects more data. Genie (Bruce et al., 2024) learns from video without recorded actions, DINO-WM (Zhou et al., 2024) learns only the dynamics on a frozen pretrained representation, and LeWorldModel (Maes et al., 2026), a joint-embedding predictive architecture (JEPA), learns representation and dynamics together from pixels. Ha & Schmidhuber (2018) report what task-success evaluation can miss: a policy optimized inside their model scores well there and fails in the real environment. They respond by raising the sampling temperature, which makes the model noisier, and Dreamer 4 (Hafner et al., 2025b) penalizes its policy's KL divergence from a behavior-cloned one. Both are safeguards, but neither states how far from the data a prediction can be trusted.

Identifiability of learned state representations. This line asks when the latent state generating the observations can be identified from data. Its results concern the encoder rather than the action-conditioned transition, and a guarantee at an unrecorded action needs both. Nonlinear ICA established that the latent is not identifiable without additional structure (Hyvärinen & Pajunen, 1999), and that non-stationarity, temporal dependence, or an auxiliary variable can supply it (Hyvarinen et al., 2019; Khemakhem et al., 2020). More recently, LeJEPA (Balestriero & LeCun, 2025) regularizes the embedding distribution toward an isotropic Gaussian, and Klindt et al. (2026) prove that for a stationary process with independent additive noise and Gaussian latents, LeJEPA recovers the true latents up to a linear orthogonal map, a result they describe as addressing the encoder side.

Bounds on compounding model error. A parallel line bounds how a per-step model error accumulates along a rollout, while this paper bounds the per-step error itself. Some works take that error as a uniform bound holding everywhere. Asadi et al. (2018) combine this uniform bound with the model's Lipschitz constant to bound the multi-step prediction error, and Timor et al. (2026) extend that analysis to learned reward models. Janner et al. (2019) replace the per-step term with an empirical estimate of model generalization, averaged over the states a policy visits, and branch short rollouts from real data. Yet the per-step error at a single query can grow with the query's distance to the recorded data, and a held-out score does not show it.

3 Theoretical Framework

A world model is asked what-if questions (Levine et al., 2020): in this situation, what would happen under an action other than the one taken? The dataset alone does not answer that, because it shows the world only where it was recorded. We open with the setup and the elementary facts it implies, and then give the two results the rest of the paper builds on: a Lipschitz constant declared for the predictor turns distance from the data into a bound (§3.2), and the encoder sets that distance (§3.3).

3.1 SETUP

3.1.1 THE WORLD

Observations lie in a metric space $(\mathcal{X}, d_{\mathcal{X}})$ and actions in a metric space $(\mathcal{A}, d_{\mathcal{A}})$. An observation often mixes the state that matters with detail that does not affect it.

Definition 1 (a split)**.** A *split* of the observation is a metric space $(\mathcal{Z}, d_{\mathcal{Z}})$ of *relevant states*, a set $\mathcal{U}$ of *nuisances* carrying no metric, and a *mixing function* $\rho : \mathcal{Z} \times \mathcal{U} \to \mathcal{X}$. Every observation has the form $x = \rho(z, u)$.

For example, in a short camera clip of a robot, the joint angles and velocities are the relevant state, and lighting and background are nuisance.

Definition 2 (a self-contained split)**.** A split is *self-contained* (Ni et al., 2024) if the relevant state evolves on its own: there is a *transition function* $T : \mathcal{Z} \times \mathcal{A} \to \mathcal{Z}$ with $z' = T(z, a)$ whenever an observation $\rho(z, u)$ is followed by $\rho(z', u')$ under an action a. T is a property of the world, unknown to the modeler.

The robot's split is self-contained, since lighting does not move the robot. Treating the velocities as nuisance would break this, because the angles alone do not determine the next angles.

3.1.2 THE MODEL

The modeler sees only observations and actions, not the relevant state.

Definition 3 (a world model)**.** A *world model* is a metric space $(\mathcal{S}, d_{\mathcal{S}})$ of *model states*, with an *encoder* $g : \mathcal{X} \to \mathcal{S}$ and a *predictor* $f : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$: surrogates for the relevant state and for T.

Faithfulness measures how well the encoder keeps the relevant state and drops the nuisance.

Definition 4 (a faithful world model)**.** A world model is (α, ε)*-faithful to the split*, with distortion $0 \le \alpha < 1$ and encoder error $\varepsilon \ge 0$, if there exists an *ideal map* $h : \mathcal{Z} \to \mathcal{S}$ such that

$$(1-\alpha)\, d_{\mathcal{Z}}(z, \tilde{z}) \;\le\; d_{\mathcal{S}}\big(h(z), h(\tilde{z})\big) \;\le\; (1+\alpha)\, d_{\mathcal{Z}}(z, \tilde{z}) \qquad \text{for all } z, \tilde{z} \in \mathcal{Z},$$

and the encoder lands within ε of it: $d_{\mathcal{S}}\big(g(\rho(z, u)), h(z)\big) \le \varepsilon$ for all $z \in \mathcal{Z}$, $u \in \mathcal{U}$.

In a faithful world model, a prediction close to the encoding $g(x')$ of an observation $x' = \rho(z', u')$ is also close to $h(z')$, the model's copy of the relevant state, whatever the nuisance.

3.1.3 THE DATA

The *dataset* D is a fixed finite set of recorded transitions (x, a, x'), whose actions were chosen by a *behavior policy*. Write $\mathcal{R}_{\mathcal{X}} := \{(x, a) : (x, a, x') \in D\}$ and $\mathcal{R}_{\mathcal{S}} := \{(g(x), a) : (x, a) \in \mathcal{R}_{\mathcal{X}}\}$ for the recorded observation–action pairs and their encoded version, the *support*. Each recorded transition gives the predictor a *recorded target* $g(x')$.

At each state the behavior policy visits, the dataset holds only actions it took there. Recording longer adds more of that policy's actions, never the ones it avoids.

The *loss* averages the squared distance from prediction to target:

$$L(f) \;:=\; \frac{1}{|D|} \sum_{(x,a,x') \in D} d_{\mathcal{S}}\big(f(g(x), a),\, g(x')\big)^2.$$

Because the loss is computed on encodings, a constant encoder makes every target the same, and a constant predictor then scores $L = 0$ having learned nothing. This is a *collapse* (Maes et al., 2026). Faithfulness makes it costly: a constant encoder confines $h(\mathcal{Z})$ to a ball of radius ε, so the lower bound in Definition 4 forces $\varepsilon \ge (1-\alpha)\,\mathrm{diam}(\mathcal{Z})/2$.

3.1.4 THE QUERY

A what-if question names a situation and an action together, so a *query* is a pair $q \in \mathcal{S} \times \mathcal{A}$ that the model is asked about.

Definition 5 (the distance to the data)**.** Let d be a metric on pairs, the *pair distance*. For a pair q and a set $\mathcal{R}$ of pairs, $\mathrm{dist}(q, \mathcal{R}) := \inf_{r \in \mathcal{R}} d(q, r)$.

For simplicity, we take the pair distance to be the sum $d\big((s,a),(\tilde{s},\tilde{a})\big) := d_{\mathcal{S}}(s,\tilde{s}) + d_{\mathcal{A}}(a,\tilde{a})$, and likewise on $\mathcal{Z} \times \mathcal{A}$. Other pair distances can also work, with different constants.

A query lies *off support* when $\mathrm{dist}(q, \mathcal{R}_{\mathcal{S}}) > 0$. It can ask, for example, what follows a visited state under an action the policy never took there.

Definition 6 (the true answer)**.** Let the split be self-contained and let h be an ideal map (Definition 4). For a query $q = (h(z), a)$, the *true answer* is $F(q) := h\big(T(z,a)\big)$, the model's copy of the next relevant state.

The lower bound in Definition 4 makes h injective, so F is well defined. Every result below holds for any choice of h.

3.1.5 The guarantee

A guarantee must hold for every predictor the data and the modeler allow, not only the one that training produced. The architecture (Gunasekar et al., 2018), the optimization (Neyshabur et al., 2014) and the initialization bias the trained predictor, but that bias is implicit and states no limit on how fast the prediction can change. A guarantee therefore rests on a limit the modeler declares.

Definition 7 (the declared inductive bias)**.** The modeler *declares* a constant $\lambda_{\mathcal{S}} > 0$: the predictor is to be $\lambda_{\mathcal{S}}$-Lipschitz from $\mathcal{S} \times \mathcal{A}$ to $\mathcal{S}$, measured in the pair distance. That class of predictors is the *declared inductive bias*.

Definition 8 (an admissible predictor)**.** Fix a *fit tolerance* $\delta \geq 0$. A predictor is *admissible* if it lies in the declared class and predicts every recorded target to within δ: $d_{\mathcal{S}}\big(f(g(x),a), g(x')\big) \leq \delta$ for every $(x, a, x') \in D$.

Definition 9 (the worst query error)**.** The *query error* of a predictor f at a query q is $d_{\mathcal{S}}\big(f(q), F(q)\big)$, and its worst value over admissible predictors is $W(q) := \sup\big\{d_{\mathcal{S}}\big(f(q), F(q)\big) : f \text{ admissible}\big\}$.

3.1.6 What the data leaves open

The next three statements are elementary, but they state the problem: the data leaves a what-if answer free (Fact 1), and predictors that fit it score alike on held-out data (Fact 2). Off support, disagreement between predictors can still be measured, and Fact 3 bounds it with no assumption about the world.

Fact 1 (the dataset determines nothing off support)**.** *L depends on a predictor only through its restriction to $\mathcal{R}_{\mathcal{S}}$. Hence for any f, any $q \notin \mathcal{R}_{\mathcal{S}}$ and any $v \in \mathcal{S}$, some predictor f', not necessarily in the declared class, has $f'(q) = v$ and $L(f') = L(f)$.*

Fact 2 (held-out scores cannot separate predictors that fit)**.** *Let two predictors each predict every recorded target to within δ. Then on any subset of the dataset, including any held-out subset, their losses differ by at most δ^2.*

A held-out score also cannot show what they do off support: however the held-out pairs are chosen, they come from the same behavior policy as the rest, so they never include the actions it avoids.

Fact 3 (twin disagreement)**.** *For any admissible f_1 and f_2 and any query q, $d_{\mathcal{S}}\big(f_1(q), f_2(q)\big) \leq 2\lambda_{\mathcal{S}}\ \mathrm{dist}(q, \mathcal{R}_{\mathcal{S}}) + 2\delta$.*

If any two predictors disagree at a query, then whatever value is correct, one of them misses it by at least half their gap.

3.2 From distance to a bound

The declared constant turns distance from the data into a bound on query error. Fact 1 leaves a prediction off support free. The declared constant confines it, provided the world changes no faster than declared: nearby states under nearby actions lead to nearby next states.

Assumption 1 (the dynamics respect the declared constant)**.** Let the split be self-contained, and let λ be T's Lipschitz constant in the pair distance. The declared constant covers the world's rate, allowing for the distortion α of Definition 4: $(1+\alpha)\,\lambda/(1-\alpha) \leq \lambda_{\mathcal{S}}$.

Without distortion this reads $\lambda \le \lambda_{\mathcal{S}}$: the world changes no faster than declared. The factor allows for the model's space stretching the world's distances by up to $1+\alpha$ and shrinking them by up to $1-\alpha$. Faithfulness is a demand on the model, while this assumption is a claim about the world. A dataset can never confirm the assumption, and can refute it only once the encoder's error is known.

The next result, which we call the *band*, comes in two forms. The first is simpler but uses the ideal map h, which is not available in practice. The second uses the query the encoder forms from an observation, which is available, and pays for it with a larger encoder term.

Proposition 1 (the band)**.** *Let the model be (α, ε)-faithful and let Assumption 1 hold.*

(i) For every query $q \in h(\mathcal{Z}) \times \mathcal{A}$,

$$W(q) \;\le\; 2\lambda_{\mathcal{S}}\, \mathrm{dist}(q, \mathcal{R}_{\mathcal{S}}) \;+\; (1+\lambda_{\mathcal{S}})\,\varepsilon \;+\; \delta.$$

(ii) For every observation $x = \rho(z,u)$, action a and admissible f, the query $q = (g(x), a)$ formed by the encoder satisfies

$$d_{\mathcal{S}}\big(f(q), h(T(z,a))\big) \;\le\; 2\lambda_{\mathcal{S}}\, \mathrm{dist}(q, \mathcal{R}_{\mathcal{S}}) \;+\; (1+4\lambda_{\mathcal{S}})\,\varepsilon \;+\; \delta.$$

The band grows with the distance to the data at twice the declared constant, above a floor set by the encoder error and the fit tolerance. The proof (Appendix A) chains triangle inequalities, and the step from a recorded pair to a nearby query is the one Lipschitz interpolation uses (Calliess et al., 2020).

Remark 1 (the declared limit need not be linear in distance). Proposition 1 needs only a bound on how far the declared class and F can move over a given distance. If both move at most $\omega(t)$ over any distance t, for a nondecreasing ω, then $W(q) \le \omega\big(\mathrm{dist}(q, \mathcal{R}_{\mathcal{S}})\big) + \omega\big(\mathrm{dist}(q, \mathcal{R}_{\mathcal{S}}) + \varepsilon\big) + \varepsilon + \delta$. Lipschitz is the case $\omega(t) = \lambda_{\mathcal{S}} t$.

Proposition 2 (tightness and failure of the band)**.** *For every $\lambda \ge \lambda_{\mathcal{S}}$ there are a $(0,0)$-faithful model, a dataset, a λ-Lipschitz transition function consistent with it, and an admissible predictor whose query error equals the band at every query if $\lambda = \lambda_{\mathcal{S}}$, and exceeds it off support if $\lambda > \lambda_{\mathcal{S}}$, a case Assumption 1 excludes.*

A constant declared too high makes the band loose. One declared too low can make it false, and the dataset may not reveal that it did (Appendix A).

3.3 The encoder sets the distance

Discarding nuisance brings a query closer to the data. Distance is measured in the encoder space, so the encoder decides it. Two observations of one situation, such as a robot under different lighting, can lie far apart in the observation space, and a faithful encoder maps both to within ε of one point. A query's distance to the data then depends on how far its relevant state and action lie from a recorded pair, not on how its observation looks. At a recorded situation and action, under Assumption 1, the encoder error and the fit tolerance are the whole cost, whatever nuisance the observation carried.

Corollary 1 (the encoder buys coverage)**.** *Let the model be (α, ε)-faithful, and suppose the dataset records a transition from an observation of the relevant state z_r, under any nuisance, with action a_r. Then (i) every query $q = (h(z), a)$ satisfies $\mathrm{dist}(q, \mathcal{R}_{\mathcal{S}}) \le (1+\alpha)\, d_{\mathcal{Z}}(z, z_r) + d_{\mathcal{A}}(a, a_r) + \varepsilon$, and (ii) if Assumption 1 also holds and $(z, a) = (z_r, a_r)$, then $W(q) \le (1+3\lambda_{\mathcal{S}})\varepsilon + \delta$.*

Remark 2 (aliasing buys the appearance of coverage). Discarding relevant state also shrinks the distance, but the coverage it buys is not real. If an encoder *aliases* two relevant states $z_1 \ne z_2$, mapping observations of both to one model state, a query about z_2 looks covered by a record of z_1. Definition 4 then forces $\varepsilon \ge (1-\alpha)\, d_{\mathcal{Z}}(z_1, z_2)/2$, so the band stays valid but wide, and the data alone may not reveal that cost.

4 Guiding Principles

Judge a prediction by its distance from the data. A random held-out split rarely tests a model far from the data, and near the data all predictors that fit score alike (Fact 2), even when they answer differently where a planner searches. For each query, find its distance to the nearest recorded pair in the model's space, and report it beside the prediction. Where possible, collect data near the queries

where it is largest. For the band of Proposition 1(ii), measure δ as the worst error on the recorded pairs, set ε to at least half the largest distance between the encodings of two observations of one state, and declare $\lambda_{\mathcal{S}}$ as the next paragraph describes. If $\lambda_{\mathcal{S}}$ covers the world's rate, the band holds for every predictor the dataset admits. For an estimate rather than a bound, split held-out data by distance, as blocked cross-validation (Roberts et al., 2017) does. Such an estimate is optimistic far from the data on most datasets (Figure 11) and cannot reach an unrecorded action.

Declare the constant as an assumption, and report disagreement as a lower bound on the error. Any guarantee away from the data needs an assumption (Fact 1), and the constant is the band's only one about the world. The data bounds $\lambda_{\mathcal{S}}$ only from below, by the largest secant between recorded pairs once 2δ is subtracted from the target distance (Milanese & Novara, 2004; Calliess et al., 2020). Declare $\lambda_{\mathcal{S}}$ above this bound from knowledge of the system where it exists, and not from the trained model's secants, gradient norms (Weng et al., 2018) or weights (Virmaux & Scaman, 2018), which describe the model, not the world. Use them instead to check the model's own rate against $\lambda_{\mathcal{S}}$, since the band covers only models that respect the constant. The band then bounds the error from above, but only if the constant holds, and even a careful one can fail without warning (Proposition 2). Disagreement gives a lower bound on the error with no constant. It is the largest distance between two models' predictions at a query, such as ensemble members (Chua et al., 2018), and one of the two has a query error of at least half the disagreement. A disagreement above Fact 3's bound shows that a model breaks the constant and so falls outside the band. A small disagreement does not show a small error, since both can be wrong in the same way (Figure 13).

Buy coverage with varied actions, and discard only nuisance. To narrow the band, shrink a query's distance from the data, which the recorded actions and the encoder set (Corollary 1(i)). For the actions, add noise to the behavior policy's action (Forssell & Ljung, 1999) at the states the planner will query, wide enough to span the actions it will search (Figure 3(b)). Logging longer under the same policy adds only the actions it takes, and adding noise needs influence over logging, which the offline setting often denies. For the encoder, train it by prediction in $\mathcal{S}$ rather than by reconstructing the observation, which tends to keep the nuisance (Zhang et al., 2020) and inflates ε. Prediction alone can collapse (§3.1), so add a term that spreads the embedding distribution (Balestriero & LeCun, 2025). That term can still merge different states, which the data may not reveal (Remark 2), so push apart states known to differ with a supervised contrastive loss (Khosla et al., 2020). A trained encoder may still keep part of the nuisance, and it pays in distance for what it keeps (Figure 4).

Plan with distance as a penalty. The band bounds the error at every query, so a planner can penalize uncertain actions by degree instead of cutting them off at a threshold as MOReL does (Kidambi et al., 2020), which can rule out the new actions the planner is meant to find. At one step, ε and δ are the same for every candidate, so the band measures a candidate's uncertainty by its distance to the data alone. MOPO (Yu et al., 2020) measures it by the largest standard deviation an ensemble member predicts. Because the band is conservative, penalize each candidate by its distance times a small weight, tuned as MOPO tunes its own (Appendix E.2). The penalty is then a heuristic, not a bound.

5 Experiments

Each experiment tests a claim of §3 or §4. All use Minari's datasets (Younis et al., 2024) of the MuJoCo halfcheetah, ant and humanoid, each recorded under three behavior policies, and train their models on the same six train/validation splits. Unlike a practitioner, we have the simulator, so the two experiments that test the band can measure the world's rate far from the data. Where a figure shows one dataset, Appendix D shows the others, and Appendix C gives the details.

Models that score alike diverge far from the data. This tests Facts 1 and 2 on trained models. For each dataset we choose the architecture by its held-out score and train six models that differ only in the rows they train on. On every dataset the six score alike on held-out data, yet their error and disagreement grow with distance from the data (Figure 1(a)). Their errors stay close to one another, but their disagreement exceeds the error itself, so they are wrong in different directions. Distance thus says where to trust them. Disagreement tracks the error here, but it can fall while the error rises, as under a strong weight penalty (Figure 13).

The band is within a small factor of the worst error. To test how tight Proposition 1 is for trained models, an adversarial search looks for the largest error at each distance from the data, over states

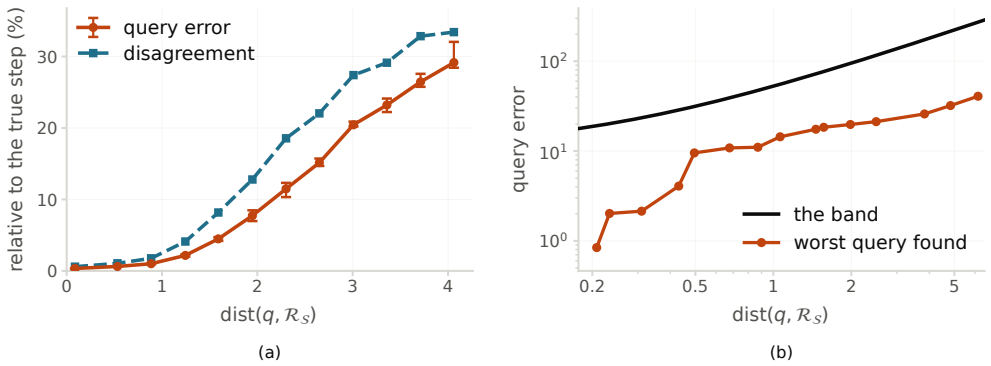


Figure 1: **(a) Models that score alike on held-out data diverge with distance, and (b) the band comes within a small factor of the worst error the search finds.** Halfcheetah/expert, against the distance from the query to the nearest recorded pair. (a) Query error and disagreement of six models, relative to the true step. Error is their median, with bars for their range, and disagreement is the largest distance between two of their predictions. (b) The band and the worst query the search found in each distance bin, in absolute units, on log axes.

and actions the simulator can execute. We set λ_S to the largest secant the simulator or the model realizes over the queries searched, the smallest covering both. The fit tolerance δ is the model's worst error on the recorded pairs, and ε is zero since the encoder keeps the simulator's state. On every dataset, the band at its closest is 2.0 to 4.0 times the worst error found (Figure 1(b), Table 1), and 2.0 to 4.5 times with λ_S fitted on other queries, where it still holds at every query (Appendix C.4).

A constant declared too low can break the band. To test Proposition 2, we declare each model's constant at the largest rate it realizes over the queries searched, the model's own rate that §4 warns against. Each model then respects its constant, so any crossing means the world breaks Assumption 1. To push the constant below the world's rate, we retrain the first experiment's models under an L_2 weight penalty (Krogh & Hertz, 1991) at three strengths. The strongest lowers the constant on every dataset (Table 4). Wherever the constant covers the world's rate, the worst query stays below the band, as Proposition 1 forces. The strongest penalty puts the constant below the world's rate on every dataset, yet the worst query crosses the band on only three, on every split there (Figure 2).

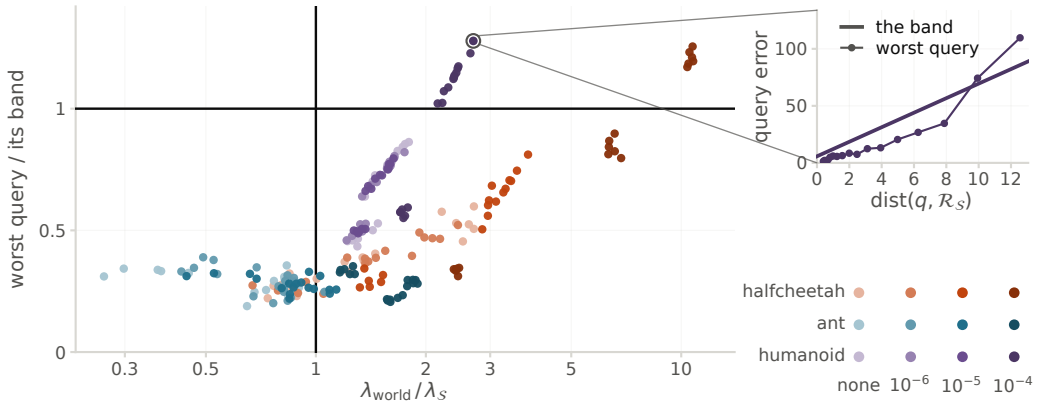


Figure 2: **At the model's own rate, a strong weight penalty can break the band, and only where the constant falls below the world's rate.** Each point is a trained model on one of the nine datasets, colored by environment and lighter for a weaker L_2 penalty. On the horizontal axis, a value past 1 means the world breaks Assumption 1. On the vertical axis, a value past 1 means the worst query exceeds the band. The panel on the right magnifies the circled model, on humanoid/expert under the strongest penalty: its band and the worst query the search found in each distance bin, in absolute units.

Widely varied actions lower the error far from the data. This tests §4's advice to buy coverage with varied actions. Keeping each dataset's size fixed, we replace a share of it with new transitions: a recorded state, its recorded action plus Gaussian noise, and the next state from the simulator. With moderate noise the error far from the data falls on every dataset, even at states that received no varied action, and on most datasets further as the share grows (Figure 3(a)). At a fixed share, narrow noise barely helps while wider noise helps far more (Figure 3(b)), so how widely the actions vary matters more than how many are varied. Most queries far from the data that the new transitions did not bring closer also improve (Appendix C.5), so varying the action helps beyond shrinking the distance.

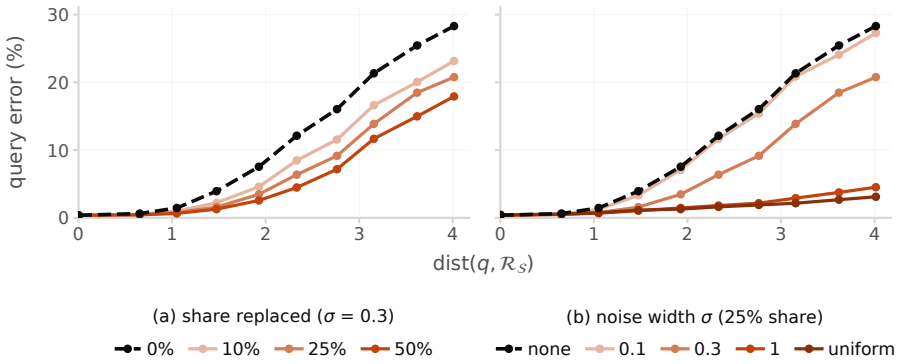


Figure 3: **Varying the action lowers the error far from the data, more by width than by share.** Query error on halfcheetah/expert when part of the dataset is replaced by transitions whose recorded action gets Gaussian noise, (a) by the share replaced and (b) by the width of the noise, up to a uniformly drawn action. Distance is to the nearest recorded pair of the original dataset. Error is relative to the true step, and only queries at states that received no varied action are scored.

An encoder must discard nuisance, and only nuisance. This tests §4's advice to discard only nuisance, and Remark 2. Humanoid's observation carries the actuator forces, which the next state does not depend on (Appendix C.1 explains why), so they are nuisance. Each query keeps a recorded observation and action and changes only those columns, to values recorded elsewhere. An encoder that keeps only the relevant state leaves the query on the data, and its error does not move. One that keeps the full observation carries the change into its model space, so the query drifts from the data and its error grows (Figure 4). One trained on the full observation to predict the next state in $\mathcal{S}$, with a term against collapse, discards most of the nuisance and stays close to the data, but it also loses part of the relevant state, so its error grows anyway. Distance does not reveal this aliasing.

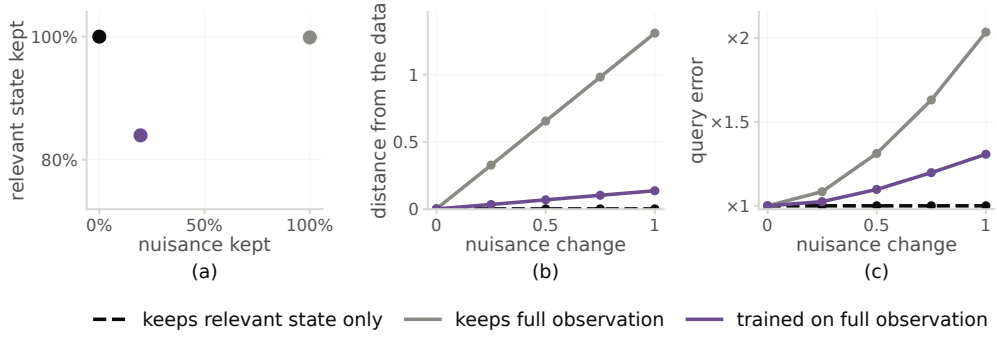


Figure 4: **An encoder pays in distance for the nuisance it keeps, and in error for the relevant state it loses.** Three encoders on humanoid/expert: one that keeps only the relevant state, one that keeps the full observation, and one trained on it to predict the next state. Each query changes only the nuisance, the actuator forces, in quarter steps toward values recorded elsewhere. (a) What each encoder keeps: the share of the relevant state a probe recovers, and its nuisance spread relative to the encoder that keeps the full observation. (b) Distance to the data, in units of relevant-state distance. (c) Query error, relative to the same encoder before the change. Medians over the six splits.

6 DISCUSSION

Definitions of a good representation. The standard definitions are stated either over all actions or in expectation, and neither form pins down the answer at a single query. Bisimulation (Givan et al., 2003; Ferns et al., 2012), the closest standard notion to faithfulness, quantifies over all actions, including those the data lacks. The information bottleneck (Tishby et al., 2000) and distances between distributions (Ben-David et al., 2006) work in expectation, where an unrecorded query carries no weight. Faithfulness avoids both, because it constrains only how the encoder maps situations and never mentions actions. The declared constant handles the actions the data lacks, and if it covers the world's rate, the band holds at each query.

Identification and coverage. Whether a dataset determines the dynamics is an old question, and at an action it never recorded, most existing answers are all-or-nothing. Closed-loop identification (Forssell & Ljung, 1999) asks whether the recorded actions vary enough to reveal the dynamics, which a fixed feedback policy may not allow, since the state sets its action. Offline reinforcement learning charges an infinite price at any reachable pair the behavior policy cannot produce (Chen & Jiang, 2019), or keeps the policy near the recorded actions (Fujimoto et al., 2019; Kumar et al., 2019), which Li et al. (2022) relax to a learned distance on average. Where the data leaves the dynamics undetermined, Proposition 1 still bounds the error from above under the declared constant, by a band that widens with the distance to the recorded pairs, as set-membership identification does for states measured directly (Milanese & Novara, 2004), and half the disagreement of two models bounds the larger of their errors from below, with no constant.

Generalization and domain adaptation. Both bound an average error under a distribution, while a planner asks about single queries, often off the distribution the data came from. A PAC bound (Shalev-Shwartz & Ben-David, 2014) holds under the behavior policy's own distribution, however large the sample. Importance weighting (Cortes et al., 2010) and conformal prediction under covariate shift (Tibshirani et al., 2019) assume that the source covers the target. Divergence bounds (Ben-David et al., 2010) drop that premise but rest on the best predictor's error on both distributions, which depends on the answers at the queries, the very thing the data leaves open. The band replaces that unknown with the declared constant and bounds the error at each query.

Underspecification and ensembles. Many predictors fit a dataset equally well. Facts 1 and 2 state it as a property of the loss, the underspecification literature (D'Amour et al., 2022) observes it in practice, and the set is called a Rashomon set (Fisher et al., 2019). An ensemble (Lakshminarayanan et al., 2017) holds a few of these predictors, and half the disagreement of two members bounds the larger of their errors only from below (§4). A small disagreement does not necessarily imply a small error, since the members can be wrong in the same way (Figure 13). Offline model-based reinforcement learning nonetheless treats a small predicted spread as a small error, using it where its guarantees assume an upper bound on model error (Yu et al., 2020; Kidambi et al., 2020). Lu et al. (2021) show that the spread correlates with the true error, but a correlation does not bound the error at a single query.

Causality. A what-if question asks for the outcome of an intervention, and the data is observational. The obstacle, though, is positivity (Petersen et al., 2012) rather than confounding. The action was never taken, so there is nothing to adjust for. Learning a causal model does not escape this. An agent needs such a model to keep its regret bounded under local interventions, in an environment its decisions do not change (Richens & Everitt, 2024), but data from one policy cannot supply it alone, because building it needs the very actions the policy never took.

Limits. The evidence is from control benchmarks with learned predictors, not from a large-scale world model. Except in the encoder experiment, the encoder is exact, so $\varepsilon = 0$ and only the predictor is tested. The band covers one step where a planner asks about a horizon, and composing it with the compounding-error line would extend it. The theory takes the dynamics to be deterministic for simplicity, and Remark 4 says what changes without that. Appendix F continues this discussion.

AI USE STATEMENT

In this work, we used generative AI tools to write code, to help design, run and analyze experiments, and to help prove mathematical claims. We also used them to create figures, search the literature, and draft and polish parts of the paper. We did not use them to generate synthetic data or to clean

or reformat datasets. We directed the work and verified every part of it. We checked the proofs, double-checked the code, and inserted every citation by hand after reading the cited paper. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

Reproducibility statement

All experiments use public data, Minari's MuJoCo datasets, and step the MuJoCo simulator directly for ground truth (Appendix C). Appendix A proves Propositions 1 and 2 in full, and §3 states every definition and assumption they use. The Facts follow from the definitions, and Corollary 1 from Definition 4 and Proposition 1. Appendix C gives the dataset construction, the encoder, the network, training and selection, the query set and distance, the worst-query search with its validity checks and held-out constants, and the coverage sweep, and Appendix E the blocked cross-validation, planning and weight-decay experiments. Trained weights reproduce bit-for-bit only on the same GPU type. Code will be released upon publication.

References

Kavosh Asadi, Dipendra Misra, and Michael Littman. Lipschitz continuity in model-based reinforcement learning. In *International conference on machine learning*, pp. 264–273. PMLR, 2018.

Randall Balestriero and Yann LeCun. Lejepa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.

Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. *Advances in neural information processing systems*, 30, 2017.

Jake Bruce, Michael D Dennis, Ashley Edwards, Jack Parker-Holder, Yuge Shi, Edward Hughes, Matthew Lai, Aditi Mavalankar, Richie Steigerwald, Chris Apps, et al. Genie: Generative interactive environments. In *Forty-first international conference on machine learning*, 2024.

Jan-Peter Calliess, Stephen J Roberts, Carl Edward Rasmussen, and Jan Maciejowski. Lazily adapted constant kinky inference for nonparametric regression and model-reference adaptive control. *Automatica*, 122:109216, 2020.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *International conference on machine learning*, pp. 1042–1051. PMLR, 2019.

Sayak Ray Chowdhury and Aditya Gopalan. On kernelized multi-armed bandits. In *International Conference on Machine Learning*, pp. 844–853. PMLR, 2017.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. *Advances in neural information processing systems*, 31, 2018.

Corinna Cortes, Yishay Mansour, and Mehryar Mohri. Learning bounds for importance weighting. *Advances in neural information processing systems*, 23, 2010.

Alexander D'Amour, Katherine Heller, Dan Moldovan, Ben Adlam, Babak Alipanahi, Alex Beutel, Christina Chen, Jonathan Deaton, Jacob Eisenstein, Matthew D Hoffman, et al. Underspecification presents challenges for credibility in modern machine learning. *Journal of Machine Learning Research*, 23(226):1–61, 2022.

Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 134(1):19–67, 2005.

Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine learning research*, 7(Jan):1–30, 2006.

Omar Darwiche Domingues, Pierre Ménard, Matteo Pirotta, Emilie Kaufmann, and Michal Valko. Kernel-based reinforcement learning: A finite-time analysis. In *International Conference on Machine Learning*, pp. 2783–2792. PMLR, 2021.

Norman Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. *arXiv preprint arXiv:1207.4114*, 2012.

Christian Fiedler. Lipschitz and Hölder continuity in reproducing kernel hilbert spaces. *arXiv preprint arXiv:2310.18078*, 2023.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of machine learning research*, 20(177):1–81, 2019.

Urban Forssell and Lennart Ljung. Closed-loop identification revisited. *Automatica*, 35(7):1215–1241, 1999.

Scott Fujimoto, David Meger, and Doina Precup. Off-policy deep reinforcement learning without exploration. In *International conference on machine learning*, pp. 2052–2062. PMLR, 2019.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial intelligence*, 147(1-2):163–223, 2003.

Suriya Gunasekar, Jason D Lee, Daniel Soudry, and Nati Srebro. Implicit bias of gradient descent on linear convolutional networks. *Advances in neural information processing systems*, 31, 2018.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. *Advances in neural information processing systems*, 31, 2018.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025a.

Danijar Hafner, Wilson Yan, and Timothy Lillicrap. Training agents inside of scalable world models. *arXiv preprint arXiv:2509.24527*, 2025b.

Aapo Hyvärinen and Petteri Pajunen. Nonlinear independent component analysis: Existence and uniqueness results. *Neural networks*, 12(3):429–439, 1999.

Aapo Hyvarinen, Hiroaki Sasaki, and Richard Turner. Nonlinear ica using auxiliary variables and generalized contrastive learning. In *The 22nd international conference on artificial intelligence and statistics*, pp. 859–868. Pmlr, 2019.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32, 2019.

Ilyes Khemakhem, Diederik Kingma, Ricardo Monti, and Aapo Hyvarinen. Variational autoencoders and nonlinear ica: A unifying framework. In *International conference on artificial intelligence and statistics*, pp. 2207–2217. PMLR, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

David Klindt, Yann LeCun, and Randall Balestriero. When does lejepa learn a world model? *arXiv preprint arXiv:2605.26379*, 2026.

Torsten Koller, Felix Berkenkamp, Matteo Turchetta, and Andreas Krause. Learning-based model predictive control for safe exploration. In *2018 IEEE conference on decision and control (CDC)*, pp. 6059–6066. IEEE, 2018.

Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

Aviral Kumar, Justin Fu, Matthew Soh, George Tucker, and Sergey Levine. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in neural information processing systems*, 32, 2019.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Jianxiong Li, Xianyuan Zhan, Haoran Xu, Xiangyu Zhu, Jingjing Liu, and Ya-Qin Zhang. When data geometry meets deep function: Generalizing offline reinforcement learning. *arXiv preprint arXiv:2205.11027*, 2022.

Jeremiah Liu, Zi Lin, Shreyas Padhy, Dustin Tran, Tania Bedrax Weiss, and Balaji Lakshminarayanan. Simple and principled uncertainty estimation with deterministic deep learning via distance awareness. *Advances in neural information processing systems*, 33:7498–7512, 2020.

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Cong Lu, Philip J Ball, Jack Parker-Holder, Michael A Osborne, and Stephen J Roberts. Revisiting design choices in offline model-based reinforcement learning. *arXiv preprint arXiv:2110.04135*, 2021.

Lucas Maes, Quentin Le Lidec, Damien Scieur, Yann LeCun, and Randall Balestriero. Leworld-model: Stable end-to-end joint-embedding predictive architecture from pixels. *arXiv preprint arXiv:2603.19312*, 2026.

Mario Milanese and Carlo Novara. Set membership identification of nonlinear systems. *Automatica*, 40(6):957–975, 2004.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Tianwei Ni, Benjamin Eysenbach, Erfan Seyedsalehi, Michel Ma, Clement Gehring, Aditya Mahajan, and Pierre-Luc Bacon. Bridging state and history representations: Understanding self-predictive rl. In *International Conference on Learning Representations*, volume 2024, pp. 23555–23569, 2024.

Il-Seok Oh. A tutorial on world models and physical ai. *arXiv preprint arXiv:2606.12783*, 2026.

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

Maya L Petersen, Kristin E Porter, Susan Gruber, Yue Wang, and Mark J Van Der Laan. Diagnosing and responding to violations in the positivity assumption. *Statistical methods in medical research*, 21(1):31–54, 2012.

Jonathan Richens and Tom Everitt. Robust agents learn causal world models. In *International Conference on Learning Representations*, volume 2024, pp. 15786–15817, 2024.

David R Roberts, Volker Bahn, Simone Ciuti, Mark S Boyce, Jane Elith, Gurutzeta Guillera-Arroita, Severin Hauenstein, José J Lahoz-Monfort, Boris Schröder, Wilfried Thuiller, et al. Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8):913–929, 2017.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Sean Sinclair, Tianyu Wang, Gauri Jain, Siddhartha Banerjee, and Christina Yu. Adaptive discretization for model-based reinforcement learning. *Advances in Neural Information Processing Systems*, 33:3858–3871, 2020.

Karl-Theodor Sturm. *Probability measures on metric spaces of nonpositive curvature*. SFB 611, 2003.

Yiyou Sun, Yifei Ming, Xiaojin Zhu, and Yixuan Li. Out-of-distribution detection with deep nearest neighbors. In *International conference on machine learning*, pp. 20827–20840. PMLR, 2022.

Ryan J Tibshirani, Rina Foygel Barber, Emmanuel Candes, and Aaditya Ramdas. Conformal prediction under covariate shift. *Advances in neural information processing systems*, 32, 2019.

Nadav Timor, Ravid Shwartz-Ziv, Micah Goldblum, Yann LeCun, and David Harel. On training in imagination. *arXiv preprint arXiv:2605.06732*, 2026.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. *arXiv preprint physics/0004057*, 2000.

Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.

Joost Van Amersfoort, Lewis Smith, Yee Whye Teh, and Yarin Gal. Uncertainty estimation using a single deep deterministic neural network. In *International conference on machine learning*, pp. 9690–9700. PMLR, 2020.

Aladin Virmaux and Kevin Scaman. Lipschitz regularity of deep neural networks: analysis and efficient estimation. *Advances in neural information processing systems*, 31, 2018.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. *arXiv preprint arXiv:1801.10578*, 2018.

Grady Williams, Andrew Aldrich, and Evangelos A Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40 (2):344–357, 2017.

Denis Yarats, David Brandfonbrener, Hao Liu, Michael Laskin, Pieter Abbeel, Alessandro Lazaric, and Lerrel Pinto. Don't change the algorithm, change the data: Exploratory data for offline reinforcement learning. *arXiv preprint arXiv:2201.13425*, 2022.

Omar G. Younis, Rodrigo Perez-Vicente, John U. Balis, Will Dudley, Alex Davey, and Jordan K Terry. Minari, September 2024. URL `https://doi.org/10.5281/zenodo.13767625`.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization. *Advances in neural information processing systems*, 33:14129–14142, 2020.

Amy Zhang, Rowan McAllister, Roberto Calandra, Yarin Gal, and Sergey Levine. Learning invariant representations for reinforcement learning without reconstruction. *arXiv preprint arXiv:2006.10742*, 2020.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

A PROOFS

Proof of Proposition 1, given for completeness. F is $\lambda_{\mathcal{S}}$-Lipschitz. For $s = h(z)$ and $\tilde{s} = h(\tilde{z})$, the upper bound of Definition 4, Assumption 1 and the lower bound give

$$\begin{aligned} d_{\mathcal{S}}\big(F(s,a), F(\tilde{s},\tilde{a})\big) &\le (1+\alpha)\,\lambda\,\big(d_{\mathcal{Z}}(z,\tilde{z}) + d_{\mathcal{A}}(a,\tilde{a})\big) \\ &\le \tfrac{1+\alpha}{1-\alpha}\,\lambda\,\big(d_{\mathcal{S}}(s,\tilde{s}) + d_{\mathcal{A}}(a,\tilde{a})\big) \;\le\; \lambda_{\mathcal{S}}\, d\big((s,a),(\tilde{s},\tilde{a})\big). \end{aligned}$$

Clause (i). Let f be admissible and $q \in h(\mathcal{Z}) \times \mathcal{A}$. Take a recorded transition from $x = \rho(z_r, u)$ to $x' = \rho(z', u')$ under a_r, with pair $r = (g(x), a_r)$. Self-containment gives $h(z') = F(\tilde{r})$ for $\tilde{r} = (h(z_r), a_r)$, and faithfulness gives $d(r, \tilde{r}) \le \varepsilon$. So

$$\begin{aligned} d_{\mathcal{S}}\big(f(q), F(q)\big) &\le d_{\mathcal{S}}\big(f(q), f(r)\big) + d_{\mathcal{S}}\big(f(r), g(x')\big) + d_{\mathcal{S}}\big(g(x'), h(z')\big) + d_{\mathcal{S}}\big(F(\tilde{r}), F(q)\big) \\ &\le \lambda_{\mathcal{S}}\, d(q,r) + \delta + \varepsilon + \lambda_{\mathcal{S}}\big(d(q,r) + \varepsilon\big), \end{aligned}$$

by the Lipschitz bound on f, admissibility, faithfulness and the Lipschitz bound on F. Minimizing over the recorded pairs gives (i).

Clause (ii). Put $q = (g(x), a)$ with $x = \rho(z, u)$, and $\tilde{q} = (h(z), a)$. Faithfulness gives $d(q, \tilde{q}) \le \varepsilon$, so $\operatorname{dist}(\tilde{q}, \mathcal{R}_{\mathcal{S}}) \le \operatorname{dist}(q, \mathcal{R}_{\mathcal{S}}) + \varepsilon$. Since $F(\tilde{q}) = h(T(z, a))$,

$$d_{\mathcal{S}}\big(f(q), h(T(z,a))\big) \;\le\; d_{\mathcal{S}}\big(f(q), f(\tilde{q})\big) + d_{\mathcal{S}}\big(f(\tilde{q}), F(\tilde{q})\big) \;\le\; \lambda_{\mathcal{S}}\varepsilon + W(\tilde{q}),$$

and bounding $W(\tilde{q})$ by (i) gives the stated bound. □

Proof of Proposition 2. Take $\mathcal{S}$ normed, h the identity with $\alpha = \varepsilon = 0$, a unit vector e, $\delta = 0$, and recorded targets all equal to p. Let $\lambda \ge \lambda_{\mathcal{S}}$ and put $T(y) := p - \lambda \operatorname{dist}(y, \mathcal{R}_{\mathcal{S}})\, e$ and $f(y) := p + \lambda_{\mathcal{S}} \operatorname{dist}(y, \mathcal{R}_{\mathcal{S}})\, e$. Since $\operatorname{dist}(\cdot, \mathcal{R}_{\mathcal{S}})$ is 1-Lipschitz and vanishes on $\mathcal{R}_{\mathcal{S}}$, T is λ-Lipschitz and f is $\lambda_{\mathcal{S}}$-Lipschitz, and both return p at every recorded pair: T is consistent with the dataset, f is admissible, and Assumption 1 holds at $\lambda = \lambda_{\mathcal{S}}$ and fails for $\lambda > \lambda_{\mathcal{S}}$. At every q they differ by $(\lambda + \lambda_{\mathcal{S}}) \operatorname{dist}(q, \mathcal{R}_{\mathcal{S}})$, while the band is $2\lambda_{\mathcal{S}} \operatorname{dist}(q, \mathcal{R}_{\mathcal{S}})$, so the error equals the band at $\lambda = \lambda_{\mathcal{S}}$ and exceeds it off support for $\lambda > \lambda_{\mathcal{S}}$. For a fit tolerance $\delta > 0$, replacing f by $f + \delta e$ adds δ to both the error and the band. □

B FURTHER REMARKS

Remark 3 (what faithfulness concedes). Faithfulness cannot be had from the identifiability literature in general. Aligning an encoder with a chosen split is a nonlinear-ICA requirement, and nonlinear ICA is unidentifiable in general (Hyvärinen & Pajunen, 1999), so passive data does not reveal which split an encoder has aligned itself with. Our condition is weaker, since relevant distinctions need only survive up to constants and no canonical parameterization of z is needed, and where identifiability does hold it delivers more: recent work (Klindt et al., 2026) recovers the latent up to a linear orthogonal map for its own split, the case $\alpha = 0$, $\varepsilon = 0$. But those results reach the encoder, not the self-containment Definition 2 asks of the split, which quantifies over every action. That passive data cannot confirm faithfulness is Remark 2, and §4 takes up what can be done.

Remark 4 (determinism). Stochastic dynamics change what the model is asked to predict, not the argument. Here the transition function and the encoder are functions, so the setting is deterministic. With noise the recorded target is a draw rather than a value, and the loss is minimized by the conditional Fréchet mean of the next model state given the pair, the conditional mean itself when $\mathcal{S}$ is an inner-product space and any barycenter that exists otherwise (Sturm, 2003), since $\mathcal{S}$ carries a metric and no linear structure. The fit tolerance δ is then read against the minimum attainable error, and everything in §3 applies to that minimizer, except that the irreducible noise adds a term at every comparison with a recorded target, one that no data and no restriction on the class removes.

Remark 5 (separate constants). The predictor and the world can have different constants. Declare the predictor λ_f-Lipschitz, and let Assumption 1 read $(1+\alpha)\lambda/(1-\alpha) \le \lambda_F$, which makes F Lipschitz with constant λ_F. The proof of Proposition 1 then runs unchanged, with the slope $2\lambda_{\mathcal{S}}$ replaced by $\lambda_f + \lambda_F$, and the encoder terms by $(1 + \lambda_F)\varepsilon$ in (i) and $(1 + 2\lambda_f + 2\lambda_F)\varepsilon$ in (ii). Proposition 2's construction, with a λ_f-Lipschitz f, attains this band at $\lambda = \lambda_F$, and Fact 3's bound becomes $2\lambda_f \operatorname{dist}(q, \mathcal{R}_{\mathcal{S}}) + 2\delta$. Assumption 1 then involves only λ_F, while λ_f is a property of the

model, which can be enforced by construction or bounded from its weights (Virmaux & Scaman, 2018). Since $\lambda_f + \lambda_F \leq 2\max(\lambda_f, \lambda_F)$, the bands of §5, which set λ_S to the larger of the two rates, are at least as wide as these.

C Experimental details

C.1 Datasets and the encoder

The data. The nine datasets are Minari's `mujoco/{halfcheetah,ant,humanoid}/{expert,medium,simple}-v0` (Younis et al., 2024). From each we draw whole episodes at random until we have $200{,}000$ transitions. We draw episodes rather than individual rows because the behavior policy's own structure is the thing under study.

Ground truth. The true next state is the simulator's. We step MuJoCo (Todorov et al., 2012) 3.2.3, the version the datasets were generated with, directly from the same model file and at the same frame skip. Stepping this way reproduces the recorded transitions to numerical precision.

The encoder. An observation gives the robot's configuration and its velocities, leaving out where the body sits horizontally in the world, which the dynamics do not depend on. So for halfcheetah the observation is already a self-contained split (Definition 2) and the encoder is the identity on it. Ant and humanoid carry further columns that the simulator computes as it steps: contact forces, body inertias, actuator forces. The next state does not depend on them given the state, so they are nuisance rather than new information, and we drop them, which takes ant from 105 columns to 27 and humanoid from 348 to 45. The kept columns are the simulator's state apart from horizontal position. On all nine datasets $\alpha = 0$ and $\varepsilon = 0$ in Definition 4, and every experiment except the encoder experiment measures the predictor alone.

The encoder experiment. Humanoid's dropped columns differ. The actuator forces are the previous action times the actuator gear, so they change while the relevant state does not, and the rest are functions of the state. A query replaces the actuator forces with those of another recorded action, and the true next observation comes from the simulator. We compare the encoder above, one keeping all 348 columns, and one trained on them to predict its next state in model space with a contrastive term (Oord et al., 2018) whose negatives are the rest of the batch. Each encoder is scored in its own model space, rescaled so that representation distances match relevant-state ones, and on that scale a uniformly drawn action lies about six units from the data. What an encoder keeps is the share of the relevant state a probe recovers from its representation and its nuisance spread relative to the keeping encoder. The previous action is largely predictable from the state, so these queries are combinations the data itself rarely holds.

C.2 Models, training and selection

The network. A fully connected network with $\tanh$ activations, predicting the state increment rather than the next state.

Training. Mean squared error on the increment, with inputs and target put on a common scale. Adam (Kingma & Ba, 2014), batch $1{,}024$, $600{,}000$ steps, and the learning rate annealed on a cosine schedule (Loshchilov & Hutter, 2016) from 10^{-3} to 10^{-5}. The L_2 penalty adds to the loss a strength times the sum of squared weights, biases excluded.

Selection. Nine architectures (widths 128 to 2048, depths 3, 4 and 6), each trained on six $80/20$ splits of every dataset. We select the one with the lowest *median* held-out score across its six splits, read at its best held-out checkpoint.

C.3 Queries and the distance

The query set. Disagreement is a maximum over pairs of models, so it means something only if the models are asked the same question. Each dataset therefore gets one query set, built once and used by every run. It has three blocks: $3{,}000$ recorded pairs, $3{,}000$ queries pairing a recorded state with a uniformly drawn executable action, and a stratified block of $6{,}000$. The stratified block is what the figures plot, except where the worst-query search supplies the queries instead. It interpolates from

each recorded action toward targets drawn inside the action box and at its corners, and then keeps equal numbers per distance interval, so the far half of the axis is not starved. Every candidate is executable because the box is convex.

The distance. Each state and action coordinate is divided by its standard deviation over the data, putting them on a common scale. On that scale $\mathrm{dist}(q, \mathcal{R}_\mathcal{S}) = \min_{r \in \mathcal{R}_\mathcal{S}} \big(\|\Delta s\| + \|\Delta a\|\big)$, searched exhaustively over the recorded pairs rather than approximated. Curves are binned by quantile in this distance, so every bin holds the same number of queries. Curves of the worst query the search found use log-spaced bins instead. Here $\mathcal{R}_\mathcal{S}$ holds all $200{,}000$ recorded pairs, including the fifth each model held out, and the δ of §5 is a model's worst error over the same pairs. Relative error understates how fast error grows with distance, since the true step is larger far from the data. Figure 6 plots the same curves in absolute units.

C.4 The worst-query search

The search. We take 192 recorded pairs as starting points and look for the query that maximizes error in an interval of distances from one of them, repeating this over twelve intervals so the whole axis is covered. It runs per dataset and per split, in two stages: a screen of 48 random directions from each starting point in each interval, and then a projected-gradient ascent of 25 steps, each taking the best of four step sizes, from 24 starting points per distance interval. A candidate may never move closer to the recorded data than where it began, nor past its distance interval's far edge, so the ascent cannot buy error by drifting back toward it.

What moves. A query is a state together with an action, and the search may displace either. We divide the displacement between the two and sweep that division over five settings, from action-only to state-only, giving each the same budget. The action-only setting reproduces a search that never moves the state, and serves as a control.

Validity. MuJoCo does not reject an impossible state. It warns, silently resets, and returns a finite and unremarkable next state, which a search maximizing error will walk straight into. Every candidate is therefore checked and a failing one dropped rather than scored: the candidate must lie inside the dataset's own box and its joints' limits, the simulator must neither warn on it nor silently reset, and the candidate state is passed through the encoder, with its projection what gets queried.

The constants. $\lambda_\mathcal{S}$ is the maximum secant, between a query and its nearest recorded pair, that either the simulator or the model realizes over the queries evaluated. The secant grows sharply as the displacement shrinks, so $\lambda_\mathcal{S}$ covers only displacements at least as large as the smallest one searched, t_0 (Remark 1 with $\omega(t) = \lambda_\mathcal{S} \max(t, t_0)$), which is all the band needs at the queries evaluated, since with $\varepsilon = 0$ the proof of Proposition 1 applies the constant only between a query and its nearest recorded pair. $\lambda_\mathcal{S}$ is a lower bound, since a harder search could raise it. This is why §5 reports how close the worst query comes, rather than that nothing can cross the band.

Held-out constants. Fitting $\lambda_\mathcal{S}$ on the queries the band is scored on makes the band hold there by construction. We therefore also fit it on a random half of the starting pairs, or on the nearer half by distance, and score the band on the remaining queries. The fitted constant can then be up to 43% smaller than the full one, yet every held-out query on all 54 models stays inside the band. A constant fitted on some queries thus covers others.

Tightness of the band. Table 1 gives how close the worst query the search found comes to the band along the whole distance axis. We cut the axis into five fifths that hold equal numbers of queries, and in each fifth take the smallest ratio of the band to the query error over the queries the search found there. The tightest of the five is the factor §5 reports. The median and loosest show how much looser the band is elsewhere on the axis. Where the world sets $\lambda_\mathcal{S}$ (halfcheetah/medium and /expert and the three humanoid datasets), the closest query has a world secant within a quarter of $\lambda_\mathcal{S}$ and a model secant at most a quarter of it. The model barely moves there, so the error is close to the world's own change, and the band exceeds it by little more than its factor of two, plus δ (Remark 5). A constant fitted on other queries (*Held-out constants* above) removes the coincidence between the query that sets $\lambda_\mathcal{S}$ and the one scored, and the band's closest approach is then 2.0 to 4.5 times the worst error (Table 1, last column).

Table 1: **How close the worst query comes to the band along the distance axis,** on the common scale of the distance. Unregularized models, nine datasets, six splits. Of the five fifths of the axis, the columns give the tightest, the median and the loosest, each as the median over the six splits $\pm$ the median absolute deviation. The last column gives the tightest when λ_S is fitted on a random half of the starting pairs and the other half is scored (*Held-out constants*), as the median over the six splits and both halves $\pm$ the median absolute deviation. Lower is tighter.

	band / query error			
Dataset	tightest	median	loosest	held-out
halfcheetah/simple	3.77 ± 0.36	6.59 ± 0.28	7.47 ± 0.56	3.86 ± 0.21
halfcheetah/medium	3.23 ± 0.11	5.94 ± 0.04	9.96 ± 0.28	3.32 ± 0.15
halfcheetah/expert	3.23 ± 0.19	4.54 ± 0.18	5.77 ± 0.14	3.18 ± 0.26
ant/simple	3.38 ± 0.14	4.14 ± 0.03	4.63 ± 0.25	3.73 ± 0.24
ant/medium	3.99 ± 0.30	5.30 ± 0.10	5.76 ± 0.08	4.46 ± 0.41
ant/expert	3.04 ± 0.06	4.79 ± 0.48	6.68 ± 0.25	3.31 ± 0.34
humanoid/simple	2.62 ± 0.03	4.06 ± 0.06	4.58 ± 0.06	2.87 ± 0.69
humanoid/medium	2.01 ± 0.01	3.74 ± 0.26	4.25 ± 0.12	2.03 ± 0.33
humanoid/expert	2.04 ± 0.02	4.07 ± 0.16	4.96 ± 0.22	2.05 ± 0.16

C.5 The coverage sweep

The data. Every arm keeps 200,000 transitions. It keeps the first part of the dataset and replaces the rest, a share of 10%, 25% or 50%, with as many transitions at distinct states drawn from the part it keeps, each under the recorded action plus Gaussian noise of standard deviation σ clipped to the action box, with the next state from the simulator. The noise is in the simulator's action units, and the box is $[-1, 1]$ for halfcheetah and ant and $[-0.4, 0.4]$ for humanoid. The share sweep uses $\sigma = 0.3$, and the width sweep replaces 25% and uses $\sigma \in \{0.1, 0.3, 1\}$ and a uniform action, all at the same states.

The distance check. Among the queries farthest from the data, we keep those the added transitions brought less than 5% closer to the arm's own recorded pairs. With half the data replaced, most of them have lower error than the first experiment's on the same split, on every dataset and every split.

C.6 Compute

Training ran on a shared cluster of NVIDIA GPUs of several types. Weights reproduce bit-for-bit only on the same GPU type.

D Results on all nine datasets

Each figure here extends a figure of §5 to all nine datasets: query error and disagreement (Figure 5, and Figure 6 in absolute units), the band and the worst query (Figure 7), coverage by share and by width (Figures 8 and 9), and the encoder on the other two behavior policies (Figure 10). Axes are per panel rather than shared, since the datasets differ severalfold in how far their queries reach, so the shapes are comparable across panels and the heights are not.

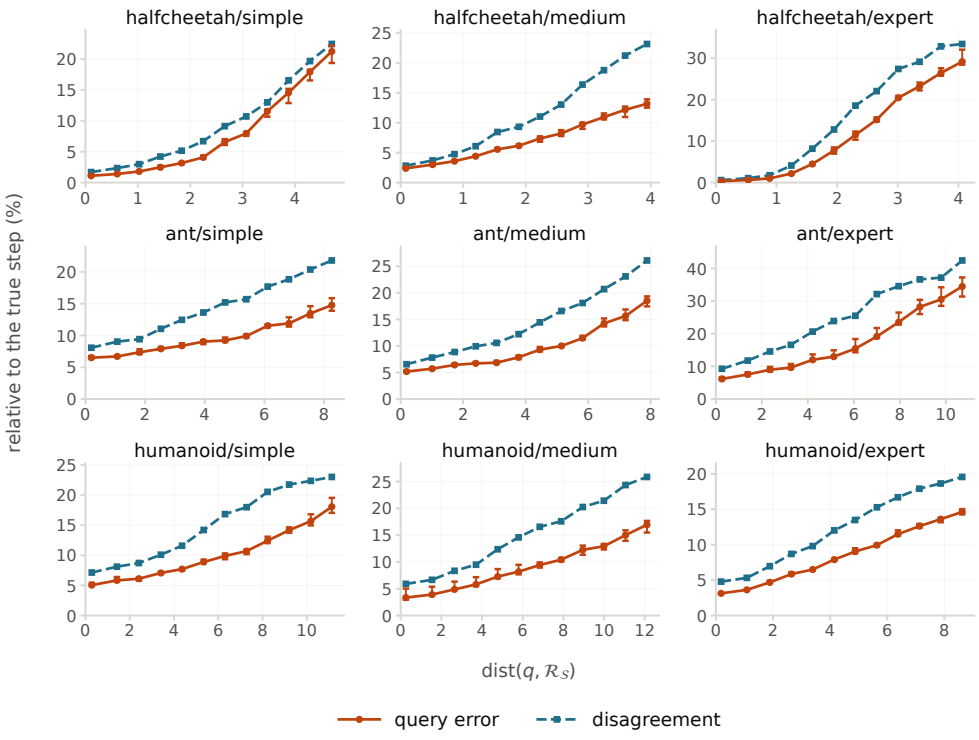


Figure 5: **On every dataset, query error and disagreement grow with the distance from the data.** Three environments, each at three behavior policies. Construction is Figure 1(a)'s, and the architecture is selected per dataset by the same rule.

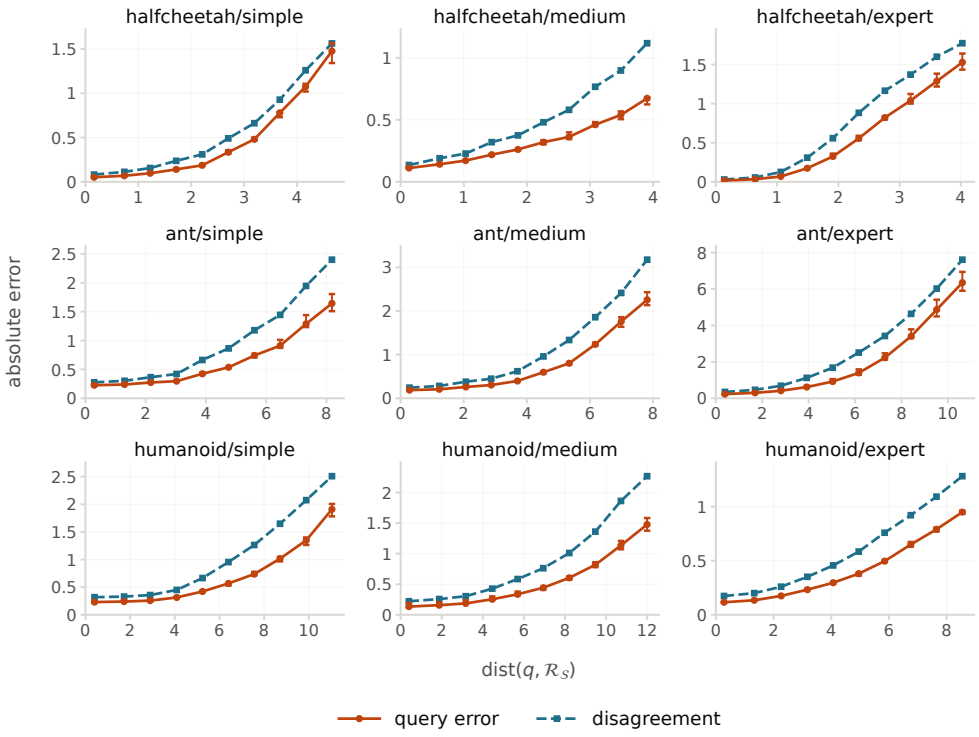


Figure 6: **The same growth holds in absolute units.** Construction is Figure 5's, with error and disagreement not divided by the true step.

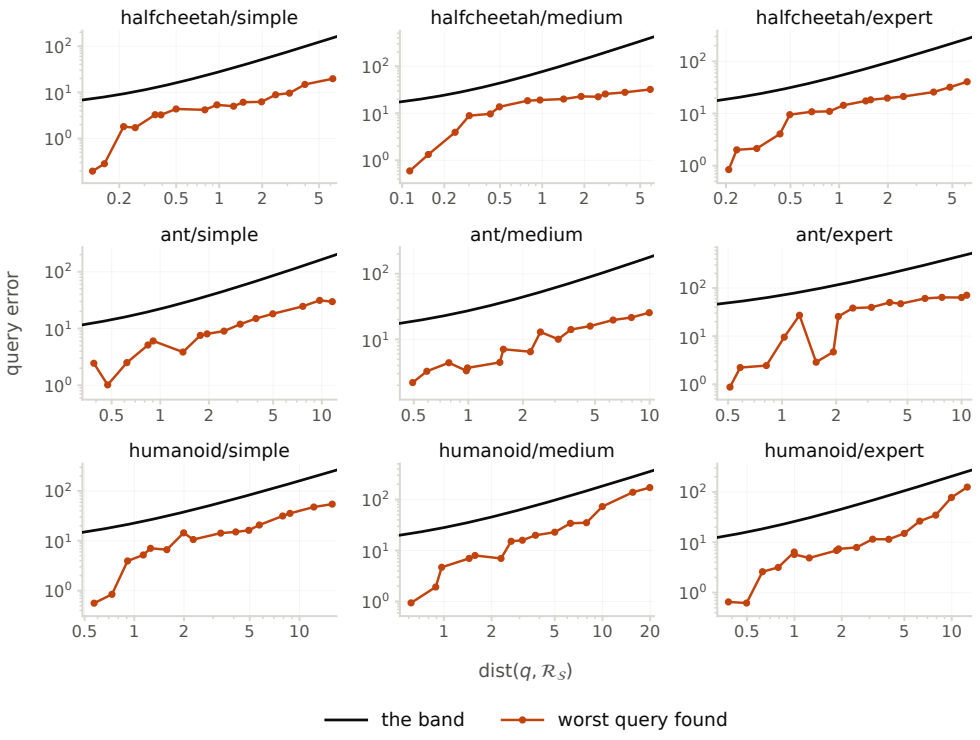


Figure 7: **On every dataset the band comes within a small factor of the worst query the search found.** Construction is Figure 1(b)'s.

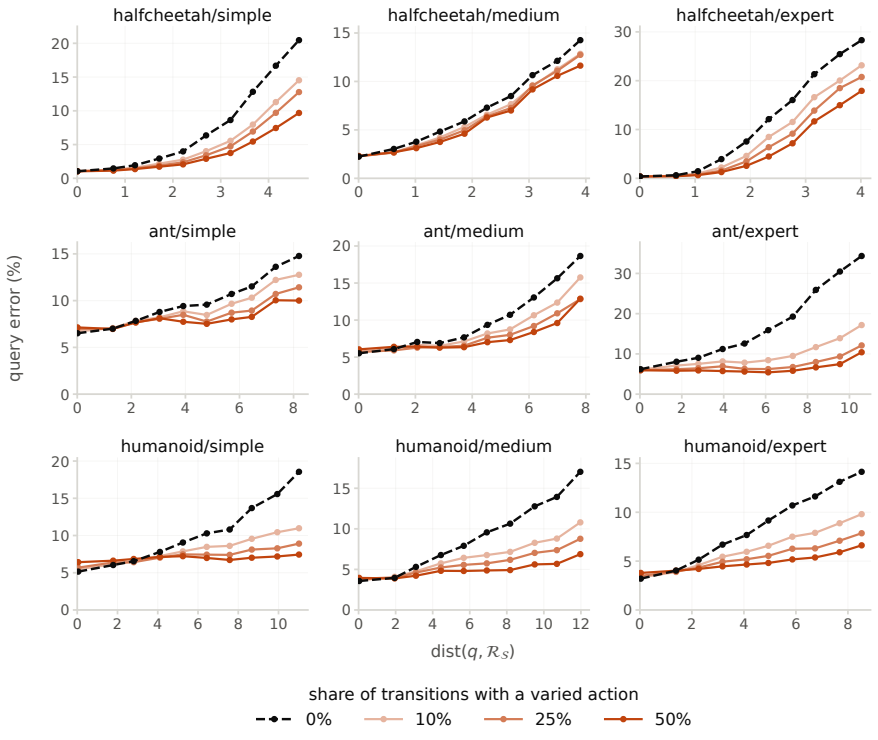


Figure 8: **On every dataset the error far from the data falls when part of it carries varied actions, and on most, further as the share grows.** Coverage by share on all nine datasets. Construction is the left panel of Figure 3.

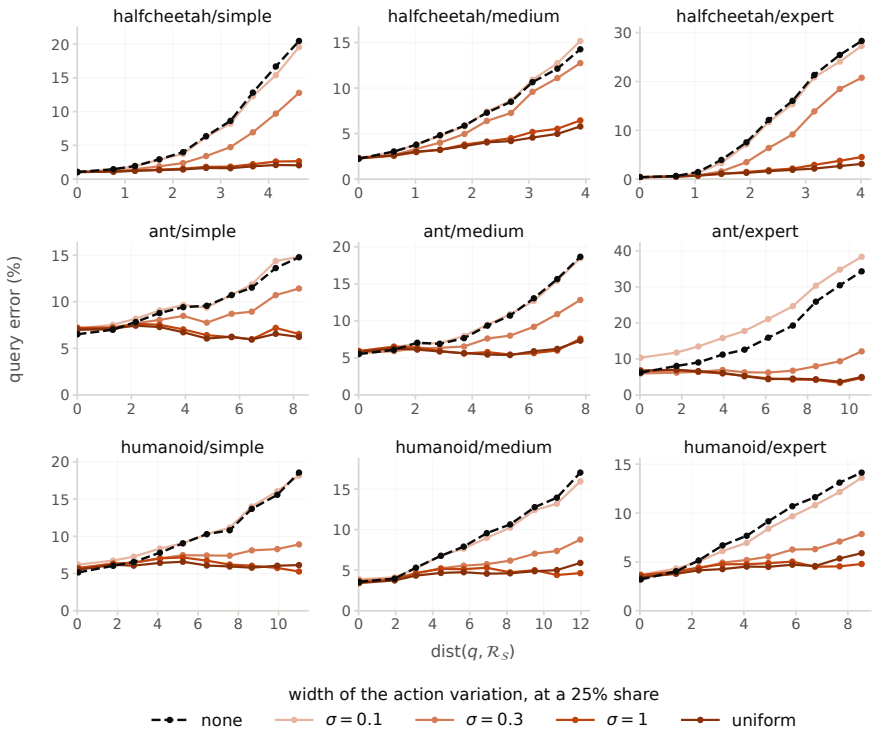


Figure 9: **On every dataset wider noise lowers the error far more than narrow noise.** Coverage by width on all nine datasets. Construction is the right panel of Figure 3.

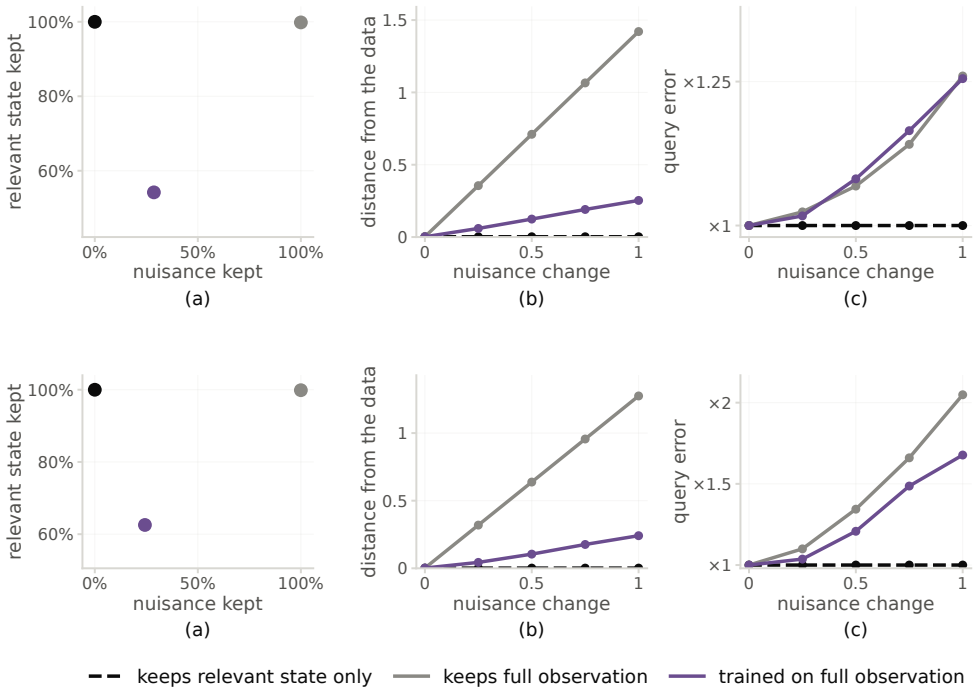


Figure 10: **The encoder result repeats on the other two behavior policies. humanoid/simple** (top) and **humanoid/medium** (bottom). Construction is Figure 4's.

E FURTHER EXPERIMENTS

E.1 BLOCKED CROSS-VALIDATION

Folds split by distance reach farther, and still read optimistic. This tests whether held-out data can estimate the error far from the recorded pairs when the folds are split by distance rather than at random. We hold out one fold at the edge of the recorded actions and one at random, train on the rest, and score every held-out pair and every query by its distance to the nearest pair the model trained on. On all nine datasets the blocked fold reaches farther from the data than the random fold, and its error rises with the query error (Figure 11). It is not an unbiased estimate: far from the data it reads below the query error on most datasets, while on ant it reads above it near the data, and where both folds reach the random one often reads lower still. So the measurement locates the error better than a random split, and understates it far from the data.

The folds and models. We hold out two folds of 10% each and train on the remaining 80%. The blocked fold is the 10% of recorded pairs whose action lies farthest from the mean action of the 200 nearest recorded states, so it removes the edge of the recorded actions as one region. The random fold is drawn from the other rows. Each dataset uses Figure 1's architecture, trained over three draws of the random fold and read at its final checkpoint. For every held-out pair and every query we record the distance to the nearest pair the model trained on and the model's absolute error there.

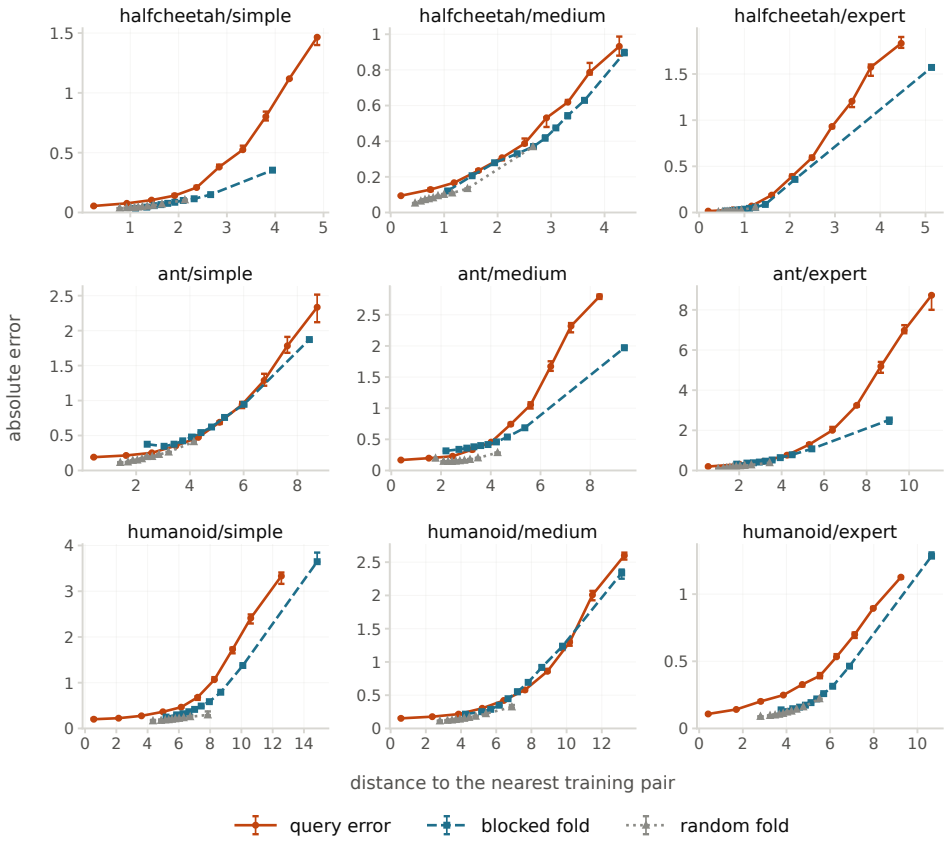


Figure 11: **The blocked fold reaches farther than the random one, and still reads below the query error far from the data on most datasets.** Absolute error against the distance to the nearest pair the model trained on, for off-support queries and for pairs held out as a block at the edge of the recorded actions or at random. Each curve is the median over three models and the bars are their range.

E.2 PLANNING WITH A DISTANCE PENALTY

At one step, distance ranks candidates better than an ensemble's spread. This tests §4's advice to plan with distance as a penalty. At recorded states the planner picks the candidate action whose predicted next state lands nearest a goal, placed at twice the change a perturbed action makes, so reaching it needs an action the data does not hold. One predictor serves every arm, and only the penalty differs: none, a cut-off on ensemble disagreement as in MOReL (Kidambi et al., 2020), the disagreement itself in place of MOPO's predicted standard deviation (Yu et al., 2020), and distance to the data. The ensemble has five members trained from different initializations, so that a shared one does not narrow their disagreement. Each weight or threshold is tuned against the simulator, as those methods do, on held-out states. For these goals, penalizing distance lowers regret against no penalty and against the cut-off on all nine datasets, and more than penalizing the spread on eight (Figure 12), significantly across datasets in each case (Table 3). Table 2 gives the mean regrets. Within a single dataset, six draws resolve the gap over the spread on two. The tuned cut-off never beats no penalty. With its weight left at one, the band penalizes so heavily that the planner keeps the recorded action, as a MOReL cut-off at its strictest would.

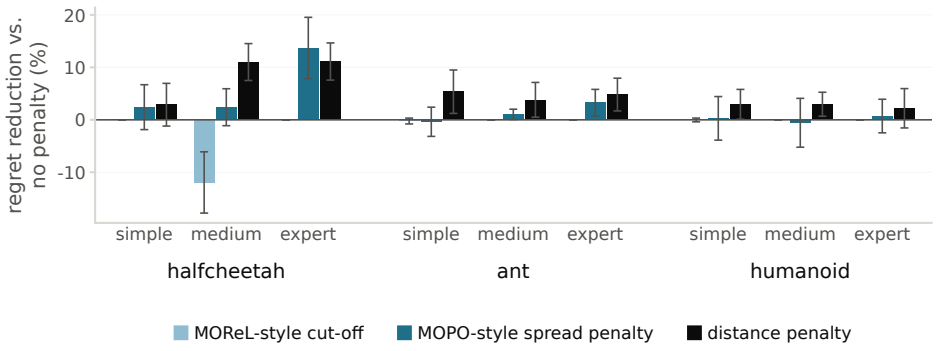


Figure 12: **Penalizing distance to the data lowers regret on every dataset, and on most more than penalizing an ensemble's spread.** Each bar is how much a penalty lowers regret against no penalty, in percent, for goals beyond the recorded action's reach, on all nine datasets. Mean over six draws of test problems with the models fixed, with $\pm$ one standard deviation. The tuned cut-off forbids almost nothing on most datasets, so its bars sit near zero.

The task. Each test draw takes 100 recorded states at random. At each, the candidates are the recorded action and 512 Gaussian perturbations of it at each of eight scales, from $1/64$ to 1 times half the action range, clipped to the action box, as a sampling-based planner would propose them, such as the cross-entropy method (De Boer et al., 2005) or MPPI (Williams et al., 2017). Each state also gets 16 goals: we perturb the recorded action at one of 16 scales over the same range and place the goal twice as far from the recorded action's outcome as that perturbation lands. The planner picks the candidate minimizing the predicted distance to the goal plus its penalty, and regret is how much farther its landing falls than the best candidate's, in the output metric of Appendix C. The design choices were set on separate draws.

The arms. Every arm uses one predictor, the split-0 model of Appendix C, and only the penalty differs. Disagreement is the largest distance between the predictions of an ensemble, the best five by held-out score of seven models trained from different initializations with the predictor's architecture, schedule and training rows, as MOPO builds it, so a shared initialization does not narrow the disagreement. The predictor is not a member, and the disagreement stands in for MOPO's variance heads, which deterministic models lack. The cut-off forbids candidates whose disagreement exceeds a threshold, and always allows the recorded action. Our penalty is $c_0 \left(s(z)/\bar{s}\right) 2\lambda_{\mathcal{S}} \operatorname{dist}(q, \mathcal{R}_{\mathcal{S}})$, where $s(z)$ is the norm of the predictor's finite-difference gradient with respect to the action at the state's recorded pair, $\bar{s}$ its median over the tuning states, and $\lambda_{\mathcal{S}}$ the constant of §5. It is a heuristic built on the band, not a bound.

Tuning. Each arm is tuned on its own on 100 held-out states of split 0, with the same task and the simulator's answer, as offline model-based methods tune theirs. The weights range over 26 log-

spaced values from 10^{-4} to 10, and the threshold over 20 quantiles of the disagreement and none at all.

Significance. Table 3 tests the distance penalty against each other arm, on the percent reduction in regret that Figure 12 plots. Within a dataset the test is a paired t-test over the six draws. Across datasets it is the Wilcoxon signed-rank test over the nine datasets' mean differences, which Demšar (2006) recommends for comparing two methods over multiple datasets. On six datasets the tuned cut-off leaves regret unchanged on every draw, so there its column repeats the no-penalty one.

Other results. With an ensemble of the other split models, our penalty is also below the spread's on all nine datasets (Wilcoxon $p = 0.004$). For goals at the perturbed action's own outcome, with the original ensemble, it is below the spread's on eight datasets ($p = 0.07$) and below the cut-off's on all nine. With its weight left at one, $c_0 = 1$, the band picks the recorded action at almost every goal and does 4 to 41 times worse than no penalty.

Table 2: **Regret of the four planners for goals beyond the recorded action's reach,** in units of 10^{-2} of the output metric. Each entry is the mean over six test draws $\pm$ the standard deviation.

Dataset	no penalty	MOReL-style cut-off	MOPO-style spread	ours
halfcheetah/simple	4.17 ± 1.03	4.17 ± 1.03	4.05 ± 0.96	4.05 ± 1.03
halfcheetah/medium	9.26 ± 1.42	10.35 ± 1.55	9.04 ± 1.47	8.23 ± 1.20
halfcheetah/expert	6.69 ± 0.88	6.69 ± 0.88	5.81 ± 1.09	5.96 ± 0.86
ant/simple	10.81 ± 1.97	10.83 ± 1.95	10.84 ± 1.95	10.18 ± 1.54
ant/medium	9.73 ± 1.28	9.73 ± 1.28	9.62 ± 1.24	9.34 ± 1.08
ant/expert	29.79 ± 2.23	29.79 ± 2.23	28.79 ± 1.73	28.31 ± 1.34
humanoid/simple	6.78 ± 0.94	6.78 ± 0.92	6.74 ± 0.85	6.57 ± 0.86
humanoid/medium	3.59 ± 0.30	3.59 ± 0.30	3.60 ± 0.28	3.48 ± 0.28
humanoid/expert	2.85 ± 0.35	2.85 ± 0.35	2.83 ± 0.34	2.78 ± 0.30

Table 3: **p-values of the distance penalty against each other arm,** for goals beyond the recorded action's reach, per dataset and across datasets. Bold is $p < 0.05$, and † marks the one case where the other arm lowers regret more.

Dataset	vs no penalty	vs cut-off	vs spread
halfcheetah/simple	0.14	0.14	0.81
halfcheetah/medium	**<0.001**	**<0.001**	**0.001**
halfcheetah/expert	**<0.001**	**<0.001**	$0.18^\dagger$
ant/simple	**0.02**	**0.02**	**0.009**
ant/medium	**0.04**	**0.04**	0.06
ant/expert	**0.01**	**0.01**	0.16
humanoid/simple	0.05	0.06	0.09
humanoid/medium	**0.02**	**0.02**	0.08
humanoid/expert	0.21	0.21	0.44
Across datasets	**0.004**	**0.004**	**0.03**

E.3 WEIGHT DECAY

Weight decay can lower the error, but not the world's share of the band. To test whether a common regularizer, an L_2 weight penalty (Krogh & Hertz, 1991), can stand in for declaring the constant, we retrain the models of §5's first experiment at three strengths. The strongest penalty lowers the models' rate on every dataset (Table 4), but the band's constant must cover the world's rate, which no penalty changes, so the band's slope cannot fall below it. Far from the data, a mild penalty lowers the error on most datasets and a strong one underfits and raises it, while every strength we tried lowers the disagreement (Figure 13, and Figures 14 and 15 on all nine datasets). A falling disagreement is thus no sign of a falling error.

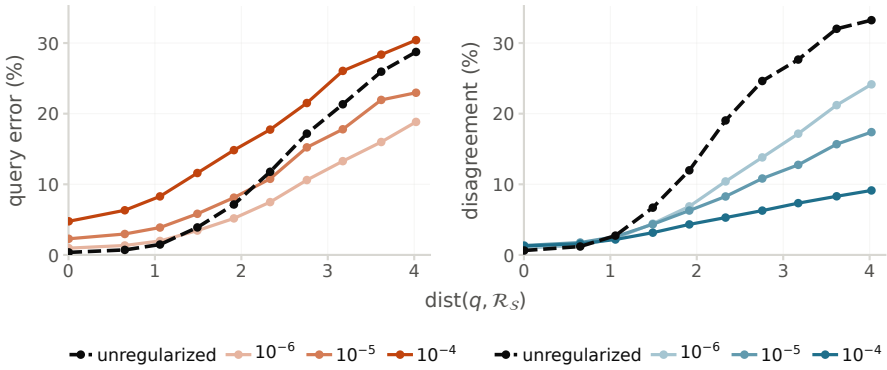


Figure 13: **Far from the data, a mild weight penalty lowers the error and a strong one raises it, while every strength lowers the disagreement.** Query error and disagreement on halfcheetah/expert against the distance from the query to the nearest recorded pair, both relative to the true step, for the unregularized models and under an L_2 weight penalty at one strength per decade. Error is the median over the six models, and disagreement is the largest distance between two of their predictions.

The realized rate. Table 4 gives the rate the models realize, the largest secant a model realizes over the queries the search of Appendix C.4 evaluated, for the unregularized models and under the L_2 penalty.

Table 4: **The rate the models realize, with and without an L_2 weight penalty,** on the common scale of the distance. Each entry is the median over the six splits $\pm$ the median absolute deviation.

	L_2 penalty strength			
Dataset	none	10^{-6}	10^{-5}	10^{-4}
halfcheetah/simple	11.49 ± 0.37	11.80 ± 0.47	8.01 ± 0.13	4.77 ± 0.01
halfcheetah/medium	25.71 ± 1.51	24.16 ± 1.05	9.72 ± 0.31	3.10 ± 0.03
halfcheetah/expert	8.24 ± 0.41	10.44 ± 0.48	7.17 ± 0.11	3.37 ± 0.06
ant/simple	7.46 ± 0.28	7.27 ± 0.10	7.04 ± 0.14	5.18 ± 0.04
ant/medium	7.99 ± 0.63	7.40 ± 0.52	6.48 ± 0.23	4.58 ± 0.10
ant/expert	23.43 ± 1.35	19.92 ± 1.30	9.85 ± 0.93	4.32 ± 0.08
humanoid/simple	5.87 ± 0.29	5.74 ± 0.12	5.84 ± 0.10	4.41 ± 0.06
humanoid/medium	5.99 ± 0.12	6.26 ± 0.08	6.39 ± 0.03	4.27 ± 0.08
humanoid/expert	5.50 ± 0.15	5.70 ± 0.05	5.62 ± 0.05	3.42 ± 0.06

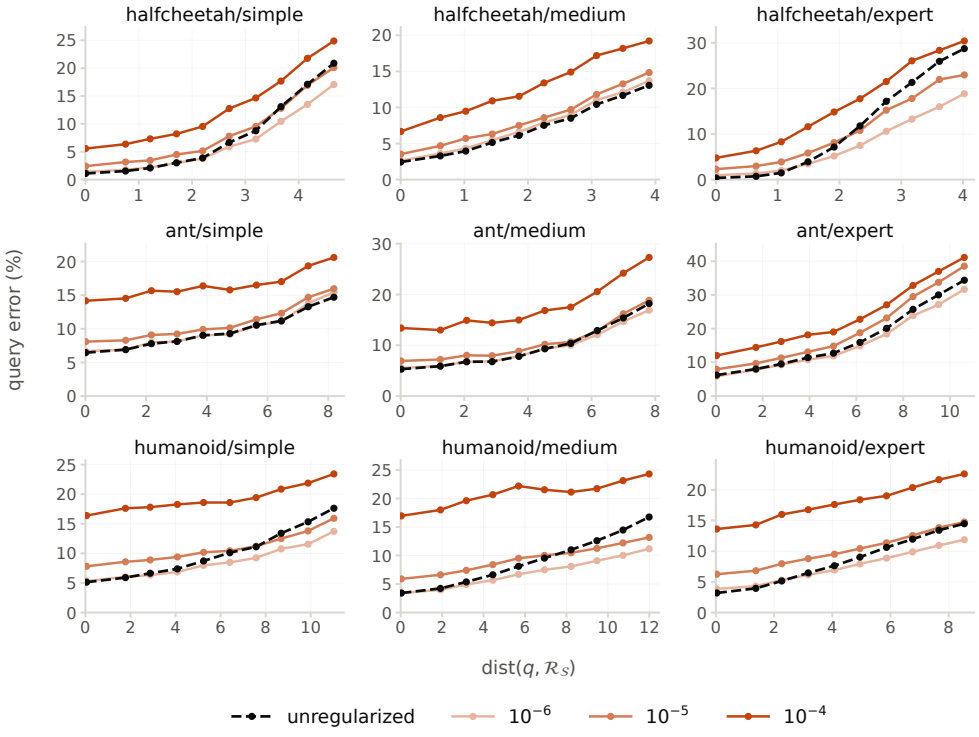


Figure 14: **Far from the recorded pairs a mild penalty lowers the error on most datasets, and one too strong raises it.** The weight-decay sweep on all nine datasets: query error. Construction and strengths are Figure 13's.

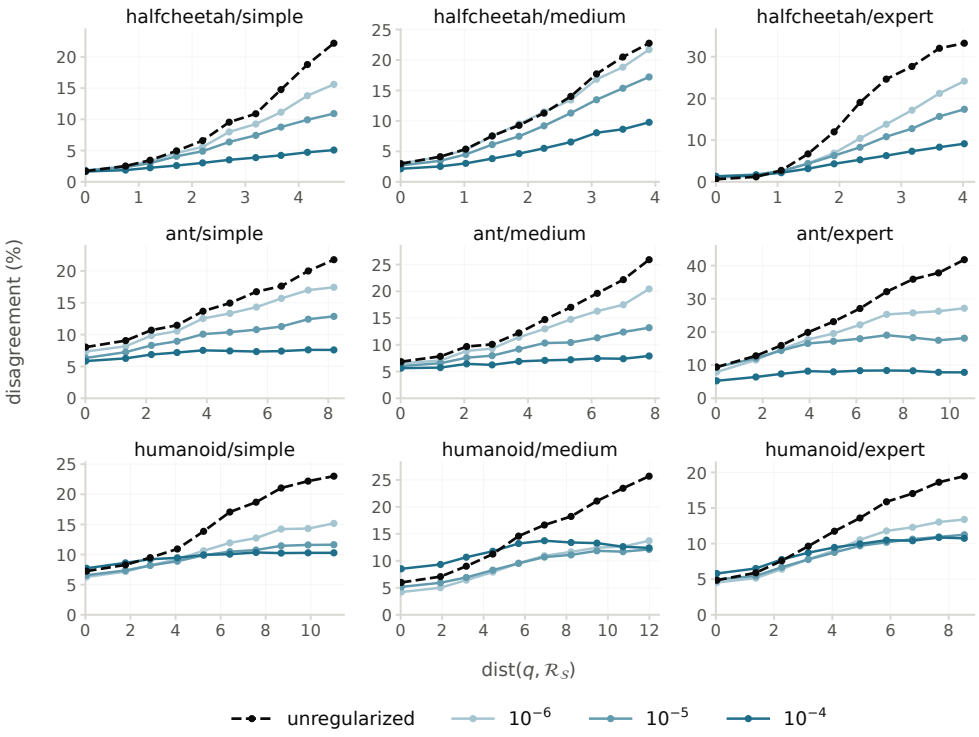


Figure 15: **Far from the recorded pairs every strength lowers the disagreement.** The weight-decay sweep on all nine datasets: disagreement. Construction and strengths are Figure 13's.

F FURTHER DISCUSSION

Joint embedding. A recent joint-embedding world model (Maes et al., 2026) trains the encoder by prediction in model space and prevents collapse with a condition on the embedding distribution (Balestriero & LeCun, 2025), the recipe §4 adopts. Such a condition does not by itself stop an encoder from merging different states, which the data may not reveal (Remark 2). Our encoder, trained by prediction with a contrastive term against collapse, lost part of the relevant state in the same way (Figure 4). Nor can any encoder settle the answer at an action never recorded (Fact 1). The band bounds the error of that answer under the declared constant.

Self-containment. The split of the observation is the one choice that cannot be fixed later. A weaker encoder costs a larger ε. But if the split leaves out something the dynamics depend on, such as a robot's velocities (§3.1), no encoder trained on it can help (Remark 3).

Kernel confidence bounds. If a function's kernel norm is bounded and the noise is sub-Gaussian, a Gaussian-process posterior mean errs, with high probability, by at most a multiple of its posterior standard deviation (Chowdhury & Gopalan, 2017). Safe model-based reinforcement learning plans with such bounds (Koller et al., 2018), or establishes where a learned policy is stable and carries that guarantee to nearby points with a Lipschitz constant, the step the band also takes (Berkenkamp et al., 2017). A kernel-norm bound also supplies such an ω (Fiedler, 2023), though Remark 1's band uses only the nearest recorded pair, where a posterior conditions on all of them, as the optimal bounds of set-membership identification do (Milanese & Novara, 2004), and so can be tighter on a given dataset. The kernel norm is declared like the constant, and a dataset cannot confirm it either. These bounds concern one model's prediction with no encoder in front, while the band covers every predictor that fits the data and respects the declared constant, and measures distance through the encoder.

Reinforcement learning on metric spaces. This line uses the band's step to decide where to explore, while we use it to decide how far to trust a trained model. Its exploration bonus bounds an estimate's error at a query by a Lipschitz constant times the distance to the samples (Sinclair et al., 2020; Domingues et al., 2021). There the bound holds for the algorithm's own estimator, on a metric of the true state given in advance. Here it holds for any predictor that fits the data and respects the constant, whatever produced it, with distance measured through a learned encoder.

Distance-aware single models. A single classifier can read uncertainty off a distance in its own representation, kept bi-Lipschitz in the input (Liu et al., 2020) or sensitive to it (Van Amersfoort et al., 2020). That resembles faithfulness, but the distance is to the input rather than to the relevant state, and neither bounds an error. The distance to the nearest training features also serves as an out-of-distribution score (Sun et al., 2022), and it too flags a query without bounding its error.

Compounding error. Bounds on how a per-step error accumulates along a rollout (Asadi et al., 2018) assume Lipschitz constants for both the learned model and the world, and need only the smaller of the two. The band instead needs a constant covering each, and one below the world's rate can make it false (Proposition 2).

Exploratory data. The data, not only the algorithm, decides what offline learning can reach. An unmodified off-policy algorithm trained on data from reward-free exploration matches or beats methods designed for offline reinforcement learning (Yarats et al., 2022). The band gives one reason such data helps a world model: it brings the queries a planner makes closer to the data. Our coverage sweep asks for less than a new exploratory dataset, only noise on the recorded action at recorded states at a fixed dataset size, and finds that how widely the actions vary matters more than how many are varied (Figure 3).