CAUSAL WORLD MODELS VIA DISENTANGLED AUTONOMOUS DYNAMICS AND AGENT INTERVENTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicting a transition does not uniquely determine which changes arise from background dynamics and which are associated with an action. We introduce the Causal Disentanglement World Model (CDWM), a structured neural world model that assigns these contributions to an Environment Pathway and an Intervention Pathway. The recurrent environment branch excludes the current action, while the feed-forward intervention branch conditions on state and action. Their sum supports Monte Carlo tree search, and the intervention magnitude defines an Agency Bonus for exploration. This organization makes pathway contributions explicit, while the encoder and both pathways remain learned black-box functions. We establish conditional-mean identification under state sufficiency, a valid null-action anchor, action coverage, realizability, and global population risk minimization; the architecture alone does not establish these conditions. Evaluation on 26 Atari100k games, branch-removal tests, and a Venture bonus ablation provides evidence about task performance, pathway dependence, and exploration, respectively. Representation visualizations and supplementary prediction studies offer complementary diagnostics, not empirical proofs of causal identification. The results support a useful structural inductive bias with inspectable component outputs, rather than a fully transparent or unconditionally identified model of Atari dynamics.

1 INTRODUCTION

For planning, it matters not only what changes after an action, but which part of that change the action caused. A visual environment continues to evolve while an agent acts: objects move, hazards approach, and observations may fluctuate independently of the chosen control. A predictive model can fit the resulting transition without distinguishing these sources. Yet a planner must compare alternative actions, and an exploring agent must decide which changes are worth seeking. Both decisions benefit from separating autonomous evolution from the agent's contribution.

Model-based reinforcement learning addresses sample efficiency by learning dynamics that can be queried before committing to actions (Luo et al., 2022; Ha & Schmidhuber, 2018; Hafner et al., 2020; Schrittwieser et al., 2020; Ye et al., 2021). A conventional latent model predicts the next state from the current state and action. This objective supervises the aggregate transition, but it does not specify how to attribute that transition to environmental evolution or intervention. The distinction is not that every unstructured predictor is necessarily confounded: under sufficient state conditioning, observational and interventional predictions can coincide. Rather, accurate prediction alone does not identify a unique decomposition of the mechanisms underlying that prediction. With partial observability or limited action coverage, the interpretation of learned action effects becomes more demanding.

We study a structured alternative, the *Causal Disentanglement World Model* (CDWM). Figure 1 illustrates the central separation: the environment continues to evolve, while candidate actions alter the controlled part of the transition. The model represents a latent update through an Environment Pathway and an Intervention Pathway. The former is recurrent and does not receive the current action; the latter is a feed-forward mapping of state and action. Their outputs are added to the

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.

current latent state and used within Monte Carlo tree search (MCTS). This exposes an *inspectable decomposition* of the predicted increment. It does not make the full system transparent: the encoder, recurrent dynamics, and intervention network remain black-box neural functions. Interpretability here concerns named, queryable pathway contributions, not human-readable latent coordinates or recovered physical laws.

Architecture alone, however, does not make this split unique. An action-independent function can be added to one pathway and subtracted from the other without changing the prediction. Our analysis therefore combines the input restriction with a null-action anchor, following the identifying conditions in Proposition 1. Under an additive generative model, adequate action coverage, and population risk minimization, the environmental component is the expected change under the null action, and the intervention component is the difference from that baseline. These conditions distinguish a causal interpretation from an unconstrained architectural attribution.

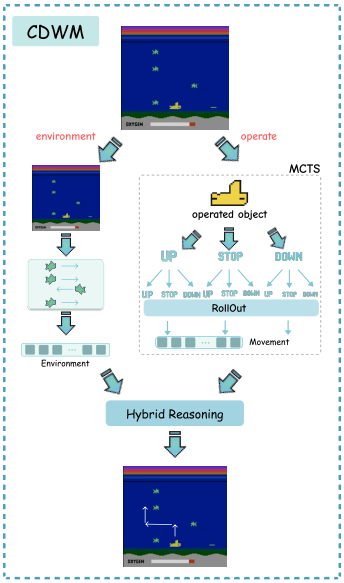

Figure 1: Dual-path planning.

The decomposition also guides exploration. Prediction-error rewards can respond to environmental variation that the agent cannot control (Pathak et al., 2017; Burda et al., 2019). Instead, our Agency Bonus is computed from the magnitude of the predicted intervention and normalized using running statistics. It favors actions with a substantial modeled effect relative to the local baseline. Under the identifying assumptions, independent zero-mean transition noise does not change the population intervention component. Proposition 2 states this conditional robustness property, not a guarantee that every learned effect is useful for the task or that finite-data estimation is insensitive to noise.

Our experiments use three seeded runs (0, 1, and 42). The 26-game Atari100k comparison measures task performance; branch removal tests dependence on a trained component; and the Venture comparison isolates the bonus within the dual-path model on that task. Representation plots describe latent geometry, not game-playing ability or absence of cross-pathway leakage. Supplementary studies examine forced-action prediction, mode switching, continuous control, and cost. These distinct measurements support the practical utility of the structural prior without establishing either benchmark-wide attribution of its gains or the identifying assumptions for every learned Atari model.

2 Problem Formulation

We consider model-based reinforcement learning in a partially observable environment (Åström, 1965). A representation $s_t = h_\theta(o_{\le t}, a_{<t})$ summarizes the available history; replay data $\mathcal{D}$ supervise a model that is later queried on candidate actions. The conventional prediction objective is

$$\min_{\phi} \; \mathbb{E}_{(s_t, a_t, s_{t+1}) \sim \mathcal{D}}[-\log p_\phi(s_{t+1} \mid s_t, a_t)]. \tag{1}$$

It constrains the total transition, but does not specify which part of a change would have occurred without the action. Our target is this attribution, not merely another parameterization of the same next-state predictor.

Prediction and intervention. Replay supervises $P(s' \mid s, a)$, whereas planning asks about $P(s' \mid s, \mathrm{do}(a))$ (Pearl, 2009). These distributions coincide when the state blocks the action–outcome backdoor path in the assumed causal graph. With omitted confounders, they need not coincide. We therefore make state sufficiency an explicit condition; excluding the current action from one network is not itself a remedy for missing state information. Appendix F.2 separates this causal condition from the function-identification argument below.

Assumption 1 (Additive reference dynamics)**.** *Fix a latent representation sufficient to adjust for action assignment. Its structural transition has finite second moments and satisfies*

$$s_{t+1} := s_t + f^\star_{\text{env}}(s_t) + f^\star_{\text{agent}}(s_t, a_t) + \epsilon_t, \qquad \mathbb{E}[\epsilon_t \mid s_t, a_t] = 0. \tag{2}$$

There is a reference action a_0 with $f^\star_{\text{agent}}(s, a_0) = 0$. The noise is exogenous in the state-sufficient causal model.

The autonomous increment and action increment may each be nonlinear in their inputs. Their outputs are added; neither network is assumed linear. A null action removes the intervention contribution, not the environment's evolution. The independent Gaussian example $\epsilon_t \sim \mathcal{N}(0, \sigma^2 I)$ is sufficient, but the conditional-mean argument only requires the stated zero-mean noise condition.

Why the input restriction is not enough. Write $\Delta(s, a) = \mathbb{E}[s' - s \mid s, a]$. An action-free environmental branch and an action-conditioned branch can fit this mean through $\Delta(s, a) = f_{\text{env}}(s) + f_{\text{agent}}(s, a)$. Yet any admissible state function $d(s)$ yields an equally predictive split:

$$f'_{\text{env}}(s) = f_{\text{env}}(s) + d(s), \qquad f'_{\text{agent}}(s, a) = f_{\text{agent}}(s, a) - d(s). \tag{3}$$

This attribution ambiguity is invisible to Equation (1): its two adjustments cancel. It is visible to an intrinsic reward based on $\|f_{\text{agent}}(s, a)\|_2$, which can change while every predicted successor stays fixed. The reference is therefore needed for both the meaning of the pathways and the meaning of the bonus.

There is an important distinction between action comparison and absolute attribution. At an exact predictive fit, $f_{\text{agent}}(s, a) - f_{\text{agent}}(s, b) = \Delta(s, a) - \Delta(s, b)$ for any two supported actions, regardless of the shift. An unanchored model can therefore compare mean successors correctly while assigning an arbitrary magnitude to either intervention separately. The anchor does not create predictive information; it selects a common origin for the component that CDWM reuses as an intrinsic signal.

Proposition 1 (Anchored identification)**.** *Under Assumption 1, suppose each compared discrete action, including a_0, has positive conditional probability at the relevant states. Let the function classes contain the true pair, let the environment branch exclude the current action, and enforce $f_{\text{agent}}(s, a_0) = 0$. Every global minimizer of population squared prediction risk then has the unique component functions*

$$f_{\text{env}}(s) = \Delta(s, a_0), \qquad f_{\text{agent}}(s, a) = \Delta(s, a) - \Delta(s, a_0). \tag{4}$$

The equality is almost sure on the supported state–action distribution.

Proof sketch. Squared risk fixes the sum at $\Delta(s, a)$. Evaluating that sum at the supported reference action and using the learned zero constraint fixes $f_{\text{env}}(s) = \Delta(s, a_0)$. Subtracting this baseline recovers $f_{\text{agent}}(s, a)$. Any alternative shift in Equation (3) would violate the same zero constraint unless $d(s) = 0$. Appendix F.2 gives the risk decomposition and complete proof.

The proposition identifies baseline and contrast functions in a fixed representation, not neural parameters or physical factors. Since both functions can be nonlinear, addition alone does not exclude complex conditional means: any supported mean admits a reference-plus-difference split. The substantive claim is that the reference is a valid autonomous baseline. For Atari, a `NOOP` label does not establish this, a finite replay buffer does not establish full action coverage, and neural training does not certify global population optimality. A state that omits a changing regime can also invalidate causal sufficiency. These are conditions of the proof, not facts inferred from game scores or separated representation clusters. Section 3.1 translates the input restriction and anchor into the learned transition; Section 3.2 uses the identified difference for exploration; Section 3.3 connects the proof's squared risk to the training loss. The penalty alternative is treated separately in Appendix F.3.

3 Causal Disentanglement World Model

CDWM parameterizes the two components of Section 2. Read Figure 2 in three stages: encode a state, predict the background increment and action increment, then use their sum for planning and the latter's magnitude for the Agency Bonus. The game frames illustrate this computation, not measured counterfactual outcomes. Proposition 1 specifies when the learned components admit the intended causal interpretation.

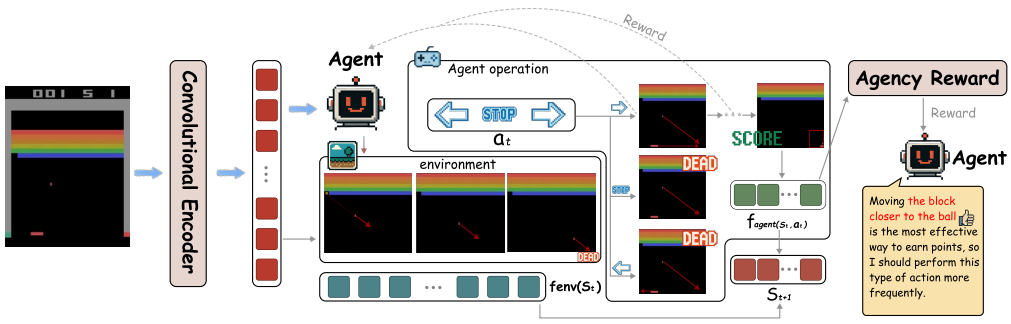

Figure 2: CDWM: dual-path transitions and intervention-based exploration.

3.1 Dual-Path Latent Dynamics

The Environment Pathway f_{env} uses a gated recurrent unit followed by an MLP; the Intervention Pathway f_{agent} uses a gated feed-forward MLP on the state and action encoding. The current action is excluded from the environment branch. Recurrent context is implicit in s and is carried consistently through the rollout. In particular, comparing candidate actions must retain the same pre-action state and context, rather than changing the baseline separately for each candidate. Layer dimensions and prediction heads appear in Appendix B.

For a fixed state, let $\hat{s}'(a)$ denote the predicted successor. With the null-action constraint $f_{\text{agent}}(s, a_0) = 0$, the learned model gives

$$\hat{s}'(a) = s + f_{\text{env}}(s) + f_{\text{agent}}(s, a), \qquad \hat{s}'(a) - \hat{s}'(a_0) = f_{\text{agent}}(s, a). \tag{5}$$

The environmental term defines a shared reference; the intervention term is its predicted action contrast. The second identity is algebraic under the learned anchor, whereas equality to the true conditional-mean effect requires Proposition 1. This is functional interpretability: the contributions can be queried and ablated, but their neural implementations remain black boxes.

Under the independent Gaussian example in Assumption 1, the identified mean and noise law give $P(s' \mid s, \text{do}(a)) = \mathcal{N}(\hat{s}'(a), \sigma^2 I)$ at the ideal solution. The contrast in Equation (5) compares these means, not independently sampled disturbances. This connects the learned predictor to the intervention distribution without assigning transition noise to the action.

The two restrictions play different roles. Current-action exclusion makes the baseline common to alternatives, but still permits the shift in Equation (3). The anchor fixes that shift. It can be imposed as a hard output constraint; the source's vanishing-penalty alternative is analyzed in Appendix F.3. The layer tables and reported loss do not specify which was used in the experiments. The identification claim for an implemented model therefore requires this training detail, in addition to the GRU/MLP architecture. Neither restriction excludes past-action information from the state or guarantees that it is causally sufficient.

3.2 Agency-Guided Planning

The action contrast in Equation (5) separates agency from the total observed change. Its magnitude defines the Agency Bonus:

$$r_t^{\text{i}} = \sigma\left(\frac{\|f_{\text{agent}}(s_t, a_t)\|_2 - \mu_\phi(s_t)}{\sigma_\phi(s_t) + \delta}\right). \tag{6}$$

Here $\sigma(\cdot)$ is the sigmoid, $\delta > 0$, and $\mu_\phi(s_t), \sigma_\phi(s_t)$ are running statistics of intervention magnitudes in similar states. The bounded intrinsic reward favors modeled effects above this local baseline. It reuses the dynamics model rather than learning a separate curiosity predictor. Unlike a next-state prediction error, its numerator does not include the realized transition residual.

The two forms of centering must not be conflated. The reference action a_0 fixes the origin of the action effect in Equation (4); the running mean $\mu_\phi(s)$ rescales magnitudes after that origin has been fixed. Statistical normalization cannot replace causal anchoring: shifting a vector before taking its norm can change which actions receive larger bonuses, even if their total predictions agree.

Proposition 2 (Population separation from transition noise)**.** *Under Proposition 1, varying independent zero-mean transition noise while holding the structural functions fixed leaves the identified intervention and its magnitude unchanged at shared supported state–action pairs. The normalized reward in Equation* (6) *is also unchanged when its normalization statistics are held fixed.*

Proof sketch. The noise term vanishes in both conditional means in Equation (4); their difference is unchanged. In contrast, the conditional-mean predictor's squared error retains the noise variance. Appendix C.2 proves the statement and gives the scalar example from the source analysis. This is a population property, not a finite-data guarantee: estimation error or changing normalization statistics can still change the bonus. Nor does a large effect necessarily imply task progress (Appendix E.5).

MCTS queries Equation (5) and the reward head to obtain $(\hat{s}', \hat{r})$. For a parent state–action pair (s, a), it augments the immediate reward and backs up

$$G = \hat{r} + \lambda_{\text{agency}} r^{\text{i}}(s, a) + \gamma V. \tag{7}$$

The intrinsic contribution is attached to the same action contrast that produced $\hat{s}'$. Here V denotes a successor value or the return already accumulated from deeper in the tree. PUCT selects search actions, and root visit counts determine the executed action. Algorithm 2 uses the same parent-state convention. The backup is a planning rule; Proposition 2 does not assert that it improves task return.

3.3 Joint Learning and Interaction

The encoder, dynamics, and heads have parameters $\Theta = \{\theta, \phi, \psi\}$. For a replay transition, the squared dynamics loss is

$$\mathcal{L}_{\text{dynamics}}(s, a, s') = \|s' - s - f_{\text{env}}(s) - f_{\text{agent}}(s, a)\|_2^2. \tag{8}$$

Its population expectation is the risk minimized in Proposition 1. In Appendix F.2, Equation (19) separates this risk into error in the *total predicted increment* and irreducible noise. Prediction supervision fixes the sum; the reference constraint fixes its attribution. A small empirical loss alone does not establish recovery of either mechanism.

For trajectory unrolls of length K, training combines this loss with policy, value, and reward supervision:

$$\mathcal{L}(\Theta) = \mathbb{E}_{\tau \sim \mathcal{D}} \left[\sum_{k=0}^{K-1} \left(\mathcal{L}^{(k)}_{\text{policy}} + \lambda_v \mathcal{L}^{(k)}_{\text{value}} + \lambda_r \mathcal{L}^{(k)}_{\text{reward}} + \lambda_d \mathcal{L}^{(k)}_{\text{dynamics}} \right) \right]. \tag{9}$$

The dynamics term at each step uses Equation (8). The other heads supply the planner's task-directed predictions; they do not remove the ambiguity in Equation (3). A penalty-based anchor would add the term in Appendix F.3, whose finite training coefficient is not specified in the source. Moreover, joint optimization changes the encoder, whereas the identification proof fixes the representation. The population statement and the end-to-end training objective are therefore linked, but not interchangeable guarantees.

Algorithm 1 connects action comparison, reward augmentation, and replay supervision. Actual transitions enter the buffer before an update; planning queries do not themselves constitute environmental evidence. The reported configuration uses five-step unrolls and 50 MCTS simulations per decision (Appendix B).

Algorithm 1 CDWM interaction and replay learning

Input: Encoder h_θ, pathways (f_{env}, f_{agent}), heads f_ψ; training schedule and identifying constraint
Output: Trained parameters Θ and replay buffer $\mathcal{D}$

Initialize Θ and $\mathcal{D} \leftarrow \varnothing$
for each episode **do**
  Observe o_0 and initialize recurrent context
  for $t = 0, \ldots, T-1$ **do**
    $s_t \leftarrow h_\theta(o_{\le t}, a_{<t})$
    $a_t \leftarrow \mathrm{MCTS}(s_t, g_\phi, f_\psi)$ ▷ Transition (5); bonus (6)
    Execute a_t; observe $(o_{t+1}, R_t, \text{done})$
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{(o_t, a_t, R_t, o_{t+1}, \text{done})\}$
    if an update is due and replay is ready **then**
      Sample trajectory unrolls from $\mathcal{D}$
      Update Θ with (9), using (8) and the null-action constraint
    end if
    if done **then break**
  end for
end for

4 Experiments

We separate four questions: how well the complete agent performs, whether a trained agent depends on each pathway, how the bonus changes exploration, and what the representations encode. Each study measures a different object; none is used as a substitute for the assumptions of Proposition 1. Appendix G summarizes these evidence boundaries and the reporting protocol.

4.1 Benchmark and Evaluation Protocol

Atari100k contains 26 games with 100,000 environment interactions per run, approximately 400,000 frames (Ye et al., 2021). Our experimental results use three seeded runs, with seeds 0, 1, and 42; this statement does not assign the same protocol to published baseline entries. Human-normalized score is $\mathrm{HNS} = (\text{agent} - \text{random})/(\text{human} - \text{random})$. Table 1 gives raw game returns. Bold and underlined values mark the largest and second-largest distinct model scores, with ties treated equally; random and human returns are references.

The comparisons include SimPLe, SPR, TWM, IRIS, STORM, DreamerV3, DramaXS, BBF, and EZ-V2 (Kaiser et al., 2020; Schwarzer et al., 2021; Robine et al., 2023; Micheli et al., 2023; Zhang et al., 2023; Hafner et al., 2024; Wang et al., 2025; Schwarzer et al., 2023; Wang et al., 2024). The three-run protocol provides replication, but the main table does not report between-run dispersion. Small differences therefore remain descriptive, not claims of statistical significance. In particular, seed IDs alone do not determine error bars or confidence intervals. Architecture and settings appear in Appendix B; aggregation details are discussed in Appendix G.1.

4.2 Performance on Atari100k

CDWM has strong reported returns on Bank Heist (1962), Breakout (462), Krull (10642), and PrivateEye (10296). The advantage is not uniform: EZ-V2 scores higher on Alien and Asterix, BBF on BattleZone, and SPR on UpNDown. These are *complete-system* comparisons, not estimates of the isolated contribution of either the dynamics decomposition or the Agency Bonus.

The reported CDWM mean and median are 2.81 and 1.24, compared with 2.69 and 1.23 for EZ-V2. Table 1 retains these aggregates pending reconciliation with the game-level entries.[1] We do not infer statistical superiority or a causal mechanism from this small reported aggregate margin. The following studies address narrower questions.

[1]The source mixes ratio-like method aggregates with a human reference of 100. Applying its stated HNS formula to the 26 displayed CDWM returns gives approximately 2.60 and 1.17, rather than 2.81 and 1.24. The daggered values are retained, not treated as a verified overall ranking. Three seeds do not by themselves resolve the aggregation discrepancy.

Table 1: Atari100k returns and reported aggregates; our runs use seeds 0, 1, and 42.

Game	Random	Human	SimPLe	SPR	TWM	IRIS	STORM	Dreamer V3	Drama XS	BBF	EZ-V2	CDWM
Alien	228	7128	617	823	675	420	984	959	820	1173	**1558**	1285
Amidar	6	1720	78	176	122	143	205	139	131	245	185	**284**
Assault	222	742	527	570	683	1524	801	706	539	**2091**	1758	1678
Asterix	210	8503	1128	969	1117	854	1028	932	1632	3946	**61810**	23765
Bank Heist	14	753	34	375	467	53	641	649	137	733	1317	**1962**
BattleZone	2360	37188	5032	15642	5068	13074	13540	12250	10860	**24460**	14433	18456
Boxing	0	12	8	36	78	70	80	78	78	86	75	**88**
Breakout	2	30	16	18	20	84	16	31	7	371	400	**462**
Chopper Command	811	7388	1078	965	1697	1565	1888	420	1642	**7549**	1197	2364
Crazy Climber	10780	35829	62584	39642	71820	59324	66776	97190	83931	58432	**112363**	72453
DemonAttack	152	1971	208	532	350	2034	165	303	201	13341	**22774**	16742
Freeway	0	30	18	23	24	31	34	0	15	26	0	**35**
Frostbite	65	4335	242	1623	1476	259	1316	909	785	2385	1136	**2554**
Gopher	258	2412	645	682	1675	2236	**8240**	3730	2757	1331	3869	2765
Hero	1027	30826	2657	6274	7254	7037	11044	11161	7946	7819	9705	**14896**
Jamesbond	29	303	112	366	362	463	509	445	372	**1130**	468	579
Kangaroo	52	3035	238	3423	1240	838	4208	4098	1384	**6615**	1887	3972
Krull	1598	2666	2805	3684	6349	6616	8413	7782	9693	8223	9080	**10642**
KungFu Master	258	22736	16541	13762	24555	21760	26183	21420	23920	18992	**28883**	25766
MsPacman	307	6952	1480	1316	1588	999	2673	1327	2270	2008	2251	**2860**
Pong	-21	15	13	-5	19	15	11	18	15	17	**21**	**21**
PrivateEye	25	69571	47	108	87	100	7781	882	90	41	100	**10296**
Qbert	164	13455	1289	735	3331	746	4522	3405	796	4447	**16058**	13260
RoadRunner	12	7845	5641	13622	9109	9615	17564	15565	14020	**33427**	27517	18692
Seaquest	68	42055	683	558	774	661	525	618	497	1233	1974	**2464**
UpNDown	533	11693	3350	**23162**	15982	3546	7985	7667	7387	12102	15224	18765
Mean†	0	100	0.37	0.68	0.96	1.04	1.27	1.12	1.05	2.26	2.69	2.81
Median†	0	100	0.14	0.41	0.51	0.29	0.58	0.49	0.27	0.92	1.23	1.24

4.3 Pathway Dependence across Tasks

Figure 3 evaluates a converged model after deactivating either branch. Within each game, the bars are *Env-ablated*, *Full Model*, and *Act-ablated*, from left to right. Pink panels denote the six control-dominant tasks and blue panels the six environment-dominant tasks; these colors identify task groups, not pathway identities. Compare bars *within* a panel: the games use different score scales.

For Pong, the returns are 13, 21, and 4; for Seaquest, they are 589, 2464, and 1546. The opposite sensitivities support task-dependent *functional reliance*. They do not establish that either branch contains only its designated factor. Removal also changes a trained model's input to the planner and may change the bonus if it is recomputed from the removed branch. This is therefore not a capacity-matched retraining ablation or a clean isolation of dynamics from intrinsic reward.

4.4 What the Representation Visualization Measures

For each sampled state–action pair, the study extracts the vectors $f_{\text{env}}(s_t)$ and $f_{\text{agent}}(s_t, a_t)$ at early, middle, and late checkpoints. Figure 4 shows density views of their reported 2D t-SNE projections. The axes are projection coordinates, not returns, game positions, or elapsed training time. The underlying observations are pathway outputs, not successful or unsuccessful episodes.

The late view has two density modes, but neither is pathway-labelled. Geometry alone does not demonstrate better returns, causal identification, or absence of leakage.

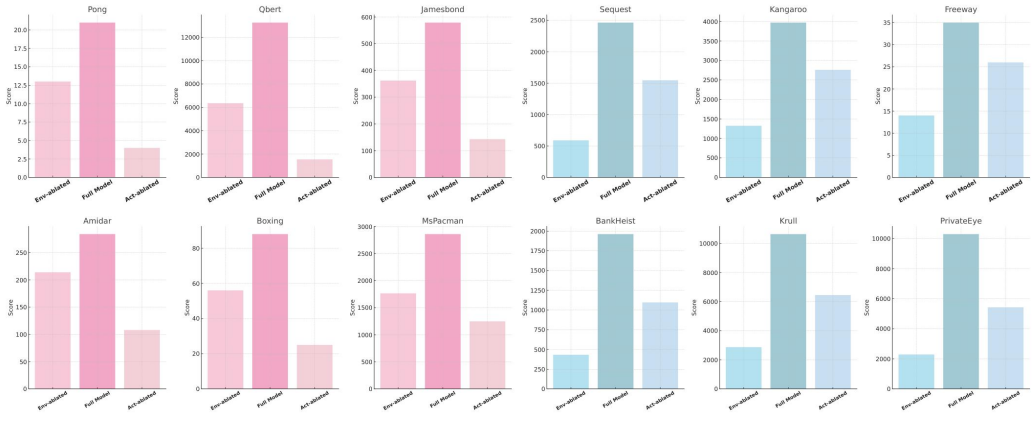

Figure 3: Within-game returns after removing either pathway from the trained model.

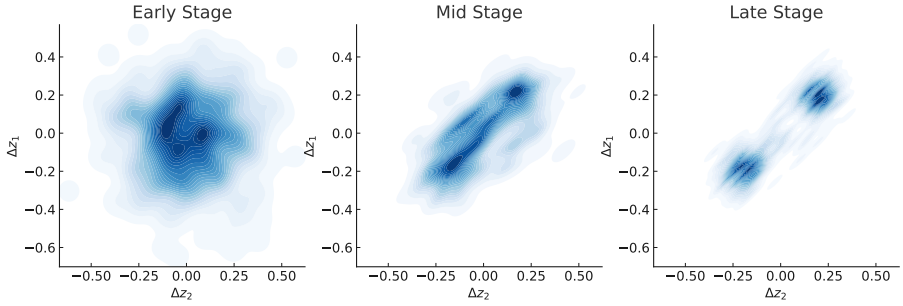

Figure 4: Projected pathway-output densities at early, middle, and late training.

4.5 Agency Bonus and Exploration

The Venture comparison holds the dual-path architecture and optimization setup fixed and removes the Agency Bonus; the no-bonus condition includes a constant reward offset to match scale. It tests the bonus *within CDWM on Venture*, not its contribution across all 26 Atari100k games. In Figure 5, the upper panels measure return and unique latent-hash states during the first 1000 steps; the lower panel tracks the bonus. These measure task reward, a coverage proxy, and the intrinsic signal, respectively.

The bonus-enabled condition has higher plotted early return and coverage. The three-run seed protocol is now explicit, but the shaded-band statistic remains unspecified. The source describes 10,000 steps for the lower panel, whereas the retained axis reaches 100,000; this discrepancy still requires reconciliation. Appendix E.5 reports that larger bonus weights can increase coverage while reducing return. Thus the evidence supports a task-specific exploration benefit, not a general claim that stronger agency always improves performance. Benchmark-wide separation of dynamics and bonus contributions remains unmeasured in the supplied results.

Supplementary evidence. Appendix E separates forced-action prediction, Light-Switch recovery, four DM Control tasks, DCI scores, and bonus sensitivity by their metric and comparison. Lower forced-action MSE tests prediction under that protocol, not purity of either branch. The reported 11.7 V100-equivalent training hours and 872 dynamics forward passes/s describe different cost quantities; forward-pass throughput is not full MCTS throughput (Appendices B.3 and C.4).



Figure 5: Venture return, latent-hash coverage, and Agency Bonus over training.

5 Related Work

Model-based reinforcement learning. World models support planning or policy learning through imagined trajectories (Luo et al., 2022; Plaat et al., 2020; Hafner et al., 2020; Schrittwieser et al., 2020; Ye et al., 2021). Representation learning and sequence models provide complementary approaches (Schwarzer et al., 2021; Robine et al., 2023; Micheli et al., 2023; Zhang et al., 2023; Wang et al., 2025). CDWM focuses on a different object: the attribution of a predicted transition to an autonomous baseline and an action-dependent component. Accurate prediction does not uniquely determine this split; Section 2 states the anchor and coverage conditions needed to identify it.

Causal structure in reinforcement learning. Causal RL studies how assumptions about the data-generating process affect prediction, generalization, and explanation (Zeng et al., 2025; Deng et al., 2023; Kaddour et al., 2022). Related approaches learn representations using independent mechanisms or structural knowledge (Komanduri et al., 2024; 2022), or explain behavior through causal models (Yu et al., 2024; Madumal et al., 2020). CDWM places the autonomous/intervention distinction inside the planner's transition model, with an explicitly conditional identification argument.

Intrinsic motivation. Intrinsic rewards guide exploration when external rewards are sparse (Aubret et al., 2019; 2023). Prediction-error curiosity motivates separating controllable effects from unpredictable observations (Pathak et al., 2017; Burda et al., 2019); causal exploration addresses related questions (Nguyen et al., 2024; Yang et al., 2024). The Agency Bonus reuses the learned intervention pathway rather than a separate curiosity predictor. Its population noise argument follows from the anchored model; finite-data accuracy and task alignment remain empirical questions.

6 Conclusion and Limitations

CDWM exposes a background increment and an action-conditioned increment within a neural world model, reusing the latter for exploration. The reported results support useful structure and task-dependent pathway reliance, not full transparency: the encoder and both branches remain black boxes. Identification is conditional on the stated assumptions, which are not verified throughout Atari. Three seeded runs clarify replication but do not establish significance, exclude leakage, or isolate dynamics and bonus gains across the full suite. Run-level uncertainty, aggregate reconciliation, and the implemented anchor remain material to these claims (Appendix G).

AI USE STATEMENT

Generative AI tools assisted with language polishing, sentence refinement, and manuscript organization. The authors take full responsibility for the final manuscript.

REFERENCES

Karl J. Åström. Optimal control of markov processes with incomplete state information i. *Journal of Mathematical Analysis and Applications*, 10(1):174–205, 1965. doi: 10.1016/0022-247X(65)90154-X.

Arthur Aubret, Laetitia Matignon, and Salima Hassas. A survey on intrinsic motivation in reinforcement learning, 2019. URL https://arxiv.org/abs/1908.06976.

Arthur Aubret, Laetitia Matignon, and Salima Hassas. An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey. *Entropy*, 25(2):327, February 2023. ISSN 1099-4300. doi: 10.3390/e25020327. URL http://dx.doi.org/10.3390/e25020327.

Yuri Burda, Harri Edwards, Deepak Pathak, Amos Storkey, Trevor Darrell, and Alexei A. Efros. Large-scale study of curiosity-driven learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJNwDjAqYX.

Zhihong Deng, Jing Jiang, Guodong Long, and Chengqi Zhang. Causal reinforcement learning: A survey, 2023. URL https://arxiv.org/abs/2307.01452.

David Ha and Jürgen Schmidhuber. World models, 2018. URL https://zenodo.org/record/1207631.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination, 2020. URL https://arxiv.org/abs/1912.01603.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models, 2024. URL https://arxiv.org/abs/2301.04104.

Jean Kaddour, Aengus Lynch, Qi Liu, Matt J. Kusner, and Ricardo Silva. Causal machine learning: A survey and open problems, 2022. URL https://arxiv.org/abs/2206.15475.

Łukasz Kaiser, Mohammad Babaeizadeh, Piotr Miłos, Błażej Osiński, Roy H Campbell, Konrad Czechowski, Dumitru Erhan, Chelsea Finn, Piotr Kozakowski, Sergey Levine, Afroz Mohiuddin, Ryan Sepassi, George Tucker, and Henryk Michalewski. Model based reinforcement learning for atari. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1xCPJHtDB.

Aneesh Komanduri, Yongkai Wu, Wen Huang, Feng Chen, and Xintao Wu. Scm-vae: Learning identifiable causal representations via structural knowledge. In *2022 IEEE International Conference on Big Data (Big Data)*, pp. 1014–1023, 2022. doi: 10.1109/BigData55660.2022.10021114.

Aneesh Komanduri, Yongkai Wu, Feng Chen, and Xintao Wu. Learning causally disentangled representations via the principle of independent causal mechanisms, 2024. URL https://arxiv.org/abs/2306.01213.

Fan-Ming Luo, Tian Xu, Hang Lai, Xiong-Hui Chen, Weinan Zhang, and Yang Yu. A survey on model-based reinforcement learning, 2022. URL https://arxiv.org/abs/2206.09328.

Prashan Madumal, Tim Miller, Liz Sonenberg, and Frank Vetere. Explainable reinforcement learning through a causal lens. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34, pp. 2493–2500. AAAI Press, 2020. doi: 10.1609/aaai.v34i03.5631.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models, 2023. URL https://arxiv.org/abs/2209.00588.

Minh Hoang Nguyen, Hung Le, and Svetha Venkatesh. Variable-agnostic causal exploration for & reinforcement learning. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2024, Vilnius, Lithuania, September 9–13, 2024, Proceedings, Part II*, pp. 216–232, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-70343-0. doi: 10.1007/978-3-031-70344-7_13. URL https://doi.org/10.1007/978-3-031-70344-7_13.

Deepak Pathak, Pulkit Agrawal, Alexei A. Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction, 2017. URL https://arxiv.org/abs/1705.05363.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2009. ISBN 9780521895606.

Aske Plaat, Walter Kosters, and Mike Preuss. Deep model-based reinforcement learning for high-dimensional problems, a survey, 2020. URL https://arxiv.org/abs/2008.05598.

Jan Robine, Marc Höftmann, Tobias Uelwer, and Stefan Harmeling. Transformer-based world models are happy with 100k interactions. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=TdBaDGCpjly.

Julian Schrittwieser, Ioannis Antonoglou, Thomas Hubert, Karen Simonyan, Laurent Sifre, Simon Schmitt, Arthur Guez, Edward Lockhart, Demis Hassabis, Thore Graepel, Timothy Lillicrap, and David Silver. Mastering atari, go, chess and shogi by planning with a learned model. *Nature*, 588 (7839):604–609, December 2020. ISSN 1476-4687. doi: 10.1038/s41586-020-03051-4. URL http://dx.doi.org/10.1038/s41586-020-03051-4.

Max Schwarzer, Ankesh Anand, Rishab Goel, R Devon Hjelm, Aaron Courville, and Philip Bachman. Data-efficient reinforcement learning with self-predictive representations, 2021. URL https://arxiv.org/abs/2007.05929.

Max Schwarzer, Johan Obando-Ceron, Aaron Courville, Marc Bellemare, Rishabh Agarwal, and Pablo Samuel Castro. Bigger, better, faster: Human-level atari with human-level efficiency, 2023. URL https://arxiv.org/abs/2305.19452.

Shengjie Wang, Shaohuai Liu, Weirui Ye, Jiacheng You, and Yang Gao. Efficientzero v2: Mastering discrete and continuous control with limited data, 2024. URL https://arxiv.org/abs/2403.00564.

Wenlong Wang, Ivana Dusparic, Yucheng Shi, Ke Zhang, and Vinny Cahill. Drama: Mamba-enabled model-based reinforcement learning is sample and parameter efficient. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=7XIkRgYjK3.

Yupei Yang, Biwei Huang, Shikui Tu, and Lei Xu. Boosting efficiency in task-agnostic exploration through causal knowledge. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, IJCAI '24, 2024. ISBN 978-1-956792-04-1. doi: 10.24963/ijcai.2024/591. URL https://doi.org/10.24963/ijcai.2024/591.

Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=OKrNPg3xR3T.

Zhongwei Yu, Jingqing Ruan, and Dengpeng Xing. Explainable reinforcement learning via a causal world model, 2024. URL https://arxiv.org/abs/2305.02749.

Yan Zeng, Ruichu Cai, Fuchun Sun, Libo Huang, and Zhifeng Hao. A survey on causal reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 36(4):5942–5962, 2025. doi: 10.1109/TNNLS.2024.3403001.

Weipu Zhang, Gang Wang, Jian Sun, Yetian Yuan, and Gao Huang. Storm: Efficient stochastic transformer based world models for reinforcement learning, 2023. URL https://arxiv.org/abs/2310.09615.

SUPPLEMENTARY MATERIAL

The appendices retain the source manuscript's baseline descriptions, architectural settings, theoretical analysis, limitations, additional experiments, identification argument, and an evidence/reporting map. Appendix B also gives the detailed planning procedure referenced by Algorithm 1.

A BASELINE COMPARISONS

The Atari100k comparison retains all nine reference methods. SimPLe uses video prediction (Kaiser et al., 2020), and SPR uses self-predictive representations (Schwarzer et al., 2021). TWM, IRIS, and STORM provide Transformer-based comparisons (Robine et al., 2023; Micheli et al., 2023; Zhang et al., 2023); DreamerV3 and Mamba-based DramaXS provide further world-model references (Hafner et al., 2024; Wang et al., 2025). BBF and EZ-V2 are retained as distinct data-efficient control systems (Schwarzer et al., 2023; Wang et al., 2024), not variants of CDWM.

Table 1 reproduces the NeurIPS report rather than new baseline replications. Its missing seed-level statistics and normalization discrepancy limit the interpretation of aggregate rankings. Auxiliary prediction and timing comparisons require their own protocols: a return table does not establish that those operations were measured identically across systems.

B ARCHITECTURE, TRAINING, AND PLANNING DETAILS

B.1 MODEL ARCHITECTURE

The encoder maps four stacked 64×64 RGB observations to a 512-dimensional latent state. Tables 2–4 retain the reported convolutional encoder, recurrent environmental branch, feed-forward intervention branch, and policy/value/reward heads. The number of actions A is game-specific. These tables describe layer structure; they do not specify how null-action anchoring was imposed in the evaluated implementation.

Table 2: Image encoder: four RGB frames to a 512-dimensional latent state.

Layer	Output shape
Input frame stack	$12 \times 64 \times 64$
Conv2d ($k = 4, s = 2, p = 1$) + BatchNorm + ReLU	$32 \times 32 \times 32$
Conv2d ($k = 4, s = 2, p = 1$) + BatchNorm + ReLU	$64 \times 16 \times 16$
Conv2d ($k = 4, s = 2, p = 1$) + BatchNorm + ReLU	$128 \times 8 \times 8$
Conv2d ($k = 4, s = 2, p = 1$) + BatchNorm + ReLU	$256 \times 4 \times 4$
Flatten	4096
Linear + LayerNorm	512
Output latent state s_t	$\mathbb{R}^{512}$

Table 3: Dual-path dynamics, with latent dimension $D = 512$.

Module / operation	Output shape
Environment Pathway: input state	D
GRU cell (hidden dimension D)	D
Two-layer MLP (ReLU, hidden dimension D)	D
Intervention Pathway: $[s_t, \text{onehot}(a_t)]$	$D + A$
Two-layer gated MLP (GELU, hidden dimension D)	D
$\Delta\hat{s}_{t+1} = f_{\text{env}}(s_t) + f_{\text{agent}}(s_t, a_t)$	D
$\hat{s}_{t+1} = s_t + \Delta\hat{s}_{t+1}$	D

Table 4: Policy, value, and reward heads operating on the latent state.

Head	Architecture	Input / hidden / output
Policy π_ψ	Three-layer MLP, ReLU	$D/D/A$
Value V_ψ	Three-layer MLP, ReLU	$D/D/1$
Reward R_ϕ	Three-layer MLP, ReLU	$D/D/1$

B.2 Training Hyperparameters

Our experimental runs use seeds 0, 1, and 42. The reported configuration uses the same hyperparameters across the 26 Atari games (Table 5). The rollout horizon is five and each planning decision uses 50 simulations. The source lists separate world-model and actor/planner learning rates but does not give a parameter-group map for the encoder and all heads. It also leaves the precise loss targets, terminal handling, action-selection temperature, and null-anchor implementation underspecified. No additional settings are inferred here.

Table 5: Reported CDWM hyperparameters for Atari100k.

Group	Hyperparameter	Value
Training	Replay buffer capacity	100,000
	Batch size	128
	Unroll length / imagination horizon	5
	Optimizer	AdamW
	Optimizer coefficients	$\beta_1 = 0.9,\ \beta_2 = 0.999$
	Gradient clipping norm	1.0
World model	Latent dimension D	512
	Dynamics loss weight λ_d	2.0
	Learning rate	3×10^{-4}
Actor / planner	Discount γ	0.997
	MCTS simulations per step	50
	Value loss weight λ_v	0.25
	Reward loss weight λ_r	1.0
	Agency Bonus weight λ_{agency}	0.05
	Learning rate	3×10^{-5}

B.3 Training Cost

Table 6 retains approximate training costs in V100-equivalent hours. The report states that starred values were extrapolated from reported hardware specifications using DreamerV3 conversion assumptions. They should therefore be distinguished from direct measurements on a shared device. The source does not specify whether each entry is a per-game run, an aggregate over games, or how seed counts enter the accounting; the 11.7-hour CDWM entry is retained without assigning a new denominator.

Table 6: Approximate V100-equivalent training hours reported for Atari100k.

Method	V100 hours (approx.)
SimPLe	240*
SPR	10
TWM	20*
IRIS	168*
STORM	9.3*
DreamerV3	12
DramaXS	7*
BBF	13*
EZ-V2	16
CDWM	11.7*

B.4 MCTS WITH THE AGENCY BONUS

Algorithm 2 expands the planning call in Algorithm 1. It retains PUCT selection and root visit-count action choice, while writing the reward augmentation and backward return recursion explicitly. The intrinsic term is evaluated at the parent state and chosen action, before replacing that state with its predicted successor. This avoids attaching a previous action to the next state's intervention estimate. An implementation may cache the model outputs on existing edges; that choice does not alter the reward definition.

Algorithm 2 MCTS with intervention-based reward augmentation

Input: Root state s_{root}, model g_ϕ, heads f_ψ, simulations M, γ, $\lambda_{\mathrm{agency}}$
Output: Root action $a^\star$

Initialize tree $\mathcal{T}$; evaluate $f_\psi(s_{\mathrm{root}})$ and expand root
for $m = 1, \dots, M$ **do**
  $s \leftarrow s_{\mathrm{root}}$; $\mathcal{P} \leftarrow [\,]$
  while s is expanded in $\mathcal{T}$ **do**
    $a \leftarrow \arg\max_b [Q(s,b) + U(s,b)]$ ▷ PUCT selection
    $(\hat{s}', \hat{r}) \leftarrow g_\phi(s,a)$ ▷ Transition (5)
    $b \leftarrow r^{\mathrm{i}}(s,a)$ ▷ Agency Bonus (6)
    Append $(s, a, \hat{r} + \lambda_{\mathrm{agency}} b)$ to $\mathcal{P}$
    $s \leftarrow \hat{s}'$ ▷ Carry the corresponding recurrent context
  end while
  $(\boldsymbol{p}, V) \leftarrow f_\psi(s)$; expand leaf with priors $\boldsymbol{p}$
  $G \leftarrow V$
  for $(s, a, \bar{r})$ in reverse order of $\mathcal{P}$ **do**
    $G \leftarrow \bar{r} + \gamma G$ ▷ Backup (7)
    $N(s,a) \leftarrow N(s,a) + 1$; $W(s,a) \leftarrow W(s,a) + G$
    $Q(s,a) \leftarrow W(s,a)/N(s,a)$
  end for
end for
return $\arg\max_a N(s_{\mathrm{root}}, a)$

The procedure is the nonterminal planning core specified by the source model. The exact terminal-state convention and PUCT constants must be supplied by the implementation. Likewise, the state-similarity rule and update schedule for μ_ϕ, σ_ϕ, and the stabilizer δ, are necessary to reproduce the bonus but are not fixed by the numerical hyperparameter table.

C STRUCTURAL ANALYSIS AND COMPUTATIONAL TRADE-OFFS

C.1 PREDICTION VERSUS MECHANISM ATTRIBUTION

Table 7 organizes the model families considered in the source discussion by their prediction form and interpretation requirements. It should not be read as a universal ranking of causal capability. Under the state-sufficient graph in Appendix F.2, an unrestricted predictor can estimate the same

conditional transition mean as CDWM. What CDWM adds is a named, anchored allocation of that mean to two components.

Table 7: Model representations and the conditions needed for action-effect interpretation.

Model family	Representation / query	Interpretation requirement
Monolithic	$s' \sim g_\theta(s, a)$; query a candidate action	State sufficiency and action coverage; no explicit component split
Video prediction	$o' \sim g_\theta(o, a)$; predict observations	Observational accuracy alone does not assign change to distinct mechanisms
Linear–Gaussian	$s' = As + Ba$	Structural meaning depends on the causal and data assumptions
CDWM	$s' = s + f_{\text{env}}(s) + f_{\text{agent}}(s, a)$	Null-action anchor and the conditions of Proposition 1

The split is useful only when its interpretation is justified. Current-action exclusion specifies an input restriction. Equation (3) shows why it is not, by itself, an identification argument. Null-action normalization removes the additive ambiguity; action coverage permits the baseline and alternatives to be compared. Neither assumption is implied by a high return or by separation in a two-dimensional embedding. The formal result therefore concerns supported state–action pairs and the risk optimum, rather than every unobserved intervention a planner could query.

C.2 Agency Bonus under Independent Transition Noise

Proposition 2 formalizes the population argument used for the bonus in Section 3.2. Its premise is Proposition 1, including the null-action anchor; it is not a statement about every finite-data fit of two branches.

Proof of Proposition 2. Consider noise laws indexed by ν, each satisfying the assumptions and $\mathbb{E}[\epsilon^{(\nu)} \mid s, a] = 0$, with the same structural functions. On shared supported state–action pairs,

$$\Delta_\nu(s, a) = f^\star_{\text{env}}(s) + f^\star_{\text{agent}}(s, a), \qquad \Delta_\nu(s, a) - \Delta_\nu(s, a_0) = f^\star_{\text{agent}}(s, a). \tag{10}$$

The noise law therefore changes neither conditional mean nor their anchored contrast. Proposition 1 identifies that same contrast under each law. Its norm is unchanged, and applying the same normalization and sigmoid preserves the equality of the bonus. In contrast, at the true mean predictor $g^*(s, a) = s + \Delta_\nu(s, a)$,

$$\mathbb{E}_\nu[\|s' - g^*(s, a)\|_2^2 \mid s, a] = \mathbb{E}_\nu[\|\epsilon^{(\nu)}\|_2^2 \mid s, a] = \operatorname{tr} \operatorname{Cov}_\nu(\epsilon^{(\nu)} \mid s, a). \tag{11}$$

The prediction residual retains noise covariance even when the action contrast does not. This proves the stated distinction; it does not assert identical state visitation or identical fitted normalization statistics across noisy environments. □

Scalar illustration. Consider the scalar example from the source analysis, $s_{t+1} = s_t + \mu(s_t) + a_t + \xi_t$, where $\xi_t \sim \mathcal{N}(0, \sigma^2)$ is independent of the action and state. The optimal conditional-mean predictor is $\hat{s}_{t+1} = s_t + \mu(s_t) + a_t$. Its irreducible squared error is

$$\mathbb{E}[(s_{t+1} - \hat{s}_{t+1})^2 \mid s_t, a_t] = \sigma^2, \qquad f_{\text{agent}}(s_t, a_t) = a_t \tag{12}$$

under an anchored decomposition with $a_0 = 0$. The prediction error increases with noise variance, while the population intervention magnitude $|a_t|$ does not. Table 8 summarizes this distinction and its scope in sparse-reward settings.

The conclusion is about a population component, not the training variance of its neural estimate. Finite data, a misspecified state, or a changing encoder can alter the estimated magnitude. The normalized sigmoid bonus in Equation (6) is not necessarily zero when the magnitude is zero, and controllable changes need not be novel or useful. For the same reason, the intervention norm is not established here as a lower bound on empowerment; that interpretation requires an additional result not supplied by the source.

Table 8: Exploration signals under noise and sparse reward: scope of the comparison.

Setting	Prediction error	Novelty / entropy	Agency Bonus
Independent transition noise	Irreducible error can stay high	Response depends on what is encoded and counted	Population intervention magnitude is unchanged under the stated assumptions
Sparse external reward	Can reward prediction difficulty	Can reward state diversity	Rewards modeled action effects, not necessarily task progress

C.3 Structural Restrictions and Expressiveness

The two neural pathways can be nonlinear even though their outputs are added. Consequently, a mode switch does not establish a universal inability to fit a transition mean. The more specific concern is attribution when a relevant mode is absent from the state or from the recurrent context used by the environmental branch. In that case, an intervention residual can absorb an unmodeled change of regime. Appendix D.2 makes this conditional failure case explicit; the Light-Switch results in Appendix E.2 concern recovery in one evaluated domain, not a general identifiability guarantee.

C.4 Forward-Pass Throughput

The source measures batch-one dynamics-model forward passes on a single V100, with latent dimension $D = 512$. Table 9 retains its parameter counts, throughput entries, and relative speeds. CDWM is reported at 872 ± 18 passes per second, compared with 945 ± 15 for DreamerV3 and 890 ± 22 for the listed BBF configuration. The throughput ratio to DreamerV3 is approximately 0.92, consistent with the source's 7.7% reduction.

Table 9: Reported batch-one forward-pass throughput on a V100 GPU.

Method	Architecture	Params (M)	Passes/s	Relative
DreamerV3	RSSM	18.5	945 ± 15	$1.00\times$
BBF	ResNet	21.2	890 ± 22	$0.94\times$
CDWM	GRU + MLP	19.8	872 ± 18	$0.92\times$

These values are not complete MCTS decisions per second: one decision can require many model evaluations, policy/value calls, and tree operations. The source does not define the repeated-run protocol or the uncertainty represented by $\pm$, and it does not fully specify comparable operations for every baseline. The smaller feed-forward branch may permit efficient implementation, but the table does not measure the benefit of parallelizing the two pathways or establish real-time control performance.

D Limitations and Mode-Switching Analysis

D.1 Scope of the Additive Interpretation

CDWM assigns autonomous evolution to f_{env} and the difference induced by an action to f_{agent}. The null-action anchor gives these components a reference point, but does not make the representation state-sufficient. When an action changes a persistent environmental regime, a model that omits that regime can produce a misleading decomposition even if it predicts some observed transitions well. The limitation concerns the state and the intended semantics of the branches, not simply the presence of nonlinear neural functions.

D.2 A Conditional Mode-Switching Failure Case

The source's formal example augments the environment with a mode m_t that can be changed by an action:

$$m_{t+1} = \mathcal{T}(m_t, a_t), \qquad s_{t+1} = s_t + f^\star_{\text{env}}(s_t, m_{t+1}) + f^\star_{\text{agent}}(s_t, a_t) + \epsilon_t. \tag{13}$$

A switching action a_{switch} changes m_t to m'_t; a non-switching action leaves it unchanged. Suppose the learned state does not resolve this mode and the environmental predictor approximates the mixture

$$f_{\text{env}}(s_t) \approx \mathbb{E}_{m\sim P(m|s_t)}[f^\star_{\text{env}}(s_t, m)]. \tag{14}$$

This is the mode-averaging condition used in the example, not a consequence of current-action exclusion alone. A recurrent environmental model that successfully tracks m_t need not satisfy this approximation.

Attribution at the switch. Prediction training compares the total model update with the observed successor:

$$\mathcal{L}_{\text{dyn}} = \mathbb{E}\|s^{\text{true}}_{t+1} - s_t - f_{\text{env}}(s_t) - f_{\text{agent}}(s_t, a_t)\|^2_2. \tag{15}$$

At a switching action, fitting the mean transition then requires the intervention branch to account for both the source's direct-action term and the discrepancy between the new regime and the environmental baseline:

$$f_{\text{agent}}(s_t, a_{\text{switch}}) \approx f^\star_{\text{agent}}(s_t, a_{\text{switch}}) + \ell(s_t, m'_t), \tag{16}$$

where $\ell(s, m') = f^\star_{\text{env}}(s, m') - \mathbb{E}_{m\sim P(m|s)} f^\star_{\text{env}}(s, m)$ is the regime-mismatch term. This changes the interpretation of the learned branch relative to the designated *direct* term in Equation (13). It is not, on its own, proof that a predicted total action effect is incorrect: a mode change caused by the action can be part of that total effect.

Prediction after the switch. After a mode change, consider a non-switching null action. The model predicts $\hat{s}^{\text{cf}}_{t+1} = s_t + f_{\text{env}}(s_t) + f_{\text{agent}}(s_t, a_0)$, whereas the conditional mean in the current mode is $s_t + f^\star_{\text{env}}(s_t, m_t) + f^\star_{\text{agent}}(s_t, a_0)$. If the null-action contribution is correctly anchored, their difference is

$$\text{Error}_{\text{cf}} = f_{\text{env}}(s_t) - f^\star_{\text{env}}(s_t, m_t) \approx \mathbb{E}_m[f^\star_{\text{env}}(s_t, m)] - f^\star_{\text{env}}(s_t, m_t). \tag{17}$$

The error persists when the current regime differs from the mixture that the state represents. Thus mode tracking is material to the validity of a null-action prediction. A quick empirical recovery does not prove exact causal attribution during the switch.

D.3 Exploration and Evaluation Limits

The Agency Bonus measures a norm in the learned latent coordinates. A visually small change can have important future consequences, while a large controllable change may have little task value. Excessive weighting can therefore distract from the external objective, as the Venture sensitivity table illustrates. Neither a general novelty guarantee nor an empowerment bound follows from the norm alone.

The current evidence also leaves aggregate normalization, run-level uncertainty, and several implementation choices unresolved. The use of three run seeds (0, 1, and 42) does not by itself define an error bar or justify a significance claim. The main benchmark and the supplementary continuous-control results should not be merged into a claim about arbitrary environments or real-world deployment. In particular, the identification argument applies to a fixed state representation with coverage and a specified anchor, whereas the practical system jointly learns the encoder and branches.

D.4 Future Directions

The source proposes richer interactions in which an intervention pathway modulates the environmental dynamics, for example through generated weights or affine feature modulation. Such extensions would require their own identification conditions and experimental validation. Further directions include partially observable 3D domains and multi-agent settings, together with direct tests of null-action anchoring, action coverage, and the reliability of intervention predictions under policy changes.

E Additional Experiments

The following studies report different measurements: one-step prediction error, recovery after a regime change, task return, representation metrics, and the response to bonus weight. Our experiments use seeds 0, 1, and 42, as in Section 4.1; published comparison entries do not inherit that protocol automatically. The descriptions below separate what each study tests from the details still needed to establish comparability. They are not additional Atari100k aggregate scores. Appendix G gives a common evidence map.

E.1 Prediction under Forced Actions

The prediction study compares one-step mean squared error on transitions sampled from the behavior policy with error under forced actions, including rare or counter-intuitive choices. Table 10 reports results for Breakout and Seaquest.

Table 10: One-step prediction MSE under behavior-policy and forced-action evaluation.

	Breakout		Seaquest	
Method	Observational	Forced action	Observational	Forced action
DreamerV3	0.042	0.089	0.051	0.112
BBF	0.038	0.076	0.048	0.095
CDWM	**0.035**	**0.041**	**0.042**	**0.055**
Reported gap[†]	+17%	+54%	+21%	+51%

For CDWM, the Breakout error changes from 0.035 to 0.041 and the Seaquest error from 0.042 to 0.055. The corresponding DreamerV3 entries are 0.042 to 0.089 and 0.051 to 0.112. These measurements support a smaller degradation in the reported forced-action evaluation. They do not alone establish that the latent pathways recover unique causal factors. For a like-for-like test, predictions must be compared on the same starting states, forced-action distribution, horizon, and target coordinates. These settings are not fully specified in the report. In particular, MSE values in separately learned latent spaces need not have a common scale. A shared target space and the prediction head used for every baseline must be documented; the presence of a BBF entry does not itself specify such a head. The daggered "gap" row is reproduced from the source; its denominator is not defined consistently enough to use it as an additional quantitative conclusion.

E.2 Mode Switching

The Light-Switch toy domain changes the environmental regime through an action. Table 11 reports MSE before the switch, at the switching step, and five steps later, together with a recovery time. CDWM has a transient error increase but lower error than the monolithic comparison at the reported checkpoints, with recovery listed as approximately three rather than fifteen steps.

Table 11: Prediction MSE and recovery time in the reported Light-Switch domain.

Method	Stable phase	Switch step	Five steps later	Recovery
Monolithic	0.021	0.354	0.128	~ 15 steps
CDWM	0.018	0.112	0.034	~ 3 steps

This result is compatible with recovery of recurrent context, but the table does not separate state inference from parameter updates. The recovery threshold and whether learning continues during evaluation are not given. It therefore does not establish that the architecture satisfies the identifying assumptions during a switch, nor that all nonlinear mode changes are harmless.

E.3 Continuous Control

Table 12 retains the DM Control 100k results on Cheetah Run, Walker Walk, Cartpole Swing, and Finger Spin. CDWM has the largest reported mean return on three of the four tasks, while Effi-

cientZero has the largest on Cheetah Run. The reported aggregate is 756 for CDWM and 750 for EfficientZero.

Table 12: Reported returns on four DM Control 100k tasks.

Method	Cheetah Run	Walker Walk	Cartpole Swing	Finger Spin	Mean
DreamerV3	520 ± 45	610 ± 55	780 ± 20	650 ± 40	640
EfficientZero	680 ± 30	720 ± 40	810 ± 15	790 ± 35	750
CDWM	645 ± 38	745 ± 42	825 ± 18	810 ± 30	756

These observations extend the reported evaluation beyond discrete Atari actions. Our run seeds are 0, 1, and 42. The meaning of the existing $\pm$ terms and the per-seed observations are not supplied, and neither the continuous-action planning procedure nor its null-anchor implementation is fully specified. Observation modality, action repeat, evaluation episodes, and baseline training budgets also need a common protocol. We retain the values without interpreting the six-point reported aggregate margin as statistically significant or extending the discrete-action identification theorem to this setting.

E.4 Representation Metrics

The DCI analysis uses Atari RAM-derived factors, such as agent and enemy positions, to evaluate representation disentanglement, completeness, and informativeness. Table 13 reports CDWM scores of 0.78, 0.74, and 0.85, respectively.

Table 13: Reported disentanglement, completeness, and informativeness scores.

Method	Disentanglement	Completeness	Informativeness
SimPLe	0.32	0.45	0.68
DreamerV3	0.41	0.52	0.81
CDWM	0.78	0.74	0.85

DCI asks whether factors can be predicted from the representation and how that predictive information is distributed. The scores complement Figure 4, but do not determine which pathway contains which factor unless reported separately by branch. A high combined score is compatible with the same factor appearing in both branches. It therefore does not verify Proposition 1 or exclude leakage. The games, RAM factor definitions, branch inputs to the metric, train/test split, probe class, and per-seed scores remain necessary for a reproducible comparison.

E.5 Agency Bonus Weight

Table 14 varies λ_{agency} on Venture. The reported mean return rises from 185 without the bonus to 1420 at 0.05, then falls to 620 at 0.50. Exploration coverage continues to increase from 12% to 85% across that range.

Table 14: Venture return and exploration coverage across Agency Bonus weights.

λ_{agency}	Venture return (mean)	Exploration coverage
0.0 (no bonus)	185	12%
0.01	850	45%
0.05	1420	78%
0.10	1105	82%
0.50	620	85%

The weight 0.05 is the best among the five reported choices, not an established global optimum. The divergence between coverage and return shows why controllable change is not sufficient evidence of task progress. It also does not establish that excessive weighting causes attraction to *uncontrollable*

noise; the observed distraction is distinct from the independent-noise calculation in Appendix C.2. An empowerment interpretation remains a hypothesis rather than a proved lower bound.

F IDENTIFICATION UNDER A NULL-ACTION ANCHOR

This appendix proves Proposition 1 using the conditions stated in Section 2. The proof concerns the population risk corresponding to the dynamics loss in Equation (8), not convergence of the complete joint objective in Equation (9). The environmental mean, intervention contrast, reference constraint, and noise residual use the same notation as the main text.

F.1 ASSUMPTIONS AND THE IDENTIFIED OBJECT

Fix the latent representation and let D be its population state–action transition law. Assumption 1 gives $\Delta s = f^\star_{\text{env}}(s) + f^\star_{\text{agent}}(s,a) + \epsilon$ with $\mathbb{E}[\epsilon \mid s,a] = 0$ and finite second moments. The causal interpretation also requires the state-sufficient, exogenous-noise graph described below. The true reference satisfies $f^\star_{\text{agent}}(s,a_0) = 0$: the world may continue to evolve under a_0, but its designated intervention contribution is zero.

The compared action set is discrete. For D_s-almost every state, each compared action, including a_0, has positive behavior-policy probability. The function classes contain the true anchored pair and population squared risk is minimized globally. The learned environment branch has no current-action input; the learned intervention branch satisfies $f_{\text{agent}}(s,a_0) = 0$. These are respectively the input and output restrictions discussed in Section 3.1. A genuine reference and its coverage are assumptions, not consequences of calling an action `NOOP`.

The conclusion identifies functions almost surely on the supported distribution. It identifies neither neural parameters nor latent coordinates and does not prove that a learned encoder is sufficient. In continuous action spaces, positive density alone does not identify the value on a probability-zero reference slice; an extension needs additional regularity or reference-action data. The discrete statement is not automatically a theorem about the supplementary continuous-control experiments.

F.2 PROOF OF ANCHORED IDENTIFICATION

Proof of Proposition 1. **Prediction fixes the sum.** Assumption 1 implies

$$\Delta(s,a) := \mathbb{E}[\Delta s \mid s,a] = f^\star_{\text{env}}(s) + f^\star_{\text{agent}}(s,a). \tag{18}$$

Let $u(s,a) = f_{\text{env}}(s) + f_{\text{agent}}(s,a)$ and let $\mathcal{R}_{\text{dyn}} = \mathbb{E}_D[\mathcal{L}_{\text{dynamics}}(s,a,s')]$ be the population counterpart of Equation (8). Expanding the square gives

$$\mathcal{R}_{\text{dyn}} = \mathbb{E}_D\|u(s,a) - \Delta(s,a)\|_2^2 + \mathbb{E}_D\|\epsilon\|_2^2. \tag{19}$$

The cross term vanishes because $\mathbb{E}[\langle \epsilon, \Delta - u\rangle \mid s,a] = \langle \mathbb{E}[\epsilon \mid s,a], \Delta - u\rangle = 0$. The realizable true pair attains the second term. Thus every global minimizer has $u(s,a) = \Delta(s,a)$ almost surely; prediction risk fixes the sum but not yet its attribution.

Coverage makes the reference slice available. For each compared discrete action a, write $p_a(s) = \Pr(A = a \mid S = s) > 0$ almost surely. Zero prediction discrepancy implies $\mathbb{E}_s[p_a(s)\|u(s,a) - \Delta(s,a)\|_2^2] = 0$, so the discrepancy is zero D_s-almost surely for that action. In particular, the reference slice a_0 cannot be ignored by a population minimizer. Evaluating it and using the learned anchor yields

$$f_{\text{env}}(s) = \Delta(s,a_0) = f^\star_{\text{env}}(s), \qquad f_{\text{agent}}(s,a) = \Delta(s,a) - \Delta(s,a_0) = f^\star_{\text{agent}}(s,a). \tag{20}$$

The first equality fixes the autonomous baseline; subtracting it from the fitted sum fixes the intervention. This is exactly Equation (4).

The same anchor excludes every alternative shift. Any second fitted decomposition differs by $d(s) = f'_{\text{env}}(s) - f_{\text{env}}(s)$ and has the opposite shift in its intervention branch, as in Equation (3). Both intervention functions vanish at a_0, so $d(s) = 0$ almost surely. Their component functions coincide, proving uniqueness on the supported law. □

Why the identified difference is an intervention contrast. The structural graph has $s \to a$, $s \to s'$, and $a \to s'$, with no remaining common cause of a and s' after conditioning on s. With consistency and the exogenous-noise condition, conditioning blocks $a \leftarrow s \to s'$, yielding the adjustment identity (Pearl, 2009)

$$P(s' \mid s, \mathrm{do}(a)) = P(s' \mid s, a). \tag{21}$$

Taking means at a and a_0 shows that the identified component is the difference between their mean successors. Equation (5) implements this difference in the learned model. The graph assumption licenses the causal reading; the anchor makes the split unique. Neither claim follows from current-action exclusion alone. This is a statement about conditional-mean interventions, not reconstruction of an individual trajectory's unobserved outcome.

F.3 The Penalty Alternative

The source also proposes selecting the normalized split with a null-action penalty:

$$\mathcal{R}_\lambda = \mathcal{R}_{\mathrm{dyn}} + \lambda \mathbb{E}_s \| f_{\mathrm{agent}}(s, a_0) \|_2^2, \qquad \lambda \to 0^+. \tag{22}$$

Within the realizable class of predictive minimizers, the shift in Equation (3) incurs $\lambda \mathbb{E}_s \|d(s)\|_2^2$. If the anchored representative is feasible and the penalty is evaluated on the relevant state support, its zero penalty selects $d = 0$. This is the normalization argument used by the source. It does not supply a convergence theorem for a sequence of neural optimizers or a finite empirical penalty setting. The coefficient, anchor-state sampling, and optimization schedule remain unspecified in the supplied implementation description.

Section 3.3 therefore keeps Equation (9) as the reported four-term objective and identifies the penalty as a separate constraint. Algorithm 1 names the required identifying constraint without inventing its finite training coefficient. Applying Proposition 1 to the reported model still requires confirmation of the implemented anchor.

F.4 Finite-Data Scope

The source additionally records the proposed finite-sample form

$$\| \hat{f}_{\mathrm{env}} - f^\star_{\mathrm{env}} \|_{L_2(\mathcal{D}_s)} = O\left(\mathfrak{R}_n(\mathcal{F}_{\mathrm{env}}) + \sqrt{\frac{\log(1/\delta)}{n_{a_0}}} \right), \tag{23}$$

where $\mathfrak{R}_n$ is a Rademacher complexity and n_{a_0} represents effective null-action coverage, described in the source as scaling with $n \inf_s \pi(a_0 \mid s)$. Analogous coverage factors are suggested for the intervention component. This unverified expression is retained for author review, not used as a guarantee in either proposition.

The missing steps are material: the source does not specify boundedness, sampling dependence, function-class conditions, or the conversion from excess squared risk to a function-norm rate. Equation (19) controls the squared error of the *sum*; control of the separate branches also needs reference coverage. Replay dependence and a changing encoder require further analysis. The proved claim remains conditional population identification; finite-data recovery and optimization are distinct questions.

G Evidence Scope and Reporting Details

The empirical claims concern usefulness and observable dependence, while the identification propositions concern population functions under explicit assumptions. Table 15 connects each reported measurement to the claim it can support. The distinctions are important because a useful predictor, an inspectable decomposition, and an identified causal mechanism are different objects.

Table 15: What each study measures and what it does not establish.

Study	Measured object	Boundary of the inference
Atari100k	Complete-agent game returns	No isolated dynamics/bonus effect or significance from point scores alone
Branch removal	Change in a trained agent's return	Functional reliance, not retrained architectural advantage or branch purity
Venture bonus test	Return and latent-hash coverage with/without bonus	Task-specific effect, not a 26-game component attribution
t-SNE visualization	Geometry of projected pathway outputs	Neither game-playing performance nor pathway-labelled causal factors
Forced actions	One-step prediction MSE under a changed action protocol	Requires matched states and targets; no separate branch-effect test
DCI	Factor predictability and distribution across latent coordinates	No proof of identification or absence of duplicated information

G.1 Replication and Aggregation

Our experimental results are based on three seeded runs with IDs 0, 1, and 42. This identifies the run protocol, but not which sources of randomness were varied. Between-training variability requires independently trained checkpoints; three evaluations of a single checkpoint instead measure evaluation variability. The run records must distinguish these cases. The seed declaration applies to our experiments and does not establish a common training protocol for numbers adopted from prior work.

For a metric $y_{g,k}$ on game g and run k, the arithmetic summary would be $\bar{y}_g = (y_{g,0}+y_{g,1}+y_{g,42})/3$, with sample standard deviation $s_g = [\sum_{k\in\{0,1,42\}}(y_{g,k} - \bar{y}_g)^2/2]^{1/2}$. A standard error would instead be $s_g/\sqrt{3}$; neither quantity is a confidence interval by definition. The current game table does not provide the three observations needed to compute either. Existing uncertainty marks in the supplementary tables and shaded plots are retained without assigning them a new statistical interpretation. Three repeats do not on their own establish that a small gain is statistically significant.

Aggregation also needs a declared order. For fixed random and human references, human normalization is affine in a game's return, so averaging over seeds commutes with normalization. The cross-game arithmetic mean likewise commutes with seed averaging when the same complete game set is used. Consequently, three-seed averaging alone cannot reconcile the main table's inconsistent mean. Medians do not generally commute with seed averaging; their aggregation order must be stated separately. Seed-level returns, evaluation episodes, checkpoint selection, normalization references, and any missing-run handling are therefore needed to reconcile Table 1.

G.2 Reading the Figures

Figure 2 illustrates computation: an encoded state feeds the background pathway and the action pathway; their outputs form the predicted increment, and the latter also supplies the intrinsic signal. The game screenshots and narrative callout are illustrative. They are not a measurement showing that the indicated action is optimal or that a specific learned coordinate is a physical cause.

Figure 3 should be read within each game. The central bar is the full model, with one branch removed on either side. Panel colors denote task groups rather than a probability of control or a branch identity. The experiment deactivates a pathway in a converged model, rather than retraining a smaller architecture. The precise handling of the bonus after intervention-branch removal needs to

be recorded: recomputing it from the modified branch can change both the transition model and the intrinsic reward.

In Figure 4, an underlying sample is a branch-output vector for a state–action pair; the plotted objects are density contours of the reported two-dimensional projection. The supplied palette does not distinguish branch labels. Thus even the two visible late-stage modes cannot be assigned to the two pathways from this plot alone. The exact training checkpoints, sampled games, number of states, preprocessing, projection settings, density smoothing, and seed aggregation are not specified. Cross-checkpoint geometric comparison also requires a documented projection procedure. No return-versus-separation analysis is reported, so the plot is not used to explain game performance.

The Venture figure reports three different quantities: return, a latent-hash coverage proxy, and bonus magnitude. The no-bonus comparator includes a constant reward offset, so the documented comparison is not necessarily identical to setting the intrinsic contribution to zero without any other reward change. The offset rule, hash definition, time-axis discrepancy, and shaded-band statistic must be resolved from the run and plotting records.

G.3 Structural Interpretation versus Leakage

Excluding a_t from f_{env} guarantees that its output does not change when only the candidate action is changed at a fixed state and recurrent context. This is an architectural fact, not evidence that the branch contains no information correlated with behavior. State history can contain previous actions and predictors of the current policy. Conversely, the intervention branch receives the state and can encode background information as well. Clustering or decoding action labels without conditioning on this shared state would not by itself test unwanted causal leakage.

A more direct check of the intended attribution would compare branch predictions with repeated environment transitions from matched starting states under a reference action and alternative actions. The two targets are the reference mean increment for f_{env} and the mean action-minus-reference contrast for f_{agent}, as in Equation (4). Such a test also needs an adequate reference, a common coordinate system, and a defined treatment of transition noise. The reported forced-action table measures total prediction error, not these separate branch targets. No branch-level leakage measurement is claimed in this manuscript.