TRANSFORMERS DEVELOP SEPARABLE POSTERIOR AND VALUE REPRESENTATIONS IN A SIMPLE DECISION TASK

Anonymous authors
Paper under double-blind review

ABSTRACT

How does a transformer make decisions when it is uncertain about the world? A network could form a posterior over hidden states and value its options under that posterior, or it could map observations to choices with no such intermediate structure. We investigate this question with a transformer trained by reward alone on a simple decision task. Hidden states are assigned payoffs, and offers composed of sets of hidden states are presented to the transformer. The offers and state payoffs are given in context, and the exact posterior and the value of every offer are known in closed form. We anticipate and then find that both quantities are linearly represented in the residual stream ($R^2 > 0.99$), in separate subspaces, and that the representations extend to offers never seen in training. Interventions show that both are used for choice: steering the posterior shifts choice by the amount the ground truth predicts, and does so through the value subspace, while steering the value shifts choice and leaves the posterior fixed. None of this is required by the objective, which constrains only the difference between the offer values. A pretrained language model, Qwen3-1.7B-Base, trained on the same task, develops the same organization. Our work extends the posterior geometry found under next-token prediction to choice, and gives a setting in which the quantities that drive a model's decisions can be measured against ground truth and intervened on.

1 INTRODUCTION

Neural networks make decisions based on partial observations of the world. In principle, such decisions can be made in two stages: first, observations of the world form a posterior over latent states, and then the value of those states is integrated with the posterior to make choices (Åström, 1965; Smallwood & Sondik, 1973; Kaelbling et al., 1998). Whether transformers decide this way, or some other way, can be tested by examining their internal activations. We ask whether models form an internal decision system in which uncertainty about the latent state of the world is represented separately from the value of candidate actions, and whether both are used to choose. If transformers do develop such a system, the representations that determine a model's decisions could be identified, monitored, and intervened on before they manifest in harmful behavior (Hubinger et al., 2019; Shah et al., 2022).

A transformer's residual stream can carry information about the latent state of the data-generating process (Li et al., 2023; Nanda et al., 2023). For hidden Markov processes, the posterior over the latent states is linearly represented with the fractal geometry that optimal prediction implies (Shai et al., 2024). Studies investigating value representations in transformers have probed for the utility function estimated from their own stated preferences (Mazeika et al., 2026), or have shown that a transformer trained for in-context policy evaluation comes to estimate value in its forward pass, by temporal-difference learning (Wang et al., 2025). The models in which the posterior's geometry has been established predict tokens over states that carry no value, and studies of value in transformers do not measure the posterior. To our knowledge, no study has measured the two in one network causally against ground truth.

To test whether transformers form the posterior and the value of each option and use them to choose, we train a small transformer on a synthetic decision task with known latent structure, in which both

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.

quantities are exactly computable. The transformer chooses between two sets of latent states, which we call *offers*. On every episode, the offers and the payoff of each state are defined in context, observations about the active state arrive in context, and the reward is the payoff of a state drawn from the chosen offer. The network is trained by reward alone, and neither the posterior nor the offer value is supervised. Because the latent process and payoffs are known, the posterior and every offer value are computable in closed form, providing ground truth for the network's internal representations.

The reinforcement objective rewards only the choice, so a network could map the observations, payoffs, and offers to the value difference directly, or value each offer without a shared posterior, and choose identically (Section 2.4). The offer value is not itself a linear function of the posterior, because the payoff of a choice is drawn from the posterior restricted to the chosen set (Section 2.2). Its linear readability therefore does not follow from the posterior's.

We find that a small transformer trained by reward alone forms the posterior and the value of each offer, keeps them in separate subspaces, and chooses through them. In summary, we contribute (i) a decision task that extends the hidden-Markov setting of Shai et al. (2024; 2026) to choice, providing exact ground truth for the posterior and value of every offer; (ii) dissociable linear representations of the posterior and offer values, in separate subspaces, that generalize to offers never presented during training; and (iii) evidence that both representations are causally used for choice, with a posterior steer changing the downstream offer values and the choice by the amount the ground truth predicts, a value steer changing choice while leaving the posterior fixed, and the value representation carrying the effect of the posterior on choice. We replicate our findings in a large language model, Qwen3-1.7B-Base, which carries additional readable, but causally inert, representations (Section 3.5). Section 4 situates these findings within the decision-theoretic literature.

2 Task and methods

2.1 The decision task

We train a transformer to choose between two sets of latent states. The latent state is the joint state of two independent three-state Markov chains, so there are nine states in all. On every episode the two sets, the payoff of each state, and observations about which state is active are all given in context. The model chooses one set, and its reward is the payoff of a state drawn from that set. We call each set an *offer* and write X for the set of states it contains. The standing offers and state payoffs are set once at the beginning of each episode.

Writing η_s for the posterior probability of state s given the observations so far, and S for the random variable for the state that is active, the expected payoff of choosing X is a conditional expectation. The payoff of state s is $u(s) \in \{-1, 0, +1\}$. Restricting the posterior to X puts mass $\eta_s/P(X)$ on each $s \in X$. Accordingly, the expected payoff of choosing X is a ratio of two linear functions of the posterior, each depending on X:

$$Q(X) \;\coloneqq\; \mathbb{E}[u(S) \mid S \in X] \;=\; \sum_{s \in X} \frac{\eta_s}{P(X)}\, u(s) \;=\; \frac{m(X)}{P(X)}, \tag{1}$$

where $P(X) = \sum_{s \in X} \eta_s$ is the offer probability and $m(X) = \sum_{s \in X} \eta_s\, u(s)$ is the payoff mass. We call $Q(X)$ the *offer value*—the action value of choosing X. The offer value depends on the posterior, and changes whenever the observations do.

Calculating the offer value from a posterior and the state payoffs, as in Eq. (1), is one way to solve the task, but not necessarily the method that the network uses. Different internal computations produce the same choices, so behavior alone cannot distinguish (i) a network that represents the posterior and the offer values from (ii) one that maps the context to a choice without them. We therefore ask whether the network contains internal representations that play the roles of η and Q, and test three properties of them, adapted from Herrmann & Levinstein (2026): *Readability*: the activations carry η and Q linearly, across offers never presented in training. *Dissociability*: the two quantities are represented in separate subspaces, so an input that should change only one of them moves only its representation. *Causal organization*: steering the posterior and value representations affects choice as the equation predicts. Each representation moves choice when written to directly, and the offer value mediates the posterior's effect on choice.

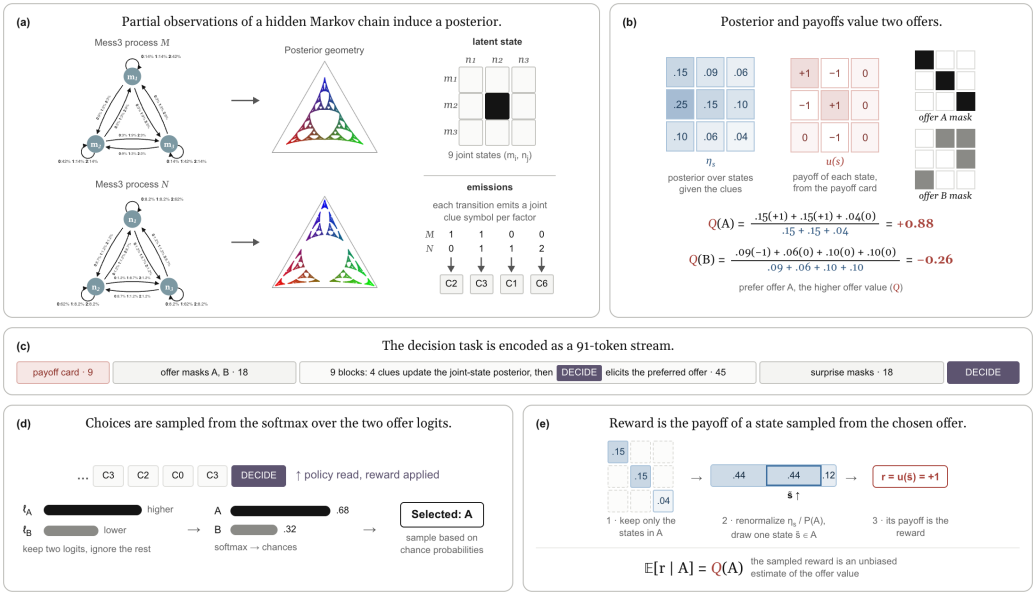

Figure 1: Overview of the simple decision task. (a) The latent state is the product of two independent three-state Mess3 factors. Each factor induces its own fractal posterior geometry, and each clue token carries one emission from each factor. (b) Payoffs associated with states, and the sets of states that make up each offer, are given in context. Ground-truth offer values $Q(X)$ are calculated from the known posterior, payoffs, and offers by Eq. (1). (c) Each episode comprises 91 tokens. The state payoffs and offers are first given, followed by nine rounds in which four clues are provided and the model is prompted to choose its preferred offer. Finally, the model is presented with a pair of unseen offers and is prompted to make a tenth decision. (d) At each readout the policy keeps only the two response logits and samples the choice from their softmax. (e) Choosing X restricts the posterior to X, renormalizes, samples one state, and returns its payoff as reward, so the expected reward is exactly $Q(X)$.

2.2 A WORLD WITH EXACT GROUND TRUTH

These tests require a ground truth for η and Q, so we construct a synthetic world in which every term of Eq. (1) is computable in closed form (Figure 1). In each episode the latent state is one of the nine joint states. In context, the model is shown the payoff of every state, is offered two sets of states, and reads a stream of clue tokens. Each clue is a partial observation of the evolving latent state, and the model chooses between the two offers repeatedly as the evidence accumulates. Choosing X pays the payoff of a state drawn from the exact posterior restricted to X, so that the expected reward of choosing X is exactly $Q(X)$ by construction (Figure 1e). Sampling the paid state from the posterior restricted to X makes the offer value a ratio, $Q(X) = m(X)/P(X)$, rather than a linear function of the posterior, so that a linear read of the value is evidence the network computed it. The sampled reward is the only training signal the model receives. The ground-truth η and Q are never supplied to the model, and no next-token loss is used.

The sequence of latent states is generated by a hidden Markov process, following Shai et al. (2024; 2026). The latent state is the joint state of two independent three-state factors, M and N. Clue tokens are emitted as each factor transitions, so the posterior η over the nine joint states updates in closed form at every clue (Appendix B). Because the factors evolve independently and every clue token carries one emission from each, the joint posterior is the tensor product of the factor posteriors. The factored world hypothesis predicts that the network represents their direct sum $\eta_M \oplus \eta_N$, two orthogonal 2-simplices in four dimensions rather than the eight of the joint simplex, as Shai et al. (2026) found for next-token prediction. Throughout the paper, *the posterior* means this pair of factor posteriors (η_M, η_N), from which the joint η follows. It is not the joint distribution itself, nor the probability $P(X)$ of an offer, which is a sum over a subset of η (Eq. (1)).

From the joint posterior η at each position (whose calculation is shown in Eq. (4) of Appendix B), one can immediately obtain the exact offer probability $P(X) = \sum_{s \in X} \eta_s$ and, together with the state payoffs $u(s)$, one can also obtain the offer value $Q(X)$ of every offer via Eq. (1) (see Figure 1b). We call the policy $\pi_\beta = \mathrm{softmax}(\beta\, Q)$ over the two offers the target policy, where the inverse temperature β sets how sharply the choice probabilities concentrate on the higher-valued offer. In the experiments presented, we set $\beta = 4$.

Each episode is a stream of 91 tokens (Figure 1c). The first nine provide the payoff for each of the nine joint states, which we refer to as the payoff card. The next eighteen tokens present the two offer masks, introducing a standing pair of disjoint offers. Each mask has nine symbols, indicating whether a state is in the offer. Of the 511 nonempty subsets of the nine states, 64 are withheld and never offered during training, so probes can be tested on unseen offers. The withheld masks are drawn at random and stratified over the structural classes of a mask (single states, their complements, sets fixed by one factor, exclusive-or sets, product sets, unions of factor sets, and generic sets), so that every class is represented. After the payoffs and the offers are presented, nine rounds occur within an episode, each presenting four clue tokens followed by a neutral `DECIDE` token, at which point the choice between the standing offers is read out and its reward sampled. The two offers each have a corresponding symbol in the vocabulary, neither of which ever appears in the stream. At each `DECIDE` token, the model's logits for these two symbols act as the choice logits, and the choice is sampled from their softmax (Figure 1d, Appendix A). A terminal surprise pair, shown only at the end, precedes a tenth and final readout, so the last decision values a pair not presented earlier in the episode. A policy that scores every readout therefore needs a representation from which any offer can be valued, since a comparison built up for the standing offers would be insufficient.

2.3 Model and objective

We train a four-layer transformer to choose between the two offers, and later apply the same objective to a pretrained language model, Qwen3-1.7B-Base (Section 3.5). Additionally, we replicate the primary results across four independent training seeds (Appendix J). Because the choice never alters the stream, each readout is a contextual bandit, and we update the model with REINFORCE (Williams, 1992) with entropy regularization. The loss at each readout is

$$\ell = -r \log \pi(a) - H(\pi)/\beta \,, \tag{2}$$

where the policy π is the softmax over the two response logits, $a \sim \pi$ is the sampled choice, $r \in \{-1, 0, +1\}$ is the sampled reward of Section 2.2, and $H(\pi)$ is the policy entropy. The loss is averaged over the batch and the ten readouts. The entropy weight is $1/\beta$, so the objective's optimum is the target policy $\pi_\beta = \mathrm{softmax}(\beta Q)$ defined above (derivation in Appendix A). It is possible that the same offer values could be learned by predicting rewards. We use reinforcement learning, rather than prediction, because our question concerns representations that guide the model's own choices.

The toy transformer has two attention heads per layer, a 64-dimensional residual stream, and approximately 208,000 parameters. Every training step samples fresh episodes. The full settings for both models are given in Appendix C.

2.4 Hypothesis

We hypothesize that a transformer trained on this task makes decisions by computing the posterior probabilities and offer values. It forms a posterior over each factor of the latent state, computes a value for each offer from the two factor posteriors and the payoff card, and chooses according to the difference between the two values. We predict that both are linearly represented in the residual stream and, since one posterior supports the valuation of any offer, that these representations extend to offers never presented during training.

The training objective does not require an internal decision system with posterior and value representations. REINFORCE with an entropy bonus fixes the optimal policy at $\pi_\beta = \mathrm{softmax}(\beta Q)$ and supervises no internal quantity (Williams, 1992). There is no value head as in AlphaZero (McGrath et al., 2022): the optimum fixes only the difference $\beta(Q(A) - Q(B))$ (Appendix A), so value representations are a finding rather than a requirement. A network could compute this difference directly from the clues, payoff card, and offers, or value each offer without a shared posterior. Furthermore, even if the posterior and offer values are represented, they need not be linearly readable nor causally

used. These alternatives can produce the same choices and can be differentiated only by the representations found within the network. Establishing that $Q(A) + Q(B)$ is represented would thus show that the individual offer values are computed, rather than just their difference.

Changing the payoff card or offers should move the offer values and leave the posterior fixed, while changing the clues moves both. Steering the posterior should move the downstream offer values in accordance with Eq. (1) and choose with them, and steering the offer values should move choice and leave the posterior fixed. If the offer values mediate the posterior's effect on choice, inserting or removing the value component of a posterior steer should reproduce or remove its behavioral effect.

3 RESULTS

3.1 CHOICE FOLLOWS THE TARGET POLICY

We first verify that the toy network has learned the task. After training, we sample $8{,}192$ fresh episodes and, at each of their ten readouts, compare the probability the network assigns to choosing offer A with the probability the target policy $\pi_\beta = \mathrm{softmax}(\beta Q)$ assigns to it, computed from the exact offer values. Over the $81{,}920$ decisions the Pearson correlation of the two is 0.999 ($p = 10^{-5}$, the permutation floor; one-sided whole-episode permutation test, $100{,}000$ permutations; 0.002 for an initialized network). This shows that the model has learned to act as the target policy does, but leaves open whether it computes the posterior and the offer values when choosing.

3.2 THE POSTERIOR AND THE OFFER VALUES ARE LINEARLY REPRESENTED IN THE RESIDUAL STREAM

If the model computes the posterior and the offer values, both should be linearly decodable from the residual stream. Every probe and causal intervention is applied to the residual stream at the sixth `DECIDE` token. We fit a single linear regression from the activations to the concatenated targets: (i) the two factor posteriors (η_M, η_N) of Section 2.2 and (ii) the offer values $Q(A)$ and $Q(B)$ of the two offers. The probe is an affine map $\hat{b} = Wa + c$, fit once by ordinary least squares (Appendix D). We report five-fold cross-validated R^2 per target, and take as the posterior site the earliest layer at which the mean cross-validated R^2 of the two factor posteriors reaches 0.95, and as the value site the earliest layer at which the pooled R^2 of the two offer values does. The null is a probe of the same form, fit to an untrained network at initialization.

The probe recovers the two factor posteriors at cross-validated $R^2 = 0.997$ and 0.994 at the posterior site (layer 0) and the two offers' values at 0.999 at the value site (layer 2) (Appendix E). While the specific value site varied across replication seeds, it always formed downstream of the posterior representation, consistent with an offer value computed downstream of the posterior (Appendix J). Probes of the same form fit fresh to the untrained network read the two factor posteriors at $R^2 = 0.197$ and 0.299 and the offer values at 0.093 (Appendix F, Figure 5). The decoded factor posteriors reproduce the fractal geometry that Bayesian updating predicts for each factor, and the decoded value pair spreads over the full range of $(Q(A), Q(B))$. The gap $Q(A) - Q(B)$ is expected to be linearly readable near the output, since the unembedding is linear and choice follows $\mathrm{softmax}(\beta Q)$ (Section 3.1). To test whether both offer values are represented separately, we decode their sum, $Q(A) + Q(B)$, from the residual stream. Linear probes recover this at $R^2 = 0.999$ at the value site, indicating that the individual offer values are represented, rather than just their difference. The gap, $Q(A) - Q(B)$, is decoded at $R^2 = 0.999$. To check if probes generalize to held-out offers, we fit a probe at the final layer and score it on episodes in which one standing offer is replaced by each of the 64 held-out masks and the other by a random training mask disjoint from it, with the payoff card and clues unchanged. The offer probability composed from the decoded factor posteriors, $\hat{P}(X) = \sum_{s \in X} (\hat{\eta}_M \otimes \hat{\eta}_N)_s$, tracks the exact $P(X)$ at $R^2 = 0.989$ (Figure 2b), and the direct reading $\hat{Q}(X)$ tracks the exact offer value at 0.998 (Figure 2c). The offer value is thus linearly readable for offers never presented during training, so the representation generalizes over offers rather than merely memorizing trained ones, and the posterior remains linearly decodable with novel masks in context.

In addition to representing the factor posteriors η_M and η_N, it is also possible for the network to represent the probability of each offer as a single direction. This would require the model to

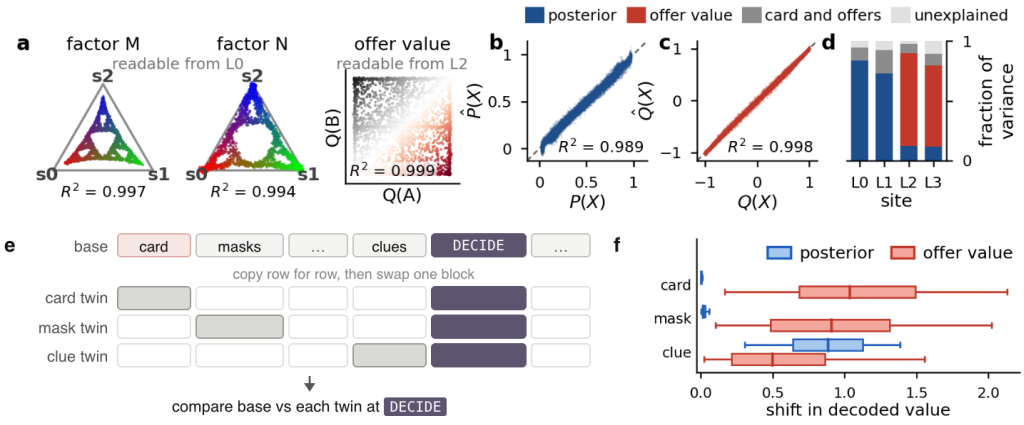

Figure 2: The posterior and the offer values are separately represented and generalize to offers withheld from training. (a) Linear probes recover the two factor posteriors (from layers 0–3) and the two offer values (from layers 2–3) from the residual stream. Points are colored by the ground truth. (b, c) Linear probes, fit at the final layer, evaluated with 64 offers never presented during training: the offer probability composed from the decoded factor posteriors, showing the posterior remains decodable under novel masks (b), and the offer value read directly (c), against their exact values. R^2 is pooled over the held-out offers. (d) Share of residual-stream variance attributed to each quantity by held-out encoding regressions, by layer. (e) The value and posterior representations are moved separately. Twin episodes, each differing from the base episode in only one of the payoffs, the offers, or the clues, are compared against the base. (f) Shift in the decoded posterior (probability units, layer 0) and offer values (payoff units, layer 2) from base to twin. Boxes are quartiles, whiskers the 5th–95th percentiles. Card and mask twins move the decoded values and not the posterior, while clue twins move both.

multiply the joint-state posteriors by the offer mask and sum across the probabilities. A probe fit directly to the offer probability reads at a median per-offer R^2 of 0.31, while the offer probability composed from the decoded factor posteriors reaches 0.985. We next ask whether the payoff mass of an offer, $m(X)$ of Eq. (1), is linearly readable. A probe fit directly to $m(X)$ reads at 0.74, but the mass is recoverable mainly through the offer value. Regressing the ground-truth $m(X)$ on the ground-truth $Q(X)$ and decoding the residual results in a median R^2 of 0.12 (Appendix G, Table 1). The network linearly represents the posteriors and the offer values, but not, as a single direction, the intermediate terms $P(X)$ and $m(X)$ from Eq. (1).

The posterior and offer values also explain the majority of the variance of the residual stream, as an unsupervised decomposition shows (Figure 2d). We apply principal component analysis to the residual stream on the $8{,}192$ evaluation episodes, then attribute each component's variance to the posterior, the offer values, or the card and offers by held-out encoding regressions on the analytic quantities, with the remainder unexplained (Appendix H). Figure 6 gives the component-level view. At the posterior site (layer 0) the posterior accounts for 0.834 of the total variance, whereas the offer values account for none of it. At the value site (layer 2), by contrast, the posterior accounts for 0.122 of the variance, while the offer values account for 0.774. At the posterior site, the first four components of the PCA encode the factored posteriors, whereas at the value site, the first two principal components reflect the offer value of the two options (Appendix H).

3.3 The two subspaces are dissociable

We next ask whether the posterior and value representations can be moved independently (Figure 2e–f). For each episode, we form three twin episodes. For each twin, we replace only one section of the sequence: the payoff card, the standing offers, or the four clues before the sixth `DECIDE` token. Card and mask twins leave the analytic posterior unchanged, so Eq. (1) predicts a shift in the internal value representation, but not the posterior representation. For the clue twin, a shift in the posterior and value representations is expected. The posterior and value are decoded at their respective read sites.

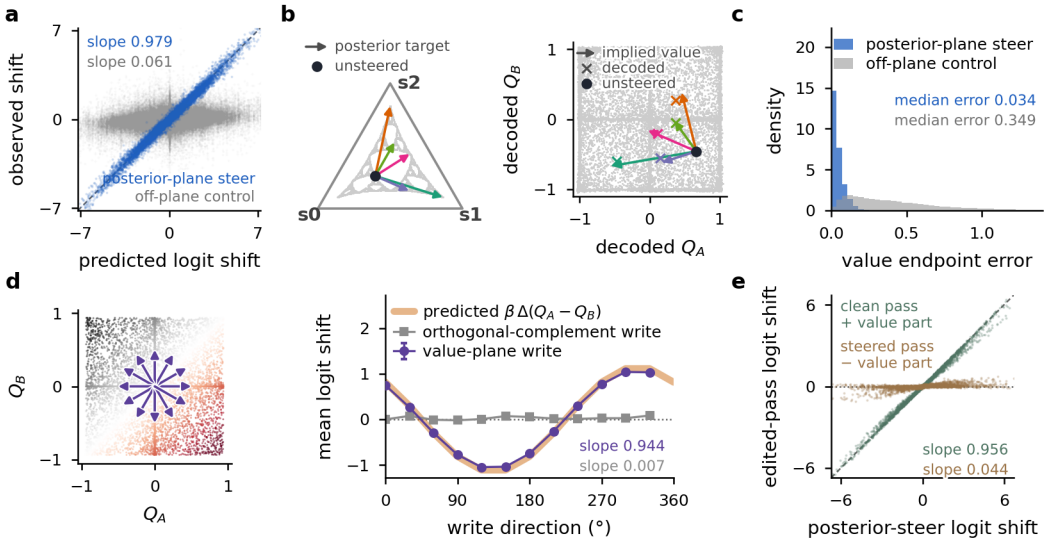

Figure 3: Posterior and value representations move choice. (a) Steering the posterior representation shifts choice as the counterfactual predicts, while matched off-plane edits do not. (b) Posterior steering shifts offer value representations to match the values implied by the posterior target. Example showing five steering directions applied to a single episode. Left: steering targets in the posterior simplex. Right: decoded value versus value implied by posterior steering targets. (c) Distribution of value endpoint error across episodes: posterior-plane steering matches the predicted value, while off-plane controls do not. (d) Value-plane edits along twelve directions shift choice by the predicted $\beta\Delta(Q_A - Q_B)$, while matched orthogonal edits have no effect (bars: 95% CI). (e) The value-plane component of a posterior edit is sufficient (insertion) and necessary (removal) for its effect. Posterior edits act through the value subspace rather than directly on choice.

Shifts are measured over the $4{,}143$ of $8{,}192$ sampled episodes in which both offers' probabilities are at least 0.02 in the base episode and in every twin. Shifts in the decoded posterior are Euclidean distances between the decoded probability vectors after projection onto the simplex plane, taken over both factors together, on a scale where two vertices of a factor's simplex are $\sqrt{2}$ apart; we call these probability units. Swapping the payoff card or the offer masks moves the decoded posterior by only 0.004 and 0.016 probability units respectively, while moving the decoded offer values by 1.03 and 0.91 payoff units (Figure 2f). Swapping the clues moves both: the posterior by 0.885 probability units, and the offer values by 0.50 payoff units (all median shifts over episodes). These results show that the offer value representations can be moved independently of the posterior representations.

3.4 Both representations are causally used for choice

Decoding alone cannot show that a representation is used, so we intervene on each subspace and measure the effect on choice (Figure 3). Causal interventions are applied to the encoding subspace of the posterior or value: the span of the cross-covariance between the activations and that quantity's analytic targets. Appendix K gives each intervention's construction, dose, and controls. We define the value and posterior write sites as the earliest layer at which the steer reaches 95% of its largest effect across layers (Appendix E).

Posterior steering targets are the analytic posteriors of other episodes, sampled to be distant from the episode's own (Appendix K), so every analytic target lies within the probability simplex. We evaluate the causal effect of steering the representation by comparing the post-steer logit shift to an exact counterfactual: the logit shift the target policy would exhibit if the posterior were set to the analytic target. The observed shift follows the exact counterfactual at slope 0.979 (factor M) and 0.982 (factor N), while matched isometric off-plane steering moves the choice logit at 0.061 and 0.078 (Figure 3a). In the toy model the posterior write site coincides with the posterior read site, layer 0. The edit also carries through the network's internal value representations. We compare the decoded shift in $(Q(A), Q(B))$ at the value site to the shift implied by Eq. (1), measuring the Euclidean dis-

tance between them. We find a median distance of 0.034 and 0.047 payoff units for factors M and N, respectively, against 0.349 and 0.639 for off-plane steering (Figure 3b–c, Appendix K).

We next steer the value subspace directly. Choice responds to the value gap alone, so a write of the displacement $(\Delta y_A, \Delta y_B)$ should shift the choice logit by $\beta\,(\Delta y_A - \Delta y_B)$. Choice should change maximally along the gap axis and not at all for directions in which both offers are raised equally. We steer the value encoding subspace in twelve equally spaced directions at the value write site, scaling the steering vector to ensure that it remains within a valid range of the subspace (Appendix K). We find that the steering-induced shift of the logits follows the predicted shift at a pooled slope of 0.944, against a slope below 0.01 for a matched-norm sweep in the orthogonal complement (Figure 3d). We find a minimal effect on the decoded posterior at the final layer after value steering, with a shift of 0.024 probability units against 0.330 under a posterior-plane sweep of either factor, indicating that the value steer leaves the posterior representation unaffected. Next, we show shifts in the value encoding plane mediate the effect of posterior steering. Inserting only the value-plane component of a posterior steer reproduces the full behavioral effect (slope $0.956/0.960$) and removing it abolishes the effect ($0.044/0.039$, Figure 3e). We test whether steering the payoff mass or offer probability directly, while holding the offer value fixed, affects choice. We find no such change, with slopes of -0.010 and -0.002, respectively, against controls of -0.004 and 0.003 (Appendix K, G). Thus, both the posterior and value representations are causal for choice, with the value representation mediating the effect of the posterior.

3.5 The same organization forms in a large language model

The full set of analyses was replicated in a large language model, Qwen3-1.7B-Base, a 28-layer pretrained transformer with a 2,048-dimensional residual stream, using the same world, objective, trainer, and site rules (Appendix C). Every result of Sections 3.1 to 3.4 holds (Table 2, Figures 4, 7 and 8). There are two notable differences between the toy model and Qwen3. The steering slopes sit below the toy's, 0.79 against 0.94 for the value sweep, while every control stays near zero. The residual stream of Qwen3 also carries the offer probability and the payoff mass of an offer as coordinates of their own, which the toy does not. Probes applied directly to the offer probability and payoff mass reach an R^2 of 0.89 and 0.95, respectively, against the toy's 0.31 and 0.74. When we regress the payoff mass on the offer value and decode the residual, the R^2 is 0.84, against the toy's 0.12, indicating that the payoff mass is represented beyond what the offer value can explain. However, the offer probability and the payoff mass representations in Qwen3 are not causal for choice. Steering that moves the payoff mass or offer probability while holding the offer value fixed leaves choice unmoved (slopes 0.017 and 0.007 against controls of 0.000 and 0.001, Appendix K).

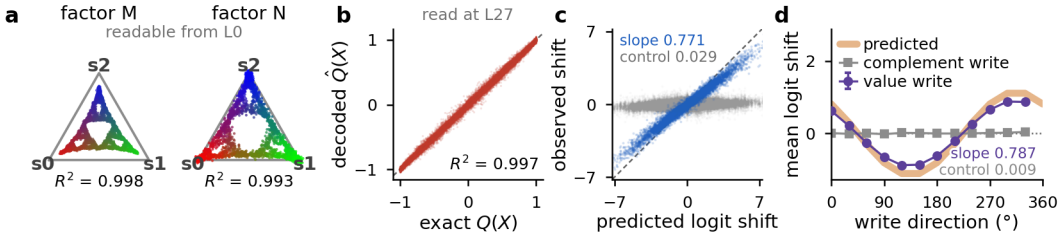

Figure 4: Replication in a large language model. Qwen3-1.7B-Base, trained using the same world and objective, replicates the organization of the toy model. (a) Linear projections of the residual stream at layer 0 recover the two factor posteriors. Points are colored by the ground-truth factor probabilities. (b) Directly decoded offer value against its exact value over the 64 offers never presented during training, read at the final layer, layer 27. R^2 is pooled over the held-out offers. (c) Editing the represented posterior (factor M) at layer 14 shifts choice as the exact counterfactual predicts. Matched off-plane edits do not. (d) Value-plane edits along twelve directions at layer 19 shift choice by the predicted $\beta\Delta(Q_A - Q_B)$. Matched orthogonal edits do not (bars: 95% CI).

4 A DECISION-THEORETIC READING

Neural networks are often discussed as if they are agents with beliefs and desires. A model is said to *believe* it is being evaluated, *want* to complete its objective, or *deceive* us (Dennett, 1989; Goldstein & Lederman, 2025; Greenblatt et al., 2024; Schoen et al., 2025). Yet it has not been established whether a network internally represents beliefs and desires and uses them to act, or whether the vocabulary is a projection of our own self-description onto its behavior (Herrmann & Levinstein, 2025; Chalmers, 2025). Behavior alone cannot settle the question, because different belief–desire pairs rationalize the same preferences (Herrmann & Levinstein, 2026; Armstrong & Mindermann, 2018). Interpreting model activations therefore provides a means of testing whether decision-theoretic definitions of belief and desire apply to transformers.

The Jeffrey–Bolker description of an agent (Bolker, 1967; Jeffrey, 1990) gives this vocabulary a precise definition that can be mapped onto the task. Where standard expected utility theory (von Neumann & Morgenstern, 1944; Savage, 1972) attaches utility to outcomes and probability to states, in the Jeffrey–Bolker form both attach to propositions: the desirability of a proposition is its expected utility on the assumption that it is true, so desire is defined in terms of belief. The task has this form built in. An offer is a proposition about the latent state, true whenever the state is one of those it contains. Choosing it samples from the posterior restricted to it, and its value is its Jeffrey–Bolker desirability. With the joint posterior $\eta = \eta_M \otimes \eta_N$ as the probability over states, the desirability of X is $\mathbb{E}[u(S) \mid S \in X] = \sum_{s \in X} \eta_s \, u(s)/P(X)$, which is Eq. (1). The belief is thus the joint posterior η, the measure over the states from which the probability $P(X)$ of any proposition is derived, and the desire for a proposition is its offer value $Q(X)$. In this description the two attitudes are jointly constrained, and neither can be attributed in isolation, since a direction that tracks the true state counts as a belief only if it combines with a desire to drive behavior (Herrmann & Levinstein, 2026).

We chose the three properties of Section 2.1 as prerequisites for attributing belief and desire to a network's internal states, aligning with the standards for belief representation proposed by Herrmann & Levinstein (2026). Readability is accuracy under a linear probe, causal organization is use, and dissociability is what lets the two attitudes be attributed separately. The results satisfy all three. To our knowledge this is the first case in which both attitudes are measured in one network against ground truth and causally linked, with the desire computed from the belief and driving choice. Had a posterior steer moved choice without moving the decoded offer values, or had removing the value-plane component of a posterior steer left its effect on choice intact, the posterior would have moved choice by some route other than the value computed from it, and no desire computed from the belief would have been driving choice (Section 3.4). The three prerequisites are necessary conditions for the attribution, but not sufficient.

5 DISCUSSION AND CONCLUSION

A transformer trained only on the outcomes of its choices in a simple decision task develops separable and causally used representations of the posterior over latent states and offer value. We find these representations, and the fractal posterior geometry of Shai et al. (2024; 2026), despite the fact that the reward objective only constrains the difference between the offer values (Appendix A). In a preliminary analysis the toy model was first trained via next-token prediction, forming the posterior representations, and then by reward; offer value representations still formed, so this organization does not depend on the posterior being formed by reward alone.

Our results come from one simple decision task rather than a naturalistic setting, and the Qwen3 replication did not present observations and offers in plain language. Furthermore, the payoff of each state was supplied to the model in context rather than learned from reward alone. The probes and encoding subspaces were fit using ground truth values.

If this organization holds in larger models and naturalistic settings, we may be able to monitor, and to intervene on, the process that governs a model's choices.

AI USE STATEMENT

In this work, we used generative AI tools for implementing methods and designing or providing feedback on research methodology or experiments. We have not used generative AI tools for proposing or refining hypotheses, interpreting results, generating synthetic data sets, helping to develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assistance in the writing of proofs, assistance with translation, cleaning and reformatting datasets, or supporting qualitative and thematic data analysis. Additionally, we used generative AI tools to draft and revise prose for clarity. We reviewed all AI-assisted work and take responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The data-generating process is provided in closed form in the appendix. All training values and analysis parameters needed for replication are provided in the appendix. All results from the primary toy model were replicated in four additional seeds and a single seed of a large language model (Qwen3-1.7B-Base). An anonymized repository is provided as supplementary information with this submission along with instructions for using the code.

REFERENCES

Stuart Armstrong and Sören Mindermann. Occam's razor is insufficient to infer the preferences of irrational agents. In *Advances in Neural Information Processing Systems*, volume 31, pp. 5603–5614, 2018.

Karl Johan Åström. Optimal control of Markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10:174–205, 1965.

Ethan D. Bolker. A simultaneous axiomatization of utility and subjective probability. *Philosophy of Science*, 34(4):333–340, 1967.

David J. Chalmers. Propositional interpretability in artificial intelligence. *arXiv preprint arXiv:2501.15740*, 2025.

Daniel C. Dennett. *The Intentional Stance*. MIT Press, 1989.

Simon Goldstein and Harvey Lederman. What does ChatGPT want? An interpretationist guide. Manuscript, 2025.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*, 2024.

Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, 87:96–110, 2014. doi: 10.1016/j.neuroimage.2013.10.067.

Daniel A. Herrmann and Benjamin A. Levinstein. Standards for belief representations in LLMs. *Minds and Machines*, 35(1):5, 2025.

Daniel A. Herrmann and Benjamin A. Levinstein. *Radical AI Interpretability*. Elements in Philosophy of AI. Cambridge University Press, 2026. In press.

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*, 2019.

Richard C. Jeffrey. *The Logic of Decision*. University of Chicago Press, 2nd edition, 1990.

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.

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.

Mantas Mazeika, Xuwang Yin, Rishub Tamirisa, Jaehyuk Lim, Bruce W Lee, Richard Ren, Long Phan, Norman Mu, Oliver Zhang, and Dan Hendrycks. Utility engineering: Analyzing and controlling emergent value systems in ais. *Advances in Neural Information Processing Systems*, 38: 140822–140868, 2026.

Thomas McGrath, Andrei Kapishnikov, Nenad Tomašev, Adam Pearce, Martin Wattenberg, Demis Hassabis, Been Kim, Ulrich Paquet, and Vladimir Kramnik. Acquisition of chess knowledge in AlphaZero. *Proceedings of the National Academy of Sciences*, 119(47):e2206625119, 2022.

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*, pp. 16–30, 2023.

Leonard J. Savage. *The Foundations of Statistics*. Courier Corporation, 1972.

Bronson Schoen et al. Stress testing deliberative alignment for anti-scheming training. *arXiv preprint arXiv:2509.15541*, 2025.

Rohin Shah, Vikrant Varma, Ramana Kumar, Mary Phuong, Victoria Krakovna, Jonathan Uesato, and Zac Kenton. Goal misgeneralization: Why correct specifications aren't enough for correct goals. *arXiv preprint arXiv:2210.01790*, 2022.

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*, volume 37, pp. 75012–75034, 2024.

Richard D. Smallwood and Edward J. Sondik. The optimal control of partially observable Markov processes over a finite horizon. *Operations Research*, 21(5):1071–1088, 1973.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1944.

Jiuqi Wang, Ethan Blaser, Hadi Daneshmand, and Shangtong Zhang. Transformers can learn temporal difference methods for in-context reinforcement learning. In *International Conference on Learning Representations*, 2025.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3):229–256, 1992.

A THE OPTIMUM OF THE TRAINING OBJECTIVE

At one readout write $z \in \mathbb{R}^2$ for the two response logits, $\pi = \mathrm{softmax}(z)$ for the policy, and $Q = (Q_A, Q_B)$ for the exact offer values. Treating the sampled action and reward as fixed when differentiating, the sample gradient of Eq. (2) is the score-function estimator of Williams (1992) and satisfies $\mathbb{E}_{a,r}[\nabla_z \ell] = -\nabla_z J$, where $J(z) = \sum_a \pi_a Q_a + H(\pi)/\beta$ is the entropy-regularized expected reward. This identity concerns the expected sample gradient; it does not identify the expected sample loss with $-J$.

Fixed point. Using the softmax derivative $\partial\pi_a/\partial z_b = \pi_a(\delta_{ab} - \pi_b)$ and the entropy derivative $\partial H/\partial z_b = -\pi_b(\log\pi_b + H)$,

$$\frac{\partial J}{\partial z_b} \;=\; \pi_b\Big[(Q_b - \textstyle\sum_a \pi_a Q_a) \;-\; \frac{1}{\beta}(\log\pi_b + H)\Big]. \tag{3}$$

Since $\pi_b > 0$, stationarity requires $\log\pi_b = \beta Q_b + c$ with c independent of b, so the unique stationary policy is $\pi_\beta = \mathrm{softmax}(\beta Q)$. As a function of π on the simplex, J is strictly concave, since the reward term is linear in π and the entropy is strictly concave, so this policy is its maximum. Adding the same constant to both logits leaves the softmax unchanged, so the objective fixes only the difference $z_A - z_B = \beta\,(Q_A - Q_B)$ and says nothing about z_A or z_B on their own. This is a pointwise optimum over action distributions, not a guarantee of convergence or of unique stationary network weights.

B Data-generating process

The posterior of Section 2.2 is computed in closed form following the approach of Shai et al. (2024; 2026). Tokens are emitted as the latent Markov chain transitions between states. These transitions are governed by token-labeled transition matrices $\{T^{(x)}\}$, where each $T^{(x)}_{ij} = \Pr(x, s_j \mid s_i)$ is the joint probability of emitting token x and transitioning to state s_j, given state s_i. If the posterior over the latent states is η (a row vector), then the update rule for the new posterior η' after seeing token x is

$$\eta' = \frac{\eta\, T^{(x)}}{\eta\, T^{(x)}\mathbf{1}}, \qquad \eta(x_{1:\ell}) = \frac{\eta^{(\emptyset)} T^{(x_1)} \cdots T^{(x_\ell)}}{\eta^{(\emptyset)} T^{(x_1)} \cdots T^{(x_\ell)}\mathbf{1}}, \tag{4}$$

where the second form gives the posterior induced by an entire sequence, starting from the stationary distribution $\eta^{(\emptyset)}$. In our world the generator is composed of two independent factors M and N, each a three-state hidden Markov process. A clue token $x = (a, b)$ carries one emission from each factor, and the joint transition operator can be expressed as the tensor product of the factors' operators,

$$T^{(x)} = T^{(a)}_M \otimes T^{(b)}_N, \tag{5}$$

which is the conditionally independent form of Shai et al. (2026). Because the factors evolve independently and every clue reveals both emissions, the joint posterior never leaves the product-state manifold: $\eta(x_{1:\ell}) = \eta_M(x_{1:\ell}) \otimes \eta_N(x_{1:\ell})$ at every position. Each factor is a Mess3 process (Shai et al., 2024), defined by a hop probability x and an emission accuracy α as in Shai et al. (2026), with $(x, \alpha) = (0.15, 0.60)$ for M and $(0.11, 0.79)$ for N.

C Architecture and training details

The toy model used is a four-layer GPT-style transformer with two attention heads per layer, $d_{\mathrm{model}} = 64$, $d_{\mathrm{mlp}} = 256$, ReLU activations, LayerNorm, learned positional embeddings, and a 91-position context holding exactly one episode, giving 208,081 parameters. Training is 24,000 steps of AdamW (PyTorch defaults, no gradient clipping) at batch size 2,048, with the learning rate held at 10^{-3} for 6,000 steps then decayed linearly to 2×10^{-5} by step 20,000. Every step generates a fresh batch of episodes, so training sees 4.9×10^7 episodes (4.9×10^8 readouts).

The same objective and trainer are applied to Qwen3-1.7B-Base, a 28-layer decoder-only transformer with $d_{\mathrm{model}} = 2{,}048$ and 16 attention heads per layer. Each of the world's 17 symbols is mapped to one token of the pretrained vocabulary, and the embedding rows and unembedding columns are restricted to those 17 tokens, so the model reads the same token stream as the toy. All remaining parameters, 1.41×10^9 in all, are trained. Training is 10,000 steps of AdamW (PyTorch defaults, no gradient clipping) at batch size 512, with the learning rate held at 10^{-5} for 2,500 steps then decayed linearly to 2×10^{-7} by step 8,333. Training consumes 5.1×10^6 episodes; the sampled reward reaches the target policy's level within the first few hundred steps and ends at 0.335 with policy entropy 0.286.

D Linear probes

The linear probe of Section 3.2 is a single affine map from the residual-stream activations $a_i \in \mathbb{R}^{d_{\mathrm{resid}}}$ to eight concatenated targets: the two factor posteriors (three probabilities each) and the two

offer values, $b_i = (\eta_M, \eta_N, Q(A), Q(B))_i \in \mathbb{R}^8$. It is fit by ordinary least squares,

$$\hat{b} = Wa + c, \qquad (W, c) = \arg\min_{W,c} \sum_i \|b_i - (Wa_i + c)\|^2. \tag{6}$$

$R^2 = 1 - \mathrm{SS}_{\mathrm{res}}/\mathrm{SS}_{\mathrm{tot}}$ is pooled over a quantity's coordinates and reported for η_M, η_N and $(Q(A), Q(B))$. Five-fold cross validation is used to evaluate model fit, on the episodes that pass the validity rule of Appendix K.

E Site selection and depth profiles

Read sites are the earliest layer at which the cross-validated R^2 reaches 0.95. Write sites are determined by first steering at each layer and regressing the observed choice-logit shift on the predicted shift. The write site is the earliest layer whose slope reaches 95% of the maximum slope across layers. Every steering result in the paper is then measured on evaluation episodes drawn independently of the fitting episodes. In the toy model the posterior write site is layer 0 and the value write site is layer 2. Posterior steering drops to zero by layer 2, consistent with the posterior being used to compute the offer value rather than acting on choice directly. In the pretrained transformer posterior steering rises to a maximum at layer 14 before falling to zero over layers 15 to 19. Steering value has no effect until layer 14, after which it rises and plateaus from layer 19 (sites in Table 2).

F Formation across training

A fresh probe of the form of Eq. (6) is fit at 23 points during training and validated using five-fold cross-validation. Posterior and values are read at their respective read sites. Figure 5 shows projected activations beside the analytic ground truth. Both quantities cross $R^2 = 0.9$ at step 1,000 of 24,000. The untrained network is the null of Section 3.2. At step 0 the fresh probes read the posterior at 0.197/0.299 and the offer values at 0.093.

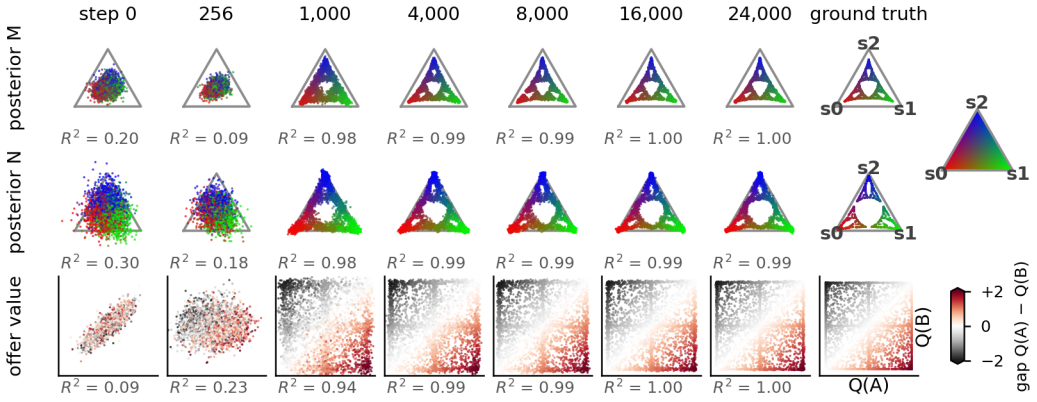

Figure 5: Formation across training. Linear projections of the residual stream onto the two factor posteriors (layer 0) and the offer-value pair (layer 2) at seven training steps, from fresh probes of the form of Eq. (6), beside the analytic ground truth. Each point is one episode, colored by the analytic posterior or the analytic gap $Q(A) - Q(B)$; each printed R^2 is the probe's cross-validated score.

G Direct and composed reads of offer probability and payoff mass

When we discuss posterior representations, we refer to the two factor posteriors (η_M, η_N), from which the joint posterior over the nine states follows. The offer probability $P(X) = \sum_{s \in X} \eta_s$ and the payoff mass $m(X) = \sum_{s \in X} \eta_s\, u(s)$ are sums over that joint posterior, and were not included in the decoder of Section 3.2. To test whether they are nonetheless represented as quantities of their

own, probes of the form of Eq. (6) are fit to $(P(A), P(B))$ and $(m(A), m(B))$ at every layer. Their outputs are the direct reads of $P(X)$ and $m(X)$. The composed read $\hat{P}(X) = \sum_{s \in X} (\hat{\eta}_M \otimes \hat{\eta}_N)_s$ is the offer probability assembled from the decoded factor posteriors, as in Figure 2b. Because $m(X) = P(X)\,Q(X)$, the payoff mass may be partly decodable from the offer value. To account for this, we regress the analytic $m(X)$ on the analytic $Q(X)$ and decode the residual, which we call the payoff mass beyond Q. Table 1 scores every read on the same episodes, at the final layer, per held-out offer. The factor posteriors themselves, read on those episodes, score 0.992 and 0.986, so the direct read of the offer probability falls well short of what its ingredients support.

Table 1: In the toy model the posterior and the offer value, but not the offer probability or the payoff mass, are linearly represented. Per-offer R^2 over the 64 held-out offers, read at the final layer: median and interquartile range. Each quantity is read by an affine probe of the form of Eq. (6) fit to it, except the composed offer probability, which is assembled from the decoded factor posteriors.

Quantity	**Median R^2**	**Interquartile range**
Factor posterior η_M	0.992	0.992–0.992
Factor posterior η_N	0.986	0.985–0.986
Composed offer probability $\hat{P}(X)$	0.985	0.983–0.986
Direct offer probability $P(X)$	0.31	0.10–0.48
Offer value $Q(X)$	0.998	0.998–0.998
Sum of the offer values $Q(A) + Q(B)$	0.998	0.998–0.998
Gap of the offer values $Q(A) - Q(B)$	0.999	0.999–0.999
Payoff mass $m(X)$	0.74	0.68–0.77
Payoff mass beyond Q	0.12	0.02–0.21

H Principal component analysis of the residual stream

To measure how much of the residual stream's variance the posterior and the offer values account for, we apply principal component analysis to decompose the residual stream. We then fit encoding models to identify which components relate to the posterior, to the offer values, and to the card and offers. For each episode, the joint posterior (9 dimensions), offer values $(Q(A), Q(B))$, and the payoff card and offers (the payoff card as 27 one-hot levels and the two standing offers as their 18 mask symbols, 45 dimensions) are used as inputs for the encoding models. We then fit an affine map from these feature groups to each of the principal components. Each map is fit on half the episodes and evaluated on the other half. Four models are fit for each principal component: one for the posterior, one for the offer value, one for the payoff card and offer masks, and one with all of these features concatenated. We then take the R^2 of the combined model, R^2_{all}, as the fraction of that component's variance the features explain, and attribute it to the three groups in proportion to their individual fits:

$$\text{share}_g \;=\; \frac{R^2_g}{\sum_{g'} R^2_{g'}}\, R^2_{\text{all}}, \qquad g \in \{\text{posterior},\ \text{offer values},\ \text{card and offers}\}, \tag{7}$$

with $1 - R^2_{\text{all}}$ unexplained. Each share is then multiplied by the fraction of the total variance that the component carries. Figure 6a plots these values for the first eight components at each layer. At the posterior site, the first four components carry 0.857 of the total variance, and 0.971 of that is attributed to the posterior. These four components form two pairs, one for each factor (Figure 6b). At the final layer, the first two components carry the offer values (Figure 6c).

I Large language model replication

Table 2 lists every result of the replication in Qwen3-1.7B-Base (Section 3.5) beside the toy's. Figures 7 and 8 are the pretrained transformer's counterparts of Figures 2a–c and 3, produced by the same analyses.

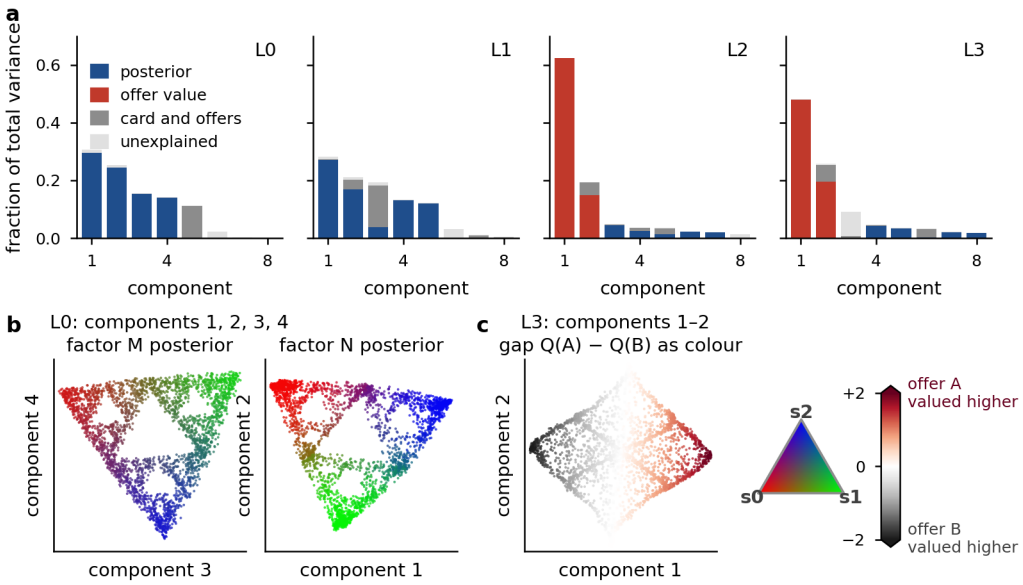

Figure 6: The top principal components encode the posterior and offer values. (a) Variance of the first eight components at each layer, attributed to the posterior, the offer values, and the card and offers as in Appendix H; the unexplained remainder is light grey. (b) The posterior site's leading components, colored by the analytic factor posterior. (c) The final layer's first two components, colored by the analytic gap $Q(A) - Q(B)$.

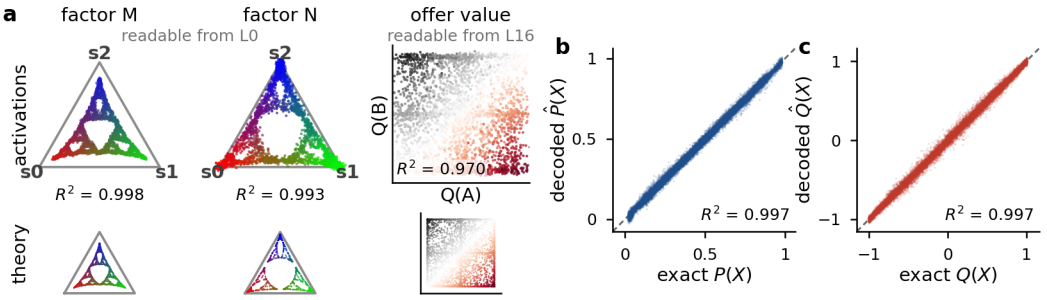

Figure 7: In Qwen3-1.7B-Base, the posterior and the offer values are separately represented and generalize to offers withheld from training. Figure 2a–c for Qwen3-1.7B-Base. (a) Linear projections of the residual stream recover the two factor posteriors (layer 0) and the values of the two offers (layer 16), each read at the earliest layer where its cross-validated R^2 reaches 0.95. Points are colored by the ground-truth factor posterior or offer-value gap $Q(A) - Q(B)$. (b, c) Decoders of the same form, fit at the final layer (layer 27), generalize to the 64 offers never presented during training: the offer probability composed from the decoded factor posteriors (b) and the directly decoded offer value (c) track their analytic counterparts. Dashed lines indicate equality; R^2 is pooled over the held-out offers.

J Replication across seeds

Five toy models were trained from independent initializations under the settings of Appendix C; the paper reports the first. Table 3 lists every quantity of Table 2 for all five, each read and steered at its own sites.

Table 2: Comparison between toy model and Qwen3. Linear reads are at each quantity's read site. Each slope is the regression of the observed choice-logit shift on the predicted shift, the exact counterfactual for the posterior steer and $\beta\,\Delta(Q_A - Q_B)$ for the value steer; the row below gives its matched control. Insertion and removal are the mediation tests of Appendix K. Shifts in $\hat{Q}$ are in payoff units and shifts in $\hat{\eta}$ in probability units (Section 3.3). Withheld-offer reads are at the final layer over the 64 offers never presented during training, pooled over contexts or as the median over offers as marked.

Quantity	**Unit**	**Toy**	**Qwen3**
Behavior			
Choice vs. target policy	Pearson r	0.999	0.999
Linear reads at the read sites			
Factor posteriors (M, N)	CV R^2	0.997, 0.994	0.998, 0.993
Offer values	CV R^2	0.999	0.968
Value sum $Q(A) + Q(B)$	CV R^2	0.999	0.967
Value gap $Q(A) - Q(B)$	CV R^2	0.999	0.969
Withheld offers, composed $\hat{P}(X)$	pooled R^2	0.989	0.997
Withheld offers, $\hat{Q}(X)$	pooled R^2	0.998	0.997
Withheld offers, direct $P(X)$	median R^2	0.31	0.89
Withheld offers, $m(X)$	median R^2	0.74	0.95
Withheld offers, $m(X)$ beyond Q	median R^2	0.12	0.84
Dissociation			
Card twin, offer value	$\hat{Q}$ shift	1.03	1.06
Card twin, posterior	$\hat{\eta}$ shift	0.004	0.011
Mask twin, offer value	$\hat{Q}$ shift	0.91	0.88
Mask twin, posterior	$\hat{\eta}$ shift	0.016	0.012
Clue twin, posterior	$\hat{\eta}$ shift	0.885	0.883
Clue twin, offer value	$\hat{Q}$ shift	0.496	0.515
Steering			
Posterior steer (M, N)	slope	0.979, 0.982	0.771, 0.801
off-plane control	slope	0.061, 0.078	0.029, 0.029
Value endpoint error (M, N)	$\hat{Q}$ error, median	0.034, 0.047	0.18, 0.25
off-plane control	$\hat{Q}$ error, median	0.349, 0.639	1.57, 1.49
Value steer	slope	0.944	0.787
off-plane control	slope	0.007	0.009
Posterior under value steer	$\hat{\eta}$ shift, RMS	0.024	0.019
under posterior steer	$\hat{\eta}$ shift, RMS	0.330	0.277
Mediation, insertion (M, N)	slope	0.956, 0.960	0.892, 0.899
Mediation, removal (M, N)	slope	0.044, 0.039	0.114, 0.109
Sites			
Read sites (posterior, value)	layer	0, 2	0, 16
Write sites (M, N, value)	layer	0, 0, 2	14, 11, 19

K STEERING PROCEDURES

Every intervention of Section 3.4 is written through the encoding plane of a quantity, Φ, an orthonormal basis of the span of the cross-covariance between the activations and its two analytic targets. A target displacement Δy of the probe read is written as

$$\delta \;=\; \Phi\, M^{-1}\, \Delta y, \qquad M \;=\; W_y\, \Phi, \tag{8}$$

where W_y is the linear part of a two-target affine probe of the form of Eq. (6) fit at the write site, so that the probe reads back exactly Δy. The targets are the factor posterior's two coordinates in the simplex plane for posterior steering and $(Q(A), Q(B))$ for value steering. The write goes through Φ rather than along the probe's weights, since the probe's weights need not align with the direction

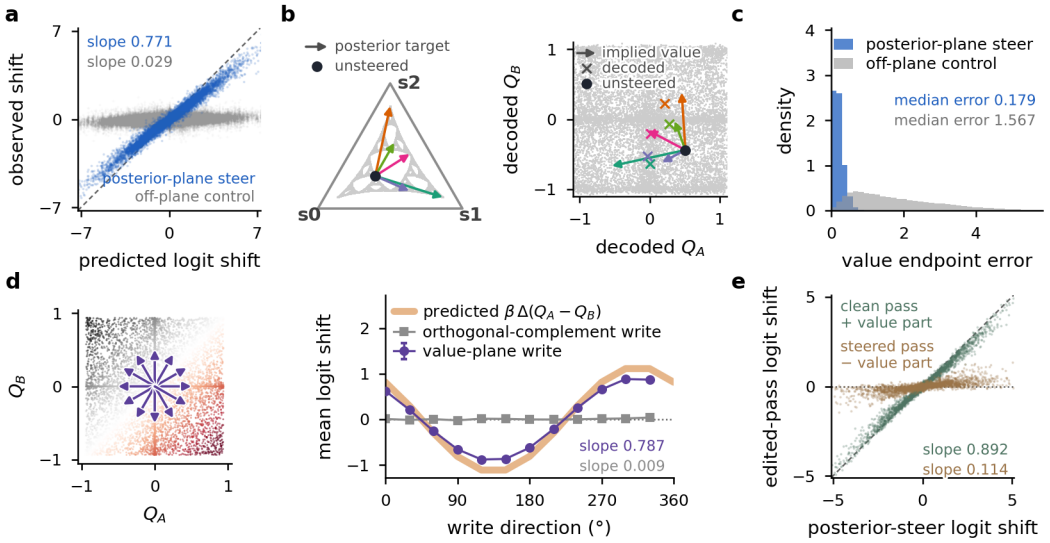

Figure 8: In Qwen3-1.7B-Base, posterior and value representations move choice. Figure 3 for Qwen3-1.7B-Base. (a) Editing the represented posterior (factor M) at layer 14 shifts choice in proportion to the exact counterfactual (slope 0.771); matched off-plane edits do not (0.029). (b) Posterior steering shifts offer value representations toward the values implied by the posterior target. Example showing five steering directions applied to a single episode. Left: steering targets in the posterior simplex. Right: decoded value at layer 16 versus value implied by posterior steering targets. (c) Distribution of value endpoint error across episodes: posterior-plane steering approaches the predicted value (median 0.18), while off-plane controls do not (1.57). (d) Value-plane edits along twelve directions at layer 19 shift choice in proportion to the predicted $\beta\Delta(Q_A - Q_B)$ (slope 0.787); matched orthogonal edits do not (0.009; bars: 95% CI). (e) The value-plane component of a posterior edit is sufficient (insertion, 0.892) and largely necessary (removal, 0.114) for its effect.

along which a variable is expressed in the activations (Haufe et al., 2014). Planes and their probes are fit on the first half of the decoder-fitting episodes, the first 2,048 of 4,096 for the toy and the first 8,192 of 16,384 for Qwen3. Edits are evaluated on 16,384 fresh episodes drawn independently of the fitting episodes. Because the division by $P(X)$ in Eq. (1) is ill-conditioned at small P, episodes are used only when both offers have probability at least 0.02, both in fitting the value plane and in every evaluation. For posterior steers this must hold both before the steer and under the target posterior.

Posterior steering. Each episode receives one target posterior, drawn uniformly from the farthest half of the fitting episodes' posteriors by distance in the simplex plane, and is written with the displacement from its analytic posterior to that target. The observed choice-logit shift is regressed on the exact counterfactual $\beta\,\Delta(Q_A - Q_B)$, recomputed with the steered factor's posterior replaced by the target and the other factor unchanged. The control writes the same displacements, norms preserved, into twelve random rank-2 planes orthogonal to the posterior plane, and a single slope is fit to all twelve. The endpoint error of Figure 3b–c is the distance between the decoded $(Q(A), Q(B))$ shift at the value site and the shift the target posterior implies, taken as a median over episodes.

Value direction sweep. Twelve equally spaced directions in the $(Q(A), Q(B))$ plane are written at the value write site, each episode at a radius that keeps the written values within the attainable range $[-1, 1]^2$: 0.8 of the episode's distance to the boundary, capped at 0.25 payoff units, with episodes whose radius falls below 0.05 dropped. The predicted choice-logit shift is $\beta\,(\Delta y_A - \Delta y_B)$. The control writes the same norms along twelve directions drawn in the orthogonal complement of the value plane. To check that value steering leaves the posterior representation unmoved, the posterior is decoded at the final layer after each write. As a positive control, a direction sweep is done within each factor's posterior encoding subspace at that factor's write site, in which the radius is set to the mean per-coordinate standard deviation of the fitting episodes' posteriors in the simplex plane.

Table 3: Toy model replication across five seeds. Every quantity of Table 2 for the toy model reported throughout the paper and four further models trained from independent initializations under identical settings. Each model is read and steered at its own sites. Units as in Table 2.

Quantity	**Unit**	**Toy**	**Seed 1**	**Seed 2**	**Seed 3**	**Seed 4**
Behavior						
Choice vs. target policy	Pearson r	0.999	0.999	0.999	0.999	0.999
Linear reads at the read sites						
Factor posterior M	CV R^2	0.997	0.996	0.997	0.997	0.997
Factor posterior N	CV R^2	0.994	0.994	0.995	0.994	0.995
Offer values	CV R^2	0.999	0.998	0.957	0.998	0.969
Value sum $Q(A) + Q(B)$	CV R^2	0.999	0.997	0.956	0.998	0.968
Value gap $Q(A) - Q(B)$	CV R^2	0.999	0.998	0.957	0.998	0.969
Withheld offers, composed $\hat{P}(X)$	pooled R^2	0.989	0.986	0.989	0.983	0.988
Withheld offers, $\hat{Q}(X)$	pooled R^2	0.998	0.997	0.997	0.997	0.998
Withheld offers, direct $P(X)$	median R^2	0.31	0.52	0.29	−0.14	0.00
Withheld offers, $m(X)$	median R^2	0.74	0.74	0.72	0.70	0.73
Withheld offers, $m(X)$ beyond Q	median R^2	0.12	0.19	0.07	0.08	0.10
Dissociation						
Card twin, offer value	$\hat{Q}$ shift	1.03	1.04	1.03	1.06	1.05
Card twin, posterior	$\hat{\eta}$ shift	0.004	0.007	0.007	0.005	0.007
Mask twin, offer value	$\hat{Q}$ shift	0.91	0.87	0.85	0.89	0.85
Mask twin, posterior	$\hat{\eta}$ shift	0.016	0.030	0.018	0.019	0.009
Clue twin, posterior	$\hat{\eta}$ shift	0.885	0.895	0.889	0.901	0.882
Clue twin, offer value	$\hat{Q}$ shift	0.496	0.505	0.477	0.501	0.519
Steering						
Posterior steer, M	slope	0.979	0.963	0.964	0.965	0.964
off-plane control, M	slope	0.061	0.073	0.073	0.084	0.093
Posterior steer, N	slope	0.982	0.977	0.976	0.975	0.980
off-plane control, N	slope	0.078	0.093	0.087	0.096	0.111
Value endpoint error, M	$\hat{Q}$ error, median	0.034	0.044	0.076	0.042	0.095
off-plane control, M	$\hat{Q}$ error, median	0.349	0.403	0.630	0.370	0.704
Value endpoint error, N	$\hat{Q}$ error, median	0.047	0.059	0.110	0.059	0.134
off-plane control, N	$\hat{Q}$ error, median	0.639	0.686	1.176	0.663	1.093
Value steer	slope	0.944	0.944	0.978	0.896	0.939
off-plane control	slope	0.007	−0.016	0.006	0.017	0.006
Posterior under value steer	$\hat{\eta}$ shift, RMS	0.024	0.004	0.005	0.018	0.005
under posterior steer	$\hat{\eta}$ shift, RMS	0.330	0.327	0.327	0.330	0.326
Mediation, insertion, M	slope	0.956	0.933	0.943	0.939	0.922
Mediation, insertion, N	slope	0.960	0.950	0.953	0.933	0.947
Mediation, removal, M	slope	0.044	0.069	0.059	0.061	0.078
Mediation, removal, N	slope	0.039	0.048	0.048	0.064	0.057
Sites						
Posterior read and write (M, N)	layer	0	0	0	0	0
Value read	layer	2	3	2	2	1
Value write	layer	2	3	3	2	3

Payoff mass and offer probability steering. To test whether the network uses the payoff mass and the offer probability, two four-dimensional encoding subspaces are fit at the value write site by the construction of Eq. (8), each with four targets: the two offer values and the two payoff masses beyond Q of Appendix G, or the two offer values and the two offer probabilities. These are fit on all decoder-fitting episodes with both offers at probability at least 0.02 and evaluated on the first $1{,}024$ evaluation episodes. In each subspace, twelve directions are written in the payoff-mass or offer-

probability coordinates, while the two offer-value coordinates of the displacement are set to zero. Each episode's radius is set by the same rule as in the value sweep: 0.8 of the episode's distance to the boundary of the attainable range of offer values, capped at 0.25 payoff units, and dropping episodes whose radius falls below $.05$. The choice-logit shift is then regressed on $\beta\,(\Delta y_A - \Delta y_B)$, with Δy the written payoff-mass or offer-probability displacement. A matched-norm control writes the same displacements in the subspace's orthogonal complement. A positive control sweeps the subspace's two offer-value coordinates instead, with the payoff-mass or offer-probability coordinates set to zero; it reproduces the value-sweep slope, 0.942 against 0.944 in the toy and 0.786 against 0.787 in Qwen3. These results are consistent across all layers tested.

Mediation. Steering the posterior shifts the downstream value representation. To test whether that shift is the mechanism through which posterior steering affects choice, an insertion and a removal intervention are performed. For both, the posterior steer is first run and the change it produces in the value encoding subspace at the value write site is recorded, along with the choice-logit shift. For insertion, the same episodes are run without steering the posterior, the recorded change in the value subspace is added to the residual stream, and the resulting logit shift is compared with that of the steered pass. For removal, the steered pass is run again with the recorded change subtracted, and its logit shift is compared with that of the steered pass. As a control, steering vectors on a plane orthogonal to the value subspace, rescaled to the same norm, are used in place of the value subspace.