DEEP REINFORCEMENT LEARNING AGENTS REPRESENT REWARD-DEPENDENT BELIEF GEOMETRY

Anonymous authors
Paper under double-blind review

ABSTRACT

Bayesian beliefs are sufficient statistics of the history of a partially observed process, and classical separation results show that optimal control can be attained using these beliefs. However, it is unclear whether belief-like representations actually emerge in model-free deep reinforcement learning (RL), which trains policies not for prediction but to maximize reward. To address this question, we study transformer policies trained by policy gradients and fit affine probes into their residual streams, scanning for Bayesian posteriors. Experiments show that reward alone suffices to recover beliefs, even in the absence of latent-state supervision or auxiliary losses. However, we also find that the full belief geometry is not always what the agent learns. Using a family of tasks, in which the reward–action structure requires progressively finer distinctions between latent states, we show that distinctions that are irrelevant for control and reward are selectively lost, and that the learned representation is better described by the beliefs of a coarse-grained POMDP than by the full environmental posterior. Changing the reward leads to the same effect: only reward-relevant latent factors are accounted for within the belief geometry. Taken together, these results suggest a learning-dependent version of the separation principle, in which the environment determines how beliefs are updated while the reward determines which beliefs are worth maintaining.

1 INTRODUCTION

Under what conditions do agents build world models comprised of latent environmental variables that affect their task performance? Classic theoretical results state that optimal policies in partially observed scenarios can be decomposed into two modules: an inference module that computes Bayesian beliefs over the environment's state, and a control module that plans based on those beliefs. Concretely, the *separation principle* in control theory shows that the optimal controller in idealised settings decomposes into Kalman filtering and standard control over those estimates (Åström, 1965; Striebel, 1965; Wonham, 1968). A similar decomposition turns a partially observable Markov decision process (POMDP) into a fully observable belief MDP (Kaelbling et al., 1998).

Despite their elegance, these theoretical results only show that the optimal policy *can* be factored into inference and control modules, but they do not guarantee that agents *must* implement policies in this manner. Indeed, a neural policy trained end-to-end via stochastic gradient descent is shaped to generate actions that maximise reward, not to perform Bayesian inference. Even a unique optimal policy does not imply a unique internal realization — the same history-to-action map can be implemented by a filter and controller, by a large look-up table, or by a distributed computation with no legible intermediate posterior.

Recent work has found belief geometry in transformers trained for next-token prediction (Shai et al., 2024), and control-relevant belief information in recurrent networks (Lambrechts et al., 2022; Hennig et al., 2023). These findings motivate two questions that drive this research:

1. Do Bayesian belief-like representations emerge spontaneously from reward-only policy optimization as implemented by deep RL methods?
2. If belief-like representations emerge, what aspects of the environment are they tracking?

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.

To address these questions, we study transformers trained to maximize reward through reinforcement learning and compare their residual-stream representations with full and task-quotiented Bayesian beliefs. The main results in this work are:

- We extend techniques for belief probing to make it applicable to POMDPs, using action-conditioned Bayesian targets.
- We show that the belief geometry becomes linearly encoded in transformer policies trained to maximize reward, without incentivizing inference via dedicated modules or loss terms.
- We show that learned coarse-grainings inherit symmetries imposed by action and reward.

We interpret these observations in light of the separation principle (Åström, 1965; Striebel, 1965; Wonham, 1968): environmental dynamics constrain how beliefs update, while the control objective helps select the state space over which a bounded learned agent maintains beliefs.

2 From belief-state optimality to learned beliefs

2.1 Beliefs in partially observable control

Consider a POMDP with latent states $S_t \in \mathcal{S}$, observations $O_t \in \mathcal{O}$, actions $A_t \in \mathcal{A}$, transition kernel $\kappa(s' \mid s, a)$, observation kernel $\nu(o \mid s)$, reward $r(s, a)$, and discount $\gamma \in [0, 1)$. A history-dependent policy selects actions according to

$$A_t \sim \pi(\cdot \mid H_t), \qquad H_t = (O_0, A_0, \ldots, A_{t-1}, O_t). \tag{1}$$

The Bayesian belief is the posterior $b_t(s) = \Pr(S_t = s \mid H_t)$, which updates after executing $A_t = a$ and observing $O_{t+1} = o$ as

$$b_{t+1}(s') \propto \nu(o \mid s') \sum_{s \in \mathcal{S}} \kappa(s' \mid s, a) b_t(s). \tag{2}$$

This recursive update $b_{t+1} = \phi(b_t, a, o)$ implies that b_{t+1} depends on H_t only through b_t, so one can build a *belief MDP* with b_t as states, A_t as actions, and reward $\bar{r}(b, a) = \sum_s b(s) r(s, a)$. Its optimal policy pulls back to an optimal policy for the original POMDP (Kaelbling et al., 1998).

We use belief states of transducer environments (Barnett & Crutchfield, 2015; Boyd et al., 2016), which generalize POMDPs (Rosas et al., 2025). As shown in App. A, the belief states of transducers function like those of the hidden Markov models (HMMs) that are currently being used to reveal belief geometry in the residual stream of transformers trained on next-token prediction (Shai et al., 2024; 2026; Piotrowski et al., 2025).

2.2 A separation principle is not a learning guarantee

The above results, together with classic results from optimal control theory (Åström, 1965; Striebel, 1965; Wonham, 1968), motivate an inference–control factorization of optimal policies:

$$H_t \xrightarrow{\text{inference}} b_t \xrightarrow{\text{control}} \hat{\pi}(\cdot \mid b_t). \tag{3}$$

Here, the inference block turns the history H_t into the belief b_t via Bayesian filtering (Särkkä & Svensson, 2023), depending on the environment dynamics and observation model but not on the reward structure. Conversely, the control block uses the belief b_t to generate a distribution over actions that maximise reward.

Importantly, the above factorisation identifies the structure of one possible realization of an optimal policy, but does not guarantee that a given optimiser will build it that way. Indeed, a given history-to-action policy can be implemented in multiple ways.

Despite not being unique, there are reasons to expect that deep learning systems may realize the factored presentation of the policy due to their inductive bias towards simplicity. Indeed, recent work has shown that gradient-trained deep learning architectures preferentially learn maps that factor through a low-dimensional intermediate representation (Jacot, 2023; Huh et al., 2023); when several circuits implement the same input–output map, training tends to converge on the most efficient ones (Varma et al., 2023); and transformers specifically favor short-program solutions (Bhattamishra et al., 2023; Zhou et al., 2024).

2.3 TASK-SUFFICIENT BELIEFS

An agent may learn how to calculate the belief b_t because it supports general prediction of environmental states, and hence can be used to pursue any possible goal. However, the full belief b_t is in general not necessary for building a policy that maximises a given reward (Castro et al., 2009). Thus, a resource-constrained agent pursuing a single objective may compress histories to only preserve reward-relevant information.

Following Castro et al. (2009), consider a coarse-graining of beliefs into blocks given by a mapping ψ. A classic condition for coarse-grained beliefs to preserve optimal control is bisimulation (Givan et al., 2003), which guarantees that states in the same block have equal rewards and identical transition mass into every block under every action:

$$\psi(b) = \psi(b') \Rightarrow \begin{cases} \bar{r}(b,a) = \bar{r}(b',a), \\ \sum_{x:\psi(x)=\bar{b}} \kappa(x \mid b,a) = \sum_{x:\psi(x)=\bar{b}} \kappa(x \mid b',a), \quad \forall \bar{b}, a. \end{cases} \tag{4}$$

These conditions connect controlled lumpability of beliefs to reward-respecting state abstraction (Li et al., 2006). Note that $\bar{b}_t$ admits an autonomous Bayesian update only if the observation model also respects the partition.

2.4 HYPOTHESES

Our investigations are motivated by the conjecture that, for learners with resource limitations or simplicity biases, learned representations tend toward *recursively updatable belief representations at the coarsest resolution sufficient for optimal control*, rather than merely grouping histories that warrant the same current action. Such a representation $z_t = f(H_t)$ must support both an update $z_{t+1} = F(z_t, A_t, O_{t+1})$ and an optimal policy $\pi^*(\cdot \mid H_t) = g(\cdot \mid z_t)$. Histories requiring the same action now may still need to remain distinct if subsequent observations make their optimal actions diverge. This offers a common interpretation of the emergence of rich internal belief representations in models not trained directly on it. Our experiments interrogate a subset of the hypotheses implied by this conjecture and only where belief geometry is linearly decodable. These hypotheses are:

H1: Reward-only emergence.
Rich and accurate Bayesian belief geometry over an environment's hidden states can emerge under reward-only RL, without a probability-calibrating prediction loss.

H2: Task-dependent coarsening.
Reward–action structure favors the coarsening of belief representations along task-irrelevant distinctions. We test two routes to this coarsening:

H2a: State–action symmetries.
Control symmetries favor coarsening of the belief representation itself, rather than only the controller's (actor's) readout of a full belief.

H2b: Reward dependence.
Changing reward alone can select which belief factors are learned, even with fixed observations, transition dynamics, and available actions.

3 BELIEF GEOMETRY EMERGES IN REINFORCEMENT LEARNING

For all experiments in this paper, we follow Shai et al. (2024) by building training environments using variants of `Mess3`: a three-state, three-symbol hidden Markov process whose Bayesian belief states have a nontrivial geometry (for details see Appendix B).

3.1 EXTENDING PROBING UNDER CLOSED-LOOP CONTROL

To study the extent to which RL agents build beliefs, we extend the methodology introduced by Shai et al. (2024), who found that the geometry of belief states can be uncovered by fitting a linear probe from the transformer's activations when trained on next-token prediction. A crucial difference between

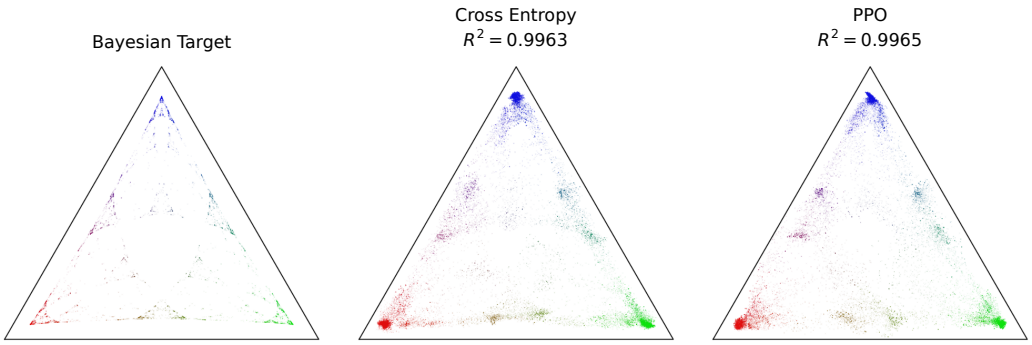


Figure 1: **Target and decoded beliefs.** (**Left**) Exact Bayesian predictive-state beliefs on held-out histories. (**Middle & Right**) Affine decodings of those beliefs from post-final-layer-normalization residual-stream activations of transformers trained with cross-entropy and PPO, respectively. Corresponding points share target-derived colors.

the predictive setting and closed-loop RL is that the same observation can imply different beliefs depending on what the agent did. To account for this, we extend the probe's target construction using an action-conditioned transducer, in which each executed action selects the dynamics through which the next observation updates the belief. By using this, the probe targets both the agent's observations and its actions. This is implemented using the bidirectional mixed-state matrix (BDMSM) recursion for forward-belief updating introduced by Rosas et al. (2025). Technical details about this can be found in Appendix A.

3.2 Decoding beliefs from the residual stream

Using the probing techniques described above, we first asked whether a transformer trained to maximize reward in a simple task develops a representation of the Bayesian posterior, despite receiving no belief supervision. To investigate this question, we trained policies with proximal policy optimization (PPO) (Schulman et al., 2017), applying a reward of 1 if its action matches the next token and 0 otherwise (see Appendix B). Thus, in this first experiment actions affect the reward but do not feed back into the environment, making this setting equivalent to a contextual bandit (Foster & Rakhlin, 2020).

Our results show that the residual stream of an agent trained in this environment linearly represents Bayesian beliefs with high accuracy — comparable to that of a control minimizing cross-entropy loss. Figure 1 shows the decoded beliefs and R^2 probe error. These results provide a first example of belief representation arising from a policy trained via PPO, supporting H1.

The emergence of belief geometry was demonstrated in predictive settings by (Shai et al., 2024; Piotrowski et al., 2025; Shai et al., 2026), who found that transformers trained on next-token prediction developed rich belief geometry over the underlying hidden states. However, please note that minimizing cross-entropy incentivizes exact estimates of the next-token probability, while maximizing expected correctness (as we do here) requires only selecting a most probable token.

4 Reconstructed beliefs correspond to control requirements

To further investigate how beliefs emerge in agents trained via deep RL using this probe methodology, we introduce the 'Belief-Quotient Ladder': three experiments arranged by increasing requirements for state-action resolution. Each experiment is the same variant of `Mess3` but where the agent's actions modify the transition probabilities between states in different ways, giving rise to a POMDP. For all experiments, states 1 and 2 have symmetric baseline transition dynamics but different emission distributions. State 3 is the sole reward state, yielding reward 1 on every time step it is occupied; states 1 and 2 yield zero. There are three discrete actions: Noop, Tilt X, and Tilt Y. Noop preserves the baseline dynamics, while the effects of Tilt X and Tilt Y are the only changes across the three experimental conditions, named below.

Table 1: A hierarchy of control requirements

	One-step reward maximizing action		
Condition	State 1	State 2	State 3
Trivial Quotient	Tilt X	Tilt X	Tilt X
Reward-State Quotient	Tilt X	Tilt X	Noop
Full-State Quotient	Tilt X	Tilt Y	Noop

For each condition, we train a transformer agent using complete-episode REINFORCE with an annealed entropy bonus, no belief target, and no predictive auxiliary loss, across 15 seeds. All conditions use the same 64-dimensional, four-block transformer and a target budget of 10 million environment steps. Appendix C specifies the environment, architecture, optimization, and probe construction.

4.1 Full-belief decodability follows control demands

All 15 seeds in each condition achieved stable reward-state occupancy well above baseline. Yet, as the distinctions required for control become coarser, a linear probe fit to the full Bayesian belief associated with the HMM's mixed-state presentation becomes less accurate, as shown in Figure 2(a). Full-belief decodability is therefore highest under Full-State Quotient and lowest under Trivial Quotient, despite similarly strong behavior. Shuffling coherent token–action histories or replacing them with random valid inputs drives probe error back to the no-information level, showing that the on-policy fits depend on the structured history presented to the network. The simplex view in Figure 2(c) shows that aggregate probe accuracy does not fully describe the recovered geometry. The exact targets contain fine branches within each token-associated region, whereas the decoded beliefs merge much of this structure, most strongly under Trivial and Reward-State Quotient. This mismatch motivates asking whether the full belief is the right target.

4.2 Reward-relevant state distinctions emerge selectively

Next, we try to identify more appropriate belief targets across the variants. The structure of the task gives a more specific prediction than an ordering of full-belief error. Each element of the target belief $b_t = (b_{1,t}, b_{2,t}, b_{3,t})$ is the posterior probability of one hidden state. To test which state distinctions are linearly accessible, we construct two scalar probe targets:

$$\rho_t = b_{3,t}, \qquad \delta_t = b_{1,t} - b_{2,t}. \tag{5}$$

The reward-state target ρ_t measures whether the representation tracks occupancy of state 3. The state-distinction target δ_t measures whether it distinguishes state 1 from state 2. Together with the normalization of the belief vector, these two targets reconstruct the full, three-state belief. As shown in Figure 2(b), we find that the Trivial Quotient action map is constant. Notably, the Reward-State Quotient map depends on whether the state is in $\{1, 2\}$ or $\{3\}$ but not on the distinction between states 1 and 2. The Full-State Quotient map distinguishes all three states.

Compared to the Full-State Quotient, the probe information absent under Reward-State Quotient is therefore not a uniform loss of belief fidelity: at least part of it is specifically the distinction between states 1 and 2, which is less useful in that condition. This result supports H2a, while arguing against treating belief emergence as an inevitable consequence of learning in partially observed settings.

4.3 So what does the Reward-State Quotient agent actually believe?

The above clues inform a better target. Reward-State Quotient depends only on the partition $\{\{1, 2\}, \{3\}\}$, but there are at least two ways an agent could represent the information needed for that distinction. It could compute the full three-state posterior and expose only $b_{3,t}$ to its controller. Alternatively, it could coarsen the observations using $g(\mathrm{A}) = g(\mathrm{B}) =$ not-C and $g(\mathrm{C}) = \mathrm{C}$, then recursively update a self-contained two-state filter. The resulting coarse belief c_t is not identical to $\rho_t = b_{3,t}$ from the full filter. Both assign probability to reward state 3, but condition on different observation histories: the full filter retains the identity of tokens A and B, whereas the coarse filter discards it.

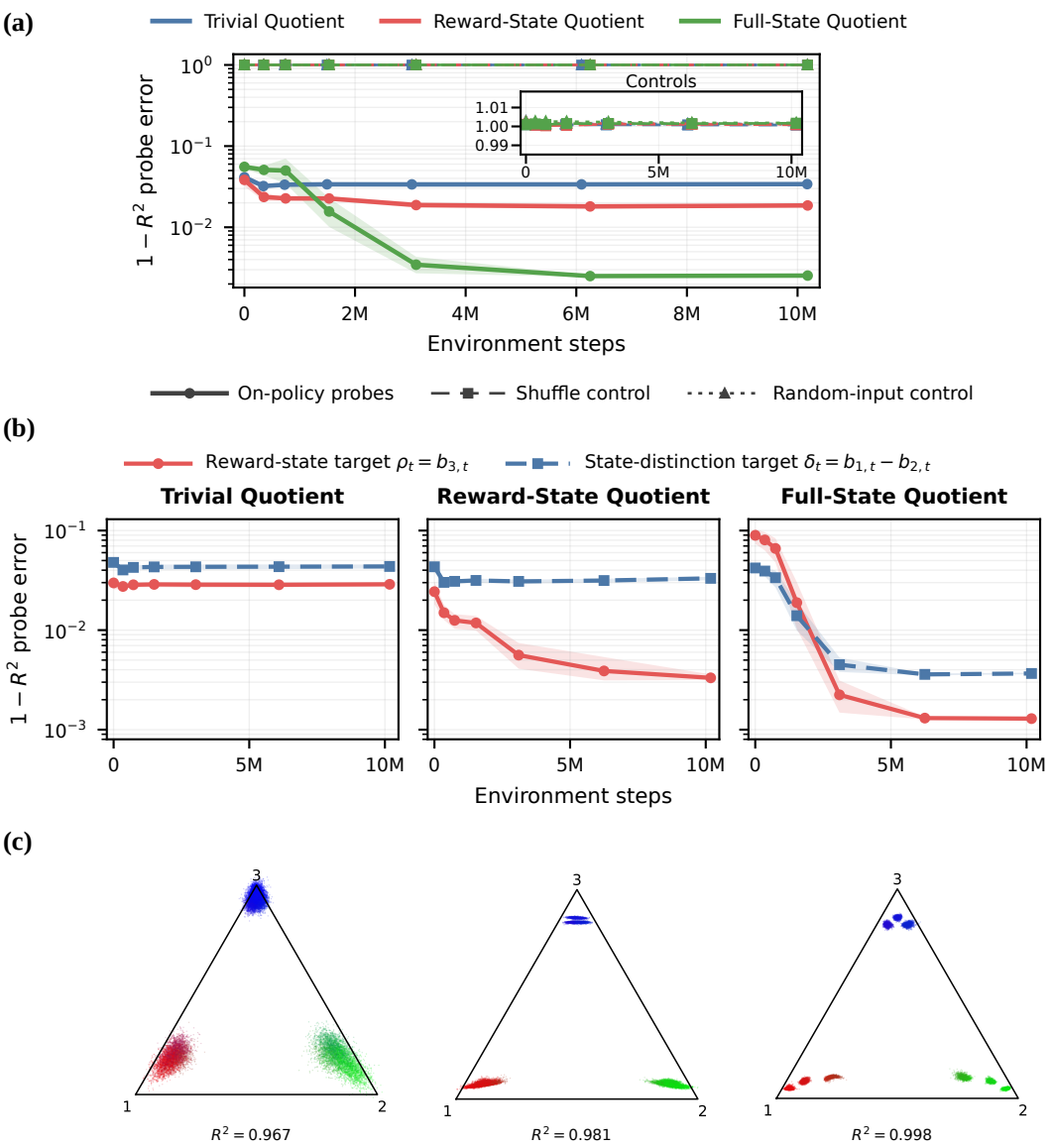


Figure 2: **Control demands predict belief accessibility. (a)** Accessibility of the exact three-state Bayesian belief, b_t. The inset magnifies frozen-network shuffle and random-input controls. **(b)** Belief accessibility of the reward state, ρ_t, and distinctions between the non-rewarding states, δ_t. **(c)** Simplex plots of the decoded beliefs. The target is the exact belief state, b_t.

This quotient is itself an exact two-state POMDP. Let $\mathcal{A} = \{1, 2\}$ denote the non-reward class and $\mathcal{B} = \{3\}$ the reward class. States 1 and 2 have identical transition mass into $\mathcal{A}$ and $\mathcal{B}$ under every action, and identical probabilities of emitting the coarse observations. These equalities establish exact controlled lumpability, an action-conditioned analogue of symmetry-induced abstract belief updating (Rosas, 2025). Writing $c_t \in \Delta^1$ as the two-entry row vector $(c_t[\mathcal{A}], c_t[\mathcal{B}])$, the quotient belief is updated recursively in its own state space:

$$c_t = \mathrm{norm}\left[\left(c_{t-1}\widehat{T}_{a_{t-1}}\right) \odot \widehat{E}_{:,\tilde{y}_t}\right], \tag{6}$$

Each $\widehat{T}_a$ is a 2×2 row-stochastic matrix, with source classes indexing rows and destination classes indexing columns. The 2×2 emission matrix $\widehat{E}$ has quotient classes as rows and coarse observations as

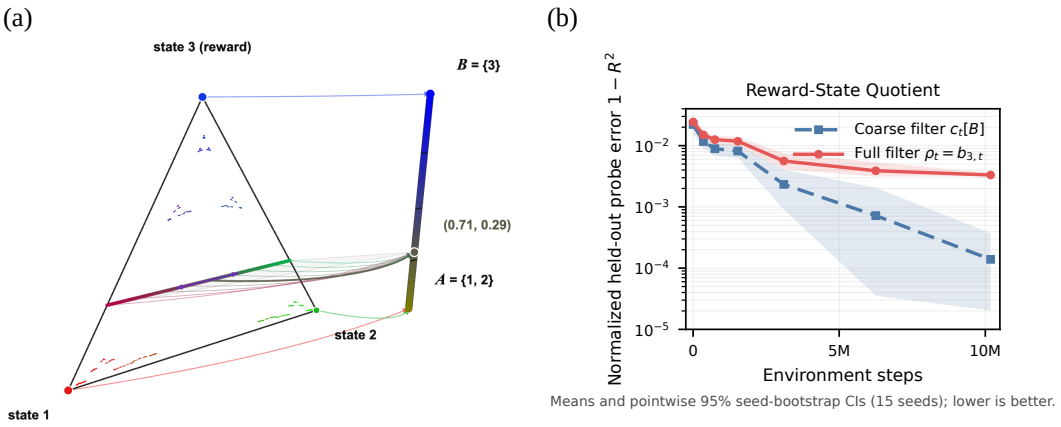


Figure 3: **Reward-State Quotient preferentially exposes a one-dimensional belief.** Left: the quotient map collapses the three-state belief 2-simplex to a two-class belief 1-simplex. Right: normalized held-out affine-probe error for the exact quotient belief and the corresponding full-filter marginal.

columns. The term $\widehat{E}_{:,\widetilde{y}_t}$ selects the likelihood column for the current coarse observation $\widetilde{y}_t = g(y_t)$. The operator $\mathrm{norm}(x) = x/\sum_i x_i$ makes the entries sum to one. Here a_{t-1} is the preceding executed action. The probe target $c_t[\mathcal{B}]$ is the quotient filter's posterior probability of reward class $\mathcal{B}$.

We fit separate probes to $c_t[\mathcal{B}]$ and ρ_t on paired rollout histories at every checkpoint. From initialization to the final checkpoint, the quotient target becomes markedly better fitted (Figure 3). The two curves begin with similar error, but the coarse-filter error finishes more than an order of magnitude below the full-filter target. The representation is more closely linearly aligned with the exact belief of a quotient than with the full-information posterior, in support of H2a. We therefore demonstrate a worked example of how scoring only against the full environmental posterior can understate belief-like structure when the learned representation preserves coarser, task-relevant distinctions.

Because the targets are correlated, the affine-probe fit alone does not show that the network literally executes the recursion in Eq. (6) or causally uses the decoded quantity. Activation interventions remain necessary for those stronger claims.

This design also exposes a methodological point: a belief probe should be evaluated against the belief state induced by the task's control-relevant quotient, not automatically against the environment's maximally resolved posterior. Target choice makes probing a comparative diagnostic: a target encoding the process sufficient for hypothesized strategies (good-enough strategies, reward-hacking, avoidance of difficult-to-learn features, etc.) tests support for a representational hypothesis.

5 Reward alone selects which belief factors remain accessible

The previous section studies a setting where the state-action map changes while holding reward fixed. As a complementary evaluation, we fix the state-action map but vary reward.

We consider a setting where the latent variable is the Cartesian product of two independent factors: each is the same Full-State Quotient POMDP above. Each factor emits one of three tokens, and the pair is encoded as a single nine-valued joint token. The agent observes this joint token together with the previous joint action and chooses one of nine joint actions, comprising every pair of the three per-factor actions. There are two conditions: in Joint Reward, occupancy of state 3 contributes reward for both factors. Meanwhile, in Unrewarded Factor 2, only factor 1 contributes; factor 2 remains observable, dynamic, and controllable but has no effect on return. We train a transformer agent on each condition for 30 million environment steps using REINFORCE and fit separate affine probes to the exact three-state belief of each factor, across 15 seeds.

The results in figure 4 supply a reward-only counterpart to the ladder. Under Joint Reward, training improves access to both factors in parallel: their probe-error trajectories remain closely matched

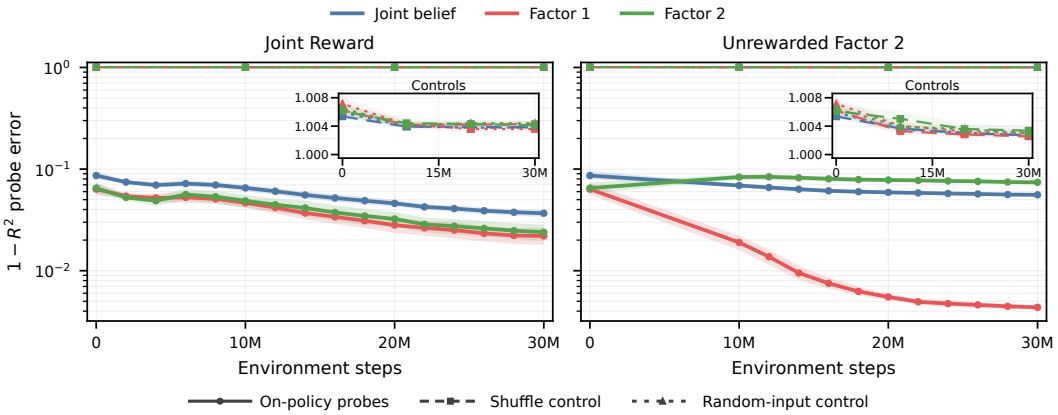


Figure 4: **Reward selects belief factors without changing the action map.** Normalized held-out affine-probe error $1 - R^2$ for each factor's exact three-state belief. Joint Reward trains both factors symmetrically; Unrewarded Factor 2 removes only the second factor's reward contribution. Insets show frozen-network history-shuffle and random-input controls.

and converge to similarly low levels. But when factor 2 is unrewarded, histories that imply different beliefs about factor 2 but the same belief about factor 1 are equivalent for maximizing return. A task-sufficient representation can therefore merge these histories, quotienting out factor 2, even though only reward has changed. Training selectively improves access only to the factor that determines return rather than indiscriminately encoding every predictable and controllable latent process. Reward can therefore induce a belief quotient without any change to the physical action map, supporting H2b. Appendix D gives the complete design.

6 Related Work

Belief representations in neural sequence models. This work extends previous work from Shai et al. (2024); Piotrowski et al. (2025); Shai et al. (2026) to capturing beliefs in POMDP settings. Lambrechts et al. (2022) find that recurrent Q-networks encode beliefs of variables relevant to optimal control and lose mutual information with irrelevant variables. Our work adds an explicitly geometry-resolved, action-conditioned test.

Learning explicit or approximate beliefs. Approximate belief representations have long been used to make POMDP control tractable (Rodríguez et al., 1999). Modern methods often add privileged state supervision, a generative model, or a predictive objective to encourage useful memory. Wang et al. (2023) explicitly decouple belief modeling from policy optimization and learn compact state features relevant to reward. Kuo et al. (2025) show that predictive coding can make meta-RL representations more compact and Bayes-aligned than reward-only training. These methods ask how to construct better beliefs, while our work instead asks what belief-like structure spontaneously appears when the policy is not instructed to construct one.

State abstraction and task-induced invariance. State abstraction identifies when multiple states can be treated as equivalent without sacrificing value (Li et al., 2006); bisimulation metrics relax exact equivalence into quantitative state similarity (Ferns et al., 2004). Castro et al. (2009) studied bisimulation of belief states for designing efficient Deep-RL objectives induce representational invariances specific to the task being optimized (Halvagal et al., 2026). Our contribution is to express those invariances directly in Bayesian belief coordinates.

7 Discussion

The classical belief-MDP construction separates inference from control after the latent state space has been specified. Our results demonstrate this representation-relative separation principle in deep

RL. While the environment's dynamics establish the fundamental belief geometry, reward determines which distinctions are useful enough to preserve. Our results suggest that deep-RL agents implicitly construct belief-like representations as an instrumental goal, which are then used for downstream planning.

Our findings have interesting implications for world models and generalization. Our first experiment demonstrates that RL can result in learning rich belief geometry without an explicit prediction loss. We later show that agents generally may not represent the environment beyond what the task requires. If reward changes, so that a previously irrelevant distinction becomes important, an agent with a quotient belief must relearn its representation as well as its controller. An agent that maintains a fuller world belief via, say, an auxiliary prediction loss, may adapt by changing only the control map.

The results also suggest a concrete interpretability strategy for agents trained in POMDPs. Instead of asking only whether a model contains a named latent variable, one can derive a family of candidate probe targets from the environment and reward, then test their geometry, update law, and causal influence. Candidates are motivated by 1) analyzing decoded simplex representations against the environment as a target and 2) symmetries that can be motivated by reward and action representation.

Limitations and future work. A random network can preserve recent observations well enough for a fitted linear map to recover coarse belief branches. Consequently, the figures presented in this work normalize error within seed and checkpoint, retain initialization as a feature-quality control, and compare on-policy fits with frozen-history shuffle and random-input controls. However, even a high probe decoding accuracy does not directly imply that the actor causally uses the decoded quantity. Investigating activation interventions is an important direction for future work, which could establish a mechanistic interpretation of belief geometry.

Our environments are deliberately small and exactly filterable. They also do not heavily require deliberate information gathering. Future work may study settings requiring sacrificing immediate reward to obtain an observation that improves a later decision. Such experiments would test whether the learned belief supports planning and active information acquisition, not only reactive control.

Future work may also explore the effects of the bounty of RL algorithms and optimizers, and how splitting actor and critic networks requires different belief targets.

Reproducibility Statement

Code, experiment manifests, JSON outputs, and scripts to regenerate all experiments and figures are available at https://anonymous.4open.science/r/deep-beliefs-5C34.

AI Disclosure

One of the authors architected the RL harness design and rules used by the codebase. The architecture is composed of two repositories: 1) a rigorous and slow-moving harness built on top of RLlib that requires extensive human review for all PRs, and 2) an experiments repo that receives human review on experiment scripts but not over many helper scripts and, of course, not over programmatically generated code such as retained .json output or experiment manifests. LLMs wrote the vast majority of the code, with human review as described. Many tests were written and smoke tests were run. However, the most effective method of verification was executing many alternative experiments that tested hypotheses from alternative angles such as modified environments, related probe targets, varied hyperparameters, etc. Only a small subset made it into the paper. Another valuable test was the local and accurate reproduction of results found in the papers that motivated our work.

While LLMs helped research relevant and prior work, the authors manually researched, aided, and even wrote some of the prior work. The authors also researched this line of work at conferences and in individual discussions with other practitioners.

LLMs were not used in developing theoretical models or conceptual frameworks.

All hypotheses were generated by the authors. All results were also interpreted by the authors. LLMs helped suggest some of the diagnostic tests but not all and selection and interpretation was performed by the authors. Implementation was performed by LLMs in the codebase described above.

Regarding the text, LLMs were helpful in providing prose on which to build, but the authors manually wrote the vast majority of the body text and certainly reviewed every line. The appendix benefited from more direct LLM input, although it too was reviewed line by line for correctness and further edited for accuracy and conciseness for the reader. LLMs were used for spell checking.

LLMs did not write this disclosure.

REFERENCES

Karl Johan Åström. Optimal control of markov processes with incomplete state information i. *Journal of mathematical analysis and applications*, 10:174–205, 1965.

Nix Barnett and James P Crutchfield. Computational mechanics of input–output processes: Structured transformations and the ϵ-transducer. *Journal of Statistical Physics*, 161(2):404–451, 2015.

Satwik Bhattamishra, Arkil Patel, Varun Kanade, and Phil Blunsom. Simplicity bias in transformers and their ability to learn sparse Boolean functions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5767–5791, Toronto, Canada, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.317.

Alexander B Boyd, Dibyendu Mandal, and James P Crutchfield. Identifying functional thermodynamics in autonomous maxwellian ratchets. *New Journal of Physics*, 18(2):023049, 2016.

Pablo Samuel Castro, Prakash Panangaden, and Doina Precup. Equivalence relations in fully and partially observable markov decision processes. In *ProceedingsIJCAI'09*, pp. 1653 – 1658, 2009.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pp. 162–169, 2004. URL https://arxiv.org/abs/1207.4114.

Dylan Foster and Alexander Rakhlin. Beyond ucb: Optimal and efficient contextual bandits with regression oracles. In *International conference on machine learning*, pp. 3199–3210. PMLR, 2020.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. *Artificial intelligence*, 147(1-2):163–223, 2003.

Manu Srinath Halvagal, Sebastian Lee, and SueYeon Chung. Task-induced representational invariances depend on learning objective in deep RL. *arXiv preprint arXiv:2606.01868*, 2026. doi: 10.48550/arXiv.2606.01868. URL https://arxiv.org/abs/2606.01868.

Jay A Hennig, Sandra A Romero Pinto, Takahiro Yamaguchi, Scott W Linderman, Naoshige Uchida, and Samuel J Gershman. Emergence of belief-like representations through reinforcement learning. *PLOS Computational Biology*, 19(9):e1011067, 2023.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856.

Arthur Jacot. Implicit bias of large depth networks: a notion of rank for nonlinear functions. In *International Conference on Learning Representations (ICLR)*, 2023.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. *Artificial Intelligence*, 101(1–2):99–134, 1998. doi: 10.1016/S0004-3702(98)00023-X.

Po-Chen Kuo, Han Hou, Will Dabney, and Edgar Y. Walker. Predictive coding enhances meta-RL to achieve interpretable bayes-optimal belief representation under partial observability. In *Advances in Neural Information Processing Systems*, 2025. doi: 10.48550/arXiv.2510.22039. URL https://arxiv.org/abs/2510.22039.

Gaspard Lambrechts, Adrien Bolland, and Damien Ernst. Recurrent networks, hidden states and beliefs in partially observable environments. *Transactions on Machine Learning Research*, 2022. doi: 10.48550/arXiv.2208.03520. URL https://arxiv.org/abs/2208.03520.

Lihong Li, Thomas J. Walsh, and Michael L. Littman. Towards a unified theory of state abstraction for MDPs. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics*, 2006. URL http://rbr.cs.umass.edu/aimath06/proceedings/P21.pdf.

Mateusz Piotrowski, Paul M Riechers, Daniel Filan, and Adam S Shai. Constrained belief updates explain geometric structures in transformer representations. *arXiv preprint arXiv:2502.01954*, 2025.

Andrés Rodríguez, Ronald Parr, and Daphne Koller. Reinforcement learning using approximate belief states. In *Advances in Neural Information Processing Systems*, pp. 1036–1042, 1999. URL https://proceedings.neurips.cc/paper/1999/file/158fc2ddd52ec2cf54d3c161f2dd6517-Paper.pdf.

Fernando Rosas, Alexander Boyd, and Manuel Baltieri. Ai in a vat: Fundamental limits of efficient world modelling for agent sandboxing and interpretability. *arXiv preprint arXiv:2504.04608*, 2025.

Fernando E. Rosas. Symmetries at the origin of hierarchical emergence. *arXiv preprint arXiv:2512.00984*, 2025. doi: 10.48550/arXiv.2512.00984. URL https://arxiv.org/abs/2512.00984.

Simo Särkkä and Lennart Svensson. *Bayesian filtering and smoothing*, volume 17. Cambridge university press, 2023.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. doi: 10.48550/arXiv.1707.06347.

Adam Shai, Loren Amdahl-Culleton, Casper L Christensen, Henry R Bigelow, Fernando E Rosas, Alexander B Boyd, Eric A Alt, Kyle J Ray, and Paul M Riechers. Transformers learn factored representations. *arXiv preprint arXiv:2602.02385*, 2026.

Adam S. Shai, Sarah E. Marzen, Lucas Teixeira, Alexander Gietelink Oldenziel, and Paul M. Riechers. Transformers represent belief state geometry in their residual stream. In *Advances in Neural Information Processing Systems*, 2024. doi: 10.52202/079017-2387. URL https://arxiv.org/abs/2405.15943.

Charlotte Striebel. Sufficient statistics in the optimum control of stochastic systems. *Journal of Mathematical Analysis and Applications*, 12(3):576–592, 1965.

Vikrant Varma, Rohin Shah, Zachary Kenton, János Kramár, and Ramana Kumar. Explaining grokking through circuit efficiency. *arXiv preprint arXiv:2309.02390*, 2023. doi: 10.48550/arXiv.2309.02390.

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*, 2023. URL https://proceedings.mlr.press/v202/wang23ah.html.

Walter Murray Wonham. On the separation theorem of stochastic control. *SIAM Journal on Control*, 6(2):312–326, 1968.

Hattie Zhou, Arwen Bradley, Etai Littwin, Noam Razin, Omid Saremi, Josh Susskind, Samy Bengio, and Preetum Nakkiran. What algorithms can transformers learn? A study in length generalization. In *International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=AssIuHnmHX.

A TRANSDUCER BELIEF STATES

The following supplement connects the transducer construction above to the implemented probe. To match the main text's decision-time history, it labels the observation following A_t as O_{t+1} rather than O_t; the corresponding operator is denoted $K^{a,o}$.

A.1 Action-conditioned Bayesian targets

A transducer maps actions to observations through a latent memory state (Barnett & Crutchfield, 2015; Boyd et al., 2016; Rosas et al., 2025). We use row-vector beliefs and the decision-time history $H_t = (O_0, A_0, \ldots, A_{t-1}, O_t)$ from Section 2.1. Define the action–observation operator

$$K^{a,o}_{ij} = \Pr(S_{t+1} = j, O_{t+1} = o \mid S_t = i, A_t = a). \tag{7}$$

Each $K^{a,o}$ is nonnegative and substochastic; summing over observations recovers the transition matrix $T_a = \sum_o K^{a,o}$, with $T_a\mathbf{1} = \mathbf{1}$. The joint kernel permits observations to depend on the transition itself, not only on its destination.

Let $b_t(i) = \Pr(S_t = i \mid H_t)$. For a policy using only H_t and independent sampling randomness, A_t is conditionally independent of S_t given H_t. The policy's action likelihood therefore cancels from Bayes' rule: executed actions select the environmental kernel rather than providing an additional latent-state measurement. After action a and observation o,

$$Z_t(a, o) = b_t K^{a,o}\mathbf{1} = \Pr(O_{t+1} = o \mid H_t, A_t = a), \tag{8}$$

$$b_{t+1}(j) = \frac{\sum_i b_t(i) K^{a,o}_{ij}}{Z_t(a, o)}, \qquad b_{t+1} = \frac{b_t K^{a,o}}{b_t K^{a,o}\mathbf{1}}. \tag{9}$$

This requires $Z_t(a, o) > 0$. It is Eq. 8 of Rosas et al. (2025), transposed into the implementation's row-vector convention and indexed so that A_t precedes O_{t+1}. The shared `predictive_belief_update` routine evaluates precisely this normalized matrix product in double precision.

For transition-then-emission dynamics, $K^{a,o} = T_a D_o$, where $D_o = \mathrm{diag}(\nu(o \mid i))$; this recovers Eq. (2). For emission-then-transition dynamics the order is $D_o T_a$, while edge-emitting processes use their joint operators directly. Passive HMM filtering is the special case in which $K^{a,o}$ is independent of a. The extension therefore changes target construction, not probe capacity.

A.2 Relation to the bidirectional mixed-state matrix

The BDMSM of Rosas et al. (2025) retains a joint posterior over the initial and current latent states. In our orientation, define

$$R_t(i, j) = \Pr(S_0 = i, S_t = j \mid H_t), \qquad R_0 = \mathrm{diag}(b_0). \tag{10}$$

This is the transpose of their matrix. Their Eq. 15 becomes

$$R_{t+1} = \frac{R_t K^{a,o}}{\mathbf{1}^\top R_t K^{a,o}\mathbf{1}}. \tag{11}$$

The current-state marginal is $b_t = \mathbf{1}^\top R_t$, whereas $R_t\mathbf{1}$ is the posterior over the initial state. Left-multiplying Eq. (11) by $\mathbf{1}^\top$ gives Eq. (9). Our probe needs only this forward marginal and does not construct the full BDMSM or a retrodictive target. Equivalently, along a realized history,

$$b_t = \frac{b_0 K^{a_0,o_1} \cdots K^{a_{t-1},o_t}}{b_0 K^{a_0,o_1} \cdots K^{a_{t-1},o_t}\mathbf{1}}. \tag{12}$$

A.3 Reset, timing, and alternative targets

The initial belief b_0 includes only information revealed at reset. For a reset emission from a prior μ, this is $b_0 = \mu D_{o_0}/(\mu D_{o_0}\mathbf{1})$; if reset reveals no token, it remains μ. There is no preceding action at reset. Subsequent targets align with the activation after processing the current observation and before selecting the next action: they use the *preceding executed action*, not the action currently being predicted. Filtering proceeds through complete episode histories before warmup rows are excluded. Rewards are not included in these targets, matching their exclusion from the policy's observation history in our experiments.

Temporal alignment is task-specific. In the delayed passive task, the filter after the latest visible token requires one further predictive transition to reach the decision-time target, as specified in Eq. (20). An edge-emitting operator that already arrives at the target state must not receive an extra

transition. Experiment adapters specify these operators and check reconstructed beliefs against public environment diagnostics. The exact targets use all available history since reset, even when the network has a finite receptive field.

Alternative belief targets require their own state space, observation map, prior, and operators. A projection of a full posterior and a filter run on coarsened observations need not agree. In particular, the quotient target in Eq. (6) is updated in its own state space rather than obtained by summing coordinates of b_t. We compare targets on paired histories with separately fitted probes and target-specific variance normalization; a better fit establishes comparative linear accessibility, not a unique internal Bayesian algorithm.

A.4 Affine probe fitting and scoring

Following the affine-decoding approach of Shai et al. (2024), let $h_n \in \mathbb{R}^d$ be a frozen network activation and $y_n \in \mathbb{R}^k$ its aligned belief target, both written as row vectors. For each checkpoint and target, we fit

$$(\widehat{W}, \widehat{c}) = \arg\min_{W,c} \left\{ \sum_{n \in \mathcal{D}_{\text{fit}}} \|h_n W + c - y_n\|_2^2 + \lambda \|W\|_{\text{F}}^2 \right\}, \qquad \widehat{y}_n = h_n \widehat{W} + \widehat{c}. \tag{13}$$

Here $W \in \mathbb{R}^{d \times k}$ and the intercept c is unpenalized. Unlike the original unregularized regression, our ridge implementation centers features and targets on fitting data and solves the augmented least-squares system with an SVD-based solver. Beliefs supervise only this readout; the agent receives no belief loss. Predictions are scored without clipping or projection onto the probability simplex.

For every seed, checkpoint, and target, let $\bar{y}_{\text{test}}$ be the mean target over the independent test set. The normalized held-out error is

$$1 - R^2 = \frac{\sum_{n \in \mathcal{D}_{\text{test}}} \|\widehat{y}_n - y_n\|_2^2}{\sum_{n \in \mathcal{D}_{\text{test}}} \|y_n - \bar{y}_{\text{test}}\|_2^2} = \frac{\text{MSE}}{\widehat{\text{Var}}(y)}. \tag{14}$$

Both MSE and target variance are averaged over rows and coordinates; this is a global variance-normalized score, not an unweighted average of coordinatewise R^2 values. Zero normalized error denotes exact decoding, one matches the test-mean reference, and values above one are worse; the ratio is undefined for a constant target. Test-set centering defines the reporting baseline and is not used to fit the probe. Rollouts weight histories by their occurrence under the evaluation policy, so changes in visitation can also change the score.

B Token Prediction

It is important to note that minimizing cross-entropy rewards exact estimates of the next token probability, while maximizing expected correctness requires only selecting a most probable token. In this `Mess3` HMM, the former distribution determines the full predictive-state posterior, whereas histories with different posteriors can warrant the same optimal action.

B.1 Belief emergence in reinforcement learning

Following Shai et al. (2024), we specify `Mess3` by three token-labelled transition matrices $T^{(x)}$, one for each token $x \in \{\text{A}, \text{B}, \text{C}\}$. In the transition-then-emission convention,

$$T^{(x)}_{ij} = \Pr(X_t = x, S_t = j \mid S_{t-1} = i). \tag{15}$$

Rows and columns index source and destination states 1, 2, and 3, respectively. The matrices, with the same numerical parameters as in Shai et al. (2024), are

$$T^{(\mathrm{A})} = \begin{pmatrix} 0.765 & 0.00375 & 0.00375 \\ 0.0425 & 0.0675 & 0.00375 \\ 0.0425 & 0.00375 & 0.0675 \end{pmatrix}, \tag{16}$$

$$T^{(\mathrm{B})} = \begin{pmatrix} 0.0675 & 0.0425 & 0.00375 \\ 0.00375 & 0.765 & 0.00375 \\ 0.00375 & 0.0425 & 0.0675 \end{pmatrix}, \tag{17}$$

$$T^{(\mathrm{C})} = \begin{pmatrix} 0.0675 & 0.00375 & 0.0425 \\ 0.00375 & 0.0675 & 0.0425 \\ 0.00375 & 0.00375 & 0.765 \end{pmatrix}. \tag{18}$$

Each entry is a joint token-and-transition probability; individual $T^{(x)}$ matrices are not row-stochastic, but their sum is. The stationary initial distribution is $\pi = (1/3, 1/3, 1/3)$. For a row-vector belief η over the source state, observing token x gives the mixed-state update

$$\eta' = \frac{\eta T^{(x)}}{\eta T^{(x)} \mathbf{1}}, \tag{19}$$

where $\mathbf{1}$ is the all-ones column vector. Unlike Shai et al. (2024), episodes contain 512 steps, with the first episode length randomized independently across vectorized environments to reduce synchronization artifacts. We chose this implementation because its distribution is from the same source and its easier integration into our reinforcement learning oriented infrastructure.

B.2 Observation timing and Bayesian target

Let X_t denote the token emitted by state S_t. The agent at decision step t sees X_{t-1} and predicts X_t; the subsequent step reveals X_t and returns $r_t = \mathbf{1}\{A_t = X_t\}$. At reset no token has been revealed, so the input is an all-zero vector. Subsequent inputs are three-dimensional one-hot tokens. Actions and rewards are not appended to the network input, and actions do not affect the process.

For this timing convention, the probe target is $b_t = \Pr(S_t \mid X_0, \ldots, X_{t-1})$, the predictive belief over the state emitting the token to be guessed. Let $\eta_t = \Pr(S_t \mid X_0, \ldots, X_t)$ be the filtered mixed state obtained from Eq. (19), starting from $\eta_{-1} = \pi$. The probe target and next-token probabilities are then

$$b_t = \eta_{t-1} \sum_{x \in \{\mathrm{A},\mathrm{B},\mathrm{C}\}} T^{(x)}, \tag{20}$$

$$\Pr(X_t = x \mid X_{<t}) = \eta_{t-1} T^{(x)} \mathbf{1}. \tag{21}$$

Thus b_t is one transition ahead of the filtered belief after the same visible history, with $b_0 = \pi$ by stationarity. The exact filter conditions on all tokens since reset, not on rewards; the network has only a 10-observation window. Neither agent is trained against these beliefs.

B.3 Shared architecture and distinct objectives

Both runs use the same seed and the same transformer configuration. A bias-free linear map embeds the token input into a 64-dimensional residual stream, with learned absolute position embeddings. Four pre-LayerNorm causal blocks each contain one attention head with eight-dimensional queries, keys, and values and a 256-dimensional ReLU feed-forward sublayer. Both sublayers have residual connections. Linear $64 \to 3$ and $64 \to 1$ heads produce token logits and a scalar value estimate.

Both objectives use Adam with learning rate 10^{-4}, a configured batch size of 32,768 sampled steps per learner, minibatches of 4,096, and six shuffled epochs per batch. PPO uses clipping parameter 0.2, value-loss coefficient 0.5, and $\gamma = \lambda = 0$, with no entropy bonus, KL penalty, or auxiliary prediction loss. Neither PPO nor cross entropy were tuned and there are certainly better implementations depending on the metric to be maximized. However, our hypothesis is that rich belief geometry *can* emerge from a reward objective alone and this example accomplishes the desired goal.

We index training by the number of sampled environment steps: each step reveals one new token and scores one prediction. Figure 5 shows probe error at initialization and at checkpoints across training.

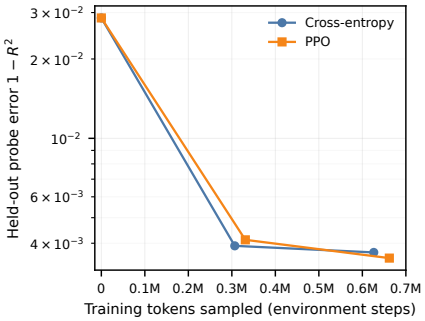


Figure 5: **Belief accessibility during passive token prediction.** Held-out affine belief decoding for transformers trained with cross-entropy and PPO, plotted against newly revealed training tokens.

Table 2: Belief-Quotient Ladder conditions. The three action entries maximize $\Pr(S_{t+1} = 3 \mid S_t = i, a)$ separately in latent states 1, 2, and 3.

Condition	State-conditional actions in states 1 / 2 / 3	Action partition	Candidate belief
Trivial Quotient	Tilt X / Tilt X / Tilt X	$\{\{1, 2, 3\}\}$	constant
Reward-State Quotient	Tilt X / Tilt X / Noop	$\{\{1, 2\}, \{3\}\}$	two-state quotient
Full-State Quotient	Tilt X / Tilt Y / Noop	$\{\{1\}, \{2\}, \{3\}\}$	full three-state belief

C Belief-Quotient Ladder Experiment

C.1 Environment and Control Requirements

The baseline transition matrix is

$$T = \begin{pmatrix} 0.75 & 0.15 & 0.10 \\ 0.15 & 0.75 & 0.10 \\ 0.30 & 0.30 & 0.40 \end{pmatrix}. \tag{22}$$

The initial state distribution is uniform. States 1, 2, and 3 emit tokens A, B, and C, respectively, with probability 0.85, and either other token with probability 0.075. At each time step, the agent receives reward one if the current latent state is state 3 and zero otherwise, $r_t = \mathbf{1}[S_t = 3]$; this reward is evaluated before the action-induced transition. The current token and previous executed action form the six-dimensional observation.

Noop applies T unchanged. Tilt X and Tilt Y modify only the log weight of destination state 3 in each source-state row. For effect size $e = 1.5$ and condition-specific direction vector d_a, the controlled transition is

$$T_a(i, j) = \frac{T(i, j) \exp(e\, d_a(i) \mathbf{1}[j = 3])}{\sum_k T(i, k) \exp(e\, d_a(i) \mathbf{1}[k = 3])}. \tag{23}$$

The Tilt X and Tilt Y direction vectors are $(+1, +1, +1)$ and $(-1, -1, -1)$ for Trivial Quotient, $(+1, +1, -1)$ and $(-1, -1, -1)$ for Reward-State Quotient, and $(+1, -1, -1)$ and $(-1, +1, -1)$ for Full-State Quotient.

Episodes contain 1,024 steps. Episode de-syncing is used: the first episode length is randomized uniformly between 1 and 1,024 to decorrelate resets from training batches; subsequent episodes have exactly 1,024 steps.

C.2 Exact Reward-State Quotient POMDP

For Reward-State Quotient, let $\mathcal{A} = \{1, 2\}$ and $\mathcal{B} = \{3\}$, and merge tokens A and B into not-C. The coarse prior is $(2/3, 1/3)$. With state rows $(\mathcal{A}, \mathcal{B})$ and observation columns (not-C, C), the emission

matrix is

$$\widehat{E} = \begin{pmatrix} 0.925 & 0.075 \\ 0.15 & 0.85 \end{pmatrix}. \tag{24}$$

The action-dependent transition matrices, with rows and columns $(\mathcal{A}, \mathcal{B})$, are

$$\widehat{T}_{\text{Noop}} = \begin{pmatrix} 0.90 & 0.10 \\ 0.60 & 0.40 \end{pmatrix}, \qquad \widehat{T}_{\text{Tilt X}} = \begin{pmatrix} 0.667572 & 0.332428 \\ 0.870509 & 0.129491 \end{pmatrix}, \tag{25}$$

$$\widehat{T}_{\text{Tilt Y}} = \begin{pmatrix} 0.975808 & 0.024192 \\ 0.870509 & 0.129491 \end{pmatrix}. \tag{26}$$

These matrices sum destination probabilities within each quotient block; the fine-state source rows 1 and 2 agree for every action. Together with their equal coarse emission probabilities, this verifies exact controlled lumpability. At reset, the prior is conditioned on the first coarse observation. Later updates use the preceding executed action and current coarse observation as in Eq. (6). Complete episode prefixes are filtered before applying the probe warmup mask.

C.3 Model and training

All three conditions use the same approximately 201,000-parameter banded causal transformer. It has a 64-dimensional residual stream, four pre-layer-normalized blocks, one 64-dimensional attention head per block, rotary position embeddings on queries and keys, and GELU feed-forward sublayers of width 256. Each layer has a 32-position attention band, giving an effective 128-step raw-observation receptive field across four layers. The latest post-final-layer-normalization residual is used for probing. Each condition is trained independently for 15 seeds per condition. Training uses baseline-free (no learned critic) REINFORCE with $\gamma = 0.99$. The reward-to-go advantages are mean-centered and divided by their batch standard deviation before each update. Each collected batch receives one full-batch update without minibatching or reuse. The nominal batch size is 32,768 steps; complete-episode collection with 24 vectorized environments per sampling runner can overshoot that target. The optimizer is Adam with learning rate 4.2×10^{-4}. The entropy coefficient decreases linearly from 0.03 at initialization to zero at 9 million sampled steps and remains zero thereafter. We also experimented with PPO and SAC; both showed rapid convergence. However, we ultimately chose to use the bare-bones REINFORCE route to focus the discussion on state-action and reward induced belief quotients rather than artifacts of the panoply of reinforcement learning algorithms.

Complete-episode sampling produces final checkpoints between approximately 10.05 and 10.19 million steps. Checkpoints are log-spaced plus each run's final checkpoint. Appendix E describes the held-out probe rollouts, controls, and statistical reporting.

D Reward Relevance experiment

D.1 Factored environment and reward manipulation

The environment is the Cartesian product of two independent copies of the Full-State Quotient factor from Appendix C. Each factor has the transition matrix in Eq. (22), the same three-token emission channel, and the same action-dependent transitions defined by Eq. (23). The joint latent state $(S_t^{(1)}, S_t^{(2)})$ therefore has nine values. The initial distribution is the product of two uniform distributions, emissions are conditionally independent across factors, and each joint transition is the Kronecker product of the selected per-factor transition matrices.

The action space contains all nine ordered pairs of Noop, Tilt X, and Tilt Y. Thus each factor retains the Full-State Quotient action map: Tilt X is optimal in state 1, Tilt Y in state 2, and Noop in state 3. The policy observation is an 18-dimensional vector containing a one-hot encoding of the current nine-valued joint token followed by a one-hot encoding of the preceding nine-valued joint action.

The two conditions differ only in the pre-transition occupancy reward:

$$r_t^{\text{Joint Reward}} = \mathbf{1}[S_t^{(1)} = 3] + \mathbf{1}[S_t^{(2)} = 3], \tag{27}$$

$$r_t^{\text{Unrewarded Factor 2}} = \mathbf{1}[S_t^{(1)} = 3]. \tag{28}$$

Consequently, the Unrewarded Factor 2 condition admits an exact task-sufficient quotient that discards the second factor and retains only the three-state belief of factor 1. Joint Reward instead requires control of both factors. Episodes contain 1,024 steps, with the first episode length randomized uniformly between 1 and 1,024 to decorrelate resets from batch boundaries.

D.2 Model and optimization

Both conditions use the same model, training algorithm, and number of seeds per condition described in Appendix C.3 but were trained to 30 million sampled environment steps and did not use entropy regularization.

D.3 Belief probes, controls, and aggregation

At each retained checkpoint, the frozen greedy policy generates independent 20,000-step probe-fitting and 20,000-step held-out rollouts from eight vectorized environments. Complete episode histories are retained, and the first eight steps of each episode are excluded from fitting and scoring. The probed representation is the latest-position residual after the final transformer block and before the final layer normalization.

Separate affine probes target the exact action-conditioned three-state predictive belief of each factor; a joint nine-state belief is also computed as a diagnostic. Probe coefficients are selected by ten-fold cross-validation over SVD pseudoinverse cutoffs $\{10^{-15}, 10^{-10}, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}\}$, then refit on the complete fitting rollout and evaluated on the independent held-out rollout. For each seed, checkpoint, and factor, the reported error is

$$1 - R^2 = \frac{\mathrm{MSE}(\widehat{b}_t^{(k)}, b_t^{(k)})}{\widehat{\mathrm{Var}}(b_t^{(k)})}. \tag{29}$$

Figure 4 shows exact checkpoint steps shared by all 15 seeds within each condition; trajectories are not interpolated. Lines are arithmetic means of the per-seed normalized errors. Pointwise 95% intervals use 50,000 percentile-bootstrap resamples of complete seed trajectories, with the same resampled indices for both factor targets. The reported factor contrast is $(1 - R^2)_{\text{factor 2}} - (1 - R^2)_{\text{factor 1}}$ within seed and checkpoint before aggregation.

Frozen-network controls are evaluated at initialization and at 10, 20, and 30 million steps. They either shuffle complete current-token/previous-action pairs within each episode while preserving reset rows or sample valid pairs with replacement from empirical reset and non-reset pools. Both controls keep the original time-indexed belief targets, so their normalized error approaches the no-information value of one. They are descriptive off-distribution checks, not calibrated permutation tests or causal interventions.

E Probe construction and statistical reporting

For the passive and Belief-Quotient Ladder experiments, at each retained checkpoint the frozen policy produces independent probe-fitting and probe-test rollouts under greedy action selection. An affine ridge probe with penalty 10^{-6} maps the post-final-layer-normalization residual at the most recent position to its target. Exact beliefs are advanced recursively using every realized preceding-action/current-observation pair and are independently checked against environment diagnostics. Reward Relevance uses the separately specified probe protocol in Appendix D.

The Belief-Quotient Ladder full-belief, scalar, and quotient-filter probes use 60,000 fitting rows and 80,000 independent held-out rows from 16 vectorized environments. Complete episode histories are retained through reset, and the first 64 steps of each episode are excluded from scoring. The full and coarse filters are evaluated on paired histories, with a separate probe fit for each target.

The frozen-history controls instead use independent 20,000-row fitting and test streams and exclude the first 128 steps, matching the transformer's effective receptive field. The shuffle control permutes complete current-token/previous-action pairs within each episode while keeping reset rows fixed. The random-input control samples valid pairs from separate empirical pools. Both retain the original

rollout's time-indexed belief as the probe target, so they are descriptive out-of-distribution controls rather than calibrated permutation tests or causal interventions.

For every seed, checkpoint, and target, the reported normalized held-out error is

$$1 - R^2 = \frac{\mathrm{MSE}}{\widehat{\mathrm{Var}}(y)}. \tag{30}$$

Normalization precedes aggregation. The ladder figures show arithmetic means over 15 independent training seeds and pointwise 95% percentile intervals from 50,000 bootstrap resamples of complete seed trajectories. The same resampled seed indices are used for targets compared within a panel. These are pointwise intervals, not simultaneous confidence bands.