SAME STATE, DIFFERENT FUTURE: LATENT INTERCHANGEABILITY IN LEARNED ACTION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Latent action and transition models are often intended for repeated computation: a latent that decodes correctly may then be consumed by subsequent learned operations. We ask a question narrower than whether rollout error can accumulate: if two latent representations correspond to the same correctly decoded observable state, are they functionally interchangeable for the next learned operation? In controlled D4 models at $d \in \{128, 64\}$, compressed action operators achieve essentially perfect primitive decoded execution yet recover 0/16 withheld two-action products under direct composition. A paired evaluation freezes the model and downstream operator, then substitutes the encoder-produced latent for the action-produced latent of the same correctly decoded observable state. This change alone restores 16/16 withheld products and 64/64 products overall in every registered seed, without retraining. The contrast demonstrates non-interchangeability for the tested downstream operators and evaluation distribution; re-encoding serves only as a diagnostic intervention. Supporting comparisons show that unstructured composers matched in parameter scale fit all observed products but fail withheld products, whereas a deliberately compatible full-state representation composes exactly. A deterministic gridworld reproduces the immediate recursive-use failure and rescue after intermediate replacement as a second controlled setting, not an unseen-state generalization test. Across these two deterministic toy environments, the results suggest that decoded correctness and downstream computational usability can be distinct evaluation targets for latent models intended for recursive use.

1 INTRODUCTION

Learned latent action and transition models are often intended to support continued computation, not merely a single prediction. A planner may apply a learned transition repeatedly to imagine candidate futures; a simulator may propagate an internal state for many steps; and a model-based controller or sequential reasoner may compose several learned operations before decoding an answer. In each case, the output of one learned operation becomes the input to the next. A latent state is therefore useful not only when it decodes to the correct observation once, but also when it remains usable by subsequent learned operators. We distinguish these requirements as primitive executability, representational reusability, and compositional reusability.

The difficulty of learning representations that remain reliable under rollout is well established. Exposure bias describes the mismatch created when a recursively deployed model consumes its own outputs rather than training-distribution inputs (Bengio et al., 2015). PlaNet uses latent overshooting to train multi-step transition priors toward observation-conditioned posteriors and discusses decoded observation overshooting (Hafner et al., 2019); TD-MPC and TD-MPC2 align predicted latents with target-encoder representations for planning (Hansen et al., 2022; 2024). Learned Koopman methods likewise seek representations that evolve within approximately invariant subspaces (Takeishi et al., 2017; Lusch et al., 2018). Recent work analyzes recursive prediction on self-induced states and provenance (Green et al., 2026), representation and transition-model constraints for stable latent dynamics (Huang, 2026), and rollout-decoded reconstruction for free-running latents (Shah & Shrestha, 2026). The latter two are recent preprints; Green et al. is an arXiv paper accepted at an ICML 2026 workshop. Together, this literature establishes the broad one-step/rollout mismatch and motivates consistency and re-encoding remedies. Our question is narrower.

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

We isolate two representations of the same observable state. Let E be an encoder, D a decoder, and M_a the learned operator for action a. After applying a to x, compare the operator-produced latent and the encoder-produced latent

$$z_{\text{op}} = M_a E(x), \qquad z_{\text{enc}} = E(a(x)). \tag{1}$$

Both may correspond to the same correctly decoded next state,

$$\text{disc}(D(z_{\text{op}})) = \text{disc}(D(z_{\text{enc}})) = a(x), \tag{2}$$

without necessarily supporting the same future computation. If two latent representations correspond to the same correctly decoded observable state, are they necessarily interchangeable for the next learned operation? We call them *functionally interchangeable* with respect to a frozen downstream operator M_b when substituting one for the other preserves downstream decoded behavior. Operationally, we test whether $\text{disc}(D(M_b z_{\text{op}})) = \text{disc}(D(M_b z_{\text{enc}}))$ while keeping M_b and every learned parameter fixed. This is a local substitution test for the specified operator and evaluation distribution, not a global equivalence relation over latent states.

In controlled models of the eight symmetries of a square, compressed linear action operators achieve essentially perfect decoded primitive execution but recover 0/16 withheld two-action products under direct latent composition. The central experiment holds the checkpoints, inputs, targets, and downstream operator fixed. In this essentially decoded-correct regime, $M_a E(x)$ can decode to the correct intermediate state while the next operator fails when it consumes that latent; replacing only the intermediate with $E(a(x))$ restores 16/16 withheld products and 64/64 products overall, without retraining. A decode/discretize/re-encode path implements the same-state substitution at evaluation time. Supporting controls bound this finding: two parameter-scale-matched unstructured composers fit all 48 observed products yet recover 0/16 withheld products, whereas a deliberately compatible full-state operator representation composes exactly. A deterministic gridworld then reproduces the immediate one-step/recursive-use separation and rescue after intermediate replacement outside the group benchmark; it is supporting replication rather than the novelty claim.

Our contributions are threefold:

- We operationalize a paired same-state substitution diagnostic that compares decoded-equivalent latents under the same frozen downstream operator.
- We provide a frozen-model substitution experiment in which a correctly decoded operator-produced latent fails under the next operator, while the encoder-produced latent of the same observable state succeeds.
- We combine capacity and observed-fit controls, a full-state structural positive control, and a sequential replication to separate the measured effect from simple undercapacity and artifacts specific to the D4 benchmark.

The evidence is deliberately scoped to fully observed, deterministic, controlled toy systems, mostly linear learned action operators, and three registered seeds. The gridworld center-start set was included in one-step training and is not an unseen-state generalization test. Accordingly, this is a diagnostic and evaluation contribution, not a new rollout-training algorithm, a universal claim about latent interchangeability, or broad validation in real-world world models.

2 METHODS

2.1 PROBLEM SETUP

Let $x \in \mathcal{X}$ be an observable discrete state, $E : \mathcal{X} \to \mathbb{R}^d$ an encoder, $M_a \in \mathbb{R}^{d \times d}$ the learned operator for action a, and D a decoder. Primitive execution and direct composition of a followed by b predict

$$\hat{x}_1 = \text{disc}(D(M_a E(x))), \qquad \hat{x}_2 = \text{disc}(D(M_b M_a E(x))), \tag{3}$$

where disc takes the per-cell argmax. *Primitive executability* is correct decoded single-action behavior. For the paired test, $z_{\text{op}} = M_a E(x)$ is the action-produced latent and $z_{\text{enc}} = E(a(x))$ is the encoder-produced latent. When both decode to the same observable state, they are *functionally interchangeable* for a specified downstream operator if substituting one for the other preserves decoded behavior. This local definition induces *representational reusability* for the paired intermediates and *compositional reusability* for recursive action application; it is not a universal latent equivalence relation.

2.2 D4 Benchmark and Models

The benchmark uses independently sampled 5×5 grids over 10 tokens and the eight square symmetries. Its 8×8 Cayley table uses (a, b) for a followed by b; a preregistered split withholds 16 ordered pairs and designates 48 as seen. Generic/capacity controls receive only the 48 seen pair targets, while operator models train only on primitive triples $(x, a, a(x))$; their pair split is evaluation-only. Independent deterministic streams generate training and evaluation grids, and main comparisons use seeds 0–2.

The full-state positive control acts with learned 250×250 matrices on a 250D cell/token one-hot representation and composes as M_bM_a. Compressed models replace this representation with a learned linear encoder and decoder and use $d \in \{128, 64\}$; the 250D model is the main positive control. The capacity comparison keeps full-state primitive operators frozen and contrasts matrix multiplication with a 501,616-parameter operator composer and a 538,120-parameter pair MLP, both trained on the 48 seen products. These controls match parameter scale, not architecture or optimization. Architecture and optimization details are in Appendix A.

2.3 Evaluation and Same-State Substitution

Cell accuracy measures correct cells; exact accuracy requires a fully correct grid. Cayley recovery counts identified D4 products among 48 seen, 16 withheld, or all 64 pairs. Its test follows the output: capacity controls use target-class prediction (unstructured) or row-wise argmax-permutation matching (structured), while learned-operator evaluations require at least 99% exact grids per pair. These share targets and denominators, not output-space criteria. Figures 1–2 report means and population SD over seeds; Figures 3–4 use stored means and sample SD. Exact counts are per seed; sample counts and settings are in Appendix A.

The central diagnostic loads frozen $d = 128$ and $d = 64$ checkpoints and holds inputs and operators fixed. The four registered path definitions are

$$\begin{aligned} \text{direct:} \quad & D(M_bM_aE(x)), \\ \text{oracle reset:} \quad & D(M_bE(a(x))), \\ \text{decode/re-encode:} \quad & D(M_bE(\mathrm{disc}(D(M_aE(x))))), \\ \text{re-anchor:} \quad & D(M_bP(M_aE(x))), \qquad P(z) = E(\mathrm{disc}(D(z))). \end{aligned} \tag{4}$$

Reported discrete outcomes apply disc to each displayed decoder output. Under this implementation, decode/re-encode and re-anchoring are mathematically identical at the reported two-step endpoint. Figure 3 therefore shows that shared path once rather than presenting duplicate bars; oracle reset remains distinct because it uses the true intermediate state $a(x)$. No operator is retrained. We also report the descriptive distances $\|M_aE(x) - E(a(x))\|_2$ and $\|P(M_aE(x)) - E(a(x))\|_2$; distance neither defines functional interchangeability nor establishes a manifold.

2.4 Sequential Gridworld Replication

The second setting is a deterministic 7×7 grid with one agent, four cardinal actions, and wall-clamped boundary moves. Linear-operator models at $d = 32$ and $d = 16$ train only by one-step cross-entropy on draws covering all 196 state–action transitions, including the nine center positions. Those positions are an evaluation stratum, not unseen states. We evaluate 50 deterministic mixed-action sequences at $H \in \{1, 2, 4, 8\}$ using direct rollout, decode/re-encode after every action, and oracle reset to the true state. Figure 4 reports exact-state accuracy; secondary protocols appear in Appendix B.

3 Results

3.1 Parameter Scale Does Not Explain the Observed Composition Gap

Figure 1 first tests whether parameter scale or observed-pair fit explains the composition gap. All conditions recover 48/48 observed products in every seed. Structured matrix multiplication, with no trainable composition parameters or pair labels, recovers 16/16 withheld products; the 501,616-parameter operator composer and 538,120-parameter pair MLP each recover 0/16. Thus, parameter

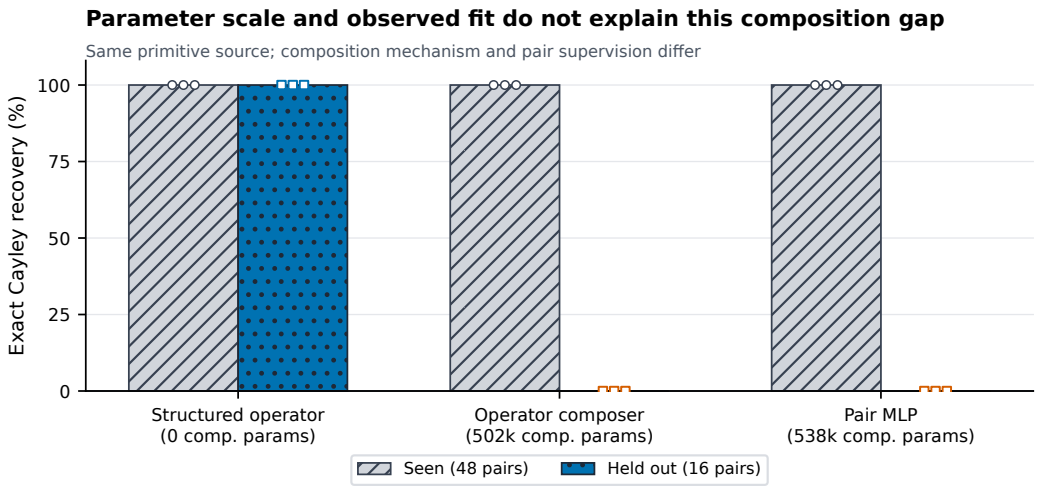

Figure 1: **Parameter scale and observed fit do not explain this composition gap.** Bars show correct D4-product identification (action-class prediction for unstructured controls; row-wise argmax-operator matching for structured multiplication); markers show seeds 0–2. All recover 48/48 seen; structured multiplication recovers 16/16 withheld, while both unstructured controls recover 0/16. This rules out parameter scale and observed fit as complete explanations within this comparison; composition rule, parameterization, and optimization also differ.

scale and observed-table fit do not explain the structured result in this comparison. Architecture and optimization also differ, so this does not imply that capacity is generally irrelevant or that every unstructured composer must fail. Having rejected this simple explanation, we next ask whether learned primitives remain reusable after compression.

3.2 Primitive Correctness Does Not Ensure Recursive Reuse

Figure 2 asks whether decoded primitive correctness is sufficient for direct two-action composition. The $d = 250$ condition is an architecturally distinct, parameter-free full-state identity-representation control; $d = 128$ and $d = 64$ are learned compressed encoder–operator–decoder models. At $d = 250$, primitive cell and exact accuracy, held-out composition cell accuracy, and Cayley recovery are exact (16/16 withheld; 64/64 full). At $d = 128$, primitive cell and exact accuracy remain 100%, but held-out composition cell accuracy is $(10.1793 \pm 0.0834)\%$, with 0/16 withheld and 0/64 full recovery. At $d = 64$, primitive cell accuracy is $(99.9927 \pm 0.0054)\%$ and primitive exact accuracy is $(99.850 \pm 0.134)\%$, yet composition reaches only $(10.1758 \pm 0.4312)\%$, again with 0/16 and 0/64 recovery. This is approximately the 10% uniform per-cell chance level for the 10-token vocabulary. These are population SD over the registered seeds. The 128D/64D failure therefore cannot be explained by material primitive underfitting, although changing dimension also changes the learned representation and is not identified as the cause. This creates the central diagnostic question: does the next operator fail because of the observable intermediate, or because of its latent provenance?

3.3 Same-State Latents Need Not Be Functionally Interchangeable

Figure 3 answers this question with the paired same-state substitution in frozen 128D/64D models. Direct continuation from the correctly decoded operator-produced intermediate obtains $(10.1303 \pm 0.1265)\%$ and $(10.2310 \pm 0.5464)\%$ held-out cell accuracy, respectively, with 0/16 Cayley recovery. The small numerical difference from Section 3.2 reflects the larger 5,000-example evaluation used here rather than a different checkpoint or training condition. Oracle reset supplies $E(a(x))$; decode/re-encode supplies $E(\mathrm{disc}(D(M_a E(x))))$. The registered re-anchor $P(M_a E(x))$ is exactly the latter under the implemented definition of P, so Figure 3 plots that shared path once. Both distinct replacement paths supply an encoder-produced representation of the same observable intermediate to the identical frozen downstream operator. Changing only this intermediate restores 16/16 withheld

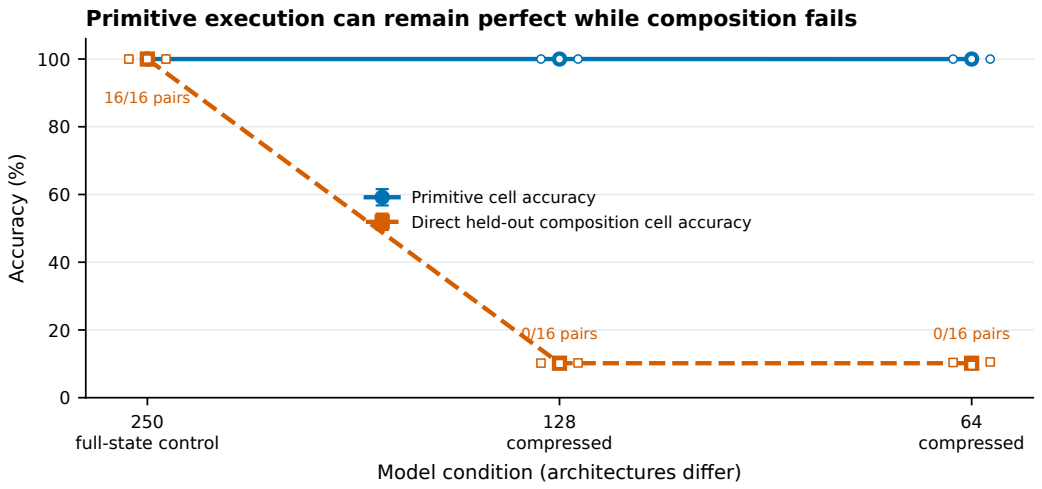

Figure 2: **Near-perfect primitive execution does not imply direct compositional reuse.** Lines compare decoded primitive accuracy with direct held-out composition; markers and error bars show seeds 0–2 and population SD. The horizontal axis denotes categorical model conditions, not an isolated dimension sweep: 250D is the full-state identity-representation control, whereas 128D/64D are learned compressed models. The 250D control recovers 16/16 withheld products. At 128D/64D, primitive execution remains essentially perfect but direct composition recovers 0/16, motivating the same-state test in Figure 3. Dimension alone is not identified as the cause.

and 64/64 full products in every registered seed without retraining. At 128D both replacement paths reach 100% held-out cell accuracy; at 64D their lowest mean is $(99.9839 \pm 0.0089)\%$. Error terms are sample SD.

The descriptive ℓ_2 distance to $E(a(x))$ changes after re-anchoring from 18.3357 to 0.0000 at 128D and from 26.4382 to 0.0105 at 64D. The behavioral substitution—not this coordinate-dependent distance—establishes that the two same-state latents are not functionally interchangeable for the tested next operators and distribution. The substitution is a diagnostic intervention, not a learned correction or proof of a latent manifold.

3.4 Replication in Sequential Rollout

Finally, Figure 4 asks whether the immediate recursive-use failure appears outside D4. It evaluates deterministic gridworld trajectories from the nine center positions, which entered one-step training and are an evaluation stratum rather than unseen states. At 32D and 16D, all paths have 100% exact-state accuracy at horizon 1. Direct rollout is 0% at horizons 2, 4, and 8, while decode/re-encode and oracle reset remain 100% through horizon 8; all displayed sample SD are zero. The immediate recursive-use failure and rescue therefore replicate outside D4, but not in a stochastic, partially observed, or realistic world model.

4 Related Work

Invariant and compositional latent dynamics. Learned Koopman methods seek representations with approximately invariant linear evolution $E(f(x)) \simeq KE(x)$, often combining reconstruction, latent dynamics, multistep prediction, or forward–backward consistency (Takeishi et al., 2017; Lusch et al., 2018; Azencot et al., 2020). Li et al.'s "compositional Koopman operators" instead assemble shared object and interaction blocks across variable physical systems (Li et al., 2020). Thus, invariant subspaces, latent consistency, and compositional linear dynamics are established objectives. They motivate our narrower evaluation question: does decoded primitive success certify enough consistency for the next learned action?

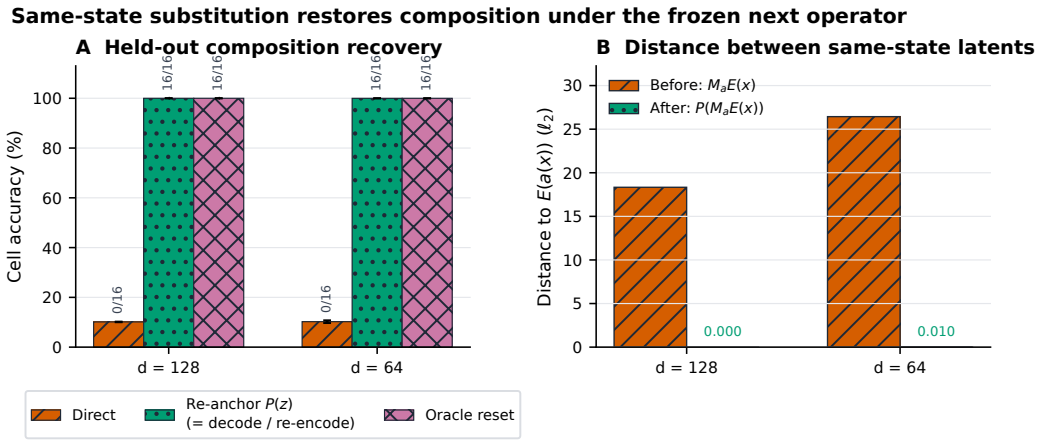

Figure 3: **Same-state substitution isolates latent provenance under an identical frozen downstream operator.** Panel A holds the models and next operator fixed at 128D/64D. Direct continuation from the correctly decoded action-produced latent recovers 0/16 withheld products. Oracle reset uses $E(a(x))$; the re-anchor $P(z)$ is identical here to decode/re-encode and is therefore shown once. Both distinct replacements recover 16/16 in every seed, without retraining (sample SD over seeds 0–2). Panel B reports descriptive ℓ_2 distance to $E(a(x))$; distance is not treated as causal.

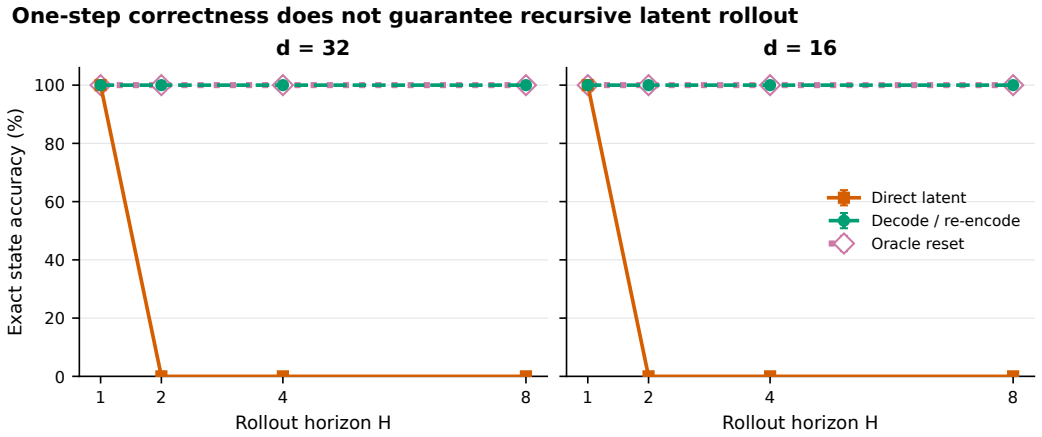

Figure 4: **Supporting replication outside D4.** Exact-state accuracy on mixed-action center-start trajectories is 100% for all paths at horizon 1; at horizons 2, 4, and 8, direct rollout is 0% while decode/re-encode and oracle reset remain 100% (sample SD zero over seeds 0–2). The immediate reuse gap therefore persists beyond Cayley-table scoring. Center positions were included in one-step training, so this is not unseen-state generalization.

Recursive prediction and observation anchoring. Multi-step mismatch is likewise established. PlaNet matches rolled transition priors to observation-informed posteriors and studies decoded observation overshooting (Hafner et al., 2019); SPR predicts future momentum-encoder targets (Schwarzer et al., 2021); and TD-MPC/TD-MPC2 explicitly align predicted next latents with target encoder representations for planning (Hansen et al., 2022; 2024). These objectives are analogous in purpose to enforcing $M_a E(x) \simeq E(a(x))$, which we do not claim as new. Exposure bias more generally concerns training on ground-truth histories but deployment on model outputs (Bengio et al., 2015).

Recent work sharpens this concern. Green et al. show that one-step supervision can underidentify recursive prediction on self-induced states and separate provenance effects (Green et al., 2026); their arXiv paper was accepted at an ICML 2026 workshop. The SJEPA and control-theory preprints

respectively analyze representation and transition-model constraints for stable latent dynamics and prediction error on planner-reachable states and its relation to control performance (Huang, 2026; You et al., 2026). Closest mechanistically, the RDR preprint trains a decoder on free-running rollout latents via decoded reconstruction (Shah & Shrestha, 2026). Together, these works establish the broad one-step/recursive mismatch and related remedies. Our evaluation-only test focuses on a different empirical object: a correctly decoded intermediate paired with the encoder-produced latent of the same observable state under an identical frozen downstream operator. Green et al. study recursive forecasting and provenance under underidentification, whereas RDR changes training to improve long-horizon prediction; here, only the intermediate changes.

Systematic generalization and positioning. SCAN, CFQ, and COGS already show that fitting observed combinations need not yield systematic recombination (Lake & Baroni, 2018; Keysers et al., 2020; Kim & Linzen, 2020). Accordingly, the 48/48 seen versus 0/16 withheld result serves as a capacity and observed-fit control within a comparison whose models also differ in architecture and optimization. The paper's contribution is instead the paired same-state interchangeability diagnostic and its controlled evidence, with structural and capacity controls and a second deterministic setting.

5 Discussion and Limitations

5.1 Interpretation

Decoded correctness and functional interchangeability are distinct evaluation targets in the tested models. In the essentially decoded-correct regime, $z_{\mathrm{op}} = M_a E(x)$ and $z_{\mathrm{enc}} = E(a(x))$ correspond to the same observable intermediate yet behave differently under the same frozen M_b. The paired same-state substitution exposes this difference directly.

This asymmetry is expected from the training support: M_b is optimized on encoder outputs $E(x)$, whereas direct composition supplies $M_a E(x)$. Correct decoding constrains the observable state but does not require $M_a E(x) = E(a(x))$ or place the former on the downstream operator's training support. This provides a mechanism consistent with the intervention; it is not claimed as the only possible cause.

For models intended for recursive use, this distinction motivates complementing one-step accuracy with downstream reuse tests. Functional interchangeability is local: two latents may be interchangeable for one operator and distribution but not another. Nor do geometry, decoded agreement, and downstream behavior necessarily coincide. Euclidean distance is descriptive and coordinate-dependent here; the behavioral substitution carries the interpretation. We identify neither a formal manifold nor a necessary geometric condition. This is an evaluation recommendation, not a theorem or a measurement of planning and policy quality.

The same-state substitution is diagnostic, not a learning method. The frozen downstream operator succeeds when only the action-produced latent is replaced by the encoder-produced latent of the same state, localizing this failure to the intermediate's form or provenance. This does not make re-encoding optimal, universally necessary, or unique. The structural and capacity controls bound task impossibility, parameter scale, and observed-pair fit as complete explanations within their comparisons; gridworld supplies supporting replication outside D4. None establishes realistic planning validity or universal failure of unstructured models.

Latent-consistency objectives and multi-step or rollout-decoded training are plausible ways to reduce this mismatch. Testing such remedies lies outside the frozen diagnostic study: their omission limits claims about mitigation, but it is not an omitted proof obligation for the paired substitution result reported here.

5.2 Limitations

Both environments are deterministic, fully observed discrete toys. They do not test perceptual ambiguity, stochasticity, partial observability, natural observations, planning return, or real-world control. The central result uses primarily linear action operators with simple encoders and decoders and three seeds; prevalence in large nonlinear world models is unknown. All 196 gridworld one-

step transitions entered optimization, so center starts are an evaluation stratum, not unseen-state generalization.

The encoder-produced latent is privileged by construction because downstream operators train on encoder outputs. Same-state substitution therefore diagnoses a train–deployment mismatch but does not prove that encoder anchoring is the only suitable design. Re-encoding may also add computation, depend on decoder accuracy, and discard information; these tradeoffs were not evaluated. The experiments test only specified downstream operators and evaluation distributions, so success or failure need not transfer to other continuations. Finally, the contribution is intentionally narrow relative to closely related work on provenance under recursive prediction, representation and transition-model constraints for stable latent dynamics, rollout-decoded training, and Koopman/latent consistency (Green et al., 2026; Huang, 2026; Shah & Shrestha, 2026; Takeishi et al., 2017; Hansen et al., 2022). Our claim remains limited to the controlled paired diagnostic.

5.3 Future Work

Future work could test same-state interchangeability in nonlinear, stochastic, or partially observed models; measure consequences for planning and control; compare downstream operators and tasks; and evaluate training objectives that encourage interchangeability. None is tested here. Across the present systems, the results suggest that observable correctness and computational usability are distinct evaluation targets. For models intended for continued latent computation, we therefore motivate testing downstream usability rather than inferring it from one-step decoded performance alone.

AI use statement

Generative AI tools assisted with experiment design, method and code implementation, result interpretation and checking, figure preparation, literature search, documentation, and manuscript drafting and editing. The authors reviewed all AI-assisted work, verified code and quantitative claims against frozen artifacts, and take responsibility for the final content, claims, and artifacts.

References

Omri Azencot, N. Benjamin Erichson, Vanessa Lin, and Michael Mahoney. Forecasting sequential data using consistent Koopman autoencoders. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 475–485. PMLR, 2020. URL https://proceedings.mlr.press/v119/azencot20a.html.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/e995f98d56967d946471af29d7bf99f1-Abstract.html.

Riku Green, Zahraa S. Abdallah, and Telmo M. Silva Filho. Exposure bias as epistemic underidentification in recursive forecasting. *arXiv preprint arXiv:2606.12990*, 2026. Accepted at the ICML 2026 EIML workshop.

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.

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

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.

Yongchao Huang. SJEPA: Learning elegant latent dynamics with hybrid symbolic-neural predictors. *arXiv preprint arXiv:2608.04060*, 2026.

Daniel Keysers, Nathanael Schärli, Nathan Scales, Hylke Buisman, Daniel Furrer, Sergii Kashubin, Nikola Momchev, Danila Sinopalnikov, Lukasz Stafiniak, Tibor Tihon, Dmitry Tsarkov, Xiao Wang, Marc van Zee, and Olivier Bousquet. Measuring compositional generalization: A comprehensive method on realistic data. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SygcCnNKwr.

Najoung Kim and Tal Linzen. COGS: A compositional generalization challenge based on semantic interpretation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pp. 9087–9105. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.731. URL https://aclanthology.org/2020.emnlp-main.731/.

Brenden Lake and Marco Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 2873–2882. PMLR, 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

Yunzhu Li, Hao He, Jiajun Wu, Dina Katabi, and Antonio Torralba. Learning compositional Koopman operators for model-based control. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1ldzA4tPr.

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

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

Rishi Shah and Rishav Shrestha. Rollout-decoded reconstruction for long-horizon prediction in latent world models. *arXiv preprint arXiv:2608.25017*, 2026.

Naoya Takeishi, Yoshinobu Kawahara, and Takehisa Yairi. Learning Koopman invariant subspaces for dynamic mode decomposition. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3a835d3215755c435ef4fe9965a3f2a0-Abstract.html.

Hanzhe You, Yonggang Zhang, Maohao Ran, Zhiqin Yang, Zhenyuan Zhang, Wei Xue, Jun Song, Xinmei Tian, and Yike Guo. A control theory of predictability in latent world models. *arXiv preprint arXiv:2607.10362*, 2026.

A Implementation and Training Details

Additional model details. The full-state control applies eight learned 250×250 matrices directly to the 250D one-hot grid. Compressed models use a bias-free $250 \to d$ encoder, eight $d \times d$ operators, and a biased 250-logit decoder; both train only on primitive transitions. The capacity-control operator composer maps each flattened primitive matrix to eight features and applies a 64-unit MLP to the concatenated pair. The pair MLP uses three 512-unit ReLU layers on two one-hot actions.

No condition uses validation-based selection or early stopping; all main comparisons use seeds 0, 1, and 2. Table 1 gives the complete production settings.

Evaluation uses 5,000 fresh examples for the full-state and re-anchoring studies and 500 fresh grids per seed for compressed primitive and composition tests. Unstructured controls train as one full batch of 48 pairs.

For the eight D4 actions—identity, rotations by $90°$, $180°$, and $270°$, horizontal reflection, transpose, vertical reflection, and anti-diagonal reflection, indexed 0 through 7 in that order—the fixed withheld ordered pairs are $(1,2)$, $(3,3)$, $(2,1)$, $(1,1)$, $(1,4)$, $(2,5)$, $(3,6)$, $(0,7)$, $(4,1)$, $(5,2)$, $(6,3)$, $(7,0)$, $(4,5)$, $(5,4)$, $(6,7)$, and $(7,6)$. The other 48 ordered pairs form the observed split.

Table 1: Paper-relevant production training protocols. CE denotes per-cell cross-entropy.

Family	Optimizer	LR / weight decay	Batch	Epochs	Training data and loss
Full-state operator	AdamW	$10^{-3}/10^{-5}$	256	200	50k primitive transitions; CE
Unstructured controls	Adam	$10^{-3}/0$	48	200	all 48 seen pair labels; action CE
Compressed operator	Adam	$10^{-3}/10^{-4}$	256	200	50k primitive transitions; CE
Gridworld operator	AdamW	$10^{-3}/10^{-4}$	64	150	9,800 draws with replacement from 196 transitions; CE

B Supplemental Evaluation Details

Composition metrics apply the primitive definitions to $b(a(x))$, pooled over selected pairs and grids. Capacity controls predict the composed action class; the structured condition compares the row-wise argmax of $M_b M_a$ with the target operator. Gridworld exact-state accuracy requires every cell to match the deterministic target.

The gridworld represents each cell by a two-way one-hot vector (empty or agent) and flattens the 7×7 grid to 98 features. It uses four $d \times d$ operators and a biased 98-logit decoder. Each cardinal move clamps the corresponding coordinate to $[0, 6]$; an outward move at a boundary is therefore a self-transition. Training draws 9,800 samples with replacement from all 196 one-step state–action transitions. Evaluation has four repeated-action and 50 deterministic mixed-action length-eight sequences. Figure 4 uses the nine center starts; the other 40 positions and repeated-action suite are secondary diagnostics.