CONTENT: CONTACT-FORCE PREDICTION FOR LATENT WORLD MODELS ENHANCES DYNAMIC MANIPULATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent world models are powerful encoders of both observations and environment dynamics in robot learning. This study applies a world model to a dynamic manipulation task in which a robot uses a stick to push a ball rolling down a tilted board up into a goal zone. In this task, it is crucial that the observed *task-relevant world state*, such as the ball state, be accurately encoded in the latent space. Yet, although the regularisers of world models prevent *collapse* of the latent space, the representation can still *degenerate and discard* such information, thereby degrading the performance of model predictive control. We show that the discarded information can be recovered by *predicting contact forces*, which anchor the object representation in the latent space. We exploit this recovery to propose *ConTent, contact-force prediction for latent world models*, which outperforms the LeWorldModel baseline on our manipulation task. Our method predicts contact forces at training time and raises the success rate from 2% to 52% when planning with the cross-entropy method. Fine-tuning the predictor on on-policy data further improves it to 77%.

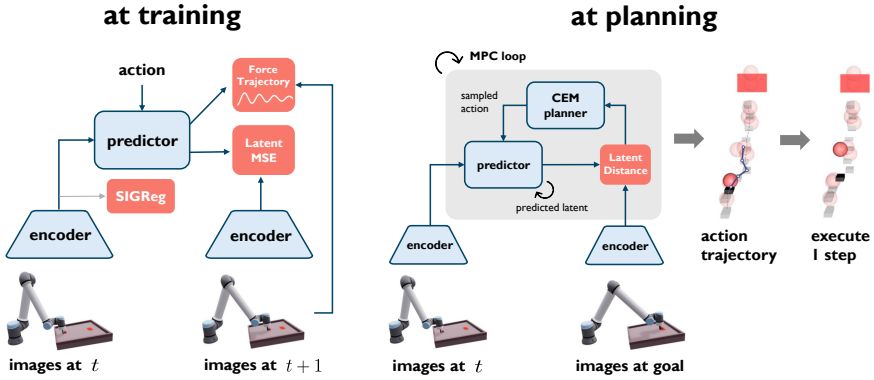

Figure 1: ConTent pipelines at training and planning. *Left*, training: the encoder and the predictor are trained with a latent prediction loss and the SIGReg regulariser, and a force head on the predictor feature predicts the contact-force trajectory of the transition. *Right*, planning: the cross-entropy method searches action sequences by the latent distance between the rolled-out prediction and the encoded goal image, and the first action is executed. No force input is used at planning.

1 INTRODUCTION

Dynamic object manipulation is at once the most exacting of the feats robot learning attempts and the most universal of the capacities a physical intelligence acquires. A manipulator acts on its environment only through contact, and it has to drive the resulting motion, produced jointly by the dynamics of the robot and of the environment, to an arbitrary goal state. A robot is therefore trained to generate trajectories towards a goal state, which is achieved by learning, implicitly or explicitly, the dynamics of the environment and the effect of acting on it (Chi et al., 2025). We study this problem on the tilted-table ball-pushing task of Schneider et al. (2022), in which an arm whose end

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.

effector is a rigid stick pushes a ball across a tilted board into a goal zone. Because the manipulation is non-prehensile, the ball under the stick sits at an unstable equilibrium, and the policy has to stabilise it continuously and learn its dynamics to a high accuracy. The task was proposed and solved with model-based reinforcement learning, in which dynamics models learned online from state observations are used for model predictive control with an exploration bonus (Schneider et al., 2022). Designing a task reward and learning online at deployment are, however, impractical in many real environments, which motivates learning the policy from offline data.

Latent world models provide an alternative way to learn such task-specific dynamics (Zhou et al., 2024; Sobal et al., 2025; Maes et al., 2026). They learn from offline data by encoding observations into a latent space and predicting the dynamics of the environment there. At planning time, the latent obtained by embedding a goal observation lets them run task optimisation through planning in latent space without an explicit task reward. Joint-embedding predictive architectures (JEPAs) learn such models directly from pixels (Sobal et al., 2025; Maes et al., 2026), yet with no reconstruction loss, so the encoder may discard appearance details and retain only what serves prediction. The SIGReg (Sketched Isotropic Gaussian Regularization) regulariser of LeJEPA (Balestriero & LeCun, 2025) stabilises this training, with a guarantee that the learned latent space does not collapse. However, JEPA encoders are still observed to discard local information even without collapse (Zeng et al., 2026; Mur-Labadia et al., 2026), which can harm downstream tasks. Our study shows that this discarding is precisely what limits manipulation, as the object representation degenerates in the latent space and planning fails with it.

Tactile sensing is an increasingly active ingredient of dexterous manipulation research. Tactile signals carry temporally dense, spatially local information about contact that vision lacks, and feeding touch to a world model can improve planning in contact-rich tasks when its representation is compatible with vision (Zhang et al., 2026). Our study exploits contact information to repair precisely this *degeneration* of JEPAs and proposes a method, *ConTent, contact-force prediction for latent world models*. *ConTent* predicts the contact force between the stick and the ball during training only, thereby anchoring the object dynamics in the latent space and improving the task performance of model predictive control with the cross-entropy method (CEM). Unlike prior work that uses touch as a policy input, measured (Lee et al., 2019; Niu et al., 2026) or predicted (Fan & Guo, 2026), *ConTent* never takes the force as an input, either to the encoder or at planning time.

Our contributions are as follows. (i) We show how a latent world model discards the task-relevant information through mathematical analysis and experiments. (ii) We propose a method predicting contact force which anchors the task-relevant information, and show that the method raises the success rate of the dynamic manipulation task from 2% to 52%. (iii) We explore the upper bound of the method by dataset aggregation with on-policy planning, and it raises the success rate further to 77%.

2 PRELIMINARIES

The task world we work on is the tilted-table ball-pushing task proposed by Schneider et al. (2022). A UR10e arm whose end effector is a rigid stick pushes a ball across a tilted board, and an episode succeeds when the ball reaches a goal zone on the board. The world state is $x \in \mathbb{R}^n$, here the configuration of the arm together with that of the ball, and (x, x') is the pair of states at two nearby timesteps of a trajectory, related by the world's own dynamics, i.e. the motion of the arm and the ball. The transition is driven by the action $a \in \mathbb{R}^d$ applied between the two timesteps, in our case a Cartesian target for the stick tip, constrained to the plane of the tilted board and mapped to joint targets by inverse kinematics, so states are drawn as $x \sim p(x)$, $a \sim \pi(a \mid x)$ and $x' \sim p(x' \mid x, a)$. The state is never seen directly, but an unknown generative process generates the observations as $o = g(x)$, e.g. rendering a scene into an image or measuring a contact force. A single frame carries V camera views of the same state at once, so an observation is the stack $o = \left(o^1, \ldots, o^V\right) \in \mathbb{R}^{D \times V}$ of V images of D pixels each. In our setting $V = 3$, namely a front, a left and a wrist view. The training set is $\mathcal{O} = \{(o_j, a_j, o'_j)\}_{j=1}^N$ and we write (o_t, a_t, o_{t+1}) as (o, a, o') unless otherwise noted. All data are collected in a simulation of this task, described in Section 5.

The learner is a pair of maps with separate parameters, an encoder $\mathsf{enc}_\theta : \mathbb{R}^{D \times V} \to \mathbb{R}^m$ that turns an observation into a vector $z = \mathsf{enc}_\theta(o)$, and a predictor $\mathsf{pred}_\phi : \mathbb{R}^m \times \mathbb{R}^d \to \mathbb{R}^m$ that has to

produce the vector of the second frame of a positive pair from the vector of the first and the action applied in between, $\widehat{z}' = \mathsf{pred}_\phi(z, a)$. Following Klindt et al. (2026), we work with the composed map $h_\theta := \mathsf{enc}_\theta \circ g : \mathbb{R}^n \to \mathbb{R}^m$ as the embedding, so that a positive pair of states gives the embedding pair $(h_\theta(x), h_\theta(x'))$ together with the action a that separates them. The two maps are trained jointly, by minimising one objective over (θ, ϕ). The objective adds to the prediction error a regulariser $S \geq 0$ that scores the embedding distribution alone, vanishing exactly at $\mathcal{N}(0, I_m)$, e.g. SIGReg (Balestriero & LeCun, 2025).

$$\mathcal{L}(h_\theta, \mathsf{pred}_\phi) := \underbrace{\mathbb{E}\big[\|\mathsf{pred}_\phi(h_\theta(x), a) - h_\theta(x')\|^2\big]}_{\text{prediction}} + \lambda \underbrace{S\big(\mathsf{law}\, h_\theta(x)\big)}_{\text{regulariser}}, \qquad \lambda > 0. \tag{1}$$

The regulariser is introduced to minimise the integrated squared bias (ISB) of readouts for arbitrary downstream tasks. Most downstream tasks reduce to a target $\tau : \mathbb{R}^n \to \mathbb{R}$ on the world state, e.g. a coordinate of the arm or of the ball, read out of the embedding by a probe trained on a finite sample $\mathcal{D} = \{(h_\theta(x_j), \tau(x_j))\}_{j=1}^N$. The probe error splits into a part the ISB governs, which Balestriero & LeCun (2025) bound for linear, kernel and k-NN probes alike, and a part no probe can remove (Lemma 1, Appendix A.B); nothing below uses the particular form of the probe, and the next section analyses the part the regulariser leaves untouched.

3 Analysis

All proofs are in Appendix A and Appendix B. The statements are about population optima, so we write h for an arbitrary measurable encoder in place of h_θ and work with $\mathcal{L}(h) := \min_{\mathsf{pred}} \mathcal{L}(h, \mathsf{pred})$, the objective with the best predictor.

Assumption 1: The world

(i) **Independence.** $p(x_i) \perp p(x_j)$ and $p(x_i' \mid x_i, a) \perp p(x_j' \mid x_j, a)$ for all $i \neq j$, with identically distributed, zero-mean coordinates of unit variance.

(ii) **Stationarity.** $\mathsf{law}(x_t, a_t) = \mathsf{law}(x, a)$ for every t along a trajectory $(x_t, a_t)_{t=1}^T$.

(iii) **Deterministic dynamics.** $x_{t+1} = F(x_t, a_t)$.

(iv) **Open-loop actions.** $a_{t:T} \perp x_t$ for every t.

Item (i) is used only for Lemma 3, Theorem 2 and the closed forms that express $\mathrm{tr}\,\mathrm{Cov}(x_\Delta)$ and $\mathbb{E}\|x_\Delta\|^4$ in n_Δ (Theorem 1, Corollary 1, Proposition 3 and Appendix A.C); it is incompatible with contact, which couples coordinates, so the retention results do not use it. Item (iii) replaces their additive-noise transition with the deterministic one of the simulator. Item (iv) idealises exploratory data collection: it holds for the scripted 16.8% of our data but not for the expert rollouts, whose policy reads the ball state, and Remark 1 states what survives in that case.

What the regulariser does and does not control. The part of the probe error the ISB governs is minimised by an isotropic Gaussian embedding law, yet the part no probe can remove, the Bayes risk $\mathbb{E}[\mathrm{Var}(\tau(x) \mid z)]$ of predicting $\tau(x)$ from $z = h(x)$, is left untouched. The regulariser sees only $\mathsf{law}\, h(x)$, and encoders with the same law can differ in this floor by the full variance of τ (Lemma 3, Appendix B), so it cannot tell whether the object is kept.

Collapse and discard. Single out the coordinates whose fate we want to follow, $\Delta \subseteq \{1, \ldots, n\}$, $n_\Delta := |\Delta|$, write x_Δ and $x_{\setminus\Delta}$ for the corresponding subvectors, and let $\mathsf{dec}_\psi : \mathbb{R}^m \to \mathbb{R}^{n_\Delta}$ be a measurable decoder that reads x_Δ off the embedding.

Definition 1: Collapse and discard

The embedding suffers *dimensional collapse* if $\lambda_{\min}(\mathrm{Cov}\, h(x)) = 0$ and *complete collapse* if $\mathrm{Cov}\, h(x) = 0$ (Jing et al., 2022). Independently of that, we measure how much of a group of latent variables survives in the embedding by the smallest error with which it can be decoded,

$$\mathsf{err}_\Delta(h) := \min_\psi \mathbb{E}\big\|x_\Delta - \mathsf{dec}_\psi(h(x))\big\|^2 = \mathbb{E}\big[\,\mathrm{tr}\,\mathrm{Cov}(x_\Delta \mid h(x))\big], \tag{2}$$

and $\mathsf{err}_{\setminus\Delta}(h)$ likewise. We say that x_Δ is *retained* if $\mathsf{err}_\Delta(h) = 0$ and *discarded* if $\mathsf{err}_\Delta(h) = \operatorname{tr}\operatorname{Cov}(x_\Delta)$, the error of a constant decoder, which is n_Δ under Assumption 1(i).

$\mathsf{err}_\Delta(h)/\operatorname{tr}\operatorname{Cov}(x_\Delta)$ is the population version of the probe error $1 - R^2$ our experiments report (Klindt et al., 2026), and $\mathsf{err}_\Delta(h) = 0$ holds exactly when this Bayes risk vanishes for every readout $\tau(x_\Delta)$ of the object (Appendix A.C). Encoders with the same law can differ in err_Δ by the full amount n_Δ (Lemma 3), so whether the object is kept is decided by the prediction term, and that term does not ask for it. Under the deterministic dynamics of Assumption 1(iii), if the object does not move the remaining coordinates, as for a light ball pushed by a stiff position-controlled arm, an encoder of $x_{\setminus\Delta}$ alone and an encoder of the full state both attain zero loss, so the objective is indifferent between discarding and retaining the object (Proposition 3, Appendix B). With noisy transitions the tie is broken against the object whenever it is the least predictable group and the embedding cannot hold every coordinate (Theorem 2). Neither statement conflicts with the identifiability result of Klindt et al. (2026), which recovers all latents when they are Gaussian with additive-noise transitions and $m = n$, none of which holds here. Driving err_Δ down therefore asks for a term that no encoder can fit while discarding the object, which an auxiliary target supplies.

An auxiliary target and retention through the dynamics. An *auxiliary target* is a square-integrable random vector $\tau^* \in \mathbb{R}^q$ carried by the transition, either a function of the state, $\tau^* = \mathsf{aux}(x)$, or of the transition, $\tau^* = \mathsf{aux}(x, a)$, the contact force being of the second kind. Predicting it from the latent and the action adds to Equation (1) the *auxiliary loss* with the best head, $\mathcal{L}^*(h) := \min_\psi \mathbb{E}\|\tau^* - \mathsf{aux}_\psi(h(x), a)\|^2$; $\mathcal{L}_{\text{force}}$ in Equation (7) is the surrogate of this loss we train with, the loss of a gated classification-and-regression head, and its infimum is approached exactly when the contact indicators and the in-contact forces are predicted exactly. A contact force is sparse in time, $\mathsf{aux} \equiv 0$ off contact, so a single transition pins the object down only on contact, and what carries that information across the contact-free steps is the prediction term. Consider therefore a trajectory $(x_t, a_t)_{t=1}^T$ with $z_t := h(x_t)$ and $\tau_t^* := \mathsf{aux}(x_t, a_t)$, under Assumption 1(ii)–(iv), none of which concerns the encoder. The *multi-step auxiliary loss* and the *observability residual* are

$$\mathcal{L}_t^{*(T)}(h) := \sum_{k=0}^{T-t} \min_{\mathsf{aux}_k} \mathbb{E}\|\tau_{t+k}^* - \mathsf{aux}_k(z_t, a_{t:T})\|^2, \quad \mathsf{risk}_t^{(T)} := \min_\varphi \mathbb{E}\|x_{\Delta,t} - \varphi\|^2, \tag{3}$$

in which $\varphi = \varphi(x_{\setminus\Delta,t:T}, a_{t:T}, \tau_{t:T}^*)$ runs over readouts of the retained trajectory, the future actions and the future targets.

Theorem 1: Multi-step retention

Assume Assumption 1(ii)–(iv) and let h be any encoder. Then for every $1 \le t \le T$:

(i) if the prediction term of $\mathcal{L}(h)$, $\mathcal{L}^*(h)$ and $\mathsf{err}_{\setminus\Delta}(h)$ all vanish, then $\mathsf{err}_\Delta(h) \le \mathsf{risk}_t^{(T)}$;

(ii) if there are $\alpha > 0$, $\beta \ge 0$ and a set $\mathcal{S}_{t:T}$ of trajectory segments with $\mathbb{P}\big((x_{t:T}, a_{t:T}) \notin \mathcal{S}_{t:T}\big) \le \delta$ such that any two segments in $\mathcal{S}_{t:T}$ generated from the same actions satisfy

$$\alpha\,\|x_{\Delta,t} - \tilde{x}_{\Delta,t}\| \le \|\tau_{t:T}^* - \tilde{\tau}_{t:T}^*\| + \beta\,\|x_{\setminus\Delta,t} - \tilde{x}_{\setminus\Delta,t}\|, \tag{4}$$

then, with $\mu_4 := \mathbb{E}[x_i^4]$ and $\mathbb{E}\|x_\Delta\|^4 = n_\Delta(n_\Delta - 1 + \mu_4)$ under Assumption 1(i),

$$\mathsf{err}_\Delta(h) \le \frac{2}{\alpha^2}\Big(\mathcal{L}_t^{*(T)}(h) + \beta^2\,\mathsf{err}_{\setminus\Delta}(h)\Big) + 2\sqrt{2\delta\,\mathbb{E}\|x_\Delta\|^4}. \tag{5}$$

For $T = t$ part (i) reduces to a one-step guarantee, stated together with its tightness as Proposition 1 in Appendix A.C. Behind (i), at zero loss two states the encoder confuses produce the same targets at every future step, so the only uncertainty that survives is the part no future target distinguishes, which is $\mathsf{risk}_t^{(T)}$. Part (ii) is the quantitative version and assumes nothing about h, holding with the measured losses $\mathcal{L}_t^{*(T)}(h)$ and $\mathsf{err}_{\setminus\Delta}(h)$. Its constants sit on the observability inequality Equation (4), not on the target map, so the contact force, which is discontinuous at the onset of contact, needs no smoothness. Lipschitz continuity enters only through the learned predictor and head, which are functions of the latent, and then Lemma 2 in Appendix A.F bounds the multi-step

loss by compounded one-step errors, so at positive loss the guarantee degrades with the number of contact-free steps before the next contact.

For the force target, $\tau^* = (c_s, f_s)_{s=1}^S$ vanishes off contact, so what is retained depends on when the contacts happen. Write Γ for the set of transitions (x, a) during which the stick touches the object, $E_t := \{(x_{t+k}, a_{t+k}) \in \Gamma \text{ for some } 0 \le k \le T - t\}$ for the event that the object is touched within the horizon, and $\delta_t^{(T)} := \mathbb{P}(E_t^c)$ for the probability that it is not.

Corollary 1: The contact force retains the object

Assume Assumption 1(ii)–(iv) and, as in Theorem 1(i), that the prediction term, $\mathcal{L}^*(h)$ and $\mathsf{err}_{\setminus\Delta}(h)$ vanish. Assume further

(i) **Contact force.** $\mathbf{1}_\Gamma(x, a) = \gamma(\mathsf{aux}(x, a))$ for some γ.

(ii) **Observability.** $x_\Delta = \varphi(x_{\setminus\Delta}, a, \mathsf{aux}(x, a))$ for some measurable φ and all $(x, a) \in \Gamma$.

(iii) **Backward deterministic dynamics.** $x_{\Delta,t} = \mathsf{back}_k(x_{\Delta,t+k}, x_{\setminus\Delta,t:t+k}, a_{t:t+k-1})$ whenever the transitions $t, \dots, t + k - 1$ are off Γ.

Then

$$\mathsf{err}_\Delta(h) \le \mathbb{E}\big[\|x_{\Delta,t}\|^2 \mathbf{1}_{E_t^c}\big] = \delta_t^{(T)}\, \mathbb{E}\big[\|x_{\Delta,t}\|^2 \,\big|\, E_t^c\big] \le \delta_t^{(T)} B^2 \quad \text{when } \|x_\Delta\| \le B. \tag{6}$$

Item (ii) says that on contact the object state is observable from the substep force trajectory together with the motion of the stick, and (iii) holds for a freely rolling ball. Sparseness of contact in time is therefore harmless: the residual is governed by $\delta_t^{(T)}$, the probability that the object is not touched again within the horizon, not by the fraction of transitions in contact. What is not claimed, closed-loop data, history-dependent predictors, and zero loss for a constrained encoder, is recorded in Remark 1 and Proposition 2 of the appendix.

4 Method

Our idea is to leverage the *contact force* as the auxiliary target of Section 3 for dynamic object manipulation, under the motivation that it anchors the information of the object at contact into the latent space and thereby serves the downstream task. Below we present the design of the architecture for the contact force and of the planning algorithm, and the search for the upper bound of our method through fine-tuning.

Model architecture and loss. As drawn on the left of Figure 1, the predictor is asked for the future contact force together with the future latent. A force head aux_ψ reads the trajectory of the contact force over the S substeps the predicted transition covers, at a frequency higher than the control rate, out of the feature $u = \mathsf{trunk}_\phi(z, a)$ the predictor produces from the latent z and the action a before it projects that feature to the predicted latent $\widehat{z}'$. Writing $(c_s, f_s)_{s=1}^S$ for the trajectory the transition carries, with $c_s \in \{0, 1\}$ the contact indicator and f_s the contact force of substep s, and $(\gamma_s, \widehat{f}_s)_{s=1}^S = \mathsf{aux}_\psi(u)$ for the readout, the force loss is

$$\mathcal{L}_{\text{force}} := \frac{1}{S} \sum_{s=1}^{S} \Big(\ell_{\text{BCE}}\big(\gamma_s,\, c_s\big) + c_s\, \ell_{\text{Huber}}\big(\widehat{f}_s - f_s\big) \Big), \tag{7}$$

a class-balanced binary cross-entropy on the contact indicators plus a Huber regression restricted to the substeps in contact. The objective adds $w\, \mathbb{E}\big[\mathcal{L}_{\text{force}}\big]$ with a weight $w > 0$ to Equation (1), so the force head is trained jointly with the encoder and the predictor, by minimising one objective over (θ, ϕ, ψ).

The encoder and the predictor follow LeWorldModel (LeWM; Maes et al., 2026): the encoder embeds each frame, and a transformer predicts the next latent from the latent history and the action, followed by an MLP projection head. The force head is a two-layer MLP (192–512–512, LayerNorm, GELU) with two linear readouts that give a contact gate and a planar force for each of the ten substeps (0.38 M parameters).

Model predictive control is executed directly in the representation space of the latent world model, as drawn on the right of Figure 1. An arbitrary goal observation is embedded with the encoder, the cost of a candidate action sequence is the mean squared error between the latent it is predicted to reach and the goal latent, and the sequence is optimised under the frozen model. We use the cross-entropy method (Maes et al., 2026) as that optimiser. At each step the method draws 300 candidate sequences from a diagonal Gaussian of zero mean and unit variance, refits it to the 30 best candidates, and repeats this for 30 iterations, with every action component bounded to the normalised range $[-1, 1]$. In the ball-pushing task the prediction error in the dynamics of the ball is unavoidable, and responding to it quickly affects the success of the task critically. We therefore use CEM in a receding-horizon manner, with a horizon of 4 as a prediction of one second and an execution of one step, as justified later.

Fine-tuning with a frozen encoder. To probe how far the predictor can be improved where the planner actually looks, we fine-tune it on on-policy data, in the spirit of data aggregation for model learning (Ross & Bagnell, 2012). We collect the candidates CEM evaluates and roll each of them out in the simulator from the same start state. Under naive joint training of the encoder and the predictor, the improvement in the accuracy of the predictor is achieved spuriously, as the encoder and SIGReg distort the latent space rather than the predictor learning the dynamics. This is fatal for our task, which demands a high prediction accuracy for the ball. We therefore freeze the encoder after representation learning and also the predictor's MLP projection head. Updating this head induces a shift in the mean of the predicted latent distribution, which hinders learning of the dynamics by the predictor. We therefore fine-tune the remaining predictor parameters on these rollouts while keeping both the encoder and the projection head fixed. The effect of freezing the encoder is evaluated in Appendix C. On the candidates the planner samples we try two objectives. (i) The *distance loss* is the prediction term of Equation (1) left as it is. (ii) The *ranking loss* replaces that term by the order of those candidates, as the pairwise logistic loss $\mathrm{softplus}((y(a_i) - y(a_j))/\tau)$ of Burges et al. (2005), where $y(a)$ is the cost the planner assigns to a candidate a, the pair is ordered by the true state distance to the goal the candidate reaches when it is rolled out, and τ is the standard deviation of y over the candidates of that start point.

5 Experiments

The environment is built in MuJoCo (Todorov et al., 2012) and parallelised on GPU with MJWarp (Google DeepMind & NVIDIA, 2025). At planning time the goal is given as an observation drawn from a fixed set of goal states and embedded by the same encoder. We compare against the original LeWM without the force head (Maes et al., 2026). Reinforcement learning, which needs a reward and online interaction, is outside our comparison; the expert serves as a reference.

Simulation and dataset. As the expert we ran the model-based MPC agent of Schneider et al. (2022), which observes the state of the arm and of the ball, under MJWarp and recorded its episodes with exploration noise. Camera images are rendered at 4 Hz and 224×224 pixels, and the force labels are read from the contact solver at 40 Hz: f_s is the mean contact force between the stick and the ball over substep s, projected onto the board plane, and c_s indicates whether the solver reports a contact within the substep (Appendix D). We divide the force by the ball mass, which is constant, so f_s is the contact force up to a constant factor. Each episode is 150 control frames long (37.5 s). The dataset contains 26,133 episodes (3,919,950 control frames), of which 83.2% are expert rollouts with exploration noise and 16.8% are scripted trajectories. The scripted part is mixed in because, with demonstrations from a single policy, the latent of the next observation becomes predictable from the current observation alone and the training objective then no longer forces the predictor to condition on the action. For the probing experiments, we also train the models on a small dataset of 2,613 episodes, one tenth of the full dataset, drawn from it uniformly at random.

Metrics. We evaluate with the ball probe error, the ball readout score and the success rate, on two tasks that differ in difficulty, the centre init (easy) and the bottom init (difficult), which start every episode at the centre of the board and at the bottom of it respectively. Unless otherwise noted, every metric is averaged over models trained with three seeds. The *ball probe error* measures what the latent keeps about the ball. A probe, an MLP described in Appendix D, is trained on the frozen embedding to read the position of the ball, with disjoint held-out episodes for training, selection

and scoring, and is averaged over three probe seeds for each model. We report $1 - R^2$ on the test episodes, so lower is better. The *ball readout score* applies the same probe to the latent the predictor reaches after h steps and normalises it as $(R^2 - R_0^2)/(1 - R_0^2)$, where R_0^2 is that of the persistence readout which repeats the ball position of the conditioning frame. The *success rate* is the share of 100 planning episodes per cell, run from a fixed set of goals and start states, in which the ball reaches the goal zone. An episode succeeds if the ball enters the goal zone at any timestep.

5.A Qualitative comparison

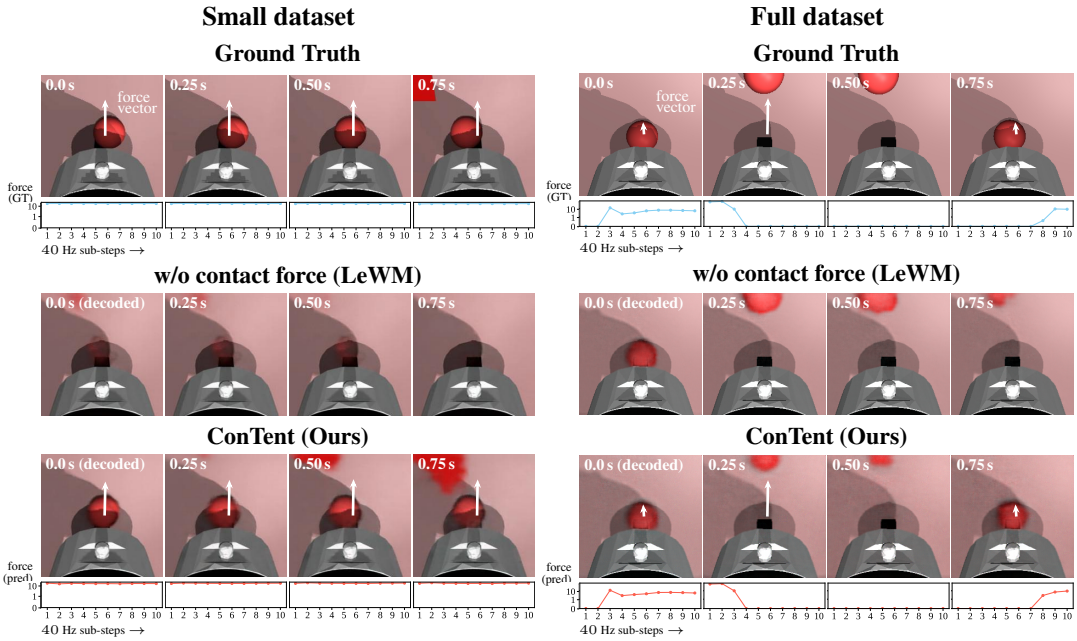

Figure 2: Decoded wrist-view rollouts on the small (left) and the full dataset (right). The first column decodes the encoded observation, and the others decode latents rolled out by the predictor. Arrows show the contact force (ground truth, top; predicted, bottom), and the strips show its magnitude over the ten 40 Hz substeps on a symmetric-log axis.

We observed distinct effects on the discarding of information and on its retention through contact-force prediction in the ball-pushing task. In Figure 2 a decoder is post-trained on the models trained on the small dataset and on the full dataset to evaluate the information the latent keeps. That decoder maps the 192-dimensional latent through a linear layer to a $512 \times 7 \times 7$ feature and five upsampling convolutional blocks that output the three camera views at 224×224, and it is trained by a pixel mean squared error on held-out episodes with the encoder and the predictor frozen. On the small dataset the ball disappears in the decode of LeWM, without contact-force prediction, whereas the model with contact-force prediction decodes the ball. On the full dataset the decodes of both models keep the ball, but once the predictor is rolled out, LeWM without contact force cannot predict the dynamics of the ball. Next, this retention of the ball and this prediction of its dynamics also affect the downstream task with CEM. On the bottom-init task LeWM without contact force reaches a success rate of only 2%, whereas the proposed model reaches 52% by the prediction term of the contact force alone. Online CEM fine-tuning further improves it to 77%, which Section 5.B takes up. As Figure 3 shows, LeWM does not minimise the latent goal distance or the true state distance through CEM, whereas the proposed model searches trajectories in the direction that minimises both.

5.B Quantitative comparison

Figure 4 is an evaluation of the effect of contact-force prediction over the whole training, which backs up the qualitative observation. The ball retention of ConTent lags behind that of LeWM from 0 to 40,000 steps, and after that it is consistently above that of LeWM. This trend agrees with the trend of the distance the ball could approach under CEM, and this distance is the obvious factor

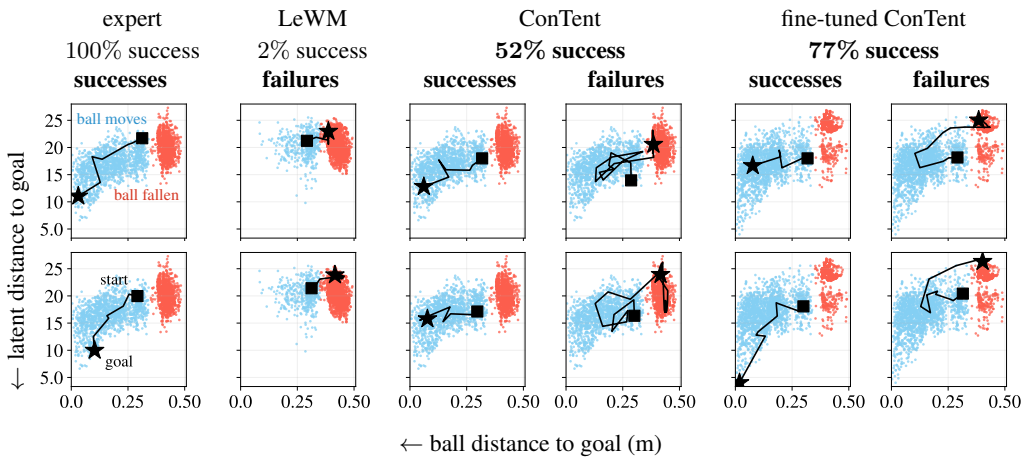

Figure 3: Closed-loop CEM planning from the bottom start, one episode per panel and two episodes per column. The horizontal axis is the distance of the ball to the goal in metres and the vertical axis is the distance of the encoded observation to the encoded goal in the 192-dimensional latent space; the square marks the first and the star the last frame of the episode. The background is the scatter of all evaluation frames of the model that produced the panel, and the expert column is embedded with the ConTent encoder. Red points are frames in which the ball has fallen off the stick and blue points frames in which it has not, and the failed episodes end in the red region.

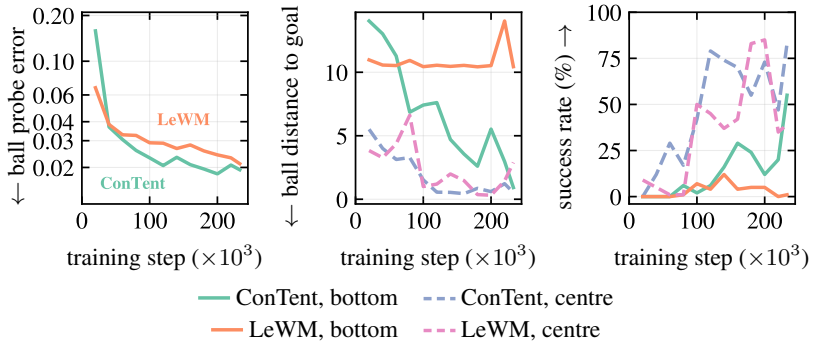

Figure 4: Ball probe error, ball distance to goal and success rate along training, for the two models from the centre start and from the bottom start. The ball distance to goal is the median over the episodes of the smallest distance the ball reaches from the goal, in units of the radius of the goal zone, so that an episode succeeds below 1. The checkpoints are taken every 20,000 steps and evaluated with planning at $h = 4$.

behind the rise of the success rate. We consider this to be because the adaptation to the additional force prediction term takes tens of thousands of steps, but finally it is beneficial for the retention of the ball state in the encoding and for the formation of a predictable latent space.

The effect of the ball information is tightly coupled with the prediction horizon. The model difference in ball retention in Figure 4 does matter especially when the predictor is rolled out, and it then weighs heavily on the success rate of the task. Figure 5 shows the effect of the planning horizon on the prediction accuracy of the ball state and on the CEM success rate. ConTent improves the prediction accuracy from horizon 1 to 6 and loses significance beyond horizon 6. This corresponds to the result that the CEM success rate peaks at horizons 3 and 4 and loses significance beyond horizon 6.

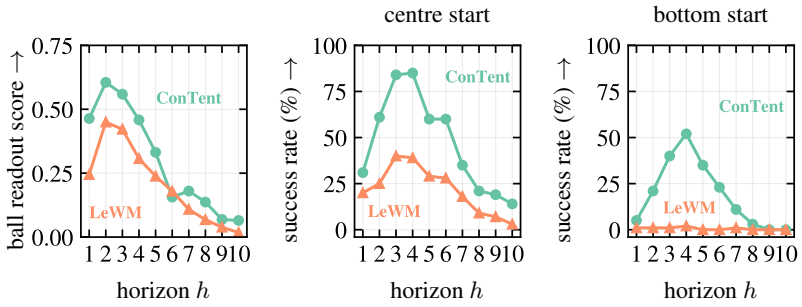

Figure 5: Ball readout score and success rate against the planning horizon h, from the centre start and from the bottom start, with the checkpoints of the training seeds, the goals and every other planner setting held fixed.

Fine-tuning alone is not enough. Figure 6 shows the result of training the models further, after the training on the offline data, with the on-policy data obtained by CEM. The encoder and the projection head are frozen throughout this fine-tuning, and the reason for freezing them is written in Appendix C. One fine-tuning epoch collects the 9,000 action sequences the planner evaluates in that round, which are 300 candidates over 30 iterations, and consumes 8,960 of them in 140 gradient steps of 64 rollouts each. On the easy task (centre init) no significant improvement is seen. On the difficult task (bottom init) an improvement is seen only for the fine-tuning that uses the ranking loss and ConTent. A plausible explanation is that a latent space that has not discarded the information the task needs is what makes further improvement by dataset aggregation effective. We discuss the effect of the two losses in the Discussion.

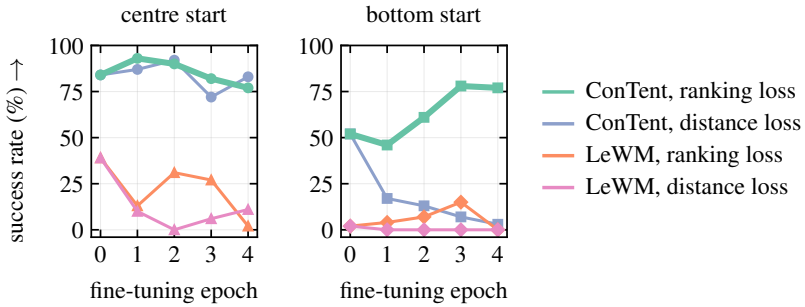

Figure 6: Success rate along the fine-tuning epochs, planned at $h = 4$, from the centre start and from the bottom start. Epoch 0 is the model before fine-tuning.

6 Discussion

Through analysis and experiments, this study shows that a latent world model trained without contact information tends to discard object information, and that incorporating contact forces improves the success rate of dynamic object manipulation by anchoring this information in the latent space. Otherwise, the loss of this information degrades the prediction of the dynamics as the horizon grows, lowering the success rate of dynamic tasks. By exploiting contact forces only as an auxiliary training target, i.e. predicting the substep contact-force trajectory of each transition from the predictor feature with a lightweight force head, we raise the CEM success rate on our task from 2% to 52%, surpassing the LeWM baseline. Fine-tuning by dataset aggregation fails in many setups, but is improved by the ranking loss. A plausible explanation is that, for the predictor, accurately predicting the absolute latent state (general-purpose information) and predicting the relative one (task-specific ranking information) are in a trade-off, and that the latter is the more efficient of the two for the task. This suggests that, although a world-model policy can be learned from offline data, on-policy learning based on a reward, e.g. reinforcement learning, could be effective, and it will be a key direction for future research.

AI USE STATEMENT

We used large language model (LLM) assistants in the following ways. **Literature search:** LLMs were used to find and summarise related work; every cited reference was checked by the authors against the original source. **Implementation and execution:** LLMs assisted in writing code for the simulation environment, the data-collection pipeline, training and evaluation scripts, and plotting, and in running experiments. All code was reviewed by the authors, and all reported numbers come from runs executed and verified by the authors; no results were generated by an LLM. **Drafting and writing:** LLMs drafted parts of the text from the authors' notes and results, and were used to polish wording throughout; the authors revised all text. **Mathematics:** LLMs assisted in formulating and proving the statements in Section 3 and Appendices A–B; the authors checked every step of every proof. The research question, the method, and the experimental design are due to the authors. The authors take full responsibility for the content of this paper, including any errors.

ETHICS STATEMENT

This work is carried out entirely in simulation and involves no human subjects and no personal data. The authors declare no conflict of interest.

REPRODUCIBILITY STATEMENT

Section 5 and Appendix D give the simulator, the dataset and the training settings, the appendix gives the assumptions and the complete proofs, and the code will be released.

REFERENCES

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

Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using gradient descent. In *International Conference on Machine Learning (ICML)*, 2005.

Cheng Chi, Zhenjia Xu, Siyuan Feng, Eric Cousineau, Yilun Du, Benjamin Burchfiel, Russ Tedrake, and Shuran Song. Diffusion Policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, 44(10–11):1684–1704, 2025. doi: 10.1177/02783649241273668.

Weijia Fan and Daqiang Guo. PredTac: Learning contact-rich manipulation with predicted touch. *arXiv preprint arXiv:2609.15198*, 2026.

Google DeepMind and NVIDIA. MuJoCo Warp. https://github.com/google-deepmind/mujoco_warp, 2025.

Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022.

Alexander S. Kechris. *Classical Descriptive Set Theory*, volume 156 of *Graduate Texts in Mathematics*. Springer, New York, 1995.

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

Michelle A. Lee, Yuke Zhu, Krishnan Srinivasan, Parth Shah, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Self-supervised learning of multimodal representations for contact-rich tasks. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

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.

Lorenzo Mur-Labadia, Matthew Muckley, Amir Bar, Mido Assran, Koustuv Sinha, Mike Rabbat, Yann LeCun, Nicolas Ballas, and Adrien Bardes. V-JEPA 2.1: Unlocking dense features in video self-supervised learning. *arXiv preprint arXiv:2603.14482*, 2026.

Yaru Niu et al. Learning versatile humanoid manipulation with touch dreaming. *arXiv preprint arXiv:2604.13015*, 2026.

Joshua Robinson, Li Sun, Ke Yu, Kayhan Batmanghelich, Stefanie Jegelka, and Suvrit Sra. Can contrastive learning avoid shortcut solutions? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

Stéphane Ross and J. Andrew Bagnell. Agnostic system identification for model-based reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2012.

Tim Schneider, Boris Belousov, Georgia Chalvatzaki, Diego Romeres, Devesh K. Jha, and Jan Peters. Active exploration for robotic manipulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022. Project page: https://sites.google.com/view/aerm.

Vlad Sobal, Wancong Zhang, Kyunghyun Cho, Randall Balestriero, Tim G. J. Rudner, and Yann LeCun. Learning from reward-free offline data: A case for planning with latent dynamics models. *arXiv preprint arXiv:2502.14819*, 2025.

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 (IROS)*, 2012.

Yihao Xue, Siddharth Joshi, Eric Gan, Pin-Yu Chen, and Baharan Mirzasoleiman. Which features are learnt by contrastive learning? On the role of simplicity bias in class collapse and feature suppression. In *International Conference on Machine Learning (ICML)*, 2023.

Xi Zeng, Haojie Ren, and Ziying Song. PhyLatent: Learning dynamics-relevant representations for JEPA world models. *arXiv preprint arXiv:2608.05720*, 2026.

Zhiyuan Zhang, Pokuang Zhou, Kaidi Zhang, Adeesh Desai, Temitope Amosa, Davood Soleymanzadeh, Jiuzhou Lei, Yuhao Zhou, Minghui Zheng, and Yu She. ContactWorld: What representations matter for vision-tactile latent world models in contact-rich manipulation. *arXiv preprint arXiv:2606.13877*, 2026.

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

Throughout this appendix x is the world state and x' its successor, h is the composed map from states to embeddings, $z = h(x)$ the embedding of the first frame, τ the downstream target and τ^* the auxiliary target. Assumption 1(i) is invoked only where stated. All random vectors are assumed square integrable, so every conditional expectation below is well defined.

A.A THREE FACTS USED THROUGHOUT

For square-integrable random vectors v and ζ write $\mathcal{V}(v \mid \zeta) := \mathbb{E}\|v - \mathbb{E}[v \mid \zeta]\|^2 = \mathbb{E}[\operatorname{tr}\operatorname{Cov}(v \mid \zeta)]$. Since $\mathbb{E}[v \mid \zeta]$ is a function of ζ and, by the Pythagorean identity $\mathbb{E}\|v - f(\zeta)\|^2 = \mathcal{V}(v \mid \zeta) + \mathbb{E}\|\mathbb{E}[v \mid \zeta] - f(\zeta)\|^2$, is the best predictor of v among functions of ζ,

- (F1) $\mathcal{V}(v \mid \zeta) = \min_f \mathbb{E}\|v - f(\zeta)\|^2$, and $\mathcal{V}(v \mid \zeta) = 0$ if and only if $v = f(\zeta)$ a.s. for some function f (namely $f = \mathbb{E}[v \mid \cdot]$);
- (F2) if $u = g(\zeta)$ a.s. then $\mathcal{V}(v \mid \zeta) \le \mathcal{V}(v \mid u)$, because every decoder $f(u) = f(g(\zeta))$ of u is a decoder of ζ;
- (F3) if a is independent of x and u, ζ are functions of x, then $\mathcal{V}(u \mid \zeta, a) = \mathcal{V}(u \mid \zeta)$, because the conditional law of u given (ζ, a) is its conditional law given ζ; the same holds when the policy reads only $x_{\setminus\Delta}$ and $x_{\setminus\Delta}$ is a function of ζ.

Profiling the predictor out of Equation (1) changes neither the value nor the minimisers,

$$\mathcal{L}(h) := \min_{\mathsf{pred}} \mathcal{L}(h, \mathsf{pred}) = \mathbb{E}\big[\operatorname{tr}\operatorname{Cov}\big(h(x') \mid h(x), a\big)\big] + \lambda S\big(\mathsf{law}\, h(x)\big), \tag{8}$$

the minimum being attained by the conditional mean, by (F1), so the prediction term sees the encoder only through the information $h(x)$ carries.

In this notation $\mathsf{err}_\Delta(h) = \mathcal{V}(x_\Delta \mid h(x))$, the prediction term of Equation (8) is $\mathcal{V}(h(x') \mid h(x), a)$, $\mathcal{L}^*(h) = \mathcal{V}(\tau^* \mid h(x), a)$ and $\mathsf{risk} = \mathcal{V}(x_\Delta \mid x_{\setminus\Delta}, a, \tau^*)$.

A.B PROOF OF LEMMA 1

Lemma 1: Risk decomposition

Let $\tau : \mathbb{R}^n \to \mathbb{R}$ be a downstream target, let $\widehat{\tau}_{\mathcal{D}}$ be any probe trained on a finite sample $\mathcal{D} = \{(h(x_j), \tau(x_j))\}_{j=1}^N$ drawn independently of the test point, let $z = h(x)$, and let $b(z) := \mathbb{E}[\tau(x) \mid z] - \mathbb{E}_{\mathcal{D}}[\widehat{\tau}_{\mathcal{D}}(z)]$. Then

$$\mathbb{E}\big[(\tau(x) - \widehat{\tau}_{\mathcal{D}}(z))^2\big] = \underbrace{\mathbb{E}\big[\operatorname{Var}(\tau(x) \mid z)\big]}_{\text{(i) Bayes risk}} + \underbrace{\mathbb{E}\big[b(z)^2\big]}_{\text{(ii) squared bias}} + \underbrace{\mathbb{E}\big[\operatorname{Var}_{\mathcal{D}}(\widehat{\tau}_{\mathcal{D}}(z))\big]}_{\text{(iii) variance}}. \tag{9}$$

Balestriero & LeCun (2025) show that anisotropy of the embedding inflates the bias and the variance of linear probes and that the isotropic Gaussian minimises the integrated squared bias of k-NN and kernel probes, which concerns terms (ii) and (iii). Term (i) is a floor for every probe and is left untouched.

Proof of Lemma 1. Write $\mu(z) := \mathbb{E}[\tau(x) \mid z]$ and $\bar{\tau}(z) := \mathbb{E}_{\mathcal{D}}[\widehat{\tau}_{\mathcal{D}}(z)]$, so that $b(z) = \mu(z) - \bar{\tau}(z)$, and split the error into $E_1 := \tau(x) - \mu(z)$, $b(z)$ and $E_3 := \bar{\tau}(z) - \widehat{\tau}_{\mathcal{D}}(z)$. E_1 is centred given z, E_3 is centred over $\mathcal{D}$ for every fixed z, and $\mathcal{D}$ is drawn independently of the test point, so conditioning on z kills all three cross terms. Squaring and taking expectations leave the three squares, which are the three terms of Equation (9). □

A.C ONE-STEP RETENTION

Call τ^* Δ-*retaining* if $\mathsf{risk} = 0$, which holds, for instance, if $x_\Delta \mapsto \mathsf{aux}(x_{\setminus\Delta}, x_\Delta, a)$ is invertible for every fixed $(x_{\setminus\Delta}, a)$.

Proposition 1: One-step retention

Assume Assumption 1(iv), which for a single transition reads $a \perp x$.

(i) If $\mathcal{L}^*(h) = 0$ and $\mathsf{err}_{\setminus\Delta}(h) = 0$, then $\mathsf{err}_\Delta(h) \le \mathsf{risk}$.

(ii) Conversely, if $\tau^* = \mathsf{aux}(x)$ and $m \ge n - n_\Delta + q$, the encoder $h_{\tau^*}(x) := (x_{\setminus\Delta}, \mathsf{aux}(x), 0, \dots, 0)$ has $\mathcal{L}^*(h_{\tau^*}) = 0$, $\mathsf{err}_{\setminus\Delta}(h_{\tau^*}) = 0$ and $\mathsf{err}_\Delta(h_{\tau^*}) = \mathsf{risk}$.

Hence the auxiliary term alone forces err_Δ to zero exactly when the target is Δ-retaining, and never below risk.

Proof of Proposition 1. (i) By (F1), $\mathcal{L}^*(h) = 0$ gives $\tau^* = \mathsf{aux}_{\psi^*}(z, a)$ a.s. and $\mathsf{err}_{\setminus\Delta}(h) = 0$ gives $x_{\setminus\Delta} = \eta(z)$ a.s. For any readout φ, $\varphi(x_{\setminus\Delta}, a, \tau^*) = \varphi(\eta(z), a, \mathsf{aux}_{\psi^*}(z, a))$ a.s. is a readout of (z, a), so $\mathcal{V}(x_\Delta \mid z, a) \le \mathbb{E}\|x_\Delta - \varphi(x_{\setminus\Delta}, a, \tau^*)\|^2$; minimising over φ gives $\mathcal{V}(x_\Delta \mid z, a) \le \mathsf{risk}$, and (F3), by Assumption 1(iv), turns the left-hand side into $\mathsf{err}_\Delta(h)$.

(ii) $\tau^* = \mathsf{aux}(x)$ and $x_{\setminus\Delta}$ are coordinates of $h_{\tau^*}(x)$, so both losses vanish, and $h_{\tau^*}(x)$ and $(x_{\setminus\Delta}, \mathsf{aux}(x))$ are functions of each other, so by (F2) twice $\mathsf{err}_\Delta(h_{\tau^*}) = \mathcal{V}(x_\Delta \mid x_{\setminus\Delta}, \mathsf{aux}(x))$, which equals $\mathcal{V}(x_\Delta \mid x_{\setminus\Delta}, a, \mathsf{aux}(x)) = \mathsf{risk}$ by (F3) under Assumption 1(iv). □

Proof of the two readings. If $\mathsf{err}_\Delta(h) = 0$ then $x_\Delta = \xi(z)$ a.s. by (F1), so $\tau(x_\Delta) = \tau(\xi(z))$ and $\mathrm{Var}(\tau(x_\Delta) \mid z) = 0$ for every τ; conversely $\mathsf{err}_\Delta(h) = \sum_{i\in\Delta} \mathbb{E}[\mathrm{Var}(x_i \mid z)]$, so term (i) vanishing for $\tau = x_i$, $i \in \Delta$, forces $\mathsf{err}_\Delta(h) = 0$. For a τ^* that depends on $x_{\setminus\Delta}$ alone, $\mathsf{risk} = \mathcal{V}(x_\Delta \mid x_{\setminus\Delta}, a) = \mathcal{V}(x_\Delta \mid x_{\setminus\Delta})$ by (F3) under Assumption 1(iv), which is $\mathrm{tr}\,\mathrm{Cov}(x_\Delta) = n_\Delta$ under Assumption 1(i) since $x_\Delta \perp x_{\setminus\Delta}$. □

A.D Proof of Theorem 1

Proof. Throughout, Assumption 1(iv) and (F3) give $\mathcal{V}(x_{\Delta,t} \mid z_t, a_{t:T}) = \mathcal{V}(x_{\Delta,t} \mid z_t)$, which is $\mathsf{err}_\Delta(h)$ by Assumption 1(ii); likewise $\mathcal{V}(x_{\setminus\Delta,t} \mid z_t, a_{t:T}) = \mathsf{err}_{\setminus\Delta}(h)$.

(i) Let the prediction term of Equation (8), $\mathcal{L}^*(h)$ and $\mathsf{err}_{\setminus\Delta}(h)$ vanish. By (F1) there are pred, aux_{ψ^*} and η with $h(x') = \mathsf{pred}(h(x), a)$, $\tau^* = \mathsf{aux}_{\psi^*}(h(x), a)$ and $x_{\setminus\Delta} = \eta(h(x))$ a.s. for one transition, and by Assumption 1(ii) the same holds at every time s, so by Assumption 1(iii) $z_{s+1} = \mathsf{pred}(z_s, a_s)$ and $\tau^*_s = \mathsf{aux}_{\psi^*}(z_s, a_s)$ a.s. Induction on k gives $z_{t+k} = \mathsf{pred}_k(z_t, a_{t:t+k-1})$ with $\mathsf{pred}_0(z) = z$ and $\mathsf{pred}_{k+1}(z, a_{t:t+k}) = \mathsf{pred}(\mathsf{pred}_k(z, a_{t:t+k-1}), a_{t+k})$, hence $\tau^*_{t+k} = \mathsf{aux}_{\psi^*}(\mathsf{pred}_k(z_t, a_{t:t+k-1}), a_{t+k})$ and $x_{\setminus\Delta,t+k} = \eta(\mathsf{pred}_k(z_t, a_{t:t+k-1}))$ are functions of $(z_t, a_{t:T})$. For any readout φ, $\varphi(x_{\setminus\Delta,t:T}, a_{t:T}, \tau^*_{t:T})$ is therefore a.s. a readout of $(z_t, a_{t:T})$, so $\mathcal{V}(x_{\Delta,t} \mid z_t, a_{t:T}) \le \mathbb{E}\|x_{\Delta,t} - \varphi(x_{\setminus\Delta,t:T}, a_{t:T}, \tau^*_{t:T})\|^2$, and minimising over φ gives the bound by $\mathsf{risk}^{(T)}_t$.

(ii) Let $\tilde{x}_t$ be drawn from the conditional law of x_t given z_t, conditionally independent of x_t; by Assumption 1(iv) this is also its conditional law given $(z_t, a_{t:T})$. Generate the copy segment $(\tilde{x}_{t:T}, a_{t:T})$ from $\tilde{x}_t$ by Assumption 1(iii) with the same actions. Then, given $(z_t, a_{t:T})$, any function v of the segment and its copy $\tilde{v}$ are conditionally i.i.d., and expanding the square,

$$\mathbb{E}\big[\|v - \tilde{v}\|^2 \,\big|\, z_t, a_{t:T}\big] = 2\,\mathbb{E}[\|v\|^2 \mid z_t, a_{t:T}] - 2\,\|\mathbb{E}[v \mid z_t, a_{t:T}]\|^2 = 2\,\mathrm{tr}\,\mathrm{Cov}(v \mid z_t, a_{t:T}). \quad (10)$$

Let E be the event that both segments lie in $\mathcal{S}_{t:T}$, so $\mathbb{P}(E^c) \le 2\delta$, and split $\mathbb{E}[\|x_{\Delta,t} - \tilde{x}_{\Delta,t}\|^2 \mid z_t, a_{t:T}]$ into the parts on E and on E^c.

On E, square Equation (4), use $(u+v)^2 \le 2u^2 + 2v^2$ and drop the indicator on the right, $\alpha^2\|x_{\Delta,t} - \tilde{x}_{\Delta,t}\|^2 \mathbf{1}_E \le 2\|\tau^*_{t:T} - \tilde{\tau}^*_{t:T}\|^2 + 2\beta^2\|x_{\setminus\Delta,t} - \tilde{x}_{\setminus\Delta,t}\|^2$. Taking conditional expectations and using Equation (10) for each τ^*_{t+k} and for $x_{\setminus\Delta,t}$, then expectations, we find that the right-hand side has expectation $4\mathcal{L}^{*(T)}_t(h) + 4\beta^2\mathsf{err}_{\setminus\Delta}(h)$, because $\mathbb{E}[\mathrm{tr}\,\mathrm{Cov}(\tau^*_{t+k} \mid z_t, a_{t:T})]$ is the k-th term of Equation (3) by (F1) and $\mathbb{E}[\mathrm{tr}\,\mathrm{Cov}(x_{\setminus\Delta,t} \mid z_t, a_{t:T})] = \mathsf{err}_{\setminus\Delta}(h)$.

On E^c, Cauchy–Schwarz gives $\mathbb{E}[\|x_{\Delta,t} - \tilde{x}_{\Delta,t}\|^2 \mathbf{1}_{E^c}] \le (\mathbb{E}\|x_{\Delta,t} - \tilde{x}_{\Delta,t}\|^4)^{1/2}\,\mathbb{P}(E^c)^{1/2} \le (16\,\mathbb{E}\|x_\Delta\|^4)^{1/2}(2\delta)^{1/2}$, using $\|u - v\|^4 \le 8(\|u\|^4 + \|v\|^4)$ and $\mathsf{law}\,\tilde{x}_{\Delta,t} = \mathsf{law}\,x_{\Delta,t} = \mathsf{law}\,x_\Delta$.

Adding the two parts, we see that the left-hand side has expectation $2\mathsf{err}_\Delta(h)$ by Equation (10); dividing by 2 gives Equation (5). Under Assumption 1(i) the fourth moment is

$$\mathbb{E}\|x_\Delta\|^4 = \sum_{i\in\Delta}\mathbb{E}[x_i^4] + \sum_{i\neq j\in\Delta}\mathbb{E}[x_i^2]\,\mathbb{E}[x_j^2] = n_\Delta\big(n_\Delta - 1 + \mu_4\big), \tag{11}$$

which is $n_\Delta(n_\Delta+2)$ in the Gaussian case $\mu_4 = 3$. □

Remark 1: Closed-loop data

Without Assumption 1(iv) every step above goes through with $\mathcal{V}(x_{\Delta,t} \mid z_t, a_{t:T})$ in place of $\mathsf{err}_\Delta(h)$, and the two can differ, because the actions then carry information about the object; if the policy reads only $x_{\setminus\Delta}$ and $\mathsf{err}_{\setminus\Delta}(h) = 0$, (F3) still applies and the conclusions hold as stated. Two further caveats apply. With a predictor that reads the history $z_{1:t}$ the same argument forces retention in the history, not in z_t alone. And whether the minimiser of the full objective reaches zero one-step loss is a separate question, answered for unrestricted measurable encoders by Proposition 2.

A.E Proof of Corollary 1

Proof. By Theorem 1(i), $\mathsf{err}_\Delta(h) \le \mathsf{risk}_t^{(T)}$, so it suffices to exhibit one readout of the triple $(x_{\setminus\Delta,t:T}, a_{t:T}, \tau^*_{t:T})$ whose error equals the right-hand side of Equation (6). Write $\kappa_t := \min\{k \ge 0 : (x_{t+k}, a_{t+k}) \in \Gamma\}$ for the time to the next contact, so that $E_t = \{\kappa_t \le T - t\}$. By condition (i) of Corollary 1 each indicator $\mathbf{1}\{(x_{t+k}, a_{t+k}) \in \Gamma\} = \gamma(\tau^*_{t+k})$, hence κ_t and the event E_t are functions of $\tau^*_{t:T}$. On $\{\kappa_t = k\}$ with $k \le T - t$, the transition at time $t + k$ is in Γ, so by condition (ii) $x_{\Delta,t+k} = \varphi(x_{\setminus\Delta,t+k}, a_{t+k}, \tau^*_{t+k})$, and the transitions $t, \dots, t + k - 1$ are off Γ, so by condition (iii) $x_{\Delta,t} = \mathsf{back}_k(x_{\Delta,t+k}, x_{\setminus\Delta,t:t+k}, a_{t:t+k-1})$, where $\mathsf{back}_0(u,\cdot) = u$. Define the readout

$$\varphi^\star := \sum_{k=0}^{T-t}\mathbf{1}\{\kappa_t = k\}\,\mathsf{back}_k\Big(\varphi\big(x_{\setminus\Delta,t+k},\, a_{t+k},\, \tau^*_{t+k}\big),\, x_{\setminus\Delta,t:t+k},\, a_{t:t+k-1}\Big), \tag{12}$$

which equals $x_{\Delta,t}$ on E_t and 0 on E_t^c. Its error is $\mathbb{E}\|x_{\Delta,t} - \varphi^\star\|^2 = \mathbb{E}[\|x_{\Delta,t}\|^2\mathbf{1}_{E_t^c}]$, which is Equation (6). The bound $\delta_t^{(T)}B^2$ is immediate, and under Assumption 1(i) Cauchy–Schwarz gives $\mathbb{E}[\|x_\Delta\|^2\mathbf{1}_{E_t^c}] \le (\mathbb{E}\|x_\Delta\|^4)^{1/2}\,\mathbb{P}(E_t^c)^{1/2}$ with Equation (11). □

A.F The multi-step loss through a Lipschitz predictor

Fix a predictor pair $(\mathsf{pred}, \mathsf{aux}_\psi)$ with one-step errors $\varepsilon_{\mathrm{p}} := \mathbb{E}\|h(x') - \mathsf{pred}(z,a)\|^2$ and $\varepsilon_{\mathrm{aux}} := \mathbb{E}\|\tau^* - \mathsf{aux}_\psi(z,a)\|^2$, Lipschitz in the latent uniformly in the action with constants L_{pred} and L_ψ. Put $\widehat{z}_{t+k} := \mathsf{pred}_k(z_t, a_{t:t+k-1})$ with pred_k as in the proof of Theorem 1, and $\widehat{\tau}^*_{t+k} := \mathsf{aux}_\psi(\widehat{z}_{t+k}, a_{t+k})$.

Lemma 2: Error compounding

Under Assumption 1(ii) and (iii), $\mathbb{E}\|\widehat{z}_{t+k} - z_{t+k}\|^2 \le 2\varepsilon_{\mathrm{p}}\Sigma_k$, where $\Sigma_k := \sum_{j=0}^{k-1}(2L^2_{\mathsf{pred}})^j$, and

$$\mathcal{L}_t^{*(T)}(h) \le \sum_{k=0}^{T-t}\mathbb{E}\big\|\tau^*_{t+k} - \widehat{\tau}^*_{t+k}\big\|^2 \le \varepsilon^{(T-t)}_{\mathrm{traj}} := \varepsilon_{\mathrm{aux}} + \sum_{k=1}^{T-t}\big(2\varepsilon_{\mathrm{aux}} + 4L_\psi^2\varepsilon_{\mathrm{p}}\Sigma_k\big), \tag{13}$$

so under the hypotheses of Theorem 1(ii), $\mathsf{err}_\Delta(h) \le \frac{2}{\alpha^2}\big(\varepsilon^{(T-t)}_{\mathrm{traj}} + \beta^2\mathsf{err}_{\setminus\Delta}(h)\big) + 2\sqrt{2\delta\,\mathbb{E}\|x_\Delta\|^4}$.

Proof. With $e_k := \mathbb{E}\|\widehat{z}_{t+k} - z_{t+k}\|^2$, $e_0 = 0$ and, by Assumption 1(iii), $\widehat{z}_{t+k+1} - z_{t+k+1} = \big(\mathsf{pred}(\widehat{z}_{t+k}, a_{t+k}) - \mathsf{pred}(z_{t+k}, a_{t+k})\big) + \big(\mathsf{pred}(z_{t+k}, a_{t+k}) - h(F(x_{t+k}, a_{t+k}))\big)$, so the inequality $\|u+v\|^2 \le 2\|u\|^2 + 2\|v\|^2$, the Lipschitz bound and Assumption 1(ii) give $e_{k+1} \le 2L^2_{\mathsf{pred}}e_k + 2\varepsilon_{\mathrm{p}}$,

whence the first claim. Similarly $\mathbb{E}\|\tau^*_{t+k} - \widehat{\tau}^*_{t+k}\|^2 \le 2\varepsilon_{\text{aux}} + 2L_\psi^2 e_k$ for $k \ge 1$ and $= \varepsilon_{\text{aux}}$ for $k = 0$. Since $\widehat{\tau}^*_{t+k}$ is a function of $(z_t, a_{t:T})$, (F1) gives the first inequality of Equation (13); summing gives the second, and substituting in Equation (5) the last claim. □

B Supplementary results

B.A The regulariser cannot see term (i)

Lemma 3: The regulariser cannot see term (i)

Let S depend on the encoder only through $\mathsf{law}\, h(x)$. If $\mathsf{law}\, h_1(x) = \mathsf{law}\, h_2(x)$, then $S(h_1) = S(h_2)$, whereas $\mathsf{err}_\Delta(h_1)$ and $\mathsf{err}_\Delta(h_2)$, and hence term (i) of Lemma 1, may differ. Concretely, under Assumption 1(i), if $n_\Delta \le r \le \min(m,\, n - n_\Delta)$, then taking $U \in \mathbb{R}^{m\times r}$ with orthonormal columns and $h_j(x) := Ux_{I_j}$ for two index sets of size r, one containing Δ and one disjoint from it, gives $\mathsf{law}\, h_1(x) = \mathsf{law}\, h_2(x)$ while $\mathsf{err}_\Delta(h_1) = 0$ and $\mathsf{err}_\Delta(h_2) = n_\Delta$. For the target $\tau(x) = x_i$ with $i \in \Delta$ the corresponding values of term (i) are 0 and 1.

The point of the lemma is that two encoders can agree on everything the regulariser can see and still differ in what they keep about the world. The witness is a pair of coordinate selections of the same size, one containing Δ and one avoiding it.

Proof of Lemma 3. The witness below is linear for convenience and not by necessity, since composing both members with one fixed measurable injection leaves the argument intact. The first claim is immediate, since S is a function of $\mathsf{law}\, h(x)$ and equal laws give equal values.

For the second claim, fix r with $n_\Delta \le r \le \min(m,\, n - n_\Delta)$, let $U \in \mathbb{R}^{m\times r}$ have orthonormal columns, write $h_I(x) := Ux_I$, and choose two index sets of size r,

$$I_1 \supseteq \Delta, \qquad I_2 \subseteq \{1, \dots, n\} \setminus \Delta, \tag{14}$$

which exist precisely under that constraint on r. Put $h_1 := h_{I_1}$ and $h_2 := h_{I_2}$. By Assumption 1(i) the coordinates are i.i.d., so x_{I_1} and x_{I_2} have the same law, hence so do their images under U, and $S(h_1) = S(h_2)$ because S depends only on the law.

What each encoder keeps. U has orthonormal columns and is therefore injective, so $h_1(x) = Ux_{I_1}$ determines x_{I_1}, and $\Delta \subseteq I_1$ gives x_Δ exactly,

$$\mathrm{Cov}(x_\Delta \mid h_1(x)) = 0, \qquad \text{so} \qquad \mathsf{err}_\Delta(h_1) = 0. \tag{15}$$

In the other direction h_2 is a function of $x_{\setminus\Delta}$ alone, and the coordinates of x are independent, so x_Δ is independent of $h_2(x)$ and

$$\mathrm{Cov}(x_\Delta \mid h_2(x)) = \mathrm{Cov}(x_\Delta) = I_{n_\Delta}, \qquad \text{so} \qquad \mathsf{err}_\Delta(h_2) = n_\Delta. \tag{16}$$

Term (i) for a concrete target. Take $\tau(x) = x_i$ for some $i \in \Delta$. Then $h_1(x)$ determines x_i, so $\mathrm{Var}(\tau(x) \mid h_1(x)) = 0$, while x_i is independent of $h_2(x)$, so $\mathrm{Var}(\tau(x) \mid h_2(x)) = \mathrm{Var}(x_i) = 1$. Term (i) of Lemma 1 is the expectation of these conditional variances, i.e. 0 and 1 respectively. The argument uses only Assumption 1(i), so it is independent of Theorem 2. □

In particular S is constant on $\{h : \mathsf{law}\, h(x) = \nu\}$, so on every such set the objective $\mathcal{L}(h)$ ranks encoders by the prediction term alone.

B.B An explicit family

Theorem 2: An explicit family, where the tie is broken by predictability

Assume Assumption 1(i) with transitions $p(x_i' \mid x_i, a)$ that may be stochastic; Assumption 1(iii) is not assumed, since under it every $c_i = 0$, all index sets tie, and Proposition 3 applies instead. Fix $r \le \min(m, n)$ and a matrix $U \in \mathbb{R}^{m\times r}$ with orthonormal columns. For an index set

$I \subseteq \{1, \dots, n\}$ with $|I| = r$, define the encoder $h_I(x) := Ux_I$, where x_I is the subvector of x indexed by I. Then

(i) $\mathsf{law}\, h_I(x)$ is the same for *every* such I. Write ν_r for it. In particular $S(h_I)$ takes the same value for all such I, for every S that depends on the encoder only through the embedding law. With a Gaussian marginal, $\nu_r = \mathcal{N}(0, UU^\top)$.

(ii) The prediction term of Equation (1) equals

$$\min_{\mathsf{pred}} \mathbb{E}\big[\|\mathsf{pred}(h_I(x), a) - h_I(x')\|^2\big] = \sum_{i \in I} c_i, \qquad c_i := \mathbb{E}\big[\mathrm{Var}(x'_i \mid x_i, a)\big], \tag{17}$$

where c_i is the irreducible error of predicting coordinate i at the next timestep from its current value.

(iii) Equation (17) is a sum of per-coordinate terms, so $\mathcal{L}(h_I)$ is minimised over $\{I : |I| = r\}$ exactly by the index sets collecting the r coordinates with the smallest c_i. If $r \le n - n_\Delta$ and every coordinate outside Δ is strictly more predictable than every coordinate inside it, i.e. $\max_{i \notin \Delta} c_i < \min_{i \in \Delta} c_i$, then every minimiser satisfies $I \cap \Delta = \emptyset$, and

$$\mathsf{err}_\Delta(h_I) = n_\Delta \quad (x_\Delta \text{ is discarded}), \qquad \mathsf{err}_{\setminus\Delta}(h_I) = n - n_\Delta - r. \tag{18}$$

Part (i) says the regulariser cannot distinguish the members of this family, part (ii) evaluates the prediction term on each member, and part (iii) combines them.

Proof of Theorem 2. (i) All members induce the same embedding law. By Assumption 1(i) the coordinates of x are i.i.d., so the subvector x_I has the same law for every index set I with $|I| = r$, and therefore so does its image $h_I(x) = Ux_I$. Write ν_r for that law. Since S depends on h_I only through $\mathsf{law}\, h_I(x)$, the value $S(h_I) = S(\nu_r)$ is the same for all admissible I. If the marginal is in addition Gaussian then $x_I \sim \mathcal{N}(0, I_r)$ and a linear image of a Gaussian is Gaussian, so $\nu_r = \mathcal{N}(0,\, UI_rU^\top) = \mathcal{N}(0, UU^\top)$.

(ii) Value of the prediction term. Decompose $\mathbb{R}^m$ into the range of U and its orthogonal complement and write a candidate predictor as $\mathsf{pred}(u, a) = U\omega(u, a) + \chi(u, a)$ with $\chi(u, a) \perp \mathrm{range}(U)$. The target $h_I(x') = Ux'_I$ lies in $\mathrm{range}(U)$, so by orthogonality

$$\|\mathsf{pred}(h_I(x), a) - h_I(x')\|^2 = \big\|U\big(\omega(h_I(x), a) - x'_I\big)\big\|^2 + \|\chi(h_I(x), a)\|^2 \tag{19}$$

$$= \|\omega(h_I(x), a) - x'_I\|^2 + \|\chi(h_I(x), a)\|^2, \tag{20}$$

where the last step uses $\|Uv\| = \|v\|$ for orthonormal columns. The second term is minimised by $\chi \equiv 0$. Since U is injective, conditioning on $(h_I(x), a)$ is the same as conditioning on (x_I, a), so the first term is minimised by the conditional mean, giving

$$\min_{\mathsf{pred}} \mathbb{E}\big[\|\mathsf{pred}(h_I(x), a) - h_I(x')\|^2\big] = \mathbb{E}\big[\|\mathbb{E}[x'_I \mid x_I, a] - x'_I\|^2\big] \tag{21}$$

$$= \sum_{i \in I} \mathbb{E}\big[\mathrm{Var}(x'_i \mid x_I, a)\big]. \tag{22}$$

By Assumption 1(i) the pairs (x_i, x'_i) are independent across i given the action, so conditioning on the whole subvector is the same as conditioning coordinatewise, $\mathrm{Var}(x'_i \mid x_I, a) = \mathrm{Var}(x'_i \mid x_i, a)$, and the sum is $\sum_{i \in I} c_i$ by the definition of c_i in Theorem 2. This is Equation (17).

(iii) Which index sets minimise the objective. By (i) the regulariser contributes the same constant $\lambda S(\nu_r)$ to $\mathcal{L}(h_I)$ for every admissible I, so minimising $\mathcal{L}$ over this family is minimising the prediction term alone. It is a sum over $i \in I$ of the per-coordinate term c_i, so minimising over index sets of a fixed size means collecting the r coordinates with the smallest c_i.

Under the stated separation every coordinate outside Δ beats every coordinate inside it, and since $r \le n - n_\Delta$ there are enough of them, so every minimiser satisfies $I \cap \Delta = \emptyset$. For such an I the map $x \mapsto Ux_I$ is a function of $x_{\setminus\Delta}$ only, and x_Δ is independent of $x_{\setminus\Delta}$. Hence x_Δ is independent of $h_I(x)$ and

$$\mathrm{Cov}(x_\Delta \mid h_I(x)) = \mathrm{Cov}(x_\Delta) = I_{n_\Delta}, \qquad \text{so} \qquad \mathsf{err}_\Delta(h_I) = n_\Delta. \tag{23}$$

Finally U is injective, so $h_I(x)$ pins down the r coordinates in I exactly and leaves the other $n - n_\Delta - r$ coordinates of $x_{\setminus\Delta}$ independent and identically distributed, giving $\mathsf{err}_{\setminus\Delta}(h_I) = n - n_\Delta - r$. □

The family $h_I = Ux_I$ is a witness and not a model of what a network computes, and the linear form costs nothing, since composing every member with one fixed measurable injection preserves part (i). On this family, discarding is a property of the optimum rather than a failure of optimisation; it relies on noisy transitions and on $r < n$, and for $m = n$ with Gaussian latents Klindt et al. (2026) show that every latent is recovered. This complements the empirical literature on feature suppression in self-supervised learning (Robinson et al., 2021; Xue et al., 2023) and is distinct from dimensional collapse (Jing et al., 2022). The costs c_i average over the actions the behaviour policy took, so predictability, and hence what the optimum keeps, moves with the data-collection policy, a further reason for the exploratory pretraining data Klindt et al. (2026) recommend.

B.C Population minimisers attain zero loss

Proposition 2: Population minimisers attain zero loss

Assume Assumption 1(iii), let p have a density, and let S vanish at $\mathcal{N}(0, I_m)$. Then, over measurable encoders $h : \mathbb{R}^n \to \mathbb{R}^m$ and for every $m \geq 1$ and every $w > 0$, the minimum of $\mathcal{L}(h) + w\mathcal{L}^*(h)$ is 0, and every minimiser has a zero prediction term, $\mathcal{L}^*(h) = 0$ and $S(h) = 0$.

Proof. By the isomorphism theorem for measures (Kechris, 1995, Thm. 17.41) there are Borel bijections $f_1 : \mathbb{R}^n \to [0,1]$ carrying p to Lebesgue measure and $f_2 : \mathbb{R}^m \to [0,1]$ carrying $\mathcal{N}(0, I_m)$ to Lebesgue measure; $h_\star := f_2^{-1} \circ f_1$ is a bijection with $\mathsf{law}\, h_\star(x) = \mathcal{N}(0, I_m)$, so $S(h_\star) = 0$. With $\mathsf{pred}(z, a) := h_\star(F(h_\star^{-1}(z), a))$ and $\mathsf{aux}_\star(z, a) := \mathsf{aux}(h_\star^{-1}(z), a)$, Assumption 1(iii) gives $h_\star(x') = \mathsf{pred}(h_\star(x), a)$ and $\tau^* = \mathsf{aux}_\star(h_\star(x), a)$, so the prediction term and $\mathcal{L}^*(h_\star)$ vanish. The three terms are nonnegative and $\lambda, w > 0$, so the minimum is 0 and every minimiser has each term equal to 0. □

This is the one non-elementary step of the analysis, and it is a statement about the population objective over all measurable encoders, the setting of Section 3. The encoder $h_\star$ is not continuous, so nothing is claimed for a continuous network with $m < n$.

B.D The base objective is indifferent to the object

Proposition 3: The base objective is indifferent to the object

Assume Assumption 1(iii), let p have a density and let S vanish at $\mathcal{N}(0, I_m)$. Assume further that the object does not move the remaining coordinates, i.e. there is a measurable $F_{\setminus\Delta}$ with $x'_{\setminus\Delta} = F_{\setminus\Delta}(x_{\setminus\Delta}, a)$. Then, for every $m \geq 1$:

(i) there is a measurable encoder h_{disc} with $\mathcal{L}(h_{\mathrm{disc}}) = 0$ and $\mathsf{err}_\Delta(h_{\mathrm{disc}}) = \mathcal{V}(x_\Delta \mid x_{\setminus\Delta})$, which is $\mathrm{tr}\,\mathrm{Cov}(x_\Delta)$ (discarded) when $x_\Delta \perp x_{\setminus\Delta}$, and n_Δ under Assumption 1(i);

(ii) there is a measurable encoder h_{ret} with $\mathcal{L}(h_{\mathrm{ret}}) = 0$ and $\mathsf{err}_\Delta(h_{\mathrm{ret}}) = 0$;

(iii) $\mathcal{L}^*(h_{\mathrm{disc}}) = \mathcal{V}(\tau^* \mid x_{\setminus\Delta}, a)$. If this is positive, h_{disc} minimises $\mathcal{L} + w\mathcal{L}^*$ for no $w > 0$, whereas every minimiser of $\mathcal{L} + w\mathcal{L}^*$ has a zero prediction term and $\mathcal{L}^* = 0$ (Proposition 2), so, under Assumption 1(ii) and (iv), Theorem 1(i) applies to it whenever it also retains $x_{\setminus\Delta}$.

Proof. (i) The law of $x_{\setminus\Delta}$ has a density, so by the isomorphism theorem for measures (Kechris, 1995, Thm. 17.41) there is a Borel bijection $T : \mathbb{R}^{n-n_\Delta} \to \mathbb{R}^m$ carrying it to $\mathcal{N}(0, I_m)$. Put $h_{\mathrm{disc}}(x) := T(x_{\setminus\Delta})$. Then $S(h_{\mathrm{disc}}) = 0$, and with $\mathsf{pred}(z, a) := T\big(F_{\setminus\Delta}(T^{-1}(z), a)\big)$ we have $h_{\mathrm{disc}}(x') = \mathsf{pred}(h_{\mathrm{disc}}(x), a)$, so the prediction term vanishes and $\mathcal{L}(h_{\mathrm{disc}}) = 0$. Since T is a bijection, $h_{\mathrm{disc}}(x)$ and $x_{\setminus\Delta}$ are functions of each other, and (F2) applied twice gives $\mathsf{err}_\Delta(h_{\mathrm{disc}}) = \mathcal{V}(x_\Delta \mid x_{\setminus\Delta})$, which equals $\mathrm{tr}\,\mathrm{Cov}(x_\Delta)$ under independence, and n_Δ under Assumption 1(i).

(ii) Take $h_{\mathrm{ret}} := h_\star$ from the proof of Proposition 2, which does not use $w > 0$; $h_\star$ is a bijection, so $\mathsf{err}_\Delta(h_\star) = 0$ by (F1).

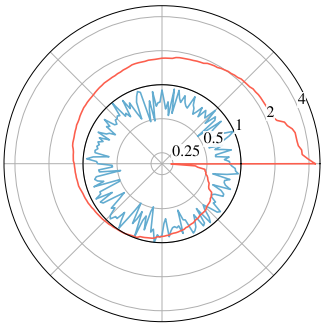

Figure 7: Per-dimension latent range of the region the planner samples, divided by the range of the frozen model over the whole probe set. Each radius is one latent dimension, the radius is logarithmic, and the dimensions are sorted by the fine-tuned curve. The black circle is the ratio 1. Blue is the model before fine-tuning and red the same model after fine-tuning the encoder along with the predictor; the red ranges leave the unit circle and reach about 4.

(iii) By (F2) applied twice, $\mathcal{L}^*(h_{\text{disc}}) = \mathcal{V}(\tau^* \mid h_{\text{disc}}(x), a) = \mathcal{V}(\tau^* \mid x_{\setminus\Delta}, a)$. If it is positive then $(\mathcal{L} + w\mathcal{L}^*)(h_{\text{disc}}) > 0$, while by Proposition 2 the minimum is 0 and is attained only with every term equal to 0. □

The assumption idealises a light ball pushed by a stiff position-controlled arm: the contact force depends on the ball, but the motion of the arm does not. For the contact force, $\mathcal{V}(\tau^* \mid x_{\setminus\Delta}, a) > 0$ as soon as contact occurs with positive probability and the ball state is not a function of the arm state.

Remark 2: Weak coupling

Suppose $x'_{\setminus\Delta} = F_{\setminus\Delta}(x_{\setminus\Delta}, a) + \varepsilon\, G(x, a)$ with $\mathbb{E}\|G\|^2 < \infty$, $x_{\setminus\Delta} \sim \mathcal{N}(0, I)$ and $m = n - n_\Delta$. Then $h(x) := x_{\setminus\Delta}$ has $S(h) = 0$ and prediction term $\varepsilon^2\, \mathcal{V}(G \mid x_{\setminus\Delta}, a) \le \varepsilon^2\, \mathbb{E}\|G\|^2$, while $\mathsf{err}_\Delta(h) = \mathcal{V}(x_\Delta \mid x_{\setminus\Delta})$. A discarding encoder is then a near-minimiser, and the excess loss that pushes the encoder to keep the object is of the order of the squared effect of the object on the remaining coordinates.

C Fine-tuning with a Frozen Encoder

Under naive joint fine-tuning of the encoder and the predictor, the prediction loss on the on-policy data decreases spuriously, because the encoder and SIGReg distort the latent space instead of the predictor learning the dynamics. We therefore freeze the encoder. We also freeze the MLP projection head of the predictor, since updating it shifts the mean of the predicted latents.

When the model is fine-tuned on a dataset restricted to the search space of the planner, the sampled observations are biased, and so are the latents they produce. SIGReg inflates this importance-sampled subspace, and the latent space loses the global properties it had acquired, e.g. the latent distance to the goal. To raise the prediction accuracy of the world model through dataset aggregation while keeping the latent space fixed, we therefore fine-tune with a frozen encoder. Figure 7 shows this inflation per latent dimension.

D Training and Measurement Details

Optimisation. The prediction loss is a mean squared error with weight 1.0, the SIGReg regulariser of Balestriero & LeCun (2025) has weight 0.09 and the force loss has weight 1.0. We optimise with AdamW (learning rate 5×10^{-5}, weight decay 10^{-3}) under a linear-warmup cosine-annealing schedule, with batch size 16 per GPU on four GPUs, for 233,000 steps. All simulation and training ran on NVIDIA H200 GPUs.

Force labels. Within each 40 Hz substep the contact impulse between the stick and the ball reported by the solver is accumulated over the simulator steps, projected onto the board plane, and divided by the ball mass and the substep duration, so that f_s is the mean contact force per unit ball mass over that substep, expressed in the board frame in m/s^2. The contact label of a substep is set when the solver reports a contact between the stick and the ball at any simulator step within it.

Probe. The probe is an MLP with two hidden layers of 512 units, trained, selected and scored on disjoint sets of held-out episodes of the dataset, the same sets for every model, and averaged over three seeds.