TANGIBLE AND EDITABLE CAUSAL ACTION-TO-OBJECT PATHWAYS IN ROBOT WORLD MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Robot world models can predict action outcomes yet still learn questionable action-to-object dependencies, making their predictions difficult to trust and their failures difficult to diagnose or correct. Existing black-box transitions obscure how the current action causally contributes to an object's predicted change within the learned model, while post-hoc attribution does not expose a separate causal pathway that can be directly revised. We address this limitation by making the model-internal causal contribution of the current action an explicit, auditable, and editable component of the world model. This enables potentially unreasonable learned action effects to be examined against independent simulator interventions and locally modified without retraining the entire dynamics network. To achieve this, we introduce a state-conditioned causal action-to-object pathway that participates directly in the learned transition and separates the current action's contribution from the remaining object dynamics. The pathway is learned jointly with the world model without direct action-effect or counterfactual supervision. We further develop a minimum-change causal mechanism edit that assigns the same executed action a prescribed alternative contribution to the predicted object transition while keeping the current state and action fixed. Experiments on Meta-World manipulation tasks show that the exposed pathway faithfully reflects the model's forward computation, aligns with independently measured action influence in the simulator, and supports precise targeted editing. Separate closed-loop interventions further demonstrate that modifying the learned pathway changes robot behavior and can yield exploratory success gains in selected tasks without retraining. These results show that world models can be designed not only to predict the consequences of actions, but also to expose, audit, and locally revise the internal causal mechanisms through which those consequences are represented, providing a foundation for assessing the reliability of opaque world models and exploring targeted interventions in failing tasks.

1 INTRODUCTION

World models (Ha & Schmidhuber, 2018) enable robots to anticipate the consequences of their actions and plan through learned dynamics. Latent-space planning (Hafner et al., 2019) makes such reasoning possible in compact learned representations, while scalable model-based control (Hansen et al., 2024) demonstrates its effectiveness across diverse continuous-control tasks. Learning from imagined trajectories (Hafner et al., 2025) further illustrates the potential of predictive world models. However, predictive accuracy alone cannot establish whether a world model attributes an object's predicted change to a reasonable action-dependent mechanism. When this dependence is entangled with other dynamics, questionable model-internal causal attributions can remain hidden, making predictions harder to scrutinize and planning failures harder to diagnose.

This opacity becomes particularly consequential when planning fails. A world model may attribute an object's predicted motion to an action coordinate with little corresponding influence in the environment. Even when short-term predictions appear reasonable, an incorrect action dependency may mislead the planner when it evaluates alternative actions. This reflects the broader distinction between predictive associations and causal mechanisms (Schölkopf et al., 2021). Without access to the learned action-to-object mechanism, it is difficult to distinguish an unreasonable predicted action effect from failures elsewhere in decision making. Prediction accuracy alone therefore provides

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.

an incomplete basis for assessing the reliability of learned dynamics. Interventional representation learning further motivates examining learned dependencies through controlled interventions (Ahuja et al., 2023). Exposing action-to-object dependencies enables independent auditing against simulator interventions and makes potentially implausible action effects accessible for targeted revision. This provides a concrete route from diagnosing opaque world-model predictions to testing local corrections on frozen models, without retraining the entire dynamics network.

Existing approaches address different aspects of this problem. Structured world models (Kipf et al., 2020) represent environment entities separately, while object-oriented dynamics (Zhao et al., 2022) organize their interactions. Action–object binding (Biza et al., 2022) makes the association between actions and objects more explicit. Post-hoc attribution (Sundararajan et al., 2017) measures the sensitivity of predictions to individual inputs, and perturbation-based visualization (Greydanus et al., 2018) examines the responses of learned agents to controlled changes. Model editing offers complementary capabilities: MEND (Mitchell et al., 2022) learns targeted parameter updates, ROME (Meng et al., 2022) modifies internal computations associated with factual associations, and MEMIT (Meng et al., 2023) extends factual editing to larger sets of associations. However, identifying an influential action coordinate does not itself expose a separately editable transformation through which that action contributes to an object's predicted transition. Similarly, altering a prediction does not necessarily revise the action-dependent mechanism that produced it. Mechanism-level diagnosis requires a pathway that participates in prediction, supports independent auditing, and can itself be directly modified.

We address this gap by making the model's local causal action-to-object pathway explicit within its primary-object transition. Our state-conditioned operator separates the current action's contribution from the remaining object dynamics and is learned jointly with the world model, without direct action-effect labels or counterfactual supervision. Unlike post-hoc attribution, this pathway participates in ordinary forward prediction, so its coordinate-wise influence can be inspected and audited against independent same-state simulator interventions. This makes otherwise opaque learned action dependencies testable and establishes an explicit intervention target for examining potentially unreasonable predicted action effects.

Importantly, the exposed operator is also a direct target for post-training intervention. At a fixed physical context and executed action, we construct a minimum-change operator update that assigns the same action a prescribed alternative contribution to the model's predicted object transition. We derive its unique minimum-Frobenius-norm solution and show that action-space directions orthogonal to the anchor action are preserved. Unlike retraining the world model, this intervention modifies the effective action-to-object mapping at the evaluated context without updating the model's global parameters. A questionable predicted action effect can consequently be examined and locally revised while the executed action and other transition branches remain unchanged. For failure recovery, this isolates a specific learned dependency as a testable intervention target rather than modifying the entire dynamics network. Candidate corrections can then be evaluated with frozen model parameters, connecting mechanism-level diagnosis to task-specific behavioral improvement without retraining.

We evaluate the approach on ten independently trained Meta-World (Yu et al., 2020) manipulation tasks. The exposed pathway matches the model's action-dependent forward computation to numerical precision and shows alignment with independently measured simulator effects. A targeted-editing audit verifies precise alternative contribution assignment at fixed states and actions. Closed-loop interventions further demonstrate that modifying the pathway affects robot behavior and can yield exploratory success gains in selected tasks without retraining. Together, these experiments establish the pathway's internal fidelity, external auditability, and behavioral relevance, while distinguishing precise mechanism editing from the separate problem of selecting reliably beneficial edits.

Our contributions are threefold. First, we introduce an explicit action-to-object operator within learned world-model dynamics, exposing an otherwise implicit action dependence for direct inspection and independent auditing. Second, we derive a minimum-Frobenius-norm mechanism edit that assigns a fixed action a prescribed alternative predicted contribution while preserving orthogonal action-space directions. Third, we demonstrate that the same pathway supports forward prediction, external auditing, and precise post-training intervention, providing a concrete foundation for scrutinizing opaque world models and exploring targeted corrections of underperforming tasks without retraining.

2 Related Work

Structured World Models. World models for robotic manipulation now span latent dynamics, action-conditioned predictive models, and other forms of predictive representation (Wang et al., 2026). Object-level latent dynamics explicitly represent interacting entities (Kipf et al., 2020), while generative object-centric models extend this structure to temporally coherent imagination (Lin et al., 2020). Object-oriented dynamics further support compositional generalization across changing object configurations (Zhao et al., 2022), and compositional world models have recently been applied to robot imagination (Zhou et al., 2024). In contrast, control-oriented latent world models such as TD-MPC2 represent action effects implicitly inside an action-conditioned latent transition (Hansen et al., 2024). Action–object binding makes the destination of control more explicit by associating actions with individual objects (Biza et al., 2022). These approaches structure objects, interactions, or the routing of control, but the transformation from the current action to a specific object's predicted change generally remains implicit inside the transition.

Causal Analysis of Learned Dynamics. Causal representation learning seeks latent variables whose structure and responses to intervention reflect underlying causal factors (Schölkopf et al., 2021). Interventional representation learning studies how controlled interventions constrain the recovery of such latent factors (Ahuja et al., 2023). Within sequential decision making, factorized world models have been used to estimate causal influence between agents and object dynamics (Zholus et al., 2022), while causal world models can construct explanatory chains connecting actions, environment variables, and reward (Yu et al., 2023). Object-oriented causal dynamics models instead learn dependencies among object variables directly from interaction data (Yu et al., 2024). Dynamic causal models can further represent causal relations that change as interactions evolve (Orujlu et al., 2026). These methods primarily address the identification, discovery, or explanation of causal dependencies. An explicit action-to-object transformation that participates in the ordinary forward transition and subsequently serves as a direct intervention target remains less explored.

Model Editing. Model editing studies post-training modification of learned behavior without full retraining. MEND maps desired input–output corrections to localized parameter updates (Mitchell et al., 2022), while ROME targets internal computations associated with factual associations (Meng et al., 2022). MEMIT extends this form of editing to substantially larger sets of associations (Meng et al., 2023). Related ideas have begun to appear in learned dynamics: ReDRAW adapts pretrained world models through latent-state dynamics residuals that compensate for changes in system dynamics (Lanier et al., 2026). These approaches demonstrate that model parameters and latent dynamics can be altered after training. Direct editing of an explicit action-to-object mechanism while keeping both the current state and executed action fixed remains comparatively unexplored.

3 Method

Our objective is to make the contribution of the current robot action explicit inside the learned object dynamics, rather than recovering action influence only after prediction through gradients or perturbation tests. We therefore structure the transition so that the action-dependent contribution to the primary object is represented by a state-conditioned operator that participates directly in the forward computation. This construction provides a single model component that supports prediction, auditing, and intervention.

3.1 Explicit Action-to-Object Dynamics

We build on the latent world-model formulation used for continuous control in TD-MPC2 (Hansen et al., 2024), while changing how the current action enters the learned object dynamics. Standard action-conditioned latent transitions jointly transform a latent state and action into the next latent state, leaving the contribution of the current action entangled with the remaining transition computation. We instead provide the current action with a dedicated route into the dynamics of the task-relevant primary object.

Structured physical context. We factorize the latent state as $z = [z_R, z_P, z_G]$, where z_R, z_P, and z_G denote the robot, primary-object, and goal representations, respectively. Physical dynamics

operate on the robot–object context $h = [z_R, z_P]$. The goal representation is carried unchanged through the transition, while reward, value, policy, and planning components continue to condition on the full latent state. This separation preserves task information for decision making without treating the goal as part of the physical robot–object dynamics.

The shared physical-dynamics backbone produces $G_P(h)$, the state-dependent base component of the primary-object transition. We define the effective action-to-object operator as $R(h) = p(h)A(h)$, where $p(h) \in [0, 1]$ is a scalar state-conditioned gate and $A(h) \in \mathbb{R}^{d_P \times d_a}$ maps the robot action into the primary-object latent space. Here, d_P and d_a denote the primary-object latent dimension and action dimension, respectively.

Given the executed robot action a, the scale β controls the magnitude of its explicit action-dependent contribution. We combine this contribution with the base transition and apply SimNorm, the group-wise softmax normalization used in TD-MPC2 (Hansen et al., 2024), with groups of eight features:

$$\ell'_P = G_P(h) + \beta R(h)a, \qquad z'_P = \mathrm{SimNorm}(\ell'_P). \tag{1}$$

We use $\beta = 1$ in the main experiments; sensitivity to this scale is examined in Appendix E. Architectural dimensions and further implementation details are provided in Appendix B.

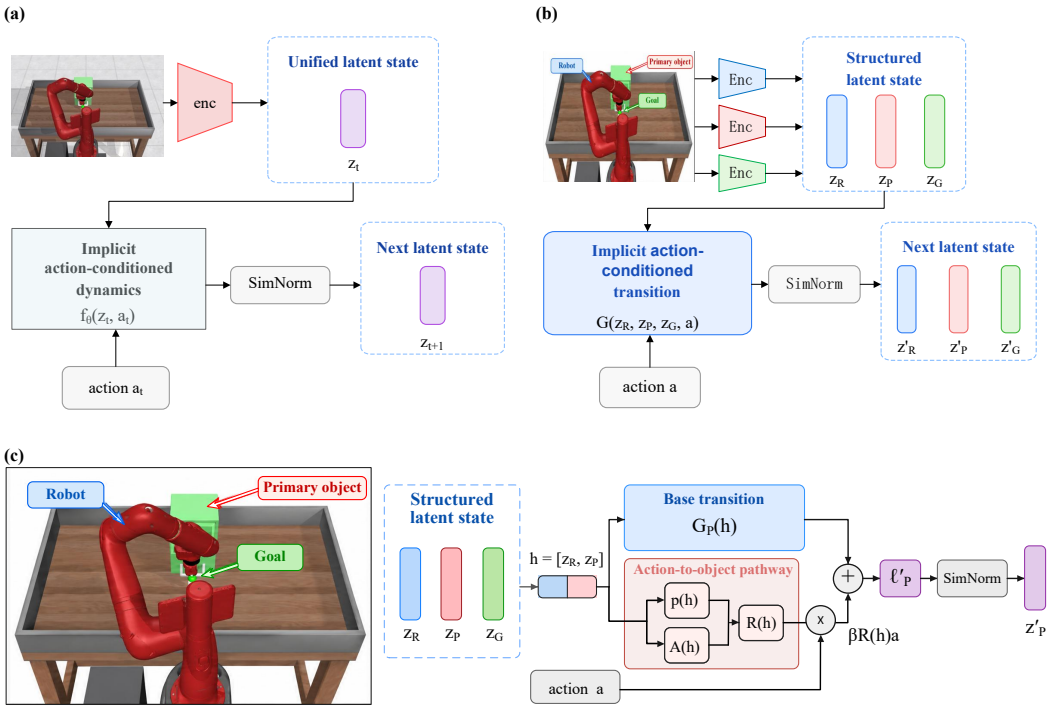

Figure 1: Comparison of latent transition architectures. (a) TD-MPC2 uses a unified latent representation and implicit action-conditioned dynamics. (b) Our factorized baseline uses separate encoders for the robot (z_R), primary object (z_P), and goal (z_G), while retaining a unified implicit transition. (c) Our model uses the physical context $h = [z_R, z_P]$ and introduces an explicit state-conditioned action-to-object pathway in the primary-object transition. Only the primary-object transition is shown in (c).

Explicit action routing. Figure 1(c) illustrates how the current action enters the primary-object transition through $R(h)$. For a fixed physical context h, the mapping from the current action to its contribution is linear, while the operator's dependence on h remains nonlinear. The effective operator $R(h)$ is the product $p(h)A(h)$ used in the forward transition, rather than an additional network module.

To distinguish the learned mapping from the contribution produced by a particular action, we use the following notation. The matrix $R(h)$ denotes the complete action-to-object mapping at the current physical context. For action coordinate j, the vector $r_j(h) = R(h)e_j$ denotes the corresponding coordinate-specific pathway, where e_j is the j-th standard basis vector. For the currently executed action, $c(h, a) = R(h)a$ denotes the realized action-dependent contribution. Thus, $R(h)$, $R(h)e_j$, and $R(h)a$ refer respectively to the operator itself, an individual action-coordinate pathway, and the contribution generated by the current action.

Forward faithfulness. The explicit pathway is part of the transition computation itself rather than an auxiliary quantity extracted after prediction. At a fixed physical context h, consider an arbitrary action perturbation Δa. Applying Eq. 1 to a and $a + \Delta a$ and subtracting the two pre-SimNorm predictions yields

$$\ell'_P(h, a + \Delta a) - \ell'_P(h, a) = \beta R(h)\Delta a. \tag{2}$$

The identity holds exactly before SimNorm because $G_P(h)$ and $R(h)$ remain fixed when only the action is perturbed. It establishes that $R(h)$ is the operator through which changes in the current action enter the pre-normalization primary-object prediction. No derivative, surrogate explainer, or separately fitted perturbation model is required to define the pathway. The coordinate magnitude $s_j(h) = \|R(h)e_j\|_2$ therefore provides an intrinsic action-influence score, which is evaluated against independent same-state simulator interventions in Sec. 4.2.

Complete transition and pathway learning. The explicit action-to-object decomposition is introduced only in the primary-object branch, which is the focus of the pathway analysis and mechanism editing. The same shared physical-dynamics backbone also produces $G_R(h)$, the state-dependent base component of the robot transition. The robot branch combines this component with an action-conditioned residual: $\ell'_R = G_R(h) + \gamma_R F_R(h, a)$, followed by $z'_R = \mathrm{SimNorm}(\ell'_R)$. Here, $F_R(h, a)$ is the robot-branch action-conditioned residual, and γ_R controls its scale. The main models use $\gamma_R = 1$. This asymmetry is intentional: the method exposes how the robot action contributes to the predicted primary-object transition while leaving the robot-state update in the underlying latent-dynamics form.

The goal representation is carried forward unchanged, giving the complete transition $f_\theta(z, a) = [z'_R, z'_P, z_G]$. The explicit pathway is optimized jointly with the latent-dynamics, reward, and value objectives inherited from the TD-MPC family (Hansen et al., 2022; 2024). It receives no direct action-effect labels, paired simulator interventions, or counterfactual transition supervision. The pathway is therefore learned through the ordinary world-model objective and is subsequently evaluated using intervention signals that were unavailable during training.

3.2 Causal Mechanism Editing

Once the action-dependent contribution is exposed as an explicit operator, the learned transition admits an intervention that differs from changing the action input. The intervention acts on the mapping from the current action to its predicted primary-object consequence while keeping both the physical context and executed action fixed. We refer to this operation as causal mechanism editing within the learned transition.

Mechanism-level intervention. For a fixed physical context h and executed action a, the original action-dependent contribution is $c = R(h)a$. Let $c^\star$ denote a prescribed alternative contribution in the primary-object latent space. Rather than replacing the action, we seek an edited operator $\widetilde{R}(h)$ satisfying

$$\widetilde{R}(h)a = c^\star, \qquad \widetilde{\ell}'_P = G_P(h) + \beta c^\star. \tag{3}$$

The physical context h, the base transition $G_P(h)$, and the executed action a are identical before and after the intervention. Only the effective action-to-object mapping is changed. The edit therefore assigns a different learned consequence to the same action rather than evaluating the model under a different action.

This distinction separates an input intervention from a mechanism intervention. Replacing a changes the action supplied to the learned dynamics. Editing $R(h)$ instead preserves that action and modifies the learned mapping through which it contributes to the predicted object transition. A reference

action may be used to construct $c^\star$, but that reference action is never supplied to the edited transition; the forward computation continues to receive the original action a.

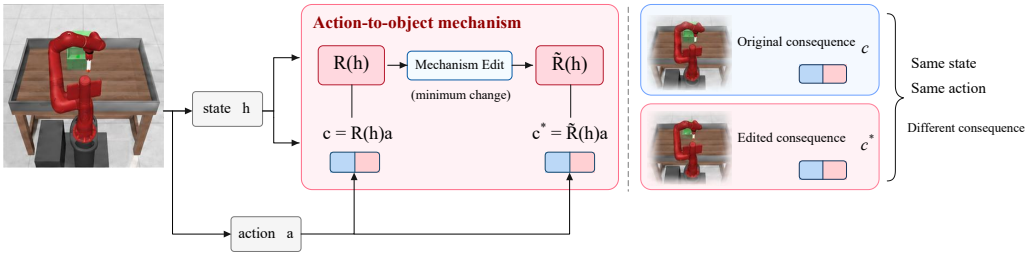

Figure 2: Targeted mechanism editing. With the physical state and executed action fixed, the action-to-object operator is changed from $R(h)$ to $\widetilde{R}(h)$ so that the same action is assigned a prescribed alternative object-level consequence. The edit acts on the learned mechanism rather than on the action input.

Minimum-change operator update. Figure 2 illustrates the same-state, same-action intervention. Let $\widetilde{R}(h)$ denote the edited operator and $R(h)$ the original operator at the fixed physical context h. The operator update is $\Delta R = \widetilde{R}(h) - R(h)$. To realize the prescribed contribution $c^\star$ for the fixed action a, we minimize $\|\Delta R\|_F^2$ subject to $(R(h) + \Delta R)a = c^\star$. For any nonzero anchor action a, this problem has the unique solution

$$\Delta R^\star = \frac{\big(c^\star - R(h)a\big)a^\top}{\|a\|_2^2}, \qquad \widetilde{R}(h) = R(h) + \Delta R^\star. \tag{4}$$

The update has rank at most one: its output-space component is the required correction $c^\star - R(h)a$, while its action-space component is aligned with the anchor action a. It therefore introduces only the operator change required to realize the prescribed consequence.

Editing guarantee. Equation 4 realizes the requested contribution exactly: substitution gives $\widetilde{R}(h)a = c^\star$. The guarantee applies to the pre-SimNorm transition; the subsequent normalized representation is obtained using the same nonlinear SimNorm operation as in the original world model.

The same solution also has an action-space locality property. For every direction b orthogonal to the anchor action, $a^\top b = 0$ implies $\Delta R^\star b = 0$, and therefore $\widetilde{R}(h)b = R(h)b$. The intervention consequently changes the operator only along the action-space component required by the anchor action while preserving all orthogonal directions. This locality concerns directions in action space rather than individual columns of the operator. Complete proofs of exact target realization, uniqueness, and orthogonal locality are given in Appendix A.

Post-training intervention. The edit is applied to the effective operator evaluated at the current physical context and does not require retraining the world model or updating its global parameters. Its purpose is therefore not to rewrite the entire dynamics network, but to make a learned transition mechanism directly manipulable at inference time. The same quantity $R(h)$ participates in ordinary forward prediction, provides an intrinsic pathway for auditing action influence, and serves as the direct target of mechanism-level intervention. Numerical stabilization used in the implementation is described in Appendix B.

4 Experiments

We evaluate the proposed representation along three dimensions. First, we test whether the exposed operator faithfully corresponds to the action-dependent computation used by the learned transition

and whether its coordinate-wise influence aligns with independent simulator interventions. Second, we test whether the operator can be edited to realize prescribed alternative consequences while keeping the current state and executed action fixed. Third, we intervene on the pathway during closed-loop control to test whether modifying the learned mechanism has systematic behavioral consequences.

4.1 Experimental setup

We evaluate on ten Meta-World manipulation tasks (Yu et al., 2020) using low-dimensional state observations. Each task is trained independently, rather than using a jointly trained multi-task agent. For each task and model variant, we train five independent models with seeds 0–4, yielding 50 frozen final checkpoints per variant. All variants use the same TD-MPC2 learning and planning pipeline and exactly 20,000 ordinary environment interactions per task–seed run. No paired simulator interventions, direct action-effect labels, or counterfactual transition pairs are used during training. All pathway audits and post-training interventions are performed on frozen models.

The latent state contains robot, primary-object, and goal representations of dimensions 160, 224, and 128, respectively, and the robot action is four-dimensional. Main experiments use $\beta = 1$ and $\gamma_R = 1$. Training uses the latent-consistency, reward, and value objectives inherited from the TD-MPC family (Hansen et al., 2022; 2024).

Figure 1 illustrates the three architectures compared in Table 1. TD-MPC2 uses a unified latent representation and an implicit action-conditioned transition. Factorized is our representation-structured baseline: it uses separate encoders for the robot, primary object, and goal, but retains a unified implicit transition over the complete latent state. Our full model uses the same encoder architecture and further introduces structured physical dynamics with an explicit action-to-object operator. This comparison evaluates control competency across the three architectures; the contribution of the exposed pathway is examined separately through direct interventions.

Table 1: Control competency across ten Meta-World tasks.

Model	Robot–Object–Goal Factorization	Explicit Action-to-Object Operator	Success ↑
TD-MPC2	–	–	0.524 ± 0.044
Factorized (our)	✓	–	0.710 ± 0.040
Ours	✓	✓	0.754 ± 0.043

4.2 Validation of the explicit action-to-object pathway

We first verify that the exposed operator corresponds to the action-dependent computation of the learned transition. At a fixed physical context h, we perturb the action and compare the resulting pre-SimNorm transition change with the change predicted by Eq. 2. The mean numerical discrepancy is 2.41×10^{-7}, confirming forward faithfulness to numerical precision.

We next examine whether the pathway identifies action coordinates that affect the primary object in the simulator. From identical simulator states, we apply paired action perturbations $a + \delta e_j$ and $a - \delta e_j$, with $\delta = 0.2$. We measure the normalized primary-object response $\rho_j(s)$ as the distance between the two resulting object positions divided by 2δ. A coordinate is labeled active when $\rho_j(s) > 10^{-4}$. These external labels are never used during training. The precise definition is given in Appendix C. We compare the intrinsic score $s_j(h) = \|R(h)e_j\|_2$ against Jacobian and finite-difference sensitivities of the Factorized baseline. Alignment is measured by AUPRC and its lift over positive-class prevalence. Reach has no positive intervention pairs and is excluded, leaving nine non-null tasks.

Figure 3 summarizes both tests. The explicit pathway achieves an aggregate AUPRC of 0.342, compared with 0.313 for both post-hoc baselines. It achieves positive lift on all $9/9$ non-null tasks, versus $7/9$ for each baseline. Task-wise results remain mixed: the pathway performs better on five tasks and worse on four. The audit therefore supports consistent external alignment with a modest aggregate advantage, rather than uniform superiority.

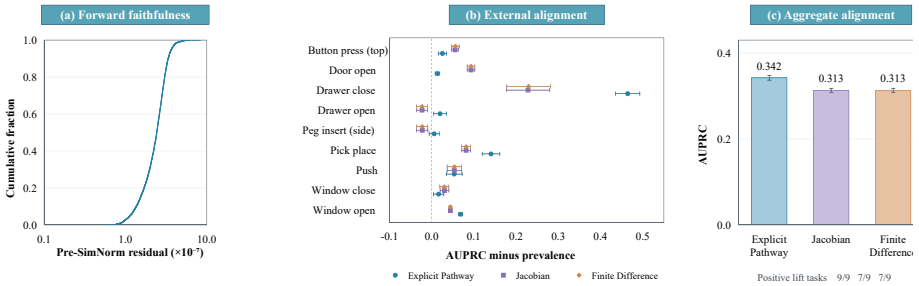

Figure 3: Forward faithfulness and external alignment. (a) Cumulative fraction of probes with pre-SimNorm discrepancy below the indicated threshold. (b) Task-wise AUPRC lift over prevalence, evaluated against independent simulator interventions. (c) Mean AUPRC and number of tasks with positive lift across the nine non-null tasks.

4.3 Mechanism Editing and Closed-Loop Effects

We evaluate the editable pathway at two levels: whether a prescribed change can be realized precisely inside the learned transition, and whether modifying the pathway can affect downstream robot behavior.

Precise mechanism editing. We first audit minimum-change editing on nine frozen checkpoints from Drawer-Close, Push, and Door-Open. Using 900 archived state–action anchors, two target families, and four action coordinates, we evaluate 7,200 edits. The targets remove or sign-flip the contribution associated with one action coordinate. In every case, the original state and executed action remain fixed; only the effective action-to-object mapping is modified.

The edited operator realizes its prescribed model-internal contribution with a maximum residual of 3.44×10^{-7}. The robot and goal branches remain unchanged, and action-space directions orthogonal to the anchor action are preserved. These results establish precise local control over the learned mechanism, without implying that every prescribed edit is behaviorally beneficial. Full target-realization curves, transition residuals, and numerical locality diagnostics are provided in Figure 5 and Table 6 in Appendix F.

Behavioral consequences. We next examine whether the pathway matters for closed-loop control. Removing $R(h)$ reduces mean terminal success from 0.754 to 0.586, while fixing the gate to $p(h) = 1$ reduces it to 0.456. The gate intervention is a factor-level diagnostic, since $p(h)$ and $A(h)$ are not separately identifiable. Full ablation results appear in Appendix F.

To test selective post-training intervention, we scale one column of $R(h)$ by α, leaving all trained parameters and other columns unchanged. Unlike minimum-change targeted editing, this intervention does not prescribe a fixed-action object consequence. It instead changes the action dependence available to the planner during closed-loop control. Table 2 reports selected positive outcomes.

Table 2: Selected coordinate-wise pathway edits. Push and Door-Open are exploratory; Pick-Place uses held-out initial states. Changes are absolute differences in success rates (percentage points), displayed with percentage signs.

Task	Edit	Original	Edited	Change
Push	$y, \alpha = 1.5$	86.7%	91.7%	+5.0%
Door-Open	$x, \alpha = 2$	63.3%	70.0%	+6.7%
Pick-Place	$z, \alpha = 2$	28.0%	52.0%	+24.0%

The Pick-Place edit increases observed success from 28% to 52% on held-out initial states, but the improvement is not statistically significant after correction for multiple comparisons. Push and Door-Open are exploratory results; exploratory gains on Window-Close do not persist on held-out states. Complete results, including unsuccessful edits, are reported in Appendix F.3.



Figure 4: Selected paired closed-loop rollouts under coordinate-wise pathway scaling. **(a)** Door-Open (x, $\alpha = 2$): both rollouts succeed. **(b)** Pick-Place (z, $\alpha = 2$): the original rollout fails, while the edited rollout succeeds.

Qualitative robot behavior. Figure 4 visualizes selected paired closed-loop rollouts. In Door-Open, both the original and edited models complete the task. In Pick-Place, the original rollout fails while the edited rollout succeeds. The frame sequences show the resulting robot behavior, rather than only numerical changes to the model's internal representation. Together with the editing audit, these examples connect precise model-internal intervention to observable downstream consequences without retraining the world model. They demonstrate behavioral relevance, not guaranteed recovery of unsuccessful tasks.

5 Conclusion

We studied whether a robot world model can expose not only the consequence predicted for an action, but also the learned pathway through which that action contributes to the transition of a task-relevant object. Our model introduces an explicit state-conditioned action-to-object operator inside the forward dynamics, making the current action contribution directly accessible during prediction rather than recovering it only through post-hoc analysis.

The resulting pathway serves three roles within the same learned transition. It is forward-faithful because it participates directly in the primary-object prediction, auditable because its coordinate-wise influence can be compared with independent same-state simulator interventions, and editable because the operator itself can be modified after training while the physical context and executed action remain fixed. The minimum-change edit provides a precise mechanism-level intervention: the same action can be assigned a prescribed alternative object-level consequence without changing the action input. Closed-loop interventions further show that modifying this pathway produces systematic behavioral consequences, establishing that it is not merely an auxiliary explanatory quantity.

These results separate two capabilities that are often conflated in learned dynamics. Exposing and editing an internal transition mechanism is one problem; selecting an intervention that improves long-horizon task performance is another. The present work addresses the former and shows that an action-to-object pathway can be made simultaneously predictive, auditable, and directly manipulable inside a robot world model.

Limitations. The exposed pathway is a learned causal mechanism within the model transition, not a recovery of the environment's physical causal structure. Our evaluation is limited to low-dimensional Meta-World tasks, without visual observations or real-robot experiments, and the external audit measures primary-object motion as the observable intervention effect. Extending the approach to high-dimensional perception, longer-horizon dynamics, and real robots remains future work. Exact editability also does not determine which intervention will improve downstream control.

Reproducibility Statement

We provide proofs of the editing properties in Appendix A, implementation and training details in Appendix B, the external simulator audit protocol in Appendix C, and the targeted-editing protocol in Appendix D. Sensitivity analyses are reported in Appendix E, while complete intervention results, including positive, negative, and unchanged editing outcomes, are provided in Appendix F. All pathway audits and post-training editing experiments use frozen checkpoints. Additional retraining experiments for pathway-scale sensitivity are documented separately.

AI Use Statement

We used OpenAI ChatGPT (GPT-5.6 Sol) and OpenAI Codex (GPT-6 Astra and GPT-5.6 Sol) to assist with code generation and debugging, figure polishing and formatting, mathematical derivations and proof drafting, and manuscript editing, including language polishing and grammar checking. All AI-assisted code, mathematical arguments, figures, and text were reviewed and verified by the authors. Generative AI was not used to generate experimental data or reported numerical results, and all final scientific decisions and interpretations were made by the authors. The authors take full responsibility for the final content of the paper.

References

Kartik Ahuja, Divyat Mahajan, Yixin Wang, and Yoshua Bengio. Interventional causal representation learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 372–407. PMLR, 2023. URL https://proceedings.mlr.press/v202/ahuja23a.html.

Ondrej Biza, Robert Platt, Jan-Willem van de Meent, Lawson L. S. Wong, and Thomas Kipf. Binding actions to objects in world models. In *ICLR 2022 Workshop on the Elements of Reasoning: Objects, Structure, and Causality*, 2022. URL https://research.google/pubs/binding-actions-to-objects-in-world-models/.

Samuel Greydanus, Anurag Koul, Jonathan Dodge, and Alan Fern. Visualizing and understanding Atari agents. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1792–1801. PMLR, 2018. URL https://proceedings.mlr.press/v80/greydanus18a.html.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://papers.nips.cc/paper/2018/hash/2de5d16682c3c35007e4e92982f1a2ba-Abstract.html.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2555–2565. PMLR, 2019. URL https://proceedings.mlr.press/v97/hafner19a.html.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):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://proceedings.iclr.cc/paper_files/paper/2024/hash/cf73d57b6dcda32b293df7c2d5341f49-Abstract-Conference.html.

Nicklas A. Hansen, Hao Su, and Xiaolong Wang. Temporal difference learning for model predictive control. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 8387–8406. PMLR, 2022. URL https://proceedings.mlr.press/v162/hansen22a.html.

Thomas N. Kipf, Elise van der Pol, and Max Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1gax6VtDB.

JB Lanier, Kyungmin Kim, Armin Karamzade, Yifei Liu, Ankita Sinha, Kat He, Davide Corsi, and Roy Fox. Adapting world models with latent-state dynamics residuals. In *Proceedings of The 8th Annual Learning for Dynamics and Control Conference*, volume 331 of *Proceedings of Machine Learning Research*, pp. 117–144. PMLR, 2026. URL https://proceedings.mlr.press/v331/lanier26a.html.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6140–6149. PMLR, 2020. URL https://proceedings.mlr.press/v119/lin20f.html.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Turan Orujlu, Christian Gumbsch, Martin V. Butz, and Charley M. Wu. Causal process models: Reframing dynamic causal graph discovery as a reinforcement learning problem. In *Proceedings of the Fifth Conference on Causal Learning and Reasoning*, volume 323 of *Proceedings of Machine Learning Research*, pp. 1450–1480. PMLR, 2026. URL https://proceedings.mlr.press/v323/orujlu26a.html.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954. URL https://ieeexplore.ieee.org/document/9363924.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3319–3328. PMLR, 2017. URL https://proceedings.mlr.press/v70/sundararajan17a.html.

Fangyuan Wang, Ziyuan Wang, Guorui Pei, Mengshi Zhang, Canxi Liang, Jun Hu, Zhongxuan Li, Jinsong Wu, Ning Han, Zeqing Zhang, Jiaming Qi, Hongmin Wu, Shiyao Zhang, Pai Zheng, Jia Pan, David Navarro-Alarcon, Sichao Liu, and Peng Zhou. World models for robotic manipulation: A survey. *SmartBot*, 2026. doi: 10.1002/smb2.70053. URL https://doi.org/10.1002/smb2.70053. Early View, e70053.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 1094–1100. PMLR, 2020. URL https://proceedings.mlr.press/v100/yu20a.html.

Zhongwei Yu, Jingqing Ruan, and Dengpeng Xing. Explainable reinforcement learning via a causal world model. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pp. 4540–4548, 2023. doi: 10.24963/ijcai.2023/505. URL https://www.ijcai.org/proceedings/2023/505.

Zhongwei Yu, Jingqing Ruan, and Dengpeng Xing. Learning causal dynamics models in object-oriented environments. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57597–57638. PMLR, 2024. URL https://proceedings.mlr.press/v235/yu24j.html.

Linfeng Zhao, Lingzhi Kong, Robin Walters, and Lawson L. S. Wong. Toward compositional generalization in object-oriented world modeling. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 26841–26864. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhao22b.html.

Artem Zholus, Yaroslav Ivchenkov, and Aleksandr Panov. Factorized world models for learning causal relationships. In *ICLR 2022 Workshop on Objects, Structure and Causality*, 2022. URL https://openreview.net/forum?id=BCGfDBOIcec.

Siyuan Zhou, Yilun Du, Jiaben Chen, Yandong Li, Dit-Yan Yeung, and Chuang Gan. RoboDreamer: Learning compositional world models for robot imagination. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 61885–61896. PMLR, 2024. URL https://proceedings.mlr.press/v235/zhou24f.html.

A Proofs of the Editing Properties

This appendix establishes three properties used in the main text: (i) the exposed operator exactly represents the action-dependent component of the pre-normalization transition at a fixed context; (ii) the proposed targeted edit is the unique minimum-Frobenius-norm operator change realizing a prescribed consequence; and (iii) this edit leaves action directions orthogonal to the anchor action unchanged.

A.1 Exact Forward Action Effect

Recall the primary-object transition

$$\ell'_P(h, a) = G_P(h) + \beta R(h)a. \tag{5}$$

Proposition 1. For a fixed robot–object context h and any action perturbation Δa,

$$\ell'_P(h, a + \Delta a) - \ell'_P(h, a) = \beta R(h)\Delta a. \tag{6}$$

Proof. Since h is fixed, both $G_P(h)$ and $R(h)$ are unchanged by the action perturbation. Therefore,

$$\ell'_P(h, a + \Delta a) = G_P(h) + \beta R(h)(a + \Delta a) \tag{7}$$

$$= G_P(h) + \beta R(h)a + \beta R(h)\Delta a. \tag{8}$$

Subtracting $\ell'_P(h, a) = G_P(h) + \beta R(h)a$ gives

$$\ell'_P(h, a + \Delta a) - \ell'_P(h, a) = \beta R(h)\Delta a. \tag{9}$$

The identity is exact for the pre-SimNorm transition. After SimNorm, the corresponding difference is generally nonlinear.

A.2 Minimum-Frobenius-Norm Targeted Edit

For a fixed context h, write $R = R(h)$ for brevity. Given a nonzero anchor action a and a desired alternative contribution $c^\star$, we seek an operator update ΔR such that

$$(R + \Delta R)a = c^\star. \tag{10}$$

Equivalently, with

$$d = c^\star - Ra, \tag{11}$$

the constraint becomes

$$\Delta R a = d. \tag{12}$$

Proposition 2. For $a \neq 0$, the unique minimum-Frobenius-norm solution of

$$\min_{\Delta R} \|\Delta R\|_F^2 \qquad \text{s.t.} \qquad \Delta R a = d \tag{13}$$

is

$$\Delta R^\star = \frac{d a^\top}{\|a\|_2^2} = \frac{(c^\star - Ra)a^\top}{\|a\|_2^2}. \tag{14}$$

Proof. Consider the Lagrangian

$$\mathcal{L}(\Delta R, \lambda) = \frac{1}{2}\|\Delta R\|_F^2 + \lambda^\top(\Delta R a - d), \tag{15}$$

where λ is the multiplier associated with the equality constraint.

Stationarity with respect to ΔR gives

$$\nabla_{\Delta R}\mathcal{L} = \Delta R + \lambda a^\top = 0, \tag{16}$$

and hence

$$\Delta R = -\lambda a^\top. \tag{17}$$

Substituting into the constraint yields

$$-\lambda a^\top a = d, \tag{18}$$

so, because $a \neq 0$,

$$\lambda = -\frac{d}{\|a\|_2^2}. \tag{19}$$

Therefore,

$$\Delta R^\star = \frac{d a^\top}{\|a\|_2^2}. \tag{20}$$

The objective $\|\Delta R\|_F^2$ is strictly convex and the constraint is affine, so this feasible stationary point is the unique minimizer.

Substituting the solution back into the edited operator verifies exact target realization:

$$(R + \Delta R^\star)a = Ra + \frac{(c^\star - Ra)a^\top a}{\|a\|_2^2} \tag{21}$$

$$= Ra + (c^\star - Ra) = c^\star. \tag{22}$$

A.3 Orthogonal Action-Space Locality

Proposition 3. Let $\widetilde{R} = R + \Delta R^\star$ be the minimum-change operator from Proposition 2. For every action direction b satisfying $a^\top b = 0$,

$$\widetilde{R}b = Rb. \tag{23}$$

Proof. Using the solution from Proposition 2,

$$\Delta R^\star b = \frac{(c^\star - Ra)a^\top b}{\|a\|_2^2}. \tag{24}$$

If b is orthogonal to a, then $a^\top b = 0$, and therefore

$$\Delta R^\star b = 0. \tag{25}$$

It follows immediately that

$$\widetilde{R}b = (R + \Delta R^\star)b = Rb. \tag{26}$$

Thus, the targeted edit changes the operator only along the anchor-action direction required to realize $c^\star$, while preserving all orthogonal action directions.

B Implementation Details

Latent representation. The learned state is factorized as $z = [z_R, z_P, z_G]$, where z_R, z_P, and z_G denote the robot, primary-object, and goal representations, respectively. Their dimensions are 160, 224, and 128, giving a 512-dimensional latent state. The physical transition operates on the robot–object context $h = [z_R, z_P]$. The goal representation is carried unchanged through the learned transition, i.e., $z'_G = z_G$, while reward, value, policy, and planning components continue to condition on the full latent state including z_G. The robot action is four-dimensional.

Physical transition. The robot and primary-object base transitions are produced from the shared physical context h. We denote the shared physical-dynamics backbone by B_θ and write

$$[G_R(h), G_P(h)] = B_\theta(h).$$

The primary-object transition is

$$\ell'_P = G_P(h) + \beta R(h)a, \qquad z'_P = \mathrm{SimNorm}(\ell'_P),$$

with $\beta = 1$ in the main models.

The effective action-to-object operator is defined as

$$R(h) = p(h)A(h),$$

where $p(h) \in [0, 1]$ is a scalar sigmoid gate and $A(h) \in \mathbb{R}^{224\times 4}$ is a state-conditioned action operator. The action-dependent contribution in the forward transition is $R(h)a$, where $R(h)$ is the effective operator defined above rather than an additional network module. All pathway analyses and interventions in the paper are defined using this effective operator.

The robot branch uses a separate action-conditioned residual,

$$\ell'_R = G_R(h) + \gamma_R F_R(h, a), \qquad z'_R = \mathrm{SimNorm}(\ell'_R),$$

where F_R receives both the physical context and the current action, and $\gamma_R = 1$ in the main models. The complete learned transition is thus

$$f_\theta(z, a) = [z'_R, z'_P, z_G].$$

SimNorm uses groups of size eight.

Training objective. The action-to-object pathway is learned jointly with the world model rather than from separate causal supervision. The model objective is

$$\mathcal{L}_{\mathrm{model}} = 20\,\mathcal{L}_{\mathrm{con}} + 0.1\,\mathcal{L}_r + 0.1\,\mathcal{L}_Q,$$

where $\mathcal{L}_{\mathrm{con}}$ is the latent-consistency objective and $\mathcal{L}_r$ and $\mathcal{L}_Q$ train reward and value prediction. Because predicted latent states are used in subsequent reward and value prediction, gradients to $p(h)$ and $A(h)$ are not restricted to a single one-step primary-object consistency term.

The formal training runs use no paired simulator interventions, counterfactual transition pairs, direct action-effect labels, or auxiliary targets for $R(h)$, $p(h)$, or $A(h)$. The pathway is therefore induced through ordinary world-model learning rather than explicitly supervised to match the simulator intervention audit used later for evaluation.

Numerical implementation of targeted editing. The theoretical minimum-change update derived in Appendix A is

$$\Delta R^\star = \frac{(c^\star - R(h)a)a^\top}{\|a\|_2^2}.$$

For numerical stability, the implementation instead uses

$$\Delta R_\epsilon = \frac{(c^\star - R(h)a)a^\top}{\|a\|_2^2 + \epsilon}, \qquad \epsilon = 10^{-12}.$$

The formal minimum-norm and orthogonal-locality results correspond to $\epsilon = 0$. With the implementation value above, the deviation from the exact pseudoinverse solution remains at numerical precision in our audit.

C External Simulator Audit

The external audit evaluates whether the action influence exposed by the learned pathway corresponds to measurable action influence in the simulator. The audit is performed entirely after training using frozen checkpoints; no simulator intervention data are used to supervise the pathway.

Same-state action intervention. For a simulator state s, recorded action a, and action coordinate j, we construct two perturbed actions

$$a_j^+ = a + \delta e_j, \qquad a_j^- = a - \delta e_j,$$

where e_j is the j-th standard basis vector and $\delta = 0.2$. Both perturbed actions are evaluated from the same underlying simulator state, so the measured difference isolates the response to the intervention on coordinate j.

Let $\phi_P(o)$ denote the position of the task-relevant primary object extracted from observation o. We define the external simulator response as

$$\rho_j(s) = \frac{\left\|\phi_P(o_j^+) - \phi_P(o_j^-)\right\|_2}{2\delta}.$$

A state–action-coordinate pair is labeled *positive* when $\rho_j(s) > 10^{-4}$ and *negative* otherwise. Thus, the external label indicates whether perturbing an action coordinate produces measurable primary-object motion from the same simulator state.

Probe bank. The frozen audit uses three source families, five seeds, and ten probe times, giving 150 simulator states per task. Each state is evaluated along all four action coordinates, yielding 600 intervention pairs per task and 6,000 pairs across the ten-task Meta-World suite. Reach contains no positive intervention pairs under this protocol and is therefore excluded from AUPRC evaluation, leaving nine non-null tasks.

Pathway and sensitivity scores. For the explicit action-to-object pathway, coordinate j receives the score

$$s_j(h) = \|R(h)e_j\|_2.$$

This score is read directly from the explicit pre-SimNorm pathway and measures the magnitude of the action-to-object mapping associated with that coordinate.

For the post-hoc baselines, let $F_P^{\mathrm{fac}}(h, a)$ denote the completed primary-object latent transition of the Factorized model. Jacobian sensitivity is

$$s_j^{\mathrm{Jac}}(h, a) = \left\| \frac{\partial F_P^{\mathrm{fac}}(h, a)}{\partial a_j} \right\|_2.$$

In the implementation, this directional derivative is evaluated using a Jacobian–vector product along the j-th action basis direction.

Finite-difference sensitivity uses the same symmetric action perturbations as the frozen audit bank, with $\delta = 0.2$:

$$s_j^{\mathrm{FD}}(h, a) = \frac{\left\|F_P^{\mathrm{fac}}(h, a + \delta e_j) - F_P^{\mathrm{fac}}(h, a - \delta e_j)\right\|_2}{2\delta}.$$

Both post-hoc scores are therefore computed from the completed Factorized transition, whereas $s_j(h)$ is read directly from the explicit pathway. All three scores are evaluated against the same frozen simulator-derived labels; the simulator intervention is used only to define the external evaluation target.

Evaluation metric. We evaluate the continuous influence scores using the *area under the precision–recall curve* (AUPRC). Each state–action-coordinate pair has a binary external label and a continuous influence score. AUPRC therefore measures how well the score ranks truly object-affecting action coordinates ahead of inactive ones over all possible decision thresholds.

AUPRC is particularly appropriate here because positive action–object interactions are sparse. Under such class imbalance, classification accuracy can remain high by predicting mostly negative cases, whereas precision–recall evaluation focuses directly on retrieval of the positive effects.

For each task, the fraction of positive intervention pairs defines the positive-class prevalence, which serves as the random-ranking baseline. We therefore additionally report

$$\text{AUPRC lift} = \text{AUPRC} - \text{prevalence}.$$

A positive lift indicates that an action-influence score ranks externally active action coordinates better than random ordering.

Aggregate results. Across the nine non-null tasks, the explicit pathway obtains an aggregate AUPRC of 0.342, compared with 0.313 for both Jacobian and finite-difference sensitivity. The pathway achieves positive AUPRC lift over prevalence on all $9/9$ tasks, while each post-hoc baseline achieves positive lift on $7/9$ tasks. At the individual-task level, the pathway performs better than the sensitivity baselines on five tasks and worse on four. Accordingly, we interpret the audit as evidence of consistent external alignment with a modest aggregate advantage, rather than uniform superiority across tasks.

D Targeted Editing Protocol

This appendix describes the evaluation protocol used to test whether the learned action-to-object pathway can be edited to assign a fixed action a prescribed alternative primary-object consequence.

Evaluation set. The targeted-editing audit uses nine frozen final checkpoints from Drawer-Close, Push, and Door-Open, with three independently trained seeds per task. For each checkpoint, we use 100 archived state–action anchors, yielding 900 anchors in total. All edits are performed after training on frozen models; no parameter update or additional environment interaction is used.

Each anchor consists of a fixed latent state and its recorded action. We evaluate all four action coordinates and construct two alternative consequences per coordinate, resulting in

$$900 \times 4 \times 2 = 7{,}200$$

targeted edits.

Fixed-state, fixed-action intervention. For an anchor context h and recorded action a, the original action-dependent contribution is

$$c = R(h)a.$$

The purpose of the audit is to modify the learned mapping from a to its primary-object consequence while keeping both h and the actually supplied action a unchanged.

A reference action a_{ref} is introduced only to define a desired alternative contribution

$$c^\star = R(h)a_{\text{ref}}.$$

Importantly, a_{ref} is never passed to the edited transition. The edited model continues to receive the original recorded action a. This distinction ensures that a successful edit cannot be explained by simply replacing the action input.

Coordinate-consequence removal. For action coordinate j, we construct a_{ref} by setting only that coordinate to zero while leaving all remaining coordinates unchanged. The resulting target is

$$c^\star_{\text{remove}} = R(h)a_{\text{ref}}.$$

This target asks the edited pathway to assign the original action the contribution that would have been produced had coordinate j contributed nothing to the reference action.

Sign-flip transplantation. For the second target family, the sign of coordinate j is reversed while all remaining coordinates are preserved. The target is again defined as

$$c^\star_{\text{flip}} = R(h)a_{\text{ref}}.$$

This construction tests whether the learned pathway can transplant a different coordinate-specific consequence onto the original action rather than merely suppressing an effect.

Minimum-change edit. Given either target $c^\star$, we edit the effective operator using

$$\widetilde{R}(h) = R(h) + \Delta R_\epsilon,$$

where

$$\Delta R_\epsilon = \frac{\big(c^\star - R(h)a\big)a^\top}{\|a\|_2^2 + \epsilon}, \qquad \epsilon = 10^{-12}.$$

The exact minimum-Frobenius-norm solution corresponds to $\epsilon = 0$ and is proved in Appendix A. The small positive ϵ is used only for numerical stability.

After the edit, the transition is evaluated using the original state and original action:

$$\widetilde{\ell}'_P = G_P(h) + \beta \widetilde{R}(h)a.$$

Thus, the only intended change is the action-to-object mapping itself.

Target-realization audit. We evaluate the edit at three levels. First, we measure the contribution-space residual

$$\left\| \widetilde{R}(h)a - c^\star \right\|_2.$$

Second, we compare the actual edited pre-SimNorm transition against the transition implied by the prescribed target. Third, we repeat the comparison after SimNorm. These checks distinguish exact realization of the requested action contribution from possible numerical effects introduced by the subsequent normalization.

Intervention invariance. Because the central claim concerns editing the mechanism while holding the cause fixed, we explicitly verify that the executed action is unchanged by the intervention. We also audit the robot and goal branches to ensure that the targeted edit does not modify unrelated components of the learned transition. Across the formal audit, action mutation as well as robot- and goal-branch changes are zero.

Minimum-norm and locality verification. We independently compare the closed-form update against an SVD-based pseudoinverse solution of the same constrained minimum-norm problem. The two solutions agree to numerical precision. We additionally evaluate the edited operator on action directions orthogonal to the anchor action. As established in Appendix A, the exact update satisfies $\Delta R^\star b = 0$ whenever $a^\top b = 0$; the numerical audit verifies the same locality up to floating-point error.

Together, these evaluations test a stronger property than merely changing a prediction: with state and executed action fixed, the learned action-to-object mechanism can be modified so that the same action is assigned a prescribed alternative primary-object consequence.

E Sensitivity and Continuous Mechanism Editing

This section examines three quantities that are fixed in the main experiments but play different roles in the proposed framework. The pathway scale β controls the magnitude of the explicit action-to-object contribution, the simulator intervention radius δ controls the external audit, and α controls a post-training coordinate-wise edit of the learned operator. These analyses are performed separately because β is a model-scale parameter, δ is an evaluation-protocol parameter, and α is an intervention variable.

Sensitivity to the pathway scale β. The primary-object transition contains the explicit contribution $\beta R(h)a$. We use $\beta = 1$ in the main experiments. For a fixed trained checkpoint, multiplying the intrinsic score $s_j(h) = \|R(h)e_j\|_2$ by any positive fixed β applies the same positive scaling to every score and therefore leaves its AUPRC unchanged. The more relevant question is whether changing β affects closed-loop behavior or whether training relies critically on a particular fixed scale.

We first change β only at inference time for the frozen main models. We report terminal-success changes relative to $\beta = 1$ within the same evaluation protocol. For $\beta \in \{0.5, 1, 2\}$, the changes are

-0.003, 0, and -0.040, respectively. These paired comparisons use their own evaluation seed set and are distinct from the competency evaluation in Table 1.

We additionally retrain models at each scale on three representative tasks—Drawer-Close, Push, and Door-Open—using three independent seeds per scale. The $\beta = 1$ condition reuses the corresponding checkpoints from the main experiment.

Table 3: Sensitivity to the fixed pathway scale β. Frozen-model results are terminal-success changes relative to $\beta = 1$ under the same evaluation protocol. Retrained results use Drawer-Close, Push, and Door-Open with three seeds per scale.

β	Frozen Δ success	Retrained AUPRC	Retrained success	Relative $\|\beta p(h)A(h)\|_F$
0.5	-0.003	0.431	0.722	$0.94\times$
1.0	0	0.459	0.783	$1.00\times$
2.0	-0.040	0.451	0.528	$0.99\times$

The retrained models substantially compensate for the imposed scale. Relative to $\beta = 1$, the mean effective pathway norm $\|\beta p(h)A(h)\|_F$ is $0.94\times$ for $\beta = 0.5$ and $0.99\times$ for $\beta = 2$. The effective pathway magnitude is therefore comparatively stable after retraining, while excessively scaling the pathway at inference time can still degrade closed-loop control. We use $\beta = 1$ as the canonical scale in the main experiments rather than treating β as a tuned performance parameter.

Robustness to the simulator intervention radius δ. The external audit in Sec. 4.2 uses paired same-state interventions $a \pm \delta e_j$ with the nominal radius $\delta = 0.2$. To test whether the external-alignment result depends on this choice, we repeat the audit with $\delta \in \{0.1, 0.2, 0.3\}$ using a common-center protocol. The simulator states, source trajectories, center actions, and pathway scores are shared across the three conditions; only the intervention radius changes.

Table 4: Robustness of the external simulator audit to the intervention radius δ. AUPRC statistics are computed over the nine non-null tasks.

δ	Mean AUPRC	Mean lift over prevalence	Positive-lift tasks
0.1	0.3401	$+0.08895$	9/9
0.2	0.3436	$+0.09031$	9/9
0.3	0.3456	$+0.09081$	9/9

The resulting external labels are also highly stable. Pairwise binary intervention-label agreement is 99.23% between $\delta = 0.1$ and 0.2, 99.37% between $\delta = 0.2$ and 0.3, and 98.93% between $\delta = 0.1$ and 0.3. The common-center $\delta = 0.2$ result is also close to the independently constructed main audit. These results indicate that the external-alignment conclusion is not specific to the nominal intervention radius.

As an additional protocol check, varying the activity threshold from 10^{-5} to 10^{-3} preserves positive mean AUPRC lift. We therefore use $\delta = 0.2$ and the 10^{-4} activity threshold as fixed nominal settings, rather than selecting them based on the reported pathway performance.

Continuous coordinate-wise mechanism editing with α. The targeted edits in the main text prescribe a desired alternative contribution $c^\star$. We additionally test whether the exposed mechanism supports continuous, coordinate-specific modulation rather than only a binary removal or replacement.

For action coordinate j, define a diagonal action-space operator $D_j(\alpha)$ that equals the identity except for its j-th diagonal entry, which is set to α. We edit the effective operator as

$$\widetilde{R}_{\alpha,j}(h) = R(h)D_j(\alpha), \qquad \alpha \in \{0, 0.5, 1, 1.5, 2\}.$$

Only the selected coordinate pathway is scaled; the physical context h and the executed action a remain unchanged. For $r_j(h) = R(h)e_j$, the corresponding pre-SimNorm primary-object transition satisfies

$$\widetilde{\ell}'_P(\alpha) - \widetilde{\ell}'_P(1) = \beta(\alpha - 1)r_j(h)a_j.$$

Thus, $\alpha = 0$ removes the selected coordinate pathway, $\alpha = 1$ recovers the unedited model, and values above or below one continuously strengthen or attenuate the contribution of that coordinate without replacing the executed action.

We evaluate this intervention on the nine frozen checkpoints from Drawer-Close, Push, and Door-Open used in the targeted-editing audit, with three seeds per task. Each checkpoint uses 20 matched initial simulator snapshots. The $\alpha = 1$ trajectory is shared as the baseline for all coordinate edits, producing 3,060 main closed-loop episodes. No model is retrained.

We first verify the intervention at fixed states. From the baseline trajectories, we sample 900 real state–action anchors at decision steps 0, 25, 50, 75, and 99. Across all four action coordinates and all five values of α, this yields 18,000 fixed-state comparisons. All comparisons satisfy the predicted coordinate-wise identity within floating-point precision. The maximum absolute pre-SimNorm theory error is 4.92×10^{-7}, the maximum relative target-magnitude error is 2.38×10^{-7}, and both the maximum off-target coordinate error and maximum action mutation are zero.

To summarize the downstream behavioral sensitivity without selecting a coordinate post hoc, we compute, for each task and coordinate, the mean absolute change in terminal success across the four non-baseline α values. Table 5 reports this quantity in percentage points.

Table 5: Closed-loop sensitivity to continuous coordinate-wise pathway editing. Entries are the mean absolute task-level terminal-success change, in percentage points, over $\alpha \in \{0, 0.5, 1.5, 2\}$ relative to $\alpha = 1$.

Task	x	y	z	Gripper
Drawer-Close	1.67	0.42	2.50	1.67
Push	6.67	6.67	7.08	4.58
Door-Open	15.00	6.25	9.17	7.08

Although Table 5 reports absolute changes to summarize intervention sensitivity without selecting favorable edits post hoc, some individual edits also yield higher mean terminal success. For example, scaling the y pathway in Push with $\alpha = 1.5$ increases success from 86.7% to 91.7%, while scaling the x pathway in Door-Open with $\alpha = 2$ increases it from 63.3% to 70.0%. These improvements are task- and coordinate-dependent rather than uniform.

The effect of continuous pathway scaling is therefore strongly task- and coordinate-dependent. Drawer-Close is comparatively insensitive over this range, whereas Push and Door-Open show larger changes for several coordinates. Importantly, no single value of α or single coordinate produces a uniformly beneficial behavioral change across tasks. The continuous-editing experiment therefore supports two conclusions. First, the exposed operator can be modulated smoothly and selectively while preserving the original state, action, and non-target pathways. Second, the existence of a precise editable mechanism does not imply that a universally beneficial edit can be selected without an additional task-level criterion.

F Additional Intervention Results

F.1 Numerical Audit of Targeted Editing

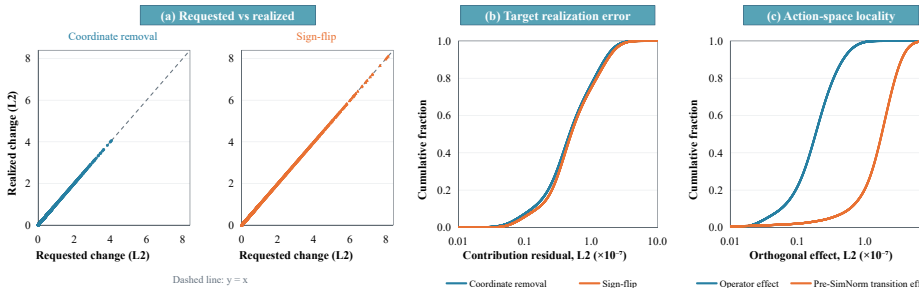

Figure 5: Quantitative audit of targeted pathway editing over 7,200 interventions. **(a)** Requested versus realized consequence changes. **(b)** Target-realization residuals. **(c)** Orthogonal action-space locality.

Table 6: Properties of targeted causal editing. The state and executed action are fixed for every intervention.

Property	Audit	Result
Target realization	Contribution residual	3.44×10^{-7}
Transition fidelity	Pre/post-SimNorm residual	$6.81/3.08 \times 10^{-7}$
Same-action intervention	Action mutation	0
Branch locality	Robot/goal change	0/0
Minimum-change solution	SVD agreement	3.33×10^{-16}

F.2 Closed-Loop Pathway Ablations

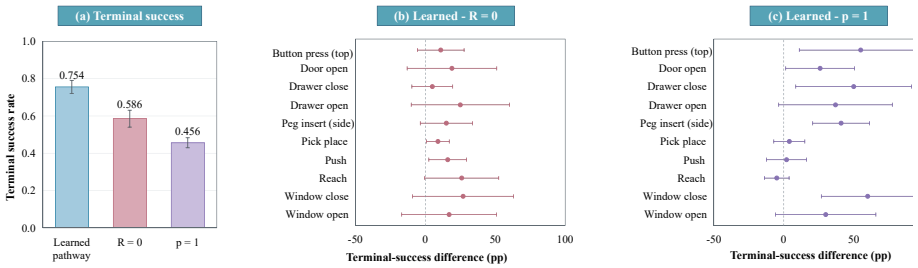

Figure 6: Closed-loop consequences of pathway intervention. **(a)** Aggregate terminal success. **(b)** Task-wise effect of pathway removal ($R = 0$). **(c)** Task-wise effect of gate saturation ($p = 1$).

F.3 Complete Results of Coordinate-wise Editing

We report all evaluated coordinate-wise editing conditions, including positive, negative, and unchanged outcomes. For action coordinate j, the intervention scales only the corresponding column of $R(h)$ by α. The unedited model corresponds to $\alpha = 1$. These interventions are distinct from the minimum-norm targeted edit studied in Sec. 4.3.

Original three-task exploration. Table 7 reports the complete terminal-success sweep on Drawer-Close, Push, and Door-Open, using three frozen checkpoints per task. Each checkpoint uses 20 matched initial conditions. All entries are percentages.

Table 7: Complete terminal-success results for the original three-task exploratory sweep. All four action coordinates and all evaluated α values are included. The $\alpha = 1$ column is the shared unedited reference.

Task	Coord.	$\alpha = 0$	0.5	1	1.5	2
Drawer-Close	x	100.0	95.0	100.0	100.0	98.3
	y	100.0	100.0	100.0	98.3	100.0
	z	91.7	98.3	100.0	100.0	100.0
	Gripper	100.0	95.0	100.0	98.3	100.0
Push	x	83.3	81.7	86.7	80.0	75.0
	y	73.3	81.7	86.7	91.7	83.3
	z	80.0	78.3	86.7	81.7	78.3
	Gripper	80.0	81.7	86.7	83.3	83.3
Door-Open	x	28.3	51.7	63.3	56.7	70.0
	y	45.0	58.3	63.3	65.0	63.3
	z	55.0	51.7	63.3	56.7	53.3
	Gripper	63.3	66.7	63.3	51.7	50.0

Additional seven-task exploration. Table 8 reports all 84 discovery conditions over seven additional tasks. All five available frozen training seeds are evaluated. The reported values are terminal-success percentages, and the discovery results are not independent confirmation.

Table 8: Complete seven-task discovery results. No evaluated coordinate or editing strength is omitted. Original denotes $\alpha = 1$; all values are terminal-success percentages. Button-Press denotes Button-Press-Topdown.

Task	Coord.	Original	$\alpha = 0.5$	1.5	2
Button-Press	x	100	100	100	100
	y	100	100	100	100
	z	100	92	100	100
	Gripper	100	100	100	100
Drawer-Open	x	80	76	80	76
	y	80	80	80	68
	z	80	80	80	80
	Gripper	80	80	80	80
Peg-Insert-Side	x	52	52	48	44
	y	52	44	48	48
	z	52	40	56	40
	Gripper	52	52	56	48
Pick-Place	x	40	24	36	28
	y	40	36	40	24
	z	40	28	40	48
	Gripper	40	20	40	32
Reach	x	84	84	84	88
	y	84	84	84	84
	z	84	76	84	84
	Gripper	84	84	84	84
Window-Close	x	84	84	96	84
	y	84	100	84	84
	z	84	96	92	88
	Gripper	84	96	88	88
Window-Open	x	96	100	88	92
	y	96	100	96	96
	z	96	100	96	100
	Gripper	96	100	96	92

Held-out confirmation. Table 9 reports all six candidates selected for held-out confirmation. The independent statistical unit is the training seed ($n = 5$). Confidence intervals are two-sided, unadjusted Student-t intervals over paired seed-level differences.

Table 9: Complete held-out confirmation results. Terminal-success changes and their 95% confidence intervals are in percentage points. No candidate passed Holm correction.

Task	Edit	Original	Edited	Δ (pp)	95% CI (pp)	Return Δ
Pick-Place	$z, \alpha = 2$	28	52	+24	$[+3.22, +44.78]$	+62.31
Window-Close	$x, \alpha = 1.5$	92	92	0	$[-17.56, +17.56]$	+45.29
	$y, \alpha = 0.5$	92	80	-12	$[-49.66, +25.66]$	-46.67
	$z, \alpha = 0.5$	92	88	-4	$[-15.11, +7.11]$	+77.15
	$z, \alpha = 1.5$	92	84	-8	$[-45.66, +29.66]$	-84.11
	Gripper, $\alpha = 0.5$	92	84	-8	$[-21.60, +5.60]$	+30.61

Although Pick-Place exhibits a positive held-out effect, it does not survive correction for multiple comparisons. The exploratory Window-Close candidates do not reproduce their positive discovery outcomes. These results demonstrate that editing the exposed pathway can alter downstream control, but that selecting consistently beneficial interventions remains an open problem.