Recovering Mental Representations from Large Language Models with Markov Chain Monte Carlo

Jian-Qiao Zhu (jz5204@princeton.edu)
Department of Computer Science
Princeton University

Haijiang Yan (haijiang.yan@warwick.ac.uk)
Department of Psychology
University of Warwick

Thomas L. Griffiths (tomg@princeton.edu)
Department of Psychology and Computer Science
Princeton University

Abstract

Simulating sampling algorithms with people has proven a useful method for efficiently probing and understanding their mental representations. We propose that the same methods can be used to study the representations of Large Language Models (LLMs). While one can always directly prompt either humans or LLMs to disclose their mental representations introspectively, we show that increased efficiency can be achieved by using LLMs as elements of a sampling algorithm. We explore the extent to which we recover human-like representations when LLMs are interrogated with Direct Sampling and Markov chain Monte Carlo (MCMC). We found a significant increase in efficiency and performance using adaptive sampling algorithms based on MCMC. We also highlight the potential of our method to yield a more general method of conducting Bayesian inference *with* LLMs.

Keywords: Mental representation, Large Language Models, Markov Chain Monte Carlo, Gibbs Sampling, Bayesian inference

Introduction

How do we know what representations artificial intelligence (AI) systems are using? For "white box" machine learning models, such as decision trees and Bayesian models, the representations are typically transparent and directly tied to the features and architecture of the model. Interpreting these models often involves looking at the coefficients, rules, or structures they use to make predictions. However, state-of-the-art AI systems frequently employ "black box" deep neural networks (e.g., LeCun et al., 2015; Vaswani et al., 2017), which are notoriously difficult to interpret.

The increasingly proprietary nature of models used in AI can also mean that their internal mechanisms are not readily accessible, posing a significant challenge for researchers who seek to understand the representations used by these models. Historically, the representations used by neural network models have been identified by analyzing the activation patterns of artificial neurons (e.g., Kornblith et al., 2019). However, the efficacy of neuron-level approaches diminishes as AI systems expand in both depth and the number of model parameters. In this context, we propose an alternative approach, drawing inspiration from cognitive psychology, to investigating the representations used by AI systems via their behaviors (i.e., the outputs they produce).

Cognitive psychologists have spent decades developing methods for elucidating the content of individuals' mental representations, such as the structure of object categories and the utilities assumed to different choice actions (Sanborn et al., 2010; Shepard & Arabie, 1979; Torgerson, 1958). These mental representations, while not directly observable, can be inferred through the analysis of behavior. In this paper, we adapt behavioral methods based on sampling from subjective probability distributions to AI systems. We evaluate the efficiency and performance of three such methods, with the goal of exploring the correspondence between the representations inferred from AI systems and those of humans.

Our focus in this paper is on recovering color representations of an object, which can be defined within a 3D space. This choice is strategic: it addresses the concern that in simpler domains certain behavioral methods are not distinguishable from each other, while in more complex domains the visualization of results becomes challenging. Formally, an agent's color representation can be conceptualized as a probability distribution over a color space x, conditioned upon a given object c, expressed as $p(x|c)$. Here, x represents a color defined in terms of Hue, Saturation, and Lightness (HSL) values. For instance, the mental representation of a strawberry's color would be represented as a probability distribution across a range of colors, each specified by unique HSL parameters.

Our analysis focuses on GPT-4 as an example system, based on its impressive ability to solve a wide range of problems that were previously only solvable by humans. Its capabilities extend to engaging in open-ended dialogues and demonstrating a surprising familiarity with visual concepts (Bubeck et al., 2023; Rathje et al., 2023). The remainder of this paper is dedicated to applying behavioral methods to extract and analyze GPT-4's representation of color. It is important to note, however, that the applicability of these behavioral methods is not confined solely to GPT-4. Indeed, these methods can be readily adapted and applied to other AI systems, provided they possess the necessary knowledge base. This flexibility highlights the potential for broader implications and uses of our methodological approach in the evolving landscape of AI research.

Background

Our work draws on a class of cognitive psychology methods that elicit mental representations in humans by integrating people into sampling algorithms. A notable example of such an approach is the World Color Survey (Kay et al., 2009). In this survey, people are presented with colors

In L. K. Samuelson, S. L. Frank, M. Toneva, A. Mackey, & E. Hazeltine (Eds.), *Proceedings of the 46th Annual Conference of the Cognitive Science Society.*

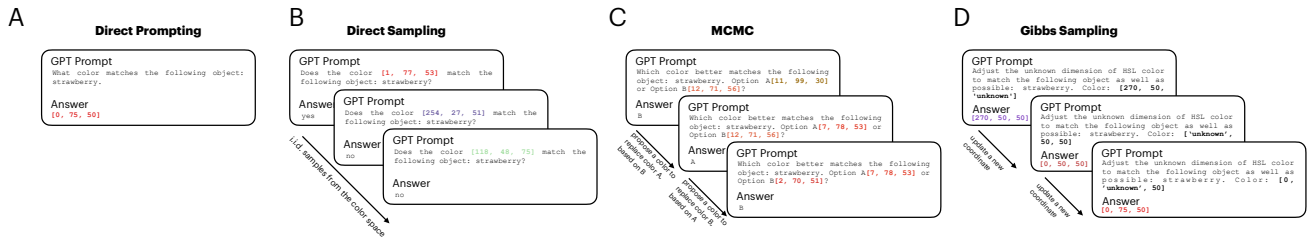

Figure 1: Illustrations of the four behavioral methods used to recover mental representations for GPT-4. **(A) Direct Prompting with GPT-4:** GPT-4 is directly prompted to generate a HSL color code corresponding to a specified object. **(B) Direct Sampling with GPT-4:** In this iterative process, a random HSL color code is sampled and presented to GPT-4, which then evaluates the extent to which this color matches the target object. **(C) Markov chain Monte Carlo (MCMC) with GPT-4:** Each iteration involves proposing a new color, derived from the previously selected color, and then deciding whether to accept this new color or retain the old one. **(D) Gibbs Sampling with GPT-4:** In each step, GPT-4 is tasked with deducing and filling in a missing dimension of the HSL color code to better match the target object. In all panels, HSL color codes are colorized to assist easier comparison.

that exhaustively sample from the color space, and they are then asked to provide evaluations of these colors (Kay et al., 2009). However, exhaustively enumerating every possible stimulus quickly becomes infeasible for dealing with high-dimensional or continuous-scale stimuli because the space is simply too vast to explore thoroughly. In contrast, more recent methods that have adopted adaptive sampling algorithms, such as Markov chain Monte Carlo (MCMC), to explore people's representation more efficiently. These methods have shown enhanced efficacy in exploring the structure of mental representations in domains including color, emotional prosody, face, and fruit (e.g., Sanborn and Griffiths, 2007).

Probing Large Language Models

Approaches to probing and interpreting information encoded in LLMs at the neuronal level primarily involve associating internal representations with external properties. This is done by training a secondary classifier on the activation of artificial neurons, with the aim of predicting specific properties (Alain & Bengio, 2016). Researchers typically use a trained LLM to generate representations, then employ another classifier that uses these representations to predict a certain property. This method has shown promise in assessing whether LLMs encode syntactic information (Belinkov, 2022) and, more recently, in analyzing the semantic structure of sentences (Zhang, McCoy, et al., 2023). However, we diverge from these methods by focusing on recovering representations from LLMs using behavioral methods, making our work complementary to existing approaches.

From Behaviors to Representations

As shown in Figure 1, we tested four behavioral methods, which can be broadly categorized into two classes: static and adaptive. Static methods typically involve presenting participants with a predefined set of stimuli, selected by the researcher prior to the commencement of the experiment. These methods do not modify the stimuli in response to participants' judgments during the course of the experiment. Examples of static methods include Direct Prompting and Direct Sampling.

In contrast, adaptive methods dynamically tailor the selection of stimuli for participants based on their previous responses. This approach allows for a more dynamic and responsive experimentation process. Notable examples of adaptive methods are Markov chain Monte Carlo (MCMC) and Gibbs Sampling with People (Harrison et al., 2020; Sanborn & Griffiths, 2007). Both methods iteratively adjust the selection of stimuli, with the aim of achieving a more accurate representation of the participant's mental state by considering their prior judgments.

Direct Prompting with GPT-4

Perhaps the most basic behavioral method to elicit an agent's mental representation involves instructing it to introspectively disclose it. In GPT-4, this could be achieved by directly prompting the model to reveal the conditional probability $p(x|c)$ by providing the object c.

For example, in exploring the color representation of a strawberry, we directly prompted GPT-4 with the following text: *"You are a participant in a color judgment task. You will be asked to describe an object's color in each question. Your objective is to generate an apt color code in HSL format to match the given object as well as possible. Remember, it's essential to answer the question with a single HSL code, even if the generated color or the object might seem unusual at times. Please limit your response to just the three values of the HSL code, for example, 'h, s, l'. What color matches the following object: strawberry."*

Direct Sampling with GPT-4

An alternative static method that circumvents the need for the agent to explicitly report a full color code is Direct Sampling. In this approach, the researcher randomly sample a valid HSL color code at each step, denoted as $x_i \sim p(x)$. $p(x)$ is a uni-

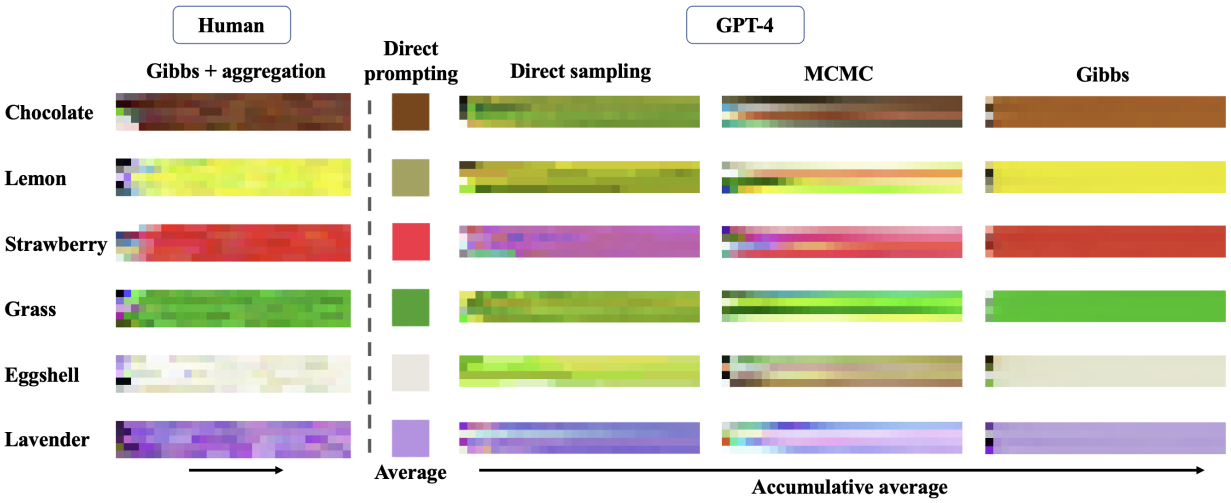

Figure 2: The evolution of the mean color representation across successive iterations. Each row within a color patch represents a single chain. Human data were adapted from Harrison et al. (2020).

form distribution over the entire color space. Subsequently, GPT-4 is tasked with determining whether the sampled color corresponds to the specified object, indicated by $\mathbb{1}_c$. That is, GPT-4 is only required to make a binary choice at each step, simplifying the task of directly reporting a color from scratch. Gradually, we approximate the conditional probability with the positive examples that were classified as the object:

$$x_{\mathbb{1}_c} \sim p_c(x) = p(x|c) \tag{1}$$

Using the same strawberry example, we implemented Direct Sampling with GPT-4 as follows: *"You are a participant in a color judgment task. You will see a question about whether a color (represented in HSL format) matches an object. Simply answer either 'yes' or 'no' based on your interpretation of the object's color in the question. Does the color [300, 97, 48] match the following object: strawberry?"*

Markov Chain Monte Carlo with GPT-4

MCMC with People (MCMCP) is a well-established adaptive method to elicit people's mental representations (Sanborn & Griffiths, 2007; Sanborn et al., 2010). We adapted the method to GPT-4. The key idea is to construct a Markov chain whose stationary distribution is $p(x|c)$, and thus the sequence of states generated by this chain can be interpreted as samples from the stationary distribution.

The Markov chain is initiated with an arbitrary value, x.[1] To progress the chain, a new candidate value, x', is generated by sampling from a proposal distribution $q(x'|x)$. Then the agent makes a decision on whether to accept x' based on its relative probability compared to x under the target distribution $p(x|c)$. This process hinges on two key assumptions: (i) the proposal distribution is symmetric, $q(x'|x) = q(x|x')$, and (ii) the probability of accepting the proposed value matches the Barker acceptance function (Barker, 1965):

$$A(x'|x,c) = \frac{p(x'|c)}{p(x'|c) + p(x|c)} \tag{2}$$

Under these conditions, the sequence of states generated by this Markov chain will converge to a stationary distribution that is consistent with $p(x|c)$.

Here we specify the proposal distribution for 90% of trials as a multivariate Gaussian distribution with a covariance matrix that is an identity matrix multiplied by 30: $q(x'|x) = N(x, 30\mathbf{I}_3)$. On the other 10% of trials, the proposed stimulus was sampled uniformly within the color space, facilitating large jumps in the stimulus space (Martin et al., 2012). In each iteration, GPT-4 was given a binary choice between two options, x and x'. The positions of these options were randomized.

The structure of the prompts implementing MCMC with GPT-4 is as follows: *"You are a participant in a color choice task. You will see a question with two color options in HSL format. Simply choose either Option A or Option B. Remember, it's essential to pick one color that better matches the object in the question, even if the choices might seem unusual at times. Please limit your response to just 'A' or 'B'. Which color better matches the following object: strawberry. Option A[0, 53, 12] or Option B[274, 81, 47]?"*

Gibbs Sampling with GPT-4

Gibbs Sampling with People (GSP) is a recent extension of the MCMCP method (Harrison et al., 2020). Gibbs Sampling involves cyclically sampling from each dimension based on the conditional probability $p(x_k|x_{-k}, c)$ (Geman & Geman, 1984). Analogously, in GSP, participants contribute to the update of coordinates. This is achieved by adjusting a slider corresponding to the current stimulus dimension, x_k, while

[1] As suggested by an anonymous reviewer, the chain can alternatively be initialized with the color values elicited from directly prompting GPT-4.

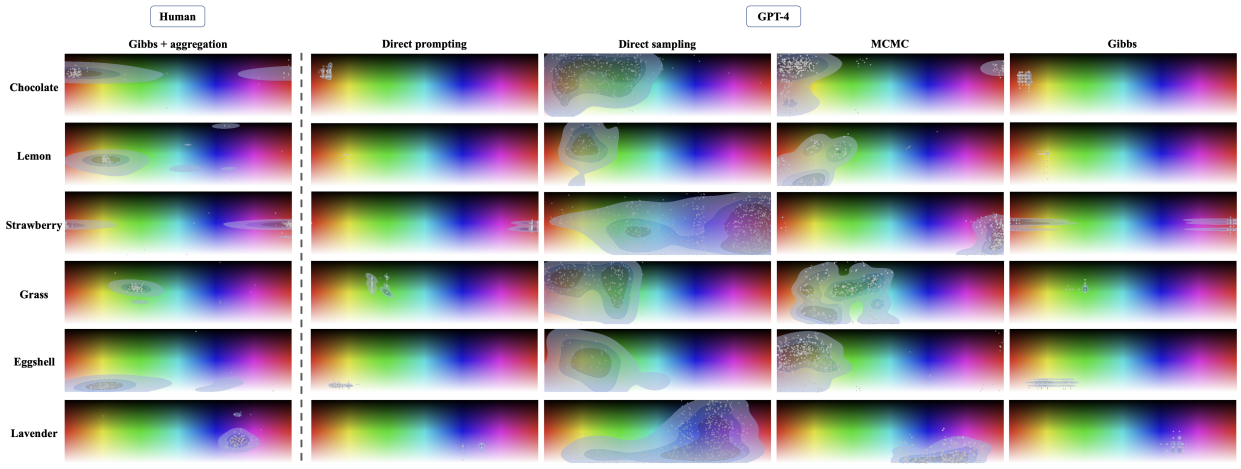

Figure 3: Samples in the color space produced by humans and those generated by GPT-4 using the four behavioral methods (displayed as columns). The overlaid contours are estimates derived from kernel density using a Gaussian kernel with a bandwidth of 1.

keeping the other dimensions, x_{-k}, constant (Harrison et al., 2020). The fundamental assumption in GSP is that participants select an option i for the k-th dimension following a specific probability distribution:

$$p(\text{choose } i) = p(x_k = i | x_{-k}, c) \tag{3}$$

If satisfied, this process will converge to a stationary distribution matching $p(x|c)$.

We provided specific prompts that implement Gibbs Sampling with GPT-4 as follows: *"You are a participant in a color judgment task. You will see an object and a color code in HSL format, however, one dimension of the given HSL color code is unknown. Your objective is to assign an apt integer to the unknown dimension to make the HSL color code match the given object as well as possible. Remember, it's essential to complete the color, even if the generated color might seem unusual at times. Please limit your response to just the value you'd like to assign to the unknown dimension. Adjust the unknown dimension of HSL color to match the following object as well as possible: strawberry. Color: [270, 50, 'unknown']"*

Recovering Color Representations from GPT-4

To recover mental representations for a low-dimensional perceptual domain, color, we employed a variety of behavioral methods to engage with GPT-4. Most of these methods have been used to elicit human representations (Harrison et al., 2020; Sanborn & Griffiths, 2007; Sanborn et al., 2010). Hypothesizing that GPT-4 can mimic human decision-making processes, we substituted human participants with GPT-4, enabling us to harvest samples directly from the LLM's color representation.

Stimuli

Our experimental design mirrors the human study conducted by Harrison et al. (2020), which used the HSL color values. Hue values range from 0 to 360, while both saturation and lightness extend from 0 to 100. We aimed to recover the representations of six specific objects within this color spectrum, enabling direct comparisons with corresponding human representations. These objects are 'Chocolate', 'Lemon', 'Strawberry', 'Grass', 'Eggshell', and 'Lavender', each offering a distinct color profile for analysis.

We adapted the human data from Harrison et al. (2020). Their research demonstrated that an aggregated GSP method is particularly effective in eliciting color representations from human participants. To briefly describe the process, each aggregated GSP chain was randomly initialized with an HSL color. Participants manipulated one color dimension at a time using a slider. For each iteration, judgments from five participants were aggregated and their mean value was used as the seed for the next iteration. Participants were only allowed to participate in a given chain only once to ensure within-chain trial independence (Harrison et al., 2020).

In the experiment conducted by Harrison et al. (2020), participants received the following instructions: "In each trial of this study you will be presented with a word and a color and your task will be to modify that color using a slider such that it best matches the target word. No prior expertise is required to complete this task, just answer what you intuitively think is the right color." Participants then completed up to 20 trials, responding to the prompt: "Adjust the slider to match the following word as well as possible: ⟨word⟩".

Procedure

GPT-4 was assigned to tasks of recovering color representations for the six objects tested in Harrison et al. (2020) through Direct Prompting, Direct Sampling, MCMC, and Gibbs Sampling. Detailed descriptions of the implementation for each method, along with corresponding visual illustrations, are presented in Figure 1. For all six objects, we configured GPT-4's temperature at 1.0. This setting makes

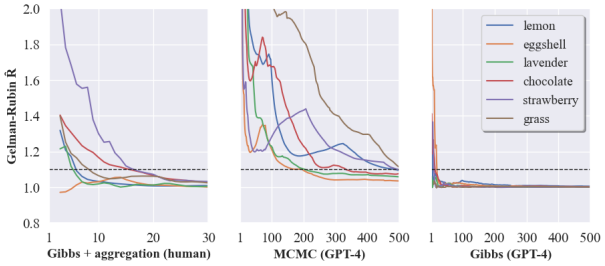

Figure 4: Cumulative $\hat{R}$ of Gibbs Sampling with People plus aggregation **(left)**, MCMC with GPT-4 **(middle)**, and Gibbs Sampling with GPT-4 **(right)**. Reaching the threshold of 1.1 suggests convergence of the Markov chain.

the model's outputs based on the model's learned probabilities. To ensure robustness and reliability in our findings, we ran all four behavioral methods for a total of 500 iterations. For methods that are based on sampling algorithms (Direct Sampling, MCMC, and Gibbs Sampling), we reinitialized the methods four times. This results in a cumulative total of 2000 samples, which were generated across four distinct chains. Figure 3 displays representative samples produced by each method, and Figure 2 depicts the evolution of the mean color representation across successive iterations.

Results

Convergence Diagnostic for Markov Chains

The convergence of the Markov chains in MCMC and Gibbs sampling can be assessed using the Gelman-Rubin diagnostic (Gelman & Rubin, 1992). This diagnostic calculates the ratio of within-chain variance to between-chain variance, denoted as $\hat{R}$, serving as an indicator of the extent of convergence. A threshold of $\hat{R} \leq 1.1$ is commonly adopted as a criterion for satisfactory convergence in Markov chains. We present cumulative $\hat{R}$ values in Figure 4. In alignment with previous empirical studies involving human participants (Harrison et al., 2020), the MCMC with GPT-4 exhibited the slowest rate of convergence. In contrast, the Gibbs sampling method demonstrated significantly quicker convergence, typically reaching stability within 10 iterations.

Representational Alignment of Humans and GPT-4

Upon verifying the convergence of the Markov chains, our analysis investigated the alignment of color representations between humans and GPT-4. First, the color space was discretized into a $18\times10\times10$ grid across the H-S-L dimensions, adopting a broader bin width to minimize the impact of minor color variations. Next, we estimated the probability density within each defined bin.

For the purpose of this comparison, we selected the human color representations derived from the GSP+aggregation condition reported in Harrison et al. (2020), as this method has demonstrated best performance in recovering human mental representations.

Table 1: Distributional and mode distances (indicated in parentheses) between GPT-4 and human representations.

	Direct Prompting	Direct Sampling	MCMC	Gibbs Sampling
Choc.	.99 (9.2)	.96 (4.6)	**.85** (**4.0**)	.95 (7.7)
Lemon	1.00 (13.5)	.99 (5.0)	**.95** (**3.2**)	1.00 (9.3)
Strwb.	**.80** (5.1)	.93 (5.5)	.93 (**4.7**)	.93 (9.1)
Grass	1.00 (6.3)	.99 (5.9)	**.98** (**5.5**)	.99 (6.5)
Eggsh.	.98 (3.7)	1.00 (4.9)	.96 (5.7)	**.87** (**3.6**)
Lav.	1.00 (5.4)	.87 (3.8)	**.81** (5.8)	.97 (**3.6**)

Note. Human representations for these objects were derived from data reported in Harrison et al. (2020). Bold numbers represent the best correspondence with human among the four behavioral methods (smaller is better). From top to bottom, the tested objects are 'Chocolate', 'Lemon', 'Strawberry', 'Grass', 'Eggshell', and 'Lavender'.

We developed two metrics to evaluate the representational alignment between humans and GPT-4. The first metric aims to quantify the overall agreement between the two distributions, $\hat{p}_{\text{human}}(x|c)$ and $\hat{p}_{\text{GPT-4}}(x|c)$. For this purpose, we employed the Hellinger distance:

$$H^2(\hat{p}_{\text{human}}, \hat{p}_{\text{GPT-4}}) = \frac{1}{2} \sum_{dx \in \mathcal{X}} \left(\sqrt{\hat{p}_{\text{human}}(dx)} - \sqrt{\hat{p}_{\text{GPT-4}}(dx)} \right)^2$$

The Hellinger distance is symmetric and bounded between 0 and 1, where 0 indicates identical distributions and 1 indicates maximum dissimilarity. This bounded range can be more intuitive and easier to interpret than unbounded measures. It is more robust when dealing with distributions that have zero probabilities.

While assessing the overall distributional alignment is crucial, it is also important to examine the most probable or representative mental state (i.e., $\arg\max_x p(x|c)$). Accordingly, we measured the Euclidean distance between the modes of the mental representations as derived from GPT-4 and humans. This second metric allows for a focused comparison of the most probable representational in both representations.

We calculated both metrics based on each of the 500-sample chains generated by Direct Sampling, MCMC, and Gibbs Sampling. Then these values were averaged over 4 repetitions of the sampling process. The resulting data are summarized in Table 1. Moreover, the progression of representational alignment throughout these iterations is depicted in Figure 5.

We found that among the methods employed, MCMC with GPT-4 exhibits notably superior performance in closely approximating the overall distributions and the modes of most human color representations (see Table 1 and Figure 5). Meanwhile, the other adaptive method, Gibbs Sampling with GPT-4, showed best performance in accurately representing eggshell and the mode of lavender. In contrast, both static

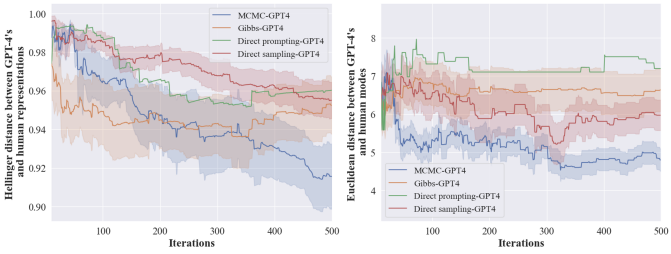

Figure 5: Comparing color representations in humans and GPT-4. **(left)** Hellinger distance between the color representations derived from GPT-4 and those from humans. **(right)** Euclidean distances between the modes of representations from GPT-4 and humans. In both measures, lower numerical values are indicative of a stronger correspondence. Shaded areas indicate ±SEM.

methods, including Direct Prompting and Direct Sampling with GPT-4, significantly lag behind in performance. Overall, the integration of GPT-4 with adaptive methods was more efficient than the integration with static methods in replicating human color representations.

Discussion

We developed and evaluated a novel class of adaptive methods with LLMs, using color as a case study. Our approach is grounded in two fundamental design principles: first, the incorporation of LLM outputs as integral components in sampling algorithms, and second, the dynamic modification of prompts based on previous responses from the LLMs. We tested integrating GPT-4 with various sampling algorithms, including Direct Sampling, MCMC, and Gibbs sampling. The objective was to recover human-like color representation. Our findings demonstrate that adaptive methods (MCMC and Gibbs sampling) significantly surpass the performance of static methods (Direct Prompting and Direct Sampling).

Towards Doing Bayesian Inference with LLMs

While we have focused on recovering conditional probabilities like $p(x|c)$ from GPT-4, the success of the methods we have presented here suggests that they could be adapted to sampling from other distributions. This capability is crucial in Bayesian inference, where many problems involve approximating the posterior probability of hypotheses h given data d, $p(h|d)$. This posterior probability is, in essence, a form of conditional probability distribution. Our methods could significantly broaden the scope for applying LLMs in Bayesian inference tasks. This can be achieved by constructing Markov chains with LLMs, which can be framed as simple as either choice-based or estimation-focused tasks.

The adaptive methods we employed are especially noteworthy. These methods dynamically alter prompts based on previous responses from LLMs, presenting a promising avenue for effectively conducting Bayesian inference. They not only simplify the task format for LLMs but also offer a more efficient means to navigate through the hypothesis space. While more advanced sampling algorithms such as Hamiltonian Monte Carlo (Betancourt, 2017) and the No-U-Turn Sampler (Hoffman & Gelman, 2014) could replace MCMC and Gibbs Sampling, the optimal choice of sampling algorithm should be determined by a combination of the target distribution's geometry and the response characteristics of the LLMs. This is because there are crucial assumptions that need to be satisfied (e.g., those outlined in Equations 2 and 3 for MCMC with GPT-4 and Gibbs sampling with GPT-4 respectively) to ensure that the sampling algorithms effectively converge to the correct target distribution.

It is important to note the distinction between our approach and other recent approaches for implementing Bayesian inference *using* LLMs (e.g., Wong et al., 2023; Zhang, Wong, et al., 2023). For example, Wong et al.'s (2023) proposal primarily leverages LLMs as translators, converting natural language inputs into probabilistic programming language statements. These statements are then subjected to Bayesian inference. This process essentially transforms LLMs into intermediaries, facilitating the translation from natural languages, which are inherently challenging for Bayesian inference, to symbolic representations that are more amenable to probabilistic programming languages, such as Church (Goodman et al., 2012). In contrast, our proposal advocates for a more direct usage of LLMs in Bayesian inference, positioning them as the primary computational mechanism rather than mere translators. Our findings suggest that constructing a Markov chain *with* LLMs for Bayesian inference might be more efficient compared to prompting LLMs directly.

Limitations and Future Directions

Our study underscores the potential of adaptive methods in recovering color representations from LLMs. Further investigation is required to assess the applicability of behavioral methods across different domains and to verify if LLMs employ these human-like representations in solving cognitive task. The efficacy of adaptive methods is heavily contingent upon the congruence between our presupposed assumptions regarding the nature of LLM responses and the actual response patterns exhibited by these models. Recent research also suggests that LLMs may exhibit a yes-response bias, which could complicate analyses of yes/no responses (Dentella et al., 2023).

In addition, there are various hyperparameters in the sampling algorithms and the LLMs, such as the proposal distributions and the temperature, that offer opportunities for fine-tuning. Tailoring these parameters to specific domains could potentially enhance the performance of these algorithms and minimize the total number of token requests for LLMs. Our research paves the way for future explorations into optimizing these parameters to achieve greater efficiency and accuracy.

Acknowledgments. This work and related results were made possible with the support of the NOMIS Foundation. H. Yan acknowledges the Chancellor's International Scholarship from the University of Warwick for additional support.

References

Alain, G., & Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*.

Barker, A. A. (1965). Monte Carlo Calculations of the Radial Distribution Functions for a Proton-Electron Plasma. *Australian Journal of Physics*, *18*(2), 119–134. https://doi.org/10.1071/ph650119

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

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.

Bubeck, S., Chandrasekaran, V., Eldan, R., Gehrke, J., Horvitz, E., Kamar, E., Lee, P., Lee, Y. T., Li, Y., Lundberg, S., et al. (2023). Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*.

Dentella, V., Günther, F., & Leivada, E. (2023). Systematic testing of three language models reveals low language accuracy, absence of response stability, and a yes-response bias. *Proceedings of the National Academy of Sciences*, *120*(51), e2309583120.

Gelman, A., & Rubin, D. B. (1992). Inference from Iterative Simulation Using Multiple Sequences. *Statistical Science*, *7*(4), 457–472. https://doi.org/10.1214/ss/1177011136

Geman, S., & Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 721–741.

Goodman, N., Mansinghka, V., Roy, D. M., Bonawitz, K., & Tenenbaum, J. B. (2012). Church: A language for generative models. *arXiv preprint arXiv:1206.3255*.

Harrison, P., Marjieh, R., Adolfi, F., van Rijn, P., Anglada-Tort, M., Tchernichovski, O., Larrouy-Maestri, P., & Jacoby, N. (2020). Gibbs sampling with people. *Advances in Neural Information Processing Systems*, *33*, 10659–10671.

Hoffman, M. D., & Gelman, A. (2014). The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *Journal Machine Learning Research*, *15*(1), 1593–1623.

Kay, P., Berlin, B., Maffi, L., Merrifield, W. R., & Cook, R. (2009). *The world color survey*. CSLI Publications Stanford, CA.

Kornblith, S., Norouzi, M., Lee, H., & Hinton, G. (2019). Similarity of neural network representations revisited. *International Conference on Machine Learning*, 3519–3529.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444.

Martin, J. B., Griffiths, T. L., & Sanborn, A. N. (2012). Testing the Efficiency of Markov Chain Monte Carlo With People Using Facial Affect Categories. *Cognitive Science*, *36*(1), 150–162.

Rathje, S., Mirea, D.-M., Sucholutsky, I., Marjieh, R., Robertson, C., & Van Bavel, J. J. (2023). GPT is an effective tool for multilingual psychological text analysis. *PsyArXiv*.

Sanborn, A. N., & Griffiths, T. L. (2007). Markov chain Monte Carlo with people. *Advances in Neural Information Processing Systems*, *20*.

Sanborn, A. N., Griffiths, T. L., & Shiffrin, R. M. (2010). Uncovering mental representations with Markov chain Monte Carlo. *Cognitive Psychology*, *60*(2), 63–106.

Shepard, R. N., & Arabie, P. (1979). Additive clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review*, *86*(2), 87–123.

Torgerson, W. S. (1958). *Theory and methods of scaling.* Wiley.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *30*.

Wong, L., Grand, G., Lew, A. K., Goodman, N. D., Mansinghka, V. K., Andreas, J., & Tenenbaum, J. B. (2023). From word models to world models: Translating from natural language to the probabilistic language of thought. *arXiv preprint arXiv:2306.12672*.

Zhang, C. E., Wong, L., Grand, G., & Tenenbaum, J. B. (2023). Grounded physical language understanding with probabilistic programs and simulated worlds. *Proceedings of the Annual Conference of the Cognitive Science Society*.

Zhang, L., McCoy, R. T., Sumers, T. R., Zhu, J.-Q., & Griffiths, T. L. (2023). Deep de finetti: Recovering topic distributions from large language models. *arXiv preprint arXiv:2312.14226*.