BUILDING STATE, USING STATE:
REPRESENTATION AND UTILIZATION IN MULTI-STEP TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-step tasks require a model to do two things well: maintain the state needed for future decisions and use that state when producing actions or answers. We study these problems separately in language models and partially observed reinforcement-learning agents. In a controlled maze, map-related information becomes linearly decodable before strong task performance, and matched belief swaps show that this internal map content affects behavior. Yet a representation-targeted auxiliary that raises decodability leaves tight-horizon success essentially unchanged (0.093 to 0.094), while policy-level supervision raises it to 0.497. Craftax shows the same asymmetry at larger scale: recurrent state is necessary for strong performance and contains hidden spatial information, but substantially increasing position decodability does not improve reward. We then ask whether this diagnosis can be used to improve the models. It can. After informative state has formed, interventions aimed at the state-to-behavior mapping improve downstream performance in both domains. Post-convergence efficiency pressure raises Craftax score from 15.09 to 16.56, and latent-guided distillation raises low-data answer accuracy from 0.86 to 0.94 in the language setting. These results suggest that multi-step learning can be limited by state utilization even after useful internal state has already been learned.

1 INTRODUCTION

Success on a multi-step task depends on more than remembering the past. The model must compress its history into a useful internal state, then make decisions that actually depend on that state. These two requirements are often treated as one because both are evaluated through the same final reward or accuracy. They need not fail together.

Consider a navigation agent that has already inferred where an unseen wall gap lies. Poor navigation could still arise because the policy does not use that information effectively. The same issue appears in a simpler language setting: a model may track a running counter in its hidden state yet remain unreliable when asked to report the value. In both cases, low task performance is compatible with a useful internal representation. The remaining error sits downstream of representation formation.

This distinction changes what one would try to improve. Better state learning is a sensible response when the relevant information is missing. It is less obviously useful when the information is already available and the policy or readout fails to exploit it. Neither task performance nor probe accuracy resolves the ambiguity on its own. Probe accuracy only establishes that a variable can be recovered by the chosen decoder, not that the model uses that code for its own computation (Hewitt & Liang, 2019; Belinkov, 2022; Elazar et al., 2021; Canby et al., 2025). Related work has also found latent knowledge or world-state information that remains recoverable when language-model outputs are unfaithful (Burns et al., 2023; Feng et al., 2025). We ask how this gap develops during learning, whether representation and use can be changed separately, and whether the distinction points to a useful training intervention.

We study three settings. The first is a partially observed maze in which the hidden map is simple enough to measure throughout training and to manipulate through recurrent-state interventions. The second is Craftax (Matthews et al., 2024), where memory supports long-horizon behavior in a much richer RL environment. The third is counter tracking in decoder-only language models, where the latent state is known exactly and can be separated from an explicit written state. We also test

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.

pretrained Qwen2.5 models (Yang et al., 2024), with behavioral replications in Phi-3.5 (Abdin et al., 2024) and DeepSeek-R1-Distill-Llama (DeepSeek-AI, 2025).

Across these settings, the same pattern appears. Useful state can become available before behavior extracts its full value. In the maze, map-related decodability rises before strong success and before an intervention-based measure of map use. In the counter task, the correct state is almost perfectly decodable even when answer accuracy is low. The separation also survives direct intervention. Swapping maze beliefs changes behavior, so the recurrent state is not merely correlated with the task. However, making a measured aspect of that state easier to decode has almost no effect on difficult behavior, whereas policy-level supervision does. Craftax gives a second example: a position-prediction auxiliary greatly improves position decodability without improving reward.

The final question is whether this is only a diagnostic observation. We find that it is not. Once useful state is present, changing the training pressure on the mapping from state to behavior can improve the original task. In RL, generic efficiency pressure improves performance after convergence, including in Craftax. In language, an already-learned latent state can supervise a more data-efficient answer readout. The losses differ because the output interfaces differ, but the intervention principle does not depend on a task-specific world-model architecture or a hand-coded solution procedure. We therefore use *task-agnostic* to describe the principle, not a single loss shared verbatim across domains.

Contributions. The paper makes three empirical claims. Internal task state and effective use of that state can develop on different timescales. Representation-targeted and behavior-targeted interventions can move these quantities separately. Finally, targeting utilization after state formation can improve downstream performance in both RL and language. The controlled maze supplies the causal testbed; Craftax and pretrained language models test how far the pattern carries beyond it.

2 Measuring state and its use

We use *internal task state* to mean information accumulated from previous observations or tokens that remains useful later in the sequence. In navigation, this includes spatial information about an unseen environment and is naturally related to a learned world model. In the language experiments, the state is deliberately simpler, such as a running counter or an intermediate arithmetic value. We do not assume that the two domains implement this state in the same way. The common question is functional: what information survives from the past, and how much does later behavior depend on it?

Representation. We measure representation primarily with held-out linear probes fitted to frozen hidden states. A high score means that the target variable is accessible to that decoder from the chosen representation. It does not show that the model itself reads the same linear directions.

Utilization. Use is measured separately. In the maze, we compare policy actions with reference actions and intervene on recurrent beliefs while matching observable decision points. In language models, we perturb explicit intermediate state and measure the effect on later predictions. We also compare objectives that improve state decodability with objectives that act on the downstream policy or readout.

Tasks. The controlled maze is a 7×7 partially observed navigation problem with a hidden wall gap. The agent sees its position, the goal, and a local 3×3 wall window, but not the full map. We train a causal transformer policy with PPO (Schulman et al., 2017). Craftax-Symbolic presents only a local view and scalar statistics, with no absolute position, and requires memory for exploration and long-horizon control (Matthews et al., 2024). The controlled language task applies $+1/-1$ operations to a mod-10 counter under next-token prediction. Pretrained models are tested on the same state-tracking structure and on coherent arithmetic-trace corruptions. Full protocols are in the appendix.

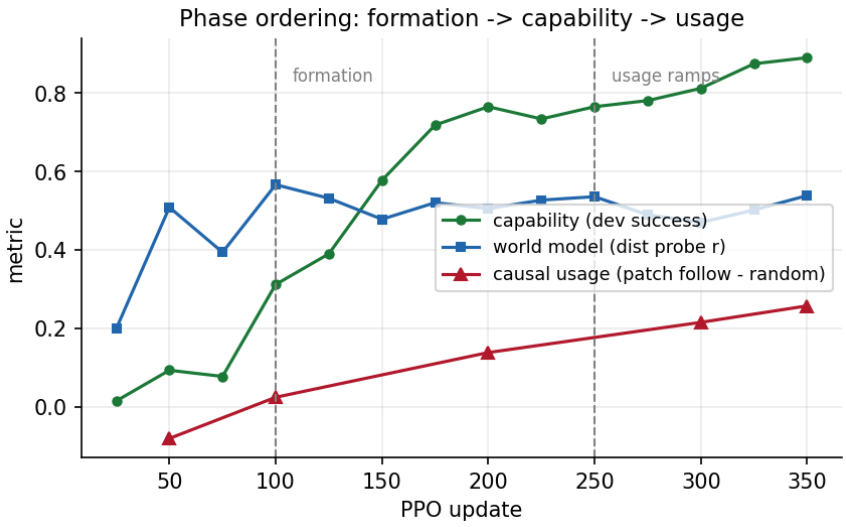


Figure 1: **Map-related information appears before strong behavioral use.** Linear decodability rises early in maze training. Task success and the matched activation-patching metric improve later. The three curves use different quantities, so only their temporal ordering should be compared.

3 State can form before behavior uses it well

3.1 Training dynamics in the controlled maze

The maze lets us compare representation, task performance, and causal use throughout training. A linear distance-to-goal probe crosses the prespecified threshold $r \geq 0.4$ at update 75 ± 42 across five seeds. Development success crosses 0.5 later, at update 155 ± 19. We also track an activation-patching measure: a belief from another maze is inserted at a matched decision point, and we ask whether the action shifts toward the action appropriate for that other maze. This effect is close to its random-pairing control early in training and grows later (Fig. 1). The thresholds are only operational summaries, but their ordering is consistent across seeds. Map-related information becomes available before the policy uses it reliably.

3.2 A similar gap in language-model state tracking

We next use a setting in which the latent variable is known exactly. A small decoder-only transformer receives sequences of $+1/-1$ operations on a mod-10 counter. Depending on the condition, intermediate counter values are written into the sequence, and the final answer appears either in every training example or in only 10% of them. The final counter is highly linearly decodable from an operations-only hidden state in all four conditions (0.94–1.00), but answer accuracy depends strongly on the training interface. With rare answers and no written state, accuracy is only 0.34 despite probe accuracy of 0.96. Writing the intermediate state raises answer accuracy to 0.87. Frequent answer supervision raises it to 0.99.

training condition	counter probe	answer accuracy
state written, answer frequent	1.00	0.99
state written, answer rare	0.98	0.87
no written state, answer frequent	1.00	0.99
no written state, answer rare	0.96	0.34

The point of this toy problem is narrow. It does not show that written reasoning is generally unnecessary. It shows that, when a state variable is already recoverable from the hidden representation, the final answer can still be limited by how that state is exposed to the output layer. Written state tokens provide an additional autoregressive route from the maintained state to the answer.

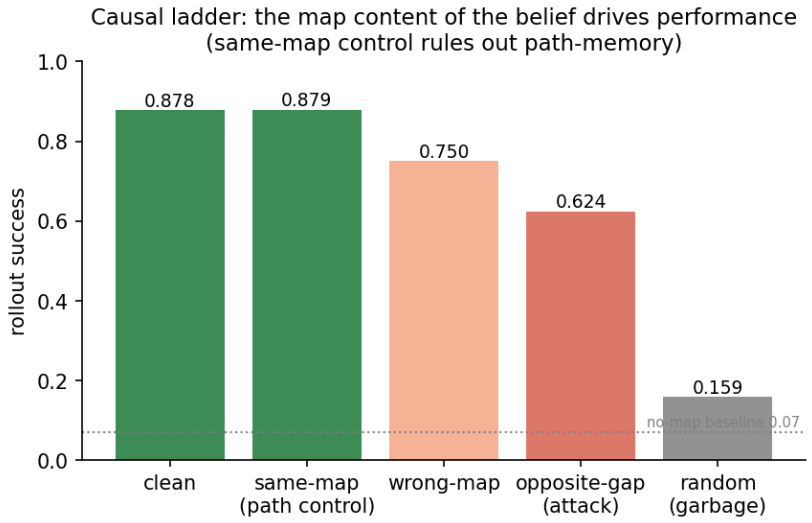


Figure 2: **Maze behavior depends on map content in the recurrent state.** Swapping in a belief from another path on the same map leaves performance unchanged. Wrong-map and targeted opposite-gap beliefs reduce success, and random recurrent state is strongly destructive. This rollout-level protocol produces larger effects than the tighter matched-state test because the false belief is preserved for longer.

3.3 Hidden spatial state in Craftax

Craftax tests the same question in a harder environment. Under our partially observed interface, the agent sees a local terrain window and scalar statistics but not its absolute position. A GRU policy scores 15.09 (95% CI [14.85, 15.34]) across five training seeds, compared with 9.75 [9.59, 9.91] for a memoryless MLP given the same observation. Resetting the trained GRU state at every step reduces the score to 2.22 [2.06, 2.38]. Whatever the recurrent state contains, the policy depends on it heavily.

Linear probes recover hidden spatial variables from that state, including player position and direction to a unique late-game feature. Position R^2 varies across checkpoints and probe protocols, so we report the standardized comparison used for the representation intervention below and tabulate the remaining values in the appendix. The important fact here is simply that spatial information is present before we add any explicit position supervision.

4 Representation and utilization can be pulled apart

Finding useful information in a hidden state is not enough to establish that the policy depends on that information. We therefore intervene on the maze belief directly, then compare training objectives that target representation quality with objectives that target behavior.

4.1 Map content has a causal effect on the maze policy

At 2,876 matched decision points, position, goal, and trajectory length are held fixed while the hidden map differs. Swapping in a belief from another path through the same map leaves success unchanged (0.964 clean versus 0.979 for the same-map control). A belief from a different map lowers success to 0.948. Choosing a belief whose hidden wall gap is on the opposite side lowers success further, to 0.901. Correct observations resume after the swap, so the policy can repair the injected error. A rollout-level intervention that preserves the false belief for longer produces larger effects (Fig. 2). The same-map control matters here: recurrent-state perturbation by itself is not what hurts the policy. The map content does.

The causal role of the full belief should not be identified with the particular directions found by a linear probe. A probe predicts a reference optimal action from the belief with 0.649 accuracy, whereas

Table 1: **Representation-targeted and policy-targeted training produce different behavioral effects.** The representation auxiliary has little effect at tight horizons. Belief-optimal policy supervision accounts for most of the gain.

training arm	h=12	h=15	h=18	h=60
baseline	0.093	0.216	0.366	0.822
+ representation auxiliary	0.094	0.219	0.393	0.870
+ belief-optimal policy supervision	**0.497**	**0.658**	0.714	0.827
+ both	0.479	0.644	**0.717**	0.837

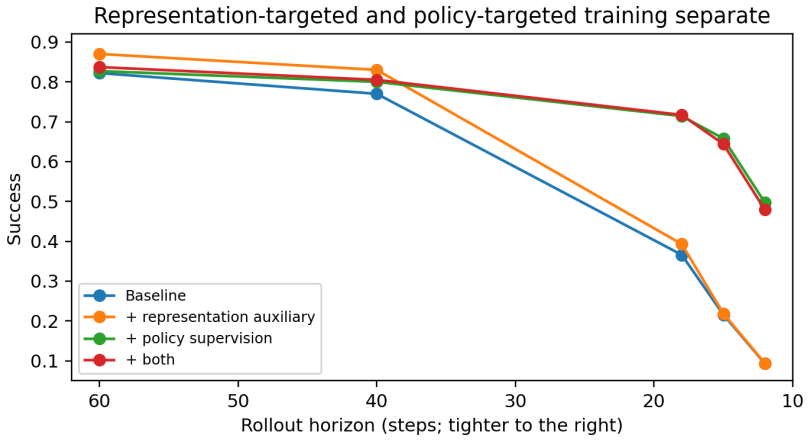


Figure 3: **Improving the measured representation is not enough to recover difficult behavior.** The representation auxiliary raises distance decodability from 0.44 to 0.60 in the corresponding comparison, yet tight-horizon success is almost unchanged. Policy supervision produces the large behavioral gain. Because that supervision also changes rollouts and state visitation, we treat it as a policy-level utilization intervention rather than a pure readout intervention.

the learned policy agrees with that action on 0.462 of the same states. Iterative nullspace erasure (Ravfogel et al., 2020; Elazar et al., 2021) reduces distance decodability from 0.625 to 0.173 after eight iterations without reducing success; matched random subspaces perturb the logits more. This says something specific: the easiest linear distance code is not a necessary localized control direction under this intervention. It does not say that the recurrent map state is irrelevant. The belief swaps show the opposite. This distinction is also consistent with recent work showing that causal probing methods trade off completeness and selectivity (Canby et al., 2025).

4.2 Better decodability does not by itself fix the policy

The strongest separation comes from a 2×2 maze experiment. We cross a next-view prediction auxiliary, which improves state decodability, with belief-optimal policy supervision during PPO. Evaluation then tightens the rollout horizon so that inefficient decisions are increasingly costly.

The numbers make the dissociation sharp. The representation auxiliary raises distance-to-goal decodability from 0.44 to 0.60, but h=12 success moves from 0.093 to 0.094. Policy supervision raises h=12 success to 0.497 without requiring the auxiliary. This does not imply that representation quality is unimportant in general. It shows that, once a useful state exists, improving one measurable property of that state may leave the main behavioral bottleneck untouched.

Craftax gives a second version of the same result. An auxiliary head predicts player position and direction to the necromancer. Held-out position decoding rises from roughly 0.44–0.50 to 0.886–0.893 across evaluation seeds, while raw task score remains essentially 15.09. A large gain in linear accessibility is therefore not sufficient to improve this policy either.

4.3 WRITTEN STATE CAN BECOME THE BEHAVIORALLY PRIVILEGED CHANNEL IN LANGUAGE MODELS

The language experiments expose the separation through the model's input-output channel rather than through a recurrent-state swap. In the toy counter model, shifting every written intermediate state by a coherent $+\delta$ shifts the final answer by roughly the same amount. Qwen2.5-3B behaves similarly: answer shifts are 0.99, 1.98, 2.99, and 5.01 for $\delta = 1, 2, 3, 5$. The operation sequence is unchanged. Only the written intermediate state is corrupted.

This result fits with prior interventions on chain-of-thought, which find that final answers can depend strongly on written reasoning in some settings and weakly in others (Lanham et al., 2023), and with evidence that generated rationales need not identify the true causes of a prediction (Turpin et al., 2023). Our claim is narrower. In a state-tracking task where the counter can be recovered from operations-only activations, the explicit written state can still dominate the final answer. We do not extend that claim to tasks in which the written trace may itself perform essential computation. The GSM8K analysis in the appendix is included partly to make that boundary explicit.

5 IMPROVING UTILIZATION IMPROVES DOWNSTREAM PERFORMANCE

The previous experiments suggest a simple training rule. Once a useful internal state exists, the next optimization target need not be a richer state representation. It may be the mapping from that state to behavior. We test this idea in RL and language. The objectives differ across domains, but both operate downstream of state formation. This is the sense in which the intervention principle is task agnostic.

5.1 RL: POST-CONVERGENCE REWARD PRESSURE IMPROVES BEHAVIOR

We first return to the maze. Starting from a trained policy, we fine-tune with increasing per-step penalties. The environment and architecture are unchanged, and map-related probe quality is already high. At h=12, success is 0.043 under the smallest control penalty, then rises to 0.128 at penalty 0.03 and 0.262 at 0.1. A penalty of 0.3 is too large and performance falls again. The distance probe does not rise monotonically with the behavioral improvement (Appendix Fig. 7). The gain therefore does not require a corresponding improvement in this measure of representation quality.

We apply the same efficiency pressure to a converged Craftax GRU. Fine-tuning for 200 updates with a per-step cost, while evaluating with the original unpenalized reward, raises mean score from 15.09 to 16.56 across five paired seeds. The mean paired difference is $+1.47$ (95% CI $[+1.20, +1.74]$, paired $t = 9.68$, $p = 0.0006$), and all five seeds improve. This is downstream training applied to a recurrent representation that is already in place; by contrast, the explicit position-decoding auxiliary did not improve reward.

The Craftax result supports the intervention principle but not a stronger mechanistic claim. We have not yet measured, on the same pre/post checkpoints, whether the fine-tuned policy becomes more sensitive to a particular spatial code. We therefore report an improvement in performance after state formation, not proof that a specific decoded map feature mediates the gain.

5.2 LANGUAGE: LATENT STATE IMPROVES A LOW-DATA ANSWER READOUT

For the language analogue, we train a new answer readout on operations-only hidden states. With limited labeled data, the readout is trained either from scratch or with a KL term toward the distribution produced by the latent counter probe. The base model is frozen, and its counter representation is already near ceiling. The intervention changes how that existing state is mapped to the answer rather than teaching the model a new counter representation.

At 32 labeled examples, latent-guided distillation reaches 0.94 answer accuracy, compared with 0.86 for a readout trained from scratch. Both reach 1.0 by 128 labels. This experiment is intentionally small, but it serves a specific role: the same state-versus-use diagnosis suggests a productive intervention in a next-token model as well as in an RL policy.

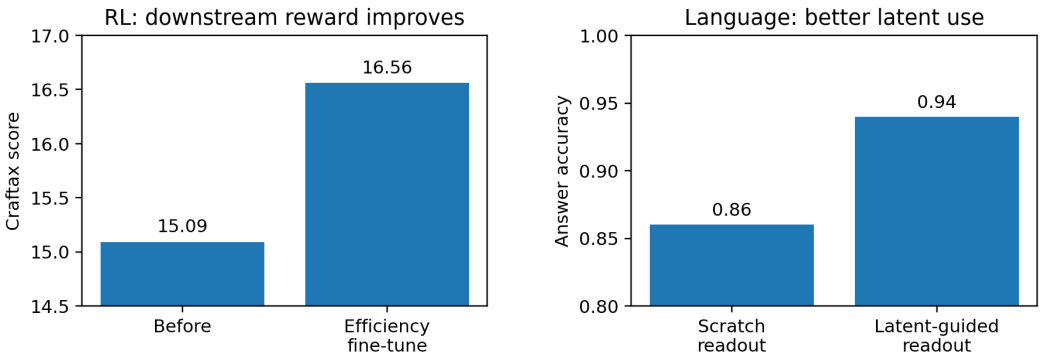


Figure 4: **Intervening on state use improves downstream performance in both domains.** Left: post-convergence efficiency pressure improves the original, unpenalized Craftax score. Right: latent-guided distillation improves answer accuracy when only 32 labeled counter examples are available. The losses are different, but both target the mapping from an existing internal state to behavior.

These interventions move the paper beyond diagnosis. In the maze, representation-targeted and policy-targeted training produce very different behavioral effects. In Craftax, extra position supervision improves decodability but not reward, whereas downstream efficiency pressure improves reward. In language, latent state provides a useful teacher for the answer mapping. The common lesson is operational: when a model already carries useful state, training can sometimes gain more by exploiting that state than by making it easier to decode.

6 Related work

Internal state and world models. Sequence models trained on Othello and chess acquire board-state representations that can be decoded and, in several cases, edited to steer predictions (Li et al., 2023; Nanda et al., 2023; Karvonen, 2024). Spies et al. (Spies et al., 2024) identify causally relevant world-model features in maze-solving transformers. Feng et al. (Feng et al., 2025) recover propositional world states that remain faithful in several settings where LM outputs are not, and Burns et al. (Burns et al., 2023) recover latent knowledge without using model outputs. Our experiments start from the same broad observation that behavior can understate what is available internally, then focus on learning dynamics and intervention: when does useful state appear, when does behavior begin to use it, and what training signal improves the latter?

Probing and causal use. Probe accuracy depends on the probe and does not by itself establish that the probed variable drives behavior (Hewitt & Liang, 2019; Belinkov, 2022). INLP and amnesic probing make this concern testable by removing linearly accessible information and measuring the behavioral effect (Ravfogel et al., 2020; Elazar et al., 2021). Canby et al. (Canby et al., 2025) show that causal probing methods can trade off completeness against selectivity. We therefore use probes as accessibility measurements and reserve stronger claims for matched belief interventions or training manipulations.

Written reasoning and latent computation. Interventions on chain-of-thought show that final answers can condition strongly on the written trace, but that dependence varies by task and model (Lanham et al., 2023). Generated explanations can also omit or misstate causal influences on the answer (Turpin et al., 2023). Our counter experiments isolate a simpler case in which the relevant state variable is independently known and recoverable from operations-only activations. This lets us ask whether the written state becomes the channel that controls the answer without treating all chain-of-thought as a state readout.

Belief and world models in RL. Model-based RL explicitly learns predictive latent dynamics and trains behavior using those models (Ha & Schmidhuber, 2018; Hafner et al., 2021; 2025). Work on partially observable RL likewise studies recurrent or predictive representations that summarize

history for control (Hefny et al., 2018; Wang et al., 2023). Our agents are not built with a separately specified model-and-planner interface. We ask whether, even under end-to-end reward training, the quality of the recurrent state and the quality of its behavioral use remain distinguishable bottlenecks.

7 Discussion and limitations

The experiments point to a practical decomposition of failure in multi-step tasks. Sometimes the model has not formed the state needed for future decisions. Sometimes useful state is already there and the policy or readout does not extract enough value from it. Reward, accuracy, and probe scores each blur this distinction in a different way. The maze interventions show why it matters: improving a measured representation and improving behavior are not interchangeable operations.

Our use of "task agnostic" is deliberately limited. We do not present one universal objective that is copied unchanged from RL to language. The commonality is the intervention target. Efficiency pressure in RL and latent-guided distillation in language both leave the base state-learning problem alone and put additional pressure on how existing state reaches behavior. A stronger result would use one identical objective across both domains.

There are several other limitations. Linear decodability is only one view of representation quality, so the failure of one representation auxiliary does not imply that every possible representation intervention would fail. Belief-optimal supervision in the maze changes state visitation as well as local action choice. The Craftax efficiency experiment lacks a matched pre/post causal measure of dependence on spatial state, which is why we do not attribute the reward gain to a specific map code. Finally, the counter tasks are not a general model of mathematical reasoning. On GSM8K, a problem-only probe for answer magnitude is at chance in our analysis, which is consistent with the written trajectory performing computation rather than merely exposing a simple answer variable that was already present.

These limitations leave the central distinction intact. A weak output is not evidence that useful state is absent, and a strong probe is not evidence that the state is being used well. The useful question is which side of that divide currently limits the system. When state is already informative, the experiments here show that training the route from state to behavior can improve downstream performance.

8 Conclusion

Models solving multi-step tasks must maintain information from the past and turn it into future decisions. Our experiments show that these abilities can separate during learning. Useful task state may appear before behavior makes good use of it, and making that state easier to decode need not improve performance. The separation is not only diagnostic. After informative state has formed, interventions aimed at its downstream use improve performance in both reinforcement learning and language. This suggests a simple distinction for training multi-step systems: before asking the model to learn a better state, check whether the state it already has is being used well.

Reproducibility statement

The appendix specifies the task definitions, prompts, architectures, optimization settings, probe protocols, interventions, seed counts, and additional ablations used in the experiments. Together, these sections document the settings used for the reported training and evaluation runs.

AI use statement

Generative AI tools were used during drafting, language editing, and literature discovery. The authors reviewed the final manuscript, checked the cited sources, and take responsibility for the claims, analyses, and reported results. All experimental results in the paper come from the authors' research code.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

Marc E. Canby, Adam Davies, Chirag Rastogi, and Julia Hockenmaier. How reliable are causal probing interventions? In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics*, pp. 857–878. Association for Computational Linguistics, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Yanai Elazar, Shauli Ravfogel, Alon Jacovi, and Yoav Goldberg. Amnesic probing: Behavioral explanation with amnesic counterfactuals. *Transactions of the Association for Computational Linguistics*, 9:160–175, 2021.

Jiahai Feng, Stuart Russell, and Jacob Steinhardt. Monitoring latent world states in language models with propositional probes. In *International Conference on Learning Representations*, 2025.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Danijar Hafner, Timothy Lillicrap, Mohammad Norouzi, and Jimmy Ba. Mastering atari with discrete world models. In *International Conference on Learning Representations*, 2021.

Danijar Hafner, Jurgis Pasukonis, Jimmy Ba, and Timothy Lillicrap. Mastering diverse control tasks through world models. *Nature*, 640(8059):647–653, 2025. doi: 10.1038/s41586-025-08744-2.

Ahmed Hefny, Zita Marinho, Wen Sun, Siddhartha Srinivasa, and Geoffrey Gordon. Recurrent predictive state policy networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1949–1958, 2018.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pp. 2733–2743. Association for Computational Linguistics, 2019.

Adam Karvonen. Emergent world models and latent variable estimation in chess-playing language models. In *Conference on Language Modeling*, 2024.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *International Conference on Learning Representations*, 2023.

Michael Matthews, Michael Beukman, Benjamin Ellis, Mikayel Samvelyan, Matthew Thomas Jackson, Samuel Coward, and Jakob Nicolaus Foerster. Craftax: A lightning-fast benchmark for open-ended reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35104–35137, 2024.

Neel Nanda, Andrew Lee, and Martin Wattenberg. Emergent linear representations in world models of self-supervised sequence models. In *Proceedings of the 6th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 16–30. Association for Computational Linguistics, 2023.

Shauli Ravfogel, Yanai Elazar, Hila Gonen, Michael Twiton, and Yoav Goldberg. Null it out: Guarding protected attributes by iterative nullspace projection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. Technical Report arXiv:1707.06347, 2017.

Alex F. Spies, William Edwards, Michael I. Ivanitskiy, Adrians Skapars, Tilman Räuker, Katsumi Inoue, Alessandra Russo, and Murray Shanahan. Transformers use causal world models in maze-solving tasks. *arXiv preprint arXiv:2412.11867*, 2024.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Andrew Wang, Andrew C. Li, Toryn Q. Klassen, Rodrigo Toro Icarte, and Sheila A. McIlraith. Learning belief representations for partially observable deep rl. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 35970–35988, 2023.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

A Language-model experiments

A.1 Toy counter model

Task. Sequences contain 6–15 $+1/-1$ operations on a mod-10 counter. Depending on the condition, a token of the form "state=N" is written every third operation. A final "ans=N" token appears in either 100% or 10% of training sequences. The four conditions cross the presence of written state with frequent versus rare answer supervision.

Model and training. We train a three-layer GPT-style decoder with width 96 and four attention heads using next-token cross-entropy. Optimization uses AdamW, learning rate 3×10^{-4}, cosine decay, batch size 256, and 3,000–6,000 steps. We run two seeds per 2×2 cell. Reported values are means over the final checkpoints. In the rare-answer, no-written-state condition, answer accuracy remains low from step 1,500 through 6,000.

Probes. Linear classifiers are fit on last-token hidden states from training sequences and evaluated on held-out sequences. For the operations-only probe, all explicit state tokens are removed. This probe measures whether the final counter is linearly recoverable from the hidden state induced by the operations alone.

Written-state intervention. Every explicit "state=N" token is replaced by "state=(N+δ) mod 10" while the operation sequence is unchanged. For $\delta \in \{1, 2, 3, 5, 7\}$, the mean answer shift is 1.38, 2.00, 2.96, 5.10, and 6.75. Figure 5 shows the dose response.

Low-data answer readout. We train a linear answer readout on operations-only hidden states using N labeled examples, either from scratch or with a KL regularizer toward the latent probe distribution. At $N = 32$, the regularized readout reaches 0.94 accuracy, compared with 0.86 from scratch. Both reach 1.0 by $N = 128$. This experiment tests whether an existing latent decoder can provide useful supervision; it is not intended as a general LM fine-tuning result.

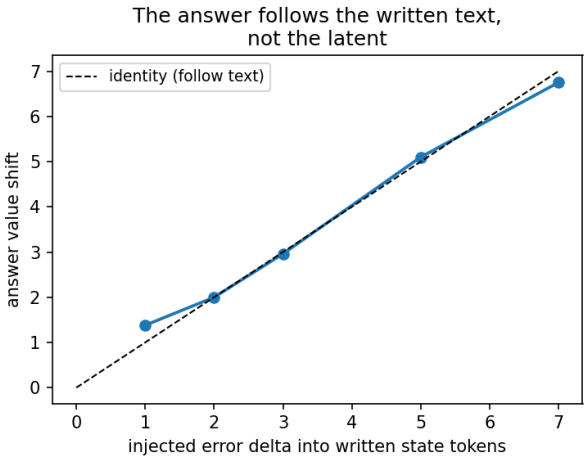


Figure 5: Toy counter model: shifting every written state by $+\delta$ shifts the final answer by approximately the same amount. The operation sequence itself is unchanged.

A.2 Pretrained language models

Models. The primary experiments use Qwen2.5-1.5B/3B-Instruct (Yang et al., 2024) in bf16 without fine-tuning. Behavioral replications use Phi-3.5-mini-instruct (Abdin et al., 2024) and DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI, 2025).

Counter format. A prompt gives a mod-10 operation sequence and, in the written-state condition, intermediate values after each operation. Linear probes use operations-only prompts in which the intermediate values are removed. On Qwen2.5, counter decoding from operations-only hidden state reaches 0.93–0.98 depending on model/layer/evaluation setting (chance 0.10), while the model answers correctly 0.96 of the time with a clean written trace.

Trace corruption. We coherently shift every written counter state by $+\delta$ while keeping the operations fixed. For Qwen2.5-3B, the final answer shifts by 0.99, 1.98, 2.99, and 5.01 for $\delta = 1, 2, 3, 5$ (three seeds). Phi-3.5 and DeepSeek-R1-Distill-Llama show the same qualitative trace-following effect, although the strength of linear counter decoding is substantially lower in those models. We therefore claim replication of the behavioral intervention, not replication of Qwen's probe magnitude.

Arithmetic cascade. In a four-term addition format, we add $+\delta$ to the result of the first written step and recompute each later written step from the corrupted value so that the trace remains internally coherent but wrong. Qwen2.5 shifts the final answer by 1.00, 2.00, and 3.00 for $\delta = 1, 2, 3$; the $\delta = 5$ condition produces a smaller mean shift of 4.76 as self-correction begins. DeepSeek replicates the exact $\delta = 1, 3$ shifts on a format it solves reliably. This experiment establishes dependence on the written computation; it does not establish that the correct final answer was separately represented before the trace was written.

A removed control from the main text. We exclude an operations-only corruption control from the main evidence. Once written states are stripped, clean and corrupted prompts can become identical, making the comparison uninformative about causal robustness of the latent representation. Any future version of this control should preserve a nontrivial clean-versus-corrupt latent difference and use one consistent probe protocol.

A.3 GSM8K boundary analysis

We evaluate Qwen2.5-3B and 7B-Instruct on 600 sampled GSM8K test problems each with the prompt: "Solve the problem step by step. Keep the reasoning concise. End with exactly one line: '#### N' where N is the numeric answer." Generation is greedy with a 1,280-token limit.

For the 3B model, overall answer accuracy is 0.810. Among 114 errors, 108 final answers equal the last number written in the preceding reasoning (0.947; Wilson 95% CI [0.890, 0.976]). The corresponding fraction for the 7B model is 72% under the completion-filtered analysis. This is a descriptive property of the generated traces and, by itself, does not establish a representation-utilization gap.

A linear probe on the problem-only hidden state for a five-bin answer-magnitude label is at chance (0.211 vs. 0.20). In contrast to the counter task, our probe therefore does not identify a comparable pre-existing answer variable before the written computation. We use this result to limit the scope of the state-tracking interpretation. It does not support a sharp general boundary between "state" and "computation."

B CONTROLLED MAZE

B.1 TASK AND AGENT

The environment is a 7×7 grid containing a horizontal wall with one gap. The gap is at least one column from both start and goal. At each step, the agent observes its current position, the goal position, and a local 3×3 wall window; the full map is hidden. The baseline policy is a three-layer, four-head causal transformer with width 96 and linear policy/value heads. It is trained using PPO (clip 0.2, GAE $\lambda = 0.95$, $\gamma = 0.99$, learning rate 5×10^{-4}, four epochs) with reward $+1$ at the goal, -0.01 per step, and a small count-based exploration bonus. The baseline has no auxiliary world-model loss. Additional width-48, width-192, GRU, two-gate, and key-door variants are used for robustness checks.

Simple reactive baselines perform poorly (Euclidean-greedy 5–8%; wall-following 0%). These baselines are evidence that the task is difficult without memory, not a formal impossibility result for every memoryless policy.

B.2 EMERGENCE AND PHASE ORDERING

Across 350 PPO updates, development success rises from 0.016 to 0.891 and test success reaches 0.771. A no-map baseline reaches only 7–11%. Three independently trained seeds achieve test success 0.77/0.73/0.74. A distance-to-goal probe rises from approximately $r = 0.20$ to 0.57 by update 100 and then plateaus; unseen-map occupancy is decoded at 0.91 against a 0.77 all-free floor, and position at 0.97.

Using the thresholded metrics reported in the main paper, distance-probe formation occurs at update 75 ± 42 and development success ≥ 0.5 at 155 ± 19. The activation-patching usage metric rises later. These thresholds are operational summaries rather than evidence of discrete learning phases.

B.3 CAUSAL BELIEF INTERVENTIONS

History removal. Replacing the accumulated belief with a current-observation-only state reduces success from 79–88% to 0%.

Matched activation patching. For 1,167 pairs matched on position, goal, and trajectory length but drawn from different maps, the original policy matches its own map's reference optimal action on 58.1% of decisions. After receiving the other map's belief, it matches the other map's reference action on 54.1%, compared with 28.3% under random pairing.

Matched rollout interventions. On 2,876 matched decision points (grouped four per maze), clean success is 0.964. Swapping a belief from a different trajectory on the same map gives 0.979; swapping a different-map belief gives 0.948; the targeted opposite-gap intervention gives 0.901. Because the observation stream remains correct, the policy can partially recover after a wrong belief.

Rollout-level causal ladder. A separate, less tightly matched rollout intervention produces the values shown in Fig. 6: clean 0.878, same-map path control 0.879, wrong-map 0.750, opposite-

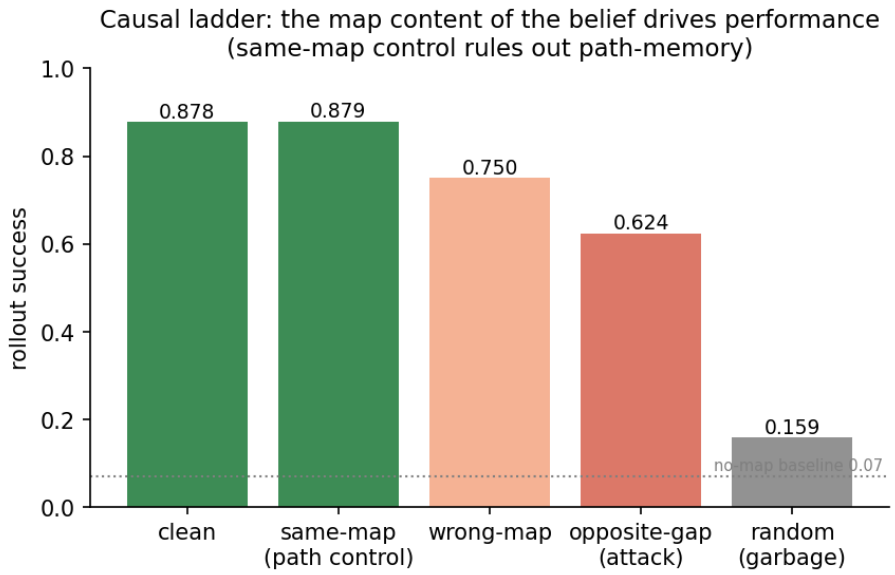


Figure 6: Rollout-level belief intervention. Same-map swaps preserve performance, while wrong-map and targeted opposite-gap beliefs reduce success. This protocol is distinct from the matched-state analysis reported in the main text.

gap attack 0.624, and random garbage belief 0.159. These numbers should not be mixed with the matched-state effect sizes above.

B.4 Probe directions versus behavior

A linear probe predicts the reference optimal action from the belief with accuracy 0.649 (majority 0.462). Against a belief-optimal action target, the corresponding values are 0.611 and 0.412. Freezing the transformer and retraining only the policy/value heads yields 0.656 probe accuracy and 0.452 policy agreement, so the discrepancy is not removed by fitting a fresh linear head under the same policy objective.

The distance-probe direction is nearly orthogonal to the optimal-action probe subspace (observed maximum cosine in the 1.2th percentile of a 500-sample random-subspace null). Iterative concept erasure reduces distance decodability from 0.625 to 0.173 after eight iterations while success remains unchanged; random subspaces of the same dimensionality perturb logits 4–20 times more. Under these interventions, the easiest linear distance code is therefore not a necessary localized control direction. This does not make the broader map representation behaviorally irrelevant; the belief-swap experiments show that it matters.

B.5 Representation auxiliary and belief-optimal policy supervision

The 2×2 experiment crosses a next-view prediction auxiliary with belief-optimal policy supervision. The auxiliary raises distance decodability from 0.44 to 0.60 in the reported comparison. Tight-horizon success values are reproduced in Table 1. Because policy supervision changes rollouts and state visitation as well as local action choice, we do not interpret it as a pure readout intervention.

B.6 Post-convergence efficiency fine-tuning

Starting from the trained baseline, we fine-tune three seeds with per-step penalties. At h=12, success is 0.043 for penalty 0.01, 0.128 for 0.03, 0.262 for 0.1, and 0.097 for 0.3. The distance-probe metric remains in a similar range and does not increase monotonically with the behavior gain (Fig. 7). We therefore describe the result as improved efficient behavior without a corresponding improvement in this probe, rather than claiming that representation quality is exactly constant.

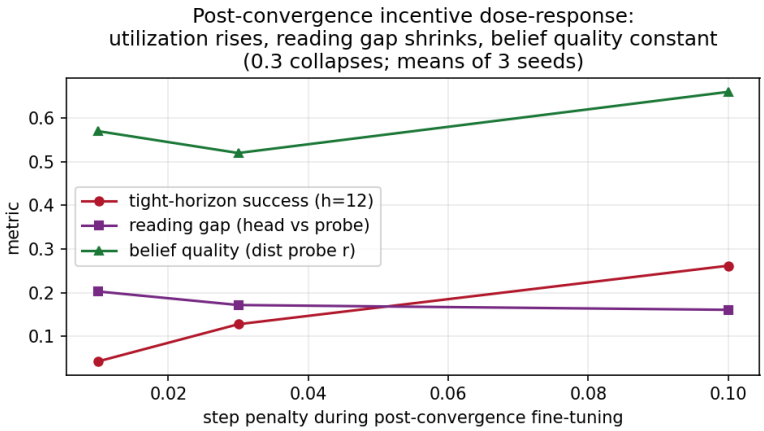


Figure 7: Post-convergence maze fine-tuning. Moderate step penalties improve tight-horizon success; an overly large penalty collapses performance. The distance-probe metric does not track the behavioral improvement monotonically.

B.7 Additional maze variants

A two-layer GRU shows the same qualitative emergence of map-related probe information. Removing the exploration bonus still produces a distance probe of approximately 0.46. A two-gate variant has a larger gap between reference-action probe accuracy and policy agreement, and a key-door variant provides an extreme example in which a probe for hidden state reaches 0.4–0.6 (chance 0.02) while task capability remains near zero. Width sweeps (48/96/192) increase reference-action probe accuracy while policy agreement changes much less. We keep these as robustness checks because the direction of capacity trends is not consistent enough across domains to support a general scaling claim.

C Craftax

Environment. We use Craftax-Symbolic (Matthews et al., 2024) with the default reward and a 17-action subset. The observation is a local 9×9 one-hot terrain window plus six scalar statistics (health, food, hunger, energy, level, and drink). Absolute position is not included.

Policies and training. The recurrent agent uses a GRU with hidden size 128; the memoryless comparison uses an MLP on the same observation. PPO uses clip 0.2, GAE $\lambda = 0.95$, $\gamma = 0.99$, learning rate 2.5×10^{-4}, four epochs, eight minibatches, entropy coefficient 0.01, and gradient norm 0.5. Training uses 1,024 environments for 128 steps over 600 updates. Headline GRU/MLP, recurrent-state reset, and efficiency-fine-tuning claims use five training seeds. The representation auxiliary uses three.

Evaluation. Fresh-rollout score is measured with 512 environments and 256 steps over three evaluation seeds. Step-budget evaluations cap rollout length between 32 and 256 steps.

Spatial probes. A post-hoc linear regression predicts normalized player position from GRU state using a 70/30 train/test split over collected beliefs. The reported R^2 values come from different checkpoints and probe/evaluation protocols. The baseline post-hoc estimate is around 0.5, a targeted-edit checkpoint reaches 0.807, and the auxiliary-trained model's own held-out decoding reaches 0.886–0.893. These values should be read within their stated protocols rather than as directly interchangeable estimates.

Representation auxiliary. An additional head predicts player position and direction to the unique necromancer tile, with squared-error loss added to PPO at weight 1.0. Across three seeds, this raises

held-out position decoding from approximately 0.44–0.50 to 0.886–0.893 while raw Craftax score remains 15.09. The main paper uses this result only to establish that increasing measured linear accessibility is not sufficient to improve the policy.

Recurrent-state necessity. Across five seeds, GRU score is 15.09 [14.85,15.34] and MLP score is 9.75 [9.59,9.91]. Resetting GRU state at every step gives 2.22 [2.06,2.38], a reduction of approximately 85%. This intervention establishes that accumulated recurrent state is load-bearing. It does not isolate which decoded variable within that state is responsible.

Targeted position edit. On the auxiliary checkpoint with post-hoc position $R^2 = 0.807$, the minimum hidden-state edit that makes the linear probe read a shifted position changes 0.8% of actions, compared with 2.4% for a same-norm random edit. We treat this as evidence that the probe direction itself is not a dominant action-control direction, not as evidence that position information in the full recurrent state is unused.

Efficiency fine-tuning. The trained GRU is fine-tuned for 200 updates with per-step reward penalties of 0.03, 0.1, or 0.3 applied to the learning signal; evaluation reports the original unpenalized reward. The selected moderate condition raises mean score from 15.09 to 16.56 across five paired seeds (difference $+1.47$, 95% CI $[+1.20, +1.74]$, paired $t = 9.68$, $p = 0.0006$). All five paired differences are positive. An overly large penalty degrades performance. A matched pre/post utilization measure on these checkpoints would be needed to attribute the gain to increased use of a particular spatial representation.

Design controls. Three controls affected the final benchmark design. First, BFS distance to a resource is poorly approximated by linear position on a 48×48 map ($R^2 \approx 0.02$), so the maze's distance-regression probe does not transfer cleanly. Second, reward shaping for approach-to-target behavior provides a local compass that a memoryless policy can exploit, obscuring the role of memory; the reported results therefore use the unshaped task. Third, under full observability the memory advantage largely disappears, confirming that partial observability is central to the setup.

D DETERMINISTIC-LATENT SEQUENCE-MODEL CONTROL

We train a small GPT on next-token prediction of random self-avoiding walks. The latent task state is the visited set, which is deterministic from the walk history. The model predicts valid steps at 99.9% compared with 21.5% for a random-neighbor baseline, and local legality is linearly decodable at 0.88. Erasing the top linear probe directions leaves validity essentially unchanged. This control reinforces a methodological point used in the main paper: a variable may be decodable and behaviorally important at the level of the full hidden state even when the particular directions selected by a linear probe are not the causal code used by the model.