ACTION-COORDINATE WORLD MODELS FOR PRECISE CONTROL FROM OFFLINE DATA

Anonymous authors
Paper under double-blind review

ABSTRACT

World models are trained to predict and then used to plan. Where the task demands precision, the two objectives come apart: a model that fits its observations can still leave the planner without a route to the target. We study tasks in which objects have to be placed precisely to maximize a signal, although their positions are never observed directly and a context both displaces them and distorts every movement. For them, we introduce action-coordinates, latent representations in which actions act linearly and the target sits at the origin, and build a world model architecture on top of them that moves structure from the planner into the representation. Our theory gives conditions under which action-coordinates exist, the structure they impose, and when they can be learned from high-dimensional observations, together with a closed-form policy and conditions under which it is optimal. On challenging benchmarks of high-fidelity, visually observed tasks, action-coordinate world models outperform planner-based and gradient-based world models and confirm the assumptions of our theory.

`https://zangentoga.github.io/ac-wm-githubio/`

1 INTRODUCTION

Model-based reinforcement learning promises control where interaction is expensive: a world model is fitted to logged transitions, and a planner searches it for the action to apply, by imagining rollouts in a learned latent space (Hafner et al., 2020), by training on the robot itself (Wu et al., 2022), or by gradient-based planning on the predicted return (Psenka et al., 2026). This division of labor leaves the search with the harder half, where planning over a learned model is considerably harder than the accuracy of that model suggests (Chang et al., 2026). Offline, these difficulties compound (Levine et al., 2020b): the planner exploits the model where the data end, a failure mode that model-based offline methods counter with uncertainty penalties or pessimism (Yu et al., 2020; Kidambi et al., 2020). Tasks on high-precision technical systems that are operated under partial observability, and whose reward is highly sensitive to small misalignments, are a failure case of this division: a residual the model considers small can be a large misalignment, and if no single image reveals the context, identifying it across frames falls to the planner, which cannot return it to the model.

We ask how much of the control problem in such tasks a learned representation can carry. Latent spaces in which actions act simply are a familiar track (Section 1.3), where the operator that moves the latent is typically learned as well. We take a more radical route that makes planning in the latent trivial: we learn *action-coordinate systems*, latent coordinates in which actions act linearly and the target sits at the origin. Solving such a task is then steering the latent to zero, which is closed-form, and no search is left to a planner. The heavy lifting falls to the encoder, which has to absorb the distortion a context causes into coordinates that remain linear in the actions.

In Section 2, we formalize action-coordinate systems, study when they exist, and when a closed-form policy is step-optimal. Section 3 settles when an encoder can realize them from images and actions alone, how much history it needs, and that a memoryless encoder suffices when the context is identified from probing steps instead. Our benchmark is a challenging set of contextual, visually observed tasks from optics, accelerator physics and robotics, two of which run on real hardware, which we bring under one mathematical structure and show to admit action-coordinates (Section 1.2). Finally,

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.

we solve these tasks from high-dimensional offline data alone, with fewer episodes and a better final reward than other model-based and model-free baselines (Section 4).

1.1 Problem formulation

We study episodic reachability problems on contextual, partially observed systems. At state $s \in \mathcal{P} \subset \mathbb{R}^m$, an action $a \in \mathcal{A} \subset \mathbb{R}^d$ changes the system's state through a dynamics $s' = f(s, a, W)$ depending on a unobserved *context* $W \in \mathcal{W}$ which is drawn once per episode and stays fixed while the episode runs. We assume that $f(\cdot, \cdot, W)$ is smooth for every $W \in \mathcal{W}$ and *driftless*, that is, $f(s, a, W) \neq s$ if and only if $a \in \mathcal{A} \setminus \{0\}$, and *reachable*, i.e. for all $s, s' \in \mathcal{P}$ and $W \in \mathcal{W}$ there is an $a_{1:n} \in \mathcal{A}^n$ with $f(s, a_{1:n}, W) = s'$. Moreover, $\mathcal{A} \subset \mathbb{R}^d$ is compact, convex and symmetric with 0 in its interior, the canonical case being the box $\mathcal{A} = [-1, 1]^d$. Neither s nor W is observed and instead, the agent reads a high-dimensional noised observation $O(s, W) \in \mathcal{O}$ together with a reward $R(s, W)$ that measures how well the system performs at s. An episode ends once a step limit T is reached. Solving such a task is a reinforcement learning problem on a contextual Markov decision process (Modi et al., 2018), in which W is a hidden parameter of dynamics, observation and reward that stays fixed while the episode runs. No single observation determines W, so a policy π draws the action a_t from the observations and actions seen so far, $a_t \sim \pi(\cdot \mid o_{1:t}, a_{1:t-1})$ with $o_i := O(s_i, W)$, and we search for a policy that maximizes the expected return $\mathcal{J}(\pi) = \mathbb{E}\left[\sum_{t=1}^{T} \gamma^{t-1} R(s_t, W)\right]$, where the expectation covers the observation noise and the policy as well, and $\gamma \in (0, 1]$ discounts later steps. As reward is paid at every step, $\mathcal{J}$ enforces to reach the set

$$\mathcal{M}_W := \arg\max_{s \in \mathcal{P}} R(s, W) \subset \mathcal{P}, \tag{1}$$

early and to stay there. Maximizing the return is thus the reachability problem we started from, steering to $\mathcal{M}_W$ in as few steps as possible.

1.2 Environments

We use the following environments which share the structure of Section 1.1: an image as the observation, a reward that is maximal at a target set displaced by context, and an actuation the same context rescales. LensAlignment and Relign are active alignment tasks (Burkhardt et al., 2025), each moving a lens system, one is a Cooke triplet the other one inspired by a real-world optical system, whose individual lenses are decentered and tilted anew in every episode. BeamAlignment steers a charged-particle beam onto the diagnostic screen of the ARES experimental area (Kaiser et al., 2022). Interferobot overlaps the two beams of an optical interferometer (Sorokin et al., 2020). LightTunnel tunes focus, gain and two polarizers of a camera looking into a light tunnel (Gamella et al., 2025). SpheroEnv drives a rolling-ball robot to a goal in a planar arena (Siffl, 2026). Figure 1 shows example trajectories, Section E the details. The tasks differ in what that context is, a manufacturing defect, an incoming beam, an actuator calibration, the color of the illumination or an unknown action frame, in whether the target set is a point or a line, as for BeamAlignment, where 5 magnets steer 4 beam parameters, and in where they run: LightTunnel is a physical device and SpheroEnv has a hardware twin, the others are simulators. Some also expose a low-dimensional vector observation, which separates perception from control. Two properties make this class hard. First, W has to be inferred from few high-dimensional observations, and unseen contexts occur at test time. Second, R changes abruptly near $\mathcal{M}_W$ (see Figure 9), so only precise movements reach the target, and the attainable value of $R(\cdot, W)$ is unknown, leaving no absolute scale for progress.

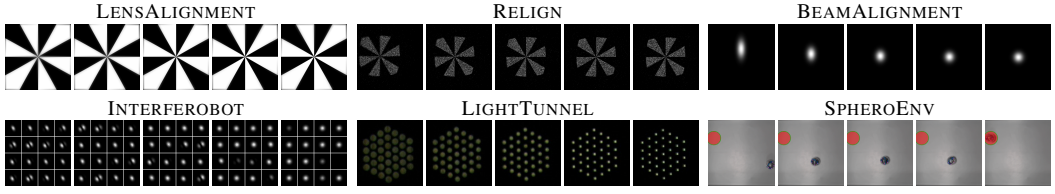


Figure 1: Trajectories towards the optimum per environment, read left to right.

1.3 Related work

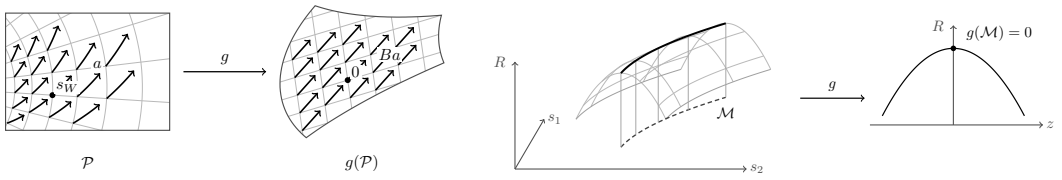
Planning with a learned world model hands the action to an optimizer searching a model trained for prediction. Descending the gradient of a learned forward model goes back to Jordan & Rumelhart (1992); Schmidhuber (1990); Nguyen & Widrow (1990) and returns in gradient-based planning with neural world models (Jyothir S V et al., 2023; Spieler & Behnke, 2026; Deb et al., 2026) and in policies trained through the model (Heess et al., 2015; Hafner et al., 2020; Xu et al., 2022; Amos et al., 2018); over long horizons these gradients explode (Metz et al., 2021) or stall in local optima (Psenka et al., 2026), which sampling planners (Hafner et al., 2019; Hansen et al., 2024) escape at the cost of many model evaluations (Bharadhwaj et al., 2020). Either optimizer exploits a model trained for prediction and repurposed as an objective over actions, the *objective mismatch* (Lambert et al., 2020; Parthasarathy et al., 2025), which decision-aware losses (Farahmand et al., 2017; Nikishin et al., 2022) and planner-shaped latents (Srinivas et al., 2018) reduce while keeping the search. In contrast, we remove it: in action-coordinates the step to the target is a linear solve (Section 3.3), with no horizon to differentiate through and no action objective to exploit.

Actions acting on a latent by a fixed transformation go back to Bowling et al. (2005)'s semidefinite-programming embedding into distance-preserving points. Descendants learn a linear group representation (Quessard et al., 2020; Keurti et al., 2023) or fix known transformations (Delliaux et al., 2025; Higgins et al., 2018; Caselles-Dupré et al., 2019). Others linearize the dynamics: locally with a state-dependent gain and iLQR (Watter et al., 2015; Levine et al., 2020a); globally in a lifted Koopman space (Lusch et al., 2018; Korda & Mezić, 2018; Takeishi et al., 2017; Proctor et al., 2018; Bruder et al., 2021); per episode, as a contextual or hidden-parameter MDP (Hallak et al., 2015; Doshi-Velez & Konidaris, 2016; Perez et al., 2020; Lee et al., 2020; Zintgraf et al., 2020; Forgione et al., 2023; Zhang et al., 2019; O'Connell et al., 2022); or with the goal as temporal or quasimetric distance (Park et al., 2024; Wang et al., 2023). All learn this operator. We prescribe it instead: translation collapses the representation to one gain, the sole entry point for context, fitted by ridge regression over a learned delay-embedding window (Bakarji et al., 2023) with actions replacing a free lag. The latent is globally linear and driftless, of the action's dimension rather than lifted, with the target at the origin and its norm ordered by reward rather than by a learned value.

Offline model-based RL keeps the policy where the model is trusted, by an uncertainty penalty (Yu et al., 2020), a pessimistic MDP (Kidambi et al., 2020), or MPC through a model learned from logs (Argenson & Dulac-Arnold, 2021); latent-prediction models plan offline to an observed goal without policy or reward model (Zhou et al., 2025; Sobal et al., 2025), whereas our goal is context-displaced and inferred. Partial coverage needs data covering the competing policy's occupancy (Xie et al., 2021; Uehara & Sun, 2022). Our anchored collection reverses an outward walk from the optimum physically, as do reverse curricula and backward imagination, online or via a learned model (Nair et al., 2020; Florensa et al., 2017; Wang et al., 2021), and this high-quality, low-coverage data challenges model-free offline learning (Schweighofer et al., 2022).

Optical and particle-accelerator alignment instantiate this problem physically and are almost entirely addressed model-free (see Section E.1 and Section E.2); the classical fixed linear map from misalignment to aberration is the gain we learn per context from data. In Schmähling et al. (2026), augmentations for model-free offline RL are studied for positioning tasks.

2 Action-coordinate systems



(a) Dynamics $f(s, a) = e^{a_1}\mathrm{Rot}(a_2) \cdot s$ with $\mathrm{Rot}(t)$ rotation matrix by t and log-polar g.

(b) Reward anchoring of dynamics from Example 5.

Figure 3: Action-coordinate systems and reward anchoring.

In this section, we formulate and study coordinate systems in which the dynamics f becomes easier to control with respect to a task specified by a reward R. For a sequence $a_{1:n} \in \mathcal{A}^n$ we write $f(s, a_{1:n}, W) := f(\dots f(f(s, a_1, W), a_2, W) \dots, a_n, W)$ for the position it steers s to. Throughout, the position space $\mathcal{P} \subseteq \mathbb{R}^m$ is open and simply connected. The *actuation Jacobian* is the smooth map $J_W \colon \mathcal{P} \to \mathbb{R}^{m \times d}$ whose columns $J_{W,i} := \partial_i f(\cdot, 0, W)$ are the partial derivatives with respect to the i-th action dimension. The dynamics f is *fully actuated* if $m = d$ and $J_W(s)$ is invertible for every $s \in \mathcal{P}$. All proofs are in Section A, and Section D discusses when our assumptions are violated in real-world systems.

2.1 Definition and properties

We start by studying under which conditions a dynamics f admits a coordinate change where movement becomes linear and additive in the actions. For simplicity, the context W is fixed and dropped from the notation, so $f(s, a) := f(s, a, W)$ and $J_W = J$.

Definition 1 (Action-coordinates)**.** Let $1 \leq \ell \leq d$. An *action-coordinate system of rank* ℓ is a pair (g, B) of a map $g \colon \mathcal{P} \to \mathbb{R}^\ell$ and a matrix $B \in \mathbb{R}^{\ell \times d}$ of rank ℓ satisfying

$$g(f(s, a)) = g(s) + B \cdot a \quad \text{for all } (s, a) \in \mathcal{P} \times \mathcal{A}. \tag{2}$$

We call g the *chart* and B the *gain* of the action-coordinate system and it is *full* if $\ell = d$.

On a real-world system, an action-coordinate system can be thought of the inverse kinematics of its mechanism. A mechanism whose position is a function of its actuator coordinate $q \in \mathbb{R}^d$ alone has a forward kinematics $s = \kappa(q)$ with a diffeomorphism $\kappa \colon \mathbb{R}^d \to \mathcal{P}$, and a command a that advances q by Γa for a fixed invertible gain $\Gamma \in \mathbb{R}^{d \times d}$ moves the position to $f(s, a) = \kappa\left(\kappa^{-1}(s) + \Gamma a\right)$. The pair (κ^{-1}, Γ) is then a full action-coordinate system, however non-linear κ is, so the chart an encoder learns is the inverse kinematics. Section D says what κ and Γ are on industrial hardware.

Action-coordinate systems ask the raw action to enter linearly. A wider notion replaces $B \cdot a$ by $B \cdot \phi_\eta(g(s), a)$ for a learned feature map ϕ_η, as in Koopman realizations (Proctor et al., 2018; Bruder et al., 2021) (Section A.6). Our formulation is more restrictive but allows for telescoping latent increments. A policy sees only observations and rewards, and a system with a chart responds to every action sequence exactly like the linear system it charts to. In Section C, we study a more general class of non-linear dynamics admitting action-coordinates. Definition 1 fixes neither origin nor basis: if (g, B) is an action-coordinate system, then so is $(Lg + v, LB)$ for every $L \in \mathbb{R}^{\ell' \times \ell}$ and every $v \in \mathbb{R}^{\ell'}$ (Proposition 3). For fully actuated dynamics, a smooth g is coupled with B by the differential equation $Dg(s) = B \cdot J^{-1}(s)$ (Proposition 4).

Definition 2 (Reversible dynamics)**.** The dynamics f is *reversible* if $f(f(s, a_1, W), a_2, W) = f(s, a_1 + a_2, W)$ for all $s \in \mathcal{P}$, all $a_1, a_2 \in \mathcal{A}$ with $a_1 + a_2 \in \mathcal{A}$, and all $W \in \mathcal{W}$.

Applying first a_1 and then a_2 in a reversible dynamics is the same as applying $a_1 + a_2$ in one step provided it is in $\mathcal{A}$, so in particular an action a is undone by $-a$. The environments from Section 1.2 are, at least to some extent, reversible and hence admit action-coordinates:

Theorem 1 (Reversibility)**.** *A fully actuated dynamics is reversible if and only if it admits a full action-coordinate system.*

Note that Theorem 1 is not a complete characterization of the existence of action-coordinate systems, as non-reversible dynamics can still admit them (Example 7).

2.2 Reward anchoring

So far the coordinates carry no information about the task. Ideally, the chart g normalizes the target in the sense that $g(s) = 0$ exactly for the s in the target set. We analyze such anchors for general sets $\mathcal{M} \subset \mathcal{P}$ and for $\mathcal{M}_W$ from equation 1. If $\mathcal{M}_W$ is a single point, we write $\mathcal{M}_W = \{s_W\}$.

Definition 3 (Anchored coordinates)**.** Let $\mathcal{M} \subseteq \mathcal{P}$. A map $g \colon \mathcal{P} \to \mathbb{R}^\ell$ is *anchored at* $\mathcal{M}$ if $g^{-1}(0) = \mathcal{M}$. An action-coordinate system (g, B) is *anchored at* $\mathcal{M}$ if g is anchored at $\mathcal{M}$. It is *R-anchored* for the context W if it is anchored at $\mathcal{M}_W$.

For a singleton $\mathcal{M} = \{s_0\}$, anchoring is a translation of g by $-g(s_0)$ (Proposition 3), and it forces the action-coordinate system to be full (Lemma 1). For a larger $\mathcal{M}$ it is a genuine demand, and the action-coordinate system needs to be *separating* (see Section A.1). Intuitively, if $g(s) = g(s')$ for two different $s, s' \in \mathcal{M}$, then the coordinates cannot tell the two states apart, and the system is separating if the actions in $\ker B$, which leave g where it is, lead from one state to the other. Thus, a target set $\mathcal{M}$ larger than a single point that is sent to 0 by g, as for BEAMALIGNMENT, forces an action-coordinate system of rank $\ell < d$. Example 5, illustrated in Figure 3b, anchors an action-coordinate system at a non-trivial set $\mathcal{M}$. We want to emphasize that the restriction to driftless dynamics is crucial here: Although action-coordinate systems for dynamics with drifts can exist (Remark 4), anchoring them at $\mathcal{M}_W$ at the same time may be impossible.

Given an R-anchored action-coordinate system (g, B) for a context W, maximizing the reward is equivalent to steering to 0 in action-coordinates. To define the optimal action, we need a norm on the latent space. One action step moves the coordinates by an element of $B \cdot \mathcal{A} = \{Ba : a \in \mathcal{A}\} \subset \mathbb{R}^\ell$, a compact convex symmetric set with 0 in its interior because $\mathcal{A}$ is and B has rank ℓ. For $z \in \mathbb{R}^\ell$, the Minkowski gauge $\|z\|_B := \inf\{t > 0 : z \in t\, B \cdot \mathcal{A}\}$ is a norm on $\mathbb{R}^\ell$ whose closed unit ball is $B \cdot \mathcal{A}$, and $\lceil \|z\|_B \rceil$ is the minimal number of action steps needed to reach z in the latent space. This proves that the following policy is step-optimal:

$$\pi^*_B(z) \in \arg\min_{a \in \mathcal{A}} \|z + Ba\|_B.$$

Theorem 2 (Step-optimality)**.** *Let (g_W, B_W) be an R-anchored action-coordinate system for the context W. Then $\pi^*_{B_W}$ reaches the target $\mathcal{M}_W$ in the minimal number of steps from every $s \in \mathcal{P}$.*

Although R-anchored action-coordinate systems are only unique up to a linear transformation L, Theorem 2 is invariant to these transformations: replacing (g, B) by (Lg, LB) replaces $B \cdot \mathcal{A}$ by $L\,B \cdot \mathcal{A}$ and leaves $\|Lg(s)\|_{LB} = \|g(s)\|_B$. Evaluating π^*_B means solving the convex problem $\min_{a \in \mathcal{A}} \|z + Ba\|_B$ at every step. A closed-form alternative uses the Tikhonov-regularized pseudoinverse of B,

$$\pi_{B,\varepsilon}(z) = \mathrm{scale}_{\mathcal{A}}\left(-\left(B^\top B + \varepsilon I\right)^{-1} B^\top z\right) \in \mathcal{A} \tag{3}$$

which is the Moore–Penrose pseudoinverse for $\varepsilon \to 0$ and where $\mathrm{scale}_{\mathcal{A}}$ scales the vector into $\mathcal{A}$ along its own ray (see also Section G.2 for implementation details). If B is not invertible, $\pi_{B,0}$ is not guaranteed to be step-optimal (Remark 1), but for full systems it is.

Corollary 1 (Closed-form policy for full systems)**.** *Let (g_W, B_W) be an R-anchored action-coordinate system for the context W which is full. Then $\pi_{B_W,0}$ reaches the target $\mathcal{M}_W$ in the minimal number of steps from every $s \in \mathcal{P}$.*

Note that $\mathcal{M}_W$ need not be a singleton in Corollary 1 (Example 7), and Remark 2 gives further conditions under which closed-form policies are step-optimal.

In practice, the tasks of Section 1.2 relax this termination criterion to a tolerance C: reaching $\mathcal{T}_C := \{s \in \mathcal{P} : R(s, W) \geq C\}$ suffices. The direction from which the latent approaches the target then matters, as $\mathcal{T}_C$ need not be a ball of $\|\cdot\|_B$.

Definition 4 (Radiality)**.** The reward R is *radial* in an action-coordinate system (g, B) for $W \in \mathcal{W}$ if there is a strictly decreasing $\rho_W \colon [0, \infty) \to \mathbb{R}$ with $R(s, W) = \rho_W\big(\|g(s)\|_B\big)$ for all $s \in \mathcal{P}$.

Intuitively, radiality asks the reward to fall off equally fast in every direction of the action-coordinates, measured in $\|\cdot\|_B$ rather than the Euclidean norm (Remark 3), and it implies R-anchoring (Lemma 5). Example 6 shows a task whose reward is highly asymmetric in the original coordinates but radial in a suitable action-coordinate system, and Figure 9 shows that the environments of Section 1.2 satisfy the definition up to a small defect. For a threshold $C < \rho_W(0)$ and the radius $r := \rho_W^{-1}(C)$, radiality gives $\mathcal{T}_C = g^{-1}(\{y \in \mathbb{R}^\ell : \|y\|_B \leq r\})$, and π^*_B reaches this set step-optimally. For $C = \rho_W(0)$ this is a special case of Theorem 2.

Corollary 2. *Let R be radial in an action-coordinate system (g, B) for $W \in \mathcal{W}$ with profile ρ_W. Then π^*_B reaches $\mathcal{T}_C$ from every $s \in \mathcal{P}$ in the minimal number of steps for all $C < \rho_W(0)$.*

3 LEARNING ACTION-COORDINATES

While the previous section deals with the existence of action-coordinates, this section asks whether an encoder e_θ can output them from a stacked window of k observations and actions. For $W \in \mathcal{W}$, $s \in \mathcal{P}$, and actions $a_{1:k-1} \in \mathcal{A}^{k-1}$ we set $s_1 = s$ and $s_{i+1} = f(s_i, a_i, W)$ for $1 \le i < k$ and define

$$h(W, s, a_{1:k-1}) := \big(O(s_1, W), \dots, O(s_k, W)\big) \times \big(a_1, \dots, a_{k-1}\big) \in \mathcal{O}^k \times \mathcal{A}^{k-1}.$$

3.1 REALIZABILITY AND IDENTIFIABILITY

Section B studies under which conditions an encoder e realizes the charts $(g_W)_{W \in \mathcal{W}}$ of a family of action-coordinate systems, that is, $e\big(h(W, s, a_{1:k-1})\big) = g_W(s_k)$ (Definition 6), and which history length k this requires. Simple examples show that $k = 1$ can be insufficient because of symmetries in the observations (Example 8). Moreover, action-coordinate systems that are equivalent up to a linear transformation (Proposition 3) may require different horizons, and the gap can be d large (Proposition 6). Proposition 5 makes this precise: realizing a system with the *simpler* gain $B_W = I$ requires a history which is at most d observations longer than realizing one with a more complex B_W. At test time, a memoryless encoder ($k = 1$) can hence rely on action-coordinates whose gain B_W is estimated from d transitions *once* per episode, while an encoder for a system with $B = I$ needs d past observations in *every prediction*. Proposition 6 establishes this for observations that are an injective function of the offset to the target. At training time, on the other hand, a memoryless encoder has to satisfy more involved constraints, leading to a potentially more brittle and inefficient training (see Section F.3).

3.2 A LATENT WORLD MODEL FORMULATION

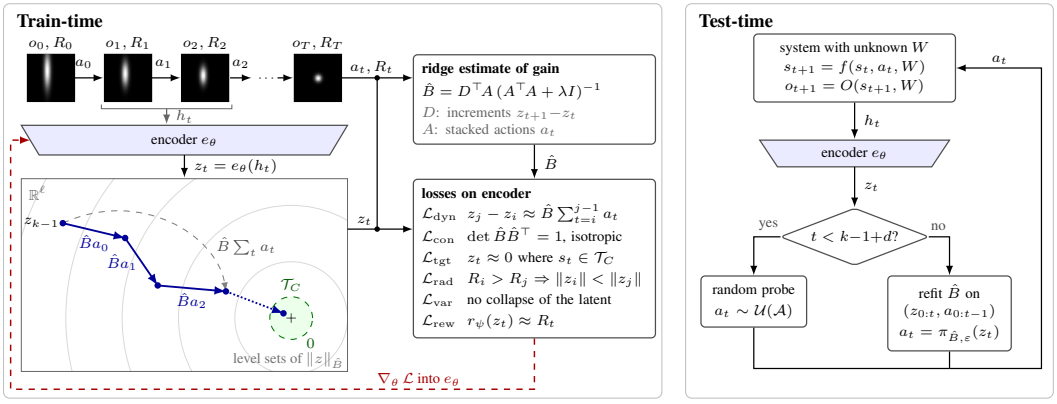


Figure 4: AC-WM at train- and test-time with fitted gain $\hat{B}$.

The core idea is an action-coordinate world model, *AC-WM*, that learns an action-coordinate system for all contexts $W \in \mathcal{W}$ simultaneously: an encoder e_θ is constrained in a way so that it imitates the chart g_W of an action-coordinate system (g_W, B_W). Its rank ℓ and the history length k are hyperparameters. The target set essentially decides ℓ: an R-anchored system is full whenever $\mathcal{M}_W$ is a singleton (Lemma 1), while a separating system anchored at larger $\mathcal{M}_W$ has $\ell < d$ (Section A.1). Let $\tau = (o_0, a_0, R_0), \dots, (o_{T-1}, a_{T-1}, R_{T-1}), (o_T, R_T)$ be a trajectory for a fixed but unobserved context W, so that $o_t = O(s_t, W)$ and $s_{t+1} = f(s_t, a_t, W)$, and let $z_t := e_\theta(h_t) \in \mathbb{R}^\ell$ be the latent of the window $h_t := (o_{t-k+1}, a_{t-k+1}, \dots, a_{t-1}, o_t)$, defined for $k - 1 \le t \le T$. If e_θ realizes an action-coordinate system, there is a $B_W \in \mathbb{R}^{\ell \times d}$ with $z_{t+1} = z_t + B_W\, a_t$, and telescoping along τ gives $z_j - z_i = B_W \cdot \sum_{t=i}^{j-1} a_t$ for all $k - 1 \le i < j \le T$. Although B_W is context-dependent, it is not a parameter or output of our model: at test time we estimate it from data in a rolling fashion (Section 3.3), and at train time the embedding is constrained to admit a B_W for every $W \in \mathcal{W}$ through the ridge estimate

$$\hat{B}_{i,n}(\theta) := D_{i:n}(\theta)^\top \cdot A_{i:n} \left(A_{i:n}^\top \cdot A_{i:n} + \lambda I\right)^{-1} \in \mathbb{R}^{\ell \times d}, \tag{4}$$

with a regularization parameter $\lambda > 0$, stacked actions $A_{i:n} := (a_i, \dots, a_{i+n-1})^\top \in \mathbb{R}^{n\times d}$, and latent increments

$$\begin{aligned} D_{i:n}(\theta) :=& (e_\theta(h_{i+1}) - e_\theta(h_i), \dots, e_\theta(h_{i+n}) - e_\theta(h_{i+n-1}))^\top \\ =& (z_{i+1} - z_i, \dots, z_{i+n} - z_{i+n-1})^\top \in \mathbb{R}^{n\times \ell}. \end{aligned}$$

Then, we get $D_{i:n}(\theta)^\top \approx \hat{B}_{i,n}(\theta) \cdot A_{i:n}^\top$. We write $\hat{B}(\theta, \tau) := \hat{B}_{k-1,T-k+1}(\theta)$ for the estimate of the full trajectory, and as $\hat{B}(\theta, \tau)$ is differentiable in θ, it back-propagates into the encoder e_θ:

$$\mathcal{L}_{\mathrm{dyn}}(\theta) = \frac{1}{\binom{T-k+2}{2}} \sum_{k-1\le i<j\le T} \left\| e_\theta(h_j) - e_\theta(h_i) - \hat{B}(\theta, \tau) \sum_{t=i}^{j-1} a_t \right\|^2. \tag{5}$$

The term $\mathcal{L}_{\mathrm{dyn}}$ allows the trivial solution $e_\theta = 0$, for which $D_{i:n}(\theta) = 0$ and $\hat{B}(\theta) = 0$. To avoid this collapse, we keep $\hat{B}(\theta)$ at rank ℓ with a loss that works reliably across dimensions $\ell \times d$:

$$\mathcal{L}_{\mathrm{con}}(\theta) = \left(\log\det(\hat{B}(\theta)\hat{B}(\theta)^\top)^{1/2} \right)^2 + \mathrm{ReLU}\left(\frac{\|\hat{B}(\theta)\|_F^2}{\ell \, \det(\hat{B}(\theta)\hat{B}(\theta)^\top)^{1/\ell}} - \nu \right) \tag{6}$$

for a threshold $\nu \ge 1$. Both terms of $\mathcal{L}_{\mathrm{con}}$ read $\hat{B}$ through its ℓ singular values, whose product is $\det(\hat{B}\hat{B}^\top)^{1/2}$. The first term forces that product to 1, so a smaller dynamics residual cannot be bought by shrinking. The second bounds the anisotropy of $\hat{B}$, as the quotient is at least 1 by the inequality of arithmetic and geometric means, with equality exactly for equal singular values, and is scale-invariant. As the ridge term shrinks $\hat{B}$ to zero wherever $A_{i:n}^\top A_{i:n}$ is near-singular, $\mathcal{L}_{\mathrm{dyn}}$ pins $\hat{B}$ only along the directions the actions explore, so the train data needs sufficiently random actions. In full systems we can pin $\hat{B}(\theta, \tau) = I \in \mathbb{R}^{\ell\times\ell}$ making $\mathcal{L}_{\mathrm{dyn}}$ easier, but requiring larger k (Proposition 5). The term $\mathcal{L}_{\mathrm{dyn}}$ is invariant to latent translations $z \mapsto z + v$, which reward anchoring fixes: Definition 3 asks $g_W^{-1}(0) = \mathcal{M}_W$, so ideally $e_\theta(h_t) = 0$ exactly when $s_t \in \mathcal{M}_W$. As in high-fidelity reachability tasks, the target often is to reach $\mathcal{T}_C \supseteq \mathcal{M}_W$, we relax this to

$$\mathcal{L}_{\mathrm{tgt}}(\theta) = \tfrac{1}{|\tau\cap\mathcal{T}_C|} \textstyle\sum_{h_t\in\tau\cap\mathcal{T}_C} \|e_\theta(h_t)\|^2. \tag{7}$$

As $\mathcal{L}_{\mathrm{tgt}}$ is only defined for trajectories that visit the target set, it imposes requirements on the data collection, too (Section 4). To use all collected data, we derive a further term from the reward signal. Radiality (Definition 4) makes $R(\cdot, W) = \rho_W(\|g_W(\cdot)\|_B)$ for a strictly decreasing profile ρ_W, so $R_i > R_j$ forces $\|z_i\|_{B_W} < \|z_j\|_{B_W}$ within an episode, and a latent that respects this ordering on every pair is radial for the profile the data defines. As ρ_W is unknown, we constrain the ordering alone, which is invariant to every strictly monotone reparametrization of the reward:

$$\mathcal{L}_{\mathrm{rad}}(\theta) = \frac{\sum_{i\ne j} |R_i - R_j| \cdot \mathrm{ReLU}\Big(\mathrm{sign}(R_i - R_j)\big(\|e_\theta(h_i)\|^2 - \|e_\theta(h_j)\|^2\big)\Big)}{\sum_{i\ne j} |R_i - R_j| + \varepsilon} \tag{8}$$

with $\varepsilon > 0$ for numerical stability. We use $\|\cdot\|^2$ instead of $\|\cdot\|_{\hat{B}(\theta)}$ in $\mathcal{L}_{\mathrm{rad}}$ to preserve differentiability and to avoid couplings with the terms relying on the gain. The anisotropy bound of $\mathcal{L}_{\mathrm{con}}$ keeps the two norms comparable, and radial systems with $B = I$ exist (Example 6). On flat reward landscapes $\mathcal{L}_{\mathrm{rad}}$ imposes no sharp constraint, and an independent reward predictor r_ψ with $\mathcal{L}_{\mathrm{rew}}(\theta, \psi) := \frac{1}{T-k+2} \sum_{t=k-1}^{T} \left(r_\psi(z_t) - R_t\right)^2$ helps. Finally, to prevent a collapse of the latent onto a constant or a proper subspace of $\mathbb{R}^\ell$, which Lemma 3 rules out for a chart, we add a variance and decorrelation term: with σ_{pq} the covariance of the coordinates p and q of the latents z_t pooled over a batch, $\sigma_p := \sqrt{\sigma_{pp}}$, and a floor $\sigma_{\min} > 0$, $\mathcal{L}_{\mathrm{var}}(\theta) = \frac{1}{\ell}\sum_{p=1}^{\ell}(\sigma_{\min} - \sigma_p)_+ + \frac{1}{\ell}\sum_{p\ne q}\sigma_{pq}^2$. All terms constrain the same encoder against one another,

$$\mathcal{L}(\theta, \psi) = \lambda_{\mathrm{dyn}}\mathcal{L}_{\mathrm{dyn}}(\theta) + \lambda_{\mathrm{con}}\mathcal{L}_{\mathrm{con}}(\theta) + \lambda_{\mathrm{tgt}}\mathcal{L}_{\mathrm{tgt}}(\theta) + \lambda_{\mathrm{rad}}\mathcal{L}_{\mathrm{rad}}(\theta) + \lambda_{\mathrm{var}}\mathcal{L}_{\mathrm{var}}(\theta) + \lambda_{\mathrm{rew}}\mathcal{L}_{\mathrm{rew}}(\theta, \psi),$$

and Section G.1 discusses the tensions between them and the weights we use.

3.3 Control and identification at test time

At test time the encoder stands in for the chart g_W and the batch estimate $\hat{B}$ for the gain B_W. Before suggesting actions at test time for context $W \in \mathcal{W}$, the encoder needs k observations to fill

its window and d transitions, the algebraic minimum needed to determine $\hat{B}$ with equation 4. Thus, an episode opens with $n_{\text{probe}} \geq k - 1 + d$ random actions and plans from then on. Drawing those actions at random makes the estimate of $\hat{B}$ well-posed, since $A_{i:n}$ then has full rank with probability 1. Either part of the budget vanishes on its own: a memoryless encoder needs no window, a gain pinned to $\hat{B} = I$ needs no identification. At every step, $\hat{B}$ is refitted on the full history so far if $\hat{B}$ is not pinned to I. The action applied at latent $z_t = e_\theta(h_t)$ is $a_t := \pi_{\hat{B},\varepsilon}(z_t)$ from equation 3, where $\pi_{\hat{B},\varepsilon}$ is the closed-form policy from Corollary 1 read in the learned coordinates. Recall that when $\ell < d$, $\pi_{\hat{B},\varepsilon}$ serves as a closed-form heuristic approximation rather than the exact step-optimal policy of Corollary 1 and for $\mathcal{A}$ a box, the true optimal policy would require to solve a low-dimensional linear program at every step (Remark 2).

4 Experiments

Data collection Good data for offline RL covers $\mathcal{P} \times \mathcal{A}$ and contains trajectories solving the task (Schweighofer et al., 2022). We therefore use an *anchored* data collection: each episode starts in $\mathcal{M}_W$, walks outward for T steps with actions drawn uniformly from $\alpha \cdot \mathcal{A}$, $0 < \alpha \leq 1$, and is stored in reversed order and negated actions. By reversibility (Definition 2), this is a random-action trajectory that ends in the target. More details on collection, training, and evaluation are in Section E.

Comparison to existing planner-based offline world model schemes We compare AC-WM against *vanilla* world model schemes consisting of an encoder e_ϕ, a latent dynamics g_ϕ and a reward head r_ϕ of the same architecture, trained jointly. All three maximize over $\mathcal{A}$ the model's own one-step estimate $J(a) := r_\phi(g_\phi(e_\phi(h_t), a)) - \rho\|a\|_1$, damped by an ℓ_1 penalty, and differ only in how: RG-WM by projected gradient ascent, S-WM by cross-entropy sampling, and MPC-WM by that sampler on the sum of J over a sequence of length 5. We also compare against three baseline policies: *Random* executes uniform actions from $\mathcal{A}$, *context-blind* drives to $s^* = \arg\max_{s \in \mathcal{P}} \mathbb{E}_W[R(s, W)]$, a nominal setting that maximizes the expected reward over the context distribution without reading the context, and *optimal* is the optimal policy for each context. Figure 5 shows the result: AC-WM outperforms the other world models, and on most environments it is close to step-optimal once the probing steps are done. The exception is BeamAlignment, which needs more steps to reach the target because the policy $\pi_{\hat{B},\varepsilon}$ is not guaranteed to be step-optimal here. Why vanilla world models fail so often, above all on the image-based tasks, is not the encoder: on the SpheroEnv frames both models read the image through the same encoder, and the task has only $d = 2$ actions. The reason is that a vanilla world model needs more data before its planner starts working (Figure 23a in Section E). There, we also compare the learned embedding of AC-WM to those of RG-WM and find that AC-WM learns a robust representation of the ground-truth positions (Figure 23b and Figure 15). For AC-WM, an analysis of the influence of k and whether the gain should be fitted or pinned can be found in Section E for each environment.

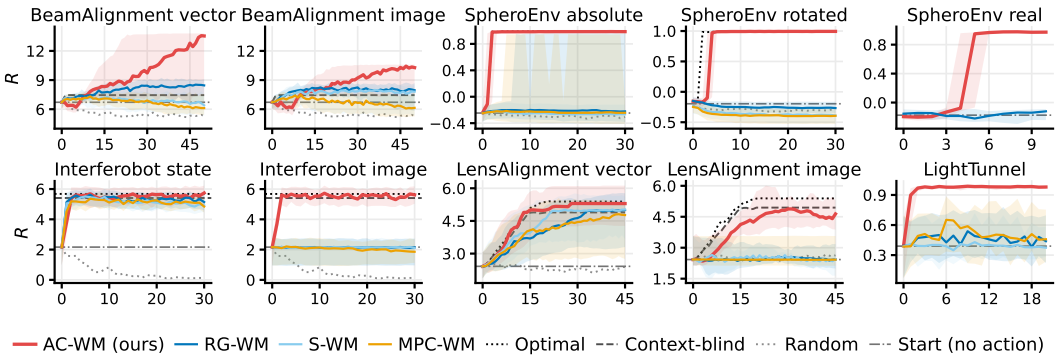


Figure 5: Reward against the evaluation step, best configuration per method (Section E).

Comparison against offline model-free RL We compare AC-WM against IQL (Kostrikov et al., 2022) and TD3+BC (Fujimoto & Gu, 2021) on anchored and random data of same size (Figure 6).

Both baselines are tuned to the task, and on random data AC-WM anchors on the 50 states closest to $\mathcal{M}_W$. AC-WM ends highest on every panel on either dataset. Anchored data lift IQL on every task and TD3+BC on three. On every task, AC-WM is at least as good on anchored as on random data.

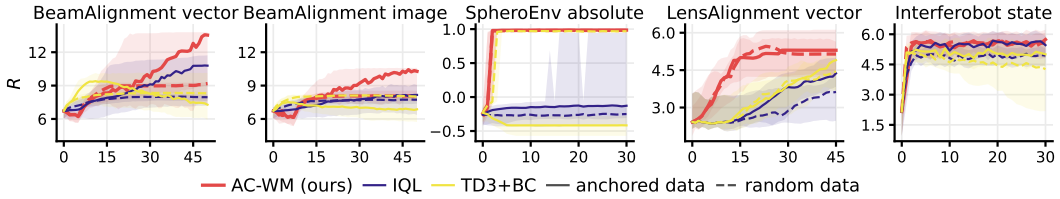
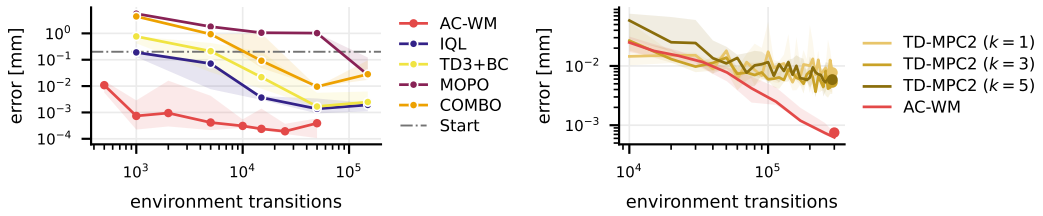


Figure 6: Reward against the evaluation step on anchored and random data.

Offline and online sample-efficiency We trained AC-WM under varying amounts of data on BEAMALIGNMENT with vector observations. Offline (Figure 7a, Table 3), no dataset size lets IQL, TD3+BC, MOPO (Yu et al., 2020) or COMBO (Yu et al., 2021) reach what AC-WM has with 10^3 transitions. Although not primarily designed for the online regime, we let AC-WM collect its own train data (Figure 7b): from 500 random-walk episodes on, the controller collects in rounds, a tenth of its steps random, and a fresh model is trained after every round with the frames closest to the target as its anchors. TD-MPC2 (Hansen et al., 2024), a state-of-the-art online model-based method, runs over the same budget with windows $k \in \{1, 3, 5\}$ (Section E). From 10^5 transitions on, AC-WM lies below every TD-MPC2 curve. However, AC-WM trained offline on $1.5 \cdot 10^4$ anchored transitions ends below what it reaches after $3 \cdot 10^5$ online transitions (Figure 7a).



(a) Offline comparison.

(b) Online comparison.

Figure 7: Final beam error on BEAMALIGNMENT against the number of transitions.

Ablations on assumptions and design choices In Section F, a detailed ablation study is performed to understand the dependence of AC-WM on the theoretical assumptions and design choices given in Sections 2 and 3. There, we show that AC-WM also works even if the underlying dynamics does not admit an action-coordinate system, but also that its performance relies crucially on the right choice of ℓ and k. We also analyze contributions to the anchor mechanism.

5 CONCLUSION AND LIMITATIONS

We present a mathematically grounded, highly efficient, and effective offline model-based RL method for high-fidelity and high-dimensional reachability tasks. Where the theory of our method applies, it applies in the interior, and Section D discusses violations in real environments. Our experiments cover the reachability tasks of Section 1.1, which come with special assumptions: the closed-form policy steers a latent to the origin, so a task whose solution is a trajectory rather than a target set, or one that needs credit assignment over a horizon, lies outside both the theory and the architecture. As for other offline methods, the performance relies on trajectories that solve the task, and creating such data can be resource-intensive. We saw a particularly good performance of our method when anchored data was used, which may be infeasible for some tasks. Our mathematical foundation, specifically the anchoring mechanism (Remark 4) is restricted to driftless and reversible kinematics. Continuous control systems governed by non-zero autonomous drift, second-order momentum, non-holonomic velocity constraints, or mechanical hysteresis lie outside the formulation. Extending and generalizing our ideas to these systems is an interesting direction for future work.

AI USE STATEMENT

The present work is the result of an iterative process in which the authors have checked, strengthened, streamlined and steered the output of generative AI tools. We used these tools to formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, and to create and edit software code to run our experiments. We used it to search for examples and counter-examples for our hypothesis, in order to sharpen our statements and to find the right level of generality. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here. Our study involves no human subjects and no personal data. All environments but two are simulators; the exceptions are LIGHTTUNNEL and SPHEROENV, two physical setups whose cameras record no persons. Every dataset is collected by us from these systems, and none contains third-party content.

REPRODUCIBILITY STATEMENT

For the reproducibility of our work, we provide the code as supplemental material of this submission. It will be made publicly available upon acceptance. We fully describe the experimental details and hyperparameters to reproduce the results for our method and baselines in the appendix.

REFERENCES

Brandon Amos, Ivan Dario Jimenez Rodriguez, Jacob Sacks, Byron Boots, and J. Zico Kolter. Differentiable MPC for end-to-end planning and control. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 31, 2018. URL https://arxiv.org/abs/1810.13400.

Arthur Argenson and Gabriel Dulac-Arnold. Model-based offline planning. In *International Conference on Learning Representations (ICLR)*, 2021. URL https://arxiv.org/abs/2008.05556.

Karl Johan Åström and Richard M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, Princeton, NJ, 2 edition, 2021. ISBN 978-0-691-19398-4.

Joseph Bakarji, Kathleen Champion, J. Nathan Kutz, and Steven L. Brunton. Discovering governing equations from partial measurements with deep delay autoencoders. *Proceedings of the Royal Society A*, 479(2276), 2023. doi: 10.1098/rspa.2023.0422.

Homanga Bharadhwaj, Kevin Xie, and Florian Shkurti. Model-predictive control via cross-entropy and gradient-based optimization. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control (L4DC)*, 2020. URL https://arxiv.org/abs/2004.08763.

Michael Bowling, Ali Ghodsi, and Dana Wilkinson. Action respecting embedding. In *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 65–72. ACM, 2005. doi: 10.1145/1102351.1102360. URL https://doi.org/10.1145/1102351.1102360.

Daniel Bruder, Xun Fu, and Ram Vasudevan. Advantages of bilinear Koopman realizations for the modeling and control of systems with unknown dynamics. *IEEE Robotics and Automation Letters*, 6(3):4369–4376, 2021. doi: 10.1109/LRA.2021.3068117. URL https://doi.org/10.1109/LRA.2021.3068117.

Matthias Burkhardt, Tobias Schmähling, Pascal Stegmann, Michael Layh, and Tobias Windisch. Active alignments of lens systems with reinforcement learning, 2025. URL https://arxiv.org/abs/2503.02075.

Hugo Caselles-Dupré, Michael Garcia-Ortiz, and David Filliat. Symmetry-based disentangled representation learning requires interaction with environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019. URL https://arxiv.org/abs/1904.00243.

Wei-Di Chang, Mikael Henaff, Brandon Amos, Gregory Dudek, and Scott Fujimoto. The surprising difficulty of search in model-based reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=gTbSqYqumM.

Rohan Deb, Stephen J. Wright, and Arindam Banerjee. Inference time policy optimization for offline rl with differentiable world models, 2026. URL https://arxiv.org/abs/2603.22430.

Thomas Delliaux, Nguyen-Khanh Vu, Vincent François-Lavet, Elise van der Pol, and Emmanuel Rachelson. Learning abstract world models with a group-structured latent space, 2025. URL https://arxiv.org/abs/2506.01529.

Finale Doshi-Velez and George Konidaris. Hidden parameter Markov decision processes: A semiparametric regression approach for discovering latent task parametrizations. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI)*, 2016. URL https://arxiv.org/abs/1308.3513.

Amir-massoud Farahmand, André Barreto, and Daniel Nikovski. Value-aware loss function for model-based reinforcement learning. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017. URL http://proceedings.mlr.press/v54/farahmand17a/farahmand17a.pdf.

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. Reverse curriculum generation for reinforcement learning. In *Proceedings of the 1st Conference on Robot Learning (CoRL)*, 2017. URL https://arxiv.org/abs/1707.05300.

Marco Forgione, Filippo Pura, and Dario Piga. From system models to class models: An in-context learning paradigm. *IEEE Control Systems Letters*, 7:3513–3518, 2023. URL https://arxiv.org/abs/2308.13380.

Scott Fujimoto and Shixiang Shane Gu. A minimalist approach to offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.06860.

Juan L. Gamella, Jonas Peters, and Peter Bühlmann. Causal chambers as a real-world physical testbed for AI methodology. *Nature Machine Intelligence*, 7(1):107–118, 2025. ISSN 2522-5839. doi: 10.1038/s42256-024-00964-x. URL https://doi.org/10.1038/s42256-024-00964-x.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1811.04551.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1912.01603.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual Markov decision processes, 2015. URL https://arxiv.org/abs/1502.02259.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2310.16828.

Kramer Harrison. Optiland, 2026. URL https://github.com/optiland/optiland.

Nicolas Heess, Greg Wayne, David Silver, Timothy Lillicrap, Tom Erez, and Yuval Tassa. Learning continuous control policies by stochastic value gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015. URL https://arxiv.org/abs/1510.09142.

Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations, 2018. URL https://arxiv.org/abs/1812.02230.

Simon Hirlaender and Niky Bruchon. Model-free and Bayesian ensembling model-based deep reinforcement learning for particle accelerator control demonstrated on the FERMI FEL, 2020. URL https://arxiv.org/abs/2012.09737.

Enjie Hu, Jiajian He, Jingwen Zhou, Zheng Ren, Huaze Sun, Tingting Jiang, Huajun Feng, and Yueting Chen. Fast and accurate active alignment of camera lenses with physics-informed deep learning. *Optics Express*, 33(10):21256–21273, 2025. doi: 10.1364/OE.560123.

Michael I. Jordan and David E. Rumelhart. Forward models: Supervised learning with a distal teacher. *Cognitive Science*, 16(3):307–354, 1992. doi: 10.1207/s15516709cog1603_1. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1603_1.

Jyothir S V, Siddhartha Jalagam, Yann LeCun, and Vlad Sobal. Gradient-based planning with world models, 2023. URL https://arxiv.org/abs/2312.17227.

Verena Kain, Simon Hirlander, Brennan Goddard, Francesco Maria Velotti, Giovanni Zevi Della Porta, Niky Bruchon, and Gianluca Valentino. Sample-efficient reinforcement learning for CERN accelerator control. *Physical Review Accelerators and Beams*, 23:124801, 2020. doi: 10.1103/PhysRevAccelBeams.23.124801.

Jan Kaiser, Oliver Stein, and Annika Eichler. Learning-based optimisation of particle accelerators under partial observability without real-world training. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato (eds.), *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 10575–10585. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/kaiser22a.html.

Jan Kaiser, Chenran Xu, Annika Eichler, and Andrea Santamaria Garcia. Bridging the gap between machine learning and particle accelerator physics with high-speed, differentiable simulations. *Phys. Rev. Accel. Beams*, 27:054601, May 2024a. doi: 10.1103/PhysRevAccelBeams.27.054601. URL https://link.aps.org/doi/10.1103/PhysRevAccelBeams.27.054601.

Jan Kaiser, Chenran Xu, Annika Eichler, Andrea Santamaria Garcia, Oliver Stein, Erik Bründermann, Willi Kuropka, Hannes Dinter, Frank Mayet, Thomas Vinatier, Florian Burkart, and Holger Schlarb. Reinforcement learning-trained optimisers and Bayesian optimisation for online particle accelerator tuning. *Scientific Reports*, 14(1):15733, 2024b. ISSN 2045-2322. doi: 10.1038/s41598-024-66263-y. URL https://www.nature.com/articles/s41598-024-66263-y.

Hamza Keurti, Hsiao-Ru Pan, Michel Besserve, Benjamin F. Grewe, and Bernhard Schölkopf. Homomorphism AutoEncoder – learning group structured representations from observed transitions. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2207.12067.

Rahul Kidambi, Aravind Rajeswaran, Praneeth Netrapalli, and Thorsten Joachims. MOReL: Model-based offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2005.05951.

Seonghui Kim, Ho-Soon Yang, Yun-Woo Lee, and Sug-Whan Kim. Merit function regression method for efficient alignment control of two-mirror optical systems. *Optics Express*, 15(8):5059–5068, 2007. doi: 10.1364/OE.15.005059.

Milan Korda and Igor Mezić. Linear predictors for nonlinear dynamical systems: Koopman operator meets model predictive control. *Automatica*, 93:149–160, 2018. doi: 10.1016/j.automatica.2018.03.046. URL https://doi.org/10.1016/j.automatica.2018.03.046.

Ilya Kostrikov, Ashvin Nair, and Sergey Levine. Offline reinforcement learning with implicit Q-learning. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2110.06169.

Nathan Lambert, Brandon Amos, Omry Yadan, and Roberto Calandra. Objective mismatch in model-based reinforcement learning. In *Proceedings of the 2nd Conference on Learning for Dynamics and Control (L4DC)*, 2020. URL https://arxiv.org/abs/2002.04523.

John M. Lee. *Introduction to Smooth Manifolds*, volume 218 of *Graduate Texts in Mathematics*. Springer, New York, 2 edition, 2012. doi: 10.1007/978-1-4419-9982-5.

Kimin Lee, Younggyo Seo, Seunghyun Lee, Honglak Lee, and Jinwoo Shin. Context-aware dynamics model for generalization in model-based reinforcement learning. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020. URL https://arxiv.org/abs/2005.06800.

Nir Levine, Yinlam Chow, Rui Shu, Ang Li, Mohammad Ghavamzadeh, and Hung Bui. Prediction, consistency, curvature: Representation learning for locally-linear control. In *International Conference on Learning Representations (ICLR)*, 2020a. URL https://arxiv.org/abs/1909.01506.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020b. URL https://arxiv.org/abs/2005.01643.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nature Communications*, 9(1):4950, 2018. doi: 10.1038/s41467-018-07210-0. URL https://doi.org/10.1038/s41467-018-07210-0.

Stepan Makarenko, Dmitry Igorevich Sorokin, Alexander Ulanov, and Alexander Lvovsky. Aligning an optical interferometer with beam divergence control and continuous action space. In *Proceedings of the 5th Conference on Robot Learning*, pp. 918–927. PMLR, 2022. URL https://proceedings.mlr.press/v164/makarenko22a.html.

Luke Metz, C. Daniel Freeman, Samuel S. Schoenholz, and Tal Kachman. Gradients are not all you need, 2021. URL https://arxiv.org/abs/2111.05803.

Aditya Modi, Nan Jiang, Satinder Singh, and Ambuj Tewari. Markov decision processes with continuous side information. In Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan (eds.), *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pp. 597–618. PMLR, 07–09 Apr 2018. URL https://proceedings.mlr.press/v83/modi18a.html.

Suraj Nair, Mohammad Babaeizadeh, Chelsea Finn, Sergey Levine, and Vikash Kumar. Time reversal as self-supervision. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2020. URL https://arxiv.org/abs/1810.01128.

Derrick H. Nguyen and Bernard Widrow. Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, 10(3):18–23, 1990. doi: 10.1109/37.55119.

Evgenii Nikishin, Romina Abachi, Rishabh Agarwal, and Pierre-Luc Bacon. Control-oriented model-based reinforcement learning with implicit differentiation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022. URL https://arxiv.org/abs/2106.03273.

Jalo Nousiainen, Chang Rajani, Markus Kasper, and Tapio Helin. Adaptive optics control using model-based reinforcement learning. *Optics Express*, 29(10):15327–15344, 2021. doi: 10.1364/OE.420270.

Michael O'Connell, Guanya Shi, Xichen Shi, Kamyar Azizzadenesheli, Anima Anandkumar, Yisong Yue, and Soon-Jo Chung. Neural-Fly enables rapid learning for agile flight in strong winds. *Science Robotics*, 7(66):eabm6597, 2022. doi: 10.1126/scirobotics.abm6597. URL https://doi.org/10.1126/scirobotics.abm6597.

Seohong Park, Tobias Kreiman, and Sergey Levine. Foundation policies with Hilbert representations. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. URL https://arxiv.org/abs/2402.15567.

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

Christian F. Perez, Felipe Petroski Such, and Theofanis Karaletsos. Generalized hidden parameter MDPs: Transferable model-based RL in a handful of trials. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020. URL https://arxiv.org/abs/2002.03072.

Joshua L. Proctor, Steven L. Brunton, and J. Nathan Kutz. Generalizing Koopman theory to allow for inputs and control. *SIAM Journal on Applied Dynamical Systems*, 17(1):909–930, 2018. doi: 10.1137/16M1062296. URL https://doi.org/10.1137/16M1062296.

Michael Psenka, Michael Rabbat, Aditi S. Krishnapriyan, Yann LeCun, and Amir Bar. Parallel stochastic gradient-based planning for world models. In *ICLR 2026 the 2nd Workshop on World Models: Understanding, Modelling and Scaling*, 2026. URL https://openreview.net/forum?id=mA5eDiVlG6.

Robin Quessard, Thomas D. Barrett, and William R. Clements. Learning disentangled representations and group structure of dynamical environments. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2002.06991.

Lea Richtmann, Viktoria-S. Schmiesing, Dennis Wilken, Jan Heine, Aaron Tranter, Avishek Anand, Tobias J. Osborne, and Michèle Heurs. Model-free reinforcement learning with noisy actions for automated experimental control in optics. *Transactions on Machine Learning Research*, 2025. URL https://arxiv.org/abs/2405.15421.

Tobias Schmähling, Matthias Burkhardt, and Tobias Windisch. Trajectory-level data augmentation for offline reinforcement learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=LvYuEB4POi.

Jürgen Schmidhuber. An on-line algorithm for dynamic reinforcement learning and planning in reactive environments. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 253–258, San Diego, CA, USA, 1990.

Kajetan Schweighofer, Andreas Radler, Marius-Constantin Dinu, Markus Hofmarcher, Vihang Patil, Angela Bitto-Nemling, Hamid Eghbal-zadeh, and Sepp Hochreiter. A Dataset Perspective on Offline Reinforcement Learning, 2022. URL http://arxiv.org/abs/2111.04714.

Takuma Seno and Michita Imai. d3rlpy: An offline deep reinforcement learning library. *Journal of Machine Learning Research*, 23(315):1–20, 2022. URL https://www.jmlr.org/papers/v23/22-0017.html.

Kilian Siffl. Sim2real reinforcement learning for vision-based active positioning. masterthesis, University of Applied Sciences Kempten, 2026.

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, 2025. URL https://arxiv.org/abs/2502.14819.

Dmitry Sorokin, Alexander Ulanov, Ekaterina Sazhina, and Alexander Lvovsky. Interferobot: Aligning an optical interferometer by a reinforcement learning agent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 13238–13248. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/hash/99ba5c4097c6b8fef5ed774a1a6714b8-Abstract.html.

Jonathan Spieler and Sven Behnke. Dream-MPC: Gradient-based model predictive control with latent imagination, 2026. URL https://arxiv.org/abs/2605.04568. International Conference on Machine Learning (ICML).

Aravind Srinivas, Allan Jabri, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Universal planning networks: Learning generalizable representations for visuomotor control. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/abs/1804.00645.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. URL https://arxiv.org/abs/1710.04340.

Masatoshi Uehara and Wen Sun. Pessimistic model-based offline reinforcement learning under partial coverage. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2107.06226.

Jianhao Wang, Wenzhe Li, Haozhe Jiang, Guangxiang Zhu, Siyuan Li, and Chongjie Zhang. Offline reinforcement learning with reverse model-based imagination. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2110.00188.

Tongzhou Wang, Antonio Torralba, Phillip Isola, and Amy Zhang. Optimal goal-reaching reinforcement learning via quasimetric learning. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023. URL https://arxiv.org/abs/2304.01203.

Manuel Watter, Jost Tobias Springenberg, Joschka Boedecker, and Martin Riedmiller. Embed to control: A locally linear latent dynamics model for control from raw images. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 28, 2015. URL https://arxiv.org/abs/1506.07365.

Philipp Wu, Alejandro Escontrela, Danijar Hafner, Ken Goldberg, and Pieter Abbeel. DayDreamer: World Models for Physical Robot Learning, 2022. URL http://arxiv.org/abs/2206.14176.

Tengyang Xie, Ching-An Cheng, Nan Jiang, Paul Mineiro, and Alekh Agarwal. Bellman-consistent pessimism for offline reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2106.06926.

Jie Xu, Viktor Makoviychuk, Yashraj Narang, Fabio Ramos, Wojciech Matusik, Animesh Garg, and Miles Macklin. Accelerated policy learning with parallel differentiable simulation. In *International Conference on Learning Representations (ICLR)*, 2022. URL https://arxiv.org/abs/2204.07137.

Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. MOPO: Model-based offline policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. URL https://arxiv.org/abs/2005.13239.

Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, and Chelsea Finn. COMBO: Conservative offline model-based policy optimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. URL https://arxiv.org/abs/2102.08363.

Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew J. Johnson, and Sergey Levine. SOLAR: Deep structured representations for model-based reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, 2019. URL https://arxiv.org/abs/1808.09105.

Gaoyue Zhou, Hengkai Pan, Yann LeCun, and Lerrel Pinto. DINO-WM: World models on pre-trained visual features enable zero-shot planning. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. URL https://arxiv.org/abs/2411.04983.

Luisa Zintgraf, Kyriacos Shiarlis, Maximilian Igl, Sebastian Schulze, Yarin Gal, Katja Hofmann, and Shimon Whiteson. VariBAD: A very good method for Bayes-adaptive deep RL via meta-learning. In *International Conference on Learning Representations (ICLR)*, 2020. URL https://arxiv.org/abs/1910.08348.

A Mathematical details for Section 2

This section provides proofs and helper statements to prove the statements of Section 2.

A.1 Separability and Anchoring

In this section, we study what anchoring at larger target sets $\mathcal{M}$ requires of an action-coordinate system (g, B). An action $a \in \ker B$ moves the state like any other, but equation 2 leaves g where it is, so the motion happens inside a level set of g. The states such actions reach from s form a set that a full system shrinks to $\{s\}$ and a system of rank $\ell < d$ does not. To anchor the chart g on a target set later, we need to understand which state null actions reach.

Definition 5 (Separating coordinates)**.** Let (g, B) be an action-coordinate system. Then $\mathcal{N}_B(s) := \{f(s, a_{1:n}) : n \geq 0,\ a_{1:n} \in (\mathcal{A} \cap \ker B)^n\}$ is the *null orbit* of s under B. The action-coordinate system is *separating* if for all $s, s' \in \mathcal{P}$, $g(s) = g(s')$ implies $s' \in \mathcal{N}_B(s)$.

A first easy consequence is that action-coordinate systems that are anchored at a single point must be full systems:

Lemma 1 (Anchored systems at singletons are full)**.** *Let (g, B) be an action-coordinate system anchored at $\mathcal{M} \subset \mathcal{P}$ with $|\mathcal{M}| = 1$. Then (g, B) is full.*

Proof. Assume that (g, B) is not full, then $\dim(\ker(B)) > 0$. Because 0 is in the interior of $\mathcal{A}$ by our basic assumptions, the set $\ker(B) \cap \mathcal{A} \setminus \{0\}$ is non-empty. Let $a \in \ker(B) \cap \mathcal{A} \setminus \{0\}$ and write $\mathcal{M} = \{s\}$, then $g^{-1}(0) = \{s\}$. Moreover, $g(f(s, a)) = g(s) + Ba = 0$ because $g(s) = 0$ and $Ba = 0$. Hence, $f(s, a) \in g^{-1}(0) = \{s_W\}$. Particularly $f(s_W, a) = s_W$ and thus a stands still, which contradicts the general assumption on f that only the zero action stands still. □

Proposition 1 (Anchoring)**.** *Let (g, B) be a separating action-coordinate system of rank ℓ and let $\emptyset \subsetneq \mathcal{M} \subseteq \mathcal{P}$. Then some translate of (g, B) is anchored at $\mathcal{M}$ if and only if $\mathcal{M} = \mathcal{N}_B(s)$ for some, and then every, $s \in \mathcal{M}$. In particular, if (g, B) is full, such a translate exists only for $|\mathcal{M}| = 1$.*

Proof. Write $\mathcal{M} = g^{-1}(c)$ for the translate $(g - c, B)$ and fix $s_* \in \mathcal{M}$. Let $\mathcal{M} = \mathcal{N}_B(s_*)$. Every chain out of s_* with its actions in $\ker B$ has its sum there too, so g is constant on $\mathcal{M}$ by Lemma 3, and $\mathcal{M} \subseteq g^{-1}(c)$ for $c := g(s_*)$; conversely $g(s') = c$ puts s' in $\mathcal{N}_B(s_*) = \mathcal{M}$ by Definition 5. For the other direction, let $\mathcal{M} = g^{-1}(c)$. Every $s' \in \mathcal{M}$ has $g(s') = g(s_*)$ and hence lies in $\mathcal{N}_B(s_*)$ by separation, while $\mathcal{N}_B(s_*) \subseteq g^{-1}(c)$ by Lemma 3. The choice of s_* is immaterial because $\mathcal{N}_B$ is constant along its own orbits. For the last statement, a full (g, B) has $\ker B = \{0\}$, which makes $\mathcal{N}_B(s) = \{s\}$ and collapses $\mathcal{M}$ to that one point. □

Example 1 (Separation is strictly weaker than reversibility)**.** Let $\mathcal{P} = \mathbb{R}^2$, $\mathcal{A} = [-1, 1]^2$ and

$$f(s, a) = \left(s_1 + a_1,\ (1 + a_1^2)\, s_2 + a_2\right),$$

a carriage on the first axis that stretches the second one by however far it travels in a step. Then $J \equiv I$, so f is fully actuated, and (g, B) with $g(s) := s_1$ and $B := (1\ \ 0)$ is a separating system of rank 1: its preimage are the vertical lines, which the null actions $(0, a_2)$ already sweep out. But f is not reversible, since at $s = (0, 1)$ and $a = b = (\frac{1}{2}, 0)$ the second coordinate of $f(f(s, a), b)$ is $\frac{25}{16}$ while that of $f(s, a + b)$ is 2.

Checking Definition 5 directly asks for the null orbits of B. Reversibility avoids that and makes every action-coordinate system of the dynamics separating at once.

A full system separates exactly when g is injective, since $\ker B = \{0\}$ makes $\mathcal{N}_B(s) = \{s\}$. Reversibility is sufficient for separation, but strictly stronger (Example 1), as the next statement shows:

Proposition 2 (Reversibility implies separation)**.** *Every action-coordinate system of a reversible dynamics is separating.*

Proof. Let f be reversible and write $\phi_a := f(\cdot, a)$ for $a \in \mathcal{A}$. Reversibility at $b = -a$, which is admissible because $\mathcal{A}$ is symmetric, gives $\phi_{-a} \circ \phi_a = \phi_0 = \mathrm{id}$ in both orders, so each ϕ_a is a bijection with inverse ϕ_{-a}, and $\phi_b \circ \phi_a = \phi_{a+b} = \phi_a \circ \phi_b$ whenever $a, b, a + b \in \mathcal{A}$. We first show that an action sequence summing to zero returns every state to itself. Let $\sum_{i=1}^{n} a_i = 0$. Because 0 is interior to $\mathcal{A}$, there is an N for which every sum of at most n of the vectors $a_1/N, \ldots, a_n/N$

lies in $\mathcal{A}$. Convexity puts ka_i/N in $\mathcal{A}$ for every $k \le N$, so reversibility applied $N-1$ times gives $\phi_{a_i} = \phi_{a_i/N}^N$ and hence

$$\phi_{a_n} \circ \cdots \circ \phi_{a_1} = \phi_{a_n/N}^N \circ \cdots \circ \phi_{a_1/N}^N.$$

The nN factors on the right commute pairwise by the choice of N, so they may be regrouped into N copies of $\phi_{a_n/N} \circ \cdots \circ \phi_{a_1/N}$, and each of those merges by reversibility, its partial sums lying in $\mathcal{A}$ for the same reason, into $\phi_{(\sum_i a_i)/N} = \phi_0$. The composition is therefore the identity.

Now let $g(s) = g(s')$. Lemma 2 gives an $a_{1:n}$ with $f(s, a_{1:n}) = s'$ and $\sum_i a_i \in \ker B$. Since 0 is interior to $\mathcal{A}$ and $\ker B$ is a subspace, there is an N for which $w := \frac{1}{N}\sum_i a_i$ lies in $\mathcal{A} \cap \ker B$. The chain $a_{1:n}$ and the N copies of w have the same sum, so concatenating the first with the reverse of the second yields a chain summing to zero, which is the identity by the previous paragraph. Inverting its N trailing factors gives $\phi_{a_n} \circ \cdots \circ \phi_{a_1} = \phi_w^N$, hence $s' = \phi_w^N(s) \in \mathcal{N}_B(s)$, which is separation. □

A.2 Proofs

This section provides proofs of the statements of Section 2. Note that if $\mathcal{A}$ is convex, then $\{\sum_{i=1}^n a_i \,:\, a_{1:n} \in \mathcal{A}^n\} = n\mathcal{A}$. This identity is used throughout this section.

Proofs of properties of action-coordinate systems

Lemma 2. *Let (g, B) be an action-coordinate system and let $s, s' \in \mathcal{P}$. Then $g(s) = g(s')$ if and only if $f(s, a_{1:n}) = s'$ for some $a_{1:n} \in \mathcal{A}^n$ with $\sum_{i=1}^n a_i \in \ker B$.*

Proof. If such an $a_{1:n}$ exists, Lemma 3 gives $g(s') = g(s) + B\sum_i a_i = g(s)$. Conversely, reachability provides an $a_{1:n}$ with $f(s, a_{1:n}) = s'$, and Lemma 3 turns $g(s) = g(s')$ into $B\sum_i a_i = 0$. □

Lemma 3. *Let (g, B) be an action-coordinate system. Then $g(f(s, a_{1:n})) = g(s) + B\sum_{i=1}^n a_i$ for every $s \in \mathcal{P}$ and every $a_{1:n} \in \mathcal{A}^n$, and $g(\mathcal{P}) = \mathbb{R}^\ell$.*

Proof. The identity is equation 2 applied n times. For the image, fix $s \in \mathcal{P}$. Every $a_{1:n} \in \mathcal{A}^n$ has $f(s, a_{1:n}) \in \mathcal{P}$, so the identity and convexity of $\mathcal{A}$ put $g(s) + n\,B \cdot \mathcal{A}$ inside $g(\mathcal{P})$ for every n. As 0 is interior to $\mathcal{A}$ and B has rank ℓ, the set $B \cdot \mathcal{A}$ contains a ball around 0, and the union of $g(s) + n\,B \cdot \mathcal{A}$ over n is all of $\mathbb{R}^\ell$. □

Proposition 3. *Let (g, B) and (g', B') be action-coordinate systems of rank ℓ and ℓ'. Then $\ker B \subseteq \ker B'$ if and only if there are $L \in \mathbb{R}^{\ell' \times \ell}$ and $v \in \mathbb{R}^{\ell'}$ with $g' = Lg + v$ and $B' = LB$. Conversely, $(Lg + v, LB)$ is an action-coordinate system of rank ℓ' for all $L \in \mathbb{R}^{\ell' \times \ell}$ of rank ℓ' and all $v \in \mathbb{R}^{\ell'}$.*

Proof. Let $\ker B \subseteq \ker B'$ and let B^+ be the pseudo-inverse of B. Then B^+B is the orthogonal projection onto $(\ker B)^\perp$, on which B' and $B'B^+B$ agree, while both vanish on $\ker B$, so $L := B'B^+$ satisfies $LB = B'$ and is the only such matrix because B is surjective. Fix $s_0 \in \mathcal{P}$ and put $v := g'(s_0) - Lg(s_0)$. For $s \in \mathcal{P}$, reachability gives an $a_{1:n}$ with $f(s_0, a_{1:n}) = s$, and Lemma 3 applied to both systems gives $g(s) = g(s_0) + B\sum_i a_i$ and $g'(s) = g'(s_0) + B'\sum_i a_i = g'(s_0) + LB\sum_i a_i$, hence $g' = Lg + v$. The other direction follows from $\ker B \subseteq \ker LB$, and the converse is immediate from equation 2. □

Proposition 4. *Let f be fully actuated and let (g, B) be an action-coordinate system of rank ℓ. Then g is smooth with $Dg(s) = B \cdot J^{-1}(s)$ for every $s \in \mathcal{P}$.*

Proof. Fix $s \in \mathcal{P}$. The map $a \mapsto f(s, a)$ is smooth on $\mathcal{A}$, sends 0 to s, and has derivative $J(s)$ there, which is invertible under full actuation. Since 0 lies in the interior of $\mathcal{A}$, the inverse function theorem restricts it to a diffeomorphism from an open $V \subseteq \mathcal{A}$ containing 0 onto an open neighbourhood of s. On that neighbourhood equation 2 reads $g = g(s) + B\,f(s, \cdot)^{-1}$, so g is smooth at s, and differentiating $g(f(s, a)) = g(s) + Ba$ at $a = 0$ gives $Dg(s)J(s) = B$. As B has rank ℓ and $J(s)$ is invertible, $Dg(s) = BJ^{-1}(s)$ has rank ℓ. □

Proof of Theorem 1. Let f be reversible and fully actuated and write $\phi_a := f(\cdot, a)$ for $a \in \mathcal{A}$, so that Proposition 7 makes it flow-generated. Smoothness of J makes the solution of equation 13 and equation 14 unique, and $\phi_{(t+h)a} = \phi_{ha} \circ \phi_{ta}$ makes $t \mapsto \phi_{ta}(s)$ solve both, so ϕ_{ta} is the time-t flow of $J(\cdot)a$. Reversibility gives $\phi_b \circ \phi_a = \phi_{a+b} = \phi_a \circ \phi_b$ whenever $a, b, a+b \in \mathcal{A}$, so two of these flows commute, and flows of smooth vector fields commute if and only if their Lie bracket vanishes (Lee, 2012, Chapter 9), so $[J(\cdot)a, J(\cdot)b] = 0$ for all $a, b \in \mathcal{A}$. As 0 is interior to $\mathcal{A}$, bilinearity turns this into equation 16, and Proposition 8 provides a full action-coordinate system.

For the converse, let (g, B) be a full action-coordinate system, so that $B \in \mathbb{R}^{d\times d}$ is invertible and, by Proposition 4, g is smooth with $Dg(s) = BJ^{-1}(s)$ invertible for every $s \in \mathcal{P}$. Write $D_s f$ and $D_a f$ for the derivatives of f in its first and in its second argument. The remaining argument reads equation 2 twice: differentiated in a it makes $t \mapsto f(s, ta)$ follow the field $J(\cdot)a$, differentiated in s it makes f carry that field along, and together the two turn $t \mapsto f(f(s, ta), tb)$ into a solution of the same initial value problem as $t \mapsto f(s, t(a+b))$. Both derivatives are taken in the interior of $\mathcal{A}$ and extend to its boundary by continuity, which suffices below, since convexity of $\mathcal{A}$ puts ta in that interior for every $a \in \mathcal{A}$ and every $t \in [0, 1)$.

Differentiating equation 2 with respect to a gives $Dg(f(s,a))\, D_a f(s,a) = B$ and hence

$$D_a f(s,a) = Dg(f(s,a))^{-1} B = J(f(s,a)), \tag{9}$$

where the second equality is $Dg = BJ^{-1}$ at $f(s,a)$. Differentiating equation 2 with respect to s gives $Dg(f(s,a))\, D_s f(s,a) = Dg(s)$ and hence

$$D_s f(s,a)\, J(s) b = J(f(s,a))\, b \qquad \text{for all } b \in \mathbb{R}^d, \tag{10}$$

because both sides have the image Bb under the invertible $Dg(f(s,a))$, by $Dg(s)J(s) = B$ on the left and by $Dg(f(s,a))J(f(s,a)) = B$ on the right.

Let now $s \in \mathcal{P}$ and let $a, b \in \mathcal{A}$ with $a+b \in \mathcal{A}$. The curves $\sigma(t) := f(s, t(a+b))$, $q(t) := f(s, ta)$ and $\Psi(t) := f(q(t), tb)$ are defined on $[0,1]$ because $\mathcal{A}$ is convex with $0 \in \mathcal{A}$, and $\sigma(0) = \Psi(0) = s$. By equation 9, $\dot\sigma(t) = D_a f(s, t(a+b))(a+b) = J(\sigma(t))(a+b)$ and $\dot q(t) = J(q(t))a$, so that f is flow-generated in the sense of Definition 7. The chain rule, equation 10 at the position $q(t)$ and equation 9 then give

$$\dot\Psi(t) = D_s f(q(t), tb)\, \dot q(t) + D_a f(q(t), tb)\, b = J(\Psi(t))\, a + J(\Psi(t))\, b = J(\Psi(t))(a+b).$$

The field $J(\cdot)(a+b)$ is smooth, so σ and Ψ solve the same initial value problem and agree on $[0,1)$, and at $t = 1$ by continuity. This is $f(f(s,a), b) = f(s, a+b)$, which is reversibility. □

Proofs of step-optimality statements

Proof of Theorem 2. First, let

$$N(s, W) = \min\{n \geq 0 : \exists\, a_{1:n} \in \mathcal{A}^n \text{ with } f(s, a_{1:n}) \in \mathcal{M}_W\}$$

be the minimal number of steps needed to reach the target $\mathcal{M}_W$ from $s \in \mathcal{P}$. As $N(s, W)$ is a minimum over *all* action sequences, it is the optimal number of steps. Write $g := g_W$, $B := B_W$ and $z := g(s)$, and note that the minimum defining π^*_B is attained because $\mathcal{A}$ is compact and $a \mapsto \|z + Ba\|_B$ is continuous. We first compute $N(s, W)$. Lemma 3 gives $g(f(s, a_{1:n})) = z + B\sum_i a_i$, whose cumulative actions run over $n\mathcal{A}$ as $\mathcal{A}$ is convex, so the latents reachable in n steps are exactly $z + n\, B \cdot \mathcal{A}$. As $\mathcal{M}_W = g^{-1}(0)$ by Definition 3, a chain of length n steers s into $\mathcal{M}_W$ if and only if $0 \in z + n\, B \cdot \mathcal{A}$, which by symmetry of $B \cdot \mathcal{A}$ and homogeneity of $\|\cdot\|_B$ holds if and only if $\|z\|_B \leq n$. Hence $N(s, W) = \lceil \|z\|_B \rceil$. We next show that one step of π^*_B reduces the norm by one,

$$\min_{a \in \mathcal{A}} \|z + Ba\|_B = \max\{0, \|z\|_B - 1\}. \tag{11}$$

By definition of the gauge, $B \cdot \mathcal{A}$ is the closed unit ball of $\|\cdot\|_B$, so Ba ranges over $\{u : \|u\|_B \leq 1\}$ as a ranges over $\mathcal{A}$. The triangle inequality gives $\|z + u\|_B \geq \|z\|_B - 1$ for every such u, and the value is attained by $u = -z/\|z\|_B$ when $\|z\|_B \geq 1$ and by $u = -z$ otherwise, which also settles the case $\|z\|_B < 1$. Writing z_k for the latent after k steps of π^*_B from $z_0 := z$, repeated use of equation 11 gives $\|z_k\|_B = \max\{0, \|z\|_B - k\}$, so $z_k = 0$ for $k = \lceil \|z\|_B \rceil$. The policy therefore reaches $g^{-1}(0) = \mathcal{M}_W$ in $N(s, W)$ steps, and by the first paragraph no action sequence reaches it sooner. □

Lemma 4. *Let $B \in \mathbb{R}^{\ell \times d}$ be of rank ℓ and $z \in \mathbb{R}^{\ell}$, then $\|z\|_B = \min\{\|a\|_I : a \in \mathbb{R}^d,\ Ba = z\}$.*

Proof. The set $\{a \in \mathbb{R}^d : Ba = z\}$ is a non-empty closed affine subspace because B has rank ℓ, and the sublevel sets of the norm $\|\cdot\|_I$ are compact, so the infimum m of $\|\cdot\|_I$ over that set is attained and the right hand side of the equation of the statement is a minimum. First, let $t > 0$ with $z \in t\,B \cdot \mathcal{A}$, so that $z = t\,Ba$ for an $a \in \mathcal{A}$. Then ta solves $B(ta) = z$ and has $\|ta\|_I = t\|a\|_I \leq t$, since $\mathcal{A}$ is the closed unit ball of $\|\cdot\|_I$. Taking the infimum over all such t gives $m \leq \|z\|_B$. Second, let $a \in \mathbb{R}^d$ with $Ba = z$ and put $t := \|a\|_I$. For $t > 0$, the same unit ball gives $a/t \in \mathcal{A}$, hence $z = t\,B(a/t) \in t\,B \cdot \mathcal{A}$ and $\|z\|_B \leq t$. For $t = 0$, the action is $a = 0$ and so is z. Taking the minimum over all such a gives $\|z\|_B \leq m$. □

Proof of Corollary 1. Write $B := B_W$. It suffices to show that $\pi_{B,0}(z)$ attains the minimum defining $\pi^*_B(z)$ for every latent z, since Theorem 2 then applies to it. The minimizer need not be unique, because $B \cdot \mathcal{A}$ is a polytope whenever $\mathcal{A}$ is one and $\|\cdot\|_B$ is then not strictly convex. A full system has B invertible, so $\pi_{B,0}(z) = \mathrm{scale}_{\mathcal{A}}(-B^{-1}z)$ and $-B^{-1}z$ is the only solution of $Ba = -z$, which by Lemma 4 and the symmetry of $\|\cdot\|_B$ gives $\|-B^{-1}z\|_I = \|-z\|_B = \|z\|_B$. The scaling therefore divides $-B^{-1}z$ by $\max\{1, \|z\|_B\}$, and the latent it reaches is

$$z + B\,\pi_{B,0}(z) = z - \frac{z}{\max\{1, \|z\|_B\}} = z\Big(1 - \frac{1}{\max\{1, \|z\|_B\}}\Big),$$

whose norm is $\|z\|_B - 1$ for $\|z\|_B \geq 1$ and 0 otherwise, which is equation 11. □

Lemma 5. *Every action-coordinate system where R is radial is R-anchored.*

Proof. Recall that we assume $\mathcal{P}$ to be open and thus g is surjective onto $\mathbb{R}^{\ell}$ by Lemma 3. Let (g, B) be an action-coordinate system for W in which R is radial with profile ρ_W. Particularly, there is an $s_0 \in \mathcal{P}$ with $g(s_0) = 0$, and every $s \in \mathcal{P}$ has $\|g(s)\|_B \geq 0 = \|g(s_0)\|_B$. The profile is strictly decreasing, so $R(s_0, W) = \rho_W(0)$ is the maximum of $R(\cdot, W) = \rho_W(\|g(\cdot)\|_B)$, and an s attains it if and only if $\|g(s)\|_B = 0$, which for a norm is $g(s) = 0$. Hence $\mathcal{M}_W = g^{-1}(\{0\})$ and it is non-empty. □

Proof of Corollary 2. Fix a threshold $C \leq \rho_W(0)$ and $s \in \mathcal{P}$, write $z := g(s)$, and let

$$N_C(s, W) := \min\{n \geq 0 : \exists\, a_{1:n} \in \mathcal{A}^n \text{ with } f(s, a_{1:n}) \in \mathcal{T}_C\}$$

be the minimal number of steps from s into $\mathcal{T}_C$. As it is a minimum over *all* action sequences, it is the optimal number of steps, and we show that

$$N_C(s, W) = \lceil \max\{0, \|z\|_B - r\} \rceil \tag{12}$$

and that π^*_B attains it. As in the proof of Theorem 2, Lemma 3 and convexity of $\mathcal{A}$ make the latents reachable from z in n steps exactly $z + n\,B \cdot \mathcal{A} = \{y \in \mathbb{R}^{\ell} : \|y - z\|_B \leq n\}$, so n steps steer s into $\mathcal{T}_C$ if and only if $\|z + u\|_B \leq r$ for some u with $\|u\|_B \leq n$. Such a u exists if and only if $\|z\|_B \leq n + r$: the triangle inequality gives one direction, and $u := -z$ for $\|z\|_B \leq n$, respectively $u := -n\,z/\|z\|_B$ otherwise, the other. As the number of steps is a non-negative integer, this is equation 12. Writing z_k for the latent after k steps of π^*_B from $z_0 := z$, repeated use of equation 11, which every selection of π^*_B attains, gives $\|z_k\|_B = \max\{0, \|z\|_B - k\}$. This first drops to r at $k = N_C(s, W)$, where π^*_B therefore reaches $\mathcal{T}_C$, and by the first part no action sequence reaches it sooner. □

A.3 Remarks

Remark 1 ($\pi_{B,0}$ not optimal if B not full). Clearly, the policy $\pi_{B,\varepsilon}$ exists even if B is not full, but it is not guaranteed to be step-optimal in that case. Consider $B = (1\ \ 3)$ and $\mathcal{A} = [-1, 1]^2$. At the latent $z = 4$, the action $a = (-1, -1)$ reaches the target 0 in precisely one step. On the other hand, $\pi_{B,0}$ applies $\mathrm{scale}_{\mathcal{A}}(-0.4, -1.2)$ which costs two steps due to the rescaling to the box $\mathcal{A}$. The difference comes from the fact that $\pi_{B,0}$ takes the Euclidean-smallest solution of $Ba = -z$ and scales it to $\mathcal{A}$, which is not necessarily the $\|\cdot\|_I$-smallest solution. For a full system, the solution set is the single point $-B^{-1}z$, on which every norm agrees.

Remark 2 (Closed-forms of π_B^*). If B is invertible, then π_B^* admits the closed-form $\pi_{B,0}$ of Corollary 1. If B is not invertible, no closed-form expression is guaranteed a priori. This is because an optimal action is a $\|\cdot\|_I$-smallest solution of $Ba = -z$ rescaled to $\mathcal{A}$: that solution has norm $\|z\|_B$ by Lemma 4, so the rescaled action lies in $\mathcal{A}$ and attains equation 11. The solutions of $Ba = -z$ form an affine subspace of dimension $d-\ell$, and which of its points is $\|\cdot\|_I$-smallest is decided by $\mathcal{A}$. For an ellipsoid $\mathcal{A} = \{a \in \mathbb{R}^d : a^\top Q a \le 1\}$ with Q positive definite, we have $\|a\|_I = \sqrt{a^\top Q a}$ and Lagrange multipliers give the Q-weighted pseudoinverse

$$\pi_B^*(z) = \mathrm{scale}_{\mathcal{A}}\left(-Q^{-1}B^\top\left(BQ^{-1}B^\top\right)^{-1}z\right),$$

which for $Q = I$ is $\pi_{B,0}$, so a Euclidean ball of actions makes the pseudoinverse policy step-optimal at every rank. For the box $\mathcal{A} = [-1,1]^d$, we have $\|a\|_I = \|a\|_\infty$ and computing π_B^* essentially requires computing a solution of the linear program $\min_{a\in\mathbb{R}^d,\ t\in\mathbb{R}}\{t : Ba = -z,\ -t\mathbf{1} \le a \le t\mathbf{1}\}$. For $\ell \ge 2$ the program has no closed-form solution in general.

Remark 3 (Mismatched termination sets). Radiality in $\|\cdot\|_B$ is what makes $\mathcal{T}_C$ a ball of the norm that counts steps. A reward radial in another norm gives a termination set of a different shape, and only a bracket survives: if $r_-\, B\cdot\mathcal{A} \subseteq g(\mathcal{T}_C) \subseteq r_+\, B\cdot\mathcal{A}$, the same two steps of the proof give

$$\left\lceil \max\{0, \|z\|_B - r_+\}\right\rceil \ \le\ N_C(s,W)\ \le\ \left\lceil \max\{0, \|z\|_B - r_-\}\right\rceil,$$

whose upper bound π_B^* again attains, so it spends at most $\lceil r_+ - r_-\rceil$ steps over the optimum. A Euclidean profile against $B\cdot\mathcal{A} = [-1,1]^\ell$ is the canonical mismatch, where $r_+/r_- = \sqrt{\ell}$.

Remark 4 (Dynamics with drift). Let $f(s,a) := f'(s,a) + b$ for a driftless f' and a fixed $b \in \mathbb{R}^m \setminus \{0\}$, the model of a system that keeps moving when no command is sent, such as an axis under a gravity load. Since 0 lies in the interior of $\mathcal{A}$, putting $a = 0$ in equation 2 gives

$$g(s+b) = g(f(s,0)) = g(s) \qquad \text{for all } s \in \mathcal{P},$$

so a chart of f is blind along the drift and constant on the orbit $s + n\,b$, $n \ge 0$, which the zero action walks. This does not forbid a chart, it forces the drift into the kernel of the gain. On $\mathcal{P} = \mathbb{R}^d$ with the canonical $f'(s,a) = s + Wa$ of Example 2 for an invertible $W \in \mathbb{R}^{d\times d}$, that periodicity turns equation 2 into $g(s+Wa) = g(s) + B\cdot a$, so (g,B) is an action-coordinate system for f' as well and Proposition 3 gives $g = BW^{-1}(\cdot - s_0) + v$ for some $s_0 \in \mathcal{P}$ and $v \in \mathbb{R}^\ell$. The condition then reads $W^{-1}b \in \ker B$, which lower ranks satisfy: for $W = I$ and $b = (0,1)^\top$ on $\mathbb{R}^2$, the pair $g(s) := s_1$ and $B := (1\ \ 0)$ is a rank-1 action-coordinate system of a drifted dynamics. What a drift does rule out is a full system, where $\ker B = 0$ forces $b = 0$, and, for any drifted f, an anchored one: $g^{-1}(0)$ is invariant under $s \mapsto s + b$ and hence unbounded as soon as it is non-empty, so it is never a bounded target set $\mathcal{M}_W$ (Definition 3). A drifted dynamics therefore has charts only where they are useless for control, with π_B^* holding the latent at 0 while the position leaves along b at every step. The environments of Section 1.2 are all driftless: with the zero action sent, every one of them stays where it is.

A.4 Examples

Example 2 (Undistorted movements)**.** Let $f(s,a) = s + Wa$ be the canonical example of Section 1.1, with an invertible $W \in \mathbb{R}^{d\times d}$, and let $s_0 \in \mathcal{P}$. Then (g,B) with $g(s) := s - s_0, \qquad B := W$ is a full action-coordinate system, since $g(f(s,a)) = s + Wa - s_0 = g(s) + B\cdot a$ for all s and a. Composing it with a $C \in \mathbb{R}^{\ell\times d}$ of rank ℓ gives the rank-ℓ system $(C(\cdot - s_0), CW)$, which reads ℓ linear combinations of the position and ignores the rest.

Example 3 (Multiplicative movements)**.** Let $\mathcal{P} = (0,\infty)$, let $f(s,a) = s\,e^a$ and let $s_0 > 0$. The pair (g,B) with $g(s) := \log\frac{s}{s_0}$ and $B := 1$ is a full action-coordinate system, since $g(f(s,a)) = \log\frac{s\cdot e^a}{s_0} = g(s) + B\cdot a$ for all s and a.

Example 4 ($L_f = 0$ with non-linear f)**.** Let $\mathcal{P} = \mathbb{R}^2$, let $h\colon \mathbb{R}\to\mathbb{R}$ be smooth and let

$$f(s,a) = \left(s_1 + a_1,\ s_2 + a_2 + h(s_1+a_1) - h(s_1)\right).$$

This can be interpreted as a carriage that carries a tool, where the carriage travels along the first axis with state s_1 controlled by a_1 and the tool has height s_2 along the second axis that is raised by a_2 relative to the carriage. Now, the carriage rides a ramp described by h, so advancing it by a_1 lifts

the tool by $h(s_1 + a_1) - h(s_1)$ in addition to what a_2 contributes. For a base point $s_0 = (p, q)$, the pair (g, B) with

$$g(s) := (s_1,\ s_2 - h(s_1) + h(p)) - s_0$$

and $B = I$ is a full action-coordinate system, since the two copies of $h(s_1 + a_1)$ cancel in $g(f(s, a)) = g(s) + a$. It is easy to see that g is injective and hence $L_f = 0$ by Corollary 3. However, for $h(t) = t^2/2$ the dynamics is $f(s, a) = \left(s_1 + a_1,\ s_2 + a_2 + s_1 a_1 + \frac{a_1^2}{2}\right)$ which is affine in neither s nor a.

Example 5 (A one-dimensional target $\mathcal{M}_W$)**.** Reconsider the dynamics from Example 4 and assume the reward penalizes the deviation of the tool's height above the ramp from a desired clearance $c > 0$. Specifically, $R(s_1, s_2) := -(s_2 - h(s_1) - c)^2$. The set of maximizers of R is

$$\mathcal{M} := \{(s_1, s_2) \in \mathbb{R}^2 : s_2 = h(s_1) + c\},$$

which is a connected 1-manifold, and (g, B) with $g(s) := s_2 - h(s_1) - c$ and $B := (0\ \ 1)$ is an action-coordinate system of rank $\ell = 1$ anchored at $\mathcal{M}$. It is separating, because $\ker B = \mathbb{R} \times \{0\}$ and the carriage actions $f(s, (t, 0)) = (s_1 + t,\ s_2 + h(s_1 + t) - h(s_1))$ already sweep out the preimage through s. They move the state at every step, as the standing assumptions require, and they move it along $\mathcal{M}$, which is therefore the whole $\ker B$-orbit of each of its points and anchored in the sense of Proposition 1. Figure 3b shows the reward in both coordinate systems.

Example 6 (Radial reward)**.** Consider a d-dimensional version of f from Example 3, that is: $\mathcal{P} = (0, \infty)^d$ and $f(s, a) = (s_1 e^{a_1}, \ldots, s_d e^{a_d})$ with the reward

$$R(s, W) := \exp\Big(- \max_{i=1,\ldots,d} \log^2 \tfrac{s_i}{s_{W,i}} \Big),$$

which is maximal at s_W. It is clearly not symmetric with respect to s_W. Example 3 gives the action-coordinates $g(s) = \big(\log(s_1/s_{W,1}), \ldots, \log(s_d/s_{W,d})\big)$ with $B = I$, so that $\|\cdot\|_B = \|\cdot\|_\infty$ for the box $\mathcal{A} = [-1, 1]^d$. Substituting it into R yields $\hat{R}(z) = \exp\big(-\|z\|_\infty^2\big)$, i.e., a radial reward in (g, I) with $\rho(t) := e^{-t^2}$.

Example 7 (Full but not fully actuated)**.** Let $\mathcal{P} = \mathbb{R}^3$ and $\mathcal{A} = [-1, 1]^2$ and let

$$f(s, a) := (s_1 + a_1,\ s_2 + a_2,\ s_3 + s_1 a_2 - s_2 a_1)\,.$$

The pair (g, B) with $g(s) := (s_1, s_2)$ and $B := I$ is an action-coordinate system, since

$$g(f(s, a)) = (s_1 + a_1,\ s_2 + a_2) = g(s) + Ba$$

for all $s \in \mathcal{P}$ and $a \in \mathcal{A}$, and it is full, as B has rank $\ell = 2 = d$. The dynamics is not fully actuated, however, since its actuation Jacobian

$$J(s) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ -s_2 & s_1 \end{pmatrix}$$

has $m = 3$ rows and $d = 2$ columns and hence is not invertible although its rank is d everywhere. Being full is a property of the rank of B and leaves m free, and the two notions meet only for $m = d$. Consequently, (g, B) is not separating, as g identifies the whole line $\{(s_1, s_2)\} \times \mathbb{R}$ while $\ker B = \{0\}$ makes the null orbit $\mathcal{N}_B(s)$ the singleton $\{s\}$. Proposition 2 is not contradicted, since f is not reversible,

$$f(f(s, a), b) - f(s, a + b) = (a_1 b_2 - a_2 b_1)\, e_3,$$

even though a single action is still undone by its negative, $f(f(s, a), -a) = s$.

A.5 Connection to linear placement errors

The definition of reversibility given in Definition 2 is a special case of (Schmähling et al., 2026, Definition 3.4). There, a dynamics has *linear placement-errors*, or LPE, if there is a constant $L \geq 0$ with $\|f\left(s, \sum_{i=1}^n a_i\right) - f(s, a_{1:n})\| \ \leq\ L \sum_{i=1}^n \|a_i\|$ for every $s \in \mathcal{P}$ and every $a_{1:n} \in \mathcal{A}^n$ with $\sum_{i=1}^n a_i \in \mathcal{A}$. We write L_f for the smallest such constant.

Intuitively, L_f bounds how far the endpoint of a chain drifts from the endpoint of its summed action, relative to the length of the chain. As LPE with $L_f = 0$ for $n = 2$ is exactly reversibility, it follows from Proposition 2 that if $L_f = 0$ every action-coordinate system is separating. For full action-coordinate systems, the converse is also true. The condition $L_f = 0$ should not be confused with f being linear, instead still can cover arbitrarily complex non-linearities (see Example 4).

Corollary 3 (LPE and separation)**.** *Let f have LPE and let (g, B) be a full action-coordinate system. Then $L_f = 0$ if and only if (g, B) is separating.*

Proof of Corollary 3. The condition $L_f = 0$ for a sequence of length $n = 2$ makes f reversible in the sense of Definition 2, so Proposition 2 gives the first claim. For a full system, separation is the injectivity of g, and Lemma 3 together with equation 2 gives $g(f(s, a_{1:n})) = g(s) + B\sum_i a_i = g(f(s, \sum_i a_i))$ whenever $\sum_i a_i \in \mathcal{A}$, which injectivity turns into $f(s, a_{1:n}) = f(s, \sum_i a_i)$, that is, $L_f = 0$. □

A.6 Lifted variant

The lifted variant $z_{t+1} = z_t + B_W \phi_\eta(z_t, a_t)$, in which a learned feature map ϕ_η replaces the raw action, is the residual form of a model both lines already study: Proctor et al. (2018) lift state and input jointly into the dictionary, and the bilinear realization of Bruder et al. (2021) lets the input multiply the lifted state. Leaving the features to a network and the coefficients that multiply them to a per-context fit is equally standard in learning-based adaptive control, where O'Connell et al. (2022) adapt such coefficients online on top of a basis shared across conditions. We keep $\phi_\eta(z_t, a_t) = a_t$ nonetheless. The lifted variant leaves the ridge estimate equation 4 intact, regressing the latent increments on $\phi_\eta(z_t, a_t)$ instead of on a_t, but it costs the two properties the construction rests on. The gain keeps its shape only if ϕ_η maps into $\mathbb{R}^d$, which the determinant in equation 6 and the inversion in equation 3 presuppose; even for a square gain, planning has to invert $\phi_\eta(z, \cdot)$ numerically. Since ϕ_η reads the latent, the increments no longer telescope, so the all-pairs identity

$$z_j - z_i = B_W \cdot \textstyle\sum_{t=i}^{j-1} a_t, \qquad k - 1 \le i < j \le T,$$

holds only along a rollout.

B Realizability of action-coordinate systems

This section provides insights when encoders can realize action-coordinate systems. We take a mathematical perspective and for simplicity, we remove stochasticity in the observation function O here. Moreover, we assume that $\mathcal{M}_W = \{s_W\}$ is a singleton for every $W \in \mathcal{W}$. In that case, it follows automatically that B_W of every R-anchored action-coordinate system (g_W, B_W) is square and invertible (see Lemma 1). The next definition formalizes the notion of realizability of action-coordinate systems, which is the main object of study in this section.

Definition 6 (Realizable action-coordinates)**.** A family $(g_W)_{W\in\mathcal{W}}$ of R-anchored maps $\mathcal{P} \to \mathbb{R}^d$ is *realizable at horizon* k if there are a map e and an open dense $U \subseteq \mathcal{A}^{k-1}$ such that

$$e\big(h(W, s, a_{1:k-1})\big) = g_W(s_k)$$

for every $W \in \mathcal{W}$, every $s \in \mathcal{P}$ and every $a_{1:k-1} \in U$, and an action-coordinate family $(g_W, B_W)_{W\in\mathcal{W}}$ is *k-realizable* if $(g_W)_{W\in\mathcal{W}}$ is. We define the minimal such k as

$$k_{\mathrm{rel}} := \min\{k \ge 1 : (g_W, B_W)_{W\in\mathcal{W}} \text{ is } k\text{-realizable}\},$$

In addition, define the minimal horizons

$$k_{\mathrm{fit}} := \min\{k \ge 1 : \exists (g_W)_{W\in\mathcal{W}} \text{ of some action-coordinate family is } k\text{-realizable}\},$$
$$k_{\mathrm{pin}} := \min\{k \ge 1 : \exists (g_W)_{W\in\mathcal{W}} \text{ of an action-coordinate family with } B_W = I \text{ is } k\text{-realizable}\},$$

all ∞ if no such k exists.

The reason for the open dense U in Definition 6 is that the encoder can only see the actions through their effect on the observations, and some actions may not move the system at all. For instance, in the extreme case that $a_i = 0$ for $1 \le i \le k$, the same observation repeats k times. However, requiring $U = \mathcal{A}^{k-1} \setminus \{0\}$ does not suffice, as we show in Example 8, where U has to satisfy algebraic constraints. The example also shows that a history of $k > 1$ is necessary already in fairly simple dynamics.

Example 8 (Why $k = 1$ does not suffice in general)**.** Consider a 1-dimensional space $\mathcal{P} = \mathbb{R}$ with context $W = (w_1 \quad w_2)$, $w_1 \neq 0$, $f(s, a, W) = s + w_1 \cdot a$ and $s_W = w_2$. Moreover, assume that $O(s, W) = (s - w_2)^2$, which is the same symmetry that we expect of O in the environments of Section 1.2, where a misalignment of either sign degrades the image in nearly the same way. A pair of action-coordinates (g_W, B_W) is given by $g_W(s) = (s - w_2)/w_1$ and $B_W = 1$ and because the actuation Jacobian $J \equiv w_1$ is constant, thus Proposition 8 gives that the action-coordinate systems of W anchored at $\{w_2\}$ are the pairs $(b \cdot g_W, b \cdot B_W)$ with $b \neq 0$. Reading the observation in action-coordinates and substituting $z := b \cdot g_W(s)$ gives

$$O(s, W) = O((b \cdot g_W)^{-1}(z), W) = \left(\frac{z w_1}{b}\right)^2$$

Thus, mathematically, it is impossible to learn the action-coordinate z from $O(s, W)$ alone and hence this system is not realizable at horizon $k = 1$. For $k = 3$, however, the encoder can identify the action-coordinates, for instance the pair for $b = 1$. To see this, write $o_i := O(s_i, W)$ for the three observations of a window $h(W, s_1, a_{1:2})$, whose positions are $s_2 = s_1 + w_1 a_1$ and $s_3 = s_2 + w_1 a_2$. The observed increments are quadratic in the context but linear in the pair $u := w_1(s_1 - w_2)$ and $v := w_1^2$,

$$\begin{aligned} o_2 - o_1 &= 2a_1\, u + a_1^2\, v, \\ o_3 - o_2 &= 2a_2\, u + \left(2a_1 a_2 + a_2^2\right) v, \end{aligned}$$

a system with determinant $2a_1 a_2(a_1 + a_2)$. Whenever the two actions and their sum are non-zero it identifies u and v and these can be analytically derived from $h(W, s_1, a_{1:2})$, like using Cramer's rule. Thus,

$$e(h(W, s_1, a_{1:2})) := \frac{u}{v} + a_1 + a_2 = \frac{w_1(s_1 - w_2)}{w_1^2} + a_1 + a_2 = g_W(s_1) + a_1 + a_2 = g_W(s_3)$$

returns the action-coordinates of the state the window ends at. In the language of Definition 6, the action-coordinate system has $k_{\text{rel}} = 3$.

The gap between k_{fit} and k_{pin} is what a window pays for carrying the gain along with the position. That price is at most d observations, and the argument is the ridge estimate equation 4 of the method itself, evaluated inside a single window instead of once per episode.

Proposition 5. *Let $k_{\text{fit}} < \infty$. Then $k_{\text{fit}} \leq k_{\text{pin}} \leq k_{\text{fit}} + d$.*

Proof. The family with $B_W = I$ is one of the families the minimum defining k_{fit} ranges over, which gives $k_{\text{fit}} \leq k_{\text{pin}}$.

To show $k_{\text{pin}} \leq k_{\text{fit}} + d$, let the action-coordinate family $(g_W, \tilde{B}_W)_{W \in \mathcal{W}}$ be realizable at horizon k_{fit} by a map e and an open dense $U \subseteq \mathcal{A}^{k_{\text{fit}}-1}$. Fix a window $h := h(W, s, a_{1:k_{\text{fit}}+d-1})$, with positions $s_1, \ldots, s_{k_{\text{fit}}+d}$ and observations $o_1, \ldots, o_{k_{\text{fit}}+d}$. For $0 \leq j \leq d$, the entries $(o_{1+j}, \ldots, o_{k_{\text{fit}}+j})$ and $a_{1+j:k_{\text{fit}}+j-1}$ of h are the window $h(W, s_{1+j}, a_{1+j:k_{\text{fit}}+j-1})$, so

$$e_j(h) := e(o_{1+j}, \ldots, o_{k_{\text{fit}}+j}, a_{1+j:k_{\text{fit}}+j-1}) = g_W(s_{k_{\text{fit}}+j})$$

whenever the j-th block lies in U. By equation 2 the consecutive differences are $e_{j+1}(h) - e_j(h) = \tilde{B}_W a_{k_{\text{fit}}+j}$ for $0 \leq j \leq d-1$, so, writing $A(h)$ for the matrix with columns $a_{k_{\text{fit}}}, \ldots, a_{k_{\text{fit}}+d-1}$ and $D(h)$ for the matrix of those differences, $D(h) = \tilde{B}_W A(h)$ and hence $\tilde{B}_W = D(h)A(h)^{-1}$ wherever $A(h)$ is invertible. The map

$$h \mapsto \left(D(h)A(h)^{-1}\right)^{-1} e_d(h) = \tilde{B}_W^{-1} g_W(s_{k_{\text{fit}}+d})$$

is therefore defined on the windows whose $d+1$ blocks lie in U and whose $A(h)$ is invertible, an open dense subset of $\mathcal{A}^{k_{\text{fit}}+d-1}$ as a finite intersection of open dense sets. By Proposition 3 its value is $B_W^{-1} g_W(s_{k_{\text{fit}}+d})$ for any R-anchored action-coordinate system (g_W, B_W) of W, and the converse of that proposition makes $(B_W^{-1} g_W, I)$ one as well, so the value is the action-coordinate of $s_{k_{\text{fit}}+d}$ in the anchored system with gain I. That family is thus realizable at horizon $k_{\text{fit}} + d$. As k_{pin} is the minimum of all such horizons, $k_{\text{pin}} \leq k_{\text{fit}} + d$. □

The next proposition establishes that when the observation is an injective derived quantity of the offset to the target, then k_{fit} and k_{pin} attain the widest possible gap Proposition 5 allows. In these cases, particularly, fitting B *in every episode* allows the encoder to be memoryless.

Proposition 6. *Let $d \geq 1$ and let $f(s, a, W) := s + A_W a$ with context (A_W, s_W) consisting of an invertible $A_W \in \mathbb{R}^{d\times d}$ and a target $s_W \in \mathbb{R}^d$. Moreover, let $O(s, W) := \iota(s - s_W)$ with $\iota\colon \mathbb{R}^d \to \mathcal{O}$ be injective. Then $k_{\text{fit}} = 1$ and $k_{\text{pin}} = d + 1$.*

Proof. Clearly, the pair (g_W, B_W) with $g_W(s) := s - s_W$ and $B_W := A_W$ is an action-coordinate system for every context, and it is R-anchored because $g_W^{-1}(0) = \{s_W\} = \mathcal{M}_W$. A window of length 1 is the single observation $o_1 = \iota(s - s_W)$, from which the inverse ι^{-1} on the image of ι recovers that coordinate, so the encoder $e(o_1) := \iota^{-1}(o_1)$ realizes $(g_W)_{W\in\mathcal{W}}$ at horizon 1 and gives $k_{\text{fit}} = 1$. The remaining claim concerns the family

$$g'_W(s) := A_W^{-1}(s - s_W),$$

which satisfies $g'_W(s + A_W a) = g'_W(s) + a$ and is anchored at $\{s_W\}$, so (g'_W, I) is an R-anchored action-coordinate system with gain I. Its chart carries all the context and its gain none. It is the only one: if $(\tilde{g}_W, I)$ is another, then $\tilde{g}_W(s_W) = 0$ by anchoring, and every $s \in \mathcal{P}$ is reachable from s_W, as $a_i := A_W^{-1}(s - s_W)/n$ lies in $\mathcal{A}$ for n large and steers s_W to s in n steps. Lemma 3 then gives $\tilde{g}_W(s) = \sum_i a_i = A_W^{-1}(s - s_W)$, that is, $\tilde{g}_W = g'_W$. Hence k_{pin} is the smallest horizon at which $(g'_W)_{W\in\mathcal{W}}$ is realizable.

Our strategy for the lower bound $k_{\text{pin}} \geq d + 1$ is to exhibit, for every horizon $k \leq d$, two contexts that produce one and the same window but disagree on the coordinate of the state it ends at, which no map can return. Let $1 \leq k \leq d$, let e be a map and let $U \subseteq \mathcal{A}^{k-1}$ be open dense. Density makes U non-empty, so fix $a_{1:k-1} \in U$ and a context $W = (A_W, s_W)$, and consider

$$S := \operatorname{span}(a_1, \ldots, a_{k-1}).$$

Being spanned by $k-1 \leq d-1$ vectors, S is a proper subspace, so there is a $v \neq 0$ with $v \perp S$, and $A_W^{-\top} v \neq 0$ gives an $s \in \mathcal{P}$ with $v^\top A_W^{-1}(s - s_W) \neq 0$. Fix any $u \neq 0$ and put $A_\epsilon := A_W + \epsilon\, uv^\top$. For small $\epsilon > 0$ it inherits the invertibility of A_W, so $W_\epsilon := (A_\epsilon, s_W)$ is a context.

The window $h := h(W, s, a_{1:k-1})$ consists of the actions and of the observations $o_i = \iota(s_i - s_W)$ of the positions $s_i = s + A_W \sum_{j<i} a_j$. The partial sums of the actions lie in S, on which A_ϵ agrees with A_W, and W_ϵ carries the same target, so it produces the same positions and hence the same observations, that is, $h(W_\epsilon, s, a_{1:k-1}) = h$. The two contexts disagree on s_k, however: the Sherman–Morrison formula gives

$$g'_{W_\epsilon}(s_k) - g'_W(s_k) = -\frac{\epsilon\, v^\top A_W^{-1}(s_k - s_W)}{1 + \epsilon\, v^\top A_W^{-1} u}\, A_W^{-1} u,$$

which is non-zero whenever $v^\top A_W^{-1}(s_k - s_W) \neq 0$, because $A_W^{-1} u \neq 0$ and the denominator is non-zero for small ϵ. Now $s_k - s_W = (s - s_W) + A_W \sum_{i<k} a_i$ gives $A_W^{-1}(s_k - s_W) = A_W^{-1}(s - s_W) + \sum_{i<k} a_i$, whose second summand v annihilates, so $v^\top A_W^{-1}(s_k - s_W) = v^\top A_W^{-1}(s - s_W) \neq 0$ by the choice of s. One window therefore carries two coordinates, so e fails Definition 6 at horizon k on U. As e and U were arbitrary, $(g'_W)_{W\in\mathcal{W}}$ is not realizable at any $k \leq d$.

For the upper bound, let $U \subseteq \mathcal{A}^d$ be the set of $a_{1:d}$ with $\det(a_1, \ldots, a_d) \neq 0$. It is open as the complement of the zero set of a polynomial, that polynomial does not vanish identically because $a_i := \alpha e_i$ lies in $\mathcal{A}$ and has determinant α^d, and a zero set with empty interior leaves U dense. On U the window determines the distortion: its observations give the offsets $\iota^{-1}(o_i) = s_i - s_W$, whose differences are $A_W a_i = \iota^{-1}(o_{i+1}) - \iota^{-1}(o_i)$ for $1 \leq i \leq d$, where the unknown target cancels, so A_W is known on the basis $a_1, \ldots, a_d$ of $\mathbb{R}^d$. The map e that recovers A_W this way and returns $A_W^{-1}\iota^{-1}(o_{d+1})$ hence realizes $(g'_W)_{W\in\mathcal{W}}$ at horizon $d + 1$, and $k_{\text{pin}} = d + 1$ follows. The target is never recovered on the way. □

Dropping injectivity can raise both horizons above $d + 1$, as in Example 8, where $d = 1$ and the squared observation forces $k_{\text{fit}} = k_{\text{pin}} = 3$. Both bounds are attained (see Remark 5).

Remark 5 (Sharp bounds of Proposition 5). Proposition 6 attains the upper bound with $k_{\text{fit}} = 1$ and $k_{\text{pin}} = d+1$. Example 8 attains the lower bound with $k_{\text{fit}} = k_{\text{pin}} = 3$. To prove it, the system given there has $B_W = 1$, so $k_{\text{pin}} \leq 3$. On the other hand, it is easy to see that no $k < 3$ can realize any family due to symmetries of the observation function $O(s, W) = (s - w_2)^2$. Hence, $k_{\text{fit}} \geq 3$.

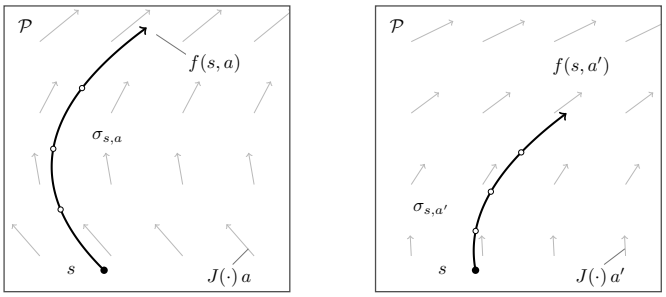


Figure 8: Flow-generated dynamics on a two-axis stage whose second axis rides on a pivoting arm, so that driving it also displaces the first axis by a gain that grows with how far the arm stands out. The vector field $J(\cdot)a$ on $\mathcal{P}$ is displayed in grey.

C FLOW-GENERATED DYNAMICS

In this section, we seek to understand more general conditions of dynamics where action-coordinates that disentangle the movement exist. A first class of functions f are those generated by flows. Again, we omit the context W in this section for simplicity, but the results hold for every $W \in \mathcal{W}$.

Definition 7 (Flow-generated dynamics)**.** For $s \in \mathcal{P}$ and $a \in \mathcal{A}$, consider the conditions

$$\dot{\sigma}(t) = J\left(\sigma(t)\right) \cdot a, \tag{13}$$

$$\sigma(0) = s, \tag{14}$$

$$\sigma(1) = f(s,a) \tag{15}$$

on a curve $\sigma\colon [0,1] \to \mathcal{P}$. The dynamics f is *flow-generated* if for all such s and a there is a $\sigma_{s,a}$ satisfying equation 13–equation 15.

In real-world systems, what an actuator receives is a rate, not a destination: a piezo stage driven at constant speed, or a screw advanced by a fixed number of motor increments (see Section D). Actuators whose response depends on where they currently stand, such as a lever whose gain grows with its extension, are covered as well.

First, we see that reversibility is sufficient for a dynamics to be flow-generated.

Proposition 7 (Reversibility and existence)**.** *Let $f\colon \mathcal{P} \times \mathcal{A} \to \mathcal{P}$ be smooth, fully actuated and reversible. Then f is flow-generated.*

Proof. Fix $s \in \mathcal{P}$ and $a \in \mathcal{A}$ and let $\sigma(t) := f(s, ta)$, which is defined on $[0,1]$ because $\mathcal{A}$ is convex and symmetric. For $t, t+h \in [0,1]$, reversibility of f (Definition 2) gives

$$\sigma(t+h) = f(s,(t+h)a) = f(f(s,ta),ha) = f(\sigma(t),ha),$$

so $\dot{\sigma}(t) = \partial_h f(\sigma(t),ha)|_{h=0} = J(\sigma(t))a$ and σ solves equation 13–equation 15, which makes f flow-generated. □

For flow-generated dynamics, a single condition on the actuation Jacobian decides whether *full* action-coordinates exist. Its proof can be made very short by borrowing machinery of differential geometry, but we give a more constructive and self-contained proof here.

Proposition 8 (Existence of full action-coordinates)**.** *Let f be flow-generated and fully actuated. Then f admits a full action-coordinate system if and only if*

$$DJ_i(s)\,J_j(s) = DJ_j(s)\,J_i(s) \qquad \text{for all } s \in \mathcal{P},\ i,j \in [d]. \tag{16}$$

In that case, for a given $s_0 \in \mathcal{P}$, every invertible $B \in \mathbb{R}^{d\times d}$ admits exactly one g with $Dg = BJ^{-1}$ and $g(s_0) = 0$, the pair (g,B) is a full action-coordinate system, and every full action-coordinate system is a translate of such a pair.

Proof. Full actuation gives $m = d$ and makes $J(s)$ invertible for every $s \in \mathcal{P}$, so the inverse J^{-1} is defined on all of $\mathcal{P}$ and, by Cramer's rule, smooth. Throughout this proof, ∂_k is the partial derivative in the k-th state coordinate, so that $(DJ_m)_{jk} = \partial_k J_{jm}$ for the Jacobian of the m-th column J_m of J. The strategy of the proof is to turn equation 16 into the integrability condition

$$\partial_k \big(J^{-1}\big)_{ij} = \partial_j \big(J^{-1}\big)_{ik} \qquad \text{for all } s \in \mathcal{P},\ i, j, k \in [d], \tag{17}$$

that is, every row of J^{-1} has a symmetric Jacobian, and then builds the chart with the Poincaré lemma. To see the equivalence, collect the difference of the two sides of equation 17 in the matrices $T_i(s) \in \mathbb{R}^{d\times d}$ with entries $(T_i)_{jk} := \partial_k \big(J^{-1}\big)_{ij} - \partial_j \big(J^{-1}\big)_{ik}$ and fix $i, m, n \in [d]$. The (i, m)-th entry of $J^{-1}J = I$ reads $\sum_j (J^{-1})_{ij} J_{jm} = \delta_{im}$ and is constant in s, so differentiating it along the n-th column of J, that is applying $\sum_k J_{kn}\partial_k$, gives

$$\sum_{j,k} J_{jm} J_{kn}\, \partial_k \big(J^{-1}\big)_{ij} = -\sum_j \big(J^{-1}\big)_{ij} \sum_k \big(\partial_k J_{jm}\big) J_{kn} = -\big(J^{-1}\, DJ_m\, J_n\big)_i.$$

Exchanging m and n and renaming the summation indices j and k turns the left hand side into $\sum_{j,k} J_{jm} J_{kn}\, \partial_j (J^{-1})_{ik}$, so subtracting the two identities gives

$$\big(J^\top T_i\, J\big)_{mn} = -\Big(J^{-1}\big(DJ_m\, J_n - DJ_n\, J_m\big)\Big)_i. \tag{18}$$

As both, J and J^{-1} are invertible, the left hand side of equation 18 vanishes for all i, m, n if and only if all T_i vanish, which is equation 17 and the right hand side vanishes for all i, m, n if and only if $DJ_m J_n = DJ_n J_m$ holds for all m, n, which is equation 16. Read as vector fields, the columns of J have the Lie bracket $[J_m, J_n] = DJ_n\, J_m - DJ_m\, J_n$, so equation 16 states that they commute and equation 17 that the rows of J^{-1} are closed 1-forms.

To prove the condition of the theorem, let now (g, B) be a full action-coordinate system. By Proposition 4, g is smooth with $Dg = BJ^{-1}$, that is $\partial_k g_i = \sum_p B_{ip}(J^{-1})_{pk}$. Its second derivatives commute, hence

$$\sum_p B_{ip}\,(T_p)_{kj} = \partial_j \partial_k g_i - \partial_k \partial_j g_i = 0$$

for all i, j, k, and B is invertible, so all T_p vanish, which is equation 17, and equation 16 follows.

Conversely, assume equation 16 and fix an invertible $B \in \mathbb{R}^{d\times d}$. As B is constant, $\partial_k (BJ^{-1})_{ij} = \sum_p B_{ip}\, \partial_k (J^{-1})_{pj}$, so the rows of BJ^{-1} inherit equation 17 from those of J^{-1}: each of them is a smooth map $\mathcal{P} \to \mathbb{R}^d$ with a symmetric Jacobian on the open and simply connected $\mathcal{P}$ and hence, by the Poincaré lemma, the gradient of a function $g_i \colon \mathcal{P} \to \mathbb{R}$, unique up to an additive constant because $\mathcal{P}$ is connected. The normalization $g(s_0) = 0$ fixes these d constants, so there is exactly one $g = (g_1, \ldots, g_d)$ with $Dg = BJ^{-1}$ and $g(s_0) = 0$. It satisfies equation 2: for $s \in \mathcal{P}$ and $a \in \mathcal{A}$, the chain rule and equation 13 give

$$\frac{d}{dt} g\big(\sigma_{s,a}(t)\big) = Dg\big(\sigma_{s,a}(t)\big) J\big(\sigma_{s,a}(t)\big) a = BJ^{-1}\big(\sigma_{s,a}(t)\big) J\big(\sigma_{s,a}(t)\big) a = Ba,$$

which does not depend on t, so integrating over $[0, 1]$ and inserting equation 14 and equation 15 gives $g(f(s, a)) = g(s) + Ba$. As B is invertible, (g, B) has rank d and hence is full.

Finally, let (g', B') be any full action-coordinate system. Then $B' \in \mathbb{R}^{d\times d}$ is invertible and $Dg' = B'J^{-1}$ by Proposition 4, so g' and the chart g constructed above for the gain B' have the same derivative on $\mathcal{P}$. Hence $g' = g + g'(s_0)$, which makes (g', B') the translate of (g, B') by $g'(s_0)$. □

Example 9 (Movement without full action-coordinates)**.** Let now $m = d = 2$ with $J(s) = \operatorname{diag}(1, e^{s_1})$, a system whose second actuator acts with a gain that depends on the state of the first. Integrating equation 13 gives

$$f(s, a) = \left(s_1 + a_1,\ s_2 + a_2 e^{s_1} \frac{e^{a_1} - 1}{a_1}\right),$$

read as $s_2 + a_2 e^{s_1}$ at $a_1 = 0$, which is not affine in a. The gain never vanishes, so f is fully actuated on $\mathcal{P} = \mathbb{R}^2$ and $f(s, a) = s$ holds only for $a = 0$. Here $DJ_2(s)J_1(s) = e^{s_1} e_2$ while $DJ_1(s)J_2(s) = 0$, so equation 16 fails and no *full* action-coordinate system exists by Proposition 8. Started at the origin, one unit on the first actuator followed by one unit on the second reaches $(1, e)$, the reverse order reaches $(1, 1)$, and holding both at once reaches $(1, e - 1)$. No encoder can make that difference disappear. The first coordinate survives it, however: the pair $g(s) := s_1$ and $B := e_1^\top$ satisfies equation 2 and is an action-coordinate system of rank 1, blind to s_2 by construction.

D ACTION-COORDINATES AND REAL-WORLD SYSTEMS

We emphasize that action-coordinates are a mathematical model of the dynamics. On real-world systems, saturation effects of actions and states can break the mathematical assumptions we made here. For instance, we assume that if $a \neq 0$, then $f(s, a) \neq s$. This is violated if an actuator is held at a travel limit. It is also why $\mathcal{P}$ is taken open: Lemma 3 has the chart cover all of $\mathbb{R}^\ell$, which a state space clipped to a closed box cannot support, since a state resting on one of its faces stops moving. Another example is the reversibility assumption of f, which is violated if the system has backlash, dead zones, or other hysteresis effects. Such effects are present in the environments of Section 1.2, specifically in the physical environments LIGHTTUNNEL and SPHEROENV, and although our theory holds only away from the travel limits of actions and positions, the experiments of Section 4 show that the learned coordinates still steer these systems to their targets.

Whether a real actuator admits action-coordinates is decided by its kinematics and by what a command does to it. A mechanism whose position is a function of the actuator coordinate alone has a forward kinematics $\kappa\colon \mathbb{R}^d \to \mathcal{P}$, a diffeomorphism if the mechanism is fully actuated, so that $s = \kappa(q)$ for the actuator coordinate $q \in \mathbb{R}^d$. If a command a advances that coordinate by Γa for a fixed invertible gain $\Gamma \in \mathbb{R}^{d\times d}$, the position moves to

$$f(s, a) = \kappa\left(\kappa^{-1}(s) + \Gamma a\right).$$

Particularly, the actuator coordinates are full action-coordinates (κ^{-1}, Γ) however non-linear the forward kinematics κ is. Over a fraction t of the period the coordinate advances by $t\Gamma a$, so the curve $t \mapsto f(s, ta)$ has velocity $\dot{s} = J(s) \cdot a$ with $J(s) := D\kappa\left(\kappa^{-1}(s)\right)\Gamma$, which makes the dynamics flow-generated (Definition 7).

The gain is what the drive delivers, and both modes in which an industrial drive is commanded advance the actuator coordinate that way. In *position mode* the drive regulates q against the reference $q + a$ of a relative command, which is the case $\Gamma = I$. In *velocity mode* it holds one rate over the control period, and that rate is a linear function $\dot{q} = \Gamma a$ of the action, so the coordinate advances by Γa over the period. What produces that rate is a state-feedback controller: writing ξ for the internal state of the drive, which holds the motor current, the motor rate and the integrator states of its loops, and u for the motor input, the linearized drive and its control law are

$$\dot{\xi} = \mathbf{A}\xi + \mathbf{B}u, \qquad \dot{q} = \mathbf{C}\xi, \qquad u = -\mathbf{K}\xi + \mathbf{V}a,$$

whose closed loop $\dot{\xi} = (\mathbf{A} - \mathbf{BK})\xi + \mathbf{BV}a$ has a Hurwitz matrix by design, so ξ settles at $(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{BV}a$ and $\Gamma = \mathbf{C}(\mathbf{BK} - \mathbf{A})^{-1}\mathbf{BV}$ follows, see Åström & Murray (2021, Chapter 7). The rate is Γa only once that loop has settled, which is why the control period is chosen long against the settling time of $\mathbf{A} - \mathbf{BK}$.

The chart, on the other hand, is determined by the mechanism. The forward kinematics κ composes the travel of each actuator with the geometry that turns it into a position: for the steering mirrors of INTERFEROBOT it carries their screw turns through the mirror tilts into the position of the beam on the camera, and for a linear stage it is the pitch of its screw. Where those levers do not change over the travel that is used, κ is affine, $D\kappa \equiv W$ is constant and f becomes $f(s, a) = s + Wa$.

E DETAILS OF EXPERIMENTS IN ENVIRONMENTS

This section details the experiments of Section 4 for each environment of Section 1.2. Unless stated otherwise, all environments share the same training and evaluation template. Table 1 lists the default dataset parameters of the offline comparisons, and Table 2 the best configuration per environment, observation, and world model scheme, which is the one drawn in Figure 5. Every dataset is collected with at least three seeds, which also seed training and evaluation, except on real hardware or where rendering is too slow.

World-model templates. Vector and image observations use different encoder architectures. On the low-dimensional vector observations, AC-WM encodes with a multilayer perceptron, as the embedding dimension ℓ is always smaller than the observation dimension, while the planner-based schemes take the vector observation itself as their latent. On images, the encoder is one of four architectures and Table 2 lists the best one per environment. The `ae` architecture is a convolutional

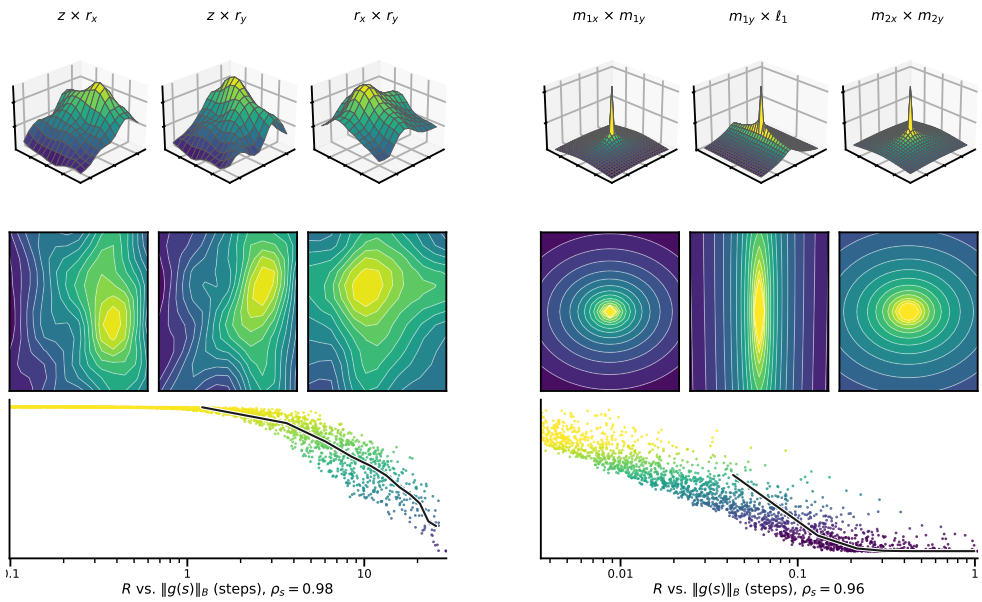
environment	observation	n	T	step size α	drift	eval. episodes
BEAMALIGNMENT	beam vector	1500	10	0.3	–	100
BEAMALIGNMENT	screen image	1500	10	0.3	–	100
SPHEROENV sim-abs	rendered image	500	10	0.3	–	100
SPHEROENV sim-rot	rendered image	500	10	0.3	–	100
SPHEROENV real	camera image	150	10	1.0	–	20
INTERFEROBOT	handle state	1500	20	0.05	–	100
INTERFEROBOT	fringe image	1500	20	0.05	–	100
LENSALIGNMENT	MTF vector	1000	60	1.0	0.7	20
LENSALIGNMENT	rendered image	400	40	1.0	0.7	12
LIGHTTUNNEL	camera image	150	20	0.5	0.5	12

Table 1: Default dataset configurations collected for offline experiments.

panel	method	window k	gain B	encoder	training
BEAMALIGNMENT vector	AC-WM	1	B fit ($\ell = 4$)	MLP	240 epochs
BEAMALIGNMENT vector	RG-WM	4	–	–	200 epochs
BEAMALIGNMENT vector	S-WM	4	–	–	200 epochs
BEAMALIGNMENT vector	MPC-WM	4	–	–	200 epochs
BEAMALIGNMENT image	AC-WM	3	B fit ($\ell = 4$)	keypoint	patience 20
BEAMALIGNMENT image	RG-WM	4	–	ae	400 epochs
BEAMALIGNMENT image	S-WM	4	–	ae	400 epochs
BEAMALIGNMENT image	MPC-WM	4	–	ae	400 epochs
SPHEROENV absolute	AC-WM	1	$B = I$	keypoint	patience 20
SPHEROENV absolute	RG-WM	1	–	keypoint	200 epochs
SPHEROENV absolute	S-WM	1	–	keypoint	200 epochs
SPHEROENV absolute	MPC-WM	1	–	keypoint	200 epochs
SPHEROENV rotated	AC-WM	1	B fit	keypoint	patience 20
SPHEROENV rotated	RG-WM	4	–	keypoint	200 epochs
SPHEROENV rotated	S-WM	4	–	keypoint	200 epochs
SPHEROENV rotated	MPC-WM	4	–	keypoint	200 epochs
SPHEROENV real	AC-WM	1	B fit	keypoint	patience 20
SPHEROENV real	RG-WM	3	–	keypoint	600 epochs
INTERFEROBOT state	AC-WM	1	$B = I$	MLP	800 epochs
INTERFEROBOT state	RG-WM	1	–	–	200 epochs
INTERFEROBOT state	S-WM	4	–	–	200 epochs
INTERFEROBOT state	MPC-WM	1	–	–	200 epochs
INTERFEROBOT image	AC-WM	1	$B = I$	sharpness	800 epochs
INTERFEROBOT image	RG-WM	1	–	ae	200 epochs
INTERFEROBOT image	S-WM	1	–	ae	200 epochs
INTERFEROBOT image	MPC-WM	1	–	ae	200 epochs
LENSALIGNMENT vector	AC-WM	1	$B = I$	MLP	400 epochs
LENSALIGNMENT vector	RG-WM	1	–	–	400 epochs
LENSALIGNMENT vector	S-WM	1	–	–	400 epochs
LENSALIGNMENT vector	MPC-WM	1	–	–	400 epochs
LENSALIGNMENT image	AC-WM	5	$B = I$	spectral	400 epochs
LENSALIGNMENT image	RG-WM	4	–	spectral	50 epochs
LENSALIGNMENT image	S-WM	4	–	spectral	50 epochs
LENSALIGNMENT image	MPC-WM	4	–	ae	50 epochs
LIGHTTUNNEL	AC-WM	1	$B = I$	spectral	400 epochs
LIGHTTUNNEL	RG-WM	1	–	spectral	400 epochs
LIGHTTUNNEL	S-WM	1	–	spectral	400 epochs
LIGHTTUNNEL	MPC-WM	1	–	spectral	400 epochs

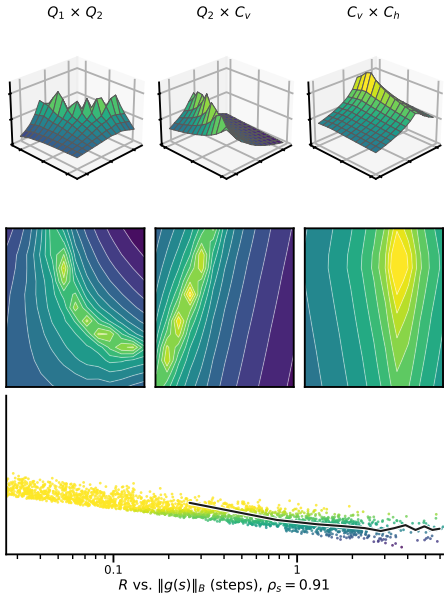
Table 2: The best configurations for each world model template per environment used in Figure 5.

autoencoder. The `keypoint` architecture is a spatial soft-argmax: convolutions produce 32 heat maps and the latent is a linear image of their spatial expectations. The `sharpness` trunk consists of strided convolutions, pooled to a 5×5 grid and flattened instead of averaged, so that it retains where in the field of view the image structure changes. This is what the experiments on INTERFEROBOT require, where the pattern stays in place and only its local appearance carries information. The

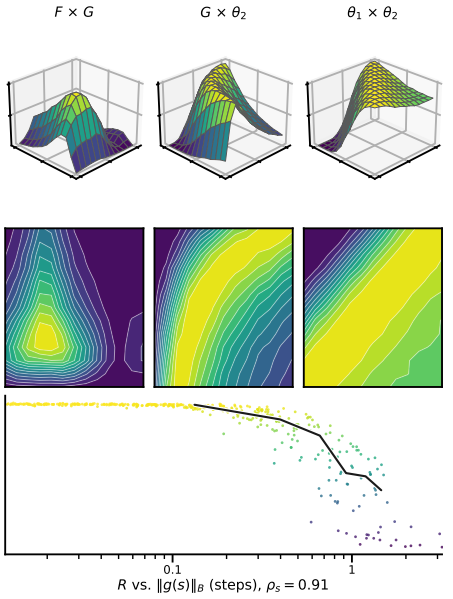


(a) LENSALIGNMENT

(b) INTERFEROBOT



(c) BEAMALIGNMENT



(d) LIGHTTUNNEL

Figure 9: Radiality of the reward of four of the environments at one fixed context W. *Top and middle rows:* $R(\cdot, W)$ over the position box as a surface and as its contour, one column per pair of position coordinates with the remaining ones held at a point from $\mathcal{M}_W$. *Bottom row:* the test of Definition 4. Each point is a position with its radius $\|g(s)\|_B$ drawn log-uniformly and its direction uniformly with ρ_s the rank correlation.

`spectral` trunk replaces those convolutions by a parameter-free radial power spectrum per block, a spatially resolved estimate of the modulation transfer function, followed by a small multilayer perceptron. With `ae` on RG-WM, S-WM, or MPC-WM, training adds a reconstruction loss. RG-WM, S-WM and MPC-WM are trained on the same data as AC-WM with a reconstruction loss on the encoder, and differ only in how they maximize $J(a) := r_\phi(g_\phi(e_\phi(h_t), a)) - \frac{0.01}{d}\|a\|_1$ at test-time: RG-WM by 40 projected Adam steps at learning rate 0.02, S-WM by the cross-entropy method with 512 samples, 64 elites and 5 iterations, and MPC-WM by that sampler over a planning

horizon of 5 actions, of which the first is then applied. Their window is padded with the first frame at the episode start, so they plan from the first step. In some figures below, we also show their unpadded behavior, where the first $k-1$ steps are random probes that fill the window with real frames, as AC-WM's history encoder needs.

Model-free baselines. IQL and TD3+BC run in `d3rlpy` (Seno & Imai, 2022) with its default hyperparameters, except $\gamma = 0.5$, chosen on the BEAMALIGNMENT vector from $\{0.99, 0.8, 0.5\}$ since an episode has $T = 10$ steps, and z-scored rewards for IQL. On images they read the frame through the keypoint encoder of AC-WM, as the convolutional trunk of `d3rlpy` does not learn SPHEROENV. The window is $k = 1$, on LENSALIGNMENT $k = 10$. Training runs for $2 \cdot 10^5$ gradient steps in Figure 7a and until an in-environment evaluation of 20 episodes has plateaued in Figure 6; the reported model is the best evaluated checkpoint. On random data, where no episode starts at the target, $\mathcal{L}_{\text{tgt}}$ of AC-WM contracts the 50 frames closest to it.

method	setting
IQL	$\gamma = 0.5$, reward z-scored
TD3+BC	$\gamma = 0.5$
MOPO	rollout horizon 5, penalty coefficient 0.5, $\gamma = 0.5$
COMBO	rollout horizon 5, conservative weight 5, $\gamma = 0.5$

Table 3: Settings of the offline baselines of Figure 7a that deviate from the defaults of their implementations.

Model-based baselines. MOPO and COMBO run in OfflineRL-Kit[1] with $\gamma = 0.5$ for $3 \cdot 10^5$ gradient steps; Table 3 lists the settings that deviate from the defaults.

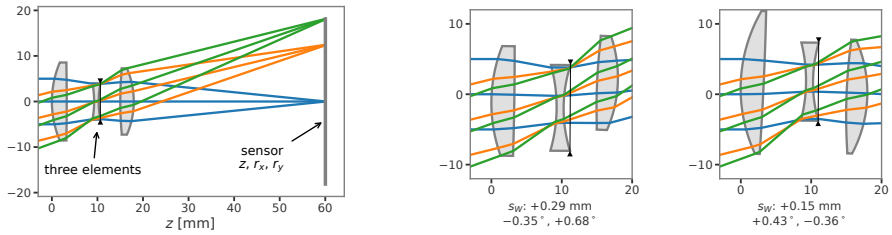
TD-MPC2 runs with the official code, its 5M-parameter preset and planning horizon 3 for $3 \cdot 10^5$ transitions on the k most recent frames; its curve is its own evaluation during training, 10 episodes every 10^4 transitions. The online AC-WM starts from 500 random-walk episodes and collects in rounds of growing size with the current model as controller and a random action with probability 0.1; after every round a fresh model is trained on the whole buffer with the offline recipe, anchored on the frames within the goal tolerance and at least the 50 closest, and evaluated on 100 episodes.

E.1 ACTIVE ALIGNMENTS OF LENS SYSTEMS (LENSALIGNMENT)

This environment is an alignment task of a Cooke triplet, built with the `optiland` library Harrison (2026). Every episode decenters and tilts the three lens elements by a small random amount, and the latent context W encodes that defect (Figure 10). The task is to reposition the sensor in the three degrees of freedom that affect the image, defocus and two tilts, towards a context-dependent optimal pose s_W that the defect displaces. The reward $R(s, W)$ is the *modulation transfer function* (MTF), summed over 18 field points. Two observation maps share this reward: those MTF values and a rendered image of a resolution target. Rendering runs on GPU, but one transition still costs far more time than in the other simulated environments, so the image observation uses fewer evaluation episodes and fewer datasets. Learning-based alignment of optics and accelerators is almost entirely model-free and trained in simulation with a randomized context (Kain et al., 2020; Burkhardt et al., 2025) or directly on the experiment (Richtmann et al., 2025); the exceptions that learn a dynamics model and plan with it are an ensemble model for beam steering (Hirlaender & Bruchon, 2020) and a model of the adaptive-optics loop that absorbs a mis-registration between mirror and sensor (Nousiainen et al., 2021). Classical active alignment inverts a *sensitivity table*, a linear map from misalignment to aberration derived from the nominal design (Kim et al., 2007), and recent work infers the hidden lens tolerances first and then plans with a known optical model (Hu et al., 2025).

A disadvantage of datasets anchored at s_W is that they cover little of the position space, even for long trajectories with $T = 60$. We therefore add a drift to the anchored walk, whose direction is resampled every five steps. The target set $\mathcal{M}_W$ is a single point. Figure 9a shows the reward landscape, Figure 12 the convergence for the image observation and Figure 11 for the vector case.

[1] `https://github.com/yihaosun1124/OfflineRL-Kit`



(a) The undisturbed system.

(b) Two contexts with random defects.

Figure 10: The Cooke triplet of LENSALIGNMENT, ray-traced at three field angles: (a) the undisturbed system and (b) two random defects.

On the MTF vector, AC-WM is best with $k = 1$ and B pinned to I, and RG-WM and S-WM also beat the context-blind policy. On the image, no world model scheme other than AC-WM works at all, and AC-WM itself only barely matches the context-blind policy on average. The burden this environment puts on the encoder is visible in k: the MTF vector needs only $k = 1$ for a policy above context-blind, the image needs $k = 5$ and the `spectral` encoder with FFT layers.

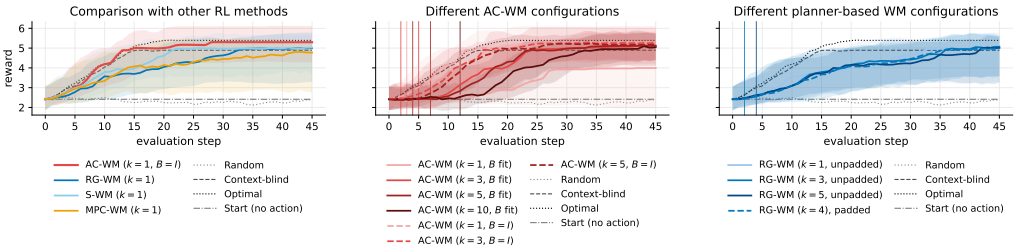


Figure 11: Reward against the evaluation step for LENSALIGNMENT with vector observation.

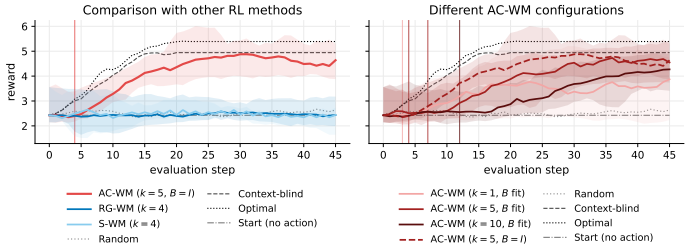


Figure 12: Reward against the evaluation step for LENSALIGNMENT with image observations.

E.2 PARTICLE ACCELERATOR BEAM ALIGNMENT (BEAMALIGNMENT)

This environment is the ARES Experimental-Area transverse beam-tuning task, a simplified port of Kaiser et al. (2022) simulated with the differentiable beam-dynamics library `cheetah` Kaiser et al. (2024a). The agent steers quadrupole magnets and steering correctors along a short beamline to focus and center an incoming charged-particle beam on a diagnostic screen. The context W encodes that incoming beam's centroid and width, drawn fresh every episode and observed only through their effect on the measured beam. Kaiser et al. (2024b) compare reinforcement learning against Bayesian optimization on this task. The reward is built from the beam vector, the measured beam's centroid and width in the two transverse axes. It is a weighted sum of the logarithm of the beam's width and the logarithm of the distance between its centroid and the desired beam. That distance is the *beam error* and is measured in millimeters. In this environment, episodes do not terminate on success and thus we measure the capability of hold on target, denoted *final beam error*, additionally.

As the positions of 5 magnets are evaluated on 4 beam parameters, the target set $\mathcal{M}_W$ is one-dimensional and an R-anchored action-coordinate system must have rank $\ell = d - 1 = 4$. We nevertheless test several settings: $\ell = 5$ with fitted and with pinned B, and $\ell = 4$ with a fitted $\hat{B} \in \mathbb{R}^{4\times5}$. The observation is either the rendered screen image or the vector of magnet settings and measured beam parameters.

Figure 13 and Figure 14 show the results for vector and image observations, respectively. In both plots, the theoretical optimal reference lies above the plotted range due to the logarithmic scale.

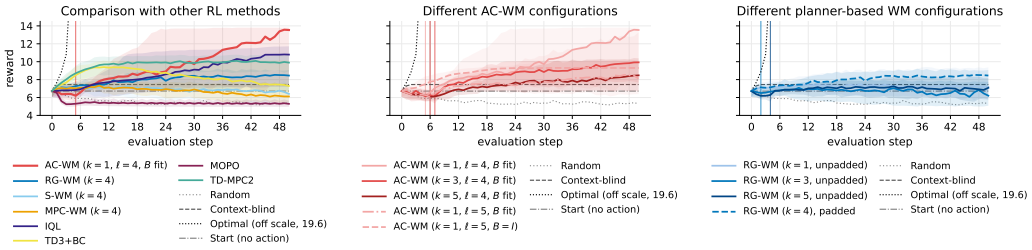


Figure 13: Reward against the evaluation step for BEAMALIGNMENT with vector observation.

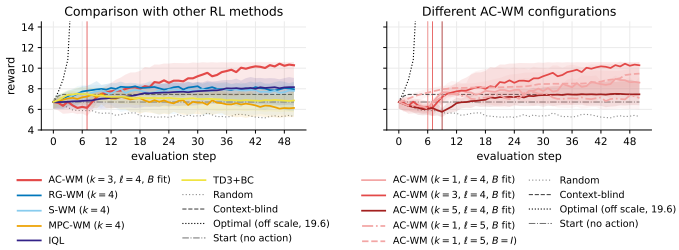


Figure 14: Reward against the evaluation step for BEAMALIGNMENT with image observation.

We further ablate the step size α of the anchored data collection by training the best AC-WM configuration on datasets collected at varying α (Table 4), and we vary the dimensionality of the target set $\mathcal{M}_W$ (Table 5) in a controlled way by freezing one actuator at an optimal value, which turns $\mathcal{M}_W$ from a one-dimensional curve into a single point.

Finally, Figure 15 shows, for AC-WM and RG-WM, the learned embedding of a two-dimensional slice of magnet settings through the optimum, for three contexts. The AC-WM panels are consistent with a 4-dimensional lattice centered at the origin, which the PCA embedding projects onto a plane.

step α	seeds	final [mm]	best [mm]
0.1	3	0.0639	0.0299
0.3	3	0.0024	0.0023
0.6	3	0.0041	0.0026
1.0	3	0.0330	0.0210

Table 4: Ablation for step size α at $n = 1500$ episodes.(30 steps evaluation)

E.3 INTERFEROMETER ALIGNMENTS (INTERFEROBOT)

This environment is `gym_interf` Sorokin et al. (2020), which implements the interferometer-alignment task of Makarenko et al. (2022). The task is to overlap a reference beam with a second beam on a camera by controlling two mirrors, with one x and one y screw each, and one telescope lens mount m. The image observation comes from a piezo scan over the relative phase in $[0, 2\pi]$, yielding a stack of 16 interference images at different phases. The scan starts at a random phase, so the fringes shift from scan to scan, while the local fringe contrast, spacing and direction are invariant

observation	actuators	latent ℓ	$\hat{B}$	final [mm]	best [mm]	$\sigma_{\min}(\hat{B})$
beam	5	5	5×5	0.110	0.038	0.10
beam	4 (one frozen)	4	4×4	0.015	0.013	0.23
beam	5	4	4×5	0.011	0.007	0.26
image	5	5	5×5	0.089	0.035	0.10
image	4 (one frozen)	4	4×4	0.023	0.013	0.17
image	5	4	4×5	0.040	0.021	0.24

Table 5: Ablation on target set dimensionality. No-action reference 0.185 mm.

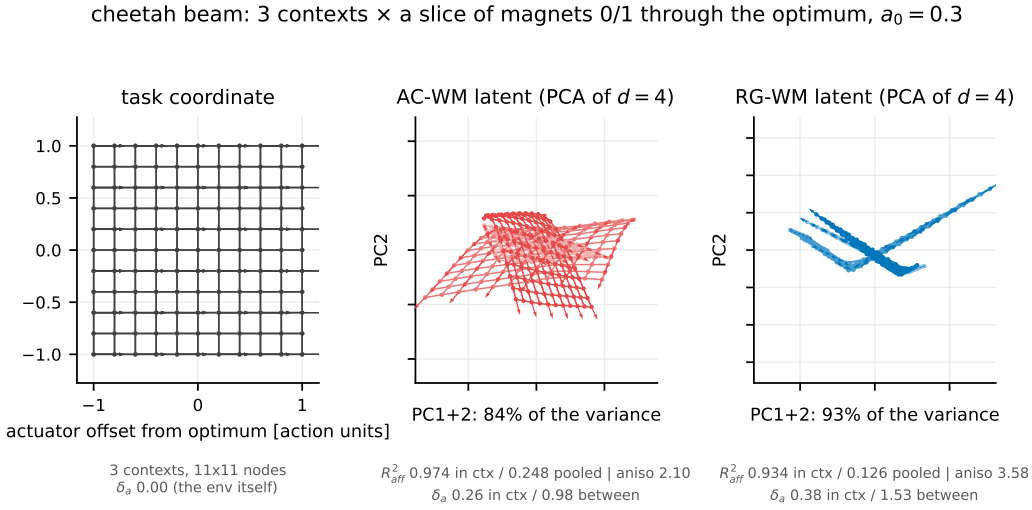


Figure 15: Visualizations of learned embeddings of AC-WM and RG-WM for different contexts in BEAMALIGNMENT

to that shift. The vector observation holds s together with hand-engineered geometric properties of the fringe pattern, among them beam radius and curvature.

For a position $s = (x_1, x_2, y_1, y_2, m)$, the fringe *visibility* $V(s) \in [0, 1]$ of that scan states how completely the two beams interfere, with $V(s) = 1$ for perfect alignment. The reward is $R(s, W) := V(s) - \log(1 - V(s))$ as in Sorokin et al. (2020, Equation 4). The target set $\mathcal{M}_W$ is a single point s_W, the position where the two beams overlap perfectly. The logarithmic term rewards only a high-quality final alignment and concentrates the reward mass in a small region around the optimum (Figure 9b), which makes this task a particularly hard one. Data collection therefore uses actions from $\alpha \cdot \mathcal{A}$ with $\alpha = 0.05$. The per-episode context W is a calibration of the actuators, which rescales how far a given action moves the optics and shifts s_W.

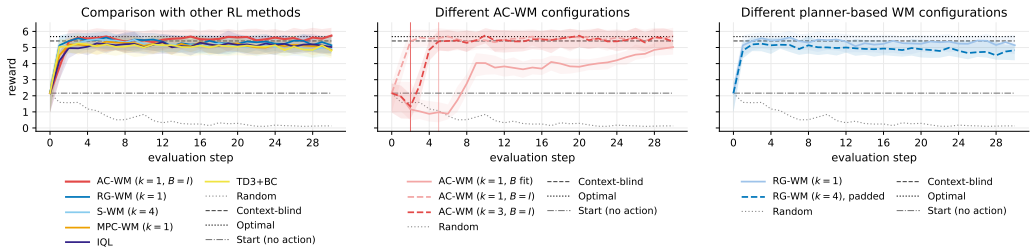


Figure 16: Reward against the evaluation step for INTERFEROBOT with vector observations.

Figure 16 and Figure 17 show the results for vector and image observations, respectively. In both figures, context-blind and optimal policies are represented by their median final performance over

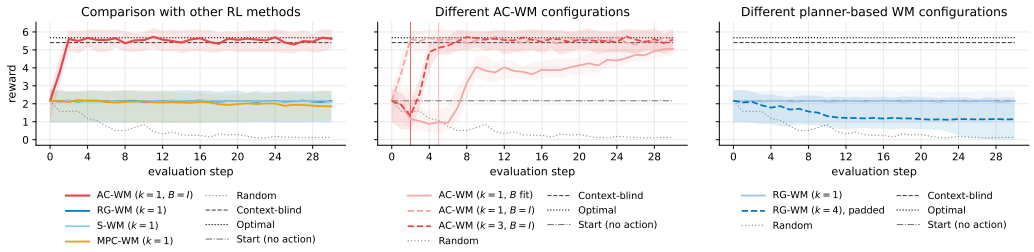


Figure 17: Reward against the evaluation step for INTERFEROBOT with image observations.

100 episodes, due to high fluctuations in the reward from step to step. AC-WM uses $\ell = 5$ and is best with $k = 1$ and B pinned to I, where it reaches the optimum on the vector observation and on the image alike.

The dimension and the shape of the observation invite encoders beyond those of Table 2, and we also test one built on the parameter-free demodulation standard in phase-shifting interferometry. It improves individual configurations, in particular AC-WM with fitted B, but never reaches the best results. On the image observation, the `sharpness` encoder is best for AC-WM and the convolutional autoencoder for the other planners. The best planner-based world models come close to the optimum on the vector observation, though they stay below context-blind, and fail completely on the image. They improve when the optimization over the prediction horizon is restricted to $\alpha \cdot \mathcal{A}$ with $\alpha = 0.05$, the action set of the data collection.

E.4 CAMERA TUNING IN A LIGHT TUNNEL (LIGHTTUNNEL)

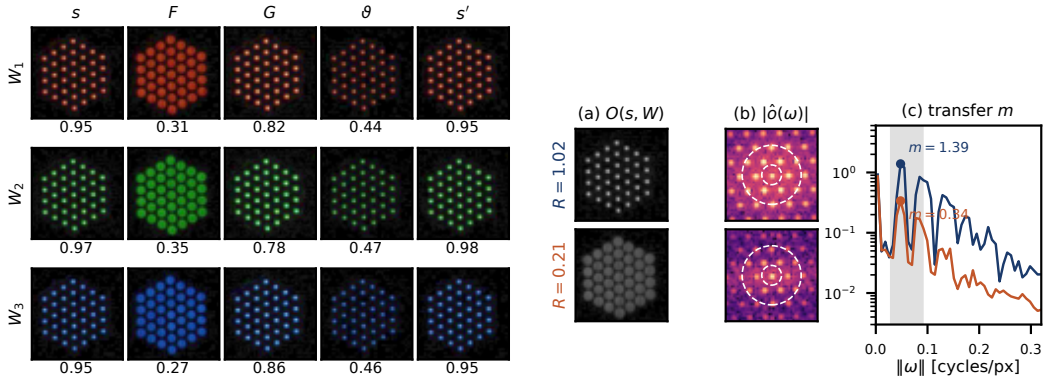
This environment is implemented within the light tunnel of the Causal Chambers Gamella et al. (2025), a physical device in which a camera looks down a tube at a hexagonal array of light-emitting diodes, through two linear polarizers. All experiments run on that device. The task is to image the array sharply at a prescribed brightness. The state is the camera focus F, the sensor gain G, both discretized into 64 steps, and the angles θ_1 and θ_2 of the two polarizers. Actions apply increments to each of these four axes, so the dynamics is fully actuated and reversible with $d = 4$. The context W is the color of the LED array, drawn every episode and set at the diodes. The observation is a single camera frame $O(s, W) \in \{0, \dots, 255\}^{N \times N \times 3}$. Figure 18a shows the effect of each action for three contexts.

Every episode draws a random color and sets the actuators to a uniformly random position $s = (F, G, \theta_1, \theta_2)$. The reward $R(s, W)$ is computed from $O(s, W)$ alone. Let $o := O(s, W)$, let $\bar{o} \in \mathbb{R}^{N \times N}$ be its average over the three color channels and let $\hat{o}$ be the discrete Fourier transform of $\bar{o}$. Then $L(s, W) := \frac{1}{N^2} \sum_x \bar{o}(x)$ is the mean intensity and $m(s, W) := \frac{2}{\hat{o}(0)} \max_{\|\omega\| \in \mathcal{B}} \left|\hat{o}(\omega)\right|$ is the modulation transfer at the lattice frequency, where $\mathcal{B} := [0.03, 0.09]$ in cycles per pixel is the radial band that brackets it. The level L is the exposure, the light the frame received: the color sets how much the diodes emit, and the gain and the polarizer pair scale it on the way to the sensor (see Figure 18b). The transfer m measures sharpness as the fraction of the array's own contrast that survives the imaging. Defocus smears the diodes into the gaps between them and collapses m, while the ratio to the mean level makes m indifferent to how much light the frame received. With these, the reward is

$$R(s, W) := \frac{m(s, W)}{m^*} \exp\Big(-\beta \log^2 \frac{L(s, W)}{L^*}\Big), \tag{19}$$

where m^*, L^* and β are global calibration constants chosen so that R is bounded by 1 and spreads over the position space.

The target set $\mathcal{M}_W$ is a two-dimensional surface, so an R-anchored action-coordinate system of LIGHTTUNNEL has rank $\ell = 2$. The first direction of $\mathcal{M}_W$ turns the two polarizers together: by Malus' law, only the angle difference $\vartheta = \theta_1 - \theta_2$ affects the camera image. The second trades ϑ against G, as a change of ϑ can be compensated by an adjustment of the gain. Figure 9d shows their non-linear interplay in the reward landscape.



(a) Effect of actions and context.

(b) How R is read off a single frame.

Figure 18: Mechanics within LIGHTTUNNEL.

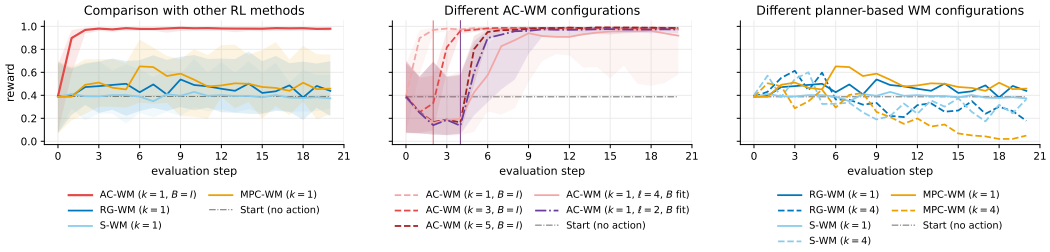


Figure 19: Reward against the evaluation step for LIGHTTUNNEL.

The experiments run on the real device, so we collect a single anchored image dataset of 150 episodes with $T = 20$ steps each. Figure 19 shows the results. For AC-WM we test $\ell = 4$ with $B = I$ under varying k, and $\ell \in \{2, 4\}$ with a fitted B. All these configurations reach the optimum and differ only in the number of steps they need: once the probing phase is over, all of them converge at nearly the same speed. The probing phase costs $k - 1 + d$ steps if B is fitted and $k - 1$ steps if $B = I$, so $k = 1$ with $B = I$ converges fastest. That configuration learns an action-coordinate system of rank $\ell = 4$, although an R-anchored one must have rank $d - \dim(\mathcal{M}_W) = 2$. We have no explanation for this, but assume that the learned full-rank system is anchored at a single canonical point in the interior of $\mathcal{M}_W$.

E.5 BOLT POSITIONING (SPHEROENV)

This environment follows the setup of Siffl (2026). The task is to navigate a rolling-ball robot, a Sphero BOLT+, to a target s_W in a planar area $\mathcal{P} = [0, 1]^2$ using only RGB camera observations, into which both the ball and the target are rendered (Figure 22a). For the ground-truth position $s \in \mathcal{P}$ of the ball and a target s_W drawn uniformly at random in each episode, the reward is $R(s, W) = -\|s - s_W\|$, with a bonus for landing within a tolerance C of the target. We refer to Siffl (2026) for further details.

On real hardware, the robot moves relative to its own coordinate system, which is fixed within an episode and need not agree with the arena's. The distortion between the two systems and the target s_W form the per-episode context W, which a world model must identify from the probing part of a trajectory instead of reading it off a frame. The SAPIEN simulation also offers an easier variant in which the robot's coordinate system is aligned with the arena's.

Figure 20, Figure 21 and Figure 22b show the results for the two simulated variants and for the real hardware. Despite careful tuning of the hyperparameters, and even on the easier variant without distortion, no world model other than AC-WM works at $n = 500$ episodes in simulation. We therefore vary the number of episodes n and the fraction of anchored ones (Figure 23a): at $n =$

3,000, RG-WM solves the task as well as AC-WM. AC-WM is best on anchored data and worst on purely random data, and at $n = 3{,}000$ the planner overtakes it there. This suggests that AC-WM learns a globally valid representation from a small number of episodes already. Figure 23b supports this: distances in the latent space of AC-WM are globally consistent, while the latent space of RG-WM is distorted and shows no repeating structure.

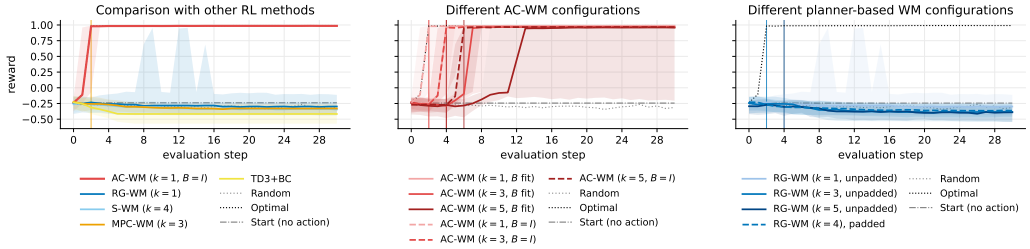


Figure 20: Reward against the evaluation step for SPHEROENV (simulation) in absolute coordinates.

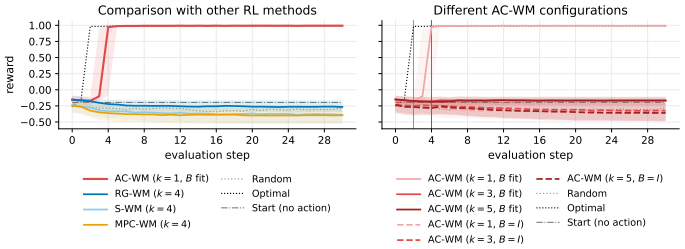
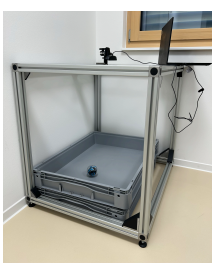


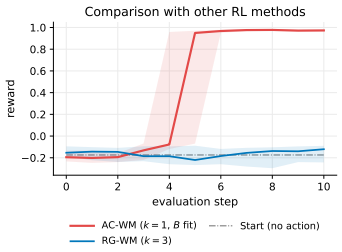
Figure 21: Reward against the evaluation step for SPHEROENV (simulation) with context-dependent rotated coordinates.

The main challenge of this task is that the distortion between the two coordinate systems becomes visible only through interaction. In the easier variant (Figure 20), AC-WM solves the task for every k and whether the gain is fitted or pinned, with small differences in the number of steps to the optimum that come from the probing phase. In the harder variant (Figure 21), AC-WM works only if B is fitted in every episode, although a realization with $B = I$ and $k = 3$ exists.

On real hardware we therefore train the best AC-WM configuration of the harder simulated variant, and RG-WM with $k = 3$ and a longer schedule (Table 2). Data collection on hardware is slow, so the dataset has only $n = 150$ episodes of $T = 10$ steps (Table 1). Trained on it, AC-WM solves the task consistently (Figure 22b).

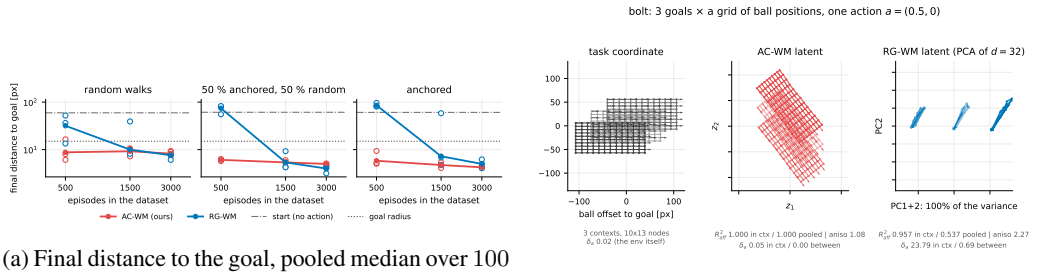


(a) Hardware setup.



(b) Reward against the evaluation step for the physical SPHEROENV.

Figure 22: The physical SPHEROENV bench.



(a) Final distance to the goal, pooled median over 100 episodes for 3 seeds, against the training-set size and the fraction of anchored episodes.

(b) Learned embeddings of AC-WM and RG-WM for different contexts.

Figure 23: Data regime and latent geometry on SPHEROENV.

F ABLATIONS ON ASSUMPTIONS AND DESIGN CHOICES

In this section, we study the learning behavior of AC-WM in a controlled way, by breaking one of the hypotheses of Section 2 at a time. We use problem settings of analytic systems with explicit f, O and R in varying dimensionality. We consider the noised observation and reward functions

$$O(s, W) = \tanh\big(C \cdot u(s, W) + b\big) + \eta, \qquad \eta \sim \mathcal{N}(0, \sigma_o^2 I) \tag{20}$$

$$R(s, W) = \exp\big(-(\delta(s, W)/\sigma_R)^2\big) \tag{21}$$

with $u(s, W) \in \mathbb{R}^q$ carrying all context-dependence, $C \in \mathbb{R}^{3q \times q}$ and $b \in \mathbb{R}^{3q}$ context-independent constants drawn once, δ a deviation that vanishes exactly on $\mathcal{M}_W$, where σ_R acts as a normalizing factor. If not stated otherwise, we use $u(s, W) = s - s_W$ and $\delta(s, W) = \|s - s_W\|$, with s_W the target state of the episode. The action set is the box $\mathcal{A} = [-1, 1]^d$. Unless stated otherwise, the experiments gather $n = 1500$ anchored episodes of length $T = 10$ collected at action scale $\alpha = 0.3$, use early stopping with patience 20 in training, and 100 evaluation episodes of 60 steps at the minimal probing budget over 3 seeds each seeding the dataset, the training and the evaluation.

In most figures, we aggregate performance over evaluation steps, episodes, and seeds. Here, a single evaluation episode is represented by the running mean over five consecutive steps of its reward, maximized over the windows that start in the second half of the evaluation episode. Multiple evaluation episodes are aggregated by the median over the pooled episodes to get robust statistics.

F.1 DYNAMICS WITH NO ACTION-COORDINATES

In this experiment, we want to understand if AC-WM can solve tasks where the underlying dynamic does not admit an action-coordinate system.

Let $\mathcal{P} = \mathbb{R}^2$, let $\mathcal{A} = [-1, 1]^2$ and define for $\lambda \in \mathbb{R}$

$$f_\lambda(s, a, W) := s + \mathrm{Rot}(\lambda s_1)\, A_W\, a \qquad \mathrm{Rot}(t) := \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix} \tag{22}$$

with an invertible A_W drawn per episode. Then f_λ is smooth and driftless because $\mathrm{Rot}(\lambda s_1) A_W$ is invertible, which also makes its actuation Jacobian $J(s) = \mathrm{Rot}(\lambda s_1) A_W$ invertible everywhere, so f_λ is fully actuated. For $\lambda = 0$ it is Example 2 and thus f_λ admits a full action-coordinate system.

Next, we show that for $\lambda \neq 0$, it admits no action-coordinate system of any rank. Assume otherwise and let (g, B) be one of rank ℓ and let $b^\top$ be a row of B, which is non-zero as B has rank ℓ. By Proposition 4 the corresponding coordinate g_i is smooth with

$$\nabla g_i(s)^\top = b^\top J^{-1}(s) = c^\top \mathrm{Rot}(-\lambda s_1)$$

with $c^\top := b^\top A_W^{-1} \neq 0$. Then $\partial_1 g_i = c_1 \cos(\lambda s_1) - c_2 \sin(\lambda s_1)$ and $\partial_2 g_i = c_1 \sin(\lambda s_1) + c_2 \cos(\lambda s_1)$. Neither depends on s_2, so $\partial_2 \partial_1 g_i = 0$, while on the other hand $\partial_1 \partial_2 g_i = \lambda\,(c_1 \cos(\lambda s_1) - c_2 \sin(\lambda s_1))$. Since g is smooth, both mixed partials must agree, so $\lambda\,(c_1 \cos(\lambda s_1) - c_2 \sin(\lambda s_1)) = 0$ for every s_1 and since $c \neq 0$, $\lambda = 0$ follows.

Yet, no canonical measurement that measures the strength of *not having* an action-coordinate system exists, as this is a binary property. In our experiment, we use the LPE constant L_f of Schmähling et al. (2026, Definition 3.4) (Section A.5) as a proxy for this as this is a generalization of reversibility and can be computed from the equations of the system alone. Figure 24 sweeps λ over the non-zero range $[0.1, 16]$, about two decades, and shows the performance of AC-WM. There, we see that AC-WM copes with deviations up to $\lambda = 2$, starts to fail at $\lambda = 4$, where the three seeds split, two collapsing and one still near-perfect, and has fully collapsed by $\lambda = 8$.

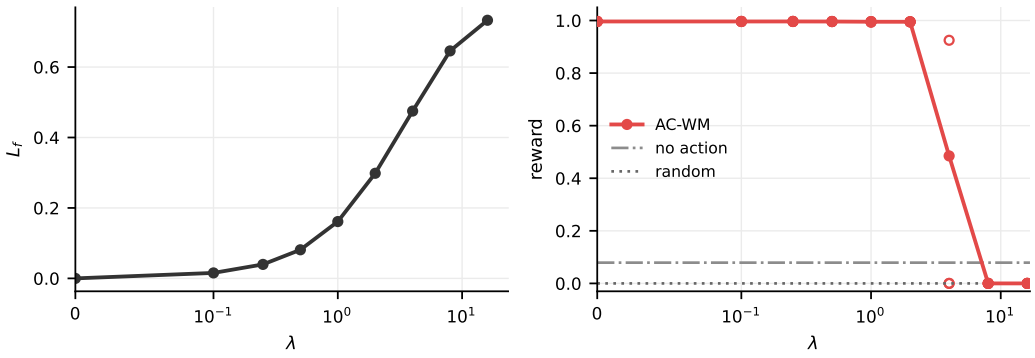


Figure 24: Dependence of AC-WM on existence of action-coordinate systems. *Left:* L_f of Schmähling et al. (2026), computed from the equations of the system and involving no model. *Right:* the reward AC-WM reaches.

F.2 Dependence on latent dimension ℓ

A crucial hyperparameter of AC-WM is the latent dimension ℓ of the encoder, which is the rank of the R-anchored action-coordinate system it is asked to learn. In most applications, we argue that ℓ is known from the physics of the system. Nevertheless, we study the effect of ℓ on the performance of AC-WM in a controlled setting.

Here, we use two systems with $d = 3$ actuators, one with a full-rank target coordinate system and one with a rank-2 one trained for every feasible $\ell \in \{1, 2, 3\}$. The first dynamics is $f(s, a, W) = s + A_W a$ on $\mathbb{R}^3$, whose anchored system is full and has rank 3. The second dynamics is a three-dimensional extension of Example 5 with $h(t) = t^2/2$ and a second tool at the height s_3 and ramp $-\frac{2}{3}h$:

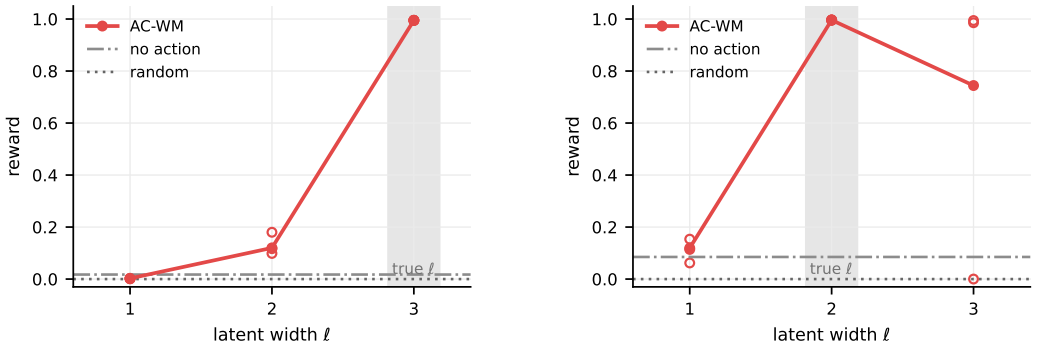
$$f(s,a) := \big(s_1+\alpha a_1,\ s_2+\alpha a_2+h(s_1+\alpha a_1)-h(s_1),\ s_3+\alpha a_3-\tfrac{2}{3}\big(h(s_1+\alpha a_1)-h(s_1)\big)\big). \quad (23)$$

Each tool is asked for a clearance of its own above its ramp, so with the clearances $c_W \in \mathbb{R}^2$ of the episode the deviation is $\delta(s, W) := \|g(s)\|$ for the pair of clearance deviations $g(s) := \big(s_2 - h(s_1) - c_{W,1},\ s_3 + \frac{2}{3}h(s_1) - c_{W,2}\big)$. Two of the three coordinates are hence pinned and the carriage is free, which makes $\mathcal{M}_W = g^{-1}(0)$ a connected 1-manifold and $(g, \alpha\,(e_2\ \ e_3)^\top)$ an action-coordinate system anchored at it of rank $\ell = 2$.

Figure 25 shows the performance of AC-WM when ℓ varies for both systems, showing that the performance of AC-WM is best when ℓ matches the true rank of the system, and degrades when ℓ is too small or too large.

F.3 Dependence on history window k

Next, we study the effect of the history window k on the performance of AC-WM. We use the canonical system $f(s, a, W) = s + A_W a$, which is precisely the setting studied in Proposition 6, where it is shown that if the gain B_W is fitted per episode, then $k = 1$ suffices, whereas if the gain is pinned to I, then $k = d + 1$ is required. Here, we use the observation function $O(s, W) = s - s_W$ and reward $R(s, W) = -\|s - s_W\|$ with randomized target s_W to make learning easier and to focus on the effect of k and the gain. All experiments take place in $d = 2$ and we vary $k \in \{1, 2, 3, 4\}$.



(a) Task with $\mathcal{M}_W$ a single point and anchored system has $\ell = 3$.

(b) Task with $\mathcal{M}_W$ a 1-manifold in $\mathbb{R}^3$ and system anchored at it has rank $\ell = 2$.

Figure 25: The latent width ℓ against the rank of the system anchored at $\mathcal{M}_W$.

The results are in Figure 26. There, we see that when the gain is fitted, $k = 1$ suffices to obtain the best possible performance, already for a very small amount of data. Interestingly, the performance degrades when k is increased and even larger amounts of data cannot prevent this degradation. This is probably due to the fact that this saturation effect in the latent space blocks efficient interplay between chart and gain. On the other hand, when the gain is pinned to I, we see that learning improves the larger k is, but even at $k = 3$, which provides realizability, the performance is not yet as good as when the gain is fitted. This emphasizes that realizability is not necessarily learnability: Proposition 6 states that an encoder with window $k = d + 1$ exists, not that training finds it from finitely many episodes. The pinned arm has to learn the inversion of A_W from data, which the fitted arm obtains per episode in closed form from equation 4.

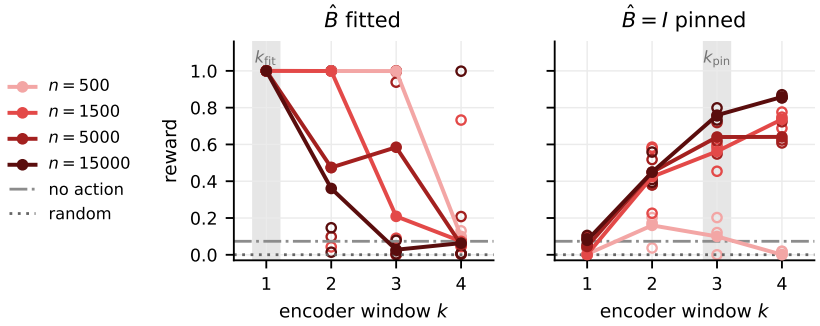


Figure 26: The encoder window k against the amount of data when the gain is fitted or pinned.

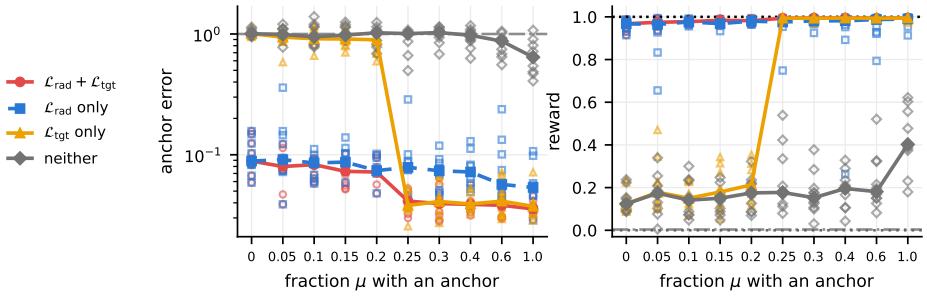
F.4 Contributions to anchoring mechanism

This section explores the contributions of the loss-terms $\mathcal{L}_{\text{rad}}$ and $\mathcal{L}_{\text{tgt}}$ to the anchoring mechanism of AC-WM. We take $f(s, a, W) = s + A_W a$ and O from equation 20 with $u(s, W) = s - s_W$ and use AC-WM with $k = 1$ and fit B in every episode, which Proposition 6 shows is theoretically enough to learn the system. Although using the boxed action space $\mathcal{A} = [-1, 1]^d$ does not give precise radial rewards $\|\cdot\|_{A_W}$, the reward is still strictly decreasing in $\|s - s_W\|$. The proposition only settles representability, but as every translate $(g_W + v, A_W)$ is again an action-coordinate system by Proposition 3, of which only $v = 0$ is anchored, so the question is which term makes training select that one. As all other loss terms are invariant under translation, the ablation on $\mathcal{L}_{\text{rad}}$ and $\mathcal{L}_{\text{tgt}}$ constitutes a complete ablation of the anchoring mechanism.

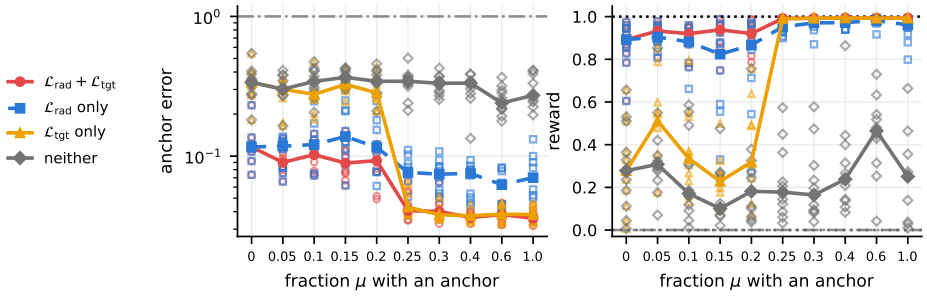
As the performance of the final policy alone can not distinguish between a latent that is not anchored and a model whose training has not converged, we instead measure how well the anchoring mech-

anism works by measuring $\|e_\theta(O(s_W, W))\|$ in the scale of the latent coordinates. As that scale is itself fixed only up to the gauge of Proposition 3, we read the number against a reference measured the same way, the median radius $\|e_\theta(O(s, W))\|$ over reset positions s.

We train AC-WM with and without $\mathcal{L}_{\text{rad}}$ and $\mathcal{L}_{\text{tgt}}$ respectively when the fraction μ of anchored episodes in the training data varies. Those episodes are collected anchored, starting at the target and hence carrying a frame for $\mathcal{L}_{\text{tgt}}$, and the remaining $1 - \mu$ from a random start displaced by a fixed $c \neq 0$, that is drawn around $s_W + c$ and not around the target. Were both regimes centred at s_W, the observation $O(s_W, W)$ would sit at the centre of the data, and any bias that puts the latent origin at the mean of the data would anchor the model for free: the measurement would then report an anchor that no loss term produced. We run this for $d \in \{2, 3\}$ on 500 collected episodes of length 10, over ten seeds, and set $\|c\|$ to one half-width of the reset box. Figure 27 shows the result for $d = 2$ and $d = 3$. Below a threshold of anchored episodes, $\mathcal{L}_{\text{tgt}}$ is inactive and only using it in training does not provide any anchoring. In these data regimes, the anchoring is essentially carried by $\mathcal{L}_{\text{rad}}$ alone. The term $\mathcal{L}_{\text{tgt}}$ unfolds its effect once enough episodes carry a target frame, boosting the anchoring of the latent coordinates. Interestingly, here, anchoring seems solely carried by $\mathcal{L}_{\text{tgt}}$ with only marginal better anchoring if both terms are used. Clearly, using neither term introduces invariances in the latent space making effective control impossible.



(a) $d = 2$.



(b) $d = 3$.

Figure 27: Anchoring error and reward over fraction of anchored episodes in dataset.

F.5 Closed-form heuristics against optimization over action space

Section 3.3 calls $\pi_{\hat{B},\varepsilon}$ a heuristic where $\ell < d$, and Remark 2 gives the program the step-optimal action solves there.

We analyze these two policies on a system highly over-actuated, with $d = 5$ actuators and $\ell = 2$ latent coordinates. Concretely, we use $f(s, a, W) = s + A_W \cdot a$ on $\mathbb{R}^2$, actions from $\mathcal{A} = [-1, 1]^5$ and the deviation $\delta(s, W) = \|s - s_W\|$, where $A_W := \frac{\alpha}{\|A\|_2} \cdot R_W \cdot A$ with

$$A := \left(\begin{smallmatrix} 1 & 2 & 4 & 8 & 16 \\ 16 & 8 & 4 & 2 & 1 \end{smallmatrix}\right) \in \mathbb{R}^{2\times 5},$$

the spectral norm $\|\cdot\|_2$ and a rotation $R_W \in \mathbb{R}^{2\times 2}$ drawn uniformly per episode. Particularly, the context is the pair (s_W, R_W). An action-coordinate system anchored at $\mathcal{M}_W = \{s_W\}$ is (g_W, A_W)

with $g_W(s) = s - s_W$. At latent z, a step-optimal action is hence a $\|\cdot\|_\infty$-smallest solution of $A_W a = -z$, whose solutions form an affine subspace of dimension $d - \ell$, while $\pi_{\hat{B},\varepsilon}$ computes the Euclidean-smallest one. The special shape of A makes the two action point in very different directions, as in Remark 1. We run the two policies, the closed-form heuristics $\pi_{\hat{B},\varepsilon}$ from equation 3 and the step-optimal policy $\pi^*_{\hat{B}}$ from Theorem 2 obtained from a solution of

$$\operatorname*{arg\,min}_{(a,t)\in\mathbb{R}^{d+1}} \left\{ t \;:\; \hat{B}a = -z,\; -t\cdot\mathbf{1}_d \le a \le t\cdot\mathbf{1}_d \right\}$$

and scaled to $\mathcal{A}$ as explained in Remark 2. Figure 28 shows the reward against the evaluation step where we see that the closed-form heuristic also reaches the optimum but requires more steps.

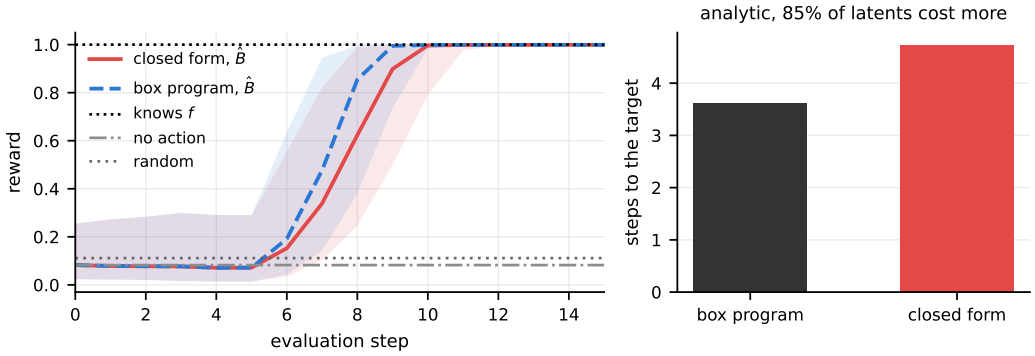


Figure 28: Performance of $\pi^*_{\hat{B}}$ and $\pi_{\hat{B},\varepsilon}$ on an over-actuated system with $\ell = 2$ and $d = 5$.

G Implementation details

G.1 Tensions in loss terms

All experiments share one set of weights, $\lambda_{\text{dyn}} = \lambda_{\text{con}} = \lambda_{\text{rad}} = \lambda_{\text{var}} = 1$, $\lambda_{\text{tgt}} = 2$ and $\lambda_{\text{rew}} = 0.1$, together with the anisotropy threshold $\nu = 4$ in equation 6 and the variance floor $\sigma_{\min} = 1$. Two environments deviate. On LensAlignment and LightTunnel, we use $\lambda_{\text{tgt}} = 32$, and on LensAlignment additionally $\lambda_{\text{dyn}} = 0.03$ for the MTF observation and $\lambda_{\text{dyn}} = 0.06$ for the rendered image.

These loss terms create tensions in the training, the sharpest is between $\mathcal{L}_{\text{tgt}}$ and $\mathcal{L}_{\text{var}}$: the first contracts the frames of $\tau \cap \mathcal{T}_C$ onto the origin, the second keeps a floor under the spread of every latent coordinate over the batch. The frames in $\mathcal{T}_C$ are a very small part of a batch and contracting them alone barely moves that spread. What the floor resists is the global shrinking of the latent by which the encoder achieves it. Raising the weight of $\mathcal{L}_{\text{tgt}}$ therefore need not sharpen the anchor, as it contracts the latent near the target and away from it alike, and the resolution improves by far less than the weight. The floor competes with $\mathcal{L}_{\text{con}}$ as well, since a latent of larger scale carries a gain of larger determinant, which the first term of equation 6 pins to 1.

G.2 Implementation of $\text{scale}_{\mathcal{A}}$

In this section we give a closed form for the action $\text{scale}_{\mathcal{A}}(v)$ of equation 3. Writing $\|\cdot\|_I$ for $\|\cdot\|_B$ with $B = I$, the gauge of the action set itself whose closed unit ball is $\mathcal{A}$, it is

$$\text{scale}_{\mathcal{A}}(v) := \frac{v}{\max\{1, \|v\|_I\}},$$

so a v outside $\mathcal{A}$ is pulled back onto its boundary and one inside stays untouched. This is not a component-wise saturation: for the box $\mathcal{A} = [-1,1]^d$ it divides by $\max\{1, \|v\|_\infty\}$ rather than clipping every coordinate, so the action keeps the direction the pseudoinverse picked and only its length is cut.

G.3 NUMERICAL STABILITY

For numerical stability we compute $|\det \hat{B}|$ in equation 6 as the product of the diagonal of R in a QR decomposition of $\hat{B}^\top$, and the action equation 3 as the least-squares solution of the stacked system $[\hat{B}; \sqrt{\varepsilon} I]\, a = [-z; 0]$. Both are the quantities written above, but neither forms $\hat{B}^\top \hat{B}$ or $\hat{B}\hat{B}^\top$, whose condition number is the square of that of $\hat{B}$. The per-episode estimates equation 4 are occasionally near-singular in every task, and there the normal equations lose the gradient of equation 6 and perturb the planned action in single precision, while the QR and least-squares forms are backward stable.