REVEALING THE BIASES OF LARGE LANGUAGE MODELS BY INFERRING PRIOR DISTRIBUTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) have been known to exhibit a wide range of social biases including name-nationality associations, occupation-gender associations and racial stereotypes. Existing approaches to evaluating these biases typically involve measuring surface level behavior where a model is fed a dataset of prompts and its outputs are scored relative to ground truth answers. Inspired by prior elicitation in cognitive science, we propose a different perspective that reconstructs latent bias as a prior underlying the model's behavior. We formulate the problem as a Bayesian inverse problem where, given prompt response observations on a bias benchmark dataset, we infer a low-dimensional latent distribution that summarizes the model's stereotyping tendencies. This paper develops the mathematical foundation, a practical reconstruction algorithm, and experiments using the Bias Benchmark for QA (BBQ) benchmark to infer the prior distributions of LLMs over responses in ambiguous social contexts.

1 INTRODUCTION

LLMs often produce outputs that reflect potentially harmful stereotypes and biases present in training data (Kotek et al., 2023; Gallegos et al., 2024; Bai et al., 2025; Hu et al., 2025). To evaluate social bias in NLP systems, prior work has developed association tests and benchmark datasets designed to elicit stereotyping (Nangia et al., 2020; Nadeem et al., 2021; Parrish et al., 2022). While these benchmarks provide valuable measurements, they typically summarize behavior through scores computed from generated text or from the observable probability token assignments. Thus, they measure the extent to which a model exhibits certain biases, but fail to explain why the preferences arise.

In this paper, we propose a different perspective. Rather than treating bias benchmarks only as evaluation metrics, we take a Bayesian approach and use them to probe language models in a way that lets us reconstruct a latent space that describes the models' prior beliefs. Concretely, we tackle the question: *Can we reconstruct a latent prior distribution over possible biased responses that explains an LLM's stereotypical preferences?*

A Bayesian framing for this problem is natural because LLMs are inherently probabilistic systems that predict a probability distribution over tokens conditioned on previous ones (Bengio et al., 2003; Radford et al., 2019; Siddiqui et al., 2025). The motivation for this approach comes from prior elicitation methods in cognitive science, where behavioral judgments have been used to infer the prior distributions that best explain human predictions (Garthwaite et al., 2005; O'Hagan et al., 2006). In this line of work, observed judgments can be used to diagnose which priors and prior parameters people use in certain task domains (Griffiths & Tenenbaum, 2006). Related ideas have also been extended to LLMs where prompt-based probing has been used to elicit prior distributions from language models in specific domains (Zhu & Griffiths, 2024).

Building on these ideas, we introduce a Bayesian surrogate model in which an LLM behaves as if it first samples a discrete latent "bias state" and then generates outputs according to a state-conditional response model. The central unknown is the prior distribution that summarizes how the model allocates probability across bias hypotheses in social contexts. We interpret this distribution as a summary of the systematic preference structure in an LLM. It is important to note that we are not assuming that LLMs are inherently Bayesian in that they are using Bayesian optimization to construct their outputs. Instead, we use a Bayesian surrogate model to analyze and explain the latent prior beliefs that are induced by the model's parameters.

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.

Mathematically, the Bayesian inverse problem framework provides a principled theoretical background for latent prior reconstruction (Dashti & Stuart, 2013). Given observations derived from token log-probabilities on candidate prompt completions, we define a likelihood function for the model and obtain a posterior distribution that approximates the desired latent prior. The main contributions of this paper are:

- A mathematical formulation for latent prior reconstruction as a Bayesian inverse problem.
- A practical reconstruction algorithm based on token log-probabilities using Bayesian inference.
- An empirical demonstration using the Bias Benchmark for QA (BBQ) dataset (Parrish et al., 2022).

2 Related Work

Quantifying Social Bias in Language Models. As language models become more powerful and increasingly widely used, AI ethics and safety have become central concerns (Leslie, 2019; Ferrara, 2023). As a result, the literature on social biases in these models proposes a broad range of evaluation protocols analyzing model behavior on different levels from embeddings to probabilities and generated text (Gallegos et al., 2024). Early work demonstrated that word embeddings encode implicit social associations analogous to those revealed by human implicit association tests and introduced methods for measuring such effects (Bolukbasi et al., 2016). Subsequent work extended association testing to sentences, allowing LLM biases to be probed in a way closer to implicit association tests in psychology (Caliskan et al., 2017). Another family of benchmarks measures bias directly from the model outputs, often through controlled choice tasks in which models select among a fixed set of options and are compared against an unbiased ground truth (Parrish et al., 2022). Other benchmark tests take this a step further and compare model probability assignments across minimally different texts or token options with different levels of stereotyping (Nangia et al., 2020; Nadeem et al., 2021).

Prior Elicitation in Cognitive Science and LLMs. In Bayesian models of cognition, priors are treated as structured assumptions that shape inductive judgments (Griffiths et al., 2008; Tenenbaum, 1998). Thus, reconstructing or eliciting a cognitive model's prior can provide insight into the implicit beliefs of a system. The central idea in prior elicitation is that observed predictions and choices can be used to infer or at least validate the priors that best explain behavior in a certain task domain (Griffiths & Tenenbaum, 2011; Griffiths et al., 2008). Work on everyday judgments has shown that human predictions can often be well modeled as Bayesian updates under latent priors, helping to explain why predictions take the forms that they do (Griffiths et al., 2008). More recently, these ideas have been extended to prior elicitation in LLMs through prompt-based iterated learning procedures that resemble Monte Carlo sampling schemes (Zhu & Griffiths, 2024). These experiments reflect those done on human judgment and provide evidence that a Bayesian framework can be useful for characterizing LLM predictions. We extend this line of work to analyze stereotype bias in LLMs. However, our approach differs in that we treat benchmark derived log-probability observations as data from which we reconstruct a latent distribution via Bayesian inversion rather than through sampling and iterated interaction.

Bayesian Computation and Uncertainty Quantification. Once the problem of prior elicitation is posed as finding a posterior distribution from a set of observations, posterior computation and uncertainty quantification can be carried out using Bayesian inference through optimization-based approximations, sampling methods, or variational methods (Tipping, 2003; Goan & Fookes, 2020). The common optimization-based approach that will be used in this paper is to use a maximum a posteriori (MAP) estimation to provide a point summary (Murphy, 2012) and then apply Laplace approximations by using local curvature (via the Hessian) around the MAP solution to approximate posterior uncertainty (Tierney & Kadane, 1986). On the other hand, sampling-based approaches like Markov Chain Monte Carlo (MCMC) and Hamiltonian Monte Carlo (HMC) can provide asymptotically exact posterior approximations under standard conditions (Hastings, 1970; Duane et al., 1987), but are often computationally expensive and reserved for settings requiring high precision (Salimans et al., 2015). Variational inference offers a scalable optimization-based alternative by approximating the posterior within a tractable distribution family (Blei et al., 2017), and may be useful for future higher-dimensional prior elicitation. However, for our use case, Laplace approximation provides a simpler and more interpretable Hessian-based uncertainty estimate.

3 BAYESIAN INVERSE PROBLEM FRAMEWORK

In this section we provide a high-level overview[1] of the Bayesian inverse problem (Stuart, 2010; Dashti & Stuart, 2013). The goal of an inverse problem is to infer some unknown parameter u from observations y generated by a noisy forward model

$$y = G(u) + \eta, \tag{1}$$

where $G : U \to Y$ is a forward operator from the space of unknowns U to the space of possible observations Y and η represents observation noise.

In the Bayesian framework, u is treated as a random variable so that we can characterize its posterior distribution given observations y (Stuart, 2010). Specifically, a prior distribution is assigned to u with density $\pi_0(u)$ and Bayesian inference is used to combine this prior with a likelihood describing the probability of observing y given u. Assuming the noise has density ρ, the likelihood is given by $\rho(y \mid u) = \rho(y - G(u))$. Bayesian inference then gives a posterior distribution of the form

$$\pi^y(u) \propto \rho(y - G(u))\pi_0(u), \tag{2}$$

where the proportionality constant depends only on the observations y. It is often convenient to express the likelihood in terms of a potential function $\Phi(u; y) = -\log \rho(y - G(u))$. In this setting, this function is also often referred to as the negative log likelihood. Rewriting Equation 2 using this potential gives us

$$\pi^y(u) \propto \exp(-\Phi(u; y))\pi_0(u). \tag{3}$$

In this form, we can see that the negative log likelihood $\Phi(u; y)$ encodes the agreement between the observations and those predicted by u while the prior encodes our beliefs about plausible values of the unknown parameter. Together, this provides a complete characterization of the posterior distribution over u, which serves as a solution to the inverse problem.

4 OUR FORMULATION

We now interpret the abstract Bayesian inverse problem framework discussed above in context of the problem of reconstructing an LLM's latent bias prior from token level probability observations on a controlled bias task. The purpose of this section is to identify, in the notation of Section 3, the unknown variable u, the data space Y, the forward model G, and the likelihood ρ. In doing so, we obtain a concrete finite-dimensional Bayesian inverse problem whose solution is a posterior distribution over a latent probability vector summarizing the model's systematic bias tendencies.

4.1 UNKNOWN VARIABLE AND OBSERVABLES

To begin, let us fix a bias domain in which we have K mutually exclusive response categories. For instance, for a task related to gender biases, we may take $K = 3, \{\text{Male, Female, Unknown}\}$. We represent the latent bias prior as a probability vector $\theta \in \Delta^{K-1}$ where Δ^{K-1} is the set of possible probability vectors in $\mathbb{R}^K$. For computational convenience, we parameterize θ via log-ratios relative to a reference category K, defining $u \in U = \mathbb{R}^{K-1}$ and $u_k = \log(\theta_k/\theta_K)$, $k \in [K-1]$. For a given u in our unknown parameter space U, the corresponding probability vector $\theta(u)$ can then be recovered via the inverse transformation

$$\theta_k(u) = \frac{e^{u_k}}{1 + \sum_{j=1}^{K-1} e^{u_j}}, \quad k \in [K-1], \qquad \theta_K(u) = \frac{1}{1 + \sum_{j=1}^{K-1} e^{u_j}}. \tag{4}$$

Thus, the unknown variable in our inverse problem is the vector $u \in U = \mathbb{R}^{K-1}$ which we interpret as an encoding of the latent bias prior over the K candidate hypotheses in log-ratios. In the running example, if we transform $u \in U \mapsto \theta \in \Delta^2$ via Equation 4, then θ_1, θ_2, and θ_3 represent the model's latent prior mass on the hypotheses Male, Female, and Unknown, respectively. It is important to stress that this latent variable is not meant to represent an explicit internal object stored by the LLM but rather it is a low-dimensional surrogate parameter whose posterior distribution summarizes the systematic preference structure reflected in the model's behavior.

[1]For a more rigorous measure-theoretic treatment, including the use and derivation of the necessary Bayes' formula and Radon–Nikodym derivatives, we refer the reader to Appendix A.

To define the observational data and data space Y, let $x_1, \ldots, x_m$ be prompts with the same shared answer set of K candidate answers. For each x_i, the model outputs log-probabilities $\ell_i \in \mathbb{R}^K$ across the K possible answers. Passing to the same log-ratio representation, we define $y_i \in \mathbb{R}^{K-1}$, $y_{i,k} = \ell_{i,k} - \ell_{i,K}$. Thus, our full observable data take the form $y = (y_1, \ldots, y_m) \in Y = (\mathbb{R}^{K-1})^m$. This choice of representation in log-ratios removes the redundancy inherent in representing a probability by all K coordinates simultaneously since the probabilities sum to one and thus only contain $K - 1$ degrees of freedom. Furthermore, it allows us to work in the entirety of $\mathbb{R}^{K-1}$ instead of the constrained simplex. In the next section, we will also see how this yields a particularly simple form for the forward operator.

4.2 Prompt Dependent Evidence and Forward Operator

The key modeling issue is how the prompts $x_1, \ldots, x_m$ should enter the inverse problem as they inherently affect the likelihood in the formulation. To tackle this, we model each prompt as contributing a prompt-specific evidence term while preserving a common latent prior across prompts.

Formally, let $\Lambda : \{x_1, \ldots, x_m\} \to \mathbb{R}^K$ be a known map assigning each prompt $x_i \in \{x_1, \ldots, x_m\}$ to an evidence vector $\Lambda(x) = (\Lambda_1(x), \ldots, \Lambda_K(x))$. We interpret $\Lambda(x_i)$ as encoding the prompt dependent evidence supplied by x_i in favor of each candidate category. Thus u represents a global latent bias prior shared across all prompts while $\Lambda(x_i)$ captures the local way in which the prompt x_i shifts or reweighs that prior. Passing again to log-ratios, we define the offset $\delta_i \in \mathbb{R}^{K-1}$, $\delta_{i,k} = \Lambda_k(x_i) - \Lambda_K(x_i), \quad k \in [K-1]$. This transformation ensures that only relative difference in the evidence effect the model and allows us to be consistent with the log-ratio representation.

We can now define the prompt dependent forward operator directly in the log-ratio space. For each prompt x_i, define $G(x_i, \cdot) : U \to \mathbb{R}^{K-1}$ by $G(x_i, u) = u + \delta_i$. Collecting these prompt-based maps together, we get the full forward operator as $G : U \to Y$ where $G(u) = (G(x_1, u), \ldots, G(x_m, u))$.

We can now place the model into the exact form of Equation 1 discussed in Section 3. We assume the observed log-ratios satisfy $y = G(u) + \eta$, where η is some noise $\eta = (\eta_1, \ldots, \eta_m) \in Y$ accounting for residual variability in the log-probability outputs of the LLM beyond what is explained by the prompt dependence and the forward operator G. This is now precisely the finite-dimensional noisy inverse problem previously formalized in Section 3.

4.3 Prior, Likelihood, and Posterior

To obtain a tractable formulation, we specify the natural isotropic Gaussian prior measure μ_0 on u so that $u \sim \mathcal{N}(0, \tau^{-1} I)$ and assume independent isotropic Gaussian noise $\eta_i \sim \mathcal{N}(0, \sigma^2 I)$ for all $i \in [m]$ where $\tau > 0$ is a precision parameter and σ^2 is the variance. Let π_0 denote the corresponding prior density. Note that this prior induces a logistic-normal distribution on the implied probability vector $\theta(u)$. Equivalently, on the full data space Y, $\eta \sim \mathcal{N}(0, \sigma^2 I)$ on $Y = (\mathbb{R}^{K-1})^m$. We then obtain from Equation 2 that the posterior density satisfies

$$\pi^y(u) \propto \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{m} \|y_i - G(x_i, u)\|^2 - \frac{\tau}{2} \|u\|^2 \right).$$

Substituting in for $G(x_i, u)$, we find that the negative log-likelihood satisfies the potential formulation from Equation 3 where

$$\Phi(u; y) = \frac{1}{2\sigma^2} \sum_{i=1}^{m} \|y_i - u - \delta_i\|^2. \tag{5}$$

In summary, we have now posed the bias prior elicitation problem as a Bayesian inverse problem. The unknown variable is the log-ratio vector u and the observables are the log-ratios extracted from the token log-probabilities generated by the LLM. Furthermore, the forward operator is the map based on the evidence vectors $\Lambda(x_i)$ and the likelihood is obtained from the noise model discussed in Section 3.

5 RECONSTRUCTION ALGORITHM

We now face the computational problem of reconstructing the posterior distribution over u from the observed data. Our approach consists of two stages: (1) computing a MAP estimate using gradient based optimization, and (2) approximating the posterior distribution via a Laplace approximation around the MAP point.

5.1 GRADIENT BASED OPTIMIZATION

Taking the negative log-likelihood from Equation 5 and combining with the Gaussian prior $u \sim \mathcal{N}(0, \tau^{-1}I)$, our negative log-posterior, up to an additive constant, is

$$\mathcal{J}(u) = \frac{1}{2\sigma^2} \sum_{i=1}^{m} \|y_i - u - \delta_i\|^2 + \frac{\tau}{2}\|u\|^2. \tag{6}$$

Thus, our MAP estimator is $u_{\text{MAP}} \in \arg\min_{u \in \mathbb{R}^{K-1}} \mathcal{J}(u)$. The objective $\mathcal{J}(u)$ is smooth and convex, and can thus be optimized efficiently using first-order methods by computing the gradient $\nabla_u \mathcal{J}(u)$ (Boyd & Vandenberghe, 2004). In practice, we implement u as a trainable parameter in a PyTorch model, and define a loss function corresponding to $\mathcal{J}(u)$. The optimization proceeds over minibatches of data, where each batch contributes a stochastic estimate of the gradient. PyTorch's automatic differentiation computes gradients via backpropagation, and the optimizer updates u iteratively. The details of this procedure can be seen in Algorithm 1.

Algorithm 1: MAP Reconstruction via Gradient-Based Optimization

Input: Observed log-ratios $\{y_i\}_{i=1}^m$, prompt offsets $\{\delta_i\}_{i=1}^m$, noise level σ, prior precision τ
Output: MAP estimate u_{MAP}

```
Initialize u ∈ R^{K-1}
for each epoch do
    for each minibatch B do
        Compute predictions ŷ_i = u + δ_i for i ∈ B
        Compute loss J_B(θ) = 1/(2σ²) Σ_{i∈B} ||y_i − ŷ_i||₂² + τ/2 ||u||₂²
        Update u using Adam or SGD
    end for
end for
return final value of u as the MAP estimate u_MAP
```

5.2 LAPLACE APPROXIMATION FOR UNCERTAINTY QUANTIFICATION

To capture uncertainty of the MAP estimate, we use a Laplace approximation centered at u_{MAP}. Let $H_{\text{MAP}} = \nabla_u^2 \mathcal{J}(u_{\text{MAP}})$ denote the Hessian of the objective at the MAP solution. Then $u \mid y \approx \mathcal{N}(u_{\text{MAP}}, H_{\text{MAP}}^{-1})$. We interpret this uncertainty on the simplex by transforming samples from this Gaussian approximation through the inverse log-ratio map in Equation 4.

In practice, after Algorithm 1 converges, we fit the local curvature of the loss using the training data. We implement this efficiently with `laplace-torch` (Daxberger et al., 2021), which computes an approximate Gaussian posterior over the unknown parameter u. We then sample from this posterior to compute uncertainty estimates as summarized in Algorithm 2.

Algorithm 2: Laplace Approximation with `laplace-torch`

Input: Trained model with parameter u_{MAP}, training data
Output: Approximate posterior over u

```
Initialize Laplace approximation object
Fit curvature using training data
Obtain Gaussian approximation N(u_MAP, H_MAP^{-1})
Sample u^(n) and compute θ^(n) = θ(u^(n))
return posterior samples and uncertainty estimates
```

6 EXPERIMENTS ON SHIFT CIPHERS

Before applying our method to large-scale bias benchmark datasets, we begin with a controlled experiment designed to validate whether our framework can recover a meaningful latent prior from LLM outputs. The goal of this experiment is to isolate a setting in which prompt-dependent effects are minimized and so that we can test the basic structure of our algorithm.

Recent work has explored the presence of cognitive biases in language models across a variety of tasks including reasoning under uncertainty and structured problem solving settings such as decoding shift ciphers (McCoy et al., 2024a;b). Motivated by these observations, we construct a simple shift-cipher task in which the model is asked to infer the shift used to produce a given ciphertext. By carefully designing the inputs, we obtain a setting in which there is no meaningful evidence favoring a particular answer, allowing us to probe the model's baseline preference over shifts.

6.1 PROBLEM SETUP

We consider a simple shift cipher in which each plaintext letter is shifted by $n \in \{0, \ldots, 25\}$ positions in the alphabet, wrapping from z to a. Rather than providing meaningful encrypted text, we generate prompts $\{x_i\}_{i=1}^{m}$ consisting of random lowercase strings. The candidate answers are the possible shifts $\{0, \ldots, 25\}$, so this corresponds to the setting of Section 4 with $K = 26$.

Since the strings are random, they do not encode any particular plaintext, so all shifts are equally plausible. In the notation of Section 4, this corresponds to the special case in which the prompt dependent evidence map $\Lambda(x_i)$ is constant across all categories. This setting thus isolates the latent parameter u as the sole systematic component of the model's outputs making it an ideal test case for evaluating whether our method can recover a stable prior distribution.

For our experiments, we use the Google Gemma 3 (Team et al., 2025) and Meta Llama 3.2 (Grattafiori et al., 2024) models as our lightweight LLM models. A detailed review of the training procedure and hyperparameters as well as the training loss curves can be seen in Appendix C.

6.2 RESULTS

We first validate our algorithm on a synthetic control dataset where we inject a known prior distribution into the ciphertext observation model while removing the LLM models. This experiment serves to verify that the algorithm behaves as intended under a controlled setting. Further details on this validation experiment can be found in Appendix B.

For the ciphertext experiments involving the lightweight LLMs, across all three models, the recovered prior distributions exhibit clear non-uniform structure despite the absence of prompt dependent signal. Furthermore, the credible intervals are relatively tight, suggesting that the inferred priors are well identified. Notably, each model exhibits a distinct latent prior structure as shown in Figure 1.

In particular, the Gemma models display a sharper concentration on a small number of shifts, particularly at low rotations. This effect is especially pronounced for Gemma 3 4B IT, whose recovered prior has a dominant mode around a single shift. In this case, the bias might reflect the fact that the traditional Caesar cipher is performed with a shift of 3. In contrast, Llama 3.2 3B IT produces a more diffuse prior with greater probability mass on higher rotations than the Gemma models.

6.3 INTERPRETATION

This experiment validates our framework in a setting where the prompt-dependent evidence $\Lambda(x_i)$ is deliberately removed. The non-uniform priors suggest that LLMs can encode systematic preferences over answer categories independent of the input prompt. This aligns with prior observations of task-based cognitive biases in LLMs from McCoy et al. (2024a), but here we provide a quantitative framework for explicitly recovering and characterizing the latent prior beliefs.

While simplified, this experiment shows that we can successfully extract latent bias structure from model outputs. In more realistic settings where $\Lambda(x_i)$ varies, this framework can help disentangle prompt-driven effects from latent bias, providing a principled tool for analyzing LLM behavior.

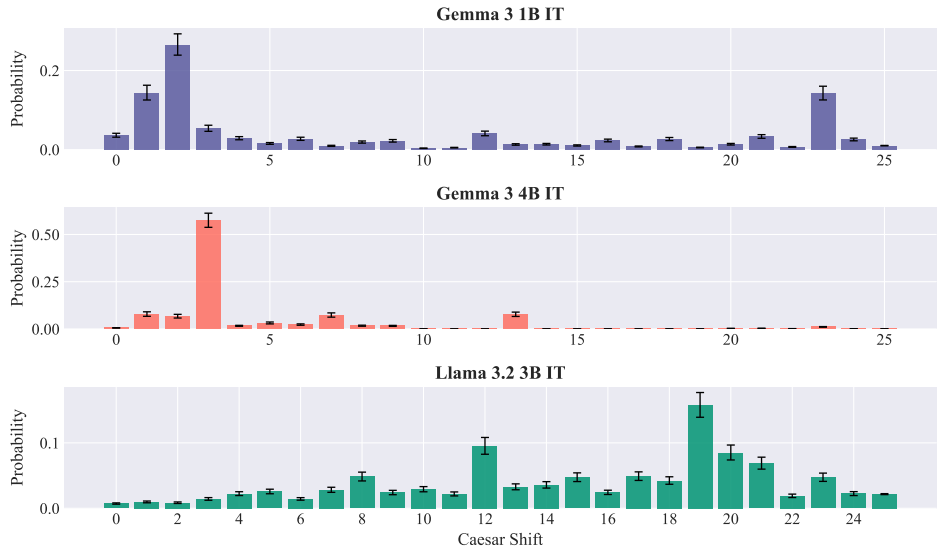

Figure 1: Recovered prior distributions over shifts for Gemma 3 1B IT, Gemma 3 4B IT, and Llama 3.2 3B IT. Error bars denote 95% credible intervals obtained via Laplace approximation. Even in the absence of input signals, all models exhibit non-uniform prior distributions, with notable differences across architectures and scales.

7 Experiments on BBQ

Having validated the reconstruction framework on a controlled task, we next apply it to a real bias benchmark setting. We use the Bias Benchmark for Question Answering (BBQ) dataset introduced by Parrish et al. (2022). BBQ is a multiple choice question answering benchmark for evaluating social bias in language models. BBQ is organized around nine social dimensions, including age, nationality, socioeconomic status, and sexual orientation. It also distinguishes between ambiguous contexts where the question lacks sufficient information to determine the answer, and disambiguated contexts where additional information resolves the ambiguity. We focus exclusively on the ambiguous context portion of BBQ. This subset plays the same conceptual role as the random string shift cipher experiment as it suppresses prompt-specific evidence which simplifies our model.

7.1 Problem Setup

The raw BBQ answer choices vary across examples, even within the same social dimension. To place all examples in a common latent prior framework, we map each item's three answer choices into a reduced semantic answer space with three categories: $\{\mathrm{S}, \mathrm{A}, \mathrm{U}\}$, where S denotes the answer choice consistent with the intended *Stereotyping*, A denotes the answer choice that goes *Against* the intended stereotyping, and U denotes the answer choice corresponding to an *Unknown* answer.

Because social dimensions differ substantially in context, we define a separate latent prior for each BBQ dimension. Only six dimensions (age, disability status, gender identity, race / ethnicity, religion, sexual orientation) admit direct mappings to the reduced semantic answer space. We therefore restrict our analysis to these dimensions. For each dimension d, we solve the inverse problem for the latent parameter $u^{(d)} \in \mathbb{R}^{K-1}$ from Section 4 with $K = 3$. A detailed review of the training procedure and hyperparameters as well as the training loss curves can be seen in Appendix C.

7.2 Results

Figure 2 shows that the credible intervals for the recovered priors are relatively tight across all models and social dimensions, indicating that the inferred priors are well identified. This suggests that despite the prompt and answer choice variations that exist within each social dimension, the reduction to the answer space $\{\mathrm{S}, \mathrm{A}, \mathrm{U}\}$ still produces stable and interpretable results.

We next compare the recovered latent priors with the standard BBQ bias score metric. In the BBQ framework, positive scores indicate preference for stereotype-consistent answers S, negative scores indicate preference for stereotype-inconsistent answers A, and values near zero indicate no clear systemic bias (Parrish et al., 2022). However, this scalar score combines two distinct behaviors: whether the model abstains under ambiguity, captured by probability mass on U, and whether the non-abstaining probability mass is directionally skewed towards S or A. Our reconstruction separates these effects by recovering the full latent distribution.

This distinction is visible in the Age social dimension. Figure 3, which visualizes the BBQ bias scores for each model, shows that both Gemma 3 4B IT and Llama 3.2 3B IT have large positive BBQ scores which indicate stereotype-consistent behavior. However, Figure 2 shows that the underlying priors differ substantially. While both models have low probability mass in U contributing to high bias scores, Gemma 3 4B IT assigns significantly more probability mass to S relative to A indicating a stronger directional bias not fully captured by the aggregate score metric.

A similar ambiguity appears when comparing the Disability Status and Religion dimensions for Gemma 3 4B IT. The BBQ scores for these dimensions are both positive and similar in magnitude but the recovered priors indicate a different mechanism of bias. In Disability Status, the model assigns low probability to U and substantial probability mass to both S and A which suggests overconfidence in the ambiguous setting. On the other hand, for the Religion dimension, the model assigns a majority of its probability mass to U, and the positive BBQ score arises primarily because the remaining probability mass is more skewed towards S than towards A.

The recovered priors also expose cases where near-balanced or moderate BBQ scores can hide low uncertainty. For example, we note that Gemma 3 1B IT often assigns comparable probability mass to S and A while failing to place a majority on U. Such behavior is not strongly directionally biased, yet is still undesirable because the model is making unsupported social inferences. These examples illustrate that similar BBQ benchmark scores can arise from fundamentally different underlying behaviors. Our framework resolves this ambiguity by recovering the full latent distribution over the answer choices $\{S, A, U\}$.

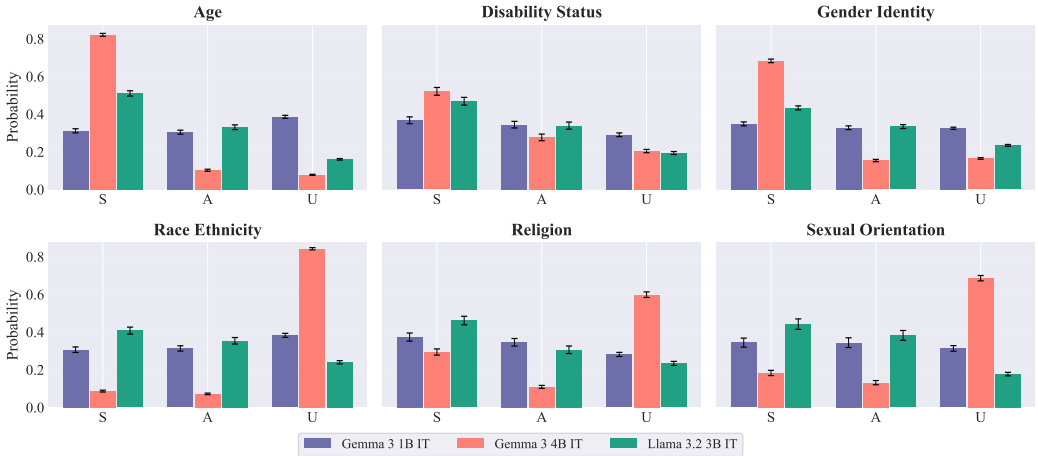

Figure 2: Recovered latent priors over the reduced answer space $\{S, A, U\}$ for each BBQ social dimension, with 95% credible intervals derived from the Laplace approximation. Each model exhibited largely different latent priors across the different social dimensions.

7.3 Interpretation

The BBQ ambiguous context experiment gives us a concrete demonstration of how our method reveals the latent structure in model behavior that is not captured by the BBQ benchmark scores. Although benchmarks such as BBQ provide useful measures of accuracy and directional bias, they do not reveal the underlying mechanisms responsible for biased results. Our experiment shows that there are many components of bias corresponding to different behaviors in models such as directional bias and overconfidence that cannot be articulated by a single scalar score value.

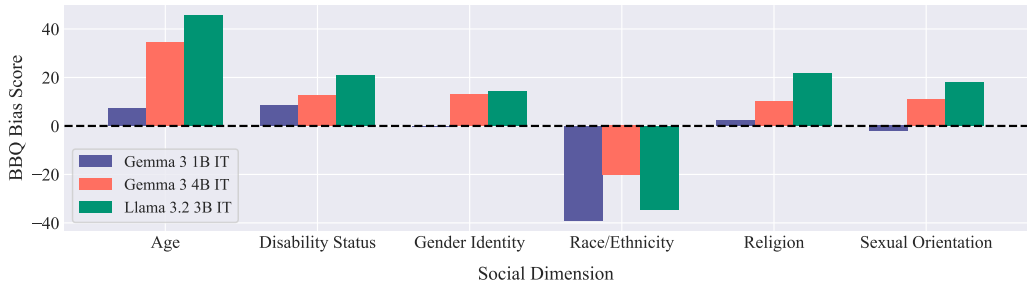

Figure 3: BBQ bias scores for each model and social dimension. Positive values indicate bias towards the stereotype-consistent answer S, negative values indicate bias towards the stereotype-inconsistent answer A, and values near zero indicate no systematic bias in either direction.

More broadly, the experiment demonstrates that latent prior elicitation provides a more principled way to analyze biased model behavior. By constructing a reduced answer space to represent prompt variation, our approach was able to reconstruct a stable prior structure that represents the underlying preferences and tendencies in each model.

8 Limitations and Future Directions

Our approach, as it stands, still has several limitations. First, it relies on mapping raw answers to a reduced answer space, such as the $\{\mathrm{S}, \mathrm{A}, \mathrm{U}\}$ representation used in our BBQ experiments. While this abstraction worked for the cases we analyzed, such a clean reduction may not be possible in general and may obscure nuances and finer-grained distinctions important for interpretation. Second, our experiments focused on settings where the prompt-dependent evidence term $\Lambda(x_i)$ is constant, yielding a simplified forward model. This assumption is appropriate for some contexts, but does not capture the general setting in which prompts carry meaningful information.

These limitations suggest several directions for future work. One direction is to develop richer latent representations beyond the low-dimensional reduced answer spaces used here, allowing the framework to capture more nuanced model behavior. Another direction is to incorporate prompt-dependent evidence through $\Lambda(x_i)$, potentially learning Λ from prompt embeddings or another representation.

More broadly, latent prior elicitation is not limited to bias analysis. The same framework could be applied to other domains where LLMs display systematic behavior under uncertainty including reasoning tasks or algorithmic tasks such as the shift cipher problem considered in this paper. Exploring these broader applications may reveal additional implicit structures encoded within LLMs.

Taken together, our results suggest that studying the latent distributions governing LLM behavior can provide valuable insight into the reasons why models behave the way they do. The Bayesian inverse problem formulation therefore offers a promising framework for uncovering these latent mechanisms underlying language model predictions.

9 Conclusion

In this work, we introduced a framework for eliciting latent priors from large language models by formulating the problem as a Bayesian inverse problem. Across both controlled synthetic experiments and real benchmark evaluations, we demonstrated that it is possible to recover stable and interpretable prior structures from model outputs. Furthermore, we demonstrated using the well-established BBQ bias benchmark that prior elicitation under this framework provides us with much more nuance in explaining model behavior that was not previously characterized through these benchmark datasets. While benchmarks like BBQ do provide useful measures of accuracy and directional bias, they do not capture the underlying reasons for the biased results. Our latent prior formulation provides an extra layer of explainability by providing the latent distribution which gives rise to these bias score results.

AI USE STATEMENT

In this work, we used generative AI tools to edit the manuscript to improve grammar, readability, and conciseness. All AI-assisted edits were reviewed by the authors and the authors take full responsibility for the final content of this work.

ETHICS STATEMENT

We evaluate social bias in large language models using controlled experimental settings. In particular, model behavior in our experiments should not be interpreted as definitive statements about the broader societal impact of a model.

REPRODUCIBILITY STATEMENT

For reproducibility, the source code for the project is included in the supplementary materials. Furthermore, specific hyperparameters used for the algorithms outlined in Section 5 can be found in Section C.1.1. The specific experimental setups for each experiment can be found in Sections B.1, 6.1, and 7.1.

REFERENCES

Xuechunzi Bai, Angelina Wang, Ilia Sucholutsky, and Thomas L Griffiths. Explicitly unbiased large language models still form biased associations. *Proceedings of the National Academy of Sciences*, 122(8):e2416228122, 2025.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(Feb):1137–1155, 2003.

David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

Tolga Bolukbasi, Kai-Wei Chang, James Zou, Venkatesh Saligrama, and Adam T Kalai. Man is to computer programmer as woman is to homemaker? Debiasing word embeddings. In D Lee, M Sugiyama, U Luxburg, I Guyon, and R Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Anton Bovier. Stochastic processes I. *Lecture notes, Berlin Mathematical School*, 2006.

Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 978-0-521-83378-3.

Aylin Caliskan, Joanna J Bryson, and Arvind Narayanan. Semantics derived automatically from language corpora contain human-like biases. *Science*, 356(6334):183–186, 2017.

Masoumeh Dashti and Andrew M Stuart. The Bayesian approach to inverse problems. *arXiv preprint arXiv:1302.6989*, 2013.

Erik Daxberger, Agustinus Kristiadi, Alexander Immer, Runa Eschenhagen, Matthias Bauer, and Philipp Hennig. Laplace redux - effortless Bayesian deep learning. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20089–20103. Curran Associates, Inc., 2021.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222, 1987.

Emilio Ferrara. Should ChatGPT be biased? Challenges and risks of bias in large language models. *arXiv preprint arXiv:2304.03738*, 2023.

Isabel O Gallegos, Ryan A Rossi, Joe Barrow, Md Mehrab Tanjim, Sungchul Kim, Franck Dernoncourt, Tong Yu, Ruiyi Zhang, and Nesreen K Ahmed. Bias and fairness in large language models: A survey. *Computational linguistics*, 50(3):1097–1179, 2024.

Paul H Garthwaite, Joseph B Kadane, and Anthony O'Hagan. Statistical methods for eliciting probability distributions. *Journal of the American Statistical Association*, 100(470):680–701, 2005.

Ethan Goan and Clinton Fookes. Bayesian neural networks: An introduction and survey. In *Case Studies in Applied Bayesian Data Science: CIRM Jean-Morlet Chair, Fall 2018*, pp. 45–87. Springer, 2020.

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

Thomas L Griffiths and Joshua B Tenenbaum. Optimal predictions in everyday cognition. *Psychological Science*, 17(9):767–773, 2006.

Thomas L Griffiths and Joshua B Tenenbaum. Predicting the future as Bayesian inference: People combine prior knowledge with observations when estimating duration and extent. *Journal of Experimental Psychology: General*, 140(4):725–743, 2011.

Thomas L Griffiths, Charles Kemp, and Joshua B Tenenbaum. *Bayesian Models of Cognition*, pp. 59–100. Cambridge Handbooks in Psychology. Cambridge University Press, 2008.

W K Hastings. Monte carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Tiancheng Hu, Yara Kyrychenko, Steve Rathje, Nigel Collier, Sander Van Der Linden, and Jon Roozenbeek. Generative language models exhibit social identity biases. *Nature Computational Science*, 5(1):65–75, 2025.

Hadas Kotek, Rikker Dockum, and David Sun. Gender bias and stereotypes in large language models. In *Proceedings of the ACM Collective Intelligence Conference*, pp. 12–24, 2023.

David Leslie. Understanding artificial intelligence ethics and safety. *arXiv preprint arXiv:1906.05684*, 2019.

R Thomas McCoy, Shunyu Yao, Dan Friedman, Mathew D Hardy, and Thomas L Griffiths. Embers of autoregression show how large language models are shaped by the problem they are trained to solve. *Proceedings of the National Academy of Sciences*, 121(41):e2322420121, October 2024a.

Thomas McCoy, Shunyu Yao, Dan Friedman, Matthew Hardy, and Thomas L Griffiths. When a language model is optimized for reasoning, does it still show embers of autoregression? An analysis of OpenAI o1. *arXiv preprint arXiv:2410.01792*, 2024b.

Kevin P Murphy. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012. ISBN 978-0-262-01802-9.

Moin Nadeem, Anna Bethke, and Siva Reddy. StereoSet: Measuring stereotypical bias in pretrained language models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 5356–5371, Online, August 2021. Association for Computational Linguistics.

Nikita Nangia, Clara Vania, Rasika Bhalerao, and Samuel R Bowman. CrowS-pairs: A challenge dataset for measuring social biases in masked language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1953–1967, Online, November 2020. Association for Computational Linguistics.

Anthony O'Hagan, Caitlin E Buck, Alireza Daneshkhah, J Richard Eiser, Paul H Garthwaite, David J Jenkinson, Jeremy E Oakley, and Tim Rakow. *Uncertain Judgements: Eliciting Experts' Probabilities*. Wiley, 1 edition, July 2006. ISBN 978-0-470-02999-2 978-0-470-03331-9. doi: 10.1002/0470033312.

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. BBQ: A hand-built bias benchmark for question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2086–2105, Dublin, Ireland, May 2022. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

Tim Salimans, Diederik Kingma, and Max Welling. Markov chain Monte Carlo and variational inference: Bridging the gap. In *International Conference on Machine Learning*, pp. 1218–1226, 2015.

Shoaib Ahmed Siddiqui, Yanzhi Chen, Juyeon Heo, Menglin Xia, and Adrian Weller. On evaluating LLMs' capabilities as functional approximators: A Bayesian evaluation framework. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5826–5835, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics.

Andrew M Stuart. Inverse problems: A Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, et al. Gemma 3 Technical Report. *arXiv preprint arXiv:2503.19786*, 2025.

Joshua B Tenenbaum. Bayesian modeling of human concept learning. In M Kearns, S Solla, and D Cohn (eds.), *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1998.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. *Journal of the American Statistical Association*, 81(393):82–86, 1986.

Michael E Tipping. Bayesian inference: An introduction to principles and practice in machine learning. In *Summer school on machine learning*, pp. 41–62. Springer, 2003.

Jian-Qiao Zhu and Thomas L Griffiths. Eliciting the priors of large language models using iterated in-context learning. *arXiv preprint arXiv:2406.01860*, 2024.

A Measure-Theoretic Background

This section collects the measure-theoretic background needed for a rigorous statement of the inverse problem described in Section 3. We will define some required measure-theoretic objects and present some results relevant to the Bayesian formulation of the inverse problem. The structure of this section parallels that of Stuart (2010); however, the derivations and proofs are our own.

A.1 Basic Preliminaries

We begin by reviewing the basic objects of study that we will be using throughout this paper.

Definition A.1. A *measurable space* is a pair $(\Omega, \mathcal{F})$, where Ω is the sample space and $\mathcal{F}$ is the σ-algebra of events on Ω. A *measure space* is a triple $(\Omega, \mathcal{F}, \mu)$, where μ is the measure. If $\mu(\Omega) = 1$, then μ is called a *probability measure*, and $(\Omega, \mathcal{F}, \mu)$ is called a *probability space*.

Definition A.2. A *Banach space* is a complete normed vector space $(X, \|\cdot\|)$. If X is a topological space, then $\mathcal{B}(X)$ denotes the Borel σ-algebra generated by the open subsets of X.

In our formulation of the Bayesian inverse problem, we will mainly be working under the assumption that Ω in any measurable space $(\Omega, \mathcal{F})$ is a separable Banach space $(X, \|\cdot\|_X)$ and $\mathcal{F}$ is the Borel σ-algebra $\mathcal{B}(X)$. Now we introduce a basic measure-theoretic relation between measures.

Definition A.3. Let $(\Omega, \mathcal{F})$ be a measurable space, and let μ, ν be measures on it. We say that μ is *absolutely continuous* with respect to ν, and write $\mu \ll \nu$, if

$$\nu(A) = 0 \quad \Longrightarrow \quad \mu(A) = 0, \qquad \forall A \in \mathcal{F}.$$

If both $\mu \ll \nu$ and $\nu \ll \mu$, then μ and ν are called *equivalent*.

Additionally, it will be important to formally establish the idea of conditional distributions as our overarching goal is to reconstruct the latent bias priors conditioned on the dataset prompts and LLM output probabilities.

Definition A.4. Let $(S, \mathcal{A})$ and $(T, \mathcal{B})$ be measurable spaces, and let ν be a probability measure on $S \times T$. Denote the second marginal of ν by

$$\nu_T(B) := \nu(S \times B), \qquad B \in \mathcal{B}.$$

A family $\{\nu^y\}_{y \in T}$ of probability measures on $(S, \mathcal{A})$ is called a *regular conditional distribution of x given y* under ν if:

1. for every $A \in \mathcal{A}$, the map $y \mapsto \nu^y(A)$ is $\mathcal{B}$-measurable;
2. for every $A \in \mathcal{A}$ and $B \in \mathcal{B}$,

$$\nu(A \times B) = \int_B \nu^y(A)\, \nu_T(dy).$$

We note that when S and T are, as we assume, separable Banach spaces equipped with their Borel σ-algebras, regular conditional distributions exist and thus for the following subsections it is enough to assume existence when needed (Bovier, 2006).

A.2 Finite-Dimensional Bayes' Formula

With the basic definitions out of the way, the main goal is to rigorously derive Bayes' formula in a way that is applicable to our prior reconstruction problem.

We begin by stating the following fundamental result from measure theory.

Theorem A.5 (Radon–Nikodym Theorem). *Let μ and ν be measures on the same measurable space $(\Omega, \mathcal{F})$. If $\mu \ll \nu$ and ν is σ-finite then there exists a ν-measurable function $f : \Omega \to [0, \infty]$ such that*

$$\mu(A) = \int_A f d\nu \qquad \forall A \in \mathcal{F}.$$

This function f is unique ν-almost everywhere and is known as the *Radon-Nikodym derivative* of μ with respect to ν. The derivative is written as

$$\frac{d\mu}{d\nu}(x) = f(x).$$

This Radon-Nikodym derivative plays an important role in our derivation as it allows us to compare measures and express one in terms of another. It also helps us generalize the idea of a probability density which is essential for our formal derivation of the required tools to discuss the Bayesian inverse problem.

We continue with the derivation of the relevant finite-dimensional Bayes formula with the following theorem which utilizes the Radon-Nikodym derivative.

Theorem A.6 (Conditional Change of Measure). *Let μ and ν be probability measures on $S \times T$, where $(S, \mathcal{A})$ and $(T, \mathcal{B})$ are measurable spaces. Let $(x, y) \in S \times T$ denote a generic point. Assume that $\mu \ll \nu$, and that μ has a Radon-Nikodym derivative $\phi := \frac{d\mu}{d\nu}$ with respect to ν. Assume further that the conditional distribution of x given y under ν exists which we denote by $\nu^y(dx)$. Then the conditional distribution of $x|y$ under μ, denoted $\mu^y(dx)$, exists. Moreover,*

$$\mu^y \ll \nu^y,$$

and the Radon–Nikodym derivative is given by

$$\frac{d\mu^y}{d\nu^y}(x) = \begin{cases} \dfrac{1}{c(y)}\phi(x, y) & \text{if } c(y) > 0, \\ 1 & \text{otherwise}, \end{cases} \tag{7}$$

where

$$c(y) := \int_S \phi(x, y)\, \nu^y(dx) \qquad \text{for all } y \in T.$$

Proof. Let ν_T and μ_T be the T-marginals of ν and μ respectively. Since

$$\mu(dx, dy) = \phi(x, y)\nu(dx, dy),$$

and since ν decomposes as

$$\nu(dx, dy) = \nu^y(dx)\nu_T(dy),$$

we have, for measurable $A \subseteq S$ and $B \subseteq T$,

$$\mu(A \times B) = \int_B \int_A \phi(x, y)\nu^Y(dx)\nu_T(dy).$$

If we define

$$c(y) := \int_S \phi(x, y)\nu^y(dx),$$

then, taking $A = S$, we get

$$\mu_T(B) = \mu(S \times B) = \int_B c(y)\nu_T(dy).$$

Thus, $c = \frac{d\mu_T}{d\nu_T}$. We can then define for μ_T-a.e. y,

$$\mu^y(A) := \frac{1}{c(y)} \int_A \phi(x, y)\nu^y(dx), \qquad c(y) > 0.$$

Notice, the set $\{y : c(y) = 0\}$ is μ_T-null since

$$\mu_T(\{y : c(y) = 0\}) = \int_{c=0} c(y)\nu_T(dy) = 0.$$

Thus, μ^y may be defined arbitrarily on this set. Let us take for these values $\mu^y = \nu^y$. Then μ^y is a probability measure for μ_T-a.e. y and moreover

$$\int_B \mu^y(A)\mu_T(dy) = \int_B \frac{1}{c(y)} \int_A (\phi(x, y)\nu^y(dx)c(y)\nu_T(dy) = \int_B \int_A \phi(x, y)\nu^y(dx)\nu_T(dy) = \mu(A\times B).$$

Therefore μ^y is the conditional distribution of x given y under μ. Finally, for μ_T-a.e. y with $c(y) > 0$,

$$\mu^y(A) = \int_A \frac{1}{c(y)}\phi(x, y)\nu^y(dx),$$

so $\mu^y \ll \nu^y$ and

$$\frac{d\mu^y}{d\nu^y}(x) = \frac{1}{c(y)}\phi(x, y) \qquad \text{for } c(y) > 0.$$

For $c(y) = 0$, since we took $\mu^y = \nu^y$ for this set, we get Radon-Nikodym derivative 1 and thus proves the claim. □

As we will see, this conditional change of measure theorem is precisely what we need to formalize the finite-dimensional Bayes' formula in rigorous measure-theoretic language.

Corollary A.7 (Finite-Dimensional Bayes' Formula). *Let $u \in \mathbb{R}^n$ and $y \in \mathbb{R}^q$ be random variables defined on a common probability space. Assume that the prior law of u admits a Lebesgue density $\rho(u)$ on $\mathbb{R}^n$, and that the conditional law of y given u admits a Lebesgue density $\rho(y \mid u)$ on $\mathbb{R}^q$. Assume also that*

$$Z(y) := \int_{\mathbb{R}^n} \rho(y \mid u)\rho(u)\, du > 0.$$

Then the conditional law of u given y admits a density $\rho(u \mid y)$ with respect to Lebesgue measure on $\mathbb{R}^n$, and

$$\rho(u \mid y) = \frac{\rho(y \mid u)\rho(u)}{\int_{\mathbb{R}^n} \rho(y \mid u)\rho(u)\, du}. \tag{8}$$

Proof. Let us define two measures on $\mathbb{R}^n \times \mathbb{R}^q$ as follows. First, let

$$\nu_0(du, dy) := \rho(u)\, du\, dy.$$

This is the product of the prior law of u with the Lebesgue measure on $\mathbb{R}^q$. Second, let ν denote the joint law of (u, y). By definition of conditional density, ν has density

$$(u, y) \mapsto \rho(y \mid u)\rho(u)$$

with respect to the Lebesgue measure $du\, dy$, and thus

$$\nu(du, dy) = \rho(y|u)\rho(u)du\, dy.$$

Therefore $\nu \ll \nu_0$ and the Radon-Nikodym derivative is clearly

$$\frac{d\nu}{d\nu_0}(u, y) = \rho(y \mid u).$$

Notice, under ν_0, the conditional law of u given y is simply the prior law of u because ν_0 is the product measure. Thus

$$\nu_0^y(du) = \rho(u)du.$$

Let us now apply Theorem A.6 to the pair of measures ν and ν_0 on $\mathbb{R}^n \times \mathbb{R}^q$. Notice, while ν_0 is not a probability measure, it is σ-finite, and the theorem extends to this setting. In particular, Equation 7 then implies that the conditional law ν^y of u given y under ν satisfies

$$\frac{d\nu^y}{d\nu_0^y}(u) = \frac{\rho(y \mid u)}{\int_{\mathbb{R}^n} \rho(y \mid u)\nu_0^y(du)}$$

for $Z(y) = \int_{\mathbb{R}^n} \rho(y \mid u)\rho(u)\, du > 0$. Substituting in $\nu_0^y(du) = p(u)du$, we obtain

$$\frac{d\nu^y}{d\nu_0^y}(u) = \frac{\rho(y \mid u)}{\int_{\mathbb{R}^n} \rho(y \mid u)\rho(u)du}.$$

Since ν_0^y has density $\rho(u)$ with respect to the Lebesgue measure, it follows that ν^y has density

$$\rho(u \mid y) = \frac{d\nu^y}{du}(u) = \frac{d\nu^y}{d\nu_0^y}(u) \cdot \frac{d\nu_0^y}{du}(u).$$

Thus, we get precisely

$$\rho(u \mid y) = \frac{\rho(y \mid u)\rho(u)}{\int_{\mathbb{R}^n} \rho(y \mid u)\rho(u)du}.$$

□

This gives a very powerful expression for the probability of a random variable u given an observation of y, which requires only the knowledge of the prior probability density $\rho(u)$ and the conditional probability density $\rho(y \mid u)$. Both of these are readily available in most modeling scenarios as will be the case in our application.

B Controlled Validation on Synthetic Data

Before applying this algorithm to LLMs, we provide a synthetic experiment with a known groundtruth prior θ^* to validate our method's ability to reconstruct the latent parameter that it was designed to infer. We structure this experiment with the general context of ciphertext rotations which is the setting of the LLM based experiment in Section 6.

B.1 Experimental Setup

We begin with a known structured groundtruth prior $\theta^* \in \Delta^{25}$ and set varying prompt evidence $\Lambda(x_i)$ for each synthetic data point. For our prior, we use a softmax sum of sines and cosines for a multimodal groundtruth. Furthermore, each data point is set to have prompt evidence arbitrarily favoring shifts of 12 or 18. This could occur, for example, if the data in the probing dataset has a few particularly common default categories.

Since we want to control the output observations y that we feed our model, in this experiment, observations are synthetic logits generated from the known prior, the varying prompt-evidence, and controlled Gaussian noise. We compare our method to the naive approach of taking the empirical mean of the outputs plus bootstrapped confidence intervals to see how our method converges to the true prior distribution more effectively than a basic average over the posterior distribution.

B.2 Reconstruction Accuracy

Figure 4 demonstrates that the distribution produced by our reconstruction algorithm closely matches the known prior distribution. In particular, the 95% credible intervals obtained from the Laplace-based uncertainty quantification contain the known prior.

We also see clearly the advantage of our reconstruction algorithm over the naive empirical mean over the posterior. The empirical mean is influenced by varying prompt specific evidence and incorrectly allocates probability mass away from the true known prior.

Overall, this control experiment provides evidence that our reconstruction algorithm is able to reproduce a distribution that closely matches a known set prior.

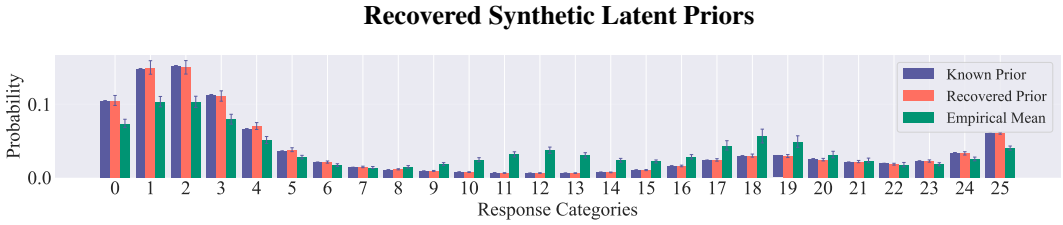

Figure 4: The recovered prior using our reconstruction algorithm much more closely resembles the control prior distribution. Meanwhile, the naive empirical mean method is biased by the varying prompt evidence.

C Experimental Details and Training Dynamics

In this section, we provide experimental details and training dynamics for the experiments conducted on shift ciphers (Section 6) and BBQ (Section 7). The code to replicate the experiments can be found in the supplementary materials. Furthermore, all experiments were run on MIG-partitioned NVIDIA A100 GPUs with 10 GB of GPU memory through an internal cluster. We obtained the underlying models from Hugging Face under model IDs `google/gemma-3-1b-it`, `google/gemma-3-4b-it`, and `meta-llama/Llama-3.2-3B-Instruct`.

C.1 Shift Ciphers

C.1.1 Training Procedure and Hyperparameters

Each prompt x_i is constructed by sampling a string of random length with characters drawn uniformly from $\{a, \ldots, z\}$. These strings are then embedded into a fixed prompt template of the form:

```
"The following text was encrypted using a Caesar
cipher: <string>. What is the shift (0--25)? Answer
with a single integer."
```

For each prompt x_i, we computed the model's conditional log-probabilities over candidate answers corresponding to the integers $0, \ldots, 25$.

We fit the MAP estimator described in Section 5 using the PyTorch implementation outlined in Algorithm 1. The latent parameter $u \in \mathbb{R}^{25}$ is initialized randomly over an isotropic Gaussian $\mathcal{N}(0, \tau^{-1}I)$ and optimized using the Adam optimizer.

In our experiments, we use the following hyperparameters: number of prompts $m = 200$, learning rate $= 10^{-2}$, number of epochs $= 400$, prior precision $\tau = 1.0$. Since our prompt dependence is constant, we derive σ from the variation in the model's output probabilities. After convergence, we compute the MAP estimate u_{MAP} as well as the uncertainty as outlined in Algorithm 2 and transform the results into the probability vector θ_{MAP} using the inverse log-ratio map to obtain a distribution over shifts.

C.1.2 Training Dynamics

For the shift cipher experiments, we find that across all three language models, the training loss decreases smoothly over time as shown in Figure 5. This behavior confirms that the derived MAP objective is well conditioned in this setting.

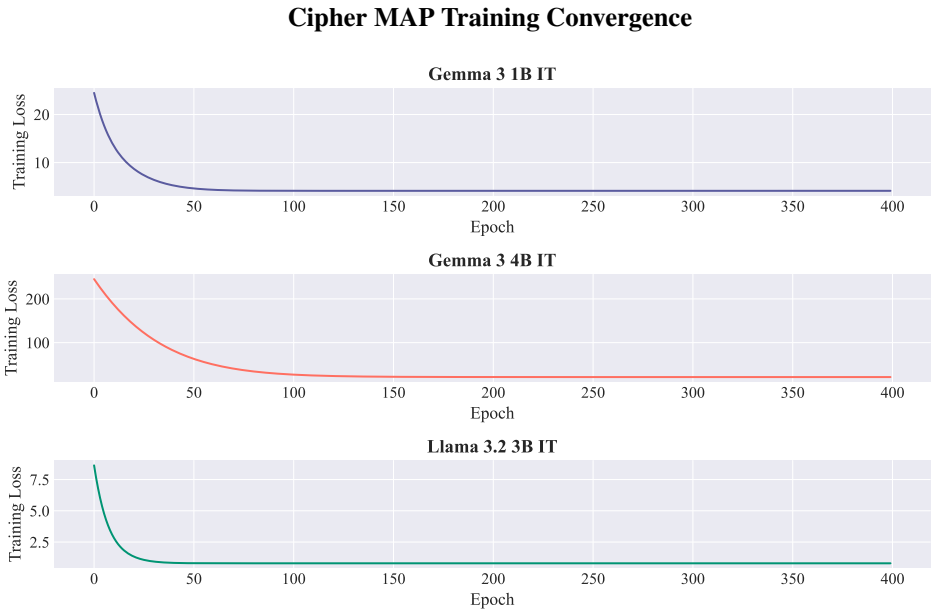

Figure 5: Training loss history for the Adam driven MAP optimization for Gemma 3 1B IT, Gemma 3 4B IT, and Llama 3.2 3B IT on the ciphertext problem. The smooth loss curves indicate a stable training objective.

C.2 BBQ Experiments

C.2.1 Training Procedure and Hyperparameters

For each BBQ item, we construct a prompt by concatenating the context, questions, and answer options in a fixed multiple choice format. Concretely, if the three answer choices are a_1, a_2, and a_3, we use a prompt template of the form:

> "Context: <context>. Question: <question>.
> Options: A. <a_1>, B. <a_2>, C. <a_3>. Answer with only
> A, B, or C."

For each model and social dimension d, we evaluate the model's conditional probabilities over the tokens corresponding to A, B, and C, and normalize these probabilities across the three choices before mapping to the reduced answer space $\{\mathrm{S}, \mathrm{A}, \mathrm{U}\}$. Then we estimate the latent parameter $u^{(d)}$ by minimizing the MAP objective from Equation 6.

In our experiments we optimize this objective for each social dimension using Adam with the same fixed hyperparameters from the ciphertext experiment across all models and dimensions. The number of prompts m for these experiments, however, comes directly from the number of prompts in each social dimension of the BBQ dataset. Again, we derive σ from the variation in the model's output probabilities and we run Algorithm 1 to find the MAP estimate u_{MAP} before running Algorithm 2 for uncertainty quantification.

C.2.2 Training Dynamics

Examining the optimization dynamics of the MAP estimation procedure, Figure 6 shows the excess objective over training epochs for each model and social dimension. We plot the excess objective

because the absolute scale of the objective differs across models and dimensions which is expected due to variations in the number of examples and variability of observations y_i.

Notably, the convergence behavior differs between models. The Gemma 3 1B IT model exhibits the smallest initial objective values and demonstrates the fastest convergence which is reasonable due to its small model size. On the other hand, the Gemma 3 4B IT model displays the largest initial objective values and takes the largest number of epochs to converge to a stable baseline.

Most importantly, however, we notice that across all models and all social dimensions, the optimization exhibits smooth decay towards zero, indicating stable convergence of our Adam-based optimization procedure.

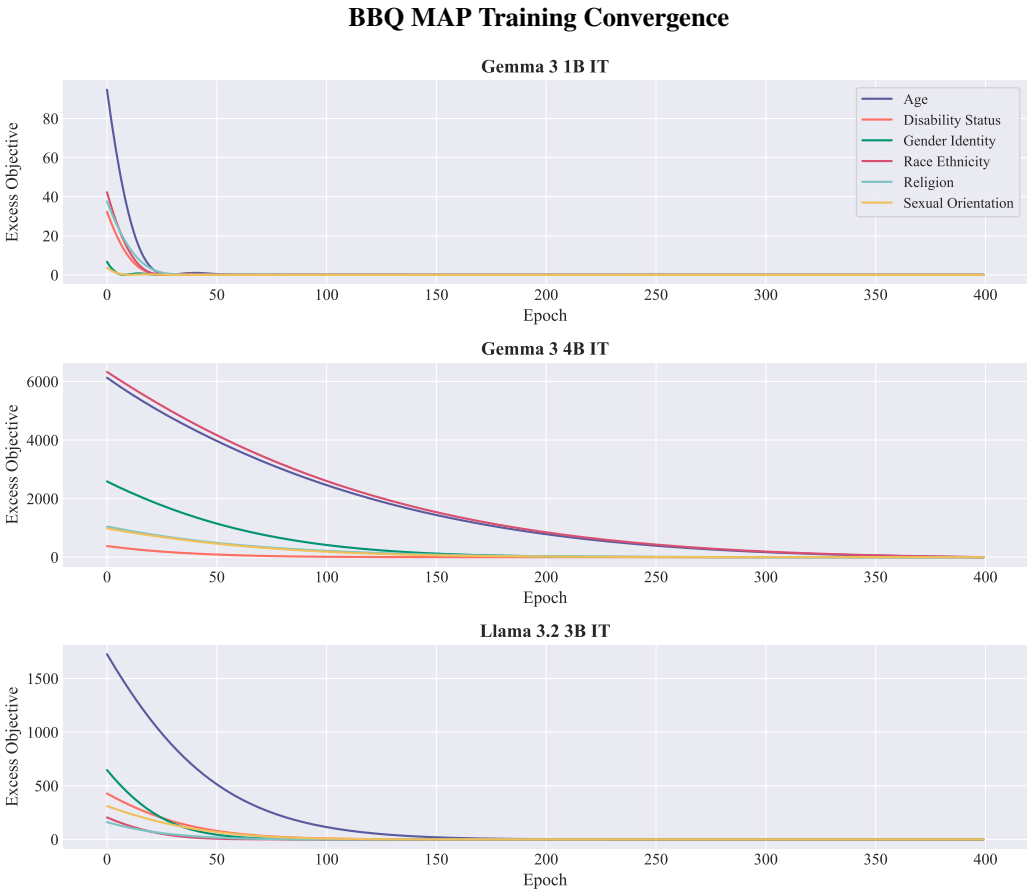

Figure 6: Training loss history for the Adam driven MAP optimization for Gemma 3 1B IT, Gemma 3 4B IT, and Llama 3.2 3B IT. The training curves are shown separately for each social dimension. We note that each training run exhibits stable convergence.