Human-like Few-Shot Learning with Distribution Calibration

in partial fulfillment of Honors in Computer Science

Thesis Student: Alyssa Loo\(^1\,\!^2\) | Advisor: Ellie Pavlick\(^1\,\!^2\) | Reader: Roman Feiman\(^2\,\!^3\)

\(^1\)Department of Computer Science \quad \(^2\)Program in Linguistics \quad \(^3\)Department of Cognitive, Linguistics and Psychological Sciences

1 Abstract

This paper investigates the behavior of large language models (LLMs) on the paradigm of rule learning experiments, previously used with humans. Demonstrating a close correlation with human learning trajectories on the rule learning task is suggestive evidence of a hypothesis generation and inference procedure similar to human subjects. Such arguments have previously been applied to Bayesian probabilistic models, and we now examine if large language models are equally as viable as models of human learning. We present three contributions: (1) a method of distribution calibration that enables human-like rule learning at the degree of fit as achieved by Bayesian probabilistic models; (2) results that show LLMs are sensitive to shared abstract components between different rules and; (3) evidence that LLMs may be implicitly Bayesian models, albeit with different hypotheses spaces from human subjects.

2 Motivation

Currently, human learning curves in the rule learning paradigm are best modeled using Bayesian models that implement a probabilistic language of thought (pLoT) [Carcassi and Szymanik 2023, Romano et al. 2018, Piantadosi et al. 2016]. Such models are endowed with an innate library of logical primitives, and apply probabilistic methods to iteratively compose hypotheses to account for the data they are given until the target rule is learned.

Neural network models learn very differently. They have no interpretable innate concepts, and hence require cognitively implausible amounts of data to train. Traditionally, these facts have made neural networks nonviable as competing models of any reasoning, much less to a Bayesian pLoT. Neural networks’ involvement in the modeling of reasoning has typically been as modules in explicit symbolic systems: proposal generators for symbolic Bayesian inference models [Ellis 2023, Qiu et al. 2024] or as translators between natural language and the symbolic language of a solver [Olausson et al. 2023].

However, after substantial pretraining, modern LLMs seem to acquire significant internal structure, and exhibit impressive sample efficiency via in-context learning. Recent work has proposed that in-context learning implements implicit Bayesian optimization [Akyürek et al. 2023], with pretrained LLMs performing Bayesian concept selection over concepts learned from pretraining data [Xie et al. 2022]. At the same time, LLMs also exhibit similar heuristic effects on reasoning as humans [Dasgupta et al. 2023]. These facts suggest that beyond only being components in symbolic models of reasoning,
pretrained LLMs may be compared on equal footing to Bayesian pLoT models as models of reasoning in themselves—possibly, as one that models both Bayesian inference procedure and heuristic influences on reasoning. The outcome of such a comparison would be significant, as it will open the door to a fundamentally different model for understanding how humans learn abstract, compositional rules.

We obtain LLMs’ learning curves on the same set of exemplar data from psycholinguistic rule learning experiments used to evaluate pLoT models [Piantadosi et al., 2016], and investigate to what extent LLMs’ probability outputs over exemplar labels can model the human posterior probability over exemplar labels. If LLMs’ probability outputs account for human responses given unfolding evidence across a wide range of rules and at least as well as the Bayesian models, then it is evidence that they should be considered to be as viable a model of human learning as Bayesian models. Such a finding would be significant since, unlike Bayesian models that require a predetermined grammar of logical primitives to generate hypotheses, LLMs are initialized with no ‘innate’ primitives nor any explicit inference algorithm. This would enable mechanistic approaches to understand the models’ process of learning the target rule, as a proxy for understanding how human subjects do so.

3 Method

Task We adapt the rule-learning experiment of [Piantadosi et al., 2016], henceforth referred to as PTG16. In the human experiment, each subject was instructed to learn the meaning of a nonce adjective that described some objects but not others. One to five objects were shown at a time as a set, and subjects were told to select only the objects described by the nonce adjective. As objects varied along the dimensions of size, color and shape, the meaning of the nonce adjective could pertain to a feature dimension (e.g. “An object is ‘wudsy’ if it is blue”), a combination of feature dimensions (e.g. “An object is ‘wudsy’ if it is blue or small, but not both.”), or how objects’ features related to other objects in the set (e.g. “An object is ‘wudsy’ if it is the only medium-sized object in its set”). Participants were given feedback on the correct labeling of objects after every set. We reframe the word learning task as simply a classification task to label whether objects are in the class described by the rule or not.

Data For each of the 112 distinct rules, we take two exemplar lists containing 25 object sets from the paper. As each object set contains one to five objects randomly sampled from the space of all possible objects, there is variation in the total number of objects in each exemplar list ($\mu=74.7$, $\sigma=6.40$). Each exemplar list is paired with the labeling data of, on average, 22 human subjects ($\mu=22.3$, $\sigma=1.44$) performing the rule learning task. Full details on exemplar list construction and human subject data collection can be found in PTG16. We hold out one exemplar list per rule to compute all reported metrics in the paper, while the other exemplar list is used as training data for model fitting where required. The split between held-out and training lists is the same as in PTG16.

We perform tuning on two settings: (a) ‘Tuned112’ where the training lists from all 112 rules are used for tuning, and (b) ‘Tuned92’ where 20 randomly chosen rules’ training lists are excluded from tuning, such that these 20 rules are entirely unseen by this model.

Model We want to select a model that has a reasonable baseline capacity to perform the task, such that potential failure to achieve a good fit to the human posterior on the task is not simply due to having a model too under-powered to perform the task at all. From a model selection experiment
detailed in Appendix A, we select Gemma-7B as the best model that performed within human subject distribution on the rule learning task across all rules.

**Finetuning** Borrowing methods from Knowledge Distillation [Wu et al., 2024], we use Forward KL-Divergence as loss to calibrate the model’s posterior distribution over labels to match that of humans. This can be interpreted as treating human subjects as the ‘teacher model’ from which we want to distill an inference procedure onto the LLM as a ‘student model’. From the model’s probability distribution over its vocabulary when predicting the label of an object, the probability mass of the True label and False label is collated by summing the probabilities of “True” and “False” tokens up to variation in leading spaces and capitalization. The loss function is defined as the KL-divergence between the model’s probability mass assignment for True and False to that of the proportion of human subjects that responded True and False respectively (Equation 1)

$$L = \sum_x D_{KL}(H_x(l)||M_x(l)) = \sum_x \sum_{l \in \{T,F\}} H_x(l) \ln \frac{H_x(l)}{M_x(l)}$$  

where $x$ are objects to label, $l$ is the true/false label, $M_x(l)$ is the probability mass assigned by the model to label $l$ for object $x$, and $H_x(l)$ is the proportion of human subjects that responded with label $l$ to object $x$. Finetuning was done on two GPUs using QLoRA, tuning about 2% of model parameters with the model quantized at 4-bit. Parameters and fine-tuning duration are detailed in Appendix B.

4 Results

![Figure 1a](image1.png)

(a) $R^2$ values across all objects from all rules.

![Figure 1b](image2.png)

(b) Each point plots the $R^2$ on objects within one rule. Rules are split by propositional and quantification rules. Box plots show median, IQR and upper/lower fences.

Figure 1: $R^2$ correlation values between model and human posterior probabilities for $P(\text{True})$ between different models.

Figure 1a shows the $R^2$ correlation between different models’ posterior $P(\text{True})$ with that of human subjects on held-out lists, with comparisons to the $R^2$ reported in Ellis [2023] (“Ellis23”) and the $R^2$ computed with the posteriors generated from PTG16’s Bayesian model using a FOL grammar. Distribution tuning improves the model’s fit to human responses greatly, to the extent that the tuned model explains 77.8% of human subject responses. This is closely competitive with the explicit Bayesian model from PTG16, which explains 79.5% of variance. We note that the pretrained model’s probabilities already explain a non-trivial amount of human subject response variance at 61.5%.

1We use the same data from PTG16’s model as used in Ellis 2023’s Bayesian Program Learning baseline.
We also show the distribution of $R^2$ on different rules in Figure 1b. The distribution of $R^2$ across different rules for the tuned model is highly similar to PTG16’s Bayesian model, especially for FOL rules. For propositional rules, the distribution of rules is centred at a slightly higher $R^2$ for PTG16’s Bayesian model than the tuned LLM.

The improvement in $R^2$ correlation between the human and model posterior on the held-out list, despite tuning only on the training list, supports that the LLM is generalizing the calibration of its posterior beyond specific object lists. Furthermore, comparing the Tuned92 and Tuned112 model shows that their $R^2$ scores are near-identical. Particularly, we compare the predictions of the Tuned112 model and the Tuned92 model specifically on rules that were held out from the Tuned92 model in Figure 2a. Model trajectories are near-identical even though these rules are entirely unseen for the Tuned92 model. This is likely because these 20 held-out rules share components with seen rules. For instance, tuning on “circle xor blue” and “not (circle xor blue)” may have enabled generalized calibration to the unseen rule “circle xor (not blue)”. This contrasts with Figure 2b, where we compare the Tuned112 model’s trajectory on two additional rules from Kellis23 that utilize majority/minority concepts, which are components not otherwise used in any of PTG16’s 112 rules. In this setting, the Tuned112 model behaves nearly identically to the pretrained model. These results together strongly suggest that the posterior calibration generalizes well to both unseen object lists and unseen rules, but is sensitive to abstract components shared between rules such as to determine that rules are in-distribution.

5 Discussion and Future Work

LLMs as Implicit Bayesian Learners with Broader Hypothesis Space. Previous work in Xie et al. [2022] characterized in-context learning as Bayesian concept selection, where examples in context allow LLMs to incrementally weight the prompt concept over all other concepts from pretraining. An interpretation for this setting is that while a human learner likely searches only over a hypothesis space pragmatically salient to the rule learning experiment, an LLM may initially search over a broader hypothesis space of tasks learned from pretraining that share similar input or output domains (e.g. sentiment classification, fact verification). Given more examples, however, the hypothesis space constrains to concepts salient to the rule learning experiment. This hypothesis makes two predictions, that (1) LLMs should correlate more with human subjects after initial examples and (2) LLMs should correlate with the Bayesian model more after initial examples.

Both predictions are correct as shown in Figure 3, where we measure $R^2$ correlation coefficients with both human subjects and PTG16’s Bayesian model with excluding up to the first 35 objects of all object lists. Excluding just the initial 1 to 3 objects of each rule from the $R^2$ calculation, where each rule has an average of 75 objects, causes the tuned LLM to easily match PTG16’s explicit Bayesian model in $R^2$ correlation to human subjects. Excluding initial objects also improves the fit with PTG16’s Bayesian model, for both the tuned and pretrained model. Ultimately, however, a ceiling effect is observed with a non-trivial 15% unexplained variance between the LLMs and human subjects and between the LLM and Bayesian model. For the difference between LLMs and the Bayesian model, it may be because LLMs search over rules of a different ‘grammar’ than the FOL-grammar of PTG16’s Bayesian model. For that between LLMs and humans, it may also be a case of having different priors. It may also be that, while the initial motivation proposed that LLMs’ sensitivity to heuristics may enable a greater fit to human responses, it may ultimately be operating with heuristics.
(a) Accuracy trajectories on the held-out list on the top three, median three and bottom three rules from the 20 rules held-out from Tuned92, ranked by $R^2$ (Tuned112).

(b) Accuracy trajectories for two additional rules and human data from Kellis23 that use majority/minority judgments, which is not used in any of the 112 rules from PTG16.

Figure 2: Learning trajectories on different rules, with a natural language description of the rule on top of each graph. For human subjects, $y$-values are the proportion correct for each example; for models, $y$-values are the probability mass assigned to the correct label. $x$-values are the index of objects in the object list. The $R^2$ values reported are the correlation between $P(True)$ for human subjects and each model. Dotted lines are the chance baseline for the rule, calculated as the accuracy from guessing ‘True’ at the rate of the proportion of ‘True’s in the object list.

that are different from that of human subjects.

**Role of Distribution Calibration**  Distribution calibration enabled the tuned model to fit human responses on some rules where the pretrained model oscillated around the chance baseline. It is unclear if calibration is teaching the model new rule concepts, or merely enabling the up-weight of latent concepts that the model already acquired in pretraining. The latter would predict that distribution calibration would yield no improvement in fit from the pretrained model if the concept was never acquired by the pretrained model from training data. This would potentially be an insightful probe into what concepts are acquired from pretraining. We leave investigating distribution calibration on lower capacity models, which are more likely to have not acquired complex concepts from their pretraining, to future work.

**‘In-Distribution’ Rules**  The contrasting results on different sets of unseen rules pose a natural question about what rules are considered ‘in-distribution’ given a tuning set. We leave it to future
work to experiment with partitioning the set of tuned rules to investigate how rules with different components may interact with each other.

6 Conclusion

We propose a method that succeeds in calibrating LLMs’ probability distributions to match that of humans’ posterior distributions on the rule learning task, such that they are on par with Bayesian probabilistic models of the task. Our results provide suggestive evidence of a sensitivity to shared abstract structure, and also of LLMs implementing a Bayesian inference procedure in in-context learning. Future work will investigate how mechanistically studying these tuned models may yield new insights about what concepts are able to be acquired from data, or how human subjects perform induction in the rule learning task.

References


Kevin Ellis. Human-like Few-Shot Learning via Bayesian Reasoning over Natural Language, September 2023.

Theo X. Olausson, Alex Gu, Benjamin Lipkin, Cedegao E. Zhang, Armando Solar-Lezama, Joshua B. Tenenbaum, and Roger Levy. LINC: A Neurosymbolic Approach for Logical Reasoning by


Taiqiang Wu, Chaofan Tao, Jiahao Wang, Zhe Zhao, and Ngai Wong. Rethinking Kullback-Leibler Divergence in Knowledge Distillation for Large Language Models, April 2024.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An Explanation of In-context Learning as Implicit Bayesian Inference, July 2022.
Appendices

A Model Selection Experiment

Method We test a range of recent models across different sizes and in both chat and completion formats. Details about how prompts were written for each format are given in Appendix C. We measure for each rule the proportion of correct labels across all objects (“overall”). Furthermore, to account for that early objects are trivially less accurately labeled than later objects due to the limited starting information about the rule, we also additionally measure the proportion correct on the last quarter of objects (“last quarter”). A quarter contains 18 objects on average ($\mu=18.6$, $\sigma=1.6$).

Metrics For completion models, objects labeled with completions that were not variants of “True” or “False” subject to leading/trailing spaces and alternate capitalization were discarded from the labeling data for these models. For chat models, labels could not be extracted for some objects because the model reiterated entirely different objects for labeling, or abstained from labeling. These objects were also discarded from the labeling data for these models.

In computing overall accuracy for each model, we find the mean accuracy only over all objects that the model gave a valid label. In computing the last quarter of object accuracy, we take the last quarter of the original list and, for each model, exclude any objects given invalid labels by that model. This is because the information the model is given for the labels of previous objects is irrespective of whether it returned valid labels for the previous objects.

Results Results are reported in Table 1 first for all rules, then for each subset of rules. On propositional rules, the best models GPT-4, Mixtral 8x7b Instruct, and Gemma (7B) exceed human performance on mean accuracy averaged across all rules.

Mean human accuracy and its lower standard deviation bound is a generous baseline, as the human accuracy distribution is heavily left-tailed for many rules. We hence examine the top models on a rule-by-rule basis in Figure 4, comparing models’ last quarter accuracy to the median last quarter accuracy of human subject data for each rule. We use the median to compare against the typical performance of a human subject for the rule, and percentiles to characterize how poor model performance is relative to the human subject distribution.

For propositional rules, GPT-4 matches the human median at the ceiling on almost all of them and remains within second quartile of subjects otherwise. Gemma (7B) and Mixtral 8x7b Instruct mostly achieve this level of parity, but are within the bottom 10-25% of human subjects for a minority of propositional rules. On quantified rules these three models remain within the top 75% of human subjects for most rules. While GPT4 and Mixtral 8x7b Instruct have three to four rules for which they are within the bottom 1-5% of human subjects, Gemma-7b even in the worst-case is within the bottom 10% of subjects. We conclude that there is sufficient evidence that the top models succeed on the rule learning task, as they are close to the human median on a majority of rules and are at worst within the human learner distribution for almost all rules.

---

2 The reported average is the average of all relevant rules’ mean accuracy. SD for each reported average is propagated error, i.e. the sum of constituent rules’ accuracy variance divided by the number of constituent rules.
<table>
<thead>
<tr>
<th>Model</th>
<th>All Rules</th>
<th>Propositional Rules</th>
<th>FOL Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Overall</td>
<td>Last Quarter</td>
<td>Overall</td>
</tr>
<tr>
<td><strong>Chat Models</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GPT4</td>
<td>0.764</td>
<td>0.813</td>
<td><strong>0.908</strong></td>
</tr>
<tr>
<td>GPT 3.5</td>
<td>0.685</td>
<td>0.746</td>
<td>0.761</td>
</tr>
<tr>
<td>Llama2-70b-Chat</td>
<td>0.681</td>
<td>0.716</td>
<td>0.787</td>
</tr>
<tr>
<td>Mixtral 8x7B Instruct</td>
<td>0.769</td>
<td>0.815</td>
<td>0.875</td>
</tr>
<tr>
<td><strong>Completion Models</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Llama2-70b</td>
<td>0.734</td>
<td>0.794</td>
<td>0.799</td>
</tr>
<tr>
<td>Llama2-7b</td>
<td>0.760</td>
<td>0.799</td>
<td>0.820</td>
</tr>
<tr>
<td>Gemma-7b</td>
<td><strong>0.795</strong></td>
<td><strong>0.840</strong></td>
<td>0.886</td>
</tr>
<tr>
<td>Gemma-2b</td>
<td>0.751</td>
<td>0.772</td>
<td>0.771</td>
</tr>
<tr>
<td>GPT2-XL</td>
<td>0.654</td>
<td>0.684</td>
<td>0.663</td>
</tr>
<tr>
<td>GPT2</td>
<td>0.622</td>
<td>0.707</td>
<td>0.634</td>
</tr>
<tr>
<td>Human</td>
<td>0.774±</td>
<td>0.835±</td>
<td>0.858±</td>
</tr>
<tr>
<td></td>
<td>0.0109</td>
<td>0.0148</td>
<td>0.106</td>
</tr>
</tbody>
</table>

Table 1: Results for all rules and for the two subsets of rules requiring only propositional logic and rules requiring first-order-logic (FOL) over the object set. Model averages above the lower SD bound of human averages for the column are underlined. The highest scores in each column are bolded.

Figure 4: Each x-value is a rule with two y-values plotted: the median last quarter accuracy of human subjects (gray) and of the model in the given row (blue). The inter-graph fill represents the difference between the last quarter accuracies over an interval of rules: blue indicates greater model than human accuracy in that interval and gray vice versa. Rules are split into rule type by column. Within each row, rules are sorted in descending order of the accuracy delta for that model. Shaded red regions show the intervals of rules where model accuracy is within the 25\textsuperscript{th}, 20\textsuperscript{th}, 10\textsuperscript{th}, 5\textsuperscript{th} and 1\textsuperscript{st} percentile of human last quarter accuracies respectively.
B Finetuning Details

Finetuning was done on two Quadro RTX 6000 GPUs. We loaded pretrained Gemma (7B) at 4-bit precision using NF4 quantization with double quantization, with a 16-bit compute datatype using bitsandbytes. We apply rank-stabilized Q-LoRA on all linear modules with $r = 64$, $\alpha = 32$, dropout at 0.05, and without updating biases, using the PeftModel class. Training was done with the Hugging Face Trainer class, using paged 8-bit Adam as the optimizer with 30 warmup steps, a learning rate of $10^{-5}$, batch size of 1, gradient accumulation over every 4 steps and gradient checkpointing.

Tuned112 model was fine-tuned for 1000 steps (35.7 epochs) taking 2 hours. Tuned92 was also fine-tuned for 1000 steps (51. In the dataset ablation without labels, the model was fine-tuned for 1400 steps (50 epochs) taking 2.6 hours. In the fine-tuning ablation where loss from all tokens were propagated, the model was fine-tuned for 1000 steps (35.7 epochs) taking 4.5 hours.

C Prompts

C.1 Completion Models

# Instructions
Learn the secret rule to label the objects in groups correctly.
The rule may consider the color, size and shape of objects, and may also consider the other objects in each group.
If an object in a group follows the rule, it should be labeled 'True'. Otherwise it should be labeled "False".

# Quiz
## Group 1
small blue rectangle
medium green triangle
large blue circle

## Group 1 Answer Key
small blue rectangle -> True
medium green triangle -> True
large blue circle -> False

Listing 1: Completions query example. Model was given Group 1 and Group 2 and their labels as example sets, and completions for Group 3 onwards were extracted object-by-object with teacher-forcing on previous examples.

C.2 Chat Models

{"role": 'system': 'content': "Learn the secret rule to label the objects in groups correctly. The rule may consider the color, size and shape of objects, and may also consider the other objects in each group. If an object in a group follows the rule, it should be labeled 'True'. Otherwise it should be labeled 'False'. Be concise and always follow the arrow '->' with a object's label."}

{"role": 'user', 'content': "Label the following objects in Group 1:
- small blue rectangle"}
The labels for the objects in Group 1 are:
- small blue rectangle → True
- medium green triangle → False
- large blue circle → True

Listing 2: Chat query example. Model was given Group 1 and Group 2 and their labels as example sets, and completions for Group 3 onwards were extracted set-by-set with teacher-forcing on previous sets.