BAYESIAN FINE-TUNING YIELDS LANGUAGE MODELS THAT ARE AS BAYESIAN AS THEIR BELIEFS ALLOW

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models (LMs) are increasingly used for tasks that require reasoning about hidden variables from a few observations, for which Bayesian inference is the normatively correct solution. While supervised fine-tuning of an LM on the outputs of an optimal *Bayesian* model leads to near-Bayesian behavior, standard supervised fine-tuning (SFT) on the true answers for the task falls short of it. But behavior alone does not tell us *why* tuning on a *Bayesian* or an *oracle* (true answers) signal differs: whether the resulting LM represents Bayesian beliefs, acts on them, or turns them into a choice the way Bayes' rule does. To compare them, we formulate increasingly demanding requirements for an LM to count as a Bayesian decision maker, spanning its behavior, representations, and computations, and test them on a flight recommendation task. The Bayes-trained LM acts Bayesian, encodes quantities of Bayes' rule in its middle layers, and uses the encoded belief for the recommendation to a certain extent. The oracle-trained LM differs from it both in the beliefs it holds and whether it reads beliefs out into recommendations. Exchanging beliefs between the LMs transfers a part of the Bayesian advantage. Bayes fine-tuning thus installs usable Bayesian beliefs in an LM for reasoning under uncertainty in a way standard SFT on oracle answers cannot, highlighting the advantage of nuanced supervision.

1 INTRODUCTION

Language models (LMs) excel at many tasks, including those that require reasoning under uncertainty, such as inferring a user's latent preferences to make adequate action recommendations, as in the task in focus in this paper (Brown et al., 2020; Shao et al., 2024). While pretrained LMs often fail on such tasks (Pournemat et al., 2025; Samanta et al., 2026), and standard supervised fine-tuning on *golden* task labels (SFT) only leads to suboptimal performance, specific *Bayesian* supervised fine-tuning can produce LMs with behavior that matches a Bayesian decision maker *superficially* (Qiu et al., 2026). However, superficial Bayes-like behavior does not guarantee following the hierarchical layering of computational steps suggested by normative Bayesian theory (see Figure 1): choosing an action (layer L1), based on a policy derived from the expected utility over relevant alternatives (L2), which is based on uncertain beliefs informed by the observational evidence (L3). In principle, an LM could show Bayesian behavior (L1) without showing internal Bayesian computation at "deeper" levels (L2 or L3), especially since in the normative Bayesian framework the mappings from beliefs to policies, and from policies to best choices are many-to-one. Reversely, though perhaps less likely *a priori*, as each computational step presupposes the previous one, an LM may also master the earlier steps but fail to reach the correct action. Understanding these computational steps in LMs is not only important for predictability (e.g., Mitchell, 2026) but crucially understudied.

We address this gap and ask here how consistently LMs follow these layers of Bayesian decision making. We test each layer with behavioral evaluation, probing of representations, and causal interventions, organized as four increasingly demanding requirements that, if all met, would constitute converging evidence for genuine computation of Bayes' rule in an LM (Figure 1, right): that the LM **acts Bayesian** (**R1**, § 3), that it **represents** the beliefs Bayes' rule prescribes (**R2**, § 4), that it causally **acts on those beliefs** (**R3**, § 5), and that it **computes** a choice policy in a Bayes optimal way using these beliefs (**R4**, § 6). We apply these tests to a triple of LMs from Qiu et al. (2026): an unmodified pretrained model (the STARTLM), the BAYESLM, fine-tuned to match the flight recommendations of an ideal Bayesian decision maker (BAYESASSIST), and the ORACLELM, fine-tuned

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.

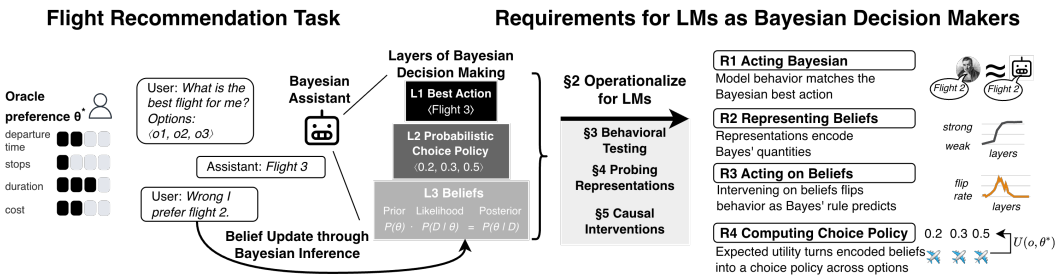

Figure 1: **Overview of our framework for analyzing to which extent LMs instantiate Bayesian inference**. We test the framework on a flight recommendation task (left), which is optimally solved through Bayesian decision making (middle). We operationalize the consistency with Bayes-optimal decision making in LMs through four increasingly fine-grained requirements capturing different layers of Bayesian inference (right), showing that LMs are less consistent with increasing requirements.

on the true flight preference (oracle). The two fine-tuned models share the STARTLM, the task, and the fine-tuning procedure, and differ only in the training signal. Comparing them against R1–R4 thus lets us locate where oracle (ORACLELM) diverges from Bayesian fine-tuning (BAYESLM). If the BAYESLM exhibits genuine Bayesian decision making, we expect it to be more Bayes-consistent than the ORACLELM not only in its behavior but also in its representations and computations.

At R1 and R2, we find that BAYESLM, and ORACLELM to a lesser degree, does match the BAYESASSIST: its best action (L1), its choice policy (L2), and the quantities of Bayes' rule encoded in its representations (L3) are all Bayes-consistent. Testing R3, we find partial consistency only in BAYESLM. Causal interventions let us inject a specific prior about the user's preferences, most effectively in the layers identified under R2, but no single token or layer carries the effect. Testing R4 via plugging LMs' beliefs into the explicit assistant, we find that the step from beliefs (L3) to choice policy (L2) follows expected utility for the BAYESLM more than for the ORACLELM: the BAYESLM is as Bayes-accurate as its beliefs allow, while ORACLELM's beliefs carry more information than is read-out for behavioral predictions. Patching beliefs between the two models confirms this: the BAYESLM's beliefs raise the ORACLELM's Bayes accuracy, but not to the same level as BAYESLM. **Thus, Bayesian fine-tuning installs Bayes-consistent beliefs and a read-out that forms a more optimal choice policy that oracle fine-tuning does not install.**[1]

2 EXPERIMENTAL SET-UP

We use the flight recommendation task and the models of Qiu et al. (2026). This section introduces the task, the normatively correct Bayesian assistant, the language models to be investigated, as well as target measures, terminology, notation and the test sets used in the following analysis sections.

Recommendation task. The flight recommendation task of Qiu et al. (2026) has two actors: the user and the assistant. Each trial consists of five consecutive rounds, $D_i = \langle d_1, \ldots, d_i \rangle$ for $1 \leq i \leq 5$, where the single data point $d_i = \langle O_i, c_i \rangle$ for round i, and where the assistant recommends one of three flight options $O_i = \{o_1, o_2, o_3\}$ for the user in each round (Figure 1, left). Each flight option is represented as a four-place vector $o_j \in \mathbb{R}^4$ of numerical values for each of the four features (departure time, duration, number of stops and a price), and the user has some weighted *preference* θ over these features, which are unknown to the assistant. Following Qiu et al. (2026), we only consider preference values from the set $\{-1, -0.5, 0, 0.5, 1\}$, with Θ as the set of all valid preference value vectors. The assistant observes the user's most preferred option $c_i \in O_i$ at the end of each round i, which allows the assistant to sequentially update their beliefs about the user's latent preference, which is held constant. Each c_i is the outcome of (strict) maximization of utility for a fixed θ. The *utility* $U(o_j, \theta)$ of option o_j under preference θ is given by the vector dot-product $U(o_j, \theta) = o_j \cdot \theta = \sum_{k=1}^{4} o_{jk}\, \theta_k$.

Bayesian Assistant. Since the user's preferences are not directly observable, the optimal BAYESASSIST captures *uncertainty* about them as a *probabilistic belief about the latent prefer-*

[1] All materials are available at `https://anonymous.4open.science/r/bayes-paper-3CF2`.

ence $P \in \Delta(\Theta)$. This belief is updated based on D_i by computing the *posterior beliefs* $P(\theta \mid D_i)$ via Bayes' rule. To capture the round-based task structure, the Bayesian posterior can be written as a sequence of belief updates via recursion over sets $D_{\leq i}$ containing the first i rounds:

$$P(\theta \mid D_{\leq i}) \propto P(c_i \mid o_i, \theta) \times P(\theta \mid D_{\leq i-1}), \quad i = 1, \ldots, 5, \text{ with } P(\theta \mid D_{\leq 0}) := P(\theta) \qquad (1)$$

Throughout, we use the notation "belief$_i$" to indicate the posterior beliefs after seeing i data points. So, the initial prior beliefs $P(\theta)$ before any observation of user choice is written as "belief$_0$". Note that belief$_i$ is also the assistant's *prior* for making a recommendation at round $i + 1$.

Given any belief P_Θ about preferences and a triple of options $\langle o_1, o_2, o_3 \rangle$, the *expected utility* of option o_j is defined as $\text{EU}(o_j, P_\Theta) = \sum_\theta P_\Theta(\theta) U(o_j, \theta)$.[2] We consider the *Bayesian policy* for (approximately) rational choice under softmax-optimization of EU, with scalable rationality parameter β. Notice that the softmax operation entails that the ratio of probabilities assigned to any two options is a function of the difference in *EU* between them, so that soft policies predict mistakes when the differences between best and second-best options are small (which are more or less pronounced depending on the softmax parameter β). The (strict) *Bayes-optimal recommendation* is the (in our case: unique) option o_i with positive probability under the Bayesian policy with $\beta \to \infty$.

Language models. We test three related LMs from Qiu et al. (2026). A STARTLM, the BAYESLM fine-tuned on the BAYESASSIST's recommendations at each round, and the ORACLELM fine-tuned on the oracle (the true user choice). Comparing the latter two isolates the effect of the standard oracle signal against the Bayesian one. We focus on Gemma-2 9B (Team et al., 2024) and report Llama-3.1 8B (Grattafiori et al., 2024) and Qwen-2.5 7B (Qwen et al., 2025) in Appendix H.

LMs' choices, policies and beliefs. We compare the predictions of the different LMs against the normative BAYESASSIST at three different layers of analysis (see Figure 1). **L1** is the *best action*, the option the policy recommends; **L2** is the *choice policy* that turns P_Θ into a probability distribution over the three flight options; **L3** are the *beliefs* P_Θ about the user's latent preference, given by the prior, likelihood and posterior of Bayes' rule. For the BAYESASSIST, L1 to L3 are given. For the LMs, the best action (L1) is the arg-max of their policy. The *LM policy* (L2) is the conditional probability the LM assigns to the flight option tokens ("1", "2", or "3") after the prompt that corresponds to the data points $D_{\leq i}$. The LMs' beliefs (L3) are not transparent, but are here obtained via linear probes and referred to as *decoded beliefs*. In doing so, we linearly probe also for LMs' representations of the *decoded likelihood* (see below). We also obtain the *decoded policy* and the *decoded best action* by probing the LM's choice policy. In this way, we obtain an internal counterpart to L1 and L2 to check whether and how LMs align internally with what they emit.

Measures of accuracy and consistency. We account for absolute goodness-of-fit via corresponding measures across the three layers of Bayesian decision making. On L1, we call an LM recommendation *Bayes correct* if it matches the Bayesian recommendation, and *Oracle correct* if it matches the true user choice, allowing us to calculate *Bayes accuracy* or *Oracle accuracy*, accordingly. Note that the BAYESASSIST's oracle accuracy is not perfect (see Appendix C). At L2, we consider the *(decoded) policy divergence* (for a given β) as the KL-divergence between the BAYESASSIST's policy for softmax parameter β and the LM (decoded) policy. For L3, the *belief divergence* is the KL-divergence between the BAYESASSIST's beliefs and the LM's decoded belief.

Test sets. We evaluate these models on four test sets that isolate a single property of Bayes' rule. The *original* set, held out from fine-tuning (Qiu et al., 2026), tests whether a measure matches the assistant at all, and the others additionally test for Bayes consistency. The *shuffled* set reorders the evidence within the first four rounds and tests the expected invariance to evidence order. On the *low noise* (one user's choices is random), we expect small effects as the assistant's beliefs mostly recover by round 5; on the *high noise* set (three random choices), we expect a substantial effect where the assistant remains uncertain. As predicted by the BAYESASSIST and confirmed by comparing the BAYESLM's performance on original and shuffled datasets, the models are invariant to the order of evidence. Accordingly, the location of noise injections in the noisy test sets is not a relevant factor.

[2]Qiu et al. (2026) assume that each feature dimension is treated independently in P_Θ, so expected utility maximization reduces to a best response to the feature-wise mean beliefs.

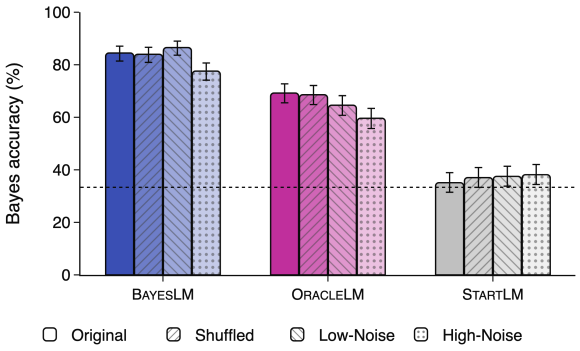

Figure 2: Bayes accuracy on the four test sets, for the BAYESLM, ORACLELM, and STARTLM. Chance agreement is at $1/3$ (dashed line).

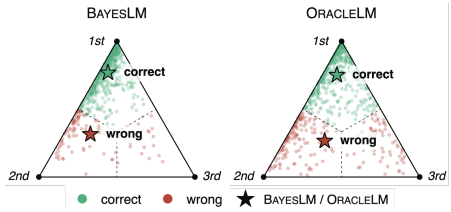

Figure 3: LMs' policies (dots) and their average (star) on the *original* set, in the simplex based on BAYESASSIST rankings (top corner: most probable choice), colored by Bayes-correct (green) or not (red).

3 ACTING BAYESIAN: BEHAVIORAL ANALYSIS

To address the behavioral requirement **R1**, we investigate the Bayes accuracy and the policy divergence of BAYESLM (L1, L2) and compare it with ORACLELM, and STARTLM as a lower bound.

Bayes accuracy is consistent across test sets. We compare the evaluated LMs with the BAYESASSIST behaviorally across all four test sets. While we expect no difference between *original* and *shuffled* for BAYESLM, we do expect that noisy evidence should decrease its Bayes accuracy, but only under *high-noise*. In line with Qiu et al. (2026), we also expect that ORACLELM is less Bayes-accurate but have no expectations about its Bayes-consistency. We test for significant effects by regressing Bayes correctness on the main effects of the LM, the test set, and their interaction.[3]

Results on the *original* set in Figure 2 show that BAYESLM achieves a significantly ($p < 10^{-4}$) higher Bayes accuracy of 0.84 than ORACLELM (0.69), replicating the results in Qiu et al. (2026). For BAYESLM and ORACLELM, we find that the accuracy did not significantly differ when comparing results on *original* with *shuffled* or *low-noise* sets ($p > 0.05$). Also, the accuracy on the *high-noise* set is significantly lower for them ($p = 0.022$). STARTLM, in contrast, achieves near-chance accuracy across all sets. These results show that both BAYESLM and ORACLELM are Bayes consistent across test sets at the level of the best action (L1), as R1 requires, but Bayesian fine-tuning allows training a model more capable of reasoning under uncertainty than oracle fine-tuning.

Policy divergence is Bayes-coherent. After evaluating the best action (L1), we turn to the choice policy (L2), derived from the LMs' token probabilities, and ask whether it aligns with the Bayesian policy. We calculate the policy divergence for each model, and expect the BAYESLM divergence to stay lower than that of ORACLELM. The LMs' policies are more decisive, i.e., place more probability mass on the chosen option than the assistant, so we report results for the softmax parameter $\beta = 4$, and provide parameter fitting details and discussion in Appendix C. The divergence on instances where the models' predictions were Bayes-correct is 0.10 for BAYESLM, and 0.26 for ORACLELM. On incorrect predictions, the $\beta = 4$ scaling reduced the divergence only for BAYESLM (from 0.42 to 0.38) and increased it for ORACLELM (0.66 to 0.90), suggesting that the LMs differ in the Bayes-consistency of the policies that lead to behaviorally incorrect predictions.

Since the Bayesian policy is a softmax over expected utilities (§ 2), we expect an LM's wrong recommendation to be the BAYESASSIST's second choice rather than its last. Figure 3 confirms this for BAYESLM. When it fails, its choice policy accounts for the second most probable choice of the assistant (left corner) with an average probability of 52%, and the least-probable choice (right corner) with 17%. In contrast, ORACLELM fails more uniformly with an average probability of 48% and 26% on the 2nd and 3rd choices. We expect Bayes-consistent failures to arise particularly when the Bayesian policy prefers no clear single choice, i.e., has higher entropy. To test this expectation, we regress the LMs' Bayes correctness on the entropy of the BAYESASSIST's policy, the factor LM

[3]For all regression tests in this paper we apply p-value correction using Bonferroni-Holm (Holm, 1979) with a significance level $\alpha = 0.05$, reporting all details in Appendix B.

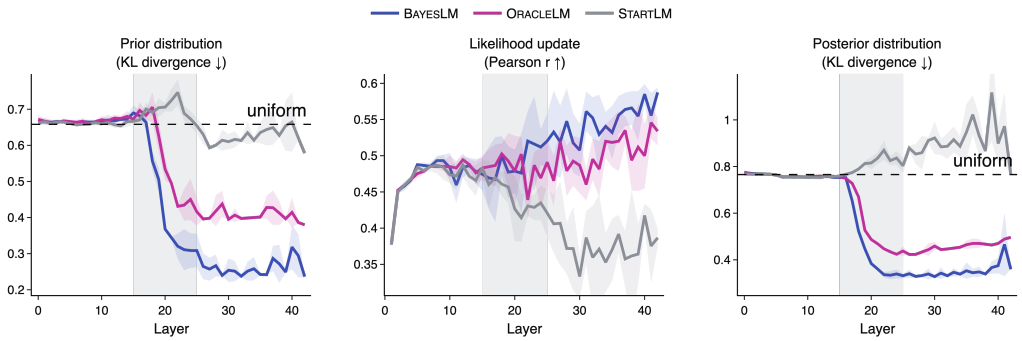

Figure 4: **The quantities of Bayes' rule emerge in the middle layers.** Layer-wise probing for the prior, the likelihood update, and the posterior, each scored against the BAYESASSIST. Dashed lines mark the reference KL between a uniform distribution and the assistant; the grey area marks the layers 15–25 with the main effect; the bands show the deviation across the 20 probes per layer.

and their interaction. Indeed, both fine-tuned models fail more often when the assistant's policy is more uniform ($\beta = -0.49$, $p < 10^{-4}$), and the interaction shows this dependence is stronger for the BAYESLM ($\beta = -0.62$, $p = 0.014$). The BAYESLM's policy entropy also correlates significantly ($p < 10^{-4}$) more strongly with the assistant's (Pearson $r = 0.61$) than the ORACLELM's does ($r = 0.30$), showing that it better captures BAYESASSIST's uncertainty.

In sum, BAYESLM selects the best-action (L1) significantly more Bayes-accurately than ORACLELM, but both models pass consistency criteria across different test sets. We see that the difference is driven by the choice policy (L2), where BAYESLM shows a lower policy divergence than ORACLELM, and fails in a more Bayesian way. Behavior alone, however, cannot tell whether LMs encode beliefs that are consistent with the assistant, which is the question we ask next.

4 REPRESENTING BELIEFS: PROBING ANALYSIS

As incorrect beliefs can potentially lead to the same policy as Bayes-consistent beliefs, a model can agree with the BAYESASSIST at L1 and L2 while diverging from it at L3. We therefore turn to **R2** and ask whether theoretically relevant quantities—prior, likelihood, and posterior—, as well as the policy, are encoded in the model's hidden states in a Bayes-consistent way.

Decoded beliefs are consistent with the assistant. We test whether the prior, likelihood and posterior of the BAYESASSIST can be retrieved via linear probes from an LMs' hidden state $h_{\ell,t} \in \mathbb{R}^{d_{\text{model}}}$ at token position t and layer $\ell \in \{0, \ldots, 42\}$ at the final round 5 from the fine-tuning data of Qiu et al. (2026). We read out at three positions: the *feedback* token of round 4, the *end of input*, and the *digit* position that predicts the flight-number token (see Figure 7). Focusing on belief_4 (prior) and belief_5 (posterior) over each feature j with five possible values, we fit distributional probes with a Kullback–Leibler (KL) loss and report the belief divergence as defined in § 2. We approximate the likelihood term via the magnitude of the belief update, computing the labels for the probes as the penultimate belief update of the BAYESASSIST as $\text{belief}_4 - \text{belief}_3$, and the the LM's representation as the difference between the hidden layers at feedback tokens of the respective turns $h_{l,t_4} - h_{l,t_3}$. We fit a regression probe and report its Pearson correlation with the true update.

If BAYESLM differs from ORACLELM not only superficially, we expect it to encode these quantities more strongly (i.e., lower divergences). STARTLM provides a reference point for what a probe achieves with presumably little to no signal. Indeed, all three quantities become decodable from layers 15–25 on in fine-tuned models (Figure 4). For BAYESLM, the divergence for belief_4 and belief_5 drops to 0.25 and 0.35 at the best layers, and the correlation with the likelihood update rises to $r = 0.59$. The ORACLELM shows the same pattern more weakly, with best values of 0.40, 0.45, and 0.53, while the STARTLM shows none of it. As in § 3, beliefs of both BAYESLM and ORACLELM are order-invariant and respond adequately to noisy input, suggesting that they mainly differ in how strongly they encode beliefs, while both being Bayes-consistent (see Appendix D).

Analyzing the Bayes-consistency of decoded policies. Bayes-consistency at L2 asks for more than the beliefs being present, since the policy those beliefs imply must be encoded as well. We therefore probe the choice policy over the three flight options with the same distributional setup, read out at the digit position (cf. Figure 7). Figure 5 shows the average of the predicted distributions, which start uniform (center of simplex) around layer 15 and leave it during the middle layers, in the same window in which the beliefs arise. Over all conversations, the BAYESLM comes close to the BAYESASSIST's choice policy ($\beta = 1$) by layer 42, while the ORACLELM stops before. Accordingly, the Bayes accuracy of the decoded policy (derived via argmax) rises from chance to 0.87 for the BAYESLM and 0.71 for the ORACLELM, and to only 0.37 for the STARTLM.

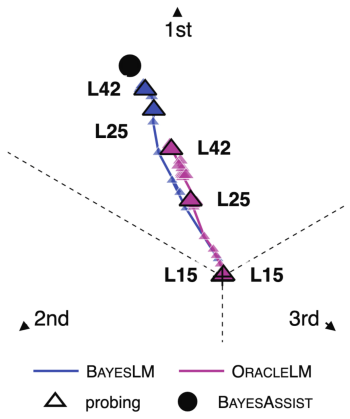

Figure 5: Path of the average decoded choice policy (△) of BAYESLM and ORACLELM and the optimal one under the BAYESASSIST (• top corner).

To derive the choice policy consistently with the softmax optimization of EU in Bayesian decision making (§ 2), LMs need to encode at least the latent belief$_4$ about the user's preference and the resulting choice policy over flight options. Consequently, we may expect the divergence of LMs' actual (token-based) policy to reflect the probing success of the LMs' beliefs and decoded policy. We test this by regressing the (actual) policy divergence on the probed policy (or belief) divergence, the model, and their interaction. As expected, the (actual) policy divergence increases with the decoded policy divergence ($\beta = 0.19, p < 10^{-4}$), and the ORACLELM's policy divergence is significantly worse than that of the BAYESLM ($\beta = -0.11, p < 10^{-4}$), without significant interaction ($\beta = 0.01, p = 0.59$). In contrast, regressing the policy divergence against the decoded belief divergence has no significant effect of belief ($p > 0.5$). The results are visualized in the Appendix (Figure 10).

In sum, these results provide evidence that the representations of BAYESLM are more Bayes-consistent (**R2**) than those of the ORACLELM, and for both, a better-decoded policy tracks the behavioral-policy divergence, and thereby subsequent Bayes accuracy, whereas a better-decoded belief does not. However, this is not evidence that the belief does not matter. A high-entropy policy (exactly when the models fail) could potentially come from an incorrect belief representation resulting in downstream errors, or be the result of correct computations with an option set where all options have close to equal expected utility under the current belief. We therefore cannot tell from the probes whether the belief is causally involved in the model's predictions at all, and if how it is used differs between models. We address these questions next by intervening on beliefs directly.

5 ACTING ON BAYESIAN BELIEFS: INTERVENTION ANALYSIS

A model can carry a Bayesian belief in its hidden states and still reach its answer by another route (Ravichander et al., 2021). So if an LM is Bayesian in its decision-making, its recommendation must use that belief. We test this by injecting a specific prior into the model (Vig et al., 2020; Todd et al., 2023; Ghandeharioun et al., 2024) and checking whether the behavioral recommendation changes to the one the new belief implies (**R3**).

The recommendation follows the belief. To intervene on the belief$_4$ representation, we build a prototypical hidden state v for each of the four features and each value direction (positive/negative) at a given layer, averaging its representations for 500 representative samples (i.e., samples with biggest belief values for the respective feature under the BAYESASSIST) from the training data. We patch this prototype v into the residual stream h at the *feedback* token and the next 32 tokens, with $\alpha = 0.5$ across 8 subsequent layers as $h' = (1-\alpha) * h + \alpha * v$, resulting in a patched belief$'_4$. We use the same v for all 32 tokens, but a separate one per layer. The intervention spans tokens and layers because we found that an edit at a single position and layer caused no behavioral effect, suggesting that the information is distributed (see Appendix E for additional results). We compare the edit against an edit with a random vector of the same norm. If an LM's recommendation is computed from an encoded belief in a Bayes-consistent way, it should change if the recommendation changes under the BAYESASSIST's policy resulting from belief$'_4$, and stay the same otherwise. We therefore

report accuracy averaged equally across recommendation-change and no-change samples (balanced Bayes accuracy), then averaged over the eight feature-direction settings.

At layers 18–25 the BAYESLM reaches a balanced accuracy of 0.77 (visualized in Figure 14 in the Appendix), and the intervention works across test sets (see Appendix E). The random vector barely changes the answer at all—so the injected belief, not the disturbance through patching itself, drives the model's recommendations. The effect is confined to the layers where the belief becomes decodable, peaking around layers 15–22 and falling back to chance in the upper layers. Since the belief_4 remains decodable in those upper layers without being steerable there, this underscores that decodability alone does not imply a causal role. ORACLELM's accuracy of 0.52, on the other hand, remains close to chance, suggesting that it does not *use* its encoded beliefs.

Forming Bayesian beliefs relies on distributed information. Based on the observation that a large-scale intervention is necessary to produce recommendation changes, we speculate that it might be because of the computational route for Bayes' rule in LMs. Concretely, Bayesian inference can equivalently be realized with *sequential* or *one-swoop* accommodation of evidence for calculating the posterior, where the former would require maintaining intermediate posteriors, and the latter gathering information across context. We offer a novel exploration whether the LM exhibits higher consistency with one of these routes. We do so by applying masks that allow attention flow consistent with only one of the routes (details are in Appendix F). Results reveal a substantial drop from original Bayes accuracy of 0.84 to 0.68 under the mask allowing one-swoop, and to 0.56 under the mask allowing sequential flow. This suggests that the BAYESLM indeed tends to follow a distributed, more one-swoop-like implementation, although both masks lead to accuracy drops.

In sum, BAYESLM partially (not reaching ceiling) uses its encoded beliefs for its recommendation (**R3**), while ORACLELM's predictions are less sensitive to interventions on beliefs, so the models differ in the information their recommendations use. However, the interventions do not show which *computations* form the recommendation in the LM, which we address next.

6 FORMING A BAYESIAN POLICY FROM BELIEFS

Without investigating the computations which use the represented beliefs, it remains unclear to what extent the quality of the belief representation (L3) affects the recommendation behavior in the different LMs: divergence between LM policies and BAYESASSIST could be, e.g., because of miscalibrated belief representations, or because correct beliefs are used inconsistently with Bayes-optimal decision making. So next we focus on **R4** and provide exploratory analyses of how the LMs' behavioral policy might be formed with the represented beliefs (L3→L2).

Cross-model patching. In § 4 we showed that BAYESLM's representations are more Bayes-consistent than other LMs', and that only BAYESLM's beliefs are causally involved in the recommendation task (§ 5). If the latter is simply due to more task-relevant, accurate content of BAYESLM's beliefs at L3, we expect that patching BAYESLM's beliefs into another model will increase its balanced accuracy on the tests from § 5 up to the level of BAYESLM, while patching another model's beliefs into BAYESLM will lower it. If additionally there is a difference in how the Bayes-fine-tuned LM *forms* the recommendation from the belief representation (L3→L2), we expect other LMs patched with BAYESLM's beliefs to still underperform relative to BAYESLM. To test this, we inject the belief prototype vector v derived from one model into another, and compare the resulting accuracy against the results from injecting the model's own prototype belief. We use the best hyperparameter setting from § 5.

Figure 6 shows that BAYESLM beliefs raise the ORACLELM's balanced accuracy from 0.52 to 0.61, while ORACLELM beliefs lower the BAYESLM's from 0.77 to 0.61. This confirms that BAYESLM supplies higher-quality, task-relevant beliefs (L3), but their content alone does not explain the gap between the models, because the patched ORACLELM still underperforms. We address this next.

Deriving a neuro-symbolic behavioral policy. Previous results alone could not tell how closely the LMs' use of represented beliefs for *behavioral* predictions matches Bayesian decision making, because it is not feasible to map the LMs' forward pass transforming belief representations into the behavioral policy onto the normative symbolic calculation of the softmax-rational choice policy. To

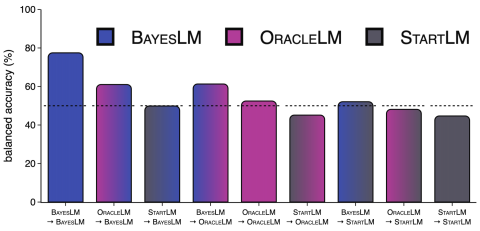

Figure 6: Balanced accuracy when different models' prototype vector is patched into another models, relative to patching each model's own prototype vector.

	neuro-symbolic	Δ over LM	KL(NeSy||LM)
BAYESLM	0.86	0.02	0.106
ORACLELM	0.78	0.09	0.354

Table 1: Bayes accuracy of neuro-symbolic against each model's own predictions, and divergence between neuro-symbolic and LMs' actual policies, representing what each model loses by its own belief read-out.

approach this calculation, we take the LMs' probed belief$_4$ (§ 4), calculate the posterior mean for each feature, and plug them into the symbolic EU calculation of the BAYESASSIST (§ 2), resulting in a *neuro-symbolic* model. Results in Table 1 show that the neuro-symbolic accuracy is nearly identical to BAYESLM's own (+0.02), showing that BAYESLM's forward pass already maps its beliefs close to the normative upper bound permitted by the quality of the encoded beliefs. The neuro-symbolic model based on the ORACLELM, on the other hand, improves over the ORACLELM by 0.09, suggesting that ORACLELM partially encodes Bayes-relevant beliefs without fully reading them out for recommendation. Both neuro-symbolic policies diverge little from the Bayesian policy (0.009 for BAYESLM, 0.017 for ORACLELM) but more from the LMs' own behavioral policies (0.106 and 0.354), suggesting that the behavioral gaps in § 4 stem from noise in policy components other than beliefs (e.g., option-set representations), and more so for ORACLELM.[4] Notably, since the BAYESLM neuro-symbolic performance is also not at ceiling yet, we speculate that remaining behavioral differences between the BAYESLM and BAYESASSIST might be partially due to noisy belief decoding caused by distributed information in the LM.

In sum, we conclude that Bayesian fine-tuning installs not only a more consistent probabilistic internal representation, but also a read-out that at least in part consumes what the model encodes for the downstream task in a way that standard oracle fine-tuning cannot.

7 RELATED WORK

Bayesian inference in connection with LMs is studied in cognitive science (Griffiths et al., 2024; Zhu & Griffiths, 2026) and machine learning (MacKay, 1995; Shi & Griffiths, 2009; Mandt et al., 2017; Papamarkou et al., 2026). This includes analyzing probabilistic reasoning in general (Nafar et al., 2024; Samanta et al., 2026), on tasks with explicitly presented probabilities, showing that LMs may deviate from normative behavior (e.g., Paruchuri et al., 2024; Zhu & Griffiths, 2024; Pournemat et al., 2025), and improving reasoning by external scaffolding (Piriyakulkij et al., 2024; Feng et al., 2025). Additional work examines whether LMs can verbalize or interpret expressions of uncertainty (Belém et al., 2024; Tang et al., 2026), their inferences about a choice policy (Liu et al., 2024), and divergence between human and model reasoning (Kamath et al., 2026). Relevant to our dataset, where the model infers preferences from sequential demonstrations (cf. Lin et al., 2022), in-context learning is often viewed as approximate Bayesian inference (Xie et al., 2021; Zhang et al., 2023; Panwar et al., 2024; Gupta et al., 2025; but see Falck et al., 2024). In contrast to studies that focus on probabilistic reasoning through model behavior alone, or on model internals alone, we build on Qiu et al. (2024; 2026) and assess LMs on a task that requires both Bayesian inference and subsequent optimal decision-making using the inferred latent variable, across the behavioral and the internal perspective. Studying model internals typically involves probing information in LMs' representations (e.g., Tenney et al., 2019; Waldis et al., 2024) or steering their behavior to uncover causal mechanisms (e.g., Turner et al., 2025; Todd et al., 2023). Related to this work, others have shown that representations encode the dynamics of beliefs on other tasks (Bortoletto et al., 2024; Bigelow et al., 2025; Guiomar et al., 2026), while the resulting behavior remains brittle, consistent with findings that probed information is not necessarily used by the model (Gekhman et al., 2025; Orgad et al., 2025; Waldis et al., 2026).

[4] We apply beta scaling to the neuro-symbolic policy as in § 3 to calculate divergence to the LM behavioral policy.

8 DISCUSSION

Results across R1–R4 (Table 2) show that Bayesian fine-tuning makes an LM as Bayesian as its beliefs allow, but not a fully genuine Bayesian decision maker. At **R1** and **R2**, BAYESLM acts Bayesian and represents the quantities of Bayes' rule. At **R3**, it uses its decoded beliefs causally, but not as fully as Bayesian decision making would require. At **R4**, its transformation of beliefs into a choice policy across options by expected utility is as good as those beliefs allow. So the gap to the BAYESASSIST lies in the beliefs, not in their read-out. Across all four requirements, BAYESLM is more consistent with the BAYESASSIST than ORACLELM, a pattern that holds across backbones (Appendix H). These results have three implications: for how we fine-tune LMs, how we evaluate them, and what we can expect from them. We discuss limitations of our study in Appendix A.

The advantage of Bayesian supervision. As the explicit assistant can recover Bayes accuracy from the decoded beliefs of BAYESLM or ORACLELM, fine-tuning makes LMs act on most of what they encode. But the beliefs of BAYESLM are superior, as patching its beliefs into ORACLELM clearly improves its Bayes accuracy. These insights align with superficial alignment theory and the fact that fine-tuning rather impacts how LMs verbalize already encoded information than changing encoding itself (Zhou et al., 2023; Waldis et al., 2026). This has a direct implication for fine-tuning practice. A training signal that reflects a Bayes-consistent uncertainty yields better reasoning under uncertainty than the standard protocol of tuning on an oracle-correct signal.

Table 2: Overview of results. Fine-tuning with Bayesian signal, but not oracle one, leads to stronger compliance with the requirements.

	R1	R2	R3	R4
STARTLM	×	×	×	×
ORACLELM	✓	✓	×	×
BAYESLM	✓	✓	(✓)	(✓)

Behavior is not enough to understand reasoning under uncertainty. A wrong recommendation can come from a wrong belief or from how a right belief is transformed into a recommendation across options. Behavioral evaluation cannot distinguish between these, so it cannot tell whether a training change improved what a model encodes or how it uses the information, a fundamental gap between encoding and expression, also reflected in other work (Gekhman et al., 2025). This requires an analytical lens on decision-making, and we provide a first one with four concrete requirements and their operationalizations for LMs.

A Bayesian fine-tuned LM is still a language model. While the BAYESASSIST performs exact inference on each instance, LMs learn one forward pass that should serve all possible user tasks, and so may only approximate the posterior. They are trained to predict tokens rather than maintain uncertainty which explains their failures on high-entropy policies. In addition, LMs compute beliefs across layers and tokens rather than at a single position, reading them out only in part. However, in principle, since the target is learnable, more or better Bayesian fine-tuning may close the remaining gap. How beliefs are formed and turned into a choice in the forward pass, however, remains open.

9 CONCLUSION

We asked to what extent an LM that behaves like a Bayesian decision maker also is one internally. To answer this, we formulate four increasingly demanding requirements for Bayesian decision making in LMs, spanning behavior, representations, and computation, and applied them to the models and flight recommendation task of Qiu et al. (2026). We show that **Bayesian fine-tuning makes an LM as Bayesian as its beliefs allow, but not a genuine Bayesian decision maker**. Our requirements allow not only to detect whether an LM is Bayesian but also to locate where it stops being so, e.g., under oracle SFT, and they provide first insights into how an LM computes with beliefs. It uses them causally, distributed across layers and tokens, and turns them into a choice policy as well as they allow. While we lay groundwork for studying Bayesian decision making in LMs, future work should extend it in two directions. First, to other tasks with a normative Bayesian solution, to test whether the pattern we find is specific to preference inference. Second, to the implementation of belief updating and of the transformation of beliefs into a choice policy across options (cf. Piotrowski et al., 2025), given that the BAYESLM's beliefs are distributed rather than held at a single position.

AI USE STATEMENT

In this work, we used generative AI tools for coding the experiments, implementing analyses and visualizations, and for feedback on the interpretation of the results. We have *not* used generative AI tools for developing the conceptual framework, data generation or designing the experiments, and the use of AI tools for mathematical proofs or for translation are not applicable to this work. Additionally, we used generative AI tools for drafting parts of the text. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text or artifacts produced with the aid of generative AI.

REFERENCES

Catarina G Belém, Markelle Kelly, Mark Steyvers, Sameer Singh, and Padhraic Smyth. Perceptions of linguistic uncertainty by language models and humans. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 8467–8502, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.483. URL https://aclanthology.org/2024.emnlp-main.483/.

Eric Bigelow, Daniel Wurgaft, YingQiao Wang, Noah Goodman, Tomer Ullman, Hidenori Tanaka, and Ekdeep Singh Lubana. Belief dynamics reveal the dual nature of in-context learning and activation steering. *arXiv preprint arXiv:2511.00617*, 2025.

Matteo Bortoletto, Constantin Ruhdorfer, Lei Shi, and Andreas Bulling. Brittle minds, fixable activations: Understanding belief representations in language models. *arXiv preprint arXiv:2406.17513*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Fabian Falck, Ziyu Wang, and Chris Holmes. Is in-context learning in large language models bayesian? a martingale perspective. *arXiv preprint arXiv:2406.00793*, 2024.

Yu Feng, Ben Zhou, Weidong Lin, and Dan Roth. Bird: A trustworthy bayesian inference framework for large language models. In *International Conference on Learning Representations*, volume 2025, pp. 8961–8989, 2025.

Zorik Gekhman, Eyal Ben David, Hadas Orgad, Eran Ofek, Yonatan Belinkov, Idan Szpektor, Jonathan Herzig, and Roi Reichart. Inside-out: Hidden factual knowledge in llms. *arXiv preprint arXiv:2503.15299*, 2025.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: a unifying framework for inspecting hidden representations of language models. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Thomas L Griffiths, Jian-Qiao Zhu, Erin Grant, and R Thomas McCoy. Bayes in the age of intelligent machines. *Current Directions in Psychological Science*, 33(5):283–291, 2024.

Gonçalo Guiomar, Elia Torre, Pehuen Moure, Victoria Shavina, Mario Giulianelli, Shih-Chii Liu, and Valerio Mante. Reasoning aligns language models to human inference. In *Mechanistic Interpretability Workshop at ICML 2026*, 2026.

Ritwik Gupta, Rodolfo Corona, Jiaxin Ge, Eric Wang, Dan Klein, Trevor Darrell, and David M Chan. Enough coin flips can make llms act bayesian. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7634–7655, 2025.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 2733–2743, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1275. URL https://aclanthology.org/D19-1275/.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1979. URL https://api.semanticscholar.org/CorpusID:122415379.

Gaurav Kamath, Sreenath Madathil, Sebastian Schuster, Marie-Catherine de Marneffe, and Siva Reddy. Humans and llms diverge on probabilistic inferences. *Transactions of the Association for Computational Linguistics*, 14:2057–2080, 2026.

Jessy Lin, Daniel Fried, Dan Klein, and Anca Dragan. Inferring rewards from language in context. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8546–8560, 2022.

Ryan Liu, Jiayi Geng, Joshua C Peterson, Ilia Sucholutsky, and Thomas L Griffiths. Large language models assume people are more rational than we really are. *arXiv preprint arXiv:2406.17055*, 2024.

David JC MacKay. Probable networks and plausible predictions-a review of practical bayesian methods for supervised neural networks. *Network: computation in neural systems*, 6(3):469, 1995.

Stephan Mandt, Matthew D Hoffman, and David M Blei. Stochastic gradient descent as approximate bayesian inference. *Journal of Machine Learning Research*, 18(134):1–35, 2017.

Melanie Mitchell. Jagged intelligence: The dangerous unknowns at the heart of llms. *The Yale Review*, 114(2):80–92, 2026. doi: 10.1353/tyr.2026.a991386. URL http://dx.doi.org/10.1353/tyr.2026.a991386.

Aliakbar Nafar, Kristen Brent Venable, and Parisa Kordjamshidi. Probabilistic reasoning in generative large language models. *arXiv preprint arXiv:2402.09614*, 2024.

Hadas Orgad, Michael Toker, Zorik Gekhman, Roi Reichart, Idan Szpektor, Hadas Kotek, and Yonatan Belinkov. LLMs know more than they show: On the intrinsic representation of LLM hallucinations. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=KRnsX5Em3W.

Madhur Panwar, Kabir Ahuja, and Navin Goyal. In-context learning through the bayesian prism. In *International Conference on Learning Representations*, volume 2024, pp. 49789–49843, 2024.

Theodore Papamarkou, Pierre Alquier, Matthias Bauer, Wray Buntine, Andrew J Davison, Gintare Karolina Dziugaite, Maurizio Filippone, Andrew YK Foong, Vincent Fortuin, Dimitris Fouskakis, et al. Position: agentic ai orchestration should be bayes-consistent. In *Forty-third International Conference on Machine Learning Position Paper Track*, 2026.

Akshay Paruchuri, Jake Garrison, Shun Liao, John B Hernandez, Jacob Sunshine, Tim Althoff, Xin Liu, and Daniel McDuff. What are the odds? language models are capable of probabilistic reasoning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 11712–11733, 2024.

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.

Wasu T Piriyakulkij, Cassidy Langenfeld, Tuan A Le, and Kevin Ellis. Doing experiments and revising rules with natural language and probabilistic reasoning. *Advances in Neural Information Processing Systems*, 37:53102–53137, 2024.

Mobina Pournemat, Keivan Rezaei, Gaurang Sriramanan, Arman Zarei, Jiaxiang Fu, Yang Wang, Hamid Eghbalzadeh, and Soheil Feizi. Reasoning under uncertainty: Exploring probabilistic reasoning capabilities of llms. *arXiv preprint arXiv:2509.10739*, 2025.

Linlu Qiu, Fei Sha, Kelsey R Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. Can language models perform implicit bayesian inference over user preference states? In *The First Workshop on System-2 Reasoning at Scale, NeurIPS'24*, 2024.

Linlu Qiu, Fei Sha, Kelsey Allen, Yoon Kim, Tal Linzen, and Sjoerd van Steenkiste. Bayesian teaching enables probabilistic reasoning in large language models. *Nature Communications*, 2026.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Abhilasha Ravichander, Yonatan Belinkov, and Eduard Hovy. Probing the probing paradigm: Does probing accuracy entail task relevance? In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty (eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 3363–3377, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.295. URL https://aclanthology.org/2021.eacl-main.295/.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Ankur Samanta, Akshayaa Magesh, Tal Lancewicki, Ayush Jain, Youliang Yu, Paul Sajda, Kaveh Hassani, Aditya Modi, Daniel R Jiang, and Yonathan Efroni. Bayesbench: Evaluating llm belief trajectories under multi-turn evidence accumulation. *arXiv preprint arXiv:2606.30850*, 2026.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Lei Shi and Thomas Griffiths. Neural implementation of hierarchical bayesian inference by importance sampling. *Advances in neural information processing systems*, 22, 2009.

Zhisheng Tang, Ke Shen, and Mayank Kejriwal. An evaluation of estimative uncertainty in large language models. *npj Complexity*, 3(1):8, 2026.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4593–4601, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1452. URL https://aclanthology.org/P19-1452/.

Eric Todd, Millicent L Li, Arnab Sen Sharma, Aaron Mueller, Byron C Wallace, and David Bau. Function vectors in large language models. *arXiv preprint arXiv:2310.15213*, 2023.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering, 2025. URL https://openreview.net/forum?id=2XBPdPIcFK.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. *Advances in neural information processing systems*, 33:12388–12401, 2020.

Andreas Waldis, Yotam Perlitz, Leshem Choshen, Yufang Hou, and Iryna Gurevych. Holmes Σ a benchmark to assess the linguistic competence of language models. *Transactions of the Association for Computational Linguistics*, 12:1616–1647, 2024. doi: 10.1162/tacl_a_00718. URL https://aclanthology.org/2024.tacl-1.88/.

Andreas Waldis, Leshem Choshen, Yufang Hou, and Yotam Perlitz. Instructions shape production of language, not processing. *arXiv preprint arXiv:2605.11206*, 2026.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Yufeng Zhang, Fengzhuo Zhang, Zhuoran Yang, and Zhaoran Wang. What and how does in-context learning learn? bayesian model averaging, parameterization, and generalization. *arXiv preprint arXiv:2305.19420*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, LILI YU, Susan Zhang, Gargi Ghosh, Mike Lewis, Luke Zettlemoyer, and Omer Levy. LIMA: Less is more for alignment. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=KBMOKmX2he.

Jian-Qiao Zhu and Thomas L Griffiths. Incoherent probability judgments in large language models. *arXiv preprint arXiv:2401.16646*, 2024.

Jian-Qiao Zhu and Thomas L Griffiths. Computation-limited bayesian updating: A resource-rational analysis of approximate bayesian inference. *Psychological Review*, 133(3):619, 2026.

A LIMITATIONS

Task scope. We compare models fine-tuned on a single task. While it is a realistic assistant recommendation task, it is highly structured and depends on certain assumptions, e.g., the independence of reward features. The ground truth BAYESASSIST generates the training data through softmax-rational decision making based on Bayesian inference over a latent user preference, and the analyzed models were trained on sequential, essentially in-context learning data. Whether our findings hold for less structured tasks, or for tasks without an analytically solvable Bayesian model, is open.

Backbones. The main reported results focus on Gemma-2-9B-Instruct (Team et al., 2024). Additional results for Qwen-2.5-7B and Llama-3.1-8B in Appendix H show qualitatively similar patterns, but we did not examine how architectural factors influence the success of Bayesian teaching, such as the lower Bayesian consistency of Qwen. Future work should additionally explore the interaction of Bayesian teaching and scaling, applying it to larger models.

Linear decodability. We use linear probes to test whether the models represent the quantities of Bayes' rule, which provides only a lower bound on the information contained in the LMs. Our control tasks (Hewitt & Liang, 2019) show that the probes are selective, and the STARTLM serves as a further control: the same probes recover far less from it, so what we decode from the fine-tuned models is installed by fine-tuning rather than an artifact of probing. Still, a non-linear read-out could recover more, and the belief divergence we report may partly reflect the limits of linear decoding rather than of the representation.

Scope of the interventions. Our causal interventions replace the prior only, at a fixed window of layers and tokens chosen for the BAYESLM. They show that the beliefs are used, but not how they are combined, and the neuro-symbolic and cross-model results test the transformation into a choice policy from outside the model rather than by intervening on it. We therefore lack an in-depth understanding of how an LM computes expected utility when transforming its beliefs into a choice policy across options. Our results provide a first step in this direction.

Table 3: Logistic regression analysis of Bayes accuracy in § 3, regressing Bayes correctness against the LM, the test set and their interaction: `correct ~ C(model) * C(set)`. The predictors are dummy coded. OR refers to odds ratio.

term	β	95% CI of the OR	OR	p
BayesLM	0.8816	$[1.8942, 3.0781]$	2.4147	$< 10^{-14}$
StartLM	-1.4257	$[0.1881, 0.3070]$	0.2403	$< 10^{-14}$
Shuffled	-0.0299	$[0.8231, 1.1444]$	0.9705	1.00
Low-Noise	-0.2102	$[0.6620, 0.9923]$	0.8105	0.209
High-Noise	-0.4215	$[0.5239, 0.8216]$	0.6561	0.002
BayesLM:Shuffled	-0.0062	$[0.7899, 1.2503]$	0.9938	1.00
StartLM:Shuffled	0.1133	$[0.9140, 1.3724]$	1.1200	1.00
BayesLM:Low-Noise	0.3784	$[1.0370, 2.0555]$	1.4600	0.1809
StartLM:Low-Noise	0.3141	$[1.0570, 1.7734]$	1.3691	0.1213
BayesLM:High-Noise	-0.0306	$[0.6877, 1.3679]$	0.9699	1.00
StartLM:High-Noise	0.5527	$[1.2711, 2.3763]$	1.7380	0.0043

Table 4: Logistic regression analysis of policy divergence in § 3, regressing Bayes correctness against the LM, the entropy of the Bayesian policy and their interaction: `correct ~ C(model) * H(Bayesian policy)`. The model is dummy coded. OR refers to odds ratio.

term	β	95% CI of the β	OR per SD	p
BayesLM	1.1634	$[0.8004, 1.5264]$	3.2007	$< 10^{-14}$
StartLM	-1.4757	$[-1.7296, -1.2219]$	0.2286	$< 10^{-14}$
H(policy)	-0.4949	$[-0.7101, -0.2798]$	0.6096	$< 10^{-14}$
H(policy):BayesLM	-0.6238	$[-1.1192, -0.1284]$	0.5359	0.0136
H(policy):StartLM	0.4290	$[0.1339, 0.7241]$	1.5357	0.0088

B Statistical analyses

Below, we report details of the regression analyses performed in the main results. For all regression models, we use z-scored normalized values for continuous predictors, and apply the Bonferroni-Holm p-value correction (Holm, 1979). The logistic regression model for Bayes accuracy in § 3 is reported in Table 3, the logistic regression for analyzing policy divergence is in Table 4. The linear regression for analyzing Bayes-consistency of the decoded policy in § 4 is in Table 5. A model with both predictors (both prior and decoded policy divergences) produced the same results.

C Behavioral Evaluations

We provide further details of the behavioral evaluations (R1: acting Bayesian) described in § 3 below.

BayesAssist's performance. The normative solution to the recommendation task is implemented by the BayesAssist, which makes recommendations based on beliefs that can be inferred from the available data of the preceding turns, which may not be fully disambiguating. Therefore, given the task data, the BayesAssist's average Oracle accuracy (i.e., match with user's true preference) on the fifth round is 0.811 (*original* set), 0.811 (*shuffled* set), 0.683 (*low noise* set), 0.420 (*high noise* set).

Dataset details. We provide an example conversation round from the *original* set in prompt format, as it was passed to the LMs, in Figure 7. Following Qiu et al. (2026), the dataset features departure time, duration, and price each had ten distinct equally spaced values, and were mapped from the verbal representations onto equally spaced numbers between 0 and 1.0 for calculations within the BayesAssist. The stops feature only had three values (verbalized: 0, 1, 2; mapped onto 0, 0.5, 1, respectively). This was used across datasets.

Table 5: Linear regression analysis of Bayes-consistency of decoded beliefs in § 4, regressing the LM's token-based behavioral policy against the LM, one of the two predictors (probed policy divergence or prior belief divergence) and their interaction. The model is dummy coded.

term	β	95% CI of the β	p
`div(behavioral policy) ~ C(model) * div(prior)` ($R^2 = 0.02$, AIC=473.7)			
Intercept	0.5037	$[0.4773, 0.5300]$	—
BayesLM	-0.0829	$[-0.1117, -0.0541]$	$< 10^{-14}$
div(prior)	0.0025	$[-0.0250, 0.0300]$	1.00
div(prior):BayesLM	0.0107	$[-0.0237, 0.0450]$	1.00
`div(behavioral policy) ~ C(model) * div(probed policy)` ($R^2 = 0.46$, AIC=-276.4)			
Intercept	0.5144	$[0.4923, 0.5364]$	—
BayesLM	-0.1064	$[-0.1283, -0.0846]$	$< 10^{-14}$
div(policy)	0.1927	$[0.1645, 0.2210]$	$< 10^{-14}$
div(policy):BayesLM	0.0083	$[-0.0217, 0.0383]$	0.5880

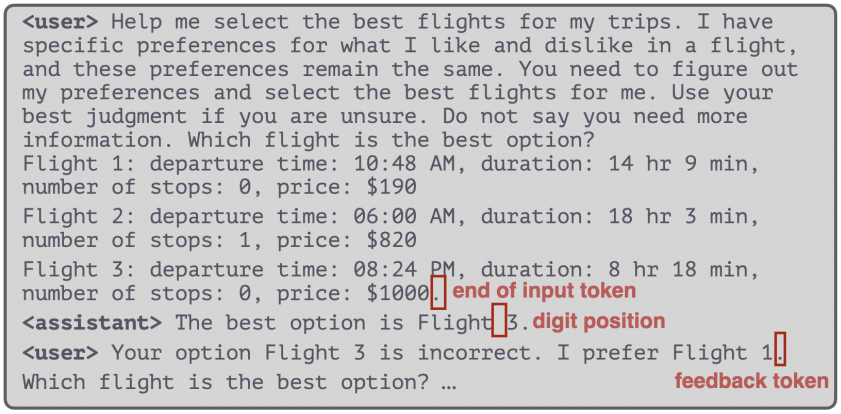

Figure 7: Prompt presenting the first round of the recommendation task, marking the probed token positions (reported probing was performed on round 5).

Policy divergence analyses. We evaluate LMs' behavioral choice policies, derived from their conditional token probabilities, by calculating the KL divergence relative to the Bayesian (BAYESASSIST's) policy (i.e., policy divergence, as introduced in § 2). The Bayesian policy can be scaled by the rationality parameter β (see § 2). Under $\beta = 1$, both LMs place more probability mass on the most probable option (i.e., emitted token, ★ in Figure 8) than the BAYESASSIST does (○ in Figure 8): while, on average, the BAYESLM's chosen option has the probability 0.74 and, ORACLELM's 0.72, the BAYESASSIST's chosen option has the average probability 0.46. This indicates that LMs' policies are generally more decisive, suggesting that they may be less sensitive to nuanced uncertainty and may be more prone to errors. For main comparisons we select a β minimizing the policy divergence across LMs by considering integer β values between 0 and 10 (the search is visualized separately for each LM in Figure 8). The policy divergence is lowest for both LMs under $\beta = 4$: on samples where the models are Bayes-correct, the KL is 0.10 for BAYESLM, and 0.26 for ORACLELM. On samples with incorrect predictions, $\beta = 4$ leads to a drop only for BAYESLM (KL of 0.42 to 0.38) and to an increase for ORACLELM (0.66 to 0.90).

Additionally, we show the correlations of the entropies of the resulting Bayesian policy with the LM policies in Figure 9.

D PROBING EXPERIMENTS

Below, more details of probing experiments from § 4 testing LMs' representations (R2: representing beliefs) are reported.

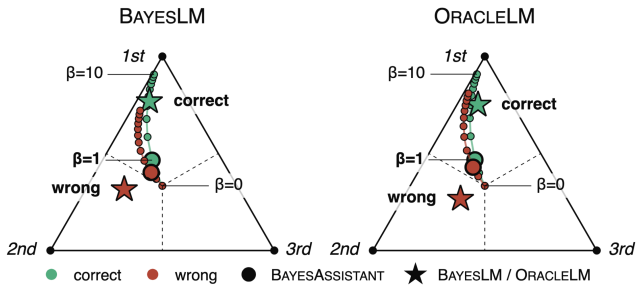

Figure 8: Visualized search over the softmax rationality parameter β for scaling the BAYESASSIST's policy: $\beta = 0$ is a uniform distribution, $\beta = 1$ corresponds to the original BAYESASSIST, $\beta = 10$ is near-deterministic, and $\beta \approx 4$ is the best match to the LMs' choice policies.

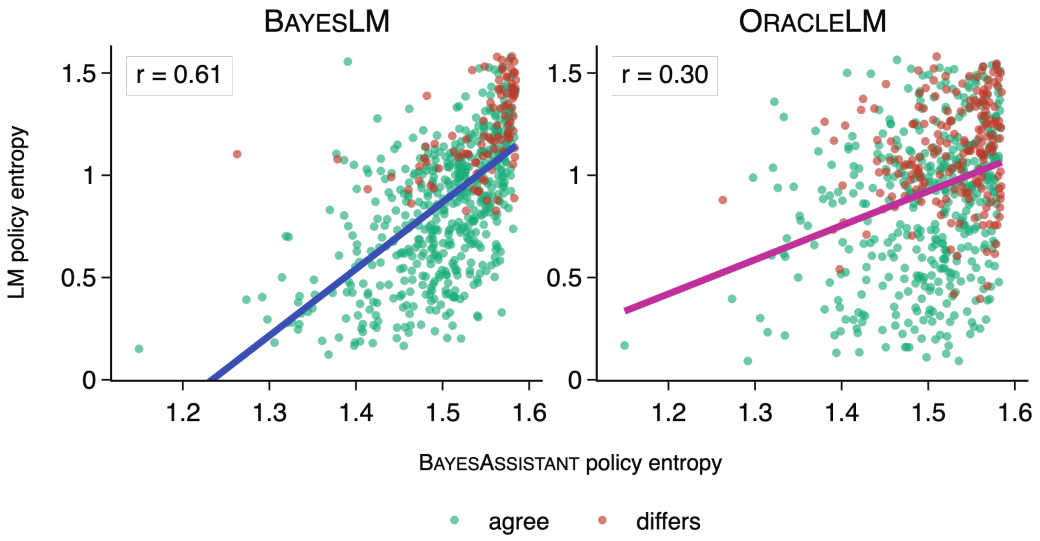

Figure 9: **The BAYESLM's uncertainty tracks the assistant's.** Entropy of the LM's behavioral choice policy against the entropy of the Bayesian policy, one point per sample of the *original* set, colored by whether the recommendation agrees with the BAYESASSIST. The BAYESLM's entropy correlates with the assistant's ($r = 0.61$), the ORACLELM's only weakly ($r = 0.30$). For both models, the differing instances are concentrated where the BAYESASSIST itself is most uncertain, but spread more for the ORACLELM.

Additional results on Bayes-consistency of decoded policies. In the main analyses, we identified that the probed policy divergence was predictive of the LMs' behavioral policy divergence. This is confirmed by visual inspection (see Figure 10), which shows that the decoded policy of BAYESLM comes close to the Bayesian policy, while the decoded policy of ORACLELM stays further away from it, in particular, on incorrect samples.

Probing on additional test sets. The main results in § 4 are only for specific token positions. Figure 12 shows probing results on the original set broken down by different probing position. The choice representations (policy divergence, top-1 choice) develop in the mid-to-late layers at the last-prompt and digit tokens — where the BAYESLM and ORACLELM models rise well above STARTLM — but stays near chance at the feedback token. This provides a sanity check that the choice is not yet committed before the new option set is seen. Belief representations (prior/posterior divergence) are decodable across all positions and depths.

Additionally, we expect Bayes-consistent decoded beliefs to be order-invariant and for respond adequately to noisy input. Figure 11 shows probing results broken down by test set. We indeed find that reordering the evidence leaves the belief divergence, averaged over layers, unchanged for

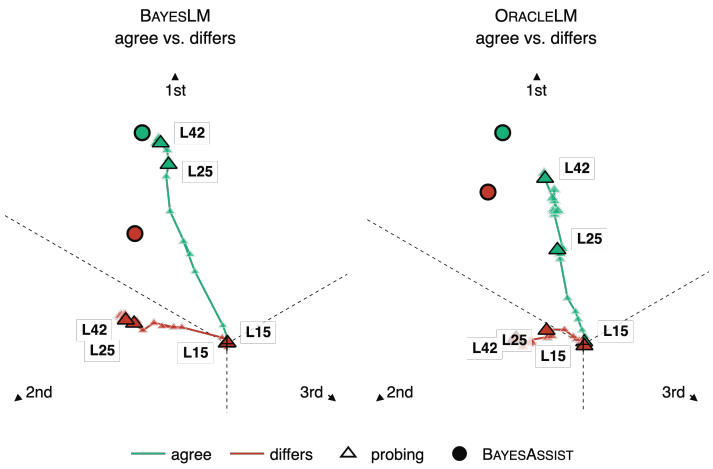

Figure 10: Mean decoded choice policy by layer on the simplex (corners ordered by Bayesian policy), split by whether the LM's recommendation agrees with the BAYESASSIST, i.e., is Bayes-correct, (green) or differs (red). Triangles mark the mean decoded choice policy for each layer, and circles mark the mean Bayesian policy of the BAYESASSIST for the same instances.

BAYESLM and ORACLELM: a regression of belief divergence against the main effect of test set shows no significant effects ($p = 0.49$ and $p = 0.43$). Under low- or high-noise, the decoded beliefs of both BAYESLM and ORACLELM have lower entropy than BAYESASSIST's, resulting in a high divergence (for BAYESLM: 0.52 on low-noise, and 0.58 on high-noise.) However, with exponential scaling of the probed beliefs with $\beta \approx 0.2$, BAYESLM's belief divergence is 0.083 (on low-noise) and 0.059 (on high-noise). In parallel with § 3, BAYESLM and ORACLELM mainly differ in how strongly they encode beliefs, but both are invariant to evidence order and noise and thereby Bayes-consistent. We speculate that the low default entropy of the probed beliefs might be an artifact of training probes on noise-free data. The ORACLELM behaves the same way.

§ 4 averages the prior probe over the four flight features. Figure 13 shows each feature separately. The pattern is the same for all four. Up to layer 15, no model does better than a uniform probe. Between layers 17 and 25, the probe loss of the BAYESLM and the ORACLELM drops well below the uniform probe and then stays flat until the last layer. The BAYESLM ends lowest on every feature (KL ≈ 0.25), and the ORACLELM ends at about 0.40. The STARTLM stays close to the uniform probe throughout, and in the belief window it even goes above it. The features differ only in how high the uniform probe sits: the targets for the number of stops are sharpest (0.77, against 0.62–0.63 for the other three). The fine-tuned models still reach the same final loss there, so they gain the most on this feature. So the averaged result in the main text is not driven by any single feature.

E ADDITIONAL RESULTS ON BELIEF INTERVENTION

This section provides details on § 5. We describe the setup, report the hyperparameter search, compare prototype patching with additive steering, and list the results per feature and direction.

Setup. For each feature, direction, and layer, we build the prototype v as the mean hidden state of the 500 training instances with the highest prior on that feature (positive direction) or the lowest (negative direction). We patch v into the residual stream h at the *feedback* token and the next 32 tokens across 8 subsequent layers as $h' = (1 - \alpha) * h + \alpha * v$, using the same v for all tokens and a separate one per layer. We evaluate on all 624 held-out instances. Between 31% and 38% of them are instances where the BAYESASSIST's recommendation changes under the patched belief, depending on the feature and direction.[5] We treat this BAYESASSIST's behavioral as the expected ground truth

[5]We operationalize the edit in the BAYESASSIST by setting the belief$_4$ of the edited feature to 1 or -1, respectively.

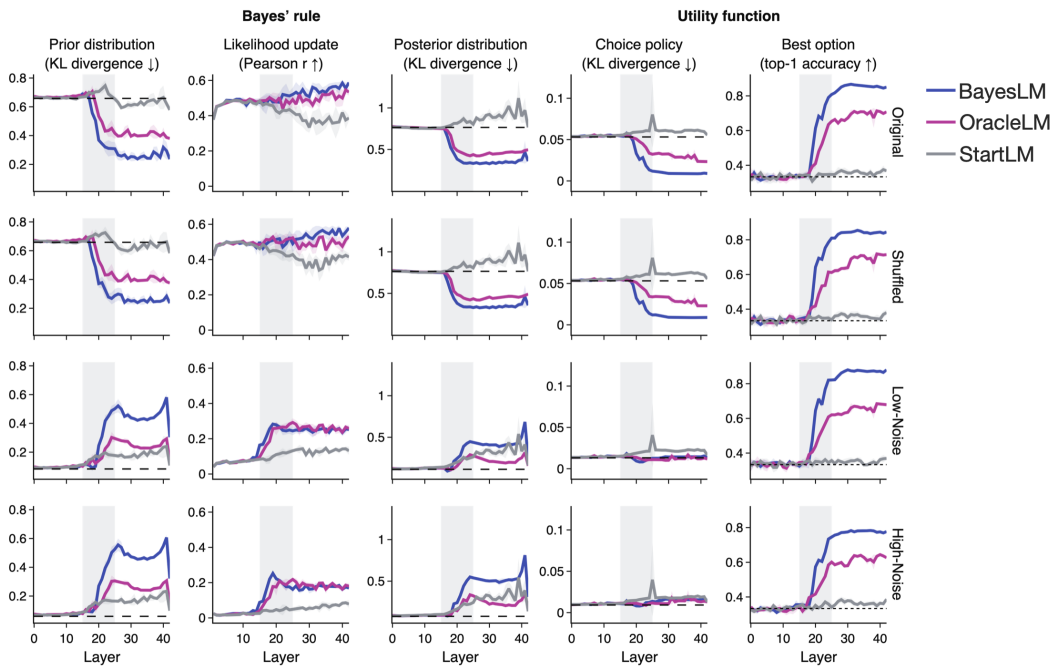

Figure 11: **Per-layer probing across the four evaluation sets (detail for Figure 4).** Each row is an evaluation set (original, shuffled, low-noise, high-noise); columns first present probed quantities of Bayes' rule, and then the probed policy and resulting prediction (argmax of probed policy). On the original and shuffled sets, the belief and choice KL curves fall in the mid-stack and the BAYESLM outperforms ORACLELM and STARTLM in the late layers. On the noisy sets the prior/posterior KL instead *rises* in the upper layers: the probes are trained on the original set and learn to represent sharper beliefs in late layers, which diverges from the flatter beliefs induced by the noisy sets. Therefore, the KL grows even as choice top-1 stays high — likely a calibration gap of the probe, not a representational failure of the model.

behavior under the intervention. We measure the accuracy of the edited LMs' via *flip accuracy*, which is the proportion of samples where the BAYESASSIST changes its recommendation upon intervention where the LM changes its recommendation upon intervention to the same option as the BAYESASSIST; and via *stay accuracy*, which is the proportion of samples where the BAYESASSIST does not change its recommendation upon recommendation for which the LM does not change the recommendation, too. We report *balanced accuracy*, which is their mean. A baseline that never changes its recommendations scores 0.5.

Hyperparameter search. We tune the intervention parameters on the duration feature in the positive direction, using a random draw of 100 instances where the recommendation should change and 100 where it should remain the same. The parameters and results are shown in Table 6. We vary the dose α which weighs the injected vector v, the number of patched layers (window width), and the number of patched tokens after the *feedback* token. As shown in Table 6, patching a single layer or only 4 tokens changes few recommendations, so the belief has to be replaced across tokens and layers. Larger doses raise flip accuracy but lower stay accuracy. Therefore, we select 32 tokens, $\alpha = 0.5$ and 8 layers. The main results with these setting are visualized in Figure 14.

Comparison with CAA steering. We also test contrastive activation addition steering (Rimsky et al., 2024), which constructs a direction d as the mean over differences between LMs' activations for instances with high belief values and instances with low belief values for a particular feature. We use the same data for computing d as in § 5. d is then added as $h' = h + \alpha * d$. We use $\alpha = 0.25$ and the same 8-layer windows, and inject either at the 16 tokens after the *feedback* token or at the digit token where the recommendation is generated. Additive steering after the feedback token reaches at most 0.59 balanced accuracy, against 0.77 for prototype patching (Figure 15). Steering at the digit token never exceeds chance, and in the early layers it corrupts the model away from the BAYESAS-

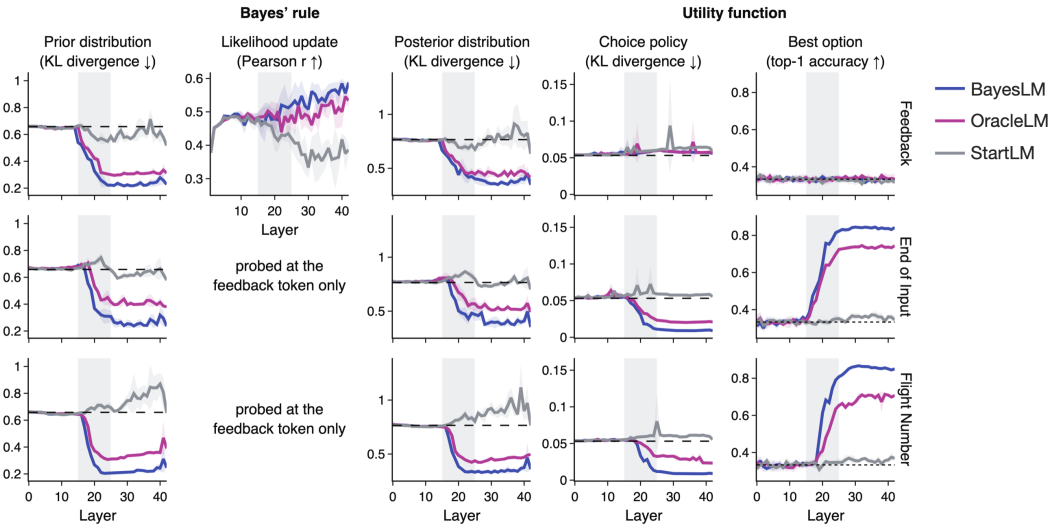

Figure 12: **Per-layer probing across token positions (detail for Figure 4).** Each row is a read-out position (feedback token, last prompt token, digit-generation token); columns first present probed quantities of Bayes' rule, and then the probed policy and resulting prediction (argmax of probed policy). The choice representation (flight-choice KL, top-1) develops in the mid-to-late layers at the last-prompt and digit tokens — where the BAYESLM and ORACLELM models rise well above STARTLM — but stays near chance at the feedback token, showing the choice is not yet committed there. Belief targets (prior/posterior KL) are decodable across all positions and depths.

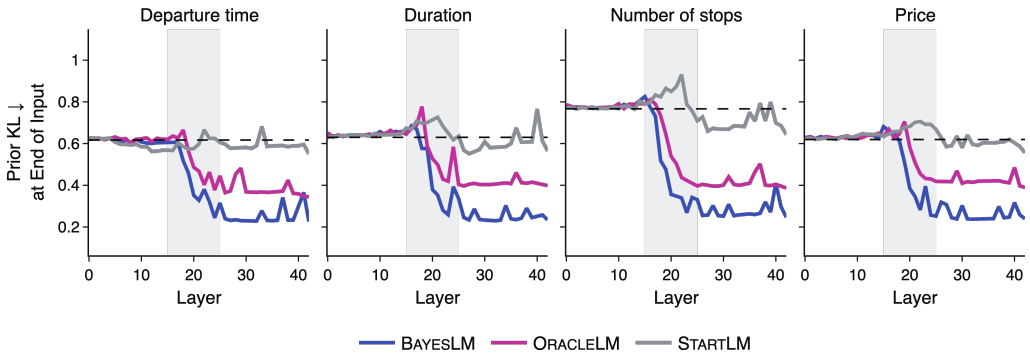

Figure 13: **Prior probing per feature.** KL of the prior probe per layer at the last-input token, on the original set. Dashed: a uniform probe for that feature; grey band: layers 15–25.

SIST's recommendation. Both methods only work in the layers where the belief is decodable, but only prototype patching replaces the belief strongly enough to change most recommendations.

Results per feature and direction. Every result in § 5 averages the eight feature-direction settings, which Table 8 lists individually. The BAYESLM achieves a balanced accuracy between 0.75 and 0.87 across all settings and test sets, so no single feature drives the average. Patching the ORACLELM or patching the BAYESLM's prior into another model stays well below those values in every setting.

Interventions on various test sets. Table 8 shows intervention results for each test set. For BAYESLM, the intervention works across test sets: its balanced accuracy on the shuffled set is 0.778, and 0.795 and 0.805 under low and high noise. The noisy sets score slightly higher because BAYESASSIST changes its recommendation more often there, so more instances fall in the group where we expect a prediction change.

setting	flip accuracy	stay accuracy	balanced accuracy	noise effect
Varied dose α, at 32 tokens and 8 layers				
$\alpha = 0.25$	0.439	0.953	0.696	0.037
$\alpha = 0.50$	0.817	0.847	0.832	0.037
$\alpha = 0.75$	0.939	0.800	0.870	0.037
$\alpha = 1.00$	0.963	0.765	0.864	0.037
Varied dose α, at 4 tokens and 8 layers				
$\alpha = 0.25$	0.098	0.988	0.543	0.012
$\alpha = 0.50$	0.244	0.965	0.604	0.012
$\alpha = 0.75$	0.280	0.976	0.628	0.012
$\alpha = 1.00$	0.280	0.976	0.628	0.012
Varied window width, at 32 tokens and $\alpha = 1$				
1 layer	0.598	0.894	0.746	0.000
2 layers	0.683	0.871	0.777	0.012
4 layers	0.817	0.847	0.832	0.012
8 layers	0.963	0.765	0.864	0.037
Varied window width, at 16 tokens and $\alpha = 1$				
1 layer	0.439	0.906	0.672	0.000
2 layers	0.463	0.918	0.691	0.012
4 layers	0.683	0.918	0.800	0.012
8 layers	0.780	0.871	0.826	0.012
Varied window width, at 4 tokens and $\alpha = 1$				
1 layer	0.159	0.941	0.550	0.000
2 layers	0.195	0.941	0.568	0.012
4 layers	0.317	0.953	0.635	0.012
8 layers	0.280	0.976	0.628	0.012

Table 6: **Hyperparameter search for prototype patching.** We select the best hyperparameter for patching using an iterative search, varying one parameter at a time while keeping the rest fixed. We used a separate set of 100 instances where the recommendation should change and 100 where it should remain the same, generated by the BAYESASSIST. We report for every setting, accuracy on instances that should flip and those that should stay, as well as the effect of injecting a norm-matched random vector (noise effect), We use 32 tokens, $\alpha = 0.5$ and 8 layers. Based on these results, we choose to patch with $\alpha = 0.5$ on 32 tokens across 8 layers, which offers the best trade-off.

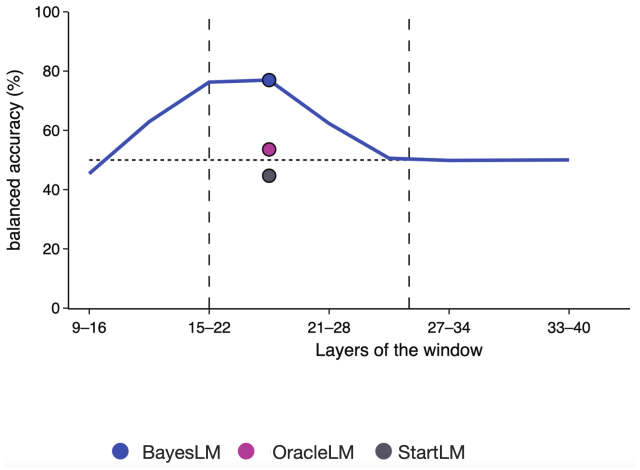

Figure 14: **Intervening on the encoded belief changes the recommendation in BAYESLM.** Balanced accuracy results of prototype belief patching.

Analyzing intervention failures in BAYESLM. We explore whether the success of the intervention depends on the magnitude of the required belief update. For this analysis, we focus on the instances where the BAYESASSIST changes its recommendation, and report the share on which the edited LM's prediction matches it. We group these instances into quartiles by the KL divergence be-

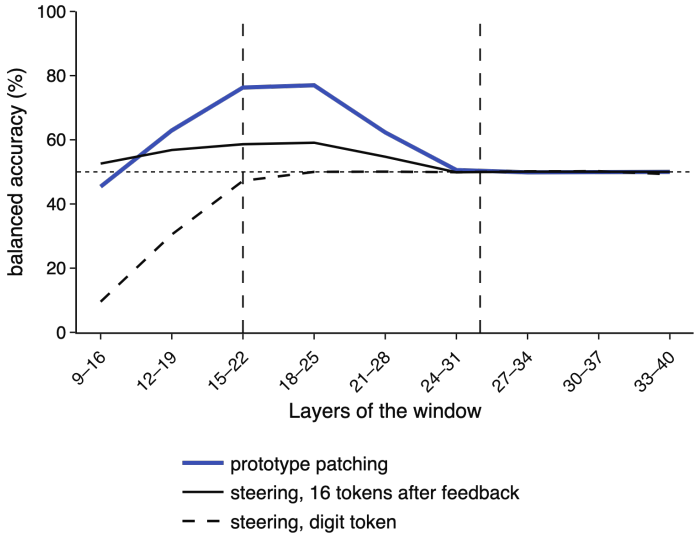

Figure 15: **Prototype patching versus CAA steering.** Balanced accuracy of the BAYESLM by layer window on the original test set, averaged over the eight feature-direction settings. Prototype patching uses 32 tokens and $\alpha = 0.5$; additive steering uses $\alpha = 0.25$ at the 16 tokens after the *feedback* token or at the digit token. The dotted line marks a model that never changes recommendations; the dashed lines mark the layers where § 4 decodes the belief.

		neuro-symbolic	Δ over LM
Llama-3.1-8B	BAYESLM	0.86	0.03
	ORACLELM	0.82	0.10
Qwen-2.5-7B	BAYESLM	0.80	0.07
	ORACLELM	0.74	0.06

Table 7: **Neuro-symbolic predictions.** We read each model's prior beliefs from its hidden states with linear probes, turn the four decoded distributions into per-attribute rewards, and pick the flight the BAYESASSIST's own rule would pick from them. *neuro-symbolic* is how often that flight matches the BAYESASSIST's recommendation, at the layer where each model scores best. Δ *over LM* is the gain over the same model answering on its own.

tween the original belief$_4$ and the patched belief$'_4$ in the BAYESASSIST. The larger the divergence, the more often the LM moves: The LM changed its recommendation on 0.664 of the instances in the smallest quartile (mean KL: 0.019), and on 0.880 of the instances in the largest (mean KL: 0.245). When the LM's recommendations change upon the intervention, they predict the correct option with equal frequency in each quartile (accuracy 0.83–0.85). So when the intervention fails, it does so because the model did not change its recommendation in the first place, not because it changed its prediction to the wrong option (see Figure 16).

F MASKING METHODS

In § 5, we explore the computational route of Bayes' rule in the LMs. To this end, we compare two mathematically equivalent calculations of the posterior given data D_n: the *sequential* an the *one-swoop* belief updating. The *sequential* update is presented in Equation 1, and accesses an intermediate beliefs after every step. The one-swoop computation is shown in Equation 2 uses the whole D_n, without requiring intermediate belief representations:

$$P(\theta \mid D_n) \propto P_H(D_n \mid \theta) \times P(\theta) = \prod_{i=1}^{n} P_H(c_i \mid o_i, \theta) \times P(\theta) \tag{2}$$

		BAYESLM							
feature	direction	original	shuffled	low noise	high noise	ORACLELM	BAYESLM→ORACLELM	BAYESLM→STARTLM	steering
costs	negative	0.775	0.799	0.817	0.818	0.566	0.582	–	0.549
costs	positive	0.769	0.765	0.766	0.766	0.530	0.612	–	0.589
departure time	negative	0.760	0.763	0.751	0.782	0.558	0.662	–	0.565
departure time	positive	0.756	0.773	0.787	0.773	0.533	0.590	–	0.628
duration	negative	0.766	0.770	0.785	0.786	0.505	0.606	0.525	0.613
duration	positive	0.759	0.761	0.786	0.809	0.530	0.604	0.584	0.597
stops	negative	0.768	0.777	0.829	0.841	0.493	0.593	0.439	0.595
stops	positive	0.806	0.817	0.841	0.866	0.570	0.649	0.535	0.590
mean		0.770	0.778	0.795	0.805	0.536	0.612	0.521	0.591

Table 8: **Balanced accuracy per feature and direction.** Prototype patching at layers 18–25 with 32 tokens and $\alpha = 0.5$. The first four columns patch the BAYESLM with its own prior on each test set. The next three use the original test set and patch the ORACLELM with its own prior, and the BAYESLM's prior into the ORACLELM and the STARTLM. The last column is CAA steering at the 16 tokens after the *feedback* token ($\alpha = 0.25$) on the original test set. The STARTLM recipient was only run for duration and stops.

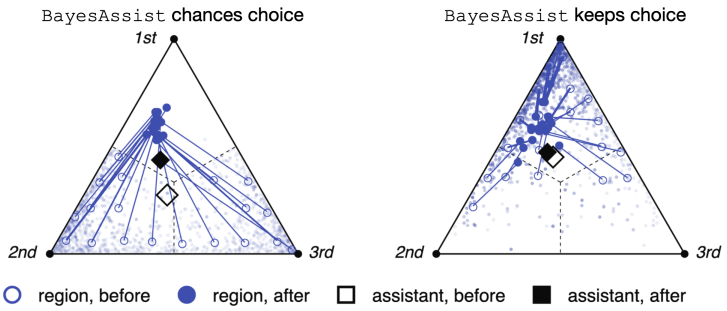

Figure 16: The choice policy on the simplex, with corners labeled by the BAYESASSIST's ranking of the three options under the updated belief f'_4, split by whether the BAYESASSIST's recommendation changes after the intervention. Open circles are the BAYESLM's mean policy before the intervention, filled circles after the intervention. The diamond marks the BAYESASSIST's own policy before (open) and after (filled).

While mathematically equivalent, in practice, using one or the other computation might not be innocuous: in an LM, a sequential computation could plausibly entail that the current beliefs are represented and used at specific context positions, whereas we assume that a one-swoop computation sources information across all tokens in the context. We investigate whether the Bayes LM processes information in a way that is consistent with sequential or one-swoop posterior computation by restricting the information flow in the model's processing of the input through attention masking. We expect that attention masking that is *less* consistent with the model's computations will lead to a *bigger* drop in the Bayes accuracy of its predictions.

Intuitively, the sequential computation posits that posterior beliefs are calculated as soon as evidence is encountered, and access to these beliefs is only needed for making a prediction on a next sample. That is, attending to option sets and feedback for them on turn i can be disabled after turn $i+1$. One-swoop calculation, on the other hand, requires only accessing the likelihood of all observations at the point where the calculation of the belief for a new prediction would occur, allowing to block attention between a recommendation and the feedback at intermediate conversation rounds preceding the target prediction. The masks operationalizing these restrictions are shown in Figure 17. The key manipulation is which tokens can access the feedback (i.e., evidence) tokens F_i: the sequential mask allows each next recommendation to attend only to the previous feedback, while the one-swoop mask allows attending to all previous options with their feedback for making the prediction at round five.

Bayesian fine-tuning seems to enable one-swoop, rather than sequential, forming of posterior beliefs: we observe a drop in Bayes accuracy of 16% when the one-swoop mask is applied to the Bayes LM, whereas the drop is 28% when the sequential mask is applied (see Figure 18). The behavioral choice policies resulting from the masks are shown in Figure 19. Randomly blocking the same number of

tokens causes a drop to an accuracy of 28-38% (depending on the conversation length where it is applied), confirming that both of these masks focus on relevant information flow.

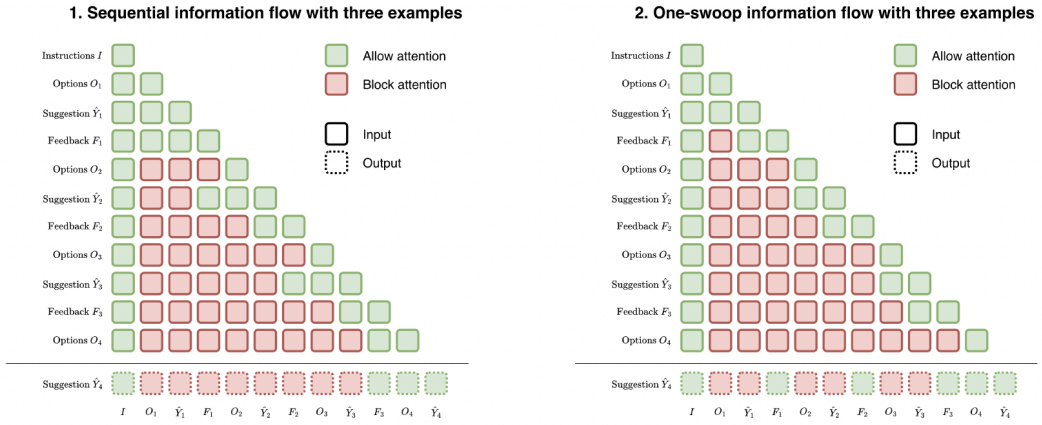

Figure 17: Overview of the attention masks applied for investigating how posterior beliefs might be formed.

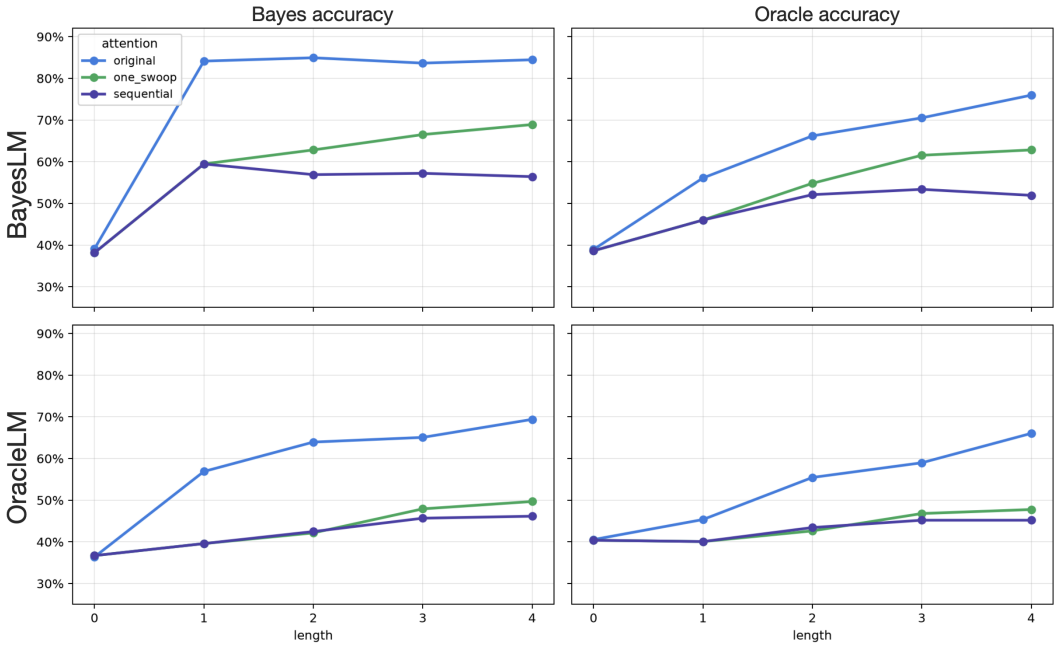

Figure 18: Overview of the attention masking accuracies (y-axis) of the BAYESLM and ORACLELM, across conversational rounds (x-axis).

G NEURO-SYMBOLIC BEHAVIORAL POLICY

The neuro-symbolic policies reported in § 6 were calculated using probed beliefs from the last layer of Gemma. We additionally calculate Bayes accuracy of neuro-symbolic policies based on beliefs from different layers and show them in Figure 20. This confirms that a neuro-symbolic policy can be derived once the beliefs are decodable, from about layer 15 on.

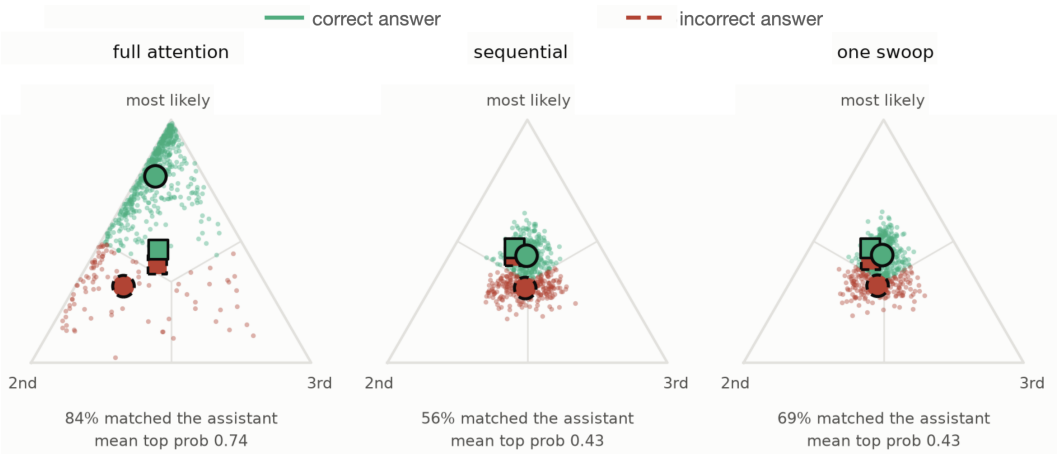

Figure 19: Distribution of the choice policy of the BAYESLM under masking.

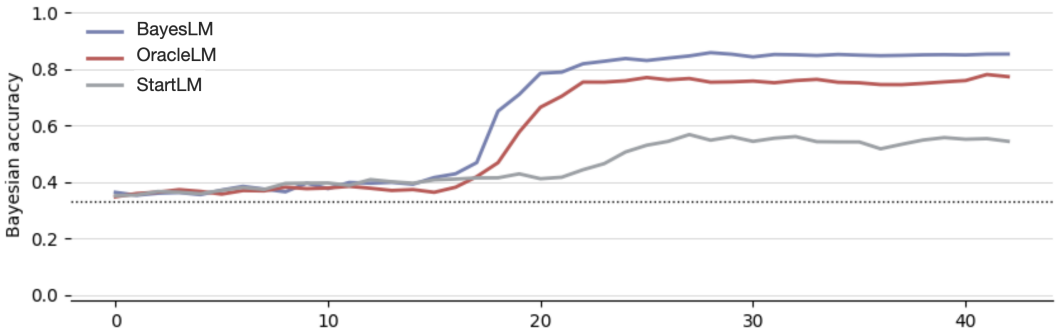

Figure 20: **Results of neuro-symbolic predictions across layers.** The development of the average Bayes accuracy of the three neuro-symbolic models when the probes stem from different model layers.

H RESULTS ACROSS BACKBONE MODELS

We repeat the main experiments addressing R1–R3 with Llama-3.1-8B-Instruct (Grattafiori et al., 2024) and Qwen-2.5-7B-Instruct (Qwen et al., 2025) as the STARTLM, each fine-tuned into a BAYESLM and an ORACLELM provided by Qiu et al. (2026). All figures show Gemma-2-9B (which was reported in the main text) alongside the two other backbones. Addressing R1 (acting Bayesian), Figure 21 shows the LMs' Bayes accuracies on the different test sets, confirming that the BAYESLM is more Bayes-accurate than the ORACLELM on every backbone, and is invariant to evidence order, and behaves as expected under noise. Figure 22 shows the LMs' oracle accuracy across test sets. The BAYESLM exceeds the ORACLELM on the original and shuffled sets for every backbone. All models lose accuracy in the presence of noise, as predicted by the BAYESASSIST, because the evidence is less informative of the user's preference. Addressing R2 (representing beliefs), Figure 23 shows the belief divergence between the decoded and the BAYESASSIST's prior on the original test set. On every backbone, the prior becomes decodable in the middle layers, and is encoded more strongly for the BAYESLM than for the ORACLELM. Addressing R3 (acting on beliefs), Figure 24 shows the balanced accuracy of the BAYESLM upon intervening on the prior. The intervention effect is largest for Gemma-2-9B, present for Llama-3.1-8B, and near chance for Qwen-2.5-7B Figure 25 shows the LMs' Bayes accuracy without masking and under the one-swoop and sequential masks. On every backbone, both masks reduce accuracy, and the sequential mask reduces it more, though the gap between the two masks is smaller for Llama-3.1-8B and Qwen-2.5-7B than for Gemma-2-9B.

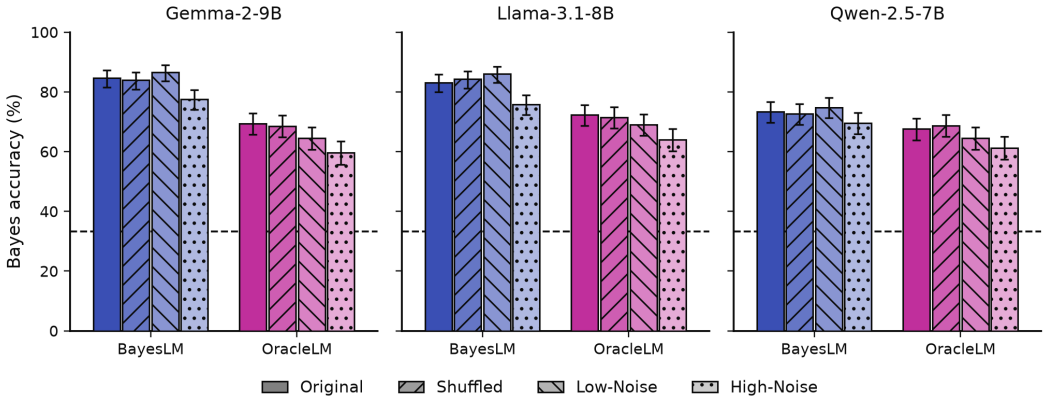

Figure 21: **Bayes accuracy across backbones (R1, § 3).** Bayes accuracy of the BAYESLM and ORACLELM on the four test sets, with 95% confidence intervals. Chance is $1/3$ (dashed line). The BAYESLM is more Bayes-accurate than the ORACLELM on every backbone, and is invariant to evidence order, and behaves as expected under noise.

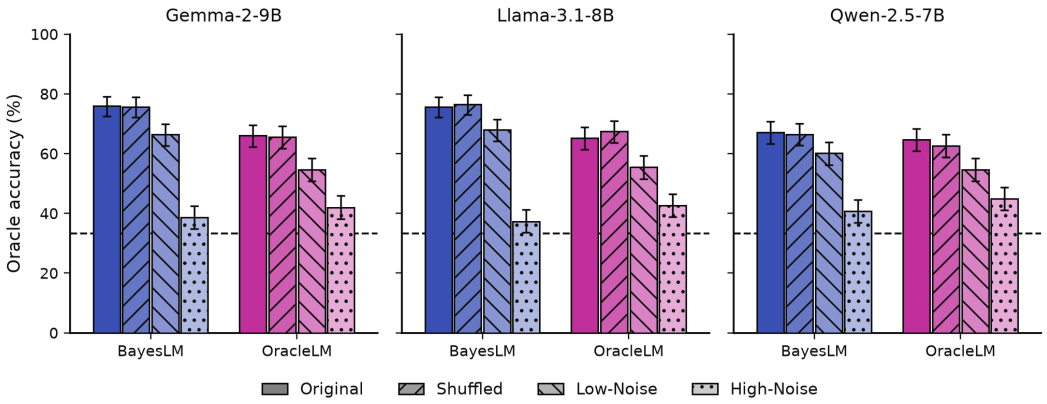

Figure 22: **Oracle accuracy across backbones (R1, § 3).** Oracle accuracy shows the proportion of recommendations matching the user's true preference. All models lose accuracy in the presence of noise, as the BAYESASSIST does, because the evidence is less informative of the user's preference. The BAYESLM exceeds the ORACLELM on the original and shuffled sets for every backbone.

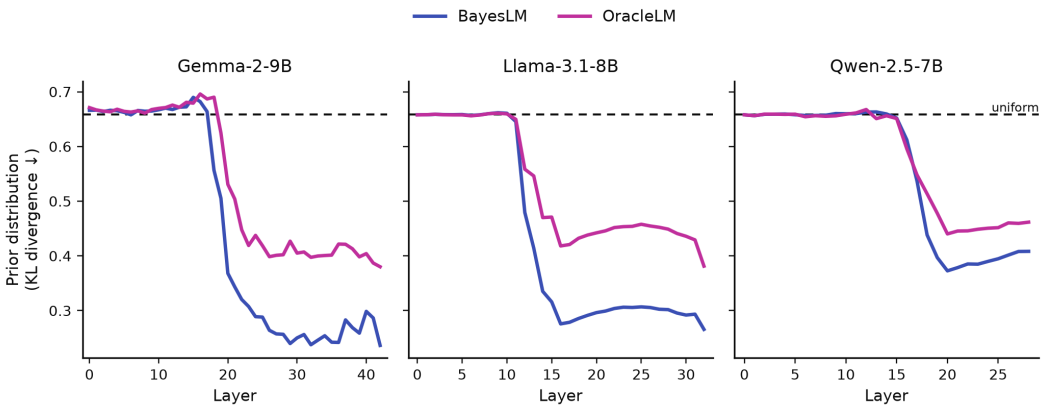

Figure 23: **Decoded prior across backbones (R2, § 4).** Belief divergence between the decoded and the BAYESASSIST's prior by layer, on the original set. The dashed line is the divergence of a uniform prior. On every backbone, the prior becomes decodable in the middle layers and more strongly for the BAYESLM than for the ORACLELM.

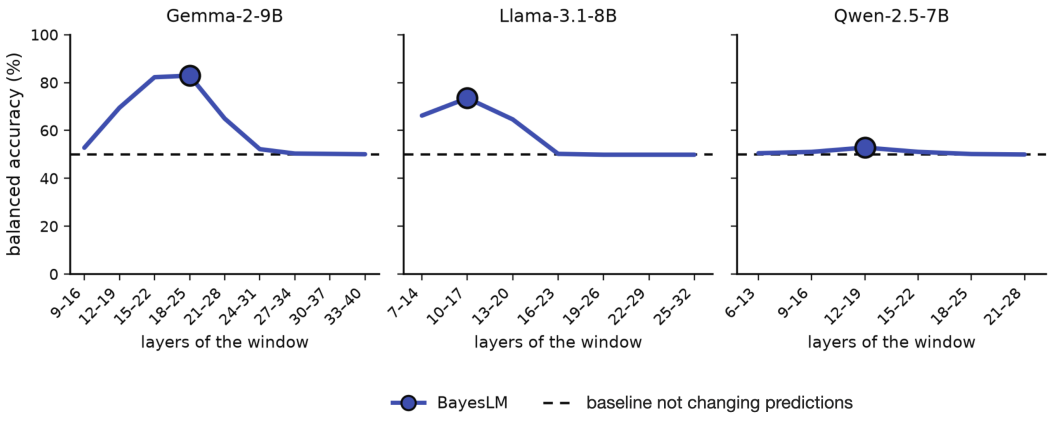

Figure 24: **Prior patching across backbones (R3, § 5).** Balanced accuracy of the BAYESLM upon intervening on the prior, by the layer window of the edit. The dashed line is a model that never changes its recommendation. The marker shows the best window. The effect is largest for Gemma-2-9B, present for Llama-3.1-8B, and near chance for Qwen-2.5-7B.

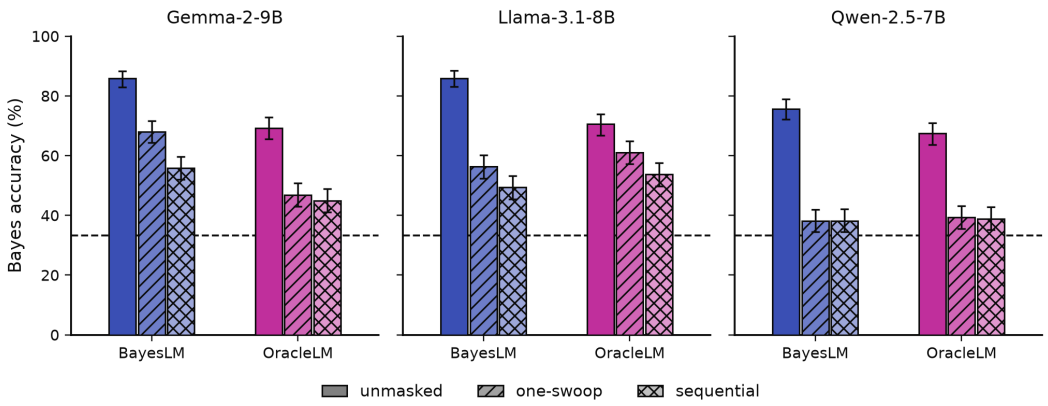

Figure 25: **Attention masking across backbones (R3, § 5).** Bayes accuracy without masking and under the one-swoop and sequential masks. Chance is $1/3$ (dashed line). On every backbone, both masks reduce accuracy, and the sequential mask reduces it more, though the gap between the two masks is smaller for Llama-3.1-8B and Qwen-2.5-7B than for Gemma-2-9B.