SAME STATES, DIFFERENT ACTUATORS: WHAT SINGLE-ACTUATOR DATA MUST COVER FOR COMPOSITIONAL WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Compositional world models aim to predict unseen actuator combinations without collecting data for every combination. For torque-driven systems, the joint response at a fixed state is the sum of single-actuator responses up to a residual of 1.7–13.6% of the actuated response, so single-actuator data suffice if each actuator's response is known at the states the combination visits. Reset-only collection observes each actuator only at states reached by that actuator alone from reset, and existing coverage measures cannot flag this: concentrability is unbounded for joint actions under any single-actuator data, and state coverage ignores which actuator is observed at which state, the dataset's *actuator–state pairing*. A matched-pool intervention changes which actuator is executed at each state of a fixed pool, keeping per-actuator action marginals fixed, and so isolates the pairing. In a linear switched class, we prove that histories engaging only a proper subset of actuators, as reset-only histories do, can leave joint behaviour undetermined even though together they reach every state, and we give a necessary and sufficient condition for component observations to determine it. A collector must therefore let a history engage several actuators; state-carrying collection does so by switching actuators without resetting. Across four MuJoCo environments and two model classes, balanced pairing reduces one-step prediction error on unseen combinations by 8.3–29.2%, and by 8.4–25.4% when the joint prediction is assembled from the model's own single-actuator responses. At the same interaction budget, state-carrying collection reduces one-step error by 6.3–34.6% relative to reset-only collection, and its advantage extends to 120 and 98 of 121 unseen actuator subsets for the two model classes and to ten-step rollouts. The artifacts are available at https://figshare.com/s/2aec1f48b5d5a038aca6.

1 INTRODUCTION

Compositional world models aim to predict new combinations of familiar components without collecting data for every combination (Neary & Topcu, 2023; Zhang et al., 2025). For systems with multiple actuators, this goal is particularly relevant when data collection is restricted to one actuator at a time, as in joint-wise excitation for robot dynamics identification (Wang et al., 2025). Such experiments provide observations of individual actuators, whereas the behaviours a model must predict can involve several actuators acting together. This raises a basic data question: *what must single-actuator observations contain to support prediction of unseen joint actions?*

For articulated systems driven by joint torques, the rigid-body equations of motion are affine in the applied torques (Featherstone, 2008). At a fixed state, the change in acceleration produced by several actuators together is therefore the sum of the changes each produces alone, up to contact forces and time discretisation, whose effect Section 3.3 measures. Single-actuator data can then determine a joint prediction, provided each actuator's response is known at the states where the combination acts.

Existing practice leaves this proviso unaddressed. Joint-wise excitation for dynamics identification can observe each joint along its own trajectories because a rigid-body model is linear in a finite set of inertial parameters, so data from any trajectory constrain the same parameters (Wang et al., 2025). A learned world model lacks this structure and predicts an actuator's response reliably only near states

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

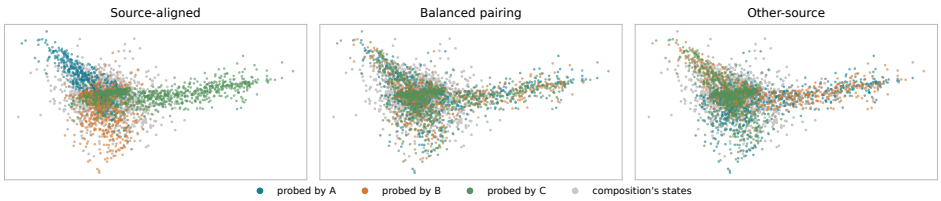

Figure 1: Actuator–state pairings on the `Hopper` matched pool. Each panel shows the same projection of the pooled states, with directions chosen to separate the actuators that prepared them; colours mark the actuator probed at each state. SOURCE-ALIGNED probes each state with the actuator that prepared it, BALANCED PAIRING spreads probes over all actuators, and OTHER-SOURCE excludes the preparing actuator. Grey points are states visited under joint actuation. The state pool and per-actuator action marginals are identical across panels.

where that actuator was observed. Under reset-only collection, which holds one actuator throughout each history, these are states the actuator *prepared* itself, that is, reached by its own actions, whereas joint actuation can drive the system into states that no single actuator reaches alone. Existing coverage notions do not register this mismatch. State–action concentrability (Chen & Jiang, 2019; Rashidinejad et al., 2021) is unbounded for a joint-action target under every single-actuator dataset, and state coverage is computed from the pooled states, so it cannot see the dataset's *actuator–state pairing*, that is, which actuator is observed at which state.

We say a dataset has *per-actuator context coverage* when the states at which each actuator is observed cover the states where the target composition acts; pooled coverage asks this only of all observed states together (Section 2). We ask three questions: does per-actuator coverage matter for learned world models when pooled states are fixed, what exactly must it satisfy for component observations to determine composition, and how can a physical collector obtain it?

For the first question we use a *matched-pool intervention* (Figure 1). Starting from a fixed pool of states prepared by reset-only histories, we construct datasets that differ only in which actuator is *probed*, that is, executed, at each state: SOURCE-ALIGNED probes each state with the actuator that prepared it, as reset-only data do; BALANCED PAIRING spreads probes evenly over all actuators; and OTHER-SOURCE excludes the preparing actuator. Each actuator is thus probed at its own states only, at both kinds of state, or at states prepared by others only. The state multiset and per-actuator action marginals stay fixed, and each probe is executed from the restored simulator state.

For the second question we move to a linear switched class in which engaging an actuator adds a position-dependent force and a separate known input drives the system, so each actuator's response still depends on the state. A history can pair actuator j with a state prepared by actuator k only if it engages both, so the set of actuators a history engages, its *actuator set*, bounds the pairings a collector can obtain; reset-only histories have actuator sets of size one. We construct two systems that differ in one actuator effect and compare their records under every history whose actuator set is a proper subset. Every position coordinate is observed and every actuator responds to the input, so neither unobserved states nor inactive actuators can account for an ambiguity. We then derive a *certificate*, a necessary and sufficient condition on the positions at which each actuator is observed, for component observations to determine the trajectory under every input and actuator schedule.

For the third question, balanced pairing restores arbitrary simulator states, which a physical collector cannot do. Guided by the linear analysis, we design *state-carrying collection*, which enlarges each history's actuator set by switching the active actuator without resetting the physical state (Figure 3), so each actuator is also probed at states prepared by others. The protocol uses a fixed switching schedule, which outperformed a certificate-inspired rule in exploratory runs (Appendix G). Its histories engage up to four actuators, a proper subset in three of the four environments, and the collection gain holds in all four.

Contributions. (1) We identify actuator–state pairing, which actuator is observed at which state, as a property of single-actuator data that pooled coverage cannot see, and reset-only collection as the case in which each actuator is observed only at states it prepared itself (Section 2). (2) In a linear switched class, we prove that histories confined to proper subsets of actuators can leave composition undetermined although together they reach the full phase space, and give a certificate

stating exactly when component observations determine every composition (Section 4). (3) We propose state-carrying collection, which switches the active actuator without resetting the physical state; at a matched budget it lowers one-step error by 6.3–34.6% relative to reset-only collection, and its advantage extends to 120 of the 121 unseen actuator subsets for the MLP ensemble and 98 for the latent model (Section 5).

2 Problem Setting

Single-actuator collection. We consider a fully observed system with m actuators and dynamics $s' = F(s, a)$, where $s \in \mathbb{R}^d$ and $a \in [-1, 1]^m$. A feasible collection protocol gathers a dataset $\mathcal{D} = \{(s_i, a_i, s'_i)\}_{i=1}^N$ using only actions $a = \alpha e_j$, where $j \in [m]$ is the selected actuator, $\alpha \in [-1, 1]$ is its amplitude, and e_j is the jth coordinate vector, so every executed action satisfies $|\operatorname{supp}(a)| \leq 1$. A history begins with an environment reset and lasts at most H transitions, ending earlier on environment termination, and the protocol may switch actuators within it while retaining the physical state. A state is *prepared* by the actions that lead to it from the reset, and by the actuators that executed them; executing αe_j from a state *probes* actuator j there. The actuators whose actions a history executes form its *actuator set*. Reset-only collection holds one actuator throughout each history, so its actuator sets have size one and each actuator is probed only at states it prepared. All executed transitions, including those that prepare states for later probes, count towards the budget N. Feasible collectors cannot restore arbitrary states, set initial states, or execute target behaviour; conditions that use such access serve as diagnostics or references.

Compositional prediction. A world model $\hat{f}(s, a)$ is trained on $\mathcal{D}$ to predict the state increment $\Delta(s, a) = F(s, a) - s$ (Chua et al., 2018; Janner et al., 2019). A target is a subset $T \subseteq [m]$ with $|T| \geq 2$. Its evaluation distribution $\mu_T(s, a)$ draws states visited from reset under random actions restricted to T and pairs them with random joint actions on T; its state marginal $\rho_T(s)$ is the composition's *context*. The main target is $T = [m]$, the all-on target, and Section 5.4 evaluates other subsets. Our primary metric is

$$\mathcal{E}_T(\hat{f}) = \mathbb{E}_{(s,a)\sim\mu_T} \left\| \hat{f}(s, a) - \Delta(s, a) \right\|_2, \tag{1}$$

where $\hat{f}$ denotes predictions expressed in the original state coordinates, after reversing any training normalisation.

Pooled and per-actuator context coverage. The dataset induces a pooled state distribution $p_{\mathcal{D}}(s) = \sum_j \pi_j \, p_{\mathcal{D}}(s \mid j)$, where π_j is the fraction of transitions that select actuator j and $p_{\mathcal{D}}(s \mid j)$ is the distribution of states at which j is probed; these components describe the actuator–state pairing. Pooled coverage asks that $p_{\mathcal{D}}$ place mass near every state where ρ_T does; *per-actuator context coverage* asks the same of every component $p_{\mathcal{D}}(\cdot \mid j)$, $j \in T$. Datasets can share the pooled mixture and per-actuator action marginals while differing in every component, so no statistic of the pooled states distinguishes them. Section 3 varies these components by intervention, and Section 4 states exactly what they must satisfy in a linear class.

3 Isolating Per-Actuator Context Coverage

This section tests whether actuator–state pairing affects compositional prediction when pooled state coverage is held fixed. We use four MuJoCo environments, `Walker2d`, `HalfCheetah`, `Hopper`, and `Ant`, with six, six, three, and eight actuators (Todorov et al., 2012; Towers et al., 2025), and two full-state learners, a five-member MLP ensemble (Lakshminarayanan et al., 2017; Chua et al., 2018) and a deterministic latent model that encodes the full state. Each dataset contains $48\,000$ transitions, and unless otherwise stated evaluation uses the all-on target of Section 2. Each condition uses five collection draws with two network initialisations each, and we report paired percentage reductions with descriptive 95% intervals across draws (Appendix D.2).

3.1 Same States, Different Actuator Pairings

We construct three datasets on a common pool of $48\,000$ states prepared by reset-only histories, with equally many states from each *source actuator*, the actuator that prepared the state (Figure 1). The

Table 1: One-step prediction error under the three matched-pool pairings. Direct scores the model's joint prediction, and Assembled scores the joint prediction built from the same model's single-actuator responses (Section 3.3); both are compared with the true joint increment at the same all-on target states. The first two columns give SOURCE-ALIGNED error in each environment's physical units, and the remaining columns give paired percentage reductions relative to it, with positive values favouring the reassignment. BALANCED abbreviates BALANCED PAIRING. Intervals are 95% intervals across five collection draws after averaging two initialisations.

		SOURCE-ALIGNED error		BALANCED (%)		OTHER-SOURCE (%)	
Environment	Learner	Direct	Assembled	Direct	Assembled	Direct	Assembled
Walker2d	MLP	5.14	5.26	16.1 ± 1.4	12.3 ± 1.6	-9.3 ± 3.3	-11.6 ± 3.1
	Latent	6.04	6.21	29.2 ± 2.2	25.4 ± 2.5	8.5 ± 1.2	6.2 ± 1.6
HalfCheetah	MLP	3.62	3.65	15.3 ± 1.4	14.2 ± 1.3	11.9 ± 2.5	11.2 ± 2.4
	Latent	5.03	4.94	25.4 ± 1.6	23.8 ± 1.6	24.3 ± 1.5	22.6 ± 1.3
Hopper	MLP	0.236	0.236	13.5 ± 1.1	12.7 ± 1.3	-18.1 ± 7.6	-18.4 ± 7.3
	Latent	0.258	0.253	19.6 ± 3.5	19.5 ± 3.1	-16.6 ± 2.4	-18.0 ± 2.4
Ant	MLP	6.80	6.73	9.8 ± 1.5	9.8 ± 1.6	1.5 ± 1.0	1.2 ± 1.0
	Latent	8.41	8.29	8.3 ± 2.2	8.4 ± 2.2	3.1 ± 1.8	2.6 ± 1.9

pool is grouped into blocks that hold one state from each source actuator and share one amplitude, and the datasets differ only in which actuator is probed at each state of a block. (1) SOURCE-ALIGNED probes every state with its source actuator. This reproduces the pairing of reset-only data, in which each actuator is observed only at states it prepared. (2) BALANCED PAIRING shifts the actuator indices cyclically by an offset drawn uniformly from $\{0, \ldots, m-1\}$, so each actuator is probed both at its own states and at states prepared by others. (3) OTHER-SOURCE shifts them by an offset drawn from $\{1, \ldots, m-1\}$, so each actuator is probed only at states prepared by others; this pairing tests whether an actuator's own states are needed. The state multiset, each actuator's amplitude multiset, the train–validation split by history, and the normalisation are identical across the three datasets, and neither target behaviour nor joint actions enter the construction. Every probe is executed from the restored simulator state, so each dataset consists of valid transitions (Appendix D).

Table 1 reports the one-step error of each model's direct joint prediction. Relative to SOURCE-ALIGNED, BALANCED PAIRING reduces the error by 8.3–29.2% in all eight environment–learner combinations. OTHER-SOURCE ranges from an 18.1% increase to a 24.3% reduction, with deterioration for both learners on `Hopper` and for the MLP ensemble on `Walker2d`. Because the state pool is unchanged, these results show that the pairing itself carries predictive information that the pooled state distribution leaves undetermined, and that probes at an actuator's own states and at states prepared by others are both needed. Pooled statistics are identical across the three datasets by construction, and per-actuator nearest-neighbour distances do not rank the datasets consistently with prediction error either (Appendix F.5), so we compare pairings by their effect on prediction.

3.2 PASSIVE DYNAMICS AND ACTUATOR RESPONSES

A joint prediction consists of a passive part, the increment under the zero action, and an actuator-driven part, and balanced pairing could improve either. To separate them, we query the simulator at each evaluation state s with the zero action and with each single-actuator action:

$$\begin{aligned} \Delta_0(s) &= F(s,0) - s, \quad & r_j(s,\alpha) &= F(s,\alpha e_j) - F(s,0), \\ \hat{\Delta}_0(s) &= \hat{f}(s,0), & \hat{r}_j(s,\alpha) &= \hat{f}(s,\alpha e_j) - \hat{f}(s,0). \end{aligned} \tag{2}$$

We then measure the passive-increment error $\|\hat{\Delta}_0 - \Delta_0\|_2$ and the actuator-response error $\|\hat{r}_j(s,a_j) - r_j(s,a_j)\|_2$ at the evaluation state–action pairs, where a_j is the jth component of the joint action, averaging the latter over actuators. These simulator queries are diagnostic; training uses only the collected datasets.

Figure 2 shows the reductions in the two errors from SOURCE-ALIGNED to BALANCED PAIRING. Actuator-response error falls by 17.6–38.0%, whereas the passive-increment reduction ranges from -9.7 to 15.4%. The relative response improvement is larger in all eight comparisons, and for the latent model on `Ant` the response error falls while the passive error rises. The intervention therefore acts mainly on the prediction of individual actuator effects.

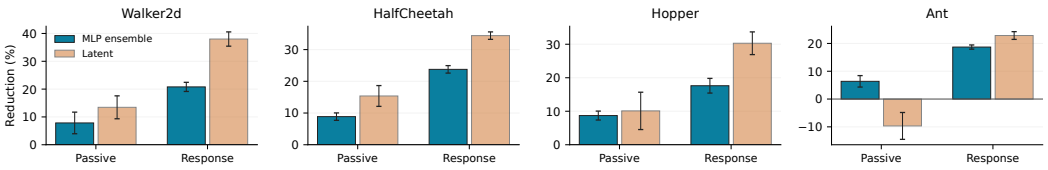

Figure 2: Passive-increment and actuator-response errors under balanced pairing. Paired percentage reductions from SOURCE-ALIGNED to BALANCED PAIRING in the two diagnostic errors at identical evaluation states, each relative to its own error. Error bars are 95% intervals across five collection draws.

We use two controls to check that this gain comes from actuator-specific information. First, we train and evaluate an auxiliary MLP ensemble with its action inputs set to zero, so that it can learn only the action-averaged increment. Its error reduction from SOURCE-ALIGNED to BALANCED PAIRING ranges from -0.3 to 1.1%, against 9.8–16.1% for the action-aware ensemble, so the gain requires knowing which actuator acted. Second, we measure held-out error on each dataset's own collected distribution. Balanced pairing raises this error in all eight comparisons, by 1.6–19.9%, so the gain on the target does not come from an easier fitting problem (Appendices E and E.3).

3.3 FROM COMPONENT RESPONSES TO JOINT PREDICTIONS

Finally, we examine how the component responses relate to the joint prediction. We form the assembled predictions, which add the passive increment and the single-actuator responses:

$$\Delta_{\text{sum}}(s,a) = \Delta_0(s) + \sum_{j\in T} r_j(s,a_j), \qquad \hat{\Delta}_{\text{sum}}(s,a) = \hat{\Delta}_0(s) + \sum_{j\in T} \hat{r}_j(s,a_j). \tag{3}$$

For the simulator, the residual $\mathbb{E}\|\Delta - \Delta_{\text{sum}}\|_2$ is 1.7–13.6% of the mean actuated response $\mathbb{E}\|\Delta - \Delta_0\|_2$ and 6.3–42.5% of the SOURCE-ALIGNED target error. The additivity premise of Section 1 therefore holds in these tasks to within a residual that is below half of the SOURCE-ALIGNED target error in every environment.

The Assembled columns of Table 1 report the error of the assembled prediction against the true joint increment. Balanced pairing reduces it by 8.4–25.4% in all eight comparisons, close to its 8.3–29.2% reduction of the direct error, and OTHER-SOURCE improves five of eight under both measures, with the same deteriorations. The component information supplied by the pairing therefore carries through to a joint prediction assembled explicitly from components, as well as to the network's direct extrapolation.

4 WHEN COMPONENT OBSERVATIONS DETERMINE COMPOSITION

Section 3 shows that, with states fixed, the pairing changes how well individual responses are learned, but a learned model cannot tell whether a pairing determines composition in principle. We therefore ask exactly which pairings do, in a linear switched class with additive actuator effects. There, the positions at which actuator j is observed play the role of $p_{\mathcal{D}}(\cdot \mid j)$, and the span of reachable states contains the support of every ρ_T. The analysis concerns exact observations and known background dynamics:

$$M\ddot{x} + C\dot{x} + \left(A + \sum_{j=1}^{m} q_j(t) S_j\right) x = Bu(t), \qquad q(t) \in \{0,1\}^m, \tag{4}$$

where $x \in \mathbb{R}^n$, the matrices $M \succ 0$, $C \succeq 0$, $A \succ 0$, and B are known, and each unknown symmetric effect satisfies $0 \preceq S_j \preceq L_j I$ for a known cap L_j. Engaging actuator j adds the position-dependent force $-S_j x$, so an observation of j at x reveals $S_j x$, the linear counterpart of r_j; amplitudes are replaced by on–off switches, and a known input u drives the system. The switches are externally selected, and a history starts from $x = \dot{x} = 0$ and records the input and the full position trajectory.

A three-dimensional example shows how the pairing enters. Each observation of actuator j at a position x reveals $S_j x$. Probing each actuator three times at a single position e_j of its own, the analogue of SOURCE-ALIGNED, and probing every actuator once at each of e_1, e_2, e_3, the

analogue of BALANCED PAIRING, give the same pooled positions and counts. The first reveals one direction of each S_j and leaves composition ambiguous; the second reveals all three and determines it (Appendix C).

Theorem 1 (Proper-subset data can leave composition undetermined)**.** *For every $m \geq 2$, there exist two systems of the form (4) with $n = m$, differing in one actuator effect, such that: (a) every S_j has rank at most two and $S_j B \neq 0$; (b) every history from zero whose actuator set is a proper subset $I \subsetneq [m]$, including arbitrary inputs, switching or simultaneous activation within I, and adaptive selection of further histories, has the same complete input–output record in both systems; (c) engaging actuators $1, \ldots, m-1$ simultaneously from rest reaches the full $2m$-dimensional phase space; and (d) the responses to a common unit-step input with every actuator engaged differ.*

In the construction (Appendix A), actuator $k < m$ couples position coordinates k and $k+1$, and the two systems differ only in how actuator m acts on coordinate m. The input enters at coordinate 1, where actuator m also acts through a known term, so it responds to the input identically in both systems. The difference appears only once coordinate m is displaced, which requires passing motion along the chain $1 \to 2 \to \cdots \to m$. A history that engages actuator m but omits some link k never displaces coordinate m, and a history without actuator m never engages the unknown effect, even though actuators $1, \ldots, m-1$ engaged simultaneously reach every state. A separating history must therefore engage every actuator; switching through them one at a time in chain order prepares coordinate m and then probes actuator m there. Proper-subset histories never observe actuator m where $x_m \neq 0$, so pooled coverage is complete while per-actuator coverage is not; reset-only collection, with actuator sets of size one, is covered by (b).

To state exactly when observations suffice, let Y_j contain the observed positions for actuator j, with exact responses $S_j Y_j$. The effects consistent with these observations and the bounds form an interval $S_{j,-} \preceq S_j \preceq S_{j,+}$ (Appendix B derives the endpoints), whose width $W_j = S_{j,+} - S_{j,-}$ determines the subspace $K_j = \ker W_j$ on which all compatible effects agree, with $\operatorname{span}(Y_j) \subseteq K_j$. Build a nominal system from the lower endpoints $S_{j,-}$, let $\mathcal{R}$ be the span of phase states it reaches from zero under arbitrary inputs and histories that switch freely among actuators with one active at a time, and let $P_x(x, \dot{x}) = x$.

Theorem 2 (Composition certificate)**.** *Every input and every switching or simultaneous schedule produces the same trajectory for all compatible actuator effects if and only if*

$$W_j P_x \mathcal{R} = \{0\} \qquad \text{for every } j \in [m]. \tag{5}$$

If the condition fails, a history that switches among single actuators separates two compatible systems.

This is the exact per-actuator coverage requirement of the linear class. Since $\operatorname{span}(Y_j) \subseteq K_j$, observing each actuator along every reachable position direction, $P_x \mathcal{R} \subseteq \operatorname{span}(Y_j)$, always suffices, and the bounds can make fewer observations enough; identifying every effect matrix is not required. Theorem 1 shows that obtaining such observations can require histories that engage every actuator. Appendix B proves both directions, and Appendix C tests noise and misspecification.

5 STATE-CARRYING COLLECTION AT A MATCHED BUDGET

5.1 PROTOCOL

Balanced pairing requires restoring simulator states, which a physical collector cannot do. This section evaluates state-carrying collection, which obtains cross-actuator pairings within a history instead: one actuator prepares a state, and the next is probed from it (Figure 3).

Each history starts from an environment reset and lasts at most H transitions. Every h steps, CARRIED redraws the active actuator uniformly, with replacement, and continues from the current state. The amplitude is redrawn uniformly in $[-1, 1]$ every 25 steps and with probability 0.2 at other steps. Every transition is retained and counts towards the budget N (Algorithm 1). The default uses $H = 100$ and $h = 25$, so a history engages at most four actuators, a proper subset in three of the four environments.

We compare CARRIED with RESET, the reset-only collection of Section 2, which shares its amplitude rule but holds one actuator throughout each history, so the comparison isolates switching within a

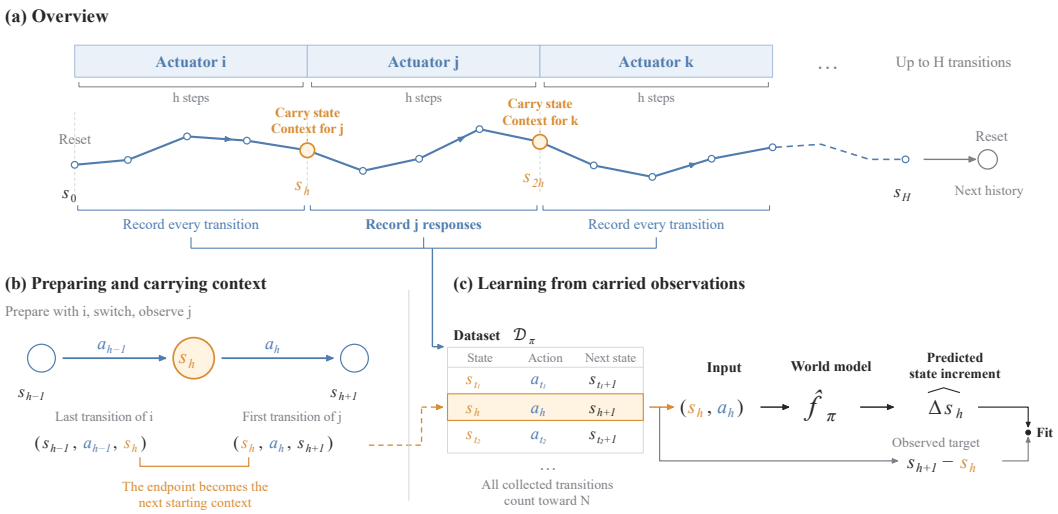

Figure 3: State-carrying collection. (a) One history of at most H transitions, in which each actuator acts for h steps and the physical state is kept across every switch. (b) At a switch, the same state s_h ends the last transition of actuator i and begins the first transition of actuator j. (c) Every recorded transition enters the training set and counts towards the transition budget. The panels show one realisation of the schedule.

Table 2: One-step error of the six feasible collectors at 48 000 transitions. Mean all-on target error over five collection draws, in each environment's physical units; bold marks the lowest in each row. CORR., CYCLE, TR.-BAL., and STEP abbreviate CORRELATED, FIXED CYCLE, TRANSITION-BALANCED, and SWITCH EVERY STEP. The last column is the paired reduction of CARRIED relative to RESET with a 95% interval.

		Hold one actuator		Switch within a history				CARRIED vs.
Environment	Learner	RESET	CORR.	CYCLE	CARRIED	TR.-BAL.	STEP	RESET (%)
Walker2d	MLP	5.20	6.71	4.55	4.15	4.14	**3.26**	20.2 ± 1.9
	Latent	5.71	7.56	5.04	4.57	4.55	**4.28**	19.9 ± 2.6
HalfCheetah	MLP	3.76	5.07	3.32	3.16	**3.14**	4.01	16.0 ± 2.0
	Latent	5.68	7.37	5.05	4.60	**4.54**	6.14	18.9 ± 2.8
Hopper	MLP	0.270	0.360	0.184	0.176	0.167	**0.153**	34.6 ± 6.4
	Latent	0.342	0.472	0.222	0.225	0.216	**0.179**	33.8 ± 9.1
Ant	MLP	8.37	9.94	7.25	6.82	7.44	**4.64**	18.6 ± 2.0
	Latent	9.91	12.01	9.30	9.28	9.18	**6.28**	6.3 ± 2.5

history. To test whether the gain depends on the particular schedule, we also compare four other feasible collectors, which hold one actuator with autocorrelated amplitudes (CORRELATED), cycle through actuators (FIXED CYCLE), balance consecutive actuator pairs (TRANSITION-BALANCED), or switch at every step (SWITCH EVERY STEP); Appendix D.3 gives every rule, and Appendix F.1 reports conditions that need initial-state access or target behaviour. The main comparisons use 48 000 transitions and the setup of Section 3. Unlike the matched pool, these protocols also change the pooled states, history lengths, and reset counts, so they measure an executable protocol's overall effect.

5.2 SWITCHING WITHIN A HISTORY

Table 2 reports the one-step error of the six feasible collectors. CARRIED reduces the error by 6.3–34.6% relative to RESET in all eight environment–learner comparisons. The decomposition of Section 3.2 shows that this gain differs from the matched-pool gain in where it falls: response error decreases by 4.7–26.7% and passive-increment error by 6.3–38.6%, with the passive improvement larger in five of eight comparisons (Appendix E).

Among the other feasible collectors, FIXED CYCLE and TRANSITION-BALANCED, like CARRIED, have lower mean error than both one-actuator collectors in every row of Table 2, and SWITCH EVERY STEP does as well except for both learners on `HalfCheetah`. TRANSITION-BALANCED matches

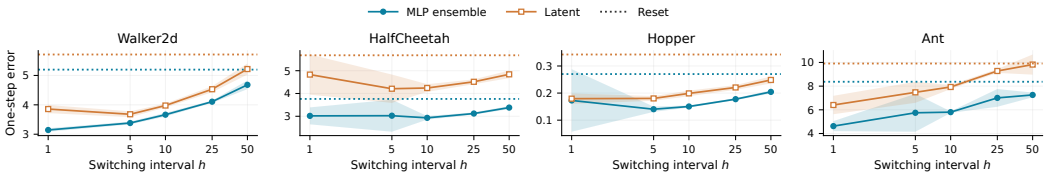

Figure 4: One-step error of CARRIED against the switching interval h. The periodic amplitude refresh is fixed at 25 steps. Dotted lines mark RESET in each learner's colour. Bands are 95% intervals across collection draws.

CARRIED, with paired intervals including zero in seven of eight comparisons, and SWITCH EVERY STEP is better than CARRIED in five with intervals excluding zero. The benefit therefore comes from switching within a history, whichever schedule is used (Appendix F.1). We keep CARRIED as the main protocol because it differs from RESET only in switching, whereas SWITCH EVERY STEP also changes the amplitude rule.

5.3 SWITCHING FREQUENCY, BUDGET, AND HISTORY LENGTH

We next examine how the gain depends on the switching rate, the budget, and the history length. A shorter switching interval can also change how often the amplitude is redrawn (Appendix D.3), so we sweep the interval over $h \in \{1, 5, 10, 25, 50\}$ with the amplitude refresh fixed at 25 steps. Figure 4 shows the one-step error of CARRIED in this sweep. Reducing h from 50 to 5 reduces the error by 10.6–31.3% in all eight comparisons, and the gain saturates there: the further reduction to $h = 1$ leaves paired intervals including zero in six (Appendix F.1 also compares amplitude rules at $h = 1$).

On `Ant`, raising the budget from 48 000 to 192 000 transitions widens the CARRIED advantage from 18.6% to 25.6% for the MLP ensemble and from 6.3% to 20.4% for the latent model, so more reset-only data does not close the gap; a longer history cap does not reproduce the gain (Appendix F.2).

5.4 UNSEEN SUBSETS AND LONGER HORIZONS

The comparisons so far score the all-on target at one step. To test other compositions, we evaluate all pairs and triples of actuators in the two six-actuator tasks, all pairs and 20 triples in `Ant`, and the three pairs in `Hopper`, 121 targets in total, each in its own context. For the MLP ensemble, CARRIED has lower mean error than RESET on 120 targets at one step and 107 at ten steps, with median reductions of 13% and 10%; the latent counts are 98 at both horizons, with medians 11% and 9% (Appendix F.6).

To test longer horizons, we roll each model out open-loop for ten steps under random action sequences restricted to the target. At ten steps CARRIED reduces error by 1.3–40.5% relative to RESET, with paired intervals excluding zero in six of eight comparisons (Figure 6 in Appendix F). On `Ant`, the mean reductions of both learners remain positive while accumulated error comes to dominate the one-step difference.

5.5 CONTROLLER TRAJECTORIES

A model used for control is queried on a controller's trajectories instead of random all-on behaviour, so we also evaluate on trajectories recorded by a fixed CEM controller acting in the true simulator. The advantage on random all-on behaviour holds across five resampled evaluation sets (Appendix F.3). On controller trajectories the paired intervals are wide, and the three that exclude zero, one of eight at one step and two of eight at ten, all favour CARRIED (Appendix F.7).

6 DISCUSSION AND LIMITATIONS

The collection gain combines the pairing with changes in the visited states, as its passive-increment improvement shows (Section 5.2); the matched pool isolates the pairing on a fixed state pool. The certificate is exact for the stated linear class. In general nonlinear dynamics, single-actuator observations leave arbitrary interaction terms unidentified, since the family $F_\lambda(s, a) = F_0(s, a) +$

$\lambda a_i a_j v(s)$ is indistinguishable under all single-actuator actions for $i \neq j$, so context coverage and the structure used to extrapolate joint actions are distinct requirements.

The experiments use fully observed simulation, two learners, and five draws per condition. Partial observation, visual inputs, hardware, and control utility are left to future work, and separating the effect of actuator count requires environments that differ only in that count.

7 Related Work

Coverage and distribution shift. Offline reinforcement learning studies learning from fixed data under state–action distribution shift (Levine et al., 2020). Model-based methods limit the resulting model error with short rollouts (Janner et al., 2019) and uncertainty penalties (Yu et al., 2020), and theoretical analyses express data sufficiency through concentrability, the density ratio between target and data distributions over states and actions (Chen & Jiang, 2019; Rashidinejad et al., 2021). For factored action spaces, a linear decomposition of the value function over sub-actions lets offline estimates generalise to rarely observed combinations (Tang et al., 2022). Empirical studies of robot demonstrations examine how dataset quality, composition, and scale affect learned policies (Mandlekar et al., 2022; Lin et al., 2025). These analyses assess a dataset through its pooled state–action distribution; our analysis concerns the per-actuator components of the state distribution, which the matched pool varies while the pooled distribution stays fixed.

Compositional learning and world models. Analyses of compositional generalisation state when models trained on some combinations of components generalise to new ones, in terms of the training support (Wiedemer et al., 2023), object-centric representations (Wiedemer et al., 2024), kernel models of compositionally structured representations (Lippl & Stachenfeld, 2025), the statistics of the training data (Okawa et al., 2023), and modular architectures (Schug et al., 2024). In a controlled system the training support is itself produced by collection, since the states at which a component can be observed depend on the actions that precede it. For dynamical systems, Neary & Topcu (2023) compose port-Hamiltonian subsystem models through their interconnection, Sehgal et al. (2024) bind object encodings to learned interaction rules, and Zhang et al. (2025) factorise multi-agent joint actions into per-agent components of a video world model. General-purpose world models learn from data gathered by their own behaviour (Hafner et al., 2020; 2025; Hansen et al., 2024; Alonso et al., 2024). We use full-state learners to keep the data question apart from representation learning and planning.

System identification and experimental design. Identifiability in interconnected systems depends on which modules are excited and measured and on switching (Weerts et al., 2018; Dreef et al., 2021), and action excitation matters for controlled world models (Zhang et al., 2026). Experimental design selects measurements for an inferential objective (Chaloner & Verdinelli, 1995; Bombois et al., 2018; Rainforth et al., 2024), while model disagreement and novelty guide exploration (Pathak et al., 2019; Shyam et al., 2019; Sekar et al., 2020). Learning to reset reduces manual resets (Eysenbach et al., 2018); our comparison instead keeps the history-length cap fixed and changes only which actuator acts within each history.

8 Conclusion

We studied what single-actuator data must contain for a world model to predict unseen actuator combinations. Pooled state coverage cannot answer this question because it ignores which actuator is observed at which state, and under reset-only collection each actuator is observed only at states it prepared itself. We isolated this actuator–state pairing with a matched-pool intervention, characterised in a linear switched class exactly when component observations determine composition, and proposed state-carrying collection, which lets one actuator prepare the states at which another is observed without restoring simulator states. Data for compositional world models should therefore be assessed by which actuator is observed at which state, and a physical collector can improve this pairing directly.

AI USE STATEMENT

We used large language models (LLMs) as general-purpose writing aids to polish wording and improve grammar, and to prepare the anonymised repository.

REPRODUCIBILITY STATEMENT

Appendices A and B provide the theoretical proofs. Appendices D–F specify data construction, training, evaluation, and statistical aggregation. Main comparisons use five independent collection draws with two network initialisations nested within each draw. The accompanying implementation, experiment manifests, per-run records, and figure-generation scripts document the reported results. Tests cover state restoration, coordinate conversion, initialisation, history-level data separation, and matched-pool invariants. Experiment specifications were recorded locally before their corresponding runs and are included with the materials. Exploratory results obtained with an earlier implementation are identified separately in Appendix G.

ETHICS STATEMENT

The experiments use simulated mechanical and locomotion systems and involve no human subjects or personal datasets. Restricting simultaneous actuation is a data-collection constraint in this study; safe exploration and deployment on physical robots require their own analysis. The reported improvements concern prediction.

REFERENCES

Eloi Alonso, Adam Jelley, Vincent Micheli, Anssi Kanervisto, Amos J. Storkey, Tim Pearce, and François Fleuret. Diffusion for world modeling: Visual details matter in atari. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6bdde0373d53d4a501249547084bed43-Abstract-Conference.html.

Xavier Bombois, Anton Korniienko, Håkan Hjalmarsson, and Gérard Scorletti. Optimal identification experiment design for the interconnection of locally controlled systems. *Automatica*, 89:169–179, 2018. doi: 10.1016/j.automatica.2017.12.014.

Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, 10(3):273–304, 1995. doi: 10.1214/ss/1177009939.

Jinglin Chen and Nan Jiang. Information-theoretic considerations in batch reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1042–1051, 2019.

Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep reinforcement learning in a handful of trials using probabilistic dynamics models. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://proceedings.nips.cc/paper_files/paper/2018/hash/3de568f8597b94bda53149c7d7f5958c-Abstract.html.

H. J. Dreef, M. C. F. Donkers, and Paul M. J. Van den Hof. Identifiability of linear dynamic networks through switching modules. *IFAC-PapersOnLine*, 54(7):37–42, 2021. doi: 10.1016/j.ifacol.2021.08.331.

Benjamin Eysenbach, Shixiang Gu, Julian Ibarz, and Sergey Levine. Leave no trace: Learning to reset for safe and autonomous reinforcement learning. In *International Conference on Learning Representations*, 2018.

Roy Featherstone. *Rigid Body Dynamics Algorithms*. Springer, Boston, MA, 2008. doi: 10.1007/978-1-4899-7560-7.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy P. Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640:647–653, 2025. doi: 10.1038/s41586-025-08744-2. URL https://doi.org/10.1038/s41586-025-08744-2.

Nicklas Hansen, Hao Su, and Xiaolong Wang. TD-MPC2: Scalable, robust world models for continuous control. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Oxh5CstDJU.

Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model: Model-based policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://papers.nips.cc/paper/2019/hash/5faf461eff3099671ad63c6f3f094f7f-Abstract.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6980.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/hash/9ef2ed4b7fd2c810847ffa5fa85bce38-Abstract.html.

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

Fanqi Lin, Yingdong Hu, Pingyue Sheng, Chuan Wen, Jiacheng You, and Yang Gao. Data scaling laws in imitation learning for robotic manipulation. In *International Conference on Learning Representations*, 2025.

Samuel Lippl and Kim Stachenfeld. When does compositional structure yield compositional generalization? a kernel theory. In *International Conference on Learning Representations*, 2025.

Ajay Mandlekar, Danfei Xu, Josiah Wong, Soroush Nasiriany, Chen Wang, Rohun Kulkarni, Li Fei-Fei, Silvio Savarese, Yuke Zhu, and Roberto Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Proceedings of the 5th Conference on Robot Learning*, volume 164 of *Proceedings of Machine Learning Research*, pp. 1678–1690, 2022.

Cyrus Neary and Ufuk Topcu. Compositional learning of dynamical system models using port-Hamiltonian neural networks. In *Proceedings of the 5th Annual Learning for Dynamics and Control Conference*, volume 211 of *Proceedings of Machine Learning Research*, pp. 679–691, 2023. URL https://proceedings.mlr.press/v211/neary23a.html.

Maya Okawa, Ekdeep Singh Lubana, Robert P. Dick, and Hidenori Tanaka. Compositional abilities emerge multiplicatively: Exploring diffusion models on a synthetic task. In *Advances in Neural Information Processing Systems*, volume 36, pp. 50173–50195, 2023. URL https://openreview.net/forum?id=frVo9MzRuU.

Deepak Pathak, Dhiraj Gandhi, and Abhinav Gupta. Self-supervised exploration via disagreement. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5062–5071, 2019. URL https://proceedings.mlr.press/v97/pathak19a.html.

Tom Rainforth, Adam Foster, Desi R. Ivanova, and Freddie Bickford Smith. Modern Bayesian experimental design. *Statistical Science*, 39(1):100–114, 2024. doi: 10.1214/23-STS915.

Paria Rashidinejad, Banghua Zhu, Cong Ma, Jiantao Jiao, and Stuart Russell. Bridging offline reinforcement learning and imitation learning: A tale of pessimism. In *Advances in Neural Information Processing Systems*, volume 34, pp. 11702–11716, 2021.

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

Simon Schug, Seijin Kobayashi, Yassir Akram, Maciej Wołczyk, Alexandra Proca, Johannes von Oswald, Razvan Pascanu, João Sacramento, and Angelika Steger. Discovering modular solutions that generalize compositionally. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=H98CVcX1eh.

Atharva Sehgal, Arya Grayeli, Jennifer J. Sun, and Swarat Chaudhuri. Neurosymbolic grounding for compositional world models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4KZpDGD4Nh.

Ramanan Sekar, Oleh Rybkin, Kostas Daniilidis, Pieter Abbeel, Danijar Hafner, and Deepak Pathak. Planning to explore via self-supervised world models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 8583–8592, 2020.

Pranav Shyam, Wojciech Jaśkowski, and Faustino Gomez. Model-based active exploration. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5779–5788, 2019.

Shengpu Tang, Maggie Makar, Michael W. Sjoding, Finale Doshi-Velez, and Jenna Wiens. Leveraging factored action spaces for efficient offline reinforcement learning in healthcare. In *Advances in Neural Information Processing Systems*, volume 35, pp. 34272–34286, 2022.

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

Mark Towers, Ariel Kwiatkowski, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, J. K. Terry, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://openreview.net/forum?id=qPMLvJxtPK.

Zhiqiang Wang, Jianhai Han, Xiangpan Li, Bingjing Guo, and Lewei Lu. Research on identification of minimum parameter set in robot dynamics and excitation strategy. *Sensors*, 25(18):5749, 2025. doi: 10.3390/s25185749.

Harm H. M. Weerts, Paul M. J. Van den Hof, and Arne G. Dankers. Identifiability of linear dynamic networks. *Automatica*, 89:247–258, 2018. doi: 10.1016/j.automatica.2017.12.013.

Thaddäus Wiedemer, Prasanna Mayilvahanan, Matthias Bethge, and Wieland Brendel. Compositional generalization from first principles. In *Advances in Neural Information Processing Systems*, volume 36, pp. 6941–6960, 2023. URL https://openreview.net/forum?id=LqOQ1uJmSx.

Thaddäus Wiedemer, Jack Brady, Alexander Panfilov, Attila Juhos, Matthias Bethge, and Wieland Brendel. Provable compositional generalization for object-centric learning. In *International Conference on Learning Representations*, 2024.

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*, volume 33, pp. 14129–14142, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/a322852ce0df73e204b7e67cbbef0d0a-Abstract.html.

Hongxin Zhang, Zeyuan Wang, Qiushi Lyu, Zheyuan Zhang, Sunli Chen, Tianmin Shu, Behzad Dariush, Kwonjoon Lee, Yilun Du, and Chuang Gan. COMBO: Compositional world models for embodied multi-agent cooperation. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YXRyYkb1im.

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

A Proof of Theorem 1

Construction. Fix $m \geq 2$ and set $n = m$, $M = C = A = I_m$, and $B = e_1$. For positive constants $a_2, \ldots, a_m, \gamma_1, \ldots, \gamma_{m-1}$, and $\theta > 0$, define

$$\begin{aligned} S_1 &= \gamma_1 (e_1 + e_2)(e_1 + e_2)^\top, \\ S_j &= a_j e_1 e_1^\top + \gamma_j (e_j + e_{j+1})(e_j + e_{j+1})^\top, \quad 2 \leq j \leq m-1, \\ S_m(\theta) &= a_m e_1 e_1^\top + \theta e_m e_m^\top. \end{aligned} \tag{6}$$

The two systems use $\theta_L \neq \theta_R$ and otherwise share all parameters. Choose common caps containing both sets of effects. Every effect is positive semidefinite with rank at most two, and every actuator is visible at the input: $S_1 e_1 = \gamma_1(e_1 + e_2)$ and $S_j e_1 = a_j e_1$ for $j \geq 2$. The dynamics are

$$\ddot{x} + \dot{x} + \left(I + \sum_j q_j(t) S_j \right) x = e_1 u(t), \qquad x(0) = \dot{x}(0) = 0.$$

Permitted proper-subset histories. Each history selects a proper subset $I \subsetneq [m]$ and uses only that subset from reset to end, including state preparation. Within I, it may use arbitrary piecewise-constant switching or simultaneous activation, piecewise-continuous scalar inputs, and arbitrary duration. The full position vector is observed, and later histories may be selected adaptively. States are not transferred between histories with different subsets.

Indistinguishability on every proper subset. If $m \notin I$, the equations do not depend on θ. If $m \in I$, at least one chain index $k \in [m-1]$ is absent. Only S_k connects coordinates k and $k+1$, so every available stiffness matrix preserves

$$V_k = \operatorname{span}(e_1, \ldots, e_k).$$

The initial state and input lie in this subspace, hence all trajectories remain in V_k. The restriction of $S_m(\theta)$ to V_k is $a_m e_1 e_1^\top$, independent of θ. Thus every permitted history has identical input–output records in both systems. Adaptive selection cannot distinguish them because identical past records produce identical choices of later histories.

Full reachability of a proper subset. Activate all actuators in $[m-1]$. The resulting stiffness matrix K is tridiagonal with nonzero adjacent entries γ_j. The entry of $K^r e_1$ in coordinate $r+1$ is $\prod_{j \leq r} \gamma_j$, and its later entries are zero, giving

$$\det[e_1, K e_1, \ldots, K^{m-1} e_1] = \prod_{j=1}^{m-1} \gamma_j^{m-j} > 0.$$

For the phase-space generator and input matrix

$$\mathsf{F} = \begin{pmatrix} 0 & I \\ -K & -I \end{pmatrix}, \qquad G = \begin{pmatrix} 0 \\ e_1 \end{pmatrix},$$

write $p(w) = (w, 0)$ and $v(w) = (0, w)$. The identities $\mathsf{F} v(w) = p(w) - v(w)$ and $\mathsf{F} p(w) = -v(Kw)$ imply that the invariant Krylov span of G contains $p(K^k e_1)$ and $v(K^k e_1)$ for every k. It therefore equals $\mathbb{R}^{2m}$. This full reachability does not reveal $S_m e_m$: actuator m is absent from the histories that explore that direction.

Different all-on responses. Let K_L and K_R be the all-on stiffness matrices and set $\delta\theta = \theta_R - \theta_L$. Then $K_R - K_L = \delta\theta e_m e_m^\top$. Bandedness gives

$$e_m^\top K_L^r e_1 = 0 \quad (r < m-1), \qquad e_m^\top K_L^{m-1} e_1 = \prod_{j<m} \gamma_j.$$

Using

$$K_R^r - K_L^r = \sum_{\ell=0}^{r-1} K_R^{r-1-\ell} (K_R - K_L) K_L^\ell,$$

we obtain $(K_R^r - K_L^r)e_1 = 0$ for $r < m$, whereas

$$(K_R^m - K_L^m)e_1 = \delta\theta \left(\prod_{j<m} \gamma_j \right) e_m \neq 0.$$

Under a common unit-step input, $\ddot{x}(0^+) = e_1$, $\dddot{x}(0^+) = -e_1$, and $x^{(k)}(0^+) = -x^{(k-1)}(0^+) - Kx^{(k-2)}(0^+)$ for $k \geq 4$. Consequently,

$$x^{(2m+2)}(0^+) = (-1)^m K^m e_1 + p_{m-1}(K)e_1,$$

where p_{m-1} has degree at most $m-1$ and coefficients independent of the system choice. The difference in this derivative is nonzero, so the analytic all-on responses differ on every positive time interval.

Sequential single-actuator identification. A history ordered $1 \to 2 \to \cdots \to m$ can distinguish the systems. Starting from a state with $x_j \neq 0$ and later coordinates zero, activating actuator j gives $\ddot{x}_{j+1} = -\gamma_j x_j \neq 0$, so a sufficiently short activation produces $x_{j+1} \neq 0$. After this preparation reaches $x_m \neq 0$, activating actuator m yields an acceleration difference $-\delta\theta x_m e_m$. This history activates only one actuator at a time, and its actuator set is $[m]$. Restricting each history to a proper subset therefore limits what the data can reveal, and a history that switches single actuators in chain order removes that limit.

B Compatible Effects and the Composition Certificate

B.1 The Completion Interval

Fix an actuator and omit its index. Let U be an orthonormal basis for $\mathrm{span}(Y)$ and complete it with Z to an orthogonal matrix $Q = [U, Z]$. In this basis,

$$Q^\top S Q = \begin{pmatrix} A_o & B_o^\top \\ B_o & D \end{pmatrix}.$$

Exact observations of SY determine $A_o = U^\top SU$ and $B_o = Z^\top SU$, leaving D unknown. The Schur-complement conditions are

$$\begin{aligned} S \succeq 0 \iff & A_o \succeq 0, \quad \mathrm{range}(B_o^\top) \subseteq \mathrm{range}(A_o), \\ & D \succeq B_o A_o^\dagger B_o^\top =: D_-, \\ S \preceq LI \iff & LI - A_o \succeq 0, \quad \mathrm{range}(B_o^\top) \subseteq \mathrm{range}(LI - A_o), \\ & D \preceq LI - B_o(LI - A_o)^\dagger B_o^\top =: D_+. \end{aligned}$$

The identity matrices in the observed and unobserved blocks have their corresponding dimensions. The compatible set is nonempty exactly when these conditions hold and $D_- \preceq D_+$. It is then the interval $\{S : S_- \preceq S \preceq S_+\}$, where

$$S_\pm = Q \begin{pmatrix} A_o & B_o^\top \\ B_o & D_\pm \end{pmatrix} Q^\top, \qquad W = S_+ - S_- = Z(D_+ - D_-)Z^\top.$$

For compatible effects S and $\widetilde{S}$, their difference satisfies $-W \preceq S - \widetilde{S} \preceq W$. All compatible effects agree on x if and only if $Wx = 0$; the maximum discrepancy is

$$\sup_{S,\widetilde{S}} \|(S - \widetilde{S})x\|_2 = \sqrt{\lambda_{\max}(W)}\sqrt{x^\top W x}.$$

The observed span lies in $\ker W$, but the bounds can determine further directions. For example, if $L = 1$, $Y = e_1$, and $Se_1 = (1/2, 1/2)^\top$, every compatible matrix has the form $\left(\begin{smallmatrix} 1/2 & 1/2 \\ 1/2 & d \end{smallmatrix}\right)$. The lower and upper semidefinite constraints require $d \geq 1/2$ and $d \leq 1/2$, respectively. Thus the entire matrix is determined although the observed span is one-dimensional.

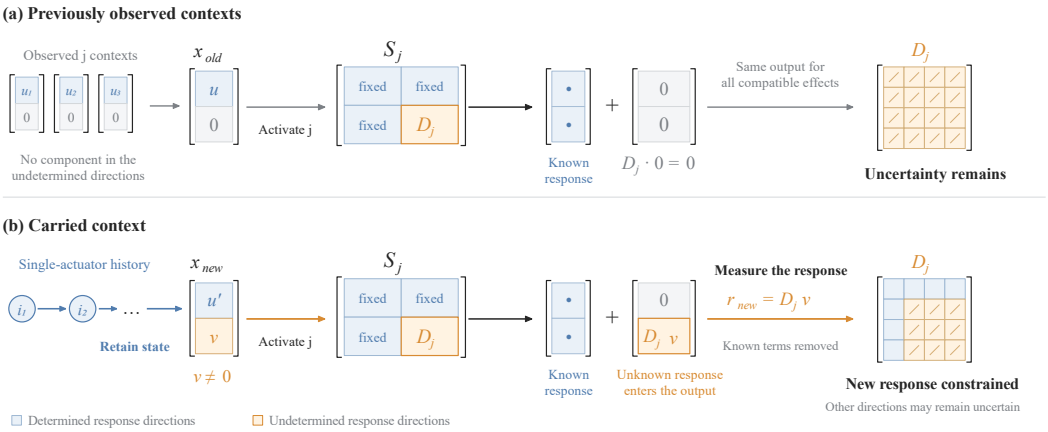

Figure 5: Previously observed and carried states for one actuator. (a) Previous observations have no component in the undetermined response directions, so the unknown block of the effect does not enter the output. (b) A carried state has a nonzero component in those directions, so the unknown block enters the measured response. Blue marks determined directions and orange marks undetermined ones.

B.2 Proof of Theorem 2

Write $z = (x, \dot{x})$, and define

$$\mathsf{F}_0 = \begin{pmatrix} 0 & I \\ -M^{-1}A & -M^{-1}C \end{pmatrix}, \quad G = \begin{pmatrix} 0 \\ M^{-1}B \end{pmatrix}, \quad E_j^- = \begin{pmatrix} 0 & 0 \\ -M^{-1}S_{j,-} & 0 \end{pmatrix}.$$

The nominal reachable subspace $\mathcal{R}$ is the smallest subspace containing $\mathrm{range}(G)$ and invariant under F_0 and $\mathsf{F}_0 + E_j^-$ for every j. Equivalently, it is the span of endpoints reachable from zero using nominal single-actuator switching histories and arbitrary inputs. This invariance also implies invariance under every E_j^- and hence under every simultaneous nominal mode.

Sufficiency. Suppose $W_j P_x \mathcal{R} = \{0\}$ for every j. Every compatible effect agrees with $S_{j,-}$ on $P_x\mathcal{R}$. Therefore each compatible mode generator agrees with the nominal generator on $\mathcal{R}$. Nominal trajectories remain in this subspace and solve the equations for every compatible system. Uniqueness of solutions yields identical trajectories for every input and schedule.

Necessity. Suppose $W_j P_x z \neq 0$ for some $z \in \mathcal{R}$ and actuator j. Because reachable endpoints span $\mathcal{R}$, a nominal single-actuator history reaches an endpoint z' with $W_j P_x z' \neq 0$. Compare the lower-endpoint system with the compatible system obtained by replacing only $S_{j,-}$ by $S_{j,+}$. If they differ during the reaching history, that history already distinguishes them. Otherwise both reach z', and appending activation of j gives an acceleration difference $-M^{-1}W_j P_x z' \neq 0$. This single-actuator history distinguishes the systems. The preparation prefix may itself include actuator j. Figure 5 shows why a carried state exposes the undetermined block of an effect.

Scope of the certificate. The result assumes independently bounded effects, externally selected switches, exact observations, and zero initial conditions. A specified nonzero initial state can be included among the generators of $\mathcal{R}$. Correlated constraints across effects require a joint compatible set. Partial observation or endogenous switching changes which measurements or separating actions are available and requires a corresponding observability or admissibility analysis.

C Numerical Study of the Linear Class

C.1 Matched Observations with Different Component Information

Set $M = C = A = B = I_3$, $L_j = 4$, and $S_j = I_3/4 + v_j v_j^\top$, where the v_j are cyclic permutations of $(1, 2, 3)^\top/4$. There are nine noiseless vector observations, three per actuator. The aligned design

Table 3: Aligned and balanced designs in the nine-observation example. Pooled rank, per-actuator observation ranks, and ranks of each actuator's completion width. Disagreement is the maximum phase-state distance between the lower- and upper-endpoint systems under the same held-out all-on input.

Assignment	Pooled rank	Component ranks	Width ranks	Disagreement
Aligned	3	$(1, 1, 1)$	$(2, 2, 2)$	0.85725
Balanced	3	$(3, 3, 3)$	$(0, 0, 0)$	0

observes actuator j at e_j three times; the balanced design observes every actuator at e_1, e_2, e_3. Both designs have the same pooled position multiset, pooled rank, and sample count.

Exact rational arithmetic gives the completion endpoints and ranks in Table 3. Both nominal reachable phase spaces have dimension six. The aligned design retains uncertainty in two directions per actuator, whereas the balanced design identifies every effect. For a common all-on input $(1, 1, 1)^\top$, a matrix-exponential integrator compares trajectories over four time units with 200 intervals. The maximum endpoint-system discrepancy is 0.85725 under aligned observations and zero under balanced observations.

The pooled observation singular values are $(\sqrt{3}, \sqrt{3}, \sqrt{3})$ in both designs. The per-actuator singular values are $(\sqrt{3}, 0, 0)$ under alignment and $(1, 1, 1)$ under balancing. The determined subspace is $\mathrm{span}(e_j)$ for aligned actuator j and $\mathbb{R}^3$ for every balanced actuator. Identical pooled spectra therefore coexist with different component information.

C.2 Actuator Ordering and Finite-Time Distinguishability

We evaluate the chain construction at dimensions $3, 4, 6, 8, 12, 16$, comparing selected proper-subset schedules with ascending and descending single-actuator orders. The selected proper-subset and descending-order schedules yield zero trajectory difference. Ascending order separates the systems; with switching interval 0.5, its maximum finite-time signal falls from approximately 3.19×10^{-4} at dimension three to 1.75×10^{-14} at dimension sixteen. The separating signal of ascending order therefore falls below realistic measurement noise as the dimension grows, so at high dimension exact distinguishability does not yield a measurable signal.

C.3 Measurement Noise and Misspecification

A separate response-prediction study uses dimensions $3, 6, 8$, five seeds, noise standard deviations 0 and 0.01, and

$$r_j(x) = S_j x + \beta \tanh(S_j x), \qquad \beta \in \{0, 0.1, 0.5\}.$$

Each actuator receives 600 observations. Observed positions lie on coordinate axes with random amplitudes, and their pooled values are fixed across aligned and balanced designs. The learner is symmetric linear least squares without a positive-semidefinite constraint and is correctly specified only at $\beta = 0$. Evaluation uses independent dense Gaussian positions and random compositions, measuring the mean Euclidean error of the summed response.

Across the 180 reported design–seed combinations, every noiseless linear balanced error is below 10^{-9}. Aligned data can fit their observed responses accurately while leaving target responses ambiguous. Noise and nonlinear perturbations introduce estimation or approximation error even when the component observation matrices span the space. For the balanced designs, component observation condition numbers range from 1.004 to 1.188; aligned designs are rank-deficient. The records include raw and sample-size-normalised spectra.

Implementation checks comprise nine completion regression tests and 70 exact-rational checks of the boundary construction on 10 fixed cases.

Table 4: Collection rules and access. The first six rows are feasible collectors. The diverse-reset row comprises two initialisation references; the last two rows require target behaviour. This gives ten conditions. Actuator timing and amplitude timing are listed as separate schedules.

Condition	Actuator schedule or additional access	Amplitude or reference design
RESET	One actuator per history	25-step refresh; probability 0.2 otherwise
CARRIED	Uniform redraw every 25 steps	Same as RESET
FIXED CYCLE	Next actuator every 25 steps	Same as RESET
TRANSITION-BALANCED	Balance ordered switches	Same as RESET
SWITCH EVERY STEP	Uniform redraw each step	Uniform redraw each step
CORRELATED	One actuator per history	AR(1), coefficient 0.95
DIVERSE-RESET	Additional reset-state perturbations	Scales 0.15 and 0.35
IN-CONTEXT	Restore target-behaviour states	Single-actuator probes
ALL-ON	Execute target behaviour	Joint-action observations

D MUJOCO EXPERIMENTAL DETAILS

D.1 ENVIRONMENTS, MODELS, AND TRAINING

We use Gymnasium MuJoCo v5 (Towers et al., 2025; Todorov et al., 2012). All four environments retain their default frame skips. States are the full simulator position and velocity, which is 29-dimensional for `Ant`. Restoration resets simulator data before setting position and velocity, preventing a previous probe's numerical warm start from carrying over.

The MLP ensemble contains five networks, each with two hidden layers of width 256 and SiLU activations; predictions are averaged at evaluation. The deterministic latent model uses a 64-dimensional encoding, a residual latent transition, and a decoder. Its loss combines reconstruction, increment prediction, and latent consistency with the next encoded state. State and increment scalings are converted explicitly when the losses are combined. Models receive standardised states and raw actions and predict standardised increments. Physical-coordinate increments are used for the main evaluation metric.

Training uses Adam (Kingma & Ba, 2015) with learning rate 10^{-3}, batch size 512, and 60 epochs, selecting the epoch by the condition's validation error. Histories are split $80/10/10$ into training, validation, and held-out partitions; no history spans partitions. Outside the matched-pool intervention, normalisation uses each condition's training split. Matched-pool conditions share a fixed scaling as described below. Initialisation seeds are set before constructing networks, and paired conditions use matching initial weights. Computation uses CPUs.

D.2 REPLICATION AND AGGREGATION

Every main design has five independent collection draws and two network initialisations per draw. For draw b, let $\bar{E}_{P,b}$ denote the error for condition P averaged over its two initialisations. The reduction of condition Q relative to reference P is

$$g_{P\to Q,b} = 100\left(1 - \frac{\bar{E}_{Q,b}}{\bar{E}_{P,b}}\right),$$

so positive values favour Q. Conditions in a comparison share their evaluation data (Appendix D.5). We aggregate these paired reductions across the five draws and report a Student-t interval with four degrees of freedom. Intervals are descriptive and unadjusted for multiple comparisons; the collection draw is the unit of replication, and environment–learner combinations, target subsets, and evaluation draws are reported within it.

D.3 COLLECTION PROTOCOLS AND ACCESS REQUIREMENTS

The default physical collection budget is $N = 48\,000$ transitions with $H = 100$. Histories end on the cap or environment termination. Every executed transition is counted and recorded, including preparation transitions. Table 4 distinguishes actuator schedules from amplitude rules and additional capabilities.

RESET, default CARRIED, FIXED CYCLE, and TRANSITION-BALANCED share one amplitude rule: the amplitude is refreshed uniformly in $[-1, 1]$ every 25 steps and with probability 0.2 at intervening steps, so comparisons among these four collectors isolate the actuator schedule. RESET holds a uniformly selected actuator throughout a history; CARRIED redraws it every 25 steps with replacement. FIXED CYCLE advances to the next actuator at each switch. TRANSITION-BALANCED selects among actuators other than the current one using the fewest observed ordered transitions from that actuator and the fewest transitions overall. SWITCH EVERY STEP redraws both the actuator and amplitude at every transition. CORRELATED uses one actuator per history with an AR(1) amplitude of coefficient 0.95 mapped through the normal cumulative distribution function.

Algorithm 1 Carried single-actuator collection

Require: Transition budget N, history cap H, switching interval h, actuator rule σ, amplitude rule η

$\mathcal{D} \leftarrow \varnothing$
while $|\mathcal{D}| < N$ **do**
  Reset the environment and observe s
  for $t = 0, \ldots, H-1$ **do**
    if $t \bmod h = 0$ **then**
      Select the active actuator j using σ
    end if
    Update amplitude α using η; set $a \leftarrow \alpha e_j$
    Execute a and observe s' and termination
    Append (s, a, s') to $\mathcal{D}$; carry $s \leftarrow s'$
    if the environment terminates or $|\mathcal{D}| = N$ **then**
      Break the current history
    end if
  end for
end while
Fit the prescribed learner to $\mathcal{D}$

The two DIVERSE-RESET references perturb the reset pose with Gaussian scales 0.15 and 0.35 in the tangent space and perturb velocity at four times the corresponding scale. They clip to joint limits and reject unhealthy or interpenetrating states. The additional ability to set initial states distinguishes them from feasible collectors. ALL-ON uses actions uniformly drawn from $[-1, 1]^m$. IN-CONTEXT restores each state from an all-on training trajectory and probes it once with a single-actuator action. These references share training states, splits, and input scaling and require target behaviour.

The protocol records reset counts, termination counts, realised history lengths, per-actuator transition counts, and ordered switch counts. Equal history caps do not force equal realised history lengths or reset counts.

Independent per-step actuator selection. If J_t is uniform and independent of the pre-action state s_t, then for any region U,

$$\Pr(s_t \in U, J_t = j) = \frac{1}{m} \Pr(s_t \in U).$$

Thus, under this sampling rule, each actuator has the same conditional pre-action state distribution in expectation. A finite dataset realises unequal empirical distributions, whose coverage of the target distribution is measured separately. Holding an actuator across several steps removes the independence, because intervening states evolve under that actuator.

Schedule and budget studies. The coupled sweep uses $h \in \{5, 10, 25, 50\}$ and refreshes amplitude at each actuator switch. The fixed-refresh sweep varies the switching interval over the same values while keeping periodic amplitude refresh at 25 steps; an additional run supplies $h = 1$. On `Ant`, separate studies vary budgets over $48\,000$, $96\,000$, and $192\,000$, or history caps over 100, 200, and 400 steps at the default budget. Each longer-history comparison includes a reset control with the same cap.

D.4 MATCHED-POOL CONSTRUCTION

Preparation histories are partitioned before constructing the pool, with equal source-actuator counts in each partition. Blocks contain one state from each source actuator and share a sampled amplitude. Shifts of zero, a uniform offset in $\{0, \ldots, m-1\}$, or a uniform offset in $\{1, \ldots, m-1\}$ implement the three assignments. Their bijective structure preserves each actuator's amplitude multiset within every partition. The conditions share input scaling and a fixed increment scaling computed from the state pool, not from condition-specific response labels. State restoration obtains the true successor for every reassigned probe.

The implementation checks equality of state arrays, trajectory groups, partitions, per-actuator amplitude multisets, and normalisation across all 20 environment–draw pools. It also checks that every training action has single-actuator support. The construction controls assignment within a fixed pool, using restoration queries and state-preparation access beyond the interaction budget available to feasible physical collectors.

D.5 Evaluation Distributions and Rollouts

For each target subset T, evaluation states are obtained from reset under random actions restricted to T. Each target uses $2\,000$ states, except the `Hopper` all-on target, which uses $10\,000$ because its error is sensitive to the evaluation draw. The same evaluation states, actions, and true simulator transitions are used for every condition in a comparison. The primary error is the mean Euclidean error of the de-standardised increment. A common-scale variant divides errors by the standard deviation of the all-on evaluation states before taking the norm; Appendix E.3 reports both headline comparisons under it.

Each target also uses 200 ten-step open-loop rollouts from evaluation states with prescribed random action sequences restricted to T. Models recursively update their predicted state from their predicted increment and receive the same action sequence as the simulator. Non-finite outcomes are counted rather than removed. Five independently generated all-on evaluation sets assess evaluation-set variation separately from collection variation.

For the controller-distribution evaluation, one fixed CEM controller per environment acts in the true simulator (Rubinstein, 1999). All models predict its held-out recorded trajectories at one and ten steps. The learned models do not choose the controller's actions, so this is a prediction experiment under a different state–action distribution.

D.6 Implementation and Result Checks

Initial weight hashes check paired initialisations. State-restoration tests check deterministic queries and coordinate conversion. Completeness checks compare model and prediction records against experiment manifests, including both learners and both initialisations, configuration values, source hashes, and checkpoint availability. Evaluation and controller predictions refer to matching model records and identical evaluation arrays. Specifications were recorded locally before the corresponding runs.

E Response-Level Diagnostics

E.1 Simulator and Model Queries

At each all-on evaluation state s, with evaluation action a fixed, restore the simulator and execute 0, each $a_j e_j$, and a separately. These queries produce the passive increment, individual responses, assembled prediction, and joint increment defined in (2)–(3). The probes are diagnostic measurements; the training datasets contain only collected transitions. Every frozen model is evaluated at the same state–action pairs without retraining.

We report passive-increment error, mean per-actuator response error, direct joint prediction error, and the error of the model's assembled prediction relative to the true joint increment. The passive and response quantities share the same model and evaluation states. Their relative improvements describe different prediction targets and therefore do not sum to the improvement in joint prediction error.

E.2 Action-Blind Control

An auxiliary MLP ensemble is trained and evaluated with its action columns set to zero. The architecture, data partitions, normalisation, optimiser, epoch budget, and initialisation seeds otherwise match the corresponding action-aware comparison. Table 6 reports the observed improvements under both input choices. The action-blind model controls for the use of action labels. Removing the actions also changes the learning problem, so we compare the two improvements side by side.

Table 5: Passive and response prediction under changes in training data. Paired percentage reductions in the two diagnostic errors at common evaluation states. Positive values favour the second condition. The first two groups are matched-pool interventions. Each percentage uses its own diagnostic error as denominator. ALIGNED, BALANCED, OTHER, and STEP abbreviate SOURCE-ALIGNED, BALANCED PAIRING, OTHER-SOURCE, and SWITCH EVERY STEP.

		MLP		Latent	
Contrast	Environment	Passive	Response	Passive	Response
ALIGNED → BALANCED	Walker2d	7.8 ± 3.9	20.8 ± 1.6	13.5 ± 4.1	38.0 ± 2.6
	HalfCheetah	8.9 ± 1.2	23.8 ± 1.2	15.4 ± 3.3	34.4 ± 1.2
	Hopper	8.7 ± 1.3	17.6 ± 2.2	10.1 ± 5.6	30.3 ± 3.4
	Ant	6.4 ± 2.1	18.7 ± 0.8	-9.7 ± 4.8	22.9 ± 1.4
ALIGNED → OTHER	Walker2d	2.8 ± 3.0	-3.6 ± 2.7	8.1 ± 6.7	19.8 ± 1.7
	HalfCheetah	8.1 ± 2.9	21.4 ± 1.6	16.2 ± 2.6	33.8 ± 1.1
	Hopper	4.1 ± 4.8	-28.6 ± 9.7	7.3 ± 4.7	-23.8 ± 5.2
	Ant	2.5 ± 0.8	8.6 ± 1.4	-11.9 ± 2.6	16.3 ± 2.5
RESET → CARRIED	Walker2d	21.4 ± 3.7	17.4 ± 1.3	17.5 ± 2.4	21.6 ± 2.8
	HalfCheetah	14.1 ± 2.8	16.3 ± 2.6	16.6 ± 3.9	19.9 ± 2.1
	Hopper	38.5 ± 4.0	26.7 ± 6.4	38.6 ± 8.7	26.0 ± 10.9
	Ant	21.1 ± 3.1	12.7 ± 2.5	6.3 ± 2.9	4.7 ± 3.2
RESET → STEP	Walker2d	36.3 ± 2.6	35.4 ± 1.0	19.8 ± 2.1	26.9 ± 3.6
	HalfCheetah	-18.5 ± 15.8	29.5 ± 1.1	-21.1 ± 13.6	27.6 ± 1.5
	Hopper	44.0 ± 21.9	41.3 ± 29.1	48.8 ± 7.3	44.2 ± 8.3
	Ant	39.0 ± 2.5	50.4 ± 0.5	30.6 ± 4.5	41.7 ± 2.3
RESET → IN-CONTEXT	Walker2d	55.9 ± 2.0	45.0 ± 1.3	45.1 ± 0.9	41.6 ± 1.4
	HalfCheetah	52.6 ± 2.2	35.6 ± 1.8	56.7 ± 2.2	44.1 ± 2.1
	Hopper	60.8 ± 3.1	54.4 ± 3.7	58.4 ± 7.6	49.6 ± 6.6
	Ant	65.1 ± 2.6	57.1 ± 1.4	54.1 ± 4.5	33.6 ± 6.4

Table 6: Action-blind and action-aware MLP controls. Paired percentage reductions in joint prediction error across five collection draws. A dash denotes an action-blind condition that was not run. Abbreviations follow Table 5.

Contrast	Environment	Action-blind (%)	Action-aware (%)
ALIGNED → BALANCED	Walker2d	0.5 ± 0.4	16.1 ± 1.4
	HalfCheetah	-0.3 ± 0.6	15.3 ± 1.4
	Hopper	0.5 ± 0.2	13.5 ± 1.1
	Ant	1.1 ± 0.9	9.8 ± 1.5
ALIGNED → OTHER	Walker2d	0.8 ± 0.4	-9.3 ± 3.3
	HalfCheetah	-0.3 ± 0.7	11.9 ± 2.5
	Hopper	0.5 ± 0.3	-18.1 ± 7.6
	Ant	0.3 ± 1.3	1.5 ± 1.0
RESET → CARRIED	Walker2d	0.1 ± 0.6	20.2 ± 1.9
	HalfCheetah	0.2 ± 1.9	16.0 ± 2.0
	Hopper	1.0 ± 0.4	34.6 ± 6.4
	Ant	5.2 ± 1.2	18.6 ± 2.0
RESET → STEP	Walker2d	—	37.2 ± 1.1
	HalfCheetah	—	-6.6 ± 12.7
	Hopper	—	43.3 ± 26.1
	Ant	—	44.5 ± 1.5
RESET → IN-CONTEXT	Walker2d	8.6 ± 0.5	49.6 ± 1.5
	HalfCheetah	16.2 ± 0.6	46.3 ± 1.9
	Hopper	7.1 ± 0.3	56.6 ± 3.8
	Ant	18.4 ± 1.5	65.9 ± 2.1

For the matched pool, the action-blind error reduction from balanced pairing ranges from -0.3 to 1.1%, compared with 9.8–16.1% for the corresponding MLP ensemble with actions. The gain therefore depends on actuator-labelled observations. Under physical collection, both errors fall: from RESET to CARRIED, the passive improvement is the larger one in five of eight comparisons, including both learners on `Hopper` and `Ant`. The collection effect therefore combines both mechanisms, which the matched pool holds apart.

Table 7: Held-out error on each condition's own collected data, and its ratio to the composition error. Own error is the one-step error on the held-out partition of the condition's collected data, and the ratio divides the all-on target error by it. Means over five collection draws after averaging two initialisations. BALANCED abbreviates BALANCED PAIRING.

		SOURCE-ALIGNED		BALANCED		RESET		CARRIED	
Environment	Learner	Own	Ratio	Own	Ratio	Own	Ratio	Own	Ratio
Walker2d	MLP	1.06	4.9	1.21	3.6	1.15	4.5	1.12	3.7
	Latent	0.989	6.1	1.14	3.8	1.54	3.7	1.36	3.4
HalfCheetah	MLP	0.388	9.3	0.397	7.7	0.35	10.8	0.36	8.8
	Latent	0.394	12.8	0.399	9.4	0.447	12.7	0.454	10.1
Hopper	MLP	0.114	2.1	0.126	1.6	0.121	2.2	0.118	1.5
	Latent	0.104	2.5	0.116	1.8	0.178	1.9	0.172	1.3
Ant	MLP	0.744	9.1	0.85	7.2	0.709	11.8	0.777	8.8
	Latent	0.738	11.4	0.885	8.7	1.02	9.7	1.08	8.6

Table 8: Own-distribution and composition error from SOURCE-ALIGNED to BALANCED PAIRING. Paired percentage reductions in held-out error on each condition's own collected data, in all-on composition error, and in composition error on a common scale that divides by the standard deviation of the evaluation states. Positive values favour BALANCED PAIRING. Intervals are 95% intervals across five collection draws.

Environment	Learner	Own distribution	Composition	Common scale
Walker2d	MLP	-14.6 ± 2.9	16.1 ± 1.4	19.2 ± 1.5
	Latent	-15.3 ± 2.6	29.2 ± 2.2	27.5 ± 1.9
HalfCheetah	MLP	-2.3 ± 2.3	15.3 ± 1.4	13.6 ± 0.8
	Latent	-1.6 ± 5.3	25.4 ± 1.6	20.0 ± 1.1
Hopper	MLP	-11.1 ± 4.3	13.5 ± 1.1	14.1 ± 0.6
	Latent	-12.0 ± 6.0	19.6 ± 3.5	17.8 ± 1.7
Ant	MLP	-14.2 ± 1.8	9.8 ± 1.5	5.1 ± 2.3
	Latent	-19.9 ± 3.5	8.3 ± 2.2	3.4 ± 1.7

E.3 FIT ON THE COLLECTED DISTRIBUTION

Every run also scores the model on the held-out partition of its own collected data, which separates a badly fitted model from one that fits what it was given and fails on the composition. Table 7 reports that error and its ratio to the all-on target error. The ratio ranges from 1.3 to 12.8 across the four conditions shown: each condition predicts its own distribution more accurately than the combination it never observed, so none produces a model that is merely undertrained.

Tables 8 and 9 give the paired changes. From SOURCE-ALIGNED to BALANCED PAIRING, own-distribution error rises in all eight comparisons, by 1.6–19.9%, while the composition error falls in all eight. The reassignment therefore does not make the data easier to fit, which rules out the explanation that balanced datasets are simply better conditioned for regression. From RESET to CARRIED, the own-distribution error shows no consistent direction, with intervals including zero in six of eight comparisons, while the composition error again falls in all eight.

The common-scale column repeats each composition comparison under the variant metric of Appendix D, which divides by the standard deviation of the evaluation states before taking the norm. Both headline comparisons keep positive intervals excluding zero in all eight cells, and no reduction moves by more than 6.8 points between the two metrics. The largest change is the collection effect on `Ant`, which rises from 18.6 and 6.3% to 25.4 and 10.8%.

E.4 SIMULATOR ADDITIVITY AND MODEL ADDITIVITY

Table 10 reports the mean norms of the passive increment, actuated response, and residual of the assembled prediction. Its Share column is the mean residual norm divided by the mean actuated-response norm. Across environments this ratio is 1.7–13.6%; relative to the SOURCE-ALIGNED target error, the residual is 6.3–42.5%. These quantities characterise one-step approximation by summed true responses.

Table 9: Own-distribution and composition error from RESET to CARRIED. The same three quantities as Table 8; positive values favour CARRIED.

Environment	Learner	Own distribution	Composition	Common scale
Walker2d	MLP	2.8 ± 8.3	20.2 ± 1.9	20.5 ± 1.1
	Latent	11.2 ± 9.7	19.9 ± 2.6	20.8 ± 2.0
HalfCheetah	MLP	-3.6 ± 13.9	16.0 ± 2.0	14.1 ± 1.9
	Latent	-2.4 ± 15.6	18.9 ± 2.8	16.5 ± 4.2
Hopper	MLP	0.7 ± 19.6	34.6 ± 6.4	33.6 ± 4.9
	Latent	1.7 ± 20.0	33.8 ± 9.1	34.1 ± 10.9
Ant	MLP	-9.8 ± 6.6	18.6 ± 2.0	25.4 ± 4.4
	Latent	-6.7 ± 12.7	6.3 ± 2.5	10.8 ± 4.8

For each model, direct additivity deviation is

$$D_{\text{add}} = \mathbb{E}_{(s,a)\sim\mu_{[m]}} \|\hat{f}(s,a) - \hat{\Delta}_{\text{sum}}(s,a)\|_2.$$

For conditions trained on single-actuator data, this deviation is 2.9–17.6% of the mean actuated response and 10–58% of the model's own joint prediction error. For the all-on-trained reference it is 1.4–9.6% of the mean actuated response. This deviation measures the discrepancy between the direct and assembled predictions themselves, which their separate errors against truth leave open. Multi-step prediction additionally requires earlier predictions to place later responses at the right states.

Table 1 measures both predictions against the true joint increment at the same evaluation states. Under the matched pool, balanced pairing reduces the direct joint error by 8.3–29.2% and the error of the model's own assembled prediction by 8.4–25.4%, with paired intervals excluding zero in all eight comparisons for each. OTHER-SOURCE improves five of eight under both measures and deteriorates in the same three cells, `Walker2d` with the MLP ensemble and `Hopper` with both learners. The Direct and Assembled columns track each other cell by cell, so the improvement the pairing produces is available to a joint prediction assembled from single-actuator responses and is not confined to the network's direct joint output. The two predictors also sit at similar absolute error, with the assembled prediction between 0.98 and 1.08 times the direct error across the 24 environment–learner–pairing cells.

Table 10: One-step simulator additivity. Mean norms at the shared all-on evaluation states. Share divides the mean residual norm by the mean actuated-response norm.

Environment	$\|\Delta_0\|$	$\|\Delta - \Delta_0\|$	$\|\Delta - \Delta_{\text{sum}}\|$	Share
Walker2d	4.75	13.3	1.75	13.2%
HalfCheetah	19.9	11.3	1.54	13.6%
Hopper	0.312	1.38	0.0241	1.7%
Ant	8.24	9.19	0.533	5.8%

E.5 TARGET-CONTEXT REFERENCES

IN-CONTEXT and ALL-ON share training states from all-on behaviour but use single-actuator and joint actions, respectively. IN-CONTEXT serves as a control on the inference drawn from the matched pool, because it improves the composition through a different route. Relative to RESET, IN-CONTEXT reduces passive-increment error by 45–65% and actuator-response error by 34–57%, with the passive relative improvement larger in all eight comparisons (Table 5). This reverses the matched-pool pattern, where the response improvement is the larger one in all eight. An intervention can therefore improve compositional prediction without acting mainly through per-actuator response information, so an improvement on the composition is not by itself evidence about which part of the prediction changed. The matched pool supports that inference because it holds the states fixed and reads the two diagnostic errors separately.

The reference recovers 89–137% of the reset-to-all-on joint prediction difference, where the contrast is $(E_{\text{Reset}} - E_{\text{InContext}})/(E_{\text{Reset}} - E_{\text{AllOn}})$. Values above 100% arise when IN-CONTEXT has lower error than ALL-ON. Target-context restoration changes the training states along with their actuator-specific observations; the matched pool isolates pairing because it keeps the original states.

Table 11: CARRIED relative to each comparator. Paired percentage reductions in one-step error at the default budget. Positive values favour CARRIED; negative values favour the comparator. Intervals are 95% intervals. CYCLE, TR.-BAL., STEP, and CORR. abbreviate FIXED CYCLE, TRANSITION-BALANCED, SWITCH EVERY STEP, and CORRELATED; the two DIVERSE-RESET rows use perturbation scales 0.15 and 0.35 and require additional initial-state access.

	Walker2d		HalfCheetah		Hopper		Ant	
Comparator	MLP	Latent	MLP	Latent	MLP	Latent	MLP	Latent
RESET	20 ± 2	20 ± 3	16 ± 2	19 ± 3	35 ± 6	34 ± 9	19 ± 2	6 ± 2
CYCLE	9 ± 3	9 ± 4	5 ± 3	9 ± 4	4 ± 12	-1 ± 8	6 ± 6	0 ± 3
TR.-BAL.	-0 ± 3	-1 ± 4	-1 ± 2	-1 ± 4	-5 ± 4	-4 ± 9	7 ± 17	-1 ± 6
STEP	-27 ± 3	-7 ± 3	21 ± 10	25 ± 7	-24 ± 36	-26 ± 10	-47 ± 4	-48 ± 7
CORR.	38 ± 2	39 ± 2	38 ± 1	38 ± 2	51 ± 6	52 ± 7	31 ± 2	23 ± 3
DIVERSE-RESET 0.15	8 ± 2	8 ± 2	5 ± 2	0 ± 3	-5 ± 5	7 ± 6	3 ± 14	4 ± 6
DIVERSE-RESET 0.35	19 ± 1	15 ± 2	3 ± 4	-16 ± 4	27 ± 4	35 ± 4	-16 ± 5	-3 ± 2

Table 12: Absolute one-step errors at 48 000 transitions. Means over five collection draws. Abbreviations follow Table 11. DIVERSE-RESET conditions require initial-state access, while the IN-CONTEXT and ALL-ON references require target behaviour. Absolute values use each environment's physical coordinates.

	Walker2d		HalfCheetah		Hopper		Ant	
Protocol	MLP	Latent	MLP	Latent	MLP	Latent	MLP	Latent
RESET	5.20	5.71	3.76	5.68	0.270	0.342	8.37	9.91
CARRIED	4.15	4.57	3.16	4.60	0.176	0.225	6.82	9.28
CYCLE	4.55	5.04	3.32	5.05	0.184	0.222	7.25	9.30
TR.-BAL.	4.14	4.55	3.14	4.54	0.167	0.216	7.44	9.18
STEP	3.26	4.28	4.01	6.14	0.153	0.179	4.64	6.28
CORR.	6.71	7.56	5.07	7.37	0.360	0.472	9.94	12.01
DIVERSE-RESET 0.15	4.53	4.97	3.32	4.61	0.167	0.242	7.08	9.68
DIVERSE-RESET 0.35	5.10	5.39	3.25	3.97	0.242	0.348	5.87	8.98
IN-CONTEXT	2.62	3.15	2.02	2.73	0.117	0.155	2.85	5.06
ALL-ON	2.44	2.86	1.86	2.38	0.116	0.157	2.79	6.37

F ADDITIONAL COLLECTION RESULTS

F.1 PROTOCOL COMPARISONS AT THE DEFAULT BUDGET

Table 11 reports the paired reduction of CARRIED relative to each comparator; positive values favour CARRIED. Table 12 gives absolute one-step errors for every condition. The absolute metric depends on each environment's physical coordinates and units, so comparisons are made within an environment. DIVERSE-RESET, IN-CONTEXT, and ALL-ON require additional access as specified in Appendix D.3. With this extra access, DIVERSE-RESET at scale 0.35 has lower mean error than CARRIED in three of eight comparisons.

The switching interval and the amplitude-refresh schedule are separate controls. Holding the amplitude rule fixed, changing $h = 50$ to $h = 5$ lowers error in all eight comparisons, with monotonic decreases over the intermediate tested values in seven. The $h = 1$ versus $h = 5$ changes are mixed, and six intervals include zero. At $h = 1$, the fixed-refresh variant and the variant that redraws the amplitude at every step differ with intervals excluding zero in three comparisons, all favouring the fixed-refresh variant. The switching interval therefore controls error over this range, and the gain saturates by $h = 5$.

F.2 BUDGET AND HISTORY LENGTH ON ANT

Table 13 separates changes in transition budget from changes in history cap. The relative one-step carried advantage increases with budget for both learners. A longer maximum history does not produce the same benefit; the latent model has negative mean carried-versus-reset reductions at caps 200 and 400. The recorded history-length distributions show that termination can prevent the cap from being reached. At ten steps, the MLP ensemble keeps the increasing trend with budget, from 1.3% to 3.3% and 6.5%.

Table 13: Budget and history-cap studies on `Ant`. The RESET, CARRIED, and TR.-BAL. (TRANSITION-BALANCED) columns show absolute one-step errors. The last columns show paired percentage reductions of CARRIED relative to RESET at each horizon. Budgets are in thousands of transitions (k). Budget and history-cap changes are separate comparisons.

Learner	Budget / cap	RESET	CARRIED	TR.-BAL.	1 step (%)	10 steps (%)
MLP	48k / 100	8.37	6.82	7.44	18.6 ± 2.0	1.3 ± 1.4
	96k / 100	7.18	5.66	5.44	21.1 ± 1.5	3.3 ± 2.2
	192k / 100	5.84	4.35	4.30	25.6 ± 1.0	6.5 ± 1.7
	48k / 200	8.17	7.12	7.54	13.0 ± 8.1	0.9 ± 6.9
	48k / 400	8.53	7.29	7.33	14.5 ± 4.1	1.2 ± 1.5
Latent	48k / 100	9.91	9.28	9.18	6.3 ± 2.5	1.5 ± 5.5
	96k / 100	8.53	7.59	7.33	10.9 ± 6.1	-0.3 ± 7.3
	192k / 100	8.02	6.38	6.37	20.4 ± 4.2	1.4 ± 9.6
	48k / 200	9.98	10.17	9.87	-1.8 ± 4.7	-4.6 ± 6.1
	48k / 400	9.92	10.49	10.35	-5.9 ± 6.9	-4.1 ± 3.5

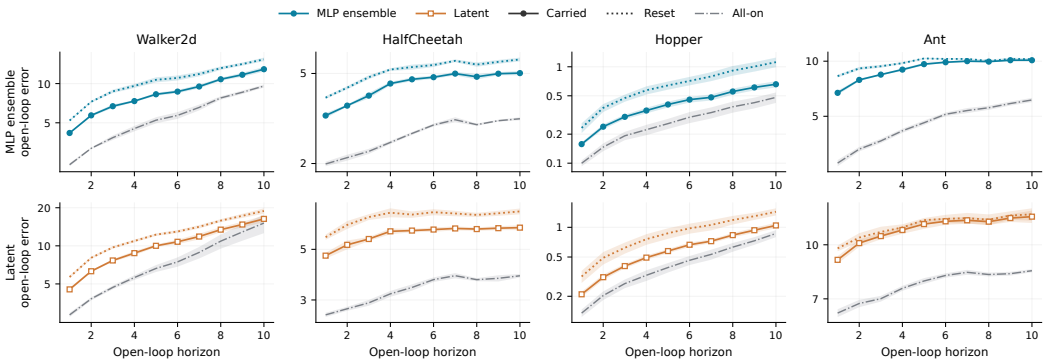

Figure 6: Absolute open-loop prediction errors for both learners. Rows show the MLP ensemble and latent model, and columns show the four environments. Curves are means over five collection draws on logarithmic axes, with 95% bands. ALL-ON is a reference that uses target behaviour.

F.3 EVALUATION-SET VARIATION

Every default-budget condition is scored on five additional independent all-on evaluation sets. The carried-versus-reset reduction remains positive in all 40 environment–learner–set combinations, with reductions of 3.0–43.4%. These evaluations measure sensitivity to resampling the same random-action target behaviour, with the collection draws held fixed.

F.4 OPEN-LOOP PREDICTION

Figure 6 gives absolute errors on the all-on target for both learners through ten steps. Absolute curves distinguish small relative differences at low error from small relative differences after substantial error accumulation. Across reported ten-step rollouts, no non-finite values were recorded.

F.5 COVERAGE MEASUREMENTS

The matched-pool audit computes median nearest-neighbour distance from evaluation states to pooled training states, and separately to states observed under each actuator. Pooled distances are identical for all three pairings in all 20 environment–draw pools. The per-actuator distance decreases by 5.3–30.5% from SOURCE-ALIGNED to BALANCED PAIRING, but does not consistently rank the three pairings by prediction error. OTHER-SOURCE has a smaller distance than SOURCE-ALIGNED in every environment while increasing error for both learners in `Hopper`. These measurements separate per-actuator from pooled coverage; ranking pairings by prediction error requires a different statistic.

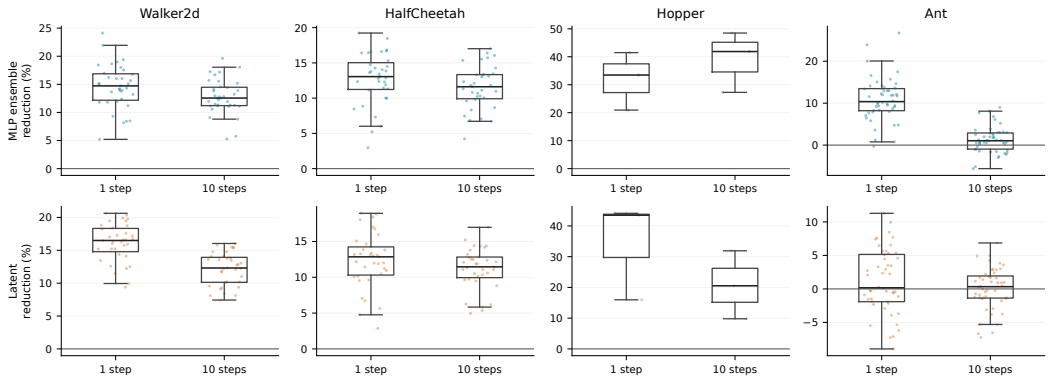

Figure 7: Distribution of paired reductions across unseen targets. Paired percentage reductions of CARRIED relative to RESET at one and ten steps. Points are target subsets; boxes show quartiles. The two rows correspond to the MLP ensemble and latent model.

F.6 UNSEEN ACTUATOR SUBSETS

The 121 evaluated subsets comprise 35 pairs and triples in each six-actuator environment, three pairs in `Hopper`, and 48 subsets in `Ant` (28 pairs and 20 triples). Each target uses its own state context and action sequences. Table 14 reports counts by environment and horizon; Figure 7 shows the distribution of relative effects. The MLP median reduction is 13% at one step, with quartiles 10–16%, and 10% at ten steps, with quartiles 2–13%. Latent-model medians are 11% and 9%. Counts and distributions describe generalisation across targets that share models and collection data.

Table 14: Unseen targets on which CARRIED has lower mean error than RESET. Counts at one and ten steps. All targets share the same trained models and collection data.

	MLP		Latent	
Environment	One step	Ten steps	One step	Ten steps
Walker2d	35/35	35/35	35/35	35/35
HalfCheetah	35/35	35/35	35/35	35/35
Hopper	3/3	3/3	3/3	3/3
Ant	47/48	34/48	25/48	25/48
Total	120/121	107/121	98/121	98/121

F.7 PREDICTION ON FIXED-CONTROLLER TRAJECTORIES

Table 15 reports paired reductions of CARRIED relative to RESET, with 95% intervals, on trajectories recorded by one fixed CEM controller per environment acting in the true simulator. The intervals are wide on these trajectories, and the three that exclude zero, one of eight at one step and two of eight at ten steps, all favour CARRIED. Since all models receive the same recorded action sequences, this result is distinct from the closed-loop planning evaluations reported in Appendix G.

Table 15: Prediction on fixed-controller trajectories.

Horizon	Environment	MLP (%)	Latent (%)
1 step	Walker2d	-9.6 ± 14.8	15.0 ± 20.1
	HalfCheetah	-9.5 ± 13.7	-24.1 ± 41.0
	Hopper	-10.2 ± 64.8	32.9 ± 44.3
	Ant	35.6 ± 14.4	6.9 ± 12.0
10 steps	Walker2d	15.1 ± 4.6	-14.6 ± 42.1
	HalfCheetah	1.0 ± 22.9	-22.9 ± 42.7
	Hopper	50.6 ± 22.9	19.2 ± 43.2
	Ant	3.5 ± 6.9	4.7 ± 25.8

G EXPLORATORY STUDIES AND IMPLEMENTATION PROVENANCE

G.1 RELATIONSHIP TO THE MAIN EXPERIMENTS

The following studies were conducted with an earlier implementation and are not pooled with the five-draw main comparisons. The main latent implementation corrects the conversion between separately normalised states and increments; training seeds are also set before network construction. Earlier latent results therefore describe the uncorrected learner. Earlier experiments additionally included an action-affine model and a world model built from TD-MPC2 modules with its value term removed, which is a component of that system rather than the complete one. Original experiment records are retained separately.

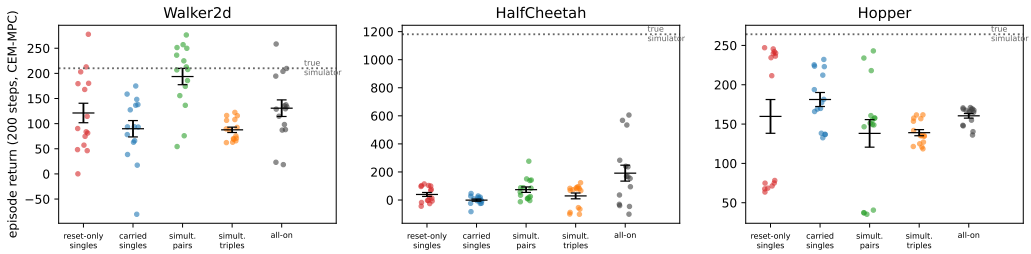

Figure 8: Exploratory plain CEM-MPC results. Episode returns from the earlier MLP study. These runs are separate from the main collection-draw experiments and from fixed-controller prediction evaluation.

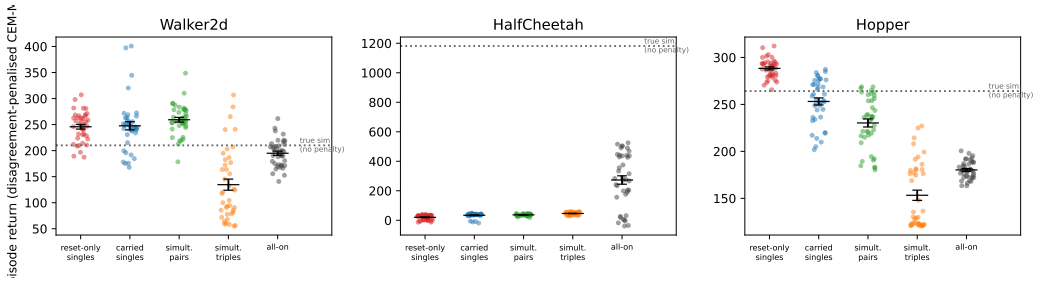

Figure 9: Exploratory disagreement-penalised CEM-MPC results. Episode returns from the earlier MLP study under a planner that subtracts ensemble disagreement. The underlying model runs use the earlier implementation.

G.2 Closed-Loop Planning

An earlier MLP ensemble was used within a common CEM-MPC planner under reset, carried, simultaneous-pair, simultaneous-triple, and all-on training conditions. Plain planning used three seeds and five episodes per seed; disagreement-penalised planning used five seeds and eight episodes per seed. Episodes were capped at 200 steps. The planning horizon was 30 for `Walker2d` and `Hopper`, and 12 for `HalfCheetah`, with 200 candidates, 20 elites, and three iterations. The penalised objective subtracted ensemble disagreement with weight one.

Plain planning produced no consistent return ordering across data conditions. Under disagreement-penalised planning, `HalfCheetah` returns were 20, 34, 37, 46, and 273 for reset, carried, pairs, triples, and all-on training, respectively; the low-order conditions reached only 2–4% of the true-simulator reference. `Walker2d` carried and reset returns were similar (248 versus 246). On `Hopper`, reset returned 288 and carried 253, compared with 230 for pairs, 153 for triples, and 180 for all-on training. These exploratory runs do not separate the data conditions by return (Figures 8 and 9).

G.3 Certificate-Based Collection and Finite Noise

In the earlier neural collection comparison, the fixed switching schedule achieved lower final prediction error than the certificate-inspired rule in all three tested tasks. The finite-noise bound exceeded realised error by a factor of approximately 10^4–10^5 at measurement tolerance 10^{-4}. A calibrated nonlinear selection criterion and a tight finite-noise trajectory bound therefore remain open beyond the exact certificate.

G.4 Model Capacity

The earlier capacity sweep used a common learning rate and epoch budget across architectures. On `Ant`, a larger residual model trained on reset data achieved error 6.21, below the small carried-trained model's 6.30, and carried data improved the same residual architecture further to 5.37. Capacity therefore compensates in part for reset-only collection, and carried collection still improves the larger model.

G.5 Actuator Withholding, Pretraining, and Passive Observations

The earlier actuator-withholding intervention changed both an actuator's role in preparing states and its opportunity to respond after other actuators, confounding the two directions of effect. The specified own-actuator attribution criterion was not met, and the coverage proxy reversed actuator rankings in `HalfCheetah`. The matched-pool intervention instead fixes state preparation and changes probe assignment.

The earlier pretrained-prior study did not meet its specified admissibility criterion, so its fine-tuning results are descriptive. Improvements from passive, action-free observations were task- and learner-dependent, with negative comparisons as well as positive ones, leaving their complementarity with carried collection open. Neither extension enters the matched-pool result or the main collection estimates.