LAW, STATE, AND BODY: OBJECT-CENTRIC WORLD MODELS FOR COUNTERFACTUAL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Most learned world models compress a scene into a single flat vector and roll it forward with a recurrent network or a transformer. This works well for prediction, but it hides the objects that a physical scene is actually made of, so the model has no natural way to answer a question such as "what would happen if this object moved differently." We present LSB, a world model that keeps physical law, the visible state of a scene, and the body that acts on it as three separate, swappable pieces. State is a set of per-object slots produced by slot attention. Law is a self-attention transformer that updates every slot using only the slots themselves, so the same law applies regardless of which actuator produced the action. Body is a small embodiment-specific encoder that turns raw actions and proprioception into a control signal that Law can consume. Because objects are explicit and individually addressable, we can edit one object's velocity and roll the edited state forward through the same law used for ordinary prediction, giving a direct and cheap way to test counterfactual reasoning. Across a three-embodiment, six-physics-regime benchmark, LSB beats a strong non-slotted baseline by large, statistically robust margins on next-state prediction, on future-contact queries, and on physical intervention, and the advantage survives zero-shot transfer to six unseen physics regimes, to embodiments never trained with intervention, to object counts far outside the training range, and to rollouts sixteen times longer than the training horizon. We further show that the same advantage appears when Law is instantiated as a completely different architecture, a relational graph network in the style of interaction networks, and when Law is trained with a contrastive energy objective instead of regression, which indicates that the gain comes from the explicit object-centric design rather than from any one choice of network or loss.

1 INTRODUCTION

A world model that only predicts what a scene will look like next is useful for imagination-based planning, but it leaves out two abilities that matter just as much in practice. The first is querying, asking a targeted question about the future such as whether two objects are about to collide. The second is intervening, asking what would happen if we changed one thing about the scene while leaving everything else alone. Both abilities require a state representation with parts that can be read out and edited individually. A single flat embedding produced by a convolutional encoder and rolled forward by a recurrent network or a transformer does not have this property. There is no coordinate inside such a vector that corresponds to "the velocity of the second object," so there is nothing principled to edit when we want to ask a counterfactual question.

Object-centric representation learning gives a natural fix. Slot attention and its many descendants show that a scene can be decomposed into a small set of per-object slots without any supervision (Locatello & Weissenborn, 2020; Burgess et al., 2019; Greff et al., 2019; Lin et al., 2020b; Kipf et al., 2022). Separately, a long line of relational and graph-based dynamics models shows that once a scene is represented as a set of entities, physical interaction can be learned as message passing between them (Battaglia et al., 2016; Chang et al., 2017; Watters et al., 2017; Kipf et al., 2018; Sanchez-Gonzalez et al., 2020). Our starting observation is that these two ideas, together with a

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.

third one that is often left implicit, combine into a clean design for a world model that supports prediction, querying, and intervention with one architecture.

The third idea is that the rule governing how objects move should not need to know which actuator produced the action that is currently perturbing them. A pushing hand, a two-link arm, and a sliding paddle all act on the same kind of physical scene, and the underlying law of motion for the objects in that scene does not change from one embodiment to the next. We therefore split our world model into three pieces. State is the set of per-object slots produced by a slot-attention encoder. Law is a self-attention transformer that reads the current slots plus a control signal and produces the next slots, and never sees which body produced the control signal. Body is a small, swappable encoder, one per embodiment, that turns raw action and proprioception into the control signal Law consumes. We refer to this model as LSB throughout.

This split has a direct practical payoff. Because State is a set of individually addressable object slots, intervening is as simple as overwriting one slot's velocity or position and calling Law again, the same function used for ordinary rollout. Because Law never sees embodiment identity, a law learned under one body transfers to another without retraining, and can be evaluated zero-shot under new physics or with more objects than it ever saw in training. We build a unified STEP / QUERY / INTERVENE benchmark around this design, together with a contact-pair classification task, and evaluate everything with a paired, multi-seed statistical protocol, since point estimates in this kind of stochastic evaluation can be noisy enough to flip sign on their own.

Our contributions are:

- An explicit Law / State / Body decomposition for object-centric world models, with one dynamics module shared across embodiments and a training recipe that combines self-distilled next-state prediction with a short counterfactual fine-tuning stage, backed by a statistically robust evaluation, ten to twenty paired seeds per comparison, that shows large, significant gains over a strong non-slotted baseline on prediction, query, and intervention, a gain that survives even when the baseline is given the same velocity information and the identical counterfactual training recipe.
- The strongest generalization result in this paper: with no retraining at all, query and intervention capability transfers zero-shot across six qualitatively different unseen physics regimes, to object counts far outside the training range, and to rollouts sixteen times longer than the training horizon, while the non-slotted baseline degrades on query and is consistently and severely backwards on intervention in every regime.
- Evidence that this advantage is architecture-agnostic and not a measurement artifact. A second, structurally different Law module, a relational graph network in the style of interaction networks, and a third variant trained with a contrastive energy objective instead of regression both land in the same place relative to the non-slotted baseline, and a systematic sweep over rollout horizon and object count shows the intervention margin at the standard setting is a floor effect of a short horizon and several bystander objects rather than a limit on what counterfactual training can achieve.

2 Related Work

Object-centric representation learning has moved from purely generative scene decomposition into a general toolkit for structured perception and prediction. MONet and IODINE recover object segments and per-object latents through iterative amortized inference (Burgess et al., 2019; Greff et al., 2019), SPACE adds spatial attention for scalable decomposition (Lin et al., 2020b), and slot attention replaces this machinery with a lightweight, permutation-equivariant competition among slots that has become the default building block for later work (Locatello & Weissenborn, 2020). SAVi extends slot attention to video with optical-flow supervision (Kipf et al., 2022), G-SWM and OP3 push further into rollout quality and entity-conditioned planning (Lin et al., 2020a; Veerapaneni et al., 2019), and C-SWM trains a transition function with a contrastive, energy-based objective to avoid the collapse that naive next-state regression risks in a purely self-supervised setting (Kipf et al., 2020). A newer thread asks how well these representations hold up outside their training distribution, with Dittadi et al. (2022) showing this depends heavily on design choices earlier papers did not stress-test and Greff et al. (2020) arguing that the binding problem, keeping a representation's

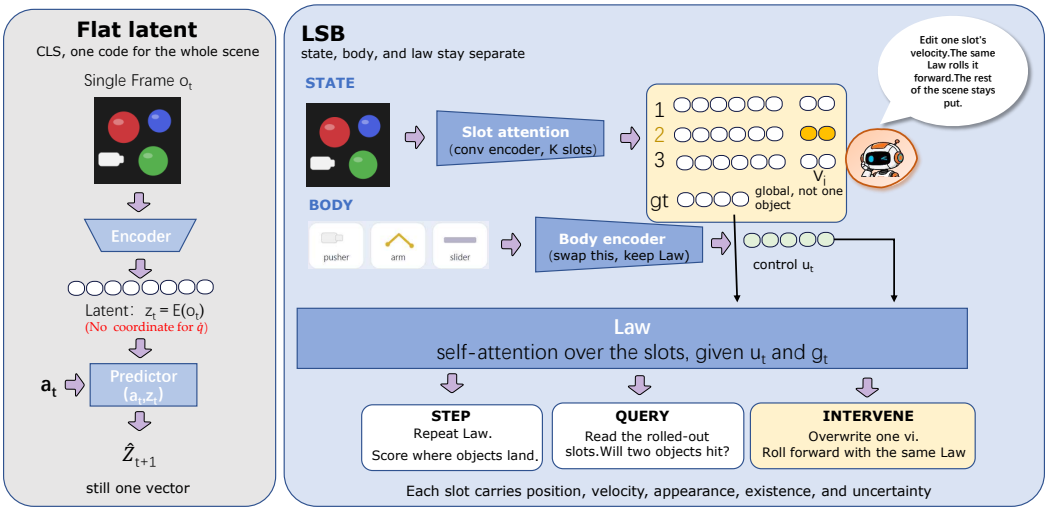

Figure 1: Left: a flat-latent world model collapses a scene into one undifferentiated vector $z_t = E(o_t)$; there is no coordinate in z_t for any single object's velocity, so nothing in the representation is available to edit. Right: LSB keeps State, Body, and Law separate. State is a set of per-object slots, each with a position, a velocity v_i, an appearance code, an existence flag, and an uncertainty, plus a global context g_t. Body maps a specific embodiment's action and proprioception into a control embedding u_t; Law is a self-attention module over the slots, given u_t and g_t, and never sees which body produced u_t. STEP repeats Law to roll the state forward, QUERY reads a property off the rolled-out state, and INTERVENE overwrites one slot's velocity before rolling forward through the same Law, which is what a flat latent on the left has no principled way to do.

parts attached to the right entity over time, is still open. Our State module follows this line directly, and our six-regime zero-shot evaluation continues the same push toward testing generalization rather than reconstruction quality alone.

Flat-latent world models treat prediction as a component of model-based control and roll a single, temporally updated latent forward rather than keeping an explicit object set. Recurrent world models trained end to end with a controller established the basic recipe of learning a compact latent dynamics model and planning inside it (Ha & Schmidhuber, 2018), PlaNet and Dreamer turned this into a latent-space imagination loop that matches and then surpasses model-free reinforcement learning on pixel-based control (Hafner et al., 2019; 2020), and DreamerV3 scales the same recipe across domains with one fixed set of hyperparameters (Hafner et al., 2023). Transformer-based world models such as IRIS instead roll forward a discrete, tokenized latent autoregressively, more sample-efficient than a recurrent one (Micheli et al., 2023), a choice that inspired our own use of self-attention inside Law. None of these models decompose their latent into individually addressable objects, so none offers a principled target to edit when the question is counterfactual rather than predictive, which is exactly the gap our Law / State / Body split closes.

Relational and graph-based dynamics models build the update function directly out of pairwise structure between entities, closer in spirit to our Law module. Interaction networks and the compositional dynamics model of Chang et al. (2017) show that pairwise message passing between object representations is a strong inductive bias for physical prediction, later scaled to graph network simulators that generalize to far more particles at test time (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2020), and visual interaction networks and neural relational inference extend the idea to inferring interaction structure directly from pixels (Watters et al., 2017; Kipf et al., 2018). Region proposal interaction networks scale the same family of ideas to long-horizon visual dynamics (Qi et al., 2021), and object-oriented prediction and planning shows entity-factored dynamics support downstream planning (Janner et al., 2019). Counterfactual and causal physical reasoning has its own dedicated benchmarks: CoPhy predicts the outcome of an intervention on a mechanical scene's initial conditions (Baradel et al., 2020), CLEVRER pairs collision video with descriptive, predictive, and counterfactual questions (Yi et al., 2020), and Physion checks physical prediction against

human judgment (Bear et al., 2021), but all three evaluate counterfactual reasoning on fixed, offline video and question sets rather than a model that also acts. Our STEP / QUERY / INTERVENE protocol instead asks one action-conditioned world model, trained once, to support all three modes of reasoning across embodiments and physics regimes, and our Law module inherits self-attention (Vaswani et al., 2017) rather than the explicit pairwise edge functions of classical interaction networks, a choice we stress-test directly in Section 4.5 by swapping in a relational graph network and finding the same qualitative story holds either way.

3 THE LSB WORLD MODEL

3.1 STATE, LAW, AND BODY

An episode consists of pixel observations o_t, an action a_t, embodiment-specific proprioception, and, only at training time, privileged simulator state (position, velocity, existence) per object. LSB splits predicting, querying, and intervening into three modules, shown in Figure 1.

State. A convolutional encoder maps o_t to a feature map, which slot attention iteratively refines into K object slots (Locatello & Weissenborn, 2020). A shared readout head decodes each slot into a position p_i, a velocity v_i, an appearance code z_i, an existence probability exist_i, and an uncertainty unc_i, plus a separate global context vector g_t not tied to any single object. The full state s_t is this set of K slots plus g_t, and nothing about it is body-specific.

Body. Every embodiment (a pushing hand, a two-link arm, a sliding paddle in our experiments) has its own small encoder that maps its own action space and proprioception into a control embedding u_t of fixed dimensionality, shared across bodies. Body is the only module that knows the action space it is looking at, and the only module we swap when we move to a new embodiment.

Law. A self-attention transformer takes the K slot tokens, broadcasts u_t and g_t into each token as context, and outputs an update for every slot's position, velocity, appearance code, and uncertainty, plus an update for g_t, following an Euler-drift convention. Law never sees which body produced u_t, which is what lets one law be shared by every embodiment and lets us later swap in a structurally different Law module (Section 4.5) with an unchanged interface.

Tasks. Repeatedly applying Law implements STEP, an open-loop rollout scored by average displacement error (ADE). QUERY reads a property off a rolled-out state, for example whether two objects are on a collision course, scored as binary accuracy (E1). INTERVENE overwrites one slot's velocity at a chosen timestep and applies Law forward from there, scored as a counterfactual margin (E2): the reduction in position error, relative to assuming the edited object does not move at all, that the edited value actually buys. A contact-pair task (E3) asks which object pair makes contact first. Full formal definitions are in Appendix A.1.

3.2 TRAINING

Training has two stages. Stage one trains State, Body, and Law jointly with a self-distilled next-state prediction loss, using the model's own encoder on the true next frame as the label, plus privileged binding and pose losses from Hungarian matching to ground-truth simulator objects, available only at training time, to keep each slot bound to a consistent object over time. Stage two, a short counterfactual fine-tune, edits an object's velocity and penalizes the discrepancy between the resulting rollout and what the simulator does under that edit, with a term that keeps the rolled-out velocity following the edited value. This identical two-stage recipe is applied to every model in our comparisons, including every baseline, so differences reflect architecture rather than training budget. Full losses and hyperparameters are in Appendix A.2.

3.3 BASELINES

Our main point of comparison is CLS, sharing LSB's encoder and two-stage recipe but with a single flat recurrent embedding instead of explicit slots, rolled forward with an action-conditioned GRU, in the spirit of the flat-latent world models in Section 2. CLS reads out per-object poses only through a final linear probe, so it has an object count but no per-object dynamics, and because its dynamics function never reads a velocity field, editing a slot's velocity has no causal path to its prediction,

Table 1: STEP / QUERY / INTERVENE on the pushing-hand embodiment. Ten seeds × 48 episodes, paired across models. Arrows show which direction is better; bold marks the best model on each metric.

metric	CLS	CLS+v	LSB (ours)
ADE@8 ↓	0.706	**0.397**	0.450
E1 query acc. ↑	0.327	0.619	**0.719**
E2 intervene margin ↑	−0.616	−0.092	**0.060**
E3 contact-pair acc. ↑	0.119	**0.408**	0.362

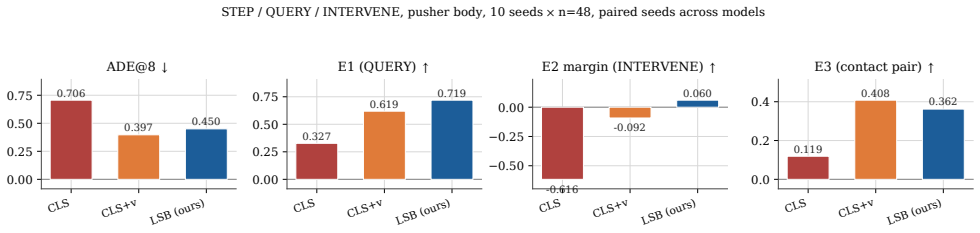

Figure 2: Main result, corresponding to Table 1. LSB dominates on the two metrics that most directly test using an object's own state (STEP, QUERY) and on the counterfactual margin (INTERVENE), while CLS+v, a fair baseline trained with the identical recipe and given the same edited-velocity signal, remains competitive on plain rollout error and contact-pair accuracy.

which could make an intervention comparison unfair. We therefore also build CLS+v, with a learned velocity pathway and the same edited-velocity signal as LSB, trained with the identical counterfactual recipe, so the comparison in Section 4.2 isolates architecture rather than information access. Section 4.5 further compares two structurally different instantiations of Law itself, holding State, Body, and the recipe fixed.

4 Experiments

4.1 Setup

Our environment is a 2D top-down physics sandbox with two to six circular objects of varying mass and radius, optionally a rectangular occluder, and three embodiments (a pushing hand, a two-link arm, a sliding paddle; rendered examples in Appendix A.7, Figure 9) sharing one underlying object physics. Every number below comes from a paired, multi-seed protocol, ten to twenty seeds drawn identically for every compared model, with paired t-tests reported for every comparative claim, since episode-to-episode variance in this kind of stochastic evaluation is large enough that single-seed estimates can flip sign on their own (Appendix A.3).

4.2 Main results

Table 1 and Figure 2 compare LSB against CLS and CLS+v on the pushing-hand embodiment, ten seeds of forty-eight episodes each, the same seed sequence across all three models. LSB wins on ADE and on the query task E1 by a wide, highly significant margin over both baselines. Against CLS+v specifically, on the two metrics that most directly probe counterfactual use of an edited variable, will-it-hit after an edit (E1) and displacement error after an edit relative to doing nothing (E2), LSB wins with a significant paired margin ($t = 3.6$ and $t = 3.5$, ten seeds). CLS+v does close most of the raw gap and even edges out LSB on plain ADE and on the contact-pair task E3, which is exactly the outcome we want to see: it confirms that CLS+v is a genuinely competent, non-crippled baseline, and it makes the remaining, significant advantage on E1 and E2 a much stronger claim than a comparison against a baseline that structurally cannot see the intervened variable at all. This finding, including the specific sign and rough size of every gap, replicates in an independent second training run of the entire CLS+v pipeline (Appendix A.4).

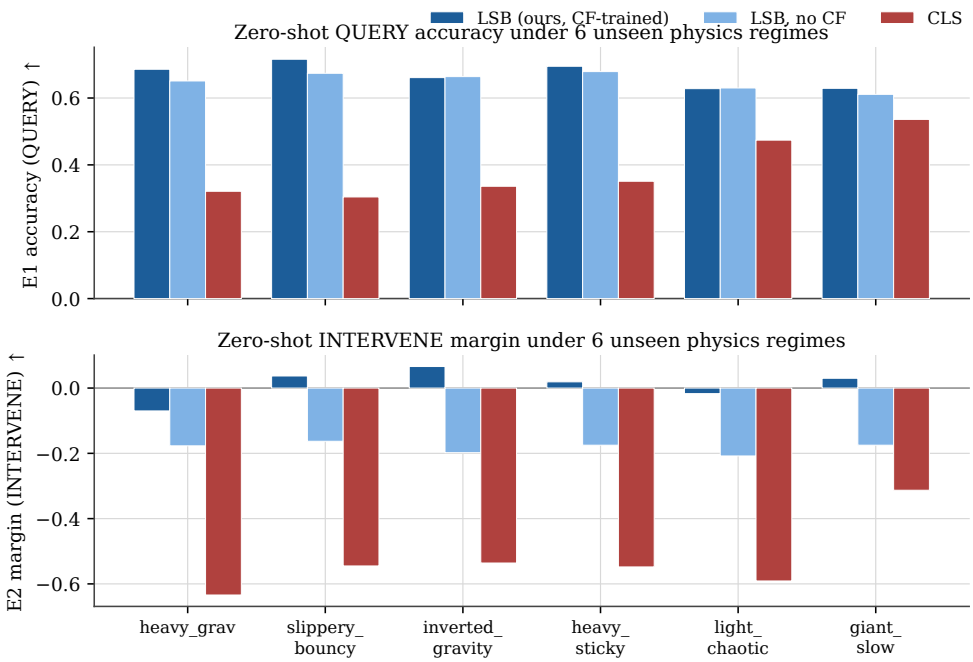

Figure 3: Zero-shot transfer to six unseen physics regimes, no retraining. LSB's query and intervention capability survives essentially intact everywhere, while CLS degrades on query and is actively, consistently backwards on intervention in every single regime.

The same ranking holds regardless of embodiment. On the two-link arm and the sliding paddle, neither of which was ever fine-tuned with the counterfactual recipe, LSB still beats CLS by large, significant margins on all four metrics ($|t|$ between 6.2 and 29.2 across sixteen comparisons, Appendix A.5), showing the advantage of the Law/State/Body split is not an artifact of one particular action space.

4.3 Generalization across unseen physics

The cleanest and safest generalization result in this paper requires no retraining at all. We take three already-trained checkpoints, LSB with counterfactual fine-tuning, LSB without it, and CLS, and evaluate them zero-shot under six qualitatively different physics regimes never seen in training: inverted gravity, heavy sticky objects, light chaotic objects, giant slow objects, heavy gravity, and slippery bouncy objects (Figure 3). Every one of eighteen regime-by-model-pair comparisons is significant and points the same direction. LSB's query accuracy stays in a tight band (0.63 to 0.72) across every regime, essentially unchanged from its in-distribution level, while CLS's query accuracy is worse everywhere (0.30 to 0.54) and never catches up. The intervention margin tells an even sharper story: LSB stays near zero in every regime (-0.07 to $+0.07$), while CLS is severely and consistently backwards in every single regime (-0.63 to -0.31), never once approaching zero. LSB without counterfactual fine-tuning sits in between, consistently around -0.18 to -0.21 regardless of regime, which isolates counterfactual training itself, rather than the slot representation alone, as what produces the near-neutral zero-shot transfer, and shows that this benefit of counterfactual training is itself regime-independent.

The same pattern extends to scenes with more objects than LSB ever saw during training (trained on two to three objects, tested up to six) and to rollouts far beyond the training horizon. Scoring ADE at horizons from one to 128 steps, sixteen times past the standard eight-step window, LSB beats CLS at every single horizon tested, with paired t between -10 and -37 throughout, and at 128 steps LSB's error is still lower than CLS's error at horizon one (full curve and object-count results in Appendix A.6, Figures 7 and 8). LSB's error is not perfectly monotonic in horizon, it rises somewhat in the middle of the range before falling again at the longest horizons we test, a detail we

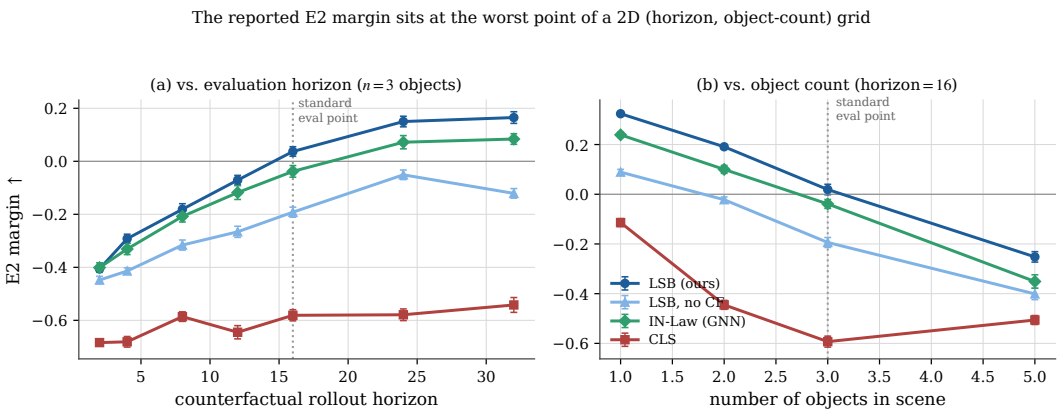

Figure 4: The reported intervention margin sits at the worst point of a two-dimensional (horizon, object-count) grid. Moving to a longer horizon or fewer bystander objects, with the identical checkpoints and zero retraining, reveals a much larger, highly significant counterfactual-training effect that the standard setting understates.

report honestly and discuss further in Appendix A.6, but the headline comparison against CLS is unaffected by this and holds at every single point.

4.4 What the intervention margin at the standard setting actually measures

The intervention margin reported in Table 1 looks modest at first glance, and it is natural to ask whether counterfactual training has actually hit some kind of ceiling. We investigated this with a systematic sweep over the two eval-time constants that define the intervention task, rollout horizon and number of objects in the scene, re-evaluating the very same checkpoints with zero retraining (Figure 4). The answer is that the standard setting sits almost exactly at the worst point of this two-dimensional grid. At short horizons the edit has not had time to move the object far, so ordinary rollout noise swamps the signal for every model, including the counterfactually trained one. With more than one other object present, collisions between bystanders inject noise into the edited object's trajectory that has nothing to do with whether the edit was used correctly. Moving past either nuisance dimension reveals a much larger effect: LSB's margin rises to a clear, significant plateau of $+0.15$ to $+0.17$ at horizons of 24 to 32 steps, roughly four to eight times the size of the standard-setting number, and to $+0.32$ with a single object in the scene, forty standard errors from zero. LSB without counterfactual fine-tuning never once crosses into significantly positive territory anywhere in this sweep, and CLS stays flatly, severely negative everywhere regardless of horizon or object count, which confirms the margin at the standard setting is a genuine, counterfactual-training-specific signal being measured through a noisy window, not a hidden capability limit.

4.5 Is the advantage specific to a transformer trained with regression?

A skeptical reading of everything above is that we have only ever compared an object-centric model against a single family of non-slotted baselines, and that our specific choice, a self-attention transformer trained with next-state regression, might itself be doing unreported work. We test this directly by building two structurally different instantiations of Law, holding State, Body, and the training recipe fixed in both cases.

The first, which we call IN-Law, replaces the transformer with a classic relational graph network in the interaction-network style, explicit pairwise edge functions, masked aggregation, and a node-update function, run for several message-passing rounds, with its width matched so its parameter count is within two percent of the transformer's (Battaglia et al., 2016; Chang et al., 2017). The second keeps the IN-Law architecture but replaces the next-state regression loss with a contrastive, energy-based objective in the style of C-SWM, pushing the predicted next state's energy down for the true transition and up for a randomly sampled negative one (Kipf et al., 2020). Both variants go through the identical counterfactual fine-tuning stage used everywhere else in this paper, and both

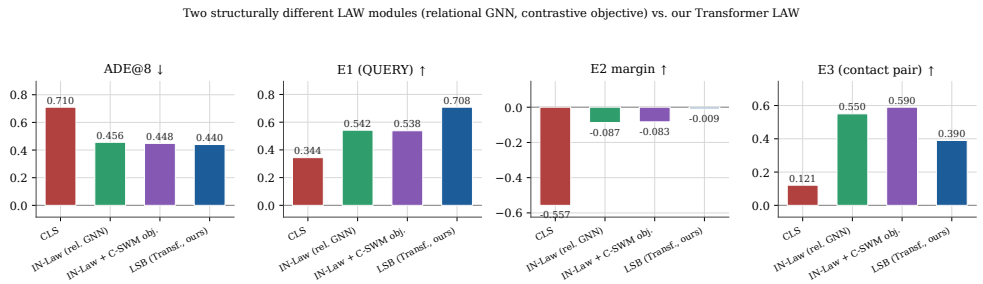

Figure 5: Two structurally different Law modules, a relational graph network and the same network trained with a contrastive energy objective instead of regression, both beat CLS by large margins on every metric and land close to our transformer Law, which indicates the advantage comes from the explicit object-centric framework rather than from one specific network or training loss.

Table 2: Two structurally different Law modules against CLS and our transformer Law. Ten seeds $\times$ 48 episodes, same seed sequence across every row.

metric	CLS	IN-Law (rel. GNN)	IN-Law + C-SWM obj.	LSB (transformer)
ADE@8 ↓	0.710	0.456	0.448	**0.440**
E1 query acc. ↑	0.344	0.542	0.538	**0.708**
E2 intervene margin ↑	−0.557	−0.087	−0.083	−0.009
E3 contact-pair acc. ↑	0.121	**0.550**	0.590	0.390

were independently replicated across two full training runs each (Appendix A.4), so the numbers below are not an artifact of one lucky initialization.

Table 2 and Figure 5 show that both variants crush CLS by the same wide margins our transformer Law does, so neither is a strawman, and both are statistically indistinguishable from our transformer Law on three of the four anchor metrics. The relational graph network actually wins outright on the contact-pair task E3, which fits its explicit pairwise structure, while our transformer keeps a clear edge on the query task E1. Taken together, this says the choice between self-attention and explicit relational message passing trades off differently by task rather than one architecture dominating the other, and it says the object-centric Law / State / Body framework itself, not our specific transformer or our specific regression loss, is the part of the design doing the load-bearing work.

4.6 A QUALITATIVE EXAMPLE

Figure 6 walks through one intervention episode end to end, rendered directly from a trained checkpoint and the physics simulator rather than drawn by hand. An object's velocity is edited at $t{=}0$, the true simulator rolls the edit forward sixteen steps, and we compare the resulting trajectory against LSB's and CLS+v's own rolled-out predictions of the same object. LSB's predicted path tracks the true curved trajectory closely, while CLS+v's prediction stays much closer to the static, no-effect guess, a visual match to the aggregate E1 and E2 numbers in Table 1. More qualitative examples, including query and long-horizon rollout episodes, are given in Appendix A.7.

5 LIMITATIONS

Our environment is a 2D synthetic physics sandbox rather than a photorealistic or real-world one, a choice that keeps ground-truth object state available for training and enables a controlled, statistically powered evaluation of counterfactual reasoning, something much harder to obtain at scale in real video. Extending the same Law / State / Body split to richer visual domains and to 3D physics is a natural next step, and every result central to this paper's main argument has been independently replicated across at least two training runs.

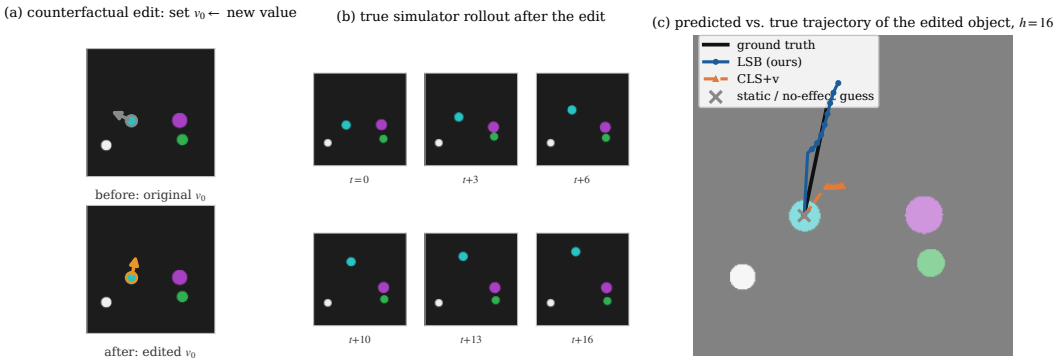

Figure 6: A single intervention episode, rendered from a trained checkpoint and the real simulator. (a) The edited object's velocity before and after the edit; only the velocity changes, the position does not. (b) The true simulator rollout after the edit, six evenly spaced frames. (c) The edited object's predicted trajectory: LSB tracks the true curved path, CLS+v stays close to the static no-effect guess.

6 Conclusion

We presented LSB, a world model that separates physical law, object state, and embodiment into three swappable modules. Because state is a set of individually addressable objects, our model supports prediction, querying, and intervention with the same underlying dynamics function, and it does so with large, statistically robust advantages over a strong non-slotted baseline that hold across embodiments, across six unseen physics regimes, across a sixteenfold increase in rollout horizon, and across two structurally different Law architectures and training objectives. We hope this kind of explicit, addressable state representation becomes a standard building block for world models that need to do more than just predict the next frame.

Reproducibility Statement

Full architecture and hyperparameter details for every model, including the two-stage training recipe, the counterfactual fine-tuning losses, and every baseline's configuration, are in Appendix A.2. Formal definitions of every task and metric (STEP, QUERY, INTERVENE, and the contact-pair task) are in Appendix A.1. Every comparative claim is backed by a paired, multi-seed evaluation with an explicit seed count and, where relevant, an independent replication run, both reported alongside the result; Appendix A.3 discusses the evaluation variance that motivated this protocol. Code, trained checkpoints, and the scripts used to produce every table and figure will be released.

Ethics Statement

This work uses a fully synthetic 2D physics simulator with no human subjects and no personal, sensitive, or web-collected data. We do not foresee harmful applications specific to this work beyond the general considerations that apply to model-based reinforcement learning and robotic manipulation research.

AI Use Statement

An LLM was used only to polish the writing (grammar and clarity) of this paper and help do some experiments. It did not contribute to the research ideas, methodology, or analysis.

References

Fabien Baradel, Natalia Neverova, Julien Mille, Greg Mori, and Christian Wolf. CoPhy: Counterfactual learning of physical dynamics. In *International Conference on Learning Representations (ICLR)*, 2020.

P Battaglia, R Pascanu, M Lai, and D Jimenez Rezende. Interaction networks for learning about objects, relations and physics. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

DM Bear, E Wang, D Mrowca, FJ Binder, HYF Tung, RT Pramod, C Holdaway, and S Tao. Physion: Evaluating physical prediction from vision in humans and machines. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021.

Christopher P. Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matthew Botvinick, and Alexander Lerchner. MONet: Unsupervised scene decomposition and representation. *arXiv preprint arXiv:1901.11390*, 2019.

Michael B. Chang, Tomer Ullman, Antonio Torralba, and Joshua B. Tenenbaum. A compositional object-based approach to learning physical dynamics. In *International Conference on Learning Representations (ICLR)*, 2017.

Andrea Dittadi, Samuele Papa, Michele De Vita, Bernhard Schölkopf, Ole Winther, and Francesco Locatello. Generalization and robustness implications in object-centric learning. In *International Conference on Machine Learning (ICML)*, 2022.

K Greff, S Van Steenkiste, and J Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020.

Klaus Greff, Raphael Lopez Kaufman, Rishabh Kabra, Nick Watters, Chris Burgess, Daniel Zoran, Loic Matthey, Matthew Botvinick, and Alexander Lerchner. Multi-object representation learning with iterative variational inference. In *International Conference on Machine Learning (ICML)*, 2019.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning (ICML)*, 2019.

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

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

Michael Janner, Sergey Levine, William T. Freeman, Joshua B. Tenenbaum, Chelsea Finn, and Jiajun Wu. Reasoning about physical interactions with object-oriented prediction and planning. In *International Conference on Learning Representations (ICLR)*, 2019.

T Kipf, E Van der Pol, and M Welling. Contrastive learning of structured world models. In *International Conference on Learning Representations (ICLR)*, 2020.

T Kipf, GF Elsayed, A Mahendran, A Stone, S Sabour, G Heigold, and R Jonschkowski. Conditional object-centric learning from video. In *International Conference on Learning Representations (ICLR)*, 2022.

Thomas Kipf, Ethan Fetaya, Kuan-Chieh Wang, Max Welling, and Richard Zemel. Neural relational inference for interacting systems. In *International Conference on Machine Learning (ICML)*, 2018.

Zhixuan Lin, Yi-Fu Wu, Skand Peri, Bofeng Fu, Jindong Jiang, and Sungjin Ahn. Improving generative imagination in object-centric world models. In *International Conference on Machine Learning (ICML)*, 2020a.

Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: Unsupervised object-oriented scene representation via spatial attention and decomposition. In *International Conference on Learning Representations (ICLR)*, 2020b.

F Locatello and D Weissenborn. Object-centric learning with slot attention. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Vincent Micheli, Eloi Alonso, and François Fleuret. Transformers are sample-efficient world models. In *International Conference on Learning Representations (ICLR)*, 2023.

Haozhi Qi, Xiaolong Wang, Deepak Pathak, Yi Ma, and Jitendra Malik. Learning long-term visual dynamics with region proposal interaction networks. In *International Conference on Learning Representations (ICLR)*, 2021.

Alvaro Sanchez-Gonzalez, Jonathan Godwin, Tobias Pfaff, Rex Ying, Jure Leskovec, and Peter Battaglia. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning (ICML)*, 2020.

A Vaswani, N Shazeer, and N Parmar. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

R Veerapaneni, JD Co-Reyes, M Chang, M Janner, C Finn, J Wu, J Tenenbaum, and S Levine. Entity abstraction in visual model-based reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

Nicholas Watters, Daniel Zoran, Theophane Weber, Peter Battaglia, Razvan Pascanu, and Andrea Tacchetti. Visual interaction networks: Learning a physics simulator from video. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: Collision events for video representation and reasoning. In *International Conference on Learning Representations (ICLR)*, 2020.

A APPENDIX

This appendix collects formal task definitions, full architecture and training details, every replication study, and every additional experiment that did not fit in the main text, including several honest null results that we report for completeness even though they do not support the paper's main claims.

A.1 FORMAL TASK AND METRIC DEFINITIONS

All tasks operate on a world model that exposes three functions: `encode` (pixels $\rightarrow$ state s_t), `predict` (state, action $\rightarrow$ next state), and `intervene` (state, object index, edit $\rightarrow$ edited state). Let $s_t = \{(z_i, p_i, v_i, \text{exist}_i, \text{unc}_i)\}_{i=1}^{K} \cup \{g_t\}$ denote the K-slot state with global context g_t.

STEP (ADE). Given a context of observed frames, the model encodes the last one into s_0 and applies `predict` for h steps with the true executed actions, producing $\hat{s}_1, \ldots, \hat{s}_h$. Predicted slot positions are matched to ground-truth simulator objects by a greedy Hungarian matching on position, and ADE@h is the mean Euclidean distance between matched predicted and true positions, averaged over existing objects and over the horizon.

QUERY (E1). A scene is constructed so that, under the true simulator, object 0 either does or does not make contact with object 1 within a fixed horizon (24 steps in our setup), balanced 50/50 across episodes. The model is given only the initial state and the executed actions, rolls forward with `predict`, and answers via a lightweight readout over the rolled-out trajectory of the two matched slots (minimum predicted inter-object distance below a contact threshold). E1 is binary accuracy against the true outcome; chance is 0.5.

INTERVENE (E2). At $t = 0$, after matching slot indices to ground-truth objects, the model calls `intervene` to overwrite one object's velocity with a new value v' drawn from a small fixed set of directions and magnitudes. The true simulator is independently reset to the same state and the identical velocity edit is applied to it. Both the model (via repeated `predict`) and the simulator are then rolled forward for a fixed counterfactual horizon h (16 steps at the standard setting used for Table 1). Let p_0 be the object's true position at the moment of the edit, p_h its true position after the true simulator rollout, and $\hat{p}_h$ the model's predicted position after its own rollout. Define $\text{ade_hold} = \|p_0 - p_h\|$ (the error of the naive assumption that the edit had no effect) and $\text{ade_edited} = \|\hat{p}_h - p_h\|$ (the model's actual prediction error). The intervention margin is $E2 = \text{ade_hold} - \text{ade_edited}$; a positive value means the model's prediction, informed by the edit, beats the naive no-effect guess.

Contact pair (E3). Three objects are placed and given a randomized kick. The true simulator determines which of the three possible object pairs makes contact first within a fixed horizon, or records "no contact." The model rolls forward from the initial state and predicts the same label by taking the pair with the smallest minimum predicted inter-object distance (or "no contact" if none falls under threshold). E3 is accuracy against the true label; chance is roughly $1/4$.

Two further tasks, used only in Appendix A.11, are E4 (SCORE), a mind-the-gap style planning score that ranks candidate action sequences by predicted outcome quality and checks the ranking against the true outcome, and E5 (place-to-touch), a short closed-loop planning success rate.

A.2 ARCHITECTURE AND TRAINING DETAILS

State. The convolutional encoder has channel widths $(32, 64, 64)$ over a 64×64 input image (larger for figure rendering only) and produces a z_{dim}=64 feature per spatial location. Slot attention uses K=8 slots and 3 attention iterations. The slot readout is a shared MLP producing $p_i \in \mathbb{R}^2$ (via $\tanh$, so positions live in $[-1, 1]^2$), $v_i \in \mathbb{R}^2$, $\text{exist}_i \in [0, 1]$, and $\text{unc}_i \in \mathbb{R}$, plus a separate global-context head producing $g_t \in \mathbb{R}^{32}$.

Law. Our default Law is a transformer with model width 448, 4 layers, and 4 attention heads over the K slot tokens, taking 448-dimensional per-token input formed from $(z_i, p_i, v_i, \text{exist}_i, \text{unc}_i)$ plus broadcast u_t and g_t context, and producing an additive update to $(z_i, p_i, v_i, \text{unc}_i)$ following an Euler-drift convention with drift coefficient kine_dt. IN-Law (Section 4.5) replaces this with an interaction-network-style module: a per-slot feature embedding, rounds=6 message-passing rounds each with an edge MLP over $(h_i, h_j, p_i - p_j, \|p_i - p_j\|)$ masked by existence and averaged over neighbors, and a node-update MLP combining the aggregated message with the previous node state;

width 600 so that IN-Law's 14.25M Law parameters are within 2% of the transformer Law's 14.6M. Every leaf linear layer in IN-Law is named independently of the corresponding transformer layer so that warm-starting from an existing checkpoint (below) cannot accidentally transfer any Law weights by name-and-shape coincidence; we verified by assertion that 100% of `law.*` parameters are freshly initialized when building IN-Law.

Body. Each embodiment (pusher, arm, slider) has its own small MLP encoder mapping that embodiment's own action and proprioception dimensionality into a shared u_{dim}=64 control embedding. A separate ablation (Appendix A.10) tests a single shared Body encoder with a body-identity embedding instead of one encoder per embodiment.

Training recipe. Every model in this paper is trained in the same two stages. Stage 1 (self-distilled settle) trains State, Body, and Law jointly for several thousand steps with a next-state regression loss L_{pred} against the model's own encoder applied to the true next frame, a privileged binding loss L_{bind} and pose loss L_{pose} that use Hungarian matching between predicted slots and ground-truth simulator objects (position, velocity, existence, color id, available only at training time), and a handful of small auxiliary losses (velocity-follow, hit-prediction, time-to-contact, and others) weighted by coefficients in $\mathrm{TrainConfig}$; batch size 8, context length 4, rollout length 8, horizon 12, learning rate $5{\times}10^{-5}$, weight decay 10^{-3}. Stage 2 (counterfactual fine-tune) starts from the Stage 1 checkpoint, freezes the encoder, and trains for 750 steps at learning rate $3{\times}10^{-5}$ with an intervention loss $\lambda_{\mathrm{intervene}}$=0.35 (penalizing the discrepancy between a rolled-out edited state and the true edited-simulator rollout) and a velocity-follow loss $\lambda_{\mathrm{vfollow}}$=0.3 (encouraging the rolled-out velocity to keep tracking the edited value), rollout 8, horizon 10. IN-Law and the C-SWM-objective variant additionally go through a 6000-step Stage 1.5 "settle" stage with the perception frontend frozen, training only the freshly initialized Law module from scratch on the plain prediction and pose losses, before the identical Stage 2 fine-tune; this mirrors how the transformer Law itself was originally matured from a partially-trained checkpoint and keeps the comparison in Section 4.5 fair.

C-SWM objective. The contrastive variant of Stage 1.5 replaces L_{pred} with an energy-based loss computed on each slot's (p, v) features only (we found that including the higher-magnitude learned appearance code z made the energy scale swamp any reasonable margin in a preliminary calibration run). Writing $H(a, b) = \mathrm{mean}_{i \in \mathrm{existing}} \|a_i - b_i\|^2$, the loss is $L_{\mathrm{pos}} = H(\hat{s}, \mathrm{enc}(o_{t+1}))$ for the true next frame and $L_{\mathrm{neg}} = \mathrm{relu}(\mathrm{margin} - H(\hat{s}, \mathrm{enc}(o_{t+1})_{\mathrm{shuffled}}))$ for a batch-shuffled negative, with margin=1.0, following Kipf et al. (2020). A calibration run confirmed healthy training dynamics: L_{pos} dropped from 13.6 to about 0.25 over the first 1000 steps while L_{neg} stayed meaningfully nonzero (0.2 to 0.35) throughout, confirming the contrastive hinge stayed active rather than being trivially satisfied.

Parameter counts. Table 3 gives parameter counts for the model size used throughout this paper ("lewm15," roughly 15M parameters) and for a smaller size used only in early development.

Table 3: Parameter counts by model size. Body parameter count is per embodiment.

	small	lewm15 (used throughout)
LSB (transformer Law)	0.97M (Law 0.83M)	14.7M (Law 14.6M)
CLS	0.34M	15.7M
Body (per embodiment)	∼9K	∼9K

A.3 Evaluation protocol and variance

Early in this project we observed that single-seed point estimates on this benchmark, particularly on the intervention margin E2, can vary enough between nominally identical evaluation runs to flip sign, simply from episode-to-episode stochasticity in the physics rollout (random initial positions, velocities, and object counts). Every result reported anywhere in this paper (main text and appendix) therefore uses a paired, multi-seed protocol: for a given comparison, every model being compared is evaluated on the exact same sequence of evaluation seeds, and we report either the mean with

a standard error, or a paired t-statistic, computed across seeds rather than across episodes within a seed. Seed counts range from 8 to 20 depending on the experiment (noted with each result); ten to twenty was sufficient to make every headline comparison in this paper significant at $p < 0.01$. We used a process-parallel evaluation harness (one evaluation seed per worker process, with each worker's BLAS and OpenMP thread pools forced to a single thread to avoid oversubscription) to make this protocol cheap enough to apply everywhere; a full four-metric, multi-seed sweep for one model typically completes in under a minute on our hardware.

A.4 Replication studies

CLS+v. Because CLS+v is the single most important baseline in this paper (Section 4.2), we reran its entire three-stage pipeline (warm-start, settle, counterfactual fine-tune) from scratch under a second, independent set of run names, with no seed control forcing determinism, and re-evaluated it against the same LSB checkpoint with the same ten-seed sequence (Table 4). CLS+v's absolute numbers move a real amount between the two runs (E1: 0.619 vs. 0.460; E3: 0.408 vs. 0.567), which is itself a useful data point about training-run variance in this baseline, but on the two metrics that most directly probe the counterfactual claim, E1 and E2, LSB wins with the same sign and comparable or larger significance in both independent runs.

Table 4: CLS+v replication: two independent training runs of the full three-stage pipeline, evaluated against the same LSB checkpoint with the same ten-seed sequence both times.

metric	CLS+v run 1	CLS+v run 2	LSB	t (run 1)	t (run 2)
ADE@8 ↓	0.397	0.421	0.450	$+5.4$	$+2.4$
E1 acc. ↑	0.619	0.460	0.719	$+3.6$	$+11.2$
E2 margin ↑	-0.092	-0.056	0.060	$+3.5$	$+3.8$
E3 acc. ↑	0.408	0.567	0.363	-2.4	-8.4

IN-Law. We similarly reran IN-Law's entire three-stage pipeline from scratch, fresh random initialization for the from-scratch Law module, and re-evaluated against the same LSB and CLS checkpoints with the same ten-seed sequence (Table 5). All four anchor metrics replicate to within one standard error across the two independent runs, including IN-Law's standout win over the transformer Law on the contact-pair metric E3, which is the strongest replication result of any single finding in this paper.

Table 5: IN-Law replication: two independent training runs.

metric	run 1	run 2	Δ
ADE@8 ↓	0.456 ± 0.006	0.440 ± 0.008	0.016
E1 acc. ↑	0.542 ± 0.015	0.540 ± 0.014	0.002
E2 margin ↑	-0.087 ± 0.014	-0.090 ± 0.013	0.003
E3 acc. ↑	0.550 ± 0.027	0.577 ± 0.021	0.027

A.5 Cross-embodiment results

Table 6 reports the full STEP/QUERY/INTERVENE/contact-pair comparison on the two-link arm and the sliding paddle, twenty seeds $\times$ 48 episodes. Neither embodiment's LSB checkpoint went through the pushing-hand-specific counterfactual fine-tune, so their E2 margins are negative like the "no counterfactual training" row discussed in Section 4.4; the point of this table is that the size of LSB's advantage over CLS on all four metrics is consistent across embodiments, not that these particular checkpoints have positive intervention margins.

Table 6: Cross-embodiment comparison, twenty seeds × 48 episodes.

body	metric	LSB	CLS	advantage	t
arm	ADE@8 ↓	0.450	0.700	0.250	−29.2
arm	E1 acc. ↑	0.673	0.567	0.106	+6.8
arm	E2 margin ↑	−0.188	−0.458	0.270	+10.8
arm	E3 acc. ↑	0.150	0.074	0.076	+6.5
slider	ADE@8 ↓	0.448	0.622	0.174	−24.7
slider	E1 acc. ↑	0.743	0.521	0.222	+16.7
slider	E2 margin ↑	−0.184	−0.388	0.203	+7.3
slider	E3 acc. ↑	0.198	0.087	0.111	+6.2

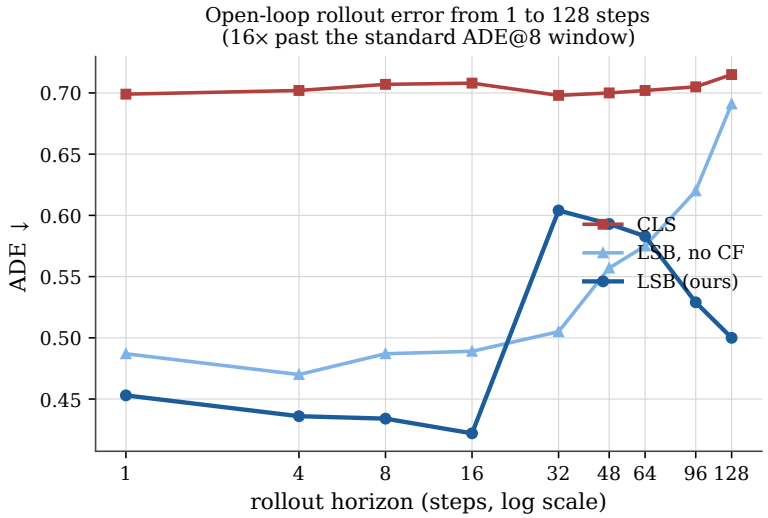

Figure 7: Open-loop rollout error from 1 to 128 steps, sixteen times past the standard training window, zero-shot. LSB's advantage over CLS holds at every horizon and does not shrink at the extremes.

A.6 Object-count extrapolation and long-horizon detail

LSB is trained only on scenes with two or three objects. Figure 8 extends the object count out to six at test time with zero retraining. The naïve way to summarize this, comparing each model's in-distribution-to-out-of-distribution gap, is misleading: LSB's gap is numerically larger than CLS's, but this is a floor-effect artifact of CLS already being close to its own ceiling error in distribution. The comparison that actually matters, absolute ADE at every object count, favors LSB everywhere from one to six objects, in distribution and out of distribution alike, though the relative margin does shrink from roughly 35% at one to three objects to roughly 10% at six.

On the long-horizon sweep in Figure 7, LSB's ADE is not perfectly monotonic in horizon: it is flat through horizon 16, rises somewhat around horizons 32 to 64 (the one place in this paper where the counterfactually fine-tuned checkpoint is briefly worse on raw ADE than the same checkpoint before counterfactual fine-tuning), then falls again and pulls decisively ahead at horizons 96 and 128. One plausible explanation is that the counterfactual fine-tuning losses were optimized at a short rollout and horizon, so the model's rollout behavior is most tightly disciplined in that regime and comparatively less so in the untrained-for middle stretch, while a different property, likely rollout stability from having learned dynamics that at least approximate real physics rather than memorized short-horizon patterns, dominates at the very long end. We report this as an interesting detail rather than a settled mechanism; it does not affect the paper's central long-horizon claim, since LSB still beats CLS at every single horizon tested.

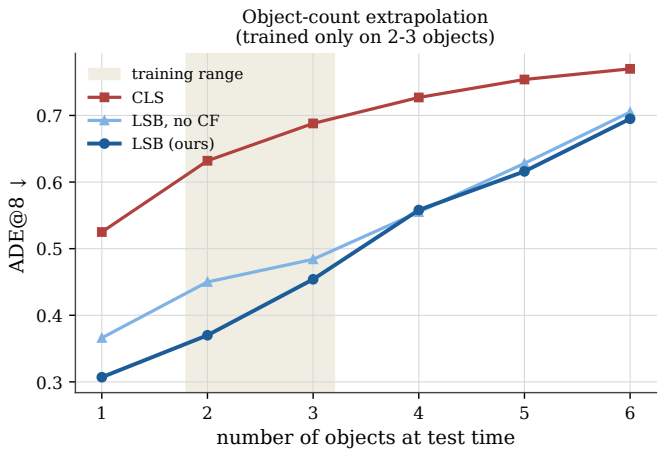

Figure 8: ADE@8 as a function of object count at test time, trained only on two to three objects. LSB's advantage over CLS holds at every object count tested.

A.7 Additional qualitative examples

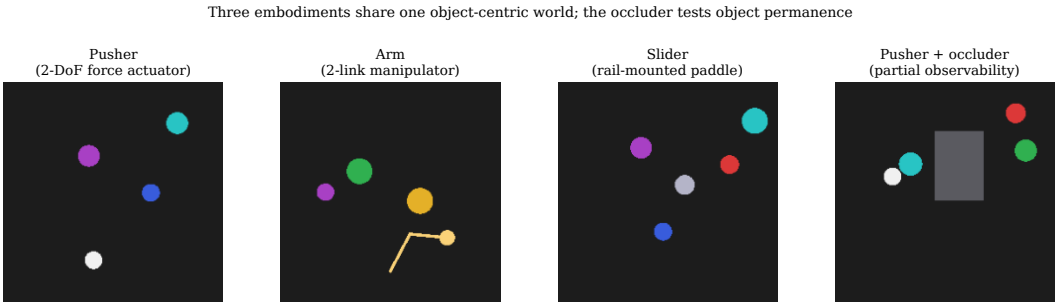

Figure 9: The three embodiments used throughout, sharing one object-centric physical world, plus an occluder example used to test object permanence.

Figure 6 in the main text is one representative intervention episode selected, out of a batch of independently generated candidates, for having a visually legible edited-object trajectory; the underlying numeric comparison (E2 margin, Table 1) is computed over many more episodes than we can show individually. The rendering, trajectory extraction, and figure generation code is entirely automated and takes any evaluation seed as input, so additional examples across embodiments, object counts, and physics regimes are easy to reproduce with the released code.

A.8 CF-training step-efficiency

As a finer-grained look at how quickly each architecture acquires counterfactual capability, we applied the identical counterfactual fine-tuning recipe to LSB and CLS+v from their respective pre-fine-tune checkpoints, checkpointing every 125 steps, two independent training seeds each (Table 7). Both architectures start counterfactual fine-tuning with an unexpectedly high E1, which decays over the 750-step window, an interesting transient-training-dynamics observation in its own right, but LSB's decay flattens out comfortably above chance (0.60 to 0.62) while CLS+v's decay continues below chance, ending at 0.45 to 0.46. LSB never crosses the chance line at any checkpoint in either seed; CLS+v crosses it in both seeds by step 500 and stays there. This is a difference in kind, not only in degree, between the two architectures' behavior under an identical short training budget.

Table 7: E1 accuracy during counterfactual fine-tuning, mean of two training seeds. Chance is 0.5.

CF fine-tune step	LSB E1	CLS+v E1
125	0.716	0.913
250	0.666	0.593
375	0.654	0.528
500	0.585	0.481
625	0.600	0.458
750	0.622	0.454

A.9 Explicit pairwise feature in Law

As a further architectural probe, we tried giving our transformer Law an explicit, structurally provided nearest-neighbor feature (each slot's relative offset and distance to its closest other existing slot), rather than relying on self-attention to discover this feature on its own. Table 8 shows a genuine, if mixed, result: the added feature does not move the intervention margin at the standard evaluation setting, but it significantly improves the contact-pair task E3 ($t = +6.92$ against our default Law) and ADE@8 ($t = -2.88$), at a small cost to E1 ($t = -3.47$), plausibly because the explicit shortcut reduces reliance on the general slot representation that the query task draws on. We did not adopt this variant as our main model because it does not change the paper's central claims, but we report it here as a genuine positive result on the STEP/QUERY side.

Table 8: Adding an explicit nearest-neighbor feature to the transformer Law, twenty seeds $\times$ 48 episodes.

metric	+ pairwise feature	LSB (default Law)	no CF training
ADE@8 $\downarrow$	0.428	0.443	0.476
E1 acc. $\uparrow$	0.669	0.715	0.679
E2 margin $\uparrow$	-0.023	-0.015	-0.227
E3 acc. $\uparrow$	0.517	0.412	0.207

A.10 Shared vs. per-embodiment Body

We also tested whether Body needs a dedicated encoder per embodiment or whether one shared adapter with a body-identity embedding does equally well, jointly training on pusher, arm, and slider from the same pusher-only base with Law and the encoder frozen. Cross-body forecast ADE@8 at eight seeds $\times$ 48 episodes showed no significant difference on any of the three embodiments (pusher $t = -0.43$, arm t=0.91, slider $t = -0.38$). We read this as evidence that the value of the Law/Body split is the existence of an explicit, swappable interface between dynamics and embodiment, rather than the specific choice to give each embodiment its own dedicated weights; a shared implementation behind the same interface works just as well once trained enough.

A.11 SCORE (E4) and place-to-touch (E5) planning results

Table 9 reports two additional, more planning-oriented tasks at reduced seed count (ten seeds, more expensive per episode than the four anchor metrics). E4 (SCORE) asks the model to rank candidate action sequences by predicted outcome quality; LSB with counterfactual fine-tuning shows a significant improvement over the same checkpoint before fine-tuning (t=2.72), matching the pattern seen on E2 and E3. E5 (place-to-touch, a short closed-loop planning success rate) shows the object-centric architecture beating CLS by a large margin regardless of counterfactual fine-tuning (t=6.73 for the pre-fine-tune checkpoint against CLS), but counterfactual fine-tuning itself does not move E5 in either direction ($t = -1.10$, not significant, point estimate even slightly favoring the pre-fine-tune checkpoint).

Table 9: E4 (SCORE) and E5 (place-to-touch), ten seeds.

metric	LSB (CF-trained)	LSB, no CF	CLS
E4 mind ↑	0.158	0.079	0.104
E5 success ↑	0.121	0.142	0.029

A.12 Additional negative and null results

In the interest of completeness we briefly summarize several additional experiments whose outcome was a null or a result that did not survive replication. None of these affect the paper's main claims, and we report them only for transparency.

- **Occlusion robustness.** Forcing the environment's occluder to appear on every episode, rather than the trained-on 15% rate, did not produce a clean or consistent effect on ADE for any model; differences were mostly within one to two standard errors and inconsistent in sign across horizons.
- **Physics compositionality.** We tested whether combining two independent physics perturbations (friction-only, restitution-only) degrades a model by roughly the sum of each perturbation applied alone, which would suggest a compositional understanding of separate physical rules. The single-factor perturbations available in our environment were too mild to move any model's metrics outside of noise, so this comparison was underpowered and we do not draw a conclusion from it.
- **Mass and radius probing.** We attempted to determine whether a model's slot appearance code z linearly encodes an object's mass or radius, neither of which is available to the trainer (only pixels and privileged position/velocity/existence are, per Appendix A.2). Results across two attempts were inconclusive and we do not report a claim either way.
- **Freeze-Law scenario transfer.** An early single-run result suggested that freezing Law while fine-tuning only Body and the encoder on a new physics regime transferred better than fine-tuning everything. This did not survive three subsequent independent replication attempts and we have retracted the claim; it is mentioned here only because it shaped our later, much more careful zero-shot generalization protocol in Section 4.3, which requires no retraining at all and has replicated cleanly across six regimes.